POLITECNICO DI MILANO

Facoltà di Ingegneria Industriale

Corso di Laurea in Ingegneria Aeronautica

Reduced Basis Method for 3D Problems governed by Parametrized PDEs and Applications

Relatore: Prof. Alfio Quarteroni

Correlatore: Dr. Ing. Gianluigi Rozza

Tesi di laurea di:

Alberto Angelo Trezzini Matr.: 721490

Anno accademico 2009/2010

Alberto Angelo Trezzini: *Reduced Basis Method for 3D Problems governed by Parametrized PDEs and Applications,* Master thesis, June 27, 2010

to my family. . .

ABSTRACT

In this thesis we will deal with the creation of a Reduced Basis (*RB*) approximation of parametrized Partial Differential Equations (*PDE*) for three-dimensional problems. The the idea behind *RB* is to decouple the generation and projection stages (Offline/Online computational procedures) of the approximation process in order to solve parametrized (*PDE*) in a fast, cheap and reliable way. The *RB* method, especially applied to 3D problems, allows great computational savings with respect to the classical Galerkin Finite Element (FE) Method. The standard *FE* method is typically ill suited to (i) iterative contexts like in optimization, sensitivity analysis and many queries in general and (ii) real time evaluation. We consider both coercive and noncoercive *PDE*s. For each class we discuss the steps to set up a *RB* approximation, either from an analytical and a numerical point of view. Then we present the applications of the *RB* method to three different problems of engineering interest and applicability: (i) a steady thermal conductivity problem in heat transfer; (ii) a linear elasticity problem; (iii) Stokes flows with emphasis on geometrical and physical parameters.

ABSTRACT

In questa tesi tratteremo la creazione di un'approssimazione a Basi Ridotte (*RB*) per Equazioni Differenziali alle Derivate Parziali (*EDP*) parametrizzate per problemi tridimensionali. L'idea alla base del metodo *RB* é quella di disaccoppiare le fasi di generazione e di proiezione del processo di approssimazione (procedura computazionale Offline/Online) in modo da risolvere *EDP* parametrizzate in modo veloce, accurato e affidabile. Il metodo *RB*, specialmente se applicato a problemi 3D, permette un grande risparmio computazionale rispetto al classico Metodo Galerkin-Elementi Finiti (*EF*). I metodi *EF* non sono adatti in un contesto iterativo, come (i) ottimizzazioni e (ii) analisi di sensitività, dove una procedura iterativa e' richiesta. In questo lavoro considereremo due classi di *EDP* per modellare problemi coercivi e non coercivi. Per ciascuna classe discuteremo i dettagli per creare un'approssimazione *RB*, sia dal punto di vista analitico, sia dal punto di vista prettamente numerico. Infine presenteremo l'applicazione del Metodo *RB* a tre differenti problemi con un certo interesse ingegneristico e applicabilitá: (i) conducibilitá termica in trasmissione del calore, (ii) elasticitá lineare e (iii) flussi di Stokes con una certa enfasi sulla scelta dei parametri geometrici e fisici.

INTRODUCTION AND MOTIVATION

ENGINEERING POINT OF VIEW

Any engineering design can be summarized in the following statement: *Given a physical process and given a set of suitable* parameters*, the physical behaviour depends on, find the optimal values of these parameters in order to obtain a desirable behaviour of the process*.
In this work the attention is focused on the designer point of view, but it is also important to recall that there are many other applications of engineering interest, such as *real time* evaluation of the performance of a system or *sensitivity analysis* with respect to certain parameters, the behavior of the system depends on.
The *physical process* belongs to any field of engineering interest:

- heat and mass transfer,
- elasticity,
- acoustics,
- fluid dynamics,
- electromagnetism,
- etc...

or even in a broader sense to any quantitative disciplines (e. g., finance, biology, ecology and medicine) and their combinations.
The physical process is analyzed by the designer in order to find the best mathematical *model* able to describe the behavior of the *system*. As *model* we refer to a system of equations and/or other mathematical relationships able to "catch" the main properties of the process and to predict its evolution in time and/or space. In this stage the engineer introduces all the simplifications that the observation and a subsequent qualitative analysis suggest to take into account.
The analytical model is then constituted by two "blocks":

1. general laws
2. constitutive equations

In this work the attention will be focused on models whose general laws are those of *continuum mechanics* which appears as conservation/balance of suitable quantities (e. g., mass, energy, linear momentum, angular momentum, ecc...).

The constitutive relationships come from experimental evidence and depends upon the features of the process in exam. The result of the combination of these two blocks is often an equation or a system of *Partial Differential Equations* PDEs. This means that in the equations the unknowns will appear along with partial derivatives with respect to multiple variables (temporal or spatial). Tol solve differential equation it is also necessary, in order to obtain the closure of the problem, specify a suitable set of boundary conditions.
Therefore the parameters in the process can be of two kind:

- physical, within this category we have:
  - coefficient of constitutive equation for the particular physical process addressed;
  - non-dimensional numbers;[1]
  - boundary conditions imposed;
- geometrical.

The desired behavior could be a particular performance of the system, such as an average temperature in a thermal block, a maximum displacement of a loaded beam, a level of vorticity in a flow-field and so on, depending on the particular process addressed.
Finally, the optimal configuration of these parameters can be found through an iterative optimization process in which a suitable *cost functional*, that depends upon the particular performance desired, has to be minimized.
To wit, the designer is interested to evaluate iteratively the *input-output* relationship to evaluate the cost functional.
since the solution of *PDEs* by classical discretization methods like finite elements, spectral methods or finite volumes, tipically involves thousands (in some cases millions) of degrees of freedom (DOFs) to obtain a "good" solution. Therefore a single evaluation of the input-output relationship is very expensive and at last, in most cases, not suitable in a *many-query* context on which the design strategy is based on.
In this context it is necessary to develop suitable *Reduced Order Modelling* ROM techniques that reduces the cost and times of the computations. *Reduced Basis Method* (RB) is one of them and this work will focus on this method that, as it will be shown, is able to reduce the computational cost of *orders of magnitude*. Moreover the RB method is a *certified* and *reliable* method because in addition to an output calculation is able to provide a *rigorous a-posteriori error estimators* on the *"exact" solution* [2].
An important remark is constituted by the fact that all the procedure is built upon a *reference domain* on which a *suitable discretization* is constructed. Then there is no need to re-discretize the domain at each iteration, re-building a mesh, or deforming the domain itself even in the case

1 It is always necessary to write the *PDEs* in a non-dimensional version in order to highlight the actual physical dependence on the parameters.
2 The discussion on our assumption of "exact" will be given in Chapter 1.

of *parameter-dependent geometry*.

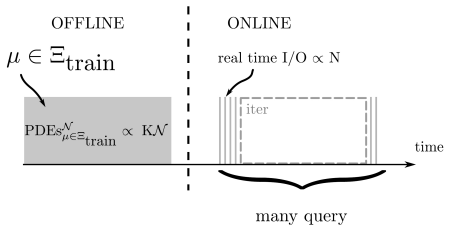
OVERVIEW ON REDUCED BASIS

As already said the *RB* method is a *reduced order method* that is able to reduce the complexity of a system without loss of information or accuracy of the results thanks to the *rigorous error bound* provided and by the properties of the *Galerkin* projection, see [PR09, RHP08].
This method does not replace an existing discretization method but "works in collaboration with it" and upon it.
In this work the discretization method adopted for the applications is the *finite element method* (FE): this choice does not constitute a limitation because the *RB* method is built over the *user-defined* assumption of *"truth"*. The choice of the correct method able to describe the physical process is demanded to the user and it will not be treated here.
The idea is to start with *FE* basis of dimension $\mathcal{N}$ and then construct a *RB* basis whose dimension N is much smaller than the former, so that $N << \mathcal{N}$.
The power of the *RB* lies in the splitting of the procedures into two parts:

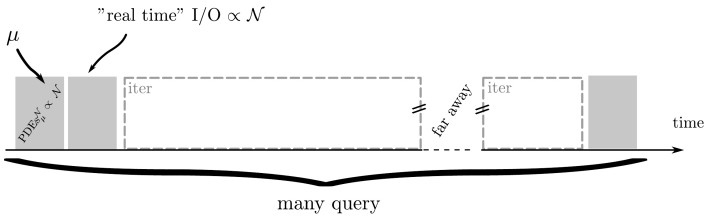
1. an Offline phase,
2. an Online phase,

where the former is $\mathcal{N}$ dependent and computed once, whereas the latter is N dependent and allows a *fast*, *cheap* and *reliable* input-output evaluation.
The role played by this decomposition is immediately clear taking in consideration for example an optimization process. The optimal configuration can be found thanks to an iterative process, e. g. the *Newton's gradient method* in which a *PDEs* solution is needed at every step of iteration. In this context the *RB* advantage is that the evaluation of the solution for every step is order of magnitude smaller; furthermore the real time evaluation of the *PDEs* solution is possible, unlike as in the *FE* case.
The splitting procedure is possible if the weak formulation of the *PDEs* can be expressed in an *affine parameter decomposition* (see Chapter 1). This is one of the *key-point* of the procedure: all the parametric dependences of the *PDEs* are actually separated by the non-parametric part, this allows to compute the latter just in a *reference domain*, whereas the former can be computed several times with a very cheap computational burden, see [Roz09].
A graphical sketch of this idea is shown in Figure 1. The *RB* method is depicted in the upper figure , whereas the *FE* classical method is depicted in the lower one.

In the *RB* sketch it appears that the *Offline* computational burden that is proportional to $K\mathcal{N}$, where $\mathcal{N}$ is the number of degrees of freedom for



(a) *RB* method



(b) *FE* method

Figure 1: *RB* computational saving

the *FE* problem and K is a positive constant that depends upon many ingredients that will be in depth discussed in Chapter 3.

RB motivation

It is worth to recall that in the *RB* methodology the so-called *Offline* part is very expensive but in a *real time* or in a *many-query context* the most important part is played by the *online* evaluation of the *input-output relationship* that in the *RB* methodology this is *very cheap*.
In addition dealing with parametrized equations it is clear that in many cases it is possible to restrict the attention on a tipically *smooth* and rather *low-dimensional parametrically induced manifold* of the functional space, that is the set of fields engendered as the input varies over the parameter domain, [PR09, RHP08]. In Section 1.2 of Chapter 1, a more involved discussion over these concepts will be given, which are the key points the reduced basis method has been built upon.

Thesis motivation

To date, the works on the *RB* method has been mostly aimed to lay the groundwork for a full comprehension of all its mathematical aspects. There are of course still open issues, such as an *a-priori convergence proof* for problems with more than one parameter [PR09].
Nonetheless, in the very few past years, dedicated software has been developed to implement the *RB* methodology. In particular we mention the `rbMIT` software, that we have used in this thesis to exploit the creation of

our *RB* system.
Unfortunately, the softwares available just deal with the 2-*dimensional case*, then our work will be devoted to create a procedure to exploit the creation of a *RB* approximation in the 3-*dimensional case* for *steady* problem.
The very first steps in this direction has been taken by F. Gelsomino during his master thesis work at the *EPFL*, focusing on scalar time-dependent problems, see [Gel10].
Our *innovative contribution* will consists in dealing with:

1. *3D* scalar problems with *higher parametric complexity* (*heat transfer applications*),
2. *3D vectorial problems* (*linear elasticity problem*),
3. an extension to *3D non-coercive problems* (*Stokes flow*).

Thesis outline

In Chapter 1 the motivations and scopes of the *RB* method will be discussed along with a briefly historical perspective, in Chapter 2 we will recall some mathematical generalities and then we will provide the abstract formulation for coercive and non-coercive problems.
In Chapter 3 the steps for the generation of the *RB* approximation spaces for the solution of parametrized *PDEs* will be explained. Later, in Chapter 4, the *affine geometry preconditions* will be presented focused on the 3D case, necessary to allow a fully decoupling between *Offline* and *Online* procedures.
In Chapter 5 we will provide a briefly insights on the software used to exploit the creation of the RB approximation.
In Chapter 6 the *RB* approximation of a heat transfer application will be addressed, then in Chapter 7 we will deal with a linear elasticity problem and finally in Chapter 8 we will treat a Stokes flow.

ACKNOWLEDGMENTS

This thesis work would not have been possible without the support of many people. I wish to express my gratitude to my advisor, Prof. Dr. Alfio Quarteroni who has enabled my exchange period at the *EPFL*. Deepest gratitude is also due to my co-advisor, Dr. Gianluigi Rozza who was abundantly helpful and offered invaluable assistance, support and guidance and without his knowledge and assistance this work would not have been successful. Special thanks also to all the MATHICSE crew: Paolo Crosetto, Matteo Lombardi, Andrea Manzoni, Laura Iapichino and Matteo Lesinigo for sharing the literature and invaluable assistance. I wish also to thanks Fabrizio Gelsomino for our funny and fruitful discussion over the *RB* method and `COMSOL` and all the Gionnis (or Gionnai): Anwar Koshakji and Davide Lupo Conti for their kindness and support. I wish to express my love and gratitude to my beloved families; for their understanding & endless love, through the duration of my studies. I also wish to express my love to Marta (Biga for friends) and give her a huge kiss for his patient and tenderness with which has supported me during these three las years; she has been my real strength! Last but not least, I wish to thank all my friends with whom I have spent the best days of my life, in particular I wish to thank: Colo, Bitta, Bighi, Catta, C and Scaso for them constant presence and their truthful friendship.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

FE Finite Element Method

PDEs Partial Differential Equations

DOFs Degrees Of Freedom

RB Reduced Basis

ROM Reduced Order Modelling

TB Thermal Block

EB Elastic Block

VP Viscous Pump

RHS Right Hand Side

1 INTRODUCTION

In Section 1.1 of this chapter the motivations and scopes of the *RB* method will be introduced.
In Section 1.2 the opportunities that can be gainfully exploited by the *RB* method are discussed; this will be done focusing even on historical perspective and on future developments related to this topic.
Sections 1.3 and 1.4 are devoted to applications of *RB* method in several fields of engineering interest.

1.1 A BRIEF INTRODUCTION TO RB

In the past few years, thanks to the increased computational performances it has been possible to use numerical simulation in the very first steps of design for a very wide spectra of fields.
Unfortunately despite this hardware improvement, the greater part of engineering problem involves the solution of partial differential equations, furthermore in a *design context* the number of solution for various configurations of attempt can become *very large* and eventually impracticable. Therefore it is necessary to develop techniques that are able to reduce the complexity of the system without a loss of information or accuracy of the results. The *RB* method is a promising approach to respond to this needs moreover this method is not only *rapid* and *efficient*, but also provides a reliable solution of partial differential equations thanks to a certified *a-posteriori error bound*.
This method provides a useful tool for engineers, in fact thanks to the very low cost of the input-output relationship evaluation, the design procedure can be enriched with highly accurate numerical simulation from the very first steps.

1.2 RB SCOPE AND HISTORICAL PERSPECTIVE

The *real-time* and *many-query contexts* represent not only computational challenges but also computational opportunities.
It is possible to identify two key opportunities that can be gainfully exploited with *RB* method [RHP08]:

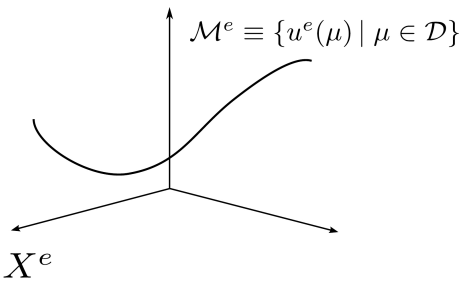
- **Opportunity I**
  In the parametric setting, the attention is restricted to a typically

smooth and rather *low-dimensional parametrically induced manifold*: the set of fields engendered as the input varies over the parameter domain; in the case of single parameter, the parametrically induced manifold is a one-dimensional filament within the infinite dimensional space which characterizes general solutions to the PDEs. Clearly, generic approximation spaces are *unnecessarily rich* and hence *unnecessarily expensive* within the parametric framework.

- **Opportunity II**
  In the *real-time* or *many-query contexts*, in which the premium is on marginal cost (or equivalently asymptotic average cost) per *input-output* evaluation, we can accept greatly increased pre-processing or *Offline* cost, not tolerable for a single or few evaluations, in exchange for greatly decreased *Online* (or deployed) cost for each new/additional input-output evaluation, see [RHP08]. Clearly, resource allocation typical for *single-query* investigations will be far from optimal for many-query and real-time exercises. We shall review the development of RB methods in terms of these two opportunities.

1.2.1 Opportunity I

Reduced Basis discretization is, in brief, a Galerkin projection on an N-dimensional approximation space that focuses on the parametrically induced manifold identified in Opportunity I.
Initial work (about in 1980s) grew out of two related streams of inquiry: from the need for more effective, and perhaps also more interactive, many-query design evaluation, for example [FM71] considers linear structural examples; and from the need of more efficient parameter continuation methods, [ASB78, NP80, Noo78, Noo82] consider nonlinear structural analysis problems.
Some modal analysis techniques proposed in the same years, *e. g.* [Nag79] deals with geometrically nonlinear behavior, are closely related to RB notions.



(a) X^e space

Figure 2: Parametrically induced manifold

At the very first moment the reduced basis method arises from the study of nonlinear elasticity problem. The development of the *RB* method has been mainly due to the engineering needs to obtain a very efficient tool in the design context.

RB *born as efficient tool to treat nonlinear elasticity problem*

The following decade saw further expansion into different applications and classes of equations, see for example [IR97] for a work dealing with

the control of PDEs, and [Pet89] for applications of reduced order methods to fluid dynamics and incompressible Navier-Stokes PDEs. However, in these early methods, the approximation spaces tended to be rather local and typically rather low dimensional in parameter (often a single parameter). In part, this was due to the nature of the applications taken in account (parametric continuation), but it was also due to the *absence* of *a-posteriori error estimators* and *effective sampling procedures*. In fact the absence of this kind of techniques did not allow a *certified* an *accurate* prediction of the error between the *"real" solution*[1] and that obtained by means of the reduced order model; the lack of a rigorous error certification is unacceptable for example in a safe engineering context, in which the reliability is an imperative.

then applied to fluid dynamic and control of fluid flows

Lack of a certified a-posteriori error bound: useless in an engineering context

Much current effort is thus devoted to development of (i) a posteriori error estimation procedures and in particular rigorous error bounds for outputs of interest [PRV+02] and (ii) effective sampling strategies, in particular for higher dimensional parameter domains such as in the works of [BTWG08, NVP05] and [Roz08].

The *a-posteriori error bounds* are of course mandatory for *rigorous certification* of any particular reduced basis (*Online*) output evaluation. However, the error estimators can also play an important role in efficient and effective (greedy) sampling procedures: the inexpensive error bounds permit us first, to explore much larger subsets of the parameter domain in search of most representative or best *snapshots*, and second, to determine when the basis functions are enough to bound the error within a certified interval.

The a-posteriori error bound *plays a dual role in the OFFLINE/ONLINE computation*

The most used sampling methods are (i) the *Greedy sampling procedure* and (ii) the *POD* (proper orthogonal decomposition): this two procedures differ under some aspects that will be discussed in Section 3.4. It is worth to anticipate that the *Greedy* procedure is optimized for higher dimension of the parameter space, while the *POD* procedure is better suited for one dimensional (typically time) domain, [AK99, KV03, Rav02, WP02].

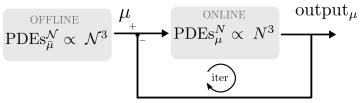
1.2.2 Opportunity II

Early work on the reduced basis method took into consideration the Opportunity II, but was not able to fully decouple the underlying standard *FE* discretization.

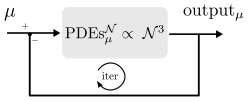
More precisely, often the *Galerkin stiffness equations* for the reduced basis system were generated by direct appeal to the high dimensional *FE* representation: essentially, pre and post multiplying the FE stiffness system by rectangular basis matrices; as a result the computational saving offered by the model reduction was not fully exploited in [Noo78, Por85] and [PL87]. The complete decoupling between the reduced order model and

1 "Real" here means the assumption of faithfulness achieved with the standard discretization method, see section 2.3.2 in the chapter 2 for further explanation on the hypothesis and on the consequences of this choice

the standard discretization model is one of the crucial point on which much of the current work has been devoted at that time.
In this work we will denote with $\mathcal{N}$ the computational complexity of the standard discretization and with N the *RB* complexity. This opportunity has been exploited thanks to an *Offline/Online procedure* that is possible in a context of an *affine parameter dependence* of the operators constituting the *PDEs*; this important concept will be recalled in Section 2.2.
One of the benefit related to this procedure is showed in figure 3; it is clear that in a many-query context, such as an optimization design process, or in a real-time closed loop control the split allows a much more faster sub-iteration (*I/O* evaluation), that should not be possible in the classical *FE* discretization nor in the reduced order methods that not implementing this efficient tool. In fact in the FE case (Figure 3b) a subiteration involves a cost proportional to $\mathcal{N}$, whereas RB involves a cost proportional only to N.



(a) RB procedure



(b) FE procedure

Figure 3: Offline/Online splitting

The Offline-Online idea is quite self-apparent and it has been treated often [IR97, JA04, Pet89], nonetheless the idea/application of *a-posteriori error bound* is more involved and recent [HRSP07, PRV+02, PRVP02].
Actually even in the case of a *non-affine parameter dependence* the development of the procedure is yet possible, although more complex procedures, that have been estabilished in the few last years [MYNA04, GMNP07], are needed in order to turn the non-affine form into an approximated affine problem.

1.3 PROBLEM ADDRESSED

This thesis will deal with *linear output functional* and *affinely parametrized linear coercive* or *non-coercive PDEs*: these classes of problems, although the former relatively simple, are relevant to many important applications in various field of engineering interest.
Although this thesis focuses on these classes of problems, the reduced basis is much more general and is able to treat nonaffine problems [GMNP07] and parabolic equations [AK99].
classes of problem addressed by the RB
Furthermore, the *RB* method can be used for nonlinear equations such as the incompressible (quadratically non-linear) *Navier-Stokes* equations, finally even the hyperbolic equations are subject of study, there are proofs

which demonstrate that RB approximation and a posteriori error estimation can be applied, although up to now there are still many issues related to *smoothness* and *stability* [HO08, PR07].
The applications chosen in this work deals with:

Parametrized linear coercive and non-coercive elliptic PDEs

1. transport equation: thermal conduction,
2. continuum mechanics: linear elasticity,
3. fluid dynamics: Stokes flow;

these problems proves a convenient expository vehicle for the methodology exploitation and are already suitable, as it will be seen, to arise a number of very specialized and involved engineering problems.

1.4 FROM 2D TO 3D

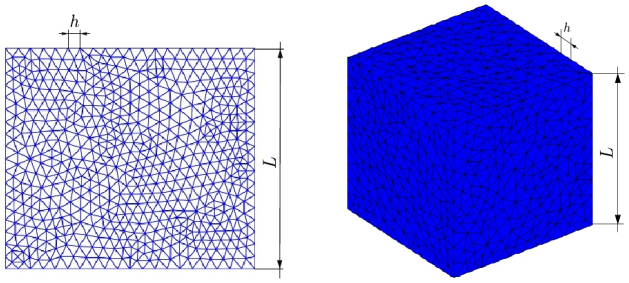
Up to now the *RB* method has been developed mainly for *2D problem*, our work constitutes the first steps in the natural extension and prosecution of the *RB* work in the 3D frame. Only few works have been specialized on this topic; the first step in this direction has been taken by F. Gelsomino who has treated a 3D *scalar elliptic coercive* equation and a *parabolic* equation with relatively simple geometry and few parameters.
The goal of this thesis is to prove, thanks to numerical evidence, that the methodology is also efficient and reliable for more involved and complex *3D applications*.
In addition more complex geometries with a greater number of parameters will be considered.
The upgrade to the 3D problem case may be quite complex because in the most cases the physical problem can not leave aside the third spatial dimension, see for example the viscous flow around a body or a thermal conduction problem in which the material has an anisotropic conductivity in the three spatial coordinates.

Focus on 3D application

This thesis is focused on real engineering applications. Some real applications will be exploited providing for each one a detailed discussion on:

1. Physical evidence,
2. Mathematical modelling,
3. *RB* approach,
4. Analysis of reliability and accuracy of the *RB* method.

This will allow us to demonstrate the power and efficiency of the method applied to several fields of engineering interest.



(a) 2D discretization **(b)** 3D discretization

Figure 4: Geometric discretization complexity

1.4.1 3D case

The passage from a lower dimensional to a greater dimensional geometric manifold is natural from an analytical point of view: in fact the method is already suitable for this extension.
The main peculiarities concerning 3D problems lie mainly in the numerical modelling:

1. much greater computational efforts,
2. more involved geometrical preconditions,
3. absence of an already available 3D software.

1- The first problem is quite self-evident, than it will be treated here briefly.
In the numerical solution of a *PDEs* the computational burden is linked to the cubic power of the degrees of freedom chosen to discretize the problem, than a 3D application involves computational costs orders of magnitude greater than the 2D case.

DOFs proliferation in the 3D case

This can be shown thanks to a simple example. Taking in account a squared geometry (see figure 4) with an edge that measures L, if a mesh is built on a cube with the same face and the same discretization h over the edges, then:

- **2D** case: DOFs $\propto \frac{L^2}{h^2} = N^2 \Rightarrow$ burden $\propto N^{2^3} = N^6$
- **3D** case: DOFs $\propto \frac{L^3}{h^3} = N^3 \Rightarrow$ burden $\propto N^{3^3} = N^9$

This extremely simple example shows that, dealing with the 3D case, the computational burden is order of magnitude greater then the 2D case, therefore in this case *RB* methodology is (if not essential) very useful in order to reduce computational cost in a reliable way.
Concerning the other two issues, it will be provided here just a brief remark, because the implications of this two aspects will affect the whole

numerical approach.

2- As already said, one of the most involved operation is carry out the affine geometric decomposition that, in the 2D is possible to deal with *rbMIT* package [PR09][2]. , while the 3D case has to be managed manually.

affine geometric decomposition issue

3- Another issue dealing with the 3D case lies in the fact that in the 2D case in the whole *RB* methodology was managed by the *rbMIT* package that works in the `Matlab` environment , while in this case the pre-process phase of *FE* matrices assembling has to be done with a *third-part software*. In this thesis the software chosen to carry out this phase has been `Comsol multyphisics`, [Com07b, Com07a], because it offers a great user versatility in the definition of the *FE* discretization as it will be shown in the chapter 5.
The last two remarks (**1-**, **2-**) will be clarified in Chapter 4 as regard the *affine geometric decomposition* and in Chapter 3 as regard the matrices assembling.

rbMIT built-in software treats only 2D case

2 The software can be downloaded at `http://augustine.mit.edu/methodology/methodology_rbMIT_System.htm`

2 PARAMETRIZED ELLIPTIC PDES

In the first part of this chapter, we will briefly introduce some generalities about *parametric* bilinear form, *parametric* linear functional, coercivity and *inf-sup* constants. In particular the abstract formulation for coercive and for non-coercive problem will be reported.

2.1 PARAMETRIC OPERATORS

In this section, definitions and properties about *parametric bilinear* and *bilinear forms* will be introduced. The conditions for the *well-posedness* of this two classes of problem are different, therefore the *coercivity constant* for the former case and the *inf-sup constants* for the parametric bilinear form will be introduced.
The theory presented here is available with further details and explanations in [PR09] for the coercive case, in [PR09, RV06] for the non coercive case.
The basic concept of functional analysis concerning Cartesian product, functional norms, bilinear forms and dual spaces are given as known, the reader can find a useful introduction to this topics in [QV97] and [Yos71].

2.1.1 Linear and bilinear parametric forms

We first introduce a closed bounded parameter domain $\mathcal{D} \subset \mathbb{R}^P$ with a typical parameter vector, or $P-$tuple, in $\mathcal{D}$ shall be denoted $\boldsymbol{\mu} = (\mu_1, \dots, \mu_P)$.
We assume that $\mathcal{D}$ is suitably regular.
It is now necessary to introduce some definition that resembles the classical definition for non-parametric linear operators.

Definition 2.1. *Let Z be an inner product space over $\mathbb{R}$, $b : Z \times Z \times \mathcal{D} \to \mathbb{R}$ is a* parametric bilinear form *if, for all $\boldsymbol{\mu} \in \mathcal{D}$, $b(\cdot, \cdot; \boldsymbol{\mu}) : Z \times Z \to \mathbb{R}$ is a bilinear form, i.e. for any $\alpha \in \mathbb{R}$ and for any $w, v, z \in Z$:*

$$b(\alpha w + v, z; \boldsymbol{\mu}) = \alpha b(w, z; \boldsymbol{\mu}) + b(v, z; \boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{D}$$

and for any $\beta \in \mathbb{R}$ and for any $w, v, z \in Z$:

$$b(w, \beta v + z; \boldsymbol{\mu}) = \beta b(w, v; \boldsymbol{\mu}) + b(w, z; \boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{D}.$$

Definition 2.2. *A parametric bilinear form* $b : Z \times Z \times \mathcal{D} \to \mathbb{R}$*is* symmetric *if, for any* $v, w \in Z$*:*

$$b(w, v; \boldsymbol{\mu}) = b(v, w; \boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{D} \tag{2.1}$$

A parametric bilinear form $b : Z \times Z \times \mathcal{D} \to \mathbb{R}$ *is* skew-symmetric *if, for any* $v, w \in Z$*:*

$$b(w, v; \boldsymbol{\mu}) = -b(v, w; \boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{D} \tag{2.2}$$

Starting from the definition 2.2 it is possible to define the symmetric and the skew-symmetric part of the bilinear form as it follows:

Definition 2.3. *Given a parametric bilinear form, we define:*

- *the symmetric part of* b *as:*

$$b_S(w, v; \boldsymbol{\mu}) = \frac{1}{2}\left(b(w, v; \boldsymbol{\mu}) + b(v, w; \boldsymbol{\mu})\right) \tag{2.3}$$

- *the skew-symmetric part of* b *as:*

$$b_{SS}(w, v; \boldsymbol{\mu}) = \frac{1}{2}\left(b(w, v; \boldsymbol{\mu}) - b(v, w; \boldsymbol{\mu})\right). \tag{2.4}$$

For the coercive case it is necessary to introduce the definition of the coercivity of the parametric bilinear form.

2.1.1.1 *Coercivity*

Definition 2.4. *We say that a parametric bilinear form* $b : Z \times Z \times \mathcal{D} \to \mathbb{R}$ *is* coercive *over* Z *if:*

$$\alpha(\boldsymbol{\mu}) = \inf_{w \in Z} \frac{b(w, w; \boldsymbol{\mu})}{\|w\|_Z^2} \tag{2.5}$$

is positive *for all* $\boldsymbol{\mu} \in \mathcal{D}$*.*

We can define $(0 <)\ \alpha_0 = \min_{\boldsymbol{\mu} \in \mathcal{D}} \alpha(\boldsymbol{\mu})$.

2.1.1.2 *Continuity*

Now it is possible to define the *continuity* of the parametric bilinear form in very similar way. We say that:

Definition 2.5. *A parametric bilinear form* $b : Z \times Z \times \mathcal{D} \to \mathbb{R}$ *is* continuous *over* Z *if:*

$$\gamma(\boldsymbol{\mu}) = \sup_{w \in Z} \sup_{v \in Z} \frac{b(w, v; \boldsymbol{\mu})}{\|w\|_Z \|v\|_Z} \tag{2.6}$$

is finite *for all* $\boldsymbol{\mu} \in \mathcal{D}$*.*

It is useful to define $\gamma_0 = \max_{\boldsymbol{\mu} \in \mathcal{D}} \gamma(\boldsymbol{\mu})$.

2.1.2 Inf-sup stability constant

2.1.2.1 *Definition*

Concerning the non-coercive case the coercivity constant for the well-posedness is replaced by the *inf-sup* stability constants [Bab71], it is useful to introduce its definition for parametric bilinear forms.

Definition 2.6. *Given a parametric bilinear form* $b : Z_1 \times Z_2 \times \mathcal{D} \to \mathbb{R}$, *we define the* inf-sup *constant as:*

$$\beta(\boldsymbol{\mu}) = \inf_{w \in Z_1} \sup_{v \in Z_2} \frac{b(w, v; \boldsymbol{\mu})}{\|w\|_{Z_1} \|v\|_{Z_2}}. \tag{2.7}$$

Here Z_1 *and* Z_2 *are* Hilbert spaces *with associated inner products and induced norms,* $(\cdot, \cdot)_{Z_1}$, $\|\cdot\|_{Z_1}$ *and* $(\cdot, \cdot)_{Z_2}$, $\|\cdot\|_{Z_2}$ *respectively.*

In general in general $\beta(\boldsymbol{\mu})$ is not necessarily strictly positive. Then:

Definition 2.7. *If there does exist a positive* β_0 *such that:*

$$\beta(\boldsymbol{\mu}) \geqslant \beta_0, \quad \forall\, \boldsymbol{\mu} \in \mathcal{D} \tag{2.8}$$

then we shall say that b *is* "inf-sup stable" *over* $Z_1 \times Z_2$.

2.1.2.2 *Supremizer operator*

We shall prove convenient to define a *"supremizing" operator* for the subsequent steps. This tool will be used dealing with the reduced basis procedure for non-coercive case (saddle point problem in this thesis); as it will be seen, its importance lies in the study of the convergence and of the algebraic stability of the system and in the construction of the reduced basis, [RV06].

Definition 2.8. *The* supremizer operator $T_\mu : Z_1 \to Z_2$ *associated with* b *is defined as:*

$$T_\mu w = \arg \sup_{v \in Z_2} \frac{b(w, v; \boldsymbol{\mu})}{\|w\|^2_{Z_2}} \tag{2.9}$$

In order to carry out the subsequent steps, an explicit representation of the supremizer operator is needed. This can be obtained thanks to the Cauchy-Schwarz inequality[1]; for any $w \in Z_1$

$$(T_\mu w, v)_{Z_2} = b(v, w; \boldsymbol{\mu}), \quad \forall\, v \in Z_2 \tag{2.10}$$

we observe that T_μ is linear.

1 We recall that the Cauchy-Schwarz states:

$$|(v, w)_Z| \leqslant \|v\|_Z \|w\|_Z \quad \forall\, v, w \in Z$$

2.1.3 Linear parametric form

Similarly as already done with the parametric bilinear form, we recall:

Definition 2.9. *Let Z be an inner product space over $\mathbb{R}$. $g : Z \times \mathcal{D} \to \mathbb{R}$ is a* parametric linear form *if, for all $\boldsymbol{\mu} \in \mathcal{D}$, and for any $w \in Z$, $g(\cdot\,;\boldsymbol{\mu}) : Z \times \mathcal{D} \to \mathbb{R}$ is a linear form. That is, for all $\alpha \in \mathbb{R}$, and for any $w, v \in Z$:*

$$g(\alpha w + v\,;\boldsymbol{\mu}) = \alpha g(w\,;\boldsymbol{\mu}) + g(v\,;\boldsymbol{\mu}) \quad \forall\, \boldsymbol{\mu} \in \mathcal{D} \tag{2.11}$$

2.1.3.1 *Continuity*

In order to ensure the well posedness of the problem, the continuity of the parametric linear form is needed. We say that:

Definition 2.10. *A parametric linear form g is* continuous *if, for all $\boldsymbol{\mu} \in \mathcal{D}$, $g(\cdot\,;\boldsymbol{\mu}) \in Z'$*

Z' denotes the *dual space*, that we recall is the space of all linear bounded functionals over Z. Note that the dual norm of a parametric linear form g, $\|g(\cdot\,;\boldsymbol{\mu})\|_{Z'}$, will of course be a (finite) function of $\boldsymbol{\mu}$ over D.

2.1.4 Coercivity eigenproblem

We recall here an additional problem which will be useful in order to evaluate a rigouros error bound as it will be seen in the section 3.6, [PR09, QSS00].
It is necessary to introduce an eigenproblem because in the subsequent analysis it will be useful to recognize that $\alpha(\boldsymbol{\mu})$ (and $\beta(\boldsymbol{\mu})$ in the non-coercive case) can be seen as the minimum eigenvalue of a *generalized eigenproblem*.

It is possible to rewrite 2.5 replacing the form b with his symmetric part, denoted with b_s:

$$\alpha(\boldsymbol{\mu}) = \inf_{w \in Z} \frac{b_s(w, w\,;\boldsymbol{\mu})}{\|w\|_Z^2} \tag{2.12}$$

it follows that $\alpha(\boldsymbol{\mu})$ can be expressed as a minimun eigenvalue.
It is useful to introduce the coercivity symmetric (generalized) eigenproblem associated with the parametric bilinear form $b : Z \times Z \times \mathcal{D} \to \mathbb{R}$.
Given $\boldsymbol{\mu} \in \mathcal{D}$, find the couple $(\chi, \lambda)_i(\boldsymbol{\mu}) \in Z \times \mathbb{R}$, $i = 1, \ldots, \dim(Z)$, such that:

$$b_S(\chi_i, v;\, \boldsymbol{\mu}) = \nu_i(\chi_i(\boldsymbol{\mu}), v)_Z \tag{2.13}$$

and

$$\|\chi_i(\boldsymbol{\mu})\| = 1 \tag{2.14}$$

of course it will be possible to sort the $\dim(Z)$ eigenvalues in ascending order such that: $\lambda_1(\boldsymbol{\mu}) < \ldots < \lambda_{\dim(Z)}(\boldsymbol{\mu})$.
It simply descends from 2.13 and 2.12 that if b is coercive, then $\alpha(\boldsymbol{\mu}) = \lambda_1(\boldsymbol{\mu}) > 0$.

2.1.5 Inf-Sup stability constant eigenproblem

Similarly as done in the coercive case, also in the non-coercive case an additional problem has to be handled in order to obtain an efficient error estimators and in this particular case even to obtain an algebraic well conditioned problem 3.6.2, [Roz08].

2.1.5.1 *Alternative expression for the inf-sup constant*

In this case it is necessary to take an intermediate step, that is rewrite the *supremizer* operator in a different fashion. It follows from the definition of the inf-sup constant 2.7, the definition of the supremizer 2.9 and 2.10 that:

$$\begin{aligned}
\beta(\boldsymbol{\mu}) &= \inf_{w\in Z_1}\sup_{v\in Z_2}\frac{b(w,v;\boldsymbol{\mu})}{\|w\|_{Z_1}\|v\|_{Z_2}} \\
&= \inf_{w\in Z_1}\frac{b(w,T_\mu w;\boldsymbol{\mu})}{\|w\|_{Z_1}\|T_\mu w\|_{Z_2}} \\
&= \inf_{w\in Z_1}\frac{(T_\mu w,T_\mu w)_{Z_2}}{\|w\|_{Z_1}\|T_\mu w\|_{Z_2}} \\
&= \inf_{w\in Z_1}\frac{\|T_\mu w\|_{Z_2}}{\|w\|_{Z_1}}
\end{aligned} \tag{2.15}$$

2.1.6 Inf-Sup eigenproblem

It readily follows from the *Rayleigh(-like) quotients* 2.15 that $\beta(\boldsymbol{\mu})$ can be easily expressed in terms of an eigenproblem.
Given a parametric bilinear form $b : Z_1 \times Z_2 \times \mathcal{D} \to \mathbb{R}$ and given $\boldsymbol{\mu} \in \mathcal{D}$, find the couple: $(\Xi,\lambda)_i(\boldsymbol{\mu}) \in Z_1 \times \mathbb{R}_{+0}$, $i = 1,\ldots,\dim Z_1$, such that:

$$(T_\mu \Xi_i(\boldsymbol{\mu}), T_\mu w)_{Z_2} = \lambda_i(\boldsymbol{\mu})\,(\Xi_i(\boldsymbol{\mu}), w)_{Z_1} \tag{2.16}$$

and

$$\|\Xi_i(\boldsymbol{\mu})\| = 1 \tag{2.17}$$

where $T_\mu w$ satisfies 2.10. The eigenvalues are then sorted in an ascending order such that $0 \leqslant \lambda_1(\boldsymbol{\mu}) \leqslant \ldots \leqslant \lambda_{\dim Z_1}(\boldsymbol{\mu})$.
The usual orthogonality between eigensolution holds; given two different eigensolution $\Xi_i(\boldsymbol{\mu})$, $\Xi_j(\boldsymbol{\mu}) \in Z_1$:

$$\big(\Xi_i(\boldsymbol{\mu}),\Xi_j(\boldsymbol{\mu})\big) = \delta_{ij}, \quad 1 \leqslant i,j \leqslant \dim Z_1 \tag{2.18}$$

where δ_{ij} is the *Kronecker-delta* symbol.

It follows from the definition of the inf-sup constant 2.15 and from the eigenproblem 2.16 that $\beta(\boldsymbol{\mu})$ can be evaluated as:

$$\beta(\boldsymbol{\mu}) = \sqrt{\lambda_1(\boldsymbol{\mu})} \tag{2.19}$$

corresponding to the square root of the smallest eigenvalues of the inf-sup eigenproblem, [QV97].

2.2 AFFINE PARAMETRIC DEPENDENCE

The *affine parametric dependence* of the bilinear form and of the linear functional is one of the most important ingredient in the offline/online decomposition and of course in the real-time input/output evaluation.
The idea is rather simple: split all the parametric dependent component by those parametrically independent.
In addition we recall also the parametric coercivity definition.

2.2.1 Affine parametric bilinear forms

With regard to the bilinear parametric forms the affine dependence states that:

Definition 2.11. *A parametric bilinear form* $b : Z \times Z \times \mathcal{D} \to \mathbb{R}$ *is* affine *in the parameter* $\boldsymbol{\mu}$ *if, for all* $v, w \in Z$*:*

$$b(w, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_b} \theta_b^q(\boldsymbol{\mu})\, b^q(w, v) \quad \forall\, \boldsymbol{\mu} \in \mathcal{D} \tag{2.20}$$

for some finite, preferably small, Q_b.

Here the $\theta_b^q(\boldsymbol{\mu}) : \mathcal{D} \to \mathbb{R}$ are (tipically very smooth) parameter- dependent functions, and $b^q(w, v) : Z \times Z \to \mathbb{R}$ are parameter-independent bilinear forms.

2.2.2 Parametric coercivity

In the scope of an affine dependence it is useful also to consider the parametric coercivity of the bilinear form.

Definition 2.12. *We say that an affine parametric (coercive) form* $b : Z \times Z \times \mathcal{D} \to \mathbb{R}$ *(definition 2.11) is* parametrically coercive *if,* $c \equiv b_s$ *(the symmetric part of* b*) admits an* affine development*:*

$$c(w, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_c} \theta_c^q(\boldsymbol{\mu})\, c^q(w, v) \quad \forall\, \boldsymbol{\mu} \in \mathcal{D} \tag{2.21}$$

that satisfies two conditions:

$$\theta_c^q(\boldsymbol{\mu}) > 0 \quad \forall w, v \in Z,\ \forall \boldsymbol{\mu} \in \mathcal{D},\ 1 \leqslant q \leqslant Q_b \tag{2.22}$$

and

$$c^q(v, v) > 0 \quad \forall v \in Z,\ 1 \leqslant q \leqslant Q_b \tag{2.23}$$

(Note that we suppose that each $c^q(w, v)$ is symmetric).

2.2.3 Parametric inf-sup condition

For the case of greatest interest in this thesis, in which b admits an affine representation, also the supremizer operator admits an affine development.

Definition 2.13. *Given $w \in Z_1$, we may express $T_\mu w$ as:*

$$T_\mu w = \sum_{q=1}^{Q_b} \theta_b^q(\boldsymbol{\mu})\, \mathbb{T}^q w \quad \forall \boldsymbol{\mu} \in \mathcal{D} \tag{2.24}$$

where the parameter-independent operators $\mathbb{T}^q : Z_1 \to Z_2$ are given by:

$$(\mathbb{T}^q w, v)_{Z_2} = b^q(w, v), \quad \forall v \in Z_2,\ 1 \leqslant q \leqslant Q_b \tag{2.25}$$

The proof of 2.24 is simple, for any $w \in Z_1$ and for any $\boldsymbol{\mu} \in \mathcal{D}$:

$$\begin{aligned}
\left(\sum_{q=1}^{Q_b} \theta_b^q(\boldsymbol{\mu})\, \mathbb{T}^q w,\, v \right)_{Z_2} &= \sum_{q=1}^{Q_b} \theta_b^q(\boldsymbol{\mu})\ (\mathbb{T}^q w, v)_{Z_2} \\
&= \sum_{q=1}^{Q_b} \theta_b^q(\boldsymbol{\mu})\, b^q(w, v) \\
&= b(w, v; \boldsymbol{\mu}) \\
&= (T_\mu w, v)_{Z_2} \quad \forall v \in Z_2
\end{aligned} \tag{2.26}$$

This decomposition shall prove useful in developing inf-sup lower bounds.

2.2.4 Affine parametric linear form

Similarly as done for the bilinear form it is worth to introduce the affine dependence for a linear bounded functional.

Definition 2.14. *We shall say that the parametric linear form $g : Z \times \mathcal{D} \to \mathbb{R}$ is affine in the parameter if, for any $v \in Z$:*

$$g(v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_g} \theta_g^q(\boldsymbol{\mu})\, g^q(v) \quad \forall \boldsymbol{\mu} \in \mathcal{D} \tag{2.27}$$

for some finite Q_g.

Once again, here the $\theta_g^q : Z \times \mathcal{D} \to \mathbb{R}$ are smooth parameter-dependent functions and the $g^q(v) : Z \to \mathbb{R}$ are parameter-independent bounded linear forms.
The affine representations 2.20 and 2.27 are not unique, though in general will exist minimum Q_b (in the former case) and Q_g (in the latter) terms of expansion able to describe the forms with an affine development.
Tipically the number of terms Q mainly depends on the complexity of the parameter-dependent geometry. This concept will be treated in the section 4.3.3 (chapter 4, pag.72) where the decomposition in the 2D and in the more involved 3D case will be considered.

2.3 ABSTRACT FORMULATION: COERCIVE PROBLEM

In this section, an abstract problem for *coercive* elliptic partial differential equations with affine parameter dependence will be introduced and in the next section the non-coercive, specialized to saddle-point problems, case will be treated. First instance the exact formulation (in weak form) of the problem will be presented, then a finite element discretization will be introduced in order to build the *"truth"* space on which the reduced basis will be built upon.
As mentioned in the introduction, the interest lies the evaluation of the *solution field* and the *output* that depend on the state equation which is solution of a *PDEs*.

2.3.1 Exact formulation

Let $\Omega \in \mathbb{R}^d$, $d = 1,2,3$ be a suitable physical domain with Lipschitz continuous boundary $\partial\Omega$. Let $\mathcal{D} \subset \mathbb{R}^P$ be the parameter domain.
Moreover let Γ, be a boundary measurable segments of $\partial\Omega$, over which we shall ultimately impose *Dirichlet boundary condition* on the components of the field variable. We next introduce a suitable scalar space Y_i^e, $1 \leqslant i \leqslant d$:

$$Y_i^e \equiv \left\{ v \in H^1(\Omega) \mid v|_{\Gamma_i^D} = 0 \right\} \tag{2.28}$$

in general $H_0^1(\Omega) \subset Y^e \subset H^1(\Omega)$. Clearly if $\Gamma = \partial\Omega$, then $X^e \equiv H_0^1$.
We then construct the space in which our *vector-valued* field variable shall reside as a Cartesian product:

$$X^e = Y_1^e \times Y_2^e \times \ldots \times Y_1^d$$

We equip X^e with an inner product $(v,w)_{X^e}$, $\forall v,w \in X^e$ and induced norm $\|w\|_{X^e} = \sqrt{(w,w)_{X^e}}$, $\forall w \in X^e$; any inner product which induces a norm equivalent to the $\left(H^1\right)^d$ is admissible.

Problem statement

It is now possible to state the problem in the "exact" space:
Let $a : Z \times Z \times \mathcal{D} \to \mathbb{R}$ be a *continuous coercive parametric bilinear* form, let $f : Z \times \mathcal{D} \to \mathbb{R}$ and l be a *parametric linear functional bounded over* X^e.
Given $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^P$, find $u(\boldsymbol{\mu}) \in X^e$ such that

$$a\left(u^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right) = f\left(v; \boldsymbol{\mu}\right) \quad \forall v \in X^e \tag{2.29}$$

and evaluate

$$s^e(\boldsymbol{\mu}) = l(u^e(\boldsymbol{\mu}); \boldsymbol{\mu}) \tag{2.30}$$

Here $s^e(\boldsymbol{\mu})$ is the output of interest, $s^e(\boldsymbol{\mu}) : \mathcal{D} \to \mathbb{R}$ is the input-output relationship and l is the linear *"output"* functional that links the input to the output through the field variable.
It follows from our hypothesis on a, f and l that the problem has a unique solution thanks to the *Lax-Milgram theorem* [QSS00, QV97].

Recalling the affine development of the bilinear form and of the linear functional (section 2.2), it is possible to write the operators in the following form; for any $\boldsymbol{\mu} \in \mathcal{D}$:

$$a(w, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \theta_a^q(\boldsymbol{\mu})\, a^q(w, v) \quad \forall v, w \in X^e \tag{2.31}$$

$$f(v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \theta_f^q(\boldsymbol{\mu})\, f^q(v) \quad \forall v \in X^e \tag{2.32}$$

$$l(v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_l} \theta_l^q(\boldsymbol{\mu})\, l^q(v) \quad \forall v \in X^e \tag{2.33}$$

for finite and *preferebly* small Q_a, Q_f, Q_l. We implicitly assume that the θ_a^q for $1 \leqslant q \leqslant Q_a$, θ_f^q for $1 \leqslant q \leqslant Q_f$ and θ_l^q for $1 \leqslant q \leqslant Q_l$ are simple algebraic expressions that can be readily evaluated in $\mathcal{O}(1)$ operations.

Compliant problem

In this section the problems considered will be *"compliant"*, i.e. :

1. $l(\cdot; \boldsymbol{\mu}) = f(\cdot; \boldsymbol{\mu})$
2. $a(w; v; \boldsymbol{\mu}) = a(v, w; \boldsymbol{\mu}) \quad \forall w, v \in X^e$

that is the *output* functional and the *load/source* functional are the same and the bilinear form is symmetric (e. g."compliancy" in linear elasticity). Considering these two hypothesis the problem 2.29-2.30 can be rewritten as follows.
Given $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^P$, find $u(\boldsymbol{\mu}) \in X^e$ such that

$$a\left(u^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}\right) = f\left(v; \boldsymbol{\mu}\right) \quad \forall v \in X^e \tag{2.34}$$

and evaluate

$$s^e(\boldsymbol{\mu}) = f(u^e(\boldsymbol{\mu}); \boldsymbol{\mu}) \tag{2.35}$$

2.3.2 Truth approximation

We focus the attention on the *"truth" approximation*. The reduced basis approximation will be built upon and the error will be measured relatives to this assumption of truth.
The role played by this assumption is very important; during the basis assembling and the error measuring the *RB* will completely "forget" the error between the *exact* solution and the *truth-assumption*. Then it is necessary to take some caution in order to ensure that this error remains suitably small for any given $\boldsymbol{\mu} \in \mathcal{D}$.
For analytical purposes, in this thesis we assume that no variational "crimes" are committed, therefore actually the *"truth"* takes the place of the *exact* statement. In this thesis the *standard finite element FE* approximation [Quao9] has been chosen to represent the truth and to measure the error in order to build the *RB* basis and evaluate the error bound for a given new set of parameter input $\boldsymbol{\mu}$.

2.3.2.1 *Galerkin projection*

We introduce a family of conforming approximation spaces $X^{\mathcal{N}} \subset X^e$ of dimension $\dim(X^{\mathcal{N}}) = \mathcal{N} < \infty$.
We then associate to our space a set of basis functions $\phi_k^{\mathcal{N}} \subset X^{\mathcal{N}}$, $1 \leqslant k \leqslant \mathcal{N}$, by construction, any member of $X^{\mathcal{N}}$ can be represented by a unique linear combination of the basis functions $\phi_k^{\mathcal{N}} \subset X^{\mathcal{N}}$.
Finally, we associate the inner products and induced norms $X^{\mathcal{N}}$ is equipped, denoted by $(v, w)_{X^{\mathcal{N}}}, \forall v, w \in X^{\mathcal{N}}$, and induced norm $\|w\|_{X^{\mathcal{N}}} = \sqrt{(w, w)_{X^{\mathcal{N}}}}$, $\forall w \in X^{\mathcal{N}}$.
This inner product, along with those related to the exact space, is explained in the subsequent section.

Inner product and induced norms

We now 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 respectively the *energy inner product* and the *energy norm* as:

$$\begin{aligned} ((w, v))_{\boldsymbol{\mu}} &= a(w, v; \boldsymbol{\mu}), \\ |||w|||_{\boldsymbol{\mu}} &= \sqrt{(w, w)_{\boldsymbol{\mu}}}, \end{aligned} \tag{2.36}$$

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

$$\begin{aligned} (w, v)_{X^e} &= ((w, v))_{\bar{\boldsymbol{\mu}}} + \tau (w, v)_{L^2(\Omega)}, \\ \|w\|_{X^e} &= \sqrt{(w, w)_{X^e}}, \end{aligned} \tag{2.37}$$

where τ is a negative real parameter and$(w,v)_{L^2(\Omega)} = \int_\Omega w\,v\,d\Omega$.
Remark 1: We note that in order to define our X^e-norm we have chosen a fixed valued of the parameter $\bar{\mu}$.
Remark 2: since $X^{\mathcal{N}} \subset X^e$, the inner products and the norms defined above are the same for the space $X^{\mathcal{N}}$.
The choice of $\bar{\mu}$ 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 [RHP08].

Problem statement

Now we can state the problem in the truth space taking the *Galerkin projection* of the problem 2.34-2.35; given $\mu \in \mathcal{D} \subset \mathbb{R}^P$, find $u(\mu) \in X^{\mathcal{N}}$ such that

$$a\left(u^{\mathcal{N}}(\mu), v; \mu\right) = f(v;\mu) \quad \forall v \in X^{\mathcal{N}} \tag{2.38}$$

and evaluate

$$s^{\mathcal{N}}(\mu) = f(u^{\mathcal{N}}(\mu);\mu). \tag{2.39}$$

Coercivity and continuity

We can define precisely the *exact* and the finite element *approximated coercivity constants* respectively, as:

$$\alpha^e(\mu) = \inf_{w\in X^e} \frac{b(w,w;\mu)}{\|w\|^2_{X^e}}, \tag{2.40}$$

$$\alpha^{\mathcal{N}}(\mu) = \inf_{w\in X^{\mathcal{N}}} \frac{b(w,w;\mu)}{\|w\|^2_{X^e}}. \tag{2.41}$$

From the coercivity hypothesis, we have that $\alpha^e(\mu) > \alpha_0$, $\forall\,\mu \in \mathcal{D}$; furthermore from our hypothesis on $X^{\mathcal{N}}$, that is a conforming space, we have that $\alpha^{\mathcal{N}}(\mu) > \alpha^e(\mu)$, $\forall\,\mu \in \mathcal{D}$. Than even after the approximation the problem remains coercive.
In the same way, the continuity constants are defined as

$$\gamma^e(\mu) = \sup_{w\in X^e}\sup_{v\in X^e} \frac{b(w,v;\mu)}{\|w\|_{X^e}\|v\|_{X^e}}, \tag{2.42}$$

$$\gamma^{\mathcal{N}}(\mu) = \sup_{w\in X^{\mathcal{N}}}\sup_{v\in X^{\mathcal{N}}} \frac{b(w,v;\mu)}{\|w\|_{X^{\mathcal{N}}}\|v\|_{X^{\mathcal{N}}}}, \tag{2.43}$$

once again from our hypothesis follows $\gamma^e(\mu) < \infty$ and $\gamma^{\mathcal{N}}(\mu) \leqslant \gamma^e(\mu)$, $\forall\,\mu \in \mathcal{D}$.

Well-posedness inheriting

The Galerkin approximation on $X^{\mathcal{N}}$ must satisfy the same conditions that the exact formulation satisfies over X^e. For the particular class of prob-

lems of interest in this section (elliptic coercive PDEs) the Galerkin formulation in fact *directly inherits* and even *improves upon* all the good properties of the exact formulation:

1. The dual norm of f over $X^{\mathcal{N}}(\subset X^e)$ is bounded by the dual norm of f over X^e;
2. symmetry is preserved;
3. a is coercive over $X^{\mathcal{N}}$ with:
$$\alpha^{\mathcal{N}}(\boldsymbol{\mu}) \geqslant \alpha^e(\boldsymbol{\mu}) \quad \forall\, \boldsymbol{\mu} \in \mathcal{D} \tag{2.44}$$
4. a is continuos over $X^{\mathcal{N}}$ with:
$$\gamma^{\mathcal{N}}(\boldsymbol{\mu}) \leqslant \gamma^e(\boldsymbol{\mu}) \quad \forall\, \boldsymbol{\mu} \in \mathcal{D} \tag{2.45}$$
5. the *affine expansions* for f and a are still valid for w, v restricted to $X^{\mathcal{N}}$;
6. a still satisfies the two conditions for parametric coercivity (2.2.2);

thus, for any $\mathcal{N}$ and associated $X^{\mathcal{N}}$, the Galerkin approximation preserves the *"parametrically coercivity* and *affine compliancy"* property.

2.4 ABSTRACT FORMULATION: NON-COERCIVE PROBLEM

In this section, the abstract problem for *non-coercive* elliptic partial differential equations with affine parameter dependence will be stated.
Concerning this class of *PDEs*, we will deal with the particular Stokes *problem*. This equations are of special interest as they model the incompressible flow of viscous fluids at low Reynolds.
Although the Stokes problem is a self-contained problem, it is also the first (main) step for the solution of the more general *nonlinear Navier-Stokes* equations.
The peculiarity of this class of *PDEs* is the loss of coercivity and to properly solve this kind of problem a more general property has to be ensured, the *inf-sup* stability condition, [Bab71, QV97].

2.4.1 Exact formulation

Let $\Omega \in \mathbb{R}^d$, $d = 1, 2, 3$ be a suitable physical domain with Lipschitz continuous boundary $\partial\Omega$. Let $\mathcal{D} \subset \mathbb{R}^P$ be the parameter domain.
Then let Γ, be a boundary measurable segments of $\partial\Omega$.

We start introducing a 3D vectorial "exact" space $V^e(\Omega)$ built as a cartesian product of a scalar space $X^e(\Omega) \in H^1_{\Gamma^D}(\Omega)$, such that

$$V^e = X^e_1 \times X^e_2 \ldots \times X^e_d \tag{2.46}$$

where:

$$H^1_{\Gamma^D} = \left\{ u \in H^1(\Omega) \mid u = 0 \ \text{ on } \Gamma^D \right\} \tag{2.47}$$

We also require a scalar space:

$$M^e = L^2(\Omega) \tag{2.48}$$

The space 2.46 will be used for the components of the velocity, whereas the space 2.48 will be used for the pressure.
Finally we require the cartesian space $X^e = V^e \times M^e$, thanks to this assumption the inner products and induced norms for these spaces are defined in the usual way (see Section 2.4.2). that is the components of the velocity are measured with an H^1 norm, whereas the pressure is measured with an L^2 norm, [QV97].

Problem statement

Let $a : V^e \times V^e \times \mathcal{D}$ and $b : V^e \times M^e \times \mathcal{D}$ be *parametric bilinear forms*, where b is *non-square*, moreover let $f : V^e \times \mathcal{D}$ and $g : M^e \times \mathcal{D}$ be parametric linear bounded functionals. The problem in the parametric weak formulation reads: find $(\mathbf{u}(\boldsymbol{\mu}), p(\boldsymbol{\mu}))\ (= \boldsymbol{\nu}) \in V^e \times M^e (= Y^e)$ such that

$$\begin{cases} a(\mathbf{u}^e(\boldsymbol{\mu}), w; \boldsymbol{\mu}) + b(p^e(\boldsymbol{\mu}), \mathbf{w}; \boldsymbol{\mu}) = f(\mathbf{w}; \boldsymbol{\mu}), & \forall\, \mathbf{w} \in V^e, \\ b(q, \mathbf{u}^e(\boldsymbol{\mu}); \boldsymbol{\mu}) = 0, & \forall\, q \in M^e \end{cases} \tag{2.49}$$

then evaluate:

$$s^e(\boldsymbol{\mu}) = l(\boldsymbol{\nu}; \boldsymbol{\mu}) = l_u(\mathbf{u}^e(\boldsymbol{\mu}); \boldsymbol{\mu}) + l_p(p^e(\boldsymbol{\mu}); \boldsymbol{\mu}) \tag{2.50}$$

here $s^e(\boldsymbol{\mu})$ is the output of engineering interest, for example a flow rate, a lift or a drag, see for example [Rozo8]. $s^e(\boldsymbol{\mu}) : \mathcal{D} \to \mathbb{R}$ is the input/output relationship, where $l_u(\,\cdot\,; \boldsymbol{\mu}) \in V^e$ and $l_p(\,\cdot\,; \boldsymbol{\mu}) \in M^e$ are linear bounded functionals $\forall\, \boldsymbol{\mu} \in \mathcal{D}$.
We now introduce the bilinear form $\mathscr{A} : Y^e \times Y^e \times \mathcal{D} \to \mathbb{R}$ associated to the Stokes problem, we recall that $\mathbf{v}, \mathbf{z} \in Y^e$:

$$\mathscr{A}(\mathbf{v}, \mathbf{z}; \boldsymbol{\mu}) = \begin{cases} a(\mathbf{u}^e(\boldsymbol{\mu}), w; \boldsymbol{\mu}) + b(p^e(\boldsymbol{\mu}), \mathbf{w}; \boldsymbol{\mu}) \\ b(q, \mathbf{u}^e(\boldsymbol{\mu}); \boldsymbol{\mu}) \end{cases} \qquad \forall\, \mathbf{v} \in Y^e \tag{2.51}$$

and the associated linear functional $\mathscr{F} : Y^e \times \mathcal{D} \to \mathbb{R}$, defined as:

$$\mathscr{F}(\mathbf{v}; \boldsymbol{\mu}) = \begin{cases} f(\mathbf{w}; \boldsymbol{\mu}) \\ 0 \end{cases} \qquad \forall\, \mathbf{v} \in Y^e \tag{2.52}$$

We note that a is symmetric, that is $\forall\, \boldsymbol{\mu} \in \mathcal{D},\ \forall\, \mathbf{u}, \mathbf{v} \in V^e$

$$a(\mathbf{u}, \mathbf{v}; \boldsymbol{\mu}) = a(\mathbf{v}, \mathbf{u}; \boldsymbol{\mu}) \tag{2.53}$$

Recalling the definition given in the section 2.1 of this chapter, in particular the definition 2.6 with regard to *continuity*, def. 2.5 with regard to *coercivity* and def. 2.7 for the *inf-sup* condition for the parametric bilinear forms, we also assume that the bilinear forms are:

1. *Continuous*: there exist $\forall\, \boldsymbol{\mu} \in \mathcal{D}$, $\gamma_a(\boldsymbol{\mu}) > 0$ and $\gamma_b(\boldsymbol{\mu}) > 0$ such that

$$\begin{aligned} a(\mathbf{w}, \mathbf{v}; \boldsymbol{\mu}) &\leqslant \gamma_a(\boldsymbol{\mu}) \|\mathbf{w}\|_{V^e} \|\mathbf{v}\|_{V^e} \quad \forall\, \mathbf{w}, \mathbf{v} \in V^e \\ b(\mathbf{w}, q; \boldsymbol{\mu}) &\leqslant \gamma_b(\boldsymbol{\mu}) \|\mathbf{w}\|_{V^e} \|q\|_{M^e} \quad \forall \in V^e,\ \forall\, q \in M^e \end{aligned} \tag{2.54}$$

2. *Stable*: there exist $\forall\, \boldsymbol{\mu} \in \mathcal{D}$, $\alpha(\boldsymbol{\mu}) \geqslant \alpha_0$ and $\beta(\boldsymbol{\mu}) \geqslant \beta_0$, such that

$$\begin{aligned} 0 < \alpha_0 \leqslant \alpha(\boldsymbol{\mu}) &= \inf_{\mathbf{v} \in V^e} \frac{a(\mathbf{v}, \mathbf{v}; \boldsymbol{\mu})}{\|\|^2_{V^e}} \qquad \forall\, v \in V^e \\ 0 < \beta_0 \leqslant \beta(\boldsymbol{\mu}) &\inf_{q \in M^e} \sup_{\mathbf{w} \in V^e} \frac{b(\mathbf{v}, q; \boldsymbol{\mu})}{\|\mathbf{v}\|_{V^e} \|q\|_{M^e}} \quad \forall\, \mathbf{v} \in V^e,\ \forall\, q \in M^e \end{aligned} \tag{2.55}$$

The conditions above are sufficient to ensure existence and uniqueness [QV97] of the solutions to problems 2.49 and 2.50.
Finally, we make the assumption of affine parameter dependence of the linear and bilinear parametric forms:

$$a(\mathbf{w}, \mathbf{v}; \boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \theta_a^q(\boldsymbol{\mu})\, a^q(w, v) \quad \forall\, \mathbf{w}, \mathbf{v} \in V^e \tag{2.56}$$

$$b(\mathbf{w}, q; \boldsymbol{\mu}) = \sum_{q=1}^{Q_b} \theta_b^q(\boldsymbol{\mu})\, b^q(\mathbf{w}, q) \quad \forall \mathbf{w} \in V^e, \forall\, q \in M^e \tag{2.57}$$

$$f(\mathbf{y}; \boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \theta_f^q(\boldsymbol{\mu})\, f^q(\mathbf{y}) \quad \forall\, \mathbf{y} \in Y^e \tag{2.58}$$

$$l(\mathbf{y}; \boldsymbol{\mu}) = \sum_{q=1}^{Q_l} \theta_l^q(\boldsymbol{\mu})\, l^q(\mathbf{y}) \quad \forall\, \mathbf{y} \in Y^e \tag{2.59}$$

for finite Q_a, Q_b, Q_f, Q_l that depends on the parametric complexity of the particular problem.

2.4.2 Truth approximation

To ensure the *stability* of the approximated problem, the *FE* basis function have to be properly chosen in order to satisfy the discrete inf-sup condition [Bab71].
There are different way to satisfy this condition, [QV97]:

1. formulate the problem 2.49 in a divergence free space,
2. make use of the so called $\mathbb{P}^2$-bubble element,
3. make use of $\mathbb{P}^1$ element for the pressure and $\mathbb{P}^2$ element for the velocity, respectively;

in this thesis we will pursue the *third* option which is quite standard.

2.4.2.1 *Galerkin projection*

We introduce a family of conforming approximation spaces:

$$Y^{\mathcal{N}} \equiv \left(V^{\mathcal{N}_v} \times M^{\mathcal{N}_p}\right) \subset Y^e \equiv (V^e \times M^e) \tag{2.60}$$

of dimension $\dim(Y^{\mathcal{N}}) = \dim\left(V^{\mathcal{N}_v}\right) + \dim\left(M^{\mathcal{N}_p}\right) = \mathcal{N} < \infty$; the subscript v and p denote the velocity and pressure spaces respectively.
We then associate to our space two set of basis functions:

- $\phi_k^v \subset V^{\mathcal{N}_v}$, $1 \leqslant k \leqslant \mathcal{N}_v$, $\mathbb{P}^2$ element
- $\phi_k^p \subset M^{\mathcal{N}_p}$, $1 \leqslant k \leqslant \mathcal{N}_p$, $\mathbb{P}^1$ element

such that, by construction, any member of $Y^{\mathcal{N}}$ can be represented by a unique linear combination of this basis functions.
We remark that, in the case of third option for the stabilization of the problem, it follows that:

$$\mathcal{N} = \mathcal{N}\mathbf{v} + \mathcal{N}_p = 4\mathcal{N}_p$$

We now define the inner product and the norm with which to equip the space $Y^{\mathcal{N}}$ and Y^e, and the *energy norm*.
For any $\mathbf{w}, \mathbf{v} \in Y^e$, we define respectively the *energy inner product* and the *energy norm* as:

$$((\mathbf{w}, \mathbf{v}))_{\mu} = \mathscr{A}(\mathbf{v}, \mathbf{w}; \boldsymbol{\mu}), \tag{2.61}$$

$$|||\mathbf{w}|||_{\mu} = \sqrt{((\mathbf{w}, \mathbf{w}))_{\mu}}, \tag{2.62}$$

moreover, for a given fixed $\bar{\boldsymbol{\mu}} \in \mathcal{D}$, we define the Y^e-*inner product* and the Y^e-*norm* as

$$(\mathbf{w}, \mathbf{v})_{Y^e} = ((\mathbf{w}, \mathbf{v}))_{\bar{\mu}}, \tag{2.63}$$

$$\|\mathbf{w}\|_{Y^e} = \sqrt{(\mathbf{w}, \mathbf{w})_{Y^e}}. \tag{2.64}$$

Problem statement

Now we can state the problem in the truth space taking the Galerkin projection of the problem 2.49-2.50. Given $\boldsymbol{\mu} \in \mathcal{D}$ find the couple $(\mathbf{u}(\boldsymbol{\mu}), p(\boldsymbol{\mu}))\ (= \boldsymbol{\nu}) \in V^{\mathcal{N}_v} \times M^{\mathcal{N}_p} (= Y^{\mathcal{N}})$ such that

$$\begin{cases} a(\mathbf{u}^{\mathcal{N}_v}(\boldsymbol{\mu}), w; \boldsymbol{\mu}) + b(p^{\mathcal{N}_p}(\boldsymbol{\mu}), \mathbf{w}; \boldsymbol{\mu}) = f(\mathbf{w}; \boldsymbol{\mu}), & \forall\, \mathbf{w} \in V^{\mathcal{N}_v}, \\ b(q, \mathbf{u}^{\mathcal{N}_v}(\boldsymbol{\mu}); \boldsymbol{\mu}) = g, & \forall\, q \in M^{\mathcal{N}_p} \end{cases} \tag{2.65}$$

then evaluate:

$$s^{\mathcal{N}}(\boldsymbol{\mu}) = l(\boldsymbol{\nu}; \boldsymbol{\mu}) = l_{\mathbf{u}}(\mathbf{u}^{\mathcal{N}_v}(\boldsymbol{\mu}); \boldsymbol{\mu}) + l_p(p^{\mathcal{N}_p}(\boldsymbol{\mu}); \boldsymbol{\mu}) \tag{2.66}$$

Continuity, coercivity and inf-sup condition

We can define precisely the finite element approximated continuity for the parametric bilinear forms a and b as:

$$\gamma_a(\boldsymbol{\mu}) = \sup_{\mathbf{w} \in V^{\mathcal{N}w}} \sup_{\mathbf{v} \in V^{\mathcal{N}_v}} \frac{a(\mathbf{w}, \mathbf{v}; \boldsymbol{\mu})}{\|\mathbf{u}\|_{V^{\mathcal{N}_v}} \|\mathbf{v}\|_{V^{\mathcal{N}_v}}} \tag{2.67}$$

$$\gamma_b(\boldsymbol{\mu}) = \sup_{\mathbf{w} \in V^{\mathcal{N}w}} \sup_{q \in M^{\mathcal{N}_p}} \frac{b(\mathbf{u}, q; \boldsymbol{\mu})}{\|\mathbf{u}\|_{V^{\mathcal{N}_v}} \|q\|_{M^{\mathcal{N}_p}}} \tag{2.68}$$

from our hypothesis on the bilinear forms (2.54) it follows that $\gamma_a(\boldsymbol{\mu}) < \infty$ and $\gamma_b(\boldsymbol{\mu}) < \infty$, $\forall\, \boldsymbol{\mu} \in \mathcal{D}$. Then, the *coercivity* of a and the *inf-sup* stability condition of b are defined respectively as:

$$\alpha(\boldsymbol{\mu}) = \inf_{\mathbf{v} \in V^{\mathcal{N}_v}} \frac{a(\mathbf{v}, \mathbf{v}; \boldsymbol{\mu})}{\|\mathbf{v}\|^2_{V^{\mathcal{N}_v}}} \tag{2.69}$$

$$\beta(\boldsymbol{\mu}) = \inf_{q \in M^{\mathcal{N}_p}} \sup_{\mathbf{v} \in V^{\mathcal{N}_v}} \frac{b(\mathbf{v}, q; \boldsymbol{\mu})}{\|\mathbf{v}\|_{V^{\mathcal{N}_v}} \|q\|_{M^{\mathcal{N}_p}}} \tag{2.70}$$

it also follow from our hypothesis (2.55) that $\alpha^{\mathcal{N}_v}(\boldsymbol{\mu}) > \alpha_0 > 0$; thanks to the proper choice of the *FE* basis function (discussed in section 2.4.2) it immediately follows that $\beta^{\mathcal{N}}(\boldsymbol{\mu}) > \beta_0 > 0$.

2.4.2.2 *Well-posedness inheriting*

The Galerkin approximation on $Y^{\mathcal{N}}$ must satisfy the same conditions that the exact formulation satisfies over Y^e.
In particular, thanks to our hypothesis on the discretization it follows that our Galerkin formulation inherit the good properties of well-posedness of the exact formulation:

1. the dual norm of f over $V^{\mathcal{N}_v}$ and is bounded by the dual norm of f and over V^e
2. a is coercive over $V^{\mathcal{N}_v}$ with

$$\alpha^{\mathcal{N}_v}(\boldsymbol{\mu}) \geqslant \alpha^e(\boldsymbol{\mu}) \quad \forall\, \boldsymbol{\mu} \in \mathcal{D} \tag{2.71}$$

3. b is inf-sup stable over $Y^{\mathcal{N}}$, with

$$\beta^{\mathcal{N}}(\boldsymbol{\mu}) \geqslant \beta^{e}(\boldsymbol{\mu}) \quad \forall\, \boldsymbol{\mu} \in \mathcal{D} \tag{2.72}$$

4. a is continuos over $V^{\mathcal{N}_v}$ with

$$\gamma_a^{\mathcal{N}_v}(\boldsymbol{\mu}) \leqslant \gamma_a^{e}(\boldsymbol{\mu}) \tag{2.73}$$

5. b is continuos over $Y^{\mathcal{N}}$ with

$$\gamma_b^{\mathcal{N}}(\boldsymbol{\mu}) \leqslant \gamma_b^{e}(\boldsymbol{\mu}) \tag{2.74}$$

6. the affine expansion for f, l, a and b (2.56-2.59) are still valid.

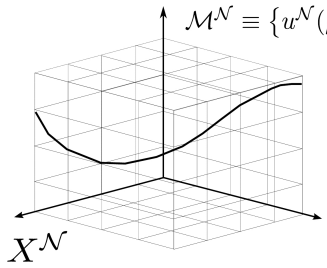
thus the *non-coercive Stokes problem* is endowed with all the properties to ensure a unique solution by *Galerkin method*.

3 RB METHOD FOR PARAMETRIZED ELLIPTIC PDES

In this chapter we will introduce the relevant steps for the generation of the rapidly convergent global RB approximation spaces for the approximation of the solution of parametrized *coercive* and *noncoercive* elliptic partial differential equations with affine parameter dependence will be explained.
Subsequently it will be possible to introduce the reduced basis approximation methodology, the sampling strategies and the construction of the reduced spaces.
Then an *a-posteriori error bound* necessary to achieve an efficient *RB* sampling it will be explained.
The *RB* methodology in the *coercive* and *noncoercive* case differs in the basis assembling procedure and in the a-posteriori error evaluation.
The *lack of coercivity* will lead to the introduction of an *additional space* -the supremizer- in the reduced base assembling.

3.1 RB IDEA

As described in the chapter 1, the Reduced Basis *RB* approach derives from the two *opportunities*: described in 1.2.1 and 1.2.2.
In particular regarding the *Opportunity I*, although $u^{\mathcal{N}}(\boldsymbol{\mu})$ is a member of the space $X^{\mathcal{N}}$ of typically very high dimension $\mathcal{N}$, in fact $u^{\mathcal{N}}(\boldsymbol{\mu})$ resides on a *low-dimensional parametrically induced manifold* $\mathcal{M} \equiv \{u^{\mathcal{N}}(\boldsymbol{\mu}) \mid \boldsymbol{\mu} \in \mathcal{D}\}$.



(a)

Figure 5: Parametrically induced manifold on $X^{\mathcal{N}}$

In figure 5 a graphical heuristic idea of the finite dimensional (*truth*) manifold $X^{\mathcal{N}}$ with the parametrically induced manifold $\mathcal{M}^{\mathcal{N}}$ (filament) is shown. The same idea in the exact infinite dimensional space is depicted in Figure 2 of Section 1.2.1.
It is thus wasteful to express the solution $u(\boldsymbol{\mu})$ as an arbitrary member of the unnecessarily rich space $X^{\mathcal{N}}$; rather, presuming that $\mathcal{M}$ is sufficiently smooth, we should represent $u(\boldsymbol{\mu})$ in terms of elements of an *ad-hoc* manifold much more lower dimensional, see [PR09, RHP08].

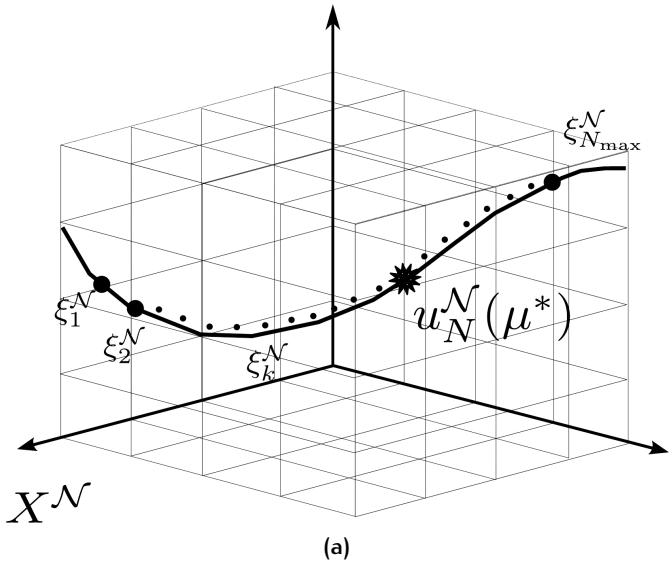


Figure 6: Approximation of $u^{\mathcal{N}}_N(\boldsymbol{\mu}^*)$

The *RB* recipe is very simple (see figure 6 for a graphical interpretation). The basic idea is to *efficiently* chose and compute N solutions or *"snapshots"* $\xi^{\mathcal{N}}_1, \xi^{\mathcal{N}}_2, \ldots, \xi^{\mathcal{N}}_N \in X^{\mathcal{N}}$ and then, for any arbitrary new $\boldsymbol{\mu}^* \in \mathcal{D}$, compute the solution $u^{\mathcal{N}}_N(\boldsymbol{\mu}^*)$ associated to this parameter thanks to an *appropriate combination* of the previously computed snapshots $\xi^{\mathcal{N}}_k$, $k = 1, \ldots, N$.
Note that "$u^{\mathcal{N}}_N(\boldsymbol{\mu})$" is not redundant; it means that this is the solution in the truth space $X^{\mathcal{N}}$ computed along the reduced manifold $\mathcal{M}^{\mathcal{N}}$, selecting N snapshots.
In the most part this thesis, if not specified, when dealing with *RB* solution we will always simply write $u_N(\boldsymbol{\mu})$ meaning the reduced solution in the truth space.

Now also the *Opportunity II* (section 1.2.1, chapter 1) can be understood; starting from the *RB* idea it is evident that are needed *at least* N solutions of the problem on the $\mathcal{N}$-dimensional *truth space*.
The *RB* approach is thus clearly *ill-suited* to the *single-query* or *few-query* situation; however, in the *real-time* and *many-query context* this *Offline* investment is readily acceptable in exchange for future asymptotic or Online computational burden reduction.

3.2 COERCIVE CASE

In this section the *RB* problem formulation in the *coercive* case is discussed.
We begin introducing the spaces and basis that allow us to build the reduced basis problem, subsequently the *creation* of the *RB* system, the *Of-*

fline/Online procedure and the *a-posteriori* error bound will be introduced.

3.2.1 RB spaces and basis

There are different possible choices for the selection of the reduced basis spaces (Hermite, Lagrange, ecc...) that will lead ultimately to different reduced order model, see [RHP08, PR09, Por85, IR01].
In the following the *Lagrange hierarchical spaces* used in this work will be discussed.

3.2.2 Lagrange hierarchical spaces

We introduce a set of linearly independent functions:

$$\xi^n \in X, \quad 1 \leqslant n \leqslant N_{max} \tag{3.1}$$

where N_{max} is the maximum dimension of the *RB* space, in terms of which we define the *RB* approximation spaces:

$$X_N = \text{span}\{\xi^n,\ 1 \leqslant n \leqslant N\} \quad 1 \leqslant N \leqslant N_{max} \tag{3.2}$$

where we assume, in order to build a "reduced basis" that the ξ^n are somehow related to the manifold $\mathcal{M}$.
By construction we obtain

$$X_N \subset X, \quad \dim(X_N) = N, \quad 1 \leqslant N \leqslant N_{max} \tag{3.3}$$

moreover, as the same property holds recursively for any nested subset of X_N, we can say that the space X is *hierarchical*.

Definition 3.1. *Given a space* X, *given* N_{max} *subsets of this space* $X_n \subset X$, $1 \leqslant n \leqslant N_{max}$, *we say that* X *is a* hierarchical (or nested) *space if:*

$$X_1 \subset X_2 \subset \ldots \subset X_{N_{max}-1} \subset X_{N_{max}} \tag{3.4}$$

The hierarchical property 3.4, as we shall see, is important in ensuring (memory) efficiency for the resulting reduced basis approximation.
To introduce the Lagrange (hierarchical) *RB* recipe, we first define a master set of parameter points $\mu^n \in \mathcal{D}$, $1 \leqslant n \leqslant N_{max}$, we then define, for given $N \in \{1, \ldots, N_{max}\}$, the Lagrange parameter samples

$$S_N = \left\{\mu^1, \ldots, \mu^N\right\}, \tag{3.5}$$

that we choose *nested* in order to build a hierarchical space, that is:

$$S_1 = \left\{\mu^1\right\} \subset S_2 = \left\{\mu^1, \mu^2\right\} \subset \ldots \subset S_{N_{max}}. \tag{3.6}$$

The associated *Lagrange RB* spaces are defined as:

$$W_N^{\mathcal{N}} = \text{span}\left\{u^{\mathcal{N}}(\boldsymbol{\mu}^n)\right\}, \quad 1 \leqslant n \leqslant N. \tag{3.7}$$

We observe that, by construction, these Lagrange spaces $W_N^{\mathcal{N}} = X_N^{\mathcal{N}}$ are hierarchical; in fact the samples S_N are nested thanks to the choice 3.6, then accordingly:

$$W_1^{\mathcal{N}} = \text{span}\left\{u^{\mathcal{N}}(\boldsymbol{\mu}^1)\right\} \subset W_2^{\mathcal{N}} = \text{span}\left\{u^{\mathcal{N}}(\boldsymbol{\mu}^2), u^{\mathcal{N}}(\boldsymbol{\mu}^1)\right\} \subset \ldots \subset W_{N_{max}}^{\mathcal{N}} \tag{3.8}$$

The $u^{\mathcal{N}}(\boldsymbol{\mu}^n)$, $1 \leqslant n \leqslant N_{max}$ are the so-called *"snaphots"* related to the low dimensional manifold $\mathcal{M}^{\mathcal{N}}$. As already mentioned in section 3.1, we would expect to well approximate any member of the manifold thanks to a good combination of the available snapshots.
In theory, in order to build the *RB* approximation spaces, it would be necessary to choose a set of parameter sample (see Section 3.5 that induces a set of linearly independent snapshot; the *greedy* sampling, that will be introduced in Section 3.4, induces linear dependent functions as N increases. In fact, if the snapshot chosen W_N contains much of the $\mathcal{D}$ induced manifold $\mathcal{M}$, then it will be clear that the new $\boldsymbol{\mu}^{N+1} \in \mathcal{D}$ will perforce be a combination of this functions.
We therefore pursue a Gram-Schmidt orthogonalization in the $(\cdot, \cdot)_X$ inner product to recover an orthonormal well-conditioned set of basis functions in order to guarantee a good algebraic stability without an ill-conditioning [PR09].

3.2.3 Orthogonal *RB* basis

To achieve the orthogonalization, we apply the already mentioned Gram-Schmidt standard orthogonalization [Meyoo]. Given the basis functions ξ^n, $1 \leqslant n \leqslant N_{max}$ (3.1), that in the Lagrange space choice are the $u(\boldsymbol{\mu}_n)$, $1 \leqslant n \leqslant N_{max}$ (3.7), we obtain the set of basis function ζ^n, $1 \leqslant n \leqslant N_{max}$ as:

$$\begin{aligned}
&\zeta^1 = \xi^1 / \|\xi^1\|_X; \\
&\texttt{for } n = 2 : N_{max} \\
&\quad z^n = \xi^n - \sum_{m=1}^{n-1} , (\xi^n, \zeta^m)_X \ \zeta^m; \\
&\quad \zeta^n = z^n / \|z^n\|_X; \\
&\texttt{end}
\end{aligned} \tag{3.9}$$

As a result of this process we obtain the orthogonality condition:

$$(\zeta^m, \zeta^n)_X = \delta_{mn} \quad 1 \leqslant m, n \leqslant N_{max} \tag{3.10}$$

where δ_{mn} is the Kronecker-delta symbol.
Finally we can express our reduced basis spaces X_N as:

$$X_N = \text{span}\{\zeta^n, \ 1 \leqslant n \leqslant N\} \quad 1 \leqslant N \leqslant N_{max} \tag{3.11}$$

Now any function $w_N \in X_N$ can be expressed as a linear combination of the reduced base X_N as:

$$w_N = \sum_{n=1}^{N} w_{N_n}\, \zeta^n \quad 1 \leqslant N \leqslant N_{max} \tag{3.12}$$

for a unique combination of (*RB*) coefficients $w_{N_n} \in \mathbb{R}$, $1 \leqslant n \leqslant N_{max}$.

3.2.4 Algebraic representation of *RB* basis

We now reconsider the orthogonalization process in order to introduce some concepts that will be necessary to build our *RB* problem starting from the *FE* original frame.
If we express our snapshots ξ^n in terms of *FE* functions ϕ_i, $1 \leqslant i \leqslant \mathcal{N}$:

$$\xi^n = \sum_{i=1}^{\mathcal{N}} \xi_i^n\, \phi_i, \quad 1 \leqslant n \leqslant N_{max}, \tag{3.13}$$

similarly we may express our *RB* orthogonalized functions ζ^n as

$$\zeta^n = \sum_{i=1}^{\mathcal{N}} \zeta_i^n\, \phi_i, \quad 1 \leqslant n \leqslant N_{max}. \tag{3.14}$$

Now, in the two cases above, we sort the *FE* coefficients in an array

$$\underline{\xi}^n \equiv \left\{ \begin{array}{cccc} \xi_1^n & \xi_2^n & \dots & \xi_{\mathcal{N}}^n \end{array} \right\}^T \quad 1 \leqslant n \leqslant, N_{max} \tag{3.15}$$

$$\underline{\zeta}^n \equiv \left\{ \begin{array}{cccc} \zeta_1^n & \zeta_2^n & \dots & \zeta_{\mathcal{N}}^n \end{array} \right\}^T \quad 1 \leqslant n \leqslant .N_{max} \tag{3.16}$$

We then introduce the algebraic representation $\underline{\underline{\mathbb{X}}}^{\mathcal{N}} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ of the inner product $(\cdot, \cdot)_X$:

$$\underline{\underline{\mathbb{X}}}^{\mathcal{N}}_{ij} = \left(\phi^i, \phi^j\right)_{X^{\mathcal{N}}} \quad 1 \leqslant i, j \leqslant \mathcal{N}. \tag{3.17}$$

The orthogonalization process can be now formulated as:

$$\begin{aligned} &\underline{\zeta}^1 = \underline{\xi}^1 / \sqrt{\underline{\xi}^{1\,T}\, \underline{\underline{\mathbb{X}}}\, \underline{\xi}^1}; \\ &\texttt{for } n = 2 : N_{max} \\ &\quad \underline{z}^n = \underline{\xi}^n - \sum_{m=1}^{n-1}, \left(\underline{\xi}^n\, \underline{\underline{\mathbb{X}}}\, \underline{\zeta}^m\right)\, \underline{\zeta}^m; \\ &\quad \underline{\zeta}^n = \underline{z}^n / \sqrt{\underline{z}^{n\,T} \underline{\underline{\mathbb{X}}}\, \underline{z}^n}; \\ &\texttt{end} \end{aligned} \tag{3.18}$$

Finally it is useful to introduce the *"basis"* matrices $\underline{\underline{Z}} \in \mathbb{R}^{\mathcal{N}\times N}$, $1 \leqslant N \leqslant N_{max}$:

$$Z_{N_{j\,n}} = \zeta_j^n \quad 1 \leqslant j \leqslant \mathcal{N},\ 1 \leqslant n \leqslant N. \tag{3.19}$$

This matrices is built in such a way that the n^{th}-column of the matrix is formed by the vector of *FE* coefficients $\zeta^n_{N_j}$, $1 \leqslant j \leqslant \mathcal{N}$ associated to the n^{th} *RB* function.

3.2.5 Galerkin projection

The projection strategy used in order to obtain the *RB* approximation is given by a Galerkin projection, which is arguably the best approach. We remark that the *RB* weak formulation has formally the same appearance as the *"exact"* weak formulation (see equations 2.29-2.30, Section 2.3.1); in this case we properly replace the *FE* truth functional space with the *RB* approximation space; in the next Section we will show how to obtain the latter from the former by means of an algebraic procedure.
The problem states: given $\boldsymbol{\mu} \in \mathcal{D}$, find $u_N \left(\equiv u_N^{\mathcal{N}}\right) \in X_N \left(X_N^{\mathcal{N}}\right)$ such that

$$a(u_N(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \forall v \in X_N \tag{3.20}$$

we evaluate

$$s_N(\boldsymbol{\mu}) = f(u_N(\boldsymbol{\mu})) \tag{3.21}$$

From coercivity and continuity hypothesis on a and f, our conforming reduced basis $X_N^{\mathcal{N}} \subset X^{\mathcal{N}}$ and from our assumption of linear independence of snapshots, the problem 3.20-3.21 admits an unique solution, [QV97, Qua09].
Thanks to the Galerkin projection, the optimality results subsequently discussed holds, see [RHP08, PR09].

Proposition 3.1. *For any* $\boldsymbol{\mu} \in \mathcal{D}$ *and* $u_N(\boldsymbol{\mu})$ *and* $s_N(\boldsymbol{\mu})$ *satisfying 3.20-3.21:*

$$|||u^{\mathcal{N}} - u_N^{\mathcal{N}}|||_{\boldsymbol{\mu}} = \inf_{w_N \in X_N^{\mathcal{N}}} |||u^{\mathcal{N}} - w_N|||_{\boldsymbol{\mu}} \tag{3.22}$$

$$\|u^{\mathcal{N}} - u_N^{\mathcal{N}}\|_X \leqslant \sqrt{\frac{\gamma^e(\boldsymbol{\mu})}{\alpha^e(\boldsymbol{\mu})}} \inf_{w_N \in X_N^{\mathcal{N}}} \|u^{\mathcal{N}} - w_N\|_X \tag{3.23}$$

as regard the output optimality results, in the compliant case *we obtain:*

$$\begin{aligned} s_N - s_N^{\mathcal{N}} &= |||u^{\mathcal{N}} - u_N^{\mathcal{N}}|||^2_{\boldsymbol{\mu}} \\ &= \inf_{w_N \in X_N^{\mathcal{N}}} |||u^{\mathcal{N}} - w_N|||^2_{\boldsymbol{\mu}} \end{aligned} \tag{3.24}$$

and furthermore

$$0 \leqslant s_N - s_N^{\mathcal{N}} \leqslant \gamma^e(\boldsymbol{\mu}) \inf_{w_N \in X_N^{\mathcal{N}}} \|u^{\mathcal{N}} - w_N\|^2_X, \tag{3.25}$$

where $\gamma^e(\boldsymbol{\mu})$ and $\alpha^e(\boldsymbol{\mu})$ are, respectively, the continuity and coercivity constants (defined in section 2.6 and 2.5) in the exact space. It will be shown that these optimality results will be used even after, replacing the

exact constants with those evaluated with the reduced space approximation.
It is also necessary to remark from the equation 3.24 that, in the *compliant case*, the error on the output is the *square* of the error of the field variable: we have the so-called *"square effect"*, this is crucial for the input/output accuracy and efficiency of the method.
Last but not least, $s_N(\mu)$ is a *lower bound* for $s^{\mathcal{N}}$, in fact: (i) $s_N(\mu) = a(u_N(\mu), u_N(\mu); \mu)$ is a positive quantity, and (ii) the error in the output is the *square* of the error of the field variable, see [RHP08].

3.2.6 Offline-Online procedure

In this section the algebraic formulation for the coercive problem will be explained. The crucial point that will be treated is the Online/Offline splitting procedure; this procedure will be equipped with an operation count to highlight the potential computational saving offered by the *RB* method.

3.2.6.1 *Algebraic formulation*

In order to apply the standard variational procedure to obtain the algebraic formulation of the problem, we first expand $u_N(\mu)$:

$$u_N(\mu) = \sum_{j=1}^{N} u_N^j(\mu) \tag{3.26}$$

now, inserting the expansion 3.26 in the problem 3.20 and choosing $v = \zeta^i, \quad 1 \leqslant i \leqslant N$ as our *test function*, we obtain the set of linear algebraic equations

$$\sum_{j=1}^{N} a(\zeta^j, \zeta^i; \mu)\, u_N^j(\mu) = f(\zeta^i; \mu), \quad 1 \leqslant i \leqslant N \tag{3.27}$$

for the reduced basis coefficients $u_N^j(\mu)$ $1 \leqslant i \leqslant N$.
The output can then be expressed as

$$s_N(\mu) = \sum_{j=1}^{N} u_N^j(\mu)\, f(\zeta^j; \mu) \tag{3.28}$$

We now express these operations in matrix form; we first introduce the array of *RB* basis coefficients $u_N(\mu)$ as

$$\underline{u}_N \equiv [u_{N_1} u_{N_2} \dots u_{N_N}] \tag{3.29}$$

It follows from 3.27 that $\underline{u}_N \in \mathbb{R}^N$ satisfies

$$\underline{\underline{A}}_N(\mu)\, \underline{u}_N(\mu) = \underline{F}_N(\mu) \tag{3.30}$$

where $\underline{\underline{A}}_N(\mu) \in \mathbb{R}^{N\times N}$ is the "stiffness matrix", and $\underline{F}_N(\mu) \in \mathbb{R}^N$ is the "load/source" array. This quantity are in particular given by:

$$\underline{\underline{A}}_{N_{i,j}}(\mu) = a(\zeta^j, \zeta^i; \mu) \quad 1 \leqslant i, j \leqslant N \tag{3.31}$$

and

$$\underline{F}_{N_i}(\mu) = f(\zeta^i; \mu) \quad 1 \leqslant i \leqslant N. \tag{3.32}$$

Finally, the output can now be expressed as

$$\underline{s}_N(\mu) = \underline{F}_N^T \, \underline{u}_N \tag{3.33}$$

It follows from our assumption of linear independence of the snapshots that the stiffness matrix is symmetric and positive definite.

3.2.6.2 *Offline-Online*

It is now possible, starting from the algebraic problem, to introduce the *Offline/Online* procedure.
The reduced basis system 3.30 is clearly of small size, in fact it is an $N \times N$ linear system that requires $\mathcal{O}(N^3)$ operation to solve it, plus $\mathcal{O}(N)$ operation to obtain the output from the equation 3.33.

By appealing to our previous assumption of affine parameter dependence discussed in Section 2.2, from equations 2.20 and 2.27, the *stiffness matrix* and *load/source vector* can be expressed, respectively, as

$$a(\zeta^m, \zeta^n; \mu) = \sum_{q=1}^{Q_a} \theta_a^q(\mu) \; a(\zeta^m, \zeta^n) \quad 1 \leqslant m, n \leqslant N \tag{3.34}$$

and

$$f(\zeta^n; \mu) = \sum_{q=1}^{Q_f} \theta_f^q(\mu) \; f(\zeta^n) \quad 1 \leqslant n \leqslant N. \tag{3.35}$$

The Offline-Online decomposition is now:

Offline

In the offline part we form:

1. the parameter independent matrices $\underline{\underline{\mathbb{A}}}_N^q \in \mathbb{R}^{N\times N}$

$$\underline{\underline{\mathbb{A}}}_{N_{m,n}}^q = a^q(\zeta^m, \zeta^n), \quad 1 \leqslant m, n \leqslant N, \;\; 1 \leqslant q \leqslant Q_a \tag{3.36}$$

2. the parameter independent vectors $\underline{\mathbb{F}}_N^q \in \mathbb{R}^N$

$$\underline{\mathbb{F}}_{N_n}^q = f^q(\zeta^n), \quad 1 \leqslant n \leqslant N, \;\; 1 \leqslant q \leqslant Q_f \tag{3.37}$$

This operations $\mathcal{N}$-dependent and hence very expensive are computed *once*.

Online

In the online stage we assemble, for any new $\boldsymbol{\mu} \in \mathcal{D}$:

1. the *RB* reduced stiffness matrix $\underline{\underline{A}}_N(\boldsymbol{\mu})$

$$\underline{\underline{A}}_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \theta_a^q(\boldsymbol{\mu})\, \underline{\underline{\mathbb{A}}}_N^q \tag{3.38}$$

2. the *RB* reduced load/source vector $\underline{F}_N(\boldsymbol{\mu})$

$$\underline{F}_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \theta_f^q(\boldsymbol{\mu})\, \underline{\mathbb{F}}_N^q \tag{3.39}$$

the operation count is actually $\mathcal{N}$-independent and hence very inexpensive.

Link between FE and RB

Before a detailed discussion over the operation count, it is necessary to provide the link between the *FE* and the *RB* stiffness matrix and load/-source vector; it is worth to remark that this operation will be completed once in the offline stage.
In particular it can be showed that the former are linked to the latter via the *"basis"* matrices $\underline{\underline{\mathbb{Z}}} \in \mathbb{R}^{\mathcal{N}\times N}$, $1 \leqslant N \leqslant N_{max}$.
The stiffness matrix 3.31 that can be written as:

$$a(\zeta^m, \zeta^n; \boldsymbol{\mu}) = \sum_{j=1}^{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \zeta_j^m\, a(\phi^i, \phi^j; \boldsymbol{\mu})\, \zeta_i^n \quad 1 \leqslant m, n \leqslant N_{max} \tag{3.40}$$

thanks to the definition of the *basis matrix* $\underline{\underline{\mathbb{Z}}}$ (3.19) and the *FE* development of the reduced basis functions ζ^n (3.14) we may rewrite the stiffness matrix 3.40 as

$$\underline{\underline{A}}_N(\boldsymbol{\mu}) = \underline{\underline{\mathbb{Z}}}^T \underline{\underline{A}}(\boldsymbol{\mu})\, \underline{\underline{\mathbb{Z}}}. \tag{3.41}$$

In the same way the reduced parametric independent stiffness matrices $\underline{\underline{\mathbb{A}}}_N^q$ 3.38 are linked to the *FE* matrices:

$$\underline{\underline{\mathbb{A}}}_N^q = \underline{\underline{\mathbb{Z}}}^T \underline{\underline{\mathbb{A}}}^q\, \underline{\underline{\mathbb{Z}}}. \tag{3.42}$$

The load/source vectors 3.33 admit a similar treatment

$$\underline{F}_N(\boldsymbol{\mu}) = \underline{\underline{\mathbb{Z}}}^T \underline{F}(\boldsymbol{\mu}). \tag{3.43}$$

finally the parameter independent load/source vector 3.39 can be written as

$$\underline{\mathbb{F}}_N^q = \underline{\underline{\mathbb{Z}}}^T \underline{\mathbb{F}}^q(\boldsymbol{\mu}) \quad 1 \leqslant q \leqslant Q_f. \tag{3.44}$$

3.2.7 Operation count and storage

Thanks to the Offline/Online splitting procedure, we have achieved an Online $\mathcal{N}$-independent stage, hence very inexpensive.
It is necessary to focus on the Offline and Online complexity to quantify the computational reduction provided by the *RB* method.
We make use of Table 1 to summarize the computational burden requested to perform: (i) Offline, the *RB* basis assembling, (ii) Online, a single input/output evaluation.

PART	ITEM	BURDEN	EQUATION
Offline	$\underline{\underline{\mathbb{Z}}}_N$	$N\,\mathcal{O}(\mathcal{N}^3)$	3.19
	$\underline{\underline{\mathbb{A}}}_N^q$	$Q_a\,N\,\underline{\underline{A}}$-matvec $+\,Q_a\,N^2\,X^{\mathcal{N}}$-inprod	3.42
	$\underline{\mathbb{F}}_N$	$Q_f\,N\,X^{\mathcal{N}}$-inprod	3.44
Online	$\underline{\underline{A}}_N$	$Q_a\,N^2$	3.38
	$\underline{F}_N$	$Q_f\,N$	3.39
	u_N	$\mathcal{O}(N^3)$	3.27
	s_N	N^2	3.33

Table 1: Offline/Online: coercive case

In this table we have denoted with "A-matvec" and "$X^{\mathcal{N}}$-inprod", the matrix-vector multiplication and the inner product between two vectors $\in X^{\mathcal{N}}$, respectively.
As regard the storage we need only to store $\underline{\underline{\mathbb{A}}}_{N_{max}}^q$ $1 \leqslant q \leqslant Q_a$ and $\underline{\mathbb{F}}_{N_{max}}^q$ $1 \leqslant q \leqslant Q_f$, then extract only the sub-matrices and sub-vectors of desired N thanks to the hierarchical base property, as explained in Section 3.4.

3.3 NON-COERCIVE CASE: STOKES PROBLEM

We introduce the projection strategy chosen in order to obtain our RB approximation for the non-coercive Stokes problem, as a particular case (of interest) as non-coercive problem, [QV97, Qua09].
The *RB* base assembling is more involved than in the coercive case.
The consequences will be the creation of an *additional* base beside the classical Lagrange base described in Section 3.2.1 in order to provide *stability* to the *RB* approximation. Even for this particular class of equations, Galerkin projection is the best approach.
We recall from Section 2.4 that the Stokes problem describes the velocity and the pressure field in a viscous flow. We look for the velocity field into a $\left(H_0^1(\Omega)\right)^3$ $(= V^e)$ space and the pressure into a $L_2(\Omega)$ $(= M^e(\Omega))$ space,

[Qua09]. We briefly recall the weak formulation for the non-coercive Stokes problem: find $(\mathbf{u}(\boldsymbol{\mu}), p(\boldsymbol{\mu}))\,(=\boldsymbol{\nu}) \in V^e \times M^e (= Y^e)$ such that

$$\begin{cases} a(\mathbf{u}^e(\boldsymbol{\mu}), w; \boldsymbol{\mu}) + b(p^e(\boldsymbol{\mu}), \mathbf{w}; \boldsymbol{\mu}) = f(\mathbf{w}; \boldsymbol{\mu}), & \forall\, \mathbf{w} \in V^e, \\ b(q, \mathbf{u}^e(\boldsymbol{\mu}); \boldsymbol{\mu}) = 0, & \forall\, q \in M^e \end{cases}$$

3.3.1 RB spaces and basis

We select some samples $S_N = \{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N\}$, where $\boldsymbol{\mu}^n \in \mathcal{D}$, $1 \leqslant n \leqslant N$ and we solve N times the problem 2.65 using the Galerkin-*FE* method. Concerning Stokes problem, the *reduced basis pressure space* is built as

$$M_N = \text{span}\{\psi_n,\ 1 \leqslant n \leqslant N\} \tag{3.45}$$

where $\psi_n = p^{\mathcal{N}_p}(\boldsymbol{\mu}^n)$.
Otherwise as regard the velocity, it is necessary to introduce a particular recipe to build the basis space. We begin recalling the definition of the *"supremizer operator"* $T_{\boldsymbol{\mu}}(\,\cdot\,)$ given in 2.9 (page 11).
Now we build the *reduced basis velocity space* enriching the velocity space with the supremizer solutions, this space will be denoted with the superscript $^{\boldsymbol{\mu}}$ to remind this key-point feature:

$$V_N^{\boldsymbol{\mu}} = \text{span}\{\xi_n,\ 1 \leqslant n \leqslant N;\ \ T^{\boldsymbol{\mu}}\psi_n,\ 1 \leqslant n \leqslant N\} \tag{3.46}$$

where $\xi_n = \mathbf{u}^{\mathcal{N}_v}(\boldsymbol{\mu}^n)$.
This choice is necessary to satisfy an equivalent inf-sup stability condition; without appending this "additional" space the *RB* problem will lead us to a solution which may be not stable.
The supremizer in the Stokes case can be computed as solution, for any ψ_n, $1 \leqslant n \leqslant N$, of the following *Poisson-like* problem, [RHP08, PR09]

$$(T^{\boldsymbol{\mu}}\psi_n, \mathbf{w})_V = b(q, \mathbf{w}; \boldsymbol{\mu}) \quad \forall\, \mathbf{w} \in V^{\mathcal{N}_v} \tag{3.47}$$

where the bilinear form b has been introduced in the exact formulation (2.49).
Finally we define the global space $Y_N^{\boldsymbol{\mu}}$ as the cartesian product of the above defined spaces $V_N^{\boldsymbol{\mu}}$ and M_N

$$Y_N^{\boldsymbol{\mu}} = V_N^{\boldsymbol{\mu}} \times M_N. \tag{3.48}$$

As already done with the coercive spaces 3.2.1, we now pursue a *Gram-Schmidt* orthogonalization in the $(\cdot,\cdot)_Y$ inner product to recover a orthonormal well-conditioned set of basis functions.
Remark: In the Stokes problem, $\mathcal{N}_{\mathbf{v}}$ denotes the number of *DOFs* of the velocity, whereas $\mathcal{N}_p$ denotes the DOFs of the pressure. We remark that, since we are dealing with a 3D problem, the velocity unknowns in each node is a vector with three components $\mathbf{v} = \{u, v, w\}$. Hence, the velocity *DOFs* are $\mathcal{N}_u + \mathcal{N}_v + \mathcal{N}_w = 3\mathcal{N}_u\,(= \mathcal{N}_{\mathbf{v}})$.

3.3.2 Orthogonal *RB* basis and space

To achieve the orthogonalization, we apply the already mentioned Gram-Schmidt orthogonalization [Meyoo]. In the Stokes case the orthogonalization procedure is similar to that relatives to the coercive case, [Meyoo].
The main difference lies in the definition of the $(\cdot, \cdot)_Y$ inner product. We have chosen to *measure* and therefore to *orthogonalize* the functions accordingly to the space to which this function belongs to; it follows that the *velocity* and the *supremizer* functions will be orthogonalized in an H^1 *metric*, whereas the *pressure* will be orthogonalized in an L^2 *metric*. In our work we have chosen to orthogonalize the *velocity* space *separately* from the *supremizer* space. There are more options possible, see [RVo6] for different orthogonalization choice.
Let we introduce our velocity and pressure basis functions respectively as $\mathbf{w}_n^\mu \in V_N^\mu$, $1 \leqslant n \leqslant N_{max}$ and $q_n \in M_N$, $1 \leqslant n \leqslant N_{max}$, we denote with $\mathbf{v}_n = (\mathbf{w}_n^\mu, q_n)$, $1 \leqslant n \leqslant N_{max}$ the generic basis function $\in V_N^\mu$. Then we denote the *orthogonal* velocity and basis functions respectively as $\zeta_n^\mu \in V_N^\mu$, $1 \leqslant n \leqslant N_{max}$ and $\psi_n \in M_N$, $1 \leqslant n \leqslant N_{max}$, finally we denote an *orthogonal* basis function as $\gamma_n \left(= (\zeta_n^\mu, \psi_n)\right) \in Y_N^\mu$, $1 \leqslant n \leqslant N_{max}$.
We now obtain our set of orthogonormal basis functions γ_n in the $(\cdot, \cdot)_Y$ inner product as

$$
\begin{aligned}
&\gamma_1 = \mathbf{v}_1 / \|\mathbf{v}_1\|_Y; \\
&\texttt{for } n = 2 : N_{max} \\
&\quad \gamma_n = \mathbf{v}_n - \sum_{m=1}^{n-1} (\mathbf{v}_n, \gamma_m)_Y \ \gamma_m; \\
&\quad \gamma_n = \gamma_n / \|\gamma_n\|_Y; \\
&\texttt{end}
\end{aligned}
\tag{3.49}
$$

as final result we obtain the orthogonality condition

$$
(\gamma_m, \gamma_n) = \delta_{mn} \quad 1 \leqslant m, n \leqslant N_{max} \tag{3.50}
$$

Our orthogonal space can be expressed, for any $1 \leqslant N \leqslant N_{max}$ as:

$$
\begin{aligned}
Y_N^\mu &= \text{span}\{\gamma_n, \ 1 \leqslant n \leqslant N\} = \\
&= \text{span}\{\zeta_n^\mu, \ 1 \leqslant n \leqslant N; \ \psi_n, \ 1 \leqslant n \leqslant N\}
\end{aligned}
\tag{3.51}
$$

3.3.3 Algebraic representation of *RB* basis

It is useful to introduce the algebraic representation of *RB* basis to better understand the link between the *FE* and *RB* matrices and vectors, that will be discussed in section 3.3.5.1, in order to exploit the Offline/Online procedure.
We need to express our basis functions in terms of the *FE* basis functions

ϕ_i , $1 \leqslant i \leqslant \mathcal{N}_{\mathbf{v}} + \mathcal{N}_p$.
We recall from Section 2.4.2 that the pressure and the velocity "lives" on different *FE* basis ($\mathbb{P}^2$-$\mathbb{P}^1$, Taylor-Hood *FE* for the velocity and pressure, respectively); moreover we recall that the velocity base V_N^{μ} is enriched by the solution $T^{\mu} q \equiv \mathbf{w}_{sup}$ of the supremizer problem.
The velocity as well as the supremizer solution, share the same basis functions; therefore to express the *RB* basis in terms of *FE* basis we need to enlarge the array of coefficients to place the $\mathcal{N}_{\mathbf{v}}$ added *FE* coefficients and duplicate the *FE* velocity basis functions.
To wit, we sort the basis functions as follows:

$$\underline{\phi} = \left\{ \begin{array}{ccc} \phi_i^v & \phi_i^v & \phi_i^p \end{array} \right\}^T \tag{3.52}$$

here ϕ_i^v, $1 \leqslant i \leqslant \mathcal{N}_v$ are the *FE* velocity basis functions that we have *duplicated* in the array since we must consider either the velocity and the supremizer basis functions, whereas ϕ_i^p, , $1 \leqslant i \leqslant \mathcal{N}_p$ are the *FE* pressure basis functions.
We remark that, since we are dealing with a 3D Stokes problem, the dimension of the *FE* basis functions array (3.52) is the following:

$$\begin{aligned} \underline{\phi} &= \left\{ \begin{array}{ccc} \mathbb{R}^{\mathcal{N}_v \times 1} & \mathbb{R}^{\mathcal{N}_v \times 1} & \mathbb{R}^{\mathcal{N}_p \times 1} \end{array} \right\}^T \\ &= \left\{ \begin{array}{ccc} \mathbb{R}^{3\mathcal{N}_u \times 1} & \mathbb{R}^{3\mathcal{N}_u \times 1} & \mathbb{R}^{\mathcal{N}_p \times 1} \end{array} \right\}^T . \end{aligned} \tag{3.53}$$

Then our basis can be expressed as

$$\mathbf{v}_n = \sum_{i=1}^{2\mathcal{N}_v + \mathcal{N}_p} \mathbf{v}_{n_i} \phi_i, \quad 1 \leqslant n \leqslant N_{max} \tag{3.54}$$

Similarly our orthogonalized functions $\boldsymbol{\gamma}_n$ can be expressed as

$$\boldsymbol{\gamma}_n = \sum_{i=1}^{2\mathcal{N}_v + \mathcal{N}_p} \gamma_{n_i} \phi_i, \quad 1 \leqslant n \leqslant N_{max}. \tag{3.55}$$

Where we have sorted the *FE* coefficients in an array, dividing those related to the velocity, to the supremizer and those relatives to the pressure:

$$\begin{aligned} \underline{\mathbf{v}}_n &= \left\{ \underline{\mathbf{w}}_n^{\mu} \; ; \; \underline{q}_n \right\} \\ &= \left\{ \left\{ \underline{\mathbf{w}}_n \; ; \; \underline{\mathbf{w}}_{sup_n} \right\} \; ; \; \underline{q}_n \right\} \\ &= \left\{ \left\{ \mathbf{w}_n^1 \dots \mathbf{w}^{\mathcal{N}_v} \right\}^T ; \left\{ \mathbf{w}_{sup_n}^1 \dots \mathbf{w}_{sup_n}^{\mathcal{N}_v} \right\}^T \; ; \; \left\{ q_n^1 \dots q_n^{\mathcal{N}} \right\}^T \right\}. \end{aligned} \tag{3.56}$$

We then introduce the algebraic representation of the inner product $(\cdot, \cdot)_Y$, denoted by $\underline{\underline{\mathbb{Y}}} \in \mathbb{R}^{(2\mathcal{N}_v + \mathcal{N}_p) \times (2\mathcal{N}_v + \mathcal{N}_p)}$:

$$\underline{\underline{\mathbb{Y}}} = \begin{bmatrix} \underline{\underline{\mathbb{V}}}^{\mu} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{\mathbb{M}}} \end{bmatrix} \left(= \begin{bmatrix} \mathbb{R}^{2\mathcal{N}_v \times 2\mathcal{N}_v} & \mathbb{R}^{2\mathcal{N}_v \times \mathcal{N}_p} \\ \mathbb{R}^{\mathcal{N}_p \times 2\mathcal{N}_v} & \mathbb{R}^{\mathcal{N}_p \times \mathcal{N}_p} \end{bmatrix} \right) \tag{3.57}$$

where the sub-matrices relatives to the velocity and supremizer are defined as follows

$$\underline{\underline{\mathbb{V}}}^{\mu} = \begin{bmatrix} \underline{\underline{\mathbb{V}}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{\mathbb{V}}} \end{bmatrix} \left(= \begin{bmatrix} \mathbb{R}^{\mathcal{N}_v \times \mathcal{N}_v} & \mathbb{R}^{\mathcal{N}_v \times \mathcal{N}_v} \\ \mathbb{R}^{\mathcal{N}_v \times \mathcal{N}_v} & \mathbb{R}^{\mathcal{N}_v \times \mathcal{N}_v} \end{bmatrix} \right) \tag{3.58}$$

$$\underline{\underline{\mathbb{V}}}_{ij} = \left(\phi_i^v, \phi_j^v\right)_V, \ 1 \leqslant i, j \leqslant \mathcal{N}_v \tag{3.59}$$

whereas the sub-matrices relatives to the pressure $\underline{\underline{\mathbb{M}}} \in \mathbb{R}^{\mathcal{N}_p \times \mathcal{N}_p}$

$$\underline{\underline{\mathbb{M}}}_{ij} = \left(\phi_i^p, \phi_j^p\right)_M, \ 1 \leqslant i, j \leqslant \mathcal{N}_p. \tag{3.60}$$

The orthogonalization process can be now explained

$$\begin{aligned}
&\underline{\gamma}_1 = \underline{\mathbf{v}}_1 / \sqrt{\underline{\mathbf{v}}_1{}^T \underline{\underline{\mathbb{V}}}\, \underline{\mathbf{v}}_1}; \\
&\texttt{for } n = 2 : N_{max} \\
&\quad \underline{\mathbf{z}}_n = \underline{\mathbf{v}}_n - \sum_{m=1}^{n-1} \left(\underline{\mathbf{v}}_n \, \underline{\underline{\mathbb{V}}}\, \underline{\gamma}_m\right) \underline{\gamma}_m; \\
&\quad \underline{\gamma}_n = \underline{\mathbf{z}}_n / \sqrt{\underline{\mathbf{z}}_n{}^T \underline{\underline{\mathbb{V}}}\, \underline{\mathbf{z}}_n}; \\
&\texttt{end}
\end{aligned} \tag{3.61}$$

Finally we introduce the *"basis"* matrices $\underline{\underline{\mathbb{Z}}}_N \in \mathbb{R}^{(\mathcal{N}_v + \mathcal{N}_p) \times 3N}$, that will be prove useful to build our *RB* system:

$$\begin{aligned}
\underline{\underline{\mathbb{Z}}}_N = &\begin{bmatrix} \underline{\underline{\mathbb{Z}}}^{vel} & \underline{\underline{\mathbb{Z}}}^{sup} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{\mathbb{Z}}}^{pre} \end{bmatrix} \\
&\left(= \begin{bmatrix} \mathbb{R}^{\mathcal{N}_v \times N} & \mathbb{R}^{\mathcal{N}_v \times N} & \mathbb{R}^{\mathcal{N}_v \times N} \\ \mathbb{R}^{\mathcal{N}_p \times N} & \mathbb{R}^{\mathcal{N}_p \times N} & \mathbb{R}^{\mathcal{N}_p \times N} \end{bmatrix} \right)
\end{aligned} \tag{3.62}$$

where the three contributes, are relatives to the velocity, supremizer and pressure respectively:

$$\begin{aligned}
&\underline{\underline{\mathbb{Z}}}^{vel}_{jn} = \zeta_j^n, && 1 \leqslant j \leqslant \mathcal{N}_v, \ 1 \leqslant n \leqslant N \\
&\underline{\underline{\mathbb{Z}}}^{sup}_{jn} = T^{\mu} \psi_j^n, && 1 \leqslant j \leqslant \mathcal{N}_v, \ 1 \leqslant n \leqslant N \\
&\underline{\underline{\mathbb{Z}}}^{pre}_{jn} = \psi_j^n, && 1 \leqslant j \leqslant \mathcal{N}_p, \ 1 \leqslant n \leqslant N
\end{aligned} \tag{3.63}$$

3.3.4 Galerkin projection

It is now possible to state the reduced problem as follows: find $(\mathbf{u}_N(\boldsymbol{\mu}), p_N(\boldsymbol{\mu})) \in V_N^{\mu} \times M_N \ \left(\equiv Y_N^{\mu}\right)$ such that

$$\begin{cases} a(\mathbf{u}_N(\boldsymbol{\mu}), \mathbf{w}; \boldsymbol{\mu}) + b(p_N(\boldsymbol{\mu}), \mathbf{w}) = f(\mathbf{w}; \boldsymbol{\mu}) & \forall w \in V_N^{\mu} \\ b(q, \mathbf{u}_N(\boldsymbol{\mu})) = 0 & \forall q \in M_N \end{cases} \tag{3.64}$$

and then evaluate

$$s_N(\mu) = l(\mathbf{v}_N;\mu) = l_v(\mathbf{u}_N;\mu) + l_p(p_N;\mu). \tag{3.65}$$

In order to satisfy the *Lax-Milgram theorem* [Bab71] it is necessary to ensure the continuity of b, the coercivity and continuity of the bilinear form a, and finally it is necessary to ensure the continuity and the inf-sup stability of b.
It is possible to show that, thanks to our hierarchical space and thanks to the choice of the spaces for the pressure and for the velocity, $\forall\,\mu \in \mathcal{D}$ the following results holds:

1. *stability*

   a) *coercivity* on a

   $$\alpha_N(\mu) \geqslant \alpha^{\mathcal{N}}(\mu) \geqslant 0 \tag{3.66}$$

   b) *inf-sup condition* on b

   $$\beta_N(\mu) \geqslant \alpha^{\mathcal{N}}(\mu) \geqslant 0 \tag{3.67}$$

2. *continuity*

   a) bilinear form a

   $$\gamma_{a_N}(\mu) \leqslant \gamma_a^{\mathcal{N}}(\mu) \leqslant \infty \tag{3.68}$$

   b) bilinear form b

   $$\gamma_{b_N}(\mu) \leqslant \gamma_b^{\mathcal{N}}(\mu) \leqslant \infty \tag{3.69}$$

finally the linear functional g remains bounded over the dual norm of V_N^μ, therefore the reduced problem 3.64 has a unique solution. Thanks to the Galerkin projection, for the *RB* approximation, the subsequent optimality result holds:

Proposition 3.2. *For any* $\mu \in \mathcal{D}$ *and* $u_N(\mu)$, $p_N(\mu)$ *and* $s_N(\mu)$ *satisfying 3.64-3.65 , see [QV97]:*

- *For the velocity*

$$|||\mathbf{u}^{\mathcal{N}_v} - \mathbf{u}_N^{\mathcal{N}_v}|||_\mu = \inf_{w_N \in V_N^\mu} |||\mathbf{u}^{\mathcal{N}_v} - \mathbf{u}_N|||_\mu \tag{3.70}$$

$$\|\mathbf{u}^{\mathcal{N}_v} - \mathbf{u}_N^{\mathcal{N}_v}\|_{V_N^\mu} \leqslant \left(1 + \frac{\gamma_a^e}{\beta^e}\right)\left(1 + \frac{\gamma_a^e}{\alpha^e}\right) \inf_{\mathbf{w}_N \in V_N^\mu} \|\mathbf{u}^{\mathcal{N}_v} - \mathbf{w}_N\|_V + \\ + \frac{\gamma_b^e}{\alpha^e} \inf_{q_N \in M_N} \|p^{\mathcal{N}_p} - q_N\|_M \tag{3.71}$$

- *For the pressure*

$$|||p^{\mathcal{N}_p} - p_N^{\mathcal{N}_p}|||_\mu = \inf_{q_N \in M_N} |||p^{\mathcal{N}_p} - q_N|||_\mu \tag{3.72}$$

$$\begin{aligned} \|p^{\mathcal{N}_p} - p_N\|_M \leqslant \frac{\gamma_a^e}{\beta^e}\left(1+\frac{\gamma_a^e}{\alpha^e}\right) \inf_{w_N \in V_N^\mu} \|u^{\mathcal{N}_v} - w_N\|_V + \\ + \left(1+\frac{\gamma_b^e}{\beta^e}+\frac{\gamma_b^e\gamma_a^e}{\alpha^e\beta^e}\right) \inf_{q_N \in M_N} \|p^{\mathcal{N}_p} - q_N\|_M \end{aligned} \tag{3.73}$$

We can see there is a coupling between the velocity and the "true" errors.
In this results, thanks to the properties 3.66-3.69 we can replace the exact constants α^e, β^e, γ_a^e, γ_b^e with their *RB* approximated counterparts α_N, β_N, γ_{a_N}, γ_{b_N}.

3.3.5 Offline-Online procedure

In this section it will be discussed the operations needed to reduce the original problem into a *RB* algebraic system, moreover the Offline/Online splitting procedure will be discussed and finally the computational burden reduction offered by the *RB* method will be explained.

Algebraic formulation

Recalling the affine dependence on the parameter of the supremizer operator (2.24) $T^\mu q : M_N \to V_N^\mu$, we write

$$T^\mu q = \sum_{k=1}^{Q_b} \phi^k(\mu)\, \mathbb{T}^k q, \tag{3.74}$$

thanks to the linearity of T^μ, we can rewrite V_N^μ (3.46) in the following fashion:

$$V_N^\mu = \text{span}\left\{ \sum_{k=1}^{\overline{Q}_b = Q_b+1} \phi^k(\mu)\, \sigma_{kn}, \;\; n = 1,\dots,2N \right\} \tag{3.75}$$

where $\phi^k(\mu)$ is defined as

$$\phi^k(\mu) = \begin{cases} \theta_b(\mu) & \text{if } \; k = 1,\dots,Q_b \\ 1 & \text{if } \; k = \overline{Q}_b \end{cases} \tag{3.76}$$

and where $\boldsymbol{\sigma}_{kn}$ are defined as

$$\boldsymbol{\sigma}_{kn} = \begin{cases} \begin{cases} 0 & \text{for } k = 1, \dots, Q_b \\ \zeta(\boldsymbol{\mu}^n) & \text{for } k = \overline{Q}_b \end{cases} & n = 1, \dots, N \\ \begin{cases} \mathbb{T}^k \psi(\boldsymbol{\mu}^{n-N}) & \text{for } k = 1, \dots, Q_b \\ 0 & \text{for } k = \overline{Q}_b \end{cases} & n = N+1, \dots, 2N. \end{cases} \tag{3.77}$$

In this way we have compacted the *reduced basis velocity space* made up of velocity ζ_n and supremizer solutions $T^{\mu}\psi_n$. In order to introduce the algebraic formulation, we now expand $\mathbf{u}_N(\boldsymbol{\mu})$ and $p(\boldsymbol{\mu})$ as a combination of the precomputed solutions as basis functions:

$$\mathbf{u}_N(\boldsymbol{\mu}) = \sum_{j=1}^{2N} u_{N_j}(\boldsymbol{\mu}) \left(\sum_{k=1}^{\overline{Q}_b} \phi^k(\boldsymbol{\mu})\, \boldsymbol{\sigma}_{kj} \right), \tag{3.78a}$$

$$p_N(\boldsymbol{\mu}) = \sum_{m=1}^{N} p_{N_m}\, \psi_m. \tag{3.78b}$$

We now expand the *RB* solution $\mathbf{u}_N$ and p_N thanks to the development above, then we choose $\mathbf{w}^{\mu} \in V_N^{\mu}$ as our *velocity test functions* and $q \in M_N$ as our *pressure test function*.
Replacing this quantities in the problem 3.64, after few operations we obtain:

$$\begin{cases} \sum_{j=1}^{2N} \underline{\underline{A}}^{\mu}_{N_{ij}}\, u_{N_j}(\boldsymbol{\mu}) + \sum_{m=1}^{N} \underline{\underline{B}}^{\mu}_{N_{im}}\, p_{N_m}(\boldsymbol{\mu}) = \underline{F}^{\mu}_i \\ \sum_{j=1}^{2N} \underline{\underline{B}}^{\mu}_{N_{jm}}\, u_{N_j}(\boldsymbol{\mu}) = \underline{G}^{\mu}_N \end{cases} \tag{3.79}$$

where the reduced matrices $\underline{\underline{A}}^{\mu}_N \in \mathbb{R}^{(2N)\times(2N)}$, $\underline{\underline{B}}^{\mu}_N \in \mathbb{R}^{(2N)\times(N)}$ and the reduced vectors $\underline{F}^{\mu}_N \in \mathbb{R}^{2N}$, $\underline{G}^{\mu}_N \in \mathbb{R}^{N}$, exploiting the affine decomposition 2.56-2.59 (pag.22), can be obtained as follows:

$$\begin{aligned} \underline{\underline{A}}^{\mu}_{N_{ij}} &= \sum_{k=1}^{Q_a} \sum_{k'=1}^{\overline{Q}_b} \sum_{k''=1}^{\overline{Q}_b} \theta_a^k(\boldsymbol{\mu})\, \phi^{k'}(\boldsymbol{\mu})\, \phi^{k''}(\boldsymbol{\mu})\, a^k\left(\boldsymbol{\sigma}_{k'i}, \boldsymbol{\sigma}_{k''j}\right) \\ &= \sum_{k=1}^{Q_a} \sum_{k'=1}^{\overline{Q}_b} \sum_{k''=1}^{\overline{Q}_b} \theta_a^k(\boldsymbol{\mu})\, \phi^{k'}(\boldsymbol{\mu})\, \phi^{k''}(\boldsymbol{\mu})\, \underline{\underline{\mathbb{A}}}^{k}_{N_{ij}} \qquad 1 \leqslant i, j \leqslant 2N \end{aligned} \tag{3.80a}$$

$$\begin{aligned}\underline{\underline{B}}^{\mu}_{N_{im}} &= \sum_{k=1}^{Q_b}\sum_{k'=1}^{\overline{Q}_b} \phi^k(\mu)\,\phi^{k'}(\mu)\,b^k\left(\boldsymbol{\sigma}_{k'i},\psi_m\right)\\ &= \sum_{k=1}^{\overline{Q}_b}\sum_{k'=1}^{\overline{Q}_b} \phi^k(\mu)\,\phi^{k'}(\mu)\,\underline{\underline{\mathbb{B}}}^{k}_{N_{im}} \qquad (3.80b)\\ &\qquad\qquad 1\leqslant i\leqslant 2N,\ 1\leqslant m\leqslant N\end{aligned}$$

$$\begin{aligned}\underline{F}^{\mu}_{N_i} &= \sum_{k=1}^{Q_f}\sum_{k'=1}^{\overline{Q}_f} \theta^k_f(\mu)\,\phi^{k'}(\mu)\,f^k\left(\boldsymbol{\sigma}_{k'i}\right)\\ &= \sum_{k=1}^{Q_f}\sum_{k'=1}^{\overline{Q}_f} \theta^k_f(\mu)\,\phi^{k'}(\mu)\,\underline{\mathbb{F}}^{k}_{N_i} \qquad (3.80c)\\ &\qquad\qquad 1\leqslant i\leqslant 2N\end{aligned}$$

$$\begin{aligned}\underline{G}^{\mu}_{N_m} &= \sum_{k=1}^{Q_g} \theta^k_g(\mu)\,g^k\left(\psi_m\right)\\ &= \sum_{k=1}^{Q_g} \theta^k_g(\mu)\,\underline{\mathbb{G}}^{k}_{N_m} \qquad (3.80d)\\ &\qquad\qquad 1\leqslant m\leqslant N.\end{aligned}$$

Then to evaluate the output (3.65), the reduced basis formulation reads:

$$\begin{aligned}s^{\mu}_N(\mu) &= \sum_{k=1}^{Q_{l_v}}\sum_{k'=1}^{\overline{Q}_b} \theta^k_{l_v}(\mu)\,\phi^{k'}(\mu)\,l^k_{\mathbf{v}}\left(\boldsymbol{\sigma}_{k'i}\right) + \sum_{k=1}^{Q_{l_p}} \theta^k_{l_p}(\mu)\,l^k_{p}\left(\psi_m\right)\\ &= \sum_{k=1}^{Q_{l_v}}\sum_{k'=1}^{\overline{Q}_b} \theta^k_{l_v}(\mu)\,\phi^{k'}(\mu)\,\underline{\mathbb{L}}^{k}_{\mathbf{v}_i} + \sum_{k=1}^{Q_{l_p}} \theta^k_{l_p}(\mu)\,\underline{\mathbb{L}}^{k}_{p_{N_m}}. \qquad (3.81)\end{aligned}$$

Finally, problem 3.64 can be written in compact $\mathbb{R}^{3N\times 3N}$ form as

$$\begin{bmatrix} \underline{\underline{A}}^{\mu}_N & \underline{\underline{B}}^{\mu}_N \\ \underline{\underline{B}}^{\mu^T}_N & \underline{\underline{0}}_N \end{bmatrix} \begin{Bmatrix} \underline{\mathbf{u}}_N \\ \underline{p}_N \end{Bmatrix} = \begin{Bmatrix} \underline{F}^{\mu}_N \\ \underline{G}^{\mu}_N \end{Bmatrix} \qquad (3.82)$$

where $\underline{\underline{0}}_N$ is a null matrix $\in \mathbb{R}^{N\times N}$.
This linear system whose unknowns are the *RB* coefficients $\mathbf{u}_i$, $1\leqslant i\leqslant 2N$ and p_m, $1\leqslant m\leqslant N$, has the same structure of a *FE* Stokes problem. The *RB* method builds a considerably smaller system (order of N) and with full matrices, where *FE* creates sparse matrices.
Then, introducing the *RB* vectors $\underline{L}_{\mathbf{v}_N}\in\mathbb{R}^{2N}$ and $\underline{L}_{p_N}\in\mathbb{R}^{N}$, the output can be written in a compact form as

$$s^{\mu}_N = \begin{Bmatrix} \underline{L}_{\mathbf{v}_N} & \underline{L}_{p_N} \end{Bmatrix} \begin{Bmatrix} \underline{\mathbf{u}}_N \\ \underline{p}_N \end{Bmatrix}. \qquad (3.83)$$

It is possible to exploit the link between the *RB* matrices and vectors introduce above, in order to provide the Offline/Online computational cost.

3.3.5.1 *Offline/Online*

The *RB* system 3.82 is clearly of small size, in fact is an $3N \times 3N$ linear system, that requires $\mathcal{O}(27N^3)$ operations to solve it, plus $9N$ operations to evaluate the output from equation 3.83. Thanks to the affine parameter dependence we have obtained an Offline/Online splitting procedure in order to build the *RB* system.

Offline

In the Offline part we form:

1. the $\boldsymbol{\mu}$-independent matrices $\underline{\underline{A}}_N^q \in \mathbb{R}^{2N\times 2N}$

$$\underline{\underline{\mathbb{A}}}^q_{N_{mn}} = a^q(\boldsymbol{\sigma}_m, \boldsymbol{\sigma}_n), \quad 1 \leqslant m, n \leqslant 2N,\ 1 \leqslant q \leqslant Q_a \tag{3.84}$$

2. the $\boldsymbol{\mu}$-independent matrices $\underline{\underline{B}}_N^q \in \mathbb{R}^{2N\times 2N}$

$$\underline{\underline{\mathbb{B}}}^q_{N_{im}} = b^q(\boldsymbol{\sigma}_i, \boldsymbol{\psi}_m), \quad 1 \leqslant i \leqslant 2N,\ 1 \leqslant m \leqslant N,\ 1 \leqslant q \leqslant Q_b \tag{3.85}$$

3. the $\boldsymbol{\mu}$-independent vectors $\underline{\mathbb{F}}^q_N \in \mathbb{R}^{2N}$

$$\underline{\mathbb{F}}^q_{N_i} = f^q(\boldsymbol{\sigma}_i), \quad 1 \leqslant i \leqslant 2N,\ 1 \leqslant q \leqslant Q_f \tag{3.86}$$

4. the $\boldsymbol{\mu}$-independent vectors $\underline{\mathbb{G}}^q_N \in \mathbb{R}^N$

$$\underline{\mathbb{G}}^q_{N_m} = g^q(\boldsymbol{\psi}_m), \quad 1 \leqslant m \leqslant N,\ 1 \leqslant q \leqslant Q_g \tag{3.87}$$

Online

In the Online stage we assemble, for any new $\boldsymbol{\mu} \in \mathcal{D}$:

1. the *RB* matrices:
   - $\underline{\underline{A}}^{\boldsymbol{\mu}}_N$, equation 3.80a;
   - $\underline{\underline{B}}^{\boldsymbol{\mu}}_N$, equation 3.80b;
2. the *RB* vectors
   - $\underline{F}^{\boldsymbol{\mu}}_N$, equation 3.80c;
   - $\underline{G}^{\boldsymbol{\mu}}_N$, equation 3.80d.

Link between FE *and* RB

We now recall, and slightly modify, the definition of the *"basis"* matrices $\underline{\underline{\mathbb{Z}}}_N$ (eq.3.62) to provide the link between the *FE* and *RB* matrices expressed in a matricial way.
In the *RB* system we have already compacted the velocity and supremizer thanks to 3.75. Therefore to obtain the desired relation accordingly to the notation used in 3.79, it will prove useful to write the basis matrices in the following compact form:

$$\underline{\underline{\mathbb{Z}}}_N^{\mu} = \begin{bmatrix} \underline{\underline{\mathbb{Z}}}_N^{vel^{\mu}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{\mathbb{Z}}}_N^{pre} \end{bmatrix} \left(= \begin{bmatrix} \mathbb{R}^{\mathcal{N}_v \times 2N} & \mathbb{R}^{\mathcal{N}_v \times N} \\ \mathbb{R}^{\mathcal{N}_p \times 2N} & \mathbb{R}^{\mathcal{N}_p \times N} \end{bmatrix} \right) \tag{3.88}$$

where:

$$\begin{aligned} \underline{\underline{\mathbb{Z}}}_{N_{jn}}^{vel^{\mu}} &= \sigma_{kn}^{j},\ 1 \leqslant k \leqslant \overline{Q}_b,\ 1 \leqslant n \leqslant 2N,\ 1 \leqslant j \leqslant 2\mathcal{N}_v \\ \underline{\underline{\mathbb{Z}}}_{N_{jn}}^{pre} &= \psi_n^j,\ \ 1 \leqslant n \leqslant N,\ 1 \leqslant j \leqslant \mathcal{N}_p \end{aligned} \tag{3.89}$$

We now recover the definition of the *RB* vectors and matrices, showing the dependence upon the *FE* basis functions.
We start from the bilinear form a of equation 3.80a:

$$\begin{aligned} a^k(\boldsymbol{\sigma}_{k'n}, \boldsymbol{\sigma}_{k''m}) &= \sum_{i=1}^{\mathcal{N}_v} \sum_{j=1}^{\mathcal{N}_v} \sigma_{k'n}^i\, a^k(\phi_i^v, \phi_i^v)\, \sigma_{k''m}^j \\ &= \sum_{i=1}^{\mathcal{N}_v} \sum_{j=1}^{\mathcal{N}_v} \sigma_{k'n}^i\, \underline{\underline{\mathbb{A}}}_{ij}^k\, \sigma_{k''m}^j \quad 1 \leqslant n, m \leqslant 2N \end{aligned} \tag{3.90}$$

then, as regard to the bilinear form b (3.80b):

$$\begin{aligned} b^k(\boldsymbol{\sigma}_{k'n}, \psi_m) &= \sum_{i=1}^{\mathcal{N}_v} \sum_{m=1}^{\mathcal{N}_p} \sigma_{k'n}^i\, b^k\left(\phi_i^v, \phi_j^p\right) \psi_m^j \\ &= \sum_{i=1}^{\mathcal{N}_v} \sum_{m=1}^{\mathcal{N}_p} \sigma_{k'n}^i\, \underline{\underline{\mathbb{B}}}_{ij}^k\, \psi_m^j \quad 1 \leqslant n \leqslant 2N,\ 1 \leqslant m \leqslant N. \end{aligned} \tag{3.91}$$

The linear functional f (3.80c):

$$\begin{aligned} f^k(\boldsymbol{\sigma}_{k'i}) &= \sum_{i=1}^{\mathcal{N}_v} \sigma_{k'n}^i\, f^k(\phi_i^v) \\ &= \sum_{i=1}^{\mathcal{N}_v} \sigma_{k'n}^i\, \underline{\mathbb{F}}^k \quad 1 \leqslant n \leqslant 2N, \end{aligned} \tag{3.92}$$

and finally g (3.80b) can be written as:

$$\begin{aligned} g^k(\psi_m) &= \sum_{i=1}^{\mathcal{N}_p} \psi_m^i \, g\left(\phi_i^p\right) N \\ &= \sum_{i=1}^{\mathcal{N}_p} \psi_m^i \, \underline{G}^k \quad 1 \leqslant m \leqslant N. \end{aligned} \tag{3.93}$$

Endowed with the definition of the basis matrix of equation 3.88, we obtain the desired matricial relation:

$$\begin{aligned} &\begin{bmatrix} \underline{\underline{A}}_N^{\mu} & \underline{\underline{B}}_N^{\mu} \\ \underline{\underline{B}}_N^{\mu^T} & \underline{\underline{0}}_N \end{bmatrix} = \\ &= \begin{bmatrix} \underline{\underline{\mathbb{Z}}}_N^{vel^{\mu}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{\mathbb{Z}}}_N^{pre} \end{bmatrix}^T \begin{bmatrix} \underline{\underline{A}} & \underline{\underline{B}} \\ \underline{\underline{B}}^T & \underline{\underline{0}} \end{bmatrix} \begin{bmatrix} \underline{\underline{\mathbb{Z}}}_N^{vel^{\mu}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{\mathbb{Z}}}_N^{pre} \end{bmatrix} \end{aligned} \tag{3.94}$$

by this, we pass (Offline) from the *FE* matrices to their *RB* counterparts.

3.3.5.2 *Operation count and storage*

Also in the Stokes case, we achieve an Online $\mathcal{N}$-independent stage, hence very inexpensive. It is necessary to focus on the Offline and Online complexity to quantify the computational reduction provided by the *RB* method.
We make use of Table 2 to summarize the computational burden requested to perform: (i) Offline, the *RB* basis assembling, (ii) Online, a single input/output evaluation.

3.4 SAMPLE/SPACE ASSEMBLING

We now discuss the procedure used to select the snapshots in order to assemble the reduced basis approximation spaces, after a few preliminaries. We then turn to the *Greedy sampling* strategy exploited in this thesis [PR09]. See also [PR09, HRSP07, RHP08] for more options sampling strategies.

We shall denote by Ξ a finite sample of points in $\mathcal{D}$. These "test" samples Ξ serve as surrogates for $\mathcal{D}$ in the calculation and presentation of errors over the parameter domain. Typically these samples are chosen by *Monte Carlo* methods with respect to a uniform or log $-$uniform density. Concerning the dimension of the sample, we always ensure that Ξ is sufficiently large that the reported results are *insensitive* to further refinement of the parameter sample.

part	item	burden	equation
Offline	$\underline{\underline{\mathbb{Z}}}_N$	$2N\,\mathcal{O}((\mathcal{N}_v+\mathcal{N}_p)^3)$	3.19
	$\underline{\underline{\mathbb{A}}}_N^k$	$Q_a\,N\,\underline{\underline{A}}$-matvec $+\,Q_a\,N^2\,X^{\mathcal{N}_v}$-inprod	3.84
	$\underline{\underline{\mathbb{B}}}_N^k$	$Q_b\,N\,\underline{\underline{B}}$-matvec $+\,Q_b\,N^2\,X^{\mathcal{N}_p}$-inprod	3.85
	$\underline{\mathbb{F}}_N$	$Q_f\,N\,X^{\mathcal{N}_v}$-inprod	3.86
	$\underline{\mathbb{G}}_N$	$Q_g\,N\,X^{\mathcal{N}_p}$-inprod	3.87
Online	$\underline{\underline{A}}_N^{\boldsymbol{\mu}}$	$Q_a\,N^2$	3.80a
	$\underline{\underline{B}}_N^{\boldsymbol{\mu}}$	$Q_b\,N^2$	3.80b
	$\underline{F}_N^{\boldsymbol{\mu}}$	$Q_f\,N$	3.80c
	$\underline{G}_N^{\boldsymbol{\mu}}$	$Q_g\,N$	3.80d
	$(\mathbf{u}_N, p_N)$	$\mathcal{O}(27N^3)$	3.27
	s_N	$9N^2$	3.83

Table 2: Offline/Online: Stokes case

Definition 3.2. *Given a function* $y : \mathcal{D} \to \mathbb{R}$*, we define the* $L^\infty(\Xi)$ *and* $L^p(\Xi)$ *norms respectively as:*

$$\begin{aligned} \|y\|_{L^\infty(\Xi)} &\equiv \max_{\boldsymbol{\mu}\in\Xi} |y(\boldsymbol{\mu})| \\ \|y\|_{L^p(\Xi)} &\equiv \left(|\Xi|^{-1} \sum_{\boldsymbol{\mu}\in\Xi} |y(\boldsymbol{\mu})|^p \right)^{1/p}. \end{aligned} \tag{3.95}$$

Definition 3.3. *Given a function* $z : \mathcal{D} \to X^{\mathcal{N}}$ *(or* X^e*), we define the* $L^\infty(\Xi; X)$ *and* $L^p(\Xi; X)$ *norms respectively as:*

$$\begin{aligned} \|z\|_{L^\infty(\Xi;X)} &\equiv \max_{\boldsymbol{\mu}\in\Xi} |z(\boldsymbol{\mu})|_X \\ \|z\|_{L^p(\Xi;X)} &\equiv \left(|\Xi|^{-1} \sum_{\boldsymbol{\mu}\in\Xi} \|z(\boldsymbol{\mu})\|_X^p \right)^{1/p}. \end{aligned} \tag{3.96}$$

Here $|\Xi|$ denotes the cardinality of (the finite number of elements in) the test sample Ξ.
We now introduce the *Greedy Lagrange spaces*, that will be used to build our *RB* approximation.

3.4.1 Greedy Lagrange spaces

We have already introduced the concept of *Lagrange spaces* (see section 3.2.1, pag.29), we now have to extend this idea to Greedy Lagrange spaces.

We remark that this strategy is not indispensable to build a basic model reduction, but it is a *prerogative* of the *Reduced Basis method*. In fact the *Greedy* sampling, we are going to discuss, can be efficiently exploited only combined to an Offline/Online splitting procedure.
The idea of this strategy is starting with a train sample Ξ_{train}, we select N parameters $\mu_1, \dots, \mu_N$ and, as already seen in section 3.2.1, we form the reduced basis space X^N as:

$$X^N = \text{span}\left\{\xi_n = u^{\mathcal{N}}(\boldsymbol{\mu_n}),\ 1 \leqslant n \leqslant N\right\}. \tag{3.97}$$

More precisely, for the *Greedy approach*, we need a also *sharp*, *rigorous* and *efficient bound* $\Delta_N^{en}(\boldsymbol{\mu})$ for the reduced basis error $\|u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_X$, where u_N is our *RB* approximation associated with the space X_N, [RHP08]. The superscript ^{en} denotes that the bound is related to the *energy norm* of the error, other options are discussed in [PR09].
To quantify the *sharpness* and *rigour* properties, we recall the *effectivity* of an error bound.

Definition 3.4. *The* effectivity *of an error bound, denoted by* η*, is defined as follows*

$$\eta_N^{en} = \frac{\Delta_N^{en}}{\|u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_X} \tag{3.98}$$

we require that

$$1 \leqslant \eta_N^{en} \leqslant \eta_{max,UB}^{en} \quad \forall\, \boldsymbol{\mu} \in \mathcal{D},\ 1 \leqslant N \leqslant N_{max} \tag{3.99}$$

where $\eta_{max,UB}^{en}$ *is finite and* N *independent.*

Proof 1. *It is possible to show that the inequality 3.99 is always fullfilled in the* RB *method, [RHP08].*

The *rigour* property is illustrated by the *left* inequality: the error bound $\Delta_N^{en}(\boldsymbol{\mu})$ is never smaller than the true error $\|u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_X$. The *sharpness* property is illustrated by the *right* inequality: $\Delta_N^{en}(\boldsymbol{\mu})$ *has not to bee much bigger* than the true error. Last, *efficient* means that the evaluation of $\Delta_N^{en}(\boldsymbol{\mu})$ is $\mathcal{N}$ independent, thanks to the Offline/Online procedure that we will show in section 3.5.2. The last property is crucial in the *Greedy* procedure, in fact it permits us to exploit a *very large* train sample Ξ_{train} in order to select the best snapshots to be include in our *RB* approximation spaces.

3.4.2 Greedy algorithm

We define $\overline{N}_{max}$, an upper bound for N_{max} and $\epsilon_{toll,min}$ the desired minimum tolerance over the error bound.
Given Ξ_{train}, $S_1 = \{\mu^1\}$ and $X^1 = \text{span}\{u^{\mathcal{N}}(\mu^1)\}$,

$$
\begin{aligned}
&\texttt{for } N = 2 : \overline{N}_{max} \\
&\quad \mu^N = \arg \max_{\mu \in \Xi_{train}} \\
&\quad \epsilon_{N-1} = \Delta_N^{en}(\mu^N) \\
&\quad \texttt{if } \epsilon_{N-1} \leqslant \epsilon_{toll,min} \\
&\qquad N_{max} = N - 1 \\
&\quad \texttt{end} \\
&\quad S_N = S_{N-1} \cup \mu^N \\
&\quad X_N = X_{N-1} + \text{span}\left\{u^{\mathcal{N}}(\mu^N)\right\} \\
&\texttt{end}
\end{aligned}
\tag{3.100}
$$

In the *Greedy* algorithm the key point is to exploit an approximated (very cheap) error bound $\Delta_N^{en}(\mu^N)$ instead of the true error (hence very expensive) $\|u^{\mathcal{N}}(\mu) - u_N(\mu)\|$.
We remark that the *Greedy* algorithm *heuristically* minimizes the *RB* error bound in $L^\infty(\Xi_{train}; X)$ norm, see [PR09, RHP08]: the algorithm evaluates the error bounds $\forall\, \mu \in \Xi_{train}$, then the next snapshot is selected such that it corresponds to the *maximum error bound*.

3.5 A-POSTERIORI ERROR BOUND

A-posteriori error bounds are crucial in the *RB* methodology. They are important for both *efficiency* and *reliability* of *RB* 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 further reduction in the *Online* computational cost.
In other words, a posteriori error estimation permits us to control the error thus allowing us to *minimize* the computational effort, [PR09].
As regards *reliability*, our *Offline* sampling procedures could not be exhaustive without a Greedy approach. For a large number of parameters P, there would be a large portion of the parameter space $\mathcal{D}$ which would remain unexplored. So, the error of a large parts of the parameter domain $\mathcal{D}$ would be uncharacterized.
The *a-posteriori* error bounds permit to *rigorously* bound the error for all

new value of parameter $\mu^* \in \mathcal{D}$. So we do not lose any confidence in the solution compared to the underlying *FE* solution while exploiting the rapid predictive power of the *RB* approximation.
As mentioned in section 3.4.1, 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 $\mathcal{D}$. Second, the bound must be reasonably *sharp*. An overly conservative error bound can yield inefficient approximations, typically N too large, or *suboptimal* engineering results, for example too much big safety margins.
For the coercive case, see [PR09, RHP08], whereas for the non-coercive case see [Roz08, Rov03].

3.5.1 Preliminaries

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

$$r(v;\mu) = f(v;\mu) - a(u_N^{\mathcal{N}}(\mu), v;\mu)) \quad \forall v \in X^{\mathcal{N}} \tag{3.101}$$

where $\left(X^{\mathcal{N}}\right)'$ is the dual space of $X^{\mathcal{N}}$.
We also introduce the function $\hat{e} : \mathcal{D} \to X^{\mathcal{N}}$, the *Riesz* representation of $r(v;\mu)$, see [Qua09]:

$$(\hat{e}(\mu), v)_X = r(v;\mu) \quad \forall v \in X^{\mathcal{N}}. \tag{3.102}$$

Finally, introducing the real error $e^{\mathcal{N}}(\mu)$ $(\equiv e(\mu))$

$$e(\mu) = u^{\mathcal{N}}(\mu) - u_N^{\mathcal{N}}. \tag{3.103}$$

Recalling that $u^{\mathcal{N}}(\mu)$ and $u_N(\mu)$ satisfies the equations 2.38 and 3.20, respectively, we get from 3.101, 3.102 that the error $e(\mu)$ satisfies the following relation

$$a(e(\mu), v;\mu) = r(v;\mu) = (\hat{e}(\mu), v) \quad \forall v \in X^{\mathcal{N}} \tag{3.104}$$

We note that for our choice of inner product 2.37, $\hat{e}(\mu) = e(\mu)$.
We then define the *dual norm* of $r(\,\cdot\,;\mu)$ associated to the dual space $\left(X^{\mathcal{N}}\right)'$:

$$\|r(v;\mu)\|_{X'} = \sup_{v \in X} \frac{r(v;\mu)}{\|v\|_X} = \|\hat{e}(\mu)\|_X \tag{3.105}$$

Note that the second equality follows from the *Riesz representation theorem*. This definition is crucial for the *Offline-Online* procedure.
Our aim is to find an approximated lower bound for $\alpha^{\mathcal{N}}(\mu)$ (see 2.5), that is a function $\alpha_{LB}^{\mathcal{N}} : \mathcal{D} \to \mathbb{R}$ such that

1. $0 < \alpha_{LB}^{\mathcal{N}}(\mu) \leqslant \alpha^{\mathcal{N}}(\mu) \quad \forall \mu \in \mathcal{D}$
2. the evaluation $\mu \to \alpha_{LB}^{\mathcal{N}}$ should be independent of $\mathcal{N}$

We will discuss the procedure to evaluate this coercivity lower bound in Chapter 3.6.

Error bound estimators

Now we can define our *energy*, *output* and *relative output error bound* estimators, that are defined respectively as, see [PR09, RHP08]:

$$\Delta_N^{en}(\mu) = \frac{\|\hat{e}(\mu)\|_X}{\left(\alpha_{LB^N}(\mu)\right)^{1/2}}, \tag{3.106a}$$

$$\Delta_N^{s}(\mu) = \frac{\|\hat{e}(\mu)\|_X^2}{\alpha_{LB^N}(\mu)}, \tag{3.106b}$$

$$\Delta_N^{s,rel}(\mu) = \frac{\|\hat{e}(\mu)\|_X^2}{\alpha_{LB^N}(\mu)\, s_N^N(\mu)} = \frac{\Delta_N^{s}(\mu)}{s_N^N(\mu)}. \tag{3.106c}$$

Effectivity estimators

As already discussed in Section 3.4, associated to each estimator theres is an *effectivity estimator* as a measure of the *quality* of the error bound estimators and are needed to certify that the *RB* method is *rigorous* and *sharp*. We introduce the following ones:

$$\eta_N^{en}(\mu) = \frac{\Delta_N^{en}(\mu)}{|||e(\mu)|||_\mu}, \tag{3.107a}$$

$$\eta_N^{s}(\mu) = \frac{\Delta_N^{s}(\mu)}{s^N(\mu) - s_N^N(\mu)}, \tag{3.107b}$$

$$\eta_N^{en,rel}(\mu) = \frac{\Delta_N^{s,rel}(\mu)}{\left(s^N(\mu) - s_N^N(\mu)\right)/s^N(\mu)}. \tag{3.107c}$$

It is can be shown that the *effectivities* are a measure of the *rigor* and *sharpness* for an error bound.

Proposition 3.3. *The following results holds (see [RHP08] for the proof):*

$$1 < \eta_N^{en}(\mu) \leqslant \sqrt{\frac{\gamma^e(\mu)}{\alpha_{LB}^N(\mu)}}, \tag{3.108}$$

$$1 < \eta_N^{s}(\mu) \leqslant \frac{\gamma^e(\mu)}{\alpha_{LB}^N(\mu)}, \tag{3.109}$$

and finally, with regard to $\eta_N^{en,rel}(\mu)$*, it can be shown that:*

$$\eta_N^{en,rel}(\mu) = (\eta_N^{en}(\,\mu))^2. \tag{3.110}$$

3.5.2 Offline-Online procedure

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

$$r(v;\mu) = \sum_{q=1}^{Q_f} \theta_f^q(\mu)\, f^q(v) + \sum_{q=1}^{Q_a}\sum_{n=1}^{N} \theta_a^q(\mu)\, u_{N_n}(\mu)\, a^q(\xi_n, v)\,. \tag{3.111}$$

This expansion directly follows from our affine assumption 2.20 and from the *RB* development $u_N(\mu) = \sum_{n=1}^{N} u_{N_n}\,\xi_n$.
Moreover, we have from the equation 3.104 that:

$$(\hat{e}(\mu), v)_X = \sum_{q=1}^{Q_f} \theta_f^q(\mu)\, f^q(v) + \sum_{q=1}^{Q_a}\sum_{n=1}^{N} \theta_a^q(\mu)\, u_{N_n}(\mu)\, a^q(\xi_n, v)\,. \quad (3.112)$$

Consequently, defining

$$(\mathcal{C}^q, v)_X = f^q(v) \qquad 1 \leqslant q \leqslant Q_f \quad (3.113a)$$

$$(\mathcal{L}_n^q, v)_X = -a^q(\xi_n, v) \qquad 1 \leqslant q \leqslant Q_a,\ 1 \leqslant n \leqslant N \quad (3.113b)$$

we can write

$$\hat{e}(\mu) = \sum_{q=1}^{Q_f} \theta_f^q(\mu)\, \mathcal{C}^q + \sum_{q=1}^{Q_a} u_{N_n}(\mu)\, \theta_a^q(\mu)\, \mathcal{L}_n^q \quad (3.114)$$

We remark that 3.113a and 3.113b are parameter-independent *Poisson-like* problems, hence $\mathcal{C}^q$ and $\mathcal{L}_n^q$ are computed *Offline*.
We thus obtain

$$\begin{aligned}
\|\hat{e}(\mu)\|_X &= \left(\sum_{q=1}^{Q_f} \theta_f^q(\mu)\, \mathcal{C}^q + \sum_{q=1}^{Q_a}\sum_{n=1}^{N} \theta_a^q(\mu)\, u_{n_N}(\mu)\, \mathcal{L}_n^q, \text{"} \right)_X \\
&= \sum_{q=1}^{Q_f}\sum_{q'=1}^{Q_f} \theta_f^q(\mu)\theta_f^{q'}(\mu) \left(\mathcal{C}^q, \mathcal{C}^{q'}\right)_X + \\
&\quad + \sum_{q=1}^{Q_a}\sum_{n=1}^{N} \theta_a^q(\mu)\, u_{N_n} \left\{ 2 \sum_{q'=1}^{Q_f} \theta_a^q(\mu)\theta_f^{q'}(\mu) \left(\mathcal{L}_n^q, \mathcal{C}^{q'}\right)_X + \right. \\
&\quad \left. + \sum_{q'=1}^{Q_a}\sum_{n'=1}^{N} \theta_a^{q'} \left(\mathcal{L}_n^q, \mathcal{L}_{n'}^{q'}\right)_X \right\}. \qquad (3.115)
\end{aligned}$$

The Offline-Online procedure is clear. In the Offline stage, we first compute $\mathcal{C}^q$, $1 \leqslant q \leqslant Q_f$ and $\mathcal{L}_n^q$, $1 \leqslant q \leqslant Q_a$, $1 \leqslant n \leqslant N$, then we compute and store the quantities:

$$\left(\mathcal{C}^q, \mathcal{C}^{q'}\right)_X \qquad 1 \leqslant q \leqslant Q_f,\ 1 \leqslant q' \leqslant Q_f \quad (3.116)$$

$$(\mathcal{L}_n^q, \mathcal{C}^q)_X \qquad 1 \leqslant q \leqslant Q_a,\ 1 \leqslant q' \leqslant Q_f \quad (3.117)$$

$$(\mathcal{L}_n^q, \mathcal{L}^q)_X \qquad 1 \leqslant q \leqslant Q_a,\ 1 \leqslant q \leqslant Q_a \quad (3.118)$$

In the *Online* stage we evaluate the expression 3.115 which consists in a sum.
The computational cost to perform this evaluation is:

$$n^2 \times Q_a^2 + 2n \times Q_a \times Q_f + n \times Q_f^2, \quad (3.119)$$

so it is $\mathcal{N}$ independent, hence very cheap.

3.6 SUCCESSIVE CONSTRAINT METHOD

We now discuss the successive constraint method (*SCM*). This tool enables the construction of lower (and upper) bounds for the *coercivity* and *inf–sup* stability constants (defined in 2.5 and 2.7 respectively), required in a posteriori error analysis of *RB* approximations. The method, based on an *Offline–Online* strategy, reduces the *Online* calculation to a small *Linear Programming problem*: the *objective* is a parametric expansion of the underlying *Rayleigh* quotients, the *constraints* reflect stability information at optimally selected parameter points. The *state of the art* method is presented in [HRSP07], see also [RHP08, PR09].

NB: *Without risk of a global comprehension loss, the reader can ⏭ to proposition 3.4*

3.6.1 Coercive case

We define

$$\mathcal{Y} \equiv \left\{ y = (y_1 \dots y_{Q_a}) \in \mathbb{R}^{Q_a} \mid \exists w_y \in X^{\mathcal{N}} \text{ s.t. } y_q = \frac{a^q(w_y, w_y)}{\|w\|_{X^{\mathcal{N}}}},\ 1 \leqslant q \leqslant Q_a \right\}. \tag{3.120}$$

We further define the *objective function* $\mathscr{F} : \mathcal{D} \times \mathbb{R}^{Q_a} \to \mathbb{R}$ as

$$\mathscr{F}(y; \boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \theta_a^q(\boldsymbol{\mu})\, y_q. \tag{3.121}$$

We may then write our *coercivity* constant as

$$\alpha^{\mathcal{N}}(\boldsymbol{\mu}) = \min_{y \in \mathcal{Y}} \mathscr{F}(y; \boldsymbol{\mu}) \tag{3.122}$$

We next introduce a *constraint box* that is the set of all the feasible value for y, defined as

$$\begin{aligned} \mathcal{B} &= \prod_{q=1}^{Q_a} \{\sigma_-^q,\ \sigma_+^q\} \\ &= \prod_{q=1}^{Q_a} \left\{ \inf_{w \in X^{\mathcal{N}}} \frac{a^q(w,w)}{\|w\|^2_{X^{\mathcal{N}}}},\ \sup_{w \in X^{\mathcal{N}}} \frac{a^q(w,w)}{\|w\|^2_{X^{\mathcal{N}}}} \right\}. \end{aligned} \tag{3.123}$$

We also introduce the two parameter set $\mathcal{S}$ and $\mathcal{P}$, that will be used to define the stability and positivity constraint, respectively:

$$\mathcal{S} = \{s_1 \in \mathcal{D}, \dots, s_k \in \mathcal{D}\}, \tag{3.124}$$

$$\mathcal{P} = \{p_1 \in \mathcal{D}, \dots, p_k \in \mathcal{D}\}. \tag{3.125}$$

Moreover, for any finite-dimensional *subset* of $\mathcal{D}$ ($= \mathcal{S}$ or $\mathcal{P}$), we denote with $\mathcal{S}^{M,\mu}$ (or $\mathcal{P}^{M,\mu}$) the set of M points closest to[1] μ in $\mathcal{S}$ (or $\mathcal{P}$).
If $M > |\mathcal{S}|$, (or $> |\mathcal{P}|$)[2], then $\mathcal{S}^{M,\mu} = |\mathcal{S}|$ (or $\mathcal{P}^{M,\mu} = \mathcal{P}$).

Lower and Upper bound

For given $\mathcal{S} \subset \mathcal{D}$, $M_\alpha \in \mathbb{N}$ (stability constraints), $M_+ \in \mathbb{N}$ (positivity constraints), we define the *lower bound* set as

$$\mathcal{Y}_{LB}(\mathcal{S};\mu) \equiv \left\{ y \in \mathcal{B} \mid \sum_{q=1}^{Q_a} \theta_a^q(\mu') y_q \geqslant \alpha^{\mathcal{N}}(\mu'),\ \forall \mu' \in \mathcal{S}^{M_\alpha,\mu};\ \sum_{q=1}^{Q_a} \theta_a^q(\mu') y_q \geqslant 0,\ \forall \mu' \in \mathcal{P}^{M_+,\mu} \right\} \tag{3.126}$$

Furthermore we define the *upper bound* set as

$$\mathcal{Y}_{UB}(\mathcal{S}) \equiv \{y^\star(\mu)(s_k),\ 1 \leqslant k \leqslant |\mathcal{S}|\} \tag{3.127}$$

for

$$y^\star(\mu) \equiv \arg\min_{y \in \mathcal{Y}} \mathscr{F}(y;\mu). \tag{3.128}$$

Finally we obtain the *coercivity lower* and *upper bound* as

$$\alpha_{LB}(\mathcal{S};\mu) = \min_{y \in \mathcal{Y}_{LB}(y;\mathcal{S})} \mathscr{F}(y;\mu), \tag{3.129}$$

$$\alpha_{UB}(\mathcal{S};\mu) = \min_{y \in \mathcal{Y}_{UB}(\mathcal{S})} \mathscr{F}(y;\mu). \tag{3.130}$$

It is possible to show that the *lower/upper bounds* provided above, *effectively bound* the coercivity constant, the subsequent result holds:

Proposition 3.4. *Given* $\mathcal{S}$, $\mathcal{P}$ *and* $M_\alpha \in \mathbb{N}$, $M_+ \in \mathbb{N}$

$$\alpha_{LB}(\mathcal{S};\mu) \leqslant \alpha^{\mathcal{N}}(\mu) \leqslant \alpha_{UB}(\mathcal{S};\mu) \qquad \forall \mu \in \mathcal{D} \tag{3.131}$$

The proof can be found on [HRSP07, RHP08].

We expect that if $\mathcal{S}$ is sufficiently large, then

1. $y^\star(\mu)$ will be sufficiently close to a member of $\mathcal{Y}_{UB}$ to provide a good *upper bound*;
2. the *stability* and *positivity* constraints in $\mathcal{Y}_{LB}$ will sufficiently restrict y to provide a good *lower bound*.

1 In the *Euclidean norm*
2 We recall that $|\cdot|$ denotes the *cardinality* of a finite set of elements

3.6.2 non-coercive case

We now address the generic non-coercive case, the *Stokes* case is a subset of this class of problems. Therefore, we will consider the global operator $\mathscr{A}$ of the Stokes problem, defined in equation 2.51 and the Babuska *inf-sup* stability constant β, see [Bab71].
We introduce the operators $T^q : X^{\mathcal{N}} \to X^{\mathcal{N}}$

$$(T^q w, v)_{X^{\mathcal{N}}} = \mathscr{A}^q(w, v) \qquad \forall v \in X^{\mathcal{N}},\ 1 \leqslant q \leqslant Q_a \tag{3.132}$$

and

$$T^{\mu} w = \sum_{q=1}^{Q} \theta^q(\boldsymbol{\mu}) \mathbb{T}^q(w) \tag{3.133}$$

It can be demonstrated, from equation 2.15, see [HRSP07], that:

$$\left(\beta^{\mathcal{N}}(\boldsymbol{\mu})\right)^2 = \inf_{w \in X^{\mathcal{N}}} \frac{(T^{\mu} w,\, T^{\mu} w)_{X^{\mathcal{N}}}}{\|w\|^2_{X^{\mathcal{N}}}} \tag{3.134}$$

which can be expanded, replacing the affine development 3.133 of the *supremizer* operator, as

$$\left(\beta^{\mathcal{N}}(\boldsymbol{\mu})\right)^2 = \inf_{w \in X^{\mathcal{N}}} \left[\sum_{q=1}^{Q} \sum_{q'=q}^{Q} \left(2 - \delta_{q\,q'}\right) \theta^q(\boldsymbol{\mu}) \theta^{q'}(\boldsymbol{\mu}) \frac{\left(\mathbb{T}^q w,\, \mathbb{T}^{q'} w\right)_{X^{\mathcal{N}}}}{\|w\|^2_{X^{\mathcal{N}}}} \right] \tag{3.135}$$

where $\delta_{q\,q'}$ is the Kronecker delta. We now identify:

$$\begin{aligned} \left(\beta^{\mathcal{N}}(\boldsymbol{\mu})\right)^2 &\mapsto \widehat{\alpha}^{\mathcal{N}} \\ \left(2 - \delta_{q\,q'}\right) \theta^q(\boldsymbol{\mu}) \theta^{q'}(\boldsymbol{\mu}),\ 1 \leqslant q \leqslant q' \leqslant \widehat{Q} &\mapsto \widehat{\theta}^q \\ \left(\mathbb{T}^q w,\, \mathbb{T}^{q'} v\right)_{X^{\mathcal{N}}},\ 1 \leqslant q \leqslant q' \leqslant \widehat{Q} &\mapsto \widehat{a}^q(w, v) \end{aligned} \tag{3.136}$$

where $\widehat{\theta}^q,\ 1 \leqslant q \leqslant \widehat{Q} \equiv Q(Q+1)/2$.
We then observe that the inf-sup constants can be rewritten as

$$\left(\beta^{\mathcal{N}}(\boldsymbol{\mu})\right)^2 \equiv \widehat{\alpha}^{\mathcal{N}}(\boldsymbol{\mu}) = \inf_{w \in X^{\mathcal{N}}} \sum_{q=1}^{\widehat{Q}} \widehat{\theta}^q(\boldsymbol{\mu}) \widehat{a}^q(w, v)\, \widehat{Q} \tag{3.137}$$

We may thus directly apply our *SCM* procedure to 3.137.

3.6.3 SCM algorithm

We now present the algorithm to exploit the evaluation of the *coercivity* (and/or *inf-sup*) constant.

The task of the *SCM* is, given a sample train $\Xi_{SCM} = \left\{\mu_{SCM}^1, \dots, \mu_{SCM}^{n_{SCM}}\right\}$ of dimension $|\Xi_{SCM}| = n_{SCM}$, to select Greedy parameters in Ξ_{SCM} and construct the sets $\mathcal{S}_k = \left\{s_1 = \mu_{SCM}^1 \cup \dots \cup s_{K_{max}} = \mu_{SCM}^{K_{max}}\right\}$. We now give the algorithm.
We define M_α, M_+, $\mathcal{P}$ and a tolerance $\epsilon_{SCM} \in]0,1[$, then we set $K_{\mathcal{S}} = 1$ and choose $\mathcal{S}_1 = \left\{s_1 = \mu_{SCM}^1\right\}$ *arbitrarily*, then

$$\begin{aligned}
&\texttt{while} \max_{\mu \in \Xi_{SCM}} \left[\frac{\alpha_{UB}(\mathcal{S};\mu) - \alpha_{LB}(\mathcal{S};\mu)}{\alpha_{UB}(\mathcal{S};\mu)}\right] > \epsilon_{SCM} \\
&\qquad s_{K+1} = \arg \max_{\mu \in \Xi_{SCM}} \left[\frac{\alpha_{UB}(\mathcal{S};\mu) - \alpha_{LB}(\mathcal{S};\mu)}{\alpha_{UB}(\mathcal{S};\mu)}\right] \\
&\qquad \mathcal{S}_{\mathcal{K}+\infty} = \mathcal{S}_{\mathcal{K}} \cup s_{K+1} \\
&\qquad K = K + 1 \\
&\texttt{end} \\
&K_{max} = K
\end{aligned} \tag{3.138}$$

Normally we set $\epsilon_{SCM} \approx 0.75$ which is a crude lower bound but with a little effect on our error bounds, [HRSP07].

3.6.4 Offline–Online procedure

We note that to compute the arg max we must solve a *linear optimization problem* or *Linear Program* (*LP*), for the lower bound $\alpha_{LB}(\mu)$, 3.130.
In the coercive case, the lower bound *LP*'s contains:

- design variables
  1. Q_a variables, $y = \{y_1, \dots, y_{Q_a}\}$;
- constraints
  1. $2Q_a$ bounding boxes for $y \in \mathcal{B}$;
  2. M_α stability;
  3. M_+ positivity.

It is clear that the operation count for the *Online* stage $\mu \to \alpha_{LB}(\mu)$ is *independent* of $\mathcal{N}$.
Nonetheless we first must determine our set $\mathcal{S}$ and obtain the $\alpha^{\mathcal{N}}(s_k)$, $1 \leqslant k \leqslant |\mathcal{S}|\ (\equiv K_{\mathcal{S}})$, by an Offline *Greedy SCM* algorithm.

Offline

In the *Offline* stage, we have to construct the set $\mathcal{B}$ (once) and then:

1. evaluate $\alpha^{\mathcal{N}}(s_k)$;
2. evaluate $y^\star(s_k)$;

3. form y_{LB};

4. perform a lower bound *LP*'s to evaluate $\alpha_{LB}(s_k)$.

The first three quantities of course depends on $\mathcal{N}$, nonetheless it is important to *remark* that there are no *cross terms* $\mathcal{O}(n_{SCM} \times \mathcal{N})$.

Online

In the *Online* stage, given a new value μ we have to perform a lower bound *LP*'s (*LP*) to evaluate $\alpha_{LB}(\mu)$. This Online stage is hence independent on $\mathcal{N}$.
In the table 3 we summarize the computational cost to evaluate the *Offline/Online* stage of the SCM:

PART	ITEM	COMPLEXITY	EQUATION
Offline	$\mathcal{B}$	$2Q_a$-eigenproblems over $X^{\mathcal{N}}$	3.123
	$\alpha^{\mathcal{N}}(s_k)$	K_{max}-eigenproblems over $X^{\mathcal{N}}$	3.126
	$y^{\star}(s_k)$	$K_{max}\, Q_a$-inner product	3.128
	y_{LB}	$\mathcal{N}\, Q_a\, K_{max}$	3.126
	$\alpha_{LB}(s_k)$	$n_{SCM}\, K_{max}$ *LP*'s of "size" $\mathcal{O}(2Q_a + M_\alpha + M_+)$	3.130
Online	$\alpha_{LB}(\mu)$	1 *LP*'s of "size" $\mathcal{O}(2Q_a + M_\alpha + M_+)$	3.39

Table 3: Offline/Online: SCM

3.7 CHOICE OF TRUTH APPROXIMATION

It would be preferable to build the *RB* approach directly upon the exact solution, but this is not in general possible. As indicated earlier, the RB approximation shall be built upon and reduced basis error will be measured relative to a "truth" Galerkin *FE* approximation. Therefore it is necessary to choose *properly* the underlying discretization.

3.7.1 Choice of $\mathcal{N}$

In order to obtain a satisfying reduced basis model able to describe in an accurate way the exact behavior of the physical process, it is necessary to chose the discretization properly; that is in order to minimize the underlying error between *exact* solution and the *truth approximation*.
Let $u^{RB}(\boldsymbol{\mu})$ be the *RB* solution of the problem and $u^{\mathcal{N}}(\boldsymbol{\mu})$ the finite element solution, than the error is the sum of (at least) two terms:

$$\|u^e(\boldsymbol{\mu}) - u^{RB}(\boldsymbol{\mu})\| = \underbrace{\|u^e(\boldsymbol{\mu}) - u^{\mathcal{N}}(\boldsymbol{\mu})\|}_{\text{neglected}} + \underbrace{\|u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|}_{\text{considered}}. \tag{3.139}$$

The minimization of the second addendum is a task delagated to the reduced basis method, on the contrary the minimization of the first is not related to the method.
Because of this, it is necessary to provide a feasible "starting point". This can be achieved thanks to the choice of a discretization method able to describe correctly the problem.
We shall require that our family of truth subspaces $X^{\mathcal{N}}$ satisfies the approximation condition:

$$\max_{\boldsymbol{\mu}\in\mathcal{D}} \inf_{w\in X^{\mathcal{N}}} \|u(\boldsymbol{\mu}) - w\|_{X^e} \to 0 \quad \text{as} \quad \mathcal{M} \to \infty. \tag{3.140}$$

The choice of a *finite element* approximation automatically fulfill this requirement because the method is *strongly consistent*, [QV97, Qua09]; thus for sufficiently large $\mathcal{N}$, it is possible to approximate $u^e(\boldsymbol{\mu})$ and $s^e(\boldsymbol{\mu})$ arbitrarily closely.
In particular we define the difference $\epsilon^{\mathcal{N}}$ between the exact solution and the approximation as:

$$\epsilon^{\mathcal{N}} = \max_{\boldsymbol{\mu}\in\mathcal{D}} \|u(\boldsymbol{\mu}) - u^{\mathcal{N}}(\boldsymbol{\mu})\|_{X^e} \overset{\mathcal{N}\to\infty}{\longrightarrow} 0. \tag{3.141}$$

In general, $\mathcal{N}$ must be chosen rather large to achieve a reasonable engineering accuracy $\epsilon^{\mathcal{N}}$.
In 3D problems the complexity is higher since there is greater variability of the solution field as the parameters changes. Therefore it is necessary to discretize the problem so that for any possible combination of the parameters the accuracy is kept under a safe tolerance. In fact it is worth to

recall that the *RB Offline* representation has to be built over a *unique truth representation* for all $\mu \in \mathcal{D}$; the truth approximation is *"frozen"* in the *RB* methodology.

3.7.1.1 *Mesh saturation*

The choice of the optimum (or at least of a suitable) *truth* solution is *not trivial*, nevertheless it is possible to verify *a-posteriori* if the discretization is enough rich to seize all the geometrical and physical complexity of the problem.
This can be achieved verifying the so called *grid saturation*. We will not go in deep into this issue, nevertheless it can be shown a glimpse just to give an idea of the issue, considered in [RHP08] and [PR09].
An example taken from [RHP08] is shown in Figure 7, dealing with a *heat conduction* problem. The figure shows the *"convergence"* of the *RB* procedure as function of the steps N, for various underlying truth approximation.
It can be seen that as $\mathcal{N}$ increases, the convergence of the method is not influenced by $\mathcal{N}$ (saturation effect).

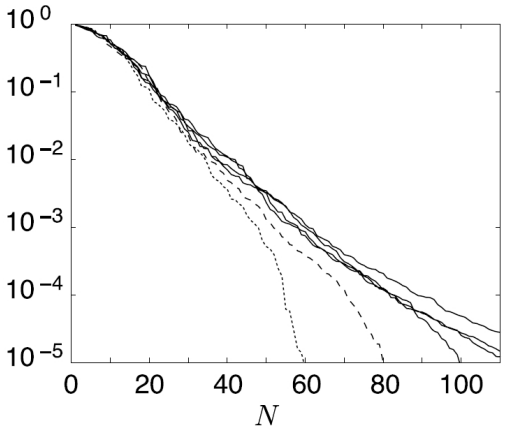


Figure 7: 2D case mesh saturation
$\mathcal{N} = 137$ (dotted), $\mathcal{N} = 453$ (dashed), and $\mathcal{N} = 661, 1737, 2545, 6808$ (all quite similar)

Therefore can be stated that if the *truth* is too poor, the *RB* method converges anyway (the *y-axis* shows the error bound as N increases), but the convergence is limited to a *subspace* of possible solution of the parametric problem.
Beyond, if $\mathcal{N}$ is enough rich, all the complexity has been captured by the discretization, therefore there is no need to go further. This is the *heuristic idea* of *mesh saturation*.

4 GEOMETRY

In this chapter, we will explain how to deal with a domain which is parameter dependent.
The *RB* method described in Chapter 3 requires that Ω is parameter independent: if we wish to consider linear combinations of snapshots, these snapshots must be defined relative to a common spatial configuration (domain) .
Then to permit geometric variations, we must interpret Ω, our parameter independent domain, as the pre-image of Ω_o, the original (actual, deformed) parameter dependent domain, see [RHP08].
The geometric transformation will yield *variable* (parameter-dependent) *coefficients* in the reference-domain linear and bilinear forms that, under suitable hypotheses to be discussed below, will take at the end the requisite affine form (2.20).

4.1 AFFINE PARAMETRIC PRECONDITION

We now introduce a domain decomposition:

$$\Omega_o(\boldsymbol{\mu}) = \bigcup_{k=1}^{K_{dom}} \Omega_o^k(\boldsymbol{\mu}) \tag{4.1}$$

where the $\Omega_o^k(\boldsymbol{\mu})$ are mutually non overlapping subdomains, that is for any $\boldsymbol{\mu} \in \mathcal{D}$

$$\Omega_o^k(\boldsymbol{\mu}) \cap \Omega_o^{k'}(\boldsymbol{\mu}) = 0 \quad 1 \leqslant k, k' \leqslant K_{dom},\ k \neq k'.$$

This coarse domain decomposition will be denoted *RB* discretization.
We now choose a parameter of reference $\boldsymbol{\mu}_{ref} \in \mathcal{D}$ and define our reference domain as $\Omega_r \equiv \Omega(\boldsymbol{\mu}_{ref})$.
We will never omit the underscript beside the domain Ω we are dealing with, to avoid any confusion between the parameter dependent *original domain* $\Omega_o(\boldsymbol{\mu})$ (sometimes for brevity, just Ω_o) and the parameter independent reference domain Ω_r.
We will build our *FE* approximation on a very fine *FE* subtriangulation of the coarse *RB* decomposition.
This *FE* subtriangulation ensures that the *FE* approximation accurately treats the perhaps discontinuous coefficients (that could arise from property and geometry variation) associated with the different subdomains.
The subtriangulation also plays an *important role* in the generation of the

affine representation.
The choice of $\boldsymbol{\mu}_{\text{ref}}$ has to be done in an optimal way to reduce both Offline and Online computational effort.
Tipically the reference domain shall be built choosing a $\boldsymbol{\mu}_{\text{ref}}$ at the *"center"* of our parameter domain $\mathcal{D}$ in order to *minimize* the *distorsion* and consequently reduce the requisite $\mathcal{N}$.

We now state our *Affine Geometry Precondition*. We can treat any *original* domain $\Omega_o(\boldsymbol{\mu})$, that *admits* a *domain decomposition* 4.1, for which, $\forall\, \boldsymbol{\mu} \in \mathcal{D}$

$$\Omega_r = \mathcal{T}^{\text{aff},k}(\Omega o(\boldsymbol{\mu})^k; \boldsymbol{\mu}) \tag{4.2}$$

for *affine mappings* $\mathcal{T}^{\text{aff},k}(\,\cdot\,;\boldsymbol{\mu}) : \Omega_o(\boldsymbol{\mu}) \to \Omega_r,\ 1 \leqslant k \leqslant K_{\text{dom}}$, that satisfy two requisites:

1. *individually bijective*
2. *collectively continuos* (interface condition), that is, given two different subdomains denoted with k and k', $\forall\, \mathbf{x}_o \in \Omega_o^k(\boldsymbol{\mu}) \cap \Omega_o^{k'}$, holds the following condition

$$\mathcal{T}^{\text{aff},k}(\mathbf{x_o}; \boldsymbol{\mu}) = \mathcal{T}^{\text{aff},k'}(\mathbf{x_o}; \boldsymbol{\mu}) \tag{4.3}$$

We have depicted the idea of the affine transformation in Figure 8. Of

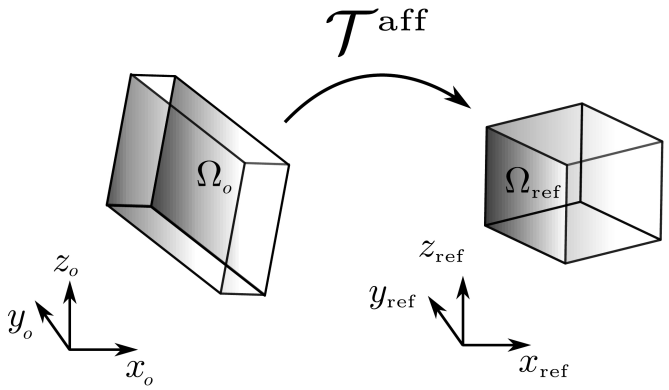


Figure 8: A 3D affine transformation

course, thanks to the requested bijective property, we can replace this definition with the forward version taking the inverse of $\mathcal{T}^{\text{aff}}$ committing any crime.
The *Affine Geometry Precondition* is a *necessary* condition for affine parameter dependence as defined in 2.20 (page 14).
Note that we purposely define K_{dom} with respect to the *exact* problem, rather than the *FE* approximation: K_{dom} is not depending on $\mathcal{N}$.
We now give a more explicit representation of the *affine transformation* $\mathcal{T}^{\text{aff}}$, to better understand how this geometry precondition will be exploited.

We state that, for $1 \leqslant k \leqslant K_{dom}$, for any $\boldsymbol{\mu} \in \mathcal{D}$ and for all $\mathbf{x}_o \in \Omega_o^k(\boldsymbol{\mu})$, the affine transformation is defined as follows:

$$\mathbf{x}_r = \mathcal{T}_i^{aff,k}(\mathbf{x}_o;\boldsymbol{\mu}) = C_i^{aff,k}(\boldsymbol{\mu}) + \sum_{j=1}^{d} G_{ij}^{aff,k}(\boldsymbol{\mu})\,\mathbf{x}_o, \quad 1 \leqslant i \leqslant d \tag{4.4}$$

for given $C^{aff,k}(\boldsymbol{\mu}) : \mathcal{D} \to \mathbb{R}^d$ and $G^{aff,k}(\boldsymbol{\mu}) : \mathcal{D} \to \mathbb{R}^{d\times d}$, that are called the *affine mapping coefficients*; we recall that d is the spatial dimension of the problem, hence in our case $d = 3$.
The affine transformation is thus the superposition of a translation $C^{aff}(\boldsymbol{\mu})$, that do not modify the shape of the domain, and a deformation $G^{aff}(\boldsymbol{\mu})$ that can be a dilation/contraction or a shear. It is worth to remark that, in this work, the transformation *must* depends only upon the parameter $\boldsymbol{\mu}$.

A more general transformation, that involves a spatial coordinates dependence, is not considered in the framework of this thesis. This kind of transformation called *"nonaffine"* has been recently adopted in the context of the *RB* methodology, for example in [MYNA04, Roz08]. The *non-affine* representation of the geometry arises from the so-called *free form deformation* techniques, which are very well suited, for example, for shape optimizations of complex geometries, see [MQR10, LR10] and [RM10].
The basic idea is that, thanks to an highly specialized technique, the so-called *"empirical interpolation"* (*EIM*), it is possible to approximate a *non-affine* tranformation thanks to a *superposition* of different *affine* transformations.
Nonetheless this results has been established in a 2D context, the extension to the 3D case in still under investigation.

We can now define the associated *Jacobians*:

$$J^{aff,k}(\boldsymbol{\mu}) = \left|\det\left(G^{aff,k}(\boldsymbol{\mu})\right)\right|, \quad 1 \leqslant k \leqslant K_{dom}, \tag{4.5}$$

which are constants in space over each subdomain. We further define, for any $\boldsymbol{\mu} \in \mathcal{D}$

$$D^{aff,k}(\boldsymbol{\mu}) = G^{aff,k}(\boldsymbol{\mu}), \tag{4.6}$$

this matrix shall prove convenient in subsequent *derivative* transformations, as we will see in section 4.3.
We may interpret our *local mappings* in terms of a *global transformation*. In particular, for any $\boldsymbol{\mu} \in \mathcal{D}$, the local mapping 4.2 induce a global *bijective piecewise-affine* mapping $\mathcal{T}^{aff} : \Omega_o(\boldsymbol{\mu}) \to \Omega_r$, such that:

$$\mathcal{T}^{aff}(\mathbf{x}_o;\boldsymbol{\mu}) = \mathcal{T}^{aff,k}(\mathbf{x}_o;\boldsymbol{\mu}), \qquad k = \min_{k'\in\{1,\dots,K_{dom}\}\,|\,\mathbf{x}\in\Omega_o^{k'}(\boldsymbol{\mu})} \tag{4.7}$$

note the one-to-one property of this mapping (and, hence the arbitrariness of our *min* choice in 4.7) is ensured by the interface condition 4.3.

In the following section the creation of an affine mappings will be discussed, subsequently the treatment of the parametric geometry dependence will be exploited thanks to an operative example. A parallel study has been carried out in [Gel10].

4.2 AFFINE MAPPINGS CREATION

For simplicity we now consider a single subdomain, nonetheless the extension to the multi-subdomains case is readily obtainable as we will see. As we consider a single subdomain in this section, we shall suppress the subdomain superscript for clarity of exposition. The procedure, in the 2D case, is explained in [RHP08], in this work it has been extended to the more general 3D case.
In the 3D case ($d = 3$, see equation 4.4) the affine mapping coefficients are $C^{aff}(\boldsymbol{\mu}) \in \mathbb{R}^3$ and $G^{aff}(\boldsymbol{\mu}) \in \mathbb{R}^{3\times 3}$, that is we have $3 + 9 = 12$ unknows to find in order to entirely define the affine transformation.
Under our assumption that the mapping is invertible, we know that the Jacobian J^{aff} of 4.5 is *strictly positive* and that the *derivative transformation matrix* D^{aff} of 4.6 is well defined.

Then the mapping coefficient can be identified by the relationship between 4 *non-planar parametrized image points* $\in \Omega_o(\boldsymbol{\mu})$, denoted with $\underline{\mathbf{z}}_o(\boldsymbol{\mu})$ and the corresponding 4 pre-image points $\in \Omega_r$, denoted with $\underline{\mathbf{z}}_r$ [1]:

$$\begin{Bmatrix} \underline{\mathbf{z}}_o^1(\boldsymbol{\mu}) \\ \underline{\mathbf{z}}_o^2(\boldsymbol{\mu}) \\ \underline{\mathbf{z}}_o^3(\boldsymbol{\mu}) \\ \underline{\mathbf{z}}_o^4(\boldsymbol{\mu}) \end{Bmatrix} = \begin{Bmatrix} \{z_{o_1}^1, z_{o_2}^1, z_{o_3}^1\} \\ \{z_{o_1}^2, z_{o_2}^2, z_{o_3}^2\} \\ \{z_{o_1}^3, z_{o_2}^3, z_{o_3}^3\} \\ \{z_{o_1}^4, z_{o_2}^4, z_{o_3}^4\} \end{Bmatrix},$$

$$\begin{Bmatrix} \underline{\mathbf{z}}_r^1 \\ \underline{\mathbf{z}}_r^2 \\ \underline{\mathbf{z}}_r^3 \\ \underline{\mathbf{z}}_r^4 \end{Bmatrix} = \begin{Bmatrix} \{z_{r_1}^1, z_{r_2}^1, z_{r_3}^1\} \\ \{z_{r_1}^2, z_{r_2}^2, z_{r_3}^2\} \\ \{z_{r_1}^3, z_{r_2}^3, z_{r_3}^3\} \\ \{z_{r_1}^4, z_{r_2}^4, z_{r_3}^4\} \end{Bmatrix},$$

In particular, for given $\boldsymbol{\mu} \in \mathcal{D}$, the application of 4.4 to the selected nodes yields to:

$$z_{r_i}^m = C_i^{aff} + \sum_{j=1}^{3} G_{ij}^m z_{o_j}^m, \quad 1 \leqslant i \leqslant 3, \quad 1 \leqslant m \leqslant 4, \tag{4.8}$$

The 4.8 provides a system made of 12 equations, by which to determine the 12 *mapping coefficients*. If we choose at least two coplanar points, than the system is singular.

1 Here we denote with the superscript one of the 4 point considered, whereas the subscript indicates one of the 3 the components (x, y, z) of the spatial coordinates.

To be more explicit, we provide a matricial representation of equation 4.8:

$$\underline{\underline{\mathbb{B}}}^{\text{aff}}\underline{\mathbf{c}}^{\text{aff}} = \underline{\mathbf{v}}^{\text{aff}} \tag{4.9}$$

Where the matrix $\mathbb{B} \in \mathbb{R}^{12\times12}$ summarizes the coefficients of the linear system, that depends upon the coordinates of the *"original"* points:

$$\underline{\underline{\mathbb{B}}}^{\text{aff}} = \begin{bmatrix} \begin{bmatrix} & \underline{\underline{I}}^{3\times3} & \end{bmatrix} & \begin{bmatrix} \underline{\mathbf{z}}_o^1 & \underline{0} & \underline{0} \\ \underline{0} & \underline{\mathbf{z}}_o^1 & \underline{0} \\ \underline{0} & \underline{0} & \underline{\mathbf{z}}_o^1 \end{bmatrix} \\ \begin{bmatrix} & \underline{\underline{I}}^{3\times3} & \end{bmatrix} & \begin{bmatrix} \underline{\mathbf{z}}_o^2 & \underline{0} & \underline{0} \\ \underline{0} & \underline{\mathbf{z}}_o^2 & \underline{0} \\ \underline{0} & \underline{0} & \underline{\mathbf{z}}_o^2 \end{bmatrix} \\ \begin{bmatrix} & \underline{\underline{I}}^{3\times3} & \end{bmatrix} & \begin{bmatrix} \underline{\mathbf{z}}_o^1 & \underline{0} & \underline{0} \\ \underline{0} & \underline{\mathbf{z}}_o^2 & \underline{0} \\ \underline{0} & \underline{0} & \underline{\mathbf{z}}_o^3 \end{bmatrix} \end{bmatrix} \tag{4.10}$$

moreover $\underline{\mathbf{c}}^{\text{aff}} \in \mathbb{R}^{12\times1}$ is the array of unknows (i.e. the *mapping coefficients*) sorted as shown in 4.11; finally $\underline{\mathbf{v}}^{\text{aff}} \in \mathbb{R}^{12\times1}$ is the array of known terms, that depends upon the coordinates of the *reference* points:

$$\underline{\mathbf{c}}^{\text{aff}} = \begin{Bmatrix} C_1^{\text{aff}} \\ C_2^{\text{aff}} \\ C_3^{\text{aff}} \\ G_{11}^{\text{aff}} \\ G_{12}^{\text{aff}} \\ G_{13}^{\text{aff}} \\ G_{21}^{\text{aff}} \\ G_{22}^{\text{aff}} \\ G_{23}^{\text{aff}} \\ G_{31}^{\text{aff}} \\ G_{32}^{\text{aff}} \\ G_{33}^{\text{aff}} \end{Bmatrix} \quad \underline{\mathbf{v}}^{\text{aff}} = \begin{Bmatrix} \underline{\mathbf{z}}_r^1 \\ \underline{\mathbf{z}}_r^2 \\ \underline{\mathbf{z}}_r^3 \\ \underline{\mathbf{z}}_r^4 \end{Bmatrix} \tag{4.11}$$

The mapping coefficients can be easily found solving the linear system 4.9 as follow:

$$\underline{\mathbf{c}}^{\text{aff}} = \underline{\underline{\mathbb{B}}}^{\text{aff}^{-1}}\underline{\mathbf{v}}^{\text{aff}}. \tag{4.12}$$

The solution of the system requires 12^3 operation, negligible if compared to the previously discussed basis assembling cost.

4.2.1 Single domain mapping

We now use an example to illustrare the procedure. We will use as test case the transformation depicted in figure 9.

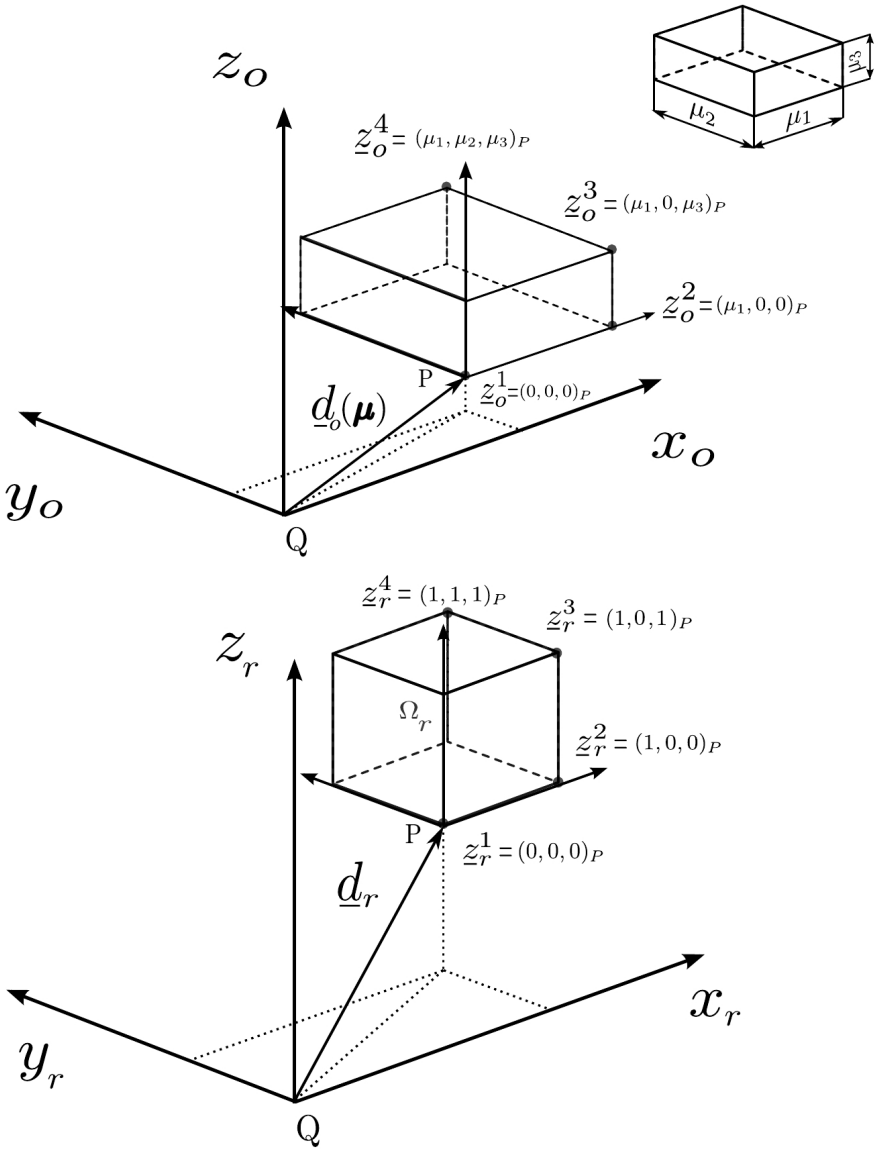


Figure 9: Affine transformation construction

With regard to the figure 9, we choose as geometrical parameters $\boldsymbol{\mu} = \{\mu_2, \mu_3, \mu_1\} = \{2, 3, 4\}$, in addition to simplify we choose $\underline{d}_r = \underline{d}_o(\boldsymbol{\mu})$, hence we can use a local system attached to the first node ($\underline{z}^1_*$).
Now, exploiting the procedure showed in section 4.2, we can build the system 4.9 by which obtain the mapping coefficients.

$$\begin{Bmatrix} C^{aff}_1 \\ C^{aff}_2 \\ C^{aff}_3 \\ G^{aff}_{11} \\ G^{aff}_{12} \\ G^{aff}_{13} \\ G^{aff}_{21} \\ G^{aff}_{22} \\ G^{aff}_{23} \\ G^{aff}_{31} \\ G^{aff}_{32} \\ G^{aff}_{33} \end{Bmatrix} = \left(\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & \mu_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \mu_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_1 & 0 & 0 \\ 1 & 0 & 0 & \mu_1 & 0 & \mu_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \mu_1 & 0 & \mu_3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_1 & 0 & \mu_3 \\ 1 & 0 & 0 & \mu_1 & \mu_2 & \mu_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \mu_1 & \mu_2 & \mu_3 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_1 & \mu_2 & \mu_3 \end{bmatrix} \right)^{-1} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{Bmatrix}. \tag{4.13}$$

Then solving the system 4.13 we obtain:

$$\underline{C}^{aff}(\boldsymbol{\mu}) = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \quad \underline{\underline{G}}^{aff}(\boldsymbol{\mu}) = \begin{bmatrix} \frac{1}{\mu_1} & 0 & 0 \\ 0 & \frac{1}{\mu_2} & 0 \\ 0 & 0 & \frac{1}{\mu_3} \end{bmatrix}. \tag{4.14}$$

The Jacobian of the transformation (4.5) is $J^{aff}(\boldsymbol{\mu}) = \frac{1}{\mu_1\mu_2\mu_3} = \frac{1}{12}$.
We remark that the Jacobian of a transformation can be seen as the ratio between the final and initial volumes on which the deformation takes place.
In order to verify the affine transformation we apply the transformation to each nodes of the *original domain* and, if correct, the *affine mapping* must trace back the corresponding nodes on the *reference domain*:

$$\boxed{\mathbf{z}^1}\ \underline{\underline{G}}^{aff} \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \checkmark \qquad \boxed{\mathbf{z}^2}\ \underline{\underline{G}}^{aff} \begin{Bmatrix} \mu_1 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} \checkmark$$
$$\boxed{\mathbf{z}^3}\ \underline{\underline{G}}^{aff} \begin{Bmatrix} \mu_1 \\ 0 \\ \mu_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix} \checkmark \qquad \boxed{\mathbf{z}^4}\ \underline{\underline{G}}^{aff} \begin{Bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix} \checkmark \tag{4.15}$$

We remark that in this simple test case the transformation is *"diagonal"* because we are deforming the domain by mean of a simple *dilation*. In the case of a shear deformation for example, we would have even the *extradiagonal* terms, [RHP08].

4.2.2 Global affine mappings

Exploited the case of a single domain, we need to extend the procedure to a multi-subdomain case. We will make use of an example as in Figure 10. In figure 10 the test case is sketched. We have two adjacent subdomains

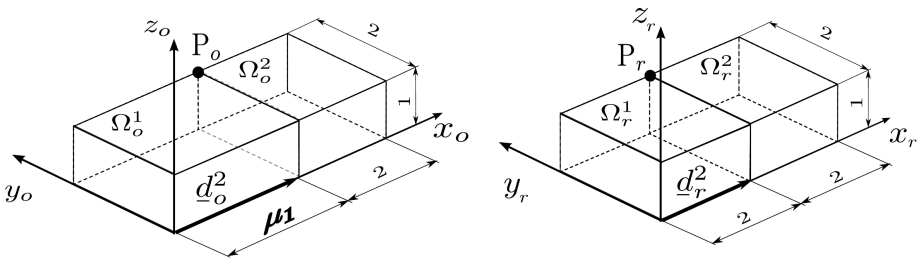


Figure 10: Global affine mappings

denoted with Ω_*^1 and Ω_*^2. The first domain can be deformed along the x-axes thanks to the parameter denoted with μ_1, all the other dimensions are held fixed.
The procedure is quite similar, we just need in addition to satisfy the *global continuity condition* 4.3. A way to satisfy it, is to use a *unique system of reference* for the different subdomains.
In this way the procedure described in section 4.2 will *implicitly* provide the suitable translation $\underline{C}^{aff}(\boldsymbol{\mu})$ to satisfy the interface condition.
Using the procedure for the two domains we obtain the following results:

$$\begin{aligned} \underline{C}_1^{aff}(\boldsymbol{\mu}) &= \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right\} \qquad \underline{\underline{G}}_1^{aff}(\boldsymbol{\mu}) = \begin{bmatrix} \frac{1}{\mu_1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \underline{C}_2^{aff}(\boldsymbol{\mu}) &= \left\{ \begin{array}{c} 2-\mu_1 \\ 0 \\ 0 \end{array} \right\} \qquad \underline{\underline{G}}_2^{aff}(\boldsymbol{\mu}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4.16}$$

We now take two *adjacent nodes* (denoted with P_o in the figure 10) on the *original domains*; if the global mappings satisfies the *interface condition*, applying the two different *affine transformation* at the node, we will obtain again two *adjacent nodes* (P_r) on the *reference domain*:

$$\mathcal{T}_1^{aff}(P_o(\boldsymbol{\mu}), \boldsymbol{\mu}) = \underline{C}_1^{aff} + \underline{\underline{G}}_1^{aff} \{\mu_1, 2, 1\}^T = \{2, 2, 1\} = P_r \quad ,\checkmark$$

$$\mathcal{T}_2^{aff}(P_o(\boldsymbol{\mu}), \boldsymbol{\mu}) = \underline{C}_2^{aff} + \underline{\underline{G}}_2^{aff} \{\mu_1, 2, 1\}^T = \{2, 2, 1\} = P_r \quad .\checkmark$$

The node P_o is identically projected into P_r thanks to the two different *affine transformations*, hence the *continuity* of the *global mapping* is satisfied.

4.3 LINEAR AND BILINEAR FORMS

In this section, we will focus on the transformations that we have to operate on the *weak forms* that arise from our system of partial differential equations, if our domain $\Omega_o(\boldsymbol{\mu})$ allows the *affine geometry precondition* described in the previous section.
We will use a simple *scalar coercive* 3D problem to show how to exploit the *geometry parametric dependence* of the domain; the Stokes case admits a straightforward extension.
The procedure is discussed in the 2D case, in [RHP08, Roz08].

4.3.1 Formulation on the original domain

Our problem is initially posed on the original domain $\Omega_o(\boldsymbol{\mu})$. We shall assume for simplicity that $X_0^e(\boldsymbol{\mu}) = H_0^1(\Omega_0(\boldsymbol{\mu}))$, which corresponds to homogeneous Dirichlet boundary conditions over the entire boundary $\partial\Omega_o(\boldsymbol{\mu})$. Given $\boldsymbol{\mu} \in \mathcal{D}$, find $u_o^e(\boldsymbol{\mu}) \in X_0^e(\boldsymbol{\mu})$ such that:

$$a_o(u_o^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f_o(v; \boldsymbol{\mu}), \quad \forall v \in X_0^e(\boldsymbol{\mu}) \tag{4.17}$$

then evaluate

$$s_o^e(\boldsymbol{\mu}) = f_o(u_o^e(\boldsymbol{\mu})). \tag{4.18}$$

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 (eq. 2.20, pag. 14).
We require that $a_o(\cdot, \cdot; \boldsymbol{\mu}) : H_0^1(\Omega_o(\boldsymbol{\mu})) \times H_0^1(\Omega_o(\boldsymbol{\mu})) \to \mathbb{R}$ can be expressed as:

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

$$= \sum_{k=1}^{K_{dom}} \int_{\Omega_o^k(\boldsymbol{\mu})} \left\{ \begin{array}{cccc} \dfrac{\partial w}{\partial x_{o_1}} & \dfrac{\partial w}{\partial x_{o_2}} & \dfrac{\partial w}{\partial x_{o_3}} & w \end{array} \right\} \mathcal{K}_{o_{ij}}^k(\boldsymbol{\mu}) \left\{ \begin{array}{c} \dfrac{\partial w}{\partial x_{o_1}} \\ \dfrac{\partial w}{\partial x_{o_2}} \\ \dfrac{\partial w}{\partial x_{o_3}} \\ w \end{array} \right\} d\Omega_o^k \tag{4.19}$$

where $\mathbf{x}_o = \{x_{o_1}, x_{o_2}, x_{o_3}\}$ denotes a point in $\Omega_o(\boldsymbol{\mu})$; and where for $1 \leqslant k \leqslant K_{dom}$, $\underline{\underline{\mathcal{K}}}_o^k : \mathcal{D} \to \mathbb{R}^{4\times4}$ is a given *symmetric positive definite* matrix (which in turn *ensures coercivity* of our bilinear form):

$$\underline{\underline{\mathcal{K}}}_o^k = \begin{bmatrix} \begin{bmatrix} & \mathbb{R}^{3\times3} & \\ & & \\ \mathbb{R}^{1\times3} & & \end{bmatrix} & \begin{bmatrix} \mathbb{R}^{3\times1} \\ \\ \mathbb{R}^{1\times1} \end{bmatrix} \end{bmatrix} \tag{4.20}$$

the upper 3×3 principal submatrix of $\underline{\underline{\mathcal{K}}}_o^k$ represent the usual *diffusity/conductivity tensor*, the element $(4,4)$ represent the identity tensor (e. g. *mass matrix* or a *reaction term*), finally the elements $\underline{\underline{\mathcal{K}}}_{o_{1:3,3}}^k$ and $\underline{\underline{\mathcal{K}}}_{o_{3,1:3}}^k$, that we set to zero because we are dealing with symmetric operators, represent the *first derivative* operators (e. g. *convective terms*).
Similarly we require that $f_o : H_o^1(\Omega_o(\mu)) \to \mathbb{R}$ can be expressed as

$$f_o(v;\mu) = \sum_{k=1}^{K_{dom}} \int_{\Omega_o^k(\mu)} \underline{\mathcal{F}}_o^k v \, d\Omega_o^k. \tag{4.21}$$

In this case we have assumed that the linear functional is only due to *volume force*, a similar treatment is possible in the case of *Dirichlet non-homogeneus* condition and/or *non-homogeneus Neumann* condition. In the chapters devoted to the mathematical modelling, we will cope with these different boundary conditions.

4.3.2 Formulation on reference domain

We now apply standard techniques to transform the problem over the original domain to an equivalent problem over the reference domain.
Given $\mu \in \mathcal{D}$, find $u^e(\mu) \in X^e \equiv H_0^1(\Omega)$ such that:

$$a(u^e, v; \mu) = f(v;\mu) \quad \forall v \in X^e \tag{4.22}$$

then evaluate

$$s^e(\mu) = f(u^e(\mu)). \tag{4.23}$$

We may then identify the relations between the *output* and the *solution field*, in the original and in the reference domain:

$$\begin{aligned} s^e(\mu) &= s_o^e(\mu) \\ u^e(\mu) &= u_o^e(\mu) \circ \mathcal{T}^{aff}(\,\cdot\,;\mu). \end{aligned} \tag{4.24}$$

The transformed bilinear form a, can be expressed as:

$$a(w, v; \mu) = \tag{4.25}$$

$$\int_{\Omega_r^k} \left\{ \begin{array}{cccc} \dfrac{\partial w}{\partial x_{r_1}} & \dfrac{\partial w}{\partial x_{r_2}} & \dfrac{\partial w}{\partial x_{r_3}} & w \end{array} \right\} \mathcal{K}_{ij}^k(\mu) \left\{ \begin{array}{c} \dfrac{\partial w}{\partial x_{r_1}} \\ \dfrac{\partial w}{\partial x_{r_2}} \\ \dfrac{\partial w}{\partial x_{r_3}} \\ w \end{array} \right\} d\Omega_r^k \tag{4.26}$$

where $\mathbf{x}_r = \{x_{r_1}, x_{r_2}, x_{r_3}\}$ denotes a point in Ω_r and where $\mathcal{K}\|_r : \mathcal{D} \to \mathbb{R}^{4\times 4}$, $1 \leqslant k \leqslant K_{dom}$ are symmetric positive definite matrices.
To obtain this matrices we first need to find the relation between the

derivative operator written in the original domain and the corresponding operator written in the original domain. In particular, we have that

$$\frac{\partial \cdot}{\partial x_{o_i}} = \frac{\partial x_{r_j}}{\partial x_{o_i}} \frac{\partial \cdot}{\partial x_{r_j}} = G_{ij}^{aff}(\boldsymbol{\mu}) \frac{\partial \cdot}{\partial x_{o_i}} = \underline{\underline{D}}^{aff}(\boldsymbol{\mu}) \frac{\partial \cdot}{\partial x_{o_i}} \tag{4.27}$$

The definition (4.6) of the *derivatives operator* $\underline{\underline{D}}^{aff,k}$ is now clear; the matrices $\underline{\underline{G}}^{aff,k}$ automatically provide the relation between the derivatives operator in the original and in the reference domain.
Moreover, since we are acting a change of variable $\mathbf{x}_o \to \mathbf{x}_r$ in the integral (4.19), recalling the equation 4.5, we get

$$d\Omega_o(\boldsymbol{\mu}) = \det(\underline{\underline{G}}^{aff}(\boldsymbol{\mu})^{-1}) d\Omega_r = \left(J^{aff,k}(\boldsymbol{\mu})\right)^{-1}. \tag{4.28}$$

It follows that, considering the equations 4.27-4.28, the relation between $\underline{\underline{\mathcal{K}}}_o^k(\boldsymbol{\mu})$ and $\underline{\underline{\mathcal{K}}}_r(\boldsymbol{\mu})$ can be written as

$$\underline{\underline{\mathcal{K}}}_r^k(\boldsymbol{\mu}) = \left(\underline{\underline{\mathcal{G}}}^k(\boldsymbol{\mu})\right)^T \underline{\underline{\mathcal{K}}}_o^k(\boldsymbol{\mu}) \, \underline{\underline{\mathcal{G}}}^k(\boldsymbol{\mu}) \, \left(J^{aff,k}(\boldsymbol{\mu})\right)^{-1} \quad 1 \leqslant k \leqslant K_{dom} \tag{4.29}$$

where we have defined $\underline{\underline{\mathcal{G}}}^k(\boldsymbol{\mu}) : \mathcal{D} \to \mathbb{R}^{4\times 4}$, $1 \leqslant k \leqslant K_{dom}$ as

$$\underline{\underline{\mathcal{G}}}^k(\boldsymbol{\mu}) = \begin{bmatrix} \underline{\underline{D}}^{aff,k}(\boldsymbol{\mu}) & \underline{0}^{3\times 1} \\ \underline{0}^{1\times 3} & 1 \end{bmatrix}. \tag{4.30}$$

Similarly, the transformed linear form can be expressed as

$$f(v;\boldsymbol{\mu}) = \sum_{k=1}^{K_{dom}} \int_{\Omega_r^k} \mathcal{F}^k(\boldsymbol{\mu}) \, v \, d\Omega_r^k. \tag{4.31}$$

Here $\mathcal{F}^k(\boldsymbol{\mu}) : \mathcal{D} \to \mathbb{R}$, $1 \leqslant k \leqslant K_{dom}$ is given by:

$$\mathcal{F}^k(\boldsymbol{\mu}) = \mathcal{F}_o^k \left(J^{aff,k}(\boldsymbol{\mu})\right)^{-1} \quad 1 \leqslant k \leqslant K_{dom} \tag{4.32}$$

We note that, in general, the $\underline{\underline{\mathcal{K}}}^k(\boldsymbol{\mu})$, $\mathcal{F}(\boldsymbol{\mu})$, will be different for each subdomain Ω^k, $1 \leqslant k \leqslant K_{dom}$. The differences can be due to *property variation* (e. g. a diffusivity of a particular subdomain) or to *geometry variation* (e. g.a characteristic dimension of the physical problem), or *both*.
We thus require, as already indicated earlier, that the *FE* approximation be built upon a *subtriangulation* of the *RB* discretization: discontinuities in *PDEs* coefficients are therefore restricted to element faces.
In this way, the boundary elements chosen for the *RB* triangulation will delimit a very well defined region of space (our *RB* subdomains), on which we assume that the parameters will be *constants in space*. This allows a simpler *identification/extraction* of the terms in the affine expansion 2.20, as we now discuss.

4.3.3 Affine form

We focus here on a, though f admits a similar treatment. We simply expand the transformed form on the reference domain, by considering in turn each subdomain Ω_r^k and *each entry* of the diffusivity/conductivity tensor $\mathcal{K}_{ij}$, $1 \leqslant ij \leqslant 4$, $1 \leqslant k \leqslant K_{dom}$.
Thus the affine form 4.26 can be written as follows:

$$\begin{aligned} a(w,v;\mu) =& \mathcal{K}_{11}^1 \int_{\Omega_r^1} \frac{\partial w}{\partial x_{r_1}} \frac{\partial v}{\partial x_{r_1}} \, d\Omega_r^1 + \dots \\ &\dots + \underbrace{\mathcal{K}_{ij}^k}_{\theta_a^q(\mu)} \underbrace{\int_{\Omega_r^k} \frac{\partial w}{\partial x_{r_i}} \frac{\partial v}{\partial x_{r_j}} \, d\Omega_r^k}_{a^q(w,v)} + \dots \\ &\dots + \mathcal{K}_{44}^{K_{dom}} \int_{\Omega_r^{K_{dom}}} w v \, d\Omega_r^{K_{dom}} \end{aligned} \tag{4.33}$$

We can then identify each component in the affine expansion: for each term in 4.33 the pre-factor of the integral represents $\theta_a^q(\mu)$, whereas the integral represents the parameter independent matrices $\underline{\underline{\mathbb{A}}}^q = a^q(w,v)$.
For a better understanding of what we have just obtained, we can have a look to the equation 3.36 (pag.34), in which we were building the *RB* system in the coercive case. The *parameter independent* matrices $\underline{\underline{\mathbb{A}}}^q$ can be now exploited even in the general case of *geometric parameter dependence*, thanks to the *geometric affine precondition*.

Affine expansion terms count

In the most general *scalar* case, the number of affine expansion terms can be (at most) $Q_a = 4 \times 4 \times K_{dom}$. Exploiting the symmetry of the bilinear form, hence of the tensor $\underline{\underline{\mathcal{K}}}^k$, only $Q_a = 10 \times K_{dom}$ terms are needed. In fact since $\mathcal{K}_{ij}^k = \mathcal{K}_{ji}^k$, $i \neq j$, the pre-factor associated to these integrals can be assembled together.
We first consider the 6 differents entries of the simmetric tensor $\mathcal{K}$ of the first subdomain, then the second subdomain and so on. Hence, the $\theta_a^q(\mu)$ and the associated *parametric independent matrices* are given by:

Dealing with the *vectorial* case, as we shall see in the *elastic block* case (7), the number of affine expansion terms can be $Q_a = 9 \times 9 \times K_{dom}$. Thanks to the symmetry of the problem and to the particular structure of the *elastic tensor*, we will limit this terms (at most) to $Q_a = 7 \times K_{dom}$.
Therefore it is crucial, in order to reduce the *RB* computational cost that depends on Q_a (see table 1), to minimize the number of terms of the affine expansion.

q	$\theta_f^q(\boldsymbol{\mu})$	$\underline{\mathbb{F}}^q$
1	$\mathcal{K}_{11}$	$\int_{\Omega_r^1} \frac{\partial w}{\partial x_{r_1}} \frac{\partial v}{\partial x_{r_1}} \, d\Omega_r^1$
2	$\mathcal{K}_{12}^1$	$\int_{\Omega_r^1} \frac{\partial w}{\partial x_{r_1}} \frac{\partial v}{\partial x_{r_2}} \, d\Omega_r^1$
7	$\mathcal{K}_{11}^2$	$\int_{\Omega_r^2} \frac{\partial w}{\partial x_{r_1}} \frac{\partial v}{\partial x_{r_1}} \, d\Omega_r^2$
Q_a	$\mathcal{K}_{11}^{K_{dom}}$	$\int_{\Omega_r^{K_{dom}}} w v \, d\Omega_r^{K_{dom}}$

Table 4: $\theta^q(\boldsymbol{\mu})$—functions and parameter independent matrices

5 SOFTWARE

In this chapter we will present the software used to *exploit* our *RB* approximation for 3D applications.
In the past few years a software has been released called `rbMIT`©, which *implements* the *RB* method.
The software has been developed at the *MIT* (Massachusetts Institute of Technology) by a research team, composed by D.B.P. Huynh, NC. Nguyen, A.T. Patera and G. Rozza.
The software is copyrighted but freely distributed and available for non-commercial purposes at the following web address
`http://augustine.mit.edu/`.
We will not go deeply inside the software, we will only present the task demanded to the software related to our particular purposes. Further explanations may be found on the software's manual present in [PR09].
In particular, within the framework of the thesis, we cannot simply make use of the `rbMIT` to build our *RB* approximation of the *FE* truth. This is due to the fact that the software is able just to exploit 1D and 2D case. Therefore dealing with 3D applications, we have used other capabilities offered by a software which would be able to fill the gap. The software chosen to collaborate with the `rbMIT` is `COMSOL multiphysics`. The *interaction* of this two softwares, will provide us the right tools to exploit the *RB* in the 3D case.

5.1 SOFWARE INTERACTION

We now briefly explain the *RB* procedure in the 2D and 3D case, focusing on the task demanded to the two software.
In the 2D case the `rbMIT` entirely covers the reduced basis methodology, starting from the geometrical setting until the Online stage, whereas in the 3D case the procedure is accomplished thanks to a strict interaction with `COMSOL`.

5.1.1 2D case

In the 2D case `rbMIT`, see Figure 11, is able to cover the whole *RB* procedure, hence actually no interaction with third party software is needed. We can summarize the main steps as follows:

1. Problem setting`rbMIT`

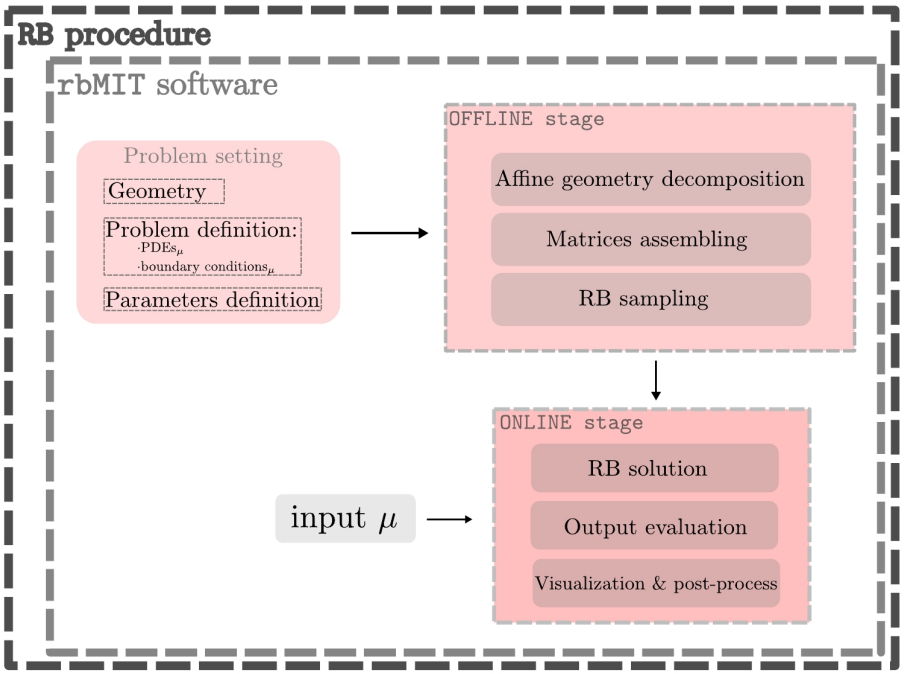


Figure 11: software 2D case

- parametric geometry definition
- PDEs definition
- boundary condition setting

2. Offline stage[rbMIT]
   - *RB* triangulation
   - *FE* meshing
   - *FE* matrices and vectors assembling
   - *RB* system assembling
3. Online stage[rbMIT]
   - *RB* solution
   - output evaluation
   - *a-posteriori error bound* evaluation
   - visualization & post-processing

5.1.2 3D case

In the 3D case, see figure 12, the *RB* procedure can be achieved thanks to an interaction of the two softwares in the `Matlab` environment. Some

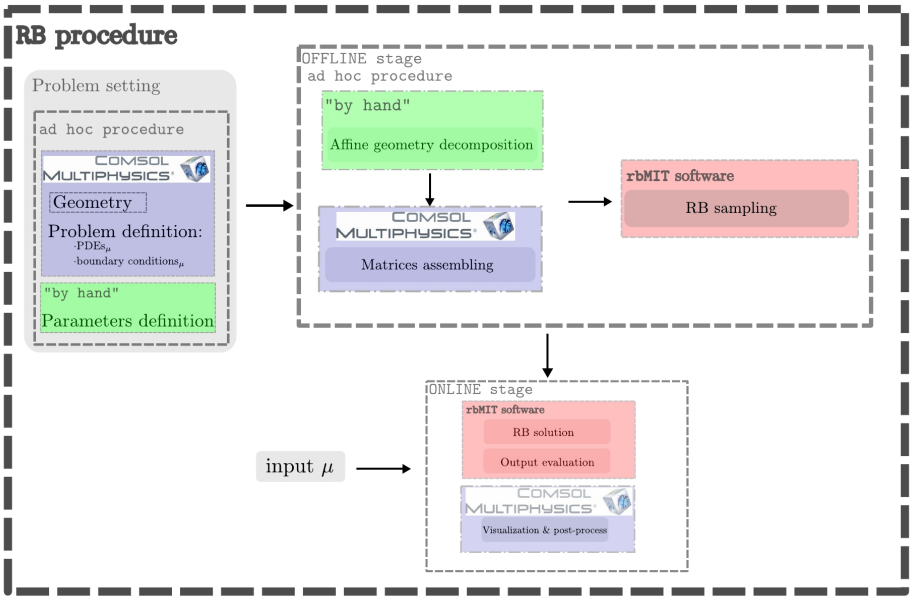


Figure 12: software 3D case

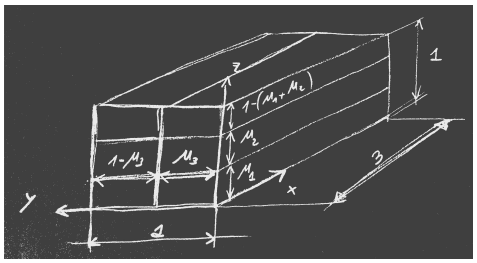
steps will be demanded to the `rbMIT`, in particular the *RB* system assembling, whereas `COMSOL` will be used to build the *FE* truth approximation, over which the *RB* approximation will be build and treated by the `rbMIT`. We can summarize the main steps as follows:

1. Problem setting
   - parametric geometry definition[by hand]
   - *PDEs* definition[COMSOL]
   - boundary condition setting[COMSOL]
2. Offline stage
   - *RB* triangulation/affine geometry precondition[by hand]
   - *FE* meshing [COMSOL]
   - *FE* matrices and vectors assembling[COMSOL]
   - *RB* system assembling[rbMIT]
3. Online stage
   - *RB* solution[rbMIT]
   - output evaluation[rbMIT]
   - *a-posteriori error bound* evaluation[rbMIT]
   - visualization & post-processing[COMSOL]

We now introduce each step summarized above.

Parametric geometry definition

In this section, equipped with blackboard and chalk, the original domain with the geometric parameter dependence is drawn.



This stage is crucial in the subsequent *affine geometric precondition*. In fact, since in the 3D case an automatic procedure to compute an *RB* discretization lacks, the user must provide a *suitable* domain decomposition able to satisfy the conditions discussed in Section 4.1.

PDEs *definition/boundary conditions*

This steps are achieved with `COMSOL`. The procedure is *innovative* in the *RB* context, hence will explain it thoroughly in section 5.2.2.

RB *discretization/affine geometry precondition*

With regard to this step, we refer the reader to the Chapter 4. In that chapter we have discussed the theoretical aspects (Section 4.1) and by means of two examples (Sections 4.2.1 and 4.2.2) we have presented the numerical procedure.

FE *meshing and assembling*

In this step the software used is `COMSOL`, the aim is to provide the set of parameter independent *FE* matrices and vectors with which build our *RB* system.
We remark that, thanks to the *affine parametric dependence* (section 2.2) and to the *affine geometric precondition* (Section 4.1), we have to perform the *meshing* and the *FE* assembling only once.
In fact, even in the case of geometrical deformation induced by geometrical parameters, the procedure shown in Section 4.3 allows us to recover the geometrical dependence using the *affine decomposition*.

RB *system assembling/Output evaluation*

In this stage the *FE* ingredients are passed to the `rbMIT` software which will provide the *RB* approximation.
We will briefly summarize the entries needed by the software and the main steps of the numerical procedure.
With regard to the entries we have to provide the following inputs:

1. Q_a terms of the affine expansion of the bilinear form, see equations 3.36-3.38:
   - *FE* matrices $\underline{\underline{\mathbb{A}}}$;
   - parametric dependent functions $\theta_a^q(\boldsymbol{\mu})$;
2. Q_f terms of the affine expansion of the linear functional, see equations 3.37-3.39:
   - *FE* vectors $\underline{\mathbb{F}}$;
   - parametric dependent functions $\theta_f^q(\boldsymbol{\mu})$;
3. Mass matrix[1]
4. Class of problem:
   - elliptic coercive;
   - elliptic non-coercive;
   - parabolic;
5. Parameter definition:
   - number of parameters;
   - parameter domain $\mathcal{D}$.

Whereas with regard to the `rbMIT` task, the software traces the procedure explained in the chapter 3. Hence, the main steps achieved are:

- Offline
  1. *Greedy* sampling for the *successive constraint method SCM*, see Section 3.6.3;
  2. *Greedy* sampling for the *RB* basis assemblingm, Section 3.4.2;
- Offline
  1. *RB* system solution, Section 3.2.6;
  2. output evaluation, as mentioned above;
  3. *a-posteriori* error bound evaluation, Section 3.5.

In the following section we introduce the `COMSOL` software and we will explain how to compute our *FE truth approximation*.
Since the `rbMIT` software is equipped by a fully descriptive manual [PR09], we will only describe the main steps to obtain the *FE* ingredients with `COMSOL`.

1 In order to build the matrix representation of the X−norm

5.2 COMSOL MULTIPHYSICS FE ASSEMBLING

COMSOL Multiphysics© is a powerful environment for modeling and solving all kind of scientific and engineering problems based on *PDEs*.
The software is able to work with one dimensional, two dimensional an three dimensional domains, thus is really well suited for our purposes, even if not in its default settings.
With `COMSOL` it is possible to deal also with *multiphysics* models which solve coupled physical phenomena. Actually this feature will not be exploited in this thesis, but of course in the following works dealing with the *RB* methodology it could be of great interest.
Thanks to the built-in *physics mode* it is possible to define the relevant physical quantities—such as material properties, loads, constraints, sources, and fluxes—rather than by defining the underlying equations. These variables, expressions, or numbers can always be applied directly to solid domains, boundaries, edges, and points independently of the computational mesh.
It is possible to access to `COMSOL Multiphysics` through a script by programming in the *MATLAB* language.
Thanks to the great flexibility of the software, we will be able to obtain the numerical ingredients required by the *RB* method in a proper way. In the following sections we will discuss the procedure used.

We begin recalling the finite element matrices and vectors needed to build our *RB* system. We then introduce some basic commands of `COMSOL` used in order to better understand the *FE* assembling procedure.

5.2.1 FE ingredients

In order to create the *RB* approximation (section 3.2 in the coercive case and section 3.3 in the Stokes case), we need an underlying *FE* approximation.
This will be accomplished providing the *FE* matrices and array built on our *reference domain* (refer to Section 4 for a detailed explanation).
To show the *FE* matrices assembling within the *COMSOL* environment, we consider a *coercive scalar* case.
Therefore, we recall from section 3.2.6 (pag.33) that we need to form:

1. Q_a parametric independent matrices $\underline{\underline{\mathbb{A}}}^q$ (eq.3.36)
2. Q_f parametric independent arrays $\underline{\mathbb{F}}^q$ (eq.3.37)

built thanks to a standard *FE* discretization of the *PDE* problem.

5.2.2 Problem definition

The main steps to assemble the *FE* matrices, are summarized as follows:

1. creation of the geometry;
2. definition of the *PDEs*;
3. imposition of the boundary condition;
4. choice of the *FE* basis functions;
5. meshing;
6. matrices assembling.

We will just consider the definition of the partial differential system *tuned for our particular purposes* along with the choice of the boundary conditions, since the other steps can be easily accomplished following the COMSOL reference manual [Como7b]. Moreover we will consider a single subdomain, the extension to the multi-subdomain case is straightforward.

5.2.2.1 *Definition of the PDEs/boundary conditions*

There are several *built-in* automatic applications in the *COMSOL* software thanks to which the stiffness matrix and the relative RHS (right hand side) can be easily build for a wide range of physical problem, such as: acoustics, electromagnetics, fluid dynamics, heat transfer, structural mechanics, transport phenomena, etc...
Unfortunately, due to our highly specialized task, this automatic way to proceed is unsuited. Our efforts will be devoted to the creation of an *ad-hoc* application able to provide the matrices arising from the affine expansion.
Therefore to build the *FE* ingredients, we will define our *PDEs* thanks to another application offered by COMSOL, the PDE mode.

PDE mode

The PDE mode is an *equation-based modeling* procedure: the system of equation is given in the coefficient form, that is the *strong formulation*, for further details we refer to the COMSOL manual [Como7a].
In the *scalar* case the coefficient form in *COMSOL* is written as

$$\begin{cases} e_a \dfrac{\partial^2 u}{\partial t^2} + d_a \dfrac{\partial u}{\partial t} + \nabla \cdot (c\nabla u - \alpha u + \gamma) + \beta \cdot \nabla u + a u = f & \text{in } \Omega \\ \mathbf{n} \cdot (c\nabla u + \alpha u - \gamma) + q u = g - h^T \mu & \text{on } \partial\Omega \\ h u = r & \text{on } \partial\Omega \end{cases} \tag{5.1}$$

where $\Omega = \Omega_r$ is our reference domain, the union of all subdomains Ω_r^k, $\partial\Omega = \partial\Omega_r$ is the domain boundary, $\mathbf{n}$ is the outward unit normal vector on $\partial\Omega$.
The first equation in the list above is the *PDEs*, which must be satisfied in

Ω. The second and third equations are the boundary conditions, which must hold on $\partial\Omega$. The second equation is a *generalized Neumann boundary condition*, whereas the third equation is a *Dirichlet boundary condition*. The system is written in the most general way; since we are dealing with elliptic equations the *temporal terms vanishes*. We now provide an explanation of the terms appearing in the equation 5.1:

- ∇ is the differential operator (gradient) defined as a column vector

$$\nabla = \left\{ \frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3} \right\}^T,$$

- $\nabla \cdot (c\nabla u)$ is due to diffusivity/conductivity, where $c \in \mathbb{R}^{3\times3}$ and it means

$$\frac{\partial}{\partial x_1}\left(c_{11}\frac{\partial u}{\partial x_1}\right) + \frac{\partial}{\partial x_1}\left(c_{12}\frac{\partial u}{\partial x_2}\right) + \ldots + \frac{\partial}{\partial x_3}\left(c_{33}\frac{\partial u}{\partial x_3}\right),$$

- $\nabla \cdot (\alpha u)$ is due to convection, where $a \in \mathbb{R}^{3\times1}$ is the convective velocity

$$\frac{\partial \alpha_1 u}{\partial x_1} + \frac{\partial \alpha_2 u}{\partial x_2} + \frac{\partial \alpha_3 u}{\partial x_3},$$

- $\nabla \cdot (\gamma)$, is due to source term, where $\gamma \in \mathbb{R}^{3\times1}$

$$\frac{\partial \gamma_1}{\partial x_1} + \frac{\partial \gamma_2}{\partial x_2} + \frac{\partial \gamma_3}{\partial x_3},$$

- $\beta \cdot \nabla u$, that will prove useful to build the Stokes problem (see Chapter 8) and it means

$$\beta_1\frac{\partial u}{\partial x_1} + \beta_2\frac{\partial u}{\partial x_2} + \beta_3\frac{\partial u}{\partial x_3},$$

  where $\beta \in \mathbb{R}^{3\times1}$.

The formulation of the *boundary conditions* considers *Dirichlet* and *Neumann conditions*.

- Generalized *Neumann* condition:
  - $(c\nabla u - \alpha u + \gamma)$ is the *flux vector* of the *homogeneus* Neumann conditions;
  - $g \in \mathbb{R}$ is the boundary source term;
  - $q \in \mathbb{R}$ is the terms related to the *Robin* boundary condition;
- *Dirichlet* condition:
  - r is the term related to the *non-homogeneus* Dirichlet conditions.

The combination of both boundary condition is possible thanks to a new dependent variable μ, which is defined *only on the boundary*. The unknown variable μ is called a *Lagrange multiplier*.
The apposition of a Neumann condition rather than a Dirichlet condition depends upon a pre-factor h, see [Como7a].
In scalar problems, h is a scalar; if $h = 0$, the Neumann condition on that boundary is deactivated and the Dirichlet condition is apposed.
In the vectorial case h is a matrix $\mathbb{R}^{3\times3}$, if $h = \underline{\underline{I}}^{3\times3}$ (=identity matrix) only the Dirichlet boundary conditions are active, whereas if h differs by the identity matrix, then the result will be a mixing of Dirichlet and Neumann conditions, coupled by the Lagrange multiplier μ.

5.2.3 A test case

In this section, thanks to a test case, we will show how to set up the problem within the COMSOL environment.
We will exploit a thermal problem, that is governed by the *Fourier equation*. The domain is constituted by a single block made by a isotropic conductive media. With regard to the figure 13, we suppose that the

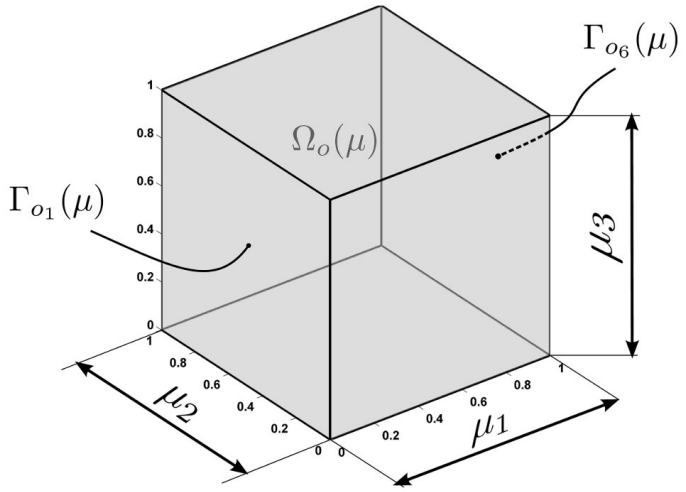


Figure 13: Original domain test case

edges are parameter dependent $\boldsymbol{\mu} = \{\mu_1, \mu_2, \mu_3\}$. Moreover, on face Γ_1 we impose an homogeneus Dirichlet condition, on Γ_6 we impose an unitary non-homogeneus Neumann condition, on all the other faces we impose an homogeneus Neumann condition.
We recall that, to create our *FE* approximation we choose a *set of reference parameters*, in order to build our reference domain Ω_r.
Hence we choose a parameter of reference $\overline{\boldsymbol{\mu}} = \{\overline{\mu}_1, \overline{\mu}_2, \overline{\mu}_3\}$ that induces a reference domain $\Omega_r = \Omega_o(\overline{\boldsymbol{\mu}})$, over which we will assemble our parame-

ter independent matrices $\underline{\underline{\mathbb{A}}}^q$ representing the discretization of the weak formulation of the following problem:

$$\begin{cases} -\nabla_r \cdot \left(\underline{\underline{K}}_r \nabla_r u\right) = 0 & \text{in } \Omega_r \\ u = 0 & \text{on } \Gamma_{r_1} \\ \dfrac{\partial u}{\partial \mathbf{n}} = 1 & \text{on } \Gamma_{r_6} \end{cases} \tag{5.2}$$

Recalling that the conductive media is isotropic[2], we can represent it as the following matrix:

$$\underline{\underline{K}}_r = \begin{bmatrix} \kappa & 0 & 0 \\ 0 & \kappa & 0 \\ 0 & 0 & \kappa \end{bmatrix}. \tag{5.4}$$

Therefore, recalling the procedure shown in Section 4.3.3, we need to build three matrices relatives to the three different entries of the conductivity tensor $\underline{\underline{K}}_r$.
Then, recalling the weak formulation 4.33, we assemble the three *FE* matrices defined as follows:

$$\begin{aligned} \underline{\underline{\mathbb{A}}}^1 &= a^1(w,v) = \int_{\Omega_r} \frac{\partial w}{\partial x_{r_1}} K_{r_{11}} \frac{\partial v}{\partial x_{r_1}} \, d\Omega_r, \\ \underline{\underline{\mathbb{A}}}^2 &= a^2(w,v) = \int_{\Omega_r} \frac{\partial w}{\partial x_{r_2}} K_{r_{22}} \frac{\partial v}{\partial x_{r_2}} \, d\Omega_r, \\ \underline{\underline{\mathbb{A}}}^3 &= a^3(w,v) = \int_{\Omega_r} \frac{\partial w}{\partial x_{r_3}} K_{r_{33}} \frac{\partial v}{\partial x_{r_3}} \, d\Omega_r. \end{aligned} \tag{5.5}$$

To form this matrices with *COMSOL* (recalling the nomenclature of equation 5.1), we assemble the *FE* matrices setting on Ω_r in the three cases:

$$\begin{aligned} c^1 &= \begin{bmatrix} \kappa & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\ c^2 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & \kappa & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\ c^3 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \kappa \end{bmatrix}, \end{aligned} \tag{5.6}$$

in addition we set to zero the constants: α, γ, β and a.
The boundary conditions, imposed on $\partial\Omega_r$[3], are the same in the three

2 That is the conductivity tensor is defined as

$$K_{ij} = \begin{cases} \kappa & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{5.3}$$

3 $\partial\Omega_r = \bigcup_{k=1}^{6} \Gamma_{r_k}$

cases.
To define the boundary conditions shown in system 5.2, we set:

- on Γ_1

$$\begin{cases} q = 0 \\ g = 0 \\ h = 1 \\ r = 0 \end{cases} \quad \text{Homogeneus Dirichlet,}$$

- on Γ_{r_1}, Γ_{r_2}, Γ_{r_3}, Γ_{r_4}, Γ_{r_5}

$$\begin{cases} q = 0 \\ g = 0 \\ h = 0 \\ r = 0 \end{cases} \quad \text{Homogeneus Neumann,}$$

- on Γ_{r_6}

$$\begin{cases} q = 0 \\ g = 1 \\ h = 0 \\ r = 0 \end{cases} \quad \text{Non-homogeneus Neumann} \tag{5.7}$$

We remark that we have chosen a coercive scalar problem on a single subdomain, hence the procedure here is very straightforward. In the part devoted to the Mathematical Modelling, we will discuss more involved problems.

COMSOL input file

We now provide, as example, the code associated to the test case discussed above, to provide a sample of the input file within the COMSOL environment.
COMSOL input file is written with the same programming language of MATLAB, hence the command list is very "user friendly".

COMSOL input file for $\underline{\underline{\mathbb{A}}}^1$ assembling

```
% COMSOL Multiphysics Model M-file

flclear fem

% Geometry
g1=block3('1','1','1','base','corner','pos',{'0','0','0'},'axis',{'0'
    ,'0','1'},'rot','0');

% Analyzed geometry
fem.geom=geomobject(g1);

% Constants
fem.const = {'k','1'};
descr.const= {'k','Conductivity'};
fem.descr = descr;

% Mesh
```

```
fem.mesh=meshinit(fem,'hauto',4);

%% Application mode 1

% FE basis functions
clear appl
appl.mode.class = 'FlPDEC';
appl.shape = {'shlag(1,''u'')'};
appl.gporder = {2,4};
appl.cporder = 1;
appl.assignsuffix = '_c';

% Boundary conditions
clear bnd
bnd.g = {0,0,1};
bnd.name = {'Homogeneus Dirichlet','Homogeneus Neumann','Inhomogeneus
     Neumann'};
bnd.type = {'dir','neu','neu'};
bnd.ind = [1,2,2,2,2,3];
appl.bnd = bnd;

% PDEs setting
clear equ
equ.f = 0;
equ.da = 0;
equ.bndgporder = 2;
equ.c = {{{'k';0;0}}};
equ.ind = [1];
appl.equ = equ;
fem.appl{1} = appl;

% Multiphysics
fem=multiphysics(fem);

% Meshextend
fem.xmesh=meshextend(fem);

% Assembling
[K,F,Null,ud] = femlin(fem); %with Dirichlet condition
```

6 THERMAL PROBLEM

6.1 INTRODUCTION

In this chapter we will exploit the creation of our *RB* problem *in the 3D case* dealing with a *scalar elliptic coercive* problem.
We deal with a steady conduction thermal problem, assuming that the thermal conductivity $\underline{\underline{K}}$ is represented by a positive definite matrix; then the unknown is the 3D field of temperature, that we will denote with $u(\mathbf{x};\boldsymbol{\mu}) \in \mathbb{R}$. This class of problem, although rather simple, is able to describe a wide range of engineering problem, see for example [Arp66].
We mention, for example, the study of the performance of an heat sink designed for the thermal management of high-density electronic components, the design of an insulated coverage of a building to reduce the energetic consumption, the control of the temperature within an engine shaft to prevent thermal stresses or deformation, etc...
Another notable application can be the non-destructive testing of mechanical components or the identification of an inclusion within a casting steel; in short, despite the simple mathematical fomrulation, this case is more than a Mathematical abstraction.

6.2 PROBLEM DESCRIPTION

We now briefly introduce the Thermal Block problem, henceforth we will refer to this case with the "TB" label.

Physical problem

The problem of a *steady-conduction* is considered here in a cubic domain.
We want to evaluate the thermal field in a non-isotropic conductive block, with an heat flux q imposed on a face of the cube, with regard to a reference environmental temperature taken on the opposite face and assuming that the other faces are insulated. The cube has an anisotropic conductivity due to an inclusion of different materials within the piece.
Dealing for example with the material science, the inclusion can be, see [Wal93]:

1. a *deficiency* of material, a hole due to gas bubble present within the casting during solidification, or due to a fatigue crack;

2. an inhomogeneity of the material, due to an unbalancing of concentration of the alloyants.

Therefore, in order to seize the behavior of such a phenomena, the *RB* parametric approach is very well suited. The thermal problem, in the 2D frame, has been treated by [RHP08, PR09] and [HRSP07].
The output of interest is the average temperature of the heated face.
Now we formulate the problem into the mathematical setting.

Analytical problem

The equation that describes the field of temperature is represented by the *Fourier equation* in which will be neglected the time derivative thanks to the hypothesis of a steady state problem (no transient effect considered) as well as the volume force that will be considered negligible.

6.3 PARAMETERS CHOICE

Due to the nature of the inclusion, its shape and dimension can vary arbitrarily, hence we introduce some geometric parameters in order to describe it. It is important to remark that, due the fact that the parametrized geometry should allow an *affine geometric precondition* (discussed in Chapter 4), we restrict our attention to a parallelepiped inclusion.
In particular, with regard to the Figure 14, we have assumed that the cubic block can vary its *dimension* along the three axis by means of a set of parameters $\boldsymbol{\mu} = \{\mu_4, \mu_5, \mu_6\}$. In addition we have chosen other three parameters, denoted by $\boldsymbol{\mu} = \{\mu_1, \mu_2, \mu_3\}$ to parametrize the *position* of the inclusion. The solid is schematized as $3 \times 3 \times 3$ cubic blocks, the central

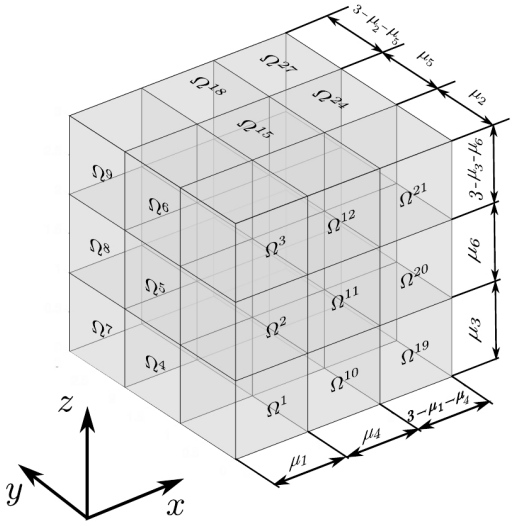


Figure 14: TB domain decomposition

block (subdomain Ω^{14} in our scheme) represents the inclusion.
Each block, due to the parametric geometry dependence of the inclusion and to the hypothesis required by an affine geometry (Section 4.1), is subjected to a geometrical parametric dependence. Each sub-block is considered *isotropic*. The conductivity constant for the central sub-block (inclusion) is another parameter, denoted by μ_7, whereas for the other sub-blocks the conductivity is the unity (reference).

Parameters domain

We now summarize the parameters and the parameter domain chosen to describe our *TB* problem as:

- the dimension and the position of the inclusion, that can be described by 6 parameters, that are 3 *translations* and 3 *dimensions*. The parameters $\mu_{1:6}$ are shown in Figure 14.
  The parameter domain for the geometrical quantities is:

$$\begin{aligned}\mathcal{D}_{\text{geom}} &= \left[\mu_1^{\min}, \mu_1^{\max}\right] \times \ldots \times \left[\mu_6^{\min}, \mu_6^{\max}\right] \\ &= [0.5, 1.45] \times \ldots \times [0.5, 1.45]\,, \end{aligned} \tag{6.1}$$

- the conductivity coefficient of the inclusion, denoted with μ_7.
  The parameter domain for this physical quantity is:

$$\mathcal{D}_{\text{physics}} = \left[\mu_7^{\min}, \mu_7^{\max}\right] = [0.1, 10]\,. \tag{6.2}$$

The parameter domain is therefore given by $\boldsymbol{\mu} \in \mathcal{D} \in \mathbb{R}^{P=7}$, such that:

$$\begin{aligned}\mathcal{D} &= \mathcal{D}_{\text{geom}} \times \mathcal{D}_{\text{physics}} \\ &= [0.5, 1.45] \times \ldots \times [0.5, 1.45] \times [0.1, 10]\,. \end{aligned}$$

Boundary conditions

With regard to the boundary conditions (Figure 15), a non-homogeneus Neumann boundary condition is imposed on Γ_6 representing an *heat flux*, an homogeneus Dirichlet boundary condition is imposed on Γ_1 representing the imposition of a temperature (adimensional, i.e. environmental temperature), whereas on the other external faces of the cube $\Gamma_{2:5}$ homogeneus Neumann conditions has been chosen, representing insulation of the walls. Finally, on the internal faces we have assumed continuity of *temperature* and *fluxes*.

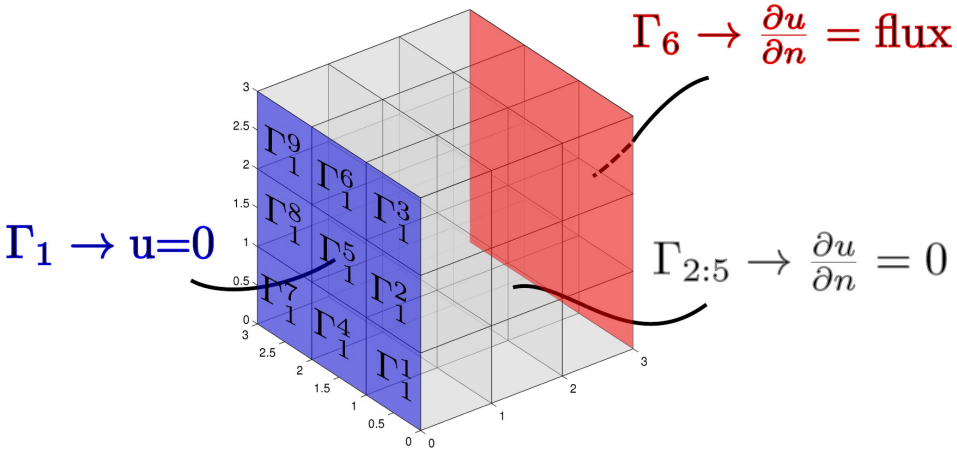


Figure 15: TB boundary conditions

6.4 TB PROBLEM FORMULATION

6.4.1 Original domain

We introduce the analytical formulation of the governing *PDEs* on the *original domain*.The equation which describes the field of temperature, within the hypothesis described in Section 6.1, is the following:

$$\begin{cases} -\nabla \cdot \left(\underline{\underline{K}}_o \nabla_o u \right) = 0 & \text{in } \Omega_o(\boldsymbol{\mu}) \\ u = 0 & \text{on } \Gamma_{o_1}(\boldsymbol{\mu}) \\ \dfrac{\partial u}{\partial \mathbf{n}} = q & \text{on } \Gamma_{o_6}(\boldsymbol{\mu}) \end{cases} \tag{6.3}$$

Multiplying the equation by a suitable *test function* v such that $v \in X^e \equiv \{v \in H_0^1(\Omega) | \, v|_{\Gamma^D} = 0\}$ and integrating over the domain Ω we obtain:

$$\int_{\Omega_o(\boldsymbol{\mu})} -\nabla_o \cdot \left(\underline{\underline{K}}_o \nabla_o u \right) v \, d\Omega_o = 0 \tag{6.4}$$

The domain Ω_o is the *original domain* on which the *PDE* is defined. The whole domain $\Omega_o(\mu)$ is decomposed (Section 4.1) in K_{dom} non overlapping subdomains such that:

$$\Omega_o(\mu) = \bigcup_{k=1}^{K_{27}} \Omega_o^k(\mu)$$

in addition, recalling the *Green Theorem* for the laplacian, [Qua09]:

$$\int_\Omega \Delta u \, v \, d\Omega = \int_\Omega \nabla u \cdot \nabla v \, d\Omega - \int_{\partial\Omega} \frac{\partial u}{\partial \mathbf{n}} v \, d\gamma$$

the equation 6.4 becomes

$$\sum_{k=1}^{27} \int_{\Omega_o^k} -\underline{\underline{K}}_o \nabla_o u \cdot \nabla_o v \, d\Omega_o^k + \sum_{k=1}^{27} \int_{\partial\Omega_o^k} \underline{\underline{K}}_o \frac{\partial u}{\partial \mathbf{n}} v \, \partial\Omega_o^k = 0$$

Thanks to the functional space chosen, the boundary terms vanish on the face on which we have imposed a zero temperature (on $\Gamma_1 \rightarrow v = 0$). The boundary terms relatives to $\Gamma_{2:5}$ vanish because we have imposed an homogeneus Neumann condition (on $\Gamma_{2:5} \rightarrow \partial u/\partial \mathbf{n} = 0$).
The internal faces contributes disappear thanks to the continuity of temperature and fluxes. Then the only remaining boundary term is the one relative to the face Γ_6 on which we have imposed an heat flux q.
Therefore, the equation 6.5 can be simplified as:

$$\sum_{k=1}^{27} \int_{\Omega_o^k} \underline{\underline{K}}_o \nabla_o u \cdot \nabla_o v \, d\Omega_o^k = \int_{\Gamma_{o_6}} \underline{\underline{K}}_o \frac{\partial u}{\partial \mathbf{n}} v \, d\Gamma_{o_6}. \tag{6.5}$$

Replacing the Neumann boundary condition (system 6.3):

$$\underline{\underline{K}}_o \nabla_o u \cdot n = \underline{\underline{K}}_o \frac{\partial u}{\partial \mathbf{n}} = q$$

in the *weak formulation* 6.5, we finally obtain:

$$\sum_{k=1}^{27} \int_{\Omega_o^k} \underline{\underline{K}}_o \nabla_o u \cdot \nabla_o v \, d\Omega_o^k = \int_{\Gamma_{o_6}} q \, v \, d\Gamma_{o_6}. \tag{6.6}$$

Introducing the bilinear form

$$a(u, v; \boldsymbol{\mu}) = \sum_{k=1}^{27} \int_{\Omega_o^k} \underline{\underline{K}}_o \nabla_o u \cdot \nabla_o v \, d\Omega_o^k \tag{6.7}$$

and the linear functional

$$f(v; \boldsymbol{\mu}) = \int_{\Gamma_{o_6}} q \, v \, d\Gamma_{o_6}, \tag{6.8}$$

we can restate the problem 6.6 as: find $u \in X^e(\Omega_o(\boldsymbol{\mu}))$, such that

$$a(u, v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \forall v \in X^e(\Omega_o(\boldsymbol{\mu})) \tag{6.9}$$

The coercivity and the continuity of the bilinear form a and the continuity of the functional f can be proved. Then the *Lax-Milgram* theorem ensures the existence and uniqueness of the solution, see [Qua09].

6.4.2 Reference domain

In this section we apply standard techniques to transform the problem statement over the *original domain* to an equivalent problem fomrulated over *reference domain*.
We shall be ultimately able to write the problem in an *affine formulation* (2.20), to exploit the crucial Offline/Online computational splitting procedure. In order to obtain the problem formulation 6.9 on the reference

domain, we need to evaluate the *affine transformation* for each subdomain $\Omega_o^k(\boldsymbol{\mu}) \in \Omega_o(\boldsymbol{\mu})$, tracing back the *derivatives operator* and all the *geometric parameter dependent quantities* to the *reference domain*, by the recipe provided in Section 4.3.2.
In order to build the *affine decomposition*, we must compute the affine mappings for each subdomain $\mathcal{T}^{aff,k}(\,\cdot\,;\boldsymbol{\mu}) : \Omega_o(\boldsymbol{\mu}) \to \Omega_r$, $1 \leqslant k \leqslant 27$, in order to evaluate:

1. the Jacobian $J^{aff,k}(\boldsymbol{\mu})$, $1 \leqslant k \leqslant 27$ (equation 4.5)

$$J^{aff,k}(\boldsymbol{\mu}) = \left|\det\left(\underline{\underline{G}}^{aff,k}(\boldsymbol{\mu})\right)\right|;$$

2. the derivatives operator $D^{aff,k}(\boldsymbol{\mu})$ (equation 4.6)

$$\underline{\underline{D}}^{aff,k}(\boldsymbol{\mu}) = \underline{\underline{G}}^{aff,k}(\boldsymbol{\mu});$$

We will not present in detail the procedure to obtain the affine transformation for all the subdomains, we refer the reader to the Section 4.2.1 and 4.2.2 for a detailed abstract explanation. In this section we just provide the results in Table 5.
We remark that the matrices $\underline{\underline{G}}^{aff,k}(\boldsymbol{\mu})$, $1 \leqslant k \leqslant 27$ are diagonal thanks to the particular choice of the geometric parameters. Once the affine mappings have been computed, we are able to rewrite the weak formulation 6.9 into the reference domain.

Bilinear form

Recalling the bilinear form defined in equation 6.7, we recall the definition of the conductivity tensor $\underline{\underline{K}}_r$ given in equation 4.29 (section 4.3.2). Due to the isotropic nature of the material, for each subdomain Ω_r^k, $1 \leqslant k \leqslant 27$, we need to extract three different affine terms, corresponding to the three different entries of the conductivity tensor, hence we would have $27 \times 3 = 81$ terms in our affine development. We obtain:

$$\begin{aligned}
a(w,v;\boldsymbol{\mu}) &= \sum_{k=1}^{27} \int_{\Omega_r^k} \underline{\underline{K}}_r(\boldsymbol{\mu})\, \nabla u \cdot \nabla v \, d\Omega_r^k \\
&= \int_{\Omega_r^1} \frac{\mu_2\mu_3}{\mu_1} \frac{\partial u}{\partial x_{r_1}} \frac{\partial v}{\partial x_{r_1}} + \frac{\mu_1\mu_3}{\mu_2} \frac{\partial u}{\partial x_{r_2}} \frac{\partial v}{\partial x_{r_2}} + \frac{\mu_1\mu_2}{\mu_3} \frac{\partial u}{\partial x_{r_3}} \frac{\partial v}{\partial x_{r_3}} \, d\Omega_r^1 + \dots \\
&\dots + \int_{\Omega_r^3} \left(-\frac{\mu_2(\mu_3+\mu_6-3)}{\mu_1} \frac{\partial u}{\partial x_{r_1}} \frac{\partial v}{\partial x_{r_1}} + \right. \\
&\quad \left. -\frac{\mu_1(\mu_3+\mu_6-3)}{\mu_2} \frac{\partial u}{\partial x_{r_2}} \frac{\partial v}{\partial x_{r_2}} - \frac{\mu_1\mu_2}{\mu_3+\mu_6-3} \frac{\partial u}{\partial x_{r_3}} \frac{\partial v}{\partial x_{r_3}} \right) d\Omega_r^3 + \dots \\
&\dots + \theta_a^{81}(\boldsymbol{\mu}) \int_{\Omega_r^{27}} \frac{\partial u}{\partial x_{r_3}} \frac{\partial v}{\partial x_{r_3}} d\Omega_r^{27}
\end{aligned} \tag{6.10}$$

sub	G_{11}^{aff}	G_{22}^{aff}	G_{33}^{aff}	C_1^{aff}	C_2^{aff}	C_3^{aff}
Ω^1	$1/\mu_1$	$1/\mu_2$	$1/\mu_3$	0	0	0
Ω^2	$1/\mu_1$	$1/\mu_2$	$1/\mu_6$	0	0	$1-\mu_3$
Ω^3	$1/\mu_1$	$1/\mu_2$	$1/(3-\mu_6-\mu_3)$	0	0	$2-\mu_6-\mu_3$
Ω^4	$1/\mu_1$	$1/\mu_4$	$1/\mu_3$	0	$1-\mu_1$	0
Ω^5	$1/\mu_1$	$1/\mu_4$	$1/\mu_6$	0	$1-\mu_1$	$1-\mu_3$
Ω^6	$1/\mu_1$	$1/\mu_4$	$1/(3-\mu_6-\mu_3)$	0	$1-\mu_1$	$2-\mu_6-\mu_3$
Ω^7	$1/\mu_1$	$1/(3-\mu_5-\mu_2)$	$1/\mu_3$	0	$2-\mu_5-\mu_1$	0
Ω^8	$1/\mu_1$	$1/(3-\mu_5-\mu_2)$	$1/\mu_6$	0	$2-\mu_5-\mu_1$	$1-\mu_3$
Ω^9	$1/\mu_1$	$1/(3-\mu_5-\mu_2)$	$1/(3-\mu_6-\mu_3)$	0	$2-\mu_5-\mu_1$	$2-\mu_6-\mu_3$
Ω^{10}	$1/\mu_5$	$1/\mu_2$	$1/\mu_3$	$1-\mu_2$	0	0
Ω^{11}	$1/\mu_5$	$1/\mu_2$	$1/\mu_6$	$1-\mu_2$	0	$1-\mu_3$
Ω^{12}	$1/\mu_5$	$1/\mu_2$	$1/(3-\mu_6-\mu_3)$	$1-\mu_2$	0	$2-\mu_6-\mu_3$
Ω^{13}	$1/\mu_5$	$1/\mu_4$	$1/\mu_3$	$1-\mu_2$	$1-\mu_1$	0
Ω^{14}	$1/\mu_5$	$1/\mu_4$	$1/\mu_6$	$1-\mu_2$	$1-\mu_1$	$1-\mu_3$
Ω^{15}	$1/\mu_5$	$1/\mu_4$	$1/(3-\mu_6-\mu_3)$	$1-\mu_2$	$1-\mu_1$	$2-\mu_6-\mu_3$
Ω^{16}	$1/\mu_5$	$1/(3-\mu_5-\mu_2)$	$1/\mu_3$	$1-\mu_2$	$2-\mu_5-\mu_1$	0
Ω^{17}	$1/\mu_5$	$1/(3-\mu_5-\mu_2)$	$1/\mu_6$	$1-\mu_2$	$2-\mu_5-\mu_1$	$1-\mu_3$
Ω^{18}	$1/\mu_5$	$1/(3-\mu_5-\mu_2)$	$1/(3-\mu_6-\mu_3)$	$1-\mu_2$	$2-\mu_5-\mu_1$	$2-\mu_6-\mu_3$
Ω^{19}	$1/(3-\mu_4-\mu_1)$	$1/\mu_2$	$1/\mu_3$	$2-\mu_4-\mu_2$	0	0
Ω^{20}	$1/(3-\mu_4-\mu_1)$	$1/\mu_2$	$1/\mu_6$	$2-\mu_4-\mu_2$	0	$1-\mu_3$
Ω^{21}	$1/(3-\mu_4-\mu_1)$	$1/\mu_2$	$1/(3-\mu_6-\mu_3)$	$2-\mu_4-\mu_2$	0	$2-\mu_6-\mu_3$
Ω^{22}	$1/(3-\mu_4-\mu_1)$	$1/\mu_4$	$1/\mu_3$	$2-\mu_4-\mu_2$	$1-\mu_1$	0
Ω^{23}	$1/(3-\mu_4-\mu_1)$	$1/\mu_4$	$1/\mu_6$	$2-\mu_4-\mu_2$	$1-\mu_1$	$1-\mu_3$
Ω^{24}	$1/(3-\mu_4-\mu_1)$	$1/\mu_4$	$1/(3-\mu_6-\mu_3)$	$2-\mu_4-\mu_2$	$1-\mu_1$	$2-\mu_6-\mu_3$
Ω^{25}	$1/(3-\mu_4-\mu_1)$	$1/(3-\mu_5-\mu_2)$	$1/\mu_3$	$2-\mu_4-\mu_2$	$2-\mu_5-\mu_1$	0
Ω^{26}	$1/(3-\mu_4-\mu_1)$	$1/(3-\mu_5-\mu_2)$	$1/\mu_6$	$2-\mu_4-\mu_2$	$2-\mu_5-\mu_1$	$1-\mu_3$
Ω^{27}	$1/(3-\mu_4-\mu_1)$	$1/(3-\mu_5-\mu_2)$	$1/(3-\mu_6-\mu_3)$	$2-\mu_4-\mu_2$	$2-\mu_5-\mu_1$	$2-\mu_6-\mu_3$

Table 5: TB affine mappings

Linear functional

In this case the parametric linear functional (equation 6.8) arises from an inhomogeneous Neumann boundary condition. This case has not been treated in Section 4.3. In order to cast the integral of equation 6.8 into the reference domain, we proceed as follows:

$$\begin{aligned} f(v;\boldsymbol{\mu}) &= \int_{\Gamma_{o_6}(\boldsymbol{\mu})} q\, v \, d\Gamma_{o_6}(\boldsymbol{\mu}) \\ &= \int_{\Gamma_{r_6}} q\, v \underbrace{\left| \left(G^{aff,k}(\boldsymbol{\mu}) \right)^{-1} \cdot \underline{e}^t \right|}_{d\Gamma_{o_6}} d\Gamma_{r_6} \end{aligned} \tag{6.11}$$

where $\underline{e}^t$ denotes the tangential unit vector and k indicates the indexes of the subdomains to which the face Γ_{r_6} belong. In particular, with regard

the chosen subdomain enumeration (Figure 14), we see that Γ_{r_6} is given by

$$\Gamma_{r_6} = \bigcup_{k=1}^{6} \Gamma_{r_6}^k. \tag{6.12}$$

Therefore the linear functional can be rewritten in an affine development as:

$$\begin{aligned} f(v;\boldsymbol{\mu}) = &\int_{\Gamma_{r_6}^1} \mu_2\mu_3\, q\, v\, d\Gamma_{r_6}^1 + \int_{\Gamma_{r_6}^2} \mu_1\mu_6\, q\, v\, d\Gamma_{r_6}^2 + \dots \\ &\dots + \int_{\Gamma_{r_6}^6} -\mu_5\,(\mu_3+\mu_6-3)\; q\, v\, d\Gamma_{r_6}^6 + \dots \\ &\dots + \theta_f^9(\boldsymbol{\mu}) \int_{\Gamma_{r_6}^9} q\, v\; d\Gamma_{r_6}^9. \end{aligned} \tag{6.13}$$

The affine decomposition is now clear and we have

$$\begin{aligned} a(u,v;\boldsymbol{\mu}) &= \sum_{1}^{81} \theta_a^q(\boldsymbol{\mu}) a^q(u,v), \\ f(v;\boldsymbol{\mu}) &= \sum_{1}^{9} \theta_f^q(\boldsymbol{\mu}) f^q(v), \end{aligned}$$

where the θ-*functions* are the parameters dependent terms which appear in the bilinear form 6.10 and in the linear functional 6.13 expressed in the reference domain.
Since the geometric parameter dependence is quite involved, we will present only few results from our set of theta functions $\theta_a^q(\boldsymbol{\mu})$, $1 \leqslant q \leqslant 81$, $\theta_f^q(\boldsymbol{\mu})$, $1 \leqslant q \leqslant 9$ in tables 6a and 6b. In the same tables we present also the definition of the $\boldsymbol{\mu}-$independent bilinear forms.

q	$\theta_a^q(\boldsymbol{\mu})$	$\underline{\underline{\mathbb{A}}}^q$
1	$\frac{\mu_2\mu_3}{\mu_1}$	$\int_{\Omega_r^1} \frac{\partial u}{\partial x_{r_1}} \frac{\partial v}{\partial x_{r_1}} \, d\Omega_r^1$
2	$\frac{\mu_1\mu_3}{\mu_2}$	$\int_{\Omega_r^1} \frac{\partial u}{\partial x_{r_2}} \frac{\partial v}{\partial x_{r_2}} \, d\Omega_r^1$
5	$\frac{\mu_1\mu_3}{\mu_2}$	$\int_{\Omega_r^1} \frac{\partial u}{\partial x_{r_2}} \frac{\partial v}{\partial x_{r_2}} \, d\Omega_r^1$
9	$\frac{\mu_2\mu_6}{\mu_1}$	$\int_{\Omega_r^2} \frac{\partial u}{\partial x_{r_2}} \frac{\partial v}{\partial x_{r_2}} \, d\Omega_r^2$
14	$\frac{\mu 1\mu_6}{\mu_5}$	$\int_{\Omega_r^5} \frac{\partial u}{\partial x_{r_2}} \frac{\partial v}{\partial x_{r_2}} \, d\Omega_r^5$
27	$\frac{\mu_1(\mu_2+\mu_5-3)}{\mu_3+\mu_6-3}$	$\int_{\Omega_r^9} \frac{\partial u}{\partial x_{r_3}} \frac{\partial v}{\partial x_{r_3}} \, d\Omega_r^9$
40	$\frac{\mu_5\mu_6\mu_7}{\mu_4}$	$\int_{\Omega_r^{14}} \frac{\partial u}{\partial x_{r_1}} \frac{\partial v}{\partial x_{r_1}} \, d\Omega_r^{14}$
60	$-\frac{\mu_2(\mu_1+\mu_4-3)}{\mu_6}$	$\int_{\Omega_r^{20}} \frac{\partial u}{\partial x_{r_3}} \frac{\partial v}{\partial x_{r_3}} \, d\Omega_r^{20}$
70	$\frac{\mu_5(\mu_3+\mu_6-3)}{\mu_1+\mu_4-3}$	$\int_{\Omega_r^{24}} \frac{\partial u}{\partial x_{r_1}} \frac{\partial v}{\partial x_{r_1}} \, d\Omega_r^{24}$
78	$\frac{(\mu_2+\mu_5-3)(\mu_1+\mu_4-3)}{\mu 6}$	$\int_{\Omega_r^{26}} \frac{\partial u}{\partial x_{r_3}} \frac{\partial v}{\partial x_{r_3}} \, d\Omega_r^{26}$

(a) TB $\theta_a^q(\boldsymbol{\mu})$-functions

q	$\theta_f^q(\boldsymbol{\mu})$	$\underline{\mathbb{F}}^q$
1	$\mu_1\mu_3$	$\int_{\Gamma_{r_6}^1} q\, v \, d\Gamma_{r_6}^1$
2	$\mu_1\mu_3$	$\int_{\Gamma_{r_6}^2} q\, v \, d\Gamma_{r_6}^2$
3	$\mu_2\mu_6$	$\int_{\Gamma_{r_6}^3} q\, v \, d\Gamma_{r_6}^3$
6	$-\mu_5(\mu_3+\mu_6-3)$	$\int_{\Gamma_{r_6}^6} q\, v \, d\Gamma_{r_6}^6$
9	$(\mu_2+\mu_5-3)(\mu_3+\mu_6-3)$	$\int_{\Gamma_{r_6}^9} q\, v \, d\Gamma_{r_6}^9$

(b) TB $\theta_a^q(\boldsymbol{\mu})$-functions

Table 6: *TB* $\theta(\boldsymbol{\mu})$-functions

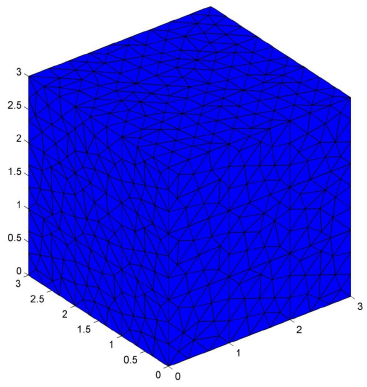
6.5 RESULTS AND VISUALIZATION

In this section we present the results obtained, linking `COMSOL` and `rbMIT`, in order to create a *RB* approximation for the *3D thermal block* example.
First we will give some informations about the *FE* approximation concerning the mesh, the basis function chosen and the entries needed by `COMSOL`, as discussed in Chapter 5.
Later on, we will focus on the results obtained with the *SCM* algorithm (Section 3.6, Chapter 3) for the error bounds calculations, then we will present the convergence of the *Greedy* procedure (Section 3.4).
Finally we will present the output evaluation for particular combinations of the parameters, along with the *certified a-posteriori error bound*, to prove that the *RB* approximation is *reliable* and *efficient*.

6.5.1 *FE* approximatiom with `COMSOL`

FE *discretization on reference domain*

We represent in Figure 16 the reference domain upon we assemble *FE* components (Chapter 5).
In the figure we also report the properties of the mesh and the basis functions chosen to discretize the *TB* problem. We recall that, since the maximum derivatives order that appears in the strong formulation of the *PDEs* is 2, then $\mathbb{P}^1$ basis function are still a good choice for the solution of the problem with the Galerkin method.
We remark that the degrees of freedom (*DOFs*) are *fewer* than the *mesh vertices* since we have imposed a *homogeneus* boundary condition on Γ_1. We have in fact eliminated *the rows and the columns* in the assembled matrices, corresponding to the nodes upon a *Dirichlet condition is imposed*, [Qua09].
The mesh has been generated by a `COMSOL` pre-process routine called `meshinit`, see [Com07a] for further explanations.

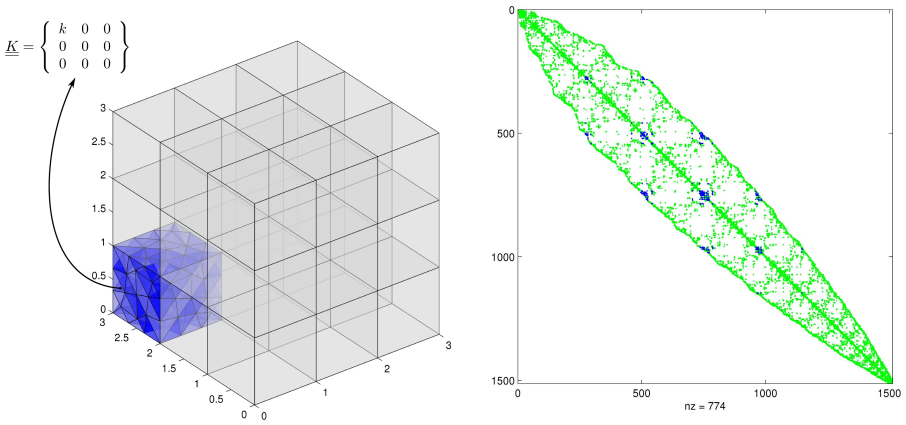


mesh vertices	1657
vertex	64
edges	432
triangle	3024
tetrahedral	7515
DOFs	1512
basis functions	$\mathbb{P}^1$

Figure 16: *TB* reference domain discretization

Matrices assembling

Following the procedure discussed in chapter 5 and in particular with regard to the notation introduced in equation 5.1, we assemble the parameter independent matrices (see table 6a and 6b) needed by the *RB* procedure.
In Figure 17 we depict a graphical view of the matrices assembling.
We consider the matrix $\underline{\underline{\mathbb{A}}}^{20}$, looking at Table 8 we note that the only subdomain that plays a role in the building of the parameter independent matrix is Ω_r^9, see Figure 17a. In Figure 17b we depict the matrix pattern.



(a) subdomain Ω_r^7 **(b)** $\underline{\underline{\mathbb{A}}}^{20}$

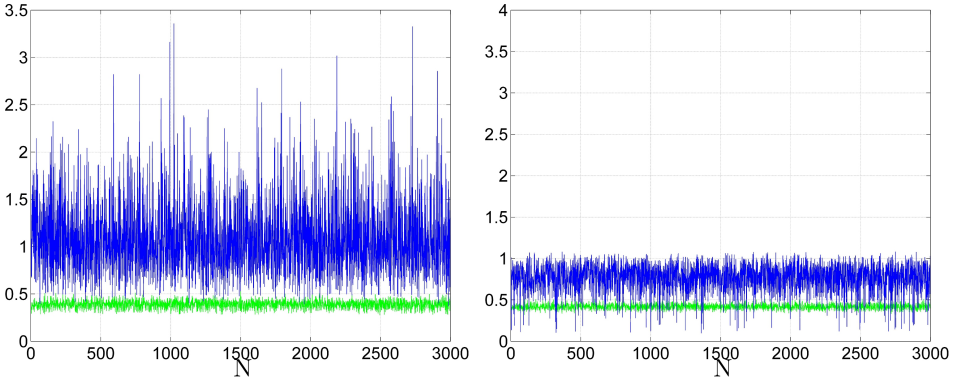
Figure 17: In figure 17b we have depicted the contributes of the local $\underline{\underline{\mathbb{A}}}^{20}$ matrix (•) to the global stiffness matrix (•).

6.5.2 *SCM* algorithm

For the *SCM* algorithm (section 3.6.3) we have took a sample train Ξ_{SCM} of size $n_{SCM} = 3000$, a tolerance $\epsilon_{SCM} = 0.7$, $M_\alpha = 16$, $M_+ = 0$ and $|\mathcal{P}| = 200$.
In figure 18a we show the α_{LB} (—) and the α_{UB} (—) for each element of the sample train Ξ_{SCM} of the first iteration $K = 1$, whereas in figure 18b we depict the same quantities for the last iteration $K = K_{max} = 4$ of the *SCM* algorithm. It is evident that the upper and lower bound for the parametric coercivity constant are converging to the exact value and restricting the possible gap between the lower and upper bound. Convergence for this problem is quite fast, see [Gel10, RHP08].

6.5.3 *Greedy* algorithm

We present the results for the *Greedy* algorithm (Section 3.4.2), during the *RB* aseembling procedure.
Here, we have chosen a sample train Ξ_{train} of size is $n_{train} = 3000$, the



(a) *TB*: First iteration of the *SCM* **(b)** *TB*: Last iteration of the *SCM*

Figure 18: *TB SCM* algorithm

tolerance is $\epsilon_{\text{toll,min}} = 0.9 \cdot 10^{-3}$ and the maximum size of the *RB* space is taken $N_{\max} = 100$. We have chosen to minimize the absolute error bound in energy norm $\Delta_N^{en}(\boldsymbol{\mu})$ (Section 3.5.1, page 52). In figure 20a we have represented the error bound $\Delta_N^{en}(\boldsymbol{\mu})$ for $1 \leqslant N \leqslant N_{\max}$. We can see that the error is monotonically decreasing. Moreover, just few basis ≈ 40 (versus $\approx$1500 *FE DOFs*) are needed to obtain a maximum error bound $\leqslant 10^{-2}$ on the temperature field for all the samples in Ξ_{train}. We remark that this result holds despite a large variation of either physical (the conductivity) and geometrical (the dimension and the position of the central block) parameters. In figure 20b we have depicted a subset of the parameters $\boldsymbol{\mu} = (\mu_2, \mu_5, \mu_6)$, automatically selected by the *Greedy algorithm* as representative snapshots. The error bounds help us to save also *Offline* computational cost since the evaluation of the error bounds during the *Greedy* procedure is ver inexpensive. It is possible to see that the algorithm often select parameters near to the bounds (upper and lower) of the parameters domain. In fact the more the parameters are chosen distant from the "center" of the set, the most the reference domain is deformed by means of the geometrical parameters. These phenomena will perforce increase the error bound, therefore the Greedy algorithm will preferably choose this outer parameters, being based on the *worst case scenario*.

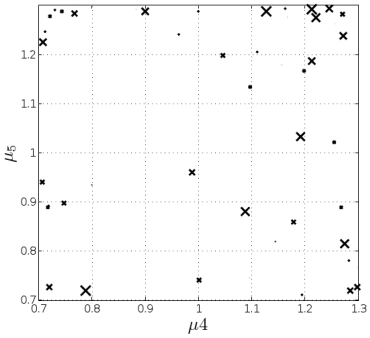
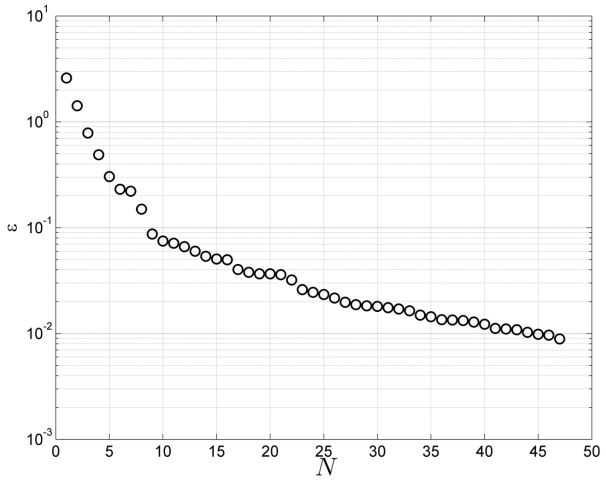
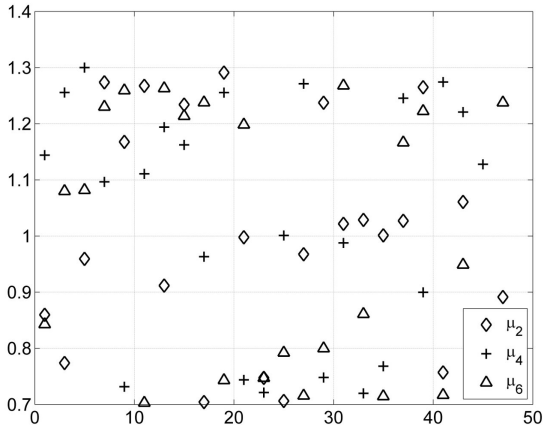


Figure 19: *Greedy* selection for parameter μ_4 and μ_5

This aspect can be better seen, looking at the figure 19, we can see that the geometrical parameters chosen are always in the outer part of the domain, where it is evident a *clustering phenomena*. In this figure, the dimension of the markers are proportional to the maximum error bound at the K-th Greedy iteration.



(a) Error bound $\Delta_N^{en}(\boldsymbol{\mu})$



(b) Parameters distribution

Figure 20: *TB* Greedy results

6.5.4 Output

Since the output is the average of the temperature on the face Γ_6 of the domain, then we are dealing with a compliant case (see Section 2.3.1). In fact we have that:

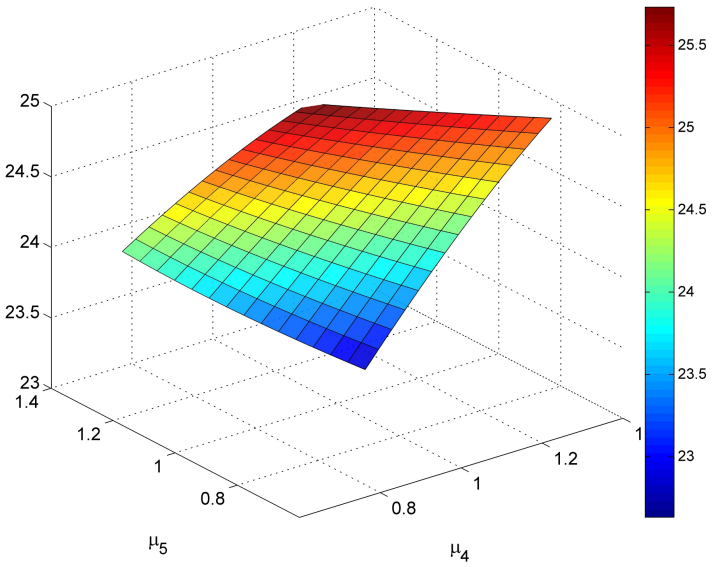
$$s_N(\boldsymbol{\mu}) = f(u_N; \boldsymbol{\mu}). \tag{6.14}$$

Since we have 7 parameters, we decided to fix some parameters and add relationship between others to obtain a graphical visualization of the output.
In particular we have chosen to vary the parameters μ_5 and μ_4, i.e. the

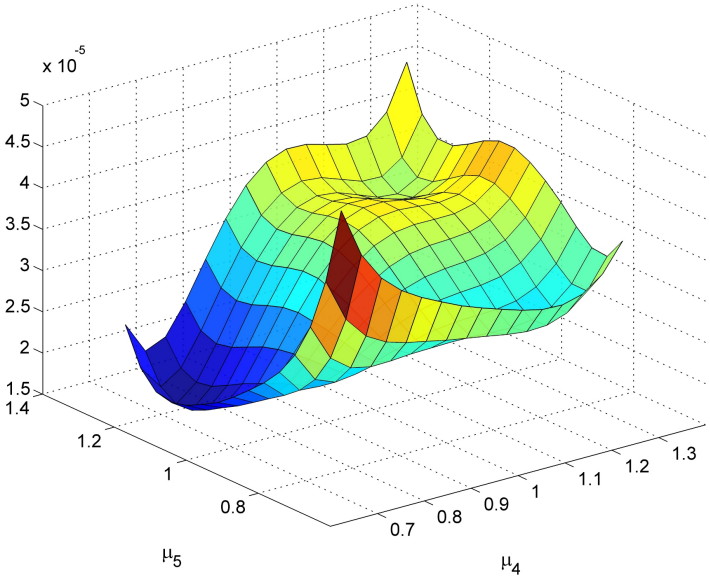
x and y dimension of the inclusion. In addition we have introduced the following relationships:

$$\begin{aligned}\mu_1 &= \frac{3-\mu_4}{2};\\ \mu_2 &= \frac{3-\mu_5}{2};\\ \mu_3 &= 1;\\ \mu_6 &= 1;\\ \mu_7 &= 1.\end{aligned}$$

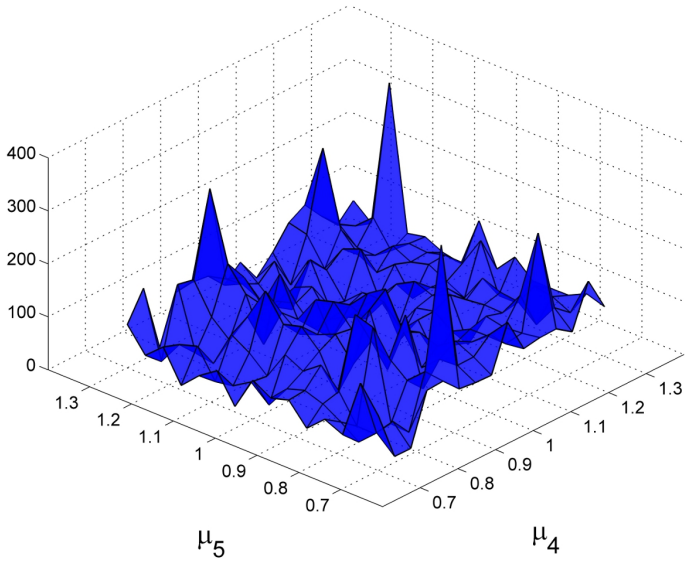
In Figure 21a we depict the tenmperature output obtained with the *RB* method, in Figure 21b we depict the *error bound* $\Delta_N^s(\boldsymbol{\mu})$ on the output between the *FE* and the *RB* method.
We can see that the output estimated error is $\leqslant 8 \cdot 10^{-5}$, hence the error on the output is effectively bounded by the square of the error on the solution field (we recall that in the Greedy we have set $\Delta_N^{en}(\boldsymbol{\mu}) \leqslant \epsilon_{toll,min} = 0.9 \cdot 10^{-3}$, $\forall \boldsymbol{\mu} \in \Xi_{train}$).
We remark that this result follows from our assumption of compliance, enabling the so called *square effect* (see equation 3.24, chapter 3.2).
In figure 21c we have depicted the ratio between the computational time needed to evaluate the output in the *FE* (denoted with $t_{FE}^s(\boldsymbol{\mu})$) and in the *RB* case ($t_{RB}^s(\boldsymbol{\mu})$) for a large test sample. We can see that the *RB* method provides a computational time saving of *two order of magnitude* with respect to the ordinary *FE* method.



(a) *TB* output $s(\boldsymbol{\mu})$



(b) *TB* error bound $\Delta_N^s(\boldsymbol{\mu})$

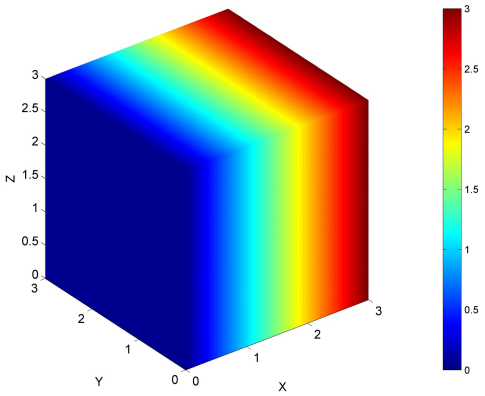


(c) $\frac{t_{FE}^s(\boldsymbol{\mu})}{t_{RB}^s(\boldsymbol{\mu})}$

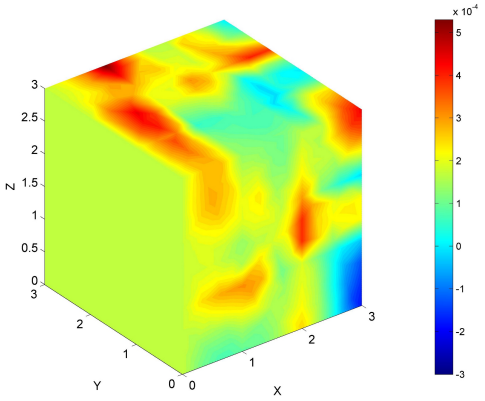
Figure 21: *TB* output results

6.5.5 Visualization

We now report the visualization of some representative *RB* solutions. On the upper figures, we show the solution for different value of the parameters $\boldsymbol{\mu}$. On the lower figures, we represent the pointwise error between the *RB* approximations and the *FE* solution.
In the first example, Figure 22, we show the solution on the reference domain. In the second example, Figure 23, we show the solution field after selecting a generic combination of parameters in the parameter domain $\mathcal{D}$. In the first case, thanks to the absence of geometrical distortion, we obtain a *smaller error bound* on the solution.

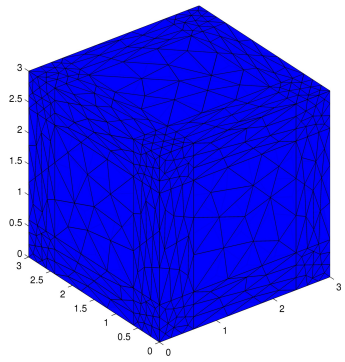


(a) Solution field: $\Delta_N^{en}(\boldsymbol{\mu}) \leqslant 2.3 \cdot 10^{-3}$

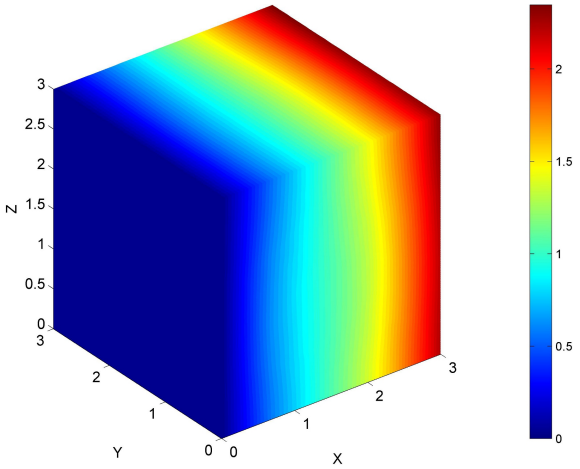


(b) Pointwise error: $u_N(\boldsymbol{\mu}) - u^{\mathcal{N}}(\boldsymbol{\mu})$

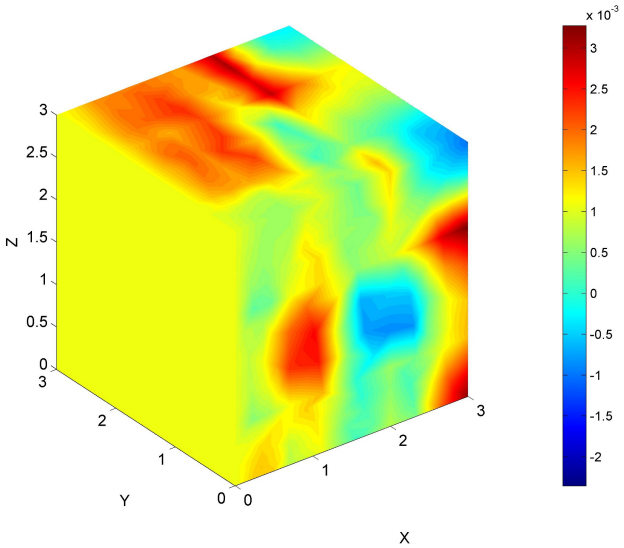
Figure 22: Example of representative solution for the *TB* problem and pointwise error for $\boldsymbol{\mu} = \{1, 1, 1, 1, 1, 1, 1\}$



(a) Deformed domain



(b) Error bound: $\Delta_N^{en}(\mu) \leqslant 6.3 \cdot 10^{-3}$



(c) Pointwise error: $u_N(\mu) - u^{\mathcal{N}}(\mu)$

Figure 23: Example of representative solution and pointwise error for $\mu = \{0.7, 0.7, 0.7, 1.3, 1.3, 1.3, 0.05\}$

7 ELASTIC PROBLEM

7.1 INTRODUCTION

In this chapter we will exploit the creation of the *RB* approximation *in the 3D case* dealing with a *vectorial elliptic coercive* problem.
We consider a *static stress application* represented by a *linear elasticity problem*, where the displacement is given by:

$$\underline{\mathbf{u}}(\mathbf{x};\boldsymbol{\mu}) = \left\{ \begin{array}{c} u \\ v \\ w \end{array} \right\}.$$

The *elastic tensor* $\mathcal{C}_{ijkl}$ is represented with the *Voigt notation*, by a positive definite matrix $\underline{\underline{C}} \in \mathbb{R}^{6\times 6}$, see [DLoo].
This case will be useful to show one of the most interesting field of application for the *RB* method. In particular in section 7.6 we will show that the *Offline/Online* splitting, may allow an efficient structural optimization that combines the accuracy of the *FE* to the cheapness of the *RB* methodology.

7.2 PROBLEM DESCRIPTION

We now briefly introduce the *linear elasticity problem* (in a cubic domain), referring to this case as the EB (elastic block) problem.

Physical problem

The *linear elasticity* is the study of how *solid objects* deform and become *internally stressed* due to prescribed *loading conditions*.
In particular we will consider a solid block formed by three superposed layer of different isotropic materials, pulled by an axial load applied on a face and clamped to the opposite one.
Our aim is to evaluate the *average displacement* of the *loaded face* consequently to the variation of the thickness, or the material composition of the three layers. We are still in a *compliant case* (Section 2.3.1 in Chapter 2.3), therefore we inherit all the good convergence properties already discussed (see Chapter 3.2).
This case is geometrically simple, nonetheless the methodology and the procedure discussed is comprehensive and provide the tools to build more involved and specialized problems.

Analytical problem

In the *EB* problem, the displacement field is governed by an *equilibrium equation* supplied with a *constitutive relationship*, i.e. *Hooke's law*, see [Lov44, DLoo]. We consider a static stress problem, hence no inertial nor dissipative terms are accounted, in addition we will not consider for simplicity volume forces.

7.3 PARAMETERS CHOICE

We introduce two parameters $\boldsymbol{\mu}_{\text{geom}} = \{\mu_1, \mu_2\}$ to describe the thickness of each layer, as depicted in Figure 24. Moreover since each layer can be made by different material, or may had been subjected to different technological process, also the *Young modulus* E is taken as a parameter, then $\boldsymbol{\mu}_{\text{physic}} = \{\mu_3, \mu_4, \mu_5\} (\equiv \{E_1, E_2, E_3\})$.
For simplicity, we have chosen to deal with isotropic material. This assumption is quite restrictive, especially in the aeronautic field where we possibly deal with composite material; the extension to the ortotropic case, which can be used to model these particular materials, is studied in [MQRo8].

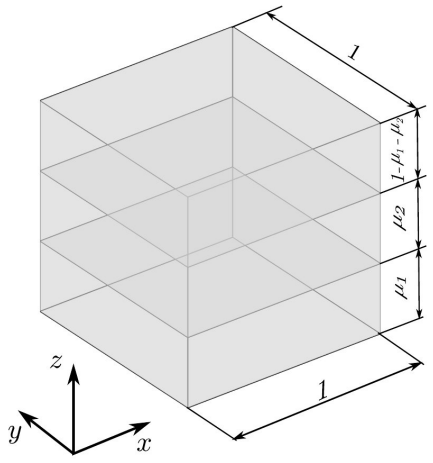


Figure 24: EB domain decomposition

Parameter domain

We now summarize the parameters and the parameter ranges chosen to describe our *EB* problem:

- the domain for the geometrical parameters, representing the thickness of the layers, is:

$$\begin{aligned}\mathcal{D}_{\text{geom}} &= \left[\mu_1^{\min}, \mu_1^{\max}\right] \times \left[\mu_2^{\min}, \mu_2^{\max}\right] \\ &= [0.1, 0.45] \times [0.1, 0.45]\,, \end{aligned} \tag{7.1}$$

- The domain for the physical parameters, representing the *Young modula* of the three layers:

$$\begin{aligned}\mathcal{D}_{\text{physics}} &= \left[\mu_1^{\min}, \mu_1^{\max}\right] \times \left[\mu_2^{\min}, \mu_2^{\max}\right] \times \left[\mu_3^{\min}, \mu_3^{\max}\right] \\ &= [0.1, 3] \times [0.1, 3] \times [0.1, 3]\,, \end{aligned} \tag{7.2}$$

The total parameter domain is therefore:

$$\begin{aligned}\mathcal{D} &= \mathcal{D}_{\text{geom}} \times \mathcal{D}_{\text{physics}} \\ &= [0.1, 0.45] \times [0.1, 0.45] \times [0.1, 10] \times [0.1, 3] \times [0.1, 3] \times [0.1, 3]\,. \end{aligned}$$

Boundary conditions

With regard to the boundary conditions (Figure 25), an non-homogeneus Neumann boundary condition, which represent a distributed load p along the x direction, is imposed on Γ_6. In addition an homogeneus Dirichlet boundary condition, which represents the clamping, is imposed on Γ_6, whereas on the other external faces of the cube $\Gamma_{2:5}$, on which we assume a *free stress condition*, homogeneus Neumann conditions has been set. Finally, on the internal faces we assume the continuity of the displacements and of stresses.

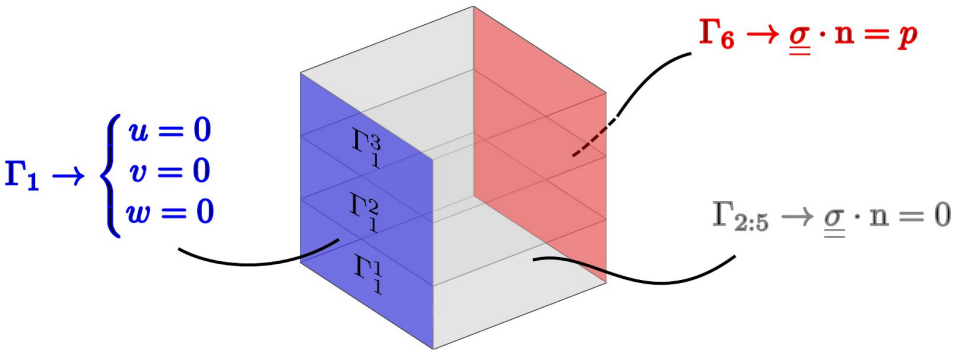


Figure 25: EB boundary conditions

7.4 EB PROBLEM FORMULATION

7.4.1 Original domain

In this section the analytical formulation of the governing *PDEs* on the *original domain* will be introduced. The equation and the boundary conditions that describes the field of displacement, within the hypothesis described in Section 7.1, is the following [Lov44, DLoo]:

$$\begin{cases} \frac{\partial \sigma_{o_{ij}}}{\partial x_{o_j}} + b_{o_i} = 0 & \text{in } \Omega_o(\boldsymbol{\mu}) \\ \underline{\mathbf{u}} = \underline{0} & \text{on } \Gamma_{o_1}(\boldsymbol{\mu}) \\ \underline{\underline{\sigma}}_o \cdot \mathbf{n} = \underline{p} = \begin{Bmatrix} p_1 \\ 0 \\ 0 \end{Bmatrix} & \text{on } \Gamma_{o_6}(\boldsymbol{\mu}) \end{cases} \tag{7.3}$$

Here $\sigma_{o_{ij}}$ is given by the *constitutive Hooke's law*, [Lov44]:

$$\sigma_{o_{ij}} = \mathcal{C}_{o_{ijkl}} \, \varepsilon_{o_{kl}}, \tag{7.4}$$

where:

- $\mathcal{C}_{o_{ijkl}}$ is the *elastic tensor* and for isotropic materials is defined as

  $$\mathcal{C}_{o_{ijkl}} = \lambda \delta_{ij} \delta_{kl} + G \left(\delta_{ik} \delta_{jl} \right),$$

  λ and G are the *Lamè constants* given by:

  $$\begin{cases} \lambda = \dfrac{E\nu}{(1+\nu)(1-2\nu)} \\ G = \dfrac{E}{2(1+\nu)} \end{cases} \tag{7.5}$$

  where E is the *Young modulus* and ν the *Poisson coefficient*.

- $\varepsilon_{o_{kl}}$ is the *linearized deformation*, given by

  $$\varepsilon_{o_{kl}} = \frac{1}{2} \left(\frac{\partial u_{o_k}}{\partial x_{o_l}} + \frac{\partial u_{o_l}}{\partial x_{o_k}} \right), \tag{7.6}$$

We now give the matrix representation $\underline{\underline{C}}$ of the *elastic tensor*, which shall prove useful for the next

$$\underline{\underline{C}} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \tag{7.7}$$

Now we derive the weak formulation for the *EB* problem.
We recall that the domain $\Omega_o(\boldsymbol{\mu})$ is made up by different subdomains of different materials, such that:

$$\Omega_o(\boldsymbol{\mu}) = \bigcup_{m=1}^{3} \Omega_o^m(\boldsymbol{\mu}),$$

furthermore each boundary $\Gamma_{o_m}(\boldsymbol{\mu})$, $1 \leqslant m \leqslant 6$ is made by different faces; for example $\Gamma_{o_6}(\boldsymbol{\mu})$ is defined as

$$\Gamma o_6(\boldsymbol{\mu}) = \bigcup_{m=1}^{3} \Gamma_{o_6}^m(\boldsymbol{\mu}).$$

The decomposition of the boundary shall prove useful in the affine decomposition of the linear functional. We now introduce a functional space $X^e = \left\{ v \in \left(H^1(\Omega_o)\right)^3 \middle| \, v = 0 \text{ on } \Gamma_{o_1} \right\}$.
Multiplying the equation 7.3 for the test function $v \in X^e$, integrating on $\Omega_o(\boldsymbol{\mu})$, using the *divergence theorem* and applying boundary conditions we get the *weak formulation* as follows, [DL00, QV97]:

$$\sum_{m=1}^{3} \int_{\Omega_o^m(\boldsymbol{\mu})} \frac{\partial v_i}{\partial x_{o_j}} \sigma_{o_{ij}} \, d\Omega_o^m(\boldsymbol{\mu}) = \sum_{m=1}^{3} \int_{\Gamma_{o_6}^m(\boldsymbol{\mu})} v_i \, p_i \, d\Gamma_{o_6}^m(\boldsymbol{\mu}). \tag{7.8}$$

Introducing the parametric bilinear form

$$\begin{aligned} a(u, v; \boldsymbol{\mu}) &= \sum_{m=1}^{3} \int_{\Omega_o^m(\boldsymbol{\mu})} \frac{\partial v_i}{\partial x_{o_j}} \sigma_{o_{ij}} \, d\Omega_o^m(\boldsymbol{\mu}) \\ &= \sum_{m=1}^{3} \int_{\Omega_o^m(\boldsymbol{\mu})} \frac{\partial v_i}{\partial x_{o_j}} \mathcal{C}_{o_{ijkl}}^m \frac{\partial w_j}{\partial x_{m_l}} \, d\Omega_o^m(\boldsymbol{\mu}), \end{aligned} \tag{7.9}$$

and the parametric linear functional

$$f(v; \boldsymbol{\mu}) = \sum_{m=1}^{3} \int_{\Gamma_{o_6}^m(\boldsymbol{\mu})} v_i \, p_i \, d\Gamma_{o_6}^m(\boldsymbol{\mu}), \tag{7.10}$$

we can restate the problem 7.3 as: find $u \in X^e(\Omega_o(\boldsymbol{\mu}))$, such that:

$$a(u, v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \forall \, v \in X^e(\Omega_o(\boldsymbol{\mu})). \tag{7.11}$$

The coercivity and the continuity of the bilinear form a and the continuity of the functional f can be proved. Then the *Lax-Milgram* theorem ensures the existence and uniqueness of the solution [Qua09].

7.4.2 Reference domain

We now apply standard techniques to transform the problem statement over the *original domain* to an equivalent problem set on the *reference domain*.
We shall be ultimately able to write our problem in an *affine form* (2.20), to exploit our crucial *Offline/Online* splitting procedure. As already seen in the *TB* problem, to perform this transformation we need to evaluate the affine mapping for each subdomain. We summarize the result for the geometrical transformation on the reference domain, in table 7.

sub	G_{11}^{aff}	G_{22}^{aff}	G_{33}^{aff}	C_1^{aff}	C_2^{aff}	C_3^{aff}
Ω^1	1	1	$1/\mu_1$	0	0	0
Ω^2	1	1	$1/\mu_2$	0	0	μ_1
Ω^3	1	1	$1/(1-\mu_1-\mu_2)$	0	0	$\mu_1+\mu_2$

Table 7: EB affine mappings

Furthermore we get:

$$d\Omega_o^m(\boldsymbol{\mu}) = \det\left(\underline{\underline{G}}^{aff,m}(\boldsymbol{\mu})\right) d\Omega_r^m \quad 1 \leqslant m \leqslant 3, \tag{7.12}$$

$$d\Gamma_{o_6}^m(\boldsymbol{\mu}) = \left|\underline{\underline{G}}^{aff,m}(\boldsymbol{\mu}) \cdot \underline{e}_t^m\right| d\Gamma_r^m \quad 1 \leqslant m \leqslant 3, \tag{7.13}$$

where $\underline{e}_t^m$ is the tangential unit vector of face m and where (see [MQR08]):

$$\left|\underline{\underline{G}}^{aff}(\boldsymbol{\mu}) \cdot \underline{e}_t\right| = \left(\sum_{i=1}^{d=3}\left(G_{ij}^{aff}\, e_{t_i}\right)^2\right)$$

Once this quantities have been evaluated, we are able to rewrite the weak formulation 6.9 into the reference domain.

Bilinear form

In the *EB* case, since we are dealing with a *vectorial* problem, the procedure to decompose the parametric bilinear form onto the *reference domain*

into an affine development is quite involved, especially in the 3D case. Recalling the bilinear form defined in 7.9, it follows:

$$
\begin{aligned}
a(u,v;\mu) &= \sum_{m=1}^{3} \int_{\Omega_o^m(\mu)} \frac{\partial v_i}{\partial x_{o_j}} \mathcal{C}_{o_{ijkl}}{}^m \frac{\partial w_j}{\partial x_{o_l}} \, d\Omega_o^m(\mu) \\
&= \sum_{m=1}^{3} \int_{\Omega_r^m} \left(G_{jj'}^{aff,m}(\mu) \frac{\partial w_i}{\partial x_{r_j}} \right) \mathcal{C}^m_{o_{ij'kl'}}(\mu) \left(G_{ll'}^{aff,r}(\mu) \frac{\partial v_k}{\partial x_{r_l}} \right) \left(J^{aff,m}(\mu) \right)^{-1} d\Omega_r^m \\
&= \frac{1}{(1+\nu)(1-2\nu)} \left(3\mu_1\mu_3 \int_{\Omega_r^1} (1-\nu) \left(\frac{\partial v_1}{\partial x_{r_1}} \frac{\partial w_1}{\partial x_{r_1}} + \frac{\partial v_1}{\partial x_{r_2}} \frac{\partial w_1}{\partial x_{r_2}} \right) + \right. \\
&+ \left(\frac{1-2\nu}{2} \right) \left(\frac{\partial v_2}{\partial x_{r_1}} \frac{\partial w_2}{\partial x_{r_1}} + \frac{\partial v_1}{\partial x_{r_2}} \frac{\partial w_1}{\partial x_{r_2}} + \frac{\partial v_3}{\partial x_{r_2}} \frac{\partial w_3}{\partial x_{r_2}} + \frac{\partial v_3}{\partial x_{r_1}} \frac{\partial w_3}{\partial x_{r_1}} \right) + \\
&+ \nu \left(\frac{\partial v_1}{\partial x_{r_1}} \frac{\partial w_2}{\partial x_{r_2}} + \frac{\partial v_1}{\partial x_{r_2}} \frac{\partial w_2}{\partial x_{r_1}} + \frac{\partial v_2}{\partial x_{r_2}} \frac{\partial w_1}{\partial x_{r_1}} + \frac{\partial v_2}{\partial x_{r_1}} \frac{\partial w_1}{\partial x_{r_2}} \right) d\Omega_r^1 + \\
&+ \frac{\mu_3}{3\mu_1} \int_{\Omega_r^1} (1-\nu) \frac{\partial v_3}{\partial x_{r_3}} \frac{\partial w_3}{\partial x_{r_3}} + \left(\frac{1-2\nu}{2} \right) \left(\frac{\partial v_1}{\partial x_{r_3}} \frac{\partial w_1}{\partial x_{r_3}} + \frac{\partial v_2}{\partial x_{r_3}} \frac{\partial w_2}{\partial x_{r_3}} \right) d\Omega_r^1 + \\
&+ \mu_3 \int_{\Omega_r^1} \nu \left(\frac{\partial v_1}{\partial x_{r_1}} \frac{\partial w_3}{\partial x_{r_3}} + \frac{\partial v_1}{\partial x_{r_3}} \frac{\partial w_3}{\partial x_{r_1}} + \frac{\partial v_2}{\partial x_{r_2}} \frac{\partial w_3}{\partial x_{r_3}} + \frac{\partial v_2}{\partial x_{r_3}} \frac{\partial w_3}{\partial x_{r_2}} \right) d\Omega_r^1 + \dots \\
&\dots + \theta_a^9(\mu) \int_{\Omega_r^3} \nu \left(\frac{\partial v_1}{\partial x_{r_1}} \frac{\partial w_3}{\partial x_{r_3}} + \frac{\partial v_1}{\partial x_{r_3}} \frac{\partial w_3}{\partial x_{r_1}} + \frac{\partial v_2}{\partial x_{r_2}} \frac{\partial w_3}{\partial x_{r_3}} + \frac{\partial v_2}{\partial x_{r_3}} \frac{\partial w_3}{\partial x_{r_2}} \right)
\end{aligned}
\tag{7.14}
$$

Linear functional

As already seen in the *TB* case, in this case the *RHS* arises from the *non-homogeneus Neumann condition* (the distributed axial load) applied on the face Γ_6.
Considering the relationship 7.13 and recalling that Γ_6 is made by three different subfaces to which correspond three different geometric transformations, starting from the linear functional 7.17 we get

$$
\begin{aligned}
f(v;\mu) &= \sum_{m=1}^{3} \int_{\Gamma_{o_6}^m(\mu)} v_i \, p_i \, d\Gamma_{o_6}^m(\mu) \\
&= 3\mu_1 \int_{\Gamma_{r_6}^1} v_1 \, p_1 \, d\Gamma_{r_6}^1 + 3\mu_2 \int_{\Gamma_{r_6}^2} v_1 \, p_1 \, d\Gamma_{r_6}^2 + \int_{\Gamma_{r_6}^3} v_1 \, p_1 \, d\Gamma_{r_6}^3
\end{aligned}
\tag{7.15}
$$

In Table 8 and 9 we present the θ-*functions* for the bilinear form and for the linear functional respectively.
We remark that, thanks to the choice of a particularly simple geometry, the terms in the affine development are rather few.

q	$\theta_a^q(\boldsymbol{\mu})$	$\underline{\underline{\mathbb{A}}}^q$
1	$3\mu_1\mu_3$	$\frac{1}{(1+\nu)(1-2\nu)}\left(\int_{\Omega_r^1}(1-\nu)\left(\frac{\partial v_1}{\partial x_{r_1}}\frac{\partial w_1}{\partial x_{r_1}}+\frac{\partial v_1}{\partial x_{r_2}}\frac{\partial w_1}{\partial x_{r_2}}\right)+ +\left(\frac{1-2\nu}{2}\right)\left(\frac{\partial v_2}{\partial x_{r_1}}\frac{\partial w_2}{\partial x_{r_1}}+\frac{\partial v_1}{\partial x_{r_2}}\frac{\partial w_1}{\partial x_{r_2}}+\frac{\partial v_3}{\partial x_{r_2}}\frac{\partial w_3}{\partial x_{r_2}}+\frac{\partial v_3}{\partial x_{r_1}}\frac{\partial w_3}{\partial x_{r_1}}\right) +\nu\left(\frac{\partial v_1}{\partial x_{r_1}}\frac{\partial w_2}{\partial x_{r_2}}+\frac{\partial v_1}{\partial x_{r_2}}\frac{\partial w_2}{\partial x_{r_1}}+\frac{\partial v_2}{\partial x_{r_2}}\frac{\partial w_1}{\partial x_{r_1}}+\frac{\partial v_2}{\partial x_{r_1}}\frac{\partial w_1}{\partial x_{r_2}}\right)\mathrm{d}\Omega_r^1$
2	$\frac{\mu_3}{3\mu_1}$	$\frac{1}{(1+\nu)(1-2\nu)}\int_{\Omega_r^1}(1-\nu)\frac{\partial v_3}{\partial x_{r_3}}\frac{\partial w_3}{\partial x_{r_3}}+\left(\frac{1-2\nu}{2}\right)\left(\frac{\partial v_1}{\partial x_{r_3}}\frac{\partial w_1}{\partial x_{r_3}}+ +\frac{\partial v_2}{\partial x_{r_3}}\frac{\partial w_2}{\partial x_{r_3}}\right)\mathrm{d}\Omega_r^1$
3	μ_3	$\frac{1}{(1+\nu)(1-2\nu)}\left(\int_{\Omega_r^1}\nu\left(\frac{\partial v_1}{\partial x_{r_1}}\frac{\partial w_3}{\partial x_{r_3}}+\frac{\partial v_1}{\partial x_{r_3}}\frac{\partial w_3}{\partial x_{r_1}}+\frac{\partial v_2}{\partial x_{r_2}}\frac{\partial w_3}{\partial x_{r_3}}+ \frac{\partial v_2}{\partial x_{r_3}}\frac{\partial w_3}{\partial x_{r_2}}\right)\mathrm{d}\Omega_r^1\right)$
4	$3\mu_2\mu_4$	$\frac{1}{(1+\nu)(1-2\nu)}\left(\int_{\Omega_r^2}(1-\nu)\left(\frac{\partial v_1}{\partial x_{r_1}}\frac{\partial w_1}{\partial x_{r_1}}+\frac{\partial v_1}{\partial x_{r_2}}\frac{\partial w_1}{\partial x_{r_2}}\right)+ +\left(\frac{1-2\nu}{2}\right)\left(\frac{\partial v_2}{\partial x_{r_1}}\frac{\partial w_2}{\partial x_{r_1}}+\frac{\partial v_1}{\partial x_{r_2}}\frac{\partial w_1}{\partial x_{r_2}}+\frac{\partial v_3}{\partial x_{r_2}}\frac{\partial w_3}{\partial x_{r_2}}+\frac{\partial v_3}{\partial x_{r_1}}\frac{\partial w_3}{\partial x_{r_1}}\right) +\nu\left(\frac{\partial v_1}{\partial x_{r_1}}\frac{\partial w_2}{\partial x_{r_2}}+\frac{\partial v_1}{\partial x_{r_2}}\frac{\partial w_2}{\partial x_{r_1}}+\frac{\partial v_2}{\partial x_{r_2}}\frac{\partial w_1}{\partial x_{r_1}}+\frac{\partial v_2}{\partial x_{r_1}}\frac{\partial w_1}{\partial x_{r_2}}\right)\mathrm{d}\Omega_r^2\right)$
5	$\frac{\mu_4}{3\mu_2}$	$\frac{1}{(1+\nu)(1-2\nu)}\int_{\Omega_r^2}(1-\nu)\frac{\partial v_3}{\partial x_{r_3}}\frac{\partial w_3}{\partial x_{r_3}}+\left(\frac{1-2\nu}{2}\right)\left(\frac{\partial v_1}{\partial x_{r_3}}\frac{\partial w_1}{\partial x_{r_3}}+ +\frac{\partial v_2}{\partial x_{r_3}}\frac{\partial w_2}{\partial x_{r_3}}\right)\mathrm{d}\Omega_r^2$
6	μ_4	$\frac{1}{(1+\nu)(1-2\nu)}\left(\int_{\Omega_r^2}\nu\left(\frac{\partial v_1}{\partial x_{r_1}}\frac{\partial w_3}{\partial x_{r_3}}+\frac{\partial v_1}{\partial x_{r_3}}\frac{\partial w_3}{\partial x_{r_1}}+\frac{\partial v_2}{\partial x_{r_2}}\frac{\partial w_3}{\partial x_{r_3}}+ \frac{\partial v_2}{\partial x_{r_3}}\frac{\partial w_3}{\partial x_{r_2}}\right)\mathrm{d}\Omega_r^2\right)$
7	$3\mu_2\mu_4$	$\frac{1}{(1+\nu)(1-2\nu)}\left(\int_{\Omega_r^3}(1-\nu)\left(\frac{\partial v_1}{\partial x_{r_1}}\frac{\partial w_1}{\partial x_{r_1}}+\frac{\partial v_1}{\partial x_{r_2}}\frac{\partial w_1}{\partial x_{r_2}}\right)+ +\left(\frac{1-2\nu}{2}\right)\left(\frac{\partial v_2}{\partial x_{r_1}}\frac{\partial w_2}{\partial x_{r_1}}+\frac{\partial v_1}{\partial x_{r_2}}\frac{\partial w_1}{\partial x_{r_2}}+\frac{\partial v_3}{\partial x_{r_2}}\frac{\partial w_3}{\partial x_{r_2}}+\frac{\partial v_3}{\partial x_{r_1}}\frac{\partial w_3}{\partial x_{r_1}}\right) +\nu\left(\frac{\partial v_1}{\partial x_{r_1}}\frac{\partial w_2}{\partial x_{r_2}}+\frac{\partial v_1}{\partial x_{r_2}}\frac{\partial w_2}{\partial x_{r_1}}+\frac{\partial v_2}{\partial x_{r_2}}\frac{\partial w_1}{\partial x_{r_1}}+\frac{\partial v_2}{\partial x_{r_1}}\frac{\partial w_1}{\partial x_{r_2}}\right)\mathrm{d}\Omega_r^3\right)$
8	$\frac{\mu_4}{3\mu_2}$	$\frac{1}{(1+\nu)(1-2\nu)}\int_{\Omega_r^3}(1-\nu)\frac{\partial v_3}{\partial x_{r_3}}\frac{\partial w_3}{\partial x_{r_3}}+\left(\frac{1-2\nu}{2}\right)\left(\frac{\partial v_1}{\partial x_{r_3}}\frac{\partial w_1}{\partial x_{r_3}}+ + \frac{\partial v_2}{\partial x_{r_3}}\frac{\partial w_2}{\partial x_{r_3}}\right)\mathrm{d}\Omega_r^3$
9	μ_4	$\frac{1}{(1+\nu)(1-2\nu)}\left(\int_{\Omega_r^3}\nu\left(\frac{\partial v_1}{\partial x_{r_1}}\frac{\partial w_3}{\partial x_{r_3}}+\frac{\partial v_1}{\partial x_{r_3}}\frac{\partial w_3}{\partial x_{r_1}}+\frac{\partial v_2}{\partial x_{r_2}}\frac{\partial w_3}{\partial x_{r_3}}+ \frac{\partial v_2}{\partial x_{r_3}}\frac{\partial w_3}{\partial x_{r_2}}\right)\mathrm{d}\Omega_r^3\right)$

Table 8: EB $\theta_a^q(\boldsymbol{\mu})$−functions

q	$\theta_f^q(\boldsymbol{\mu})$	$\underline{\mathbb{F}}^q$
1	$3\mu_1\mu_3$	$\int_{\Gamma^1_{r_6}} \nu_1\, p_1\, d\Gamma^1_{r_6}$
2	$3\mu_2\mu_3$	$\int_{\Gamma^2_{r_6}} \nu_1\, p_1\, d\Gamma^2_{r_6}$
3	$3-3\mu_2-3\mu_1$	$\int_{\Gamma^3_{r_6}} \nu_1\, p_1\, d\Gamma^3_{r_6}$

Table 9: *EB* $\theta_f^q(\boldsymbol{\mu})$—functions

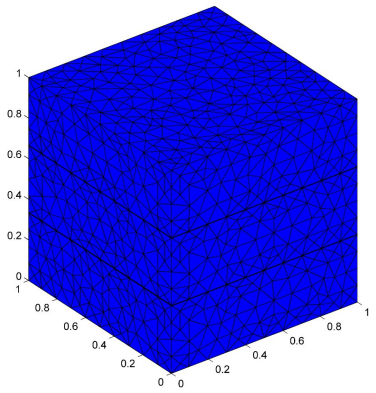
7.5 RESULTS AND VISUALIZATION

We present now some results obtained for the *3D elastic block* case.
First we give some informations about the *FE* approximation concerning the mesh, the basis function chosen and the entries needed by COMSOL, as discussed in Chapter 5.
Then, we focus on the results obtained with the *SCM* algorithm (Section 3.6, Chapter 3) and we focus on the convergence of the *Greedy* procedure (Section 3.4). We present an example for the output evaluation, the average displacement of the loaded face as function of either geometrical and physical parameters.
In the last section we will present a *RB* application in the engineering design; in particular we will deal with a simple optimization problem solved via the classical *FE* method and via the efficient *RB* method.

7.5.1 *FE* approximatiom with COMSOL

FE *discretization on reference domain*

We depict in Figure 26 the reference domain over which we assemble our *FE* components (section 5).
The mesh has been generated by COMSOL. In figure 26 we summarize the main features of the mesh along with the *FE* basis functions and the *DOFs* for the *EB* problem.

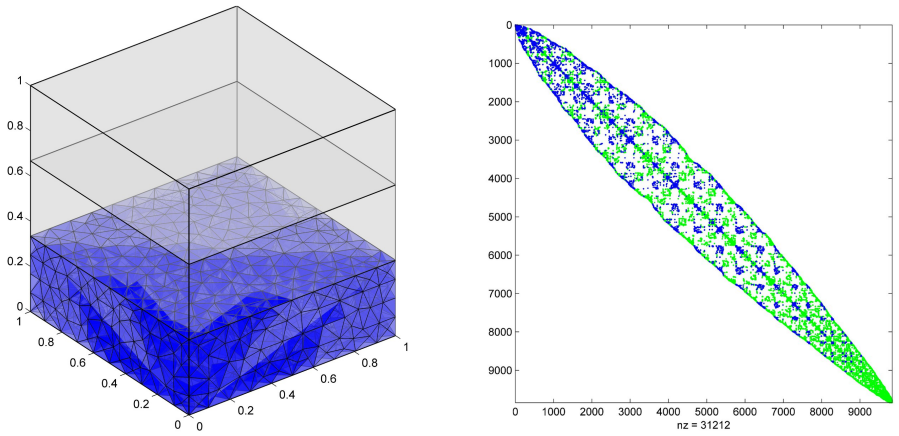


mesh vertices	3503
vertex	16
edges	208
triangle	3178
tetrahedral	17234
DOFs	10509
basis functions	$\mathbb{P}^1$

Figure 26: EB reference domain

Matrices assembling

We assemble the stiffness matrices $\underline{\underline{\mathbb{A}}}^q \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$, $1 \leqslant q \leqslant 9$ related to the *affine terms* reported in Table 8; then we form the *RHS* vectors $\underline{\mathbb{F}}^q \in \mathbb{R}^{\mathcal{N}\times 1}$, $1 \leqslant q \leqslant 3$ related to the face Γ_{r_6} on which we have imposed a unitary distributed load $\underline{p}$, as reported in Table 9.

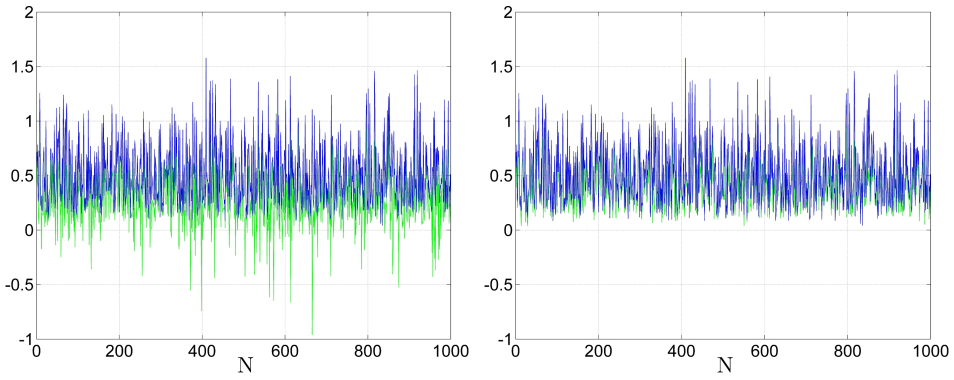


(a) subdomain Ω_r^1 **(b)** $\underline{\underline{\mathbb{A}}}^3$

Figure 27: In figure 27b we have depicted for example the contributes of the local $\underline{\underline{\mathbb{A}}}^3$ matrix (•) to the global stiffness matrix (•).

7.5.2 *SCM* algorithm

For the *SCM* algorithm (section 3.6.3) we have took a sample train Ξ_{SCM} of size $n_{SCM} = 1000$, a tolerance $\epsilon_{SCM} = 0.8$, $M_\alpha = 16$, $M_+ = 2$ and $|\mathcal{P}| = 300$. In figure 28a we show the α_{LB} (—) and the α_{UB} (—) for each element of the sample train Ξ_{SCM} for the first iteration $K = 1$, whereas in figure 28b we depict the same quantities for the last iteration $K = K_{max} = 24$ of the *SCM* algorithm. We can see ho the lower an upper bounds are approaching.
In *EB* problem, the convergence is reached with more iterations due to fact that we deal with a vectorial problem.



(a) *EB*: First iteration of the *SCM* **(b)** *EB*: Last iteration of the *SCM*

Figure 28: *EB SCM* algorithm

7.5.3 *Greedy* algorithm

We provide here the results of the *Greedy* algorithm (Section 3.4.2).
Here, we have chosen a sample train Ξ_{train} of size is $n_{train} = 3000$, the tolerance is $A\epsilon_{toll,min} = 1 \cdot 10^{-2}$ and the maximum size of the *RB* space $N_{max} = 150$. As already done in the *TB* problem, even for the *EB* we have chosen to minimize the absolute error bound in energy norm $\Delta_N^{en}(\boldsymbol{\mu})$ (section 3.5.1, page 52). In figure 29a we have represented the *a-posteriori* error bound $\Delta_N^{en}(\boldsymbol{\mu})$ for $1 \leqslant N \leqslant N_{max}$. We can see that the error is monotonically decreasing. Just few basis ≈ 30 (versus $\approx$10000 *FE DOFs*) are needed to obtain a maximum error bound $\leqslant 10^{-2}$ on the displacement field.

We have performed a Greedy algorithm minimizing the *absolute error* on the *displacement field* (over the sample train Ξ_{train}); it is clear that from an engineering point of view it would be of greater interest to minimize, for example, the error on a stress (derivative of the displacement).
This choice is supported by the *RB* method, in this case we need to perform a second *Greedy sampling* on the so-called *dual problem*. For further explanation on this issue, we refer the reader to [RHP08]. In Figure 30a we depict the *Greedy* selected geometrical parameters, whereas in Figure 30b we show the physical parameters (the parameter choice is discussed in Section 7.3). As we can see the parameters chosen by the algorithm are clustered at the lower and upper bound of the parameters domain $\mathcal{D}$.

7.5.4 Output

Since we are dealing with a compliant case , the output in this case is the *average* of the *displacement* of the face Γ_6. In fact we have that:

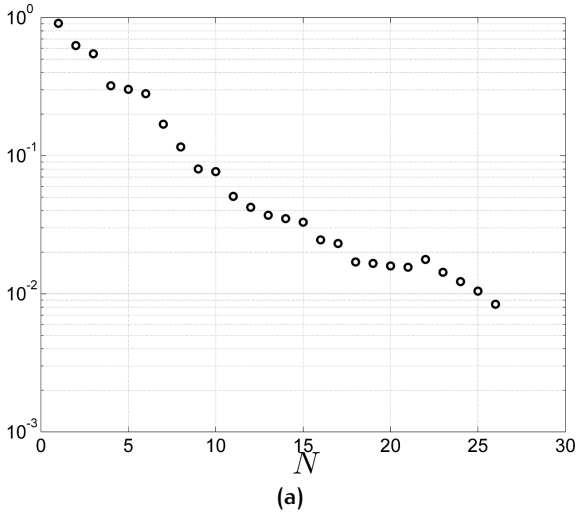
$$s_N(\boldsymbol{\mu}) = f(u_N; \boldsymbol{\mu}), \tag{7.16}$$

This choice is not mandatory, and just illustrative, since as output may be chosen for example as *average stress* on a subdomain, as maximum stress on the whole domain and so on. Unfortunately to obtain a *rigorous a-posteriori* error bound for the output, as already said in Section 7.5.3, we must perform an additional *Greedy procedure* on the so-called *dual problem*, see [RHP08, Gel10] for further explanations.

Since we have 5 parameters, we fix two parameters and add a relationship between the others to obtain a graphical visualization of the output. In particular we have chosen to vary parameters μ_2 and μ_5, i.e. the *Young modula* and the *thickness* of the central layer respectively. In addition we have set:

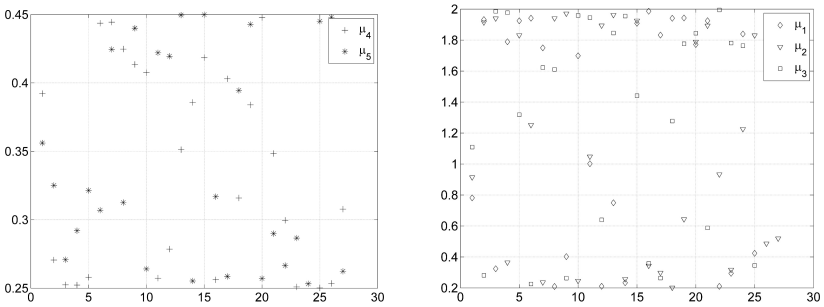
$$\mu_4 = \frac{1 - \mu_5}{2}.$$

In Figure 31a we depict the output obtained with the *RB* method, in Figure 31b we depict the a-posteriori error bound on the output $\Delta_N^s(\boldsymbol{\mu})$



(a)

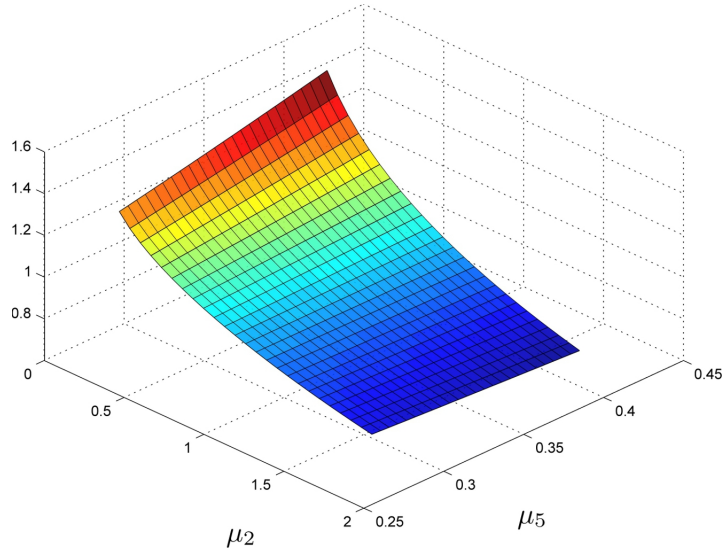
Figure 29: Error bound $\Delta_N^{en}(\boldsymbol{\mu})$ during Greedy convergence



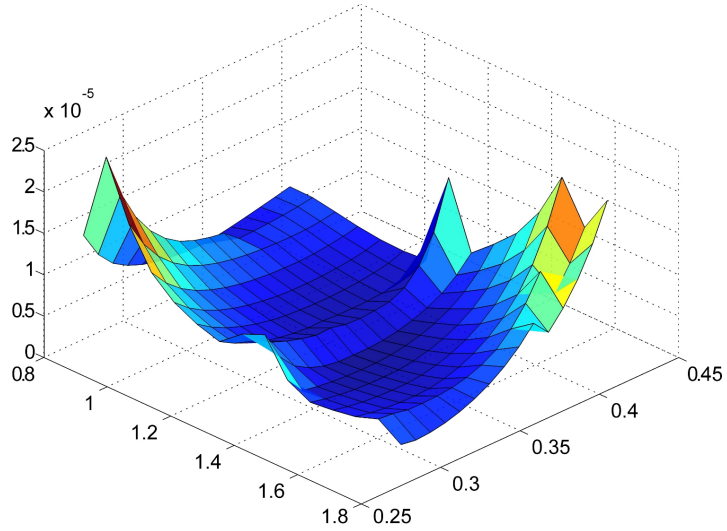
(a) Parameters μ_{geom} distribution **(b)** Parameters μ_{physic} distribution

Figure 30: *EB* Greedy results

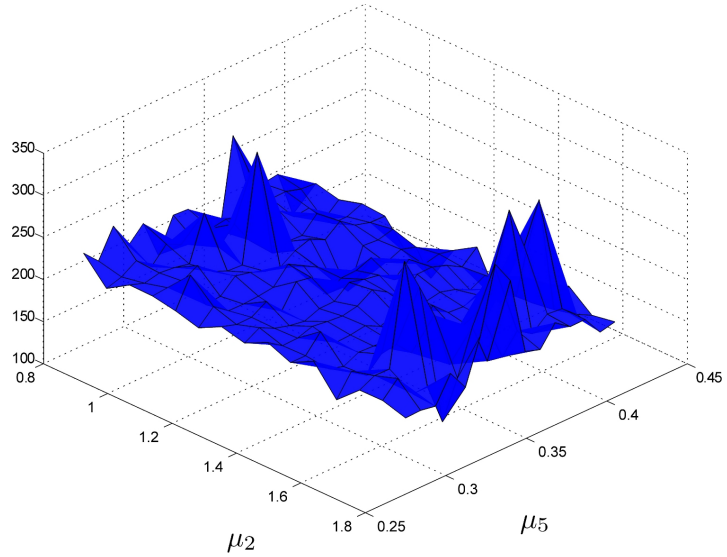
between the *FE* and the *RB* method. We can see that the output error is $\leqslant 1 \cdot 10^{-4}$, hence the error is effectively bounded by the square of the error on the solution field (we recall that in the Greedy we have set $\Delta_N^{en}(\boldsymbol{\mu}) \leqslant \epsilon_{toll,min} = 1 \cdot 10^{-2}$, $\forall \boldsymbol{\mu} \in \Xi_{train}$).
We remark again that this result follows from our assumption of compliance, that enables the so called *square effect* (equation 3.24, Chapter 3.2).
Finally in Figure 31c we have depicted the ratio between the computational time needed to evaluate the output in the *FE*: the *RB* method provides a computational time saving at leat *two order of magnitude* greater with respect to the ordinary *FE* method.



(a) *EB* output



(b) *EB* output error

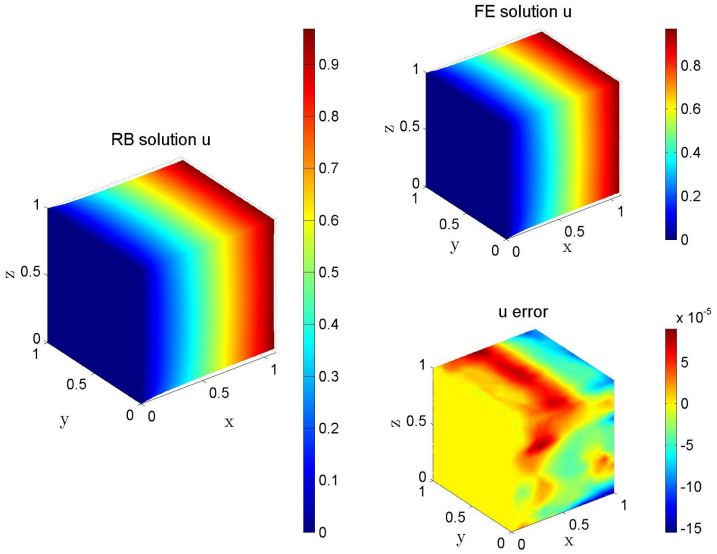


(c) $\frac{t^s_{FE}(\mu)}{t^s_{RB}(\mu)}$

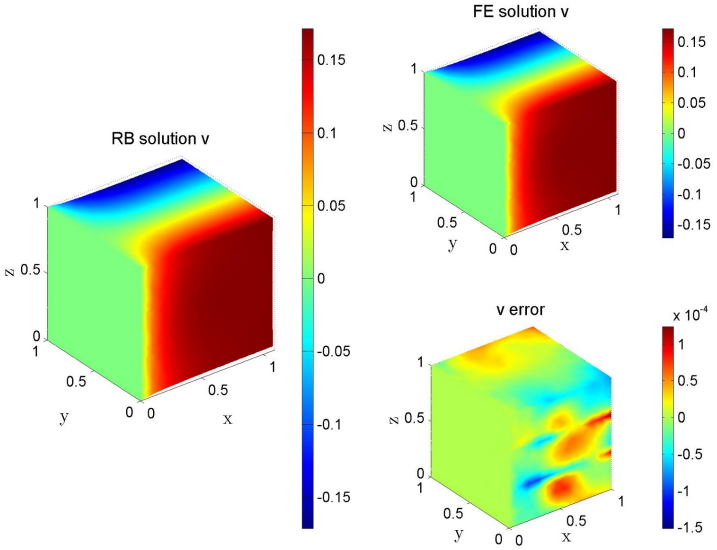
Figure 31: *EB* output results

7.5.5 Visualization

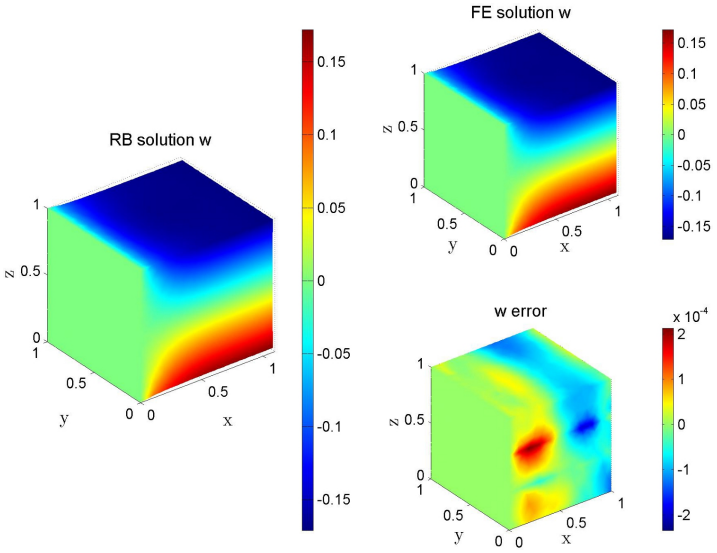
We report in this section the visualization of some representative *RB* solutions. In the first example, Figure 32, we show the solution on the reference domain. In the second example, Figure 33, we show the solution field selecting a generic combination of parameters into our parameter domain $\mathcal{D}$.
For each example we show: the three displacement (u, v, w) in the three direction x, y, z respectively and the error betweem the *RB* and *FE* solution. We can see from the pictures that the *error bound* compared to the *true error* is very sharp.



(a) u displacement and error

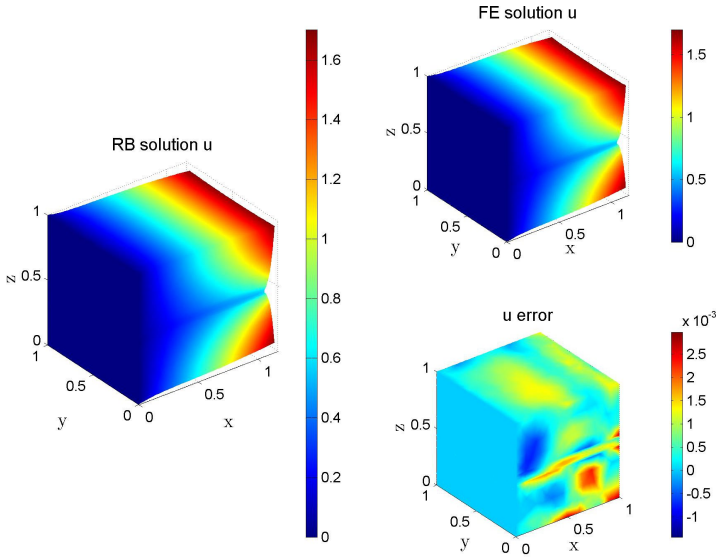


(b) v displacement and error

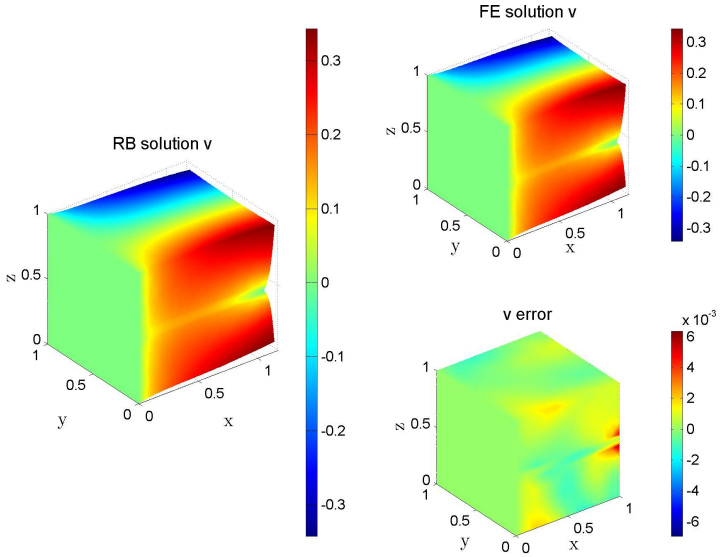


(c) w displacement and error

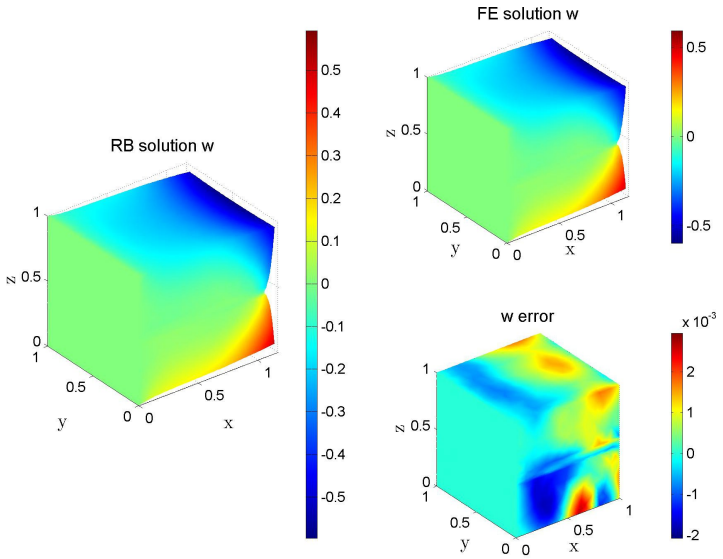
Figure 32: Example of a representative solution and error for $\boldsymbol{\mu} = \{1, 1, 1, 1/3, 1/3\}$, $\Delta_N^{\text{en}}(\boldsymbol{\mu}) \leqslant 4.17 \cdot 10^{-4}$



(a) u displacement and error



(b) v displacement and error



(c) w displacement and error

Figure 33: Example of a representative solution and error for $\boldsymbol{\mu} = \{0.5, 5, 0.5, 0.8, 0.1\}$, $\Delta_N^{\text{en}}(\boldsymbol{\mu}) \leqslant 6.7 \cdot 10^{-3}$

7.6 EB OPTIMIZATION

In this section we highlight, by means of a simple *optimization problem*, the great potential offered by the *RB* method.

Physical setup

With regard to the geometry of the *EB* problem, we assume that the upper and lower layer are made by *Aluminum*, whereas the central layer is made by *steel*.
The tickness of the central layer is our *design variable*, the other two layers have a dimension dependent upon the former. As already done in Section 7.5.4, we keep the *steel layer* centered between the Aluminum layers, so that the layers are symmetric with respect to the $x-y$ plane (see Figure 24).
We suppose that the solid is stretched from a face with a distributed load in x-direction and clamped on the other one.

layer	tickness	E $[N/mm^2]$	density $[Kg/dm^3]$
Ω_1	$\frac{1-\mu_5}{2}$	70000	2.7
Ω_2	μ_5	210000	7.8
Ω_1	$\frac{1-\mu_5}{2}$	70000	2.7

Table 10: EB optimization setup

Optimization target

We suppose that we are interested in the minimization of the average displacement of the loaded face.
Due to the different stiffness and density between the Aluminum and the Steel, if we enlarge the central layer (made of Steel) we will obtain a smaller displacement but we will also increase the weight and vice versa if we reduce the central layer we will reduce the weight but we will obtain a greater displacement.
Therefore we must cope with an *optimization problem*, so that we evaluate for example the *optimal thickness* that *minimize a suitable objective function*.
Our needs are to minimize either the displacement and the weight of the structure. The objective function has to weigh two contributions:

- the displacement of the face
- the total weight of the the structure

Hence we define the multi-objective function $\mathscr{F} : \mathcal{D} \to \mathbb{R}$ as

$$\mathscr{F}(\boldsymbol{\mu}) = \left(\frac{s(\boldsymbol{\mu})}{s_0}\right)^2 + \left(\frac{W(\boldsymbol{\mu})}{W_0}\right)^2 \tag{7.17}$$

where:

- $s(\boldsymbol{\mu})$ is the *average* x*-displacement* (the compliant output of Section 7.5.4) of the face Γ_6;
- $W(\boldsymbol{\mu})$ is the *total weight* of the solid block (sum of the weight of the three layers);
- s_0 and W_0 are the *reference displacement* and the *reference weight* respectively, obtained setting a reference parameter $\boldsymbol{\mu}_0$ equal to

$$\boldsymbol{\mu}_0 \left\{ 0.7 \cdot 10^5,\ 2.1 \cdot 10^5,\ 0.7 \cdot 10^5,\ \frac{1}{3},\ \frac{1}{3} \right\}.$$

We remark that our objective function is built so that the two contributions are on the order of unit. This choice is not mandatory, but it is the ordinary way to weigh contributions with different unit measure.
Our optimization problem states

$$\text{find } \boldsymbol{\mu} \to \min_{\boldsymbol{\mu} \in \mathcal{D}} \mathscr{F}(\boldsymbol{\mu}) \tag{7.18}$$

We will not go in deep into the minimization algorithm choice because this aspects lies outside from the thesis purposes. We have chosen a numerical tool implemented within `Matlab`, called fminbnd.
This numerical method uses either the *golden section* method and the *successive parabolic interpolation*, for further explanation see [FMM77].

7.6.1 FE-RB comparison

For the *RB* approximation we have used all the basis function selected by the Greedy. Of course in the optimization context the designer will take the minimum number of basis functions that will permit to bound the error (on the solution field or on a particular output) under a desirable threshold.
This means that the ratio between the time to obtain the optimal solution

	μ_5	iter	*DOFs*	time
RB	0.297463	11	10509	0.2028
FE	0.297467	11	26	30.7790

Table 11: EB optimization result

in the *RB* and *FE* method is

$$\frac{t_{FE}^{opt}}{t_{RB}^{opt}} \approx 150 \tag{7.19}$$

Therefore, we notice from 7.19 that the *RB* approximation provides a computational saving two order of magnitude greater than the classical *FE* method. This result is in accordance with previous tests carried out for example in [RHP08].

8 STOKES PROBLEM

8.1 INTRODUCTION

In this chapter we will exploit the creation of a *RB* approximation *in the 3D case* dealing with a *vectorial elliptic noncoercive* problem.
We deal with a steady *Stokes* problem, hence the unknowns are the three components of the velocity field plus the pressure (scalar), i.e. $\forall \mathbf{x} \in \Omega$

$$\underline{\mathbf{u}}(\mathbf{x}; \boldsymbol{\mu}) \in \mathbb{R}^4 = \{u, v, w, p\}^T$$

The lack of coercivity, since the Stokes problem is a saddle-point, see [QV97], brings some extra-difficulties in the construction of the *RB* approximation. Nonetheless, as we have seen in the previous chapters, we possess all the mathematical tools which allow us to build a proper *RB* approximation.
Up to date, this thesis is the first work that deals with a 3D Stokes problem; this work is oriented to be the first step for the construction of a *RB* approximation for *Navier-Stokes* equations in a 3D setting.
To build the *FE* ingredients (Section 3.3.5) we use the `COMSOL multiphysics` software and for the *RB* assembling we use the `rbMIT` routine available for *noncoercive problems* with few modifications to extend the procedure to 3D problems.

8.2 PROBLEM DESCRIPTION

We now briefly introduce the application chosen in order to exploit the *RB* approximation of the Stokes problem.

Physical and analytical problem

The Stokes problem is able to describe a flow at very low *Reynolds* number. The field of applicability of these equations is quite wide, e. g. hemodynamics, lubrication/micro-lubrication and *MEMS* [1] applications. In our case we will deal with a particular *MEMS* application, the Viscous Pump VP, see for example [SSG97, Woi05]. The *MEMS* are devices of characteristic dimension are between 1 to 100 [µm]. In the past few years, this *micromachines* have reached a great development thanks to the recent advance in *microtechnology process*.

1 Micro Electro-Mechanical Systems

In particular, the *VP* exploits the viscosity of a fluid to create a net flux in a micro channel. The pump (a sketch is depicted in Figure 34) is consti-

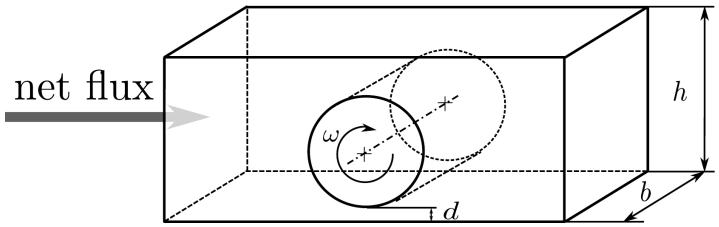


Figure 34: VP sketch

tuted by a tube in which an eccentric rotor is mounted (perpendicular to the flow direction) on a shaft and kept in rotation.
The rotor drags some fluid flow into rotation due to the viscosity, thanks to the eccentricity of the rotor there is a net flux of fluid that moves into the channel.
Many works in the last ten years have been devoted to the study of this device, especially in the design optimization context, we cite for example [SSG97] and [AHE04].
The *RB* approach is very well suited here, in fact the *Online* cheap and reliable stage permits fast evaluation of the output, hence a very inespensive design strategy. A generic viscous pump can depend on either geometrical and physical parameters. In particular, we will consider: the eccentricity of the rotor, the pressure imposed at the side of the channel and the angular speed of the shaft.
A classical optimization problem is to maximize the mass flow rate and at the same time minimize shaft power consumption when an external pressure load is applied along the channel housing the rotor, [DKC07].
The output, in our example, is the *net mass flux* obtained by the *VP*. The procedure to manage with the *RB* approximation of this problem is discussed in Section 3.3.

8.3 PARAMETERS CHOICE

In our problem, the parameters on which we will act to modify the behavior of the flow are *three*: the *eccentricity*, the *angular speed* of the rotor and the *pressure load* imposed at the channel.
We remark that the load pressure imposed (on the outlet, on the inlet we assume a null pressure) induces a flux in x direction that has to be overcome by the counter flux induced by the clockwise positive rotation of the rotor, as depicted in Figure 35. Either viscosity of the fluid and the density are kept constant. We have chosen a fluid with physical characteristic similar to the water, hence with a density $\tilde{\rho} = 1 \cdot 10^3 [Kg/m^3]$

and dynamic viscosity $\tilde{\mu} = 1 \cdot 10^{-3}$[Kg m/s]. It follows that the kinematic viscosity is $\tilde{\nu} = \mu/\rho = 1 \cdot 10^{-6}$[m/s].
The quantities with a tilde ˜ correspond to the dimensional quantities, whereas the absence of a tilde denotes non-dimensional quantities.

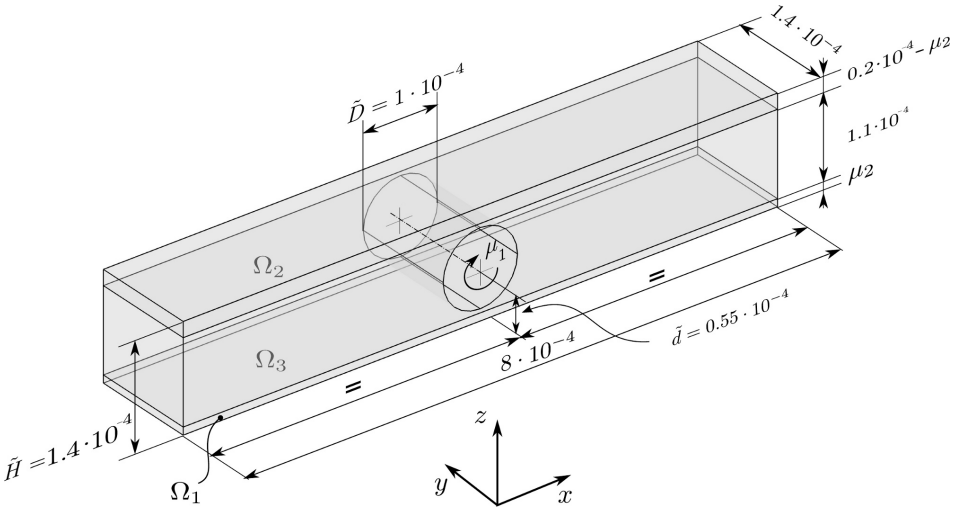


Figure 35: *VB* domain decomposition: the dimension are expressed in [m]

Parameters domain

We now summarize the parameters and the parameter domain chosen to describe our *VP* problem:

- the non-dimensional *angular speed* of the rotor, which correspond to the Reynolds number Re_D (see Section 8.4), denoted with μ_1, is defined as:

$$\mu_1 = \frac{\tilde{\omega}\tilde{D}^2}{2\tilde{\mu}} = \frac{\tilde{D}\tilde{U}}{\tilde{\mu}} = \mathrm{Re}_D, \tag{8.1}$$

  where:
  - $\tilde{U} = \dfrac{\tilde{D}\tilde{\omega}}{2}$[m/s] is the dimensional speed of reference,
  - $\tilde{\omega}$ (= [rad/s]) is the dimensional angular speed of the rotor;

- The eccentricity of the rotor, denoted with μ_2, defined as:

$$\mu_2 = 1 - \frac{\tilde{d}}{\tilde{H}/2}, \tag{8.2}$$

  where:
  1. $\tilde{H} = 1.4 \cdot 10^{-4}$ [m] is the total height of the channel;
  2. $\tilde{d}$ (= [m]) = is the height of the subdomain Ω^1.

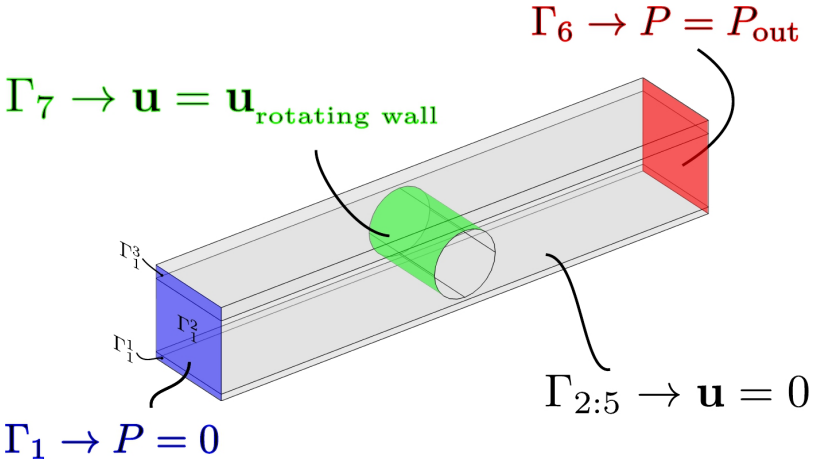


Figure 36: VP boundary conditions

- the non-dimensional load pressure imposed at the outlet of the channel, denoted with μ_3, defined as

$$\mu_3 = \frac{\tilde{P}_{\text{out}} \tilde{D}^2}{\tilde{\nu}^2 \tilde{\rho}}, \tag{8.3}$$

  where $\tilde{P}_{\text{out}}$ $(= [\text{Pa}])$ is the dimensional load pressure imposed at the channel.

The dimensional parameter domain is:

$$\begin{aligned} \mathcal{D} &= \left[\tilde{\omega}^{\text{min}}, \tilde{\omega}^{\text{max}}\right] \times \left[\tilde{P}^{\text{min}}_{\text{out}}, \tilde{P}^{\text{max}}_{\text{out}}\right] \times \left[\tilde{d}^{\text{min}}, \tilde{d}^{\text{max}}\right] \\ &[50, 200]\,[\text{rad/s}] \times [0.025, 0.25] 10^{-4} [\text{m}] \times [0.05, 10]\,[\text{Pa}] \end{aligned} \tag{8.4}$$

Thanks to the relationships given above, we can rewrite the parameter domain 8.4 in a non-dimensional way:

$$\begin{aligned} \mathcal{D} &= \left[\mu_1^{\text{min}}, \mu_1^{\text{max}}\right] \times \left[\mu_2^{\text{min}}, \mu_3^{\text{max}}\right] \times \left[\mu_3^{\text{min}}, \mu_3^{\text{max}}\right] \\ &= [0, 7.5]\, 10^{-4} \times [0.65, 0.95] \times [0, 30] \end{aligned} \tag{8.5}$$

Boundary conditions

Now we explain the boundary condition chosen to represent our physical model, (see Figure 36). With regard to the velocity boundary conditions, we have imposed an *homogeneus Dirichlet condition* on $\Gamma_{2:5}$ that represent a *no-slip* condition on the wall channel.
In addition we have imposed a *non-homogeneus Dirichlet condition* on Γ_7 that represent the no-slip condition on the rotating surface of the rotor.
We now give the explicit definition of the tangential velocity that constitutes our boundary condition. We will denote this velocity with $\tilde{\mathbf{g}} =$

$\tilde{\mathbf{g}}(\mathbf{x};\boldsymbol{\mu})$.
We recall that the rotor is normal to the channel so that the angular velocity is represented by a vector directed along the z-axis. It follows that the non-dimensional velocity $\mathbf{g}(\mathbf{x};\boldsymbol{\mu})\ \forall \mathbf{x} \in \Gamma_7$ is defined as:

$$\begin{aligned}\mathbf{g}(\mathbf{x};\boldsymbol{\mu}) &= \frac{\tilde{\mathbf{g}}(\mathbf{x};\boldsymbol{\mu})}{\tilde{U}} \\ &= \frac{2}{\tilde{D}\tilde{\omega}} \begin{Bmatrix} \tilde{\omega}\frac{\tilde{D}}{2}\cos\left(\arctan\left(\frac{x-x_O}{z-z_O}\right)\right) \\ -\tilde{\omega}\frac{\tilde{D}}{2}\sin\left(\arctan\left(\frac{x-x_O}{z-z_O}\right)\right) \\ 0 \end{Bmatrix} = \begin{Bmatrix} \cos\left(\arctan\left(\frac{x-x_O}{z-z_O}\right)\right) \\ -\sin\left(\arctan\left(\frac{x-x_O}{z-z_O}\right)\right) \\ 0 \end{Bmatrix}\end{aligned} \tag{8.6}$$

With regard to the pressure boundary conditions, we have imposed $\tilde{P} = \tilde{P}_{in} = 0$ on Γ_1 (inlet) and $\tilde{P} = \tilde{P}_{out} = \mu_3$ on Γ_6 (outlet).
The *pressure* boundary condition at the *inlet* and *outlet* is implemented as

$$\underline{\underline{\sigma}}\,\mathbf{n} = P_{in/out}, \tag{8.7}$$

where $\underline{\underline{\sigma}}$ is the non-dimensional Stokes stress tensor, $\mathbf{n}$ is the unit normal outward vector and P is the imposed non-dimensional pressure.
Finally on the internal faces we have assumed continuity of either pressure, velocity and fluxes.

8.4 VP PROBLEM FORMULATION

8.4.1 Original domain

In this section the analytical formulation of the governing PDEs for the Stokes problem on the *original domain* will be introduced. The equation that describes the pressure and velocity field, within the hypothesis described in section 8.1, is the following:

$$\begin{cases} -Re_D\,\Delta_o\hat{\mathbf{u}} + \nabla_o\hat{P} = \mathbf{f} & \text{in } \Omega_o(\boldsymbol{\mu}) \\ \nabla_o\cdot\hat{\mathbf{u}} = 0 & \text{in } \Omega_o(\boldsymbol{\mu}) \\ \hat{\mathbf{u}} = 0 & \text{on } \Gamma_{o_{2:5}}(\boldsymbol{\mu}) \\ \hat{\mathbf{u}} = \mathbf{g} & \text{on } \Gamma_{o_7}(\boldsymbol{\mu}) \\ Re_D\dfrac{\partial\hat{\mathbf{u}}}{\partial\mathbf{n}} - \hat{P}\mathbf{n} = 0 & \text{on } \Gamma_{o_1}(\boldsymbol{\mu}) \\ Re_D\dfrac{\partial\hat{\mathbf{u}}}{\partial\mathbf{n}} - \hat{P}\mathbf{n} = \mu_3 & \text{on } \Gamma_{o_6}(\boldsymbol{\mu}) \end{cases} \tag{8.8}$$

where:

- $\mathbf{u} = \{u, v, w\}^T$ is the non-dimensional field velocity, defined as:

$$\mathbf{u} = \frac{1}{\tilde{U}}\tilde{\mathbf{u}} = \frac{1}{\tilde{U}}\{\tilde{u}, \tilde{v}, \tilde{w}\}^T; \tag{8.9}$$

- P is the non-dimensional field of pressure, defined as

$$P = \frac{\tilde{P}\tilde{D}^2}{\tilde{\nu}^2\tilde{\rho}}; \tag{8.10}$$

- $Re_D = \mu_1$ is the Reynolds number, defined as:

$$Re_D = \frac{\tilde{D}\tilde{U}}{\tilde{\mu}};. \tag{8.11}$$

We remark that, tipically, the momentum equation is written so that the non-dimensional coefficient associated to the viscous stress is the inverse of Re. We have pursued another choice, i.e. to follow the notation used in literature with regard to this topic, see [SSG97, DKC07].
We now introduce two functional spaces for the velocity and for the pressure. For the velocity we choose $V^e = \left\{ \mathbf{v} \in \left(H^1(\Omega_o)\right)^3 \middle| \mathbf{v} = 0 \text{ on } \Gamma_{2:5} \right\}$, whereas for the pressure we choose $Q^e = \left\{ q \in L^2_0(\Omega_o) \right\}$, where the subscript o of the space L^2_0 denotes that we are interested in a solution with a *zero average pressure*. We recall that the domain is decomposed in three subdomains and each face of the domain is divided in subfaces, as depicted in Figure 35-36.
Then, multiplying the first equation of the problem 8.8 by a test function $\mathbf{v} \in V^e$ and the second equation by $q \in Q^e$, integrating on $\Omega_o(\boldsymbol{\mu})$ and using the *Green formula* we get the *weak formulation*:

$$\begin{cases} \sum_{k=1}^{3} \int_{\Omega_o^k(\boldsymbol{\mu})} \mu_1 \nabla_o \hat{\mathbf{u}} \cdot \nabla_o v \, d\Omega_o^k - \int_{\Omega_o^k(\boldsymbol{\mu})} \hat{P} \nabla_o \mathbf{v} \, d\Omega_o^k = 0 \\ \sum_{k=1}^{3} \int_{\Omega_o^k(\boldsymbol{\mu})} q \nabla_o \cdot \hat{\mathbf{u}} \, d\Omega_o^k = 0 \end{cases} \tag{8.12}$$

We now define the bilinear forms $a : V^e \times V^e \times \mathcal{D} \to \mathbb{R}$ and $b : V^e \times Q^e \times \mathcal{D} \to \mathbb{R}$ and the linear functional $\mathbf{f} : V^e \times \mathcal{D} \to \mathbb{R}$ as follows:

$$a(\hat{\mathbf{u}}, \mathbf{v}; \boldsymbol{\mu}) = \sum_{k=1}^{3} \int_{\Omega_o^k(\boldsymbol{\mu})} \mu_1 \nabla_o \hat{\mathbf{u}} \cdot \nabla_o \mathbf{v} \, d\Omega_o^k \tag{8.13}$$

$$b(\hat{\mathbf{u}}, q; \boldsymbol{\mu}) = \sum_{k=1}^{3} \int_{\Omega_o^k(\boldsymbol{\mu})} q \nabla_o \cdot \hat{\mathbf{u}} \, d\Omega_o^k \tag{8.14}$$

With these notations the problem 8.12 becomes:

$$\begin{cases} a(\hat{\mathbf{u}}, \mathbf{v}; \boldsymbol{\mu}) + b(\mathbf{v}, \hat{P}; \boldsymbol{\mu}) = 0 & \forall \mathbf{v} \in V^e \\ b(\hat{\mathbf{u}}, q; \boldsymbol{\mu}) = 0 & \forall q \in Q^e \end{cases} \tag{8.15}$$

Since we are dealing with *non-homogeneus Dirichlet* and *Neumann* conditions (as indicated in 8.8) and due to the choice of our functional space,

the weak formulation becomes, [Qua09]: find $(\mathbf{u}, p) \in V^e \times Q^e$ such that, [QV97]:

$$\begin{cases} a(\mathbf{u}, \mathbf{v}; \boldsymbol{\mu}) + b(\mathbf{v}, p; \boldsymbol{\mu}) = \mathbf{F}(\mathbf{v}; \boldsymbol{\mu}) & \forall \mathbf{v} \in V^e \\ b(\mathbf{u}, q; \boldsymbol{\mu}) = G(q; \boldsymbol{\mu}) & \forall q \in Q^e \end{cases} \tag{8.16}$$

Denoting a lifting of the boundary datum $\mathbf{g}$ with $\mathbf{Rg} \in \left[H^1(\Omega_o(\boldsymbol{\mu}))\right]^3$, we have placed $\mathbf{u} = \hat{\mathbf{u}} - \mathbf{Rg}$, whereas no lifting is required for the pressure field, hence $p = \hat{p}$. We remark that the function $\mathbf{g}$ arises from the no-slip condition on the rotor. The new known terms $F(\mathbf{v}; \boldsymbol{\mu})$ and $G(q; \boldsymbol{\mu})$ are defined as:

$$\mathbf{F}(\mathbf{v}; \boldsymbol{\mu}) = \sum_{k=1}^{3} \int_{\Gamma_{o_6}^k(\boldsymbol{\mu})} p_{out}\, \mathbf{v}\, d\Gamma_{o_6}^k + a(\mathbf{Rg}, \mathbf{v}; \boldsymbol{\mu}), \tag{8.17}$$

$$G(q; \boldsymbol{\mu}) = -b(\mathbf{Rg}, q; \boldsymbol{\mu}). \tag{8.18}$$

8.4.2 Reference domain

We now rewrite the problem 8.16 on the reference domain to exploit the affine decomposition.
We need to evaluate the affine mappings for each subdomain to rewrite all the parameter dependent quantities in an affine formulation.
We summarize the results for the geometric transformation in Table 12. In addition the same relationships for the Jacobian and the derivatives

sub	G_{11}^{aff}	G_{22}^{aff}	G_{33}^{aff}	C_1^{aff}	C_2^{aff}	C_3^{aff}
Ω^1	1	1	$-\frac{1}{7(\mu_2-1)}$	0	0	$-\mu_2$
Ω^2	1	1	1	0	0	0
Ω^3	1	1	$\frac{2}{7\mu_2-4}$	0	0	$\frac{7(\mu_2-1)}{11}$

Table 12: VP affine mappings

operator used in the *TB* and *EB* hold, hence we can directly step to the definition of the bilinear and linear forms on the reference domain.

Bilinear form

Starting from the bilinear form a (8.13) we can write:

$$
\begin{aligned}
a(\mathbf{u},\mathbf{v};\boldsymbol{\mu}) &= \sum_{k=1}^{3} \int_{\Omega_o^k(\boldsymbol{\mu})} \mu_1 \nabla_o \mathbf{u} \cdot \nabla_o \mathbf{v} \, d\Omega_o^k = \\
&= \sum_{k=1}^{3} \int_{\Omega_r^k(\boldsymbol{\mu})} \mu_1 \nabla^T \mathbf{u} \left(\underline{\underline{G}}^{aff,k}\right)^T \left(\underline{\underline{G}}^{aff,k}\right) \nabla \mathbf{v} \left(J^{aff,k}\right)^{-1} \, d\Omega_r^k = \\
&= 1 \left(\int_{\Omega_r^1} \mu_1 \left(\frac{\partial u_1}{\partial x_{r_1}} \frac{\partial v_1}{\partial x_{r_1}} + \ldots + \frac{\partial u_3}{\partial x_{r_2}} \frac{\partial v_3}{\partial x_{r_2}} \right) d\Omega_r^1 + \ldots \right. \\
&\ldots + \left. \int_{\Omega_r^3} \mu_1 \left(\frac{\partial u_1}{\partial x_{r_1}} \frac{\partial v_1}{\partial x_{r_1}} + \ldots + \frac{\partial u_3}{\partial x_{r_2}} \frac{\partial v_3}{\partial x_{r_2}} \right) d\Omega_r^3 \right) + \ldots \\
&\ldots + \theta_a^3(\boldsymbol{\mu}) \int_{\Omega_r^3} \left(\frac{\partial u_1}{\partial x_{r_3}} \frac{\partial v_1}{\partial x_{r_3}} + \frac{\partial u_2}{\partial x_{r_3}} \frac{\partial v_2}{\partial x_{r_3}} + \frac{\partial u_3}{\partial x_{r_3}} \frac{\partial v_3}{\partial x_{r_3}} \right) d\Omega_r^3
\end{aligned} \tag{8.19}
$$

Meanwhile the bilinear form b becomes:

$$
\begin{aligned}
b(\mathbf{u}, q; \boldsymbol{\mu}) &= \sum_{k=1}^{3} \int_{\Omega_o^k(\boldsymbol{\mu})} q \, \nabla_o \hat{\mathbf{u}} \, d\Omega_o^k = \\
&= \sum_{k=1}^{3} \int_{\Omega_o^k(\boldsymbol{\mu})} q \, \underline{\underline{G}}^{aff,k} \nabla \cdot \mathbf{u} \left(J^{aff,k}\right)^{-1} d\Omega_r^k = \\
&= 1 \left(\int_{\Omega_r^1} q \left(\frac{\partial u_1}{\partial x_{r_1}} + \ldots + \frac{\partial u_3}{\partial x_{r_3}} \right) d\Omega_r^1 + \ldots + \int_{\Omega_r^3} q \left(\frac{\partial u_1}{\partial x_{r_1}} + \ldots \right. \right. \\
&\quad \left. \left. + \ldots \frac{\partial u_3}{\partial x_{r_3}} \right) \right) + \theta_b^2(\boldsymbol{\mu}) \int_{\Omega_r^3} q \left(\frac{\partial u_1}{\partial x_{r_1}} + \ldots + \frac{\partial u_3}{\partial x_{r_3}} \right) d\Omega_r^3
\end{aligned} \tag{8.20}
$$

We present in detail the definition of each θ-function and each parameter independent matrix associated to the parametric bilinear forms a and b in Table 13-14.

Linear functional

We now briefly present the *RHS* of the Stokes problem on the *original domain*. We remark that the first addendum of equation 8.17 comes from a Neumann non-homogeneous condition, hence the treatment is standard; whereas the second addendum of equation 8.17 and the equation 8.18 arise from a non-homogeneous Dirichlet condition.
Since we have placed the Stokes equations in a non-dimensional frame, then the parametric dependence upon the angular velocity of the rotor is accounted in the bilinear form a, therefore the boundary condition g

has been set in a parameter independent way (we recall that we have set $\mu_1 = Re_D$). The linear forms $\mathbf{F}(\mathbf{v};\boldsymbol{\mu})$ and $G(q;\boldsymbol{\mu})$ can be written as:

$$\begin{aligned}\mathbf{F}(\mathbf{v};\boldsymbol{\mu}) &= \sum_{k=1}^{3}\int_{\Gamma^k_{o_6}(\boldsymbol{\mu})} P_{out}\,\mathbf{v}\,d\Gamma^k_{o_6} + a(\mathbf{Rg},\mathbf{v};\boldsymbol{\mu}) = \\ &= \mu_2\cdot 10^5\mu_3\int_{\Gamma^1_{r_6}}\mathbf{v}\,d\Gamma^1_{r_6} + \mu_3\int_{\Gamma^2_{r_6}}\mathbf{v}\,d\Gamma^2_{r_6} + \\ &\quad + \left(15\mu_3 - \mu_2\cdot 10^5/2\mu_3\right)\int_{\Gamma^3_{r_6}}\mathbf{v}\,d\Gamma^3_{r_6} + a(\mathbf{Rg},\mathbf{v};\boldsymbol{\mu}). \qquad (8.21)\\ G(q;\boldsymbol{\mu}) &= -b(\mathbf{Rg},q;\boldsymbol{\mu})\end{aligned}$$

We summarize the parameter dependent functions and the Offline parameter independent vectors associated to the linear forms in Table 15-16.

q	$\theta_a^q(\boldsymbol{\mu})$	$\underline{\underline{\mathbb{A}}}^q$
1	μ_1	$\int_{\Omega_r^1}\left(\frac{\partial u_1}{\partial x_{r_1}}\frac{\partial v_1}{\partial x_{r_1}} + \frac{\partial u_1}{\partial x_{r_2}}\frac{\partial v_1}{\partial x_{r_2}} + \frac{\partial u_2}{\partial x_{r_1}}\frac{\partial v_2}{\partial x_{r_1}} + \frac{\partial u_2}{\partial x_{r_2}}\frac{\partial v_2}{\partial x_{r_2}}\right)d\Omega_r^1 +$ $+\int_{\Omega_r^2}\left(\frac{\partial u_1}{\partial x_{r_1}}\frac{\partial v_1}{\partial x_{r_1}} + \frac{\partial u_1}{\partial x_{r_2}}\frac{\partial v_1}{\partial x_{r_2}} + \frac{\partial u_1}{\partial x_{r_3}}\frac{\partial v_1}{\partial x_{r_3}} + \frac{\partial u_2}{\partial x_{r_1}}\frac{\partial v_2}{\partial x_{r_1}} + \right.$ $\left. + \frac{\partial u_2}{\partial x_{r_2}}\frac{\partial v_2}{\partial x_{r_2}} + \frac{\partial u_2}{\partial x_{r_3}}\frac{\partial v_2}{\partial x_{r_3}}\right)d\Omega_r^2 +$ $+\int_{\Omega_r^3}\left(\frac{\partial u_1}{\partial x_{r_1}}\frac{\partial v_1}{\partial x_{r_1}} + \frac{\partial u_1}{\partial x_{r_2}}\frac{\partial v_1}{\partial x_{r_2}} + \frac{\partial u_2}{\partial x_{r_1}}\frac{\partial v_2}{\partial x_{r_1}} + \frac{\partial u_2}{\partial x_{r_2}}\frac{\partial v_2}{\partial x_{r_2}}\right)d\Omega_r^3$
2	$2\frac{(7\mu_2)-1}{(7\mu_2-4)^2}$	$\int_{\Omega_r^1}\left(\frac{\partial u_1}{\partial x_{r_3}}\frac{\partial v_1}{\partial x_{r_3}} + \frac{\partial u_2}{\partial x_{r_3}}\frac{\partial v_2}{\partial x_{r_3}} + \frac{\partial u_3}{\partial x_{r_3}}\frac{\partial v_3}{\partial x_{r_3}}\right)d\Omega_r^1$
3	$\frac{2\mu_1((7\mu_2)-1)}{(7\mu_2-4)^2}$	$\int_{\Omega_r^3}\left(\frac{\partial u_1}{\partial x_{r_3}}\frac{\partial v_1}{\partial x_{r_3}} + \frac{\partial u_2}{\partial x_{r_3}}\frac{\partial v_2}{\partial x_{r_3}} + \frac{\partial u_3}{\partial x_{r_3}}\frac{\partial v_3}{\partial x_{r_3}}\right)d\Omega_r^3$

Table 13: θ_a-functions

q	$\theta_b^q(\boldsymbol{\mu})$	$\underline{\underline{\mathbb{B}}}^q$
1	1	$\int_{\Omega_r^1} q \left(\frac{\partial u_1}{\partial x_{r_1}} + \frac{\partial u_2}{\partial x_{r_2}} + \frac{\partial u_3}{\partial x_{r_3}} \right) d\Omega_r^1 +$ $+ \int_{\Omega_r^2} q \left(\frac{\partial u_1}{\partial x_{r_1}} + \frac{\partial u_2}{\partial x_{r_2}} + \frac{\partial u_3}{\partial x_{r_3}} \right) d\Omega_r^2 +$ $+ \int_{\Omega_r^3} q \left(\frac{\partial u_1}{\partial x_{r_1}} + \frac{\partial u_2}{\partial x_{r_2}} \right) d\Omega_r^3 +$
2	$\frac{7\mu_2 - 1}{7\mu_2 - 4}$	$\int_{\Omega_r^3} q \frac{\partial u_3}{\partial x_{r_3}} d\Omega_r^3$

Table 14: θ_b-functions

q	$\theta_F^q(\boldsymbol{\mu})$	$\underline{\underline{\mathbb{F}}}^q$
1	$\frac{20^2 \mu_3}{10^2 \, 1.1^2}$	$\int_{\Gamma_{o_6}^1(\boldsymbol{\mu})} \mathbf{v} \, d\Gamma_{o_6}^1$
2	$-\frac{20^2 \mu_3 (7\mu_2 - 7)}{10^2 \, 1.1^2}$	$\int_{\Gamma_{o_6}^2(\boldsymbol{\mu})} \mathbf{v} \, d\Gamma_{o_6}^2$
3	$\frac{20^2 \mu_3 ((7\mu_2)/2 - 2)}{10^2 \, 1.1^2}$	$\int_{\Gamma_{o_6}^3(\boldsymbol{\mu})} \mathbf{v} \, d\Gamma_{o_6}^3$
4:7	$\mu_1 \theta_a^{1:3}(\boldsymbol{\mu})$	$a^{1:3}(\mathbf{Rg}, \mathbf{v})$

Table 15: θ_F-functions

q	$\theta_G^q(\boldsymbol{\mu})$	$\underline{\underline{\mathbb{G}}}^q$
1:2	$\mu_1 \theta_b^{1:2}(\boldsymbol{\mu})$	$b^{1:2}(\mathbf{Rg}, q)$

Table 16: θ_G-functions

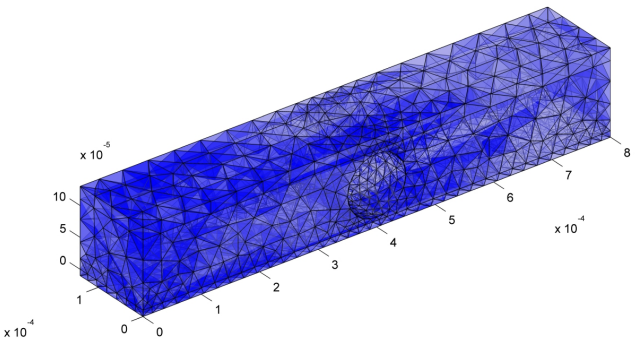
8.5 RESULTS AND VISUALIZATION

In this section we will present the results obtained for the *3D viscous pump case*.First we will give some details about the *FE* approximation concerning the mesh, the basis function and the input needed by COMSOL, then we will present the convergence results for either the *SCM* and the *Greedy* algorithm.
Below we will show the results for the output along with the computational time saving obtained with the *RB* method.

8.5.1 *FE* approximatiom with COMSOL

FE *discretization on reference domain*

We depict in Figure 37 the reference domain over which we will assemble our *FE* components; we also summarize the main characteristic of the mesh together with the *FE* basis functions and the *DOFs* for the pressure and velocity fields.

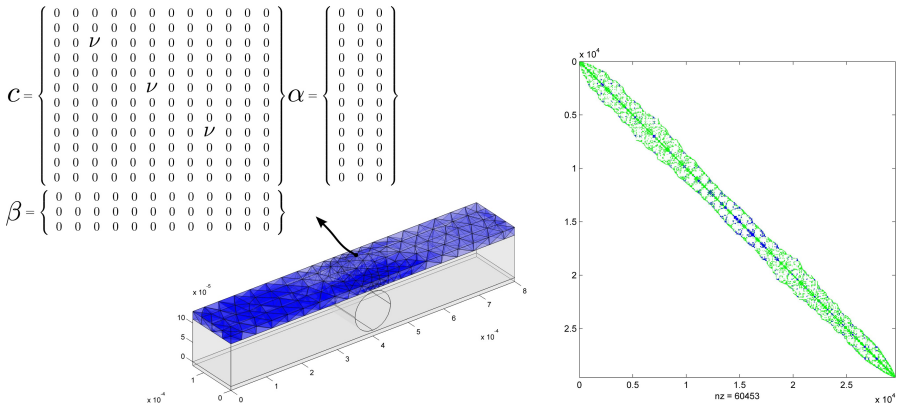


mesh vertices	2021	
vertex	24	
edges	384	
triangle	3474	
tetrahedral	8414	
basis functions	P	$\mathbb{P}^1$
	u	$\mathbb{P}^2$
DOFs	P	2021
	u	27543

Figure 37: VP reference domain

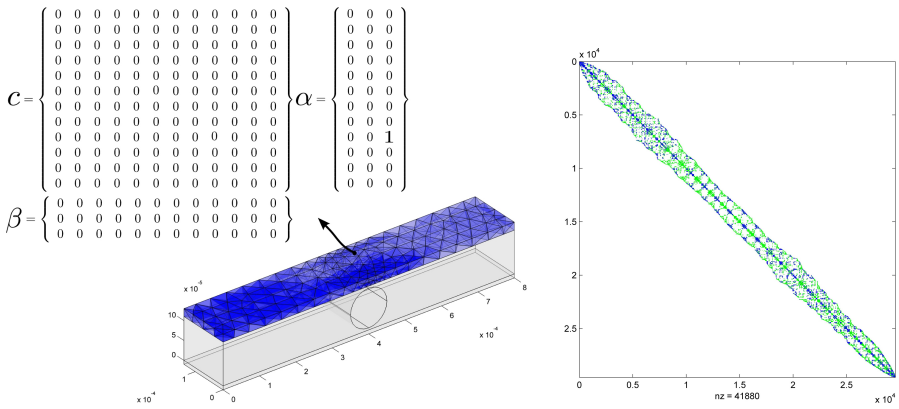
Matrices assembling

In this section, we assemble the parameter independent matrices (see Section 8.4.2) needed by the *RB* procedure.
In the Stokes case, after COMSOL assembling, we have to perform an intermediate step, as we will see in the following section.
To assemble the Stokes matrices and vectors we just need to provide to COMSOL the entries reported in the Tables 13, 14, 15 and 16.
In the *VP* case, the definition of the the Stokes problem is quite involved, we refer to the COMSOL manuals [Como7b, Como7a] for more detailed explanations.
We show in figure the assembled matrices $\underline{\underline{\mathbb{A}}}^3_{\text{COMSOL}}$ $\underline{\underline{\mathbb{B}}}^2_{\text{COMSOL}}$ (see Tables 13-14). The underscript $_{\text{COMSOL}}$ recalls that, before passing the *FE* ingredients to the *rbMIT*, we must take an intermediate step (see Section 8.5.1.1).



(a) subdomain Ω_r^1 **(b)** $\underline{\underline{\mathbb{A}}}^3_{\text{COMSOL}}$

Figure 38: In Figure 38a we have highlighted the subdomain and the definition of the coefficients used to assemble the matrix $\underline{\underline{\mathbb{A}}}^3_{\text{COMSOL}}$, whereas in Figure 38b we have depicted the pattern of this matrix (●) along with the pattern of the global stiffness matrix (●).



(a) subdomains $\Omega_r^{1:3}$ **(b)** $\underline{\underline{\mathbb{B}}}^2_{\text{COMSOL}}$

Figure 39: In Figure 39a we have highlighted the subdomains and the definition of the coefficients used to assemble the matrix $\underline{\underline{\mathbb{B}}}^2_{\text{COMSOL}}$, whereas in Figure 39b we have depicted the pattern of the matrix

8.5.1.1 *Matrices reordering*

Unlike the *TB* and *EB* case, we have denoted the assembled *FE* matrices and vectors with the superscript $^{\texttt{COMSOL}}$. In fact, in the Stokes case, we must perform a reordering on the *DOFs* of the assembled entities.
We first recall that the linear system arising from a classical *FE* approximation of the Stokes problem is the following, [QV97]:

$$\begin{bmatrix} \underline{\underline{A}}_{\mathcal{N}_v \times \mathcal{N}_v} & \underline{\underline{B}}_{\mathcal{N}_v \times \mathcal{N}_p} \\ \underline{\underline{B}}^T_{\mathcal{N}_v \times \mathcal{N}_v} & \underline{\underline{0}}_{\mathcal{N}_p \times \mathcal{N}_p} \end{bmatrix} \begin{Bmatrix} \underline{\mathbf{u}}_{\mathcal{N}_v \times 1} \\ \underline{P}_{\mathcal{N}_p \times 1} \end{Bmatrix} = \begin{Bmatrix} \underline{F}_{\mathcal{N}_v \times 1} \\ \underline{G}_{\mathcal{N}_p \times 1} \end{Bmatrix} \tag{8.22}$$

In this case, the *DOFs* of velocity and pressure are separated into the vector of unknowns.
Unfortunately COMSOL exploit a different sorting, so that the pressure and velocity *DOFs* are mixed and the linear system is set in the following way:

$$\begin{bmatrix} \underline{\underline{K}}_{\texttt{COMSOL}} \end{bmatrix} \begin{Bmatrix} \underline{\mathbf{x}}_{\texttt{COMSOL}} \end{Bmatrix} = \begin{Bmatrix} \underline{f}_{\texttt{COMSOL}} \end{Bmatrix} \tag{8.23}$$

$$\left(\begin{bmatrix} \mathbb{R}^{(\mathcal{N}_v+\mathcal{N}_p)\times(\mathcal{N}_v+\mathcal{N}_p)} \end{bmatrix} \begin{Bmatrix} \mathbb{R}^{(\mathcal{N}_v+\mathcal{N}_p)\times 1} \end{Bmatrix} = \begin{Bmatrix} \mathbb{R}^{(\mathcal{N}_v+\mathcal{N}_p)\times 1} \end{Bmatrix} \right) \tag{8.24}$$

where $\underline{\underline{K}}_{\texttt{COMSOL}}$ and $\underline{f}_{\texttt{COMSOL}}$ are the COMSOL assembled stiffness matrix and *RHS* vector respectively. The vector of *DOFs* is sorted as follows:

$$\left\{\underline{\mathbf{x}}_{\texttt{COMSOL}}\right\} = \left\{u_1, v_1, w_1, p_1, \ldots, u_{\mathcal{N}_v}, v_{\mathcal{N}_v}, w_{\mathcal{N}_v}, p_{\mathcal{N}_p}\right\}^T \tag{8.25}$$

Hence, the *FE* matrices and vectors must be reordered before being passed to rbMIT. In particular, introducing a *permutation matrix*, denoted with $\underline{\underline{M}}$, such that:

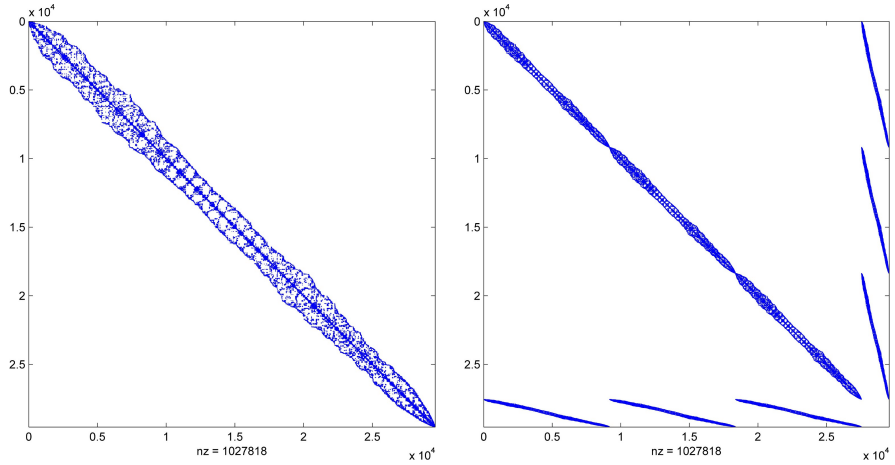
$$\begin{Bmatrix} \underline{\mathbf{u}} \\ \underline{P} \end{Bmatrix} = \left[\underline{\underline{M}}\right] \left\{\underline{\mathbf{x}}_{\texttt{COMSOL}}\right\} \tag{8.26}$$

and replacing this equation into the COMSOL linear system 8.23, we obtain the reordering conditions:

$$\begin{bmatrix} \underline{\underline{A}}_{\mathcal{N}_v \times \mathcal{N}_v} & \underline{\underline{B}}_{\mathcal{N}_v \times \mathcal{N}_p} \\ \underline{\underline{B}}^T_{\mathcal{N}_v \times \mathcal{N}_v} & \underline{\underline{0}}_{\mathcal{N}_p \times \mathcal{N}_p} \end{bmatrix} = \left[\underline{\underline{M}}\right]^T \left[\underline{\underline{K}}_{\texttt{COMSOL}}\right] \left[\underline{\underline{M}}\right] \tag{8.27}$$

$$\begin{Bmatrix} \underline{\mathbb{F}} \\ \underline{\mathbb{G}} \end{Bmatrix} = \left[\underline{\underline{M}}\right]^T \left\{\underline{f}_{\texttt{COMSOL}}\right\} \tag{8.28}$$

These operations are expensive because involves a cost $\mathcal{O}((\mathcal{N}_v + \mathcal{N}_p)^2)$, nonetheless thanks to the Offline/Online *RB* splitting procedure, they are performed just once.
We depict in Figure 40 the global stiffness matrix pattern before and after the reordering (on the *global* Stokes stiffness matrix).



(a) COMSOL ordering **(b)** classical ordering

Figure 40: *VP* matrices reordering

8.5.2 *SCM* algorithm

We recall that in this case we are interested in the estimation a lower bound of the *inf-sup* constant $\beta_N(\boldsymbol{\mu})$ (see Section 3.3).
For the *SCM* algorithm (section 3.6.3) we take a sample train Ξ_{SCM} of size $n_{SCM} = 1000$, a tolerance $\epsilon_{SCM} = 0.8$, $M_\alpha = 10$, $M_+ = 5$ and $|\mathcal{P}| = 200$.
In figure 41 we depict the *cover percentage* as function of the *SCM* algorithm iteration. The cover percentage is defined as the percentage of samples in the Ξ_{SCM} that satisfies either (i) the tolerance ϵ_{SCM} (over $(\beta_{UB} - \beta_{LB})/\beta_{UB}$, see 3.138) and (ii) the positivity of β_{LB}.

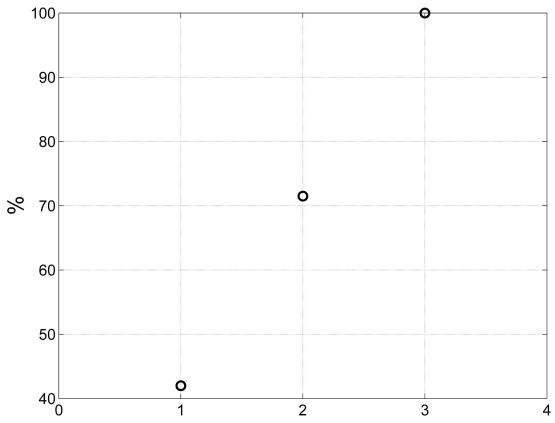


Figure 41: VP SCM algorithm: cover percentage

8.5.3 *Greedy* algorithm

We present the results for the *Greedy* algorithm (Section 3.4.2), during the *RB* assembling procedure.
We have chosen a sample train Ξ_{train} of size is $n_{train} = 3000$, a tolerance is $\epsilon_{toll,min} = 1 \cdot 10^{-6}$ and a maximum size of the *RB* space is taken $N_{max} = 100$. We have chosen to minimize the absolute error bound in the energy norm $\Delta_N^{en}(\mu)$; we recall from Chapter 2.4 that this choice corresponds to minimize H^1-norm of the error bound on the velocity and L^2-norm of error bound on the pressure. In figure 42 we have represented the

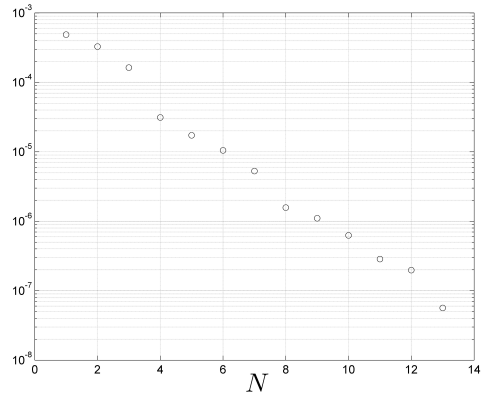


Figure 42: Error bound $\Delta_N^{en}(\mu)$

error bound $\Delta_N^{en}(\mu)$ for $1 \leqslant N \leqslant N_{max}$. We can see that the error is monotonically decreasing. Just very few basis ≈ 13 (versus $\approx$30000 *FE DOFs*) are needed to obtain a maximum error bound $\leqslant 1 \cdot 10^{-6}$ on the velocity and pressure field for all the samples in Ξ_{train}.
In Figure 43 we depict the parameter samples selected by the Greedy algorithm. We note that, the algorithm selects the samples in the worst case scenario, i.e. the parameters chosen are those clustered far from the *"center"* of the parameter domain.

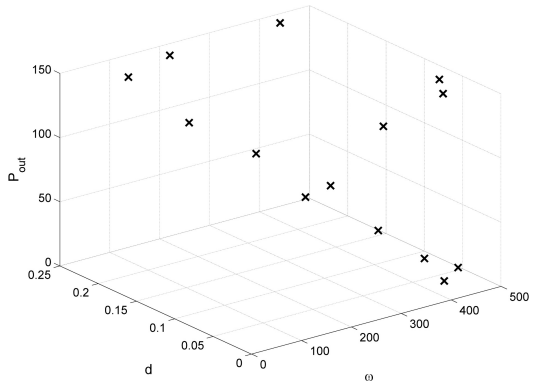


Figure 43: Parameters distribution

8.5.4 Output

The output is the *net mass flux* of fluid which moves in the channel. We recall that we are dealing with a fluid with constant density $\tilde{\rho} = 1 \cdot 10^3 [Kg/m^3]$.
In this case we are dealing with a non-compliant problem since in the *RHS* there are terms arising from the inhomogeneous Dirichlet condition (equations 8.17 and 8.18). The output can be evaluated as follows:

$$\begin{aligned} s(\boldsymbol{\mu}) &= l(u(\boldsymbol{\mu}); \boldsymbol{\mu}) \\ &= \tilde{\rho} \left(\frac{20^2 \mu_3}{10^2 \, 1.1^2} \int_{\Gamma^1_{r_6}} \mathbf{u}(\boldsymbol{\mu}) \, d\Gamma^1_{r_6} - \frac{20^2 \mu_3 (7\mu_2 - 7)}{10^2 \, 1.1^2} \int_{\Gamma^2_{r_6}} \mathbf{u}(\boldsymbol{\mu}) \, d\Gamma^2_{r_6} + \right. \\ &\quad \left. + \frac{20^2 \mu_3 ((7\mu_2)/2 - 2)}{10^2 \, 1.1^2} \int_{\Gamma^3_{r_6}} \mathbf{u}(\boldsymbol{\mu}) \, d\Gamma^3_{r_6} \right) \end{aligned} \tag{8.29}$$

If we want to recover the *"square-effect"* on the error bound of the output, we have to solve an additional problem, called the *dual problem*. Since this work is focused on another topic, we will not exploit this additional tool offered by the *RB* methodology, see [RHP08, Gel10] for a detailed explanation of the *dual-problem*.
Since we have 3 parameters, we have fixed the pressure, that is $\mu_3 = 5$ and we left vary the angular velocity μ_1 (the Reyenolds number) and the eccentricity μ_2 of the rotor.
In Figure 44 we depict the output obtained with the *RB* method. In Figure 45a we depict the *error bound* $\Delta^s_N(\boldsymbol{\mu})$ on the output, in Figure 45b we depict the computational time savings offered by the *RB* method.
As we would have expected, the greater μ_1 $(= Re_D)$ is (that we recall is

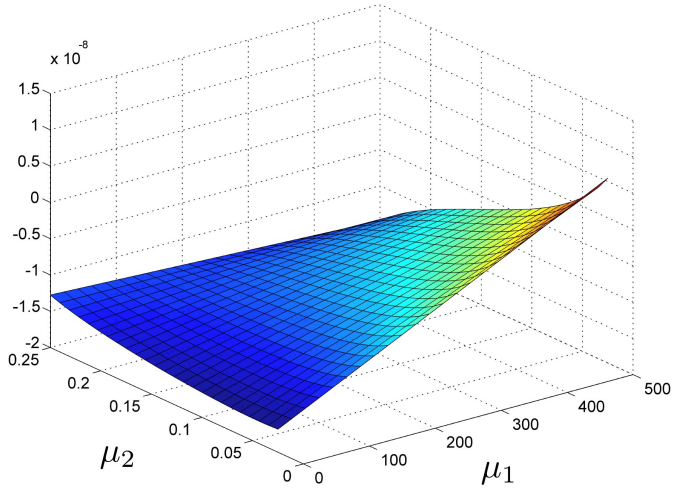
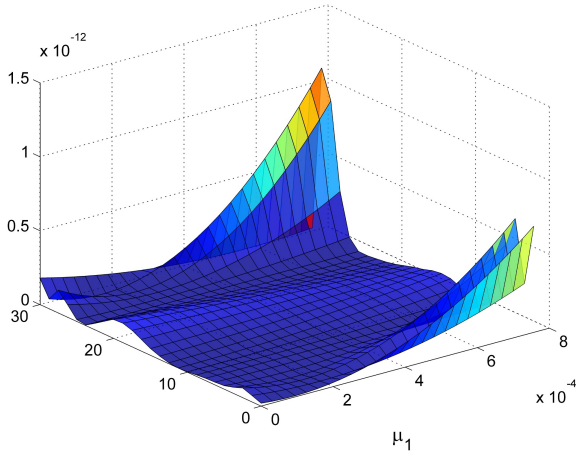
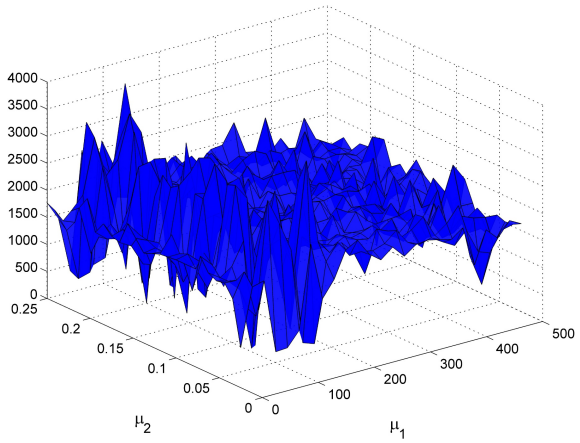


Figure 44: *VP* output: net flux

proportional to the rotor angular speed) and the more the mass net flux is.



(a) *VP* output error bound $\Delta_N^s(\boldsymbol{\mu})$



(b) $\frac{t_{FE}^s(\boldsymbol{\mu})}{t_{RB}^s(\boldsymbol{\mu})}$

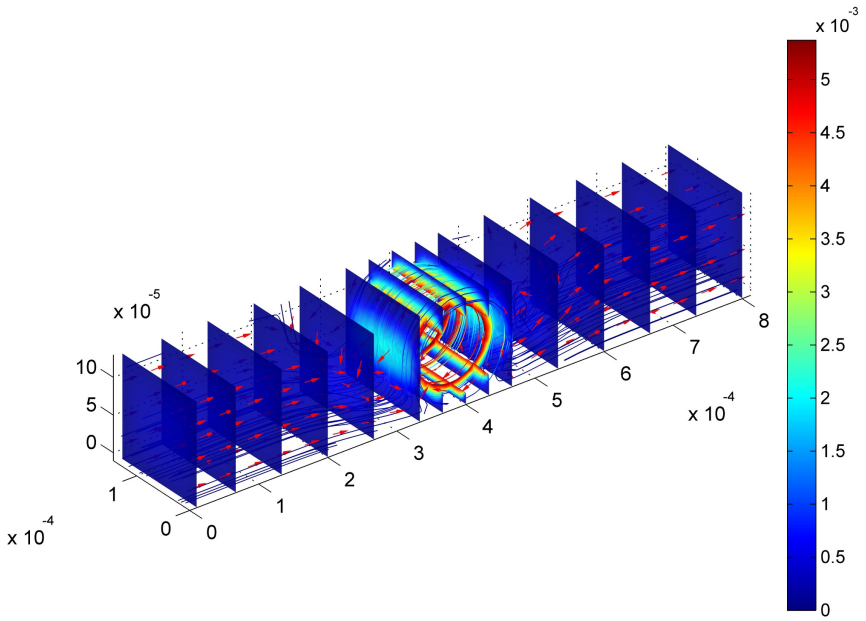
Figure 45: *VP* output results

Moreover, as it has been proved in [DKC07], the more the eccentricity (μ_2) increases, the more the mass flux increases. In fact a non-symmetry of the geometry creates an unbalancing in the velocity field that induces a net flux through the channel.
We note that, in this case, the computational saving is three orders of magnitude with respect to the *FE* method, which is a quite astonishing result.

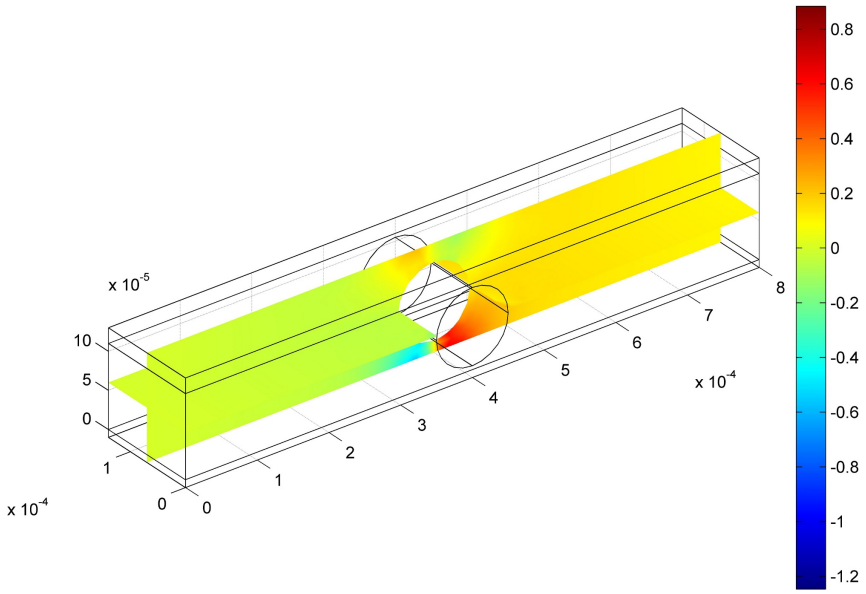
8.5.5 Visualization

We now report the visualization of some representative *RB* solutions. On the upper figures, we show the solution for different value of the parameters $\boldsymbol{\mu}$. In the first example, we show the solution on the reference domain. In the second example, we show the solution field after selecting a generic

combination of parameters in the parameter domain $\mathcal{D}$.

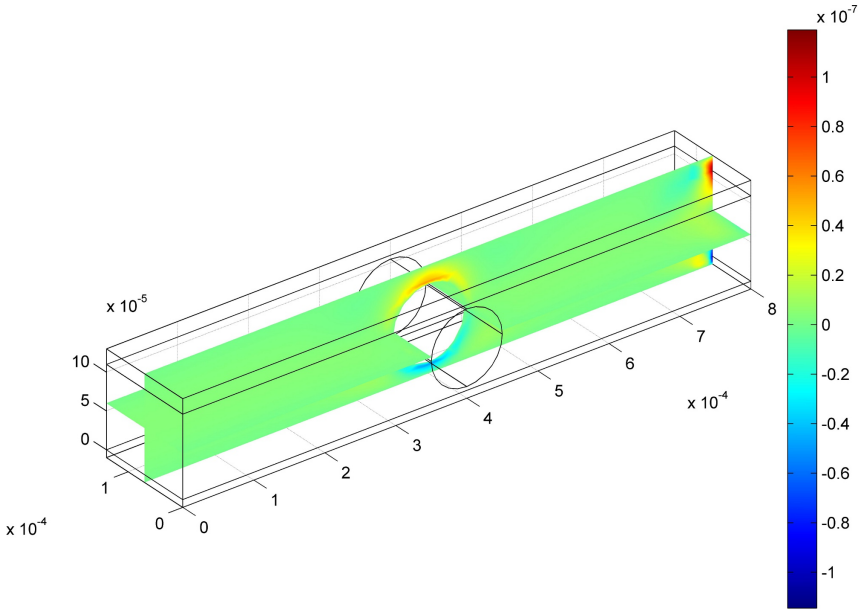


(a) RB solution: velocity field and streamlines

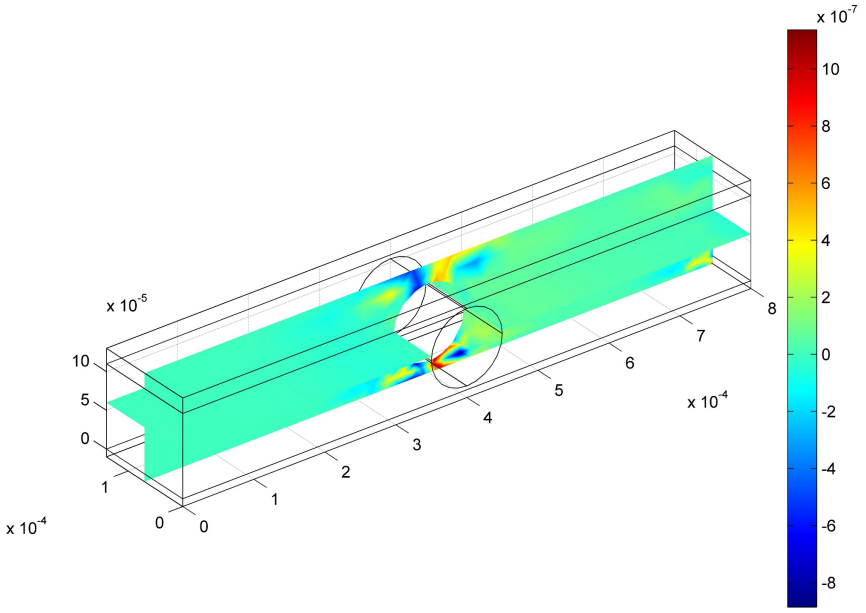


(b) RB solution: pressure field

Figure 46: Example of a representative solution and for $\boldsymbol{\mu} = \{100, 0.1, 0.1\}$, $\Delta_N^{en} \leqslant 8.95 \cdot 10^{-7}$



(a) RB solution: pointwise error on velocity field



(b) RB solution: pointwise error on pressure field

Figure 47: Pointwise error for $\mu = \{100, 0.1, 0.1\}$, $\Delta_N^{en} \leqslant 8.95 \cdot 10^{-7}$

9 SUMMARY AND CONCLUSIONS

In this thesis, we have studied the *RB* method for 3D *coercive* and *noncoercive*, *scalar* and *vectorial* problems. First, we have introduced the fundamentals of the *RB* method for *parametrized coercive and noncoercive elliptic PDEs*: (i) the affine decomposition to enable an Offline/Online splitting procedure, (ii) the *a-posteriori error estimates* to efficiently create the *RB* Greedy space (Offline) and inexpensive and rigorous error bounds for the *RB* solution and output (Online). In order to obtain an affine representation of the parametrized linear and bilinear forms we exploited an *affine geometry precondition* properly extended to the 3D case. The *RB* procedures, i.e. the Offline creation of the *RB* space and basis, the Online *RB* system solution, the input/output evaluation and post-processing were carried out thanks to the collaboration of two softwares, `COMSOL multiphysics` and `rbMIT`. We presented the applications of the *RB* method to a wide spectrum of engineering problem of interest: (i) a *steady thermal conductivity problem* in heat transfer; (ii) a *linear elasticity problem*, with regard to coercive problems; a (iii) Stokes flows application, concerning noncoercive problems.
We obtained in each applicative case a good and rapid convergence of either the *SCM* and the Greedy algorithm. Hence we experimentally proved that the *RB* method is very well suited to efficiently approximate also 3D problems with a rather involved parameter dependence, either physical and geometrical.
The Offline stage is quite expensive in the 3D context, nonetheless the very inexpensive and rigorous Online stage renders invaluable the worth of the *RB* method in many engineering field of interests: optimization, control, sensitivity analysis and real-time context.
In fact with the *RB* method we obtained (Online) a computational saving of at least two order of magnitude with respect to the *FE* approximation in the thermal and in the linear elasticity applications; corroborating the recent results obtained in parallel work [Gel10] dealing with 3D coercive problems. In the Stokes flow application, we obtained computational saving even of three order of magnitude; this result may have been possible even in consequences of a quite simple parameter dependence, nonetheless since the application is already able to seize the main properties and characteristic of a state-of-the-art viscous pump, see e. g.[SSG97, AHE04, DKC07], we concretely foresee a possible development of the *RB* method in the 3D fluid-dynamic field.

The applications chosen to exploit the *RB* approximation of 3D problems are a little, although representative, subset of the applications the

current efforts on the *RB* are devoted to. Different branches of the research field related to the *RB* method are dealing nowadays with a plenty of different problems and different context: the study of potential flows, see e. g.[Roz10], thermal problems, see [RNPD09, RHNP09], hemodynamics and biomedical devices optimization, see [MQR10], the study of nonlinear equations such as the Navier-Stokes problem, [DR09], the development of *RB* approximation in parabolic, see [NRHP10], and hyperbolic equations, see [NRP09]. Up to now, much efforts have been focused on 2D problems, our work constitutes the first steps in the natural extension and prosecution of the *RB* work in the 3D frame.

We also remark that in this work we have used a rather simple *RB* discretization, this is because in the 3D framework an automatic affine geometry precondition treatment is not available yet. On the contrary in the 2D case, the procedure has been developed and implemented, see `rbMIT` and for other works dealing with parametrically complex 2D geometries see [Qua05, Mil06].

Unfortunately, the affine parametrization of a geometry, is not enough *flexible* for some purposes, e. g.the shape-optimization of a vessel in a hemodynamic context or the design of an *air intake* in an aerodynamic context. Hence in the very few years, new techniques based on the so-called *free form deformation* has been developed in collaboration with the *RB* method, see [MQR10, LR10] and [RM10] for works dealing with shape optimization of cardiovascular geometries.

A possible and remarkable upgrade of our work would be to enrich the 3D geometry parametrization with these new free-from techniques. Recent work are devoted to this topic, which, in my opinion, will gainfully improve the power of *RB* method in the engineering context.

BIBLIOGRAPHY

[AHE04] M. Abdelgawad, I. Hassan, and N. Esmail. Transient Behavior of the Viscous Micropump. *Microscale Thermophysical Engineering*, 8:361–381, 2004.

[AK99] J.A. Atwell and B.B. King. Proper Orthogonal Decomposition for Reduced Basis Feedback Controllers for Parabolic Equations. *Mathematical and computer modeling*, 33(1-3):1–19, January 1999.

[Arp66] V.S. Arpaci. *Conduction Heat Transfer*. Addison-Wesley, 1966.

[ASB78] B.O. Almroth, P. Stern, and F.A. Brogan. Automatic choice of global shape functions in structural analysis. *AIAA Journal*, 16(7):525–528, January 1978.

[Bab71] I. Babuska. Error-bounds for Finite Element Method. *Computer Methods in Applied Mechanics and Engineering*, 16(4):322–333, December 1971.

[BTWG08] T. Bui-Thanh, K. Willcox, and O. Ghattas. Model Reduction for Large-Scale Systems with High-Dimensional Parametric Input Space. *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and material conference*, 30(6):3270–3288, October 2008.

[Como7a] Comsol. *COMSOL Multiphysics ©Modeling Guide*. COMSOL AB, 3.5a edition, 2007.

[Como7b] Comsol. *COMSOL Multiphysics ©User's Guide*. COMSOL AB, 3.5a edition, 2007.

[DKCo7] A.K. Da Silva, M.H. Kobayashi, and C.F.M. Coimbra. Optimal theoretical design of 2-D microscale viscous pumps for maximum mass flow rate and minimum power consumption. *International Journal of Heat and Fluid Flow*, 28:526–536, September 2007.

[DL00] R. Dautray and J.L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology, Vol.1*. Springer, Berlin, 2000.

[DR09] S. Deparis and G. Rozza. Reduced Basis Method for Multi-Parameter dependent Steady Navier-Stokes Equations: Applications to Natural Convection in a cavity. *Journal of Computational Physics*, 228:4359–4378, 2009.

[FM71] R. Fox and H. Miura. An approximate analysis technique for design calculations. *AIAA Journal*, 9(1):177–179, January 1971.

[FMM77] G. E. Forsythe, M. A. Malcolm, and C. B. Moler. *Computer Methods for Mathematical Computations*. Prentice-Hall Series in Automatic Computation. Wiley-VCH Verlag, Englewood Cliffs, New Jersey 07632, 1977.

[Gel10] F. Gelsomino. Exploration and Comparison of Reduced Order Modelling Techniques for Parametrized System. Master's thesis, Modeling and Scientific Computing (CMCS), Institute of Analysis and Scientific Computing (IACS), EPFL, January 2010.

[GMNP07] M.A. Grepl, Y. Maday, N.C. Nguyen, and A.T. Patera. Efficient Reduced Basis Treatment of Nonaffine and Nonlinear Partial Differential Equations. *European Series in Applied and Industrial Mathematics: Mathematical Modelling and Numerical Analysis*, 41(3):575–605, August 2007.

[HO08] B. Haasdonk and M. Ohlberger. Reduced Basis method for Finite Volume Approximations of Parametrized Linear Evolution Equations. *European Series in Applied and Industrial Mathematics: Mathematical Modelling and Numerical Analysis*, 42(2):277—-302, March 2008.

[HRSP07] D.B.P. Huynh, G. Rozza, S. Sen, and A.T. Patera. A Successive Constraint Linear Optimization Method for Lower Bounds of Parametric Coercivity and Inf–Sup Stability Constants. *Comptes Rendus de l'Acadèmie des Sciences*, 345(8):473–478, October 2007.

[IR97] K. Ito and S.S. Ravindran. A Reduced Basis Method For Control Problems Governed by PDEs. *Journal of Computational Physics*, In: Desch W., Kappel F., Kunisch K(eds) Control and Estimation of distributed parameter systems:153–168, 1997.

[IR01] K. Ito and S.S. Ravindran. Reduced Basis method for Optimal Control of Unsteady Viscous Flows. *International Journal of Computational Fluid Dynamics*, 15:97–113, October 2001.

[JA04] M. Jabbar and A. Azeman. Fast optimization of Electromagnetic-Problems: the Reduced Basis Finite Element Approach. *IEEE Transactions on Magnetics*, 40(4):2161–2163, May 2004.

[KV03] K. Kunischm and S. Volkwein. Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fluid Dynamics. *Mathematical and Computer Modeling*, 40(2):492–515, SIAM Journal on numerical analysis 2003.

[Lov44] A.E.H. Love. *A Treatise on the Mathematical Theory of Elasticity*. Dover publications, New York, 1944.

[LR10] T. Lassila and G. Rozza. Parametric Free-Form Shape Design with PDE Models and Reduced Basis Method. *Computer Methods in Applied Mechanics and Engineering*, 199:1583–1592, 2010.

[Meyoo] C.D. Meyer. *Matrix analysis and applied linear algebra*. SIAM, Philadelphia, Colorado State University, 1st edition, 2000.

[Milo6] R. Milani. Metodi a Basi Ridotte per la Risoluzione di Problemi Parametrizzati in Elasticità Lineare. Master's thesis, Modeling and Scientific Computing (CMCS), Institute of Analysis and Scientific Computing (IACS), EPFL and MOX, Politecnico di Milano, 2006.

[MQRo8] R. Milani, A. Quarteroni, and G. Rozza. Reduced Basis method for Linear Elasticity problems with many parameters. *Computer Methods in Applied Mechanics and Engineering*, 5(197):4812–4829, May 2008.

[MQR10] A. Manzoni, A. Quarteroni, and G. Rozza. Shape Optimization in Cardiovascular Geometries using Reduced Basis Method and Free-Form Deformations, 2010. submitted 2010.

[MYNA04] Barrault M., Maday Y., Nguyen N.C., and Patera A.T. An Empirical Interpolation Method: Application to Efficient Reduced Basis discretization of Partial Differential Equations. *Comptes Rendus Mathematiques*, 339(9):667–672, November 2004.

[Nag79] D.A. Nagy. Modal representation of geometrically nonlinear behaviour by the finite element method. *Computers & Structures*, 10(4):683–688, April 1979.

[Noo78] A.K. Noor. Recent advances in reduction methods for nonlinear problems. *Computers & Structures*, 13(1-3):31–44, June 1978.

[Noo82] A.K. Noor. On making large nonlinear problems small. *Computer Methods in Applied Mechanics and Engineering*, 34(2):955–985, September 1982.

[NP80] A.K. Noor and J.M. Peters. Reduced Basis Technique for Nonlinear Analysis of structures. *Computers & Structures*, 18(4):455–462, April 1980.

[NRHP10] N.C. Nguyen, G. Rozza, D.B.P. Huynh, and A.T. Patera. Reduced Basis Approximation and A-Posteriori Error Estimation for Parametrized Parabolic PDEs; Application to Real-

Time Bayesian Parameter Estimation. Technical report, EPFL-IACS report 11.2008, 2010. Computational Methods for Large Scale Inverse Problems and Uncertainty Quantification, in press.

[NRP09] N.C. Nguyen, G. Rozza, and A.T. Patera. Reduced Basis Approximation and A-Posteriori Error Estimation for Time Dependent Viscous Burgers Equation. *Calcolo*, 46(3):157–187, 2009.

[NVP05] N.C. Nguyen, K. Veroy, and A.T. Patera. *Certified Real-Time Solution of Parametrized Partial Differential Equations*. Yip S(ed) Handbook of materials modeling. Springer, Berlin, 2005.

[Pet89] J.S. Peterson. The Reduced Basis method for incompressible viscous flow calculations. *SIAM Journal on Scientific Computing*, 10(4):777–786, July 1989.

[PL87] T.A. Porsching and M.L. Lee. The Reduced Basis method for Initial Value Problems. *SIAM Journal on Numerical Analysis*, 24(6):1277–1287, December 1987.

[Por85] T.A. Porsching. Estimation of the Error in the Reduced Basis Method Solution of Nonlinear Equations. *Mathematics of Computation*, 45(172):487–496, October 1985.

[PR07] A.T. Patera and E.M. Rønquist. On the error behavior of the Reduced Basis Technique for Nonlinear Finite Element Approximations and A-Posteriori Error estimation for a Boltzmann model. *Computer Methods in Applied Mechanics and Engineering*, 196:2925—-2942, 2007.

[PR09] 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 2006-2010, Massachusetts Institute of Technology, Cambridge, MA, USA, 2009.

[PRV+02] C. Prud'homme, D. Rovas, K. Veroy, Y. Maday, A. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced Basis output bounds methods. *Journal of Fluid Engineering*, 124(1):70–80, March 2002.

[PRVP02] C. Prud'homme, D.V. Rovas, K. Veroy, and A.T. Patera. A mathematical and Computational Framework for Reliable Real-Time Solution of Parametrized Partial Differential Equations. *Mathematical Modeling and Numerical Analysis*, 36(5):747—-771, 2002.

[QSS00] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics,*. Texts in Applied Mathematics. Springer, New York, 2000.

[Qua05] A. Quaini. Metodi a Basi Ridotte per Problemi di Controllo in Fluidodinamica Ambientale. Master's thesis, Modeling and Scientific Computing (CMCS), Institute of Analysis and Scientific Computing (IACS), EPFL and MOX, Politecnico di Milano, 2005.

[Qua09] A. Quarteroni. *Numerical Models for Differential Problems*, volume 1 of *MS&A*. Springer, 2009.

[QV97] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer series on Computational Mathematics. Sringer, 1997.

[Rav02] S.S. Ravindran. Adaptive reduced order controllers for a thermal flow system using Proper Orthogonal Decomposition. *SIAM Journal on Scientific Computation*, 23(3):1924–1942, 2002.

[RHNP09] G. Rozza, D.B.P. Huynh, N.C. Nguyen, and A.T. Patera. Real-Time Reliable Simulation of Heat Transfer Phenomena. In *Heat Transfer Summer Conference Proceedings, HT-2009-88212*, S. Francisco, CA, US, July 2009. American Society of Mechanical Engineers. EPFL-IACS report 06.2009.

[RHP08] 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. *Archives of Computational Methods in Engineering*, 15:229—275, 2008.

[RM10] G. Rozza and A. Manzoni. Model Order Reduction by Geometrical Parametrization for Shape Optimization in Computational Fluid Dynamics. In *Proceedings of ECCOMAS CFD Conference, J.Pereira and A. Sequeira Editors*, Lisbon, Portugal, June, 2010. EPFL-IACS report 01.2009.

[RNPD09] G. Rozza, N.C. Nguyen, A.T. Patera, and S. Deparis. Reduced Basis Methods and A-Posteriori Error Estimators for Heat Transfer Problems. In *Heat Transfer Summer Conference Proceedings, HT-2009-88211*, S. Francisco, CA, US, July 2009. American Society of Mechanical Engineers. EPFL-IACS report 05.2009.

[Rov03] D. Rovas. *Reduced-Basis Output Bound Methods for Parametrized Partial Differential Equations*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2003.

[Roz08] G. Rozza. Reduced Basis for Stokes equations in domains with non-affine parametric dependence. *Computing and Visualization in Science*, 12(1):23–35, 2008. doi:10.1007/s00791-006-0044-7.

[Roz09] G. Rozza. An Introduction to Reduced Basis Method for Parametrized PDEs. In *In Applied and Industrial Mathematics in Italy, Vol. III, Series on Advances in Mathematics for Applied Sciences*, pages 508–519. Societá Italiana di Matematica Applicata e Industriale, 2009. EPFL-IACS report 01.2009.

[Roz10] G. Rozza. Reduced basis approximation and error bounds for potential flows in parametrized geometrie, 2010. Communications in Computational Physics, submitted 2010.

[RV06] G. Rozza and K. Veroy. On the Stability of the Reduced Basis method for Stokes Equations in parametrized domains. *Computer Methods in Applied Mechanics and Engineering*, 196:1244–1260, 2006.

[SSG97] M.C. Sharatchandra, M. Sen, and M. Gad-el-Hak. Navier–Stokes Simulations of a novel Viscous Pump. *Journal of Fluid Engineering*, 119(2):372–381, 1997.

[Wal93] P. Walker. *Chambers Dictionary of Materials Science and Technology*. Chambers Publishing, 1st edition, 1993.

[Woi05] P. Woias. Micropumps—Past, Progress and Future Prospects. *Sensors and Actuators B: Chemical*, 105(1):28–38, February 2005.

[WP02] K. Willcox and J. Peraire. Balanced Model Reduction via the Proper Orthogonal Decomposition. *AIAA Journal*, 40(11):2323—–2330, November 2002.

[Yos71] K. Yosida. *Functional Analysis*. New York: Springer-Verlag, 1971.

COLOPHON

This thesis was typeset with LaTeX using the *classicthesis* style, available via CTAN as `classicthesis`.
The bibliography has been compiled with the BibTeX reference management software.

Final Version as of June 27, 2010 at 22:49.

INDEX