UNIVERSITY OF TRENTO

Department of Mathematics


PhD in Mathematics

XXX CYCLE

# Model Order Reduction and its Application to an Inverse Electroencephalography Problem

PhD Student:
Juan Luis Valerdi Cabrera

Advisors:
Alberto Valli
Ana Alonso Rodriguez

June 2018

Referees:

Prof. Gianlugi Rozza
Scuola Internazionale Superiore di Studi Avanzati - SISSA

Prof. Michele Piana
Università Degli Studi Di Genova

*a mis padres*

# Agradecimientos

Quisiera expresar mi gratitud a las personas que me han seguido en la trayectoria del doctorado.

Agradezco a los profes Alberto Valli y Ana Alonso por el apoyo brindado en cada encuentro y por darme la oportunidad de hacer el doctorado con ellos. Gracias a mis colegas de doctorado por compartir momentos de risa y apoyo en nuestra larga jornada: Giulia Tini, Chiara Benazzoli, Marco Oppio, Elena Gaburro, Roberto Civino, Alberto Melati, Riccardo Longo, Daniel Wessel y Christian Contarino.

Quisiera dar una mención especial a Marco Oppio por estar a mi lado en los momentos más difíciles del doctorado y ayudarme a aclarar mi mente en muchas ocasiones. Por supuesto, no olvidaré cuanto me has hecho reír! Gracias amigo.

También quisiera dar unos agradecimientos especiales a mis tres compañeras de oficina, Giulia, Chiara e Elena por los buenos tiempos y las conversaciones que compartimos juntos.

Doy las gracias a mis amigos Sara Dari, Marion Llagiu, Bruno Llagiu, Valentina Tonon, Alejandra Huitrado y Giorgia Tosoni por la especial convivencia que tuvimos en la residencia y los buenos momentos que compartimos juntos.

El ultimo año fue posiblemente el más difícil pero a la vez uno de los más lindos gracias a la llegada de dos cosas: el squash y el amor. Gracias a mis compañeros de squash Ramu Peretti y Matteo Magagnini por la intensidad y el entusiasmo en el campo de juego.

El amor llego con el nombre de Elena Remonato. Gracias por cada sonrisa y abrazo que lleno mi vida de colores. También quisiera agradecer tu paciencia y apoyo, y sobretodo la fuerza que me brindaste para poder terminar. Gracias también a tu familia por acogerme como un miembro más, cuantos momentos en lindos!

Por ultimo quiero agradecer a las personas mas importantes de mi vida, mi familia, que aunque estuvieron lejos me apoyaron en todo momento de esta travesía: Mima, Abuelata, tia Aleida, tia Nidia, Aimara, Indira, Tomasito y Evelyn.

A mis padres les quiero dedicar un agradecimiento especial. Llegar a la culminación de este sueño es fruto de su amor sin límites. Todo lo que he obtenido en mi vida es gracias al sostén incondicional que me han dado. Hoy nos graduamos los tres.

El doctorado ha sido el momento de crecimiento personal más grande que he tenido y es en gran parte gracias a los momentos especiales que compartí con todas estas personas. Gracias de corazón.

Junio de 2018

# Ringraziamenti

Vorrei esprimere la mia gratitudine alle persone che mi hanno seguito nel percorso del dottorato.

Ringrazio i professori Alberto Valli e Ana Alonso per avermi dato l'opportunità di lavorare con loro e per il supporto fornitomi in ogni incontro. Grazie ai miei colleghi di studio per aver condiviso momenti di risate e per tutto il sostegno datomi in questo lungo viaggio: Giulia Tini, Chiara Benazzoli, Marco Oppio, Elena Gaburro, Roberto Civino, Alberto Melati, Riccardo Longo, Daniel Wessel e Christian Contarino.

Vorrei fare un ringraziamento speciale a Marco Oppio per essere stato al mio fianco nei momenti più difficili del dottorato ed avermi aiutato a chiarire la mente in molte occasioni. Certo, non dimenticherò quanto mi hai fatto ridere! Grazie amico.

Vorrei anche ringraziare in modo speciale i miei tre colleghi di ufficio, Giulia, Chiara ed Elena per i bei momenti e le conversazioni che abbiamo condiviso insieme.

Ringrazio i miei amici Sara Dari, Marion Llagiu, Bruno Llagiu, Valentina Tonon, Alejandra Huitrado e Giorgia Tosoni per la bella convivenza allo studentato e i bei momenti che abbiamo trascorso assieme.

L'ultimo anno è stato forse il più difficile, ma allo stesso tempo uno dei più belli grazie all'arrivo di due cose: lo squash e l'amore. Grazie ai miei colleghi di squash Ramu Peretti e Matteo Magagnini per l'intensità e l'entusiasmo in campo.

L'amore è arrivato col nome di Elena Remonato. Grazie per ogni sorriso ed abbraccio che hanno riempito la mia vita di colori. Vorrei anche ringraziare la tua pazienza e il tuo supporto e soprattutto la forza che mi hai dato per poter finire. Grazie anche alla tua famiglia per avermi accolto come un membro in più, quanti momenti belli!

Infine vorrei ringraziare le persone più importanti della mia vita, la mia famiglia. Chi, nonostante la lontananza, mi ha sostenuto in ogni momento di questo viaggio: Mima, Abuelata, zia Aleida, zia Nidia, Aimara, Indira, Tomasito ed Evelyn.

Ai miei genitori vorrei dedicare un ringraziamento speciale. Raggiungere il culmine di questo sogno è il frutto del vostro amore senza limite. Tutto ciò che ho ottenuto nella mia vita è stato possibile grazie al supporto incondizionato che mi avete dato. Oggi ci dottoriamo tutti i tre.

Il dottorato è stato il più grande momento di crescita personale che abbia avuto ed è in gran parte dovuto ai momenti speciali che ho condiviso con tutte queste persone. Grazie dal cuore.

Giugno 2018

# Contents

**Introduction** **1**

**1 Elements of RB and FOR Methods** **7**
1.1 Abstract Framework . . . . . 7
1.2 Reduced Basis Offline Stage . . . . . 10
1.3 Reduced Basis Online Stage . . . . . 16
1.4 Kolmogorov's n-width Estimates . . . . . 19
1.5 Fundamental Order Reduction Method . . . . . 27
1.6 Reduced Basis Error Estimates . . . . . 38
1.7 Empirical Interpolation Method . . . . . 45

**2 RB and FOR on an Epilepsy EEG Equation** **50**
2.1 Subtraction Approach Formulation . . . . . 51
2.1.1 RB on the Subtraction Approach. A Simple Case . . . . . 52
2.1.2 RB and FOR on the Subtraction Approach. A More Realistic Case . . . . . 64
2.2 Direct Approach Formulation . . . . . 71
2.2.1 RB on the Direct Approach . . . . . 72
2.3 Solution for the EEG Inverse Problem . . . . . 76

**3 jMOR** **89**
3.1 POD on the Laplacian operator . . . . . 90
3.2 jMOR Functions Reference . . . . . 98

**Conclusions** **101**

**Bibliography** **102**

# List of Figures


1 Epileptic spike detected with EEG . . . . . 4

1.1 Reduced basis set relations . . . . . 11

2.1 Domain $\Omega$ divided by layers. . . . . 56
2.2 RB in a compact set, isotropic subtraction approach case. Singular values and error estimates . . . . . 57
2.3 RB in a compact set, isotropic subtraction approach case. Some test errors in $\|\cdot\|_{H^1(\Omega)}$ 57
2.4 RB for $\mathcal{P} \approx \Omega$, isotropic subtraction approach case. Singular values and error estimates . . . . . 62
2.5 RB for $\mathcal{P} \approx \Omega$, isotropic subtraction approach case. Some test errors in $\|\cdot\|_{H^1(\Omega)}$ . 62
2.6 Isotropic subtraction approach case. Truth solutions for different sources . . . . . 63
2.7 Subdomains of $\Omega$: $\Omega_0, \Omega_1, \Omega_2$. . . . . 64
2.8 EIM in a compact set, anisotropic subtraction approach case. EIM sampling on $u_0$ . 66
2.9 RB in a compact set, anisotropic subtraction approach case. Sampling schemes . . 67
2.10 RB in a compact set, anisotropic subtraction approach case. Singular values and error estimates . . . . . 68
2.11 RB in a compact set, anisotropic subtraction approach case. Exact error for different parameters . . . . . 68
2.12 MOR in a compact set, anisotropic subtraction approach case. Exact error difference between RB and FOR . . . . . 70
2.13 MOR in a compact set, anisotropic subtraction approach case. Comparison between RB and FOR . . . . . 70
2.14 RB in a compact set, isotropic direct approach case. Singular values . . . . . 73
2.15 RB in a compact set, isotropic direct approach case. Projection error estimates in $\Xi_s$ 73
2.16 RB in a compact set, isotropic direct approach case. $l_2$ error with $\boldsymbol{\mu} \in \Xi_s$ . . . . . 74
2.17 RB in a compact set, isotropic direct approach case. $l_2$ error with $\boldsymbol{\mu} \notin \Xi_s$ . . . . . 74
2.18 Isotropic direct approach case. Two truth solutions . . . . . 75
2.19 RB in a compact set, isotropic direct approach case. Reduced basis functions . . . . 75
2.20 Simulated annealing on the EEG inverse problem. Average computing times . . . . 79
2.21 Approximation of the inverse EEG function. Average error in the training set $\Xi_{tr}$ and the test set $\Xi_{te}$ . . . . . 85

2.22 Approximation of the inverse EEG function, q = 16. Comparison between actual and estimated parameter, Example 1. . . . . 86
2.23 Approximation of the inverse EEG function, q = 16. Comparison between actual and estimated parameter, Example 2. . . . . 86
2.24 Approximation of the inverse EEG function. Average error in the training set $\Xi_{\text{tr}}$ and the test set $\Xi_{te}$ . . . . . 87
2.25 Approximation of the inverse EEG function, q = 128. Comparison between actual and estimated parameter, Example 1. . . . . 87
2.26 Approximation of the inverse EEG function, q = 128. Comparison between actual and estimated parameter, Example 2. . . . . 88

3.1 jMOR POD applied to the Poisson equation. Singular values and error estimation . 96
3.2 jMOR POD applied to the Poisson equation. Exact errors . . . . . 97

# Introduction

A scientific or engineer simulation is a way to reproduce a phenomenon of the real world in order to predict an output given some input of the modeled system. These simulations have great impact in everyday life as well as in engineering and science itself, e.g. weather forecast [133], civil engineering [78], biology [119], etc. A huge body of simulations are modeled by partial differential equations (PDE), which represent a wide variety of phenomena such as sound, heat, electrostatics, electrodynamics, fluid flow, elasticity, etc. [28, 67, 77, 139].

Nowadays there are diverse successful methods for numerically solving PDEs like finite element methods (FEM) [52, 132], finite volume methods (FVM) [120], spectral methods [44, 45], among others. Sometimes computing a *high-fidelity* or *full-order* approximation of a very complex PDE through these methods can be very demanding and the computation in supercomputers could take from hours to days to finish. Furthermore, in the case of PDEs depending on parameters, i.e. *parameterized PDE*, the computation of solutions for many different parameters may be extremely expensive or even impossible due to time constraint, like for example, in *inverse problems* or *optimal control* where an iterative procedure needs to solve the *forward problem* several times [90]. This setting where a PDE has to be solved numerous times with different configuration of parameters is commonly known as a *many-query context*.

A simple example of a parametrized PDE is shown by the following equation: Given $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^p$, solve

$$\begin{cases} -\operatorname{div}(\sigma(\boldsymbol{\mu})\nabla u) &= f(\boldsymbol{\mu}) & \text{in } \Omega, \\ u &= 0 & \text{on } \Gamma_D, \\ \sigma(\boldsymbol{\mu})\nabla u \cdot \mathbf{n} &= 0 & \text{on } \Gamma_N. \end{cases}$$

where the parameter set $\mathcal{P}$ represents a compact subset of $\mathbb{R}^p$, $p \geq 1$; the domain $\Omega \subset \mathbb{R}^d$, $d \geq 1$, denotes an open bounded and connected region with Lipschitz boundary, $\partial\Omega = \Gamma_D \cup \Gamma_N$ the boundary of $\Omega$ and $\mathbf{n}$ the outward unit normal vector on $\partial\Omega$. Here $\sigma$ and $f$ are two parameter-dependent functions

$$\sigma, f : \mathcal{P} \times \mathbb{R}^d \to \mathbb{R}.$$

In general, the parameters may lay in any position of the model, i.e. in the domain, boundary or initial conditions, source terms, or in the physical properties.

A way to surpass high computational expenses of a many-query context is to reduce the complexity of the high-fidelity problem through *model order reduction* (MOR) [29, 31, 83, 131]. The idea of MOR is to compute an approximation of the full-order problem from a small model in terms of degrees of freedom while keeping a good input-output accuracy. To take advantage of this technique the complexity of the *reduced-order model* (ROM) has to be independent of the dimension of the original problem.

MOR is not a new idea, it has been used since the 70s in many-query design evaluation [70] and parameter continuation methods [6]. Afterwards, they were further developed to other problems like differential equations through *reduced basis* (RB) methods [24, 69, 105]. Nevertheless, they did not have strong emphasis in the *certification* of the error which is very important because MOR techniques may be susceptible to inaccuracies on the solution. To bring rigor to these methods, a-posteriori error estimates and effective sampling strategies were researched and published at the beginning of the 00s [57, 130, 135, 165]. Nowadays, reduced basis methods are widely applied and very actively researched in numerous fields, e.g. Maxwell equations [49, 50, 51], Stokes equations [71, 134, 138], homogenization [37, 121], multi-scale methods [1, 2, 98], parabolic equations [74, 76], nonlinear problems [46, 75, 95], optimal control [59, 136, 155, 156], uncertainty quantification [38, 87, 122] and many others.

There are two main algorithms to apply RB to PDEs: *proper orthogonal decomposition* (POD) and *greedy*.

The idea of POD is to represent a collection of solutions of the differential equation with an orthonormal basis which is optimal in a least-square sense. This representation is built from a small dimensional space and retains the most important information of the solutions. One of its first application appeared in turbulent flows [11, 12, 147, 148] and nowadays is used in many other kind of equations [13, 42, 96, 97, 102, 103, 155].

As an alternative to POD we could use greedy algorithms. Differently from POD, a greedy approach does not need a precomputed collection of solutions, which can save computational time in many cases. It uses a-posteriori error estimates to select the most meaningful parameters to construct the reduced model while minimizing the computation of full-order solutions. The first greedy method was introduced in the 70s [64] and was related to optimization problems. Later by the 00s greedy methods were studied in the RB context [113, 114, 137, 142, 165], mainly for a-posteriori error estimation and a-priori convergence.

In general, RB methods use an *offline-online* approach which means there are two stages:

- *offline stage*: take advantage of the parametric dependence of the PDE and, for a selection of parameters $\boldsymbol{\mu}_j \in \mathcal{P}$ compute the corresponding high-order solutions, also called in this context *snapshots*, that will constitute the reduced basis. This stage is done only once.
- *online stage*: use the small dimension of the RB to compute a fast approximation of the high-order solution for a given parameter in $\boldsymbol{\mu} \in \mathcal{P}, \boldsymbol{\mu} \neq \boldsymbol{\mu}_j$.

The computations in the offline stage are carried out by usual numerical techniques, e.g FEM. This stage can take a huge amount of time to finish but once done the results are used and stored to build a reduced model. Then, the offline-online decoupling replaces the large algebraic system of the legacy methods by a smaller one, whose dimension is controlled by the dimension of the RB. This approach gives remarkable speedups to the point of *real time* evaluation of PDEs [10, 57, 122, 130, 134, 164].

A case in which we can obtain real time computations in the online stage is when we have an affine representation of the parametric part of the PDE, i.e.

$$f(x;\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu}) f_q(x). \tag{1}$$

This helps having less computational burden in the online stage because many operations can be precomputed in the offline stage. For example, given an linear operator $A$ we obtain

$$Af(x;\mu) = \sum_{q=1}^{Q_f} \Theta_f^q(\mu) A f_q(x),$$

and after we have computed $Af_q(x)$ for all $q = 1, \dots, Q_f$, evaluating $Af$ for different parameters is inexpensive.

In the case there is not an affine representation of the parametric part we can return to that case by using the *empirical interpolation method* (EIM). It was introduced in [23] and since then is the standard method for computing affine representations. This method is iterative and hierarchical, it achieves exponential convergence rate for analytical functions and is applicable in general domains.

The fast computation in the state of the art of RB methods is outstanding. Nevertheless, one of the aim of this thesis is to take it further.

**Objective 1**: *Improve computational times in the online stage of reduced basis methods while retaining good accuracy.*

To achieve this objective we propose two ideas: the *Fundamental Order Reduction Method* (FOR) and *offline error estimators*.

In Section 1.5 we propose the fundamental order reduction method for solving PDEs dependent of parameters. Differently from POD and greedy, the FOR method uses nonlinear combinations of the snapshots to build the new basis and does not solve the PDE from the reduced model in the online stage. In the online stage the only operations executed are simple affine evaluations like in (1).

The FOR method is not completely new, it appears in [131], but not as solver of parametric PDEs, instead as an estimator on the accuracy we could obtain from a reduced basis in finite dimensional spaces. We expand all the results found in [131] to infinite dimensional spaces and introduce new a-priori convergence results.

We also discuss in Section 1.5 some disadvantages, for example, FOR cannot be applied to all kinds of PDEs so it is not as general as POD and greedy. Also, some of the error estimates can be difficult to obtain in infinite dimensional spaces, whereas in finite dimensional spaces they are easy to compute.

For cases where FOR cannot be used, we propose in Section 1.6 some offline error estimators for standard RB techniques. After computing a RB solution the most common procedure is to estimate an a-posteriori error to certify good accuracy. Offline estimators are a class of estimators that move a-posteriori operations to the offline stage, reducing in this way the load of computations in the online stage.

In Section 1.6 we introduce two of them: *Lipschitz offline estimator* (Loe) and *Chebychev offline estimator* (Coe). Both use the regularity of the solution map to compute estimations. Loe in particular use the Lipschitz constant of the solution map and the distance of the snapshots to bound the RB error. In the case we have more regularity in the solution map we can prove that the residual will have the same kind of regularity. Coe exploits this fact to interpolate the

residual using Chebychev polynomials and obtain in the offline stage an approximant of the residual.

The rest of Chapter 1 serves as a base for Section 1.5, Section 1.6 and the later Chapter 2. We can find in Section 1.1-1.3 the basic results of the RB methods and the offline-online decoupling. In Section 1.4 we explain how to obtain Kolmogorov $n$-width estimates for proving a-priori convergence from RB spaces. And finally in Section 1.7 we present the EIM algorithm to compute affine representations. In all these sections we have also obtained some original results that are strongly interconnected with the main results in Section 1.5 and Section 1.6.

Chapter 2 focus in solving the electroencephalography (EEG) equation

$$\begin{cases} \operatorname{div}(\sigma \nabla u) = \operatorname{div}(\mathbf{p}_0 \delta_\mu) & \text{in } \Omega, \\ (\sigma \nabla u) \cdot \mathbf{n} = 0 & \text{on } \partial\Omega. \end{cases} \tag{2}$$

One application of this equation is detecting the position where an epilepsy seizure begins inside the brain. The parameter that controls the solution is the point $\mu$ where the Dirac delta function is placed. The only information we have to find $\mu$ is the electrical potential read by a collection of electrodes positioned in the head, see Figure 1.

Figure 1: Epileptic spike detected with EEG

In mathematical terms, we know $u(x_1), \dots, u(x_q)$ for $q$ points in the boundary $x_1, \dots, x_q \in \partial\Omega$, and we wish to find the polarization $\mathbf{p}_0$ and the position $\mu$ that best fits the evaluations $u(x_i), i = 1, \dots, q$.

Equation (2) is hard to treat because of the singularity of the delta function. Moreover, the general theory of RB methods do not cover this kind of equation.

Chapter 2 presents two ways to look at the EEG equation: *direct approach* [9, 163] and *subtraction approach* [14, 106, 169]. Two finite element schemes stem naturally from these formulations. For the direct approach the EEG solution is directly approximated from a finite dimensional space, while in the subtraction approach one first removes the singularity and finally has to face a standard PDE.

A significant result of this thesis is the proof that RB methods do not give reasonable solutions when applied to the direct approach, see Section 2.2.1. This seems to prevent any possible application of RB methods to the EEG problem. Instead, we show that for the alternative

formulation, the subtraction approach, RB methods furnish a suitable procedure for finding a solution in an efficient way.

We present theoretical and numerical results of the RB and FOR methods in Sections 2.1.1-2.1.2. Having known that the direct approch is not suitable for these methods, we focus on the subtraction approach. The numerical results related to it show that FOR is faster and more accurate than RB and therefore more convenient for solving the inverse problem.

The most common way of solving this inverse problem is to use iterative methods like simulated annealing. These kind of methods compute the forward problem several times so we are inside a many-query context. As explained before, we can decrease the computational effort by using model order reduction techniques. Even if we can run the online stage of RB methods very fast, we still have to execute many iterations, and therefore the inverse problem is not solved in real-time.

Another aim of this thesis is therefore to find some new procedures for the following:

**Objective 2**: *Achieve real-time solutions of inverse problems using model order reduction techniques.*

The idea that permits to obtain real-time solutions of inverse problems is avoiding iterative methods. In Section 2.3 we introduce a general methodology for solving inverse problems like the EEG using universal approximation theory [79]. This theory is the base of artificial neural networks and has been successful in numerous fields like supervised learning [68, 81, 101], reinforcement learning [144, 145, 146], inverse problems in image processing [91, 93, 153], etc.

Following this methodology we build a map

$$\varphi : \mathbb{R}^q \to \mathcal{P},$$

with the property that, given q readings of $u$ on the boundary, it returns a good approximation of the parameters that generated $u$. In the same way as RB methods, this methodology has an offline-online decoupling.

In the offline stage we constructs the map $\varphi$ through an optimization problem that fits $\varphi$ to the inverse problem. This step needs many solution of the forward problem, hence we use a reduced-order model. The function $\varphi$ is given by a simple linear combinations of smooth functions, therefore the online stage, which is the evaluation of $\varphi$, turns out to be very fast.

Let us come now to the final part of this thesis. There are several software libraries that we can use to work with RB methods, here is a comprehensive list:

- *rbMIT* [88]: is a package implemented in Matlab and companion to the book [127]. This library is very complete from the feature point of view and many examples are also available. Truth solutions are computed with FEM.
- *RBmatlab* [62, 63]: is another Matlab implementation with methods for linear and non-linear problems and general parameter dependency. The offline stage can be done with FEM, FVM or discontinuous Galerkin discretization [128].

- *Dune-RB* [63]: is a C++ module of the DUNE library [26, 27] with a strong focus on parallelism of the offline stage.
- *RBniCS* [20]: is a package developed in Python and companion to the book [83]. It has well explained tutorials to get started quickly and use FEniCS [7] as backend to compute high-order solutions.
- *pyMOR* [118]: is another Python library with good integration of external Python PDE solvers like FEniCS and Python bindings of deal.II [21] and DUNE.

The three main languages where RB has been implemented are Matlab, Python and C++ which are nowadays the most used languages in numerical computations. Nevertheless, we want to expand the availability of RB packages, as described here below.

**Objective 3**: *Implement an open source RB package in the Julia programming language.*

Julia [34, 35, 36] is a recent programming language which has been developed specifically for scientific computing. It has increased in popularity within the scientific community in the last years thanks to its C/Fortran level of performance, high-level dynamic programming like Python, parallelism design and mathematical-like syntax. Moreover, despite of being a new language, many mathematical packages have been developed with high level of maturity, e.g numerical optimization [86, 110, 161], numerical linear algebra [92, 126, 170], numerical quadrature [149, 158].

We have called jMOR [162] the RB package implemented in this thesis. This package has a black-box philosophy and use FEniCS as default to compute FEM solutions. Julia gives two advantages to jMOR:

1. **High computational speed**: Julia rivals the performance of C/Fortran which is very important for real-time computations. Also, it has metaprogramming [58] capabilities through macros, which can create specialized code for every problem and help to mitigate possible speed problems of black-box libraries.
2. **Easy to extend**: Julia syntax is similar to Matlab which is easy to read and write for mathematicians. Furthermore, jMOR can be extended easily to use any standard PDE solver from Python, Fortran or C++.

Section 3.1 shows how to use jMOR through a simple tutorial. This tutorial explains the most basic commands that are universally applicable to any equation. In Section 3.2 we describe some functions not included in Section 3.1 which are valuable for every day computations.

The version of jMOR that we present in Chapter 3 is jMOR v0.1. It has the whole backend code and data structures implemented for integrating new MOR methods. This version has POD and EIM functionality, useful functions for mathematical analysis and a simple macro for executing online computations.

# Chapter 1

# Elements of RB and FOR Methods

Reduced basis (RB) and Fundamental Order Reduction (FOR) method are Model Order Reduction (MOR) algorithms that are built on top of traditional numerical methods for differential equations to speed up computation times of parameterized problems. This chapter presents their offline-online methodology, main algorithms and error estimation results.

The organization of the content is divided in seven sections: Section 1.1 introduces a basic background in variational problems and RB methods; Section 1.2 explains the RB offline stage and POD specifically; Section 1.3 presents the RB online stage; Section 1.4 presents the theory behind a-priori estimates; Section 1.5 introduces FOR for solving parameterized equations; Section 1.6 shows new results in the computation of a-posteriori error estimates; and Section 1.7 presents the EIM algoritm for computing affine representations. The main references for this chapter are [52, 53, 83, 131, 132].

## 1.1 Abstract Framework

This section introduces from an abstract perspective the most basic definitions and results of variational problems which are related with PDEs, thereafter we present the general methodology of RB methods.

Let $V$ be a real Hilbert space, $V'$ the dual space of $V$, $a$ a bilinear form $a : V \times V \to \mathbb{R}$ and $f \in V'$, then we can consider the following variational problem:

Find $u \in V$ such that

$$a(u, v) = f(v) \qquad \forall v \in V. \tag{1.1}$$

We obtain the well-posedness of this problem by Lax-Milgram theorem.

**Theorem 1** (Lax-Milgram)**.** *Suppose $a(\cdot,\cdot)$ is continuous, i.e. there exists $\gamma > 0$ such that*

$$|a(u, v)| \leq \gamma \|u\|_V \|v\|_V, \qquad \forall u, v \in V,$$

*and coercive, i.e. there exists $\alpha > 0$ such that*

$$a(u, u) \geq \alpha \|u\|_V^2, \qquad \forall u \in V. \tag{1.2}$$

*If $f(\cdot) \in V'$, then there exists a unique solution for the variational problem (1.1). Furthermore, the solution satisfies*

$$\|u\|_V \leq \frac{1}{\alpha}\|f\|_{V'}. \tag{1.3}$$

*Proof.* See [132]. □

From a finite-dimensional subspace of $V$ is possible to compute an approximated solution of (1.1). Let for every $h > 0$, $V_h \subset V$ be a subspace of dimension $N_h$, therefore solving problem (1.1) in this subspace is equivalent to:

Find $u_h \in V_h$ such that

$$a(u_h, v_h) = f(v_h) \qquad \forall v_h \in V_h. \tag{1.4}$$

If $a(\cdot,\cdot)$ is continuous and coercive in $V$ and $f(\cdot)$ bounded in $V$, then we obtain these properties also in any subspace of $V$, thus (1.4) has a unique solution in $V_h$ because of Lax-Milgram theorem.

Let $\{\varphi^j\}_{j=1}^{N_h}$ denote a basis of $V_h$, consequently the variational problem (1.4) is equivalent to the linear system:

$$\mathbf{A}_h \mathbf{u}_h = \mathbf{f}_h, \tag{1.5}$$

where

- $\mathbf{A}_h \in \mathbb{R}^{N_h \times N_h}$ is the *stiffness matrix* with components $(\mathbf{A}_h)_{ij} = a(\varphi^j, \varphi^i)$.
- $\mathbf{f}_h \in \mathbb{R}^{N_h}$ is the *load vector* with components $(\mathbf{f}_h)_i = f(\varphi^i)$.
- $\mathbf{u}_h \in \mathbb{R}^{N_h}$ is the *solution vector* with coordinates $(u_h^{(1)}, \ldots, u_h^{(N_h)})$ in $V_h$.

The convergence of $u_h$ to $u$ when $V_h$ approximate to $V$ is a result of Cea's lemma:

**Lemma 1** (Cea)**.** *If $a(\cdot,\cdot)$ and $f(\cdot)$ satisfy the conditions of Lax-Milgram theorem in $V$ and $u$ is the solution of (1.1), then the following holds for the solution $u_h$ of (1.4),*

$$\|u - u_h\|_V \leq \frac{\gamma}{\alpha} \inf_{v_h \in V_h} \|u - v_h\|_V,$$

*where $\gamma$ and $\alpha$ are the continuity and coercivity constant respectively.*

*Proof.* See [132]. □

It may happen that problem (1.4) belongs to a collection of similar problems indexed by a parameter $\boldsymbol{\mu}$, like for instance, the one described in the Introduction:

$$\begin{cases} -\operatorname{div}(h(\boldsymbol{\mu}))\nabla u) &= s(\boldsymbol{\mu}) & \text{in } \Omega, \\ u &= 0 & \text{on } \Gamma_D, \\ h(\boldsymbol{\mu})\nabla u \cdot \mathbf{n} &= 0 & \text{on } \Gamma_N. \end{cases}$$

Variational formulation of a parameterized PDE does not vary from a non-parameterized one, the procedure to arrive to the variational problem is the same but resulting a parameterized variational problem instead.

More concretely, let a parameter set $\mathcal{P}$ be a compact subset of $\mathbb{R}^p$, $p \in \mathbb{N}$, a *parametric or parameterized variational problem* is defined by:

Given a parameter $\boldsymbol{\mu} \in \mathcal{P}$, find $u(\boldsymbol{\mu}) \in V$ such that

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \qquad \forall v \in V. \tag{1.6}$$

For a parameter $\boldsymbol{\mu} \in \mathcal{P}$, Lax-Milgram theorem guarantees again the existence and uniqueness of a solution if its hypothesis are satisfied for $a(\cdot, \cdot; \boldsymbol{\mu})$ and $f(\cdot; \boldsymbol{\mu})$. The procedure to compute an approximation follows as the non-parametric case, i.e. we obtain a discrete approximation of $u(\boldsymbol{\mu})$ with a finite-dimensional subspace of V:

Given $\boldsymbol{\mu} \in \mathcal{P}$, find $u_h(\boldsymbol{\mu}) \in V_h$ such that

$$a(u_h(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) = f(v_h; \boldsymbol{\mu}) \qquad \forall v_h \in V_h. \tag{1.7}$$

In other words, the approximation of each element of the *solution manifold*

$$\mathcal{M} := \{u(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \mathcal{P}\}$$

will belong to the *computable solution manifold*

$$\mathcal{M}_h := \{u_h(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \mathcal{P}\} \subset V_h.$$

The discrete problem (1.7) is again equivalent to a linear system like (1.5),

$$\mathbf{A}_h(\boldsymbol{\mu})\mathbf{u}_h(\boldsymbol{\mu}) = \mathbf{f}_h(\boldsymbol{\mu}), \tag{1.8}$$

the only difference resides in the introduction of a parametric dependency in each component of the equation.

Having parametric problems usually involves constructing the linear system (1.8) for many parameters and computing its solution. This may take long CPU time when the dimension of $V_h$ is large. A way to overcome this difficulty is to exploit the parametric dependencies that could exist between different solutions and reduce the dimension of the problem, this is in fact the main idea of the RB methods.

Reduced basis methods try to obtain a precise approximation of the elements of $\mathcal{M}_h$ in a *uniform way* from a *small* subspace $V_N \subset V_h$. The subspace $V_N$ is built from a set of solutions $\{u_h(\boldsymbol{\mu}^i), \boldsymbol{\mu}^i \in \mathcal{P},\ i = 1, \ldots, n_s\}$. Here the parameters selected should be a good representation of the whole parameter set $\mathcal{P}$.

In a uniform way means that with a fixed subspace $V_N$, a RB method has to be able to compute an element of $V_N$ that approximates precisely $u_h(\boldsymbol{\mu})$ regardless of the parameter selected. On the other hand, the requirement of $V_N$ being of small dimension, in the sense $N \ll N_h$, will give the possibility to improve CPU times of computations.

Notice that RB methods do not approximate the elements of $\mathcal{M}$ directly but $\mathcal{M}_h$. Therefore, for approximating an element of $\mathcal{M}$ from a RB method it is necessary that $\mathcal{M}_h \approx \mathcal{M}$ and the RB algorithm compute approximations of the elements of $\mathcal{M}_h$ with good accuracy. This is the reason why the elements in $\mathcal{M}_h$ are usually named in the literature as *truth solutions* because in the context of reduced basis methodology they are considered the "real" solutions.

Reduced basis methods are composed of two stages, one called *offline* and the other *online*. The first stage is the offline which usually consists in two main steps:

1. Discretization of the parameter set $\mathcal{P}$.
2. Construction of a basis for $V_N$.

This stage is usually done only one time and is concerned with the preparation and precomputations of all the mathematical structures that will be used in the online phase. The second stage has only one core step and is the final step in a RB method:

3. Computation of the RB solution $u_N(\boldsymbol{\mu}) \in V_N$ for a given $\boldsymbol{\mu} \in \mathcal{P}$.

The following sections go in more details on each of these steps. First we describe the offline stage and later the online stage.

## 1.2 Reduced Basis Offline Stage

The initial step of the offline stage is the discretization of $\mathcal{P}$. This is because working directly with $\mathcal{M}_h$ is difficult if $\mathcal{P}$ has infinite elements, which is the usual case.

The discretization of $\mathcal{P}$ for $n_s$ points will be denoted by the *sampling or training set*

$$\Xi_s := \{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^{n_s}\} \subset \mathcal{P},$$

and its *discrete computable solution manifold* by

$$\mathcal{M}_h^{n_s} := \{u_h(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \Xi_s\} \subset \mathcal{M}_h.$$

In general, the selection of an optimal sampling set is difficult and problem dependent, it has to be small enough for affordable CPU times of the offline stage and at the same time capture the parametric dependency of the solutions. One simple strategy to construct $\Xi_s$ is to choose the parameters uniformly over the parameter set, which is usually suitable for low-dimensional $\mathcal{P} \subset \mathbb{R}^p, p \leq 3$. For additional strategies consult [40, 43, 65, 84, 112].

Once $\mathcal{P}$ is discretized the second step of the offline phase is to construct $V_N$. Building this space vary depending on the algorithm used, but their common ground is that they take a certain amount of solutions from $\mathcal{M}_h^{n_s}$ and use linear combinations of them to construct $V_N$.

For example, let

$$\{u_h(\boldsymbol{\mu}^1), \dots, u_h(\boldsymbol{\mu}^N)\} \subset \mathcal{M}_h^{n_s}$$

be a collection of solutions or *snapshots* corresponding to a set of N selected parameters

$$S_N := \{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N\} \subset \Xi_s.$$

A possible linear combination of the snapshots could be the outcome of an orthonormalization process [137] resulting a set of N functions

$$\{\zeta_1, \dots, \zeta_N\}.$$

The functions in $\{\zeta_1, \dots, \zeta_N\}$ are called *reduced basis functions* and generate the *reduced basis space*

$$V_N := \operatorname{span}\{\zeta_1, \dots, \zeta_N\} = \operatorname{span}\{u_h(\boldsymbol{\mu}^1), \dots, u_h(\boldsymbol{\mu}^N)\} \subset V_h.$$

The offline stage finish at this point and once $V_N$ is defined, the online stage takes care of computing the elements of this subspace. For each $u_h(\boldsymbol{\mu})$, $\boldsymbol{\mu} \in \mathcal{P}$, the online stage compute an approximation $u_N(\boldsymbol{\mu}) \in V_N$ which belongs to the *reduced basis manifold*

$$\mathcal{M}_N := \{u_N(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \mathcal{P}\} \subset V_N.$$

Notice that even if the snapshots belong to $\mathcal{M}_h^{n_s}$, the reduced basis functions do not longer belong to $\mathcal{M}_h^{n_s}$ after they have been orthonormalized or gone through another kind of linear combination. Figure 1.1 gives an illustration of the set relations defined until now, note though, some interceptions between the sets could happen.

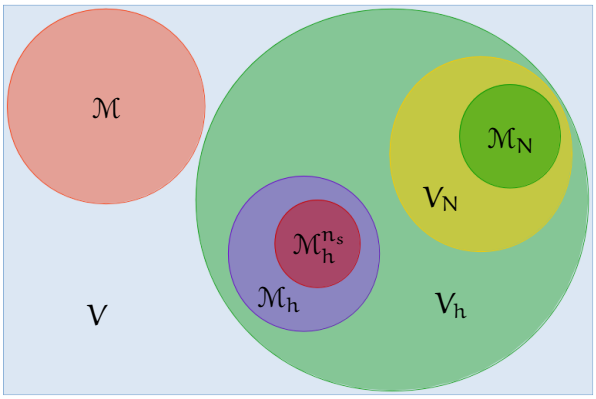


Figure 1.1: Reduced basis set relations

In order to implement $V_N$ in the computer we manipulate the snapshots in their discrete representation, i.e. using their coordinates in the basis of $V_h$, and not as real functions. Therefore, in practical matters, $V_N$ is never explicitly built but actually its *reduced basis matrix*

$$\mathbf{U}_N := [\zeta_1, \dots, \zeta_N] \in \mathbb{R}^{N_h \times N}$$

where for $j = 1, \dots, N$, $\zeta_j = \left(\zeta_j^{(1)}, \dots, \zeta_j^{(N_h)}\right)^T \in \mathbb{R}^{N_h}$ represents the j-th reduced basis function

$$\zeta_j = \sum_{i=1}^{N_h} \zeta_j^{(i)} \varphi^i,$$

with $\{\varphi^1, \dots, \varphi^{N_h}\}$ the $V_h$ basis.

There are two main algorithms to build $\mathbf{U}_N$, *greedy* and *proper orthogonal decomposition* (POD). The main focus here will be on POD, for more information about greedy consult [83, 131]. Before showing the algorithm we present some preliminary definitions and minor results:

- Let $\mathbf{X}_h$ be the matrix built with the scalar product of $V$

$$(\mathbf{X}_h)_{ij} := (\varphi^i, \varphi^j)_V, \tag{1.9}$$

where $\varphi^i$, $i = 1, \dots, N_h$, are the functions of the basis in $V_h$. Then we define the $\mathbf{X}_h$ scalar product and $\mathbf{X}_h$ norm for all $\mathbf{v}, \mathbf{w} \in \mathbb{R}^{N_h}$ by

$$(\mathbf{v}, \mathbf{w})_{\mathbf{X}_h} := \mathbf{v}^T \mathbf{X}_h \mathbf{w},$$

$$\|\mathbf{v}\|_{\mathbf{X}_h} := \sqrt{\mathbf{v}^T \mathbf{X}_h \mathbf{v}}.$$

The $\mathbf{X}_h$ norm satisfies

$$\|\mathbf{u}_h\|_{\mathbf{X}_h} = \|u_h\|_V \tag{1.10}$$

when $\mathbf{u}_h \in \mathbb{R}^{N_h}$ represents the coordinates of $u_h \in V_h$.

- If $a(\cdot, \cdot; \boldsymbol{\mu})$ is symmetric and coercive for all $\boldsymbol{\mu} \in \mathcal{P}$, then

$$(v, w)_{\boldsymbol{\mu}} := a(v, w; \boldsymbol{\mu}), \qquad \forall v, w \in V,$$

and

$$\|v\|_{\boldsymbol{\mu}} := \sqrt{(v, v)_{\boldsymbol{\mu}}}, \qquad \forall v \in V,$$

will denote the inner product and energy norm induced by the bilinear form $a(\cdot, \cdot; \boldsymbol{\mu})$. For bilinear forms independent of $\boldsymbol{\mu}$ the notation will be $(\cdot, \cdot)_a$ and $\| \cdot \|_a$.

- The reduced basis matrices considered afterwards will be orthonormal matrices $\mathbf{U}_N \in \mathcal{V}_N$,

$$\mathcal{V}_N := \{\mathbf{W} \in \mathbb{R}^{N_h \times N} : \mathbf{W}^T \mathbf{W} = \mathbf{I}_N\},$$

or orthonormal with respect to the $\mathbf{X}_h$ scalar product, i.e. $\mathbf{U}_N \in \mathcal{V}_N^{\mathbf{X}_h}$ for

$$\mathcal{V}_N^{\mathbf{X}_h} := \{\mathbf{W} \in \mathbb{R}^{N_h \times N} : \mathbf{W}^T \mathbf{X}_h \mathbf{W} = \mathbf{I}_N\}.$$

Using the standard $l_2$ scalar product

$$(\mathbf{v}, \mathbf{w})_2 := \mathbf{v}^T \mathbf{w}, \qquad \forall \mathbf{v}, \mathbf{w} \in \mathbb{R}^{N_h},$$

then the orthogonal projection of $\mathbf{x} \in \mathbb{R}^{N_h}$ onto the span of $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_N] \in \mathcal{V}_N$ is equal to

$$P_{\mathbf{W}} \mathbf{x} = \sum_{j=1}^{N} (\mathbf{x}, \mathbf{w}_j)_2 \mathbf{w}_j = \mathbf{W}\mathbf{W}^T \mathbf{x}.$$

Whereas if $\mathbf{W} \in \mathcal{V}_N^{\mathbf{X}_h}$, the $\mathbf{X}_h$ orthogonal projection onto its span correspond to

$$P_{\mathbf{W}}^{\mathbf{X}_h} \mathbf{x} = \sum_{j=1}^{N} (\mathbf{x}, \mathbf{w}_j)_{\mathbf{X}_h} \mathbf{w}_j = \sum_{j=1}^{N} \mathbf{w}_j^T \mathbf{X}_h \mathbf{x}\, \mathbf{w}_j = \mathbf{W} \begin{bmatrix} \mathbf{w}_1^T \mathbf{X}_h \mathbf{x} \\ \mathbf{w}_2^T \mathbf{X}_h \mathbf{x} \\ \vdots \\ \mathbf{w}_N^T \mathbf{X}_h \mathbf{x} \end{bmatrix} = \mathbf{W}\mathbf{W}^T \mathbf{X}_h \mathbf{x}. \tag{1.11}$$

In principle, the POD method builds $\mathbf{U}_N$ using a truncated SVD representation of the *solution matrix*

$$\mathbf{S} := \left[\mathbf{u}_1|\cdots|\mathbf{u}_{n_s}\right] = \left[\mathbf{u}_h(\boldsymbol{\mu}^1)|\cdots|\mathbf{u}_h(\boldsymbol{\mu}^{n_s})\right] \in \mathbb{R}^{N_h \times n_s},$$

which has in each column the discrete representation $\mathbf{u}_h(\boldsymbol{\mu}^j) = \{u_h^{(1)}(\boldsymbol{\mu}^j), \ldots, u_h^{(N_h)}(\boldsymbol{\mu}^j)\}^T \in \mathbb{R}^{N_h}$ of each snapshot $u_h(\boldsymbol{\mu}^j) \in V_h$,

$$u_h(\boldsymbol{\mu}^j) = \sum_{i=1}^{N_h} u_h^{(i)}(\boldsymbol{\mu}^j)\varphi^i, \qquad j = 1, \ldots, n_s.$$

To show how precise this method performs we use Schmidt-Eckart-Young theorem. This theorem gives an optimal approximation of a given matrix using sums of rank-1 matrices in the $l_2$ norm $\|\cdot\|_2$ and Frobenius norm $\|\cdot\|_F$.

**Theorem 2** (Schmidt-Eckart-Young)**.** *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be a matrix of rank $r$ with SVD decomposition*

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{Z}^T,$$

*where the left and right singular vectors are respectively*

$$\mathbf{U} = \left[\zeta_1|\cdots|\zeta_m\right] \in \mathbb{R}^{m\times m}, \quad \mathbf{Z} = \left[\boldsymbol{\Psi}_1|\cdots|\boldsymbol{\Psi}_n\right] \in \mathbb{R}^{n\times n},$$

*with singular values*

$$\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m\times n}, \quad \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0, \quad p = \min(m, n).$$

*Then for $1 \leq k \leq r$, the matrix*

$$\mathbf{A}_k := \sum_{i=1}^{k} \sigma_i \zeta_i \boldsymbol{\Psi}_i^T = \mathbf{U}_k\mathbf{U}_k^T\mathbf{A}, \quad \mathbf{U}_k = \left[\zeta_1|\cdots|\zeta_k\right],$$

*satisfies*

$$\|\mathbf{A} - \mathbf{A}_k\|_2 = \min_{\substack{\mathbf{B}\in\mathbb{R}^{m\times n} \\ \operatorname{rank}(\mathbf{B})\leq k}} \|\mathbf{A} - \mathbf{B}\|_2 = \sigma_{k+1},$$

$$\|\mathbf{A} - \mathbf{A}_k\|_F = \min_{\substack{\mathbf{B}\in\mathbb{R}^{m\times n} \\ \operatorname{rank}(\mathbf{B})\leq k}} \|\mathbf{A} - \mathbf{B}\|_F = \sqrt{\sum_{i=k+1}^{r} \sigma_i^2}. \tag{1.12}$$

*Proof.* See [72]. □

The following known result shows that taking the first N left singular vectors of $\mathbf{S}$ results an optimal RB matrix in $\mathcal{V}_N$. The sense of optimality is for $l_2$ representations in the reduced basis of all computed snapshots collectively.

**Proposition 1.** *Let* $\mathbf{S} = \left[\mathbf{u}_1|\cdots|\mathbf{u}_{n_s}\right] \in \mathbb{R}^{N_h \times n_s}$ *a solution matrix of rank* $r$ *with SVD decomposition*

$$\mathbf{S} = \mathbf{U}\Sigma\mathbf{Z}^T,$$

*where the left and right singular vectors are respectively*

$$\mathbf{U} = \left[\zeta_1|\cdots|\zeta_{N_h}\right] \in \mathbb{R}^{N_h \times N_h}, \quad \mathbf{Z} = \left[\Psi_1|\cdots|\Psi_{n_s}\right] \in \mathbb{R}^{n_s \times n_s},$$

*with singular values*

$$\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{N_h \times n_s}, \quad \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0, \quad p = \min(N_h, n_s).$$

*Then for* $N \leq r$*,*

$$\mathbf{U}_N = \left[\zeta_1|\cdots|\zeta_N\right]$$

*gives the following equality*

$$\sum_{i=1}^{n_s} \|\mathbf{u}_i - P_{\mathbf{U}_N}\mathbf{u}_i\|_2^2 = \min_{\mathbf{W}\in\mathcal{V}_N} \sum_{i=1}^{n_s} \|\mathbf{u}_i - P_{\mathbf{W}}\mathbf{u}_i\|_2^2 = \sum_{i=N+1}^{r} \sigma_i^2. \tag{1.13}$$

*Proof.* Proof from [131]. Taking in account that $\|\mathbf{A}\|_F^2 = \sum_i \|\mathbf{a}_i\|_2^2$, where $\mathbf{a}_i$ are the columns of $\mathbf{A}$ results

$$\begin{aligned}\min_{\mathbf{W}\in\mathcal{V}_N} \sum_{i=1}^{n_s} \|\mathbf{u}_i - P_{\mathbf{W}}\mathbf{u}_i\|_2^2 &= \min_{\mathbf{W}\in\mathcal{V}_N} \sum_{i=1}^{n_s} \|\mathbf{u}_i - \mathbf{W}\mathbf{W}^T\mathbf{u}_i\|_2^2 \\ &= \min_{\mathbf{W}\in\mathcal{V}_N} \|\mathbf{S} - \mathbf{W}\mathbf{W}^T\mathbf{S}\|_F.\end{aligned}$$

Since $\operatorname{rank}(\mathbf{W}\mathbf{W}^T\mathbf{S}) = N$, using Theorem 2 we achieve the minimum in the last equality when

$$\mathbf{W}\mathbf{W}^T\mathbf{S} = \mathbf{U}_N\mathbf{U}_N^T\mathbf{S},$$

so taking $\mathbf{W} = \mathbf{U}_N$ and using (1.12) yields equality (1.13). □

In a analogous way we can prove a similar statement for the $\mathbf{X}_h$ norm.

**Proposition 2.** *Let* $\mathbf{S} = \left[\mathbf{u}_1|\cdots|\mathbf{u}_{n_s}\right] \in \mathbb{R}^{N_h \times n_s}$ *a solution matrix of rank* $r$ *and* $\widetilde{\mathbf{S}} = \mathbf{X}_h^{1/2}\mathbf{S}$*. Given the SVD decomposition* $\widetilde{\mathbf{S}} = \widetilde{\mathbf{U}}\widetilde{\Sigma}\widetilde{\mathbf{Z}}^T$ *with left and right singular vectors*

$$\widetilde{\mathbf{U}} = \left[\widetilde{\zeta}_1|\cdots|\widetilde{\zeta}_{N_h}\right] \in \mathbb{R}^{N_h \times N_h}, \quad \widetilde{\mathbf{Z}} = \left[\widetilde{\Psi}_1|\cdots|\widetilde{\Psi}_{n_s}\right] \in \mathbb{R}^{n_s \times n_s},$$

*and singular values*

$$\widetilde{\Sigma} = \operatorname{diag}(\widetilde{\sigma}_1, \ldots, \widetilde{\sigma}_p) \in \mathbb{R}^{N_h \times n_s}, \quad \widetilde{\sigma}_1 \geq \widetilde{\sigma}_2 \geq \ldots \geq \widetilde{\sigma}_p \geq 0, \quad p = \min(N_h, n_s).$$

*the following equality holds for* $N \leq r$ *and* $\mathbf{U}_N = \mathbf{X}_h^{-1/2}\widetilde{\mathbf{U}}_N = \mathbf{X}_h^{-1/2}\left[\widetilde{\zeta}_1|\cdots|\widetilde{\zeta}_N\right]$*,*

$$\sum_{i=1}^{n_s} \|\mathbf{u}_i - P_{\mathbf{U}_N}^{\mathbf{X}_h}\mathbf{u}_i\|_{\mathbf{X}_h}^2 = \min_{\mathbf{W}\in\mathcal{V}_N^{\mathbf{X}_h}} \sum_{i=1}^{n_s} \|\mathbf{u}_i - P_{\mathbf{W}}^{\mathbf{X}_h}\mathbf{u}_i\|_{\mathbf{X}_h}^2 = \sum_{i=N+1}^{r} \widetilde{\sigma}_i^2. \tag{1.14}$$

*Proof.* Proof from [131]. Taking in account that with the symmetric matrix $\mathbf{X}_h$, $\|\mathbf{X}_h^{1/2}\mathbf{x}\|_2 = \|\mathbf{x}\|_{\mathbf{X}_h}$ for all $\mathbf{x} \in \mathbb{R}^{\mathbf{X}_h}$, $\|\mathbf{A}\|_F^2 = \sum_i \|\mathbf{a}_i\|_2^2$, where $\mathbf{a}_i$ are the columns of a matrix $\mathbf{A}$ and defining $\widetilde{\mathbf{W}} = \mathbf{X}_h^{1/2}\mathbf{W}$, then

$$\begin{aligned}\min_{\mathbf{W}\in\mathcal{V}_N^{\mathbf{X}_h}} \sum_{i=1}^{n_s} \|\mathbf{u}_i - P_{\mathbf{U}_N}^{\mathbf{X}_h}\mathbf{u}_i\|_{\mathbf{X}_h}^2 &= \min_{\mathbf{W}\in\mathcal{V}_N^{\mathbf{X}_h}} \sum_{i=1}^{n_s} \|\mathbf{u}_i - \mathbf{W}\mathbf{W}^T\mathbf{X}_h\mathbf{u}_i\|_{\mathbf{X}_h}^2 \\ &= \min_{\mathbf{W}\in\mathcal{V}_N^{\mathbf{X}_h}} \sum_{i=1}^{n_s} \|\mathbf{X}_h^{1/2}\mathbf{u}_i - \mathbf{X}_h^{1/2}\mathbf{W}\mathbf{W}^T\mathbf{X}_h^{1/2}\mathbf{X}_h^{1/2}\mathbf{u}_i\|_2^2 \\ &= \min_{\widetilde{\mathbf{W}}\in\mathcal{V}_N} \|\widetilde{\mathbf{S}} - \widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^T\widetilde{\mathbf{S}}\|_F^2.\end{aligned}$$

Since $\operatorname{rank}(\widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^T\widetilde{\mathbf{S}}) = N$, by Theorem 2 the minimum in the last equality is achieved when

$$\widetilde{\mathbf{W}}\widetilde{\mathbf{W}}^T\widetilde{\mathbf{S}} = \widetilde{\mathbf{U}}_N\widetilde{\mathbf{U}}_N^T\widetilde{\mathbf{S}}$$

so taking $\widetilde{\mathbf{W}} = \widetilde{\mathbf{U}}_N$, defining $\mathbf{U}_N = \mathbf{X}_h^{-1/2}\widetilde{\mathbf{U}}_N$ and using (1.12) yields equality (1.14). □

These results give a way to build the reduced basis matrix $\mathbf{U}_N$ with optimal representation of the solution matrix $\mathbf{S}$ thanks to (1.13) and (1.14), which at the same time are easy to implement and use. Nevertheless, note that they do not expose how the error behaves when $\boldsymbol{\mu} \notin \Xi_s$. In this case, the usual way to compute error estimates is a-posteriori in the online stage.

Moreover, these propositions provide a way to select the dimension of the reduced basis which would be to pick the smallest N such that (1.13) or (1.14) is satisfied for a desirable error $\varepsilon$. Other practical way to select N is to use what is called *relative information content* of the POD basis [4], i.e. given a desirable error $\varepsilon$, pick the smallest N such that

$$I(N) := \frac{\sum_{i=1}^{N} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2} \geq 1 - \varepsilon^2. \tag{1.15}$$

The expression $I(N)$ consider the percentage of information retained in $\mathbf{U}_N$ with respect to $\mathbf{S}$. In terms of Proposition 1 and 2, it means what percentage from the whole sum of singular values is held from the sum of their first N. From here onwards we will call $I(N)$ *relative POD error* to differentiate it from (1.13) and (1.14) that we will denote as *POD error*. For both ways to pick N applies the same idea, the faster the decay of the singular values is, the smaller will be N for attaining a prescribed error.

To summarize, POD method gives two possible procedures, we define them as *POD2* and *PODXh* refering to Proposition 1 and 2 respectively, one which optimize the $l_2$ norm of the error and the other optimize the $\mathbf{X}_h$ norm instead. Their procedure follows as:

1. Construct the solution matrix

$$\mathbf{S} = \left[\mathbf{u}_h(\boldsymbol{\mu}^1)|\cdots|\mathbf{u}_h(\boldsymbol{\mu}^{n_s})\right] \in \mathbb{R}^{N_h\times n_s}. \tag{1.16}$$

and make the transformation $\widetilde{\mathbf{S}} = \mathbf{X}_h^{1/2}\mathbf{S}$ if the $\mathbf{X}_h$ norm is desired.

2. For the $l_2$ norm, compute the SVD decomposition of $\mathbf{S}$ and given an error tolerance $\varepsilon$, select the smallest N less or equal to the rank of $\mathbf{S}$ such that the sum of the singular values satisfies one of these conditions,

$$\sum_{i=N+1}^{r} \sigma_i^2 \leq \varepsilon,$$

or

$$\frac{\sum_{i=1}^{N} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2} \geq 1 - \varepsilon^2.$$

For the $\mathbf{X}_h$ norm applies the same procedure but the SVD is computed to $\widetilde{\mathbf{S}}$.

3. For the $l_2$ norm, construct $\mathbf{U}_N$ taking the first N left singular vectors of $\mathbf{S}$ and for the $\mathbf{X}_h$ norm case $\mathbf{U}_N$ is the result of multiplying $\mathbf{X}_h^{-1/2}$ with the first N left singular vectors of $\widetilde{\mathbf{S}}$.

Once we finish these steps and store the reduced basis matrix we continue with the online stage to compute RB approximations. The following section explains in details the theorical results of this stage.

## 1.3 Reduced Basis Online Stage

There are two main ways to treat the online stage, one is using *Galerkin RB* (G-RB) and the other is *Least-Square RB*. The main focus will be the former because of the existence of optimal results for the problems in subsequent chapters.

The G-RB procedure is analogous to the usual Galerkin approach (1.7), but on the reduced basis space:

Given $\boldsymbol{\mu} \in \mathcal{P}$, find $u_N(\boldsymbol{\mu}) \in V_N$ such that

$$a(u_N(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) = f(v_N; \boldsymbol{\mu}) \qquad \forall v_N \in V_N. \tag{1.17}$$

Equally as before, if $a(\cdot, \cdot; \boldsymbol{\mu})$ is continuous and coercive in $V$ and $f(\cdot; \boldsymbol{\mu})$ is bounded in $V$, all these conditions satisfy in $V_N$, thus (1.17) will have an unique solution in $V_N$ because of Lax-Milgram theorem. To find the RB solution $u_N(\boldsymbol{\mu})$ we solve a linear system like (1.5),

$$\mathbf{A}_N(\boldsymbol{\mu})\mathbf{u}_N(\boldsymbol{\mu}) = \mathbf{f}_N(\boldsymbol{\mu}),$$

where its resulting vector

$$\mathbf{u}_N(\boldsymbol{\mu}) := (u_N^{(1)}(\boldsymbol{\mu}), \ldots, u_N^{(N)}(\boldsymbol{\mu}))^T$$

represents $u_N(\boldsymbol{\mu})$,

$$u_N(\boldsymbol{\mu}) = \sum_{i=1}^{N} u_N^{(i)}(\boldsymbol{\mu})\zeta_i.$$

The following classical result asserts that for symmetric and coercive bilinear forms, G-RB solutions give optimal solutions respect to the energy norm.

**Proposition 3.** *If $a(\cdot,\cdot;\mu)$ is symmetric and coercive for all $\mu \in \mathcal{P}$, then the solution of (1.17) satisfies*

$$u_N(\mu) = \underset{v \in V_N}{\operatorname{argmin}} \|u_h(\mu) - v\|_\mu, \qquad \forall \mu \in \mathcal{P}. \tag{1.18}$$

*Proof.* Using the $\mu$-orthogonality of $u_h(\mu) - u_N(\mu)$ in $V_N$ and the Cauchy-Schwarz inequality

$$\begin{aligned} \|u_h(\mu) - u_N(\mu)\|_\mu^2 &= a(u_h(\mu) - u_N(\mu), u_h(\mu) - u_N(\mu); \mu) \\ &= a(u_h(\mu) - u_N(\mu), u_h(\mu) - v; \mu) \\ &\leq \|u_h(\mu) - u_N(\mu)\|_\mu \|u_h(\mu) - v\|_\mu, \qquad \forall v \in V_N, \end{aligned}$$

therefore, $\|u_h(\mu) - u_N(\mu)\|_\mu \leq \|u_h(\mu) - v\|_\mu$ for all $v$ in $V_N$, proving that $u_N(\mu)$ is the desired minimum. □

In the case of the norm $\|\cdot\|_V$, there is also another classical optimal result. Furthermore, we add an original improvement that connects with POD in Proposition 2.

**Proposition 4.** *If $a(\cdot,\cdot;\mu)$ is symmetric and coercive for all $\mu \in \mathcal{P}$, then the solution of (1.17) satisfies*

$$\|u_h(\mu) - u_N(\mu)\|_V \leq \left(\frac{\gamma_h(\mu)}{\alpha_N(\mu)}\right)^{1/2} \inf_{v \in V_N} \|u_h(\mu) - v\|_V, \qquad \forall \mu \in \mathcal{P}, \tag{1.19}$$

*where $\gamma_h(\mu)$ and $\alpha_N(\mu)$ are the continuity constant in $V_h$ and coercivity constant in $V_N$ of $a(\cdot,\cdot;\mu)$ respectively. Moreover, if $\mu \in \Xi_s$ and $V_N$ is generated by the POD reduced basis matrix $\mathbf{U}_N$ built in Proposition 2, then*

$$\|u_h(\mu) - u_N(\mu)\|_V \leq \left(\frac{\gamma_h(\mu)}{\alpha_N(\mu)}\right)^{1/2} \sqrt{\sum_{i=N+1}^{r} \sigma_i^2}, \qquad \mu \in \Xi_s, \tag{1.20}$$

*where $\sigma_i, i = 1, \ldots, r$, are the singular values different from zero of the solution matrix $\mathbf{S}$ with snapshots $u_h(\mu), \mu \in \Xi_s$.*

*Proof.* As the bilinear form $a(\cdot,\cdot;\mu)$ is considered continuous and coercive in $V$, then there exist $\gamma_h(\mu)$ and $\alpha_N(\mu)$ such that

$$|a(u, v; \mu)| \leq \gamma_h(\mu) \|u\|_V \|v\|_V, \qquad \forall u, v \in V_h,$$

$$a(u, u; \mu) \geq \alpha_N(\mu) \|u\|_V^2, \qquad \forall u \in V_N.$$

Then using Proposition 3,

$$\begin{aligned} \alpha_N(\mu) \|u_h(\mu) - u_N(\mu)\|_V^2 &\leq a(u_h(\mu) - u_N(\mu), u_h(\mu) - u_N(\mu); \mu) \\ &= \|u_h(\mu) - u_N(\mu)\|_\mu^2 \\ &= \min_{v \in V_N} a(u_h(\mu) - v, u_h(\mu) - v; \mu) \\ &\leq \gamma_h(\mu) \inf_{v \in V_N} \|u_h(\mu) - v\|_V^2, \end{aligned}$$

The expression (1.19) results from both extremes of the previous inequality,

$$\alpha_N(\boldsymbol{\mu})\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V^2 \leq \gamma_h(\boldsymbol{\mu}) \inf_{v \in V_N} \|u_h(\boldsymbol{\mu}) - v\|_V^2. \tag{1.21}$$

Now, suppose $V_N$ is generated by the reduced basis matrix $\mathbf{U}_N \in \mathbb{R}^{N_h \times N}$ defined in Proposition 2.

First note that even if the infimum at (1.21) is over the elements in $V_N$, $u_h(\boldsymbol{\mu})$ is in $V_h$. Therefore, to compute $\|u_h(\boldsymbol{\mu}) - v\|_V$ from a discrete point of view using the equivalent norm $\|\cdot\|_{\mathbf{X}_h}$, it is necessary to change the coordinates of $v \in V_N$ to $V_h$, i.e. $\mathbf{U}_N\mathbf{v} \in \mathbb{R}^{N_h}$ with $\mathbf{v} \in \mathbb{R}^N$ the coordinates of $v$ in $V_N$. Then, from the discrete representation $\mathbf{u}_h(\boldsymbol{\mu})$ of $u_h(\boldsymbol{\mu})$ in $V_h$ and (1.10) yields

$$\inf_{v \in V_N} \|u_h(\boldsymbol{\mu}) - v\|_V^2 = \inf_{\mathbf{v} \in \mathbb{R}^N} \|\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{U}_N\mathbf{v}\|_{\mathbf{X}_h}^2, \tag{1.22}$$

for which we obtain the infimum at $\mathbf{v} = \mathbf{U}_N^T\mathbf{X}_h\mathbf{u}_h(\boldsymbol{\mu}) \in \mathbb{R}^N$,

$$\inf_{\mathbf{v} \in \mathbb{R}^N} \|\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{U}_N\mathbf{v}\|_{\mathbf{X}_h}^2 = \|\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{U}_N\mathbf{U}_N^T\mathbf{X}_h\mathbf{u}_h(\boldsymbol{\mu})\|_{\mathbf{X}_h}^2. \tag{1.23}$$

This is because $\mathbf{U}_N\mathbf{U}_N^T\mathbf{X}_h\mathbf{u}_h(\boldsymbol{\mu}) = P_{\mathbf{U}_N}^{\mathbf{X}_h}\mathbf{u}_h(\boldsymbol{\mu})$ which is the projection over the subspace generated from $\mathbf{U}_N$ in the norm $\|\cdot\|_{\mathbf{X}_h}$, see (1.11).

Let $\boldsymbol{\mu} \in \Xi_s$ be an arbitrary parameter and denote $\mathbf{u}_i = \mathbf{u}_h(\boldsymbol{\mu}_i),\ \boldsymbol{\mu}_i \in \Xi_s$ for $i = 1, \ldots, n_s$. In particular, there exists $j$ such that $\mathbf{u}_j = \mathbf{u}_h(\boldsymbol{\mu})$, then using (1.14) results

$$\|\mathbf{u}_j - P_{\mathbf{U}_N}^{\mathbf{X}_h}\mathbf{u}_j\|_{\mathbf{X}_h}^2 \leq \sum_{i=1}^{n_s} \|\mathbf{u}_i - P_{\mathbf{U}_N}^{\mathbf{X}_h}\mathbf{u}_i\|_{\mathbf{X}_h}^2 = \sum_{i=N+1}^{r} \sigma_i^2. \tag{1.24}$$

From (1.21), (1.22), (1.23) and (1.24) we conclude that

$$\alpha_N(\boldsymbol{\mu})\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V^2 \leq \gamma_h(\boldsymbol{\mu}) \sum_{i=N+1}^{r} \sigma_i^2$$

which gives the result (1.20). □

An important case of (1.20) happens when the continuity and coercivity factor are bounded or as the case of the problems presented in Chapter 2 where the bilinear form is independent of $\boldsymbol{\mu}$.

**Corollary 1.** *Under the hypothesis of Proposition 4 suppose $\gamma_h(\boldsymbol{\mu})$ and $\alpha_N(\boldsymbol{\mu})$ are uniformly bounded above and below with bounds $\gamma_0$ and $\alpha_0$ respectively, then estimate (1.20) becomes*

$$\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V \leq \left(\frac{\gamma_0}{\alpha_0}\right)^{1/2} \sqrt{\sum_{i=N+1}^{r} \sigma_i^2}, \qquad \boldsymbol{\mu} \in \Xi_s. \tag{1.25}$$

Inequality (1.25) gives an upper bound of the error for G-RB that improves the faster the decay of the singular values is or the bigger N is. As in Proposition 2, this bound only takes

in account the snapshots computed for building the reduced basis and it does not give any information when $\boldsymbol{\mu} \notin \Xi_s$.

Another interesting fact from (1.20) is that if there exists high linear dependency between the solutions and therefore the rank of S is small, then it is a good idea to take $N = r$. From the projection point of view, (1.14) shows that perfect recovery of the original snapshots from the reduced basis can be achieved, and(1.20) exposes that G-RB procedure gives the possibility of recovering them with exact precision.

As explained before, Proposition 4 does not completely certify a good RB approximation for all $\boldsymbol{\mu} \in \mathcal{P}$. A way to achieve it is to obtain an estimate of the supremum in $\mathcal{P}$ of (1.19),

$$\sup_{\boldsymbol{\mu}\in\mathcal{P}} \|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V \leq \sup_{\boldsymbol{\mu}\in\mathcal{P}} \left[ \left( \frac{\gamma_h(\boldsymbol{\mu})}{\alpha_N(\boldsymbol{\mu})} \right)^{1/2} \inf_{v\in V_N} \|u_h(\boldsymbol{\mu}) - v\|_V \right]. \tag{1.26}$$

In general, knowing if a parametic problem is *reducible*, i.e. if there exists a small finite-dimesional $V_N$ that approximate all the elements of the computable solution manifold $\mathcal{M}_h$ with an acceptable error, is a complicated task. Nevertheless, there are some results that takes in account the smoothness of the solution manifold and parametric complexity which give some conditions for reducibility. Section 1.4 and Section 1.5 go deeper in this subject.

## 1.4 Kolmogorov's n-width Estimates

This section introduces an approach to estimate a-priori error bounds of reduced basis methods. The main idea behind it is to benchmark the *Kolmogorov's n-width* [99],

$$d_n(\mathcal{M})_V := \inf_{\dim(V_n)=n} \sup_{v\in\mathcal{M}} \inf_{w\in V_n} \|v - w\|_V, \tag{1.27}$$

using the smoothness and anisotropy of the solution manifold.

This quantity exposes the best possible accuracy in the norm of $V$ when approximating the elements of $\mathcal{M}$ with linear $n$-dimensional spaces $V_n$. Its principal difference from (1.26) falls in the selection of $V_n$, i.e. Kolmogorov $n$-width ask for the best subspace of dimension $n$ while (1.26) has a fixed subspace built with POD. Even if (1.27) is not exactly (1.26), it gives an idea if it is possible to approximate the elements of $\mathcal{M}$ uniformly from finite dimensional spaces.

The main result of this section is Theorem 6 which gives estimates for the Kolmogorov $n$-width. To arrive to its formulation and proof it is necessary to introduce some background first:

1. New formulation for the dependency of the parameters through a sequence of real numbers.

2. Complex version of Lax-Milgram for the extension of the PDE to the complex domain.

3. Legendre polynomials [55] to generate a suitable $n$-dimensional subspace.

The content of this section is based entirely in [53]. The main difference relies in the proof of Theorem 6, which is the same, but is presented in a straightforward manner. Moreover, this topic will be related to the construction of practical error estimators in Section 1.6.

The first important ingredient of this section is to identify the parametric part of the equation (1.6) through a sequence of real numbers. For example, consider a well defined operator equation

$$\mathcal{A}u = f(\boldsymbol{\mu}),$$

where $f(\boldsymbol{\mu})$ belongs to a subset P of a Banach space X for all $\boldsymbol{\mu} \in \mathcal{P}$. The idea is to define a basis $(\psi_j)_{j\geq 1}$ of functions $\psi_j \in X$ such that

$$f(\boldsymbol{\mu}) = f(y) = \sum_{j\geq 1} y_j \psi_j, \quad y := (y_j)_{j\geq 1}, \quad y_j = y_j(\boldsymbol{\mu}),$$

where $y_j \in \mathbb{R}$, $j \in \mathbb{N}$, and the series converges in the X-norm for each $\boldsymbol{\mu} \in \mathcal{P}$. The sequence $(y_j)_{j\geq 1}$ is called an *affine representer* of $\mathcal{P}$.

The main advantage of affine representations is that $f(\boldsymbol{\mu})$ can now be identified in a different way with the sequence $(y_j)_{j\geq 1}$. Therefore, for each $\boldsymbol{\mu} \in \mathcal{P}$ there is one of this representers. If $\mathcal{P}$ is compact and $y_j(\boldsymbol{\mu})$ continuous with respect to $\boldsymbol{\mu}$, then each function $\psi_j$ can be normalized such that

$$\sup_{\boldsymbol{\mu}\in\mathcal{P}} |y_j(\boldsymbol{\mu})| = 1.$$

After normalization, each sequence $(y_j(\boldsymbol{\mu}))_{j\geq 1}$ will belong to the infinite-dimensional cube

$$Y := [-1, 1]^{\mathbb{N}}.$$

Two remarks are in hand, one is that taking a general sequence $(y_j)_{j\geq 1}$ from $Y$ may not make the sum

$$\sum_{j\geq 1} y_j \psi_j \tag{1.28}$$

necessarily converge in X. The affine representations that for every representer $(y_j)_{j\geq 1} \subset Y$ the sum (1.28) converges are called *complete*. In this case, the sets

$$f(\mathcal{P}) := \{f(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \mathcal{P}\},$$

$$f(Y) := \Big\{ f(y) = \sum_{j\geq 1} y_j \psi_j : (y_j)_{j\geq 1} \in Y \Big\},$$

satisfy the relation $f(\mathcal{P}) \subset f(Y)$.

The second remark is that could happen that for a specific sequence $(y_j)_{j\geq 1}$ there is no solution $u(y)$ and therefore the original equation is not well defined. If for each representer $(y_j)_{j\geq 1} \subset Y$ there exists a solution $u(y)$, then the affine representation is called *compatible*.

An important concept that helps estimate the Kolmogorov $n$-width is *anisotropy*. This regards to the fact that when $\|\psi_j\|_X$ is small the scaled variable $y_j$ in $Y$ have little influence on variations of $u(y)$ and therefore some $y_j$ has more importance than others. This makes the *solution map*

$$y \rightsquigarrow u(y) \tag{1.29}$$

highly anisotropic.

Another relevant concept for obtaining an estimation of Kolmogorov $n$-width is the smoothness of the solution map. More concretely, in many cases it is possible to extend the solutions of a parametric problem to the complex domain in a holomorphic way. Next, we extend the theory of Section 1.1 to the complex valued case.

Let $\mathcal{B}$ be the set of all sesquilinear forms defined in $V \times V$ and $V'$ the set of all antilinear functionals of $V$. The norm of $b \in \mathcal{B}$ is defined as usual,

$$\|b\| := \sup_{\|v\|_V \leq 1, \|w\|_V \leq 1} |b(v, w)|.$$

With these definitions we can consider the following problem:

Given $b \in \mathcal{B}$ and $f \in V'$, find $u \in V$ such that

$$b(u, v) = f(v), \qquad \forall v \in V. \tag{1.30}$$

The existence and uniqueness of (1.30) can be obtained from a complex version of Lax-Milgram theorem, which is presented next.

Let $\mathcal{L}(V, W)$ be the space of all linear operators $T$ from the Banach space $V$ to the Banach space $W$ with the norm

$$\|T\|_{L(V,W)} := \sup_{v \in V} \frac{\|Tv\|_W}{\|v\|_V}.$$

Given $b \in \mathcal{B}$, the expression $b(u, \cdot)$ is an antilinear functional and therefore, for $u \in V$ there exists $Bu \in V'$ such that

$$b(u, v) = (Bu, v)_{V',V}, \qquad v \in V,$$

where $(\cdot, \cdot)_{V',V}$ is the anti-dual pairing between $V$ and $V'$. Hence, $B$ is a linear operator from $V$ into $V'$ with

$$\|B\|_{\mathcal{L}(V,V')} = \|b\|.$$

Consequently, the operator $B$ is continuous given the possibility to express the equation (1.30) in an equivalent operator equation in $V'$,

$$Bu = f. \tag{1.31}$$

**Theorem 3** (Lax-Milgram)**.** *Let $b \in \mathcal{B}$ be a sequilinear form on $V \times V$ such that it is coercive, i.e. there exists $\alpha > 0$ such that*

$$|b(u, u)| \geq \alpha \|u\|_V^2, \qquad \forall u \in V.$$

*Then, $B$ in (1.31) is invertible and its inverse satisfies*

$$\|B^{-1}\|_{\mathcal{L}(V',V)} \leq \frac{1}{\alpha}.$$

*Therefore, for each $f \in V'$, problem (1.30) has a unique solution $u_f = B^{-1}(f)$ which satisfies*

$$\|u_f\|_V \leq \frac{\|f\|_{V'}}{\alpha}.$$

*Proof.* See [132]. □

The procedure of the RB methods explained in previous sections looks for approximated solutions in the form of

$$(x, \mu) \rightsquigarrow u_N(x, \mu) = \sum_{i=1}^{N} u_N^{(i)}(\mu)\zeta_i(x),$$

where $\zeta_i$ are obtained from actual solutions of (1.6). In a more general context, the approximated solutions of a parametric equation can be sought in the *separable form*

$$(x, \mu) \rightsquigarrow u_n(x, \mu) := \sum_{i=1}^{n} v_i(x)\phi_i(\mu),$$

where $v_i \in V$ and $\phi_i : \mathcal{P} \to \mathbb{R}$, or equivalently,

$$(x, y) \rightsquigarrow u_n(x, y) := \sum_{i=1}^{n} v_i(x)\phi_i(y), \tag{1.32}$$

with $\phi_i : Y \to \mathbb{R}$.

The Kolmogorov $n$-width results we show later are obtained using approximations of the form (1.32) when $\phi$ is considered a polynomial, specifically a Legendre polynomial.

Assume the following notation for the sum in (1.32),

$$\sum_{\nu \in \mathcal{F}} u_\nu \phi_\nu, \tag{1.33}$$

where $\mathcal{F}$ is a countable index, $\phi_\nu : Y \to \mathbb{R}$ and $u_\nu \in V$.

**Definition 1.** *A sequence $(\Lambda_n)_{n\geq 1}$ of finite subsets of $\mathcal{F}$ is called an exhaustion of $\mathcal{F}$ if and only if, for any $\nu \in \mathcal{F}$, there exists an $n_0$ such that $\nu \in \Lambda_n$ for all $n \geq n_0$.*

**Definition 2.** *The series (1.33) converges conditionally with limit $u$ if and only if there exists an exhaustion $(\Lambda_n)_{n\geq 1}$ of $\mathcal{F}$ such that*

$$\lim_{n\to\infty} \left\| u - \sum_{\nu \in \Lambda_n} u_\nu \phi_\nu \right\| = 0.$$

*If the sum converges for every exhaustion $(\Lambda_n)_{n\geq 1}$ of $\mathcal{F}$ then is said to converge unconditionally.*

Certainly, having unconditional convergence is more desirable so later we can choose a convenient exhaustion $(\Lambda_n)_{n\geq 1}$, i.e. the one that makes (1.33) converges the fastest to $u$. One useful result is the following classical fact from Hilbert space theory.

**Theorem 4.** *Let $(\phi_\nu)_{\nu\in\mathcal{F}}$ be an orthonormal basis of $L^2(Y, \omega)$ for some given measure $\omega$ on $Y$, and let $u \in L^2(Y, V, \omega)$. Then, the inner products*

$$u_\nu := \int_Y u(y)\phi_\nu(y)d\omega(y), \qquad \nu \in \mathcal{F},$$

*are elements of* $V$*, and the sum (1.33) converges unconditionally to* $u$ *in* $L^2(Y, V, \omega)$ *with error*

$$\left\| u - \sum_{\nu \in \Lambda_n} u_\nu \phi_\nu \right\|_{L^2(Y,V,\omega)} = \left( \sum_{\nu \notin \Lambda_n} \|u_\nu\|_V^2 \right)^{1/2},$$

*for any exhaustion* $(\Lambda_n)_{n \geq 1}$.

*Proof.* See [100]. □

We introduce now the Legendre polynomials, assume $\mathcal{F}$ the set of all sequences $\nu = (\nu_j)_{j \geq 1}$ of non-negative integers which are finitely supported and consider the separable sum

$$\sum_{\nu \in \mathcal{F}} v_\nu L_\nu(y), \quad L_\nu(y) = \prod_{j \geq 1} L_{\nu_j}(y_j), \tag{1.34}$$

where $(L_k)_{k \geq 0}$ is the sequence of Legendre polynomials on $[-1, 1]$

$$Q_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} \left[ (x^2 - 1)^n \right],$$

normalized with respect to the uniform measure,

$$\int_{-1}^{1} |Q_k(t)|^2 \frac{dt}{2} = 1.$$

It is known that Legendre polynomials are orthogonal on $L^2([-1, 1])$ and therefore $(L_k)_{k \geq 0}$ forms an orthonormal basis on $L^2(Y, \omega)$ with

$$\omega = \otimes_{j \geq 1} \frac{dy_j}{2},$$

the uniform measure over $Y$. Using Theorem 4 the coefficients $v_\nu$ are given by

$$v_\nu := \int_Y u(y) L_\nu(y) d\omega(y). \tag{1.35}$$

For completing the proof of upcoming Theorem 6 will be convenient to have Legendre polynomials renormalized in the form of

$$\sum_{\nu \in \mathcal{F}} w_\nu P_\nu(y), \quad P_\nu(y) = \prod_{j \geq 1} P_{\nu_j}(y_j),$$

where $(P_k)_{k \geq 0}$ is the sequence of Legendre polynomials on $[-1, 1]$ with the normalization

$$\|P_k\|_{L^\infty([-1,1])} = P_k(0) = 1.$$

The relation $L_k = \sqrt{1 + 2k} P_k$ gives a way to compute the coefficients $w_\nu$,

$$w_\nu := \int_Y u(y) P_\nu(y) d\omega(y) = \Big( \prod_{j \geq 1} (1 + 2\nu_j) \Big)^{1/2} v_\nu. \tag{1.36}$$

Here are two important results before introducing Theorem 6:

**Lemma 2.** *Let $p > 0$, $(c_\nu)_{\nu\in\mathcal{F}} \in l^p(\mathcal{F})$ a sequence of positive numbers and $(\Lambda_n)_{n\geq 1}$ the set of indices that corresponds to the $n$ largest $c_\nu$. Then, taking a real number $q > p$ yields*

$$\left(\sum_{\nu\notin\Lambda_n} c_\nu^q\right)^{1/q} \leq C(n+1)^{-s},$$

*with constants $C$ and $s$ equal to*

$$C = \|(c_\nu)_{\nu\in\mathcal{F}}\|_{l^p}, \quad s = \frac{1}{p} - \frac{1}{q}.$$

*Proof.* See [53]. □

**Theorem 5.** *Consider a parametric problem of the form (1.6) such that it has a complete compatible affine representer with $(\psi_j)_{j\geq 1} \in l^p(\mathbb{N})$, $p < 1$, and that the solution map $u \rightsquigarrow u(y)$ admits a holomorphic extension to an open set $\mathcal{O}$ such that the parameters $P(y) \subset \mathcal{O}$ with uniform bound*

$$\sup_{z\in\mathcal{O}} \|u(z)\|_V \leq C.$$

*Then, $(\|w_\nu\|_V)_{\nu\in\mathcal{F}} \in l^p(\mathcal{F})$.*

*Proof.* See [53]. □

Now we are ready to show the main result of this section. Some of the new findings of Section 1.5 and Section 1.6 are inspired by the ideas of its proof.

**Theorem 6.** *If the parametric problem (1.6) satisfies the conditions of Theorem 5, then for each $p \in (0, 1)$,*

$$d_n(\mathcal{M})_V \leq C_s(n+1)^{-s}, \quad n \geq 1, \quad s := \frac{1}{p} - 1. \tag{1.37}$$

*Proof.* Proof from [53]. Let $\mathcal{F}$ be the set of all $\nu = (\nu_j)_{j\geq 1}$ of non-negative integers which are finitely supported, and compute $v_\nu$ as in (1.35),

$$v_\nu = \int_Y u(y)L_\nu(y)d\omega(y).$$

From (1.36) results the coefficients of the normalized Legendre polynomials $P_\nu$

$$w_\nu = \Big(\prod_{j\geq 1}(1+2\nu_j)\Big)^{1/2} v_\nu,$$

and consequently the normalized Legendre series in the form of

$$\sum_{\nu\in\mathcal{F}} w_\nu P_\nu. \tag{1.38}$$

The main idea hereafter is to proof that this series converges unconditionally to $u$ and choose a convenient exhaustion to find the required estimate (1.37).

From Theorem 5 we know that

$$(\|w_\nu\|_V)_{\nu\in\mathcal{F}} \in l^p(\mathcal{F}),$$

for $p < 1$. Let $\mathcal{F}^* \subset \mathcal{F}$ denote the subset where $\|w_\nu\|_V \leq 1$, using the fact that the polynomials $P_\nu$ are uniformly bounded by 1 yields

$$\left\| \sum_{\nu\in\mathcal{F}^*} w_\nu P_\nu \right\|_{L^\infty(Y,V)} = \sup_{y\in Y} \left\| \sum_{\nu\in\mathcal{F}^*} w_\nu P_\nu(y) \right\|_V \leq \sum_{\nu\in\mathcal{F}^*} \|w_\nu\|_V \sup_{y\in Y} |P_\nu(y)| \leq \sum_{\nu\in\mathcal{F}^*} \|w_\nu\|_V^p < \infty,$$

and therefore the series (1.38) converges in $L^\infty(Y,V)$. On the other hand, it also converges in $L^2(Y,V,\omega)$ since

$$\left\| \sum_{\nu\in\mathcal{F}} w_\nu P_\nu \right\|_{L^2(Y,V,\omega)} \leq \int_Y \left\| \sum_{\nu\in\mathcal{F}} w_\nu P_\nu(y) \right\|_V d\omega(y) \leq \left\| \sum_{\nu\in\mathcal{F}^*} w_\nu P_\nu \right\|_{L^\infty(Y,V)} \int_Y d\omega(y) = \left\| \sum_{\nu\in\mathcal{F}^*} w_\nu P_\nu \right\|_{L^\infty(Y,V)}.$$

Taking in account that $u_\nu$ is

$$u_\nu = \frac{1}{\|P_\nu\|^2_{L^2(Y,V,\omega)}} \int_Y u(y) P_\nu(y) d\omega(y), \quad \nu \in \mathcal{F},$$

then Theorem 4 guarantees that in fact the Legendre series converges unconditionally to $u$ in $L^2(Y,V,\omega)$, and consequently converges conditionally to $u$ in $L^\infty(Y,V)$ for the trivial exhaustion $(\mathcal{F}_n)_{n\geq 1}$.

Let $\Lambda_n$ be any exhaustion of $\mathcal{F}$ and suppose $\varepsilon > 0$ is arbitrary. Also take $m_0$ such that simultaneously satisfies

$$\left\| u - \sum_{\nu\in\mathcal{F}_m} w_\nu P_\nu \right\|_{L^\infty(Y,V)} \leq \varepsilon/2,$$
$$\sum_{\nu\notin\mathcal{F}_m} \|w_\nu\|_V \leq \varepsilon/2,$$

for $m \geq m_0$. Since $(\Lambda_n)_{n\geq 1}$ is an exhaustion, there exists $n_0$ such that $\mathcal{F}_m \subset \Lambda_n$ and consequently it yields

$$\left\| u - \sum_{\nu\in\Lambda_n} w_\nu P_\nu \right\|_{L^\infty(Y,V)} \leq \left\| u - \sum_{\nu\in\mathcal{F}_m} w_\nu P_\nu \right\|_{L^\infty(Y,V)} + \sum_{\nu\notin\mathcal{F}_m} \|w_\nu\|_V \leq \varepsilon.$$

This proves that the series (1.38) converges unconditionally to $u$ in $L^\infty(Y,V)$. Moreover, an estimation of the error comes from

$$\left\| u - \sum_{\nu\in\Lambda_n} w_\nu P_\nu \right\|_{L^\infty(Y,V)} - \left\| \sum_{\nu\notin\Lambda_n} w_\nu P_\nu \right\|_{L^\infty(Y,V)} \leq \left\| u - \sum_{\nu\in\Lambda_n} w_\nu P_\nu - \sum_{\nu\notin\Lambda_n} w_\nu P_\nu \right\|_{L^\infty(Y,V)}$$
$$\left\| u - \sum_{\nu\in\Lambda_n} w_\nu P_\nu - \sum_{\nu\notin\Lambda_n} w_\nu P_\nu \right\|_{L^\infty(Y,V)} = \left\| u - \sum_{\nu\in\mathcal{F}} w_\nu P_n u \right\|_{L^\infty(Y,V)} = 0,$$

to conclude that

$$\left\| u - \sum_{\nu \in \Lambda_n} w_\nu P_\nu \right\|_{L^\infty(Y,V)} \leq \left\| \sum_{\nu \notin \Lambda_n} w_\nu P_\nu \right\|_{L^\infty(Y,V)}. \tag{1.39}$$

Consider now $\Lambda_n$ the set of indices of the $n$ largest $\|w_\nu\|_V$ and define the truncated Legendre expansion

$$u_n := \sum_{\nu \in \Lambda_n} w_\nu P_\nu.$$

Using $q = 1$ in Lemma 2 the error $\|u - u_n\|_V$ reduce to

$$\|u - u_n\|_V \leq \sum_{\nu \notin \Lambda_n} w_\nu P_\nu \leq C(n+1)^{\frac{1}{p}-1},$$

with $C = \|(\|w_\nu\|_V)_{\nu \in \mathcal{F}}\|$.

Taking the definition of Kolmogorov $n$-width with the subspace

$$V_n := \text{span}\{w_\nu : \nu \in \Lambda_n\} \subset V,$$

gives the desired result through the inequalities

$$d_n(\mathcal{M})_V \leq \sup_{v \in \mathcal{M}} \inf_{w \in \mathbf{V}_n} \|v - w\|_V \leq \sup_{y \in Y} \inf_{w \in \mathbf{V}_n} \|u(y) - w\|_V \leq \|u - u_n\|_{L^\infty(Y,V)} \leq C(n+1)^{\frac{1}{p}-1}.$$

□

There exists an ongoing research regarding Kolmogorov $n$-width bounds and exponential decay results (see [53, 125, 131] and their references) but many of them are sub-optimal in comparison to the $n$-width spaces.

Unfortunaly, is not always possible to obtain good approximations from a linear space. The following new result shows a particular case of problems that cannot be approximated using RB methods. In particular, it will be useful in Chapter 2.

**Theorem 7.** *Let the bilinear form $a(\cdot,\cdot)$ be symmetric, coercive and $\mu$-independent. Also suppose*

$$V_N = \textit{span}\{\zeta_1, \ldots, \zeta_N\} = \textit{span}\{\zeta_1^a, \ldots, \zeta_N^a\},$$

*where the elements $\zeta_i^a, i = 1, \ldots, N$, are obtained after an orthonormalization process of the original reduced basis functions with respect to the scalar product $(\cdot,\cdot)_a$. If there exist a sequence $(\mu_k)_{k \geq 1} \subset \mathcal{P}$ and an index $i \in [N+1, N_h]$ such that an element of the orthogonal complement of $V_N$ in $V_h$ with respect to $(\cdot,\cdot)_a$,*

$$V_N^a = \textit{span}\{\zeta_{N+1}^a, \ldots, \zeta_{N_h}^a\},$$

*satisfies $|f(\zeta_i^a; \mu_k)| \to \infty$ when $k \to \infty$, then*

$$\|u_h(\mu_k) - u_N(\mu_k)\|_V \xrightarrow[k \to \infty]{} \infty.$$

*Proof.* We can represent the solution $u_h(\boldsymbol{\mu}_k)$ of problem (1.7) for each $k \in \mathbb{N}$ as

$$u_h(\boldsymbol{\mu}_k) = \sum_{j=1}^{N_h} u_h^{(j)}(\boldsymbol{\mu}^k)\zeta_j^a,$$

where $u_h^{(j)}(\boldsymbol{\mu}^k) = (u_h(\boldsymbol{\mu}_k), \zeta_j^a)_a$ for all $j = 1, \ldots, N_h$. Therefore, by the orthonormality of the system $\{\zeta_1^a, \ldots, \zeta_{N_h}^a\}$ we obtain

$$|f(\zeta_i^a; \boldsymbol{\mu}_k)| = |a(u_h^{(i)}(\boldsymbol{\mu}^k)\zeta_i^a, \zeta_i^a)| = |u_h^{(i)}(\boldsymbol{\mu}^k)| \xrightarrow[k\to\infty]{} \infty.$$

Using Proposition 3 for the Galerkin solution on $V_N$ the following holds,

$$\|u_h(\boldsymbol{\mu}_k) - u_N(\boldsymbol{\mu}_k)\|_a^2 = \left\| \sum_{j=1}^{N_h} u_h^{(j)}(\boldsymbol{\mu}^k)\zeta_j^a - \sum_{j=1}^{N} u_h^{(j)}(\boldsymbol{\mu}^k)\zeta_j^a \right\|_a^2 = \sum_{j=N+1}^{N_h} |u_h^{(j)}(\boldsymbol{\mu}^k)|^2 \xrightarrow[k\to\infty]{} \infty,$$

because $N + 1 \leq i \leq N_h$ and $|u_h^{(i)}(\boldsymbol{\mu}^k)| \to \infty$ when $k \to \infty$. The desired result comes from the fact that all norms are equivalent in finite dimensional spaces. □

Theorem 6 exploits the smoothness of the solution map to obtain estimates for the Kolmogorov's n-width. Next section shows another paradigm for such estimations using the parametric complexity instead.

## 1.5 Fundamental Order Reduction Method

This section presents a new viewpoint of a technique usually applied for estimating Kolmogorov's n-width from the parametric complexity of PDEs. The main result of this technique in finite dimensional spaces is located in [131]. Here we generalize the procedure to the infinite dimensional case and add new error estimations.

With these new results we go beyond estimations and actually solve parameterized PDEs. We call this approach *Fundamental Order Reduction method* (FOR).

FOR method has three main differences with RB methods, one is that we cannot apply it to every kind of PDE. The problems that solves are linear operator equations of the kind

$$A(\boldsymbol{\mu})u = \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu}) f_q, \tag{1.40}$$

where

$$A(\boldsymbol{\mu}) := \Theta_a^1(\boldsymbol{\mu})A_1 + \Theta_a^2(\boldsymbol{\mu})A_2, \qquad f_q \in V', \quad q = 1, \ldots, Q_f,$$

and

$$A_i : V \to V', \qquad i = 1, 2,$$
$$\Theta_a^i, \Theta_f^q : \mathcal{P} \to \mathbb{R}, \qquad i = 1, 2, \qquad q = 1, \ldots, Q_f.$$

The following result express the exact solution of (1.40) and forms the base of the fundamental order reduction method. It also exposes another main difference with reduced basis methods which is guaranteed convergence to the exact solution. The finite dimensional equivalent is found in [131].

**Theorem 8.** *If the corresponding bilinear forms of* $A(\mu)$ *and* $\Theta_a^1(\mu)A_1$ *are continuous and coercive for all* $\mu \in \mathcal{P}$ *and the spectral radius*

$$\rho\Big(\frac{\Theta_a^2(\mu)}{\Theta_a^1(\mu)}A_1^{-1}A_2\Big) < 1, \qquad \forall \mu \in \mathcal{P}, \tag{1.41}$$

*then the unique solution* $u(\mu)$ *of (1.40) can be expressed as*

$$u(\mu) = \sum_{k=0}^{\infty}\sum_{q=1}^{Q_f} \frac{(-1)^k(\Theta_a^2(\mu))^k\Theta_f^q(\mu)}{(\Theta_a^1(\mu))^{k+1}}(A_1^{-1}A_2)^k A_1^{-1}f_q. \tag{1.42}$$

*Proof.* Let $\mu \in \mathcal{P}$ be an arbitrary parameter. As the bilinear form of $A(\mu)$ is continuous and coercive then there is a unique solution $u(\mu)$ of (1.40) by the Lax-Milgram theorem.

We now proceed to proof that

$$A_1^{-1}A_2 : V \to V$$

is bounded and later obtain the result (1.42). To simplify the notation the parameter dependencies will be omitted.

Take $u_n \to u$ in $V$. From the continuity of the operator

$$A : V \to V',$$

we obtain that

$$f = Au = \lim_n Au_n = \lim_n f_n,$$

where $f$ and $f_n$ are the corresponding values of $Au$ and $Au_n$.

By Lax-Milgram $\Theta_a^1A_1$ is invertible with continuous inverse, and

$$\Theta_a^1A_1u + \Theta_a^2A_2u = f$$

implies

$$u + \frac{\Theta_a^2}{\Theta_a^1}A_1^{-1}A_2u = \frac{1}{\Theta_a^1}A_1^{-1}f. \tag{1.43}$$

Using (1.43) and taking the limit

$$\lim_n \frac{\Theta_a^2}{\Theta_a^1}A_1^{-1}A_2u_n = \lim_n \frac{1}{\Theta_a^1}A_1^{-1}f_n - u_n = \frac{1}{\Theta_a^1}A_1^{-1}f - u = \frac{\Theta_a^2}{\Theta_a^1}A_1^{-1}A_2u,$$

we prove the boundedness of $A_1^{-1}A_2$.

The inverse of the operator $I + \frac{\Theta_a^2}{\Theta_a^1} A_1^{-1} A_2$ in (1.43) is well known as

$$\rho\Big(\frac{\Theta_a^2}{\Theta_a^1} A_1^{-1} A_2\Big) < 1$$

and $A_1^{-1} A_2$ is bounded,

$$u = \sum_{k=0}^{\infty} \Big(\frac{-\Theta_a^2 A_1^{-1} A_2}{\Theta_a^1}\Big)^k \frac{1}{\Theta_a^1} A_1^{-1} f.$$

The expression (1.42) follows from substituting $f = \sum_{q=1}^{Q_f} \Theta_f^q f_q$. □

To apply Theorem 8 and obtain an approximation $u_N(\boldsymbol{\mu})$ of $u(\boldsymbol{\mu})$ we divide the computations in two stages like in RB, the offline and the online. In the offline stage we compute the *fundamental basis functions*

$$\Psi_{k,q} := (A_1^{-1} A_2)^k A_1^{-1} f_q, \qquad k = 0, \ldots, N,\ q = 1, \ldots, Q_f,$$

and define the *fundamental basis space*

$$V_{N,Q_f} := \operatorname{span}\{\Psi_{k,q}\}_{k=0,\ldots,N,\ q=1,\ldots,Q_f}, \tag{1.44}$$

of dimension $NQ_f$. Instead, in the online stage we evaluate all the parameter dependent functions $\Theta_a^1, \Theta_a^2, \Theta_f^q$, $q = 1, \ldots, Q_f$, for the required parameter $\boldsymbol{\mu} \in \mathcal{P}$ and compute an approximated solution of (1.40) using the first N summands of (1.42).

The third main difference between FOR and RB lies in the procedure of both stages. In the offline stage the RB methods need a well selected sample of the parameter set to construct the reduced basis space while FOR method compute the basis independently of the parameter set, i.e. is a samplingless method. Regarding the online stage, FOR evaluates the parameter part and compute a partial sum of (1.42), whereas RB requires to solve a small linear system and therefore is more expensive in online computations.

We follow now with two quick remarks. The first one is that Theorem 8 can be extended readily to an arbitrary affine representation

$$A(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) A_q.$$

But when $Q_a > 2$, in the $n$-th summand of (1.42) will appear $Q_f(Q_a - 1)^n$ new terms to sum (see [131]), and therefore the fundamental basis space will have dimension

$$Q_f \sum_{i=1}^{N} (Q_a - 1)^i, \tag{1.45}$$

therefore not feasible for fast computations. On the other hand, if $Q_a = 2$ as in Theorem 8, then (1.45) reduce to $NQ_f$, which does not scale exponentially.

The second remark is when we have $Q_a = 1$. This is a particular case of Theorem 8 which gives the exact solution for all $\boldsymbol{\mu} \in \mathcal{P}$. We will write this case as a corollary to have quick references later from Chapter 2.

**Corollary 2.** *Suppose the bilinear form of $A_1$ is continuous and coercive. If $\Theta_a^1(\boldsymbol{\mu}) = 1$ and $\Theta_a^2(\boldsymbol{\mu}) = 0$ for all $\boldsymbol{\mu} \in \mathcal{P}$. Then*

$$u(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu}) A_1^{-1} f_q. \tag{1.46}$$

We present some error estimates with Proposition 5 and Proposition 6.

**Proposition 5.** *Suppose the bilinear form of $A_1$ is continuous and coercive with coercivity constant $\alpha_1$, and*

$$\left\| \frac{\Theta_a^2(\boldsymbol{\mu})}{\Theta_a^1(\boldsymbol{\mu})} A_1^{-1} A_2 \right\| = r(\boldsymbol{\mu}) < 1, \qquad \forall \boldsymbol{\mu} \in \mathcal{P}. \tag{1.47}$$

*Then the partial sum*

$$u_N(\boldsymbol{\mu}) = \sum_{k=0}^{N-1} \sum_{q=1}^{Q_f} \frac{(-1)^k (\Theta_a^2(\boldsymbol{\mu}))^k \Theta_f^q(\boldsymbol{\mu})}{(\Theta_a^1(\boldsymbol{\mu}))^{k+1}} (A_1^{-1} A_2)^k A_1^{-1} f_q \tag{1.48}$$

*satisfies*

$$\|u(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V \leq \frac{r^N(\boldsymbol{\mu})}{\alpha_1 |\Theta_a^1(\boldsymbol{\mu})| (1 - r(\boldsymbol{\mu}))} \sum_{q=1}^{Q_f} |\Theta_f^q(\boldsymbol{\mu})| \|f_q\|_{V'} \tag{1.49}$$

*Proof.* With the hypothesis of this proposition we can easily follow the proof in Theorem 8 and obtain a solution $u(\boldsymbol{\mu})$ as (1.42), the only difference here is the boundedness of $A_1^{-1} A_2$ which is given by (1.47).

Subtracting (1.42) and (1.48) yields

$$\begin{aligned}
\|u - u_N\|_V &= \left\| \sum_{k=N}^{\infty} \sum_{q=1}^{Q_f} \frac{(-1)^k (\Theta_a^2)^k \Theta_f^q}{(\Theta_a^1)^{k+1}} (A_1^{-1} A_2)^k A_1^{-1} f_q \right\|_V \\
&\leq \frac{1}{|\Theta_a^1|} \sum_{q=1}^{Q_f} |\Theta_f^q| \sum_{k=N}^{\infty} \left\| \left( \frac{\Theta_a^2}{\Theta_a^1} A_1^{-1} A_2 \right)^k A_1^{-1} f_q \right\|_V \\
&\leq \frac{1}{\alpha_1 |\Theta_a^1|} \sum_{q=1}^{Q_f} |\Theta_f^q| \|f_q\|_{V'} \sum_{k=N}^{\infty} \left\| \frac{\Theta_a^2}{\Theta_a^1} A_1^{-1} A_2 \right\|_V^k \\
&= \frac{r^N}{\alpha_1 |\Theta_a^1| (1 - r)} \sum_{q=1}^{Q_f} |\Theta_f^q| \|f_q\|_{V'}.
\end{aligned}$$

□

**Proposition 6.** *Suppose the hypothesis of Theorem 8 are satisfied with spectral radius*

$$\rho\left( \frac{\Theta_a^2(\boldsymbol{\mu})}{\Theta_a^1(\boldsymbol{\mu})} A_1^{-1} A_2 \right) = r(\boldsymbol{\mu}) < 1, \qquad \forall \boldsymbol{\mu} \in \mathcal{P}.$$

*If the eigenvectors $\{w_i\}_{i=1}^{\infty} \subset V$ of $A_1^{-1}A_2$ form an orthonormal basis in $V$, then the partial sum*

$$u_N(\boldsymbol{\mu}) = \sum_{k=0}^{N-1} \sum_{q=1}^{Q_f} \frac{(-1)^k(\Theta_a^2(\boldsymbol{\mu}))^k \Theta_f^q(\boldsymbol{\mu})}{(\Theta_a^1(\boldsymbol{\mu}))^{k+1}} (A_1^{-1}A_2)^k A_1^{-1} f_q$$

*satisfies*

$$\|u(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V \leq \frac{r^N(\boldsymbol{\mu})}{\alpha_1 |\Theta_a^1(\boldsymbol{\mu})| \sqrt{1 - r^2(\boldsymbol{\mu})}} \left( \sum_{q=1}^{Q_f} \left(\Theta_f^q(\boldsymbol{\mu})\right)^2 \|f_q\|_{V'}^2 \right)^{1/2}. \tag{1.50}$$

*Proof.* As the eigenvectors $\{w_i\}_{i=1}^{\infty}$ forms an orthonormal basis in $V$ there exist unique scalars $\beta_i^q$ such that

$$A_1^{-1} f_q = \sum_{i=1}^{\infty} \beta_i^q w_i, \qquad q = 1, \ldots, Q_f.$$

This equality allow us to express the operator $A_1^{-1}A_2$ in terms of its eigenvalues $\lambda_i$,

$$\left(\frac{\Theta_a^2}{\Theta_a^1} A_1^{-1}A_2\right)^k A_1^{-1} f_q = \sum_{i=1}^{\infty} \beta_i^q \left(\frac{\Theta_a^2}{\Theta_a^1}\lambda_i\right)^k w_i, \qquad q = 1, \ldots, Q_f,$$

and consequently

$$\begin{aligned}
\|u - u_N\|_V^2 &= \left\| \sum_{k=N}^{\infty} \sum_{q=1}^{Q_f} \frac{(-1)^k (\Theta_a^2)^k \Theta_f^q}{(\Theta_a^1)^{k+1}} (A_1^{-1}A_2)^k A_1^{-1} f_q \right\|_V^2 \\
&= \left\| \sum_{q=1}^{Q_f} \frac{\Theta_f^q}{\Theta_a^1} \sum_{k=N}^{\infty} (-1)^k \sum_{i=1}^{\infty} \beta_i^q \left(\frac{\Theta_a^2}{\Theta_a^1}\lambda_i\right)^k w_i \right\|_V^2 \\
&\leq \sum_{q=1}^{Q_f} \left(\frac{\Theta_f^q}{\Theta_a^1}\right)^2 \sum_{i=1}^{\infty} (\beta_i^q)^2 \sum_{k=N}^{\infty} \left(\frac{\Theta_a^2}{\Theta_a^1}\lambda_i\right)^{2k} \\
&\leq \left( \sum_{q=1}^{Q_f} \left(\frac{\Theta_f^q}{\Theta_a^1}\right)^2 \sum_{i=1}^{\infty} (\beta_i^q)^2 \right) \left( \sum_{k=N}^{\infty} r^{2k} \right) \\
&\leq \frac{r^{2N}}{\alpha_1^2 (\Theta_a^1)^2 (1 - r^2)} \sum_{q=1}^{Q_f} (\Theta_f^q)^2 \|f_q\|_{V'}^2.
\end{aligned}$$

The result (1.50) follows immediately. □

When the scalar coefficients in (1.42) are all positive, it is possible to modify the summands to make it converge faster in some cases. This modification is called Accelerated Fundamental Order Reduction (AFOR) and is based in the following result.

**Lemma 3.** *Suppose that* $0 < a < 1$ *and define*

$$d_N := \frac{(3+\sqrt{8})^N + (3-\sqrt{8})^N}{2}, \quad c_{N,k} := (-1)^k \sum_{m=k+1}^{N} \frac{N}{N+m}\binom{N+m}{2m} 2^{2m},$$

*then*

$$\left| \sum_{k=0}^{\infty} (-1)^k a^k - \sum_{k=0}^{N-1} \frac{c_{N,k}}{d_N} a^k \right| \leq \frac{2}{(1+a)(3+\sqrt{8})^N} \tag{1.51}$$

*Proof.* See [54]. □

The modification $c_{N,k}/d_N$ in the sum

$$\sum_{k=0}^{N-1} \frac{c_{N,k}}{d_N} a^k$$

is known as a Padé type approximation [41, 66]. For implementation and extensions consult [54]. Now we apply Proposition 3 to improve the convergence in the case (1.50) converges slowly.

**Proposition 7.** *Suppose the hypothesis of Theorem 8 are satisfied and define*

$$d_N := \frac{(3+\sqrt{8})^N + (3-\sqrt{8})^N}{2}, \quad c_{N,k} := (-1)^k \sum_{m=k+1}^{N} \frac{N}{N+m}\binom{N+m}{2m} 2^{2m}.$$

*If the eigenvectors* $\{w_i\}_{i=1}^{\infty} \subset V$ *of* $A_1^{-1}A_2$ *form an orthonormal basis in* $V$ *with all eigenvalues satisfying*

$$\frac{\Theta_a^2(\mu)}{\Theta_a^1(\mu)}\lambda > 0, \qquad \mu \in \mathcal{P}.$$

*Then the partial sum*

$$u_N(\mu) = \sum_{k=0}^{N-1} \sum_{q=1}^{Q_f} \frac{c_{N,k}(\Theta_a^2(\mu))^k \Theta_f^q(\mu)}{d_N(\Theta_a^1(\mu))^{k+1}} (A_1^{-1}A_2)^k A_1^{-1} f_q$$

*satisfies*

$$\|u(\mu) - u_N(\mu)\|_V \leq \frac{2}{\alpha_1 |\Theta_a^1(\mu)| \left(1 + \frac{\Theta_a^2(\mu)}{\Theta_a^1(\mu)}\lambda_{\min}\right)(3+\sqrt{8})^N} \left( \sum_{q=1}^{Q_f} \left(\Theta_f^q(\mu)\right)^2 \|f_q\|_{V'}^2 \right)^{1/2}. \tag{1.52}$$

*where the* $\lambda_{\min}$ *is the smallest eigenvalue of* $A_1^{-1}A_2$ *in absolute value.*

*Proof.* We proceed as in Proposition 6, express

$$A_1^{-1} f_q = \sum_{i=1}^{\infty} \beta_i^q w_i, \qquad q = 1, \ldots, Q_f.$$

and

$$\Big(\frac{\Theta_a^2}{\Theta_a^1}A_1^{-1}A_2\Big)^k A_1^{-1}f_q = \sum_{i=1}^{\infty}\beta_i^q\Big(\frac{\Theta_a^2}{\Theta_a^1}\lambda_i\Big)^k w_i, \qquad q = 1,\dots,Q_f.$$

Then using Proposition 3 yields

$$\begin{aligned}
\|u-u_N\|_V^2 &= \Big\| \sum_{k=0}^{\infty}\sum_{q=1}^{Q_f} \frac{(-1)^k(\Theta_a^2)^k\Theta_f^q}{(\Theta_a^1)^{k+1}}(A_1^{-1}A_2)^kA_1^{-1}f_q - \sum_{k=0}^{N}\sum_{q=1}^{Q_f}\frac{c_{N,k}(\Theta_a^2)^k\Theta_f^q}{d_N(\Theta_a^1)^{k+1}}(A_1^{-1}A_2)^kA_1^{-1}f_q\Big\|_V^2\\
&= \Big\| \sum_{q=1}^{Q_f}\frac{\Theta_f^q}{\Theta_a^1}\Big(\sum_{k=0}^{\infty}\sum_{i=1}^{\infty}(-1)^k\beta_i^q\Big(\frac{\Theta_a^2}{\Theta_a^1}\lambda_i\Big)^k w_i - \sum_{k=0}^{N}\sum_{i=1}^{\infty}\beta_i^q\frac{c_{N,k}}{d_N}\Big(\frac{\Theta_a^2}{\Theta_a^1}\lambda_i\Big)^k w_i\Big)\Big\|_V^2\\
&\le \sum_{q=1}^{Q_f}\Big(\frac{\Theta_f^q}{\Theta_a^1}\Big)^2\sum_{i=1}^{\infty}(\beta_i^q)^2\Big|\sum_{k=0}^{\infty}(-1)^k\Big(\frac{\Theta_a^2}{\Theta_a^1}\lambda_i\Big)^k - \sum_{k=0}^{N}\frac{c_N}{d_{N,k}}\Big(\frac{\Theta_a^2}{\Theta_a^1}\lambda_i\Big)^k\Big|^2\\
&\le \frac{2^2}{(\Theta_a^1)^2(3+\sqrt{8})^{2N}}\sum_{q=1}^{Q_f}(\Theta_f^q)^2\sum_{i=1}^{\infty}(\beta_i^q)^2\big(1+\frac{\Theta_a^2}{\Theta_a^1}\lambda_i\big)^{-2},\\
&\le \frac{2^2}{\alpha_1^2(\Theta_a^1)^2\big(1+\frac{\Theta_a^2}{\Theta_a^1}\lambda_{\min}\big)^2(3+\sqrt{8})^{2N}}\sum_{q=1}^{Q_f}(\Theta_f^q)^2\|f_q\|_{V'}^2,
\end{aligned}$$

and finally

$$\|u-u_N\|_V \le \frac{2}{\alpha_1|\Theta_a^1|\big(1+\frac{\Theta_a^2}{\Theta_a^1}\lambda_{\min}\big)(3+\sqrt{8})^N}\Big(\sum_{q=1}^{Q_f}(\Theta_f^q)^2\|f_q\|_{V'}^2\Big)^{1/2}.$$

□

Even if we call the method in Proposition 7 "accelerated", the estimate (1.52) will converge slower than (1.50) if

$$\frac{1}{3+\sqrt{8}} \ge r(\boldsymbol{\mu}), \qquad \boldsymbol{\mu}\in\mathcal{P},$$

where $r$ is the spectral radius

$$\rho\Big(\frac{\Theta_a^2(\boldsymbol{\mu})}{\Theta_a^1(\boldsymbol{\mu})}A_1^{-1}A_2\Big) = r(\boldsymbol{\mu}) < 1, \qquad \boldsymbol{\mu}\in\mathcal{P}.$$

With the results shown so far we can obtain estimates for the Kolmogorov's $n$-width.

**Theorem 9.** *Suppose that for all* $\boldsymbol{\mu}\in\mathcal{P}$, $\sum_{q=1}^{Q_f}\frac{\Theta_f^q(\boldsymbol{\mu})}{\Theta_a^1(\boldsymbol{\mu})}$ *is bounded.*

*i) If the hypothesis of Proposition 5 or Proposition 6 are satisfied and for all* $\boldsymbol{\mu}\in\mathcal{P}$, $r(\boldsymbol{\mu})\le r<1$, *then*

$$d_n(\mathcal{M}) \le Cr^n,$$

*where* $C$ *is independent of* $\boldsymbol{\mu}$.

*ii) If the hypothesis of Proposition 7 are satisfied, then*

$$d_n(\mathcal{M}) \leq \frac{C}{(3+\sqrt{8})^n},$$

*where* $C$ *is independent of* $\mu$.

*Proof.* We prove the result for the hypothesis of Proposition 5, the other cases are analogous. Using the space (1.44) for $n = NQ_f$ and the estimate (1.49) yields

$$\begin{aligned}
d_n(\mathcal{M})_V &= \inf_{\dim(V_n)=n} \sup_{v\in\mathcal{M}} \inf_{w\in V_n} \|v-w\|_V \\
&\leq \sup_{v\in\mathcal{M}} \inf_{w\in V_{N,Q_f}} \|v-w\|_V \\
&= \sup_{\mu\in\mathcal{P}} \inf_{w\in V_{N,Q_f}} \|u(\mu)-w\|_V \\
&\leq \sup_{\mu\in\mathcal{P}} \|u(\mu)-u_N(\mu)\|_V \\
&\leq \sup_{\mu\in\mathcal{P}} \left\{ \frac{1}{\alpha_1 |\Theta_a^1(\mu)|(1-r(\mu))r^{Q_f}} \sum_{q=1}^{Q_f} |\Theta_f^q(\mu)| \|f_q\|_{V'} \right\} r^n \\
&\leq Cr^n.
\end{aligned}$$

□

The following results are an extension of the ideas presented so far to the case where there is not a finite affine representation of the functional in (1.40).

**Theorem 10.** *Suppose that* $W$ *is a Hilbert subspace of* $V'$ *with* $\|f\|_{V'} \leq C\|f\|_W$ *for all* $f \in W$ *and there exists an orthonormal basis in* $W, \{f_q\}_{q=1}^{\infty} \subset W$, *with*

$$f(\mu) = \sum_{q=1}^{\infty} \Theta_f^q(\mu) f_q, \qquad \forall \mu \in \mathcal{P}.$$

*If the conditions of Theorem 8 are met, then the solution to the equation*

$$\Theta_a^1(\mu) A_1 u + \Theta_a^2(\mu) A_2 u = f(\mu), \qquad \forall \mu \in \mathcal{P}, \tag{1.53}$$

*can be expressed as*

$$u(\mu) = \sum_{k=0}^{\infty} \sum_{q=1}^{\infty} \frac{(-1)^k (\Theta_a^2(\mu))^k \Theta_f^q(\mu)}{(\Theta_a^1(\mu))^{k+1}} (A_1^{-1}A_2)^k A_1^{-1} f_q, \qquad \mu \in \mathcal{P}. \tag{1.54}$$

*Additionally, denote*

$$f_{Q_f}(\mu) := \sum_{q=1}^{Q_f} \Theta_f^q(\mu) f_q, \qquad \mu \in \mathcal{P},$$

$$\varepsilon_{Q_f}(\mu) := \left\| \sum_{q=Q_f+1}^{\infty} \Theta_f^q(\mu) f_q \right\|_W, \qquad \mu \in \mathcal{P},$$

$$u_{N,Q_f}(\mu) := \sum_{k=0}^{N} \sum_{q=1}^{Q_f} \frac{(-1)^k (\Theta_a^2(\mu))^k \Theta_f^q(\mu)}{(\Theta_a^1(\mu))^{k+1}} (A_1^{-1}A_2)^k A_1^{-1} f_q, \qquad \mu \in \mathcal{P},$$

*i) If the hypothesis of Proposition 5 are satisfied, then*

$$\|u(\mu) - u_{N,Qf}(\mu)\|_V \leq \frac{1}{\alpha_1 |\Theta_a^1(\mu)|(1 - r(\mu))} \Big( C\varepsilon_{Q_f}(\mu) + \|f_{Q_f}(\mu)\|_{V'} r^N(\mu) \Big), \qquad \mu \in \mathcal{P}.$$

*ii) If the hypothesis of Proposition 6 are satisfied, then*

$$\|u(\mu) - u_{N,Qf}(\mu)\|_V \leq \frac{1}{\alpha_1 |\Theta_a^1(\mu)| \sqrt{1 - r^2(\mu)}} \Big( C^2 \varepsilon_{Q_f}^2(\mu) + \|f_{Q_f}(\mu)\|_{V'}^2 r^{2N}(\mu) \Big)^{1/2}, \quad \mu \in \mathcal{P}.$$

*iii) If the hypothesis of Proposition 7 are satisfied, then AFOR gives the estimate*

$$\|u(\mu) - u_{N,Qf}(\mu)\|_V \leq \frac{1}{\alpha_1 |\Theta_a^1(\mu)|} \left( \frac{C^2}{1 - r^2(\mu)} \varepsilon_{Q_f}^2(\mu) + \frac{4 \|f_{Q_f}(\mu)\|_{V'}^2}{\left(1 + \frac{\Theta_a^2(\mu)}{\Theta_a^1(\mu)} \lambda_{\min}\right)^2} \frac{1}{(3 + \sqrt{8})^{2N}} \right)^{1/2}, \mu \in \mathcal{P}.$$

*Proof.* The procedure to obtain (1.54) is analogous to the proof in Theorem 8. In what follows we only prove i), the other cases are analogous.

$$\begin{aligned}
\|u - u_{N,Qf}\|_V = \Bigg\| & \sum_{k=0}^{\infty} \sum_{q=1}^{\infty} \frac{(-1)^k (\Theta_a^2)^k \Theta_f^q}{(\Theta_a^1)^{k+1}} (A_1^{-1}A_2)^k A_1^{-1} f_q \\
& - \sum_{k=0}^{N} \sum_{q=1}^{Q_f} \frac{(-1)^k (\Theta_a^2)^k \Theta_f^q}{(\Theta_a^1)^{k+1}} (A_1^{-1}A_2)^k A_1^{-1} f_q . \Bigg\|_V \\
\leq & \left\| \sum_{k=0}^{\infty} \frac{(-1)^k (\Theta_a^2)^k}{(\Theta_a^1)^{k+1}} (A_1^{-1}A_2)^k A_1^{-1} \sum_{q=Q_f+1}^{\infty} \Theta_f^q f_q \right\|_V \\
& + \left\| \sum_{k=N+1}^{\infty} \frac{(-1)^k (\Theta_a^2)^k}{(\Theta_a^1)^{k+1}} (A_1^{-1}A_2)^k A_1^{-1} \sum_{q=1}^{Q_f} \Theta_f^q f_q \right\|_V \\
\leq & \frac{1}{\alpha_1 |\Theta_a^1|(1 - r)} \left\| \sum_{q=Q_f+1}^{\infty} \Theta_f^q f_q \right\|_{V'} + \frac{r^N}{\alpha_1 |\Theta_a^1|(1 - r)} \left\| \sum_{q=1}^{Q_f} \Theta_f^q f_q \right\|_{V'} \\
\leq & \frac{1}{\alpha_1 |\Theta_a^1|(1 - r)} \Big( C\varepsilon_{Q_f} + \|f_{Q_f}\|_{V'} r^N \Big).
\end{aligned}$$

□

The degree of decay of $\varepsilon_{Q_f}$ in Theorem 10 depends on the smoothness of $f$ with more smoothness meaning faster decay [61].

The following corollaries are immediate consequences of previous results.

**Corollary 3.** *Suppose the conditions of Theorem 10 and additionally* $\Theta_a^1(\mu) = 1$ *and* $\Theta_a^2(\mu) = 0$ *for all* $\mu \in \mathcal{P}$. *Then*

$$u(\mu) = \sum_{q=1}^{\infty} \Theta_f^q(\mu) A_1^{-1} f_q, \tag{1.55}$$

*is the solution to equation (1.53) with estimate*

$$\|u(\mu) - u_{0,Qf}(\mu)\|_V \leq \frac{C}{\alpha_1 |\Theta_a^1(\mu)|} \varepsilon_{Q_f}(\mu), \qquad \mu \in \mathcal{P}. \tag{1.56}$$

**Corollary 4.** *Suppose the conditions of Theorem 10 and that for all* $\mu \in \mathcal{P}$, *there exist constants* $\varepsilon, r, \Theta_a^1, C_f$, *such that*

$$\varepsilon_{Q_f}(\mu) \leq \varepsilon, \qquad r^N(\mu) \leq r^N < 1,$$
$$|\Theta_a^1(\mu)| \geq \Theta_a^1 > 0, \qquad \|f_{Q_f}(\mu)\|_{V'} < C_f.$$

*i) If the hypothesis of Proposition 5 are satisfied, then there exists a constant* $C_k$ *independent of* $\mu$ *such that*

$$d_n(\mathcal{M}) \leq C_k(\varepsilon_{Q_f} + r^n).$$

*ii) If the hypothesis of Proposition 6 are satisfied, then there exists a constant* $C_k$ *independent of* $\mu$ *such that*

$$d_n(\mathcal{M}) \leq C_k(\varepsilon_{Q_f}^2 + r^{2n})^{1/2}.$$

*iii) If the hypothesis of Proposition 7 are satisfied, then there exists a constant* $C_k$ *independent of* $\mu$ *such that AFOR implies*

$$d_n(\mathcal{M}) \leq C_k\Big(\varepsilon_{Q_f}^2 + \frac{1}{(3+\sqrt{8})^{2n}}\Big)^{1/2}.$$

We exemplify now an application of FOR to a simple equation. The example has an unbounded parameter set which is usually one of the weaknesses of the RB methods. Later in Chapter 2 we show another use of FOR in a more involved equation.

**Example 1.** Consider a bounded domain $\Omega \subset \mathbb{R}^n$, $V = H_0^1(\Omega)$, $\lambda_1$ the smallest singular value of the Laplace operator, $\mathcal{P} = (1/\lambda_1, \infty)$ and the equation

$$-\mu\Delta u + u = f. \tag{1.57}$$

This equation has the structure of (1.40) with

$$\Theta_a^1(\mu) = \mu, \quad \Theta_a^2(\mu) = 1, \quad \Theta_f^1(\mu) = 1, \quad Q_f = 1,$$

therefore we can use Theorem 8 if we check Lax-Milgram conditions for the bilinear form of the operators $-\mu\Delta + I$ and $\Delta$, which is straightforward, and the spectral radius condition (1.41).

The Laplace operator $-\Delta$ is a symmetric elliptic operator, hence all its eigenvalues are positives with $\lambda_1 > 0$. Consequently,

$$\rho\Big(-\frac{1}{\mu}\Delta^{-1}I\Big) = \frac{1}{\mu}\rho\Big(-\Delta^{-1}\Big) = \frac{1}{\mu\lambda_1} < 1, \quad \forall \mu \in \mathcal{P}.$$

By Theorem 8 the exact solution to the equation (1.57) is

$$u(\mu) = \sum_{k=0}^{\infty}(-1)^k \frac{1}{\mu^{k+1}}(-\Delta)^{-(k+1)} f, \qquad \forall \mu \in \mathcal{P}.$$

To use FOR for approximating the solution we define a finite subspace $V_h \subset V$. By Theorem 8 again, the solution to (1.57) in $V_h$ is

$$u_h(\mu) = \sum_{k=0}^{\infty}(-1)^k \frac{1}{\mu^{k+1}}(-\Delta_h)^{-(k+1)} f_h, \qquad \forall \mu \in \mathcal{P}, \tag{1.58}$$

where $-\Delta_h$ is the symmetric matrix representing the Laplace operator.

For better performance we divide $\mathcal{P}$ in the sets

$$\mathcal{P}_1 := \left(\frac{1}{\lambda_1}, \frac{3+\sqrt{8}}{\lambda_1}\right), \quad \mathcal{P}_2 := \left[\frac{3+\sqrt{8}}{\lambda_1}, \infty\right),$$

and use Proposition 6 and Proposition 7 accordingly. For $\mathcal{P}_1$ we apply AFOR, estimate (1.52) entails that with the first N summands of (1.58) we obtain

$$\|u_h(\mu) - u_{h,N}(\mu)\|_V \leq \frac{2\|f_h\|_{V_h'}}{\sqrt{\mu^3(\mu-\lambda_{\min})}} \frac{1}{(3+\sqrt{8})^N},$$

where $\lambda_{\min}$ is the smallest eigenvalue of $(-\Delta_h)^{-1}$. Whereas in $\mathcal{P}_2$ we use FOR with estimate (1.50),

$$\|u_h(\mu) - u_{h,N}(\mu)\|_V \leq \frac{\lambda_1\|f_h\|_{V_h'}}{\sqrt{(\mu\lambda_1)^2-1}} \frac{1}{(\mu\lambda_1)^N}.$$

Computing the exact spectral radius value of a differential operator is a difficult task, so is better to obtain $\lambda_1$ from $V_h$ and use it in the method.

Notice that the larger $\mu$ becomes the smaller N is needed to obtain a good approximation. This makes sense as

$$-\mu\Delta u + u = f \xrightarrow[\mu\to\infty]{} -\Delta u = 0,$$

and then the solution is $u \equiv 0$.

Finally, Theorem 9 gives the following Kolmogorov's $n$-width estimate,

$$d_n(\mathcal{M}) \leq \frac{C}{(3+\sqrt{8})^n}.$$

□

The two main weaknesses of the FOR method are the limited setting where it can be applied and the difficulty to verify the hypothesis of the results. A way to surpass the lastest is to work directly in a finite dimensional space as it is easier to perform computations. Nevertheless, in the cases where FOR is applicable, it is recommended to use is it over RB because of the advantage in the offline and online stage.

In the next section we continue with estimation results regarding the RB methods.

## 1.6 Reduced Basis Error Estimates

The new contributions to the reduced basis methodology are presented in this section. They are *offline error estimators* and are based in the possibility of certifying RB approximations in the offline stage. We introduce two simple but effective estimators of this kind: *Lipschitz offline estimator (Loe)* and *Chebyshev offline estimator (Coe)*.

There are two practical methodologies commonly used to check an effective RB space. One is an heuristic way to check reducibility by applying POD with a fairly dense $\Xi_s$ and check the decay of the singular values. The other common procedure is to certify the error using the *residual*

$$r(v;\boldsymbol{\mu}) = f(v;\boldsymbol{\mu}) - a(u_N(\boldsymbol{\mu}), v;\boldsymbol{\mu}), \qquad \forall v \in V_h, \tag{1.59}$$

for a-posteriori error estimations in the online stage.

**Proposition 8.** *The residual* $r(\cdot;\boldsymbol{\mu})$ *satisfies*

$$\frac{\|r(\cdot;\boldsymbol{\mu})\|_{V_h'}}{\gamma_h(\boldsymbol{\mu})} \leq \|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V \leq \frac{\|r(\cdot;\boldsymbol{\mu})\|_{V_h'}}{\alpha_h(\boldsymbol{\mu})}. \tag{1.60}$$

*Proof.* First note that $a(u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu}), v;\boldsymbol{\mu}) = r(v;\boldsymbol{\mu})$ for all $v \in V_h$, then from the continuity of $a(\cdot,\cdot;\boldsymbol{\mu})$,

$$|r(v;\boldsymbol{\mu})| = |a(u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu}), v;\boldsymbol{\mu})| \leq \gamma_h(\boldsymbol{\mu})\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V\|v\|_V, \quad \forall v \in V_h.$$

Therefore, the first inequality of (1.60) comes from the definition of dual norm,

$$\|r(\cdot;\boldsymbol{\mu})\|_{V_h'} \leq \gamma_h(\boldsymbol{\mu})\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V.$$

The remaining inequality yields from the stability estimate (1.3) applied to the solution $u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})$ of the variational problem

$$a(u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu}), v;\boldsymbol{\mu}) = r(v;\boldsymbol{\mu}),$$

which gives

$$\alpha_h(\boldsymbol{\mu})\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V \leq \|r(\cdot;\boldsymbol{\mu})\|_{V_h'}. \qquad \square$$

Using the estimation (1.60) requires the computation of the coercivity factor and the norm of the residual (1.59) for each RB solution. If there is not an efficient offline/online decoupling, computing the residual can be a time consuming task for the online stage purpose. The main

objective of offline estimators is to speedup even further the online stage passing the workload of the online estimators to the offline stage.

The main attribute of the two offline estimators presented here use the smoothness of the solution map. The first one is based on Lipschitz continuity.

**Definition 3.** *A parameterized bilinear form $a(\cdot,\cdot;\mu)$ is Lipschitz-continuous with respect to $\mu$ (uniformly with respect to $v$ and $w$) if there exists $L_a > 0$ such that*

$$|a(v,w;\mu) - a(v,w;\mu')| \leq L_a\|v\|_V\|w\|_V\|\mu-\mu'\| \quad \forall \mu,\mu' \in \mathcal{P},\ \forall v,w \in V. \tag{1.61}$$

*Likewise, $f(\cdot;\mu)$ is Lipschitz-continuous with respect to $\mu$ (uniformly with respect $v$) if there exists $L_f > 0$ such that*

$$|f(v;\mu) - f(v;\mu')| \leq L_f\|v\|_V\|\mu-\mu'\| \quad \forall \mu,\mu' \in \mathcal{P},\ \forall v \in V. \tag{1.62}$$

**Proposition 9.** *Let $a(\cdot,\cdot;\mu)$ and $f(\cdot;\mu)$ be Lipschitz-continuous with respect to $\mu$ with coercivity factor $\alpha(\mu)$ and norm $\|f(\mu)\|_{V'}$ uniformly bounded below and above with bounds $\alpha_0$ and $\gamma_f$ respectively. Then the solution $u(\mu)$ of (1.6) is Lipschitz-continuous with respect to $\mu$,*

$$\|u(\mu) - u(\mu')\|_V \leq L_u\|\mu-\mu'\|, \qquad \forall \mu,\mu' \in \mathcal{P},$$

*with constant*

$$L_u = \frac{1}{\alpha_0}\left(L_f + L_a\frac{\gamma_f}{\alpha_0}\right). \tag{1.63}$$

*Proof.* Subtracting the equations

$$a(u(\mu),u(\mu)-u(\mu');\mu) = f(u(\mu)-u(\mu');\mu),$$

$$a(u(\mu'),u(\mu)-u(\mu');\mu') = f(u(\mu)-u(\mu');\mu'),$$

rearranging and using the Lipschitz continuity yields

$$\begin{aligned} a(u(\mu)-u(\mu'),u(\mu)-u(\mu');\mu) &= f(u(\mu)-u(\mu');\mu) - f(u(\mu)-u(\mu');\mu') \\ &\quad - a(u(\mu'),u(\mu)-u(\mu');\mu) \\ &\quad + a(u(\mu'),u(\mu)-u(\mu');\mu') \\ &\leq L_f\|u(\mu)-u(\mu')\|_V\|\mu-\mu'\| \\ &\quad + L_a\|u(\mu')\|_V\|u(\mu)-u(\mu')\|_V\|\mu-\mu'\|. \end{aligned}$$

Taking the coercivity condition (1.2) and estimates (1.3) results the Lipschitz continuity of $u$ with constant $L_u$ from

$$\alpha(\mu)\|u(\mu)-u(\mu')\|_V \leq \frac{a(u(\mu)-u(\mu'),u(\mu)-u(\mu');\mu)}{\|u(\mu)-u(\mu')\|_V} \leq L_f\|\mu-\mu'\| + L_a\frac{\|f(\mu')\|_{V'}}{\alpha(\mu')}\|\mu-\mu'\|.$$

□

The following proposition is the theoretical justification for Loe estimator. It takes advantage of the Lipschitz-continuity of the solution to obtain an error estimate of the RB solution.

**Proposition 10.** *Suppose that* $a(\cdot,\cdot;\mu)$ *and* $f(\cdot;\mu)$ *are Lipschitz-continuous in* $V$ *with* $a(\cdot,\cdot;\mu)$ *symmetric and coercive, then*

$$\|u_h(\mu) - u_N(\mu)\|_V \leq 2L_u d(\mu,\Xi_s) + \left(\frac{\gamma_h(\mu)}{\alpha_N(\mu)}\right)^{1/2} \sqrt{\sum_{i=N+1}^{r} \sigma_i^2}, \quad \forall \mu \in \mathcal{P}, \tag{1.64}$$

*where* $d(\mu,\Xi_s) := \min_{\mu' \in \Xi_s} \|\mu - \mu'\|$ *is the distance from* $\mu$ *to* $\Xi_s$.

*Proof.* The hypothesis of Proposition 4 and Proposition 9 are satisfied and therefore the estimate (1.64) results directly from applying them to

$$\|u_h(\mu) - u_N(\mu)\|_V \leq \|u_h(\mu) - u_h(\mu')\|_V + \|u_h(\mu') - u_N(\mu')\|_V + \|u_N(\mu') - u_N(\mu)\|_V,$$

with $\mu'$ the closest point to $\mu$ in $\Xi_s$. □

**Definition 4.** *The Lipschitz offline estimator (Loe) is defined by the map*

$$\Delta_N^L(\mu) := 2L_u d(\mu,\Xi_s) + \left(\frac{\gamma_h(\mu)}{\alpha_N(\mu)}\right)^{1/2} \sqrt{\sum_{i=N+1}^{r} \sigma_i^2}. \tag{1.65}$$

We can obtain a simple bound to Loe if the factors in (1.65) are bounded uniformly,

$$d(\mu,\Xi_s) \leq d_{max}, \qquad \alpha_h(\mu) \geq \alpha_0, \qquad \gamma_N(\mu) \leq \gamma_0, \qquad \forall \mu \in \mathcal{P},$$

then it results

$$\Delta_N^L(\mu) \leq 2L_u d_{max} + \left(\frac{\gamma_0}{\alpha_0}\right)^{1/2} \sqrt{\sum_{i=N+1}^{r} \sigma_i^2}, \qquad \forall \mu \in \mathcal{P}.$$

**Corollary 5.** *If* $a(\cdot,\cdot;\mu)$ *is constant with respect to* $\mu$*, i.e. the bilinear form is* $\mu$*-independent, and* $N = r$*, then*

$$\Delta_N^L(\mu) \leq 2\left(\frac{L_f}{\alpha_h} + \frac{L_f}{\alpha_N}\right) d_{max}, \quad \forall \mu \in \mathcal{P}. \tag{1.66}$$

*Proof.* As the bilinear form is $\mu$-independent (1.63) becomes

$$L_u = \frac{L_f}{\alpha_0}.$$

Using this equality with the corresponding coercivity constants in the proof of Proposition 10 results the estimate (1.66). □

Even if we have fast decay of the singular values in POD, the expression (1.65) can deteriorate anyway if $L_u$ is large. This could happen due to sharp changes in the structure of one solution to another or if $\Xi_s$ is not very dense in $\mathcal{P}$, i.e. if the distance $d_{max}$ is not small enough.

The following proposition relates the a-posteriori error estimate (1.60) and Loe.

**Proposition 11.** *Denote the a-posteriori estimate (1.60) as*

$$\Delta_N(\boldsymbol{\mu}) := \frac{\|r(\cdot;\boldsymbol{\mu})\|_{V_h'}}{\alpha_h(\boldsymbol{\mu})},$$

*then*

$$\Delta_N(\boldsymbol{\mu}) \leq \frac{\gamma_h(\boldsymbol{\mu})}{\alpha_h(\boldsymbol{\mu})}\Delta_N^L(\boldsymbol{\mu}).$$

*Proof.* From the relation (1.60) and (1.64) yields

$$\frac{\|r(\cdot;\boldsymbol{\mu})\|_{V_h'}}{\gamma_h(\boldsymbol{\mu})} \leq \|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V \leq \Delta_N^L(\boldsymbol{\mu}),$$

which implies that

$$\Delta_N(\boldsymbol{\mu}) = \frac{\|r(\cdot;\boldsymbol{\mu})\|_{V_h'}}{\alpha_h(\boldsymbol{\mu})} \leq \frac{\gamma_h(\boldsymbol{\mu})}{\alpha_h(\boldsymbol{\mu})}\Delta_N^L(\boldsymbol{\mu}).$$ □

As explained before, Loe may be of low utility in some cases. More adequate in general is the Chebyshev offline estimator (Coe) which use Chebyshev polynomials [159] to exploit higher smoothness of the solution map in the a-posteriori estimates.

**Definition 5.** *The bilinear form $a(\cdot,\cdot;\boldsymbol{\mu})$ is differentiable with respect to $\mu_i$ at a point $\boldsymbol{\mu} \in \mathcal{P}$ if, for any $v, w \in V$, the limit*

$$\frac{\partial a}{\partial \mu_i}(v,w;\boldsymbol{\mu}) = \lim_{h\to 0}\frac{1}{h}(a(v,w;\boldsymbol{\mu}+h\mathbf{e}_i) - a(v,w;\boldsymbol{\mu}))$$

*exists. In the same way, $f(\cdot;\boldsymbol{\mu})$ will be differentiable with respect to $\mu_i$ at a point $\boldsymbol{\mu} \in \mathcal{P}$ if, for any $v \in V$, the following limit exists*

$$\frac{\partial f}{\partial \mu_i}(v;\boldsymbol{\mu}) = \lim_{h\to 0}\frac{1}{h}(f(v;\boldsymbol{\mu}+h\mathbf{e}_i) - f(v;\boldsymbol{\mu})v) \tag{1.67}$$

**Proposition 12.** *Let $a(\cdot,\cdot;\boldsymbol{\mu})$ and $f(\cdot;\boldsymbol{\mu})$ be $C^k$ (analytic) maps with respect to $\boldsymbol{\mu}$, for some $k \geq 0$ with $a(\cdot,\cdot;\boldsymbol{\mu})$ coercive for all $\boldsymbol{\mu} \in \mathcal{P}$. Then, the solution map $u(\boldsymbol{\mu})$ is of class $C^k$ (analytic) with respect to $\boldsymbol{\mu}$.*

*Proof.* See [131]. □

The following proposition gives the algebraic counterpart of the a-posteriori error estimate $\Delta_N(\boldsymbol{\mu})$. This is the usual way to calculate error estimates after the RB solution is computed.

**Proposition 13.** *Denote the norm $\|\cdot\|_{\mathbf{X}_h^{-1}}$ in $\mathbb{R}^{N_h}$ as*

$$\|\mathbf{x}\|_{\mathbf{X}_h^{-1}} := \|\mathbf{X}_h^{-1/2}\mathbf{x}\|_2, \qquad \mathbf{x} \in \mathbb{R}^{N_h},$$

*then*

$$\|\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{U}_N\mathbf{u}_N(\boldsymbol{\mu})\|_{\mathbf{X}_h} \leq \frac{1}{\alpha_h(\boldsymbol{\mu})}\|\mathbf{r}_h(\mathbf{u}_N;\boldsymbol{\mu})\|_{\mathbf{X}_h^{-1}}. \tag{1.68}$$

*Proof.* The discrete residual of (1.59) evaluated in $\mathbf{u}_N(\boldsymbol{\mu})$ is

$$\mathbf{r}_h(\mathbf{u}_N;\boldsymbol{\mu}) = \mathbf{f}_h(\boldsymbol{\mu}) - \mathbf{A}_h(\boldsymbol{\mu})\mathbf{U}_N\mathbf{u}_N(\boldsymbol{\mu})$$

with satisfies the following equality using (1.5),

$$\mathbf{A}_h(\boldsymbol{\mu})(\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{U}_N\mathbf{u}_N(\boldsymbol{\mu})) = \mathbf{r}_h(\mathbf{u}_N;\boldsymbol{\mu}),$$

and consequently

$$\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{U}_N\mathbf{u}_N(\boldsymbol{\mu}) = \mathbf{A}_h(\boldsymbol{\mu})^{-1}\mathbf{r}_h(\mathbf{u}_N;\boldsymbol{\mu}). \tag{1.69}$$

Multiplying (1.69) by $\mathbf{X}_h^{1/2}$

$$\mathbf{X}_h^{1/2}(\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{U}_N\mathbf{u}_N(\boldsymbol{\mu})) = \mathbf{X}_h^{1/2}\mathbf{A}_h(\boldsymbol{\mu})^{-1}\mathbf{X}_h^{1/2}\mathbf{X}_h^{-1/2}\mathbf{r}_h(\mathbf{u}_N;\boldsymbol{\mu}),$$

and taking $\|\cdot\|_2$ at both sides yields

$$\|\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{U}_N\mathbf{u}_N(\boldsymbol{\mu})\|_{\mathbf{X}_h} \leq \|\mathbf{X}_h^{1/2}\mathbf{A}_h(\boldsymbol{\mu})^{-1}\mathbf{X}_h^{1/2}\|_2\|\mathbf{r}_h(\mathbf{u}_N;\boldsymbol{\mu})\|_{\mathbf{X}_h^{-1}}.$$

The final expression in (1.68) results from

$$\|\mathbf{X}_h^{1/2}\mathbf{A}_h(\boldsymbol{\mu})^{-1}\mathbf{X}_h^{1/2}\|_2 = \sigma_{\max}(\mathbf{X}_h^{1/2}\mathbf{A}_h(\boldsymbol{\mu})^{-1}\mathbf{X}_h^{1/2}) = \frac{1}{\sigma_{\min}(\mathbf{X}_h^{-1/2}\mathbf{A}_h(\boldsymbol{\mu})\mathbf{X}_h^{-1/2})} = \frac{1}{\alpha_h(\boldsymbol{\mu})}. \qquad \square$$

We now present the main ingredient for the Coe estimator: the smoothness of the residual as a function of the parameters.

**Proposition 14.** *Let* $a(\cdot,\cdot;\boldsymbol{\mu})$ *and* $f(\cdot;\boldsymbol{\mu})$ *be* $C^k$ *(analytic) maps with respect to* $\boldsymbol{\mu}$ *for some* $k \geq 0$*. If* $a(\cdot,\cdot;\boldsymbol{\mu})$ *is coercive for all* $\boldsymbol{\mu} \in \mathcal{P}$*, then the residual norm*

$$r_{h,2}(\boldsymbol{\mu}) := \|\mathbf{r}_h(\mathbf{u}_N;\boldsymbol{\mu})\|^2_{\mathbf{X}_h^{-1}},$$

*is a map*

$$r_{h,2}(\boldsymbol{\mu}) : \mathcal{P} \to \mathbb{R}$$

*of class* $C^k$ *(analytic).*

*Proof.* Expanding the definition of $r_{h,2}(\boldsymbol{\mu})$ in terms of vectors yields

$$r_{h,2}(\boldsymbol{\mu}) = \|\mathbf{r}_h(\mathbf{u}_N;\boldsymbol{\mu})\|^2_{\mathbf{X}_h^{-1}} = \|\mathbf{X}_h^{-1/2}\mathbf{r}_h(\mathbf{u}_N;\boldsymbol{\mu})\|_2^2 = \mathbf{r}_h(\mathbf{u}_N;\boldsymbol{\mu})^T\mathbf{X}_h^{-1}\mathbf{r}_h(\mathbf{u}_N;\boldsymbol{\mu}) =$$

$$\mathbf{f}_h(\boldsymbol{\mu})^T\mathbf{X}_h^{-1}\mathbf{f}_h(\boldsymbol{\mu}) - 2\mathbf{f}_h(\boldsymbol{\mu})^T\mathbf{X}_h^{-1}\mathbf{A}_h(\boldsymbol{\mu})\mathbf{U}_N\mathbf{u}_N(\boldsymbol{\mu}) + \mathbf{u}_N(\boldsymbol{\mu})^T\mathbf{U}_N^T\mathbf{A}_h(\boldsymbol{\mu})^T\mathbf{X}_h^{-1}\mathbf{A}_h(\boldsymbol{\mu})\mathbf{U}_N\mathbf{u}_N(\boldsymbol{\mu}). \tag{1.70}$$

This last expression shows that $r_{h,2}(\boldsymbol{\mu})$ is obtained by multiplications and sums of $C^k$ (analytic) functions and therefore $r_{h,2}(\boldsymbol{\mu})$ is of class $C^k$. $\square$

Once the residual is identified as a smooth function we open two possibilies:

- Apply non-linear optimization methods to find for which parameter the residual attains its maximum. The inconvenience of this approach is finding sub-optimal solutions giving the disadvantage of non-linear optimization methods to get stuck in local optimums.
- Approximate the residual for faster computation times and if possible look for maximum values without the need of non-linear programming.

For the Coe estimator we use Chebyshev interpolation of the residual. This choice is due to the logarithmic growth of the Lebesgue constant, good approximation properties for smooth functions and stable computations [116, 159]. The downsides are that it may be impractical on general domains and high dimensional spaces.

Chebyshev polynomials (of first kind) $\{T_n(x)\}, x \in [-1,1]$, can be defined recursively as

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \tag{1.71}$$

where the first three polynomials are

$$T_0(x) = 1, \qquad T_1(x) = x, \qquad T_2(x) = 2x^2 - 1.$$

The Chebyshev series expansion of a function $f(x)$ defined in $[-1,1]$ follows as

$$f(x) \sim S_C f(x) := \frac{1}{2}c_0T_0(x) + c_1T_1(x) + c_2T_2(x) + \cdots, \tag{1.72}$$

with

$$c_i = \frac{2}{\pi}\int_{-1}^{1}(1-x^2)^{-1/2}f(x)T_i(x)dx.$$

Pointwise convergence is guaranteed for continuous functions and is very fast in many cases.

**Theorem 11.** *If the function $f(x)$ is continuous in $[-1,1]$, then its Chebyshev series is pointwise convergent. If $f(x)$ has $m+1$ continuous derivatives in $[-1,1]$, then*

$$|f(x) - S_n^T f(x)| = O(n^{-m}), \qquad \forall x \in [-1,1].$$

*Proof.* See [116]. □

**Theorem 12.** *If $f(x)$ can be extended to a function $f(z)$ analitic on an ellipse of radius $r > 1$ which contains $[-1,1]$, then*

$$|f(x) - S_n^T f(x)| = O(r^{-n}), \qquad \forall x \in [-1,1].$$

*Proof.* See [116]. □

The simplest way to obtain a polynomial approximation with Chebyshev polynomials is to interpolate the desired function. Let denote

$$p^* = \underset{p \in \mathbb{P}_n}{\operatorname{argmin}} \|f - p\|_\infty \in \mathbb{P}$$

the best approximation of $f$ among the polynomials of degree $n$, then the interpolation error is bounded by

$$\|f - \mathcal{I}_n f\|_\infty \leq (1 + \eta_n)\|f - p^*\|_\infty, \tag{1.73}$$

where $\eta_n$ denotes the Lebesgue constant related to the interpolation points. If the the interpolation points are selected to be the zeros of the Chebyshev polynomials then the Lebesgue constant grows logarithmically

$$\eta_n \leq \frac{2}{n}\log(n+1) + 1,$$

which reduces the Runge phenomenon [116].

We can generalize the approximation and interpolation results to simple domains like general hypercubes [39, 115, 157, 160]. Moving the range $[-1, 1]$ to a general interval is a matter of making a linear transformation and extension to the multivariate case could be done through different techniques, for example, by tensor product.

**Definition 6.** *The Chebyshev offline estimator (Coe) is defined as the Chebyshev polynomial obtained by interpolating $r_{h,2}$ defined in Proposition 14.*

We can use a Coe estimator in two ways:

- online: for faster online estimator
- offline: for computing a maximum error estimate

If we do not have an affine representation of $a(\cdot,\cdot;\boldsymbol{\mu})$ or $f(\cdot;\boldsymbol{\mu})$ it may be faster to evaluate the polynomial interpolation than the residual because the expansion (1.70) needs to execute operations in the high dimensional space $V_h$. Instead, for the Chebyshev polynomial

$$p_n(\boldsymbol{\mu}) = \sum_{i=0}^{n} c_i T_i(\boldsymbol{\mu}), \qquad \boldsymbol{\mu} \in \mathcal{P},$$

we can use different strategies to evaluate it which are stable and has $O(n)$ operations [33, 85, 140].

In most cases is better to apply Coe offline instead of online. With a precise polynomial approximation we may simply compute the maximum and obtain a global error estimate. In this way we reduce the number of operations executed in the online stage.

To use Coe is recommended to have a smooth residual so the Chebyshev coefficients $c_i$ decay fast. This is because evaluating the residual in high-dimensional spaces to compute the polynomial could be CPU intensive. Also in this regard is important to have an affine representation of the parametric part to speedup the computations. Without these two points: fast decay of Chebyshev coefficients and an affine representation of the parameters, in some cases it may be expensive to construct a Coe estimator.

Also, if the residual is so small that is close to the machine epsilon, it is better to interpolate each term of the residual (1.70) individually and not the final result because some instabilities may appear in the computation of the coefficients of the polynomial.

The main disadvantage of Coe comes from the actual computation of the polynomial. Also, general and high-dimensional parameters domains are difficult to treat. Nevertheless, this is an ongoing research topic with many promising results [3, 25].

Next section explains how to compute an affine representation of the parameters when is not available. Reduced basis methods in general can be slow without it in the online stage and also helps computing the residual.

## 1.7 Empirical Interpolation Method

This section introduces the *empirical interpolation method* (EIM) [23, 75, 111] which builds an affine representation of parameterized functions. The main idea behind it is to construct iteratively an operator $\mathcal{I}_M^{\mathbf{x}}$ that interpolates the spacial variables of a function $g(\cdot;\boldsymbol{\mu})$ in a separable expansion. It has many advantages like exponential convergence for analytic functions, usefulness in general domains and a hierarchical algorithm.

First we show how to use EIM when its offline stage is done and then how to actually iterate through the algorithm.

Let $\mathcal{G}$ be a family of parameter-dependent functions

$$\mathcal{G} = \{g(\cdot;\boldsymbol{\mu}), \boldsymbol{\mu} \in \mathcal{P}\} \subset C^0(\overline{\Omega}).$$

The output of the EIM offline algorithm is:

- a set of *basis functions* $\{\rho_1, \dots, \rho_M\} \subset \operatorname{span}\{\mathcal{G}\}$
- a set of *interpolation points* or *magic points* $T_M = \{\mathbf{t}^1, \dots, \mathbf{t}^M\} \subset \overline{\Omega}$
- the interpolation operator $\mathcal{I}_M^{\mathbf{x}}$

$$\mathcal{I}_M^{\mathbf{x}} g(\mathbf{x};\boldsymbol{\mu}) := \sum_{j=1}^{M} \theta_j(\boldsymbol{\mu})\rho_j(\mathbf{x}), \qquad \mathbf{x} \in \Omega, \boldsymbol{\mu} \in \mathcal{P}. \tag{1.74}$$

Once the EIM algorithm finish and we want to compute $g(\cdot,\boldsymbol{\mu})$, for a selected $\boldsymbol{\mu} \in \mathcal{P}$, we force the coefficients $\theta_j(\boldsymbol{\mu})$ to interpolate the values of $g(\cdot,\boldsymbol{\mu})$ in $M$ points,

$$\mathcal{I}_M^{\mathbf{x}} g(\mathbf{t}^i;\boldsymbol{\mu}) = g(\mathbf{t}^i;\boldsymbol{\mu}), \qquad i = 1, \dots, M. \tag{1.75}$$

To obtain the values $\theta_j(\boldsymbol{\mu})$ we substitute (1.74) in (1.75) and solve the linear system

$$\sum_{j=1}^{M} \theta_j(\boldsymbol{\mu})\rho_j(\mathbf{t}^i) = g(\mathbf{t}^i;\boldsymbol{\mu}), \qquad i = 1, \dots, M,$$

or equivalently

$$\mathbf{B}_M \boldsymbol{\theta}(\boldsymbol{\mu}) = \mathbf{g}_M(\boldsymbol{\mu}), \qquad \forall \boldsymbol{\mu} \in \mathcal{P}, \tag{1.76}$$

where

$$(\mathbf{B}_M)_{ij} = \rho_j(\mathbf{t}^i), \quad (\theta(\boldsymbol{\mu}))_j = \theta_j(\boldsymbol{\mu}), \quad (\mathbf{g}_M(\boldsymbol{\mu}))_i = g(\mathbf{t}^i;\boldsymbol{\mu}), \qquad i,j = 1,\dots,M.$$

Thus, from (1.74) and the solution vector of (1.76) we obtain a representation of $g(\cdot,\boldsymbol{\mu})$ for every $\boldsymbol{\mu} \in \mathcal{P}$.

**EIM Algorithm.**

1. Select the parameter $\boldsymbol{\mu}^1_{\mathrm{EIM}}$ such that

$$\boldsymbol{\mu}^1_{\mathrm{EIM}} = \operatorname*{argmax}_{\boldsymbol{\mu}\in\mathcal{P}} \|g(\cdot;\boldsymbol{\mu})\|_{\mathcal{L}^\infty(\Omega)}.$$

After this define $S_1 := \{\boldsymbol{\mu}^1_{\mathrm{EIM}}\}$ and the first generating function as

$$\xi_1(\mathbf{x}) := g(\mathbf{x};\boldsymbol{\mu}^1_{\mathrm{EIM}}).$$

From $\xi_1$ select $\mathbf{t}^1$ such that

$$\mathbf{t}^1 = \operatorname*{argmax}_{\mathbf{x}\in\Omega} |\xi_1(\mathbf{x})|$$

and define $T_1 := \{\mathbf{t}^1\}$. Finally, compute the first basis function as

$$\rho_1(\mathbf{x}) := \xi_1(\mathbf{x})/\xi_1(\mathbf{t}^1).$$

For $m = 1$ the interpolation matrix $\mathbf{B}_m$ is

$$(\mathbf{B}_m)_{11} = \rho_1(\mathbf{t}^1) = 1,$$

and solving (1.76) for this particular $m$ yields the interpolation operator

$$\mathcal{I}^{\mathbf{x}}_1 g(\mathbf{x};\boldsymbol{\mu}) = g(\mathbf{t}^1;\boldsymbol{\mu})\rho_1(\mathbf{x}).$$

2. After $m-1$ steps we have obtained

$$T_{m-1} = \{\mathbf{t}^1,\dots,\mathbf{t}^{m-1}\},$$

$$S_{m-1} = \{\boldsymbol{\mu}^1_{\mathrm{EIM}},\dots,\boldsymbol{\mu}^{m-1}_{\mathrm{EIM}}\},$$

with the corresponding basis functions

$$\{\rho_1,\dots,\rho_{m-1}\}.$$

At the $m$-th step select the new parameter as

$$\boldsymbol{\mu}^m_{\mathrm{EIM}} = \operatorname*{argmax}_{\boldsymbol{\mu}\in\mathcal{P}} \|g(\cdot;\boldsymbol{\mu}) - \mathcal{I}^{\mathbf{x}}_{m-1} g(\cdot,\boldsymbol{\mu})\|_{\mathcal{L}^\infty(\Omega)} \tag{1.77}$$

and define the next generating function

$$\xi_m(\mathbf{x}) := g(\mathbf{x};\boldsymbol{\mu}^m_{\mathrm{EIM}}).$$

Then compute the residual

$$r_m(\mathbf{x}) = \xi_m(\mathbf{x}) - \mathcal{I}^x_{m-1}\xi_m(\mathbf{x})$$

and solve (1.76) with $M = m-1$ and $g = \xi_m$ to obtain the corresponding coefficients of the operator $\mathcal{I}^x_{m-1}$. Finally, define the new basis function as

$$\rho_m(\mathbf{x}) := r_m(\mathbf{x})/r_m(\mathbf{t}^m)$$

with

$$\mathbf{t}^m = \underset{\mathbf{x}\in\overline{\Omega}}{\operatorname{argmax}} |r_m(\mathbf{x})|. \tag{1.78}$$

□

In second step of the algorithm we should to prove that $\mathbf{B}_{m-1}$ is invertible. It follows directly from the fact that

$$(\mathbf{B}_{m-1})_{ij} = \rho_j(\mathbf{t}^i),$$

which implies

$$\begin{aligned}(\mathbf{B}_{m-1})_{ij} &= 0, \qquad i < j,\\ (\mathbf{B}_{m-1})_{ij} &= 1, \qquad i = j,\\ |(\mathbf{B}_{m-1})_{ij}| &\le 1, \qquad i > j,\end{aligned}$$

and hence $\mathbf{B}_{m-1}$ is lower triangular with all the elements of the diagonal different from zero.

There are some results regarding the convergence and error estimation of EIM [111, 131]. Next we enunciate an exponential a-priori estimate.

**Theorem 13.** *Suppose there exists a sequence of nested finite-dimensional spaces $\mathcal{G}_1 \subset \mathcal{G}_2 \subset \cdots$, such that there exists $c > 0$ and $\alpha > \log 4$ with*

$$d(\mathcal{G}, \mathcal{G}_M) = \sup_{\mu\in\mathcal{P}} \inf_{v_M\in\mathcal{G}_M} \|g(\cdot;\mu) - v_M\|_{C^0(\Omega)} \le ce^{-\alpha M}.$$

*Then*

$$\sup_{\mu\in\mathcal{P}} \|g(\cdot;\mu) - \mathcal{I}^{\mathbf{x}}_M g(\cdot;\mu)\|_{\mathcal{L}^\infty(\Omega)} \le ce^{-(\alpha-\log 4)M}.$$

*Proof.* See [111]. □

In the particular case of $\mu$-independent bilinear forms we can obtain a connection between RB, FOR and EIM.

**Proposition 15.** *Consider the conditions of Corollary 3 and $f(\mu) \in L^\infty(\Omega)$ for all $\mu \in \mathcal{P}$. If $\|g\|_{V'} \le C\|g\|_{L^\infty(\Omega)}$ for all $g \in V' \cap L^\infty(\Omega)$, and there exist $\varepsilon_{EIM} > 0$ independent of $\mu$ such that*

$$\sup_{\mu\in\mathcal{P}} \|f(\mu) - \mathcal{I}^{\mathbf{x}}_{Q_f} f(\mu)\|_{L^\infty(\Omega)} \le \varepsilon_{EIM}$$

*for $S_{Q_f} = \{\mu^1_{EIM}, \ldots, \mu^{Q_f}_{EIM}\}$, then*

$$\sup_{\mu\in\mathcal{P}} \inf_{\alpha_1,\cdots,\alpha_{Q_f}} \|u(\mu) - \sum_{k=1}^{Q_f} \alpha_k u(\mu^k_{EIM})\|_V \le C\varepsilon_{EIM}. \tag{1.79}$$

*Proof.* With the conditions of Corollary 3 we can transform the expression (1.79),

$$\begin{aligned}
\sup_{\mu\in\mathcal{P}}\inf_{\alpha_1,\cdots,\alpha_{Q_f}}\|u(\mu)-\sum_{k=1}^{Q_f}\alpha_k u(\mu_{EIM}^k)\|_V &= \sup_{\mu\in\mathcal{P}}\inf_{\alpha_1,\cdots,\alpha_{Q_f}}\|A_1^{-1}f(\mu)-\sum_{k=1}^{Q_f}\alpha_k A_1^{-1}f(\mu_{EIM}^k)\|_V\\
&\leq \frac{1}{\alpha_1}\sup_{\mu\in\mathcal{P}}\inf_{\alpha_1,\cdots,\alpha_{Q_f}}\|f(\mu)-\sum_{k=1}^{Q_f}\alpha_k f(\mu_{EIM}^k)\|_{V'}\\
&\leq C\sup_{\mu\in\mathcal{P}}\inf_{\alpha_1,\cdots,\alpha_{Q_f}}\|f(\mu)-\sum_{k=1}^{Q_f}\alpha_k f(\mu_{EIM}^k)\|_{L^\infty(\Omega)}\\
&\leq C\sup_{\mu\in\mathcal{P}}\|f(\mu)-\mathcal{I}_{Q_f}^{\mathbf{x}}f(\mu)\|_{L^\infty(\Omega)}\\
&\leq C\varepsilon_{EIM}
\end{aligned}$$

□

The result (1.79) shows that under certain conditions using the RB space

$$V_{Q_f}=\{u(\mu_{EIM}^1),\ldots,u(\mu_{EIM}^{Q_f})\}$$

generated with the same points of the EIM algorithm is as good as applying Corollary 2 after computing the affine representation with EIM. Nevertheless, this result does not take in account G-RB computations and the fact that Corollary 2 gives an exact solution with an affine representation. Also we can think about the RB space generated by the EIM points of being "optimal" because these points are selected in a greedy way.

In conclusion for $\mu$-independent differential operator, RB and FOR could give the same level of precision with an EIM affine representation but FOR is faster and more precise in computing actual approximations.

From the computational point of view, the EIM algorithm described before is not feasible as (1.77) and (1.78) are difficult to compute. Therefore, we need to select a fine sample $\Xi_{EIM}\subset\mathcal{P}$ and $\Omega_{N_q}=\{\mathbf{x}^k\}_{k=1}^{N_q}\subset\Omega$ to turn (1.77) and (1.78) to an enumeration problem.

With this idea we use in the EIM algorithm a vector representation

$$\mathbf{g}:\mathcal{P}\to\mathbb{R}^{N_q}\quad\text{of}\quad g:\Omega_{N_q}\times\mathcal{P}\to\mathbb{R}$$

defined as

$$(\mathbf{g}(\mu))_k=g(\mathbf{x}^k;\mu),\qquad k=1,\ldots,N_q,$$

and a matrix

$$\mathbf{Q}=\left[\rho_1|\cdots|\rho_M\right],$$

where in each column we have the discrete representation of the basis functions, i.e. $(\mathbf{Q})_{kj}=\rho_j(\mathbf{x}^k)$.

We also represent the interpolation operator $\mathcal{I}_M^{\mathbf{x}}$ by a discrete function $\mathbf{g}_M:\mathcal{P}\to\mathbb{R}^{N_q}$ given by

$$\mathbf{g}_M(\mu)=Q\theta(\mu)\in\mathbb{R}^{N_q},$$

where $\theta(\boldsymbol{\mu}) \in \mathbb{R}^M$ is the solution of a linear system equivalent to (1.75),

$$\sum_{j=1}^{M} \theta_j(\boldsymbol{\mu})(\boldsymbol{\rho}_j)_{i_m} = (\mathbf{g}(\boldsymbol{\mu}))_{i_m}, \qquad 1, \ldots, M.$$

The indices $\{i_1, \ldots, i_M\}$ correspond with the selected magic points,

$$\{\mathbf{t}^1, \ldots, \mathbf{t}^M\} = \{\mathbf{x}^{i_1}, \ldots, \mathbf{x}^{i_M}\}.$$

# Chapter 2

# RB and FOR on an Epilepsy EEG Equation

This chapter deals with a particular parameterized equation used in *electroencephalography* (EEG) for detecting *epilepsy waves* [47, 124]. The main topic is the application of model order reduction techniques presented in Chapter 1 to solve the equation faster.

An electroencephalogram is a noninvasive method used to record electrical activity of the brain with a collection of electrodes positioned on the scalp. From a mathematical point of view, EEG for epilepsy is an inverse problem, the known information are the measurements on the boundary of the domain, i.e. from the electrodes on the scalp, and we seek the electric field that generated it.

An EEG mathematical model for epilepsy [5, 82, 141] is given by

$$\begin{cases} \operatorname{div}(\sigma \nabla u) = \operatorname{div}(\mathbf{p}_0 \delta_\mu) & \text{in } \Omega, \\ (\sigma \nabla u) \cdot \mathbf{n} = 0 & \text{on } \partial\Omega. \end{cases} \tag{2.1}$$

The Lipschitz domain $\Omega$ is usually a geometry that assembles the human brain in $\mathbb{R}^d$, $d = 3$, with subdomains inside representing different elements of the brain; $\mathbf{n}$ is the unit outward normal vector; the conductivity $\sigma$ is a bounded matrix which satisfies the uniform elipticity condition; in the right hand side there are two parameters, the *polarization* $\mathbf{p}_0 \in \mathbb{R}^d$ and the *source parameter* $\mu \in \Omega$ for which the Dirac delta distribution $\delta_\mu$ is centered.

We cannot consider this equation in the classical sense, instead in a distributional sense [60]. In this context, the real vector $\mathbf{p}_0$ has meaning as a constant map from the cartesian product of distributions,

$$\mathbf{p}_0 : \mathcal{D} \times \mathcal{D} \to \mathcal{D}.$$

Equation (2.1) poses a huge challenge from the theorical and numerical point of view because the divergence of the delta function introduce a strong singularity in the solution. Nevertheless, there are two known ways to approximate its solution with theorical support, they are called *direct approach* [8, 9, 14, 163] and *subtraction approach* [8, 9, 14, 106, 168].

A relevant result in this thesis is the proof that reduced basis methods cannot be efficiently applied to the direct approach, which being a variational approach, could be presumed to be suitable for this aim. We postpone to Section 2.2 a theoretical proof related to this statement and the presentation of some numerical results that enlighten this point. Instead, we start in

Section 2.1 with the subtraction approach and its numerical approximation. Finally, in Section 2.3 we deal with the application of RB to the inverse problem to find the original source.

## 2.1 Subtraction Approach Formulation

We describe in this section the subtraction approach to solve the EEG equation (2.1). The idea is to remove the singularity with a function $u_0$,

$$u - u_0 = u_s$$

and solve a simpler equation for $u_s$.

Suppose $\sigma$ is constant with value $\sigma_0$ in a neighborhood of $\mu$. We will use the same notation $\sigma_0$ to denote the diagonal matrix with this value on the diagonal, thus there exists $\sigma_s$ such that

$$\sigma = \sigma_0 + \sigma_s. \tag{2.2}$$

For $\sigma_0$ consider the equation

$$\operatorname{div}(\sigma_0 \nabla u_0) = \operatorname{div}(\mathbf{p}_0 \delta_\mu). \tag{2.3}$$

As the fundamental solution of

$$-\operatorname{div}(\sigma_0 \nabla v) = \delta_\mu \tag{2.4}$$

is known and

$$\operatorname{div}(\mathbf{p}_0 \delta_\mu) = \operatorname{div}(\mathbf{p}_0 \cdot -\operatorname{div}(\sigma_0 \nabla v)) = -\operatorname{div}(\sigma_0 \nabla(\operatorname{div}(\mathbf{p}_0 v))),$$

then we obtain the exact solution of (2.3) by computing $\operatorname{div}(\mathbf{p}_0 \cdot v)$ in the fundamental solution of (2.4). This results

$$u_0(x;\mu) = \begin{cases} \dfrac{\mathbf{p}_0 \cdot \sigma_0^{-1}(x-\mu)}{2\pi\sqrt{\det \sigma_0}(x-\mu)\cdot\sigma_0^{-1}(x-\mu)}, & \text{if } n = 2, \\ \dfrac{\mathbf{p}_0 \cdot \sigma_0^{-1}(x-\mu)}{4\pi\sqrt{\det \sigma_0}\left((x-\mu)\cdot\sigma_0^{-1}(x-\mu)\right)^{3/2}}, & \text{if } n = 3. \end{cases} \tag{2.5}$$

If the medium is isotropic, i.e. $\sigma_0 = \sigma_0 \mathbf{I}$, then (2.5) simplifies to

$$u_0(x;\mu) = \begin{cases} \dfrac{\mathbf{p}_0 \cdot (x-\mu)}{2\pi\sigma_0 |x-\mu|^2}, & \text{if } n = 2, \\ \dfrac{\mathbf{p}_0 \cdot (x-\mu)}{4\pi\sigma_0 |x-\mu|^3}, & \text{if } n = 3. \end{cases} \tag{2.6}$$

Given that a solution $u$ of (2.1) exists, then there exists $u_s$ such that

$$u = u_0 + u_s. \tag{2.7}$$

Substituting this equality with (2.31) in (2.1) yields

$$\operatorname{div}((\sigma_0 + \sigma_s)(\nabla(u_0 + u_s))) = \operatorname{div}(\mathbf{p}_0\delta_\mu),$$

and

$$(\sigma_0 + \sigma_s)\nabla(u_0 + u_s) \cdot n = 0,$$

which gives an equation for $u_s$,

$$\begin{cases} \operatorname{div}(\sigma\nabla u_s) = -\operatorname{div}(\sigma_s\nabla u_0(\mu)) & \text{in } \Omega, \\ (\sigma\nabla u_s) \cdot \mathbf{n} = -(\sigma\nabla u_0(\mu)) \cdot \mathbf{n} & \text{on } \partial\Omega, \\ \int_\Omega u_s = 0. \end{cases} \tag{2.8}$$

We obtain its weak formulation using the standard procedure:

Given $\mu \in \Omega$, find $u_s \in H^1(\Omega)$ such that

$$\begin{cases} \int_\Omega \sigma\nabla u_s\nabla v = -\int_\Omega \sigma_s\nabla u_0(\mu)\nabla v - \int_{\partial\Omega}(\sigma_0\nabla u_0(\mu)) \cdot \mathbf{n} \cdot v & \forall\, v \in H^1(\Omega), \\ \int_\Omega u_s = 0. \end{cases} \tag{2.9}$$

Different from (2.1), we have smoothness and boundness in the first equation of (2.8) because $\sigma_s = 0$ in a neighborhood of $\mu$. Also notice that the boundary conditions in (2.8) does not present any difficulties because $u_0$ is bounded on the boundary. Therefore, the functional in the equation (2.9),

$$-\int_\Omega \sigma_s\nabla u_0(\mu)\nabla v - \int_{\partial\Omega}(\sigma_0\nabla u_0(\mu)) \cdot \mathbf{n} \cdot v,$$

is continuous in $H^1(\Omega)$.

As the average of $u_s$ is zero, which was imposed to remove additive constants, we can use Poincaré inequality to prove the elipticity in $H^1(\Omega)$ of the bilinear form in equation (2.9). On the other hand, its continuity holds, thus by Lax-Milgram the existence and uniqueness of a solution in (2.9) can be claimed for all $\mu \in \Omega$.

To approximate this solution we use a finite element space $V_h$:

Given $\mu \in \Omega$, find $u_{s,h} \in V_h$ such that

$$\begin{cases} \int_\Omega \sigma\nabla u_{s,h}\nabla v_h = -\int_\Omega \sigma_s\nabla u_0(\mu))\nabla v_h - \int_{\partial\Omega}(\sigma_0\nabla u_0(\mu)) \cdot \mathbf{n} \cdot v_h & \forall\, v_h \in V_h, \\ \int_\Omega u_{s,h} = 0. \end{cases} \tag{2.10}$$

### 2.1.1 RB on the Subtraction Approach. A Simple Case

The subtraction approach approximates the solution of the EEG equation (2.1) with a function $u_{h,s}$ defined by the sum of the fundamental solution (2.5) and the solution of (2.10),

$$u_{h,s}(\mu) := u_0(\mu) + u_{s,h}(\mu), \qquad \mu \in \mathcal{P}.$$

This gives two ways to apply model order reduction, one to $u_{h,s}$ and the other to $u_{s,h}$.

For the former, each element of the reduced basis will be composed by a singular function plus a smooth function. Having a spatial singularity is very inconvenient because it may invalidate an efficient MOR application, so it is not recommended (see Section 2.2.1).

The second idea is to apply MOR directly to $u_{s,h}$. In this way we obtain an approximation $u_{s,N}$ which we sum to $u_0$,

$$u_{N,s}(\boldsymbol{\mu}) := u_0(\boldsymbol{\mu}) + u_{s,N}(\boldsymbol{\mu}), \qquad \boldsymbol{\mu} \in \mathcal{P}.$$

If $u_{s,N}$ is close to $u_{s,h}$, then $u_{N,s}$ will be a good approximation of $u_{h,s}$.

This section concentrates in the application of RB to $u_{s,h}$ in the most basic scenario, which is the isotropic case. We present first some results regarding the approximation of $u_s$ from elements of its solution manifold and afterwards some experiments.

Consider a compact parameter set $\mathcal{P} \subset \Omega = (0,1)^d$, $\sigma = 1$ and $\mathbf{p}_0 \in \mathbb{R}^d$, then equation (2.8) reduces to

$$\begin{cases} \operatorname{div}(\nabla u_s) = 0 & \text{in } \Omega, \\ \nabla u_s \cdot \mathbf{n} = -\nabla u_0(\boldsymbol{\mu}) \cdot \mathbf{n} & \text{on } \partial\Omega, \\ \int_\Omega u_s = 0. \end{cases} \tag{2.11}$$

The corresponding weak formulation is

$$\begin{cases} \int_\Omega \nabla u_s \nabla v = -\int_{\partial\Omega} \nabla u_0(\boldsymbol{\mu}) \cdot \mathbf{n} \cdot v & \forall\, v \in H^1(\Omega), \\ \int_\Omega u_s = 0. \end{cases} \tag{2.12}$$

We prove in two ways Kolmogorov's $n$-width estimates for the manifold

$$\mathcal{M} = \{u_s(\boldsymbol{\mu}), \boldsymbol{\mu} \in \mathcal{P}\}.$$

We begin first using the theory of Section 1.4.

**Theorem 14.** *Let $\mathcal{P}$ be a compact subset of $\Omega = (0,1)^d$, then for each $s > 0$ there exists a constant $C_s$ independent of $\boldsymbol{\mu}$ such that*

$$d_n(\mathcal{M})_{H^1(\Omega)} \leq C_s (n+1)^{-s}, \qquad \forall s \in \mathbb{N}. \tag{2.13}$$

*Proof.* For every compact set $\mathcal{P} \subset \Omega$ there is $\mathcal{P}' = [p_1, p_2]^d$ such that $\mathcal{P} \subset \mathcal{P}' \subset \Omega$, therefore proving Theorem 14 for $\mathcal{P}'$ automatically prove the result for $\mathcal{P}$. As this is the case we will consider $\mathcal{P} = \mathcal{P}'$ in the proof.

We will apply Theorem 6 to obtain (2.13). This theorem asks for a normalized affine representation of the parametric part. Notice that the differential operator in the subtraction approach is independent of $\boldsymbol{\mu}$. Therefore, we focus exclusively on

$$f(\boldsymbol{\mu}) := -\int_{\partial\Omega} \nabla u_0(\boldsymbol{\mu}) \cdot \mathbf{n} \cdot v.$$

The function $u_0$ inside the integral,

$$u_0(\mathbf{x}; \boldsymbol{\mu}) = \frac{\mathbf{p}_0 \cdot (x - \boldsymbol{\mu})}{2\pi |x - \boldsymbol{\mu}|^2}, \qquad \mathbf{x} \in \partial\Omega,\ \boldsymbol{\mu} \in \mathcal{P},$$

does not have a direct affine representation, but we can use a Chebyshev series of the type (1.72) to compute one as required by Theorem 6.

Define for each $\boldsymbol{\mu} \in \mathcal{P}$ the series

$$Su_0(\mathbf{x}, \boldsymbol{\mu}) = \sum_{j \geq 1} c_j(\boldsymbol{\mu}) T_j(\mathbf{x}), \qquad \mathbf{x} \in \partial\Omega,\ \boldsymbol{\mu} \in \mathcal{P}, \tag{2.14}$$

where $T_j$ are tensorized Chebyshev polynomials like (1.34) and $c_j(\boldsymbol{\mu})$ the corresponding coefficients of $u_0$ for each $\boldsymbol{\mu} \in \mathcal{P}$.

Since $u_0$ is an analytical function in $\partial\Omega \times \mathcal{P}$ we have convergence of (2.14) and also exponential decay of the coefficients $c_j(\boldsymbol{\mu})$ for all $\boldsymbol{\mu} \in \mathcal{P}$ from a generalization of Theorem 12 for tensorized Chebyshev polynomials [160].

For each fixed $j \in \mathbb{N}$, $c_j(\boldsymbol{\mu})$ is continuous and therefore bounded in the compact set $\mathcal{P}$. Now normalize the coefficients $c_j(\boldsymbol{\mu})$ by multiplying and dividing each term of (2.14) with $|c_j(\boldsymbol{\mu}^j_{\max})|$, where

$$\boldsymbol{\mu}^j_{\max} = \underset{\boldsymbol{\mu} \in \mathcal{P}}{\operatorname{argmax}} |c_j(\boldsymbol{\mu})|.$$

In other words, we redefine the coefficients of the series (2.14) with

$$c_j(\boldsymbol{\mu}) \sim \frac{c_j(\boldsymbol{\mu})}{|c_j(\boldsymbol{\mu}^j_{\max})|},$$

$$T_j(\mathbf{x}) \sim T_j(\mathbf{x}) |c_j(\boldsymbol{\mu}^j_{\max})|.$$

We have to prove this normalized affine representation is compatible and complete (see Section 1.4). It is trivially compatible because the Chebyshev polynomials belong to $L^2(\Omega)$ and it is complete as $|c_j(\boldsymbol{\mu}^j_{\max})|$ decay exponentially and $\max_{\mathbf{x} \in \Omega} |T_j(\mathbf{x})| = 1$. For this same reason the normalized Chebyshev polynomials satisfy $(T_j)_{j \geq 1} \in l^p(\mathbb{N}),\ \forall p < 1$. Fix for $s > 0$ a value $p$ like (1.37), $p = 1/(s+1)$.

The only thing missing to apply Theorem 6 is to extent the problem (2.12) to the complex domain in a holomorphic way.

Consider an open set $\mathcal{P}'$ such that $\mathcal{P} \subset \mathcal{P}' \subset \Omega$ and $\mathcal{P}' \cap \partial\Omega = \emptyset$. Then, define the complex set

$$\mathcal{O} := (\mathcal{P}' + i\mathcal{P}', \mathcal{P}' + i\mathcal{P}').$$

The operator

$$B : v \to \operatorname{div}(\nabla v)$$

is well defined and has meaning acting from $H^1(\Omega)$ to its dual. Using Theorem 3 we obtain that $B^{-1}$ exits and is continuous.

The map $\boldsymbol{\mu} \to B^{-1} f(\boldsymbol{\mu})$ is holomorphic in $\mathcal{O}$ as $f(\boldsymbol{\mu})$ is holomorphic and $B^{-1}$ is a linear continuous operator independent of $\boldsymbol{\mu}$. Therefore, the solution map

$$u_s(\boldsymbol{\mu}) = B^{-1} f(\boldsymbol{\mu})$$

is holomorphic in $\mathcal{O}$. Finally, applying Theorem 6 we get (2.13). □

From Theorem 14 we showed that $d_n(\mathcal{M})_{H^1(\Omega)}$ decays faster than any polynomial rate but we have to take in account that the constant $C_s$ grows as $s$ is larger (see Lemma 2).

The following result is another Kolmogorov's $n$-width bound but using a FOR estimate.

**Theorem 15.** *Let $\mathcal{P}$ be a compact subset of $\Omega$, then there exists a constant $C$ and $r > 1$ independent of $\mu$ such that*

$$d_n(\mathcal{M})_{H^1(\Omega)} \leq Cr^{-n}. \tag{2.15}$$

*Proof.* Compute a Chebyshev affine representation like in the proof of Theorem 14. Equation (2.11) satisfies all the conditions of Corollary 3 therefore there exists a constant $C$ such that

$$\|u_s(\mu) - u_{0,n}(\mu)\|_{H^1(\Omega)} \leq C\varepsilon_n(\mu), \qquad \mu \in \mathcal{P},$$

for $\varepsilon_n(\mu)$ the Chebyshev affine representation error for the first $n$ terms. As seen in Theorem 14 for all $\mu \in \mathcal{P}$ there exists $r(\mu) > 1$ such that $\varepsilon_n(\mu) \leq r^{-n}(\mu)$. Moreover, $r(\mu)$ is uniformly bounded by some value $r$ because the coefficients of the Chebyshev series are uniformly bounded. Directly from the definition of $d_n(\mathcal{M})$ we obtain

$$d_n(\mathcal{M})_{H^1(\Omega)} \leq \|u_s(\mu) - u_{0,n}(\mu)\|_{H^1(\Omega)} \leq C\varepsilon_n(\mu) \leq Cr^{-n}, \qquad \forall \mu \in \mathcal{P}.$$

□

With Theorem 14 and Theorem 15 we have two similar estimates for the subtraction approach. Clearly, FOR estimate (2.15) is better than (2.13) and also have a simpler proof.

As a quick reminder, from the definition of Kolmogorov's $n$-width,

$$d_n(\mathcal{M})_V := \inf_{\dim(V_n)=n} \sup_{v \in \mathcal{M}} \inf_{w \in V_n} \|v - w\|_V,$$

we give an interpretation to (2.15). This is, for all $n \in \mathbb{N}$ there exists a linear $n$-dimensional space $V_n$ generated by solutions of (2.11) such that for all $\mu \in \mathcal{P}$ there exists $u_n(\mu) \in V_n$ satisfying

$$\|u_s(\mu) - u_n(\mu)\|_{H^1(\Omega)} \leq Cr^{-n}.$$

Therefore, problem (2.11) for a compact parameter set $\mathcal{P}$ is reducible and MOR techniques are feasible to solve it.

We explore now some numerical results for equation (2.11) in a finite element space $V_h$ with uniform mesh of dimension $N_h = 45604$ and $h = 0.0031$. The variational formulation (2.12) in $V_h$ readily follows as

$$\begin{cases} \int_\Omega \nabla u_{s,h} \nabla v = -\int_{\partial\Omega} \nabla u_0(\mu) \cdot \mathbf{n} \cdot v & \forall\, v \in V_h(\Omega), \\ \int_\Omega u_{s,h} = 0. \end{cases} \tag{2.16}$$

In real scenarios the domain $\Omega$ is usually divided by layers with different conductivities that resembles diverse structures of the head like the scalp, skull and several other tissues of the brain. From a physiological point of view, the source of epilepsy seizures is known to be located in a middle layer close to the boundary, see Figure 2.1. Anyhow, for this first experiment we will consider a larger parameter space $\mathcal{P} = [0.28, 0.72]^2$.

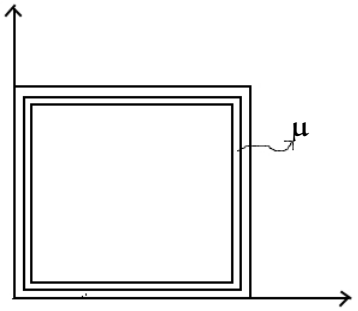


Figure 2.1: Domain $\Omega$ divided by layers.

The sampling set $\Xi_s$ was selected with an uniform discretization of $n_s = 676$ points and the POD algorithm used was PODXh. The first step to use PODXh is to compute the solution matrix

$$\mathbf{S} = \left[\mathbf{u}_{s,h}(\boldsymbol{\mu}^1)|\cdots|\mathbf{u}_{s,h}(\boldsymbol{\mu}^{n_s})\right],$$

where $\mathbf{u}_{s,h}(\boldsymbol{\mu}^j)$ is the discrete solution of (2.16) for $\boldsymbol{\mu}^j \in \Xi_s,\ j = 1,\ldots,n_s$. Later we compute

$$\widetilde{\mathbf{S}} = \mathbf{X}_h^{1/2}\mathbf{S}$$

and obtain its singular values to decide a suitable dimension. And with the last step, we build the RB space taking its first N left singular vectors (see Proposition 2).

The bilinear form in (2.16) is symmetric and $\boldsymbol{\mu}$-independent. Therefore, it is convenient to use G-RB to compute the reduced basis solutions because of the optimal result of Proposition 4 and error estimate of Corollary 1.

Figure 2.2 displays the behaviour of the singular values and estimate (1.25) for different dimensions of the reduced basis space. We can see the decay of the singular values to be fast and naturally also the corresponding error. Moreover, Figure 2.3 shows the exact errors in $\|\cdot\|_{H^1(\Omega)}$ between some truth solutions and RB solutions, it shows that with just thirty reduced basis functions we can have precise RB approximations. From the experiments only it seems we are obtaining a convergence rate of $1.9^{-N}$.

In some applications we may find the parameters near to boundary interfaces. We consider now an illustrative case, which is when $\mathcal{P} = \Omega$ and not as a compact subset of $\Omega$. Unfortunately, we cannot have good approximations from RB tecniques for all $\boldsymbol{\mu} \in \mathcal{P}$ in this case.

**Theorem 16.** *Suppose $\mathcal{P} = \Omega$ and that*

$$V_N = span\{\zeta_1,\ldots,\zeta_N\}$$

*is created by the POD method of Proposition 2. If there exists a parameter $\boldsymbol{\mu}_, \in \mathcal{P}$ such that*

$$\|u_{s,h}(\boldsymbol{\mu}_e) - u_{s,N}(\boldsymbol{\mu}_e)\|_{H^1(\Omega)} > 0, \tag{2.17}$$

*then there exists a sequence $(\boldsymbol{\mu}_k)_{k\geq 1} \subset \mathcal{P}$ that satisfies*

$$\|u_{s,h}(\boldsymbol{\mu}_k) - u_{s,N}(\boldsymbol{\mu}_k)\|_{H^1(\Omega)} \xrightarrow[k\to\infty]{} \infty. \tag{2.18}$$

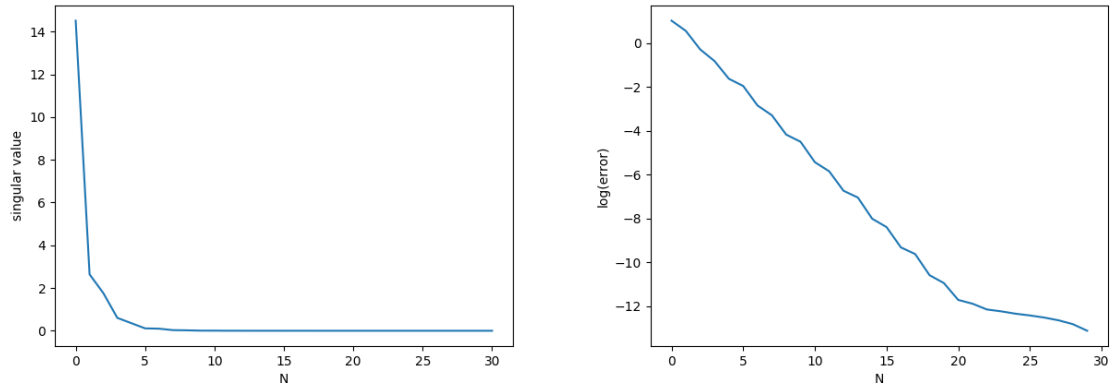


Figure 2.2: POD applied to the subtraction approach (isotropic case) in a compact parameter set. Left picture: singular values. Right picture: estimated RB error in $\Xi_s$ using (1.25).

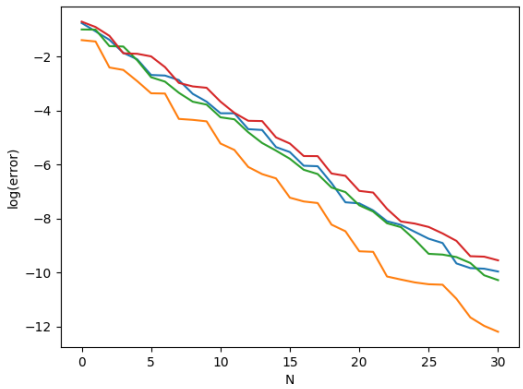


Figure 2.3: POD applied to the subtraction approach (isotropic case) in a compact parameter set. Error in $\|\cdot\|_{H^1(\Omega)}$ between four random truth solutions and RB solutions for different dimensions of the RB space.

*Proof.* To prove (2.18) we will use Theorem 7 and the notation of its proof.

The condition (2.17) implies

$$\|u_{s,h}(\boldsymbol{\mu}_e) - u_{s,N}(\boldsymbol{\mu}_e)\|_a > 0$$

because of the norm equivalence in finite dimensional spaces. Therefore, there exists $i \in \mathbb{N}$ with $N+1 \leq i \leq N_h$ such that $u_h^{(i)}(\boldsymbol{\mu}) > 0$ and consequently

$$a(u_h^{(i)}(\boldsymbol{\mu})\zeta_i^a, \zeta_i^a) = f(\zeta_i^a; \boldsymbol{\mu}) > 0. \tag{2.19}$$

Consider the right hand side of (2.12) in the finite element space,

$$f(v_h; \boldsymbol{\mu}) = \int_{\partial\Omega} \nabla u_0(\boldsymbol{\mu}) v_h \cdot \mathbf{n}. \tag{2.20}$$

The function $u_0(\boldsymbol{\mu})$ is different from zero for all $\boldsymbol{\mu} \in \Omega$, so (2.19) and (2.20) implies there exists a point $\boldsymbol{\mu}_0 = (x_0, y_0) \in \partial\Omega$ such that $\zeta_i^a(\boldsymbol{\mu}_0) \neq 0$. For simplicity consider that $x_0 = 1$, for other values the procedure is analogous, and define an arbitrary sequence $\boldsymbol{\mu}_k = (\mu_k, y_0) \in \Omega$, where $|1 - \mu_k| = \epsilon_k \to 0$ when $k \to \infty$.

Let $T_0 \in T_h$ be the triangle where $\boldsymbol{\mu}_0$ belongs, the symbols $\partial T_0$ and $\partial T_0^C$ will denote a compact interval $\partial T_0 \subset\subset T_0 \cap \partial\Omega$ where $\zeta_i^a(x) \neq 0 \ \forall x \in \partial T_0$, and its complement in $\partial\Omega$ respectively. Thus,

$$f(\zeta_i^a; \boldsymbol{\mu}_k) = \int_{\partial T_0} \nabla u_0(\boldsymbol{\mu}_k)\zeta_i^a \cdot \mathbf{n} + \int_{\partial T_0^C} \nabla u_0(\boldsymbol{\mu}_k)\zeta_i^a \cdot \mathbf{n}. \tag{2.21}$$

The integral in $\partial T_0$ reduce exactly to

$$-\frac{1}{2\pi}\int_{\partial T_0} \frac{(x-\mu_k)^2 - (y-y_0)^2}{((x-\mu_k)^2 + (y-y_0)^2)^2}\zeta_i^a,$$

which is equal to

$$-\frac{1}{2\pi}\int_{\partial T_0} \frac{(x-\mu_k)^2 - (y-y_0)^2}{((x-\mu_k)^2 + (y-y_0)^2)^2}\varphi_{T_0}.$$

The function $\varphi_{T_0}$ is a linear function obtained from the linear combination of the two basis functions in $V_h$ associated with the element $T_0$ and with support on the boundary.

Consider now $k > k_0$ such that $(y_0 - \epsilon_k^2, y_0 + \epsilon_k^2) \subset \partial T_0$ and the constants $c_{\partial T_0} = \min_{x\in\partial T_0} \varphi_{T_0}(x)$, $C_{\partial T_0} = \max_{x\in\partial T_0} |\varphi_{T_0}(x)|$, then

$$\begin{aligned}\int_{\partial T_0} \frac{(x-\mu_k)^2 - (y-y_0)^2}{((x-\mu_k)^2 + (y-y_0)^2)^2}\varphi_{T_0} &= \int_{((y_0-\epsilon_k^2, y_0+\epsilon_k^2)} \frac{(x-\mu_k)^2 - (y-y_0)^2}{((x-\mu_k)^2 + (y-y_0)^2)^2}\varphi_{T_0} \\ &\quad + \int_{((y_0-\epsilon_k^2, y_0+\epsilon_k^2)^C} \frac{(x-\mu_k)^2 - (y-y_0)^2}{((x-\mu_k)^2 + (y-y_0)^2)^2}\varphi_{T_0},\end{aligned}$$

with

$$\left|\int_{(y_0-\epsilon_k^2, y_0+\epsilon_k^2)} \frac{(x-\mu_k)^2 - (y-y_0)^2}{((x-\mu_k)^2 + (y-y_0)^2)^2}\varphi_{T_0}\right| \leq C_{\partial T_0}\int_{(y_0-\epsilon_k^2, y_0+\epsilon_k^2)} \frac{\epsilon_k^2 + \epsilon_k^4}{\epsilon_k^4} \leq C_{\partial T_0} 4,$$

and

$$\int_{((y_0-\epsilon_k^2, y_0+\epsilon_k^2)^C} \frac{(x-\mu_k)^2 - (y-y_0)^2}{((x-\mu_k)^2 + (y-y_0)^2)^2}\varphi_{T_0} \geq \frac{c_{\partial T_0}}{4\epsilon_k^4}\int_{((y_0-\epsilon_k^2, y_0+\epsilon_k^2)^C} \epsilon_k^2 - (y-y_0)^2 \xrightarrow[\epsilon_k\to 0]{} \infty.$$

In conclusion, the expression $f(\zeta_i^a; \boldsymbol{\mu}_k)$ in (2.21) goes to $\infty$ as $k \to \infty$ and therefore

$$\|u_{s,h}(\boldsymbol{\mu}_k) - u_{s,N}(\boldsymbol{\mu}_k)\|_{H^1(\Omega)} \xrightarrow[k\to\infty]{} \infty \tag{2.22}$$

applying Theorem 7. □

The condition (2.17) is easily achievable and cannot be controlled, therefore Theorem 16 exposes that for a fixed reduced basis space is not possible to get an approximation for all $\mu \in \Omega$. The reason behind it is given by the fact that $u_0(\mu)$ explodes when the parameter gets close to the boundary.

Certainly, we can always select a compact set $\mathcal{P}$ with values as close as we want to the boundary but this affects greatly the effectiveness of the RB. For example, the following Loe estimate deteriorates with closeness to the boundary.

**Theorem 17.** *Define $co(\mathcal{P})$ as the convex envelope of $\mathcal{P}$ and $q$ the distance*

$$q = dist(co(\mathcal{P}), \partial\Omega).$$

*Also suppose that the reduced basis space $V_N$ is created by the POD method of Proposition 2 with $N = r$, where $r$ is the rank of the solution matrix. Then, the Loe estimator is bounded by*

$$\Delta_N^L(\mu) \leq \frac{2\sqrt{71}(\alpha_h + \alpha_N)|\mathbf{p}_0|\sqrt{m(\partial\Omega)}d_{max}}{\alpha_h \alpha_N \pi \sigma_0 q^3}, \qquad \mu \in \mathcal{P}. \tag{2.23}$$

*Proof.* We will use Corollary 5 to obtain the estimate. The only requirement needed is a Lipschitz bound for the functional

$$f(v;\mu) = -\int_{\partial\Omega} \nabla u_0(\mu) \cdot \mathbf{n} \cdot v. \tag{2.24}$$

The gradient with respect to the spatial variables of the expression

$$u_0(\mu) = \frac{\mathbf{p}_0 \cdot (x-\mu)}{2\pi\sigma_0|x-\mu|^2}$$

is

$$\nabla u_0(\mu) = \frac{1}{2\pi\sigma_0}\left(\frac{\mathbf{p}_0}{|x-\mu|^2} - \frac{2\mathbf{p}_0(x-\mu)(x-\mu)}{|x-\mu|^4}\right).$$

From (2.24) the following holds

$$\begin{aligned}(f(\mu) - f(\mu'))^2 &= \left(\int_{\partial\Omega} (\nabla u_0(\mu) - \nabla u_0(\mu')) \cdot \mathbf{n}v\right)^2 \\ &\leq \left(\int_{\partial\Omega} |\nabla u_0(\mu) - \nabla u_0(\mu')|^2\right)\left(\int_{\partial\Omega} v^2\right).\end{aligned}$$

Define $A$ as the expression

$$A := \int_{\partial\Omega} |\nabla u_0(\mu) - \nabla u_0(\mu')|^2,$$

then

$$A \leq \frac{1}{2(\pi\sigma_0)^2}\int_{\partial\Omega}\left|\frac{\mathbf{p}_0}{|x-\mu|^2} - \frac{\mathbf{p}_0}{|x-\mu'|^2}\right|^2 + 2\left|\frac{\mathbf{p}_0(x-\mu)(x-\mu)}{|x-\mu|^4} - \frac{\mathbf{p}_0(x-\mu')(x-\mu')}{|x-\mu'|^4}\right|^2.$$

We take the two terms in the integral of A and define B and C as

$$B := \int_{\partial\Omega} \left| \frac{\mathbf{p}_0}{|x-\mu|^2} - \frac{\mathbf{p}_0}{|x-\mu'|^2} \right|^2,$$
$$C := \int_{\partial\Omega} 2\left| \frac{\mathbf{p}_0(x-\mu)(x-\mu)}{|x-\mu|^4} - \frac{\mathbf{p}_0(x-\mu')(x-\mu')}{|x-\mu'|^4} \right|^2.$$

Starting from B the following yields,

$$\begin{aligned} B &\le |\mathbf{p}_0|^2 \int_{\partial\Omega} \left( \frac{1}{|x-\mu|^2} - \frac{1}{|x-\mu'|^2} \right)^2 \\ &\le |\mathbf{p}_0|^2 \int_{\partial\Omega} \frac{\big(|x-\mu| - |x-\mu'|\big)^2 \big(|x-\mu| + |x-\mu'|\big)^2}{|x-\mu|^4 |x-\mu'|^4} \\ &\le |\mathbf{p}_0|^2 \int_{\partial\Omega} |x-\mu-x+\mu'|^2 \cdot 2\frac{|x-\mu|^2 + |x-\mu'|^2}{|x-\mu|^4 |x-\mu'|^4} \\ &\le \frac{2|\mathbf{p}_0|^2}{q^6} m(\partial\Omega) |\mu-\mu'|^2, \end{aligned}$$

Now with C we obtain

$$\begin{aligned} C &\le 2|\mathbf{p}_0|^2 \int_{\partial\Omega} \left| (x-\mu)\frac{(x-\mu)}{|x-\mu|^4} - (x-\mu)\frac{(x-\mu')}{|x-\mu'|^4} + (x-\mu)\frac{(x-\mu')}{|x-\mu'|^4} - (x-\mu')\frac{(x-\mu')}{|x-\mu'|^4} \right|^2 \\ &\le 4|\mathbf{p}_0|^2 \int_{\partial\Omega} \left( |x-\mu|^2 \left| \frac{x-\mu}{|x-\mu|^4} - \frac{x-\mu'}{|x-\mu'|^4} \right|^2 + |x-\mu-x+\mu'|^2 \left| \frac{x-\mu'}{|x-\mu'|^4} \right|^2 \right) \\ &\le 4|\mathbf{p}_0|^2 \int_{\partial\Omega} \frac{\big||x-\mu'|^4(x-\mu) - |x-\mu'|^4(x-\mu') + |x-\mu'|^4(x-\mu') - |x-\mu|^4(x-\mu')\big|^2}{|x-\mu|^6 |x-\mu'|^8} \\ &\quad + 4|\mathbf{p}_0|^2 |\mu-\mu'|^2 \frac{m(\partial\Omega)}{q^6} \\ &\le 4|\mathbf{p}_0|^2 \left( \frac{m(\partial\Omega)|\mu-\mu'|^2}{q^6} + 2\int_{\partial\Omega} \left[ \frac{|x-\mu'|^8 |\mu-\mu'|^2}{|x-\mu|^6 |x-\mu'|^8} + |x-\mu'|^2 \right. \right. \\ &\quad \left. \left. \cdot \frac{\big(|x-\mu'| - |x-\mu|\big)^2 \big(|x-\mu'|^3 + |x-\mu'|^2|x-\mu| + |x-\mu'||x-\mu|^2 + |x-\mu|^3\big)^2}{|x-\mu|^6 |x-\mu'|^8} \right] \right) \\ &\le 4|\mathbf{p}_0|^2 \left( \frac{3m(\partial\Omega)|\mu-\mu'|^2}{q^6} + 8|\mu-\mu'|^2 \int_{\partial\Omega} \frac{|x-\mu'|^6 + |x-\mu'|^4|x-\mu|^2 + |x-\mu'|^2|x-\mu|^4 + |x-\mu|^6}{|x-\mu|^6 |x-\mu'|^6} \right) \\ &\le 4|\mathbf{p}_0|^2 \left( \frac{3m(\partial\Omega)|\mu-\mu'|^2}{q^6} + 8|\mu-\mu|^2 \frac{4m(\partial\Omega)}{q^6} \right) \\ &= 4 \cdot 35 |\mu-\mu'|^2 \frac{|\mathbf{p}_0|^2}{q^6} m(\partial\Omega). \end{aligned}$$

Combining the final inequalities of B and C with A we obtain a bound of the Lipschitz constant for an arbitrary $\mathbf{p}_0$ and $\sigma_0$,

$$L_f \leq \frac{\sqrt{71}|\mathbf{p}_0|\sqrt{m(\partial\Omega)}}{\pi\sigma_0 q^3}, \tag{2.25}$$

and finally from Corollary 5 and (2.25) we conclude (2.23). □

From Proposition 9 and Theorem 17 we obtain that

$$\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_{H^1(\Omega)} \leq \frac{2\sqrt{71}(\alpha_h + \alpha_N)|\mathbf{p}_0|\sqrt{m(\partial\Omega)}d_{max}}{\alpha_h\alpha_N\pi\sigma_0 q^3}, \qquad \forall \boldsymbol{\mu} \in \mathcal{P}. \tag{2.26}$$

No element in (2.26) but one affects considerably the estimate. If the term

$$q = \text{dist}(\text{co}(\mathcal{P}), \partial\Omega)$$

goes close to zero it means that the parameters are allowed to be closer to the boundary. This makes the estimate (2.26) less precise. Also notice that we selected $N = r$ in Theorem 17 because is the maximum possible dimension we can choose from POD and in terms of approximation is the best.

In the worth cases we can always relay on the Coe estimator. The following result, which is a direct consequence of Proposition 14, allow us to use Coe and have accurate bounds of the error.

**Proposition 16.** *The residual norm $r_{h,2}$ associated with any reduced basis space of problem (2.11) on a compact $\mathcal{P}$ is analytic.*

We explore now some numerical results with $\mathcal{P} = [0.01, 0.99]^2$, here the parameters are fairly close to the boundary. The sampling set $\Xi_s$ was selected with an uniform discretization of $n_s = 2500$ points and the POD algorithm was PODXh.

Figure 2.4 displays the behaviour of the singular values and estimate (1.25) for different dimensions of the reduced basis space. It exposes that we can achieve good RB approximations of the sampling set for N close to the numerical rank $r = 569$. Moreover, Figure 2.5 shows the exact errors in $\|\cdot\|_{H^1(\Omega)}$ between some truth solutions and RB solutions. From the experiments only it seems we are obtaining an approximate convergence rate of $1.013^{-N}$.

Taking Figure 2.4 and Figure 2.5 as a reference we can conclude that the RB method is working well but not as good as the previous simulation. As predicted in Theorem 15, we have a relatively fast decay in the errors computed even if the dimension is not quite small as expected. This is a consequence of Theorem 16, the closer the parameters are to $\partial\Omega$ the more reduced basis functions are needed to supplement the blow-up of

$$u_0(\mathbf{x}; \boldsymbol{\mu}) = \frac{\mathbf{p}_0 \cdot (x - \boldsymbol{\mu})}{2\pi|x - \boldsymbol{\mu}|^2}, \qquad \mathbf{x} \in \partial\Omega,\ \boldsymbol{\mu} \in \mathcal{P},$$

near the boundary. For example, we can notice a much worse performance for $\boldsymbol{\mu} = (0.9, 0.9)$ in Figure 2.5, it has an acceptable error value but it does not perform as the other cases.

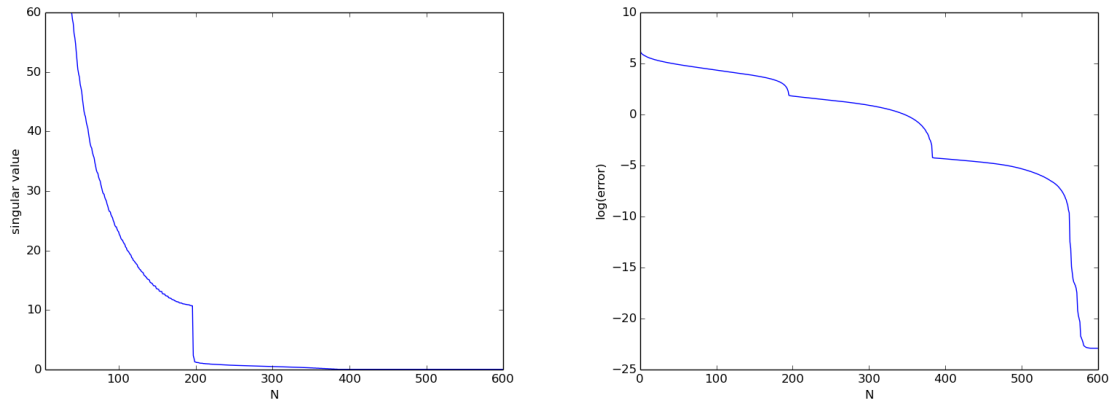


Figure 2.4: POD applied to the subtraction approach (isotropic case) for $\mathcal{P} \approx \Omega$. Left picture: singular values. Right picture: estimated RB error in $\Xi_s$ using (1.25).

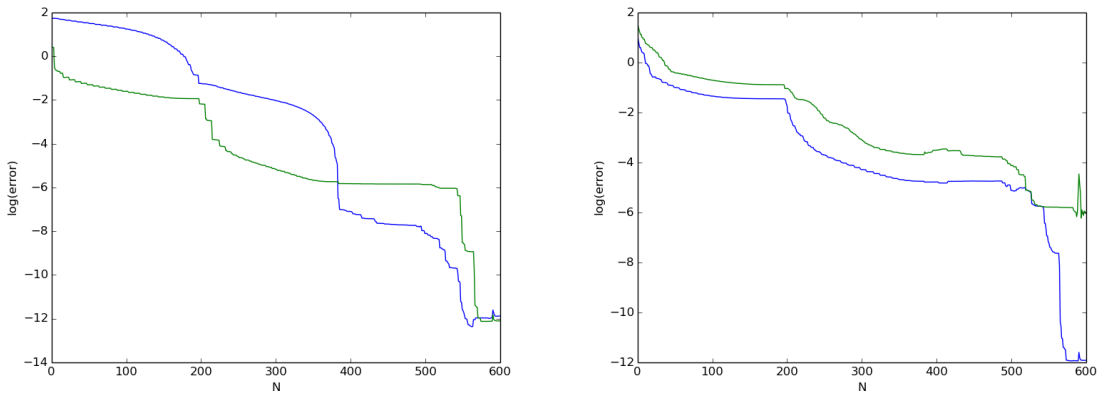


Figure 2.5: POD applied to the subtraction approach (isotropic case) for $\mathcal{P} \approx \Omega$. Error in $\|\cdot\|_{H^1(\Omega)}$ between truth solutions and RB solutions for different dimensions of the RB space. Left image: blue line for $\boldsymbol{\mu} = (0.01, 0.01)$ and green line $\boldsymbol{\mu} = (0.51, 0.51)$, both $\boldsymbol{\mu} \in \Xi_s$. Right image: blue line for $\boldsymbol{\mu} = (0.653, 0.2334)$ and green line $\boldsymbol{\mu} = (0.9, 0.9)$, both $\boldsymbol{\mu} \notin \Xi_s$.

The blow-up affects directly the functional expression

$$f(v; \boldsymbol{\mu}) = -\int_{\partial\Omega} \nabla u_0(\boldsymbol{\mu}) \cdot \mathbf{n} \cdot v, \tag{2.27}$$

and this influence the solution introducing a strong singularity on it, see Figure 2.6.

Notice also that the error for $\boldsymbol{\mu} = (0.01, 0.01)$ is much better than for $\boldsymbol{\mu} = (0.9, 0.9)$ even if it is closer to the boundary. This does not contradict the ideas explained before because $(0.01, 0.01) \in \Xi_s$ and the RB approximations in the sampling set are directly controlled by Proposition 4 and the singular values.

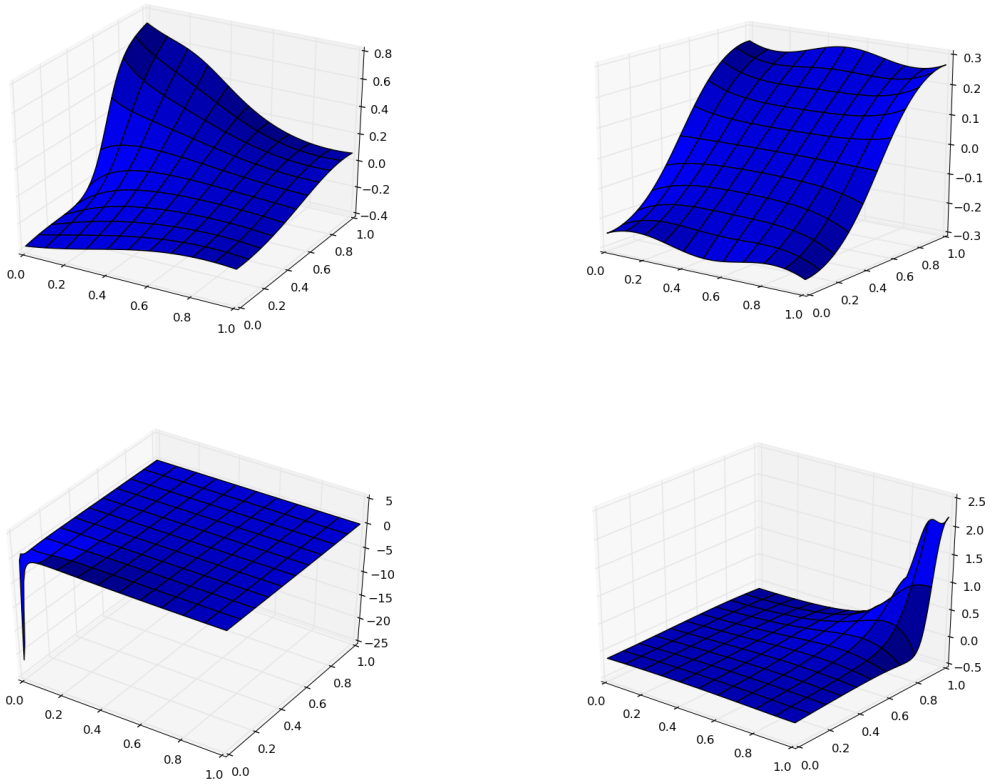

Figure 2.6: Subtraction approach for the isotropic case. Top from left to right: truth solutions for $\boldsymbol{\mu} = (0.653, 0.2334), \boldsymbol{\mu} = (0.51, 0.51)$. Bottom from left to right: truth solutions for $\boldsymbol{\mu} = (0.01, 0.01), \boldsymbol{\mu} = (0.9, 0.9)$.

We end this section with the certification of the RB solutions using offline estimators. We refer to the results proved in Section 1.6.

It is a common practice to always compute a-posteriori error estimates after every RB online computation to guarantee certain level of precision. In this EEG problem with $\mathcal{P} \approx \Omega$ we should check reliability of the RB solutions but we would like not to add extra time to the online stage, therefore offline estimators are a solution.

Note that in this particular problem the Loe estimate (2.26) does not give any valuable information of the error as it is very pessimistic, its actual bound for this experiment was $\Delta_N^L(\boldsymbol{\mu}) \leq 8582.80$. A better option is to use Coe.

The interpolation of the residual with Chebyshev polynomials is very convenient because is a one time procedure and can be executed at the offline stage to calculate the maximum value of the residual. From Proposition 16 we have a residual norm which is analytic, therefore with few evaluations of the residual we can construct an approximation that can achieve good precision.

We used the Chebfun2 package [157] for the interpolation and around the ten minutes mark Chebfun2 achieved machine epsilon precision of the residual. After computing the maximum of the polynomial and using Proposition 13 we can guarantee an error between truth solutions and RB solutions of $0.00068$ for all $\mu \in [0.01, 0.99]^2$ if we take $N = r$.

### 2.1.2 RB and FOR on the Subtraction Approach. A More Realistic Case

This section builds upon the results of the previous section to construct a good reduced model of the subtraction approach in a layered domain.

The original equation to solve is

$$\begin{cases} \operatorname{div}(\sigma \nabla u_s) = -\operatorname{div}(\sigma_s \nabla u_0(\mu)) & \text{in } \Omega, \\ (\sigma \nabla u_s) \cdot \mathbf{n} = -(\sigma \nabla u_0(\mu)) \cdot \mathbf{n} & \text{on } \partial\Omega, \\ \int_\Omega u_s = 0, \end{cases} \tag{2.28}$$

which has the variational formulation

$$\begin{cases} \int_\Omega \sigma \nabla u_s \nabla v = -\int_\Omega \sigma_s \nabla u_0(\mu) \nabla v - \int_{\partial\Omega} (\sigma_0 \nabla u_0(\mu)) \cdot \mathbf{n} \cdot v \qquad \forall\, v \in H^1(\Omega), \\ \int_\Omega u_s = 0. \end{cases} \tag{2.29}$$

In $\Omega$ consider the layers $\Omega_i \subset \Omega$, $i = 0, 1, 2$, with $\Omega_i \cap \Omega_j = \emptyset$ for $i \neq j$ like for example in Figure 2.7. Also, assume a constant conductivity $\sigma_i$ in each layer $\Omega_i$ and define in $\Omega_0$ a compact parameter set $\mathcal{P} \subset \Omega_0$.

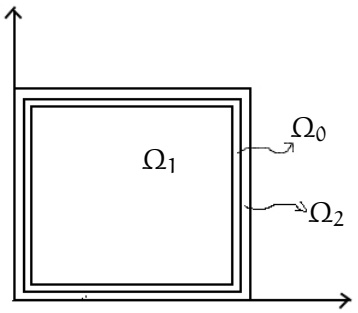


Figure 2.7: Subdomains of $\Omega$: $\Omega_0, \Omega_1, \Omega_2$.

A direct consequence of Corollary 3 gives an Kolmogorov $n$-width for the general equation (2.28).

**Theorem 18.** *If there exists a basis of functions $\{g_1, g_2, \ldots\}$ defined in $\Omega$ such that for all $\mu \in \mathcal{P}$ there exists a sequence $(\alpha_k(\mu))_{k \geq 1}$ with*

$$\sup_{\mu \in \mathcal{P}} \left\| u_0(\mu) - \sum_{k=1}^{n} \alpha_k(\mu) g_k \right\|_{L^\infty(\Omega)} \leq \varepsilon_n.$$

*Then there exists a constant $C$ independent of $\mu$ such that*

$$d_n(\mathcal{M})_{H^1(\Omega)} \leq C \varepsilon_n. \tag{2.30}$$

As explained in Section 2.1, the right hand side of (2.29) is composed by smooth functions, therefore we can obtain the following result as a consequence of Proposition 14.

**Proposition 17.** *The residual norm $r_{h,2}$ associated with any reduced basis space of problem (2.28) on a compact $\mathcal{P}$ is analytic.*

We show now some experiments. We will define a 2D configuration close to real applications and denote it as *Example* $A_\delta$. Assume the following $\boldsymbol{\sigma}_0$ which resembles the conductivity in the different layers of the brain,

$$\boldsymbol{\sigma}_0(\mathbf{x}) = \begin{cases} 0.0042 & \text{if } \mathbf{x} \in \Omega_2, \\ 0.0042 & \text{if } \mathbf{x} \in \Omega_1, \\ 0.33 & \text{if } \mathbf{x} \in \Omega_0, \end{cases} \tag{2.31}$$

with the domains similar to Figure 2.7,

$$\begin{aligned}
\Omega &:= [0,1]^2, \\
\Omega_1 &:= [0.15, 0.85]^2, \\
\Omega_2 &:= \Big( \big([0,0.1] \cup [0.9,1]\big) \times [0,1] \Big) \cup \Big( [0,1] \times \big([0,0.1] \cup [0.9,1]\big) \Big), \\
\Omega_0 &:= \Omega / (\Omega_2 \cup \Omega_1), \\
\mathcal{P}_\delta &:= \Big( \big([0.1+\delta, 0.15-\delta] \cup [0.85+\delta, 0.9-\delta]\big) \times [0.1+\delta, 0.9-\delta] \Big) \cup \\
&\qquad \Big( [0.1+\delta, 0.9-\delta] \times \big([0.1+\delta, 0.15-\delta] \cup [0.85+\delta, 0.9-\delta]\big) \Big), \qquad 0 < \delta < 0.025.
\end{aligned}$$

Following the construction of the Chebyshev series in Theorem 14 and using Theorem 18 we can obtain a Kolmogorov $n$-width bound for Example $A_\delta$.

**Proposition 18.** *Assume the setting of Example $A_\delta$. Then there exists a constant $C_\delta$ and $r_\delta > 1$ independent of $\boldsymbol{\mu}$ such that*

$$d_n(\mathcal{M}_\delta)_{H^1(\Omega)} \leq C_\delta r_\delta^{-n}. \tag{2.32}$$

We are considering as in Section 2.1.1 a compact set for the parameter set. The blow-up of $u_0$ in Theorem 16 still applies in general if $\mathcal{P} \approx \Omega_0$ as it affects the integral

$$\int_\Omega \boldsymbol{\sigma}_s \nabla u_0(\boldsymbol{\mu}) \nabla v$$

in (2.29) when $\boldsymbol{\mu}$ is close to the interfaces. Nevertheless, a possible extension of Theorem 16 depends on the configuration on the finite element space and the complexity of the layers.

In Example $A_\delta$ we have a very thin layer for the parameters and they are close to $\partial\Omega_0$. Therefore, we need to apply a good strategy for the selection of the snapshots to apply reduced basis.

A starting point is to find the parameters in $\mathcal{P}$ that are more meaningful. The solution of the equation (2.29) is completely dependent on how $u_0$ behaves with respect to the parameters

so an interesting idea would be to find the parameters that give the most important information of $u_0$ in $\mathcal{P}$.

The EIM algorithm described in Section 1.7 is a suitable choice for this. It picks different parameters in a greedy way to capture linear dependency. Also we can take in account Proposition 15 which implies that we can build a good RB space with the selected parameters from the EIM algorithm.

We fixed $\delta = 0.009$ for the experiments shown hereafter. Figure 2.8 shows the selected parameters among a very dense sampling to obtain an EIM approximation of $u_0$ with error $10^{-10}$. The right picture of Figure 2.8 shows that the most meaningful parameters are actually in $\partial\mathcal{P}_\delta$. This is expected for two reasons: (1) the closer $u_0$ is to the interface the more severe is the change of structure in $u_0$ and (2) the layer $\Omega_0$ is very thin and it may create redundancy in $u_0$ for parameters close to each other.

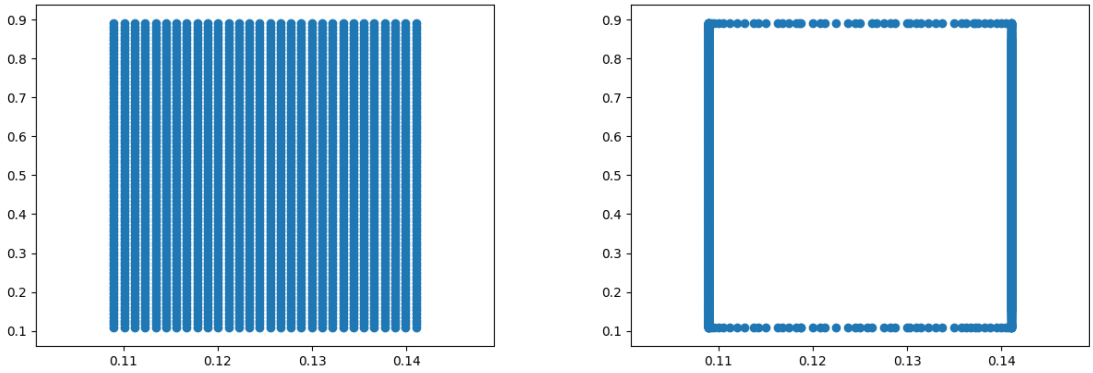


Figure 2.8: EIM applied to the subtraction approach (anisotropic case) in a compact parameter set. The pictures show only a part of $\mathcal{P}_\delta$ to improve the presentation of the parameters. Left picture: EIM sampling set for $u_0$. Right picture: Selected parameters by EIM for $u_0$.

Let us compare now four sampling schemes for POD presented in Figure 2.9: uniform, random, EIM selection and EIM selection + uniform. The quantity of parameters in the uniform and random sampling is $n_s = 2500$ while the EIM selection is $n_s = 1100$.

Figure 2.10 shows the decay of the singular values and the error estimate (1.25) for the elements in $\Xi_s$. The uniform and random schemes seems to be better fit than both EIM samplings but the exact error computations in Figure 2.11 shows clearly the opposite. This is major point of consideration because here faster decay does not imply better convergence. A reason could be to the higher redundancy in the selection of $\Xi_s$, for example, from $n_s = 2500$ the rank of the solution matrix of the random sampling is $r = 750$ in comparison to the rank complete EIM selection of $r = 1100$.

Also notice that putting together the uniform sampling with the EIM selection just add a negligible improvement in the approximation. Moreover, the two sampling versions of the EIM selection had a very close Coe estimation of $10^{-4}$.

We can exploit even more the EIM approximation computed before with the FOR method.

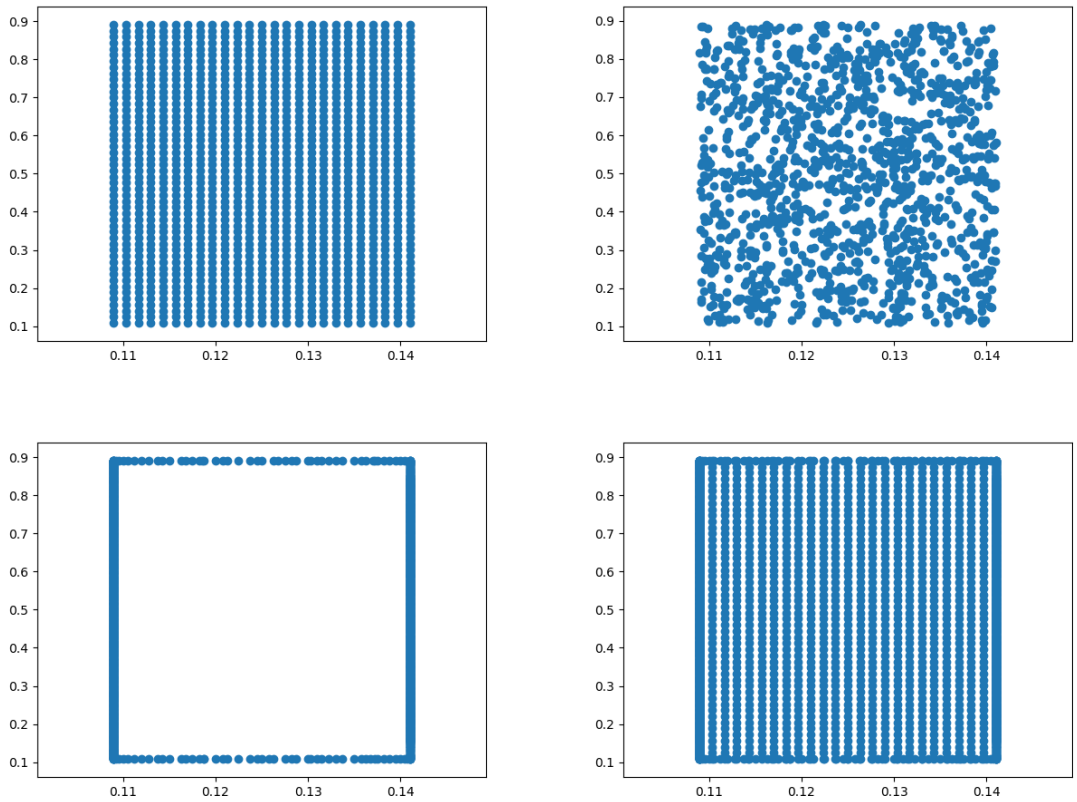

Figure 2.9: POD applied to the subtraction approach (anisotropic case) in a compact parameter set. The pictures show only a part of $\mathcal{P}_\delta$ to improve the presentation of the parameters. Top from left to right: uniform and random sampling. Bottom from left to right: EIM selection and EIM selection + uniform.

The main advantage is the $\boldsymbol{\mu}$-independence of the differential operator which we can use in Corollary 2.

Let $u_0$ has the following affine decomposition computed with EIM

$$u_0(\mathbf{x},\boldsymbol{\mu}) = \sum_{q=1}^{Q} \Theta_q(\boldsymbol{\mu}) g_q(\mathbf{x}), \qquad \mathbf{x} \in \Omega_1 \cup \Omega_3, \boldsymbol{\mu} \in \mathcal{P}_\delta,$$

then the functional in (2.29) becomes

$$-\int_\Omega \sigma_s \nabla u_0(\boldsymbol{\mu}) \nabla v - \int_{\partial\Omega} (\sigma_0 \nabla u_0(\boldsymbol{\mu})) \cdot \mathbf{n} \cdot v = \sum_{q=1}^{Q} \Theta_q(\boldsymbol{\mu}) \left( -\int_\Omega \sigma_s \nabla g_q(\mathbf{x}) \nabla v - \int_{\partial\Omega} (\sigma_0 \nabla g_q(\mathbf{x})) \cdot \mathbf{n} \cdot v \right).$$

We have then all the conditions of Corollary 2. The general computational procedure is the following:

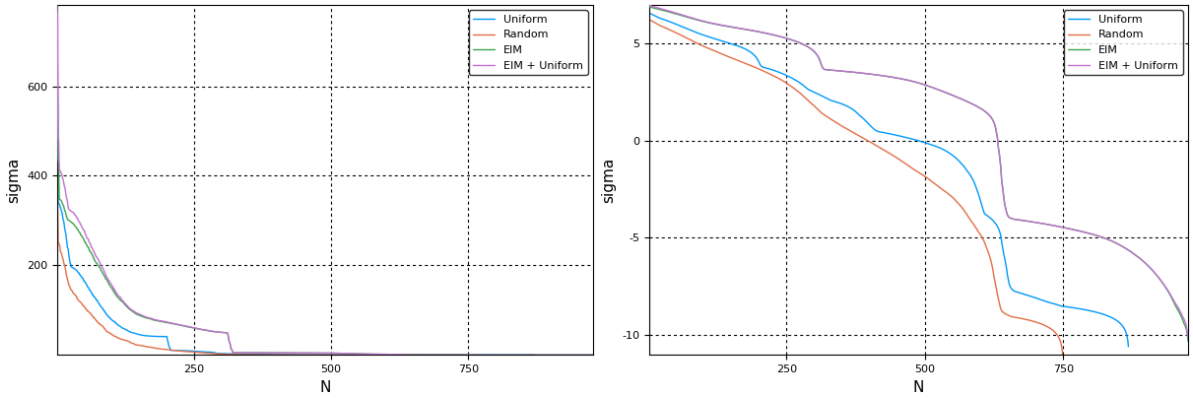


Figure 2.10: POD applied to the subtraction approach (anisotropic case) in a compact parameter set. Left picture: decay of the singular values. Right picture: estimated error in $\Xi_s$ using (1.25).

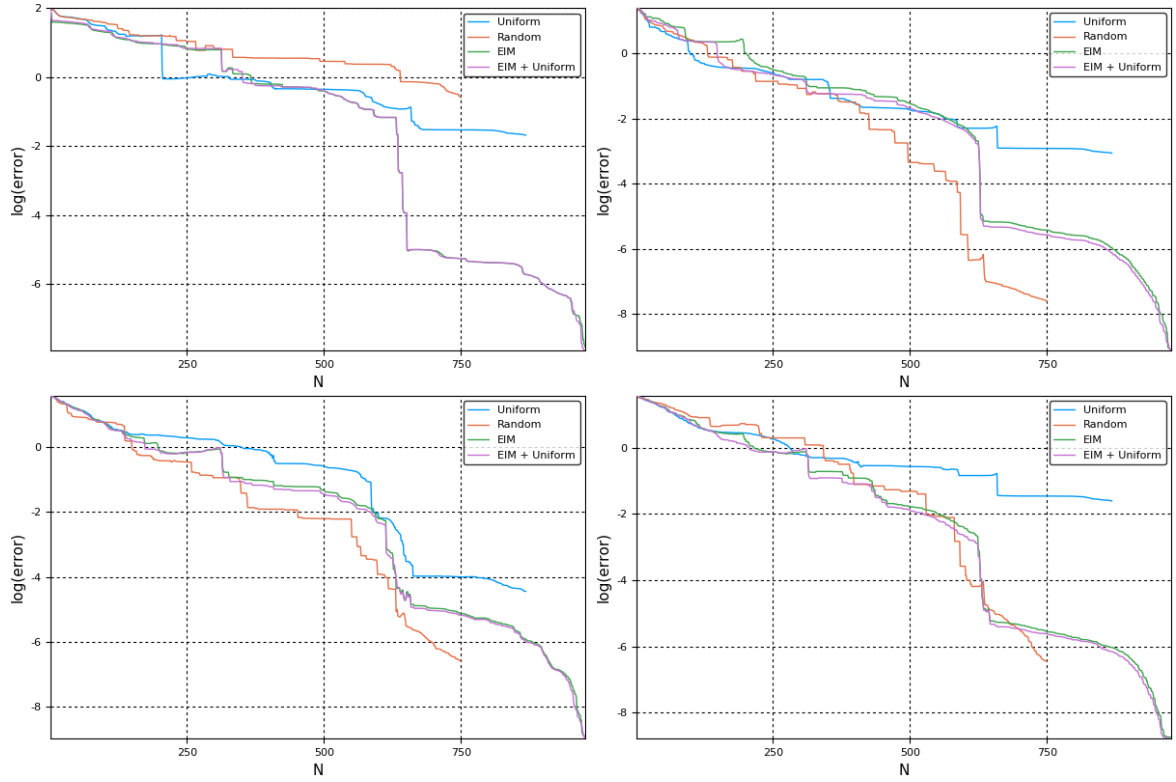


Figure 2.11: POD applied to the subtraction approach (anisotropic case) in a compact parameter set. Top from left to right: Exact error for $\boldsymbol{\mu} = (0.1091, 0.1091)$ and $\boldsymbol{\mu} = (0.125, 0.5)$. Bottom from left to right: Exact error for $\boldsymbol{\mu} = (0.135, 0.8)$ and $\boldsymbol{\mu} = (0.115, 0.3)$.

1. Compute all the vectors

$$\mathbf{g}_q := -\int_\Omega \boldsymbol{\sigma}_s \nabla g_q(\mathbf{x}) \nabla v_h - \int_{\partial\Omega} (\boldsymbol{\sigma}_0 \nabla g_q(\mathbf{x})) \cdot \mathbf{n} \cdot v_h, \qquad q = 1, \dots, Q.$$

2. Compute the solutions $\mathbf{f}_q$,

$$\mathbf{A}_h \mathbf{f}_q = \mathbf{g}_q, \qquad q = 1, \dots, Q.$$

where $\mathbf{A}_h$ is the matrix associated with the bilinear form in (2.29).

3. For any $\boldsymbol{\mu} \in \mathcal{P}$ compute an approximated solution of (2.28) as

$$\mathbf{u}_Q(\boldsymbol{\mu}) = \sum_{q=1}^{Q} \Theta_q(\boldsymbol{\mu}) \mathbf{f}_q. \tag{2.33}$$

The solutions $\mathbf{u}_Q(\boldsymbol{\mu})$ are generated by the *fundamental basis* $\{\mathbf{f}_q, q = 1, \dots, Q\}$ which are computed through the offline stage of Steps 1 and 2. Finally, the online stage is done in Step 3.

A good approximation from $\mathbf{u}_Q$ depends basically on a precise approximation of $u_0$ with EIM,

$$\begin{aligned}
\|\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{u}_Q(\boldsymbol{\mu})\|_2 &= \|\mathbf{A}_h^{-1}\mathbf{f}(\boldsymbol{\mu}) - \sum_{q=1}^{Q} \Theta_q(\boldsymbol{\mu})\mathbf{F}_q\|_2 \\
&= \|\mathbf{A}_h^{-1}\mathbf{f}(\boldsymbol{\mu}) - \mathbf{A}_h^{-1}\sum_{q=1}^{Q} \Theta_q(\boldsymbol{\mu})\mathbf{g}_q\|_2 \\
&\leq \|\mathbf{A}_h^{-1}\|_2 \|\mathbf{f}(\boldsymbol{\mu}) - \sum_{q=1}^{Q} \Theta_q(\boldsymbol{\mu})\mathbf{g}_q\|_2 \\
&= \frac{1}{\sigma_{\min}(\mathbf{A}_h)} \|\mathbf{e}_f(\boldsymbol{\mu})\|_2,
\end{aligned}$$

where $\sigma_{\min}$ denotes the smallest singular value of $\mathbf{A}_h$ and $\mathbf{e}_f$ the EIM approximation error. Analogously there is a bound for the $\mathbf{X}_h$ norm,

$$\begin{aligned}
\|\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{u}_Q(\boldsymbol{\mu})\|_{\mathbf{X}_h} &= \|\mathbf{X}_h^{1/2}(\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{u}_Q(\boldsymbol{\mu}))\|_2 \\
&= \|\mathbf{X}_h^{1/2}\mathbf{A}_h^{-1}(\mathbf{f}(\boldsymbol{\mu}) - \sum_{q=1}^{Q} \Theta_q(\boldsymbol{\mu})\mathbf{g}_q)\|_2 \\
&\leq \|\mathbf{X}_h^{1/2}\mathbf{A}_h^{-1}\|_2 \|\mathbf{f}(\boldsymbol{\mu}) - \sum_{q=1}^{Q} \Theta_q(\boldsymbol{\mu})\mathbf{g}_q\|_2 \\
&= \frac{1}{\sigma_{\min}(\mathbf{A}_h\mathbf{X}_h^{-1/2})} \|\mathbf{e}_f(\boldsymbol{\mu})\|_2.
\end{aligned}$$

Figure 2.12 and Figure 2.13 shows a comparison between the RB and FOR method, which shows that FOR is faster and more precise than RB. The computation time of the RB method is very slow, this is due to the complexity of the functional in (2.29) with the structure of the layers in the domain. Nevertheless, it can always be combined with EIM to reduce the computation time, in any case RB is slower than FOR.

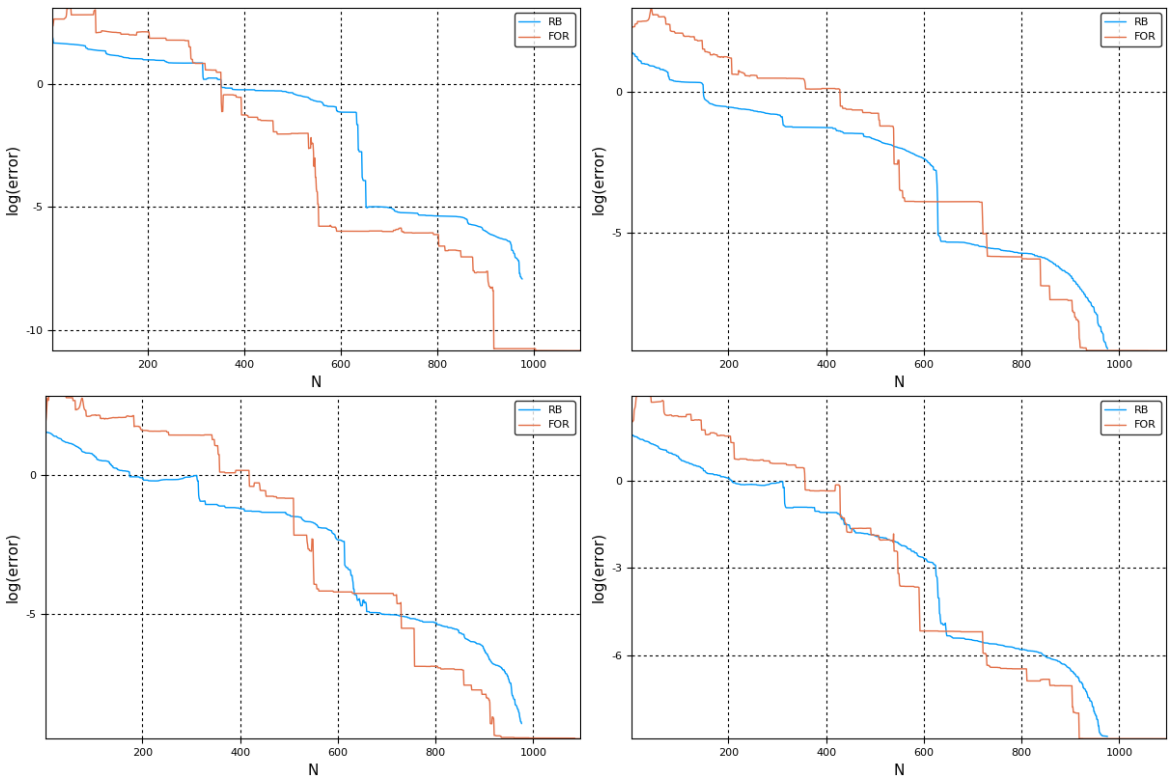


Figure 2.12: POD applied to the subtraction approach (anisotropic case) in a compact parameter set. Top from left to right: Exact errors for $\boldsymbol{\mu} = (0.1091, 0.1091)$ and $\boldsymbol{\mu} = (0.125, 0.5)$. Bottom from left to right: Exact errors for $\boldsymbol{\mu} = (0.135, 0.8)$ and $\boldsymbol{\mu} = (0.115, 0.3)$.

| | average execution time | average exact error |
|---|---|---|
| FEM | 6.70 sec | – |
| RB offline | 4.65 hr | – |
| RB online | 5.76 sec | $10^{-3}$ |
| RB + EIM offline | 3.17 hr | – |
| RB + EIM online | 0.15 sec | $10^{-6}$ |
| FOR offline | 3.17 hr | – |
| FOR online | 0.11 sec | $10^{-7}$ |

Figure 2.13: MOR applied to the subtraction approach (anisotropic case) in a compact parameter set. Comparison table between FEM, RB and FOR. RB offline and online correspond to the RB space generated from the random sampling set. RB + EIM use EIM decomposition for faster computation and EIM parameter selection as POD sampling set.

## 2.2 Direct Approach Formulation

For the direct approach we need to consider a specific trial space. Let $X$ be this trial space,

$$X := \left\{ v \in H^1(\Omega) : v \in C^1(B_{r_*}(\boldsymbol{\mu})),\ \operatorname{div}(\boldsymbol{\sigma}\nabla v) \in L^{p'}(\Omega),\ (\boldsymbol{\sigma}\nabla v)\cdot \mathbf{n} = 0 \text{ on } \partial\Omega \right\},$$

where $0 < r_* < r_0$ is a fixed number, $H^1(\Omega)$ is the Sobolev space with functions in $L^2(\Omega)$ that have first distributional derivative in $L^2(\Omega)$ and $p' \in \mathbb{R}$ (specified later). Then multiplying the first equation of (2.1) by $v \in X$ and integrating by parts yields

$$\begin{aligned}\int_\Omega \operatorname{div}(\boldsymbol{\sigma}\nabla u)v &= -\int_\Omega (\boldsymbol{\sigma}\nabla u)\cdot\nabla v + \cancel{\int_{\partial\Omega} (\boldsymbol{\sigma}\nabla u)\cdot \mathbf{n}\, v} \\ &= \int_\Omega u \operatorname{div}(\boldsymbol{\sigma}\nabla v) - \cancel{\int_{\partial\Omega} u\,(\boldsymbol{\sigma}\nabla v)\cdot \mathbf{n}} \\ &= \int_\Omega u \operatorname{div}(\boldsymbol{\sigma}\nabla v). \end{aligned} \tag{2.34}$$

The element $\operatorname{div}(\boldsymbol{\sigma}\nabla v)$ belongs to $L^{p'}(\Omega)$, thus the integral (2.34) makes sense for $u \in L^p(\Omega)$, the dual space of $L^{p'}(\Omega)$. On the other side of (2.1)

$$\int_\Omega \operatorname{div}(\mathbf{p}_0\delta_{\boldsymbol{\mu}})v = -\int_\Omega \mathbf{p}_0\cdot\nabla v\delta_{\boldsymbol{\mu}} = -\mathbf{p}_0\cdot\nabla v(\boldsymbol{\mu}). \tag{2.35}$$

Therefore, from (2.34) and (2.35) the weak formulation of (2.1) reads as:

Given $\boldsymbol{\mu} \in \Omega$, find $u \in L^p(\Omega)$ such that

$$\begin{cases} \int_\Omega u \operatorname{div}(\boldsymbol{\sigma}\nabla v) = -\mathbf{p}_0\cdot\nabla v(\boldsymbol{\mu}) \qquad \forall v \in X, \\ \int_\Omega u = 0. \end{cases} \tag{2.36}$$

The integral $\int_\Omega u = 0$ in (2.36) removes additive constants from the solution. Also notice that we cannot apply Lax-Milgram Theorem to (2.36) because the functional $-\mathbf{p}_0\cdot\nabla v(\boldsymbol{\mu})$ is unbounded in $X$.

Nevertheless, a technique called duality method gives existence and uniqueness of this equation.

**Theorem 19.** *There exists a unique solution $u \in L^p(\Omega)$ to (2.36) with range of $p$ between $1 \le p < \frac{3}{2}$ for $n = 3$ and $1 \le p < 2$ for $n = 2$.*

*Proof.* See [163]. □

We can approximate the solution of (2.36) using finite element methods. Before formulating this result we present some preliminary definitions.

Consider $P_1$ as the space of linear polynomials and consider a regular family of triangular meshes $T_h$ of $\Omega$ (see [52, 132]), where $h$ denotes the mesh size

$$h := \max_{T\in T_h} h_T,$$

with $h_T$ the diameter of T. We define then the Lagrange finite elements space of linear polynomials

$$V_h := \{v_h \in C(\Omega) : v_h|_T \in P_1 \ \forall T \in T_h\}. \tag{2.37}$$

In $V_h$ we state the finite element approximation:

Given $\boldsymbol{\mu} \in \Omega$, find $u_h \in V_h$ such that

$$\begin{cases} \int_\Omega \boldsymbol{\sigma} \nabla u_h \cdot \nabla v_h = -\mathbf{p}_0 \cdot \nabla v_h(\boldsymbol{\mu}) & \forall v_h \in V_h, \\ \int_\Omega u_h = 0. \end{cases} \tag{2.38}$$

**Theorem 20.** *Let $T_h$ be a quasiuniform family of subdivisions of a convex Lipschitz domain $\Omega$ and assume that the matrix $\boldsymbol{\sigma} \in C^1(\Omega)$. Let $u$ and $u_h$ be the respective solutions of (2.36) and (2.38). Then, there exists $h_0 > 0$ and $q_0 > 2$ such that*

$$\|u - u_h\|_{L^p(\Omega)} \leq C h^{2/p-1}$$

*for all $0 < h < h_0$ and for all $p$ such that $\frac{q_0}{q_0-1} < p < 2$. Moreover, for $1 \leq p \leq \frac{q_0}{q_0-1}$ there holds*

$$\|u - u_h\|_{L^p(\Omega)} \leq C h^s$$

*for all $0 < h < h_0$ and for all $s$ with $0 < s < 1 - \frac{2}{q_0}$.*

*Proof.* See [8]. □

### 2.2.1 RB on the Direct Approach

Model order reduction on the direct approach is a delicate task because we are not solving a standard variational problem. This section explores the application of RB to the direct approach.

Consider the parameter set $\mathcal{P} = [0.28, 0.72]^2$ without layers as in Section 2.1.1. The first step to apply POD is to discretize the selected parameter space. For the experiments presented hereafter we constructed the sampling set $\Xi_s$ with $n_s = 2500$ points uniformly distributed in $\mathcal{P}$. Following this, the next step is to build a reduced basis space $V_N$.

We build the solution matrix $\mathbf{S}$ as stated in (1.16), i.e. the columns of $\mathbf{S}$ will have the coefficients of the solutions of the FEM problem, that in the direct approach are the solutions of the variational problem

$$\begin{cases} \int_\Omega u \operatorname{div}(\boldsymbol{\sigma} \nabla v) = -\mathbf{p} \cdot \nabla v(\boldsymbol{\mu}) & \forall v \in X, \\ \int_\Omega u = 0. \end{cases} \tag{2.39}$$

Note that the results in Chapter 1 were built on Hilbert spaces and Theorem 19 shows that the solutions of (2.39) are in $L^p(\Omega)$ with $p < 2$ which is not a Hilbert space. Therefore, even if it is possible to apply POD regardless of the space we cannot use the error estimates presented in Chapter 1.

The only a-priori information that is usable in this case are the projection estimates of the procedure POD2 or PODXh according to Proposition 1 for the norm $\|\cdot\|_2$ or Proposition 2 for $\|\cdot\|_{\mathbf{X}_h}$. Figure 2.14 and Figure 2.15 displays the behaviour of the singular values of both

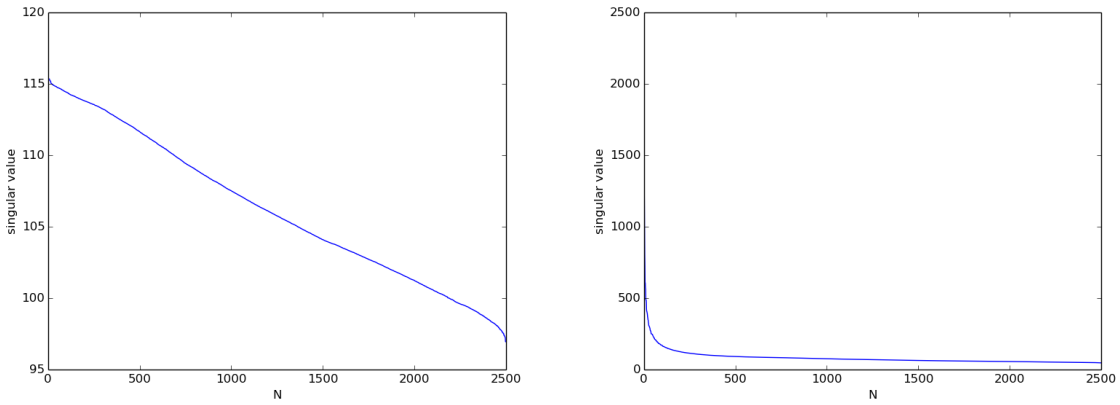


Figure 2.14: POD applied to the direct approach (isotropic case) in a compact parameter set. Left picture: singular values for PODXh. Right picture: singular values for POD2.

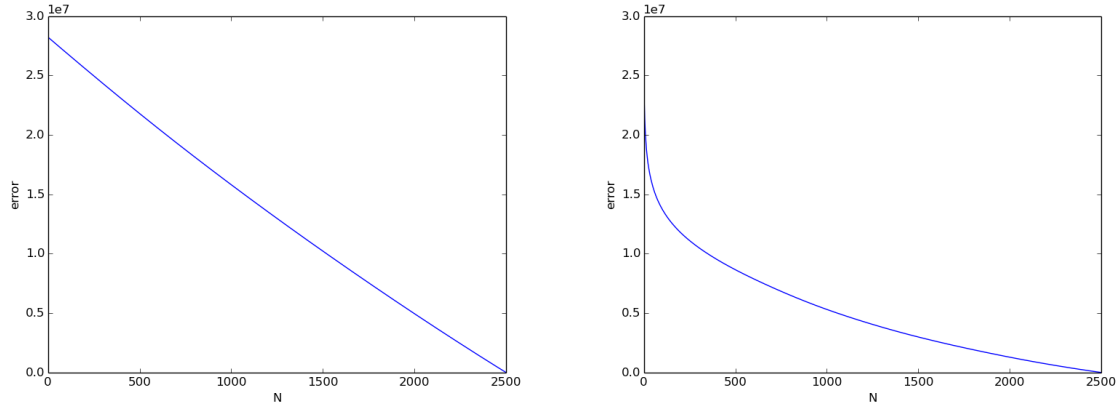


Figure 2.15: POD applied to the direct approach (isotropic case) in a compact parameter set. Left picture: projection error estimate for PODXh in $\Xi_s$. Right picture: projection error estimate for POD2 in $\Xi_s$.

procedures and their respective estimates (1.13) and (1.14) for different dimensions of the RB space.

Figure 2.14 shows that applying reduced basis is not advised, the singular values do not decay to zero and the projection errors in Figure 2.15 are far from zero if we do not take $N = n_s$. As a more concrete example, Figure 2.16 shows the exact $l_2$ errors between truth solutions and RB solutions for two parameters in $\Xi_s$ at different dimensions of the RB space. As we can notice, the error only improves when $N$ is close to 2500 which is not very good.

Moreover, for $\mu \notin \Xi_s$ the error for some parameters do not pass beyond $10^{1.903}/\text{approx}80$ as shown in Figure 2.17. The explanation of the difficulty to approximate well the truth solution seems to come from the singularity of the solution. For example, Figure 2.18 shows two

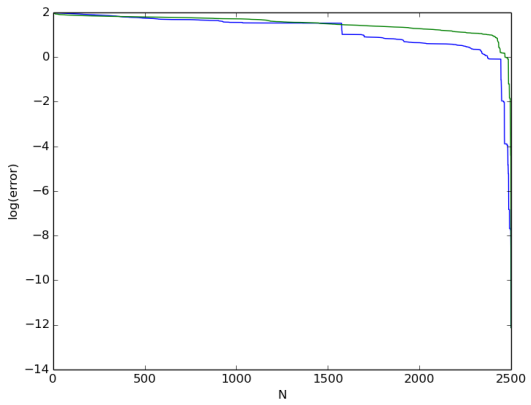


Figure 2.16: POD applied to the direct approach (isotropic case) in a compact parameter set. Exact $l_2$ error between FEM and RB for two random $\boldsymbol{\mu} \in \Xi_s$.

truth solutions which visually exhibit their linear uncorrelation coming from the singularity. Furthermore, the singularities seem to imply that the reduced basis functions have a strong tendency to become chaotic and irregular, as they are a linear combination of the snapshots, see Figure 2.19.

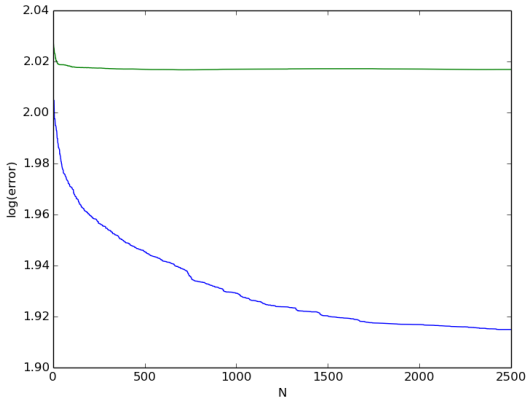


Figure 2.17: POD applied to the direct approach (isotropic case) in a compact parameter set. Exact $l_2$ error between FEM and RB for two random $\boldsymbol{\mu} \notin \Xi_s$.

There is little hope to apply successfully RB with the results of these numerical experiments. To complement, the next proposition shows that not all solutions of the 1D version can be well approximated as a linear combination of its solutions in the maximum norm.

**Proposition 19.** *Consider the following equation with parameters* $\mu \in [m, M] \subset (0, 1)$*,*

$$\begin{cases} u''(x;\mu) = \delta'_\mu \quad in\ (0,1), \\ u'(0) = u'(1) = 0. \end{cases} \tag{2.40}$$

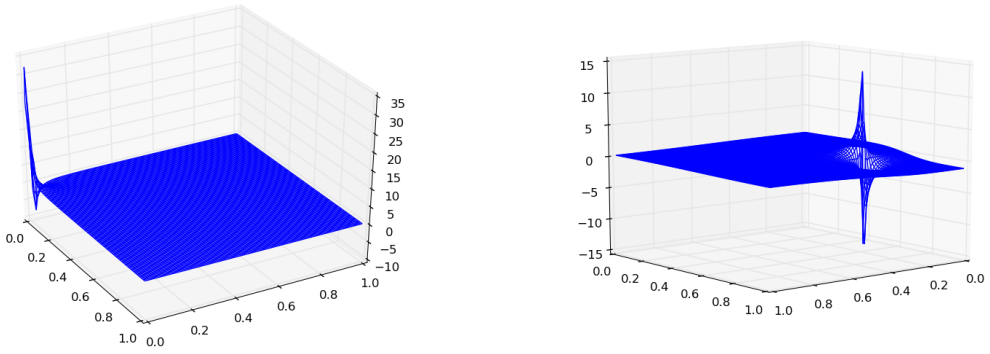

Figure 2.18: Direct approach for the isotropic case. Truth solutions of two random $\boldsymbol{\mu} \in \mathcal{P}$.

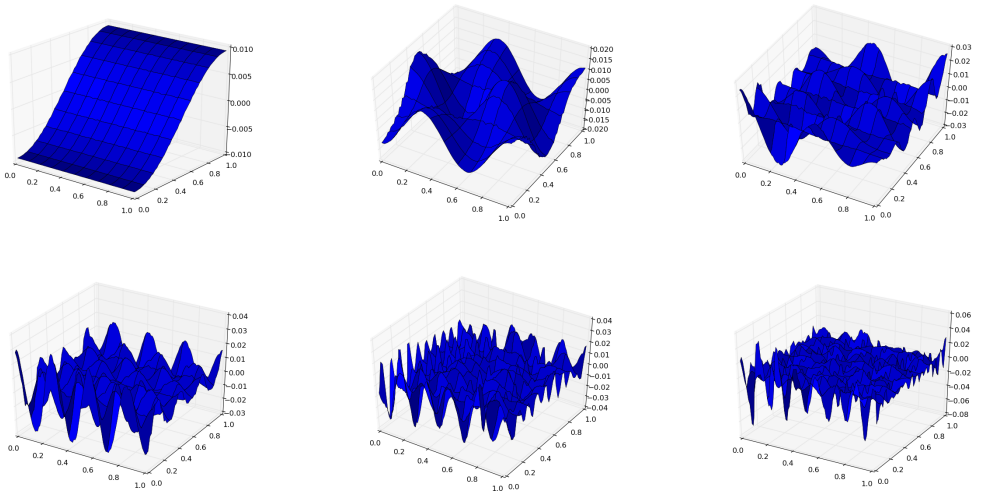

Figure 2.19: POD applied to the direct approach (isotropic case) in a compact parameter set. First, 20th, 40th, 80th, 160th and 320th element of the reduced basis for POD2.

*Then for any fixed selection of* $\mu_i \in [m, M], i = 1, \ldots, N$*, there exists an interval of values* $\mu$ *such that for any choice of the parameters* $\alpha_i \in \mathbb{R}, i = 1, \ldots, N$*, the following inequality holds*

$$\sup_{x \in [0,1]} |u(x;\mu) - \sum_{i=1}^{N} \alpha_i u(x;\mu_i)| \geq 1/2.$$

*Proof.* The solution of (2.40) is the Heaviside function

$$u(x;\mu) = \begin{cases} 0 & x \in [0,\mu) \\ 1 & x \in [\mu,1] \end{cases}$$

which can be checked directly by substitution. Suppose first that $m$ does not belong to the

selection $\{\mu_1, \ldots, \mu_N\}$, which has a minimum $u_{j_0} = m_1$ for certain $j_0$. Picking an arbitrary $\mu \in (m, m_1)$ yields that

$$u(\mu;\mu) - \sum_{i=1}^{N} \alpha_i u(\mu;\mu_i) = 1, \qquad \forall\, \alpha_i \in \mathbb{R}, i = 1, , \ldots, , N,$$

because $\mu < \mu_i$, for each $i$ and then $u(\mu;\mu_i) = 0$ for each $i$.

Suppose now that there exists $j_0$ such that $\mu_{j_0} = m$, and proceeding as before let us denote by $m_1$ the minimum of $\mu_i$ for $i \neq j_0$. For $\mu \in (m, m_1)$ we have

$$|u(\mu;\mu) - \sum_{i=1}^{N} \alpha_i u(\mu;\mu_i)| = |1 - \alpha_{j_0}|,$$

$$|u(\mu_{j_0};\mu) - \sum_{i=1}^{N} \alpha_i u(\mu_{j_0};\mu_i)| = |\alpha_{j_0}|,$$

thus

$$\sup_{x\in[0,1]} |u(x;\mu) - \sum_{i=1}^{N} \alpha_i u(x;\mu_i)| \geq \max\left(|1 - \alpha_{j_0}|, |\alpha_{j_0}|\right).$$

Since $\min_{\alpha\in\mathbb{R}} \max\left(|1 - \alpha_{j_0}|, |\alpha_{j_0}|\right) = 1/2$ (for $\alpha = 1/2$), the thesis follows. □

## 2.3 Solution for the EEG Inverse Problem

The final objective of solving the EEG equation

$$\begin{cases} \operatorname{div}(\sigma\nabla u) = \operatorname{div}(\mathbf{p}_0\delta_{\mu_0}) & \text{in } \Omega, \\ (\sigma\nabla u)\cdot \mathbf{n} = 0 & \text{on } \partial\Omega. \end{cases} \tag{2.41}$$

is to obtain the values $(\boldsymbol{\mu}_0, \mathbf{p}_0)$ that best fit the model from some measurements on the boundary.

We express the *continuous EEG inverse problem* for $\Gamma := \partial\Omega$ like this: Given $f = u|_\Gamma$, where $u|_\Gamma$ is the trace of a solution of (2.41), find the pair $(\boldsymbol{\mu}_0, \mathbf{p}_0)$ that generated $f$.

This inverse problem has been studied in [16, 17, 18]. In these research papers the authors proved several important results regarding the solvability and stability of the EEG inverse problem in its continuous presentation. Here we present two of them.

**Theorem 21.** *If two solutions $u(\boldsymbol{\mu}_1, \mathbf{p}_1)$ and $u(\boldsymbol{\mu}_2, \mathbf{p}_2)$ of (2.41) has the same boundary measurements on a nonempty and connected $\Gamma_* \subset \Gamma$, i.e. $u(\boldsymbol{\mu}_1, \mathbf{p}_1)|_{\Gamma_*} = u(\boldsymbol{\mu}_2, \mathbf{p}_2)|_{\Gamma_*}$, then $(\boldsymbol{\mu}_1, \mathbf{p}_1) = (\boldsymbol{\mu}_2, \mathbf{p}_2)$.*

*Proof.* See [16]. □

Theorem 21 guarantees that boundary measurements are enough to recover the source in equation (2.41).

**Definition 7.** *Let $\mathcal{P} \subset \Omega \times \mathbb{R}^d$ be a parameter set for the EEG problem (2.41) and*

$$f(\mathcal{P}) := \{u(\boldsymbol{\mu}, \mathbf{p})|_\Gamma : (\boldsymbol{\mu}, \mathbf{p}) \in \mathcal{P}\}.$$

*We define the EEG inverse function*

$$\varphi : f(\mathcal{P}) \to \mathcal{P},$$

*as $\varphi(f) := (\boldsymbol{\mu}, \mathbf{p})$, where $f = u(\boldsymbol{\mu}, \mathbf{p})|_\Gamma$.*

In Definition 7 we are overloading the symbol $f$. As a function of the parameters it can denote the set $f(\mathcal{P})$ or an element $f \in f(\mathcal{P})$, and at the same time acts as a real function defined on $\Gamma$. The context will differentiate the meaning.

**Theorem 22.** *Let $\mathcal{P} \subset \Omega \times \mathbb{R}^d$ be a compact parameter set, then there exists a constant $L > 0$ such that $\varphi$ satisfies the Lipschitz continuity*

$$|\varphi(f_1) - \varphi(f_2)| \leq L\|f_1 - f_2\|_{L^2(\Gamma)}, \qquad \forall f_1, f_2 \in f(\mathcal{P}).$$

*Proof.* See [16]. □

Theorem 21 and Theorem 22 a real number $s_0 > 0$ prove the well-posedness of the continuous EEG inverse problem. Also, in [16, 17, 18] there are different algorithms to construct the inverse function $\varphi$ but always having complete information of $f$ over the boundary.

In practical applications we never have continuous measurements but pointwise approximations of $f$, therefore we write the *discrete EEG inverse problem* for $\Gamma_q := \{\mathbf{x}_1, \ldots, \mathbf{x}_q\} \subset \Gamma$ as follows: Consider $f = u(\boldsymbol{\mu}, \mathbf{p})|_\Gamma$ and let $f_{meas}$ and $f_s(\mathbf{x}; (\boldsymbol{\mu}, \mathbf{p}))$ be two continuous functions on $\Gamma$ such that $f_{meas} \approx f$ and $f_s \approx f$ for all the parameters in the parameter set. Then, given the measurements

$$\hat{\mathbf{f}}_{meas} := (f_{meas}(\mathbf{x}_1), \ldots, f_{meas}(\mathbf{x}_q)),$$

find the pair $(\boldsymbol{\mu}_0, \mathbf{p}_0)$ such that

$$(\boldsymbol{\mu}_0, \mathbf{p}_0) = \underset{\boldsymbol{\mu} \in \Omega, \mathbf{p} \in \mathbb{R}^d}{\operatorname{argmin}} \|\hat{\mathbf{f}}_s(\boldsymbol{\mu}, \mathbf{p}) - \hat{\mathbf{f}}_{meas}\|_2^2, \tag{2.42}$$

where

$$\hat{\mathbf{f}}_s(\boldsymbol{\mu}, \mathbf{p}) := (f_s(\mathbf{x}_1; (\boldsymbol{\mu}, \mathbf{p})), \ldots, f_s(\mathbf{x}_q; (\boldsymbol{\mu}, \mathbf{p}))).$$

Certainly, Theorem 21 and Theorem 22 do not extend to the discrete inverse problem, therefore the well-posedness is not guarantee in general. Note also that we cannot take discrete measurements directly from $f$ because $f \in H^{1/2}(\Gamma)$.

This is a common theme in almost every inverse problem controlled by a partial differential equation. Therefore, we will introduce a methodology that applies not only to the EEG case but also to other inverse problems with the same setting.

The aim of the general methodology is to use MOR to compute large quantities of solutions of the parameterized PDE to approximate the inverse function $\varphi$ in a fast and accurate manner.

The usual approach to solve this kind of inverse problems is to use iterative methods like simulated annealing [90, 129]. As they require many iteration of the forward problem, model

order reduction is well suited tool for this task. Nevertheless, we would like to use the fast computation times to actually remove the iterative part and solve the inverse problem in real time.

First we show how to reduce the parameter space dimension in the EEG problem and then we present some numerical results using simulating annealing to compare it later with the mentioned methodology. After the simulating annealing experiments we present the theoretical background for the approximation of the inverse function $\varphi$ and finally some experimental results.

Consider $f_s$ the trace of an approximated solution of (2.41), looked for in a finite dimensional space. To reduce the parameter space dimension we will express $\mathbf{p}$ as a function of $\boldsymbol{\mu}$. Write $\mathbf{p} = \sum_{i=1}^{d} p_i \mathbf{e}_i$, where $\mathbf{e}_i$ are the Cartesian unit vectors. Then, we have

$$\hat{\mathbf{f}}_s(\boldsymbol{\mu}, \mathbf{p}) = \sum_{i=1}^{d} p_i \hat{\mathbf{f}}_s(\boldsymbol{\mu}, \mathbf{e}_i),$$

because the variational formulation of (2.41) is linear with respect to the polarization.

Express $\hat{\mathbf{f}}_s(\boldsymbol{\mu}, \mathbf{p}) = \mathbf{M}_s(\boldsymbol{\mu})\mathbf{p}$ where $\mathbf{M}_s(\boldsymbol{\mu})$ is the $p \times d$ matrix

$$\mathbf{M}_s(\boldsymbol{\mu}) := \left[ \hat{\mathbf{f}}_s(\boldsymbol{\mu}, \mathbf{e}_1) \cdots \hat{\mathbf{f}}_s(\boldsymbol{\mu}, \mathbf{e}_d) \right],$$

then, given $\boldsymbol{\mu} \in \mathcal{P}$, the optimum $\mathbf{p}_{\boldsymbol{\mu}} \in \mathbb{R}^d$ that minimizes

$$\|\mathbf{M}_s(\boldsymbol{\mu})\mathbf{p} - \hat{\mathbf{f}}_{meas}\|_2^2$$

is equivalent to determining the solution of the normal equations

$$\mathbf{M}_s^t(\boldsymbol{\mu})\mathbf{M}_s(\boldsymbol{\mu})\mathbf{p}_{\boldsymbol{\mu}} = \mathbf{M}_s^t(\boldsymbol{\mu})\hat{\mathbf{f}}_{meas}, \tag{2.43}$$

which is linear system of dimension $d \times d$.

In conclusion, we can find the optimum values $(\boldsymbol{\mu}_0, \mathbf{p}_0)$ of (2.42) solving the equivalent minimization problem

$$(\boldsymbol{\mu}_0, \mathbf{p}_0) = \underset{\boldsymbol{\mu} \in \Omega}{\operatorname{argmin}} \|\mathbf{M}_s(\boldsymbol{\mu})\mathbf{p}_{\boldsymbol{\mu}} - \hat{\mathbf{f}}_{meas}\|_2^2, \tag{2.44}$$

where $\mathbf{p}_{\boldsymbol{\mu}}$ is the solution of (2.43).

All the results of MOR of previous sections assume a fixed vector $\mathbf{p}$. The matrix $\mathbf{M}_s$ depends only on $\boldsymbol{\mu}$ with $d$ different polarization vectors $\mathbf{e}_i$. Therefore, to compute (2.44) we can construct $d$ reduced models, one for each $\mathbf{e}_i$ as explained in Section 2.1.2, and compute very fast the matrix $\mathbf{M}_s$. As we only have to take in account the parameter $\boldsymbol{\mu}$ and not $\mathbf{p}$, let's fix a compact parameter set $\mathcal{P} \subset \Omega$ for $\boldsymbol{\mu}$.

We used for the experiments the same setting as Section 2.1.2. We considered $f_{meas} = f_h$ the truth solution and $f_s = f_N$ the reduced model solution. Figure 2.20 shows a comparison of average computation times to obtain the pair $(\boldsymbol{\mu}_0, \mathbf{p}_0)$ with a precision of 1e−5 using simulating annealing. The quantity of evaluation points were $q = 16$ uniformly selected over $\Gamma$.

| | $\boldsymbol{\mu}_0 = (0.1311, 0.6)$<br>$\mathbf{p}_0 = (-0.2425, 0.9701)$ | $\boldsymbol{\mu}_0 = (0.1451, 0.2)$<br>$\mathbf{p}_0 = (-0.5161, 0.8440)$ |
|---|---|---|
| Subtraction approach FEM | 1147.51 sec | 1406.99 sec |
| Subtraction approach RB | 27.04 sec | 30.33 sec |
| Subtraction approach FOR | 18.87 sec | 21.02 sec |

Figure 2.20: Simulated annealing on the EEG inverse problem. Average computing times for a precision of 1e−5. Initial parameter in the simulated annealing algorithm: $\boldsymbol{\mu} = (0.125, 0.5)$.

The results in Figure 2.20 are fine but can be improved. The main idea with the methodology is to extend the inverse function $\varphi$ outside of $f(\mathcal{P})$ and define a new function

$$\varphi_s : \mathbb{R}^q \to \mathbb{R}^d$$

from $\varphi$ such that if $f_s(\boldsymbol{\mu}) \approx f(\boldsymbol{\mu})$ for all $\boldsymbol{\mu} \in \mathcal{P}$ and $(y_1, \ldots, y_q) = \hat{\mathbf{f}}_s(\boldsymbol{\mu})$, then $\varphi_s(y_1, \ldots, y_q) \approx \boldsymbol{\mu}$.

A function like $\varphi_s$ is usually ill-posed because it may happen that two distant sources generate the same pointwise evaluation on the approximations $f_s$. Nevertheless, we will use two universal approximation results from [48, 117] to construct a precise and stable function $\varphi_s$ given enough evaluation points. First we present some definitions.

**Definition 8.** *A function $\rho : (0, \infty) \to (\infty, \infty)$ is called modulus of continuity if it is positive, increasing, subadditive and satisfies $\rho(t) \to 0$ as $t \to 0^+$.*

**Definition 9.** *Given a modulus of continuity $\rho$ and a compact subset $K \subset L^2$, we define the set $\mathcal{F}_{\rho,K}$ as the functionals $F \in C(K) := \{F : K \to \mathbb{R} \mid F \text{ is continuous}\}$ such that*

$$|F(f_1) - F(f_2)| \leq \rho(\|f_1 - f_2\|_{L^2}), \qquad \forall f_1, f_2 \in K.$$

**Definition 10.** *We define the set (TW) of Tabuer-Wiener functions as the nonpolynomial continuous tempered distributions.*

The following result will help extending the definition of $\varphi$ outside of $f(\mathcal{P})$.

**Theorem 23.** *Let $N \geq 1, p \geq 1$ be integers, $\rho$ be a modulus of continuity, $K$ be a compact subset of $L^2(\Gamma)$, and $\phi : \mathbb{R} \to \mathbb{R}$ be an infinitely differentiable function in an open interval of $\mathbb{R}$. We further assume that there exists $b$ in this interval where all derivatives of $\phi$ are different from zero.*

*Then, there exist a constant $c > 0$, continuous linear functionals $\gamma_i : C(K) \to \mathbb{R}, i = 1, \ldots, N$, and a continuous linear operator $L : L^2(\Gamma) \to \mathbb{R}^p$ with the following property: for every $F \in \mathcal{F}_{\rho,K}$ there exist vectors $\mathbf{a}_i = \mathbf{a}_i(F) \in \mathbb{R}^p, i = 1, \ldots, N$, such that*

$$\left| F(f) - \sum_{i=1}^{N} \gamma_i(F)\phi(\mathbf{a}_i \cdot L(f) + b) \right| \leq c\big(\rho(\epsilon_1(r)) + \rho(\epsilon_2(r, N))\big), \qquad \forall f \in K,$$

*where $\epsilon_1(p) \to 0$ when $p \to \infty$ and $\epsilon_2(p, N) \to 0$ when $N \to \infty$ for any fixed $p$.*

*Proof.* See [117]. □

On the other hand the following result let us approximate any functional from pointwise evaluations of its input.

**Theorem 24.** *Suppose that $\sigma \in (TW)$, $V$ is a compact set of $C(\Gamma)$ and $F$ is a continuous functional defined on $V$. Then, for any $\varepsilon > 0$ there exist two integers $m$ and $N$, points $\mathbf{x}_1, \dots, \mathbf{x}_m \in \Gamma$ and real constants $c_i, \theta_i, \xi_{ij}, i = 1, \dots N, j = 1, \dots, m$, such that*

$$\left| F(f) - \sum_{i=1}^{N} c_i \sigma\left( \sum_{j=1}^{m} \xi_{ij} f(\mathbf{x}_j) + \theta_i \right) \right| < \varepsilon, \qquad \forall f \in V. \tag{2.45}$$

*Proof.* See [48]. □

Theorem 23 and Theorem 24 are similar but have important differences. Theorem 23 is more general but we do not have control over the functional $L_r$, on the other hand Theorem 24 is closer to our problem but the functionals that takes in account are defined in $C(\Gamma)$, which is not the case for $\varphi$. Nevertheless, combining both results we can construct an approximation of $\varphi$.

**Theorem 25.** *Let $f_s = f_s(\boldsymbol{\mu}) \in C(\Gamma)$ be the trace of an approximate solution of the EEG equation (2.41) from a finite dimensional space, and assume that $f_s$ converges to $f = u(\boldsymbol{\mu})|_\Gamma$ when $s \to 0$ uniformly for $\boldsymbol{\mu}$ in a compact parameter set $\mathcal{P}$. Then, given $\sigma \in (TW)$, for all $\varepsilon > 0$ there exist a real number $s > 0$, an integer $m$, a selection of points $\mathbf{x}_1, \dots, \mathbf{x}_m \in \Gamma$ and integers $N^{(k)}$ and real values $c_i^{(k)}, \xi_{ij}^{(k)}, \theta_i^{(k)}, k = 1, \dots, d, j = 1, \dots, m, i = 1, \dots N^{(k)}$, such that*

$$\varphi_*(g) := \left( \sum_{i=1}^{N^{(1)}} c_i^{(1)} \sigma\left( \sum_{j=1}^{m} \xi_{ij}^{(1)} g(\mathbf{x}_j) + \theta_i^{(1)} \right), \dots, \sum_{i=1}^{N^{(d)}} c_i^{(d)} \sigma\left( \sum_{j=1}^{m} \xi_{ij}^{(d)} g(\mathbf{x}_j) + \theta_i^{(d)} \right) \right) \tag{2.46}$$

*satisfies*

$$|\varphi(f(\boldsymbol{\mu})) - \varphi_*(f_s(\boldsymbol{\mu}))| < \varepsilon, \qquad \forall \boldsymbol{\mu} \in \mathcal{P}. \tag{2.47}$$

*Proof.* The main idea of the proof is to extend $\varphi$ to the functions $f_s$ using Theorem 23 and then use Theorem 24 to obtain good approximations from pointwise evaluations of $f_s$.

To use Theorem 23 we need to define a modulus of continuity. From Theorem 22 we have that $\varphi$ is Lipschitz continuous in $f(\mathcal{P})$ and therefore uniformly continuous as well. Then, is straightforward to prove that

$$\rho(t) = \sup\{|\varphi(f_1) - \varphi(f_2)| : \|f_1 - f_2\|_{L^2(\Gamma)} \leq t\},$$

is a well-defined modulus of continuity.

Assume $K = f(\mathcal{P})$ in Theorem 23, which is a compact subset of $L^2(\Gamma)$ because $\mathcal{P}$ is compact and $f$ as a function of $\boldsymbol{\mu}$ is continuous (the solution map is continuous, see Proposition 12). Therefore, $\varphi \in \mathcal{F}_{\rho, f(\mathcal{P})}$ as well as each of its components

$$\varphi = (\varphi^{(1)}, \dots, \varphi^{(d)}).$$

We will prove (2.47) for the first of the $d$ components, the others are analogous.

According to Theorem 23 given $N' \geq 1, p \geq 1$ and a specific infinitely differentiable function $\phi$ there exist for $\varphi^{(1)}$ a constant $c > 0$, real values $\gamma_i := \gamma_i(\varphi^{(1)})$, vectors $\mathbf{a}_i \in \mathbb{R}^p, i = 1, \ldots, N'$, and a continuous linear operator $L : L^2(\Gamma) \to \mathbb{R}^p$ such that

$$\left|\varphi^{(1)}(f) - \sum_{i=1}^{N'} \gamma_i \phi(\mathbf{a}_i \cdot L(f) + b)\right| \leq c\big(\rho(\epsilon_1(p)) + \rho(\epsilon_2(p, N'))\big), \qquad \forall f \in f(\mathcal{P}),$$

where $\epsilon_1(p) \to 0$ when $p \to \infty$ and $\epsilon_2(p, N') \to 0$ when $N' \to \infty$ for any fixed $p$.

Consider $\phi$ as a infinitely differential function on the real line satisfying the condition written in Theorem 23. Fix $p$ such that $c\rho(\epsilon_1(p)) < \varepsilon/6$ and then $N'$ such that $c\rho(\epsilon_2(p, N')) < \varepsilon/6$. Hence, for certain $\gamma_i, \mathbf{a}_i$ and $L$ we obtain

$$\left|\varphi^{(1)}(f) - \sum_{i=1}^{N'} \gamma_i \phi(\mathbf{a}_i \cdot L(f) + b)\right| < \varepsilon/3, \qquad \forall f \in f(\mathcal{P}). \tag{2.48}$$

Define for $g \in L^2(\Gamma)$,

$$\varphi_\#(g) := \sum_{i=1}^{N'} \gamma_i \phi(\mathbf{a}_i \cdot L(g) + b).$$

The argument inside of $\phi$, $\mathbf{a}_i \cdot L_r(g) + b$, is continuous for all $g \in L^2(\Gamma)$ and hence for a small enough $s$ we obtain

$$|\mathbf{a}_i \cdot L(f_s(\mathcal{P})) + b| \leq |\mathbf{a}_i| \|L\| \|f_s(\mathcal{P})\|_{L^2(\Gamma)} + |b| \leq |\mathbf{a}_i| \|L\| \big(\|f(\mathcal{P})\|_{L^2(\Gamma)} + 1\big) + |b|.$$

The set $f(\mathcal{P})$ is compact in $L^2(\Gamma)$, therefore there exists $C$ such that

$$\|f(\mathcal{P})\|_{L^2(\Gamma)} \leq C.$$

Define the set

$$\Phi := \Big[ -\max_{i=1,\ldots N'} |\mathbf{a}_i| \|L\| (C+1) - |b|, \max_{i=1,\ldots N'} |\mathbf{a}_i| \|L\| (C+1) + |b| \Big].$$

The function $\phi$ is infinite differentiable so is Lipschitz continuous in $\Phi$. Write $l_\phi$ as the corresponding Lipschitz constant.

From the Lipschitz continuity we can compute the estimate

$$\begin{aligned} |\varphi_\#(f(\boldsymbol{\mu})) - \varphi_\#(f_s(\boldsymbol{\mu}))| &\leq \sum_{i=1}^{N'} |\gamma_i| |\phi(\mathbf{a}_i \cdot L(f(\boldsymbol{\mu})) + b) - \phi(\mathbf{a}_i \cdot L(f_s(\boldsymbol{\mu})) + b)| \\ &\leq l_\phi \sum_{i=1}^{N'} |\gamma_i| |\mathbf{a}_i \cdot (L(f(\boldsymbol{\mu}) - f_s(\boldsymbol{\mu})))| \\ &\leq l_\phi \|L\| \|f(\boldsymbol{\mu}) - f_s(\boldsymbol{\mu})\|_{L^2(\Gamma)} \sum_{i=1}^{N'} |\gamma_i| |\mathbf{a}_i|. \end{aligned} \tag{2.49}$$

The convergence of $f_s$ to $f$ is uniform so pick $s$ in (2.49) such that

$$\|f(\boldsymbol{\mu}) - f_s(\boldsymbol{\mu})\|_{L^2(\Gamma)} < \frac{\varepsilon}{3l_\phi \|L_r\| \sum_{i=1}^{N'} |\gamma_i| \|\mathbf{a}_i|}, \qquad \forall \boldsymbol{\mu} \in \mathcal{P},$$

which implies that

$$|\varphi_\#(f(\boldsymbol{\mu})) - \varphi_\#(f_s(\boldsymbol{\mu}))| < \varepsilon/3, \qquad \forall \boldsymbol{\mu} \in \mathcal{P}. \tag{2.50}$$

Now we will use Theorem 24. The map $f_s(\boldsymbol{\mu})$ is continuous in the $L^2(\Gamma)$-norm and therefore also continuous in the $C(\Gamma)$-norm because of the norm equivalence in finite dimensional spaces, consequently $f_s(\mathcal{P})$ is compact in $C(\Gamma)$. Also by the norm equivalence $\varphi_\#$ is continuous in $f_s(\mathcal{P})$ for the $C(\Gamma)$-norm. Select a function $\sigma \in (TW)$ and apply Theorem 24 to obtain an approximation of $\varphi_\#$,

$$\varphi_*^{(1)}(f_s) := \sum_{i=1}^{N^{(1)}} c_i^{(1)} \sigma\Big( \sum_{j=1}^{m^{(1)}} \xi_{ij}^{(1)} f_s(\mathbf{x}_j^{(1)}) + \theta_i^{(1)} \Big),$$

such that

$$|\varphi_\#(f_s) - \varphi_*^{(1)}(f_s)| < \varepsilon/3, \qquad \forall f_s \in f_s(\mathcal{P}). \tag{2.51}$$

We can now obtain the estimate (2.47) from (2.48), (2.50) and (2.51),

$$\begin{aligned} \|\varphi^{(1)}(f(\boldsymbol{\mu})) - \varphi_*^{(1)}(f_s(\boldsymbol{\mu})\| \leq \|\varphi^{(1)}(f(\boldsymbol{\mu})) - \varphi_\#(f(\boldsymbol{\mu}))\| + \|\varphi_\#(f(\boldsymbol{\mu})) - \varphi_\#(f_s(\boldsymbol{\mu}))\| + \\ + \|\varphi_\#(f_s(\boldsymbol{\mu})) - \varphi_*^{(1)}(f_s(\boldsymbol{\mu}))\| < \varepsilon, \qquad \forall \boldsymbol{\mu} \in \mathcal{P}. \end{aligned} \tag{2.52}$$

The points $\mathbf{x}_1, \ldots, \mathbf{x}_m$ in (2.46) are the same for all the components of $\varphi_*$ but it may happen that Theorem 24 selects different points for each component. Nevertheless, we can consider

$$\varphi_*^{(1)}(f_s) = \sum_{i=1}^{N^{(1)}} c_i^{(1)} \sigma\Big( \sum_{j=1}^{m} \xi_{ij}^{(1)} f_s(\mathbf{x}_j) + \theta_i^{(1)} \Big), \tag{2.53}$$

where $\mathbf{x}_j, j = 1, \ldots, m^{(1)}$, are the selected points for the first component and $\mathbf{x}_j, j = m^{(1)} + 1, \ldots, m$, are all the points from the other components not present in the first. The value $m \geq m^{(1)}$ represents the quantity of different points selected over all the components. To mantain the estimate (2.52) select $\xi_{ij}^{(1)} = 0$ for all $j = m^{(1)} + 1, \ldots, m$ in (2.53) for each $i = 1, \ldots, N^{(1)}$. We can apply the same idea to all the components of $\varphi_*$ and in this way obtain an approximation that uses the same points in all its components. □

In the proof of Theorem 25 we only used the uniform continuity of the inverse function $\varphi$ and the continuity of the solution map $u(\boldsymbol{\mu})$. Therefore, this result can be extended readily to other inverse problems satisfying these conditions.

Also note that uniform convergence from finite dimensional spaces is feasible from the RB and FOR methods if we can apply the Kolmogorov's $n$-width estimates from Section 1.4 and Section 1.5. For example, in the subtraction approach we already proved some of such estimates (Section 2.1.1 and Section 2.1.2). For some Lipschitz continuous solution maps we can prove uniform convergence from pointwise convergence. We will show a proof for the subtraction approach but is easily extendable to other PDEs under the conditions of Proposition 9.

**Proposition 20.** *Assume a compact parameter set $\mathcal{P}$ and let $f_s$ be the trace of solutions of the subtraction approach equation (2.28) from finite dimensional spaces indexed by $s$ such that for all $\boldsymbol{\mu} \in \mathcal{P}$, $f_s(\boldsymbol{\mu}) \to f(\boldsymbol{\mu})$ in $L^2(\Gamma)$ when $s \to 0$. Then, $f_s$ converges to $f$ when $s \to 0$, uniformly with respect to $\boldsymbol{\mu}$.*

*Proof.* We need to prove that for all $\varepsilon > 0$ there exists $S > 0$ such if $s < S$, then

$$\|f(\boldsymbol{\mu}) - f_s(\boldsymbol{\mu})\|_{L^2(\Gamma)} < \varepsilon, \qquad \forall \boldsymbol{\mu} \in \mathcal{P}.$$

We will prove it by contradiction. Suppose that there exists $\varepsilon_* > 0$ such that for all $k \in \mathbb{N}, k \neq 0$, there is $s_k \leq 1/k$ and a parameter $\boldsymbol{\mu}_k \in \mathcal{P}$ that satisfies

$$\|f(\boldsymbol{\mu}_k) - f_{s_k}(\boldsymbol{\mu}_k)\|_{L^2(\Gamma)} \geq \varepsilon_*. \tag{2.54}$$

The set $\mathcal{P}$ is compact so from the sequence $(\boldsymbol{\mu}_k)_{k\geq 1}$ we can extract a subsequence that converges to a parameter $\boldsymbol{\mu}_* \in \mathcal{P}$. Redefine $(s_k)_{k\geq 1}$ and $(\boldsymbol{\mu}_k)_{k\geq 1}$ with the indexes of this subsequence.

The inequality (2.54) implies that

$$\varepsilon_* \leq \|f(\boldsymbol{\mu}_k) - f(\boldsymbol{\mu}_*)\|_{L^2(\Gamma)} + \|f(\boldsymbol{\mu}_*) - f_{s_k}(\boldsymbol{\mu}_*)\|_{L^2(\Gamma)} + \|f_{s_k}(\boldsymbol{\mu}_*) - f_{s_k}(\boldsymbol{\mu}_k)\|_{L^2(\Gamma)}. \tag{2.55}$$

The expressions $\|f(\boldsymbol{\mu}_k) - f(\boldsymbol{\mu}_*)\|_{L^2(\Gamma)}$ and $\|f(\boldsymbol{\mu}_*) - f_{s_k}(\boldsymbol{\mu}_*)\|_{L^2(\Gamma)}$ in (2.55) goes to zero when $k \to \infty$ by the continuity of $f$ and the assumptions of the proposition respectively.

Consider $T$ the trace operator, then

$$\|f_{s_k}(\boldsymbol{\mu}_*) - f_{s_k}(\boldsymbol{\mu}_k)\|_{L^2(\Gamma)} = \|Tu_{s_k}(\boldsymbol{\mu}_*) - Tu_{s_k}(\boldsymbol{\mu}_k)\|_{L^2(\Gamma)} \leq \|T\|\|u_{s_k}(\boldsymbol{\mu}_*) - u_{s_k}(\boldsymbol{\mu}_k)\|_{H^1(\Omega)}.$$

By Proposition 9 there exists a positive constant $\gamma$ (see Theorem 17 for an idea on how to compute exact estimates) such that

$$\|T\|\|u_{s_k}(\boldsymbol{\mu}_*) - u_{s_k}(\boldsymbol{\mu}_k)\|_{H^1(\Omega)} \leq \|T\|\frac{\gamma}{\alpha_{s_k}}|\boldsymbol{\mu}_* - \boldsymbol{\mu}_k|, \tag{2.56}$$

where $\alpha_{s_k}$ is the coercivity constant associated with the space where $u_{s_k}$ is computed. The constants $\alpha_{s_k}$, for all $k \in \mathbb{N}$, are bounded below by the coercivity constant associated with problem (2.28), therefore $1/\alpha_{s_k}$ is bounded above and (2.56) goes to zero when $k \to \infty$.

In conclusion, the right hand side of (2.55) becomes infinitesimal as $k$ grows which is a contradiction, as it has to be larger $\varepsilon_* > 0$. □

In practical scenarios we cannot choose the quantity of evaluation points or their positions, instead they are fixed. From Theorem 25 we may find that the selected integer $m$ for a desired approximation precision exceeds the fixed quantity that we have at hand. Also, another typical problem is that we cannot compute exactly (2.46). Therefore, we propose a practical approach which is common is such situations.

Let $\mathbf{x}_1, \ldots, \mathbf{x}_q$ be $q$ fixed evaluation points and $f_h(\boldsymbol{\mu})$ be the trace on $\Gamma$ of a finite element approximation for $\boldsymbol{\mu} \in \mathcal{P}$. Then, define an approximation of $\varphi$ as (2.46),

$$\varphi_*(g) := \left( \sum_{i=1}^{N^{(1)}} c_i^{(1)} \sigma\left( \sum_{j=1}^{q} \xi_{ij}^{(1)} g(\mathbf{x}_j) + \theta_i^{(1)} \right), \ldots, \sum_{i=1}^{N^{(d)}} c_i^{(d)} \sigma\left( \sum_{j=1}^{q} \xi_{ij}^{(d)} g(\mathbf{x}_j) + \theta_i^{(d)} \right) \right)$$

such that

$$|\boldsymbol{\mu} - \varphi_*(f_h(\boldsymbol{\mu}))| < \varepsilon, \qquad \forall \boldsymbol{\mu} \in \Xi_{tr},$$

where $\Xi_{tr}$ is a finite and "dense" subset of $\mathcal{P}$, and at the same time it has acceptable approximations outside of $\Xi_{tr}$ (also known as extrapolation or generalization).

To construct $\varphi_*$ given a $\sigma \in (TW)$ we can solve numerically the following problem

$$\min_{N,c,\xi,\theta} \sum_{\boldsymbol{\mu}\in\Xi_{tr}} \left| \boldsymbol{\mu} - \left( \sum_{i=1}^{N^{(1)}} c_i^{(1)} \sigma\left( \sum_{j=1}^{q} \xi_{ij}^{(1)} f_h(\mathbf{x}_j;\boldsymbol{\mu}) + \theta_i^{(1)} \right), \ldots, \sum_{i=1}^{N^{(d)}} c_i^{(d)} \sigma\left( \sum_{j=1}^{q} \xi_{ij}^{(d)} f_h(\mathbf{x}_j;\boldsymbol{\mu}) + \theta_i^{(d)} \right) \right) \right|^2$$

up to the desired precision. Here the vectors $N = (N^{(1)}, \ldots, N^{(d)}), c = (\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(d)}), \xi = (\boldsymbol{\xi}^{(1)}, \ldots, \boldsymbol{\xi}^{(d)}), \theta = (\theta^{(1)}, \ldots, \theta^{(d)})$ represent all the values that define $\varphi_*$.

To test that $\varphi_*$ extrapolates well after solving the minimization problem we can evaluate it in another finite and "dense" subset $\Xi_{te}$ of $\mathcal{P}$ such that $\Xi_{te} \cap \Xi_{tr} = \emptyset$. In other words, we can compute the average error

$$\varepsilon_{te} := |\Xi_{te}|^{-1} \sum_{\boldsymbol{\mu}\in\Xi_{te}} |\boldsymbol{\mu} - \varphi_*(f_h(\boldsymbol{\mu}))|^2.$$

As we want a "dense" training and test set to improve the reliability of $\varphi_*$ we need to compute many forward problems of (2.41) to build the sets $f_h(\Xi_{tr})$ and $f_h(\Xi_{te})$. From the finite element method point of view this may not feasible, therefore MOR will be a key tool in the construction of those sets.

We are now ready to write the methodology to solve the EEG inverse problem. This methodology has two stages, offline and online. Here are the steps:

1. **Offline**: Construct a reduced basis of dimension $M$ that approximates well the EEG equation (2.41). A good selection of $M$ parameters are the ones that represent well the linear functional in the subtraction approach, see Section 2.1.2. We prefer to use the FOR method.

2. **Offline**: Select a finite and dense training and test sets $\Xi_{tr} \subset \mathcal{P}$ and $\Xi_{te} \subset \mathcal{P}$. A random selection is a good approach.

3. **Offline**: For all $\boldsymbol{\mu} \in \Xi_{tr}, \Xi_{te}$ compute $f_M(\boldsymbol{\mu})$ using the reduced basis. For the FOR method we follow the procedure explained in Section 1.5.

4. **Offline**: Solve the minimization problem

$$\min_{N,c,\xi,\theta} \sum_{\boldsymbol{\mu}\in\Xi_{tr}} |\boldsymbol{\mu} - \varphi_*(f_M(\boldsymbol{\mu}))|^2 \tag{2.57}$$

   up to a suitable precision, determining the parameters $N, c, \xi, \theta$.

5. **Offline**: Define for all $\mathbf{y} = (y_1, \ldots, y_q) \in \mathbb{R}^q$,

$$\varphi_*(\mathbf{y}) := \left( \sum_{i=1}^{N^{(1)}} c_i^{(1)} \sigma\left( \sum_{j=1}^{q} \xi_{ij}^{(1)} y_j + \theta_i^{(1)} \right), \ldots, \sum_{i=1}^{N^{(d)}} c_i^{(d)} \sigma\left( \sum_{j=1}^{q} \xi_{ij}^{(d)} y_j + \theta_i^{(d)} \right) \right),$$

   using the values of the parameters $N, c, \xi, \theta$ just determined.

6. **Offline**: Check through the test error

$$\varepsilon_{te} = |\Xi_{te}|^{-1} \sum_{\boldsymbol{\mu}\in\Xi_{te}} |\boldsymbol{\mu} - \varphi_*(f_M(\boldsymbol{\mu}))|^2$$

   if the parameters $N, c, \xi, \theta$ are good enough to extrapolate $\varphi_*$ outside the training set. An error $\varepsilon_{te} \leq 10^{-4}$ is good enough in applications.

7. **Online**: Given the measured data $\mathbf{y}_*$, compute the candidate location as $\boldsymbol{\mu}_{\mathbf{y}_*} = \varphi_*(\mathbf{y}_*)$.

8. **Online**: Using $\boldsymbol{\mu}_{\mathbf{y}_*}$ compute an approximation $\mathbf{p}_{\mathbf{y}_*}$ of the polarization solving the linear system (2.43).

If we want to test how good is the online stage, we could solve the equation (2.41) for $(\boldsymbol{\mu}_{\mathbf{y}_*}, \mathbf{p}_{\mathbf{y}_*})$ from the finite element space or the reduced space and assess $|\mathbf{y}_* - \hat{\mathbf{f}}_M(\boldsymbol{\mu}_{\mathbf{y}_*}, \mathbf{p}_{\mathbf{y}_*})|$.

The minimization problem (2.57) has been studied in the neural networks literature [80]. The most common algorithm to solve it is the stochastic gradient descent [150], for which is known that it gives sub-optimal results because of the nonlinearity of $\sigma$. The state of the art for a large number of parameters uses GPUs [32, 56] to exploit the parallelizable structure of $\varphi_*$.

In practical computations it is usual to fix N beforehand to reduce the complexity of the minimization. With same setting as in Section 2.1.2 we found that with around $N = 10$ we obtain a good accuracy for the EEG problem. To give a motivation of this choice we present some numerical results.

Figure 2.21 shows the average training and test error for different N. The function $\sigma$ considered was $\sigma(t) = \frac{1}{1+e^{-t}}$, the sets $\Xi_{tr}, \Xi_{te}$ were generated randomly for $|\Xi_{tr}| = |\Xi_{te}| = 10^5$ elements and the quantity of evaluation points were $q = 16$ uniformly selected over $\Gamma$.

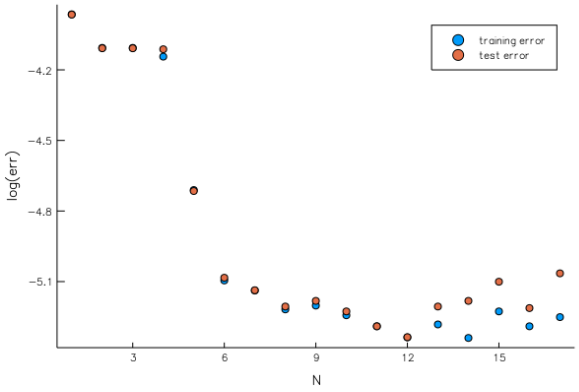


Figure 2.21: Approximation of the inverse EEG function. Average error in the training set $\Xi_{tr}$ and the test set $\Xi_{te}$ for different N.

For this experiment we picked randomly 200000 pair of parameters $(\boldsymbol{\mu}, \mathbf{p})$ and computed the FOR solution for each of them. Afterwards, we evaluated each solution in the sixteen selected points and separated the results into the training and test set. We applied the stochastic gradient descent to minimize (2.57) and iterate it many times to look for adequate parameters. The results in Figure 2.21 are the parameters that had the best average error from all the iterations.

Figure 2.21 shows that from $N = 6$ we have good approximations of the inverse function $\varphi$ but then it does not improve so much for larger N. It seems that for the selected setting the best average error of $\varphi_*$ will not pass $1e-6$. Clearly the evaluation of $\varphi_*$ is instantaneous and compared with simulated annealing we have obtained a huge improvement in CPU time. In Figure 2.20 we presented some computational times of the simulated annealing to obtain the same precision as $\varphi_*$.

After $N = 12$ there is a deterioration of the average test error. This is common for the type of approximation we are doing and is well documented [80]. The usual procedure is to take N with the smallest test error or use regularization techniques. We tried several regularization methods found in [73] but it did not improve much, so we do not include them.

Figure 2.22 and Figure 2.23 show a comparison between actual and estimated parameters for two elements of $\Xi_{te}$. The estimation of $\mathbf{p}$ was made through the linear problem (2.43).

| | N | $\mu_1$ | $\mu_2$ | $p_1$ | $p_2$ |
|---|---|---|---|---|---|
| Actual | | 0.138871 | 0.207602 | 2.756710 | 0.175281 |
| Estimation by $\varphi_*$ | 10 | 0.125529 | 0.218897 | 2.693873 | 0.081207 |
| Absolute Error | 10 | 0.013342 | 0.011295 | 0.062837 | 0.094073 |
| Estimation by $\varphi_*$ | 12 | 0.125467 | 0.218471 | 2.693468 | 0.081819 |
| Absolute Error | 12 | 0.013403 | 0.010869 | 0.063242 | 0.093461 |

Figure 2.22: Approximation of the inverse EEG function, $q = 16$. Comparison between actual and estimated parameter for an element of $\Xi_{te}$.

| | N | $\mu_1$ | $\mu_2$ | $p_1$ | $p_2$ |
|---|---|---|---|---|---|
| Actual | | 0.124902 | 0.679551 | 0.323747 | 1.212711 |
| Estimation by $\varphi_*$ | 10 | 0.124988 | 0.672039 | 0.323115 | 1.212085 |
| Absolute Error | 10 | 0.000086 | 0.007511 | 0.000632 | 0.000726 |
| Estimation by $\varphi_*$ | 12 | 0.124998 | 0.679761 | 0.323794 | 1.212797 |
| Absolute Error | 12 | 0.000096 | 0.000210 | 0.000047 | 0.000086 |

Figure 2.23: Approximation of the inverse EEG function, $q = 16$. Comparison between actual and estimated parameter for an element of $\Xi_{te}$.

The parameters in Figure 2.22 and Figure 2.23 were selected to show how the approximations fluctuate. The maximum and minimum error $\varphi_*$ achieves in this experiment are 0.04 and $1.6e-9$ respectively for $N = 12$. Certainly, there are many possible ways to construct $\varphi_*$ and the results shown here may be sub-optimal.

We present now an experiment with $q = 128$ using the same configuration as before. This quantity of measurements is common in medical applications. In our case, more measurements means more parameters to optimize, which makes more complex the optimization problem

(2.57). For a fixed N we have to compute $2N(2+q) = 260N$ parameters.

The results shown in Figure 2.24, Figure 2.25 and Figure 2.26 were computed on a discrete CPU, hence for the quantity of parameters we do not expect to find the best parameters in an appropriate time. We can notice that for $N > 14$ is not possible to improve the error in a stable way and that in two cases we have notable results. This means that there are parameters which can give good approximations but they are difficult to find with the hardware and algorithm used, nevertheless the approximations are good enough for applications.

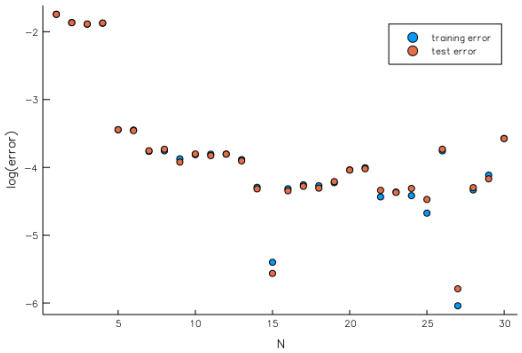


Figure 2.24: Approximation of the inverse EEG function. Average error in the training set $\Xi_{tr}$ and the test set $\Xi_{te}$ for different N.

| | N | $\mu_1$ | $\mu_2$ | $p_1$ | $p_2$ |
|---|---|---|---|---|---|
| Actual | | 0.117208 | 0.720746 | 0.481627 | 0.107150 |
| Estimation by $\varphi_*$ | 10 | 0.125857 | 0.421851 | 0.475224 | 0.097707 |
| Absolute Error | 10 | 0.008555 | 0.008346 | 0.006403 | 0.009442 |
| Estimation by $\varphi_*$ | 15 | 0.125618 | 0.703990 | 0.473414 | 0.130497 |
| Absolute Error | 15 | 0.008410 | 0.016759 | 0.008213 | 0.023347 |

Figure 2.25: Approximation of the inverse EEG function, $q = 128$. Comparison between actual and estimated parameter for an element of $\Xi_{te}$.

In conclusion, we have used the fast online stage of model order reduction techniques to construct a function that solves in real time the EEG inverse problem with good accuracy. This methodology is easily extendable to other inverse problems and can help improving computation time of slow iterative solvers.

| | N | $\mu_1$ | $\mu_2$ | $p_1$ | $p_2$ |
|---|---|---|---|---|---|
| Actual | | 0.139385 | 0.440241 | 1.938679 | 0.920604 |
| Estimation by $\varphi_*$ | 10 | 0.125857 | 0.421851 | 1.886519 | 0.885965 |
| Absolute Error | 10 | 0.013527 | 0.018390 | 0.052160 | 0.034639 |
| Estimation by $\varphi_*$ | 15 | 0.136246 | 0.440473 | 1.937756 | 0.920544 |
| Absolute Error | 15 | 0.003139 | 0.000232 | 0.000923 | 0.000060 |

Figure 2.26: Approximation of the inverse EEG function, $q = 128$. Comparison between actual and estimated parameter for an element of $\Xi_{te}$.

# Chapter 3

# jMOR

This chapter is about the exploration of jMOR, an open source package written in the Julia programming language [35]. The main purpose of this library is to provide *general* and *transparent* commands to users who want to apply model order reduction.

The transparent concept means that the final user does not have to understand how a function works or get overhelmed with its arguments. On the other hand, the general principle is about applying model order reduction to any equation independently of its implementation. We could call these two concepts together as a *black-box philosophy* [30].

In matter of execution times, a black-box library may affect negatively the efficiency of the reduced basis method, but it does not have to be the case in Julia. This is because there are two important paradigm implemented in the language: *multiple dispatch* [22] and *metaprogramming* [58].

A *dispatch* is the way a language choose which function of all its overloaded definitions is selected once is called. The traditional way is to dispatch according to the first argument of the function like it happens in C++ [151]. Instead, Julia looks for the quantity of arguments and their types individually, which is called multiple dispatch.

The first time a function is called with a specific set of arguments it is compiled at runtime with *just-in-time compilation* [15] and generates specialized *machine code* for this set. Later on, using runtime type inference, it can rival execution times of statically-compiled languages like C [36].

The other paradigm, metaprogramming, is the ability to treat the code as a data structure, which gives the possibility to manipulate code from within the language. This allows a program to generate code on its own at runtime and adapt the execution to concrete problems without specifying them manually. This means in the context of MOR that the code can adapt itself to any equation for the sake of improving computation times.

Furthermore, Julia also has good connection with C, Fortran and Python with almost no overhead. This gives the opportunity to use well developed packages in these languages for computing the snapshots. The library of choice for solving PDEs in jMOR is FEniCS [7].

FEniCS is a Python/C++ open source FEM library with good features like automated solution of variational problems, automated error control, extensive collection of finite elements, high performance, etc. FEniCS is not hardcoded inside jMOR, instead jMOR uses PyCall [94],

the Julia interface to Python. In the same way we can use any interface to access any other PDE solver.

This chapter focus on jMOR v0.1 and uses Python v2.7.12, FEniCS v2017.1 and Julia v0.5.2. We explicitly mark the explanation of a jMOR function with the symbol $*$ to express that it will not exist or will not be necessary in future versions of jMOR. One of the main objective of future versions is to automatize and simplify unimportant commands to final RB practitioners.

The sections in this chapter are devoted on how to use jMOR. Section 3.1 has a basic tutorial on how to apply POD to a Poisson's equation and Section 3.2 explains more deeply some features not exposed in the tutorial.

Even though no deep knowledge of Python, FEniCS or Julia is necessary to fully understand the content of this chapter, some basic knowledge is advised. Several good resources are: Python [143, 152]; FEniCS [104, 107]; Julia [19, 89].

## 3.1 POD on the Laplacian operator

This section shows in a tutorial-like style how to apply POD to a simple Poisson equation in jMOR. We present first the whole code for representing the equation in FEniCS and the required functions to make it compatible with jMOR. Thereafter, we explain every piece independently in more detail.

The equation to solve is:

$$\begin{cases} -\mu\Delta u &= \sin(\mu(x_1 + x_2)) & \text{in } \Omega = [0,1]^2, \\ u &= 0 & \text{on } \partial\Omega, \end{cases} \tag{3.1}$$

where $\mu \in [1,3]$.

Here is the code that represents this equation in FEniCS:

```
from fenics import *
import numpy as np
from scipy.sparse import csr_matrix

mesh = UnitSquareMesh(125, 125)
Vh = FunctionSpace(mesh, 'P', 1)

u0 = Constant(0)
def u0_boundary(x, on_boundary):
    return on_boundary

bc = DirichletBC(Vh, u0, u0_boundary)

u = TrialFunction(Vh)
v = TestFunction(Vh)

mu = Constant(1.0)

f = Expression('sin(mu*(x[0] + x[1]))', mu=mu, degree=2)
c = Constant(mu)
```

```
a = c*dot(grad(u), grad(v))*dx
L = f*v*dx

def affinedim():
    return np.array([1, 0])

def affinecoef(param):
    return [np.array([param]), 1]

def Afem(i=1):
    paramchange([1])

    a = c*dot(grad(u), grad(v))*dx
    mat = as_backend_type(assemble(a)).mat()
    return csr_matrix(mat.getValuesCSR()[::-1], shape = mat.size)

def bfem(param):
    paramchange(param)

    L = f*v*dx
    return assemble(L).array()

def Nh():
    return len(assemble(L))

def paramchange(param):
    param = param[0]

    global c
    global f
    c = Constant(param)
    f = Expression('sin(mu*(x[0]+x[1]))', mu=param, degree=2)

def eqsolve(param):
    paramchange(param)

    u1 = Function(Vh)
    a = c*dot(grad(u), grad(v))*dx
    L = f*v*dx
    solve(a == L, u1, bc)
    return u1

def offcoef(param):
```

The first three lines

```
from FEniCS import *
import numpy as np
from scipy.sparse import csr_matrix
```

import three basic libraries for the functioning of the code, in particular DOLFIN [108, 109] for FEniCS, NumPy [166] for efficient scientific computing in Python and the sparse CSR function

`scr_matrix`.

The code

```
mesh = UnitSquareMesh (125, 125)
Vh = FunctionSpace(mesh, 'P', 1)
```

generates a uniform mesh in $\Omega$ of 31250 triangles with $N_h = 15876$ and defines the globally continuous linear polynomial space $V_h$ on this mesh.

The boundary conditions are defined in the following way,

```
u0 = Constant(0)
def u0_boundary(x, on_boundary):
  return on_boundary

bc = DirichletBC(Vh, u0, u0_boundary)
```

where the function `u0_boundary` returns a special FEniCS variable that assess if a point is on $\partial\Omega$. The definition of `bc` declares that the equation will have a Dirichlet boundary condition with `u0` as the given function defined over it.

Before we define the variational formulation we need to write some placeholders for the functions of the trial and test space

```
u = TrialFunction(Vh)
v = TestFunction(Vh)
```

Now, we define $a(\cdot,\cdot)$ and $f(\cdot)$ with a temporary value for the parameter $\mu$

```
mu = Constant(1.0)

f = Expression('sin(mu*(x[0] + x[1]))', mu=mu, degree=2)
c = Constant(mu)

a = c*dot(grad(u), grad(v))*dx
L = f*v*dx
```

Notice that the FEniCS functions mimic those in the mathematical language.

The code dissected until now is the usual preamble for solving a PDE in FEniCS. For computing the solution the only thing missing is to call the solver:

```
u = Function(Vh)
solve(a == f, u, bc)
```

For using this preamble in jMOR we need to define some necessary functions to establish a communication with the solver, we call them *jMOR compatible functions*. The function definitions

are up to the user but they must satisfy some requirements in their input and output. The following list explains these requirements:

- `*affinedim()`: takes no arguments and returns an array of length two that indicates if the differential operator (first component of the array) or the functional (second component of the array) is affine. The possible values are $-1$ for expressing parameter independence, $0$ for not having an affine decomposition and $n$ for the affine dimension.

  For equation (3.1):

```
def affinedim():
  return np.array([1, 0])
```

- `affinecoef(param)`: takes the array `param` and returns an array of length two. The output has in the first component the coefficients of the operator affine representation as an array object. The second component is the same but for the functional.

  For equation (3.1):

```
def affinecoef(param):
  return [np.array([param]), 1]
```

- `Afem(params or i)`: if the differential operator is not affine, it returns the stiffness matrix for `param`. If it is affine, it returns the `i`-th stiffness matrix associated with the affine representation. The output has to be an sparse matrix in the *compressed sparse row format* (CSR).

  For equation (3.1):

```
def Afem(i=1):
 paramchange([1])

 a = c*dot(grad(u), grad(v))*dx
 mat = as_backend_type(assemble(a)).mat()
 return csr_matrix(mat.getValuesCSR()[::-1], shape = mat.size)
```

  Here `paramchange` is an auxiliary function for changing the parameter setting of the equation. The last two lines computes the stiffness matrix and convert it to the CSR format.

- `bfem(params or i)`: the same as `Afem` for the load vector. Here the output is an array.

  For equation (3.1):

```
def bfem(param):
 paramchange(param)
```

```
    L = f*v*dx
    return assemble(L).array()
```

- `*Nh()`: takes no arguments and return the dimension $N_h$ of $V_h$.

  For equation (3.1):

```
  def Nh():
    return len(assemble(L))
```

- `offcoef(param)`: returns the coefficients of the truth solution for the parameters `param`.

  For equation (3.1):

```
  def offcoef(param):
    u = eqsolve(param)

    return u.vector().array()
```

  The function `eqsolve` is an auxiliary function that calls the solver.

- `offsol(param)`: does not return anything but defines a global variable with the solution for the parameters `param`.

  For equation (3.1):

```
  def offsol(param):
    global offsolution

    offsolution = eqsolve(param)
```

- `offsolcoef(coef)`: does not return anything but defines a global variable with the FEM coefficients `coef`.

  For equation (3.1):

```
  def offsolcoef(coef):
   global offsolution

   offsolution = Function(Vh)
   offsolution.vector().set_local(coef)
```

- `offeval(x)`: after defining a global variable with `offsol` or `offsolcoef`, it returns the evaluation of that variable in `x`.

  For equation (3.1):

```
    def offeval(x):
      return offsolution(x)
```

- `Xh()`: takes no input and returns a CSR matrix representing the $\mathbf{X}_h$ matrix.

  For equation (3.1):

```
    def Xh():
     aXh = (u*v + dot(grad(u), grad(v)))*dx

     mat = as_backend_type(assemble(aXh)).mat()
     return csr_matrix(mat.getValuesCSR()[::-1], shape = mat.size)
```

Once we have all these functions defined we can compute a reduced basis with POD in jMOR. In the Julia console, also called REPL, write:

```
using jMOR
@femimport laplacian
params = 1:0.005:3
pod(params)
```

The first line

```
using jMOR
```

load the jMOR library and its dependencies.

If the definition of the PDE is in a file called laplacian.py, then the line

```
@femimport laplacian
```

import the Python code.

The sampling set $\Xi_s$ will be

```
params = 1:0.005:3
```

Finally, the line

```
pod(params)
```

runs the whole POD method and save the results in a HDF5 file [154]. The HDF5 format is a data model for representing complex objects with high performance.

All the offline steps in the POD method are done with the command `pod`. The default error tolerance is $1.0e-6$ and the dimension N is selected automatically for such error using the error estimates of Proposition 2. Further configuration of the error tolerance and the dimension N is explained in Section 3.2.

The following code shows the decay of the singular values and the Proposition 2 error estimate for different N until the rank r of the solution matrix. The results are in Figure 3.1.

```
using PyPlot

load(:r)
plot(singularvalues(r))
plot(poderrors(r))
```

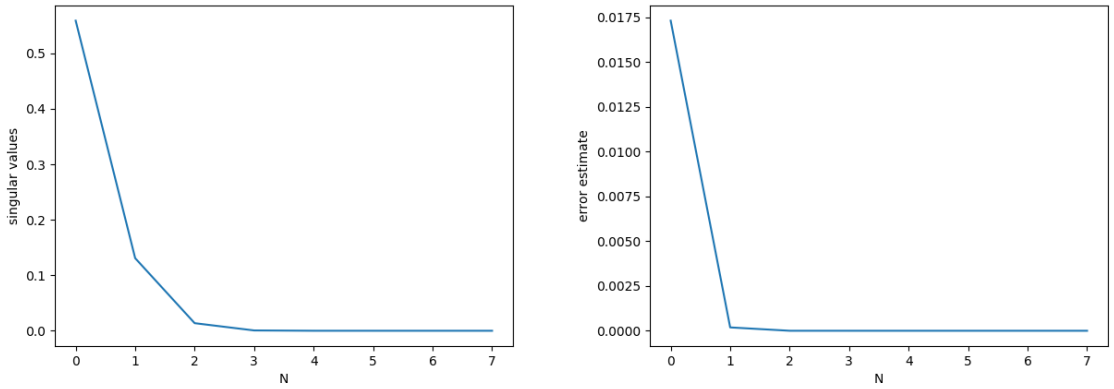


Figure 3.1: jMOR POD applied to the Poisson equation. Left picture: singular values for different N. Right picture: estimated POD error for different N.

The decay of the singular values for problem (3.1) is extremely fast and express the linear correlation between the snapshots. This gives the information that this parametric equation is reducible and suitable for the reduced basis approach.

Figure 3.2 shows the exact error in the $\mathbf{X}_h$ norm for different parameters and dimension N. The code for this figure is

```
onprepare()

plot(1:6, log(10, errors(1.485, 6)), color = "red", label = L"\mu = 1.485")
plot(1:6, log(10, errors(2.015, 6)), color = "green", label = L"\mu = 2.015")
plot(1:6, log(10, errors(2.748, 6)), color = "blue", label = L"\mu = 2.748")
plot(1:6, log(10, errors(1.719, 6)), color = "black", label = L"\mu = 1.719")
legend(loc="upper right",fancybox="true")
```

The line `onprepare()` calls a function that loads all the necessary mathematical structures to execute the online stage, this function call is only needed once per session. After `onprepare()`,

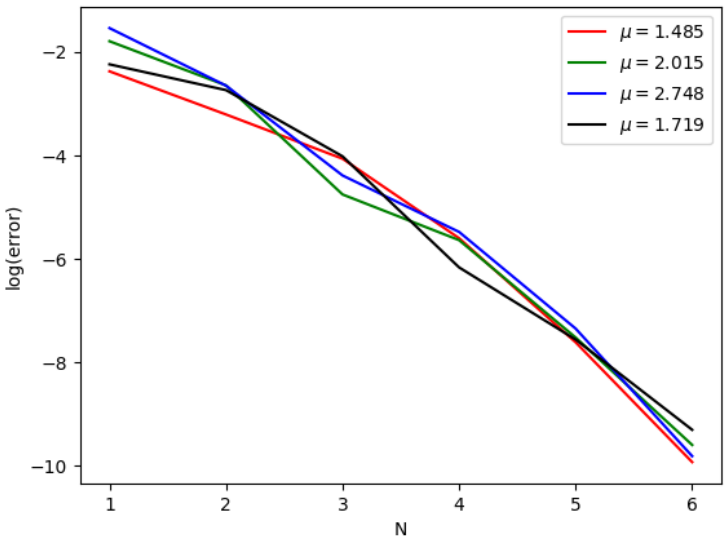


Figure 3.2: jMOR POD applied to the Poisson equation. Exact $\mathbf{X}_h$ error between truth solutions and RB solutions.

we can compute a reduced basis solution with the code

```
oncoef(param)
```

The function `oncoef(param)` gives the coefficients of the reduced basis, to convert it to the truth solution basis execute

```
ontofem(oncoef(param))
```

We can reuse all the code shown so far in other equations. For example, if we wish to apply POD to the equation:

$$\begin{cases} \text{curl curl } u + u &= (\sin(\mu(x_1+x_2)), \cos(\mu(x_1+x_2))) & \text{in } \Omega = [0,1]^2 \\ u &= 0 & \text{on } \partial\Omega, \end{cases} \tag{3.2}$$

we only have to change some lines. The definition of `Vh` is now a `VectorFunctionSpace`

```
Vh = VectorFunctionSpace(mesh, 'P', 1)
```

and regarding the elements in the variational formulation of (3.2) the code is

```
u0 = Expression(('0', '0'))
```

```
fmu = Expression(('sin(mu␣*␣(x[0]␣+␣x[1]))', 'cos(mu␣*␣(x[0]␣+␣x[1]))'), mu = mu)
a = (inner(curl(uh), curl(vh)) + inner(uh, vh))*dx
f = inner(fmu, vh)*dx
```

Other minor changes like in the jMOR compatible functions only have to take in account the vector nature of the solution u.

## 3.2 jMOR Functions Reference

In this section we describe the most useful functions in jMOR and how to use them. The specifications are for jMOR v0.1.

We first specify the types of the functions argument:

- `param`: float64 array
- `params`: array or matrix of float64 array
- `i,N`: integer
- `coef`: float64 array
- `xpoints,mupoints`: float64 array

Here are the functions description:

- `@femimport`: it is a Julia macro that loads a FEniCS code with the description of the equation to solve. Example:

```
@femimport name_of_file
```

```
@femimport name_of_file as reference_name
```

It creates a HDF5 file called `name_of_file.jld`. If we use the `as` statement, it changes the name to `reference_name`.

- `pod(params,error=1.e-6)`: computes the offline stage of the POD method for the parameter collection `params`. It saves in the HDF5 file important variables needed for the online stage or for further analysis.
  - `params`: parameters used for computing the offline stage
  - `S`: solution matrix
  - `ns`: quantity of parameters in `params`
  - `sigma`: singular values of the solution matrix

  - `N`: reduced basis dimension
  - `Nh`: full-order space dimension
  - `Afem1,Afem2...`: CSR stiffness matrices of the affine representation
  - `bfem1,bfem2...`: load vectors of the affine representation
  - `U`: reduced basis matrix
  - `Xh`: CSR matrix representing the $\mathbf{X}_h$ matrix
  - `r`: rank of the solution matrix

- `load(:var)`: it loads the variable `var` from the HDF5 file. The input is the name of the variable preceded by ":". Example:

```
load(:U)
```

- `offsol(param)`: computes the truth solution for the parameter `param` and returns the solution as a function.
- `offcoef(param)`: computes the truth solution for the parameter `param` and returns its coefficients.
- `offtime(params)`: estimates the execution time of the `pod` function.
- `offontime(param)`: compares the computation time between the reduced basis and truth solution for the parameter `param`.
- `errors(param,N)`: computes the exact error between the reduced basis and truth solution for the parameter `param` up to the dimension `N`.
- `poderrors(N)`: computes an error estimate for the POD method up to the dimension `N`.
- `singularvalues(N)`: returns the first `N` singular values of the solution matrix.
- `onbasis(i)`: returns the i-th reduced basis.
- `onprepare()`: prepares the mathematical structures to compute online solutions. It is necessary to call this function one time for every REPL section after using `@femimport`.
- `oncoef(param)`: computes the reduced basis coefficients for the parameter `param`.
- `onsol(param)`: returns the online solution as a function for the parameter `param`.
- `ontofem(coef)`: converts the online coefficients `coef` to the truth space coefficients.
- `*eimevals(g,xpoints,mupoints)`: computes the evaluation matrix of `g` for all combinations of `xpoints` and `mupoints`.
- `offeim(evals,maxiter,tol)`: computes the offline stage of the EIM algorithm using the evaluation matrix `evals` with maximum iteration `maxiter` and error tolerance `tol`.

- `oneim(param,g,Q,J,xs)`: computes the online stage of the EIM algorithm for the function `g` and the parameter `param`. The elements `Q,J,xs` are returned in this same order by the function `offeim`. Future versions of jMOR will use a Julia struct to group all these variables in one.

# Conclusions

Model order reduction methods (MOR) are good tools to achieve fast computational times while retaining good accuracy in many differential equation problems. A first general objective of this thesis is to provide new ideas for improving the effectiveness of reduced basis methods (RB). The main results in this direction are focused on the Fundamental Order Reduction (FOR) and new offline error estimators for standard RB algorithms.

In the FOR method we provide different theoretical estimators that show fast convergence rates. To exhibit one of their uses we present a simple example where the parameter space is unbounded and applied the estimators to find that a very low dimensional space is needed to solve the equation for any parameter. This method has other advantages like a fast online stage without the need of solving a linear system like in standard RB methods. Also there is a benefit on the non-linear construction of the basis, which could solve problems that RB algorithms may find difficult to solve.

Regarding RB methods, we also propose two offline estimators: Lipschitz and Chebychev offline estimators. Under the right conditions they can help obtaining good estimation of the error produced by the reduced model and at the same time reduce computational operations in the online stage.

A second result of this thesis has been the application of the FOR method and the new estimators EEG epilepsy equation. This equation, through the solution of an inverse problem, is useful for finding where epileptic seizures occur from electroencephalography readings.

The most notable difficulty this EEG equation brings is the strong singularity that it presents in the source term. We analyze two known methods for solving this equation. In the direct approach we show through numerical experiments and an example in 1D that we cannot apply RB methods effectively. On the other hand, for the subtraction approach we find that we can use MOR. We apply POD and use the offline estimators to certify good approximations without the need of computing the residual in the online stage. Afterwards we use FOR and obtain faster and better results when computing the solution.

Once we are able to compute the forward solution of the equation very fast we focus on solving the inverse problem. The typical approach in this case is to use iterative algorithms to find the source term and exploit the reduced model for obtaining fast computational times. In this work we instead propose an approximation of the inverse function using ideas from the universal approximation theory. Based on some theoretical results that apply to problem of this type, we showed that given a good approximation of the solution and enough measurement points on the boundary any PDE inverse problem where the input are discrete measurements on the boundary can be approximated by a neural network. We have applied this idea to the

EEG problem and obtained real-time results with good accuracy.

Finally, we have implemented a new MOR package in the Julia programming language to expand the availability of these techniques and at the same time provide a black-box approach to facilitate experiments. It is an open source library that any researcher or practitioner can use and contribute.

As future research paths we suggest to use the FOR method for other equations where POD and greedy algorithms work fine and compare the results. Given the theoretical convergence that FOR has, there is a huge potential to improve standard techniques, the only downside is that FOR is not as general as them. From another point view, FOR builds its basis with nonlinear operations, which could give good approximations where RB methods generally fail, i.e., when there is no linear correlation between the solutions. We also think is useful to expand the idea of the offline estimators presented here to high-dimensional problems and find better estimators using the smoothness of the residual.

It would be also interesting to run the EEG experiments on more realistic models like a sphere or a head with real measurement data and expand the theoretical results to a combination of delta functions as the source term. For the inverse part we would like to compare the obtained results with other similar solvers like Bayesian inference [123] or the lead-field approach [167] and to look for other neural network models that could approximate better the inverse function.

# Bibliography


[1] A. Abdulle and Y. Bai. "Adaptive Reduced Basis Finite Element Heterogeneous Multiscale Method". In: *Comput. Methods Appl. Mech. Eng.* **257**, pp. 203–220, 2013. (cited on p. 2)

[2] A. Abdulle and Y. Bai. "Reduced Basis Finite Element Heterogeneous Multiscale Method for High Order Discretizations of Elliptic Homogenization Problems". In: *J. Comput. Phys.* **231**(21), pp. 7014–7036, 2012. (cited on p. 2)

[3] B. Adcock, S. Brugiapaglia, and C. G. Webster. "Compressed Sensing Approaches for Polynomial Approximation of High-Dimensional Functions". In: 2017. arXiv: 1703.06987v2 [math.NA]. (cited on p. 45)

[4] K. Afanasiev and M. Hinze. "Adaptive Control of a Wake Flow using Proper Orthogonal Decomposition". In: *Shape Optimization and Optimal Design*. Ed. by J. Cagnol, M. Polis, and J. P. Zolesio. Vol. 216. Lecture notes in Pure and Appl. Math. Dekker, 1999, pp. 317–332. (cited on p. 15)

[5] R. Albanese and P. B. Monk. "The inverse source problem for Maxwell's equations". In: *Inverse Problems* **22**(3), pp. 1023–1035, 2006. (cited on p. 50)

[6] B. Almroth, P. Stern, and F. Brogan. "Automatic Choice of Global Shape Functions in Structural Analysis". In: *AAIA J* **16**(5), pp. 525–528, 1978. (cited on p. 2)

[7] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. "The FEniCS Project Version 1.5". In: *Archive of Numerical Software* **3**(100), 2015. (cited on pp. 6, 89)

[8] A. Alonso Rodriguez, J. Camaño, R. Rodriguez, and A. Valli. "A Posteriori Error Estimates for the Problem of Electrostatics with a Dipole Source". In: *Computers and Mathematics with Applications* **68**(4), pp. 464–485, 2014. (cited on pp. 50, 72)

[9] A. Alonso Rodriguez, J. Camaño, R. Rodriguez, and A. Valli. "Assessment of Two Approximation Methods for the Inverse Problem of Electroencephalography". In: *Int. J. of Numer. Anal and Model.* **1**(1), pp. 1–18, 2004. (cited on pp. 4, 50)

[10] D. Amsallem, J. Cortial, and C. Farhat. "Towards Real-Time Computational-Fluid-Dynamics-Based Aeroelastic Computations Using a Database of Reduced-Order Information". In: *AIAA J* **48**(9), pp. 2029–2037, 2010. (cited on p. 2)

[11] N. Aubry. "On the Hidden Beauty of the Proper Orthogonal Decomposition". In: *Theor. Comp. Fluid. Dyn.* **2**, pp. 339–352, 1991. (cited on p. 2)

[12] N. Aubry. “Preserving Symmetries in the Proper Orthogonal Decompostion”. In: *SIAM J. Sci. Comput.* **14**(2), pp. 483–505, 1993. (cited on p. 2)

[13] C. Audouze, F. De Vuyst, and P. B. Nair. “Reduced-Order Modeling of Parametrized PDEs using Time-Space Parameter Principal Component Analysis”. In: *Int. J. Numer. Methods Engrg.* **80**(10), pp. 1025–1057, 2009. (cited on p. 2)

[14] K. A. Awada, D. R. Jackson, J. T. Williams, D. R. Wilton, S. B. Baumann, and A. C. Papanicolaou. “Computational Aspects of Finite Element Modeling in EEG Source Localization”. In: *IEEE Trans Biomed Eng.* **44**(8), pp. 736–752, 1997. (cited on pp. 4, 50)

[15] J. Aycock. “A Brief History of Just-in-time”. In: *ACM Comput. Surv.* **35**(2), pp. 97–113, 2003. (cited on p. 89)

[16] A. Badia. “Inverse Source Problem in an Anisotropic Medium by Boundary Measurements”. In: *Inverse Problems* **21**, pp. 1487–1506, 2005. (cited on pp. 76, 77)

[17] A. Badia and M. Farah. “A Stable Recovering of Dipole Sources from Partial Boundary Measurements”. In: *Inverse Problems* **26**, 2010. (cited on pp. 76, 77)

[18] A. Badia and M. Farah. “Identification of Dipole Sources in an Elliptic Equation From Boundary Measurements: Application to the Inverse EEG Problem”. In: *J. Inv. Ill-Posed Problems* **14**(4), pp. 331–353, 2006. (cited on pp. 76, 77)

[19] I. Balbaert. *Getting started with Julia Programming Language*. Packt Publishing, 2015. (cited on p. 90)

[20] F. Ballarin, A. Sartori, and G. Rozza. *RBniCS*. `http://mathlab.sissa.it/rbnics`. (cited on p. 6)

[21] W. Bangerth, D. Davydov, T. Heister, G. Heltai L.and Kanschat, G. Kronbichler, M. Maier, B. Turcksin, and D. Wells. “The deal.II Library, Version 8.4”. In: *Journal of Numerical Mathematics* **24**, 2016. (cited on p. 6)

[22] S. Bansal. *Multiple Polymorphic Arguments in Single Dispatch Object Oriented Languages*. Ed. by S. Ranka, A. Banerjee, K. K. Biswas, S. Dua, P. Mishra, R. Moona, S.-H. Poon, and C.-L. Wang. 2010. (cited on p. 89)

[23] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. “An Empirical Interpolation Method: Application to Efficient Reduced-Basis Discretization of Partial Differencial Equations”. In: *Math. Acad. Sci. Paris* **339**(9), pp. 667–672, 2004. (cited on pp. 3, 45)

[24] A. Barret and G. Reddien. “On the Reduced Basis Method”. In: *Z. angew. Math. Mech.* **75**(7), pp. 543–549, 1995. (cited on p. 2)

[25] V. Barthelmann, E. Novak, and K. Ritter. “High dimensional polynomial interpolation on sparse grids”. In: *Advances in Computational Mathematics* **12**(4), pp. 273–288, 2000. (cited on p. 45)

[26] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klofkorn, M. Ohlberger, and Sander. O. “A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part I: Abstract Framework”. In: *Computing* **82**(2), pp. 103–119, 2008. (cited on p. 6)

[27] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klofkorn, M. Ohlberger, and Sander. O. *DUNE*. `https://www.dune-project.org/index.html`. (cited on p. 6)

[28] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 2000. (cited on p. 1)

[29] U. Baur, P. Benner, and L. Feng. *Model Order Reduction for Linear and Nonlinear Systems: a System-theoretic Perspective*. Preprint MPIMD/14-07. Available from `http://www.mpi-magdeburg.mpg.de/preprints/`. Max Planck Institute Magdeburg, Mar. 2014. (cited on p. 1)

[30] B. Beizer. *Black-Box Testing: Techniques for Functional Testing of Software and Systems*. Wiley, 1995. (cited on p. 89)

[31] P. Benner, S. Gugercin, and K. Willcox. *A Survey of Model Reduction Methods for Parametric Systems*. Preprint MPIMD/13-14. Available from `http://www.mpi-magdeburg.mpg.de/preprints/`. Max Planck Institute Magdeburg, Aug. 2013. (cited on p. 1)

[32] J. Bergstra, F. Bastien, O. Breuleux, and P. Lamblin. "Theano: Deep Learning on GPUs with Python". In: *Journal of Machine Learning Research*, 2011. (cited on p. 85)

[33] J. P. Berrut and Trefethen L. N. "Barycentric Lagrange Interpolation". In: *SIAM Rev.* **46**(3), pp. 501–517, 2004. (cited on p. 44)

[34] J. Bezanson, J. Chen, S. Karpinski, V. Shah, and A. Edelman. "Array Operators using Multiple Dispatch: a Design Methodology for Array Implementations in Dynamic Languages". In: *ARRAY'14 Proceedings of ACM SIGPLAN International Workshop on Libraries, Languages, and Compilers for Array Programming*. ACM, 2014, pp. 56–61. arXiv: `1407.3845 [cs.PL]`. (cited on p. 6)

[35] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. "Julia: A Fresh Approach to Numerical Computing". In: Nov. 2014. arXiv: `1411.1607 [cs.MS]`. (cited on pp. 6, 89)

[36] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman. "Julia: A Fast Dynamic Language for Technical Computing". In: Sept. 2012. arXiv: `1209.5145 [cs.PL]`. (cited on pp. 6, 89)

[37] S. Boyaval. "Reduced-Basis Approach for Homogenization Beyond the Periodic Setting". In: *Multiscale Model. Simul.* **7**(1), pp. 466–494, 2008. (cited on p. 2)

[38] S. Boyaval, C. L. Bris, Y. Maday, N. C. Nguyen, and A. T. Patera. "A Reduced Basis Approach for Variational Problems with Stochastic Parameters: Application to Heat Conduction with Variable Robin Coefficient". In: *Comput. Methods Appl. Mech. Eng.* **198**, pp. 3187–3206, 2009. (cited on p. 2)

[39] J. P. Boyd. "Large-Degree Asymptotics and Exponential Asymptotics for Fourier, Chebyshev and Hermite Coefficients and Fourier Transforms". In: *Journal of Engineering Mathematics* **63**(2), 2008. (cited on p. 44)

[40] T. Braconnier, M. Ferrier, J.-C. Jouhaud, M. Montagnac, and P. Sagaut. "Towards an Adaptive POD/SVD Surrogate Model for Aeronautic Design". In: *Computers and Fluids* **40**(1), pp. 195–209, 2011. (cited on p. 10)

[41] C. Brezinski. "Pade-type Approximation and General Orthogonal Polynomials". In: *Intern. Series of Numer. Anal.* **50**, 1980. (cited on p. 32)

[42] T. Bui-Thanh, M. Damodaran, and K. Willcox. “Proper Orthogonal Decomposition Extensions for Parametric Applications in Transonic Aerodynamics”. In: *AIAA Journal*, 2003. (cited on p. 2)

[43] H. J. Bungartz and M. Griebel. “Sparse Grids”. In: *Acta Numerica* **13**, pp. 147–269, 2004. (cited on p. 10)

[44] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics*. Springer, 2007. (cited on p. 1)

[45] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods: Fundamentals in Single Domains*. Springer, 2006. (cited on p. 1)

[46] C. Canuto, T. Tonn, and K. Urban. “A Posteriori Error Analysis of the Reduced Basis Method for Nonaffine Parametrized Nonlinear PDEs”. In: *SIAM J. Numer. Anal.* **47**(3), pp. 2001–2022, 2009. (cited on p. 2)

[47] B. S. Chang and D. H. Lowenstein. “Epilepsy”. In: *New England Journal of Medicine* **349**(13), pp. 1257–1266, 2003. (cited on p. 50)

[48] T. Chen and H. Chen. “Universal Approximation to Nonlinear Operators by Neural Networks with Arbitrary Activation Functions and Its Application to Dynamical Systems”. In: *IEEE Transactions on Neural Networks* **6**(4), 1995. (cited on pp. 79, 80)

[49] Y. Chen, J. S Hesthaven, Y. Maday, and J. Rodriguez. “Certified Reduced Basis Methods and Output Bounds for the Harmonic Maxwell’s Equations”. In: *SIAM J. Sci. Comput.* **32**(2), pp. 970–996, 2010. (cited on p. 2)

[50] Y. Chen, J. S Hesthaven, Y. Maday, and J. Rodriguez. “Improved Successive Constraint Method Based A Posteriori Error Estimate for Reduced Basis Approximation of 2d Maxwell’s Problem”. In: *ESAIM Math. Model. Numer. Anal.* **49**(6), pp. 1099–1116, 2009. (cited on p. 2)

[51] Y. Chen, J. S Hesthaven, Y. Maday, J. Rodriguez, and X. Xhu. “Certified Reduced Basis Method for Electromagnetic Scattering and Radar Cross Section Estimation”. In: *Comput. Methods Appl. Mech. Eng.* **233**, pp. 92–108, 2012. (cited on p. 2)

[52] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. 2nd ed. SIAM: Society for Industrial and Applied Mathematics, 2002. (cited on pp. 1, 7, 71)

[53] A. Cohen and R. DeVore. “Approximation of high-dimensional parametric PDEs”. In: *Acta Numerica* **24**, pp. 1–159, 2015. (cited on pp. 7, 20, 24, 26)

[54] H. Cohen, F. Rodriguez Villegas, and D. Zagier. “Convergence Acceleration of Alternating Series”. In: *Experimental Mathematics* **9**(1), 2000. (cited on p. 32)

[55] R. Courant and D. Hilbert. *Methods of Mathematical Physics*. Interscience Publisher, 1953. (cited on p. 19)

[56] Henggang Cui, Hao Zhang, Gregory R. Ganger, Phillip B. Gibbons, and Eric P. Xing. “GeePS: Scalable deep learning on distributed GPUs with a GPU-specialized parameter server”. In: *Proceedings of the Eleventh European Conference on Computer Systems*, 2016. (cited on p. 85)

[57] N. N. Cuong, K. Veroy, and A. T. Patera. “Certified Real-Time Solution of Parametrized Partial Differential Equations”. In: *Handbook of Materials Modeling*. Springer, 2005, pp. 1529–1564. (cited on p. 2)

[58] K. Czarnecki and U. Eisenecker. *Generative Programming: Methods, Tools, and Applications*. Addison-Wesley Professional, 2000. (cited on pp. 6, 89)

[59] L. Dede. “Reduced Basis Method and A Posteriori Error Estimation for Parametrized Linear Quadratic Optimal Control Problems”. In: *SIAM J. Sci. Comput.* **32**(2), pp. 997–1019, 2010. (cited on p. 2)

[60] A. S Demidov. *Generalized Functions in Mathematical Physics: Main Ideas and Concepts*. Nova Science Publishers, Inc., 2013. (cited on p. 50)

[61] R. DeVore. “Nonlinear Approximation”. In: *Acta Numerica* (2), pp. 51–150, 1998. (cited on p. 36)

[62] M. Dihlmann, M. Drohmann, B. Haasdonk, M. Ohlberger, and M. Schaefer. *RBmatlab*. `http://www.ians.uni-stuttgart.de/MoRePaS/software/rbmatlab/0.11.04/doc/`. (cited on p. 5)

[63] M. Drohmann, B. Haasdonk, S. Kaulmann, and M. Ohlberger. “A Software Framework for Reduced Basis Methods using Dune-RB and RBmatlab”. In: *Advances in DUNE*. Ed. by A. Dedner, B. Flemisch, and R. Klofkorn. Springer-Verlag, 2012, pp. 77–88. (cited on pp. 5, 6)

[64] J. Edmonds. “Matroids and the Greedy Algorithm”. In: *Math. Program.* **1**(1), pp. 127–136, 1971. (cited on p. 2)

[65] J. L. Eftang, Patera A. T., and E. M. Ronquist. “An hp Certified Reduced Basis Method for Parametrized Parabolic Partial Differential Equations”. In: *Mathematical and Computer Modelling of Dynamical Systems* **17**(4), pp. 395–422, 2011. (cited on p. 10)

[66] M. Eiermann. “On the Convergence of Pade-type Approximants to Analytic Functions”. In: *J. Comput. Appl. Math.* **10**(2), pp. 219–227, 1984. (cited on p. 32)

[67] L. C. Evans. *Partial Differential Equations*. 2nd ed. American Mathematical Society, 2010. (cited on p. 1)

[68] L. Fei-Fei, A. Karpathy, T. Leung, S. Shetty, R. Sukthankar, and G. Toderici. “Large-Scale Video Classification with Convolutional Neural Networks”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014. (cited on p. 5)

[69] J. P. Fink and W. C. Rheinboldt. “On the Error Behavior of the Reduced Basis Technique for Nonlinear Finite Element Approximations”. In: *Z. angew. Math. Mech.* **63**(1), pp. 21–28, 1983. (cited on p. 2)

[70] R. Fox and H. Miura. “An Approximate Analysis Technique for Design Calculations”. In: *AAIA J* **9**(1), pp. 177–179, 1971. (cited on p. 2)

[71] A.-L. Gerner and K. Veroy. “Certified Reduced Basis Methods for Parametrized Saddle Point Problems”. In: *SIAM J. Sci. Comput.* **34**(5), A2812–A2836, 2012. (cited on p. 2)

[72] G. H. Golub and C. F. van Loan. *Matrix Computations*. 4th ed. Johns Hopkins University Press, 2012. (cited on p. 13)

[73] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. (cited on p. 86)

[74] M. A. Grepl. "Reduced-Basis Approximation and A Posteriori Error Estimation for Parabolic Partial Differential Equations". PhD thesis. Massachusetts Institute of Technology, 2005. (cited on p. 2)

[75] M. A Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera. "Efficient Reduced-Basis Treatment of Nonaffine and Nonlinear Partial Differential Equations". In: *ESAIM. Math. Model. Numer. Anal.* **41**(3), pp. 575–605, 2007. (cited on pp. 2, 45)

[76] M. A. Grepl and A. T. Patera. "A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations". In: *ESAIM. Math. Model. Numer. Anal.* **39**(1), pp. 157–181, 2005. (cited on p. 2)

[77] D. J. Griffiths. *Introduction to Electrodynamics*. 4th ed. Pearson, 2012. (cited on p. 1)

[78] D. Hartmann. "Computer Simulation for Contemporary Problems in Civil Engineering". In: *Computing in Civil Engineering (2005)*. 2005, pp. 1–16. (cited on p. 1)

[79] M. Hassoun. *Fundamentals of Artificial Neural Networks*. MIT Press, 1995. (cited on p. 5)

[80] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2009. (cited on pp. 85, 86)

[81] K. He, S. Ren, J. Sun, and X. Zhang. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016. (cited on p. 5)

[82] S. He and V. G. Romanov. "Identification of Dipole Sources in a Bounded Domain for Maxwell's Equations". In: *Wave Motion* **28**(1), pp. 25–40, 1998. (cited on p. 50)

[83] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer, 2016. (cited on pp. 1, 6, 7, 11)

[84] J. S. Hesthaven, B. Stamm, and S. Zhang. "Efficient Greedy Algorithms for High-Dimensional Parameter Spaces with Applications to Empirical Interpolation and Reduced Basis Methods". In: *ESAIM: M2AN* **48**(1), pp. 259–283, 2014. (cited on p. 10)

[85] N. J. Higham. "The Numerical Stability of Baricentric Lagrange Interpolation". In: *IMA J. Numer. Anal.* **24**(4), pp. 547–556, 2004. (cited on p. 44)

[86] J. Huchette, M. Lubin, and C. Petra. "Parallel Algebraic Modeling for Stochastic Optimization". In: *HPTCDL'14 Proceedings of the 1st Workshop on High Performance Technical Computing in Dynamic Languages*. ACM, 2014, pp. 29–35. ANL: `MCS-P5181-0814`. (cited on p. 6)

[87] D. B. P. Huynh, D. Knezevic, and A. Patera. "Certified Reduced Basis Model Validation: a Frequentistic Uncertainty Framework". In: *Comput. Methods Appl. Mech. Eng.* **201**, pp. 13–24, 2012. (cited on p. 2)

[88] D. B. P. Huynh, N. C. Nguyen, A. T. Patera, and G. Rozza. *rbMIT*. `http://augustine.mit.edu/methodology/methodology_rbMIT_System.htm`. (cited on p. 5)

[89] *Introducing Julia*. https://en.wikibooks.org/wiki/Introducing_Julia. (cited on p. 90)

[90] V. Isakov. *Inverse Problems for Partial Differential Equations*. 2nd ed. Springer, 2005. (cited on pp. 1, 77)

[91] Adler J. and Oktem O. "Solving ill-posed inverse problems using iterative deep neural networks". In: *Inverse Problems* **33**(12), 2017. (cited on p. 5)

[92] N. Jakovcevic Stor, I. Slapnicar, and J. L. Barlow. "Forward Stable Eigenvalue Decomposition of Rank-One Modifications of Diagonal Matrices". In: *Linear Algebra and Its Applications* **487**, pp. 301–315, 2015. (cited on p. 6)

[93] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. "Deep Convolutional Neural Network for Inverse Problems in Imaging". In: *IEEE Transactions on Image Processing* **26**(9), pp. 4509–4522, 2017. (cited on p. 5)

[94] S. G. Johnson. *PyCall*. https://github.com/JuliaPy/PyCall.jl. (cited on p. 89)

[95] N. Jung, B. Haasdonk, and D. Kroner. "Reduced Basis Method for Quadratically Nonlinear Transport Equations". In: *Int. J. Comput. Sci. Math.* **2**(4), pp. 334–353, 2009. (cited on p. 2)

[96] M. Kahlbacher and S. Volkwein. "Galerkin Proper Orthogonal Decomposition Methods for Parameter Dependent Elliptic Systems". In: *Discuss. Math. Differ. Incl. Control Optim.* **27**(1), pp. 95–117, 2007. (cited on p. 2)

[97] M. Kahlbacher and S. Volkwein. "POD A-Posteriori Error Based Inexact SQP Method for Bilinear Elliptic Optimal Control Problems". In: *Math. Model. Numer. Anal.* **46**(2), pp. 491–511, 2012. (cited on p. 2)

[98] S. Kaulmann, M. Ohlberger, and B. Haasdonk. "A New Local Reduced Basis Discontinuous Galerkin Approach for Heterogeneous Multiscale Problems". In: *C.R. Math.* **349**(23), pp. 1233–1238, 2011. (cited on p. 2)

[99] A. Kolmogorov. "Uber die beste Annaherung von Funktionen einer gegebenen Funktionenklasse". In: *Ann. of Math* **37**, pp. 107–110, 1936. (cited on p. 19)

[100] A. N. Kolmogorov and S. V. Fomin. *Elements of the Theory of Functions and Functional Analysis*. Dover Publications, 1999. (cited on p. 23)

[101] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: 2012. (cited on p. 5)

[102] K. Kunisch and S. Volkwein. "Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fluid Dynamics". In: *SIAM J. Numer. Anal.* **40**(2), pp. 492–515, 2003. (cited on p. 2)

[103] K. Kunisch and S. Volkwein. "Galerkin Proper Orthogonal Decomposition Methods for a Parabolic Problems". In: *Num. Math.* **90**, pp. 117–148, 2001. (cited on p. 2)

[104] H. P. Langtangen. *FEniCS Tutorial*. http://hplgit.github.io/fenics-tutorial/doc/web/index.html. (cited on p. 90)

[105] M.-Y. L. Lee. “Estimation of the Error in the Reduced Basis Method Solution of Differential Algebraic Equation Systems”. In: *SIAM J. Numer. Anal.* **28**(2), pp. 512–528, 1991. (cited on p. 2)

[106] S. Lew, C. H. Wolters, T. Dierkes, C. Roer, and R. S Macleod. “Accuracy and Run-Time Comparison for Different Potential Approaches and Iterative Solvers in Finite Element Method Based EEG Source Analysis”. In: *Appl. Numer. Math.* **59**(8), pp. 1970–1988, 2009. (cited on pp. 4, 50)

[107] A. Logg, K. A. Mardal, and G. Wells. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*. Springer, 2012. (cited on p. 90)

[108] A. Logg, A. N. Wells, and J. Hake. “DOLFIN: a C++/Python Finite Element Library”. In: *Automated Solution of Differential Equations by the Finite Element Method, Volume 84 of Lecture Notes in Computational Science and Engineering*. Ed. by A. Logg, K.-A. Mardal, and G. N. Wells. Springer, 2012. Chap. 10. (cited on p. 91)

[109] A. Logg and G. N. Wells. “DOLFIN: Automated Finite Element Computing”. In: *ACM Transactions on Mathematical Software* **37**(2), 2010. (cited on p. 91)

[110] M. Lubin and I. Dunning. “Computing in Operations Research Using Julia”. In: *INFORMS Journal on Computing* **27**(2), pp. 238–248, 2015. (cited on p. 6)

[111] Y. Maday, N. C. Nguyen, A. T. Patera, and G. S. H. Pau. “A General Multipurpose Interpolation Procedure: The Magic Points”. In: *Commun. Pure Appl. Anal.* **8**(1), pp. 383–404, 2009. (cited on pp. 45, 47)

[112] Y. Maday and B. Stamm. “Locally Adaptive Greedy Approximations for Anisotropic Parameter Reduced Basis Spaces”. In: *SIAM J. Sci. Comput* **35**(6), A2417–A2441, 2013. (cited on p. 10)

[113] Y. Maday, Patera A. T., and G. Turinici. “A priori convergence theory for reduced-basis approximations of single-parametric elliptic partial differential equations”. In: *J. Sci. Comput.* **17**, pp. 437–446, 2002. (cited on p. 2)

[114] Y. Maday, Patera A. T., and G. Turinici. “Global a priori convergence theory for reduced-basis approximations of single-parameter symmetric coercive elliptic partial differential equations”. In: *C. R. Math. Acad. Sci. Paris* **335**, pp. 289–294, 2002. (cited on p. 2)

[115] J. C. Mason. “Near-Best Multivariate Approximation by Fourier Series, Chebyshev Series and Chebyshev Interpolation”. In: *Journal of Approximation Theory* **28**(4), pp. 349–358, 1980. (cited on p. 44)

[116] J. C. Mason and D. C. Handscomb. *Chebyshev Polynomials*. CRC Press, 2003. (cited on pp. 43, 44)

[117] H. N. Mhaskar and N. Hahm. “Neural Networks for Functional Approximation and System Identification”. In: *Neural Computation* **9**(1), pp. 143–159, 1997. (cited on p. 79)

[118] R. Milk, S. Rave, and F. Schindler. *pyMOR*. `http://pymor.org/`. (cited on p. 6)

[119] J. C. Misra. *Biomathematics: Modelling and Simulation*. World Scientific Publishing Company, 2006. (cited on p. 1)

[120] F. Moukalled, L. Mangani, and M. Darwish. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction With OpenFOAM and Matlab*. Springer-Verlag, 2015. (cited on p. 1)

[121] N. C. Nguyen. "A Multiscale Reduced-Basis Method for Parametrized Elliptic Partial Differential Equations with Multiple Scales". In: *J. Comput. Phys.* **227**(23), pp. 9807–9822, 2008. (cited on p. 2)

[122] N. C. Nguyen, G. Rozza, D. B. P. Huynh, and A. T. Patera. "Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Parabolic PDEs; Application to Real-time Bayesian Parameter Estimation". In: *Computational Methods for Large Scale Inverse Problems and Uncertainty Quantification*. Ed. by I. Biegler, G. Biros, O. Ghattas, M. Heinkenschloss, D. Keyes, B. Mallick, L. Tenorio, B. van Bloemen Waanders, and K. Willcox. Wiley, 2012. (cited on p. 2)

[123] N. Nguyen, B. C. Khoo, and K. Willcox. "Model order reduction for Bayesian approach to inverse problems". In: *Asia Pacific Journal on Computational Engineering*, 2014. (cited on p. 102)

[124] E. Niedermeyer and F. Lopes da Silva. *Niedermeyer's Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Ed. by D. L. Schomer and F. Lopes da Silva. 6th ed. LWW, 2010. (cited on p. 50)

[125] M. Ohlberger and S. Rave. "Reduced Basis Methods: Success, Limitations and Future Challenges". In: *Proceedings of ALGORITMY 2016*, pp. 1–12, 2016. (cited on p. 26)

[126] S. Olver and A. Townsend. "A Practical Framework for Infinite-Dimensional Linear Algebra". In: *HPTCDL'14 Proceedings of the 1st Workshop on High Performance Technical Computing in Dynamic Languages*. ACM, 2014, pp. 57–62. arXiv: `1409.5529 [math.NA]`. (cited on p. 6)

[127] A. T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations*. MIT Pappalardo Graduate Monographs in Mechanical Engineering, 2007. (cited on p. 5)

[128] D. A. di Pietro and A. Ern. *Mathematical Aspects of Discontinuous Galerkin Methods*. Springer, 2012. (cited on p. 5)

[129] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P Flannery. *Numerical Recipes: The Art of Scientific Computing*. 3rd ed. Cambridge University Press, 2007. (cited on p. 77)

[130] C. Prudhomme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici. "Reliable Real-Time Solution of Parametrized Partial Differential Equations: Reduced-Basis Output Bound Methods". In: *J. Fluids Eng* **124**(1), pp. 70–80, 2002. (cited on p. 2)

[131] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*. Springer, 2016. (cited on pp. 1, 3, 7, 11, 14, 15, 26–29, 41, 47)

[132] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer, 1994. (cited on pp. 1, 7, 8, 22, 71)

[133] I. Roulstone and J. Norbury. *Invisible in the Storm: The Role of Mathematics in Understanding Weather*. Princeton University Press, 2013. (cited on p. 1)

[134] G Rozza. "Real Time Reduced Basis Techniques for Arterial Bypass Geometries". In: *Computational Fluid and Solid Mechanics-Third MIT Conference on Computational Fluid and Solid Mechanics*. Elsevier, 2005, pp. 1283–1287. (cited on p. 2)

[135] G. Rozza. "Reduced-Basis Methods for Elliptic Equations in Sub-Domains with a Posteriori Error Bounds and Adaptivity". In: *Appl. Numer. Math.* **55**(4), pp. 403–424, 2005. (cited on p. 2)

[136] G. Rozza. "Shape Design by Optimal Flow Control and Reduced Basis Techniques: Applications to Bypass Configurations in Haemodynamics". PhD thesis. EPFL Lausanne, Switzerland, 2005. (cited on p. 2)

[137] 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". In: *Archives of Computational Methods in Engineering* **15**(3), pp. 229–275, 2008. (cited on pp. 2, 10)

[138] G. Rozza and K. Veroy. "On the Stability of the Reduced Basis Method for Stokes Equations in Parametrized Domains". In: *Comput. Methods Appl. Mech. Eng.* **196**(7), pp. 1244–1260, 2007. (cited on p. 2)

[139] M. H. Sadd. *Elasticity: Theory, Applications, and Numerics*. 3rd ed. Academic Press, 2014. (cited on p. 1)

[140] H. E. Salzer. "Lagrangian Interpolation at the Chebychev Points $X_{n,\nu} \equiv \cos(\nu\pi/n), \nu = 0(1)n$; some unnoted advantages". In: *Comput. J.* **15**, pp. 156–159, 1972. (cited on p. 44)

[141] J. Sarvas. "Basic Mathematical and Electromagnetic Concepts of the Biomagnetic Inverse Problem". In: *Phys. Med. Biol.* **32**(1), pp. 11–22. (cited on p. 50)

[142] S. Sen. "Reduced-basis approximation and a posteriori error estimation for many-parameter heat conduction problems". In: *Numer. Heat Tr.* **54**, pp. 369–389, 2008. (cited on p. 2)

[143] Z. A. Shaw. *Learn Python The Hard Way*. `http://learnpythonthehardway.org/`. (cited on p. 90)

[144] D. Silver et al. "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm". In: 2017. arXiv: `1712.01815 [cs.AI]`. (cited on p. 5)

[145] D. Silver et al. "Mastering the game of Go with deep neural networks and tree search". In: *Nature* **529**, pp. 484–489, 2016. (cited on p. 5)

[146] D. Silver et al. "Mastering the Game of Go Without Human Knowledge". In: *Nature* **550**, pp. 354–359, 2017. (cited on p. 5)

[147] L. Sirovich. "Analysis of Turbulent Flows by Means of the Empirical Eigenfunctions". In: *Fluid Dynam. Res.* **8**(1), pp. 85–100, 1991. (cited on p. 2)

[148] L. Sirovich. "Turbulence and the Dynamics of Coherent Structures, part i". In: *Quart. Appl. Math* **45**(3), pp. 561–571, 1987. (cited on p. 2)

[149] R. M. Slevinsky and S. Olver. “On the Use of Conformal Maps for the Acceleration of Convergence of the Trapezoidal Rule and Sinc Numerical Methods”. In: 2014. arXiv: 1406.3320 [math.NA]. (cited on p. 6)

[150] J. S. Spall. *Introduction to Stochastic Search and Optimization*. Wiley-Interscience, 2003. (cited on p. 85)

[151] B. Stroustrup. *Programming: Principles and Practice Using C++*. 2nd ed. Addison-Wesley Professional, 14. (cited on p. 89)

[152] C. H. Swaroop. *A Byte of Python*. http://python.swaroopch.com/. (cited on p. 90)

[153] McCann M. T., Jin K. H., and Unser M. “A Review of Convolutional Neural Networks for Inverse Problems in Imaging”. In: 2017. arXiv: 1710.04011 [eess.IV]. (cited on p. 5)

[154] The HDF Group. *Hierarchical Data Format, version 5*. http://www.hdfgroup.org/HDF5/. 1997-2017. (cited on p. 95)

[155] T. Tonn and K. and Urban. “Optimal Control of Parameter-Dependent Convection-Diffusion Problems around Rigid Bodies”. In: *SIAM J. Sci. Comput.* **32**(3), pp. 1237–1260, 2010. (cited on p. 2)

[156] T. Tonn, K. Urban, and S. Volkwein. “Comparison of the Reduced-Basis and Pod A Posteriori Error Estimators for an Elliptic Linear-Quadratic Optimal Control Problem”. In: *Math. Comput. Model. Dyn. Syst.* **17**(4), pp. 355–369, 2011. (cited on p. 2)

[157] A. Townsend and L. N. Trefethen. “An Extension of Chebfun to Two Dimensions”. In: *SIAM J. Sci. Comput.* **35**(6), pp. 495–518, 2013. (cited on pp. 44, 64)

[158] A. Townsend, T. Trogdon, and S. Olver. “Fast Computation of Gauss Quadrature Nodes and Weights on the Whole Real Line”. In: 2014. arXiv: 1410.5286 [math.NA]. (cited on p. 6)

[159] L. N. Trefethen. *Approximation Theory and Approximation Practice*. SIAM, 2012. (cited on pp. 41, 43)

[160] L. N. Trefethen. “Multivariate Polynomial Approximation in the Hypercube”. In: *Proceedings of the American Mathematical Society* **145**(11), pp. 4837–4844, 2017. (cited on pp. 44, 54)

[161] M. Udell, K. Mohan, D. Zeng, J. Hong, S. Diamond, and S. Boyd. “Convex Optimization in Julia”. In: *HPTCDL’14 Proceedings of the 1st Workshop on High Performance Technical Computing in Dynamic Languages*. ACM, 2014, pp. 18–28. arXiv: 1410.4821 [math.OC]. (cited on p. 6)

[162] J. L. Valerdi Cabrera. *jMOR*. https://github.com/valerdi/jMOR.jl. (cited on p. 6)

[163] A. Valli. “Solving an electrostatics-like problem with a current dipole source by means of the duality method”. In: *Applied Mathematics Letters* **25**(10), pp. 1410–1414, 2012. (cited on pp. 4, 50, 71)

[164] K. Veroy and A. T. Patera. “Certified Real-Time Solution of the Parametrized Steady Incompressible Navier-Stokes Equations: Rigorous Reduced-Basis A Posteriori Error Bounds”. In: *Int. J. Numer. Meth. Fluids* **47**(8), pp. 773–788, 2005. (cited on p. 2)

[165] K. Veroy, C. Prudhomme, D. Rovas, and A. Patera. “A Posteriori Error Bounds for Reduced-Basis Approximation of Parametrized Noncoercive and Nonlinear Elliptic Partial Differential Equations”. In: *Proceedings of the 16th AIAA Computational Fluid Dynamics Conference*. Vol. 3847. 2003. (cited on p. 2)

[166] S. Walt, S. C. Colbert, and G. Varoquaux. “The NumPy Array: A Structure for Efficient Numerical Computation”. In: *Computing in Science and Engineering* **13**(2), pp. 22–30, 2011. (cited on p. 91)

[167] D. Weinstein, L. Zhukov, and C. Johnson. “Lead-field Bases for Electroencephalography Source Imaging”. In: *Annals of Biomedical Engineering* **28**, pp. 1059–1065, 2000. (cited on p. 102)

[168] C. H. Wolters, H. Kostler, C. Moller, J. Hardtlein, L. Grasedyck, and W. Hackbusch. “Numerical Mathematics of the Subtraction Method for the Modeling of a Current Dipole in EEG Source Reconstruction Using Finite Element Head Models”. In: *SIAM Journal on Scientific Computing* **30**(1), pp. 24–45, 2008. (cited on p. 50)

[169] C. H. Wolters, H. Kostler, C. Moller, J. Hardtlein, L. Grasedyck, and W. Hackbusch. “Numerical mathematics of the subtraction method for the modeling of a current dipole in EEG source reconstruction using finite element head models”. In: *SIAM J. Sci. Comput.* **30**, pp. 24–45, 2007. (cited on p. 4)

[170] W. Zhang and N. J. Higham. *Matrix Depot: An Extensible Test Matrix Collection for Julia*. Tech. rep. 2015.118. Manchester Institute for Mathematical Sciences, The University of Manchester, Dec. 2015. URL: `http://eprints.ma.man.ac.uk/2426`. (cited on p. 6)