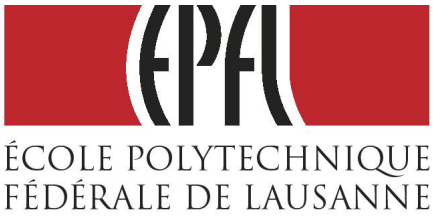


Master Thesis

Exploration and comparison of reduced order modelling techniques for parametrized system

Fabrizio Gelsomino

Supervised by
Prof. Alfio Quarteroni
Dr. Gianluigi Rozza

CMCS - Chaire de Modélisation et Calcul scientifique - Modelling and Scientific Computing Chair

January 15, 2010

Contents

Acknowledgements

First of all, I would like to thank my thesis supervisors Prof. Alfio Quarteroni and Dr. Gianluigi Rozza who approved this project and helped me to find a good topic. I especially thank Gianluigi Rozza to have followed me during my project, for his motivation, his encouragements, his patience and particularly for his knowledge and his availability (even when he slept he thought about Graetz problem).

The staff of CMCS for its kindness and its good mood has also all my gratitude. I would like to thank the soccer team which by sun and rain was always present for the very competed monday's game. I thank also Paolo Crosetto for his excellent (numerous) coffees, Anwar Koshakji for his delicious apple juice and Alberto Trezzini for our useful and long discussions about reduced basis and Comsol.

Infine ringrazio i miei genitori Antonietta e Salvatore per avermi sostenuto e incoraggiato durante tutti i miei studi. Grazie ai miei fratelli Marco e Yvan per tutti i grandi momenti passati insieme. Infine, ringrazio la mia ragazza Fanny per avermi sostenuto (e sopportato) durante i miei studi all'EPFL.

Introduction

Computing the solution of partial differential equations (PDEs) is very expensive. For realistic simulations, we need thousands degrees of freedom (DoF) to obtain good approximations of the solution. For example for the analysis and the optimization of an engineering system, we have to compute several solutions of the PDEs. We may be interested in outputs depending on the state solution : i.e. maximum or average temperature, heat transfer rates, flow rates, etc. The introduction of some parameters leads to *input-outputs relationships*. The input-parameter may represent boundary conditions and sources, geometric configurations or physical properties. So, classical discretization methods like finite element, finite volume or spectral methods are not the most appropriate. Consequently, we have to develop techniques that reduce the cost and time of the computations. These techniques are called *reduced order methods (ROM)*. Reduced basis (RB) method is one of them and is indicated to evaluate very quickly the outputs mentioned earlier. Moreover, the goal of the reduced basis method is to reduce the complexity of a system without a loss of information or accuracy of the results. This method does not replace one existing method, like the finite element (FE) method, but it is a collaboration with it. The idea is to start with a FE basis of dimension $\mathcal{N}$ and then we construct a RB space which dimension is $N << \mathcal{N}$. The RB method is particularly well suited in two contexts: the *real-time context* and the *many-query context*. The real-time context arises in applications like control engineering and in parameter estimations. The many-query context is involved in multiscale, multiphysics and optimization problems. The RB method is a response of these contexts.
In this thesis, we essentially treat *affine coercive elliptic problems* [20]. This class of problems has significant advantages for the reduced basis. The affine

parameter-dependence hypothesis enables an efficient *Offline-Online* computation. The reduced basis method reduces notably the Online cost. However, to reduce the Online cost, we have to provide much effort and preparation in the Offline step. So, this method is optimized for problems that allow rapid Online computation at the cost of a bigger Offline effort. In other words, the Offline part depends on $\mathcal{N}$ while the Online part depends only on N.

In this thesis, we will use the RB method to solve 3-dimensional problems: *the thermal fin problem* and *the time-dependent Graetz problem*. In the past, 2-dimensional problems have been successfully solved using the *rbMIT software package* [1]. This software has been developed by Anthony Patera's group at MIT. The rbMIT software is a library developed in MATLAB. It contains a symbolic part that compute the *affine decomposition* and the related matrices and another part that contains all the methodology of the RB method such as SCM and Greedies (see Chapter 1). So the user needs only to provide the equation to be solved, the parameter domain and the kind of output. For more details about the use of the rbMIT software, see [7].

However, in the 3D case, we can not use the provided symbolic solver and then we have to find other solutions. Here, we propose to link the rbMIT software with COMSOL [2]. COMSOL is a powerful environment for solving and modelling engineering problems based on PDEs. It permits to create your own geometry and generate meshes. The most important capability is that it can be interfaced with MATLAB. So the idea is to construct the geometry, the mesh and the different matrices with COMSOL and then use the rbMIT software to construct our RB space. Here, we will focus on problems with simple geometries in order to find "easily" the affine decomposition. In this thesis, we are more interested on results about convergence and computational time for the method applied to 3D problems in order to show that the RB method can be used for more complicated problems for example in the numerical simulation of the cardiovascular system.

In the first chapter, we will review the main points of the RB method for affine coercive elliptic problems. First of all, we will give some generalities about parametric bilinear forms. Then, we present the RB method and some sample strategy. We will treat the geometric variations and the constructions of affine mappings and then, we introduce the *a posteriori error bound* and some extensions for non-compliant problems.

The second chapter is the continuation of the first, where we extend our attention to *parabolic* problems. We will introduce the main changes arising with the time-dependency.

In the third chapter, we present the 3D thermal fin problem (see [3], [5] and [10]) which is a heat problem with parametrized geometry and physics, already solved in the 2D case (see [1]). Consequently, we could compare our results. In a first time, we will give a description of the problem and its mathematical description. Then, we will indicate how we construct the affine mappings and give the affine decomposition. At the end, we present some convergence results for the reduced basis method and a comparison with several ROM. The behaviour of the parametrized output is discussed too.

We next solve a time-dependent problem in chapter four; the Graetz problem (see [4], [5] and [22]). This problem is a time-dependent problem dealing with heat conduction and forced heat convection in a duct. In the same way we do for the thermal fin, we will introduce the mathematical description of the problem and some numerical results solving system with reduced basis method.

Historical on Reduced Basis

We present here a brief historical on reduced basis. For more details and references read [21] and [13].
Almroth, Stern and Brogan [6] are the first who have introduced the reduced basis method for one parameter problem. Then the method was developed by Noor [14] and extended to multi parameter problem. The need for more effectivity, many-query design evaluation and from the need for more efficiency parameter methods for nonlinear problems depending on a parameter is the reason of development of this method. This method was next extended to general system and a variety of different reduced basis approximation spaces. Then, the method has been used in fluid dynamics and for incompressible Navier Stokes equations [19].
Current research are dedicated to the development of a posteriori error estimation procedure and to the development of effective sampling strategies for many parameters spaces, in order to improve convergence and computational efficiency [20].

Chapter 1

Overview of the Reduced Basis Method: Elliptic Problems

In this chapter, we will introduce the RB method for compliant, coercive affine elliptic problems. In the first part of this chapter, we will briefly recall some generalities about parametric bilinear form and about coercivity, then the methodology is presented. The last part deals with the construction of a posteriori error bounds. Finally, we will extend the theory to the non-compliant case. The theory we present here can be found in more details principally in [20] and [15].

1.1 Parametric bilinear forms

In this section, we recall some definitions and properties about parametric bilinear forms and coercivity constants. For more details, the reader can refers to [15].
Let Z_1 and Z_2 be two vector spaces over $\mathbb{R}$. Let $\mathcal{D} \subset \mathbb{R}^P$ be a closed bounded parameter domain. A typical parameter vector in $\mathcal{D}$ shall be denoted $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_P)$. We assume that $\mathcal{D}$ is suitably regular, for example with a Lipschitz continuous boundary.

Definition 1.1. A form $b : Z_1 \times Z_2 \times \mathcal{D} \longrightarrow \mathbb{R}$ is a *parametric bilinear form*, if for all $\boldsymbol{\mu} \in \mathcal{D}$ and for any $\alpha \in \mathbb{R}$, w, $v \in Z_1$, $z \in Z_2$

$$b(\alpha w + v, z, \boldsymbol{\mu}) = \alpha b(w, z, \boldsymbol{\mu}) + b(v, z, \boldsymbol{\mu}),$$

and for any $\alpha \in \mathbb{R}$, $z \in Z_1$, w, $v \in Z_2$

$$b(z, \alpha w + v, \boldsymbol{\mu}) = \alpha b(z, w, \boldsymbol{\mu}) + b(z, v, \boldsymbol{\mu}).$$

In the next, we consider $Z_1 = Z_2 = Z$ and that $\dim(Z) < \infty$.

Definition 1.2. A parametric bilinear form $b : Z \times Z \times \mathcal{D} \longrightarrow \mathbb{R}$ is *symmetric*, if for $\boldsymbol{\mu} \in \mathcal{D}$ and for any $w, z \in Z$, $b(w, z, \boldsymbol{\mu}) = b(z, w, \boldsymbol{\mu})$.
A parametric bilinear form $b : Z \times Z \times \mathcal{D} \longrightarrow \mathbb{R}$ is *skew-symmetric*, if for $\boldsymbol{\mu} \in \mathcal{D}$ and for any $w, z \in Z$, $b(w, z, \boldsymbol{\mu}) = -b(z, w, \boldsymbol{\mu})$.

Definition 1.3. Let $b : Z \times Z \times \mathcal{D} \longrightarrow \mathbb{R}$ be a parametric bilinear form. We define:

- *the symmetric* part of b as

$$b_S(w, z, \boldsymbol{\mu}) = \frac{1}{2} \left(b(w, v, \boldsymbol{\mu}) + b(v, w, \boldsymbol{\mu}) \right), \forall v, w \in Z, \boldsymbol{\mu} \in \mathcal{D},$$

- *the skew-symmetric* part of b as

$$b_{SS}(w, z, \boldsymbol{\mu}) = \frac{1}{2} \left(b(w, v, \boldsymbol{\mu}) - b(v, w, \boldsymbol{\mu}) \right), \forall v, w \in Z, \boldsymbol{\mu} \in \mathcal{D}.$$

Definition 1.4. A parametric bilinear form $b : Z \times Z \times \mathcal{D} \longrightarrow \mathbb{R}$ is

- *positive definite*, if for $\boldsymbol{\mu} \in \mathcal{D}$ and for any $v \in Z$, $b(v, v, \boldsymbol{\mu}) \geq 0$ with equality only for $v = 0$.
- *positive semidefinite*, if for $\boldsymbol{\mu} \in \mathcal{D}$ and for any $v \in Z$ $b(v, v, \boldsymbol{\mu}) \geq 0$.

Definition 1.5. A parametric bilinear form $b : Z \times Z \times \mathcal{D} \longrightarrow \mathbb{R}$ is *coercive* over Z if

$$\alpha(\boldsymbol{\mu}) = \inf_{w \in Z} \frac{b(w, w, \boldsymbol{\mu})}{\|w\|_Z^2} \tag{1.1}$$

is positive for all $\boldsymbol{\mu} \in \mathcal{D}$.
We can then define $\alpha_0 = \min_{\boldsymbol{\mu} \in \mathcal{D}} \alpha(\boldsymbol{\mu})$.

Remark: We note that we can replace b in (1.1) by the symmetric part b_S.

Definition 1.6. We say that a parametric bilinear form $b : Z \times Z \times \mathcal{D} \longrightarrow \mathbb{R}$ is *continuous* over Z if

$$\gamma(\boldsymbol{\mu}) = \sup_{w \in Z} \sup_{v \in Z} \frac{b(w, v, \boldsymbol{\mu})}{\|w\|_Z \|v\|_Z} \tag{1.2}$$

is finite for all $\boldsymbol{\mu} \in \mathcal{D}$.
We can the define $\gamma_0 = \max_{\boldsymbol{\mu} \in \mathcal{D}} \gamma(\boldsymbol{\mu}) (< \infty)$.

1.1.1 Coercivity eigenproblem

As mentioned in the previous remark, we can rewrite (1.1) as

$$\alpha(\boldsymbol{\mu}) = \inf_{w \in Z} \frac{b_S(w, w, \boldsymbol{\mu})}{\|w\|_Z^2}. \tag{1.3}$$

It follows that $\alpha(\boldsymbol{\mu})$ can be expressed as a minimum eigenvalue.

Then, we introduce the coercivity symmetric (generalized) eigenproblem associated with the parametric bilinear form $b : Z \times Z \times \mathcal{D} \longrightarrow \mathbb{R}$:

Given $\boldsymbol{\mu} \in \mathcal{D}$, find the couple $(\xi_i(\boldsymbol{\mu}), \lambda_i(\boldsymbol{\mu})) \in Z \times \mathbb{R}$, $i = 1, \ldots, \dim(Z)$, such that

$$b_S(\xi_i, v, \boldsymbol{\mu}) = \lambda_i \left(\xi_i(\boldsymbol{\mu}), v\right)_Z, \quad \forall v \in Z \tag{1.4}$$

and

$$\|\xi_i(\boldsymbol{\mu})\| = 1. \tag{1.5}$$

We order the eigenvalues in ascending order such that $\lambda_1(\boldsymbol{\mu}) \leq \ldots \leq \lambda_{\dim(Z)}(\boldsymbol{\mu})$. This procedure will be recalled when dealing with POD, (Proper Orthogonal Decomposition). It follows from (1.3) and (1.4) that if a is coercive $\alpha(\boldsymbol{\mu}) = \lambda_1 > 0$.

1.1.2 Affine parameter dependence and parametric coercivity

A crucial ingredient for a real-time evaluation of outputs is the affine decomposition of the bilinear form $b : Z_1 \times Z_2 \times \mathcal{D} \longrightarrow \mathbb{R}$.

Definition 1.7. A parametric bilinear form b is *affine* in the parameter if

$$b(w, v, \boldsymbol{\mu}) = \sum_{q=1}^{Q_b} \theta_b^q b^q(w, v) \quad \forall w \in Z_1, \forall v \in Z_2, \tag{1.6}$$

for some finite Q_b and some parameter-dependent function $\theta_b^q : \mathcal{D} \longrightarrow \mathbb{R}$, $1 \leq q \leq Q_b$, and where $b^q(w, v) : Z_1 \times Z_2 \longrightarrow \mathbb{R}$, $1 \leq q \leq Q_b$ are parameter-independent continuous bilinear forms.

The decomposition (1.6) is non-unique, so we can find a minimal value for Q_b.

Definition 1.8. An affine parametric bilinear form $b : z \times Z \times \mathcal{D} \longrightarrow \mathbb{R}$ is *parametrically coercive* if $c = b_S$ admits an affine development such as:

$$c(w, v, \boldsymbol{\mu}) = \sum_{q=1}^{Q_c} \theta_c^q(\boldsymbol{\mu}) c^q(w, v) \quad \forall w, v \in Z, \tag{1.7}$$

that satisfies

$$\theta_c^q(\boldsymbol{\mu}) > 0 \quad \forall \boldsymbol{\mu} \in \mathcal{D},\ 1 \leq q \leq Q_c, \tag{1.8}$$

and

$$c^q(v, v) \geq 0 \quad \forall v \in Z,\ 1 \leq q \leq Q_c. \tag{1.9}$$

1.2 Elliptic coercive parametric PDEs: compliant case

In this section, we focus on elliptic problems. As mentioned in the *Introduction*, we may be interested in the evaluation of output that depends on the state equation which is solution of an elliptic PDE. The goal of this section is to introduce the *reduced basis method* in order to solve this kind of problem. At the beginning, we will give the *exact* formulation (in weak form) of the problem and a finite element discretization. Then, we will be able to introduce the reduced basis method, detailing what we do in the Offline and the Online part. The main references for this part are [20], [15] and [23].

1.2.1 Problem formulation

Let $\Omega \in \mathbb{R}^d$, $d = 1, 2, 3$ a suitable physical domain with Lipschitz continuous boundary $\partial\Omega$. Let $\mathcal{D} \subset \mathbb{R}^P$ the parameter domain. Let $X^e(\Omega)$ be a Hilbert function space such that $\left(H_0^1(\Omega)\right)^\nu \subset X^e(\Omega) \subset \left(H^1(\Omega)\right)^\nu$ where $\nu = 1$ (respectively $\nu = d$) for a scalar (respectively, vector) field. Here,

$$H^1(\Omega) = \left\{ v \in L^2(\Omega) \,|\, \nabla v \in \left(L^2(\Omega)\right)^d \right\},$$

$$H_0^1(\Omega) = \left\{ v \in H^1(\Omega) \,|\, v_{\partial\Omega} = 0 \right\},$$

and

$$L^2(\Omega) = \left\{ v \,\text{measurable} \,|\, \int_{\partial\Omega} v^2 \,\text{is finite} \right\}.$$

We associate to X^e an inner product and a norm, equivalent to the H^1 norm, denoted by $(\cdot,\cdot)_{X^e}$ and $\|\cdot\|_{X^e}$ respectively. The definitions of these quantities will be defined below.

Let $a : X^e \times X^e \times \mathcal{D} \longrightarrow \mathbb{R}$ be a continuous coercive parametric bilinear form. Let f be a continuous parametric linear functional.
We consider the following problem: Given $\boldsymbol{\mu} \in \mathbb{R}^P$ evaluate

$$\begin{aligned} &s^e(\boldsymbol{\mu}) = \ell(u^e(\boldsymbol{\mu}); \boldsymbol{\mu}), \\ &\text{where } u^e(\boldsymbol{\mu}) \in X^e(\Omega) \text{ satisfies} \\ &a\left(u^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right) = f(v; \boldsymbol{\mu}), \quad \forall v \in X^e. \end{aligned} \tag{1.10}$$

The superscript ^{e} refers to *exact*. Here $\boldsymbol{\mu}$ is the input parameter, s^e is the scalar output, ℓ is the linear output functional and $u(\boldsymbol{\mu})$ is the field variable. Under the hypothesis on the forms a and f the problem(1.10) has a unique solution.
Moreover, we assume that f and ℓ are bounded over X^e and that we have the following affine developments

$$\ell(v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_l} \theta_l^q(\boldsymbol{\mu}) \ell^q(v) \quad \forall v \in X^e, \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{1.11}$$

$$f(v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \theta_f^q(\boldsymbol{\mu}) f^q(v) \quad \forall v \in X^e, \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{1.12}$$

$$a(w, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \theta_a^q(\boldsymbol{\mu}) a^q(w, v) \quad \forall w, v \in X^e, \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{1.13}$$

for finite and relatively small Q_l, Q_f, Q_a. We consider that the θ_k^q for $1 \leq q \leq Q_k$, $k = l, f, a$ are simple algebraic expressions that can be readily evaluated in $\mathcal{O}(1)$ operations. Till the end of this section, we will consider *compliant* problems, i.e. we assume that
(i) a is symmetric
(ii) $\ell = f$

We will extend to the *non-compliant case* in section 1.6 with the introduction of a *dual problem* .

Then, the problem (1.10) can be reformulated as: Given $\boldsymbol{\mu} \in \mathbb{R}^P$ evaluate

$$\begin{aligned} & s^e(\boldsymbol{\mu}) = f(u^e(\boldsymbol{\mu}); \boldsymbol{\mu}) \\ & \text{where } u^e(\boldsymbol{\mu}) \in X^e(\Omega) \text{ satisfies} \\ & a\left(u^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right) = f(v; \boldsymbol{\mu}), \quad \forall v \in X^e. \end{aligned} \tag{1.14}$$

1.2.2 Truth approximation

In this section, we built the *truth* approximation on which we will construct our reduced basis (RB) approximation. Moreover, we will measure the error in the reduced basis approximation relative to this truth approximation, see Section 1.5.

We now introduce the space $X^{\mathcal{N}} \subset X^e$ such that $\dim(X^{\mathcal{N}}) = \mathcal{N} < \infty$, and we take the Galerkin projection of our problem 1.14: Given $\boldsymbol{\mu} \in \mathbb{R}^P$ evaluate

$$\begin{aligned} & s^{\mathcal{N}}(\boldsymbol{\mu}) = f(u^{\mathcal{N}}(\boldsymbol{\mu}); \boldsymbol{\mu}), \\ & \text{where } u^{\mathcal{N}}(\boldsymbol{\mu}) \in X^{\mathcal{N}}(\Omega) \text{ satisfies} \\ & a\left(u^{\mathcal{N}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right) = f(v; \boldsymbol{\mu}), \quad \forall v \in X^{\mathcal{N}}. \end{aligned} \tag{1.15}$$

Typically, we take $\mathcal{N}$ large to obtain $|s^e(\boldsymbol{\mu}) - s^{\mathcal{N}}(\boldsymbol{\mu})|$ small.

Then, the reduced basis will be built on this truth approximation. Before introducing the reduced basis approximation, we define the inner product and the norm over the space $X^{\mathcal{N}}$ and X^e and the *energy norm* given by the coercive bilinear form a.

For $w, v \in X^e$, we define the *energy inner product* and the *energy norm* as

$$(w, v)_{\boldsymbol{\mu}} = a(w, v; \boldsymbol{\mu}) \tag{1.16}$$

$$\|w\|_{\boldsymbol{\mu}} = \sqrt{(w, w)_{\boldsymbol{\mu}}}. \tag{1.17}$$

Moreover, for a given $\overline{\boldsymbol{\mu}} \in \mathcal{D}$, we define the X^e-inner product and the X^e-norm for $w, v \in X^e$ as

$$(w, v)_X = (w, v)_{\overline{\boldsymbol{\mu}}} + \tau(w, v)_{L^2(\Omega)} \tag{1.18}$$

$$\|w\|_X = \sqrt{(w, w)_X}, \tag{1.19}$$

where τ is a non-negative real parameter and $(w,v)_{L^2(\Omega)} = \int_\Omega wv\, d\Omega$. Since $X^{\mathcal{N}} \subset X^e$, the inner products and the norms define above are the same for the space $X^{\mathcal{N}}$.

Remark 1.9. The choice of $\overline{\boldsymbol{\mu}}$ and τ will affect the quality and efficiency of our reduced basis a posteriori error estimators, but will not affect directly our reduced basis output predictions (see [20]).

Now, we can define precisely the exact and finite element coercivity respectively as

$$\alpha^e(\boldsymbol{\mu}) = \inf_{w \in X^e} \frac{a(w,w;\boldsymbol{\mu})}{\|w\|_X^2} \tag{1.20}$$

$$\alpha^{\mathcal{N}}(\boldsymbol{\mu}) = \inf_{w \in X^{\mathcal{N}}} \frac{a(w,w;\boldsymbol{\mu})}{\|w\|_X^2}. \tag{1.21}$$

From the coercivity hypothesis, we have that $\alpha^e(\boldsymbol{\mu}) \geq \alpha_0 > 0$, $\forall \boldsymbol{\mu} \in \mathcal{D}$, and from hypothesis on $X^{\mathcal{N}}$ (conforming space), we have that $\alpha^{\mathcal{N}}(\boldsymbol{\mu}) \geq \alpha^e(\boldsymbol{\mu})$, $\forall \boldsymbol{\mu} \in \mathcal{D}$. In the same way, the continuity constants are defined as

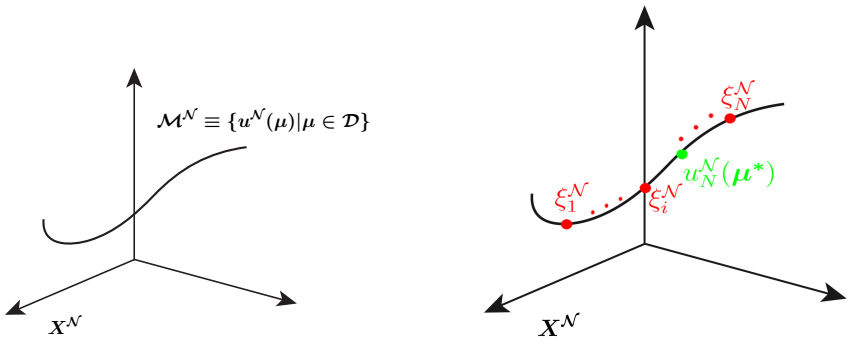
$$\gamma^e(\boldsymbol{\mu}) = \sup_{w \in X^e} \sup_{v \in X^e} \frac{a(w,v;\boldsymbol{\mu})}{\|w\|_X \|v\|_X} \tag{1.22}$$

$$\gamma^{\mathcal{N}}(\boldsymbol{\mu}) = \sup_{w \in X^{\mathcal{N}}} \sup_{v \in X^{\mathcal{N}}} \frac{a(w,v;\boldsymbol{\mu})}{\|w\|_X \|v\|_X}. \tag{1.23}$$

From hypothesis, $\gamma^e(\boldsymbol{\mu})$ is finite $\forall \boldsymbol{\mu} \in \mathcal{D}$ and that $\gamma^{\mathcal{N}}(\boldsymbol{\mu}) \leq \gamma^e(\boldsymbol{\mu})$, $\forall \boldsymbol{\mu} \in \mathcal{D}$.

1.2.3 Reduced basis approximation

Roughly speaking, the reduced basis discretization is a Galerkin projection on a N-dimensional approximation space that focuses on a low-dimensional, smooth, parametric manifold $\mathcal{M}^{\mathcal{N}} = \{u(\boldsymbol{\mu}) \,|\, \boldsymbol{\mu} \in \mathcal{D}\}$, induced by the parametric dependence. The space $X^{\mathcal{N}}$ is too general because it approximates all members of X^e. So, to approximate a solution $u^{\mathcal{N}}(\boldsymbol{\mu})$ it is sufficient to be able to approximate only functions in $\mathcal{M}^{\mathcal{N}}$. In the case of one parameter ($P = 1$), the manifold $\mathcal{M}^{\mathcal{N}}$ is a one dimensional filament Fig.1.1(a). Then, the idea is to choose and compute N functions $\xi_1^{\mathcal{N}}, \ldots, \xi_N^{\mathcal{N}} \in \mathcal{M}^{\mathcal{N}}$, called *snapshots* and then, for an arbitrary value $\boldsymbol{\mu}^* \in \mathcal{D}$, we compute the solution associated to this parameter (denoted $u_N^{\mathcal{N}}(\boldsymbol{\mu}^*)$) taking a good linear combinations of $\xi_k^{\mathcal{N}}$, $k = 1, \ldots, N$.



(a) Low-dimensional smooth parametric manifold.

(b) Approximation of $u_N^{\mathcal{N}}(\boldsymbol{\mu}^*)$ taking a linear combination of $\xi_1^{\mathcal{N}} \dots \xi_N^{\mathcal{N}}$.

Figure 1.1:

More precisely, let N_{max} be an integer. For $N = 1 \dots N_{max}$, let $X_N^{\mathcal{N}}$ be a N-dimensional subspace of $X^{\mathcal{N}}$. We assume that these spaces satisfy *the nested or hierarchical condition*, i.e.

$$X_1^{\mathcal{N}} \subset \dots \subset X_{N_{max}}^{\mathcal{N}} \subset X^{\mathcal{N}}. \tag{1.24}$$

The condition (1.24) is very important for efficiency and for reducing the computational cost. There are several spaces that satisfy the hierarchical solution, for example Taylor, Lagrange [20] and Hermite spaces, and more recently POD spaces [15]. In this work, we will focus on *Lagrange spaces* and *POD spaces*. These spaces will be introduced later (see Section 1.3). Now, we consider the Galerkin projection to obtain our reduced basis approximation:

Given $\boldsymbol{\mu} \in \mathcal{D}$, evaluate

$$\begin{aligned} &s_N^{\mathcal{N}}(\boldsymbol{\mu}) = f(u_N^{\mathcal{N}}(\boldsymbol{\mu}); \boldsymbol{\mu}), \\ &\text{where } u_N^{\mathcal{N}}(\boldsymbol{\mu}) \in X_N^{\mathcal{N}} \subset X^{\mathcal{N}} \text{ satisfies} \\ &a\left(u_N^{\mathcal{N}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right) = f(v; \boldsymbol{\mu}), \quad \forall v \in X_N^{\mathcal{N}}. \end{aligned} \tag{1.25}$$

From coercivity and continuity hypothesis on a and f, our conforming reduced basis $X_N^{\mathcal{N}} \subset X^{\mathcal{N}}$, problem (1.25) admits a unique solution. Moreover, we can demonstrate the well known Galerkin optimality results:

Proposition 1.10. See [15].. *For any $\boldsymbol{\mu} \in \mathcal{D}$ and $u_N^{\mathcal{N}}(\boldsymbol{\mu})$ and $s_N^{\mathcal{N}}(\boldsymbol{\mu})$ satisfying (1.25)*

$$\left\|u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})\right\|_{\boldsymbol{\mu}} = \inf_{w_N \in X_N^{\mathcal{N}}} \left\|u^{\mathcal{N}}(\boldsymbol{\mu}) - w_N(\boldsymbol{\mu})\right\|_{\boldsymbol{\mu}}, \tag{1.26}$$

$$\left\|u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})\right\|_X \leq \sqrt{\frac{\gamma^e(\boldsymbol{\mu})}{\alpha^e(\boldsymbol{\mu})}} \inf_{w_N \in X_N^{\mathcal{N}}} \left\|u^{\mathcal{N}}(\boldsymbol{\mu}) - w_N(\boldsymbol{\mu})\right\|_X, \tag{1.27}$$

and furthermore for the output:

$$\begin{aligned} s^{\mathcal{N}}(\boldsymbol{\mu}) - s_N^{\mathcal{N}}(\boldsymbol{\mu}) &= \left\|u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})\right\|_{\boldsymbol{\mu}}^2 \\ &= \inf_{w_N \in X_N^{\mathcal{N}}} \left\|u^{\mathcal{N}}(\boldsymbol{\mu}) - w_N\right\|_{\boldsymbol{\mu}}^2, \end{aligned} \tag{1.28}$$

as well as

$$0 < s^{\mathcal{N}}(\boldsymbol{\mu}) - s_N^{\mathcal{N}}(\boldsymbol{\mu}) \leq \gamma^e(\boldsymbol{\mu}) \inf_{w_N \in X_N^{\mathcal{N}}} \left\|u^{\mathcal{N}}(\boldsymbol{\mu}) - w_N(\boldsymbol{\mu})\right\|_X^2. \tag{1.29}$$

1.2.4 Offline-Online procedure

The Offline-Online procedure is essentially efficient thanks to the affine parameter dependence of a (1.13). First of all, we introduce a base for spaces $X^{\mathcal{N}}$ and $X_N^{\mathcal{N}}$ to obtain an algebraic system instead of (1.25). Then, let $(\phi_1^{FE}, \ldots, \phi_{\mathcal{N}}^{FE})$ a base for $X^{\mathcal{N}}$ and $(\xi^1, \ldots, \xi^N)$ a base for $X_N^{\mathcal{N}}$ such that

$$\xi^j = \sum_{k=1}^{\mathcal{N}} \xi_k^j \phi_k^{FE} \quad j = 1, \ldots N. \tag{1.30}$$

Remark 1.11. In the following, we will omit the superscript $^{\mathcal{N}}$ in the reduced basis approximation of the solution, i.e. $u_N^{\mathcal{N}} \equiv u_N$.

If we express $u_N(\boldsymbol{\mu}) = \sum_{j=1}^{N} u_{N\,j}(\boldsymbol{\mu})\xi^j$, problem (1.25) can be rewritten as:

$$\text{given } \boldsymbol{\mu} \in \mathcal{D} \text{ evaluate, } s_N(\boldsymbol{\mu}) = f(u_N(\boldsymbol{\mu})) = \sum_{j=1}^{N} u_{N\,j}(\boldsymbol{\mu}) f(\xi^j; \boldsymbol{\mu})$$

where

$$\sum_{j=1}^{N} u_{N\,j}(\boldsymbol{\mu}) a(\xi^j, \xi^i; \boldsymbol{\mu}) = f(\xi^i; \boldsymbol{\mu}) \quad 1 \leq i \leq N.$$

Now, using the affine parameter dependence of a and f we obtain the following problem:

$$\text{given } \boldsymbol{\mu} \in \mathcal{D} \text{ evaluate, } s_N(\boldsymbol{\mu}) = \sum_{j=1}^{N} \sum_{q=1}^{Q_f} u_{N\,j}(\boldsymbol{\mu}) \theta_f^q(\boldsymbol{\mu}) f^q(\xi^j)$$

where

$$\sum_{j=1}^{N} u_{N\,j}(\boldsymbol{\mu}) \theta_a^q(\boldsymbol{\mu}) a^q(\xi^j, \xi^i) = \sum_{q=1}^{Q_f} \theta_f^q(\boldsymbol{\mu}) f^q(\xi^i) \quad 1 \le i \le N.$$

The Offline stage is now clear. We must form and store values of $a^q(\xi^j, \xi^i)$ for $1 \le i, j \le N$, $1 \le q \le Q_a$, and the values of $f^q(\xi^j)$ for $1 \le j \le N$, $1 \le q \le Q_f$.

In the Online stage, we form the sum $\sum_{q=1}^{Q_a} \theta_a^q(\boldsymbol{\mu}) a^q(\xi^j, \xi^i)$ for $1 \le i, j \le N$ and then we solve the RB linear system to obtain the $u_{N\,j}(\boldsymbol{\mu})$.

For convenient reasons for the operation count, we introduce some matrices and vectors:

- the matrix

$$Z_N \in \mathbb{R}^{\mathcal{N} \times N},$$

which i^{th}-column is the vector of components ξ_k^i, $1 \le k \le \mathcal{N}$ of the function ξ^i, $1 \le i \le N$;

- the parameter-independent matrices

$$A^q = a(\phi_n^{FE}, \phi_m^{FE}), \quad 1 \le n, m \le \mathcal{N},$$

and

$$A_N^q = a(\xi^n, \xi^m), \quad 1 \le n, m \le N,$$

where $1 \le q \le Q_a$.

- the parameter-independent vectors

$$F^q = f^q(\phi_n^{FE}), \quad 1 \le n \le \mathcal{N},$$

and

$$F_N^q = f^q(\xi_n), \quad 1 \le n \le N,$$

where $1 \le q \le Q_f$.

- *the assembled FEM stiffness matrix and load vector*

$$A(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \theta_a^q(\boldsymbol{\mu}) A^q,$$

and

$$F(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \theta_f^q(\boldsymbol{\mu}) F^q,$$

- *the assembled RB stiffness matrix and load vector*

$$A_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \theta_a^q(\boldsymbol{\mu}) A_N^q,$$

and

$$F_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \theta_f^q(\boldsymbol{\mu}) F_N^q.$$

Then, using

$$a(\xi^n, \xi^m; \boldsymbol{\mu}) = \sum_{i=1}^{\mathcal{N}} \sum_{j=1}^{\mathcal{N}} \xi_i^m a^q(\phi_i^{FE}, \phi_j^{FE}; \boldsymbol{\mu}) \quad 1 \le n, m \le N,$$

we have the following relations between the matrices and vectors defined above:

$$\begin{aligned} A_N(\boldsymbol{\mu}) &= Z_N^T A(\boldsymbol{\mu}) Z_N, && (1.31)\\ A_N^q &= Z_N^T A^q Z_N, \quad 1 \le q \le Q_a, && (1.32)\\ F_N(\boldsymbol{\mu}) &= Z_N^T F(\boldsymbol{\mu}) Z_N, && (1.33)\\ F_N^q &= Z_N^T F^q Z_N, \quad 1 \le q \le Q_f, && (1.34) \end{aligned}$$

where all these quantities are defined for $1 \le N \le N_{max}$.

1.2.5 Operation count and storage

The cost of the Offline stage depends on $\mathcal{N}$, N_{max}, Q_a and Q_f, while the Online stage depends on N_{max}, Q_a and Q_f. The most important thing here

is that the cost of the Online stage is independent of $\mathcal{N}$. Then, the evaluation $\boldsymbol{\mu} \rightarrow s_N(\boldsymbol{\mu})$ is very quick and very important in the real-time and many query context.
More precisely, the Offline stage is of the order $\mathcal{O}(\mathcal{N})$ (expensive). For the Online, we have $\mathcal{O}(Q_a N_{max}^2)$ and $\mathcal{O}(Q_f N_{max})$ operation for assembling $A_N(\boldsymbol{\mu})$ and $F_N(\boldsymbol{\mu})$ respectively. We need $\mathcal{O}(N_{max}^3)$ operations to solve the RB linear system (see [17]) and finally, the evaluation of the output requires N operations.

1.3 Sample/space assembling strategies

In this section, we will presents two sampling strategies to obtain our reduced basis. The first is the *Proper Orthogonal Decomposition*, (*POD*) and the second is the *Greedy Lagrange*. In section 3.4.4 of this work, we will compare these two strategies with some examples.

Let $\Xi_{train} = \{\boldsymbol{\mu}_{train}^1, \ldots, \boldsymbol{\mu}_{train}^{n_{train}}\}$ be a finite sample set, called *test sample* of point in $\mathcal{D}$. These parameter are often chosen by Monte Carlo methods with respect to a uniform or log-uniform density. We assume that the cardinality of $|\Xi_{train}| = n_{train}$ is very large to cover all the set $\mathcal{D}$. Now, we define the following norms that will be useful for the next: for a function $y : \mathcal{D} \longrightarrow \mathbb{R}$,

$$\|y\|_{L^\infty(\Xi_{train})} = \max_{\boldsymbol{\mu} \in \Xi_{train}} |y(\boldsymbol{\mu})|,$$

and

$$\|y\|_{L^p(\Xi_{train})} = \left(|\Xi_{train}|^{-1} \sum_{\boldsymbol{\mu} \in \Xi_{train}} |y|^p(\boldsymbol{\mu}) \right)^{1/p}.$$

For a function $z : \mathcal{D} \longrightarrow X^{\mathcal{N}}$ (or X^e) we then define in the same way,

$$\|z\|_{L^\infty(\Xi_{train};X)} = \max_{\boldsymbol{\mu} \in \Xi_{train}} \|z(\boldsymbol{\mu})\|_X ,$$

and

$$\|z\|_{L^p(\Xi_{train};X)} = \left(|\Xi_{train}|^{-1} \sum_{\boldsymbol{\mu} \in \Xi_{train}} \|z(\boldsymbol{\mu})\|_X^p \right)^{1/p}.$$

1.3.1 POD RB spaces

The proper orthogonal decomposition (POD) approach is popular most notably in time-domain reduced order modelling (see [24]). The technique can also be applied within the parametric context, as we now describe (see [15]).
Given Ξ_{train}, we define the POD RB spaces $X_N^{\mathcal{N}\,POD}$ as the solution of the optimization problem

$$X_N^{\mathcal{N}\,POD} = \arg \inf_{X_N^{\mathcal{N}\,POD} \subset span\{u^{\mathcal{N}}(\boldsymbol{\mu}) \,|\, \boldsymbol{\mu} \in \Xi_{train}\}} \left\| u^{\mathcal{N}} - \Pi_{X_N^{\mathcal{N}}} u^{\mathcal{N}} \right\|_{L^2(\Xi_{train};X)}, \tag{1.35}$$

where $\Pi_{X_N^{\mathcal{N}}} : X^{\mathcal{N}} \longrightarrow X_N^{\mathcal{N}}$ refers to the projection in the X-inner product. Now, we define the correlation matrix $C^{POD} \in \mathbb{R}^{n_{train} \times n_{train}}$ given by

$$C_{ij}^{POD} = |\Xi_{ntrain}|^{-1} \left(u^{\mathcal{N}}(\boldsymbol{\mu}_{train}^i), u^{\mathcal{N}}(\boldsymbol{\mu}_{train}^j) \right)_X \quad 1 \le i, j \le n_{train}. \tag{1.36}$$

Now, if we express $u^{\mathcal{N}}(\boldsymbol{\mu}_{train}^k)$ as

$$u^{\mathcal{N}}(\boldsymbol{\mu}_{train}^k) = \sum_{q=1}^{\mathcal{N}} u^q(\boldsymbol{\mu}_{train}^k) \phi_q^{FE}, \quad \forall 1 \le k \le n_{train},$$

and if we define the vector $\boldsymbol{u}(\boldsymbol{\mu}_{train}^k) = [u^1(\boldsymbol{\mu}_{train}^k), \ldots, u^{\mathcal{N}}(\boldsymbol{\mu}_{train}^k)]^T$ and the matrix $X^{\mathcal{N}} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ is such that

$$X_{ij}^{\mathcal{N}} = \left(\phi_j^{FE}, \phi_i^{FE} \right)_X,$$

we have that

$$C_{ij}^{POD} = |\Xi|^{-1} \left(\boldsymbol{u}(\boldsymbol{\mu}_{train}^i) \right)^T X^{\mathcal{N}} \left(\boldsymbol{u}(\boldsymbol{\mu}_{train}^j) \right). \tag{1.37}$$

We then solve the following eigenproblem: find $(\boldsymbol{\psi}^{POD,\,k}, \lambda^{POD\,k}) \in \mathbb{R}^{n_{train}} \times \mathbb{R}_+^*$, $1 \le k \le n_{train}$ such that

$$C^{POD} \boldsymbol{\psi}^{POD,\,k} = \lambda^{POD,\,k} \boldsymbol{\psi}^{POD,\,k}, \quad \left(\boldsymbol{\psi}^{POD,\,k} \right)^T X^{\mathcal{N}} \boldsymbol{\psi}^{POD,\,k} = 1. \tag{1.38}$$

Arranging the eigenvalues in descending order $\lambda^{POD,\,1} \ge \lambda^{POD,\,2} \ge \ldots \ge \lambda^{POD,\,n_{train}} \ge 0$, we define $\Psi^{POD,\,k} \in X^{\mathcal{N}}$, $1 \le k \le n_{train}$ as

$$\Psi^{POD,\,k} = \sum_{m=1}^{n_{train}} \psi_m^{POD,\,k} u^{\mathcal{N}}(\boldsymbol{\mu}_{train}^m), \quad 1 \le k \le n_{train}. \tag{1.39}$$

We define N_{max} as the smallest N such that

$$\sqrt{\sum_{k=N+1}^{n_{train}} \lambda^{POD,\,k}} \leq \varepsilon_{tol,\,min}. \tag{1.40}$$

We then construct our POD RB spaces as

$$X_N^{\mathcal{N}\,POD} = span\{\Psi^{POD,\,n},\, 1 \leq n \leq N\}, \quad 1 \leq N \leq N_{max}. \tag{1.41}$$

We then take $X_N = X_N^{\mathcal{N}\,POD}$ our reduced basis space for the choice $\xi^n = \Psi^{POD,\,n}$, $1 \leq n \leq N$.

The POD is an extremely expensive method. Indeed, we have to compute all solution $u^{\mathcal{N}}(\boldsymbol{\mu})$ for all $\boldsymbol{\mu} \in \Xi_{train}$. The most expensive part is the construction of the correlation matrix and then the resolution of several eigenproblems (1.38). But the POD is very useful for some small n_{train} that is why it is used in time domain application.

1.3.2 Greedy Lagrange spaces

The idea of this strategy is starting with Ξ_{train}, we select N parameters $\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^N$ and we form the reduced basis space

$$X_N = span\left\{\xi_n = u^{\mathcal{N}}(\boldsymbol{\mu_n}),\, 1 \leq n \leq N\right\} \qquad \text{(see [20])}.$$

More precisely, for the greedy approach, we need a sharp, rigorous and efficient bound $\Delta_N^{en}(\boldsymbol{\mu})$ for the reduced basis error $\left\|u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\right\|_X$, where u_N is our RB approximation associated with the space X_N. To quantify the sharpness and rigor properties, we introduce the effectivity:

$$\eta_N^{en}(\boldsymbol{\mu}) = \frac{\Delta_N^{en}(\boldsymbol{\mu})}{\left\|u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\right\|_X}, \tag{1.42}$$

and we require

$$1 \leq \eta_N^{en}(\boldsymbol{\mu}) \leq \eta_{max,\,UB}^{en} \quad \forall \boldsymbol{\mu} \in \mathcal{D},\, 1 \leq N \leq N_{max}, \tag{1.43}$$

where $\eta_{max,\,UB}$ is finite and independent of N. The *rigor* property is illustrated by the left inequality, i.e. the error bound Δ_N^{en} is never less than the true error. The right inequality illustrates the *sharpness*, i.e. Δ_N^{en} is not

too much larger than the true error. *Efficient* means that the evaluation $\boldsymbol{\mu} \rightarrow \Delta_N^{en}(\boldsymbol{\mu})$ is independent of $\mathcal{N}$. In section 1.5 we will explain the Offline-Online procedure for compute $\Delta_N^{en}(\boldsymbol{\mu})$.
Now, we present the greedy algorithm. We define $\overline{N}_{max}$ an upper bound for N_{max}, $\varepsilon_{tol,min}$ a tolerance of the error.
Given Ξ_{train}, $S_1 = \{\boldsymbol{\mu}^1\}$ and $X_1 = span\left\{u^{\mathcal{N}}(\boldsymbol{\mu}^1)\right\}$,

$$
\begin{aligned}
\text{For } N \;=\; & 2 : \overline{N}_{max} \\
& \boldsymbol{\mu}^N = \arg \max_{\boldsymbol{\mu} \in \Xi_{train}} \Delta_{N-1}^{en}(\boldsymbol{\mu}); \\
& \varepsilon_{N-1} = \Delta_{N-1}^{en}(\boldsymbol{\mu}^N); \\
& \text{if } \varepsilon_{N-1} \leq \varepsilon_{tol,min} \\
& \qquad N_{max} = N - 1; \\
& \text{end;} \\
& S_N = S_{N-1} \cup \boldsymbol{\mu}^N; \\
& X_N = X_{N-1} + span\left\{u^{\mathcal{N}}(\boldsymbol{\mu}^{N_*})\right\}; \\
\text{end.}
\end{aligned}
$$

With POD we have to compute the snapshots for all $\boldsymbol{\mu} \in \Xi_{train}$, here we only have to compute N_{max} snapshots. In this strategy we use a posteriori error bound $\Delta_N^{en}(\boldsymbol{\mu})$ to approximate the expensive true error $\left\|u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\right\|_X$.

Remark 1.12. Note that in theory, the Greedy minimize the error $|u^{\mathcal{N}} - u_N|$ in the L^∞-norm while the POD minimize the projection error in L^2-norm.

1.4 Geometric variations

In this section, we will explain how to operate if the domain is parameter-dependent. Indeed, the reduced basis method described above requires that Ω be a parameter independent domain. If we want to consider geometric variations of the domain, we have to assume that *the reference domain* Ω is the pre-image of the parameter-dependent *original* domain $\Omega_o(\boldsymbol{\mu})$. For more details, the reader can refer to [20] and [21].

We shall assume that, for all $\boldsymbol{\mu}$ in $\mathcal{D}$, we have a domain decomposition of $\Omega_o(\boldsymbol{\mu})$,

$$
\overline{\Omega}_o(\boldsymbol{\mu}) = \cup_{k=1}^{K_{dom}} \overline{\Omega}_o^k(\boldsymbol{\mu}) \;, \tag{1.44}
$$

where the $\Omega_o^k(\boldsymbol{\mu}), 1 \le k \le K_{dom}$, are mutually non-overlapping open subdomains,

$$\Omega_o^k(\boldsymbol{\mu}) \cap \Omega_o^{k'}(\boldsymbol{\mu}) = \emptyset, \quad 1 \le k < k' \le K_{dom} \ . \tag{1.45}$$

This coarse domain decomposition will be denoted *reduced basis (RB) triangulation*.

We now choose a value $\boldsymbol{\mu}_{ref} \in \mathcal{D}$ and define our reference domain as $\Omega \equiv \Omega_o(\boldsymbol{\mu}_{ref})$. It is easy to see that

$$\overline{\Omega} = \cup_{k=1}^{K_{dom}} \overline{\Omega}^k \ , \tag{1.46}$$

$$\Omega^k \cap \Omega^{k'} = \emptyset, \quad 1 \le k < k' \le K_{dom} \ , \tag{1.47}$$

We will build a very fine finite element (FE) subtriangulation of the RB triangulation of Ω. This FE subtriangulation ensures that the FE approximation accurately treats the perhaps discontinuous coefficients (arising from property and geometry variation) associated with the different subdomains and the subtriangulation also plays an important role in the generation of our affine representation (1.13). We emphasize that $\boldsymbol{\mu}_{ref}$ only affects the accuracy of the underlying FE approximation upon which the RB discretization and *a posteriori* error estimator is built.

A necessary condition for the affine representation (1.13) is the so called *Affine Geometry Precondition*. This condition says that for any original domain $\Omega_o(\boldsymbol{\mu})$ that admits a domain decomposition (1.44) there exists *affine* mappings $\mathcal{T}^{aff,k}(\cdot;\boldsymbol{\mu})$: $\Omega^k \rightarrow \Omega_o^k(\boldsymbol{\mu})$, $1 \le k \le K_{dom}$, that are

(i) individually *bijective*, and

(ii) collectively *continuous*, i.e.

$$\mathcal{T}^{aff,k}(\boldsymbol{x};\boldsymbol{\mu}) = \mathcal{T}^{aff,k'}(\boldsymbol{x};\boldsymbol{\mu}), \quad \begin{array}{l} \forall\, \boldsymbol{x} \in \overline{\Omega}^k \cap \overline{\Omega}^{k'} \ , \\ 1 \le k < k' \le K_{dom} \ , \end{array} \tag{1.48}$$

$\forall \boldsymbol{\mu} \in \mathcal{D}$ and such that

$$\overline{\Omega}_o^k(\boldsymbol{\mu}) = \mathcal{T}^{aff,k}(\overline{\Omega}^k;\boldsymbol{\mu}), \quad 1 \le k \le K_{dom}. \tag{1.49}$$

Note that we purposely define K_{dom} with respect to the *exact* problem, rather than the FE approximation: K_{dom} can not depend on $\mathcal{N}$ (to be meaningful).

We now define our (bijective) affine mappings more explicitly: for $1 \leq k \leq K_{dom}$, any $\boldsymbol{\mu}$ in $\mathcal{D}$, and any $\boldsymbol{x} \in \Omega^k$,

$$\mathcal{T}_i^{aff,k}(\boldsymbol{x};\boldsymbol{\mu}) = C_i^{aff,k}(\boldsymbol{\mu}) + \sum_{j=1}^{d} G_{ij}^{aff,k}(\boldsymbol{\mu})\, x_j, \quad 1 \leq i \leq d\ , \tag{1.50}$$

for given $C^{aff,k}$: $\mathcal{D} \rightarrow \mathbb{R}^d$ and $G^{aff,k}$: $\mathcal{D} \rightarrow \mathbb{R}^{d\times d}$. We can then define the associated Jacobians

$$J^{aff,k}(\boldsymbol{\mu}) = |det(G^{aff,k}(\boldsymbol{\mu}))|, \quad 1 \leq k \leq K_{dom}\ , \tag{1.51}$$

where *det* denotes the determinant. Note that the Jacobian is constant in space over each subdomain. We further define, for any $\boldsymbol{\mu} \in \mathcal{D}$,

$$D^{aff,k}(\boldsymbol{\mu}) = (G^{aff,k}(\boldsymbol{\mu}))^{-1}, \quad 1 \leq k \leq K_{dom}\ ; \tag{1.52}$$

this matrix shall prove convenient in subsequent derivative transformations.

We may interpret our local mappings in terms of a global transformation. In particular, for any $\boldsymbol{\mu} \in \mathcal{D}$, the local mappings (1.49) induce a global bijective *piecewise-affine* transformation $\mathcal{T}^{aff}(\,\cdot\,;\boldsymbol{\mu})$: $\Omega \rightarrow \Omega_o(\boldsymbol{\mu})$: for any $\boldsymbol{\mu} \in \mathcal{D}$,

$$\mathcal{T}^{aff}(\boldsymbol{x};\boldsymbol{\mu}) = \mathcal{T}^{aff,k}(\boldsymbol{x};\boldsymbol{\mu}),\ k = \min_{k' \in \{1,\ldots,K_{dom}\}\ |\ x \in \overline{\Omega}^{k'}} k'; \tag{1.53}$$

note the one-to-one property of this mapping (and, hence the arbitrariness of our min choice in (1.53)) is ensured by the interface condition (1.48).

Below (Sec.1.4.1), we first consider a single subdomain. Finally, in Section 1.4.2, we discuss the incorporation of these affine mappings into our weak form.

1.4.1 Affine Mappings: Single Subdomain

As we consider a single subdomain in this section, we shall suppress the subdomain superscript for clarity of exposition. We shall focus on the three-dimensional case ($d = 3$). The 2-dimensional case is detailed in [20].

Then, the affine transformation (1.50) can be rewritten as

$$\mathcal{T}_i^{aff}(\boldsymbol{x};\boldsymbol{\mu}) = C_i^{aff}(\boldsymbol{\mu}) + \sum_{j=1}^{3} G_{ij}^{aff}(\boldsymbol{\mu})\, x_j, \quad 1 \leq i \leq d\ ; \tag{1.54}$$

we shall refer to $C^{aff}(\boldsymbol{\mu}) \in \mathbb{R}^3$ and $G^{aff}(\boldsymbol{\mu}) \in \mathbb{R}^{3\times 3}$ as the *mapping coefficients*. In this case we have 12 mapping coefficients that entirely define the affine transformation (1.54). Under our assumption that the mapping is invertible we know that our Jacobian, $J^{aff}(\boldsymbol{\mu})$ of (1.51), is strictly positive, and that the derivative transformation matrix, $D^{aff}(\boldsymbol{\mu}) = (G^{aff}(\boldsymbol{\mu}))^{-1}$ of (1.52), is well defined.

The mapping coefficient can be identified by the relationship between 4 non-colinear pre-image points, or nodes,

$$(\overline{\boldsymbol{z}}^1, \overline{\boldsymbol{z}}^2, \overline{\boldsymbol{z}}^3, \overline{\boldsymbol{z}}^4) \equiv ((\overline{z}^1_1, \overline{z}^1_2, \overline{z}^1_3), (\overline{z}^2_1, \overline{z}^2_2, \overline{z}^2_3), (\overline{z}^3_1, \overline{z}^3_2, \overline{z}^3_3), (\overline{z}^4_1, \overline{z}^4_2, \overline{z}^4_3)),$$

in Ω, and 4 parametrized image nodes,

$$(\overline{\boldsymbol{z}}^1_o(\boldsymbol{\mu}), \overline{\boldsymbol{z}}^2_o(\boldsymbol{\mu}), \overline{\boldsymbol{z}}^3_o(\boldsymbol{\mu}), \overline{\boldsymbol{z}}^4_o(\boldsymbol{\mu})) \equiv \ldots$$

$$\ldots \equiv ((\overline{z}^1_{o1}, \overline{z}^1_{o2}, \overline{z}^1_{o3}), (\overline{z}^2_{o1}, \overline{z}^2_{o2}, \overline{z}^2_{o3}), (\overline{z}^3_{o1}, \overline{z}^3_{o2}, \overline{z}^3_{o3}), (\overline{z}^4_{o1}, \overline{z}^4_{o2}, \overline{z}^4_{o3}))(\boldsymbol{\mu})$$

in $\Omega_o(\boldsymbol{\mu})$. In particular, for given $\boldsymbol{\mu} \in \mathcal{D}$, application of (1.54) to the selected nodes yields

$$\overline{z}^m_{oi}(\boldsymbol{\mu}) = C^{aff}_i(\boldsymbol{\mu}) + \sum_{j=1}^{3} G^{aff}_{ij}(\boldsymbol{\mu})\, \overline{z}^m_j, \quad \begin{array}{l} 1 \le i \le 3\ , \\ 1 \le m \le 4\ ; \end{array} \tag{1.55}$$

(1.55) constitutes 12 independent equations by which to determine the 12 mapping coefficients.

To be more explicit in our construction, we first form the matrix $\mathbb{B}^{aff} \in \mathbb{R}^{12\times 12}$ (more generally, $\mathbb{R}^{(d^2+d)\times(d^2+d)}$),

$$\mathbb{B}^{aff} = \begin{bmatrix}
1 & 0 & 0 & \overline{z}^1_1 & \overline{z}^1_2 & \overline{z}^1_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & \overline{z}^1_1 & \overline{z}^1_2 & \overline{z}^1_3 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \overline{z}^1_1 & \overline{z}^1_2 & \overline{z}^1_3 \\
1 & 0 & 0 & \overline{z}^2_1 & \overline{z}^2_2 & \overline{z}^2_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & \overline{z}^2_1 & \overline{z}^2_2 & \overline{z}^2_3 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \overline{z}^2_1 & \overline{z}^2_2 & \overline{z}^2_3 \\
1 & 0 & 0 & \overline{z}^3_1 & \overline{z}^3_2 & \overline{z}^3_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & \overline{z}^3_1 & \overline{z}^3_2 & \overline{z}^3_3 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \overline{z}^3_1 & \overline{z}^3_2 & \overline{z}^3_3 \\
1 & 0 & 0 & \overline{z}^4_1 & \overline{z}^4_2 & \overline{z}^4_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & \overline{z}^4_1 & \overline{z}^4_2 & \overline{z}^4_3 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \overline{z}^4_1 & \overline{z}^4_2 & \overline{z}^4_3
\end{bmatrix}.$$

We further introduce the vector $V^{aff}(\boldsymbol{\mu})$ of image nodal locations,

$$V^{aff} = \begin{bmatrix} \overline{z}^1_{o\,1}(\boldsymbol{\mu}) \\ \overline{z}^1_{o\,2}(\boldsymbol{\mu}) \\ \overline{z}^1_{o\,3}(\boldsymbol{\mu}) \\ \overline{z}^2_{o\,1}(\boldsymbol{\mu}) \\ \overline{z}^2_{o\,2}(\boldsymbol{\mu}) \\ \overline{z}^2_{o\,3}(\boldsymbol{\mu}) \\ \overline{z}^3_{o\,1}(\boldsymbol{\mu}) \\ \overline{z}^3_{o\,2}(\boldsymbol{\mu}) \\ \overline{z}^3_{o\,3}(\boldsymbol{\mu}) \\ \overline{z}^4_{o\,1}(\boldsymbol{\mu}) \\ \overline{z}^4_{o\,2}(\boldsymbol{\mu}) \\ \overline{z}^4_{o\,3}(\boldsymbol{\mu}) \end{bmatrix}.$$

We obtain our affine mappings

$$\begin{bmatrix} C^{aff}_1(\boldsymbol{\mu}) \\ C^{aff}_2(\boldsymbol{\mu}) \\ C^{aff}_3(\boldsymbol{\mu}) \\ G^{aff}_{11}(\boldsymbol{\mu}) \\ G^{aff}_{12}(\boldsymbol{\mu}) \\ G^{aff}_{13}(\boldsymbol{\mu}) \\ G^{aff}_{21}(\boldsymbol{\mu}) \\ G^{aff}_{22}(\boldsymbol{\mu}) \\ G^{aff}_{23}(\boldsymbol{\mu}) \\ G^{aff}_{31}(\boldsymbol{\mu}) \\ G^{aff}_{32}(\boldsymbol{\mu}) \\ G^{aff}_{33}(\boldsymbol{\mu}) \end{bmatrix} = \left(\mathbb{B}^{aff}\right)^{-1} V^{aff}(\boldsymbol{\mu})$$

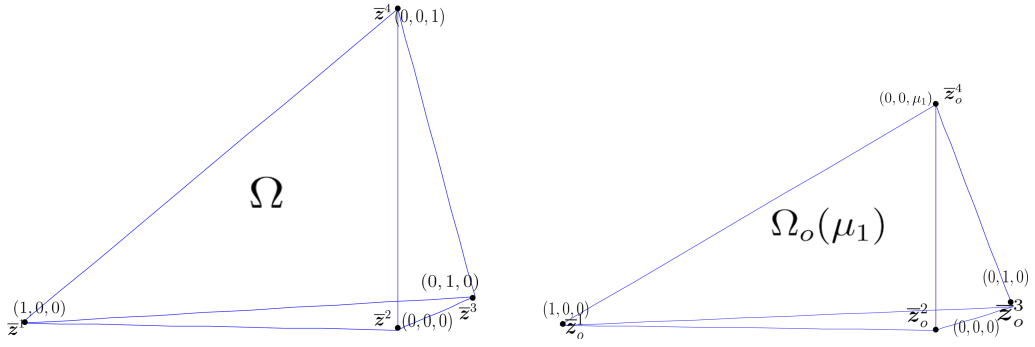
note that $\mathbb{B}^{aff}$ is non-singular under our hypothesis of non-colinear pre-image nodes.

The matrix $\mathbb{B}^{aff}$ is independent of $\boldsymbol{\mu}$; the parametric dependence derives from $V^{aff}(\boldsymbol{\mu})$. To illustrate how the parametric dependence propagates from the (desired) parametrized domain to the mapping coefficients, we give the example of a tetrahedra.

We note that parallelepipeds are the most intuitive subdomains by which to effect transformations by hand, invoking the usual translation, dilation,

rotation, and shear primitives. However, we can state a parallelism with the 2D case and consider curvy tetrahedra or curvy tetrahedra instead of triangle and curvy triangle. Here, we shall thus focus only on tetrahedra building blocks.

So, let be a tetrahedra as in figure 1.2(a) with vertices $\overline{\boldsymbol{z}}^1 = (1,0,0)$, $\overline{\boldsymbol{z}}^2 = (0,0,0)\,\overline{\boldsymbol{z}}^3 = (0,1,0)$ and $\overline{\boldsymbol{z}}^4 = (0,0,1)$ that are pre-images nodes of points $\overline{\boldsymbol{z}}_o^1 = (1,0,0)$, $\overline{\boldsymbol{z}}_o^2 = (0,0,0)$, $\overline{\boldsymbol{z}}_0^3 = (0,1,0)$ and $\overline{\boldsymbol{z}}_o^4 = (0,0,\mu_1)$ (Fig.1.2(b)). The reference domain and the original domain are respectively denoted by Ω and $\Omega(\mu_1)$. The pre-images nodes correspond to the image node for a particular value μ_{ref} of the parameter. Here, $\mu_{ref} = 1$, i.e $(\overline{\boldsymbol{z}}^1, \overline{\boldsymbol{z}}^2, \overline{\boldsymbol{z}}^3, \overline{\boldsymbol{z}}^4) = (\overline{\boldsymbol{z}}_o^1(\mu_{ref}), \overline{\boldsymbol{z}}_o^2(\mu_{ref}), \overline{\boldsymbol{z}}_o^3(\mu_{ref}), \overline{\boldsymbol{z}}_o^4(\mu_{ref}))$



(a) Reference domain Ω.

(b) Original domain $\Omega_o(\mu_1)$.

Figure 1.2: Variation on a tetrahedra

Then, we find that

$$\begin{bmatrix} C_1^{aff}(\boldsymbol{\mu}) \\ C_2^{aff}(\boldsymbol{\mu}) \\ C_3^{aff}(\boldsymbol{\mu}) \\ G_{11}^{aff}(\boldsymbol{\mu}) \\ G_{12}^{aff}(\boldsymbol{\mu}) \\ G_{13}^{aff}(\boldsymbol{\mu}) \\ G_{21}^{aff}(\boldsymbol{\mu}) \\ G_{22}^{aff}(\boldsymbol{\mu}) \\ G_{23}^{aff}(\boldsymbol{\mu}) \\ G_{31}^{aff}(\boldsymbol{\mu}) \\ G_{32}^{aff}(\boldsymbol{\mu}) \\ G_{33}^{aff}(\boldsymbol{\mu}) \end{bmatrix} = \left(\mathbb{B}^{aff}\right)^{-1} V^{aff}(\boldsymbol{\mu})$$

$$= \left(\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \mu_1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \mu_1 \end{bmatrix}.$$

Consequently, we have that $C^{aff}(\mu_1) = 0$ and

$$G^{aff}(\mu_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mu_1 \end{bmatrix}$$

It follows that

$$J^{aff}(\mu_1) = \mu_1$$

and

$$D^{aff}(\mu_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\mu_1} \end{bmatrix}.$$

1.4.2 Bilinear Form

Here, we will focus on the transformation that we have to operate on the weak form to if our domain $\Omega_o(\boldsymbol{\mu})$ allows the Affine Geometry Precondition described in the previous section.

Formulation on Original Domain

Our problem is initially posed on the original domain $\Omega_o(\boldsymbol{\mu})$. We shall assume for simplicity that $X_o^e(\boldsymbol{\mu}) = H_0^1(\Omega_o(\boldsymbol{\mu}))$, which corresponds to homogeneous Dirichlet boundary conditions over the entire boundary $\partial\Omega_o(\boldsymbol{\mu})$.

Given $\boldsymbol{\mu} \in \mathbb{R}^P$ evaluate

$$s_o^e(\boldsymbol{\mu}) = f_o(u_o^e(\boldsymbol{\mu}); \boldsymbol{\mu})$$

where $u_o^e(\boldsymbol{\mu}) \in X_o^e(\Omega)$ satisfies

$$a_o\left(u_o^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right) = f_o(v; \boldsymbol{\mu}), \quad \forall v \in X_o^e.$$

We now place conditions on a_o and f_o such that, in conjunction with the Affine Geometry Precondition, we are ensured an affine expansion of the bilinear form.

In particular, we require that $a_o(\cdot, \cdot; \boldsymbol{\mu})$: $H^1(\Omega_o(\boldsymbol{\mu})) \times H^1(\Omega_o(\boldsymbol{\mu})) \rightarrow \mathbb{R}$

can be expressed as

$$a_o(w, v; \boldsymbol{\mu}) =$$

$$\sum_{k=1}^{K_{dom}} \int_{\Omega_o^k(\boldsymbol{\mu})} \left[\begin{array}{cccc} \frac{\partial w}{\partial x_{o1}} & \frac{\partial w}{\partial x_{o2}} & \frac{\partial w}{\partial x_{o3}} & w \end{array} \right] \mathcal{K}_{o,k\,ij}(\boldsymbol{\mu}) \left[\begin{array}{c} \frac{\partial v}{\partial x_{o1}} \\ \frac{\partial v}{\partial x_{o2}} \\ \frac{\partial v}{\partial x_{o3}} \\ v \end{array} \right] , \tag{1.56}$$

where $\boldsymbol{x}_o = (x_{o1}, x_{o2}, x_{03})$ denotes a point in $\Omega_o(\boldsymbol{\mu})$. Here, for $1 \le k \le K_{dom}$, $\mathcal{K}_{o,k}$: $\mathcal{D} \to \mathbf{R}^{4\times 4}$ is a given symmetric positive definite matrix (which in turn ensures coercivity of our bilinear form): the upper 3×3 principal submatrix of $\mathcal{K}_{o,k}$ is the usual tensor conductivity/diffusivity; the $(4,4)$ element of $\mathcal{K}_{o,\ell}$ represents the identity operator (*mass matrix*); and the $(4,1), (4,2), (4,3)$ (and $(1,4), (2,4), (3,4)$) elements of $\mathcal{K}_{o,\ell}$, which can be set here to zero thanks to our current restriction to symmetric operators, permit first derivative (or *convection*) terms.

Similarly, we require that f_o: $H^1(\Omega_o(\boldsymbol{\mu})) \to \mathbb{R}$ can be expressed as

$$f_o(v) = \sum_{k=1}^{K_{dom}} \int_{\Omega_o^k(\boldsymbol{\mu})} \mathcal{F}_{o,k}(\boldsymbol{\mu}) v \ ,$$

where, for $1 \le k \le K_{dom}$, $\mathcal{F}_{o,k}$: $\mathcal{D} \to \mathbb{R}$.

Formulation on Reference Domain

We now apply standard techniques to transform the problem statement over the original domain to an equivalent problem statement over the reference domain.

Given $\boldsymbol{\mu} \in \mathbb{R}^P$ evaluate

$$s^e(\boldsymbol{\mu}) = f(u^e(\boldsymbol{\mu}); \boldsymbol{\mu})$$

where $u^e(\boldsymbol{\mu}) \in X^e(\Omega)$ satisfies

$$a\left(u^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right) = f(v; \boldsymbol{\mu}), \quad \forall v \in X^e.$$

We may then identify $s^e(\boldsymbol{\mu}) = s_o^e(\boldsymbol{\mu})$ and $u^e(\boldsymbol{\mu}) = u_o^e(\boldsymbol{\mu}) \circ \mathcal{T}^{aff}(\cdot; \boldsymbol{\mu})$.

The transformed bilinear form, a, can be expressed as

$$a(w,v;\boldsymbol{\mu}) = \sum_{k=1}^{K_{dom}} \int_{\Omega^k} \left[\begin{array}{cccc} \frac{\partial w}{\partial x_1} & \frac{\partial w}{\partial x_2} & \frac{\partial w}{\partial x_3} & w \end{array} \right] \mathcal{K}_{ij}^k(\boldsymbol{\mu}) \left[\begin{array}{c} \frac{\partial v}{\partial x_1} \\ \frac{\partial v}{\partial x_2} \\ \frac{\partial v}{\partial x_3} \\ v \end{array} \right] , \tag{1.57}$$

where $\boldsymbol{x} = (x_1, x_2, x_3)$ denotes a point in Ω. Here the $\mathcal{K}^k$: $\mathcal{D} \rightarrow \mathbb{R}^{4\times 4}$, $1 \leq k \leq K_{dom}$, are symmetric positive definite matrices given by

$$\mathcal{K}^k(\boldsymbol{\mu}) = J^{aff,k}(\boldsymbol{\mu})\, \mathcal{G}^k(\boldsymbol{\mu})\, \mathcal{K}_{o,k}(\boldsymbol{\mu})(\mathcal{G}^k(\boldsymbol{\mu}))^T, \quad \forall\, 1 \leq k \leq K_{dom} \ , \tag{1.58}$$

the $\mathcal{G}^k$: $\mathcal{D} \rightarrow \mathbb{R}^{4\times 4}$, $1 \leq k \leq K_{dom}$, are given by

$$\mathcal{G}^k(\boldsymbol{\mu}) = \left(\begin{array}{ccc} & D^{aff,k}(\boldsymbol{\mu}) & \begin{array}{c} 0 \\ 0 \end{array} \\ 0 \quad & 0 & 1 \end{array} \right) \ ; \tag{1.59}$$

$J^{aff,k}(\boldsymbol{\mu})$ and $D^{aff,k}(\boldsymbol{\mu}), 1 \leq k \leq K_{dom}$, are given by (1.51) and (1.52), respectively; and ^{T} denotes transpose.

Similarly, the transformed linear form can be expressed as

$$f(v) = \sum_{k=1}^{K_{dom}} \int_{\Omega^k} \mathcal{F}^k(\boldsymbol{\mu}) v \ .$$

Here $\mathcal{F}^k$: $\mathcal{D} \rightarrow \mathbb{R}$, $1 \leq k \leq K_{dom}$, is given by

$$\mathcal{F}^k = J^{aff,k}(\boldsymbol{\mu}) \mathcal{F}_{o,k}(\boldsymbol{\mu}), \quad \forall \ \ 1 \leq k \leq K_{dom} \ .$$

We note that, in general, the $\mathcal{K}^k(\boldsymbol{\mu})$ and $\mathcal{F}^k(\boldsymbol{\mu})$, $1 \leq k \leq K_{dom}$, will be different for each subdomain Ω^k. The differences can arise either due to property variation or to geometry variation, or both. We thus require, as already indicated earlier, that the FE approximation be built upon a *subtriangulation* of the RB triangulation: discontinuities in PDE coefficients are thereby restricted to element edges to ensure (more) rapid convergence; and identification/extraction of the terms in the affine expansion (1.13) is more readily effected, as we now discuss.

Affine Form

We focus here on a, though f admits a similar treatment. We simply expand the form (1.57) by considering in turn each subdomain Ω^k and each entry of the diffusivity tensor $\mathcal{K}_{ij}^k$, $1 \le i,j \le 3$, $1 \le k \le K_{dom}$. Thus,

$$\begin{aligned} a(w,v;\boldsymbol{\mu}) = \mathcal{K}_{11}^1(\boldsymbol{\mu}) \int_{\Omega^1} \frac{\partial w}{\partial x_1}\frac{\partial v}{\partial x_1} + \\ \mathcal{K}_{12}^1(\boldsymbol{\mu}) \int_{\Omega^1} \frac{\partial w}{\partial x_1}\frac{\partial v}{\partial x_2} + \cdots + \mathcal{K}_{44}^{K_{dom}}(\boldsymbol{\mu}) \int_{\Omega^{K_{dom}}} wv \ . \end{aligned} \tag{1.60}$$

We can then identify each component in the affine expansion: for each term in (1.60), the pre-factor represents $\Theta^q(\boldsymbol{\mu})$, while the integral represents a^q.

Taking into account the symmetry of the bilinear form, there are (at most) $Q = 10K_{dom}$ terms in the affine expansion. Indeed, since the form is symmetric, the entries $\mathcal{K}_{ij}^k$ and $\mathcal{K}_{ji}^k$, $i \neq j$ of the matrix $\mathcal{K}^k$ for $1 \le k \le K_{dom}$ multiply the same integral then these two factors can be assembled. The $\theta^q(\boldsymbol{\mu})$ are given by (for the obvious numbering scheme) $\theta^1(\boldsymbol{\mu}) = \mathcal{K}_{11}^1(\boldsymbol{\mu}), \ldots, \theta^Q(\boldsymbol{\mu}) = \mathcal{K}_{33}^{K_{dom}}(\boldsymbol{\mu})$; the $a^q(w,v)$ are given by

$$\begin{aligned} a^1(w,v) &= \int_{\Omega^1} \frac{\partial w}{\partial x_1}\frac{\partial v}{\partial x_1} \ , \\ a^2(w,v) &= \int_{\Omega^1} \frac{\partial w}{\partial x_1}\frac{\partial v}{\partial x_2} \ , \\ &\vdots \\ a^Q(w,v) &= \int_{\Omega^{K_{dom}}} wv \ . \end{aligned}$$

1.5 A posteriori error bound

A posteriori error bounds are one of the most important part of the reduced basis methodology. They are crucial for both efficiency and reliability of reduced basis approximations. As regards *efficiency*, error bounds play a role in Offline and Online stage. In the Greedy algorithm for example, the application of error bounds permits larger training sample at reduced Offline computational cost. Hence, we have a better accuracy of the reduced basis approximation which can be obtained with a smaller number N of basis functions, and hence we have a reduced Online cost. In short, a posteriori

error estimation permits us to control the error which in turn permits us to minimize the computational effort (see [15]).

As regards *reliability*, our Offline sampling procedures can not be exhaustive. For a large number of parameters P, there is a large portion of the parameter space $\mathcal{D}$ which remains unexplored. So, the error of a large parts of our parameter domain $\mathcal{D}$ is uncharacterised. The a posteriori error bounds permit to bound the error for all new value of parameter $\boldsymbol{\mu} \in \mathcal{D}$. We can be sure that constraints are satisfied, feasibility conditions are verified, and prognoses are valid (in each case not only for the reduced basis approximation but for the truth finite element solution). So we do not loose any confidence in the solution compared to the underlying FE solution while exploiting the rapid predictive power of the RB approximation (see [20]).

As mentioned earlier, the a posteriori error bound must be *rigorous* (greater or equal to the true error) for all N and all parameters values in the parameter domain $\mathcal{D}$. Non-rigorous error indicators may suffice for adaptivity, but not for reliability. Second, the bound must be reasonably *sharp*. An overly conservative error bound can yield inefficient approximations, typically for N too large. Third, we require *efficiency*, i.e, the cost of the Online evaluation and storage must be $\mathcal{N}$-independent and should be comparable to the cost of the RB output prediction (see [15]).

1.5.1 Preliminaries

First of all, let us introduce some quantities that will be useful for the next.

We define the *residual* $r : \mathcal{D} \longrightarrow \left(X^{\mathcal{N}}\right)'$ as

$$r(v; \boldsymbol{\mu}) = f(v) - a(u_n^{\mathcal{N}}, v, \boldsymbol{\mu}), \tag{1.61}$$

where $\left(X^{\mathcal{N}}\right)'$ is the *dual* space of $X^{\mathcal{N}}$. We also introduce the function $\hat{e} : \mathcal{D} \longrightarrow X^{\mathcal{N}}$, the Riesz representation of $r(v, \boldsymbol{\mu})$

$$\left(\hat{e}(\boldsymbol{\mu}), v\right)_X = r(v, \boldsymbol{\mu}) \quad \forall v \in X^{\mathcal{N}}. \tag{1.62}$$

Introducing the *error* $e^{\mathcal{N}}(\boldsymbol{\mu}) = u^{\mathcal{N}} - u_N^{\mathcal{N}} \in X^{\mathcal{N}}$, we have from (1.62) and (1.61) that

$$a(e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = r(v, \boldsymbol{\mu}) = \left(\hat{e}(\boldsymbol{\mu}), v\right)_X, \quad \forall v \in X. \tag{1.63}$$

Indeed, it follows directly from the problem statements for $u^{\mathcal{N}}(\boldsymbol{\mu})$ (1.15) and $u_N^{\mathcal{N}}(\boldsymbol{\mu})$ (1.25).

We still introduce the *dual norm*

$$\|r(:,\boldsymbol{\mu})\|_{X'} = \sup_{v\in X} \frac{r(v,\boldsymbol{\mu})}{\|v\|_X} = \|\hat{e}(\boldsymbol{\mu})\|_X \,. \tag{1.64}$$

Note that the second equality follows from the Riesz representation theorem. This definition trough the Riesz representation is crucial for the Offline-Online procedure which will be developed below.

We recall the definition of the coercivity and continuity "constants" (see (1.20)-(1.21) and (1.22)-(1.23))

$$\alpha^e(\boldsymbol{\mu}) = \inf_{w\in X} \frac{a(w,w;\boldsymbol{\mu})}{\|w\|_X^2}, \qquad \alpha^{\mathcal{N}}(\boldsymbol{\mu}) = \inf_{w\in X^{\mathcal{N}}} \frac{a(w,w;\boldsymbol{\mu})}{\|w\|_X^2}, \tag{1.65}$$

$$\gamma^e(\boldsymbol{\mu}) = \sup_{w\in X^e}\sup_{v\in X^e} \frac{a(w,v;\boldsymbol{\mu})}{\|w\|_X\,\|v\|_X}, \tag{1.66}$$

$$\gamma^{\mathcal{N}}(\boldsymbol{\mu}) = \sup_{w\in X^{\mathcal{N}}}\sup_{v\in X^{\mathcal{N}}} \frac{a(w,v;\boldsymbol{\mu})}{\|w\|_X\,\|v\|_X}, \tag{1.67}$$

such that

$$\alpha^{\mathcal{N}}(\boldsymbol{\mu}) \geq \alpha^e(\boldsymbol{\mu}) \geq \alpha_0^e > 0, \quad \forall \boldsymbol{\mu}\in\mathcal{D}.$$

In Section 1.5.3, we will give an algorithm to find *lower bound* for $\alpha^{\mathcal{N}}(\boldsymbol{\mu})$, i.e a function $\alpha_{LB}^{\mathcal{N}} : \mathcal{D} \longrightarrow \mathbb{R}$ such that

$$0 < \alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}) \leq \alpha^{\mathcal{N}}(\boldsymbol{\mu}), \quad \forall \boldsymbol{\mu}\in\mathcal{D} \tag{1.68}$$

and such that the evaluation $\boldsymbol{\mu} \to \alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})$ is independent of $\mathcal{N}$.

We define now the *energy*, the *output* and *the relative output error bound estimators* as

$$\Delta_N^{en}(\boldsymbol{\mu}) = \frac{\|\hat{e}(\boldsymbol{\mu})\|_X}{(\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}))^{\frac{1}{2}}}, \tag{1.69}$$

$$\Delta_N^{s}(\boldsymbol{\mu}) = \frac{\|\hat{e}(\boldsymbol{\mu})\|_X^2}{\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})}, \tag{1.70}$$

$$\Delta_N^{s,rel}(\boldsymbol{\mu}) = \frac{\|\hat{e}(\boldsymbol{\mu})\|_X}{\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}) s_N^{\mathcal{N}}(\boldsymbol{\mu})} = \frac{\Delta_N^s(\boldsymbol{\mu})}{s_N^{\mathcal{N}}(\boldsymbol{\mu})}, \tag{1.71}$$

respectively (see [15]).

We introduce now the *effectivities* of these errors

$$\eta_N^{en}(\boldsymbol{\mu}) = \frac{\Delta_N^{en}}{\|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}}, \tag{1.72}$$

$$\eta_N^{s}(\boldsymbol{\mu}) = \frac{\Delta_N^{s}(\boldsymbol{\mu})}{s^{\mathcal{N}}(\boldsymbol{\mu}) - s_N^{\mathcal{N}}(\boldsymbol{\mu})}, \tag{1.73}$$

and

$$\eta_N^{s,rel}(\boldsymbol{\mu}) = \frac{\Delta_N^{s,rel}(\boldsymbol{\mu})}{(s^{\mathcal{N}}(\boldsymbol{\mu}) - s_N^{\mathcal{N}}(\boldsymbol{\mu}))/s^{\mathcal{N}}(\boldsymbol{\mu})}. \tag{1.74}$$

The effectivities are a measure of the quality of the estimators. We give a proposition that show that these estimators are *rigorous* and *sharp*.

Proposition 1.13. See [20]. *For* $N = 1, 2, \ldots$

$$1 \leq \eta_N^{en}(\boldsymbol{\mu}) \leq \sqrt{\frac{\gamma^e(\boldsymbol{\mu})}{\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})}}, \quad \forall \boldsymbol{\mu} \in \mathcal{D} \tag{1.75}$$

$$1 \leq \eta_N^{s}(\boldsymbol{\mu}) \leq \frac{\gamma^e(\boldsymbol{\mu})}{\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})}, \quad \forall \boldsymbol{\mu} \in \mathcal{D} \tag{1.76}$$

Proof. We first prove that if (1.75) is true then (1.76) is also true. Indeed, since $s^{\mathcal{N}}(\boldsymbol{\mu}) - s_N^{\mathcal{N}}(\boldsymbol{\mu}) = \|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}^2$ and since $\Delta_N^s(\boldsymbol{\mu}) = (\Delta_N^{en}(\boldsymbol{\mu}))^2$, we have that

$$\eta_N^s = \frac{\Delta_N^{s,rel}(\boldsymbol{\mu})}{(s^{\mathcal{N}}(\boldsymbol{\mu}) - s_N^{\mathcal{N}}(\boldsymbol{\mu}))/s^{\mathcal{N}}(\boldsymbol{\mu})} = \frac{(\Delta_N^{en}(\boldsymbol{\mu}))^2}{\|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}^2} = (\eta_N^{en})^2.$$

So the result follows.

We now prove (1.75). It follows from (1.63) for $v = e(\boldsymbol{\mu})$ and the Cauchy-Schwarz inequality that

$$\|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}^2 \leq \|\hat{e}(\boldsymbol{\mu})\|_X \|e(\boldsymbol{\mu})\|_X . \tag{1.77}$$

Moreover by the coercivity of the bilinear form

$$\left(\alpha^{\mathcal{N}}(\boldsymbol{\mu})\right)^{\frac{1}{2}} \|e(\boldsymbol{\mu})\|_X \leq \sqrt{a(e(\boldsymbol{\mu}), e(\boldsymbol{\mu}); \boldsymbol{\mu})} = \|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}$$

and hence from (1.77) we obtain

$$\left(\alpha^{\mathcal{N}}(\boldsymbol{\mu})\right)^{\frac{1}{2}} \frac{\|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}^2}{\|\hat{e}(\boldsymbol{\mu})\|_X} \leq \|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}$$

and so we have

$$\|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}} \leq \Delta_N^{en},$$

and hence $\eta_N^{en}(\boldsymbol{\mu}) \geq 1$.

We provide now the proof for the upper bound. For $v = \hat{e}(\boldsymbol{\mu})$ and using (1.63) and the Cauchy Schwarz inequality, we have

$$\|\hat{e}(\boldsymbol{\mu})\|_X^2 \leq \|\hat{e}(\boldsymbol{\mu})\|_{\boldsymbol{\mu}} \|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}. \tag{1.78}$$

Moreover from the definition of the continuity constant

$$\|\hat{e}(\boldsymbol{\mu})\|_{\boldsymbol{\mu}} \leq (\gamma^e(\boldsymbol{\mu}))^{\frac{1}{2}} \|\hat{e}(\boldsymbol{\mu})\|_X$$

and from (1.78)

$$\begin{aligned}
\eta_N^{en} &= \frac{\Delta_N^{en}(\boldsymbol{\mu})}{\|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}} = \frac{(\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}))^{-\frac{1}{2}} \|\hat{e}(\boldsymbol{\mu})\|_X}{\|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}} = \frac{(\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}))^{-\frac{1}{2}} \|\hat{e}(\boldsymbol{\mu})\|_X^2}{\|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}} \|\hat{e}(\boldsymbol{\mu})\|_X} \\
&\leq \frac{\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}))^{-\frac{1}{2}} \|\hat{e}(\boldsymbol{\mu})\|_{\boldsymbol{\mu}} \|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}}{\|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}} \|\hat{e}(\boldsymbol{\mu})\|_X} \leq \alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}))^{-\frac{1}{2}} (\gamma^e(\boldsymbol{\mu}))^{\frac{1}{2}} \\
&= \sqrt{\frac{\gamma^e(\boldsymbol{\mu})}{\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})}}.
\end{aligned}$$

□

Remark 1.14. See [20]. Note that we have the following estimation for the upper bound in (1.75)

$$\sqrt{\frac{\gamma^e(\boldsymbol{\mu})}{\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})}} \leq \sqrt{\frac{\alpha^{\mathcal{N}}(\boldsymbol{\mu})}{\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})}} \sqrt{\frac{\gamma^e(\boldsymbol{\mu})}{\alpha^e(\boldsymbol{\mu})}}, \quad \forall \boldsymbol{\mu} \in \mathcal{D} \tag{1.79}$$

1.5.2 Offline-Online procedure

The main component of the error bound is the computation of the dual norm of the residual $\|\hat{e}(\boldsymbol{\mu})\|_X$. To develop the Offline-Online procedure, we introduce the residual expansion

$$r(v;\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \theta_f^q(\boldsymbol{\mu}) f^q(v) - \sum_{q=1}^{Q_a} \sum_{n=1}^{N} \theta_a^q(\boldsymbol{\mu}) u_{N\,n}^{\mathcal{N}}(\boldsymbol{\mu}) a^q(\xi_n, v), \quad \forall v \in X. \tag{1.80}$$

This expansion follows from our affine assumption (1.13) and from the RB representation $u_N^{\mathcal{N}}(\boldsymbol{\mu}) = \sum_{n=1}^{N} u_{N\,n}\xi_n$.

Moreover, we have from (1.63) and (1.80) that

$$(\hat{e}(\boldsymbol{\mu}), v)_X = \sum_{q=1}^{Q_f} \theta_f^q(\boldsymbol{\mu}) f^q(v) - \sum_{q=1}^{Q_a}\sum_{n=1}^{N} \theta_a^q(\boldsymbol{\mu}) u_{N\,n}^{\mathcal{N}}(\boldsymbol{\mu}) a^q(\xi_n, v), \quad \forall v \in X. \tag{1.81}$$

Consequently,

$$\hat{e}(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \theta_f^q(\boldsymbol{\mu})\mathcal{F}^q + \sum_{q=1}^{Q_a}\sum_{n=1}^{N} \theta_a^q(\boldsymbol{\mu}) u_{N\,n}^{\mathcal{N}}(\boldsymbol{\mu})\mathcal{A}_n^q, \tag{1.82}$$

where $\forall v \in X^{\mathcal{N}}$

$$(\mathcal{F}^q, v)_X = f^q(v), \qquad 1 \le q \le Q_f \tag{1.83}$$

$$(\mathcal{A}_n^q)_X = -a^q(\xi^n, v), \qquad 1 \le n \le N,\ 1 \le q \le Q_a. \tag{1.84}$$

Remark that (1.83) and (1.84) are parameter-independent Poisson-like problems. So, $\mathcal{F}^q$ and $\mathcal{A}_n^q$ are computed Offline. Then, we obtain the following expansion

$$\begin{aligned}
\|\hat{e}(\boldsymbol{\mu})\|_X^2 &= \left(\sum_{q=1}^{Q_f} \theta_f^q(\boldsymbol{\mu})\mathcal{F}^q + \sum_{q=1}^{Q_a}\sum_{n=1}^{N} \theta_a^q(\boldsymbol{\mu}) u_{N\,n}^{\mathcal{N}}(\boldsymbol{\mu})\mathcal{A}_n^q, \bullet\right)_X \\
&= \sum_{q=1}^{Q_f}\sum_{q'=1}^{Q_f} \theta_f^q(\boldsymbol{\mu})\theta_f^{q'}(\boldsymbol{\mu}) \left(\mathcal{F}^q, \mathcal{F}^{q'}\right)_X + \sum_{q=1}^{Q_a}\sum_{n=1}^{N} \theta_a^q(\boldsymbol{\mu}) u_{N\,n}^{\mathcal{N}}(\boldsymbol{\mu})\Big\{ \\
&\quad 2\sum_{q'=1}^{Q_f} \theta_f^{q'}(\boldsymbol{\mu}) \left(\mathcal{F}^{q'}, \mathcal{A}_n^q\right)_X + \sum_{q'=1}^{Q_a}\sum_{n'=1}^{N} \theta_a^{q'}(\boldsymbol{\mu}) u_{N\,n'}(\boldsymbol{\mu}) \left(\mathcal{A}_n^q, \mathcal{A}_{n'}^{q'}\right)_X \Big\}.
\end{aligned} \tag{1.85}$$

The Offline-Online procedure is clear. In the Offline stage, we first compute $\mathcal{F}^q$, $1 \le q \le Q_f$ and $\mathcal{A}_n^q$, $1 \le q \le Q_a$ and $1 \le n \le N_{max}$. After that, we form and store the quantities

$$\left(\mathcal{F}^{q_f}, \mathcal{F}^{q_f'}\right)_X, \ \left(\mathcal{F}^{q_f'}, \mathcal{A}_n^{q_a}\right)_X, \ \left(\mathcal{A}_n^{q_a}, \mathcal{A}_{n'}^{q_a'}\right)_X$$

for $1 \leq q_f,\, q_f^{'} \leq Q_f$, $1 \leq q_a,\, q_a^{'} \leq Q_a$ and $1 \leq n,\, n^{'} \leq N_{max}$.

In the Online stage we evaluate the sum (1.85).

1.5.3 Coercivity lower bound

In this section we present the *Successive Constraint Method* (SCM) that is an algorithm to construct lower bounds for the coercivity (and in the non-coercive case, inf-sup stability) constants (see [20], [11] and [15]). This method is also based on an Offline-Online procedure and it reduces considerably the Online calculation.

We recall the definition of the FE coercivity constant (1.1)

$$\alpha^{\mathcal{N}}(\boldsymbol{\mu}) = \inf_{w \in X^{\mathcal{N}}} \frac{a(w, w; \boldsymbol{\mu})}{\|w\|_X^2}, \qquad \forall \boldsymbol{\mu} \in \mathcal{D}$$

and we assume that $\alpha^{\mathcal{N}}(\boldsymbol{\mu}) > 0$, $\forall \boldsymbol{\mu} \in \mathcal{D}$ so a is coercive. However, the SCM can be extended to the non-symmetric operator and for non-coercive problems.

As anticipated previously, we want to compute $\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})$ such that $0 < \alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}) < \alpha^{\mathcal{N}}(\boldsymbol{\mu})$, $\forall \boldsymbol{\mu} \in \mathcal{D}$ and the evaluation $\boldsymbol{\mu} \rightarrow \alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})$ must be $\mathcal{N}$-independent. Let us introduce an *objective* function $\mathcal{J}^{obj} : \mathcal{D} \times \mathbb{R}^{Q_a} \longrightarrow \mathbb{R}$ given by

$$\mathcal{J}^{obj}(\boldsymbol{\mu}, y) = \sum_{q=1}^{Q_a} \theta_a^q(\boldsymbol{\mu}) y_q, \tag{1.86}$$

where $y = (y_1, \ldots, y_{Q_a})$. The coercivity constant may be express as

$$\alpha^{\mathcal{N}}(\boldsymbol{\mu}) = \inf_{y \in \mathcal{Y}} \mathcal{J}^{obj}(\boldsymbol{\mu}, y), \tag{1.87}$$

where the set $\mathcal{Y} \subset \mathbb{R}^{Q_a}$ is defined by

$$\mathcal{Y} = \left\{ y \in \mathbb{R}^{Q_a} \,|\, \exists w_y \in X^{\mathcal{N}} \; s.t. \; y_q = \frac{a^q(w_y, w_y)}{\|w_y\|_X^2},\; 1 \leq q \leq Q_a \right\}. \tag{1.88}$$

We next introduce the *continuity constraint* box

$$\mathcal{B} = \prod_{q=1}^{Q_a} \left[\inf_{w \in X^{\mathcal{N}}} \frac{a^q(w, w)}{\|w\|_X^2},\; \sup_{w \in X^{\mathcal{N}}} \frac{a^q(w, w)}{\|w\|_X^2} \right], \tag{1.89}$$

that is *bounded* from the continuity hypothesis. Finally, we introduce the *coercivity constraint* sample as

$$\mathcal{C}_J = \left\{ \boldsymbol{\mu}^1_{SCM} \in \mathcal{D}, \ldots, \boldsymbol{\mu}^J_{SCM} \in \mathcal{D} \right\}. \tag{1.90}$$

We denote by $C_J^{M,\boldsymbol{\mu}}$ the set of $M(\geq 1)$ points in $\mathcal{C}_J$ closest to a given $\boldsymbol{\mu}$ (in the Euclidean norm). If $M > J$ then $C_J^{M,\boldsymbol{\mu}} = \mathcal{C}_J$.

Lower bound

The idea is to construct the *lower bound* set $\mathcal{Y}_{LB}(\boldsymbol{\mu}, \mathcal{C}_J, M)$ for given $\mathcal{C}_J, M \in \mathbb{N}$ and any $\boldsymbol{\mu} \in \mathcal{D}$ such that $\mathcal{Y} \subset \mathcal{Y}_{LB}(\boldsymbol{\mu}, \mathcal{C}_J, M)$. This set is cheap to compute and is defined as

$$\mathcal{Y}_{LB}(\boldsymbol{\mu}, \mathcal{C}_J, M) = \left\{ y \in \mathbb{R}^{Q_a} \,|\, y \in \mathcal{B},\ \sum_{q=1}^{Q_a} \theta^q(\boldsymbol{\mu}') y_q \geq \alpha^{\mathcal{N}}(\boldsymbol{\mu}'),\ \forall \boldsymbol{\mu}' \in \mathcal{C}_J^{M,\boldsymbol{\mu}} \right\}. \tag{1.91}$$

We obtain the wanted result

Lemma 1.15. *Given $\mathcal{C}_J \subset \mathcal{D}$ and $M \in \mathbb{N}$*

$$\mathcal{Y} \subset \mathcal{Y}_{LB}(\boldsymbol{\mu}, \mathcal{C}_J, M), \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{1.92}$$

Proof. Let $y \in \mathcal{Y}$. We want to show that $y \in \mathcal{Y}_{LB}$.

- Since $y \in \mathcal{Y}$, it exists $w_y \in X^{\mathcal{N}}$ such that $y_q = \frac{a^q(w_y, w_y)}{\|w_y\|_X^2}$, $1 \leq q \leq Q_a$. Moreover

$$\inf_{w \in X^{\mathcal{N}}} \frac{a^q(w,w)}{\|w\|_X^2} \leq \underbrace{\frac{a^q(w_y, w_y)}{\|w_y\|_X^2}}_{=y^q} \leq \sup_{w \in X^{\mathcal{N}}} \frac{a^q(w,w)}{\|w\|_X^2},$$

i.e. $\mathcal{Y} \in \mathcal{B}$.

- We have also that, $\forall \boldsymbol{\mu} \in \mathcal{D}$

$$\sum_{q=1}^{Q} \theta^q(\boldsymbol{\mu}) \frac{a^q(w_y, w_y)}{\|w_y\|_X^2} = \frac{a(w_y, w_y; \boldsymbol{\mu})}{\|w_y\|_X^2} \geq \alpha^{\mathcal{N}}(\boldsymbol{\mu}).$$

Consequently, $y \in \mathcal{Y}_{LB}$. □

Finally, we define our *coercivity lower bound* as

$$\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}, \mathcal{C}_J, M) = \min_{y \in \mathcal{Y}_{LB}(\boldsymbol{\mu}, \mathcal{C}_J, M)} \mathcal{J}^{obj}(\boldsymbol{\mu}, y). \tag{1.93}$$

Moreover, we have:

Proposition 1.16. *For given $\mathcal{C}_J \subset \mathcal{D}$, $M \in \mathbb{N}$,*

$$\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}, \mathcal{C}_J, M) \leq \alpha^{\mathcal{N}}(\boldsymbol{\mu}), \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{1.94}$$

Proof. It is a direct consequence of (1.87),(1.92) and (1.93). □

The most important thing here is that the operation count to evaluate $\boldsymbol{\mu} \to \alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu})$ is independent of $\mathcal{N}$.

Upper Bound

The SCM needs the construction of an *upper bound* $\alpha_{UB}^{\mathcal{N}}(\boldsymbol{\mu}, \mathcal{C}_J, M)$ for given $\mathcal{C}_J$, $M \in \mathbb{N}$ and any $\boldsymbol{\mu} \in \mathcal{D}$. We proceed in a similar way we did for the lower bound. We construct a set easy to compute $\mathcal{Y}_{UB}(\boldsymbol{\mu}, \mathcal{C}_J, M) \in \mathbb{R}^{Q_a}$ as

$$\mathcal{Y}_{UB}(\boldsymbol{\mu}, \mathcal{C}_J, M) = \left\{ y^*(\boldsymbol{\mu}') \,|\, \boldsymbol{\mu}' \in \mathcal{C}_J^{M, \boldsymbol{\mu}} \right\}, \tag{1.95}$$

where

$$y^*(\boldsymbol{\mu}) = arg \inf_{y \in \mathcal{Y}} \mathcal{J}^{obj}(\boldsymbol{\mu}, y).$$

We then define the upper bound as

$$\alpha_{UB}^{\mathcal{N}}(\boldsymbol{\mu}, \mathcal{C}_J, M) = \min_{y \in \mathcal{Y}_{UB}(\boldsymbol{\mu}, \mathcal{C}_J, M)} \mathcal{J}^{obj}(\boldsymbol{\mu}, y). \tag{1.96}$$

We can show the followings results

Lemma 1.17. *For given $\mathcal{C}_J$, $M \in \mathbb{N}$*

$$\mathcal{Y}_{UB}(\boldsymbol{\mu}, \mathcal{C}_J, M) \subset \mathcal{Y} \quad \forall \boldsymbol{\mu} \in \mathcal{D} \tag{1.97}$$

Proposition 1.18. *Given $\mathcal{C}_J$, $M \in \mathbb{N}$*

$$\alpha_{UB}^{\mathcal{N}}(\boldsymbol{\mu}) \geq \alpha^{\mathcal{N}}(\boldsymbol{\mu}), \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{1.98}$$

1.5.4 The successive constraint method

The task of the SCM is given a sample set

$$\Xi_{train,\,SCM} = \left\{\boldsymbol{\mu}^1_{train,\,SCM}, \ldots, \boldsymbol{\mu}^{n_{train,\,SCM}}_{train,\,SCM}\right\} \subset \mathcal{D}$$

of $n_{train,\,SCM}$ parameter points, to select (greedy) parameters in $\Xi_{train,\,SCM}$ and construct the sets $\mathcal{C}_1 = \{\boldsymbol{\mu}^1_{SCM}\}, \ldots, \mathcal{C}_{J_{max}} = \{\boldsymbol{\mu}^1_{SCM}, \ldots, \boldsymbol{\mu}^{J_{max}}_{SCM}\}$. We now give the algorithm.

Given a tolerance $\varepsilon_{SCM} \in (0,1)$ we choose for $J = 1$, the set $\mathcal{C}_1 = \{\boldsymbol{\mu}^1_{SCM}\}$ arbitrarily. We then perform

$$\begin{aligned}
&\text{While} && \max_{\boldsymbol{\mu}\in\Xi_{train,\,SCM}} \left[\frac{\alpha^{\mathcal{N}}_{UB}(\boldsymbol{\mu}, \mathcal{C}_J, M) - \alpha^{\mathcal{N}}_{LB}(\boldsymbol{\mu}, \mathcal{C}_J, M)}{\alpha^{\mathcal{N}}_{UB}(\boldsymbol{\mu}, \mathcal{C}_J, M)}\right] > \varepsilon_{SCM};\\
&\boldsymbol{\mu}^{J+1}_{SCM} &&= arg \max_{\boldsymbol{\mu}\in\Xi_{train,\,SCM}} \left[\frac{\alpha^{\mathcal{N}}_{UB}(\boldsymbol{\mu}, \mathcal{C}_J, M) - \alpha^{\mathcal{N}}_{LB}(\boldsymbol{\mu}, \mathcal{C}_J, M)}{\alpha^{\mathcal{N}}_{UB}(\boldsymbol{\mu}, \mathcal{C}_J, M)}\right];\\
&\mathcal{C}_{J+1} &&= \mathcal{C}_J + \boldsymbol{\mu}^{J+1}_{SCM};\\
&J &&= J + 1;\\
&\text{end}\\
&\text{Set} && J_{max} = J;
\end{aligned}$$

Remark 1.19. - Note that we choose α_{UB} in the denominator and not α_{LB} since α_{LB} may be negative or zero.

- the choice of stopping criterion ε_{SCM} permits to estimate the upper bound for the output effectivity in (1.79) by $\frac{1}{1-\varepsilon_{SCM}}$. Indeed,

$$\begin{aligned}
\frac{\alpha^{\mathcal{N}}(\boldsymbol{\mu})}{\alpha^{\mathcal{N}}_{LB}(\boldsymbol{\mu}, \mathcal{C}_{J_{max}}, M)} &= \frac{\alpha^{\mathcal{N}}(\boldsymbol{\mu})}{\alpha^{\mathcal{N}}_{UB}(\boldsymbol{\mu}, \mathcal{C}_{J_{max}}, M) - (\alpha^{\mathcal{N}}_{UB}(\boldsymbol{\mu}, \mathcal{C}_{J_{max}}, M) - \alpha^{\mathcal{N}}_{LB}(\boldsymbol{\mu}, \mathcal{C}_{J_{max}}, M))}\\
&\leq \frac{\alpha^{\mathcal{N}}(\boldsymbol{\mu})}{\alpha^{\mathcal{N}}_{UB}(\boldsymbol{\mu}, \mathcal{C}_{J_{max}}, M)} \frac{1}{1-\varepsilon_{SCM}}\\
&\leq \frac{1}{1-\varepsilon_{SCM}}, \quad \forall \boldsymbol{\mu} \in \Xi_{train,\,SCM}.
\end{aligned}$$

And so we have that

$$\eta_N^{en}(\boldsymbol{\mu}) \leq \sqrt{\frac{\gamma^e(\boldsymbol{\mu})}{(1-\varepsilon_{SCM})\alpha^e(\boldsymbol{\mu})}}, \quad \forall \boldsymbol{\mu} \in \mathcal{D}.$$

Usually we set $\varepsilon_{SCM} = 0.75$ which is a crude lower bound but that has little deleterious effect on our error bounds.

- For the SCM algorithm, in the case where a is not symmetric, we have to consider only the symmetric part of a in the lower bound calculation.

Offline-Online procedure

In the Offline stage, we have to construct the set $\mathcal{B}$ that needs $2Q_a$ eigenproblems over $X^{\mathcal{N}}$ and we need to solve J_{max} eigenproblems to form the set $\left\{\alpha^{\mathcal{N}}(\boldsymbol{\mu}' \,|\, \boldsymbol{\mu}' \in \mathcal{C}_{J_{max}})\right\}$. We still have $J_{max}Q_a$ inner products over $X^{\mathcal{N}}$ to form $\{y^*(\boldsymbol{\mu}' \,|\, \boldsymbol{\mu}' \in \mathcal{C}_{J_{max}})\}$ and $n_{train,\,SCM}J_{max}$ linear optimization problems of size $2Q_a + M$ to perform the arg max. The computational cost depends of course of $\mathcal{N}$. However, there is not "cross-terms" like $\mathcal{O}(n_{train}\mathcal{N})$. So, we can choose n_{train} and $\mathcal{N}$ very large. The eigenproblems associated with the calculation of the $\alpha^{\mathcal{N}}(\boldsymbol{\mu}')$, $\boldsymbol{\mu}' \in \mathcal{C}_j$ can be treated very efficiently bay the Lancsoz method. In particular, for the choice of the parameter τ in (1.18)

$$\tau = \inf_{w \in X^{\mathcal{N}}} \frac{a(w, w; \overline{\boldsymbol{\mu}})}{(w, w)_{L^2(\Omega)}}, \tag{1.99}$$

it can be shown that $\lambda_{min} = \lambda_1$ is well separated from λ_2, where these λ_i are the eigenvalues of the problem (1.4). The latter ensures rapid convergence of the Lanczos procedure. As anticipated in Remark 1.9 the choice of the norm (1.18) will affect the quality and cost of the a posteriori output error bound. We understand now that the choice of $\overline{\boldsymbol{\mu}}$ affects the effectivity while the choice of τ affects Offline efficiency.

In the Online stage, for each evaluation $\boldsymbol{\mu} \to \alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}, \mathcal{C}_J, M)$, we have $\mathcal{O}(MJ_{max})$ operations to chose J_{max} points in $\mathcal{C}_{J_{max}}$ to determine the set $\mathcal{C}_{J_{max}}^{M,\boldsymbol{\mu}}$. We have $\mathcal{O}((M+1)Q_a)$ operation count to evaluate $\boldsymbol{\mu}' \to \theta_a^q(\boldsymbol{\mu}')$, $1 \leq q \leq Q_a$ and finally extract the selected M members of the pre-computed set $\left\{\alpha^{\mathcal{N}}(\boldsymbol{\mu}' \,|\, \boldsymbol{\mu}' \in \mathcal{C}_J)\right\}$ and solve the resulting linear optimization problem to obtain $\alpha_{LB}^{\mathcal{N}}(\boldsymbol{\mu}, \mathcal{C}_J, M)$. In conclusion, we note that the cost of the Online evaluation is $\mathcal{N}$-independent.

1.6 Non-compliant elliptic problems

In this section, we treat briefly the case of non-compliant elliptic problems. We recall the the formulation of our problem in that case

$$\begin{aligned} & s^e(\boldsymbol{\mu}) = \ell(u^e(\boldsymbol{\mu}); \boldsymbol{\mu}), \\ & \text{where } u^e(\boldsymbol{\mu}) \in X^e(\Omega) \text{ satisfies} \\ & a\left(u^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right) = f(v; \boldsymbol{\mu}), \quad \forall v \in X^e. \end{aligned} \tag{1.100}$$

We assume that $\ell \neq f$ and that a is a coercive not necessarily symmetric bilinear form. We called the problem (1.100) the *Primal* problem.

The RB formulation for the non-compliant state

$$\begin{aligned} & s_N^{\mathcal{N}}(\boldsymbol{\mu}) = \ell(u_N^{\mathcal{N}}(\boldsymbol{\mu}); \boldsymbol{\mu}), \\ & \text{where } u_N^{\mathcal{N}}(\boldsymbol{\mu}) \in X_N^{\mathcal{N}} \text{ satisfies} \\ & a\left(u_N^{\mathcal{N}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right) = f(v; \boldsymbol{\mu}), \quad \forall v \in X_N^{\mathcal{N}}. \end{aligned} \tag{1.101}$$

We can develop an a posteriori error bound for $s_N^{\mathcal{N}}(\boldsymbol{\mu})$.

Proposition 1.20. See [23]. *For any $\boldsymbol{\mu} \in \mathcal{D}$ and $u_N^{\mathcal{N}}(\boldsymbol{\mu})$ and $s_N^{\mathcal{N}}(\boldsymbol{\mu})$ satisfying* (1.101)

$$|s^{\mathcal{N}}(\boldsymbol{\mu}) - s_N^{\mathcal{N}}(\boldsymbol{\mu})| \leq \|\ell\|_{(X^{\mathcal{N}})'} \Delta_N(\boldsymbol{\mu}), \tag{1.102}$$

where

$$\left\|u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})\right\|_X \leq \Delta_N(\boldsymbol{\mu}). \tag{1.103}$$

Moreover,

$$\Delta_N(\boldsymbol{\mu}) = \frac{\|\hat{e}(\boldsymbol{\mu})\|_X}{\alpha_{LB}(\boldsymbol{\mu})}. \tag{1.104}$$

Proof. We have that

$$\begin{aligned} a(e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) &= r(v, \boldsymbol{\mu}) = f(v) - a(u_N^{\mathcal{N}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) \\ &= a(u^{\mathcal{N}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) - a(u_N^{\mathcal{N}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = \left(\hat{e}(\boldsymbol{\mu}), v\right)_X. \end{aligned}$$

Hence for $v = e(\boldsymbol{\mu})$ and using Cauchy-Schwarz and the definition of the coercivity we obtain

$$\alpha_{LB}(\boldsymbol{\mu}) \left\| e(\boldsymbol{\mu}) \right\|_X^2 \leq \left\| \hat{e}(\boldsymbol{\mu}) \right\|_X \left\| e(\boldsymbol{\mu}) \right\|_X$$

$$\iff \qquad \left\| e(\boldsymbol{\mu}) \right\|_X \leq \tfrac{\left\| \hat{e}(\boldsymbol{\mu}) \right\|_X}{\alpha_{LB}(\boldsymbol{\mu})}.$$

Consequently,

$$\begin{aligned} |s^{\mathcal{N}}(\boldsymbol{\mu}) - s_N^{\mathcal{N}}(\boldsymbol{\mu})| &= |\ell(u^{\mathcal{N}}(\boldsymbol{\mu})) - \ell(u_N^{\mathcal{N}}(\boldsymbol{\mu}))| = |\ell(e(\boldsymbol{\mu}))| \\ &= \frac{|\ell(e(\boldsymbol{\mu}))|}{\left\| e(\boldsymbol{\mu}) \right\|_X} \leq \underbrace{\left(\sup_{v \in X^{\mathcal{N}}} \frac{\ell(v)}{\left\| v \right\|_x} \right)}_{\left\| \ell \right\|_{(X^{\mathcal{N}})'}} \left\| e(\boldsymbol{\mu}) \right\|_X \leq \left\| \ell \right\|_{(X^{\mathcal{N}})'} \Delta_N(\boldsymbol{\mu}) \\ &= \Delta_N^{s,nc}(\boldsymbol{\mu}). \end{aligned}$$

□

For a lot of problems (and for many of outputs), we can consider the Primal only. But for many output, we can do better. Indeed, if we solve the Primal only we lose the quadratic convergence for our output in the non-compliant case (see 1.20) while in the compliant case the convergence is quadratic (see Proposition 1.10). Moreover, the effectivities

$$\eta_N^{s,nc} = \frac{\Delta_N^{s,nc}(\boldsymbol{\mu}}{|s^{\mathcal{N}}(\boldsymbol{\mu}) - s_N^{\mathcal{N}}(\boldsymbol{\mu})|} \tag{1.105}$$

can be unbounded.

To remedy these problems, we introduce a *Dual* problem.

$$\begin{aligned} &s_N^{\mathcal{N},du}(\boldsymbol{\mu}) = f(\Psi_{N_{du}}^{\mathcal{N}}(\boldsymbol{\mu}); \boldsymbol{\mu}), \\ &\text{where } \Psi_{N_{du}}^{\mathcal{N}}(\boldsymbol{\mu}) \in X_{N_{du}}^{\mathcal{N},du} \text{ satisfies} \\ &a\left(v, \Psi_{N_{du}}^{\mathcal{N}}(\boldsymbol{\mu}); \boldsymbol{\mu}\right) = -\ell(v; \boldsymbol{\mu}), \quad \forall v \in X_{N_{du}}^{\mathcal{N},du}. \end{aligned} \tag{1.106}$$

where the script du indicates the dual membership. Note that if a is symmetric, we have that $\Psi = -u$.

Remark 1.21. Now, we will omit the superscript $^{\mathcal{N}}$ for more readability. Analogously with the Dual problem we will indicate with the script pr the quantity relative to the Primal problem.

We introduce now the *Primal-Dual output correction* as

$$s_{N_{pr},N_{du}}(\boldsymbol{\mu}) = \ell(u_{N_{pr}}) - r^{pr}(\Psi_{N_{du}};\boldsymbol{\mu}), \tag{1.107}$$

where

$$\begin{aligned} r^{pr}(v;\boldsymbol{\mu}) &= f(v;\boldsymbol{\mu}) - a(u_{N_{pr}},v;\boldsymbol{\mu}), \\ r^{du}(v;\boldsymbol{\mu}) &= -\ell(v) - a(v,\Psi_{N_{du}};\boldsymbol{\mu}). \end{aligned}$$

Offline-Online procedure is the same we made before but now we have to consider also the Dual problem. In this Primal-Dual approach, we recover the quadratic output effect. Indeed,

$$\begin{aligned} |s - s_{N_{pr},N_{du}}| &= \ell(u-u_{N_{pr}}) + r^{pr}(\Psi_{N_{du}};\boldsymbol{\mu}) \\ &= -a(e^{pr},\Psi;\boldsymbol{\mu}) + a(e^{pr},\Psi_{N_{du}};\boldsymbol{\mu}) = -a(e^{pr},e^{du};\boldsymbol{\mu}). \end{aligned}$$

Applying the Galerkin optimality to Primal and Dual we obtain that

$$|s - s_{N_{pr},N_{du}}| \leq C \left(\inf_{w_N^{pr}\in X_{N_{pr}}^{pr}} \|u - w_N^{pr}\|_X \right) \left(\inf_{w_N^{du}\in X_{N_{du}}^{du}} \left\|u - w_N^{du}\right\|_X \right). \tag{1.108}$$

The Offline-Online procedure is similar to the Primal only, however we just have to evaluate the residual dual norm for Primal and Dual. Indeed, we can derive an a posteriori error bound for the output as

$$|s - s_{N_{pr},N_{du}}| \leq \Delta_N^{s,nc}, \tag{1.109}$$

where

$$\Delta_N^{s,nc} = \left\|r_N^{du}\right\|_{(X^{\mathcal{N}})'} \Delta_N(\boldsymbol{\mu}). \tag{1.110}$$

Chapter 2

Overview of the Reduced Basis Method: Parabolic problems

In this chapter, we will treat *linear parabolic* problems. We will focus only on the primal problem. A primal-dual formulation exists and is similar to the elliptic case (see 1.6). The main references for this chapter are [23], [24] and [25]. We recall briefly the main definitions that we need. The parameter domain will be denoted by $\mathcal{D}$ and it is a subset of $\mathbb{R}^P$ for $P \in \mathbb{N}$. The *time domain* shall be denoted by $I = [0, t_f]$ where t_f is the *final time*. The physical domain is denoted by $\Omega \subset \mathbb{R}^d$ with boundary $\partial\Omega$ and where d is the dimension. We may define $X^e = X^e(\Omega)$ the function space such that $(H_0^1(\Omega))^\nu \subset X^e \subset (H^1(\Omega))^\nu$, where $\nu = 1$, respectively d, for scalar, respectively vectorial problems.

2.1 Reduced Basis and a posteriori error bound

We introduce now the weak form of the $\boldsymbol{\mu}$-parametrized linear parabolic PDE:

Given $\boldsymbol{\mu} \in \mathcal{D}$, evaluate:

$$s(t;\boldsymbol{\mu}) = \ell(u^e(t;\boldsymbol{\mu});\boldsymbol{\mu}),$$

$$\text{where } u^e(t;\boldsymbol{\mu}) \in X^e \text{ satisfies} \tag{2.1}$$

$$m\left(\tfrac{\partial u^e}{\partial t}(t;\boldsymbol{\mu}), v;\boldsymbol{\mu}\right) + a\left(u^e(t;\boldsymbol{\mu}), v;\boldsymbol{\mu}\right) = g(t) f(v;\boldsymbol{\mu}), \quad \forall v \in X^e,$$

with initial condition $u^e(\boldsymbol{x}, t = 0;\boldsymbol{\mu}) = u_0^e(\boldsymbol{x};\boldsymbol{\mu})$, where $a(\cdot,\cdot;\boldsymbol{\mu})$ is bilinear, X^e-continuous and coercive, $m(\cdot,\cdot;\boldsymbol{\mu})$ is bilinear, L^2-continuous and coercive

and $f(\cdot;\boldsymbol{\mu})$ is linear bounded. The output functional $\ell(\cdot;\boldsymbol{\mu})$ is linear and bounded while $g(\cdot) \in L^2(0,t_f)$ is a *control input*. Moreover, the forms a, m, f and ℓ are affine in $\boldsymbol{\mu}$, i.e.,

$$a(v,w;\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \theta_a^q(\boldsymbol{\mu}) a^q(w,v), \quad \forall w,v \in X^e, \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{2.2}$$

$$m(v,w;\boldsymbol{\mu}) = \sum_{q=1}^{Q_m} \theta_m^q(\boldsymbol{\mu}) m^q(w,v), \quad \forall w,v \in X^e, \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{2.3}$$

$$\ell(v;\boldsymbol{\mu}) = \sum_{q=1}^{Q_\ell} \theta_\ell^q(\boldsymbol{\mu}) \ell^q(v), \quad \forall v \in X^e, \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{2.4}$$

$$f(v;\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \theta_f^q(\boldsymbol{\mu}) f^q(v), \quad \forall v \in X^e, \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{2.5}$$

$$u_0(\boldsymbol{x};\boldsymbol{\mu}) = \sum_{q=1}^{Q_u} \theta_u^q(\boldsymbol{\mu}) u_0^q(\boldsymbol{x}), \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{2.6}$$

Remark 2.1. We may also consider another type of output

$$s^e(t;\boldsymbol{\mu}) = \int_0^{t_f} h(t,t')\ell(u^e(t';\boldsymbol{\mu});\boldsymbol{\mu})\, dt' \tag{2.7}$$

for $h(t,\cdot) \in L^2(0,t_f)$ and $\ell(\cdot;\boldsymbol{\mu}) \in (X^e)'$, where $(X^e)'$ is the dual space of X^e.

We now discretize the problem (2.1) in space (FE) and in time using a *Euler backward* discretization. We introduce Δt the time step and $n_t = \frac{t_f}{\Delta t}$ the number of time steps. We still define $t^k = k\Delta t$, $0 \le k \le n_t$, $\mathbb{T} = \{t^0, \ldots, t^{n_t}\}$ and $\mathbb{K} = \{1, 2, \ldots, n_t\}$. Let us introduce $X^{\mathcal{N}} \subset X^e$ the finite element space. Then, we obtain the discretized problem, $\forall k \in \mathbb{K}$, $\forall v \in X^{\mathcal{N}}$:

$$s^{\mathcal{N}\,k}(\boldsymbol{\mu}) = \ell(u^{\mathcal{N}\,k}(\boldsymbol{\mu});\boldsymbol{\mu}),$$

where $u^{\mathcal{N}\,k}(\boldsymbol{\mu}) \in X^{\mathcal{N}}$ satisfies

$$m\left(u^{\mathcal{N}\,k}(\boldsymbol{\mu}), v;\boldsymbol{\mu}\right) + \Delta_t a\left(u^{\mathcal{N}\,k}(\boldsymbol{\mu}), v;\boldsymbol{\mu}\right) = \Delta_t g(t^k) f(v;\boldsymbol{\mu}) + m\left(u^{\mathcal{N}\,k-1}(\boldsymbol{\mu}), v;\boldsymbol{\mu}\right), \tag{2.8}$$

with initial condition $u(\boldsymbol{x}, t_0; \boldsymbol{\mu}) = u_0(\boldsymbol{x}; \boldsymbol{\mu})$, where $u_0(\boldsymbol{x}, \boldsymbol{\mu})$ is the L^2-projection of $u_0^e(\boldsymbol{x}; \boldsymbol{\mu})$. Here, $s^{\mathcal{N}\,k}(\boldsymbol{\mu}) \approx s^e(t^k; \boldsymbol{\mu})$ and $u^{\mathcal{N}\,k} \approx u^e(t^k; \boldsymbol{\mu})$.

Now, the theory we introduced for elliptic problem can be used for the parabolic. More precisely, we can express our problem in a parameter-independent domain using affine mappings (see 1.4) and we perform a RB approximation (Section 1.2.3). Then, we obtain the reduced problem: $\forall k \in \mathbb{K}$, $\forall v \in X^{\mathcal{N}}$

$$s_N^{\mathcal{N}\,k}(\boldsymbol{\mu}) = \ell(u_N^{\mathcal{N}\,k}(\boldsymbol{\mu}); \boldsymbol{\mu}),$$

where $u_N^{\mathcal{N}\,k}(\boldsymbol{\mu}) \in X_N^{\mathcal{N}}$ satisfies

$$m\left(u_N^{\mathcal{N}\,k}(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right) + \Delta_t a\left(u_N^{\mathcal{N}\,k}(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right) = \Delta_t g(t^k) f(v; \boldsymbol{\mu}) + m\left(u_N^{\mathcal{N}\,k-1}(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right), \tag{2.9}$$

where $X_N^{\mathcal{N}}$ is the reduced basis space and $dim(X_N^{\mathcal{N}}) = N << \mathcal{N}$.
Now, as in the elliptic case, we compute an $\alpha_{LB}(\boldsymbol{\mu})$, such that

$$0 < \alpha_{LB}(\boldsymbol{\mu}) < \alpha^{\mathcal{N}}(\boldsymbol{\mu})$$

using the SCM algorithm (see Section 1.5.4). We define two a posteriori error bounds Δ_N^k and $\Delta_N^{s\,k}$ as:

$$\Delta_N^k(\boldsymbol{\mu}) = \sqrt{\frac{\frac{\Delta_t}{\alpha_{LB}(\boldsymbol{\mu})} \sum_{k'=1}^{k} \left(\epsilon_N^2(t^{k'}; \boldsymbol{\mu})(1 + \Delta_t \alpha_{LB}(\boldsymbol{\mu}))^{k'-1}\right)}{(1 + \Delta_t \alpha_{LB}(\boldsymbol{\mu}))^k}}, \tag{2.10}$$

$$\Delta_N^{s\,k}(\boldsymbol{\mu}) = \left(\sup_{v \in X^{\mathcal{N}}} \frac{\ell(v)}{\|v\|_X}\right) \Delta_N^k(\boldsymbol{\mu}), \tag{2.11}$$

where $\epsilon_N(t^k; \boldsymbol{\mu}) = \left\|r^k(\cdot; \boldsymbol{\mu})\right\|_{(X^{\mathcal{N}})'}$ and $r^k(v; \boldsymbol{\mu})$ is the residual defined as

$$r^k(v; \boldsymbol{\mu}) = g(t^k) f(v) + m\left(u_N^{\mathcal{N}\,k-1}(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right) - m\left(u_N^{\mathcal{N}\,k}(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right) - \Delta_t a\left(u_N^{\mathcal{N}\,k}(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right),$$

$\forall v \in X^{\mathcal{N}}$, $\forall k \in \mathbb{K}$. Then, we have the rigorous and sharp result as in Proposition 1.13

Proposition 2.2. *For all $N \in \mathbb{N}$ and for all $k \in \mathbb{K}$, $\boldsymbol{\mu} \in \mathcal{D}$:*

$$1 \leq \frac{\Delta_N^k(\boldsymbol{\mu})}{\left\|u^{\mathcal{N}\,k}(\boldsymbol{\mu}) - u_N^{\mathcal{N}\,k}(\boldsymbol{\mu})\right\|_{L^2}} \leq C_1 \tag{2.12}$$

$$1 \leq \frac{\Delta_N^{s\,k}(\boldsymbol{\mu})}{|s^{\mathcal{N}\,k}(\boldsymbol{\mu}) - s_N^{\mathcal{N}\,k}(\boldsymbol{\mu})|} \leq C_2, \tag{2.13}$$

where C_1 and C_2 are two constants. Moreover,

$$\left\|u^{\mathcal{N}\,k}(\boldsymbol{\mu}) - u_N^{\mathcal{N}\,k}(\boldsymbol{\mu})\right\|_{L^2} \leq \Delta_N^k(\boldsymbol{\mu}), \tag{2.14}$$

$$|s^{\mathcal{N}\,k} - s_N^{\mathcal{N}\,k}(\boldsymbol{\mu})| \leq \Delta_N^{s\,k}(\boldsymbol{\mu}) \tag{2.15}$$

The last thing that we have to explain is the construction of the space $X_N^{\mathcal{N}}$. The idea is to consider the set $\mathbb{K}$ as a "small" (time-)parameter sample and the set Ξ_{train} the space-parameter sample (see 1.3) and to combine the POD (Section 1.3.1) and the Greedy (Section 1.3.2) algorithm. More precisely, with use the POD for the time-parameter and the Greedy for the space-parameter, so we have a *POD-GREEDY sampling procedure* (see [24]). In the next section we will explain in details this strategy.

2.2 POD(t)-Greedy($\boldsymbol{\mu}$) sampling procedure

Remark 2.3. In all this section we will omit the superscript $\mathcal{N}$ for the reduced basis approximations.

The algorithm is composed of two stages of POD and one stage of Greedy. As mentioned before, the POD is used in time while the Greedy is used in space. We recall that, given L elements $w_j \in X^{\mathcal{N}}$, the POD returns P X-orthogonal functions $\{\Psi_j,\, 1 \leq j \leq P\}$ such that $X_P^{POD} = span\{\Psi_j,\, 1 \leq j \leq P\}$ is optimal in the sense that

$$X_P^{POD} = arg \inf_{X_P^{POD} \subset span\{w_j,\, 1 \leq j \leq L\}} \left(\frac{1}{L} \sum_{j=1}^{L} \inf_{v \in Y_P} \|w_j - v\|_X^2 \right)^{\frac{1}{2}}.$$

For simplicity, we will write

$$X_P^{POD} = POD\left(\{w_j,\, 1 \leq j \leq L\}, P\right).$$

We introduce the parameter sample Ξ_{train} and an initial parameter $\boldsymbol{\mu}_0 \in \Xi_{train}$ and set $S = \{\boldsymbol{\mu}_0\}$. We give now the algorithm:

$$
\begin{array}{rcl}
\text{Set } Z & = & \emptyset; \\
\text{Set } N = 1 \quad \textit{and} & & \boldsymbol{\mu}_0 = \boldsymbol{\mu}^N; \\
\text{While} & & N \leq N_{max} \\
& & \{\Psi_j,\, 1 \leq j \leq P\} = POD\left(\{u^{\mathcal{N}}(t^k; \boldsymbol{\mu}^N),\, 1 \leq k \leq n_t\}, P\right); \\
& & Z \longleftarrow \{Z, \{\Psi_j,\, 1 \leq j \leq P\}\}; \\
& & N \longleftarrow N + 1; \\
& & \{\xi_n,\, 1 \leq n \leq N\} = POD(Z, N); \\
& & X_N = span\{\xi_n,\, 1 \leq n \leq N\}; \\
& & \boldsymbol{\mu}^N = arg \max_{\boldsymbol{\mu} \in \Xi_{train}} \Delta_N^K(\boldsymbol{\mu}); \\
& & S \longleftarrow \{S, \boldsymbol{\mu}^N\}; \\
\text{end.} & &
\end{array}
$$

Set $X_N = span\{\xi_n,\, 1 \leq n \leq N\}$, $1 \leq N \leq N_{max}$.

Remark 2.4. Notice that the complexity remains $\mathcal{O}(\mathcal{N}) + \mathcal{O}(n_{train})$ and is not $\mathcal{O}(\mathcal{N} n_{train})$

Roughly speaking, the Greedy selects parameter $\boldsymbol{\mu} \in \Xi_{train}$ and then we have to compute all the state solution $u^{\mathcal{N}}(t^k; \boldsymbol{\mu})$ for $1 \leq k \leq n_t$ and apply the POD procedure. Then the second POD procedure gives us spaces of dimension N.

Chapter 3

The 3D thermal fin problem

This chapter is dedicated to the resolution of our first 3 dimensional problem: *the thermal fin* (see [3], [5], [10]). This problem has already been solved in the two dimensional case in [1] (see also *http://augustine.mit.edu/workedPro--blems.htm*). As mentioned in the *Introduction*, to extend to the 3-dimensional case, we used COMSOL [2] linked with the rbMIT software [1]. We want to show that the reduced basis method can be used with success to solve more complex problems.

3.1 Problem description

In this section we will describe the Thermal Fin problem. We will explain the physics and introduce the different parameters which we consider. The main reference is [10] where there is a description for the 2D case.

3.1.1 Heat Sink

This problem considers the performance of a heat sink designed for the thermal management of high-density electronic components. The main function of a heat sink is to transfer heat from an object at a higher temperature to another at a lower temperature with greater heat capacity. The heat sink comprises of a *base/spreader* which in turn supports a number of *plate fins* exposed to flowing air (see Fig. 3.3). The high density of the heat sink combined with its large aera, due to fins, results in the rapid transfer of thermal energy to the surrounding cooler material. Then, the heat sink is cooled and whatever is in direct contact with is also cooled. In our analysis, we shall consider one half of one fin for symmetry reasons (see Fig. 3.1, [10] and [5]). We model the flowing air through a simple convection heat transfer coefficient. Our interest is at the temperature at the base of the spreader.

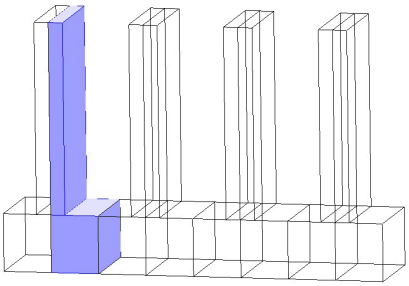


Figure 3.1: Respective Domain for a periodic heat sink.

From the engineering point of view, this problem illustrates the application of conduction analysis to an important class of cooling problems : electronic components and systems. Examples of systems that require a heat sink to reduce their temperature are microprocessors and refrigeration. On figure 3.2, we show an example of microprocessor with heat sink and a fan that produce an airflow.

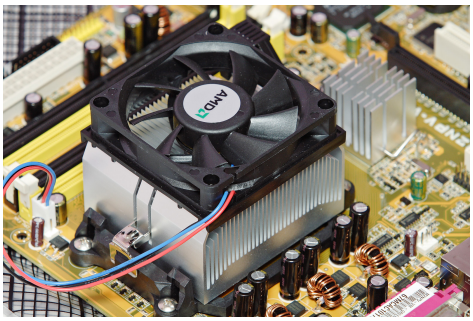


Figure 3.2: Example of heat sink.

3.1.2 Parametrized geometry and parameters

In this section, we introduce the different quantities which describe our problem. The quantities with a tilde $\sim$ correspond to dimensional quantities and the absence of a tilde denotes non-dimensional quantities.
We assume that the spreader has thermal conductivity $\widetilde{k}_{sp}$ and the plate fin has thermal conductivity $\widetilde{k}_{fin}$. The ratio of these conductivities is denoted

by

$$k = \frac{\widetilde{k}_{sp}}{\widetilde{k}_{fin}}.$$

The distance between two fin is $\widetilde{d}_{per}$ and the height of a fin is denoted by $\widetilde{L}$. We characterize the heat transfer from the fin to the air by a heat transfer coefficient $\widetilde{h}_c$. We consider $P = 3$ parameters, two of them are physical and one is geometrical. The geometrical parameter is

$$\mu_2 = \frac{\widetilde{L}}{\widetilde{d}_{per}},$$

the non-dimensional fin height. We have two other physical parameters. The first physical parameter is the Biot number defined as

$$\mu_1 = Bi = \frac{\widetilde{h}_c \widetilde{d}_{per}}{\widetilde{k}_{fin}};$$

the second one is given by the conductivity ratio

$$\mu_3 = k.$$

The parameter domain is given by $\mathcal{D} = [0.01, 0.5] \times [2, 8] \times [1, 10]$. We denote by $\boldsymbol{\mu}$ the vector of parameters, i.e.

$$\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3).$$

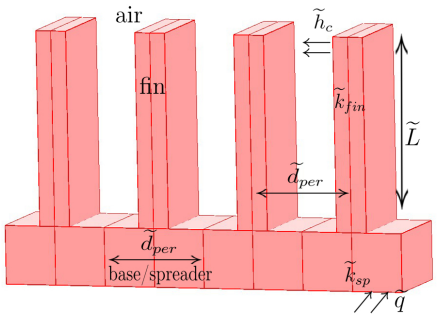


Figure 3.3: Heat Sink.

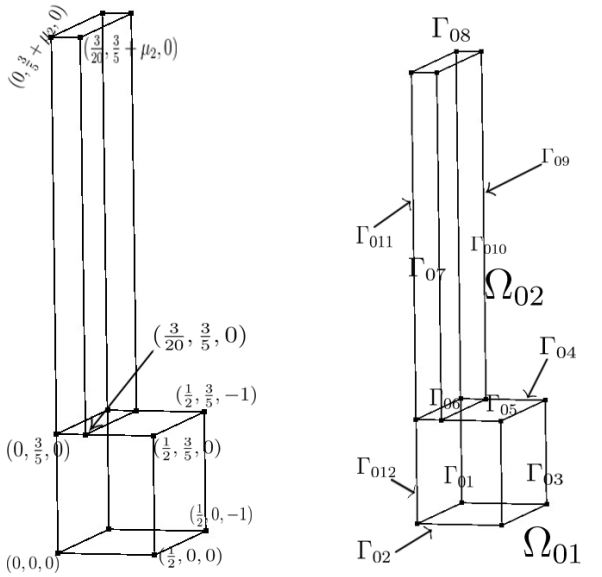
The temperature $\widetilde{T}$ is measured relative to the temperature of the air *at infinity*, $\widetilde{T}_{air}$, and non-dimensionalized with respect to $\frac{\widetilde{q}\widetilde{d}_{per}}{\widetilde{k}_{fin}}$ where $\widetilde{q}$ is the dimensional heat flux into the spreader. Then, we have that

$$\widetilde{T} = u_o \frac{\widetilde{q}\widetilde{d}_{per}}{\widetilde{k}_{fin}} + \widetilde{T}_{air},$$

where u_o is the non-dimensional temperature distribution. It follows from the previous equation that $u_o = \frac{\widetilde{T}-\widetilde{T}_{air}}{\frac{\widetilde{q}\widetilde{d}_{per}}{\widetilde{k}_{fin}}}$. The spatial coordinate $\widetilde{x} = (\widetilde{x}_{o1}, \widetilde{x}_{o2}, \widetilde{x}_{o3})$ is non-dimensionalized with respect to $\widetilde{d}_{per}$.
We identify in Figure3.4 the points defining the geometry and the different subdomains $\Omega_o^k(\boldsymbol{\mu})$, for $1 \leq k \leq 2$ considered. We define the global domain $\Omega_o(\boldsymbol{\mu})$ as

$$\Omega_o(\boldsymbol{\mu}) = \Omega_o^1(\boldsymbol{\mu}) \cup \Omega_o^2(\boldsymbol{\mu}).$$

Since we have only one geometric parameter, μ_2 and since Ω_o^1 is parameter-independent, we can write $\Omega_o(\boldsymbol{\mu}) = \Omega_o(\mu_2) = \Omega_o^1 \cup \Omega_o^2(\mu_2)$.



(a) Parametrized Geometry.

(b) Domain Boundaries.

Figure 3.4:

3.2 Mathematical description

The non-dimensional temperature $u_o(\boldsymbol{\mu})$ satisfies the conduction equation in $\Omega_o(\mu_2)$. We impose continuity of temperature and heat flux at the spreader-fin interface. Moreover, we impose zero heat flux on the horizontal exposed surfaces of the spreader and fin; uniform heat flux on the spreader base and heat-transfer convection (Robin) boundary conditions on the vertical face of the fin: the surface exposed to the flowing air.

3.2.1 Strong formulation

Mathematically, $u_o(\boldsymbol{\mu})$ satisfies

$$
\begin{cases}
-\boldsymbol{\nabla}\cdot\left(\begin{bmatrix}\mu_3 & 0 & 0\\ 0 & \mu_3 & 0\\ 0 & 0 & \mu_3\end{bmatrix}\boldsymbol{\nabla}u_o(\boldsymbol{\mu})\right)=0 & \text{in } \Omega_o^1,\\[2ex]
-\boldsymbol{\nabla}\cdot\left(\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\boldsymbol{\nabla}u_o(\boldsymbol{\mu})\right)=0 & \text{in } \Omega_o^2(\mu_2),\\[2ex]
\mu_3\frac{\partial}{\partial \boldsymbol{n}}u_o(\boldsymbol{\mu})=1 & \text{on } \Gamma_{o2},\\[2ex]
\frac{\partial}{\partial \boldsymbol{n}}u_o(\boldsymbol{\mu})=0 & \text{on } \Gamma_{o1,3,4,5,7,8,9,11,12},\\
\frac{\partial}{\partial \boldsymbol{n}}u_o(\boldsymbol{\mu})+\mu_1 u_o(\boldsymbol{\mu})=0 & \text{on } \Gamma_{o10} \text{ (Robin)},
\end{cases}
$$

where $\boldsymbol{n}$ denotes the unit outward normal and $\boldsymbol{\nabla}=\left(\frac{\partial}{\partial x_{o1}},\frac{\partial}{\partial x_{o2}},\frac{\partial}{\partial x_{o3}}\right)^T$. On Γ_{o6}, we impose continuity of temperature and heat flux.
The output of interest is

$$s(\boldsymbol{\mu})=2\int_{\Gamma_{o2}} u_o(\boldsymbol{\mu}),$$

which represents the average of the temperature on the base of the spreader, (taking into account the symmetry of the fin configuration, we just consider a half of the fin).

3.2.2 Weak formulation

In this scalar problem we take $X^e = H^1(\Omega_o(\boldsymbol{\mu}))$. The weak formulation reads as follow : *for all* $v \in X^e$, *find* $u_o(\boldsymbol{\mu}) \in X^e$ *such that*

$$\mu_3 \int_{\Omega_o^1} \nabla u_o(\boldsymbol{\mu}) \nabla v + \int_{\Omega_o^2(\mu_2)} \nabla u_o(\boldsymbol{\mu}) \nabla v + \mu_1 \int_{\Gamma_{o10}} u_o(\boldsymbol{\mu}) v = \int_{\Gamma_{o1}} v. \tag{3.1}$$

Introducing the bilinear form

$$a_o(u, v, \boldsymbol{\mu}) = \mu_3 \int_{\Omega_o^1} \nabla u(\boldsymbol{\mu}) \nabla v + \int_{\Omega_o^2(\mu_2)} \nabla u(\boldsymbol{\mu}) \nabla v + \mu_1 \int_{\Gamma_{o10}} u(\boldsymbol{\mu}) v$$

and the linear functional

$$F_o(v, \boldsymbol{\mu}) = \int_{\Gamma_{o1}} v,$$

we can rewrite (3.1) as : *find* $u_o(\boldsymbol{\mu}) \in X^e$ *such that*

$$a_o(u_o(\boldsymbol{\mu}), v, \boldsymbol{\mu}) = F_o(v), \text{ for all } v \in H^1(\Omega_o(\boldsymbol{\mu})). \tag{3.2}$$

The coercivity and the continuity of the bilinear form a_o and the continuity of the functional F_o can be proved. So the Lax-Milgram theorem ensure the existence and unicity of the solution (see [18]).

3.3 Reference geometry

This section is dedicated at the construction of affine mappings in order to obtain the affine decomposition of the bilinear form a_o and to work with parameter-independent geometry. We take $\overline{\boldsymbol{\mu}} = (0.3, 2, 5)$ as reference parameter vector (see Section 1.2.2).Then, our reference domain will be $\Omega = \Omega_o(\mu_2 = 2)$. Note that we have the following splitting for Ω

$$\Omega = \Omega_o^1 \cup \Omega_o^2\,(\mu_2 = 2) = \Omega^1 \cup \Omega^2,$$

where $\Omega^1 = \Omega_o^1$ and $\Omega^2 = \Omega_o^2\,(\mu_2 = 2)$.

3.3.1 Construction of affine mappings

Now, we want to construct an affine mapping $\mathcal{T}^{aff,k}(:;\boldsymbol{\mu}) : \Omega^k \longrightarrow \Omega_o^k(\boldsymbol{\mu})$, with $k = 1, 2$. We remind that these mappings have to be *individually bijective*, *collectively continuous* and each mapping has the general form

$$\mathcal{T}_i^{aff,k}(\boldsymbol{x}, \boldsymbol{\mu}) = C_i^{aff,k} + \sum_{j=1}^{d} G_{ij}^{aff,k}(\boldsymbol{\mu}) x_j,\ 1 \leq i \leq d$$

for given $C^{aff,k} : \mathcal{D} \longrightarrow \mathbb{R}^3$ and $G^{aff,k} : \mathcal{D} \longrightarrow \mathbb{R}^{3\times 3}$, $k = 1, 2$ (see 1.50). Since Ω_o^1 does not depend of any parameters, we have that

$$\mathcal{T}^{aff,1} = Id_1,$$

where $Id_k \colon \Omega^k \times \mathcal{D} \longrightarrow \Omega_o^k(\boldsymbol{\mu})$ is the identity operator, $k = 1, 2$.
To construct $\mathcal{T}^{aff,2}$, we choose four non-colinear points in Ω^2

$$\begin{aligned}
\overline{\boldsymbol{z}}^1 &= (0, \frac{3}{5}, 0), \\
\overline{\boldsymbol{z}}^2 &= (\frac{3}{20}, \frac{3}{5}, 0), \\
\overline{\boldsymbol{z}}^3 &= (\frac{3}{20}, \frac{3}{5}, -1), \\
\overline{\boldsymbol{z}}^4 &= (\frac{3}{20}, \frac{13}{5}, 0),
\end{aligned}$$

and four parametrized image node in $\Omega_o^2(\boldsymbol{\mu})$

$$\begin{aligned}
\overline{\boldsymbol{z}}_o^1 &= (0, \frac{3}{5}, 0), \\
\overline{\boldsymbol{z}}_o^2 &= (\frac{3}{20}, \frac{3}{5}, 0), \\
\overline{\boldsymbol{z}}_o^3 &= (\frac{3}{20}, \frac{3}{5}, -1), \\
\overline{\boldsymbol{z}}_o^4 &= (\frac{3}{20}, \frac{3}{5} + \mu_2, 0).
\end{aligned}$$

Then, we construct the matrix $\mathbb{B}^{aff,2} \in \mathbb{R}^{12\times 12}$ and the vector $V^{aff,2} \in \mathbb{R}^{12}$ as

$$\mathbb{B}^{aff,2} = \begin{bmatrix} 1 & 0 & 0 & 0 & \frac{3}{5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \frac{3}{5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3}{5} & 0 \\ 1 & 0 & 0 & \frac{3}{20} & \frac{3}{5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \frac{3}{20} & \frac{3}{5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3}{20} & \frac{3}{5} & 0 \\ 1 & 0 & 0 & \frac{3}{20} & \frac{3}{5} & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \frac{3}{20} & \frac{3}{5} & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3}{20} & \frac{3}{5} & -1 \\ 1 & 0 & 0 & \frac{3}{20} & \frac{13}{5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \frac{3}{20} & \frac{13}{5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3}{20} & \frac{13}{5} & 0 \end{bmatrix},$$

and

$$V^{aff,2} = \begin{bmatrix} 0 \\ \frac{3}{5} \\ 0 \\ \frac{3}{20} \\ \frac{3}{5} \\ 0 \\ \frac{3}{20} \\ \frac{3}{5} \\ -1 \\ \frac{3}{20} \\ \frac{3}{5} + \mu_2 \\ 0 \end{bmatrix}.$$

And we obtain the coefficients of our affine mapping :

$$\begin{bmatrix} C_1^{aff,2}(\boldsymbol{\mu}) \\ C_2^{aff,2}(\boldsymbol{\mu}) \\ C_3^{aff,2}(\boldsymbol{\mu}) \\ G_{11}^{aff,2}(\boldsymbol{\mu}) \\ G_{12}^{aff,2}(\boldsymbol{\mu}) \\ G_{13}^{aff,2}(\boldsymbol{\mu}) \\ G_{21}^{aff,2}(\boldsymbol{\mu}) \\ G_{22}^{aff,2}(\boldsymbol{\mu}) \\ G_{23}^{aff,2}(\boldsymbol{\mu}) \\ G_{31}^{aff,2}(\boldsymbol{\mu}) \\ G_{32}^{aff,2}(\boldsymbol{\mu}) \\ G_{33}^{aff,2}(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{3}{5} - \frac{3}{10}\mu_2 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \frac{\mu_2}{2} \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \left(\mathbb{B}^{aff,2}\right)^{-1} V^{aff,2}(\boldsymbol{\mu}).$$

Finally, the affine mapping reads :

$$\mathcal{T}^{aff,2}(\boldsymbol{x}, \boldsymbol{\mu}) = \begin{pmatrix} 0 \\ \frac{3}{5} - \frac{3}{10}\mu_2 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{\mu_2}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{x}.$$

3.3.2 Affine decomposition

In order to find the affine decomposition, we start to compute the Jacobian and the matrix $D^{aff,k}$, $k = 1, 2$. For $k = 1$, all these quantities are trivial, i.e. $J^{aff,1} = 1$ and $D^{aff,1} = I$, where $I \in \mathbb{R}^{3\times 3}$ is the identity matrix. For $k = 2$, we easily compute

$$J^{aff,2}(\mu_2) = \frac{\mu_2}{2}$$

and

$$D^{aff,2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{2}{\mu_2} & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Now, using the transformation (1.56), we obtain the bilinear form expressed in the reference domain :

$$\begin{aligned} a(u,v,\boldsymbol{\mu}) &= \mu_3 \left(\int_{\Omega^1} \frac{\partial}{\partial x_1} u \frac{\partial}{\partial x_1} v + \int_{\Omega^1} \frac{\partial}{\partial x_2} u \frac{\partial}{\partial x_2} v + \int_{\Omega^1} \frac{\partial}{\partial x_3} u \frac{\partial}{\partial x_3} v \right) \\ &\quad + \int_{\Omega^2} \frac{\partial}{\partial x_1} u \frac{\partial}{\partial x_1} v + \frac{2}{\mu_2} \int_{\Omega^2} \frac{\partial}{\partial x_2} u \frac{\partial}{\partial x_2} v + \int_{\Omega^2} \frac{\partial}{\partial x_3} u \frac{\partial}{\partial x_3} v \\ &\quad + \frac{\mu_1 \mu_2}{2} \int_{\Gamma_{10}} uv. \end{aligned}$$

In the same way, we obtain the linear functional

$$F(v) = \int_{\Gamma_2} v.$$

The affine decomposition is now clear, we have

$$a(u,v,\boldsymbol{\mu}) = \sum_{q=1}^{7} \theta_a^q(\boldsymbol{\mu}) a^q(u,v), \tag{3.3}$$

$$F(v,\boldsymbol{\mu}) = \theta_f^1(\boldsymbol{\mu}) f^1(v), \tag{3.4}$$

where

$$\theta_a^1 = \theta_a^2 = \theta_a^3 = \mu_3,\ \theta_a^4 = \theta_a^6 = 1,\ \theta_a^5 = \frac{2}{\mu_2} \text{ and } \theta_a^7 = \frac{\mu_1 \mu_2}{2},$$

and

$$\theta_f^1 = 1,$$

while

$$a^q(u,v) = \begin{cases} \displaystyle\int_{\Omega^1} \frac{\partial u}{\partial x_q} \frac{\partial v}{\partial x_q} & \text{for } q = 1,2,3, \\ \displaystyle\int_{\Omega^2} \frac{\partial u}{\partial x_q} \frac{\partial v}{\partial x_q} & \text{for } q = 4,5,6, \\ \displaystyle\int_{\Gamma_{10}} uv & \text{for } q = 7, \end{cases}$$

and

$$f^1(v) = \int_{\Gamma_2} v.$$

3.4 Results and Visualizations

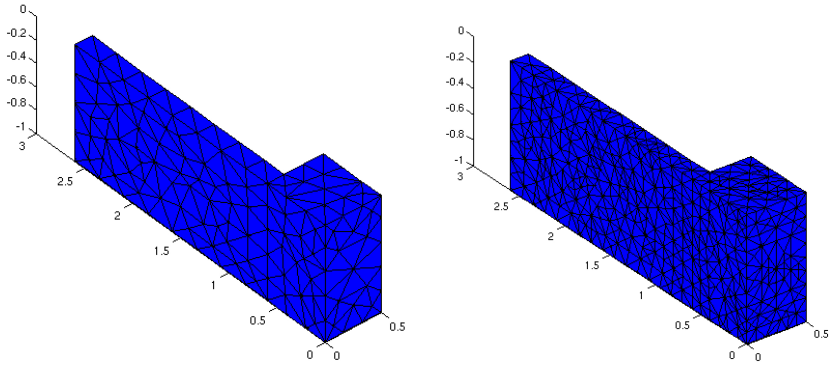
In this section, we present some results that we obtain linking COMSOL and rbMIT. More precisely, thanks to COMSOL, we drew the reference geometry of the problem and we generated a mesh. The different matrices A^q which are needed in the Offline stage (see Section 1.2.4) are extracted with COMSOL too. After that, we used the rbMIT software as in the 2-dimensional case but without the use of the symbolic part.
First we will give some properties about the mesh. After that, we will focus on the SCM (Section 1.5.4) and on two sample strategies, the Greedy (Section 1.3.2) and the POD (Section 1.3.1). We will then compare these two strategies computing the error in L^2 and L^∞-norm (see Remark 1.12). We will also be interested in the behaviour of the output s_N when we make vary the parameters and we will compare these results with the 2D case. Finally, we will represent some solutions and give their respective error.

3.4.1 Mesh

In the table 3.1, we represent the different properties of the meshes that we used. The meshes have been generated by *COMSOL* [2]. We used $\mathbb{P}_2$ element and we have 1440 and 11340 DoFs for the first and second mesh respectively.

Mesh properties	Mesh 1	Mesh 2
Vertices	251	1724
Vertex	14	14
Edge	91	176
Triangle	470	1726
Tetrahedra	712	7052
DoFs	1440	11340
Element type	$\mathbb{P}_2$	$\mathbb{P}_2$

Table 3.1: Properties of the mesh 1 and 2 for the reference geometry.

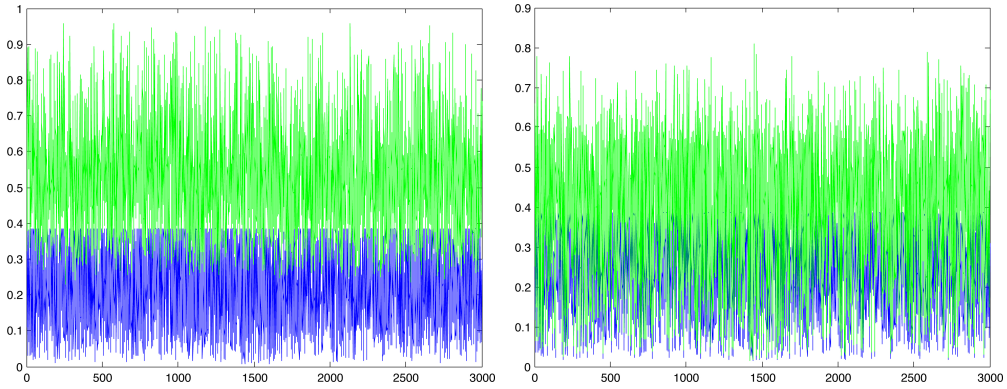


(a) Mesh 1. (b) Mesh 2.

Figure 3.5: Meshes for the reference geometry.

3.4.2 SCM

For the SCM, we took a train sample Ξ_{train} of size $n_{train} = 3000$ and a tolerance $\epsilon_{SCM} = 0.85$ (see Section 1.5.4). In Figure 3.6, we show the α_{UB} (green) and the α_{LB} (blue) for different $J = 1$ (Fig. 3.6(a)) and $J_{max} = 3$ (Fig. 3.6(b)). Graphically we see that we have good results because the



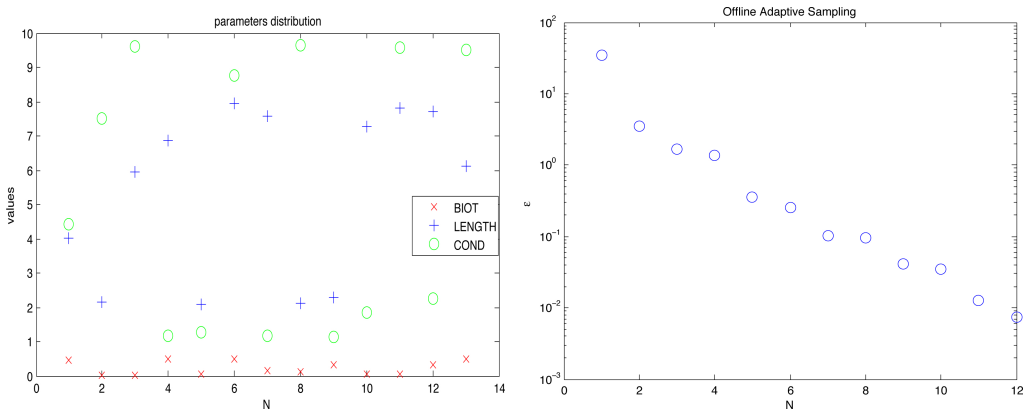
(a) First iteration of the SCM. (b) Last iteration of SCM.

Figure 3.6: Representation of α_{UB} (green) and α_{LB} (blue) for the 3D Thermal Fin problem.

lower bound and the upper bound are very close (Fig. 3.6(b)).

3.4.3 Greedy

In this section, we give the result of the Greedy algorithm (Section 1.3.2). Here, the sample size is $n_{train} = 500$, the tolerance $\epsilon_{tol,min} = 0.01$ and the $\overline{N}_{max} = 120$. In the figure 3.7(a), we have represented for each N the parameter $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)$ that was chosen automatically by the algorithm. We have obtained $N_{max} = 12$ for the primal problem. In the figure 3.7(b), we represent the error bound $\Delta_N(\boldsymbol{\mu})$ for $1 \leq N \leq N_{max}$. We see that the error is decreasing as we except in theory. We can see that few basis function are



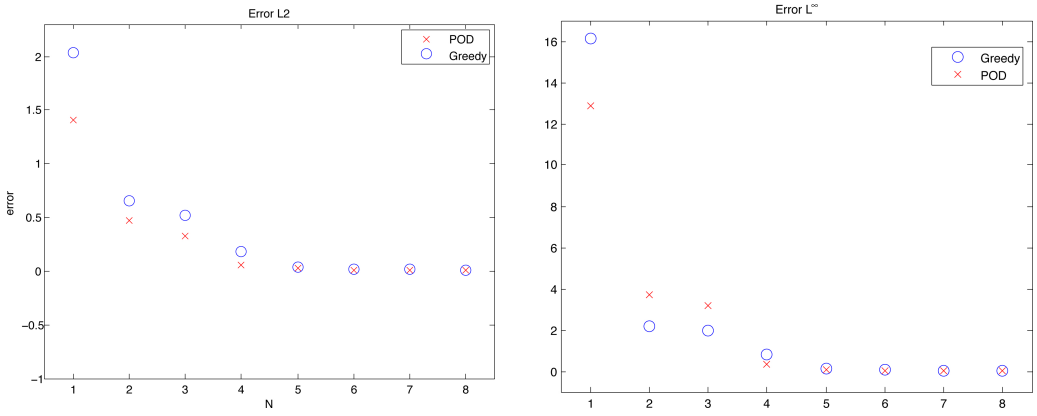
(a) Chosen parameters. (b) Error bound Δ_N.

Figure 3.7: Sample and error bound for the Greedy.

needed to build a good reduced basis approximation ($N_{max} = 12$).

3.4.4 Comparison between the POD and the Greedy

In this section, we compare two reduced order methods: the Greedy and the POD. We recall that theoretically the Greedy-RB has to minimize the RB error in L^∞-norm while the POD minimizes the projection error in L^2-norm (Remark 1.12). In figure 3.8(a) we represent the error in L^2-norm for Greedy-RB and POD approximations while in figure 3.8(b), we represent the error in L^∞-norm.



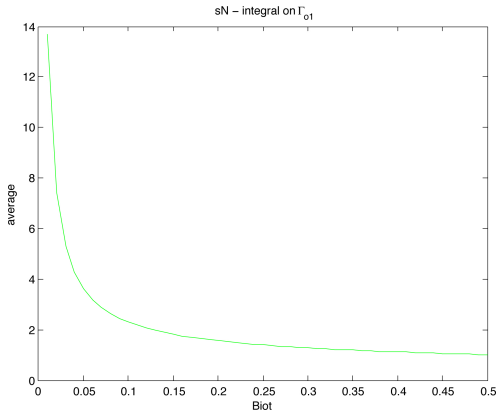
(a) $\|u^{\mathcal{N}} - u_N^{\mathcal{N}}\|_{L^2}$. (b) $\|u^{\mathcal{N}} - u_N^{\mathcal{N}}\|_{L^\infty}$.

Figure 3.8: Sample and error bound for the Greedy.

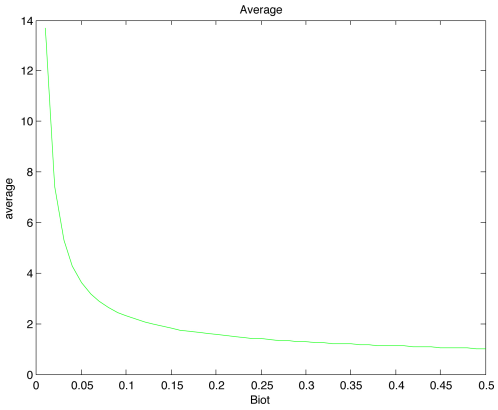
We see that the plot 3.8(a) confirms the theoretical expectation, i.e. the error of the Greedy-RB approximation is bigger that the error of the POD approximation since we are considering POD in its natural environment, i.e. by considering L^2 norm (and not L^∞).

3.4.5 Output

Here, we present the variation of the output s_N, i.e, the average of the temperature on the base of the spreader, in function of the different parameters in 3D and 2D case (see Fig. 3.9 3.10 3.11). Graphically, the result of Figure 3.9 corresponds to results expected by the theory. Indeed, if the Biot number (μ_1) increases, then there is a bigger heat transfer and so the temperature at the base decreases. Moreover, we see that the 3D case is similar to the 2D case. Since μ_1 is a physical parameter, this similitude is quite logical and expected. When we change the Biot number, the temperature decreases if the Biot number increases and this fact is independent of the dimension of the problem.



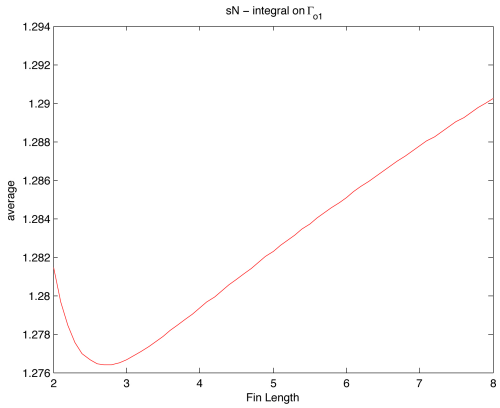
(a) Output s_N for different values of μ_1 in the 3D case.



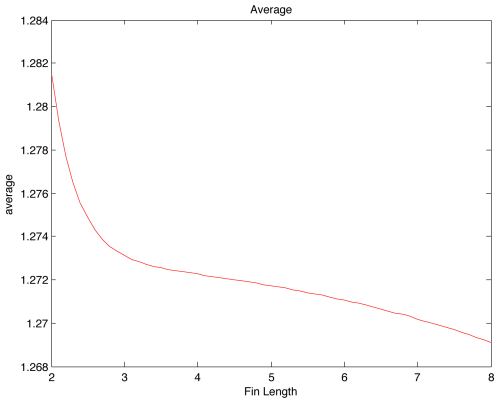
(b) Output s_N for different values of μ_1 in the 2D case.

Figure 3.9: Output with $\mu_2 = 2$ and $\mu_3 = 4$.

In Figure 3.10, we note a difference between the 3D and 2D problem when we vary μ_2. In 3D, the temperature decreases and then it increases while in the 2D case the temperature always decreases. In this case the difference is due to the fact that 3D problem is characterized by a more complex geometrical configuration where there is one more dimension (the depth) influencing the conduction.

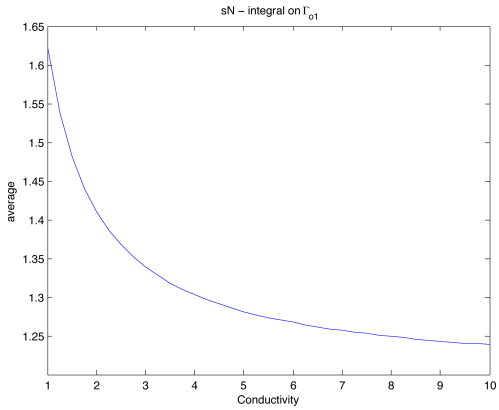


(a) Output s_N for different values of μ_2 in the 3D case.

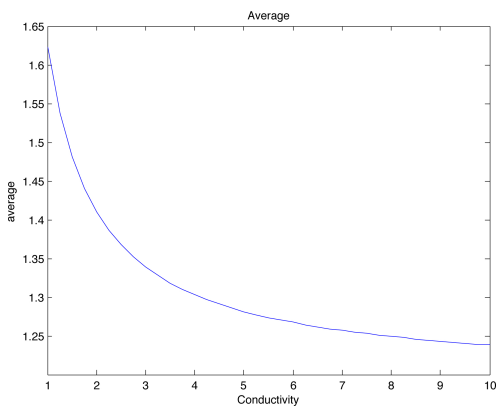


(b) Output s_N for different values of μ_2 in the 2D case.

Figure 3.10: Output with $\mu_1 = 0.3$ and $\mu_3 = 4$.



(a) Output s_N for different values of μ_3 in the 3D case.



(b) Output s_N for different values of μ_3 in the 2D case.

Figure 3.11: Output with $\mu_1 = 0.3$ and $\mu_2 = 2$.

In Figure 3.11, we still have a similitude between the 3D and 2D problem. Here μ_3 is also a physical parameter, so the same explanation as above works.

3.4.6 Computational time

In this section, we will give some computational times that show the efficiency of the method. We define two kind of computational time :

$$t^{Offline}(\mathcal{N}) = \frac{\text{Offline time to perform SCM \& Greedy}}{\text{time to evaluate } \boldsymbol{\mu} \to s^{\mathcal{N}}(\boldsymbol{\mu})},$$

that compute the *break-even*, i.e. the maximal number of solutions that we may evaluate with the FE method without using the RB.

$$t^{Online}(\mathcal{N}, N) = \frac{\text{Online time to evaluate } \partial t(\boldsymbol{\mu} \to s_N)}{\text{time to evaluate } \boldsymbol{\mu} \to s^{\mathcal{N}}(\boldsymbol{\mu})},$$

where, ∂t is the time we need to evaluate $\boldsymbol{\mu} \to s_N(\boldsymbol{\mu})$. This computational time shows the gain of time using the RB method for the solution of parametrized problems.
In this problem, we compute these computational time for the two meshes. We define $\mathcal{N}_1$ and $\mathcal{N}_2$ the dimension of the first and second mesh respectively.

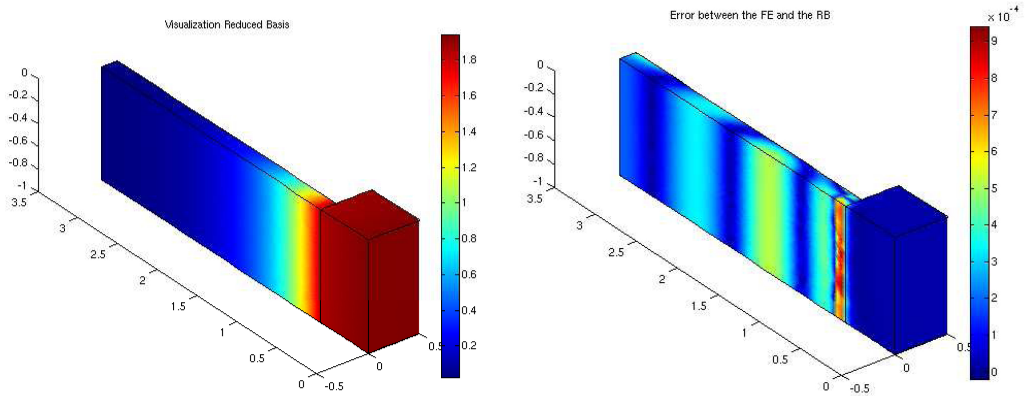
So, we have :

$$\begin{aligned} t^{Offline}(\mathcal{N}_1) &= \frac{140.1}{0.05} = 2802 \\ t^{Offline}(\mathcal{N}_2) &= \frac{706.3}{0.61} = 1157.8 \\ t^{Online}(\mathcal{N}_1, N) &= \frac{0.004}{0.05} = 0.08 \\ t^{Online}(\mathcal{N}_2, N) &= \frac{0.004}{0.61} = 0.006 \end{aligned}$$

For the first mesh, we obtain a break-even = 2802. So, it is not convenient to use the RB if we have to compute less than 2802 solutions. Here, looking at t^{Online}, we see that the use of the RB method is very gainful. We can get faster Online evaluation (8% of computational costs in the Online evaluation) .
For the second mesh the break-even = 1158 and $t^{Online}(\mathcal{N}_2, N)$ is 0.006 then the computational costs for the RB is 150 times less expensive than the one for the FE. So here it is more gainful to use RB.

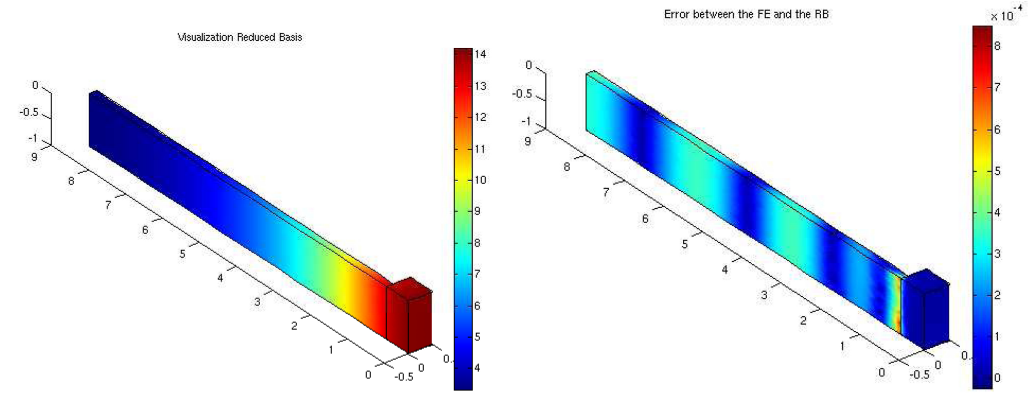
3.4.7 Visualization

This section is dedicated to the visualization of some representative reduced basis solutions. On the left figures, we show the solution for different value of the parameter $\boldsymbol{\mu}$. On the right side, we represent the error between the RB approximations and the FE solution.



(a) $\Delta_N^s = 3.36e^{-6}$. (b) Error $u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})$.

Figure 3.12: Example of representative solution and error for $\boldsymbol{\mu} = (0.5, 2.75, 10)$.



(a) $\Delta_N^s = 8.2e^{-5}$. (b) Error $u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})$.

Figure 3.13: Example of representative solution and error for $\boldsymbol{\mu} = (0.01, 8, 1)$.

We note that the error bound is in the order of 10^{-4} and so there is a good approximation.

Chapter 4

The 3D Time-Dependent Graetz Problem

In this chapter, we will treat a 3D time-dependent non-compliant problem: the *Graetz problem* (see [4], [5] and [22]). The case 2D has also be treated in [1] (see also http://augustine.mit.edu/workedProblems.htm). This problem and the thermal fin Chapter 3 show that the RB method can be used to solve also 3D steady and unsteady problems.

4.1 Problem description

The reader can refer to [1] for a description in 2D case.
This is a classical problem in literature dealing with forced heat convection combined with heat conduction in a duct . The duct is separated in two parts. This first one is made up of cold walls whereas the second part has hot walls. The temperature at inlet is imposed and the flow has a known given convective field. In Figure 4.1, we can see the duct with the cold and hot portion.

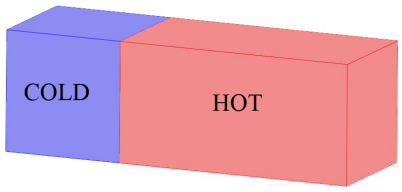


Figure 4.1: A channel with cold and hot portion.

The physical domain $\Omega_o(\boldsymbol{\mu}) = \Omega_o^1(\boldsymbol{\mu}) \cup \Omega_o^2(\boldsymbol{\mu})$ is defined in Figure 4.2(a). We will adopt the same notation as in chapter 3, i.e. the quantity with a tilde

are dimensional quantity. A point $\boldsymbol{x} = (x_{o1}, x_{o2}, x_{o3})$ is non-dimensionalized with respect to $\tilde{h}$ the width of the channel (in the x_{o3}-direction). We introduce also $\tilde{k}$ as the dimensional conductivity coefficient for the air flowing in the channel, $\tilde{\nu}$ the dimensional diffusivity and $\tilde{U}$ the reference velocity for the convective field. We introduce the *Péclèt* number defined as $\mathbb{P}e = \frac{\tilde{U}\tilde{h}}{\tilde{\nu}}$.

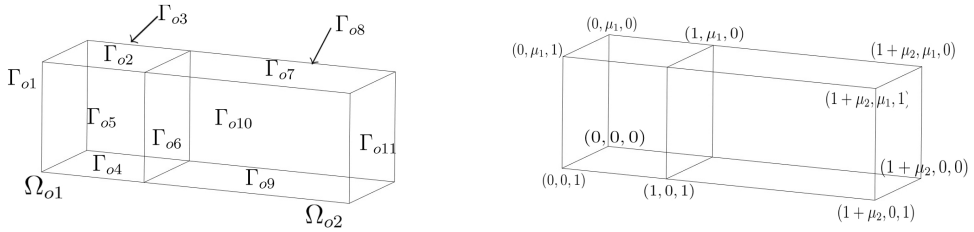
We consider 3 parameters. We denote by μ_1 and μ_2 the geometric parameters that represents the height of the cold portion and the length of the hot portion, respectively (see Figure 4.2(b)). The last one is the Péclèt number that is a physical parameter, i.e. $\mu_3 = \mathbb{P}e$. The parameter domain is given by $\mathcal{D} = [1, 2] \times [2, 10] \times [0.1, 10]$. We denote by $\boldsymbol{\mu}$ the vector of parameters, i.e.

$$\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3).$$

The temperature $u_o(\boldsymbol{\mu})$ is non-dimensionalized and defined as

$$u_o(\boldsymbol{\mu}) = \frac{\tilde{T} - \tilde{T}_{inlet}}{\tilde{T}_{hot} - \tilde{T}_{inlet}},$$

where $\tilde{T}$ is the dimensional temperature, $\tilde{T}_{inlet}$ is the dimensional temperature at inflow and $\tilde{T}_{hot}$ is the dimensional temperature of the hot walls.



(a) Domain Boundaries. (b) Parametrized Geometry.

Figure 4.2:

4.2 Mathematical description

The non-dimensionalized temperature $u_o(\boldsymbol{x}, t; \boldsymbol{\mu})$ satisfies the unsteady advection-diffusion equation in $\Omega_o(\boldsymbol{\mu})$. The time interval is $[0, T]$, where T is the final time. We impose continuity of temperature and heat flux on all

internal faces. At the inflow and on the cold walls, we impose homogeneous Dirichlet boundary condition. On the hot walls, we have non-homogeneous Dirichlet boundary condition and at the outflow we impose zero heat flux (homogeneous Neumann boundary condition).

Notation 4.1. For convenience, we will not indicate the $\boldsymbol{x}$-dependence of u_o, i.e. we will write $u_o(\boldsymbol{x}, t; \boldsymbol{\mu})$ as $u_o(t; \boldsymbol{\mu})$.

4.2.1 Strong formulation

Mathematically, $u_o(\boldsymbol{\mu})$ satisfies

$$
\begin{cases}
\frac{\partial u_o(t;\boldsymbol{\mu})}{\partial t} - \boldsymbol{\nabla} \cdot \left(\begin{bmatrix} (\mu_3)^{-1} & 0 & 0 \\ 0 & (\mu_3)^{-1} & 0 \\ 0 & 0 & (\mu_3)^{-1} \end{bmatrix} \boldsymbol{\nabla} u_o(t; \boldsymbol{\mu}) \right) + x_{o2}(1 - x_{o2}) \frac{\partial}{\partial x_1} u_o(t; \boldsymbol{\mu}) = 0 & \\
\qquad\qquad \text{in } \Omega_o(\boldsymbol{\mu}),\ t \in [0, T] & \\
u_o(t; \boldsymbol{\mu}) = 0 \qquad\qquad \text{on } \Gamma_{o1,2,3,4,5},\ t \in (0, T] & \\
u_o(t; \boldsymbol{\mu}) = g(t) \qquad\qquad \text{on } \Gamma_{o7,8,9,10},\ t \in (0, T] & \\
\frac{\partial}{\partial \boldsymbol{n}} u_o(t; \boldsymbol{\mu}) = 0 \qquad\qquad \text{on } \Gamma_{o11},\ t \in (0, T] & \\
u_o(t = 0; \boldsymbol{\mu}) = 0 \qquad\qquad \text{in } \Omega_o(\boldsymbol{\mu}) \ \text{(Initial condition)} &
\end{cases}
$$

where $\boldsymbol{n}$ denotes the unit outward normal and $g(t)$ is the *control input*. On Γ_{o6}, we impose continuity of temperature and heat flux. The convective field is $U = x_{o2}(1 - x_{o2})$.

Notation 4.2. - The homogeneous Dirichlet boundaries will be denoted by Γ_{Dh} i.e.

$$\Gamma_{Dh} = \Gamma_{o1,2,3,4,5}$$

and the non-homogeneous Dirichlet boundary will be denoted by Γ_{Dnh}, i.e.

$$\Gamma_{Dnh} = \Gamma_{o7,8,9,10}$$

We denote by $\Gamma_D = \Gamma_{Dh} \cup \Gamma_{Dnh}$.

- The Neumann boundary will be denoted by $\Gamma_N = \Gamma_{o11}$.

We are interested in two kinds of output. The first one is

$$s_1(t;\boldsymbol{\mu}) = \int_{\Omega_o(\boldsymbol{\mu})} u_o(t;\boldsymbol{\mu})\, d\boldsymbol{x} \tag{4.1}$$

which is a vector and the i^{th} component of the vector represents the average of the temperature at the i^{th} timestep. The second one is

$$s_2(t;\boldsymbol{\mu}) = \int_0^T \left(\int_{\Omega_o(\boldsymbol{\mu})} u_o(t;\boldsymbol{\mu})\, d\boldsymbol{x} \right) h(t)\, dt \tag{4.2}$$

where $h(t)$ is a function of time. This output represents the average temperature.

4.2.2 Weak formulation

Let us introduce the space $V = \{v \in H^1(\Omega_o(\boldsymbol{\mu})) \,|\, v = 0 \text{ on } \Gamma_D\}$. We will also introduce a lifting of $g(t)$, $R_g \in H^1(\Omega_o(\boldsymbol{\mu})) \times [0,T]$ such that $R_{g|\Gamma_{Dnh}} = g(t)$ and $u_o(t;\boldsymbol{\mu}) = \overline{u}_o(t;\boldsymbol{\mu}) + R_g(\boldsymbol{x},t)$ where $\overline{u}_o \in [0,T] \times V$. We can now give the weak formulation.

For all $v \in V$, find $\overline{u}_o(t;\boldsymbol{\mu}) \in [0,T] \times V$ such that

$$m_o(\frac{\partial \overline{u}_o}{\partial t}, v;\boldsymbol{\mu}) + a_o(\overline{u}_o, v;\boldsymbol{\mu}) = g(t) F_o(v;\boldsymbol{\mu}) \tag{4.3}$$

where

$$\begin{aligned} a_o(\overline{u}_o, v;\boldsymbol{\mu}) &= \frac{1}{\mu_3} \int_{\Omega_o(\boldsymbol{\mu})} \nabla \overline{u}_o(t;\boldsymbol{\mu}) \nabla v + \int_{\Omega_o(\boldsymbol{\mu})} x_{o2}(1-x_{o2}) \frac{\partial \overline{u}_o(t;\boldsymbol{\mu})}{\partial x_{o1}} v, \\ m_o(\frac{\partial \overline{u}_o}{\partial t}, v;\boldsymbol{\mu}) &= \int_{\Omega_o(\boldsymbol{\mu})} \frac{\partial \overline{u}_o(t;\boldsymbol{\mu})}{\partial t} v, \end{aligned}$$

and

$$F_o(v,\boldsymbol{\mu}) = -\int_{\Omega_o(\boldsymbol{\mu})} \frac{\partial R_g}{\partial t} v - \frac{1}{\mu_3} \int_{\Omega_o(\boldsymbol{\mu})} \nabla R_g \nabla v - \int_{\Omega_o(\boldsymbol{\mu})} x_{o2}(1-x_{o2}) \frac{\partial R_g}{\partial x_{o1}} v.$$

The function $g \in L^2(0,T)$ is called the *control input*. The coercivity and the continuity of the bilinear form a_o and the continuity of the functional F_o can be proved. So the Lax-Milgram theorem ensure the existence and unicity of the solution (see [18]).

4.3 Reference geometry

As we did for the thermal fin, we will construct the affine mappings to obtain the affine decomposition of the bilinear form a_o and F_o. In this case our reference domain is given for the reference parameter $\overline{\boldsymbol{\mu}} = (1,2,1)$, i.e. the reference domain is $\Omega = \Omega_o(\overline{\boldsymbol{\mu}})$ and the reference subdomains are $\Omega^k = \Omega_o^k(\overline{(}\boldsymbol{\mu}))$, $k = 1,2$.

4.3.1 Construction of affine mappings

We want to construct an affine mapping $\mathcal{T}^{aff,k}(:;\boldsymbol{\mu}) : \Omega^k \longrightarrow \Omega_o^k(\boldsymbol{\mu})$, with $k = 1,2$. We remind that these mappings have to be *individually bijective*, *collectively continuous* and each mapping has the general form

$$\mathcal{T}_i^{aff,k}(\boldsymbol{x},\boldsymbol{\mu}) = C_i^{aff,k} + \sum_{j=1}^{d} G_{ij}^{aff,k}(\boldsymbol{\mu}) x_j,\ 1 \leq i \leq d$$

for given $C^{aff,k} : \mathcal{D} \longrightarrow \mathbb{R}^3$ and $G^{aff,k} : \mathcal{D} \longrightarrow \mathbb{R}^{3\times 3}$, $k = 1,2$. (see (1.50)).

In this problem the two subdomains depend on the parameters. So we start to chose 4 non-colinear points in Ω^1

$$\begin{aligned}
\overline{\boldsymbol{z}}^1 &= (0,0,0),\\
\overline{\boldsymbol{z}}^2 &= (0,1,0),\\
\overline{\boldsymbol{z}}^3 &= (1,0,0),\\
\overline{\boldsymbol{z}}^4 &= (0,0,1),
\end{aligned}$$

and four parametrized image node in $\Omega_o^1(\boldsymbol{\mu})$

$$\begin{aligned}
\overline{\boldsymbol{z}}_o^1 &= (0,0,0),\\
\overline{\boldsymbol{z}}_o^2 &= (0,\mu_1,0),\\
\overline{\boldsymbol{z}}_o^3 &= (1,0,0),\\
\overline{\boldsymbol{z}}_o^4 &= (0,0,1).
\end{aligned}$$

Then, we construct the matrix $\mathbb{B}^{aff,1} \in \mathbb{R}^{12\times 12}$ and the vector $V^{aff,1} \in \mathbb{R}^{12}$ as

$$\mathbb{B}^{aff,1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and

$$V^{aff,1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mu_1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

And we obtain the coefficients of our affine mapping :

$$\begin{bmatrix} C_1^{aff,1}(\boldsymbol{\mu}) \\ C_2^{aff,1}(\boldsymbol{\mu}) \\ C_3^{aff,1}(\boldsymbol{\mu}) \\ G_{11}^{aff,1}(\boldsymbol{\mu}) \\ G_{12}^{aff,1}(\boldsymbol{\mu}) \\ G_{13}^{aff,1}(\boldsymbol{\mu}) \\ G_{21}^{aff,1}(\boldsymbol{\mu}) \\ G_{22}^{aff,1}(\boldsymbol{\mu}) \\ G_{23}^{aff,1}(\boldsymbol{\mu}) \\ G_{31}^{aff,1}(\boldsymbol{\mu}) \\ G_{32}^{aff,1}(\boldsymbol{\mu}) \\ G_{33}^{aff,1}(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \mu_1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \left(\mathbb{B}^{aff,1}\right)^{-1} V^{aff,1}(\boldsymbol{\mu}).$$

Finally, the affine mapping reads :

$$\mathcal{T}^{aff,1}(\boldsymbol{x}, \boldsymbol{\mu}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \mu_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{x}.$$

Now we do the same things to find $\mathcal{T}^{aff,2}$. We chose 4 non-colinear points in Ω^2

$$\begin{aligned} \overline{\boldsymbol{z}}^1 &= (1,0,0), \\ \overline{\boldsymbol{z}}^2 &= (3,0,0), \\ \overline{\boldsymbol{z}}^3 &= (3,1,0), \\ \overline{\boldsymbol{z}}^4 &= (1,0,1), \end{aligned}$$

and four parametrized image node in $\Omega_o^1(\boldsymbol{\mu})$

$$\begin{aligned} \overline{\boldsymbol{z}}_o^1 &= (0,0,0), \\ \overline{\boldsymbol{z}}_o^2 &= (\mu_2+1,0,0), \\ \overline{\boldsymbol{z}}_o^3 &= (\mu_2+1,\mu_1,0), \\ \overline{\boldsymbol{z}}_o^4 &= (1,0,1). \end{aligned}$$

Then, we construct the matrix $\mathbb{B}^{aff,2} \in \mathbb{R}^{12\times 12}$ and the vector $V^{aff,2} \in \mathbb{R}^{12}$

as

$$\mathbb{B}^{aff,2} = \left[\begin{array}{cccccccccccc} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 1 & 0 & 0 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 3 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{array}\right],$$

and

$$V^{aff,2} = \left[\begin{array}{c} 1 \\ 0 \\ 0 \\ \mu_2 + 1 \\ 0 \\ 0 \\ \mu_2 + 1 \\ \mu_1 \\ 0 \\ 1 \\ 0 \\ 1 \end{array}\right].$$

We obtain the coefficients of our affine mapping:

$$\begin{bmatrix} C_1^{aff,2}(\boldsymbol{\mu}) \\ C_2^{aff,2}(\boldsymbol{\mu}) \\ C_3^{aff,2}(\boldsymbol{\mu}) \\ G_{11}^{aff,2}(\boldsymbol{\mu}) \\ G_{12}^{aff,2}(\boldsymbol{\mu}) \\ G_{13}^{aff,2}(\boldsymbol{\mu}) \\ G_{21}^{aff,2}(\boldsymbol{\mu}) \\ G_{22}^{aff,2}(\boldsymbol{\mu}) \\ G_{23}^{aff,2}(\boldsymbol{\mu}) \\ G_{31}^{aff,2}(\boldsymbol{\mu}) \\ G_{32}^{aff,2}(\boldsymbol{\mu}) \\ G_{33}^{aff,2}(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} 1 - \frac{\mu_2}{2} \\ 0 \\ 0 \\ \frac{\mu_2}{2} \\ 0 \\ 0 \\ 0 \\ \mu_1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \left(\mathbb{B}^{aff,2}\right)^{-1} V^{aff,2}(\boldsymbol{\mu}).$$

Then, the affine mapping reads :

$$\mathcal{T}^{aff,2}(\boldsymbol{x}, \boldsymbol{\mu}) = \begin{pmatrix} 1 - \frac{\mu_2}{2} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \frac{\mu_2}{2} & 0 & 0 \\ 0 & \mu_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{x}.$$

4.3.2 Afiine decomposition

In order to find the affine decomposition, we start to compute the Jacobian and the matrix $D^{aff,k}$, $k = 1, 2$. We easily obtain that

$$J^{aff,1}(\boldsymbol{\mu}) = \mu_1 \quad \text{and} \quad J^{aff,2}(\boldsymbol{\mu}) = \frac{1}{2}\mu_1\mu_2,$$

and

$$D^{aff,1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\mu_1} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad D^{aff,2} = \begin{pmatrix} \frac{1}{\mu_2} & 0 & 0 \\ 0 & \frac{1}{\mu_1} & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Now, using the transformation (1.56), we obtain the time-independent bilinear form expressed in the reference domain:

$$
\begin{aligned}
a(\overline{u},v,\boldsymbol{\mu}) &= \frac{\mu_1}{\mu_3}\int_{\Omega^1}\frac{\partial}{\partial x_1}\overline{u}\frac{\partial}{\partial x_1}v+\frac{1}{\mu_1\mu_3}\int_{\Omega^1}\frac{\partial}{\partial x_2}\overline{u}\frac{\partial}{\partial x_2}v+\frac{\mu_1}{\mu_3}\int_{\Omega^1}\frac{\partial}{\partial x_3}\overline{u}\frac{\partial}{\partial x_3}v \\
&+\frac{2\mu_1}{\mu_2\mu_3}\int_{\Omega^2}\frac{\partial}{\partial x_1}\overline{u}\frac{\partial}{\partial x_1}v+\frac{\mu_2}{2\mu_1\mu_3}\frac{2}{\mu_2}\int_{\Omega^2}\frac{\partial}{\partial x_2}\overline{u}\frac{\partial}{\partial x_2}v \\
&+\frac{\mu_1\mu_2}{2\mu_3}\int_{\Omega^2}\frac{\partial}{\partial x_3}\overline{u}\frac{\partial}{\partial x_3}v+\mu_1\int_{\Omega^1}x_2(1-x_2)\frac{\partial\overline{u}}{\partial x_1}v \qquad (4.4)\\
&+\mu_1\int_{\Omega^2}x_2(1-x_2)\frac{\partial\overline{u}}{\partial x_1}v.
\end{aligned}
$$

In the same way we obtain the time-dependent bilinear form

$$
m(\frac{\partial\overline{u}}{\partial t},v;\boldsymbol{\mu})=\mu_1\int_{\Omega^1}\frac{\partial\overline{u}}{\partial t}v+\mu_1\int_{\Omega^2}\frac{\partial\overline{u}}{\partial t}v, \qquad (4.5)
$$

and the linear functional

$$
\begin{aligned}
F(v;\boldsymbol{\mu}) &= -\frac{\mu_1}{\mu_3}\int_{\Omega^1}\frac{\partial}{\partial x_1}R_g\frac{\partial}{\partial x_1}v-\frac{1}{\mu_1\mu_3}\int_{\Omega^1}\frac{\partial}{\partial x_2}R_g\frac{\partial}{\partial x_2}v-\frac{\mu_1}{\mu_3}\int_{\Omega^1}\frac{\partial}{\partial x_3}R_g\frac{\partial}{\partial x_3}v \\
&-\frac{2\mu_1}{\mu_2\mu_3}\int_{\Omega^2}\frac{\partial}{\partial x_1}R_g\frac{\partial}{\partial x_1}v-\frac{\mu_2}{2\mu_1\mu_3}\frac{2}{\mu_2}\int_{\Omega^2}\frac{\partial}{\partial x_2}R_g\frac{\partial}{\partial x_2}v \\
&-\frac{\mu_1\mu_2}{2\mu_3}\int_{\Omega^2}\frac{\partial}{\partial x_3}R_g\frac{\partial}{\partial x_3}v-\mu_1\int_{\Omega^1}x_2(1-x_2)\frac{\partial R_g}{\partial x_1}v \qquad (4.6)\\
&-\mu_1\int_{\Omega^2}x_2(1-x_2)\frac{\partial R_g}{\partial x_1}v-\mu_1\int_{\Omega^1}\frac{\partial R_g}{\partial t}v-\mu_1\int_{\Omega^2}\frac{\partial R_g}{\partial t}v.
\end{aligned}
$$

We obtain the affine decomposition

$$
a(\overline{u},v;\boldsymbol{\mu}) = \sum_{q=1}^{8}\theta_a^q(\boldsymbol{\mu})a^q(\overline{u},v), \qquad (4.7)
$$

$$
m(\frac{\partial\overline{u}}{\partial t},v;\boldsymbol{\mu}) = \sum_{q=1}^{2}\theta_m^q(\boldsymbol{\mu})m^q(\frac{\partial\overline{u}}{\partial t},v;\boldsymbol{\mu}), \qquad (4.8)
$$

$$
F(v;\boldsymbol{\mu}) = \sum_{q=1}^{10}\theta_f^q f^q(v;\boldsymbol{\mu}). \qquad (4.9)
$$

Consequently, we have that $Q_a = 8$, $Q_m = 2$ and $Q_f = 10$. We will not indicate explicitly what are the $\theta_a^{q_a}, a^{q_a}, \theta_m^{q_m}, m^{q_m}, \theta_f^{q_f}, f^{q_f}$ for $q_a \in [0, Q_a]$, $q_m \in [0, Q_m]$ and $q_f \in [0, Q_f]$, but these quantities can easily be deduced from equations (4.4)-(4.6).

4.4 Results and Visualization

4.4.1 Mesh

In the table 4.1, we represent the different properties of the mesh that we used. The mesh has been generated by *COMSOL* [2]. We used $\mathbb{P}_2$ element and we have 2977 DoFs.

Mesh properties	Value
Vertices	458
Vertex	12
Edge	92
Triangle	604
Tetrahedra	1779
DoFs	2977
Element	$\mathbb{P}_2$

Table 4.1: Property of the mesh for the reference geometry.

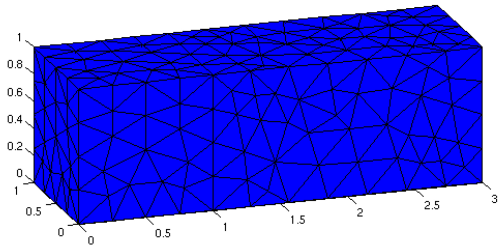
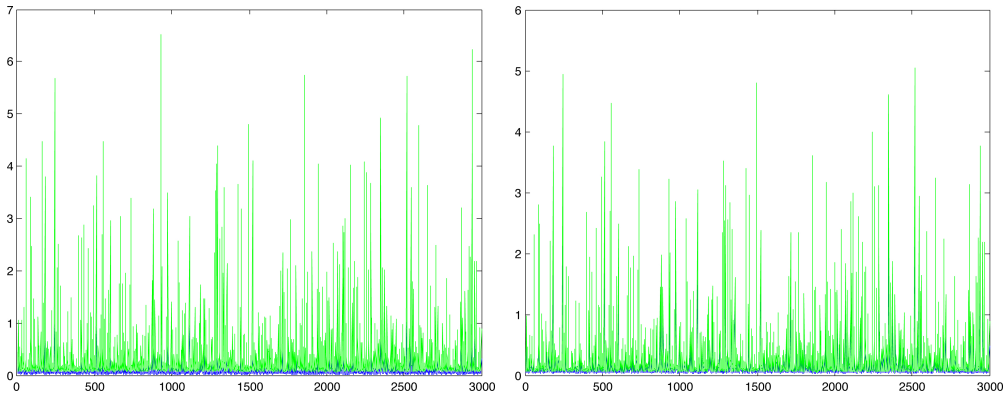


Figure 4.3: Mesh for the reference geometry.

4.4.2 SCM

For the SCM, we took a train sample Ξ_{train} of size $n_{train} = 3000$ and a tolerance $\epsilon_{SCM} = 0.85$ (see Section 1.5.4). In Figure 4.4, we show the α_{UB} (green) and the α_{LB} (blue) for different $J = 1$ (Fig. 4.4(a)) and $J_{max} = 3$ (Fig. 4.4(b)). Graphically, we can see we have good results because the lower



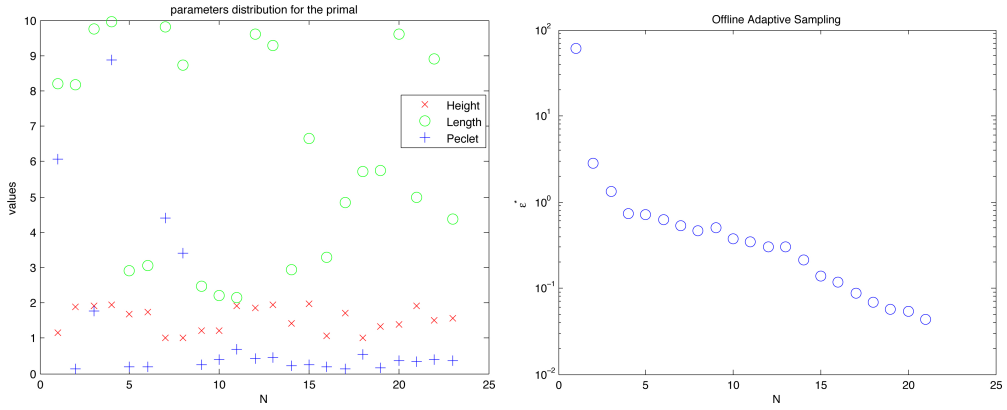
(a) First iteration of the SCM. (b) Last iteration of SCM.

Figure 4.4: Representation of α_{UB} (green) and α_{LB} (blue).

bound and the upper bound are very close (Fig. 4.4(b)).

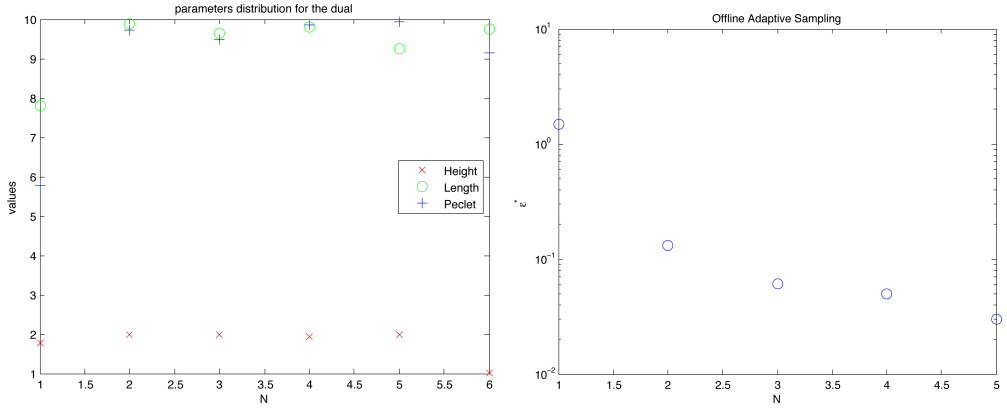
4.4.3 POD-Greedy

In this section, we give the result of the POD-Greedy-RB algorithm described in Section 2.2. Here, the sample size is $n_{train} = 500$, the tolerance $\epsilon_{tol,\,min} = 0.01$ and the $\overline{N}_{max} = 120$ (Section 2.2). Since the problem is non-compliant, we have to do the POD-Greedy for the primal and dual problem (Section 1.6). In the figure 4.5(a) and 4.6(a), we have represented for each N the parameter $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)$ that was chosen by the algorithm. We have obtained $N_{pr,\,max} = 24$ for the primal problem and $N_{du,\,max} = 6$. In the figure 4.5(b) and 4.6(b), we represent the error bound $\Delta_N(\boldsymbol{\mu})$ for $1 \leq N \leq N_{pr/du,\,max}$.



(a) Chosen parameters. (b) Error bound Δ_N.

Figure 4.5: Sample and error bound for the primal problem.



(a) Chosen parameters. (b) Error bound Δ_N.

Figure 4.6: Sample and error bound for the dual problem.

4.4.4 Output

In this section, we study the output $s_2(t;\boldsymbol{\mu}) = \displaystyle\int_0^T \left(\int_{\Omega_o(\boldsymbol{\mu})} u_o(t;\boldsymbol{\mu})\, d\boldsymbol{x} \right) dt;$ the average temperature in the duct, for different parameters $\boldsymbol{\mu}$. As we did for the thermal fin, we compare these outputs with the 2D case. For the computations, we took $\Delta_t = 0.05$ and $n_t = 150$ timesteps, then $t_f = 7.5$.

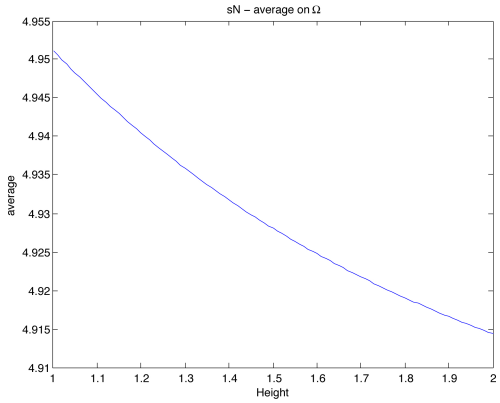
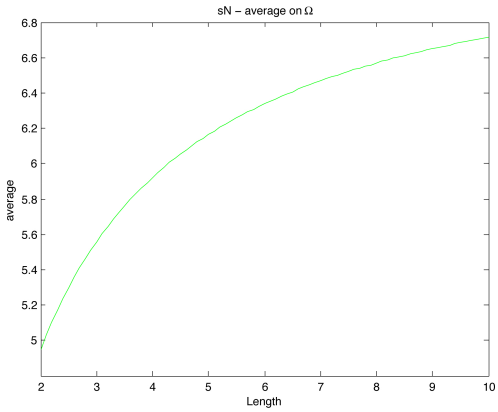
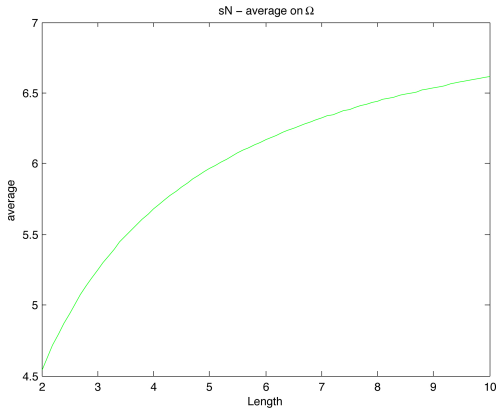


Figure 4.7: Output s_N for different values of μ_1 in the 3D case. $\mu_2 = 2$ and $\mu_3 = 1$

In Figure 4.7, we represent the evolution of the average temperature when the height of the "block" grows up. Looking at the values, we can consider that this graphics is a straight line. Physically, if we change the height only the geometry changes, but the average temperature is the same. Then if we normalize we obtain a straight line. In Figure 4.8, we see that the average



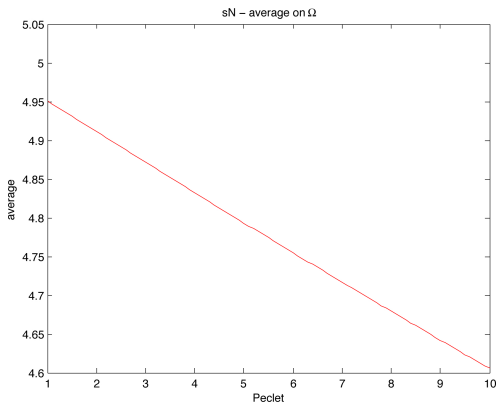
(a) Output s_N for different values of μ_2 in the 3D case.



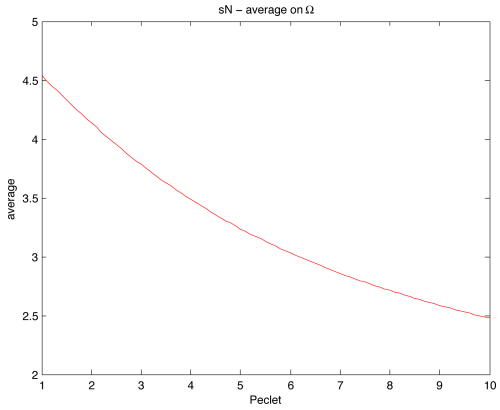
(b) Output s_N for different values of μ_2 in the 2D case.

Figure 4.8: Output with $\mu_1 = 1$ and $\mu_3 = 1$.

temperature increases when the length of the hot portion grows up. This result agrees with the physics. Indeed, if the hot zone is bigger, then the temperature will be bigger too.



(a) Output s_N for different values of μ_3 in the 3D case.



(b) Output s_N for different values of μ_3 in the 2D case.

Figure 4.9: Output with $\mu_1 = 1$ and $\mu_2 = 2$.

The Figure 4.9 shows that the average temperature decreases when the Péclèt grows up. Physically, if the Péclèt is bigger then there is more transport and then the temperature is lower. Here, we see a difference between the 3D and the 2D case. We can observe that the 3D case in less sensible to the variation of the Péclèt (looking at the difference between the first and the last value in the two cases). This difference is maybe due to two things: the difference between the geometries and the form of the convective field. Here, the convective field is a parabola which stretch out over the z-axis, then there is more transport than if we had a "dome" for the convective field.

4.4.5 Computational time

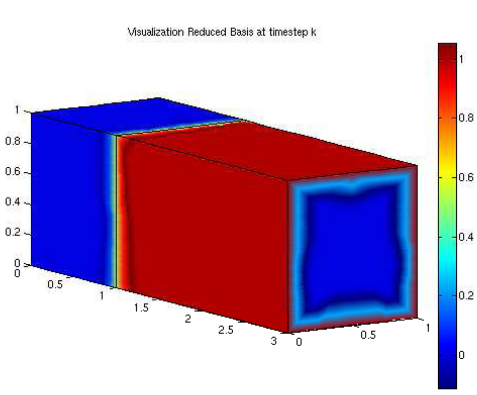
As in Section 3.4.6, we compute the different computational time that we defined. Here $\mathcal{N} = 2977$ and so :

$$\begin{aligned} t^{Offline}(\mathcal{N}) &= \frac{313.7}{9.68} = 32.4, \\ t^{Online}(\mathcal{N}, N) &= \frac{0.05}{9.68} = 0.005. \end{aligned}$$

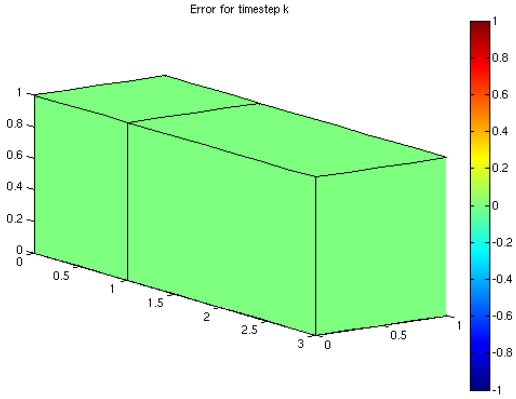
Then, if we want more than 32 solutions, the use of the RB method is more efficient and recommended. Here, looking at t^{Online}, we see that the use of the RB method is very gainful. We can get faster Online evaluation (5‰ of computational costs in the Online evaluation).

4.4.6 Visualization

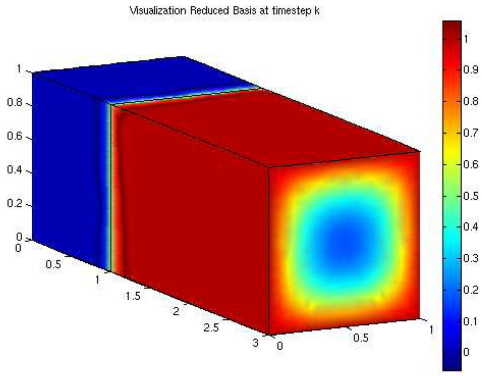
This section is dedicated to the visualization of reduced basis solutions. On Figure 4.10, we represent on the left the solution for $\boldsymbol{\mu} = (1, 2, 10)$ for different timesteps and control input $g = 1$. On the right side, we represent the error between the RB approximations and the FE solution. We can see that the error is very small and that the temperature on Γ_{o11} is getting close to 1.



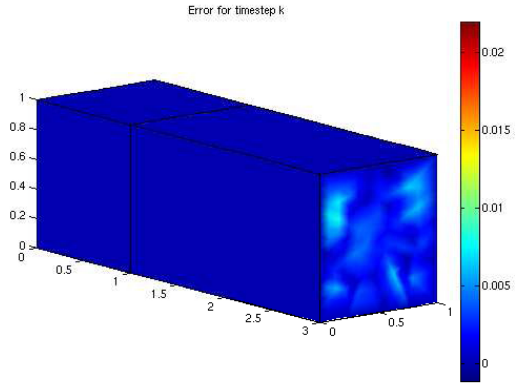
(a) Visualization at $t^0 = 0$. $\Delta_N^s = 0$.



(b) Error $u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})$.

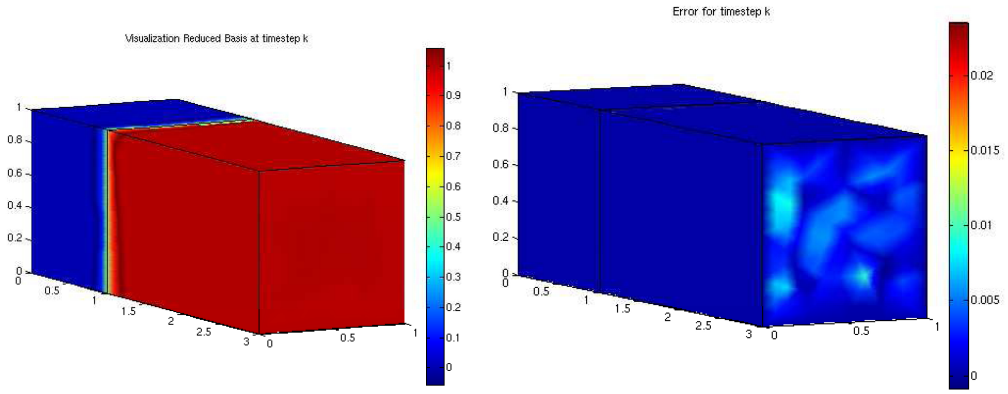


(c) Visualization at $t^6 = 0.3$. $\Delta_N^s = 0.0131$.



(d) Error $u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})$.

Figure 4.10: Example of representative solution and error for $\boldsymbol{\mu} = (1, 2, 10)$.

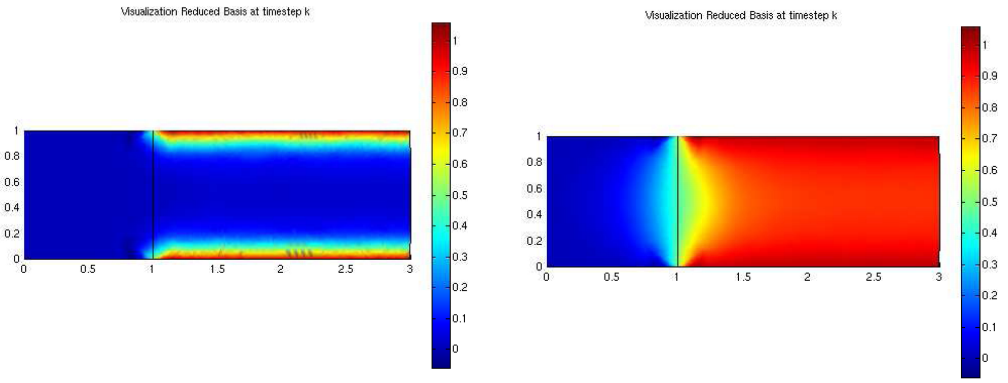


(e) Visualization at $t_f = 7.5$. $\Delta_N^s = 0.0708$.

(f) Error $u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})$.

Figure 4.10: Example of representative solution and error for $\boldsymbol{\mu} = (1, 2, 10)$.

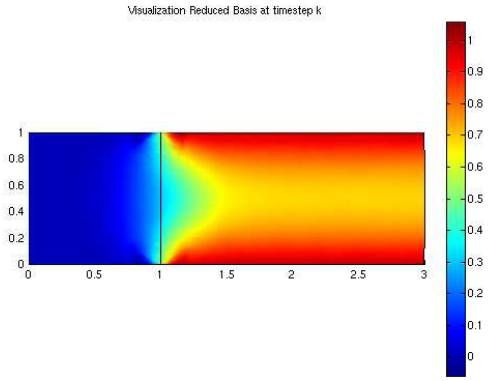
On Figure 4.11, we have taken a slice in the $x - y$ plan and we represent (on the left) a solution with dominant transport ($\mu_3 = 5$) and one with dominant diffusion ($\mu_3 = 0.1$).



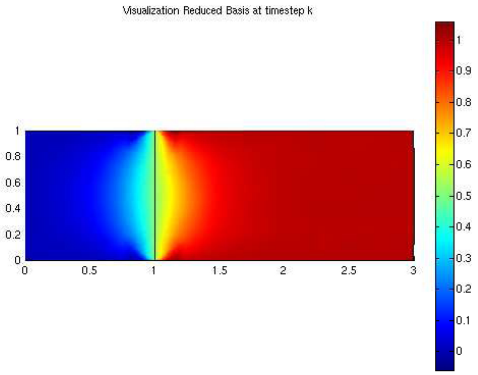
(a) Dominant transport $t^1 = 0.05$.

(b) Dominant diffusion $t^1 = 0.05$.

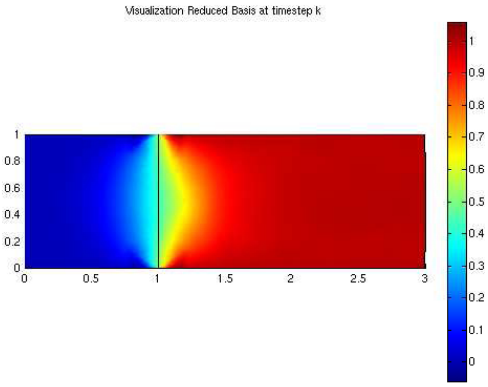
Figure 4.11: Figures on the left represent the solution for a dominant transport ($\mathbb{P}e = 5$) with $\mu_1 = 1$ and $\mu_2 = 2$. Figures on the right represent the solution for dominant diffusion ($\mathbb{P}e = 0.1$) with $\mu_1 = 1$ and $\mu_2 = 2$.



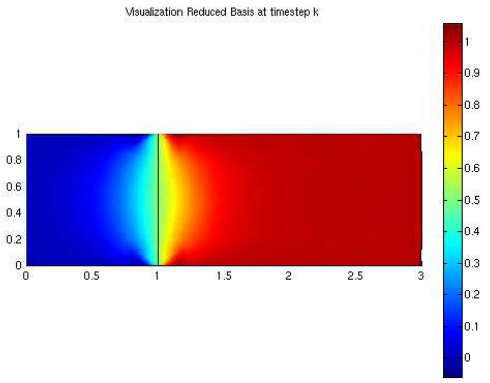
(c) Dominant transport $t^{10} = 0.5$.



(d) Dominant diffusion $t^{10} = 0.5$.



(e) Dominant transport $t_f = 7.5$.



(f) Dominant diffusion $t_f = 7.5$.

Figure 4.11: Figures on the left represent the solution for a dominant transport ($\mathbb{P}e = 5$) with $\mu_1 = 1$ and $\mu_2 = 2$. Figures on the right represent the solution for dominant diffusion ($\mathbb{P}e = 0.1$) with $\mu_1 = 1$ and $\mu_2 = 2$.

Appendix A

Appendix

Here, we will provide some details about the implementation of the two problems treated in this work, i.e. the Thermal Fin and the Greatz problem, with the rbMIT and COMSOL software. Our attention will focus on two aspects: the running of the Offline part which concerns the programmer and the running of the Online part which concerns the user.

A.1 Offline part

Here, we will briefly explain how to use COMSOL and MATLAB to run the Offline part for elliptic and parabolic problems, more specifically how to run the Offline part of the Thermal Fin (for elliptic problem) and the Graetz problem (parabolic problem). The main task of COMSOL is to construct the different matrices A^q (and M^q for parabolic problems). The function which does this task is `probname_Get_matrix.m`, for the problem named `probname`. This function has to be modified at hands for each new problems. Once we have done this, we use the function `probname_Step1_coernoncompliant.m` to specify some properties/settings about the problems as the parameter numbers, the parametric set $\mathcal{D}$ and the problem *type* ('elliptic' or 'parabolic') through the field `pr_PROBDEF.type`. Then, we call the routine `probname_Step2_1_coer_noncompliant.m` to execute the SCM. Now, if we want to use the Greedy we use the `probname_Step2_2_coer_noncompliant.m`, otherwise to use the POD (only for elliptic problems), we use the function `probname_POD_Step2_2_coer_noncompliant.m`. For the dual problem we use `probname_outputname_Step2_2_coer_noncompliant.m` or `probname_POD_outputname_Step2_2_coer_noncompliant.m` for the Greedy or the POD respectively, where `outputname` is the name of the output (in our case 'average'). Before doing this, we have to create a file `probname_Get_The-ta_q.m` and `probname_outputname_Get_Theta_q.m` where we have to spec-

ify the coefficients $\theta_{a,f,m,l}$.
Then, to launch the Offline procedure, we can call the function `rbU_probname.m` and all the steps mentioned are launched.

A.2 Online part

The idea here is to give a little tutorial about the different functions that are available for output or visualizations for steady problems (as Thermal Fin) or time-dependent problem (as Graetz problem).

A.2.1 Elliptic problem

In this section, we will focus on steady problem. We will present the different function to evaluate the output or to visualize the solution. We will take as example the Thermal Fin problem.

Output

For stationary problems as the Thermal Fin, we can use the function `Online3D` to evaluate the output `sN` and the error bound `DeltaN`. More precisely,

```
[sN,DeltaN]=Online3D(probname,mu,outputname),
```

where `probname` is the name of the problem, in our case '*TFIN3D*', `outputname` is the name of the output, here '*average*', the variable `mu` is the vector of the parameter, for example `mu=[0.3, 2, 5]`. A full syntax of `Online3D` is

```
[sN,DeltaN]=Online3D(probname,mu,outputname,type_res,N_pr,
                     N_du),
```

So, the user can also specify the size of the primal and dual basis through the variable `N_pr` and `N_du`. The variable `type_res` is a double (1, 2 or 3) which specify different kind of output :

- if `type_res`= 1, then `sN` and `DeltaN` are the standard output and error bound of our problem (here the average on the base of the spreader). We can use this function to evaluate the output if we have used the Greedy or the POD.

- if `type_res`= 2, then

`[u_RB, u_RB_FEM]=Online3D(probname,mu,outputname,2,N_pr, N_du)`,

returns the reduced basis approximation of the state solution, `u_RB` and also the reduced basis approximation of the state solution expressed in the FE basis, `u_RB_FEM`. Then, `u_RB` is a vector of dimension `N_pr` and `u_RB_FEM` is a vector of dimension $\mathcal{N}$.

- if `type_res`= 3, then

`[u_FEM]=Online3D(probname,mu,outputname,3)`

returns the FE solution `u_FEM`.

Visualization

- To visualize the RB approximation for the problem named `probname`, the output named `outputname` and for a parameter `mu`, just do

`Vis_RB3D(probname, mu, outputname,1)`.

- To visualize the FE solution, just do

`Vis_RB3D(probname, mu, outputname,2)`.

- Finally, to visualize the error between the RB approximation and the FE solution, write

`Vis_RB3D(probname, mu, outputname,3)`.

A.2.2 Parabolic problem

Now, we will focus on functions about time-dependent problems. The example that we consider is the Graetz problem.

Output

Since the resolution of the system (2.9) is the same in 2D and in 3D, then the function to evaluate the output is `Online_RB` the same that is in the rbMIT library. For an explanation see the documentation in [7]. Note that for the elliptic case we could use this function too with some modifications.

Visualization

- To visualize the RB approximation for the problem named `probname`, the output named `outputname` and for a parameter `mu`, write

```
Vis_TRB3D(probname, mu, outputname).
```

By default, the control input is equal to 1. If we want to specify another control input, we do

```
Vis_TRB3D(probname, mu, outputname,gt).
```

where gt is a time-dependent function.

- To visualize the FE solution write

```
Vis_TRB3D(probname, mu, outputname,gt,2).
```

- To visualize the RB solution at time t^k, write

```
Vis_TRB3D(probname, mu, outputname,gt,3,nk).
```

where nk is a double which indicates the timestep.

- Finally, to visualize the the error at time t^k, just do

```
Vis_TRB3D(probname, mu, outputname,gt,4,nk).
```

Bibliography

[1] *Reduced Basis at MIT. http://augustine.mit.edu/methodology.htm*, ©MIT, 2009.

[2] *Comsol Multiphysics* 3.5*a*. Guide available at *http://math.nju.edu.cn/help/mathhpc/document.htm*.

[3] V.S. ARPACI. *Conduction Heat Transfer*. Addison-Wesley, Reading, UK, 1966.

[4] V.S. ARPACI, P.S. LARSEN. *Convection Heat Transfer*. Prentice Hall, Englewood Cliffs, US, 1984.

[5] F. INCROPERA, D. DeWITT, T. BERGMANN, A. LAVINE. *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons, 2007.

[6] B.O. ALMORTH, P. STERN, F.A. BROGAN. *Automatic choice of global shape functions in structural analysis*. AIAA Journal, 16: 525-528, 1978.

[7] D.B.P. HUYNH, N.C. NGUYEN, G. ROZZA, A.T. PATERA. *Documentation for rbMIT Software: I. Reduced Basis (RB) for Dummies*. ©MIT, available at *http://augustine.mit.edu*. 2007.

[8] D.B.P. HUYNH, N.C. NGUYEN, G. ROZZA, A.T. PATERA. *Documentation for rbMIT Software: II. Time dependent problems*. ©MIT, 2008.

[9] D.B.P. HUYNH, N.C. NGUYEN, G. ROZZA, A.T. PATERA. *rbMIT Software*. ©MIT, Cambridge, 2007. Available at *http://augustine.mit.edu/methodology/methodology_ rbmit_ system.htm*.

[10] G. ROZZA, D.B.P. HUYNH, N.C. NGUYEN, A.T. PATERA. *Real-Time Reliable Simulation of Heat Transfer Phenomena*. ASME Summer Heat Transfer Conference, San Francisco, California, Paper HT2009-88212, 2009.

[11] D.B.P. HUYNH, G.ROZZA, S. SEN, A.T. PATERA. *A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants*. Comptes Rendus Mathématique, Volume 345: 3362-3366, 2007.

[12] A. MANZONI. *Shape Optimization and Optimal Control Problems in Viscous Flows Using Reduced Basis Method. A Short Guide for Programming with rbMIT and MLife*, Internal documentation, 2009.

[13] C. GUNTHER. *Reduced basis mathod for the shape optimization of racing car components*. EPFL, Master thesis, Chair of Modelling and Scientific Computing (CMCS), September 2008.

[14] A.K. NOOR, J.M. PETERS *Reduced Basis technique for nonlinear system analysis of structures*. AIAA Journal, 18(4): 455-462, 1980.

[15] A.T. PATERA and G. ROZZA. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations*. To appear in MIT Pappalardo Graduate Monographs in Mechanical Engineering, ©MIT, 2006-2009, Version 1.0.

[16] A. QUARTERONI. *Numerical Approximation of Partial Differential Equations*. Springer-Verlag, Berlin, 1997.

[17] A. QUARTERONI, R. SACCO, F. SALERI. *Numerical Mathematics*. Springer, Berlin, 2007.

[18] A. QUARTERONI. *Numerical Models for Differential Problems*. MS& A, Volume 2, Springer, 2009.

[19] A. QUARTERONI, G. ROZZA. *Numerical solution of parametrized Navier-Stokes equations by reduced basis methods*. Numerical Methods for PDEs, 23(4): 923-948, 2007.

[20] 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*. Archives of Computational Methods in Engineering, 15(3):229-275, 2008.

[21] G. ROZZA. *An Overview on Reduced Basis Approximation for Parametrized PDEs*. Institut d'Analyse et Calcul scientifique (IACS), Nr 04.2009, February 2009.

[22] G. ROZZA, N.C. NGUYEN, A.T. PATERA, S. DEPARIS. *Reduced basis method and a posteriori error estimators for heat transfer problems*. ASME Summer Heat Transfer Conference, San Francisco, California, Paper HT2009-88211, 2009.

[23] G. ROZZA. *Computational mechanics by reduced basis method*. Doctoral course given at EPFL in spring 2009.

[24] N.C. NGUYEN, G. ROZZA, A.T. PATERA. *Reduced Basis Approximation and A Posteriori Error Estimation for the Time-Dependent Viscous Burgers' Equation*. Calcolo, 46(3):157-185, 2009.

[25] N.C.NGUYEN, G. ROZZA, D.B.P.HUYNH, A.T.PATERA. *Reduced basis approximation and a posteriori error estimation for parametrized parabolic PDEs; Application to real-time Bayesian parameter estimation*. ©MIT, Department of Mechanical Engineering, Cambridge MA, USA. To appear (2010) in Computational Methods for Large Scale Inverse Problems and Uncertainty Quantification, John Wiley & Sons, UK. 2010.

Index