



imus

Programa de Doctorado "Matemáticas"

PhD Dissertation

---

**Some applications of  
reduced order modelling  
to slow-fast dynamical systems,  
turbulence models and elliptic PDEs**

---

*Author*

*Alejandro Bandera Moreno*

*Supervisors*

*Tomás Chacón Rebollo*

*Soledad Fernández García*

*Macarena Gómez Márquez*

Sevilla, 2024



*A mis abuelas.*



---

## AGRADECIMIENTOS

---

En primer lugar, quiero agradecer su trabajo a mis tutores de tesis: Macarena, Soledad y Tomás. Voy a hacer un breve repaso a nuestra historia conjunta hasta ahora. Todo este viaje comenzó en el curso 2016/17, cuando le propuse a Tomás ser su alumno interno para estudiar sistemas caóticos. Él me aceptó y me presentó a Soledad, con la que estudié bifurcaciones en ecuaciones escalares autónomas. En el curso siguiente, continué como alumno interno de Soledad, a la que se le unió Macarena, para estudiar sistemas de ecuaciones diferenciales ordinarias con múltiples escalas de tiempo. El curso 2018/19 fue mi último año de grado, y defendí mi TFG sobre órbitas canard en sistemas lineales a trozos, tutorizado por Soledad y Macarena. Además, durante ese año disfruté de una Beca de Colaboración en el departamento. En el año del máster ya empecé a meter la cabeza en los modelos reducidos, defendiendo mi TFM sobre métodos de orden reducido para ecuaciones diferenciales. Después han venido cuatro años para la elaboración de esta tesis doctoral. Como veis, son bastantes años los que llevamos trabajando juntos y que espero que sean muchos más en el futuro.

Debo hacer también una mención especial a mis profesoras de Matemáticas y Física durante el bachillerato, en particular a Natividad Cortés, mi tutora durante el último año preuniversitario. Gracias a ellas empezo este largo camino que hoy alcanza una nueva meta.

Je voudrais également remercier les personnes avec qui j'ai travaillé lors de mes séjours académiques. Simon Brillet et Mejdi Azaiez pour mes séjours à Bordeaux, et Stéphane Menozzi et Alexandre Vidal pour mon séjour à Paris. J'ai hâte de continuer à travailler avec vous à l'avenir.

A todos los que me han precedido como estudiantes de doctorado dentro del grupo de investigación M2S2M, si he podido ver lejos es porque me he subido a vuestros hombros. Espero que mi trabajo le sirva tanto a los que vengan detrás, como a mí me ha servido el vuestro. También a todos los compañeros del departamento, nos hacéis sentir como uno más desde el principio y nos hacéis la vida un poco más fácil.

Por supuesto, quiero mencionar a mis compañeros del Colegio Mayor Hernando Colón durante los cinco años de carrera, con vosotros la vida universitaria alcanzó un significado mayor. Me llevo muy buenas experiencias y amistades que no olvidaré

nunca. No puedo terminar este párrafo sin agradecer además al equipo de dirección del Colegio Mayor, del que formé parte durante mis dos primeros años de tesis. Gracias a la oportunidad que me disteis, he mejorado en lo personal y en lo profesional. Espero que nuestra relación dure muchos años más.

También mencionar a mis compañeros del IMUS durante estos cuatro años de tesis, a todos los que estáis y a todos los que os habéis ido. De no haber sido por vosotros, esto habría sido muy distinto. Han sido muchos los buenos momentos, que espero seguir compartiendo con vosotros durante mucho tiempo. Me gustaría además mencionar a mis compañeros del Doble Grado en Física y Matemáticas, me hicisteis ver que la física y las matemáticas en grupo son más divertidas.

Faltan palabras para poder agradecer a mi nebulosa todo lo que ha hecho por mí. Sólo tú sabes lo feliz que me siento de que nos hayamos cruzado, y de que podamos compartir este momento juntos. Pronto vendrán nuevos momentos que celebrar, no me cabe duda.

Por último, quiero agradecer todo su esfuerzo a mis padres, Pedro y Conchi, y a mi hermano, Víctor. A los primeros, les traje de cabeza unas semanas antes de Selectividad al decirles que quería estudiar el Doble Grado en Física y Matemáticas, y que este no estaba en Málaga. Gracias por vuestro apoyo incondicional a todas mis decisiones. Esta tesis tiene bastante de vosotros.

Muchas gracias.

*Alejandro*

*The true delight is in the finding out  
rather than in the knowing.*

---

**Isaac Asimov**

---

## RESUMEN

---

En las últimas décadas, el progreso tecnológico ha elevado el rol de las simulaciones numéricas a ser una herramienta fundamental en la mayoría de ciencias y aplicaciones tecnológicas. Para ser más exactos, las simulaciones basadas en modelos apoyados en ecuaciones diferenciales adquieren un papel crítico debido a su diversidad de aplicaciones, que cubren desde ingeniería hasta economía, incluyendo áreas tan importantes como la medicina. Estas simulaciones sirven de plataformas virtuales que permiten la realización de experimentos, contribuyendo enormemente a la comprensión de diversas propiedades de los sistemas y sus dinámicas.

Sin embargo, trabajar con los sistemas altamente complejos necesarios para aplicaciones realistas y efectivas supone un reto significante. Tales sistemas normalmente tienen asociadas decenas de miles o millones de grados de libertad, haciendo su resolución numérica prohibitivamente cara si se utilizan métodos convencionales. Esta complejidad requiere recursos computacionales extensos, necesitando horas o incluso días de computación, junto a Computación de Alto Rendimiento o arquitecturas informáticas específicas. Esto representa un gran problema, especialmente cuando se necesita una simulación en tiempo real o interactiva, o cuando se consideran múltiples valores de los parámetros, por ejemplo, en la asistencia en los procesos de toma de decisiones o diseño industrial.

Esta tesis doctoral tiene como propósito abordar el reto de los altos costes computacionales asociados con los modelos realistas, y es en este punto donde el concepto de Modelado de Orden Reducido (ROM, por sus siglas en inglés) entra en juego. La filosofía subyacente en este campo recae en reemplazar el costoso problema original por uno alternativo, computacionalmente eficiente, que mantenga las propiedades cualitativas y cuantitativas esenciales de la solución original. En general, el uso de las técnicas ROM está motivado por la necesidad de mantener un balance entre la precisión y la eficiencia computacional en varias aplicaciones, permitiendo el análisis y la manipulación de sistemas complejos de una forma más práctica y manejable.

Las principales contribuciones de esta tesis doctoral se enmarcan en tres aplicaciones diferentes: sistemas dinámicos lento-rápido, modelos de turbulencia y Ecuaciones en Derivadas Parciales (EDPs) elípticas.

Relativo a la primera, el foco se sitúa en el estudio y simulación de modelos de redes de actividad neuronal que involucran múltiples escalas de tiempo. Estos modelos se tratan por medio de un método inédito basado en la Descomposición Ortogonal Propia, con el objetivo específico de abordar una de las limitaciones que presenta, a saber, la potencial pérdida de estructura del modelo original. En este caso, la pérdida de la separación de escalas temporales.

En lo que respecta a los modelos de turbulencia, el énfasis se encuentra en el modelo de Smagorinsky y el método de Bases Reducidas. Aquí, el objetivo es salvar una limitación del método relativa al reto de obtener un estimador de error *a posteriori* basado en análisis matemático, que depende de la discretización numérica. En particular, desarrollamos un estimador de error basado en la teoría de cascada de energía de Kolmogórov.

Por último, relativo a las EDPs elípticas, el objetivo que nos planteamos se centra en la resolución de EDPs elípticas simétricas y en calcular el mejor subespacio que aproxima su solución. La investigación gira en torno al método de la Descomposición Generalizada Propia (PGD), con el propósito de abordar una limitación asociada al cálculo de los modos PGD óptimos. Específicamente, el propósito es explorar la posibilidad de calcular directamente estos modos en una variedad de Grassmann, utilizando el conocido algoritmo del Gradiente Descendente, adaptado a este marco.

---

## ABSTRACT

---

In the last decades, the technological progress has elevated the role of numerical simulations to be a fundamental tool in the majority of sciences and technical applications. More precisely, simulations based on models that are supported in differential equations acquire a critical role due to its diversity of applications, that cover from engineering to economy, and even areas so important as medicine. These simulations serve as virtual platforms enabling experimental realization, greatly contributing to the comprehension of diverse system properties and dynamics.

Nevertheless, working with highly complex systems necessary for realistic and effective applications poses a significant challenge. Such systems usually have associated ten thousands or millions of degrees of freedom, rendering their numerical solving prohibitively expensive using conventional methods. This complexity demands extensive computational resources, sometimes requiring hours or even days of computation alongside High-Performance Computing or specialized informatics architectures. This fact represents a major problem, especially when a real-time or interactive simulation is needed, or when multiple values of the parameters need consideration, such as assisting decision-making processes or industrial design.

This PhD dissertation aim to address the challenge of the high computational costs associated with those realistic models, and it is at this point where the concept of Reduced Order Modelling (ROM) comes into play. The underlying philosophy in this field lies in replacing the original costly problem with an alternative, computationally efficient one, that retains essential qualitative and quantitative properties of the original solution. Overall, the use of ROM techniques is motivated by the need to strike a balance between accuracy and computational efficiency in various applications, enabling the analysis and manipulation of complex systems in a more practical and manageable manner.

The main contributions of this PhD dissertation are encompassed in three different applications: slow-fast dynamical systems, turbulence models and elliptic Partial Differential Equations (PDEs).

Concerning the first, the focus lies on studying and simulating network models of neuronal activity involving multiple timescales. These models are going to be treated

through a novel method based on the Proper Orthogonal Decomposition method, with a specific goal of addressing one of the limitations that it presents, namely, the potential loss of structure from the original model. In this case, the lost of the separation of the timescales.

Regarding turbulence models, the emphasis is placed on the Smagorinsky model and the Reduced Basis method. Here, the aim is to overcome a limitation of the method concerning the challenge of obtaining an *a posteriori* error estimator using mathematical analysis, which depends on numerical discretization. In particular, we develop an *a posteriori* error estimator based upon the Kolmogórov's energy cascade theory.

Lastly, concerning elliptic PDEs, the goal centers on resolving symmetric elliptic PDEs and computing the best subspace approximating their solutions. The investigation revolves around the Proper Generalized Decomposition (PGD) method, aiming to address a limitation related to the computation of optimal PGD modes. Specifically, the purpose is to explore the feasibility of directly computing these modes on a Grassmann manifold, using the known Gradient Descent algorithm, adapted to this framework.

---

## CONTENTS

---

Introduction	1
1 Basics of Reduced order models	15
1.1 Differential Equations depending on parameters	16
1.1.1 Parametric Ordinary Differential Equations	16
1.1.2 Parametric Partial Differential Equations	18
1.2 Algebraic representation	20
1.2.1 Algebraic representation of systems of parametric ordinary differential equations	21
1.2.2 Algebraic representation of systems of parametric partial differential equations	21
1.3 Reduced Order Modelling	22
1.3.1 Proper Orthogonal Decomposition	23
1.3.2 Greedy Algorithm	25
1.3.3 POD+Greedy algorithm	27
1.4 Treatment of nonlinear terms in reduced order modelling	28
1.4.1 Empirical Interpolation Method	29
2 Mathematical models addressed	31
2.1 Models of neuronal Intracellular Calcium Concentration dynamics	31
2.1.1 Single-cell model of neuronal Intracellular Calcium Concentration dynamics	31
2.1.2 Two coupled-cells model of neuronal Intracellular Calcium Concentration dynamics	35
2.2 Reduced Order Model for the Smagorinsky turbulence model	43
2.2.1 Smagorinsky turbulence model	43
2.2.2 Reduced Order Model for the Smagorinsky turbulence model	46

### I Model order reduction in multiple timescale network models

Introduction	51
3 Structure-preserving methods based on the Proper Orthogonal Decomposition	55
3.1 Proper Orthogonal Block Decomposition	55

3.1.1	Properties of the Proper Orthogonal Block Decomposition	57
3.2	Automatic Proper Orthogonal Block Decomposition	59
4	Application to Multiple Timescales Network Models	63
4.1	Intracellular Calcium Concentration Network model	64
4.1.1	Network model of neuronal Intracellular Calcium Concentration dynamics	64
4.1.2	Preliminary study	69
4.1.3	Numerical Results	73
4.2	Application to bursters models	79
4.2.1	Hindmarsh-Rose Network model	79
4.2.2	Fold/Hopf Bursting Network model	87
4.3	Limitations	92
4.3.1	Ill-located blocks	92
4.3.2	Limitations in the case of nonlinear coupling	94
4.3.3	Close to chaotic behavior	96
	Conclusions and open problems	99

## II Reduced Basis modelling of turbulence with well-developed inertial range

Introduction 103

5	Reduced Basis modelling of turbulent flows based on the statistical turbulence theory	107
5.1	A posteriori error indicator based on the statistical turbulence theory	107
5.1.1	A brief introduction to the Kolmogórov $-5/3$ law	108
5.1.2	Error estimates based on the energy spectrum	109
5.1.3	Kolmogórov's <i>a posteriori</i> error indicator	116
5.2	General algorithm to build the reduced basis spaces	118
5.2.1	Parameter selection	119
6	Application to Smagorinsky turbulence model	121
6.1	Refinement of the characteristics of the method	121
6.1.1	Space-time discretization	121
6.1.2	Data of the problem and initial Condition	122
6.1.3	Selection of the POD+Greedy procedure	123
6.1.4	Treatment of the eddy viscosity term by EIM	125
6.1.5	Time variability in the error indicator	126
6.2	Numerical Results	126

6.2.1	Comparison against actual relative error	128
6.2.2	Comparison against equispaced sampling	133
	Conclusions and open problems	139
<b>III Intrinsic Proper Generalized Decomposition for Parametric Elliptic Problems</b>		
	Introduction	143
7	Computation of intrinsic Proper Generalized Decomposition modes in a matrix framework	145
7.1	Intrinsic Proper Generalized Decomposition	145
7.2	Algebraic representation of the intrinsic Proper Generalized Decomposition	147
7.3	Computation of intrinsic Proper Generalized Decomposition modes via the Gradient Descent algorithm	150
7.3.1	Gradient of a function on a matrix manifold	151
7.3.2	Gradient Descent equation on a matrix manifold	154
7.3.3	Adapted Gradient Descent algorithm to the matrix framework	156
7.4	Deflation Algorithm for the computation of intrinsic Proper Generalized Decomposition modes via the Gradient Descent algorithm	157
7.4.1	One-dimensional intrinsic Proper Generalized Decomposition	158
7.4.2	One-dimensional adapted Gradient Descent algorithm to the matrix framework	159
7.4.3	Deflation algorithm	161
8	Application to symmetric elliptic partial differential equations	165
8.1	Diffusion problem	165
8.2	Elastostatics problem	172
	Conclusions and open problems	177
	Bibliography	179



---

## INTRODUCTION

---

In the last decades, the technological progress has elevated the role of numerical simulations to be a fundamental tool in the majority of sciences and technical applications (Ahmed et al., 2017; Jordan & Mitchell, 2015). More precisely, simulations based on models that are supported in differential equations acquire a critical role due to its diversity of applications, that cover from engineering (Ahmed et al., 2017; Blocken, 2015) to economy (Albu et al., 2020; Bordag, 2015), and even medicine (Fernández-García & Vidal, 2020; Vidal et al., 2012). These simulations provide a virtual platform that allows the realization of experiments, which are of great utility for the understanding of the properties and dynamics of diverse systems.

Nevertheless, in order to achieve realistic and effective applications, we have to work with systems that present a very high complexity. These systems usually have associated ten thousands or millions of degrees of freedom (Kenway & Martins, 2014; Komatitsch et al., 2003), which makes its numerical solving extremely costly by means of traditional methods, needing up to hours or even days of computations, alongside with the use of High-Performance Computing (HPC), or the construction of ad hoc informatic architectures (Kindratenko et al., 2009; Taubenblatt, 2011). This fact represents a major problem, especially when a real-time or interactive simulation is needed or when multiple values of the parameters need to be considered, for instance, to help in the decision-making process (Erdem & Keane, 1996; Taylor et al., 1998) or in the industrial design (Hiermaier, 2007).

In the context of this thesis, we aim to address the challenge of the high computational costs associated with those realistic models, and it is at this point where the concept of Reduced Order Modelling (ROM) comes into play (Bamani & Al-Saddiki, 1991; Pan & Pal, 1995). The underlying philosophy in this field lies in replacing the original costly problem with an alternative problem in which the solution for any parameter configuration can be evaluated in a computationally efficient manner, while preserving the essential qualitative and quantitative properties of the original solution. Alongside the search for computational efficiency and real-time simulations, other primary motivations for using ROM include data compression and storage (Aharon et al., 2006), inverse problems (Egbert & Erofeeva, 2002), visualization and interpretability

(Carrizosa et al., 2017) and control problems (Ito & Ravindran, 1998). Overall, the use of ROM is motivated by the need to strike a balance between accuracy and computational efficiency in various applications, enabling the analysis and manipulation of complex systems in a more practical and manageable manner.

There exists several approaches in order to obtain the ROM. They can be broadly classified into different classes (Lassila et al., 2014; Silva et al., 2007): Proper Orthogonal Decomposition (POD) methods (Kerschen et al., 2005), Reduced Basis (RB) methods (Boyaval et al., 2010; Quarteroni et al., 2016), Balancing methods (Gugercin & Antoulas, 2004), simplified physics (Majdabadi et al., 2015), nonlinear manifold methods (Mojgani & Balajewicz, 2021) and projection based reduction, like Proper Generalized Decomposition (PGD) (Chinesta et al., 2011) and Krylov subspace methods (Bai, 2002). In this thesis, we will focus on the study and applications of three of the former approaches, the POD method, the RB method and the PGD method. For the reduction of the nonlinear terms, we highlight the EIM presented in Barrault et al. (2004), and then extended in Grepl et al. (2007). These problems have been selected within the scope of the Spanish national projects "ROM Optimization for Architecture and Design. ROAD" (RTI2018-093521-B-C31) and "Modelos de orden reducido híbridos aplicados a flujos incompresibles y redes neuronales cerebrales. HYROM" (PID2021-123153OB-C21), and it also aligns with the development of the European project "Accurate Roms for Industrial Applications. ARIA" (H2020-872442).

The POD is a numerical method that enables a reduction in the complexity of computer intensive simulations. Its main use is to decompose a physical field, for instance, velocity or pressure, into a set of functions, the so-called POD modes, modulated by some coefficients. This decomposition is performed depending on the different variables that influence its physical behaviors (Kerschen et al., 2005). It is a generalization of the Principal Component Analysis (Pearson, 1901) in the field of statistics, or the Singular Value Decomposition (SVD) in the field of linear algebra, as it refers to singular values and singular vectors of the physical field. In fact, the optimality of the POD is ensured by the celebrated Eckart and Young theorem (Eckart & Young, 1936). Once the dominant structures are extracted, usually a Galerkin projection of the governing differential equations can be employed to obtain a low-order system (Berkooz et al., 1993). The resulting low-order model is usually addressed as standard POD-Galerkin ROM.

Addressing RB methods, they were first introduced in the late seventies in Almroth et al. (1978) and Nagy (1979) for structural analysis of beam and arches. In the early

eighties, the mathematical basis developed in previous works was extended, which resulted in the presentation of several works on RB methods for structural problems (Almroth et al., 1981; Noor & Peters, 1980). Also in the eighties, the presentation of the first analysis for RB methods was made in Fink and Rheinboldt (1983a, 1983b). In those works, the effectiveness of the RB method is provided by a theoretical error estimation for a general non-linear problem. We will be focusing on the application of RB methods to fluid dynamics problems. The first works in this field were presented in the late eighties in Gunzburger (1989) and Peterson (1989).

However, the RB method made significant progress at the beginning of the new millennium with the development of a general mathematical background analysis. One of the key advancements was the creation of an *a posteriori* error bound estimator and an algorithm for the basis functions, which marked a milestone in the expansion of the RB method. The *a posteriori* error bound estimator is particularly useful in the information selection process for constructing the reduced basis space. This selection is typically performed using a Greedy algorithm. In each iteration of the Greedy algorithm, the snapshot associated with the parameter value that exhibits the highest error between the High Fidelity (HF) and RB solutions is chosen. Determining the actual error precisely can be computationally expensive. Therefore, the weak Greedy algorithm is often employed in RB methods. This algorithm replaces the exact error calculation with an error estimator, which offers a more practicable alternative. The first appearance of the weak Greedy algorithm in the context of RB methods was in Prud'Homme et al. (2002) and in Veroy et al. (2003). Furthermore, these works exploit the parameter dependence of the problem, presenting an offline/online decomposition of it. That is, the problem is divided in two phases, an offline phase, devoted to the computation of the parameter-independent matrices, and an online phase, focused on computing the RB solution depending on the parameter value. Also, in Maday et al. (2002), general *a priori* convergence properties are studied.

The need for an *a posteriori* error estimator in Reduced Basis (RB) methods is evident in their development. Typically, this estimator relies on the dual norm of the residual, providing a means to certify the method, that is, to yield an error bound for the RB approximation. In the context of linear Partial Differential Equations (PDEs), the process of obtaining an *a posteriori* error bound estimator is well-established. It involves applying continuity and coercivity results to the actual error equation (Quarteroni et al., 2016). However, when dealing with non-linear PDEs, the development of an *a posteriori* error bound estimator through mathematical analysis depends strongly

on the selected discretization. For instance, the Brezzi-Rappaz-Raviart (BRR) theory (Brezzi et al., 1980) is employed for approximating regular branches of solutions in non-linear PDEs, enabling effective error estimation in these cases. Nevertheless, like any theoretical framework, it has its limitations and challenges.

For problems with high-dimensional parameter spaces, the (weak) Greedy Algorithm can be coupled with the POD procedure. This coupled approach, commonly referred as POD+Greedy approach, was introduced in Haasdonk and Ohlberger (2008). Usually, in the POD+Greedy procedure, each greedy step is invoking a temporal compression step by performing a POD. In Haasdonk (2013), the existing convergence results for the Greedy algorithm were extended to the POD-Greedy algorithm case.

Unfortunately, the POD methods present a great disadvantage, as it needs knowledge about the original solution before any ROM approximation can be applied. This knowledge has to be large enough to capture the main behaviors of the actual solution over a range of the parameters. This can be extremely expensive in terms of computational time or storage. The PGD methods address this and other issues arising in the POD framework. A PGD method is an iterative numerical technique developed for solving PDEs under specified boundary conditions. Their first appearance was in the mid-2000 in Ammar et al. (2006). This method was designed to alleviate the ‘curse of dimensionality’ when solving problems defined in high-dimensional spaces. The main idea behind this technique is to assume that the solution admits a decomposition into a sum of basis functions which do not need to be orthogonal between them, that is, the solution admits a representation as a sum of rank-1 tensors. Then, the PGD algorithm aims to approximate the solution through successive enrichment by means of a deflation algorithm. In each iteration, a new mode is calculated and incorporated into the approximation. Therefore, the complexity of the reduced problem increases linearly. This is clearly a great improvement and allows us to work efficiently with problems defined in much higher dimensions. Furthermore, thanks to linearity and separability of the representations of the method, only one-dimensional operations are needed. In addition, this approximation is constructed without any *a priori* knowledge of the solution. There are several ways to approximate the reduced solution. A number of different PGD methods are described in Nouy (2010).

In this thesis, we will focus on the PGD method proposed in Azaiez et al. (2018), the so-called intrinsic PGD. This technique is based on the construction of recursive approximations on finite-dimensional optimal subspaces. These approximations are obtained by means of the minimization of the mean parametric error of the residual,

similar to the one introduced in Falcó and Nouy (2011). In the latter work Azaiez et al. (2020), this technique is applied along with a deflation algorithm and a Power Iterate (PI) method to compute the PGD modes for parametric elliptic PDEs both, symmetric and non-symmetric.

We begin this thesis by dedicating a chapter to introducing the fundamental concepts of model order reduction. Therefore, **Chapter 1** serves as a foundation for the subsequent sections of the thesis where we delve deeper into the topic. In this chapter, we start in Section 1.1 from the concept of differential equations depending on one or various parameters. On the one hand, we introduce a general system of parametric ODEs, mainly used in Part I. On the other hand, we present a general parametrized PDE, that will be the main object of study in Parts II and III. Then, we present in Section 1.2 the algebraic equivalent of the previous differential equations, that is, a representation of that problem through operations between matrices and vectors. The interested reader is referred to Quarteroni et al. (2016) for some examples. Section 1.3 is devoted to the argumentation of the need of the development of ROM techniques. Additionally, it provides an overview of the most commonly employed techniques, which will be detailed in the subsequent chapters, namely, the POD method and the Greedy Algorithm, along with a coupled strategy that combines the POD and Greedy procedures. In order to treat possible non-linearities that could be present in the original models, we introduce in Section 1.4 a technique that helps us to approximate any non-linear term, the Empirical Interpolation Method (EIM).

Furthermore, we dedicate a chapter to the presentation of two models of interest for the rest of the thesis. Hence, **Chapter 2** acts like a presentation for these two models of interest. On the one hand, we present in Section 2.1 a model of neuronal Intracellular Calcium Concentration, presented in Krupa et al. (2013) and studied in Fernández-García and Vidal (2020), among others. This model is used in Part I. On the other hand, in Section 2.2, we present the basic Large Eddy Simulation (LES) turbulence model, the celebrated Smagorinsky turbulence model introduced in Smagorinsky (1963), along with its associated reduced basis problem. This model is used in Part II.

The rest of the thesis is then divided into three main parts, each focusing in one particular kind of problem modelled by a system of differential equations: slow-fast dynamical problems, that will be tackled by a Proper Orthogonal Decomposition (POD) approach, turbulence problems, that will be addressed by a Reduced Basis (RB) approach, and symmetric elliptic PDEs, that will be studied under a Proper Generalized Decomposition (PGD) approach. Each part of the thesis explores and

analyzes the respective ROM method in detail. It outlines a primary difficulty that we aim to address and introduces the proposed enhancement. These enhancements are designed in a manner, so they serve to improve the study of their corresponding problem. The content of each part is outlined and explained in the following to ensure clarity and coherence throughout the thesis. Every part has an introduction, where we present the context and the main problem.

## PART I

In this part, our emphasis is on the study and simulation of network models of neuronal activity that involve different timescales, modelled by a system of ODEs. In particular, we focus on the POD method, with a specific goal of addressing one of the limitations that it presents, namely, the potential loss of structure from the original model. This part aligns with the Spanish national project HYROM.

In the last thirty years, extensive research has been groundbreaking in modeling neuronal activity networks. These models use dynamical systems coupled as networks to represent interactions among neuron assemblies, employing various paradigms and scales such as neural mass (Coombes, 2023), neural fields (Faugeras et al., 2008), mean-fields (Byrne et al., 2022), dendritic trees (Traub et al., 2001), and others. However, persistent challenges include difficulties in classifying synchronization features between network cells and the computational cost of reliable simulations for realistic-sized networks.

The challenge arises from varying timescales in neuronal activity, reflected in models using multiple timescales systems as cell dynamics within a network. Synchronization mechanisms in these systems differ from harmonic oscillators, particularly in terms of the impact of coupling types and strengths. Despite numerous studies exploring synchronization in the presence of complex phenomena characteristic of multiple timescale dynamics, such as, canard phenomena and bursting oscillations (Coombes & Wedgwood, 2023; Ermentrout & Wechselberger, 2009; Ersöz et al., 2017; Izhikevich, 2001; Semenov & Fradkov, 2021), a comprehensive analysis of synchronization types and associated conditions is needed, especially for large networks, along with their physio-pathological implications.

However, simulating large-scale network models can be computationally expensive, and while intrusive ROM approximate accurately the solution of the original system (Bandera Moreno et al., 2022), they may lead to the loss of the problem's original

structure, such as the slow-fast separation of variables common in neuroscience problems. The original problem involves quasi-sparse matrices and vectors, but reductions result in full matrices and vectors. To retain the problem's structure, various strategies have been proposed, such as a Krylov subspace-based technique for electronic circuits (Freund, 2011), and, more recently, a domain decomposition approach for fluid simulations (Anderson et al., 2022). In this thesis, we present an alternative approach using the POD method and a data-driven procedure based on collinearity analysis. This approach automatically determines the best structure for the reduced model while significantly reducing computational time.

Part I is outlined as follows:

- In **Chapter 3**, we present the theoretical side of this part. Our emphasis is on a system of parametric ODEs, yet it's worth noting that this technique is adaptable to the realm of PDEs characterized by any qualitative separation in variables.

In Section 3.1, we introduce an enhanced version of the original POD technique, designed to maintain the structural integrity of the initial system, that we name the *Proper Orthogonal Block Decomposition* (POBD). We also present some mathematical results concerning this novel technique in Section 3.1.1. More precisely, we adapt Eckart and Young Theorem to our case, obtaining an error bound for our method, that we present in Proposition 3.1.

Furthermore, in Section 3.2, we introduce a preliminary data-driven phase aimed at determining the optimal structure for the reduced model by means of a correlation analysis between variables and group of variables. This step is conducted to retain the maximum possible structural fidelity of the original model while simultaneously minimizing the number of reduced variables. The whole process of determining the structure for the reduced model and performing the POBD is named as *Automatic Proper Orthogonal Block Decomposition* (APOBD).

- **Chapter 4** is devoted to the validation and numerical study of the methods developed in Chapter 3, namely the POBD and APOBD methods. More precisely, we will apply these reduction techniques to problems arising from the mathematical modelling of biological phenomena exhibited in different cells, like neurons.

Section 4.1 will be devoted to the study of a model of Intracellular Calcium Concentration, presented in Krupa et al. (2013), studied in Fernández-García and Vidal (2020) and in Bandera Moreno et al. (2022), among others. In Section 4.2.1, we will study the celebrated Hindmarsh-Rose model, introduced in Hindmarsh

and Rose (1984) and widely studied (Barrio & Shilnikov, 2011; Bonaventura et al., 2022; Etémé et al., 2017), among others. Finally, Section 4.2.2 will be focused on the study of pancreatic  $\beta$ -cells models described by a Fold/Hopf Bursting model (Izhikevich, 2000; Pernarowski et al., 1992; Smolen et al., 1993).

Our focus involves network models characterized by a substantial number of interconnected oscillators. To evaluate the efficacy of reduction techniques, we will employ conventional metrics like speed-up, that is, the ratio between the solving computational times of the original and reduced model, and the number of equations in reduced models, in comparison to the original model. Additionally, we will use the sparsity index, an indicator reflecting the reproduction of the original problem's structure, gauged through the sparsity of the matrices associated with each reduced model.

## PART II

In this part, our emphasis is the study and simulation of turbulent flows, modelled by PDEs. Our focus would be on the RB method, with a specific goal of addressing one of the limitations that it presents, namely, the difficulty of obtaining mathematics-based *a posteriori* error estimator, and its dependence on the numerical discretization. In particular, we study the development of an *a posteriori* error estimator based upon the Kolmogórov's energy cascade theory (Kolmogórov, 1941). This part aligns with the Spanish national projects ROAD and HYROM, and with the European project ARIA.

Kolmogorov's energy cascade theory, also known as K41 theory, is a foundational concept in the modelling of turbulence (Chacón Rebollo & Lewandowski, 2014; Kolmogórov, 1941; Richardson, 1922). It elucidates the transfer of kinetic energy from large-scale eddies to progressively smaller scales in turbulent flows. This energy cascade, marked by self-similar scaling behavior, culminates in the dissipation of energy into heat at the smallest scales. The theory provides a statistical description of turbulence, serving as a guiding framework for developing turbulence models and conducting numerical simulations of turbulent flows. The energy cascade concept has led to using high-order modes of the POD expansion for modeling small scales in the inertial range. However, achieving precision with this method requires a substantial number of modes, resulting in ROM computation times comparable to a direct Full Order Model (FOM) solver (Hijazi et al., 2020). An alternative approach involves constructing ROMs for established turbulence models, particularly addressing Large Eddy

Simulation (LES) models (Germano et al., 1991; Smagorinsky, 1963). In this strategy, the focus is not on modeling turbulence through ROM, but on creating surrogate LES models for significantly faster computations. From the family of LES models, we highlight the Smagorinsky model (Smagorinsky, 1963), in which the eddy viscosity is modeled by a non-linear term, that depends on the mesh size and the modulus of the velocity gradient.

Certified RB LES Smagorinsky models for steady flows have demonstrated substantial computing speed-ups for both single and multi-parametric configurations (Ballarin et al., 2020; Caravaca García, 2022; Chacón Rebollo et al., 2023; Chacón Rebollo et al., 2017). Error certification relies on *a posteriori* error estimation procedures, formulated using the BRR theory for approximating regular branches of non-linear PDEs (Brezzi et al., 1980), leveraging the enhanced regularity of Smagorinsky model solutions. However, this approach has significant limitations. Constructing *a posteriori* error indicators is closely tied to the specific turbulence model and FOM in use, as well as to the nature of the parameters, whether physical or geometrical, considered in the ROM. For the Smagorinsky turbulence model, there exists a well-established theory (potentially extendable to more general LES models) based on the enhanced regularity compared to the Navier-Stokes equations, facilitating the development of *a posteriori* error estimation (Caravaca García, 2022). Nevertheless, these estimations are computationally intensive, involving the calculation of stability factors in space-time, that additionally depend on discretization parameters.

This part of the thesis focuses on establishing a RB method applicable to a wide range of transient LES parametric turbulence models, irrespective of the FOM employed. The key approach involves an error indicator rooted in the physics of turbulent flows rather than the mathematical analysis of models. This indicator quantifies the deviation between the energy spectrum of a trial solution and the theoretically predicted spectrum by Kolmogórov's energy cascade theory of turbulence in statistical equilibrium.

Part II is outlined as follows:

- In **Chapter 5**, we present the theoretical side of this part. The main purpose behind this chapter is the introduction of an *a posteriori* error indicator for turbulent flows based only on physical properties of turbulent flows, instead of on a mathematical analysis of the selected model and discretization. More precisely, for the development of this indicator, we have used general results from the statistical turbulence theory, attributed to Kolmogórov (Kolmogórov, 1941).

In Section 5.1, we introduce the error indicator based upon Kolmogórov's energy cascade theory, along with some basics results from this theory. Furthermore, in Section 5.1.2, we develop some error estimates for Kolmogórov's indicator.

Theorem 5.2 states that the error between two velocity fields of locally isotropic flows, that is, the statistical properties of the mean flow are invariant under rotations, solutions of the Smagorinsky turbulence model, is fully driven by the indicator. Also, in Theorem 5.3, we state that the error between the FOM and ROM solutions is driven by the estimator plus the projection error on the reduced space.

To end this chapter, we describe in Section 5.2 a general algorithm based on the *a posteriori* error indicator for the construction of the reduced basis model of LES turbulent model. Furthermore, as the indicator is not expected to vanish as the reduced dimension grows, but to converge to the indicator for the solution of the FOM, then, the usual criterion for the selection of the best parameter in the Greedy algorithm is no longer useful. Instead, we propose in Algorithm 1 new criteria for the selection of the next parameter at each step. Algorithm 1 first aims to add to the basis the information of the solutions whose energy spectrum is farther away from the theoretical one. Additionally, if the selected parameter has already been selected in a previous iteration, we are assuming that the reduced error indicator is getting close to the indicator of the FOM solution, and the previous solution plays the role of the full-order one to compare with it.

- Chapter 6 is devoted to the validation and numerical study of the methods developed in Chapter 5. In particular, we address some academic tests for the Smagorinsky turbulence model. We consider 2D periodic flows, yet meaningful, as they are designed to present the inertial spectrum predicted by Kolmogórov's theory. We assume that the Smagorinsky flow already has a part of its spectrum within the inertial range.

First, in Section 6.1, we conduct a preliminary study. Our attention in this section is specifically directed towards the handling of the eddy viscosity term, the selection of the POD+Greedy procedure and the study of the time variability of the error indicator.

Then, in Section 6.2, we present the implemented test in order to validate the performance of the use of the error indicator. We compare the errors obtained with this strategy versus those obtained with the best error indicator, that is,

the exact error. As a result, we obtain errors between the ROM and the FOM solutions quite close to the optimal ones. Furthermore, we obtain a spectral decay of the error as the dimension of the reduced basis space increases, quite close to the one obtained if the exact error is used as indicator. We obtain speed-up rates of computing times around 23, which is quite satisfying for an evolution turbulence model.

### PART III

In this part, our emphasis is on the resolution of symmetric elliptic PDEs, and the computation of the best subspace that approximates their solution. To that end, we focus on the PGD method, with the specific objective of addressing one of the limitations that it presents, namely, the computation of the optimal PGD modes. In particular, we aim to study the possible calculation of these modes directly in a Grassmann manifold by means of the Gradient Descent (GD) algorithm. This part aligns with the Spanish national projects ROAD and HYROM.

Among all the PGD methods, we wish to bring attention to the formulation of the PGD introduced in Azaïez et al. (2018), termed the intrinsic PGD. This approach is founded on constructing recursive approximations within finite-dimensional optimal subspaces. These approximations are derived through minimizing the mean parametric error of the residual, akin to the approach introduced in Falcó and Nouy (2011). In that paper, the authors establish the existence of an optimal subspace, for any given dimension, that achieves the most accurate approximation in terms of the error between the precise solution and the Galerkin solution computed within the subspace, measured by the mean parametric norm associated with the elliptic operator. This mirrors the best approximation characteristic of POD subspaces, with the distinction that in their scenario the norm is parameter-dependent. Then, they employ a deflation technique to construct a sequence of approximate solutions within finite-dimensional optimal subspaces, directly during the online step. Furthermore, they demonstrate that the partial sums of these approximations converge to the continuous solution in the mean parametric elliptic norm. Additionally, they establish a strong connection between the conventional PGD for the specified parametric problem and the deflation algorithm presented in this paper. In fact, both methods coincide if the minimum of the functional is computed exactly. This linkage suggests the feasibility of computing

the PGD expansion by directly addressing the optimization problems that lead to the determination of optimal subspaces.

In this part of the thesis, we tackle this latter problem. We propose the resolution of the intrinsic PGD for parametric symmetric elliptic PDEs directly by means of the GD algorithm in a matrix framework in order to obtain the optimal subspaces. This matrix framework can be identified with the Grassmann manifold (Bendokat et al., 2020).

Part III outlines as follows:

- In **Chapter 7**, we present the theoretical side of this part. The main purpose behind this chapter is the presentation of a novel procedure to obtain the intrinsic PGD modes by means of the GD algorithm in a Grassmann framework.

We begin this chapter with an exposition of the intrinsic PGD in its continuous version, elaborated in Section 7.1. Within this section, we also incorporate an examination of various mathematical results and properties related to the intrinsic PGD framework. Then, in Section 7.2, we display the algebraic form of the intrinsic PGD, providing a concise yet comprehensive exploration of its discrete version. We also state the connection between this discrete version and the Grassmann manifold.

Furthermore, Section 7.3 is focused on introducing the modified version of the GD algorithm for obtaining intrinsic PGD modes, as outlined in Algorithm 3. Concluding this section, we also present the deflation algorithm used in Azaïez et al. (2020). This deflation algorithm proves instrumental in deriving higher-dimensional optimal subspaces through a recursive process of obtaining one-dimensional optimal subspaces.

- **Chapter 8** is devoted to the validation and numerical study of the procedure developed in Chapter 7. In particular, we will apply the procedure to two different parametric symmetric problems.

On the one hand, in Section 8.1, we apply the procedure to a problem of parametric diffusion, where two different materials coexist in a domain. Here, the parameter will be the relationship between the diffusion of both materials. This was also the test performed in Azaïez et al. (2020), where the deflation algorithm was employed joint to a Power Iterate (PI) method for computing PGD modes in parametric elliptic PDEs. The results are very satisfactory, as we obtain very low errors with just a few modes, reaching up to 3 more orders of error in comparison with the method proposed in Azaïez et al. (2020).

On the other hand, in Section 8.2, we apply the procedure to a problem of classic elastostatics, where there is a block of material fixed on one side, which is also under the action of gravity. Here, the parameter will be the relationship between the Lamé parameters of the material. In this case, we also obtain very satisfactory results, achieving errors of the order of  $10^{-8}$ .

Finally, each part ends with a conclusion, where we highlight the main advances obtained and the remaining open problems. Moreover, we also spotlight the contributions derived from each part.



# 1

---

## BASICS OF REDUCED ORDER MODELS

---

The resolution of real life problems modelled either by systems of parametric Ordinary Differential Equations (ODEs) or by parametric Partial Differential Equations (PDEs) can be very expensive in terms of computational time, or even in terms of storage. Even more, when several resolutions of these problems for various values of the parameters are needed, or when they should be done on-the-fly to help in the decision-making, it can be unaffordable. Reduced Order Modelling (ROM) techniques aim to develop a problem of lower numerical complexity when compared to the expensive original problem, whose solution provides a good approximation of the original one in a much lesser computational time.

In this chapter, we present a short review of basic results that we will be using along this dissertation. We will cover from the concept of differential equations depending on one or various parameters to some basic methods for model order reduction, going through some techniques to tackle the nonlinear terms arising in the differential equations.

The chapter outline is the following. In Section 1.1, we present the concept of a parametric differential equation, for both, ODEs in Section 1.1.1 and PDEs in Section 1.1.2, along with some academic examples. Furthermore, we present in Section 1.2 the algebraic representation for differential equations. Section 1.3 will be devoted to the motivation of ROM and the presentation of two of the most commons techniques in model order reduction, the Proper Orthogonal Decomposition (POD) in Section 1.3.1 and the Greedy Algorithm in Section 1.3.2, and we also present a coupled POD+Greedy technique in Section 1.3.3. In Section 1.4, we present a technique to treat the nonlinear terms that could appear in the original problem, the Empirical Interpolation Method (EIM).

### 1.1 DIFFERENTIAL EQUATIONS DEPENDING ON PARAMETERS

In this section, we present the two main objects that will be the primary focus throughout this dissertation. On the one hand, we introduce a general system of parametric ODEs, mainly used in Part I. On the other hand, we present a general parametrized PDE, that will be the main object of study in parts II and III.

Depending on the nature of the parameters, these can be separated into two groups, physical and geometrical parameters. We consider as physical parameters those that arise from the physical or biological properties of the problem, like the Reynolds number in incompressible fluid flows problems or the coupling parameters in problems that model cell interactions inside a network. On another note, geometrical parameters are those involved in the shape of the problem domain. In the following, we will only consider physical parameters and let us denote by  $\gamma = (\gamma_1, \dots, \gamma_P) \in \mathcal{D}$  the parameter vector, where  $P$  is the number of total parameters present in the problem, and  $\mathcal{D}$  is the parametric set that contains all the possible parameter values.

#### 1.1.1 Parametric Ordinary Differential Equations

In this section, we will denote by  $x = (x_1, \dots, x_N) \in \mathbb{R}^N$ , the vector of unknowns, with  $N$  the total number of unknowns, the overdot notation will stand for the derivative with respect to the time variable  $t$ , and  $I_f = [0, T_f]$ , with  $T_f > 0$ , will be the time domain. Then, for every parameter vector  $\gamma \in \mathcal{D}$ , we can consider the following system of autonomous ODEs

$$\dot{x} = f(x; \gamma), \quad \text{in } I_f, \tag{1}$$

where  $f(x; \gamma)$  is a given function non-depending on time, and the overdot denotes the derivative in time.

Furthermore, we can consider an initial condition

$$x(0; \gamma) = x_0(\gamma), \quad t \in I_f, \quad \gamma \in \mathcal{D},$$

with  $x_0(\gamma)$  being a given vector depending on the parameter.

#### *Systems of ODEs with multiple timescales*

More precisely, we will focus on a specific class of parametric system of ODEs, those that exhibit multiple timescales. The separation of the timescales is fitted by a pa-

parameter, usually denoted by  $\varepsilon$ , fulfilling  $0 < \varepsilon \ll 1$ . In this case, expression (1) reads

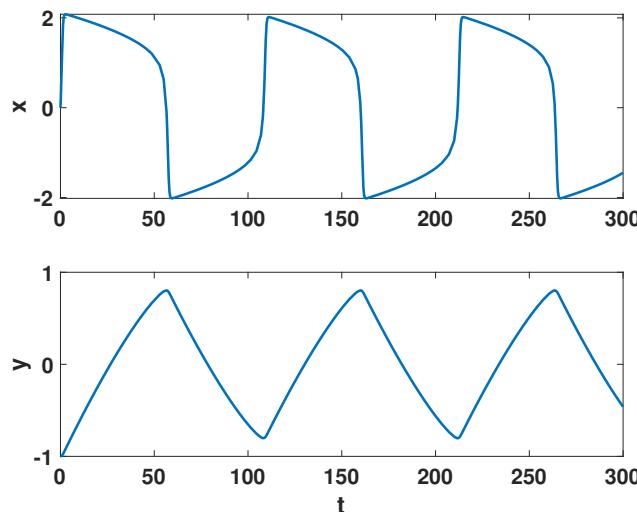
$$\begin{cases} \dot{x} = f_1(x, y; \gamma), \\ \dot{y} = \varepsilon f_2(x, y; \gamma), \end{cases} \text{ in } I_f. \quad (2)$$

The systems that exhibit two or more different timescales are the so-called *slow-fast* systems. With the timescale separation shown in system (2),  $x$  variables are called the fast variables, and  $y$  the slow ones.

For illustration purposes, we present one of the simplest systems with multiple timescales, the Van der Pol oscillator (Van der Pol, 1920, 1926; Van der Pol & Van Der Mark, 1927). This system has only two variables, one fast  $x$ , and one slow  $y$ . The governing equations of the Van der Pol oscillator can be written as

$$\begin{cases} \dot{x} = x - x^3/3 - y, \\ \dot{y} = \varepsilon x, \end{cases} \text{ in } I_f.$$

We represent in Figure 1.1, the solution over the time domain  $I_f = [0, 300]$  of the Van der Pol oscillator, for  $\varepsilon = 0.02$ . We can easily see that the fast variable  $x$  exhibits quick transitions with a strong gradient in time, while the behavior of the slow variable  $y$  does not.



**Figure 1.1:** Evolution over time of the fast variable (upper) and the slow variable (lower) of the Van der Pol oscillator for  $\varepsilon = 0.02$ .

The model order reduction for this kind of systems will be studied more in detail in Part I, dedicated to the model order reduction in multiple timescale network models.

### 1.1.2 Parametric Partial Differential Equations

In this section, let us denote by  $\Omega \subset \mathbb{R}^d$ , with  $d = 2, 3$ , a spatial domain,  $\Gamma$  will denote the boundary of  $\Omega$ , and  $Q_T = I_f \times \Omega$  stands for the spatio-temporal domain.

Then, for every parameter vector  $\gamma \in \mathcal{D}$ , we can consider the following PDE

$$\partial_t u(\gamma) + L(\gamma)u(\gamma) = f(\gamma), \quad \text{in } Q_T, \quad (3)$$

where  $f(\gamma) = f(t, x; \gamma)$  is a given function, and  $L(\gamma) = L(x; \gamma)$  is a generic elliptic operator acting on the unknown  $u(\gamma) = u(t, x; \gamma)$ .

We could also consider a PDE with no time dependence, the so-called elliptic PDEs. In that case, for every parameter vector  $\gamma \in \mathcal{D}$ , we would have the following expression

$$L(\gamma)u(\gamma) = f(\gamma), \quad \text{in } Q_T, \quad (4)$$

where  $f(\gamma) = f(x; \gamma)$  is a given function, and  $L(\gamma) = L(x; \gamma)$  is a generic elliptic operator acting on the unknown  $u(\gamma) = u(x; \gamma)$ . In the following, we consider that we are working with non-elliptic PDEs, as they are more general. Nevertheless, the similar results can be obtained for the elliptic case.

Furthermore, we can consider an initial condition

$$u(0, x; \gamma) = u_0(x, \gamma), \quad x \in \Omega, \quad \gamma \in \mathcal{D},$$

where  $u_0$  is a given function, and we also consider boundary conditions

$$\begin{aligned} u(t, x, \gamma) &= u_D(x, \gamma), & t \in I_f, \quad x \in \Gamma_D, \quad \gamma \in \mathcal{D}, \\ \frac{\partial u}{\partial n}(t, x, \gamma) &= u_N(x, \gamma), & t \in I_f, \quad x \in \Gamma_N, \quad \gamma \in \mathcal{D}, \end{aligned}$$

with  $u_D$  and  $u_N$  given functions, and  $\Gamma_D \cup \Gamma_N = \Gamma$ , with no-intersecting interiors.

Our main objective is to solve the previous PDE problems (4) and (3) numerically, so we introduce its weak formulation. Let us consider  $X = X(\Omega)$  a suitable Hilbert space endowed with the  $X$ -norm, and homogeneous Dirichlet boundary conditions. Then, for every  $t \in I_f$  and  $\gamma \in \mathcal{D}$ , we consider the problem

$$\left\{ \begin{array}{l} \text{Find } u(t; \gamma) \in X \text{ such that} \\ (\partial_t u(t; \gamma), v)_\Omega + A(u(t; \gamma), v; \gamma) = F(t, v; \gamma), \quad \forall v \in X, \end{array} \right. \quad (5)$$

where  $A(\cdot, \cdot; \gamma) : X \times X \rightarrow \mathbb{R}$ , is a bilinear operator associated to the elliptic operator  $L(\gamma)$  and the Neumann boundary condition  $u_N$ , and  $F(t, \cdot; \gamma)$  is a linear operator associated to function  $f(\gamma)$  for every  $\gamma \in \mathcal{D}$ .

The existence and uniqueness of weak solutions of the problems studied in parts II and III, that are like problem (5) or like the variational formulation of problem (4), is given by the celebrated Lax-Milgram theorem or one of its generalizations.

**Theorem 1.1 (Lax-Milgram)**

Let  $X$  be a Hilbert space and  $A(\cdot, \cdot; \gamma)$  a bilinear form in  $X$ . We assume that, for every  $\gamma \in \mathcal{D}$ , the bilinear form is

- bounded, that is,  $|A(\cdot, \cdot; \gamma)| \leq C\|u\|_X\|v\|_X$ , with  $C > 0$ ; and
- coercive, that is,  $A(\cdot, \cdot; \gamma) \geq c\|u\|_X^2$  with  $c > 0$ .

Then, for any  $f \in X'$ , with  $X'$  denoting the dual space of  $X$ , and for every  $\gamma \in \mathcal{D}$ , there exists a unique solution  $u \in X$  to the equation

$$A(u, v; \gamma) = f(v; \gamma), \forall v \in X,$$

and the following bound holds

$$\|u\|_X \leq \frac{1}{c}\|f\|_{X'}.$$

### Space-time discretization

As our aim is to solve problem (5) numerically, a discretization is needed. Through this dissertation, we will work with the finite element approach, and we will consider a Galerkin approximation of problem (5).

Let us define  $\{\mathcal{T}_h\}_{h>0}$  a uniformly regular family of triangulations of  $\Omega$ , where the lower index  $h > 0$  is related to the mesh size. Then, we consider a discrete subspace  $X_h = X_h(\Omega) \subset X$ , of finite dimension  $N_h$ , as an inner approximation of a given Hilbert space  $X$ .

Then, if we set  $u_h(0; \gamma)$  as an approximation of  $u_0(\gamma)$  in the space  $X_h$ , for a given  $t \in I_f$ ,  $\gamma \in \mathcal{D}$ , the semi-discretized equation in space is given as

$$\left\{ \begin{array}{l} \text{Find } u_h(t; \gamma) \in X_h \text{ such that} \\ (\partial_t u_h(t; \gamma), v_h)_\Omega + A(u_h(t; \gamma), v_h; \gamma) = F(t, v_h; \gamma), \quad \forall v_h \in X_h. \end{array} \right.$$

For the sake of clarity, we present now an explicit Euler scheme for time discretization. Extension to more general numerical schemes is possible, but the following analysis would be more tedious. Let  $N_t$  be a positive integer that denotes the number of time steps that will be considered,  $\Delta t = T_f / N_t$  the time step and  $t_k = k\Delta t$ , for  $k = 1, \dots, N_t$ . We also denote by  $u_h^k(\gamma)$  the approximation of  $u_h(t_k; \gamma)$ .

Then, the complete space-time discretization of problem (5) can be expressed as

$$\left\{ \begin{array}{l} \text{For any } k = 1, \dots, N_t, \text{ and } \gamma \in \mathcal{D}, \text{ assuming known } u_h^{k-1}(\gamma) \in X_h, \\ \text{find } u_h^k(\gamma) \in X_h \text{ such that} \\ M(u_h^k(\gamma), v_h; \gamma) + A(u_h^k(\gamma), v_h; \gamma) = F^k(v_h; \gamma) + M(u_h^{k-1}(\gamma), v_h; \gamma), \quad \forall v_h \in X_h, \end{array} \right. \quad (6)$$

where  $M(\cdot, \cdot; \gamma) : X \times X \rightarrow \mathbb{R}$  is defined by  $M(u, v; \gamma) = (u, v)_\Omega / \Delta t$ , and  $F^k(v_h; \gamma)$  is a suitable approximation of the second member for any  $\gamma \in \mathcal{D}$ .

Time-dependent parametric PDEs will be the main object of study in Part II, devoted to the development of reduced order models of incompressible turbulent flows. While, elliptic parametric PDEs will be the main object of study in Part III, focused on the study of the intrinsic Proper Generalized Decomposition expansion in a Grassmann manifold.

## 1.2 ALGEBRAIC REPRESENTATION

In order to treat numerically both, ODEs and PDEs, we will be working with the algebraic equivalent of models (1) and (6), respectively. The algebraic equivalent of a discrete problem, in general terms, is an algebraic equation, that is, a representation of the problem by means of operations between matrices and vectors, see Quarteroni et al. (2016) for some examples. It will also be useful when constructing the reduced models thanks to the techniques that we will present in Section 1.3.

Before we begin to develop the algebraic equivalents of models (1) and (6), we need to introduce some notation that will be useful in the latter.

### Notation 1.1

*With respect to vectors notation:*

- $\mathbf{0}_N$ , the null column vector with  $N$  components.
- $\mathbf{1}_N$ , the column vector with  $N$  components, and all its components equal to 1.

## Notation 1.2

On the other hand, some useful matrix notation will be:

- $\mathbb{I}_N$  is the identity matrix of order  $N$ .
- $\mathbb{O}_N$  is the null matrix of order  $N$ . If it has two subindexes, they will denote the number of rows and the number of columns, respectively.
- If  $\mathbf{d}$  is a column vector,  $\mathbb{D} = \text{diag}(\mathbf{d})$  is a diagonal matrix whose entries are the components of vector  $\mathbf{d}$ .

Bearing these notations in mind, we can focus now on the presentation of the algebraic versions for both, models (1) and (6).

### 1.2.1 Algebraic representation of systems of parametric ordinary differential equations

For the construction of the algebraic representation of model (1), we will consider, like in model (6), an explicit Euler scheme for time discretization. Again, extensions to more general numeric schemes is possible, but we do not cover it here for the sake of brevity.

Let  $N_t$  be a positive integer that denotes the number of time steps that will be considered,  $\Delta t = T_f / N_t$  the time step and  $t_k = k\Delta t$ , for  $k = 1, \dots, N_t$ . We also denote by  $x_k(\gamma)$  the approximation of  $x(t_k; \gamma)$ .

The algebraic representation of model (1), for every  $k = 1, \dots, N_t$ , will be the following:

$$\Delta \mathbf{x}_k(\gamma) = f(\mathbf{x}_{k-1}, \gamma) = \mathbb{A}(\gamma) \mathbf{x}_{k-1}(\gamma) + \mathbf{b}(\mathbf{x}_{k-1}(\gamma); \gamma), \quad (7)$$

where  $\Delta \mathbf{x}_k(\gamma)$  denotes the variation of the solution at time step  $k$ ,  $\mathbb{A}(\gamma)$  is the algebraic representative of the linear part of model (1), and  $\mathbf{b}(\cdot; \gamma)$  summarizes the affine part and the possible nonlinearities involved in model (1). Then, in order to obtain the next value  $\mathbf{x}_k(\gamma)$ , we just need to sum  $\mathbf{x}(\gamma)_k = \mathbf{x}_{k-1}(\gamma) + \Delta t \Delta \mathbf{x}_k(\gamma)$ .

### 1.2.2 Algebraic representation of systems of parametric partial differential equations

For the construction of the algebraic representation of model (6), we consider that we are working in a finite element framework. Then, considering the basis functions of the inner approximation  $X_h$  of  $X$  as the test functions in problem (6), allows us to

identify the latter as a system of equations. If we set  $\mathbf{u}_h^k(\gamma)$  the algebraic representative of  $u_h^k(\gamma)$ , and we obtain for  $k = 1, \dots, N_t$ , a linear system that we have to solve:

$$(\mathbb{M}(\gamma) + \mathbb{A}(\gamma))\mathbf{u}_h^k(\gamma) = \mathbb{M}(\gamma)\mathbf{u}_h^{k-1}(\gamma) + \mathbf{b}^k(\gamma), \quad (8)$$

where  $\mathbb{M}(\gamma)$  is the algebraic representative of the time derivative  $M$ ,  $\mathbb{A}(\gamma)$  is the algebraic representative of the elliptic operator and Neumann boundary condition, and  $\mathbf{b}^k(\gamma)$  summarizes the effect of function  $F^k(\cdot; \gamma)$ .

Here, we have considered, like in model (1) an explicit Euler scheme for time discretization, but again, extensions to more general numeric schemes is possible.

If we were to consider a PDE that does not depend on time (4), we just need to get rid of the algebraic representatives of the time derivative in (8). In that case, we would obtain the following linear system

$$\mathbb{A}(\gamma)\mathbf{u}_h(\gamma) = \mathbf{b}(\gamma), \quad (9)$$

where  $\mathbb{A}(\gamma)$  is the algebraic representative of the elliptic operator and Neumann boundary condition, and  $\mathbf{b}$  summarizes the effect of function  $f(\gamma)$ .

Every algebraic representation (7), (8) and (9), even in the case when there are no nonlinearities involved, could involve the resolution of matrix-vector multiplications or linear systems of very high dimension. That can be very demanding in terms of computational time and storage. In order to solve this issue, some techniques have been developed. We explain them more in detail in Section 1.3.

Algebraic representatives of type (7) will be present in the first part of this dissertation, Part I, when modelling the neural activity by means of networks of coupled oscillators. While a representative of type (8) will be present in Part II, when solving an unsteady incompressible flow. And a representative of type (9) will be present in Part III, when computing the intrinsic Proper Generalized Decomposition expansion in a Grassmann manifold.

### 1.3 REDUCED ORDER MODELLING

In this section, we focus on the argumentation of the need of the development of ROM techniques, the main theme of this dissertation, as well as on the presentation of the techniques that we will be using in the following chapters.

Although the use of ROM techniques is suitable for both, systems of parametric differential equations and parametric PDEs, the philosophy behind their use is quite different:

- In the case of a system of parametric differential equations, like the one presented in (2), our objective will be to reduce the number of equations considered in the original problem, expressing the solution of the rest as a linear combination of the solved ones. These techniques are some of the so-called hyper-reduction techniques (Farhat et al., 2020; Ryckelynck, 2005).
- In the case of parametric PDEs, like the one presented in (3), the objective behind ROM techniques will be the representation of the solution of the original problem as a linear combination of some basis functions that keep the most of the information of the solution. These techniques include the so-called Reduced Basis Method (Ali et al., 2020; Porsching & Lee, 1987; Quarteroni et al., 2016).

In the following, we present some of the most common methods for the construction of the reduced problem, that we will be using along this dissertation. These methods provide, both, the basis and dimension of the reduced problem.

### 1.3.1 Proper Orthogonal Decomposition

The POD method is based on the Singular Value Decomposition (SVD) technique, and it is one of the most popular approaches for construction of the reduced model (Kunisch & Volkwein, 2002; Pinna, 2008; Volkwein, 2013).

In the following, we will consider that we have a set of solutions of the original problem, either (1) or (6),  $\$ = (s_1, \dots, s_{N_s}) \in \mathbb{R}^{N \times N_s}$ , where  $N$  denotes the number of unknowns of problem (1) and the dimension of the discrete space in problem (6). Let us denote by snapshot a solution of the problem for a given parameter setting,  $s_i$  as the  $i$ -th snapshot of the problem, and  $\$$  as the snapshot matrix.

It is known that the following property holds for the SVD of the snapshot matrix  $\$$  (Eckart & Young, 1936; Quarteroni et al., 2016).

### Proposition 1.2

Let  $\mathcal{V}_{N_r} = \{\mathbf{W} \in \mathbb{R}^{N \times N_r} : \mathbf{W}^T \mathbf{W} = \mathbb{I}_{N_r}\}$  be the set of all orthonormal bases of  $N_r$ -dimensional subspaces of  $\mathbb{R}^N$ . Then,

$$\sum_{i=1}^{N_s} \|\mathbf{s}_i - \mathbb{U} \mathbb{U}^T \mathbf{s}_i\|_2^2 = \min_{\mathbf{W} \in \mathcal{V}_{N_r}} \sum_{i=1}^{N_s} \|\mathbf{s}_i - \mathbf{W} \mathbf{W}^T \mathbf{s}_i\|_2^2 = \sum_{i=N_r+1}^r \sigma_i^2,$$

where,  $\sigma_i$  denotes the  $i$ -th singular value,  $\mathbb{U} \in \mathbb{R}^{N \times N_r}$  arises from the SVD of  $\mathbb{S}$  and is a full matrix containing the first  $N_r$  left singular vectors in columns corresponding to the  $N_r$  greatest singular values, and  $r$  is the number of non-zero singular values.

Indeed, Proposition 1.2 holds for the SVD of any matrix.

Therefore, the SVD provides a basis that minimizes the sum of the square of the errors between each snapshot vector  $\mathbf{w}_i$  and its projection onto the reduced subspace, and the error is equal to the sum of the squares of the neglected singular values. Furthermore, Proposition 1.2 provides a natural way to select the dimension of the reduced model, as we can select the higher singular values until a certain tolerance criteria is fulfilled.

The general POD procedure to obtain the reduced basis and dimension, either for problem (1), or problem (6), can be summarized as follows:

1. First, we need to solve the original problem for several instances of the parameter vector  $\gamma \in \mathcal{D}_{train} \subset \mathcal{D}$ , where the solution is qualitatively similar, to obtain the snapshots in time, and store them in the snapshot matrix,  $\mathbb{S}$ .
2. Then, we have to compute the SVD of the snapshot matrix  $\mathbb{S} = \mathbb{U} \Sigma \mathbb{V}^T$ , and sort, in descending order, the singular values and its corresponding left singular vectors.
3. Finally, we can apply Proposition 1.2 as a measure of the information lost to select the least amount of basis functions that retain the most information, according to a certain fixed tolerance. For example, if we define  $\epsilon_{tol} > 0$  as the tolerance, then we can use a suitable criterion involving the information retained by each mode, that is,

$$I(N_r) = \frac{\sum_{i=1}^{N_r} \sigma_i^2}{\sum_{i=1}^r \sigma_i^2} \geq 1 - \epsilon_{tol}^2,$$

or we can choose the first integer  $N_r$  such that  $\sigma_{N_r}^2 \leq \epsilon_{tol}^2$ . This will provide the dimension of the reduced model  $N_r$ , along with a submatrix of  $\mathbb{U}$ , the matrix of left singular vectors, that contains the basis functions,  $\{\phi_i\}_{i=1}^{N_r}$ .

There are two main issues regarding the POD technique. On the one hand, we need to know the solution for several values of the parameter  $\mathcal{D}_{train}$ , and, as said before, we are trying to avoid the resolution of the computationally expensive original problem. On the other hand, as we have no way to determine the best size for  $\mathcal{D}_{train}$ , it can be extremely large, and even if we can afford the resolution of the original problem, the computation of the SVD of the snapshot matrix could also be very challenging.

The POD technique will be present in the two first parts of the dissertation. To begin with, in Part I, we develop a technique based on the POD that allows to retain the information of the structure of problems like (2). After that, in Part II, we will use a coupled strategy with the Greedy algorithm (see Section 1.3.2), the so-called coupled POD+Greedy algorithm, that we will introduce in more detail in Section 1.3.3.

Now, we proceed to explain another method for the construction of the reduced problem, the Greedy Algorithm.

### 1.3.2 Greedy Algorithm

In order to overcome the issue of having to solve the original problem for several instances of the parameter vector  $\gamma$ , we could think of an optimal way to sample  $\mathcal{D}$ , avoiding repeating information that we have already captured. The Greedy Algorithm is an iterative algorithm that has been proven useful for the construction of the reduced problem (Binev et al., 2011; Buffa et al., 2012; Quarteroni et al., 2016).

The Greedy Algorithm is based on iteratively enriching the reduced problem to obtain better approximations of the original solution. In each iteration of the Greedy Algorithm, we add to the reduced problem the information of the original solution of the parameter vector value that provides the largest error between the original solution and the solution of the reduced problem.

In the following, we consider that we are working with an elliptic problem (4) or with steady solutions of problem (6). For the case of unsteady solutions we refer to Section 1.3.3, where a coupled strategy with the POD technique is presented. We also assume that we have a discrete subset of the parameter space  $\mathcal{D}_{train} \subset \mathcal{D}$ .

The general Greedy Algorithm to obtain the reduced basis and dimension, can be summarized as follows:

1. At the first iteration, select an initial parameter vector value  $\gamma_1 \in \mathcal{D}_{train}$ , define the set of parameter values  $S_1 = \{\gamma_1\}$ , and the reduced space  $X_0 = \{0\}$ .
2. Then, at any iteration  $i = 1, \dots$ :
  - a) Compute the (steady) solution of the original problem,  $u_h(\gamma_i)$ .
  - b) Add the computed solution to the reduced space,  $X_i = \text{span}\{X_{i-1}, u_h(\gamma_i)\}$ .
  - c) Solve the reduced problem for every parameter value  $\gamma \in \mathcal{D}_{train}$  to obtain  $u_i(\gamma)$ .
  - d) Compute the errors between the original solution and the solutions of the reduced problem  $\|u_h(\gamma) - u_i(\gamma)\|$ , for every  $\gamma \in \mathcal{D}_{train}$ , where  $\|\cdot\|$  is the norm of the corresponding space.
  - e) If  $\max_{\gamma \in \mathcal{D}_{train}} \|u_h(\gamma) - u_i(\gamma)\|$  is lower than a certain tolerance  $\epsilon_{tol}$ , then stop the algorithm. Otherwise, set  $\gamma_{i+1} = \arg \max_{\gamma \in \mathcal{D}_{train}} \|u_h(\gamma) - u_i(\gamma)\|$ , let  $S_{i+1} = S_i \cup \{\gamma_{i+1}\}$ , and repeat from instruction a).

However, as it states, the computation of the error  $\|u_h(\gamma) - u_i(\gamma)\|$  at each step of the Greedy Algorithm still requires the knowledge of the solution of the original problem for every  $\gamma \in \mathcal{D}_{train}$ . So, the general Greedy Algorithm does not prove an improvement against the POD technique. Bearing that in mind, in practice, instead of computing the exact error, we can consider an *a posteriori* error estimator  $\Delta_i(\gamma)$ , whose computation is cheaper with respect to the computation of the exact error. The resulting algorithm, changing the exact error by an *a posteriori* error estimator, is the so-called Weak Greedy Algorithm (Binev et al., 2011; Quarteroni et al., 2016), and it is the usual one for the construction of the reduced problem.

A further difference with respect to the POD technique is that the basis obtained for the reduced space is not orthonormal by construction. Therefore, in order to avoid redundancies and possible numerical issues, the reduced space should be orthonormalized with respect to the norm of the original space. For this purpose, we will use the Gram-Schmidt orthonormalization process when needed.

The (Weak) Greedy Algorithm will be present in the two latter parts of this dissertation. On the one hand, in Part II, we will use a coupled strategy with the POD technique (see Section 1.3.1) for unsteady problems, that we will explain in more detail in Section 1.3.3. On the other hand, in Part III, a Greedy algorithm based on the Gradient Descent Algorithm in Grassmann manifolds is presented to obtain the modes of the Proper Generalized Decomposition expansion. We will explain this procedure more in detail in Chapter 7.

### 1.3.3 POD+Greedy algorithm

The POD technique (Section 1.3.1) and the Greedy algorithm (Section 1.3.2) can be applied successfully in the case of elliptic or steady problems depending on parameters with no further complexity. However, in the case of unsteady problems, the POD technique can be unaffordable because of the size of the snapshots we will have to consider, and the Greedy Algorithm may converge very slowly due to the variability of the snapshots.

For those reasons, we introduce a coupled strategy for unsteady problems combining both, the POD technique and the Greedy Algorithm. We call it the POD+Greedy strategy. The main idea behind this coupled strategy is to use the POD technique considering the time  $t$  as the parameter, and the Greedy Algorithm for the parameter vector  $\gamma$ . The whole process for the coupled POD+Greedy strategy can be summarized as follows:

1. At the first iteration, select an initial parameter vector value  $\gamma_1 \in \mathcal{D}_{train}$ , define the set of parameter values  $S_1 = \{\gamma_1\}$ , and the reduced space  $X_0 = \{0\}$ .
2. Then, at any iteration  $i = 1, \dots$ :
  - a) Compute the solution of the original problem for an interval of time,  $\{u_h^k(\gamma_i)\}_{k=1}^{N_t}$ , with  $u_h^k(\gamma_i) = u_h(t_k, \gamma_i)$ , and  $t_k = k\Delta t$ .
  - b) Update the reduced space by means of the POD technique, taking into account the previous reduced space  $X_{i-1}$  and the information about the solution for the new parameter. We obtain the next reduced space  $X_i$  of dimension  $N$ .
  - c) Solve the reduced problem for every parameter value  $\gamma \in \mathcal{D}_{train}$  to obtain  $u_N(\gamma)$ .
  - d) Compute the errors between the original solution and the solutions of the reduced problem  $\|u_h(\gamma) - u_N(\gamma)\|$ , for every  $\gamma \in \mathcal{D}_{train}$ , where  $\|\cdot\|$  is the norm of the corresponding space.
  - e) If  $\max_{\gamma \in \mathcal{D}_{train}} \|u_h(\gamma) - u_N(\gamma)\|$  is lower than a certain tolerance  $\epsilon_{tol}$ , then stop the algorithm. Otherwise, set  $\gamma_{i+1} = \arg \max_{\gamma \in \mathcal{D}_{train}} \|u_h(\gamma) - u_N(\gamma)\|$ , let  $S_{i+1} = S_i \cup \{\gamma_{i+1}\}$ , and repeat from instruction a).

Point 2 of the previous strategy admits several approaches, we present here some of the most common ones. For further information on the coupled POD+Greedy strategy, we refer to Haasdonk (2013) and Quarteroni et al. (2011).

- **Complete POD:** We store the solutions for all the previously-selected parameters, and we perform the POD to the complete matrix of snapshots. This method will be optimal in the sense of Proposition 1.2 as there is no loss of information. However, performing a POD procedure to the complete matrix of snapshots can be computationally very expensive.
- **Hierarchical POD:** We perform a POD to the snapshots corresponding to the current selected parameter. Then, we add the new basis functions to the previous reduced basis, so  $X_{i-1} \subset X_i$ . It is called hierarchical in the sense that the basis at each step contains the previous basis. However, there could be a repetition of information. For more information about this procedure, see Haasdonk (2017).
- **Incremental HAPOD:** It follows the same procedure as the Hierarchical POD, but we perform a second POD procedure to the resulting basis. This will avoid the possible repetition of information. However, we can incur in some loss of information. For more information about this procedure, see Himpe et al. (2018). This procedure has been used in Caravaca García (2022).
- **Reduced POD:** We add the snapshots corresponding to the current selected parameter to the previous reduced basis that is  $\tilde{X}_i = \{u_h^k(\gamma_i)\}_{k=1}^{N_t} \cup X_{i-1}$ . Then, we perform a POD procedure to obtain the new reduced basis  $X_i$ . This procedure could provide a lesser reduced dimension, but this could be because of a loss of information.

The coupled POD+Greedy strategy will be used in Part II of this dissertation, applied to the study of an unsteady periodic flow, where we will also perform a study comparing the presented POD approaches.

#### 1.4 TREATMENT OF NONLINEAR TERMS IN REDUCED ORDER MODELLING

One of the main drawbacks of the techniques presented in the previous is that they fail to provide approximate solutions when we are considering problems that present nonlinearities with respect to time and/or the parameter vector. An accurate treatment

of these terms is crucial, so, in order to tackle this problematic, we present in this section a technique that allows the treatment of these terms in the reduced problem.

### 1.4.1 Empirical Interpolation Method

We introduce here the EIM, presented in Barrault et al. (2004), and then extended in Grepl et al. (2007). The main objective of the EIM is to build a linear approximation of the nonlinear function with respect to time and/or the parameter vector. The EIM relies on the computation of an interpolant function over a properly selected set of points, called *magic points* (Maday et al., 2009). Therefore, any nonlinear function  $g$  can be approximated by

$$g(\mathbf{x}; t, \gamma) \approx g_M(\mathbf{x}; t, \gamma) = \sum_{i=1}^M \sigma_i(t, \gamma) q_i(\mathbf{x}), \quad \forall t \in I_f \text{ and } \gamma \in \mathcal{D},$$

for  $\sigma_i$  and  $q_i$  with  $i = 1, \dots, M$ , to be determined. Here  $\mathbf{x}$  can represent the variables, if we are working in the framework of problem (1), or points of the domain  $\Omega$ , in the case of working in the field of parametric PDEs (3). In both cases, the following mathematical treatment is similar, so we will make no further distinction.

In the following, we will consider the time as a parameter, and, with a slight abuse of notation, we name the extended parameter vector also as  $\gamma$ . We also consider a subset of the parameter space  $\mathcal{D}_{EIM} \subset \mathcal{D}$ . Then, we can summarize the procedure of the EIM algorithm to build the reduced version of the nonlinear terms as follows:

1. At the first iteration, select an initial parameter vector value  $\gamma_1 \in \mathcal{D}_{EIM}$ , and define the set of selected parameter vectors  $S_1^{EIM} = \{\gamma_1\}$ , the set of *magic points*  $T_0 = \emptyset$ , the set of functions  $W_0 = \emptyset$ , and the interpolation function  $I_0(\mathbf{x}) = 0$ .
2. Then, at any iteration  $M = 1, \dots$ :
  - a) Compute the nonlinear term and subtract the interpolation function, that is,

$$g_M(\mathbf{x}) = g(\mathbf{x}; \gamma_M) - I_{M-1}(\mathbf{x}).$$

- b) Define  $\mathbf{x}_M = \arg \sup_{\mathbf{x}} |g_M(\gamma)|$ , and  $q_M(\mathbf{x}) = \frac{g_M(\mathbf{x})}{g_M(\mathbf{x}_M)}$ .
- c) Update the set of *magic points*  $T_M = T_{M-1} \cup \{\mathbf{x}_M\}$  and the set of functions  $W_M = W_{M-1} \cup \{q_M\}$ .

- d) Set the matrix  $\mathbb{B}_{ij}^M = q_j(\mathbf{x}_i)$ , with  $1 \leq i, j \leq M$ , and solve for every  $\gamma \in \mathcal{D}_{EIM}$  the following linear system

$$\sum_{j=1}^M \mathbb{B}_{ij}^M \sigma_j(\gamma) = g(\mathbf{x}_i, \gamma), \quad 1 \leq i \leq M.$$

- e) For every  $\gamma \in \mathcal{D}_{EIM}$ , we need to compute the interpolation function  $I_M(\mathbf{x}) = \sum_{i=1}^M \sigma_i(\gamma) q_i(\mathbf{x})$ , and the interpolation errors,  $\|g(\mathbf{x}; \gamma) - I_M(\mathbf{x})\|_\infty$ .
- f) If  $\max_{\gamma \in \mathcal{D}_{EIM}} \|g(\mathbf{x}; \gamma) - I_M(\mathbf{x})\|_\infty$  is lower than a certain tolerance  $\epsilon_{tol}$ , then stop the algorithm. Otherwise, set

$$\gamma_{i+1} = \arg \max_{\gamma \in \mathcal{D}_{EIM}} \|g(\mathbf{x}; \gamma) - I_M(\mathbf{x})\|_\infty,$$

let  $S_{i+1}^{EIM} = S_i^{EIM} \cup \{\gamma_{i+1}\}$ , and repeat from instruction a).

Now, if we assume that the EIM algorithm stopped at step  $M$ , when we solve the reduced problem, we need to compute the nonlinear terms in the following way:

1. Compute the nonlinear term in the set of *magic points*  $T_M$ , that is,  $\mathbf{g}_M = (g_i)$ , with  $g_i = g(\mathbf{x}_i; \gamma)$ , for every  $\mathbf{x}_i \in T_M$ .
2. Solve the linear system  $\mathbb{B}^M \boldsymbol{\sigma} = \mathbf{g}_M$ .
3. Compute the interpolation function  $g(\mathbf{x}; \gamma) \approx I_M = \sum_{i=1}^M \sigma_i(\gamma) q_i(\mathbf{x})$ .

Nonlinear terms will appear in the first two parts of this dissertation, and the use of the EIM will be needed in both cases. On the one hand, in Part I, the EIM will be used in the context of nonlinearities of a system of parametric differential equations with multiple timescales, like problem (2). On the other hand, in Part II, the EIM will be used to compute the eddy viscosity in unsteady turbulent flows.

# 2

---

## MATHEMATICAL MODELS ADDRESSED

---

This chapter is devoted to the presentation of two models of interest for the rest of the thesis. On the one hand, we present in Section 2.1 a dynamical system that exhibits multiple timescales. In particular, we present some models of neuronal Intracellular Calcium Concentration (ICC) dynamics. On the other hand, in Section 2.2, we present a mathematical model of turbulence, the Smagorinsky turbulence model, and its associated reduced basis problem.

### 2.1 MODELS OF NEURONAL INTRACELLULAR CALCIUM CONCENTRATION DYNAMICS

In this section, we focus on the presentation of one example of a dynamical system that exhibits different timescales. In particular, we present some models of neuronal Intracellular Calcium Concentration (ICC) dynamics (Bandera Moreno et al., 2022; Fernández-García & Vidal, 2020; Krupa et al., 2013), that exhibit one fast variable and two slow ones.

We begin with the single-cell model and then progress to the interconnected two-cell model. We present here the main behaviors that appear in each of the models. In subsequent chapters, we will consider networks built from these two models.

#### 2.1.1 *Single-cell model of neuronal Intracellular Calcium Concentration dynamics*

The 3D slow-fast oscillator described in Krupa et al. (2013) serves as a single-cell system specifically crafted to model ICC variations within a single neuron. Through comparisons with experimental data, this first model has demonstrated its effectiveness in encapsulating both, the fundamental qualitative and quantitative characteristics of ICC oscillations in GnRH neurons and motoneurons. Additionally, the authors in

Krupa et al. (2013) detailed a straightforward method to adjust parameters, enabling the replication of desired frequencies, amplitudes, and durations of quiescence phases within ICC oscillations.

The system, with  $(x, y, z) \in \mathbb{R}^3$ , reads

$$\begin{cases} \dot{x} = \tau(-y + f(x) - \phi_f(z)), \\ \dot{y} = \tau\varepsilon(x + a_1y + a_2), \\ \dot{z} = \tau\varepsilon \left( \phi_r(x) - \frac{z - z_b}{\tau_z} \right), \end{cases} \quad (10)$$

where,

$$\phi_f(z) = \frac{\mu z}{z + z_0}, \quad \text{and} \quad \phi_r(x) = \frac{\lambda}{1 + \exp(-\rho(x - x_{on}))}.$$

Parameter  $\varepsilon$  fits the timescale separation and therefore  $0 < \varepsilon \ll 1$ , and  $\tau, \tau_z$  are introduced to fit the timescale of experimental data. As in the FitzHugh-Nagumo system (FitzHugh, 1961; Nagumo et al., 1962) and 3D extensions of such dynamics (see, for instance, Krupa et al. (2008) for a model of dopaminergic neurons), fast variable  $x$  stands for the cell electrical activity, while slow variable  $y$  acts as a recovery variable. Moreover, here slow variable  $z$  stands for the ICC, and its dynamics is essentially driven by sigmoid  $\phi_r$ , with a threshold on the electrical activity  $x_{on}$ . Variable  $z$  feedbacks onto the electrical activity through the Hill function  $\phi_f$  with upper bound  $\mu$ .

For  $\phi_r(x)$  close to zero,  $z$  decreases to a value close to  $z_b$  that is a quasi-steady state for  $z$  in case of electric inactivation and stands for ICC baseline. The decrease of variable  $z$  is an exponential decay with  $\tau\varepsilon/\tau_z$  rate. The coupling term  $\phi_f(z)$  onto  $x$  aims at reproducing the reduction of neuronal electrical activity when ICC increases. Such coupling has already been introduced in single neuron models to mimic the neuron hyperpolarization induced by calcium (Ermentrout et al., 2001).

The essential properties of ICC dynamics depend on the qualitative features of the coupling functions. These properties and the results of our study are preserved for other increasing functions  $\phi_f$  and  $\phi_r$  that are bounded on  $\mathbb{R}^+$ . For reproducing the excitable feature of calcium release from experimental data, function  $\phi_r$  is chosen stiff and a sigmoid is well-designed for this aim.

In the following, we consider the same parameters values as in the literature (Bandera Moreno et al., 2022; Fernández-García & Vidal, 2020). Therefore,  $f(x) = -x^3 + 4x$  and

$$\begin{aligned} a_1 &= -0.1, \quad a_2 = 0.8, \quad \varepsilon = 0.06, \quad \tau = 37, \quad \mu = 2.4, \quad z_0 = 5, \\ z_b &= 1, \quad \tau_z = 2, \quad x_{on} = -0.45, \quad \lambda = 1.75, \quad \rho = 4.5. \end{aligned}$$

In order to describe the main behaviors of system (10), we need to define the  $S$ -shaped surface  $S \equiv \{y = f(x) - \phi_f(z)\}$ , called the critical surface in the multiple timescale framework, that exhibits two fold lines:  $\mathcal{F}^- \equiv \{x = -x_f, y = f(-x_f) - \phi_f(z)\}$ , and  $\mathcal{F}^+ \equiv \{x = x_f, y = f(x_f) - \phi_f(z)\}$ .

Therefore, the critical surface consists of three sheets and the fold lines that separate them. The middle sheet  $S_m$ , contained in  $|x| < x_f$ , is repulsive for the fast dynamics. The left sheet  $S_l$ , contained in  $x < -x_f$ , as well as the right sheet  $S_r$ , contained in  $x > x_f$ , are attracting manifolds for the boundary-layer system, obtained for the limit case  $\varepsilon = 0$ ,

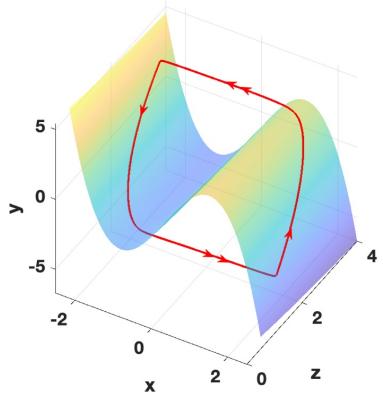
$$\begin{cases} \dot{x} = \tau(-y + f(x) - \phi_f(z)), \\ \dot{y} = 0, \\ \dot{z} = 0, \end{cases}$$

i.e. the fast  $x$ -dynamics with  $y, z$  considered as parameters.

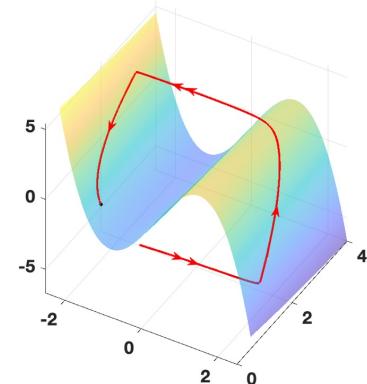
According to the location of system (10) singular point, three main asymptotic behaviors can be identified using slow-fast analysis, i.e. considering  $\varepsilon > 0$  small enough and using geometric singular perturbation theory (Fenichel, 1979). Rigorously, the phase portraits described below are obtained for any  $\varepsilon \in (0, \varepsilon_0]$  for a fixed  $\varepsilon_0 > 0$ .

We refer the reader to Fernández-García and Vidal (2020) and Krupa et al. (2013) for more detailed explanations of the following phase portraits. Here, we display figures 2.1a-2.1c for visual support. We distinguish the following three cases:

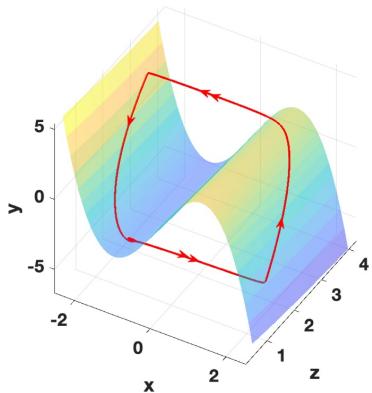
1. The singular point lies on  $S_m$  and far from the folds: then, it is unstable and there exists a globally attractive limit cycle of system (10) generating relaxation oscillations in the time series, see Figure 2.1a.
2. The singular point lies on  $S_l$ : then, it is asymptotically stable and any orbit of system (10) admits this point as  $\omega$ -limit, see Figure 2.1b. The phase portrait has the same properties if the singular point lies on  $S_r$ .
3. The singular point lies on  $S_m$  but close to a fold: then, it is unstable and system (10) generates Mixed-Mode Oscillations, see Figure 2.1c.



(a) Attractive relaxation limit cycle of the single-cell ICC model together with the S-shaped critical manifold: single and double arrows materialize the slow and fast parts of the limit cycle, respectively.



(b) Instance of an orbit of the single-cell ICC model for the case of existence of an attractive equilibrium (black point on the left sheet of the critical manifold). The system does not generate sustained oscillation.



(c) MMOs generated by the single-cell ICC model. Left panel: Periodic orbit together with the critical manifold. Right panel: magnified view of the small oscillations near the lower fold of the critical manifold.

**Figure 2.1:** Asymptotic behaviors of system (10) depending on the location of its singular point together with the S-shaped critical manifold. Single and double arrows materialize the slow and fast parts of the limit cycle, respectively.

### 2.1.2 Two coupled-cells model of neuronal Intracellular Calcium Concentration dynamics

The two-cells model is built as a linear coupling of two copies of system (10). In the following, as in Fernández-García and Vidal (2020) and in Bandera Moreno et al. (2022), we will denote  $(x_1, y_1, z_1)$ -subsystem as  $O_1$ , which depends on  $x_2$ , and  $(x_2, y_2, z_2)$ -subsystem as  $O_2$ , which depends on  $x_1$ . The system then reads as

$$\left. \begin{array}{l} O_1 \quad \left\{ \begin{array}{l} \dot{x}_1 = \tau(-y_1 + f(x_1) - \phi_f(z_1)), \\ \dot{y}_1 = \tau\varepsilon k_1(x_1 + a_1 y_1 + a_2 + c(x_1 - x_2)), \\ \dot{z}_1 = \tau\varepsilon \left( \phi_r(x_1) - \frac{z_1 - z_b}{\tau_z} \right), \\ \dot{x}_2 = \tau(-y_2 + f(x_2) - \phi_f(z_2)), \\ \dot{y}_2 = \tau\varepsilon k_2(x_2 + a_1 y_2 + a_2 + c(x_2 - x_1)), \\ \dot{z}_2 = \tau\varepsilon \left( \phi_r(x_2) - \frac{z_2 - z_b}{\tau_z} \right), \end{array} \right\} \\ O_2 \quad \left. \begin{array}{l} \end{array} \right\} \end{array} \right\} \quad (11)$$

with the coupling parameter  $c \in [-1, 1]$  and where parameters  $k_i, i = 1, 2$ , state the possible heterogeneity between both oscillators,  $O_1$  and  $O_2$ .

Let us describe the main behavioral patterns obtained in system (11). We distinguish two cases, on the one hand, we assume that the two cells are identical, that is,  $k_i = 1$ ,  $i = 1, 2$ , what we call the homogeneous case, that we explain in the following. On the other hand, we let the two cells to be slightly different, that is,  $k_i \neq 1$ , for some  $i = 1, 2$  and  $k_1 \neq k_2$ , what we call the heterogeneous case, and explain afterwards.

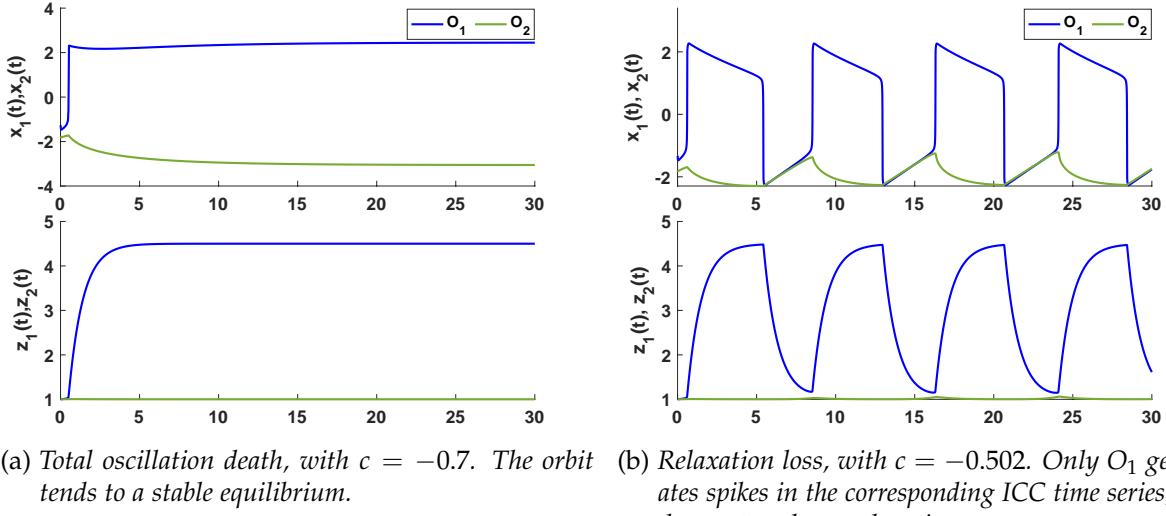
#### *Homogeneous case*

The case with  $k_i = 1, i = 1, 2$ , that is, the homogeneous case, was already studied in Fernández-García and Vidal (2020). In that work, the authors outline the different synchronization types between  $O_1$  and  $O_2$  within the attractive limit cycle of system (11) depending on the coupling strength  $c \neq 0$ .

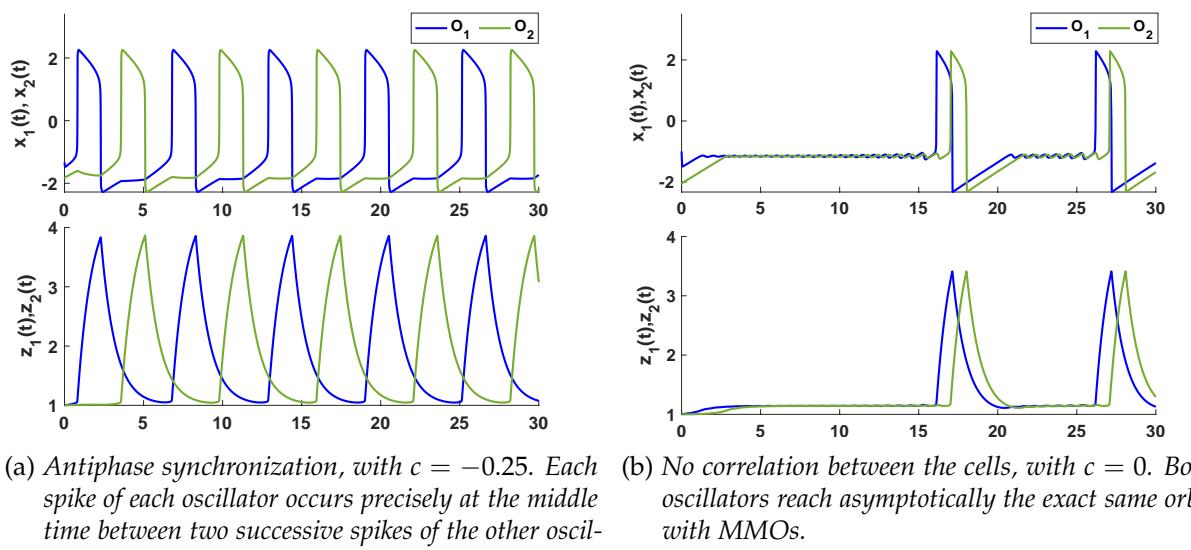
Here, we summarize the main asymptotic behaviors obtained in Fernández-García and Vidal (2020). We illustrate each case in figures 2.2-2.4.

1. *Total Oscillation Death (TOD):*  $O_1$  and  $O_2$  reach asymptotically a stable equilibrium. Figure 2.2a shows the behavior of system (11) in the time interval  $[0, 30]$  for  $c = -0.7$ .

2. *Relaxation Loss (RL)*: one oscillator produces relaxation oscillations, i.e. the orbit approaches both  $S_l$  and  $S_r$ , while the other one produces small oscillations and the orbit remains near  $S_l$ . Figure 2.2b shows the behavior of system (11) in the time interval  $[0, 30]$  for  $c = -0.502$ .



**Figure 2.2:** The  $x$  and  $z$  time series (top and bottom panel respectively) of each oscillator  $O_1$  and  $O_2$  are obtained from the two-cells model (11) for  $k_1 = k_2 = 1$  and different coupling parameters  $c < 0$ .



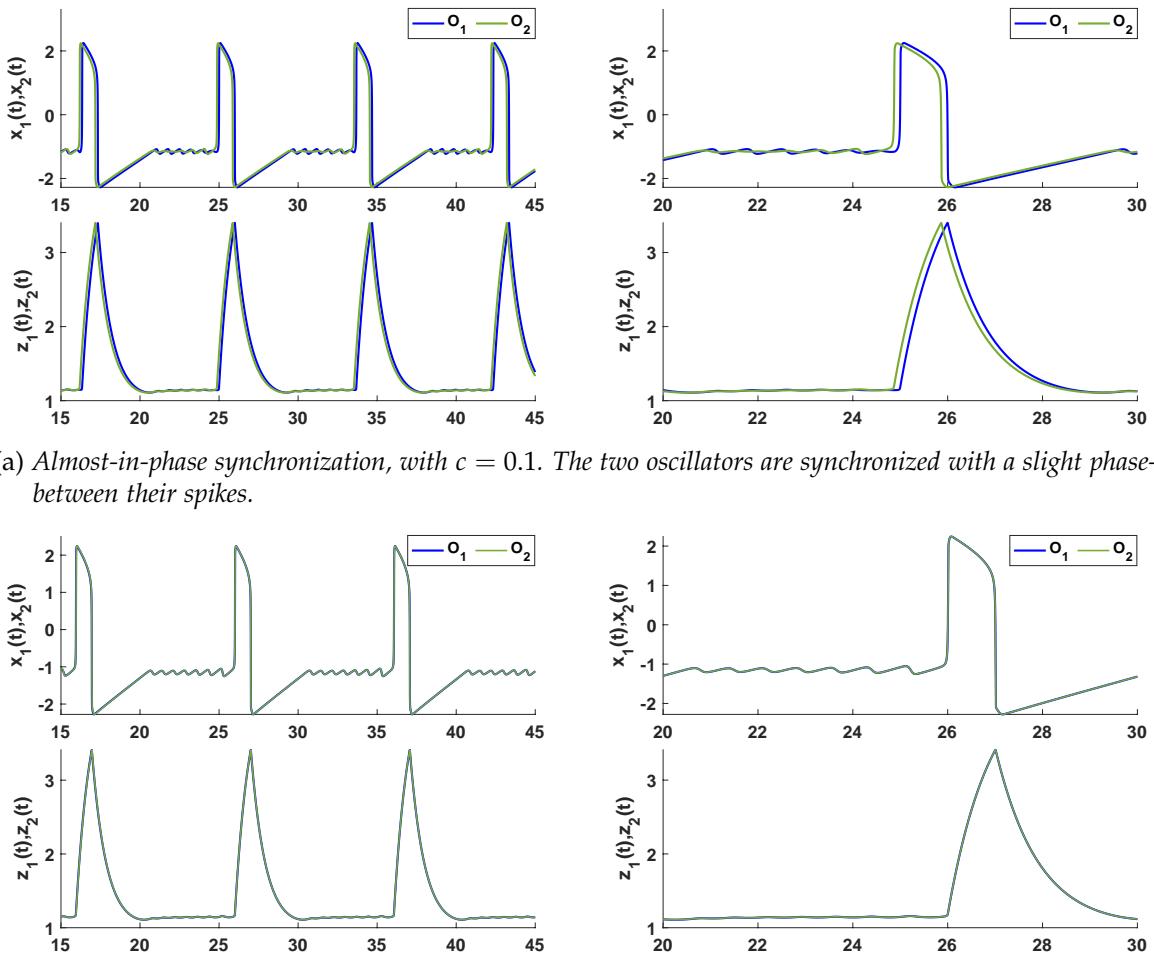
**Figure 2.3:** The  $x$  and  $z$  time series (top and bottom panel respectively) of each oscillator  $O_1$  and  $O_2$  are obtained from the two-cells model (11) for  $k_1 = k_2 = 1$  and different coupling parameters  $c \leq 0$ .

3. *Antiphase Synchronization (AS)*:  $O_1$  and  $O_2$  reach asymptotically the same orbit, and the phase-shift between them is exactly one half-period. Figure 2.3a shows the behavior of system (11) in the time interval  $[0, 30]$  for  $c = -0.25$ .
4. *Uncoupled case (U)*: when  $c = 0$  the oscillators  $O_1$  and  $O_2$  reach asymptotically the exact same orbit with MMOs. There is a phase shift between the oscillators, which depends on the initial conditions. The phase-shift will vanish if the initial conditions are identical. Figure 2.3b shows the behavior of system (11) in the time interval  $[0, 30]$  for  $c = 0$ .
5. *Almost-in-Phase synchronization (AiP)*: system (11) admits an attractive limit cycle along which  $O_1$  and  $O_2$  generate MMOs. The two 3D oscillators are synchronized with a small phase-shift. Figure 2.4a shows the behavior of system (11) in the time interval  $[15, 45]$  for  $c = 0.1$ , as well as a zoom of one of the peaks where we can see that the oscillators are not perfectly in phase.
6. *In-phase Locking synchronization (IL)*:  $O_1$  and  $O_2$  reach asymptotically the exact same orbit which is precisely the cycle existing for the uncoupled case. Therefore, the attractive limit cycle of system (11) lies on  $\{x_1 = x_2, y_1 = y_2, z_1 = z_2\}$  and is a global attractor of the system. Figure 2.4b shows the behavior of system (11) in the time interval  $[15, 45]$  for  $c = 1$ , as well as a zoom of one of the peaks.

In Fernández-García and Vidal (2020), the distribution of these behaviors according to the coupling parameter  $c$  has been theoretically studied and numerically shown for symmetric coupled system (11) (with  $k_1 = k_2$ ).

Furthermore, through numerical exploration and heuristic arguments, the authors pinpointed distinctive behaviors emerging within narrow ranges of  $c$  values, representing transitions between the aforementioned primary behaviors. To gain deeper insights into the organization of these behaviors and the significance of the most exceptional ones, examining a broader scenario and removing the constraint of symmetry between the two oscillators proves beneficial.

Although the theoretical basis relying on this symmetry loses validity, the evanescent behaviors identified in Fernández-García and Vidal (2020) become central behaviors when considering  $k_1 \neq k_2$ . The role of the heterogeneity parameters and of the coupling parameter in organizing the dynamical structure can be clarified through a numerical investigation. We will discuss this point in more detail in the subsequent section.



(a) Almost-in-phase synchronization, with  $c = 0.1$ . The two oscillators are synchronized with a slight phase-shift between their spikes.

(b) In-phase synchronization, with  $c = 1$ . The two oscillators spikes are perfectly synchronized. More precisely, the time difference between corresponding spikes of  $O_1$  and  $O_2$  tends asymptotically to 0.

**Figure 2.4:** The  $x$  and  $z$  time series (top and bottom panel respectively) of each oscillator  $O_1$  and  $O_2$  are obtained from the two-cells model (11) for  $k_1 = k_2 = 1$  and  $c > 0$ .

#### Heterogeneous case

If some heterogeneity is introduced between the two oscillators by considering different values for the recovery variable times-scale parameters  $k_1$  and  $k_2$ , then, the uncoupled oscillators ( $O_1$  and  $O_2$  with  $c = 0$ ) admit attractive limit cycles with different intrinsic frequencies. Moreover, the oscillatory time series generated by the ICC variables  $z_1$  and  $z_2$  along the corresponding limit cycles feature different amplitudes. The theoretical study performed in Fernández-García and Vidal (2020) for the homogeneous case is strongly based on symmetry, therefore it cannot be directly adapted to this case. We present here, the results obtained using numerical simulations for the study of the heterogeneous case, performed in Bandera Moreno et al. (2022).

In the following, we fix  $k_1 = 1$  and consider  $k_2 = k > 1$ , since the results for  $k < 1$  can be deduced by exchanging the roles of  $O_1$  and  $O_2$ . We further explain the synchronization patterns obtained in the heterogeneous case.

- For  $c > 0$ , while  $k \in (1, 2]$ , we obtain the same qualitative behavior as in the homogeneous case ( $k = 1$ ).
- For  $c < 0$ , we find the three behaviors obtained in the homogeneous case, but in the antiphase region  $c \in [c^0 + \delta, -\delta]$  with  $c^0 \approx -0.5$ , and  $0 < \delta \ll 1$ , besides the antiphase behavior, we obtain new synchronization patterns for  $k \in (1, 3]$  that we explain subsequently.

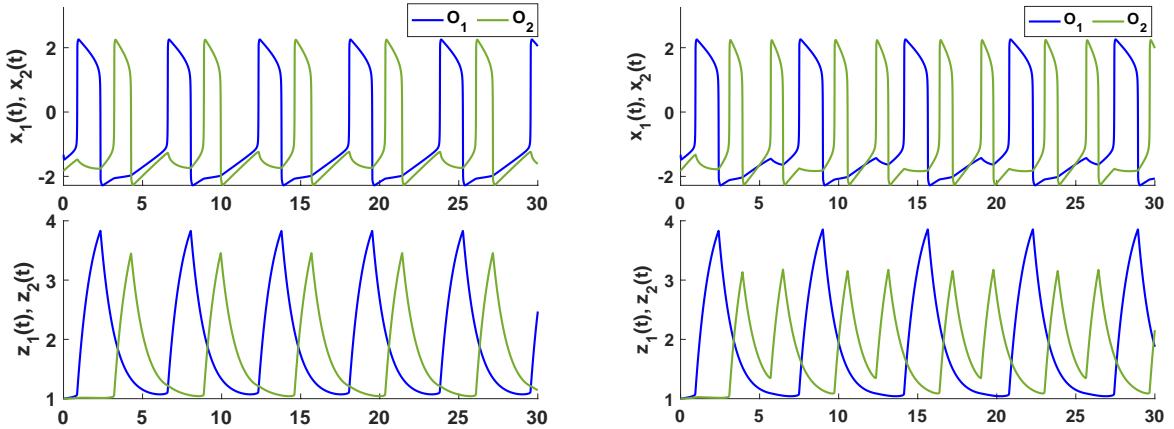
First, we need to present a definition that will help us to distinguish between the different behaviors.

### **Definition 2.1**

*In the following, we say that the signature of a periodic orbit of system (11) is  $s/1$  if, along one period of this orbit, one 3D oscillator spikes  $s$  times while the other one spikes once.*

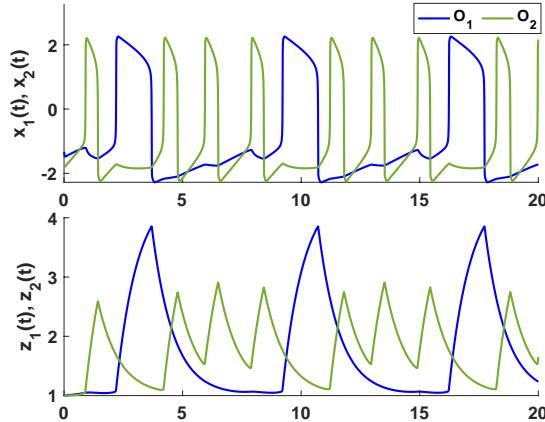
Hence, with this notation, the antiphase behavior of the homogeneous case, (see Figure 2.3a), would be denoted as  $1/1$ . Let us focus on the case  $c = -0.25$  and describe the main and transitory patterns.

- For  $k \in (1, 1.5563]$ , we obtain a behavior qualitatively similar to the one obtained in the homogeneous case, one oscillation for one oscillation (signature  $1/1$ ), see Figure 2.5a for a specific instance where  $k = 1.5$ . However, as we can observe, the behavior is not in antiphase, as it is the case in the homogeneous case (see Figure 2.3a), because the phase shift is not exactly half of the period. This is an indicator of how the heterogeneity is affecting the system. We refer to the behavior shown in Figure 2.5a as "phase-shifted synchronization".
- For  $k \in [1.6468, 2.7350]$ , one oscillator generates two pulses while the other one generates a single one along a limit cycle period, which corresponds to signature  $2/1$ . See Figure 2.5b for a specific instance where  $k = 2$ .
- For  $k \in [2.7582, 3]$ , the behavior of the system changes and one oscillator generates three pulses while the other one generates a single one along a limit cycle period, which corresponds to signature  $3/1$ . See Figure 2.5c for a specific instance where  $k = 3$ .



(a) Signature 1/1, with  $k = 1.5$ . Each oscillator spikes a single time between two consecutive spikes of the other one.

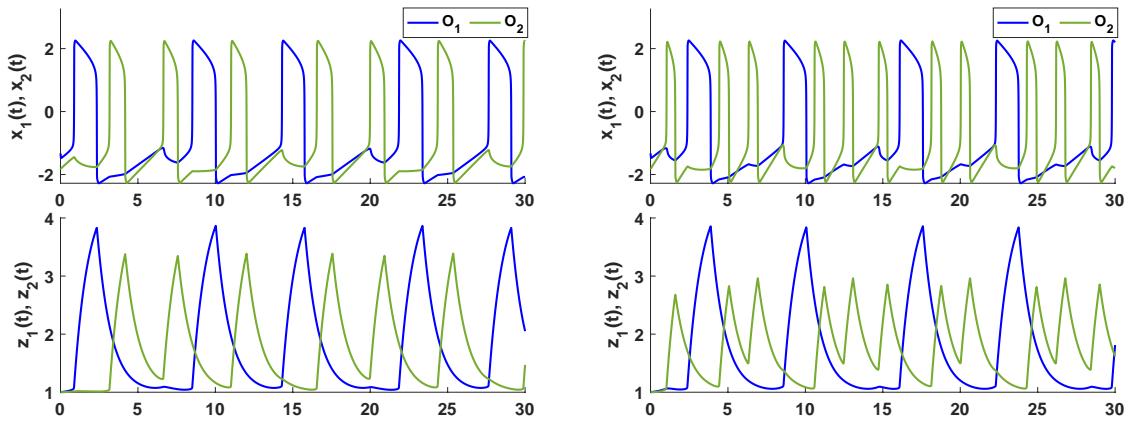
(b) Signature 2/1, with  $k = 2$ . Oscillator  $O_2$  spikes precisely two times between two consecutive spikes of oscillator  $O_1$ .



(c) Signature 3/1, with  $k = 3$ . Oscillator  $O_2$  spikes precisely three times between two consecutive spikes of oscillator  $O_1$ .

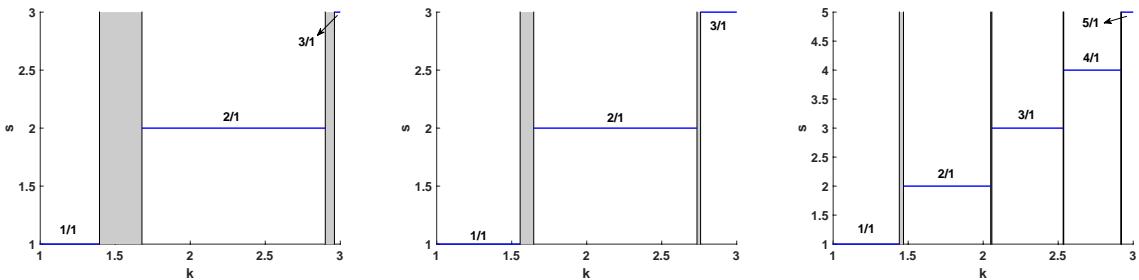
**Figure 2.5:** Phase-shifted synchronization with different signatures. The time series are generated with the two-cells model (11) with  $k_1 = 1$  and varying  $k_2 = k$ .

- For  $k \in [1.5592, 1.6454]$ , in the transition from signature 1/1 to 2/1, the system can exhibit a mixture of two signatures: between two successive spikes of  $O_1$ , oscillator  $O_2$  spikes one and two times, alternatively. Therefore, two signatures 1/1 and 2/1 alternate along the limit cycle. See left panel of Figure 2.6.
- For  $k \in [2.7372, 2.7563]$ , in the transition from signature 2/1 to 3/1, a similar case occurs: between two successive spikes of  $O_1$ , oscillator  $O_2$  spikes two and three times, alternatively. Therefore, two signatures 2/1 and 3/1 alternate along the limit cycle. See Figure 2.6, right.



**Figure 2.6:** Transitory signatures of system (11) outputs while varying  $k$ . Parameter  $k_1 = 1$ . Left panel: for  $k = 1.6$ , the signature alternates between 1/1 and 2/1. Right panel: for  $k = 2.74$ , the signature alternates between 2/1 and 3/1.

In Figure 2.7, we represent the signature of system (11) limit cycle for  $c = -0.05$  (left panel),  $c = -0.25$  (central panel) and  $c = -0.40$  (right panel) according to the value of parameter  $k$ . The blue lines represent the signature value when system (11) admits a limit cycle, while gray bands materialize the intervals of  $k$  value where the signature is not well-defined, corresponding to signature transitions. We note that the representation of the different behaviors in Figure 2.7 exhibits a Devil Staircase-like behavior, characterized by different stable behaviors for large interval of one parameter values, separated by transition intervals where the system can exhibit a mixture of these stable behaviors, and even chaotic behaviors, (Bak, 1986).

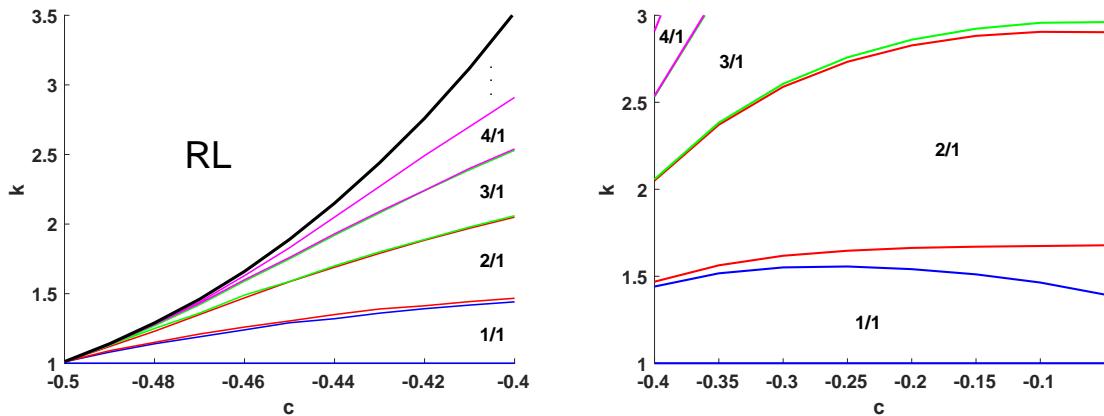


**Figure 2.7:** Signature of the two-cells model (11) limit cycle according to  $k$  values for  $c = -0.05$  (left panel),  $c = -0.25$  (central panel) and  $c = -0.40$  (right panel). The gray bands materialize the intervals of  $k$  values for which the signature is not well-defined.

Note that there are some ranges of the parameter values inside the transitions where the system does not exhibit one of the behaviors described above, but a more complex combination of the main behaviors. In fact, we also have a canard explosion,

that is, a very fast transition from small oscillations to relaxation oscillations, with a small variation of the parameters (Benoit et al., 1981; Krupa & Szmolyan, 2001). This transition happens via a family of cycles whose characteristic property is that they contain *canard segments*, i.e. segments of the orbit that follow the repulsive slow manifold  $S_m$  for a time  $O(1)$ .

Now, that we have detailed the observed patterns in the case  $c = -0.25$ , let us come back to the general case. We study the behaviors for  $k \in (1, 3]$  in the range of the coupling parameter  $c \in [-0.4, 0)$ , we obtain the subdomains in the  $(c, k)$ -plane where the system (11) exhibits each main behavior, as shown in Figure 2.8 right, where the region  $c \in [-0.4, -0.05]$  and  $k \in [1, 3]$  is displayed. We can observe that the transition intervals shrink while the coupling parameter values are lowered.



**Figure 2.8:** Regions of  $(c, k)$ -plane associated with fixed signature value of the attractive limit cycle of the two-cells model (11). The left panel shows that the intervals of  $k$  values corresponding to a given signature converge to the singleton  $\{1\}$ , while  $c$  decreases to  $-0.5$ . The right panel emphasizes the small strips in  $(c, k)$ -plane for which the signature is not well-defined.

Finally, we focus now on the range near the transition to the relaxation loss behavior, namely  $c \in [-0.5, -0.4]$ . All ranges of the main behaviors converge to the transition point for  $k = 1$ , see Figure 2.8 left, showing  $c \in [-0.5, -0.4]$  and  $k \in [1, 3.5]$ . We can also observe that the transition to the relaxation loss regime depends on the heterogeneity parameter  $k$ .

The ICC models will be used in Part I of this dissertation. Where we present the two clusters model, along with its main behaviors. However, simulating large-scale network models can be computationally expensive, and while the POD method approximate accurately the solution of the original system (Bandera Moreno et al., 2022), they may lead to the loss of the problem's original structure. Therefore, the main objective of

Part I will be to develop a reduced order model procedure based on the POD, in order to preserve the two clusters structure of model.

## 2.2 REDUCED ORDER MODEL FOR THE SMAGORINSKY TURBULENCE MODEL

This section will be devoted to the presentation of the basic Large Eddy Simulation (LES) turbulence model, the celebrated Smagorinsky turbulence model introduced in Smagorinsky (1963). In this model, the effect of the subgrid scales on the resolved scales is modeled by eddy diffusion terms (Chacón Rebollo & Lewandowski, 2014; Sagaut, 2005; Smagorinsky, 1963), that is, a characterization of the transport and dissipation of energy in the smaller scales.

Although being intrinsically discrete, we present in a first place a continuous model whose discretization can be interpreted as the Smagorinsky turbulence model. The presentation of the continuous model is included to clarify the relationship between the Navier-Stokes model with the Smagorinsky turbulence model. Then, we will derive the respective reduced order model.

### 2.2.1 Smagorinsky turbulence model

In this section, we present the Smagorinsky model as a continuous one, although it is intrinsically discrete, associated to a given discretization grid. We will just specify its dependency on the actual grid used in the expression of the eddy viscosity terms. Then, we present the Smagorinsky turbulence model in the discrete framework and some of its properties.

We thus consider a polygonal, if  $d = 2$ , or polyhedral, if  $d = 3$ , bounded domain  $\Omega \subset \mathbb{R}^d$ . We assume that its boundary is split into  $\Gamma = \Gamma_D \cup \Gamma_N$ , where  $\Gamma_D$  is the boundary relative to the Dirichlet boundary conditions, and  $\Gamma_N$  to the Neumann conditions. In order to specify the expression of the eddy viscosity, we will also consider a triangular mesh  $\{\mathcal{T}_h\}_{h>0}$  of  $\Omega$ , that we assume to be regular in the sense of Ciarlet (Ciarlet, 2002).

In the following, we will consider that the model depends on just one parameter, the Reynolds number, denoted by  $\gamma \in \mathcal{D}$ . Here, the parametric set  $\mathcal{D}$ , stands for the positive real numbers. The Reynolds numbers should be large enough to ensure that the flow is in well-developed turbulent regime. For clarity, we will also drop the dependence of  $\mathbf{u}$  and  $p$  on space and time.

Given a targeted final time  $T_f$ , we look for a velocity and pressure fields,  $\mathbf{u} \in L^2(I_f, H^1(\Omega)^d)$ , and  $p \in L^2(I_f, L_0^2(\Omega))$ , respectively, such that the next equation is verified

$$\left\{ \begin{array}{ll} \partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nabla \cdot \left( \left( \frac{1}{\gamma} + \nu_T(\mathbf{u}) \right) \nabla \mathbf{u} \right) = \mathbf{f}, & \text{in } I_f \times \Omega, \\ \nabla \cdot \mathbf{u} = \mathbf{0}, & \text{in } I_f \times \Omega, \\ \mathbf{u} = \mathbf{g}, & \text{on } \Gamma_D, \\ -p \mathbf{n} + \left( \frac{1}{\gamma} + \nu_T(\mathbf{u}) \right) \frac{\partial \mathbf{u}}{\partial \mathbf{n}} = 0, & \text{on } \Gamma_N. \end{array} \right. \quad (12)$$

In equation (12),  $\mathbf{f} \in L^2(I_f, L^2(\Omega)^d)$  denotes the kinetic momentum source and  $\mathbf{g} \in L^2(I_f, H^{1/2}(\Omega)^d)$  denotes the Dirichlet boundary conditions. The term  $\nu_T(\mathbf{u})$  stands for the eddy viscosity, given by

$$\nu_T(\mathbf{u}) = C_S^2 \sum_{K \in \mathcal{T}_h} h_K^2 |\nabla \mathbf{u}|_K \chi_K, \quad (13)$$

where  $C_S$  denotes the Smagorinsky constant estimated to be  $C_S \approx 0.18$  (Lilly, 1967; Rubino, 2014),  $h_K$  denotes the diameter of element K of the triangulation, and  $|\cdot|$  denotes the Frobenius norm in  $\mathbb{R}^{d \times d}$ .

If we were to consider  $\mathbf{g} \neq \mathbf{0}$ , then we should assume that there exists a lift function  $\mathbf{u}_D \in H^1(\Omega)^d$ , such that it is incompressible and satisfies the Dirichlet boundary conditions. With those conditions, we assure that the homogenized velocity  $\mathbf{w} = \mathbf{u} - \mathbf{u}_D$  is still incompressible and satisfies homogeneous boundary conditions on  $\Gamma_D$ . For simplicity, in the following, we will consider the case where  $\mathbf{g} = \mathbf{0}$ . Therefore,  $\mathbf{u} \in L^2(I_f, H_{\Gamma_D}^1(\Omega)^d)$ , with  $H_{\Gamma_D}^1(\Omega)^d := \{v \in H^1(\Omega)^d \text{ such that } v|_{\Gamma_D} = 0\}$ .

Let  $X_h \subset H_{\Gamma_D}^1(\Omega)^d$  and  $M_h \subset L_0^2(\Omega)$  be two finite subspaces of  $H_{\Gamma_D}^1(\Omega)^d$  and  $L_0^2(\Omega)$ , respectively. To guarantee the well-posedness of the discrete problem, we assure that these spaces satisfy the discrete uniform inf-sup condition,

$$\|q_h\|_{0,2,\Omega} \leq \sup_{v_h \in X_h} \frac{(q_h, \nabla \cdot v_h)_\Omega}{\|\nabla v_h\|_{0,2,\Omega}}, \quad \forall q_h \in M_h. \quad (14)$$

Now that we have presented the continuous version of the Smagorinsky model, we consider the following Galerkin discretization of problem (12), that is the actual unsteady Smagorinsky turbulence model,

$$\left\{ \begin{array}{l} \forall \gamma \in \mathcal{D}, t \in I_f, \text{ find } (\mathbf{u}_h(t; \gamma), p_h(t; \gamma)) \in X_h \times M_h \text{ such that} \\ (\partial_t \mathbf{u}_h, \mathbf{v}_h)_\Omega + a(\mathbf{u}_h, \mathbf{v}_h; \gamma) + a_S(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h; \gamma) \\ \quad + b(\mathbf{v}_h, p_h; \gamma) + c(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h; \gamma) = \langle \mathbf{f}, \mathbf{v}_h \rangle, \quad \forall \mathbf{v}_h \in X_h, \\ b(\mathbf{u}_h, q_h; \gamma) = 0, \quad \forall q_h \in M_h, \end{array} \right. \quad (15)$$

where we recall that we are considering a triangular mesh  $\{\mathcal{T}_h\}_{h>0}$  of  $\Omega$ , that we assume to be regular in the sense of Ciarlet.

Here,  $a(\cdot, \cdot; \gamma)$  and  $b(\cdot, \cdot; \gamma)$  are bilinear forms defined as

$$a(\mathbf{u}, \mathbf{v}; \gamma) = \frac{1}{\gamma} \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v} d\Omega, \quad b(\mathbf{v}, q; \gamma) = - \int_\Omega (\nabla \cdot \mathbf{v}) q d\Omega.$$

Also,  $c(\cdot, \cdot, \cdot; \gamma)$  is a trilinear skew-symmetric form, given by

$$c(z, \mathbf{u}, \mathbf{v}; \gamma) = \frac{1}{2} \left[ \int_\Omega (z \cdot \nabla \mathbf{u}) \mathbf{v} d\Omega - \int_\Omega (z \cdot \nabla \mathbf{v}) \mathbf{u} d\Omega \right].$$

This form is skew-symmetric in the sense that

$$c(\mathbf{v}, \mathbf{u}, \mathbf{u}; \gamma) = 0, \quad \forall \mathbf{u}, \mathbf{v} \in X. \quad (16)$$

It also holds that

$$c(\mathbf{u}, \mathbf{v}, \mathbf{w}; \gamma) \leq C \|\nabla \mathbf{u}\|_2 \|\nabla \mathbf{v}\|_2 \|\nabla \mathbf{w}\|_2, \quad \forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in X. \quad (17)$$

Finally,  $a_S(\cdot, \cdot, \cdot; \gamma)$  is a non-linear form that models the eddy viscosity effects, stated as

$$a_S(z; \mathbf{u}, \mathbf{v}; \gamma) = \int_\Omega \nu_T(z) \nabla \mathbf{u} : \nabla \mathbf{v} d\Omega.$$

The solution of problem (15) is meant to approximate the large-scales of the solution of the Navier-Stokes problem, that is problem (12) with no eddy viscosity term,  $\nu_T(\mathbf{u}) = 0$ .

The well-posedness of equation (12) was studied in John (2003) following the results presented in Ladyzenskaja (1967). For the discrete case, (equation (15)), in Chacón

Rebollo and Lewandowski (2014) provided by the classical Brezzi theory (Brezzi, 1974), and in Delgado Ávila (2018) for the finite element approximation using the more general Brezzi-Rappaz-Raviart (BRR) theory (Brezzi et al., 1980).

### 2.2.2 Reduced Order Model for the Smagorinsky turbulence model

In this section, we formulate the Reduced Basis (RB) model for the Smagorinsky turbulence model (15). This formulation is a direct adaptation of the RB method for Navier-Stokes equations (Deparis, 2008; Manzoni, 2014).

The idea behind the resolution of the Smagorinsky turbulence model in the Reduced Basis framework is the same as the one used in the resolution by means of the finite element method. That means that, we will be working also with a Galerkin projection of the continuous Smagorinsky model (12), but now the projection is performed onto the reduced basis spaces. The reduced basis spaces are low-dimensional while the finite element spaces could be very high-dimensional.

The reduced basis model for the Smagorinsky turbulence model (15) reads

$$\left\{ \begin{array}{l} \forall \gamma \in \mathcal{D} \text{ and } t \in I_F, \text{ find } (\mathbf{u}_N(t; \gamma), p_N(t; \gamma)) \in X_N \times M_N \text{ such that} \\ (\partial_t \mathbf{u}_N, \mathbf{v}_N)_\Omega + a(\mathbf{u}_N, \mathbf{v}_N; \gamma) + a_S(\mathbf{u}_N; \mathbf{u}_N, \mathbf{v}_N; \gamma) \\ + b(\mathbf{v}_N, p_N; \gamma) + c(\mathbf{u}_N, \mathbf{u}_N, \mathbf{v}_N; \gamma) = \langle \mathbf{f}, \mathbf{v}_N \rangle, \quad \forall \mathbf{v}_N \in X_N, \\ b(\mathbf{u}_N, q_N; \gamma) = 0, \quad \forall q_N \in M_N, \end{array} \right. \quad (18)$$

where  $X_N$  and  $M_N$ , respectively, denote the reduced velocity and pressure spaces. These reduced spaces can be built thanks to the procedures presented in the previous sections, like the POD (Section 1.3.1), the Greedy Algorithm (Section 1.3.2), or most commonly, a joint POD+Greedy approach (Section 1.3.3). We will be focusing on the construction of the reduced spaces by means of the latter approach in Part II.

We assume that the family of pairs of reduced spaces  $\{(X_N, M_N), N = 1, 2, \dots\}$  satisfies the discrete inf-sup condition (14). This may be achieved if we add to the obtained reduced velocity space the so-called supremizers of a basis of the reduced velocity space  $M_N$  (Ballarin et al., 2015; Stabile & Rozza, 2018). The pressure supremizer of  $q_N \in M_N$  is given by

$$T_p^\gamma q_N \in X_h \text{ such that } (\nabla T_p^\gamma q_N, \nabla \mathbf{v}_h)_\Omega = b(q_h, \mathbf{v}_h; \gamma) \quad \forall \mathbf{v}_h \in X_h.$$

Provided that the inf-sup condition is ensured, then, problem (18) is well-posed in the same sense as the finite element problem (15) is (Chacón Rebollo & Lewandowski, 2014). In Part II, we use this enrichment of the reduced velocity space to stabilize the reduced pressure discretization in our numerical experiments.

The eddy viscosity term defined in (13),  $\nu_T(\mathbf{u}(\gamma))$ , is a non-linear function of the parameter. This term will be treated by means of the EIM procedure presented in Section 1.4.1 in the reduced framework.

The Smagorinsky turbulence model will be used in Part II of this dissertation, applied to the development of a universal error indicator based on the statistical turbulence theory.



Part I

MODEL ORDER REDUCTION IN MULTIPLE  
TIMESCALE NETWORK MODELS



---

## INTRODUCTION

---

In this part, our emphasis is on the study and simulation of network models of neuronal activity that involve different timescales, modelled by a system of Ordinary Differential Equations (ODEs). In particular, we focus on the Proper Orthogonal Decomposition (POD) method, with a specific goal of addressing one of the limitations that it presents, namely, the potential loss of structure from the original model. This part aligns with the Spanish national project HYROM. The research presented in this part of the thesis is an extended version of the results published in the research paper Bandera Moreno et al. (2022).

One of the main challenges of network models of neuronal activity is the difficulty to classify and analyze the synchronization features between network cells, and the cost of reliable simulations for networks of realistic size.

Furthermore, these models usually involve different timescales, what is naturally translated into the models by the use of multiple timescales systems as cell dynamics to be coupled as a network. Synchronization mechanisms of multiple timescales systems are quite specific and different from that of harmonic oscillators (especially the impact of the coupling types and strengths on the synchronization). Nevertheless, it is well-known that synchronization of neuron assemblies plays a key role in the neuronal information treatment and diffusion, and overall in the neural behavior.

With the aim of reducing the computational complexity of the simulations of this kind of models, in this part, we will focus on the development of reduced order models for network models of neuronal activity.

Despite the ability of Reduced Order Modelling (ROM) techniques to capture the main information of these systems (see Bandera Moreno et al. (2022) for an approach that joints POD with the Empirical Interpolation Method (EIM) to a model of intracellular calcium concentration), they fail in preserving the separation of timescales of the original model, what we call the *structure* of the model. These could play a key role when dealing with systems with more than two timescales or where we want to keep that separation of timescales also in the reduced model. As a motivation, we provide a simple example of the lost of structure.

In the following, we will consider a dynamical system of  $N \in \mathbb{N}$  equations, that could be separated into two groups (for example, see Equation (2)). However, this analysis can be generalized to  $M < N$  groups. Without loss of generality, we can consider that the variables can be sorted by group.

A dynamical system in this framework could read,

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbb{A}_{11} & \mathbb{A}_{12} \\ \mathbb{A}_{21} & \mathbb{A}_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \quad (19)$$

where  $x_i$  and  $b_i$ ,  $i = 1, 2$ , are vectors of dimensions  $N_1$  and  $N_2$ , respectively, such that  $N = N_1 + N_2$  and  $\mathbb{A}_{ij}$ ,  $i, j = 1, 2$ , are real matrices with consistent dimensions.

For the sake of clarity, we will consider  $\mathbb{A}_{11}$  and  $\mathbb{A}_{22}$  as identity matrices multiplied by nonzero real scalars,  $a$  and  $b$ , respectively,  $\mathbb{A}_{21} = 0$  and  $\mathbb{A}_{12}$  a full matrix, so

$$\begin{pmatrix} \mathbb{A}_{11} & \mathbb{A}_{12} \\ \mathbb{A}_{21} & \mathbb{A}_{22} \end{pmatrix} = \begin{pmatrix} a\mathbb{I} & \mathbb{A}_{12} \\ 0 & b\mathbb{I} \end{pmatrix}.$$

If we consider system (19), and we let  $\mathbb{U}$  be the matrix arising from the POD of some snapshots of the system (see Section 1.3.1 for more information), then the reduced order system reads,

$$\dot{\mathbf{z}} = \tilde{\mathbf{A}}\mathbf{z} + \mathbf{b}_r, \quad (20)$$

where  $\mathbf{b}_r = \mathbb{U}^T \mathbf{b} = \begin{pmatrix} \mathbb{U}_{11}^T & \mathbb{U}_{21}^T \\ \mathbb{U}_{12}^T & \mathbb{U}_{22}^T \end{pmatrix} \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}$ , and

$$\begin{aligned} \tilde{\mathbf{A}} &= \mathbb{U}^T \mathbf{A} \mathbb{U} = \begin{pmatrix} \mathbb{U}_{11}^T & \mathbb{U}_{21}^T \\ \mathbb{U}_{12}^T & \mathbb{U}_{22}^T \end{pmatrix} \begin{pmatrix} a\mathbb{I} & \mathbb{A}_{12} \\ 0 & b\mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{U}_{11} & \mathbb{U}_{12} \\ \mathbb{U}_{21} & \mathbb{U}_{22} \end{pmatrix} \\ &= \begin{pmatrix} a\mathbb{U}_{11}^T \mathbb{U}_{11} + \mathbb{U}_{11}^T \mathbb{A}_{12} \mathbb{U}_{21} + b\mathbb{U}_{21}^T \mathbb{U}_{21} & a\mathbb{U}_{11}^T \mathbb{U}_{12} + \mathbb{U}_{11}^T \mathbb{A}_{12} \mathbb{U}_{22} + b\mathbb{U}_{21}^T \mathbb{U}_{22} \\ a\mathbb{U}_{12}^T \mathbb{U}_{11} + \mathbb{U}_{12}^T \mathbb{A}_{12} \mathbb{U}_{21} + b\mathbb{U}_{22}^T \mathbb{U}_{21} & a\mathbb{U}_{12}^T \mathbb{U}_{12} + \mathbb{U}_{12}^T \mathbb{A}_{12} \mathbb{U}_{22} + b\mathbb{U}_{22}^T \mathbb{U}_{22} \end{pmatrix}, \end{aligned}$$

and we have lost the structural information of the original problem and the possible separation between variables, for example, when  $a = 1$  and  $b = \varepsilon \ll 1$ .

In order to avoid this lost of structure, we will develop a novel technique that will take into account the structure of the original problem when constructing the reduced order model. That will allow us to preserve the separation of timescales. We will also,

provide a data-driven technique in order to determine the structure of the original model, which can be difficult to determine directly.

This part is structured as follows: in Chapter 3, we will present a novel technique based on the POD technique that will preserve the structure of the original problem, and we also provide a data-driven technique in order to determine the best structure for the reduced model in order to maintain as most structure of the original model while still providing a low number of reduced variables. Then, in Chapter 4, we will compare these novel techniques with the original POD for three neural network models with multiple timescales.



# 3

---

## STRUCTURE-PRESERVING METHODS BASED ON THE PROPER ORTHOGONAL DECOMPOSITION

---

In the introduction of this part, we have presented the problematic of the lost of structure when trying to obtain a reduced order model of a dynamical system directly by means of the original POD technique that we presented in Section 1.3.1. To overcome this issue, in this chapter, we develop an extension to the original POD technique, that is able to preserve the structure of the original system. We will focus on the framework of a system of parametric ODEs, although this technique can be also applied in the field of PDEs that exhibit any qualitative separation in the variables.

The structure of the chapter is as follows: in Section 3.1, we present the extension of the original POD to systems that exhibit a qualitative separation of the variables, and we also give some mathematical results concerning this novel technique in Section 3.1.1. Then, in Section 3.2, we present a previous data-driven step that can be performed in order to obtain the best structure for the reduced model in order to maintain as most structure of the original model while still providing a low number of reduced variables.

### 3.1 PROPER ORTHOGONAL BLOCK DECOMPOSITION

If we take into account the construction of the reduced model (20), we can see that, in order to maintain the main properties of the original model when we perform the reduction, we have to take care of the structure of the matrix  $\mathbb{U}$  arising from the POD technique.

Taking into account the separation of variables in the original model (19) and the action of matrix  $\mathbb{U}$  in the construction of the reduced model (20), we can see that if matrix  $\mathbb{U}$  were diagonal by blocks, we could keep the structure of the original model.

With that in mind, we divide the snapshot matrix  $\mathbb{S} = (\mathbf{w}_1 | \dots | \mathbf{w}_{N_s})$  into two blocks, as follows,  $\mathbb{S}^T = (\mathbb{S}_1^T | \mathbb{S}_2^T)$ . Now, we perform a Singular Value Decomposition (SVD) for each block, and we obtain a POD basis for each block,

$$\mathbb{S}_1 \approx \sum_{i=1}^{r_1} \sigma_i^{(1)} \mathbf{u}_i^{(1)} (\mathbf{v}_i^{(1)})^T, \quad \mathbb{S}_2 \approx \sum_{i=1}^{r_2} \sigma_i^{(2)} \mathbf{u}_i^{(2)} (\mathbf{v}_i^{(2)})^T,$$

where  $\sigma_i^{(\cdot)}$  stands for the  $i$ -th singular value,  $\mathbf{u}_i^{(\cdot)}$  is the  $i$ -th left singular vector, and  $\mathbf{v}_i^{(\cdot)}$  is the  $i$ -th right singular vector of the corresponding matrix  $\mathbb{S}_i$ . Proposition 1.2 holds for each block, so as in the original POD method, we have a suitable criterion for the selection of the reduced basis dimension for each variable. We name the reduced dimensions  $N_{r1}$  and  $N_{r2}$  for each block, respectively.

Now, we focus on the reconstruction of the original snapshot matrix as a whole. We note that each block matrix has the correct number of columns, so no change in the block right singular vectors  $\mathbf{v}_i^{(\cdot)}$  is needed. However, each block left singular vector  $\mathbf{u}_i^{(1)}$  lays in  $\mathbb{R}^{N_1}$ , so we need to complete them in order to have a vector in  $\mathbb{R}^N$  as we must, and respectively for vectors  $\mathbf{u}_i^{(2)}$ .

A direct concatenation of the block left singular vectors does not work, as we do not know how the product  $\mathbf{u}_i^{(k)} (\mathbf{v}_i^{(l)})^T$  behaves, and we have no control over it. A successful approach is to add as many zeros as needed to each block left singular vector, taking into account the position of the block matrix in the original matrix, i.e.,

$$\mathbf{u}_i^{(1)} \mapsto \begin{pmatrix} \mathbf{u}_i^{(1)} \\ \mathbf{0}_{N_2} \end{pmatrix}, \quad \mathbf{u}_i^{(2)} \mapsto \begin{pmatrix} \mathbf{0}_{N_1} \\ \mathbf{u}_i^{(2)} \end{pmatrix}.$$

With this decomposition, we can recover the original matrix as

$$\begin{pmatrix} \mathbb{U}_1 & \mathbb{O}_{N_1, N_{r2}} \\ \mathbb{O}_{N_2, N_{r1}} & \mathbb{U}_2 \end{pmatrix} \begin{pmatrix} \Sigma_1 & \mathbb{O}_{N_1, N_{r2}} \\ \mathbb{O}_{N_2, N_{r1}} & \Sigma_2 \end{pmatrix} \begin{pmatrix} \mathbb{V}_1^T \\ \mathbb{V}_2^T \end{pmatrix} = \begin{pmatrix} \mathbb{U}_1 \Sigma_1 \mathbb{V}_1^T \\ \mathbb{U}_2 \Sigma_2 \mathbb{V}_2^T \end{pmatrix} \approx \begin{pmatrix} \mathbb{S}_1 \\ \mathbb{S}_2 \end{pmatrix}.$$

As we have exploited the separation of the variables by blocks, we name this method the *Proper Orthogonal Block Decomposition* (POBD).

Here, we have considered just two blocks for simplicity, but the method can be extended to any number of blocks  $N_b$ .

### 3.1.1 Properties of the Proper Orthogonal Block Decomposition

In this section, we present some properties of the POBD. On the one hand, we will check that this novel technique is able to preserve the structure of the original problem. On the other hand, we will present a result concerning the error bound for the neglected information.

We focus first on the preservation of the structure of the original problem, if we select the matrix arising from the POBD

$$\mathbb{U} = \begin{pmatrix} \mathbb{U}_1 & \mathbb{O}_{N_1, N_{r_2}} \\ \mathbb{O}_{N_2, N_{r_1}} & \mathbb{U}_2 \end{pmatrix}, \quad (21)$$

as the new basis for the reduced subspace, the reduced model associated to the original system (19) can be obtained as follows:

$$\begin{aligned} \tilde{\mathbb{A}} &= \mathbb{U}^T \mathbb{A} \mathbb{U} = \begin{pmatrix} \mathbb{U}_1^T & \mathbb{O}_{N_2, N_{r_1}}^T \\ \mathbb{O}_{N_1, N_{r_2}}^T & \mathbb{U}_2^T \end{pmatrix} \begin{pmatrix} \mathbb{I}_{N_1} & \mathbb{A}_{12} \\ \mathbb{O}_{N_2, N_1} & \mathbb{I}_{N_2} \end{pmatrix} \begin{pmatrix} \mathbb{U}_1 & \mathbb{O}_{N_1, N_{r_2}} \\ \mathbb{O}_{N_2, N_{r_1}} & \mathbb{U}_2 \end{pmatrix} \\ &= \begin{pmatrix} \mathbb{I}_{N_{r_1}} & \mathbb{U}_1^T \mathbb{A}_{12} \mathbb{U}_2 \\ \mathbb{O}_{N_{r_2}, N_{r_1}} & \mathbb{I}_{N_{r_2}} \end{pmatrix} = \begin{pmatrix} \mathbb{I}_{N_{r_1}} & \tilde{\mathbb{A}}_{12} \\ \mathbb{O}_{N_{r_2}, N_{r_1}} & \mathbb{I}_{N_{r_2}} \end{pmatrix}. \end{aligned}$$

We remark that applying the POBD technique allows us to maintain the structure of the original system, that was the aim of this technique.

Now, we focus on obtaining an error bound on the neglected information when using the POBD for the construction of the reduced model. As done for the original POD technique, where we were able to obtain an error bound depending on the values of the neglected singular values, thanks to the properties of the SVD of a matrix, see Proposition 1.2, in our case, we are able to obtain a similar error bound applying similar techniques.

#### Proposition 3.1

Let  $\mathbb{U}$  be an orthonormal matrix built as in (21) for a dynamical system with  $N_b \geq 2$  blocks. Then, the next error bound holds

$$\sum_{i=1}^{N_S} \|\mathbf{w}_i - \mathbb{U}^T \mathbb{U} \mathbf{w}_i\|_2^2 = \sum_{j=1}^{N_b} \sum_{i=N_{r_j}+1}^{r_j} (\sigma_i^{(j)})^2.$$

**Proof:** This proof is an adaptation of the proof of Proposition 1.2 in (Quarteroni et al., 2016) to our case. We detail it for  $N_b = 2$  for the sake of clarity.

For a matrix,  $\mathbb{A} \in \mathbb{R}^{N \times N_s}$  we have  $\|\mathbb{A}\|_F^2 = \sum_{i=1}^{N_s} \|\mathbf{a}_i\|_2^2$ , where  $\mathbf{a}_i$  denotes the  $i$ -th column of  $\mathbb{A}$ . Thus, in our case we can write

$$\sum_{i=1}^{N_s} \|\mathbf{w}_i - \mathbb{U}\mathbb{U}^T\mathbf{w}_i\|_2^2 = \|\mathbb{S} - \mathbb{U}\mathbb{U}^T\mathbb{S}\|_F^2. \quad (22)$$

Now, as

$$\mathbb{S} = \begin{pmatrix} \mathbb{S}_1 \\ \mathbb{S}_2 \end{pmatrix} = \begin{pmatrix} \mathbb{S}_1 \\ \mathbb{O}_{N_2, N_s} \end{pmatrix} + \begin{pmatrix} \mathbb{O}_{N_1, N_s} \\ \mathbb{S}_2 \end{pmatrix},$$

and

$$\mathbb{U}\mathbb{U}^T\mathbb{S} = \begin{pmatrix} \mathbb{U}_1\mathbb{U}_1^T\mathbb{S}_1 \\ \mathbb{U}_2\mathbb{U}_2^T\mathbb{S}_2 \end{pmatrix} = \begin{pmatrix} \mathbb{U}_1\mathbb{U}_1^T\mathbb{S}_1 \\ \mathbb{O}_{N_2, N_s} \end{pmatrix} + \begin{pmatrix} \mathbb{O}_{N_1, N_s} \\ \mathbb{U}_2\mathbb{U}_2^T\mathbb{S}_2 \end{pmatrix},$$

along with the facts that  $\|\mathbb{S}\|_F = \|\mathbb{S}^T\|_F$  and  $\|(A|B)\|_F^2 = \|A\|_F^2 + \|B\|_F^2$ , we can decompose the last term of (22) as

$$\|\mathbb{S} - \mathbb{U}\mathbb{U}^T\mathbb{S}\|_F^2 = \|\mathbb{S}_1 - \mathbb{U}_1\mathbb{U}_1^T\mathbb{S}_1\|_F^2 + \|\mathbb{S}_2 - \mathbb{U}_2\mathbb{U}_2^T\mathbb{S}_2\|_F^2.$$

Finally, applying Proposition 1.2 to each term on the right-hand side, we obtain the desired result. ■

A similar approach for the decomposition of the domain in computational simulation of fluids was introduced in Anderson et al. (2022).

Property 3.1 provides a similar error bound to the one obtained in Property 1.2, but, in the former, the action of the blocks is taken into account. We note that the more blocks that we consider the greater the error would be, this will be tackled in the next section, Section 3.2.

Now, we give two remarks concerning the properties of the POBD. The first one acknowledging that the original POD technique is a special case of the POBD, and another one about the non-optimality of the error bound for two or more blocks.

### Remark 3.1

The POBD procedure is suitable for any number of blocks  $N_b < N$  and when considering just 1 block, we recover the classic POD method.

**Remark 3.2**

We remark that despite the similarities between the error bounds of Propositions 1.2 and 3.1, for a number of blocks  $N_b \geq 2$  the latter might not be the optimal for reducing onto a subspace of dimension  $\sum_{j=1}^{N_b} N_{r_j}$ .

In order to measure the good behavior of this method, taking into account just the number of equations of the reduced system could be not enough, as we could also need to retain some information about the structure of the original model. Bearing that in mind, we propose a new index, that we name the *sparsity index*, defined as the quotient between the number of zero entries and the number of total elements of the matrix of a system of differential equations. This index provides an indicator of the structure of the models.

		Original model
Non-zero elements		$\approx N_1 + N_2 + N_1 N_2$
Total elements		$(N_1 + N_2)^2$
Sparsity index		$\approx \frac{N_1(N_1 - 1) + N_2(N_2 - 1) + N_1 N_2}{(N_1 + N_2)^2}$
		POBD
Non-zero elements	$\approx N_r^2$	$\approx N_{r1} + N_{r2} + N_{r1} N_{r2}$
Total elements	$N_r^2$	$(N_{r1} + N_{r2})^2$
Sparsity index	$\approx 0$	$\approx \frac{N_{r1}(N_{r1} - 1) + N_{r2}(N_{r2} - 1) + N_{r1} N_{r2}}{(N_{r1} + N_{r2})^2}$

Table 3.1: Comparison of the sparsity index for between the original model, and the models arising from the POD and POBD techniques.

Table 3.1 shows the comparison between the sparsity index of the original model and the ones arising from the POD and POBD techniques for the academic test case (19). We can easily check that, for this test case, the sparsity index of the POBD systems is much more in concordance with the one of the original model, while the sparsity index of the POD system is always close to zero.

## 3.2 AUTOMATIC PROPER ORTHOGONAL BLOCK DECOMPOSITION

From Proposition 3.1, we can clearly see that the error bound scales with the number of blocks considered. That gives us an idea that the lesser the number of blocks that we consider, the lesser the error between the solution of the reduced model and the original solution will be. However, we still need to take into account the structure of

the original model. We are facing a dichotomy, should we focus on obtaining better error bounds while losing the structure of the original model, or just the opposite, give greater importance to the preservation of the structure of the problem?

In order to solve this issue, we propose here a data-driven method that will allow us to obtain the best number of blocks and their composition so that we obtain the optimal trade off between low error bounds and preservation of the structure of the original problem. We would name this method *Automatic Proper Orthogonal Block Decomposition* (APOBD). The procedure of the selection of blocks is summarized in the following. We have illustrated the steps with the academic example of the previous subsections (19),

1. Perform a previous study to obtain an *a priori* number of blocks of variables and their *a priori* configuration.
2. Once you have obtained the snapshots of the original models, the first step is to compute the correlation matrix of the snapshots  $\mathbb{C} \in \mathbb{R}^{N \times N}$ .

$$\mathbb{C} = \mathbb{S}\mathbb{S}^T = \begin{pmatrix} \mathbb{S}_1 \\ \mathbb{S}_2 \end{pmatrix} \left( \begin{array}{c|c} \mathbb{S}_1^T & \mathbb{S}_2^T \end{array} \right) = \begin{pmatrix} \mathbb{S}_1\mathbb{S}_1^T & \mathbb{S}_1\mathbb{S}_2^T \\ \mathbb{S}_2\mathbb{S}_1^T & \mathbb{S}_2\mathbb{S}_2^T \end{pmatrix} = \begin{pmatrix} \mathbb{C}_{11} & \mathbb{C}_{12} \\ \mathbb{C}_{12}^T & \mathbb{C}_{22} \end{pmatrix}.$$

3. Then, we compute the mean correlation of each block with itself and the others, so we end up with a symmetric matrix of dimension equal to the number of blocks. In order to check if the *a priori* blocks were well located, we can check that the elements of the diagonal of this matrix are close to 1.

$$\mathbb{L} = \begin{pmatrix} \frac{\sum_{i,j=1}^{N_1} (\mathbb{C}_{11})_{ij}}{N_1^2} & \frac{\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (\mathbb{C}_{12})_{ij}}{N_1 N_2} \\ \frac{\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (\mathbb{C}_{12})_{ij}}{N_1 N_2} & \frac{\sum_{i,j=1}^{N_2} (\mathbb{C}_{22})_{ij}}{N_2^2} \end{pmatrix} = \begin{pmatrix} l_{11} & l_{12} \\ l_{12} & l_{22} \end{pmatrix}, \quad \text{with } l_{ii} \approx 1.$$

4. As the third step, for each pair of different blocks  $(i, j)$ , we compute the collinearity coefficient  $cl_{ij}$ , that is, the determinant of the  $2 \times 2$  submatrix of the matrix  $\mathbb{L}$  of the previous step, corresponding to the blocks studied,

$$cl_{ij} = \det \begin{pmatrix} l_{ii} & l_{ij} \\ l_{ij} & l_{jj} \end{pmatrix}.$$

If this value is lower than a certain fixed tolerance, we can consider the two blocks of variables as a unique block.

This procedure is based on the data of the original model, as it depends only on the snapshots and the *a priori* given blocks of variables, and allow us to group the blocks of variables that behave qualitatively similar, avoiding the repetition of redundant information.

We end this section stating two remarks corresponding to the APOBD, one corresponding to a possible computational saving, and another one limiting the number of collinearity coefficients that we have to compute.

**Remark 3.3**

*Step two of the APOBD procedure, could also be seen as the first step for the POD technique, as the singular values of a matrix are the square root of the eigenvalues of the respective correlation matrix.*

**Remark 3.4**

*At most, we would need to compute  $N_b(N_b - 1)/2$  collinearity coefficients as the diagonal elements are zero,  $cl_{ii} = 0$ , and they are symmetric,  $cl_{ij} = cl_{ji}$ .*



# 4

---

## APPLICATION TO MULTIPLE TIMESCALES NETWORK MODELS

---

This chapter is devoted to the validation and numerical study of the methods developed in Chapter 3, POBD and APOBD. We will apply these reduction techniques to problems arising from the mathematical modelling of biological phenomena exhibited in cells or neurons:

- In Section 4.1, we will study a model of Intracellular Calcium Concentration (ICC) (Bandera Moreno et al., 2022; Fernández-García & Vidal, 2020; Krupa et al., 2013). See Section 2.1 for more details about this model.
- In Section 4.2, we will study two models presenting bursting dynamics, that is, alternance between quiescence phases and rapid oscillation phases. In particular, we consider two models:
  - In subsection 4.2.1, we will study the celebrated Hindmarsh-Rose model (Barrio & Shilnikov, 2011; Bonaventura et al., 2022; Etémé et al., 2017; Hindmarsh & Rose, 1984).
  - In Subsection 4.2.2, we will study pancreatic  $\beta$ -cells models described by a Fold/Hopf Bursting model (Izhikevich, 2000; Pernarowski et al., 1992; Smolen et al., 1993).

We will work with network models with a high number of coupled oscillators, and we will compare the reduction techniques by means of some classical indicators, such as speed-up and number of reduced equations, and by means of the sparsity index, an indicator of the reproduction of the structure of the original problems based on the sparsity of the reduced matrices associated to each reduced model.

#### 4.1 INTRACELLULAR CALCIUM CONCENTRATION NETWORK MODEL

In this section, our main aim is to test the aforementioned methods on the ICC model studied in Bandera Moreno et al. (2022) and Fernández-García and Vidal (2020) and presented in Section 2.1.

##### 4.1.1 Network model of neuronal Intracellular Calcium Concentration dynamics

Although some advances were made in obtaining the reduced order model for the two-cell model (11) in Bandera Moreno et al. (2022), in more realistic problems, we have more than two cells. We consider here the case where we have  $N$  different cells and symmetric coupling, so we can model the system with  $N$  coupled oscillators:

$$O_i \left\{ \begin{array}{l} \dot{x}_i = \tau(-y_i + f(x_i) - \phi_f(z_i)), \\ \dot{y}_i = \tau\varepsilon k_i(x_i + a_1 y_i + a_2 + \frac{1}{N/2} \sum_{j=1}^N c_{ij}(x_i - x_j)), \\ \dot{z}_i = \tau\varepsilon \left( \phi_r(x_i) - \frac{z_i - z_b}{\tau_z} \right), \end{array} \right. \quad (23)$$

where  $i = 1, 2, \dots, N$  and  $c_{ij} \in [-1, 1]$  with  $c_{ij} = c_{ji}$ . For  $N = 2$ , we recover system (11).

In the following, we will consider that cells have symmetric coupling, and we keep the same parametric values presented in Section 2.1. Furthermore, we set  $N_- = \lfloor N/2 \rfloor$ , that is, the greatest integer lower than or equal to  $N/2$ , and  $N_+ = \lceil N/2 \rceil$ , that is, the lowest integer greater than or equal to  $N/2$ .

In a first approach, we consider the cells separated following a modular network structure with two clusters, namely, first cluster, are the cells with indexes  $i = 1, \dots, N_-$ , and second cluster, the cells with indexes  $i = N_- + 1, \dots, N$ . This choice is based on the motoneurons network properties studied in the zebrafish embryo in De Vico Fallani et al. (2015), and has been considered previously in Bandera Moreno et al. (2022) for the ICC model. In the next subsection, we give a more detailed explanation on the ICC model and some of its properties.

We set that the cells of the same cluster are coupled with  $c_{ij} = c_\alpha$ , and that every two cells of different clusters are coupled with the same parameter  $c_\beta$ , that is  $c_{ij} = c_\beta$ ,  $\forall i = 1, \dots, N_-, \forall j = N_- + 1, \dots, N$ . We consider two different cases:

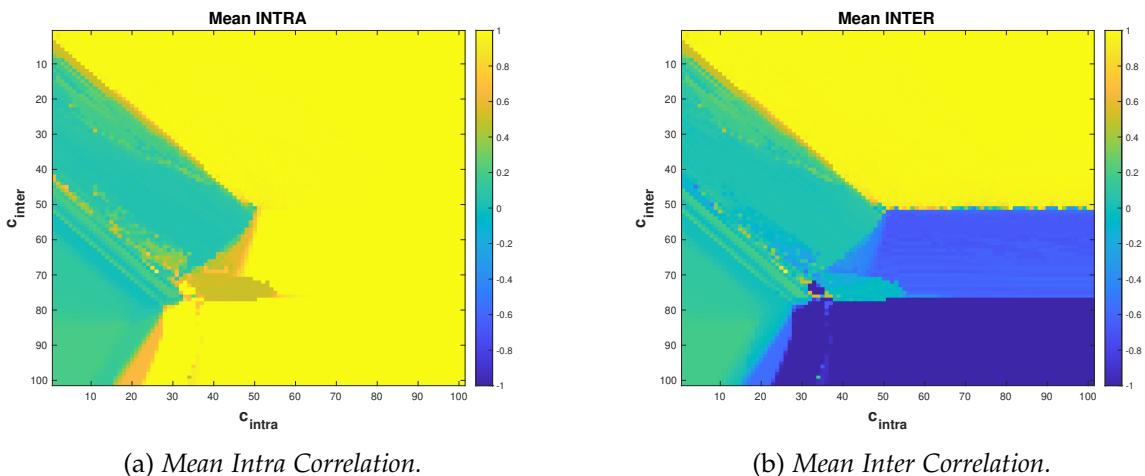
- *Homogeneous case* ( $k_i = 1$ ). All the cells have the same intrinsic amplitude and frequency.

- *Heterogeneous case* ( $k_i = 1$  for  $i = 1, \dots, N_-$  and  $k_i > 1$  for  $i = N_- + 1, \dots, N$ ). All the cells in one cluster have the same intrinsic amplitude and frequency, but the cells of the other cluster have different intrinsic amplitudes and frequencies.

Before performing any ROM technique, we need to know how the behaviors are distributed in the two cluster model (23). Therefore, in the following, we study the different behaviors that the system can exhibit by moving both  $c_\alpha$  and  $c_\beta$  in the interval  $[-1, 1]$ . In order to do so, we let the model run for a certain time, high enough to avoid the possible transient period. Then, we compute the mean correlation between cells inside each cluster, that we name *Mean Intra Correlation*, and the mean correlation between cells of different clusters, that we name *Mean Inter Correlation*.

#### *Homogeneous case*

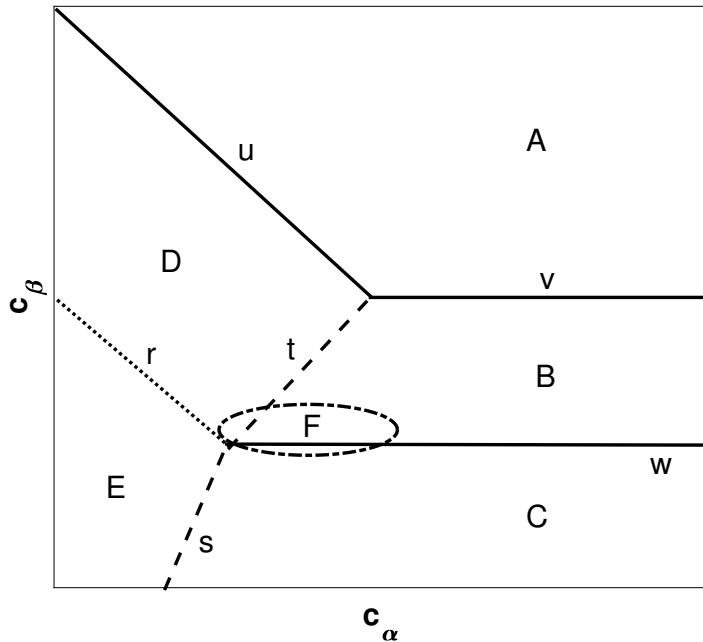
In this case, we set the computational solving time from  $T = 0$  to  $T = 100$ . The partition in  $(c_\alpha, c_\beta)$  is a uniform grid of  $101 \times 101$  elements from in  $[-1, 1]^2$ . We set the number of cells to  $N = 100$ , the heterogeneity parameters  $k_i = 1$ , so we have copies of the same cell and the initial condition far away from the attractors. We compute the correlation of the solutions for each pair of cells in the time interval  $[70, 100]$ . The results for the *Mean Intra Correlation* and *Mean Inter Correlation* are shown in Figure 4.1.



**Figure 4.1:** Mean Correlations for system (23) for different values of  $c_\alpha$  and  $c_\beta$  inside the plane  $[-1, 1]^2$ . Left panel shows the Mean Intra Correlation, and right panel shows the Mean Inter Correlation.

With a closer study, we can identify 6 different behaviors, we show the parameter plane diagram in Figure 4.2. There are three main behaviors, named A, B and C, and

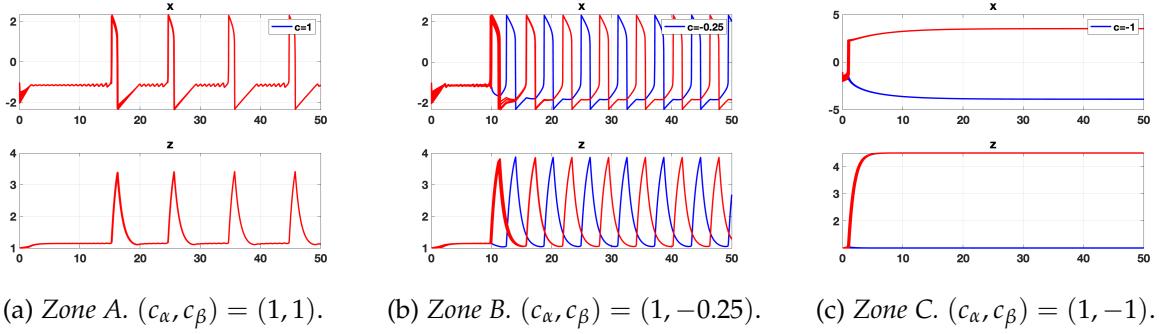
three residual behaviors, named D, E and F. We can also identify several boundaries between the behaviors, that we named  $r, s, t, u, v, w$  and these changes in behavior can exhibit a transient or not.



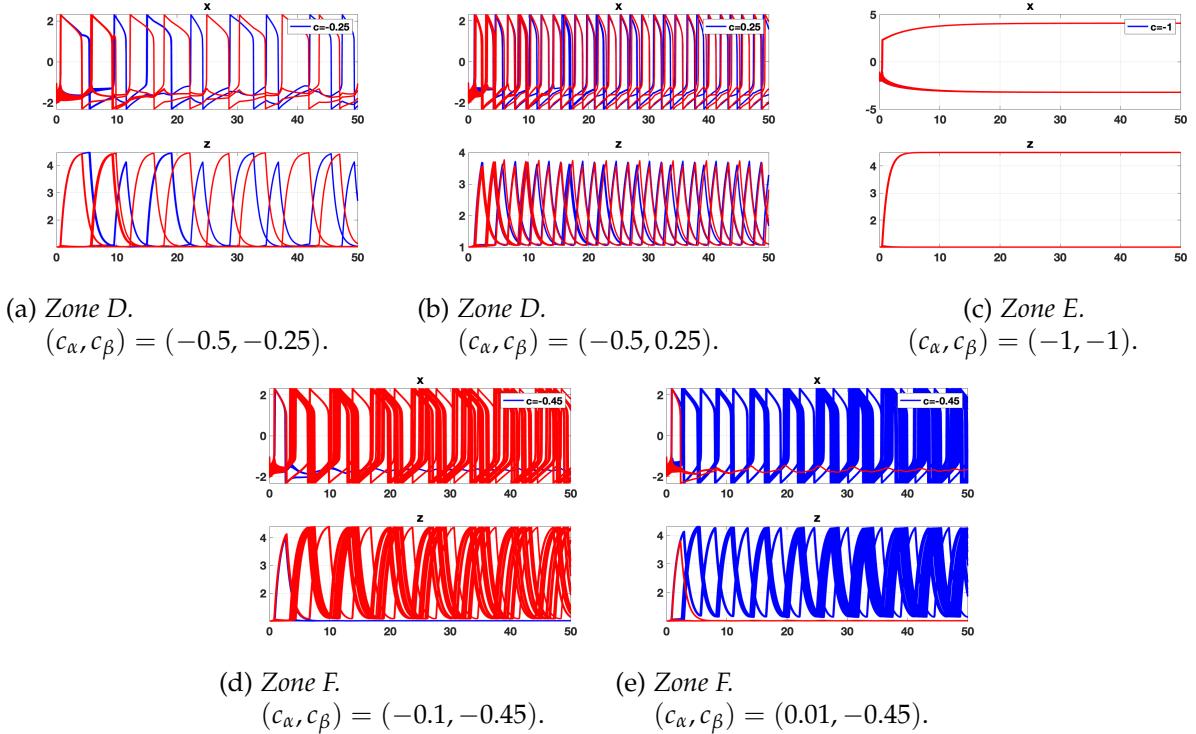
**Figure 4.2:** Diagram for the different behaviors of model (23). Zones are represented with capital letters, while the boundary between them are represented with lower case letters. Continuous lines ( $u, v, w$ ) indicate the non-existence of a transient region, while lines with other fashions ( $r, s, t$ ) indicate the existence of a transient region.

Here, we summarize the obtained behaviors of each zone:

- Every cell follows the only one cell behavior. Shown in Figure 4.3a.
- Each cluster follows the two cells antiphase behavior. Shown in Figure 4.3b.
- Each cluster follows the two cells total death behavior. Shown in Figure 4.3c.
- Each cell spikes independent of the other ones. Shown in figures 4.4a and 4.4b.
- This behavior is similar to the total death behavior, but the separation between clusters is no longer clear. Shown in Figure 4.4c.
- The cells inside a cluster spike independent of the other ones, and the cells inside the other cluster remain totally silent. Shown in figures 4.4d and 4.4e.



**Figure 4.3:** Representation of the three of the main behaviors A-C obtained for model (23). These behaviors represent the behaviors obtained if the intracluster coupling is set to be between 0 and 1, i.e., in-phase synchronization.

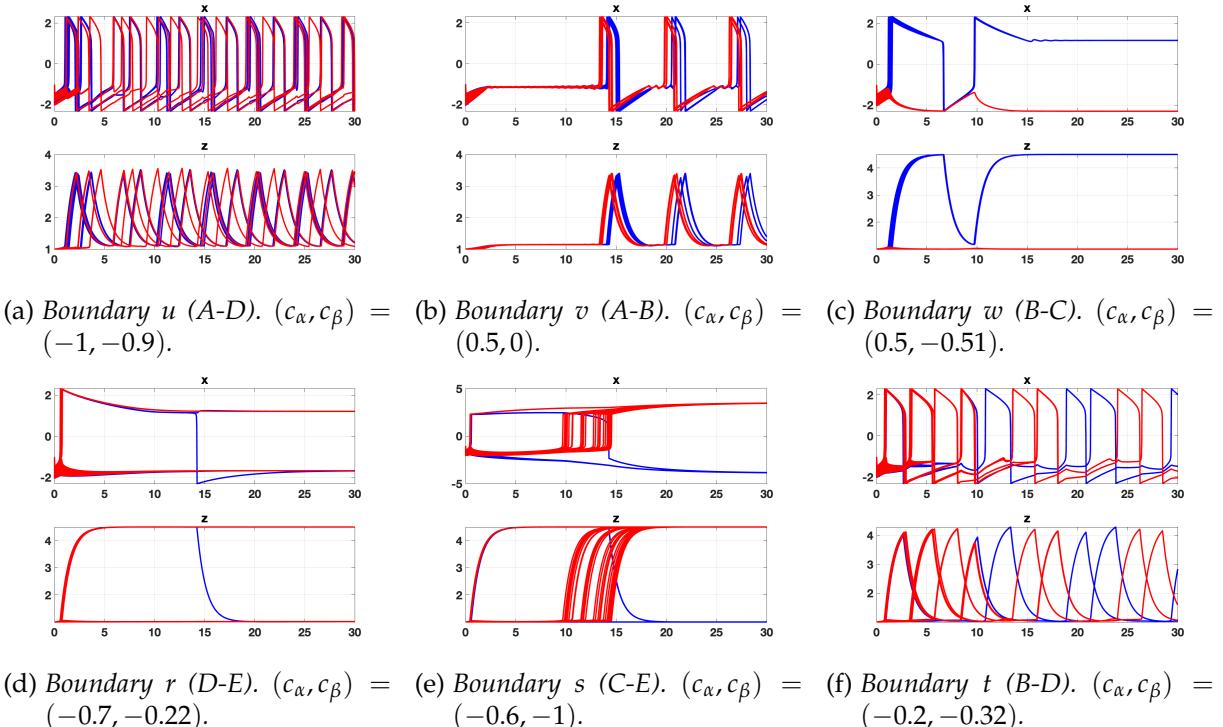


**Figure 4.4:** Representation of the residual behaviors E-F obtained for model (23). These behaviors represent the behaviors obtained if the intracluster coupling is set to be lower than 0, i.e., inhibitory synchronization.

Now, we present the existing behaviors in the transition between the zones introduced in the previous:

- u. Cells start to desynchronize, but there are some that remain synchronized. Shown in Figure 4.5a.
- v. Two clusters. Shown in Figure 4.5b.

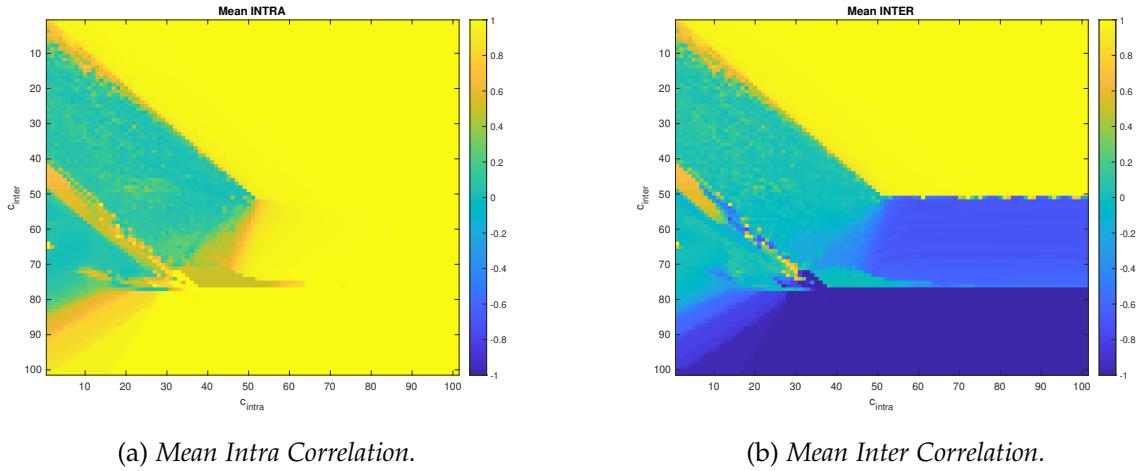
- w. Relaxation oscillation to total death of the clusters. Shown in Figure 4.5c.
- r. The cells of a cluster suddenly go from excitation to silence. Shown in Figure 4.5d.
- s. The cells of a cluster suddenly go from silence to excitation. Shown in Figure 4.5e.
- t. Cells within each cluster start to desynchronize, the antiphase behavior between cluster starts to vanish. Shown in Figure 4.5f.



**Figure 4.5:** Representation of the evolution of the solution for parameter values in the boundaries of the behaviors obtained for model (23).

### Heterogeneous case

We perform the same study as in the previous section, but we set the heterogeneous parameter randomly in  $[1, 1.5]$  following a truncated normal distribution  $\mathcal{N}(1.25, 0.25)$ . The results for the *Mean Intra Correlation* and *Mean Inter Correlation* are showed in Figure 4.6.



**Figure 4.6:** Mean Correlations for system (23) for different values of  $c_\alpha$  and  $c_\beta$  inside the interval  $[-1, 1]$ . Left panel shows the Mean Intra Correlation, and right panel shows the Mean Inter Correlation.

We can see that there are no major changes in the regions of the zones, but that zone C has expanded to the left, reducing the area of behavior E and augmenting the transient  $r$ . Zone F also transfers some area to behavior B.

#### 4.1.2 Preliminary study

In this section, we will build the algebraic version of the original network problem (23), then, we will present the behaviors that we want to study, and finally, we will determine the *a priori* number of blocks and their configuration.

##### Algebraic version

In order to build the algebraic form of the original network problem (23), we denote  $\mathbf{w} = (x_1, \dots, x_N, y_1, \dots, y_N, z_1, \dots, z_N)$  a rearrangement of the vector of unknowns. We note that we have changed the order of the variables in order to help the reader to understand the relationship between them, and that it has no consequences in the following study. We can write problem (23) as

$$\dot{\mathbf{w}} = f(\mathbf{w}, \gamma), \text{ with } \mathbf{w}(0) = \mathbf{w}_0, \quad (24)$$

where  $\gamma = (c_\alpha, c_\beta, \mathbf{k})$ , the parameter vector, with  $\mathbf{k} = (k_1, \dots, k_N)$  the vector of heterogeneity parameters, and  $c_\alpha, c_\beta$  the intra-cluster and inter-cluster coupling, respectively.

For the sake of clarity, we define some quantities subsequently:

**Notation 4.1**

$\mathbb{K} = \text{diag}(\mathbf{k})$  is a diagonal matrix whose entries are the heterogeneity parameters.

**Notation 4.2**

We define the following parameters,

$$\Sigma_i = \begin{cases} (N_- - 1)c_\alpha + N_+c_\beta, & \text{for } i = 1, \dots, N_-, \\ (N_+ - 1)c_\alpha + N_-c_\beta, & \text{for } i = N_- + 1, \dots, N. \end{cases}$$

Note that, when  $N$  is even,  $\Sigma_i = \Sigma = (N/2 - 1)c_\alpha + Nc_\beta/2, \forall i = 1, \dots, N$ .

Finally, we can consider the coupling matrix, where top left block is  $N_- \times N_-$ , top right is  $N_- \times N_+$ , bottom left is  $N_+ \times N_-$ , and bottom right is  $N_+ \times N_+$ .

$$\mathbb{M}(c_\alpha, c_\beta) = \left( \begin{array}{cccc|cccc} \Sigma_1 & -c_\alpha & \dots & -c_\alpha & -c_\beta & \dots & \dots & -c_\beta \\ -c_\alpha & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ \vdots & \ddots & \ddots & -c_\alpha & \vdots & & & \vdots \\ -c_\alpha & \dots & -c_\alpha & \Sigma_{N_-} & -c_\beta & \dots & \dots & -c_\beta \\ \hline -c_\beta & \dots & \dots & -c_\beta & \Sigma_{N_-+1} & -c_\alpha & \dots & -c_\alpha \\ \vdots & & & \vdots & -c_\alpha & \ddots & \ddots & \vdots \\ \vdots & & & \vdots & \vdots & \ddots & \ddots & -c_\alpha \\ -c_\beta & \dots & \dots & -c_\beta & -c_\alpha & \dots & -c_\alpha & \Sigma_N \end{array} \right)$$

We remark here that the matrix  $\mathbb{M}(c_\alpha, c_\beta)$  admits an affine decomposition depending on  $c_\alpha, c_\beta$  and  $\Sigma_i$ .

Taking this into account, we can write the algebraic version of system (24) as

$$\dot{\mathbf{w}} = \tau((\mathbb{A}_{np} + \mathbb{A}_p(\mathbf{k}) + \mathbb{B}(c_\alpha, c_\beta, \mathbf{k}))\mathbf{w} + \mathbf{b}_{np} + \mathbf{b}_p(\mathbf{k}) + \mathbf{b}_{nl}(\mathbf{w})), \quad (25)$$

where  $\mathbb{A}_{np}$  and  $\mathbf{b}_{np}$  represent the non-parametric part,  $\mathbb{A}_p(\mathbf{k})$  and  $\mathbf{b}_p(\mathbf{k})$  the parametric part depending on the heterogeneity parameters,  $\mathbb{B}(c_\alpha, c_\beta)$  is the parametric part relative to the coupling and  $\mathbf{b}_{nl}(\mathbf{w})$  denotes the nonlinear part of system (23). They can be expressed as follows

$$\mathbb{A}_{np} = \left( \begin{array}{c|c|c} 4\mathbb{I}_N & -\mathbb{I}_N & \mathbb{O}_N \\ \hline \mathbb{O}_N & \mathbb{O}_N & \mathbb{O}_N \\ \hline \mathbb{O}_N & \mathbb{O}_N & -\varepsilon/\tau_z \mathbb{I}_N \end{array} \right), \quad \mathbb{A}_p(\mathbf{k}) = \varepsilon \left( \begin{array}{c|c|c} \mathbb{O}_N & \mathbb{O}_N & \mathbb{O}_N \\ \hline a_0 \mathbb{K} & a_1 \mathbb{K} & \mathbb{O}_N \\ \hline \mathbb{O}_N & \mathbb{O}_N & \mathbb{O}_N \end{array} \right),$$

$$\mathbb{B}(c_\alpha, c_\beta) = \frac{\varepsilon}{N/2} \left( \begin{array}{c|c|c} \mathbf{0}_N & \mathbf{0}_N & \mathbf{0}_N \\ \mathbb{K}\mathbb{M}(c_\alpha, c_\beta) & \mathbf{0}_N & \mathbf{0}_N \\ \hline \mathbf{0}_N & \mathbf{0}_N & \mathbf{0}_N \end{array} \right),$$

$$\mathbf{b}_{np} = \frac{\varepsilon z_b}{\tau_z} \begin{pmatrix} \mathbf{0}_N \\ \mathbf{0}_N \\ \mathbf{1}_N \end{pmatrix}, \quad \mathbf{b}_p(k) = \varepsilon \begin{pmatrix} \mathbf{0}_N \\ a_2 k \\ \mathbf{0}_N \end{pmatrix}, \quad \mathbf{b}_{nl}(\mathbf{w}) = \begin{pmatrix} -x_1^3 - \phi_f(z_1) \\ \vdots \\ -x_N^3 - \phi_f(z_N) \\ \mathbf{0}_N \\ \varepsilon \phi_r(x_1) \\ \vdots \\ \varepsilon \phi_r(x_N) \end{pmatrix}.$$

With a first look into the model, we can easily see that system (25) depends on many parameters, so the reduction could be unaffordable because of the immense number of parameter vectors that we would have to consider.

In order to reduce the number of parameters to consider, we set random heterogeneity parameters  $k_i \in [1, 1.5]$  following a truncated normal distribution  $\mathcal{N}(1.25, 0.25)$ , fixed for all the simulations, that is, every cell is slightly different from each other. Furthermore, we fix  $c_\alpha = 1$ , that is, we consider that the cells within each cluster are coupled in an excitatory fashion. Therefore, we perform our reduction with respect to the inter-cluster coupling parameter  $c_\beta$ , that is, the coupling between cells of different clusters. This selection is made to be in concordance with previous works (Bandera Moreno et al., 2022), and to avoid chaos and bizarre behaviors, see Figure 4.2. We present in the following section the behaviors in which we will be focusing on.

### *Behaviors to study*

We focus now on the explanation of the behaviors that will be our object of study for the reduction of problem (25). We will focus on three main behaviors, for  $c_\alpha = 1$ , see rightmost border of Figure 4.6, that we explain more in detail in the following. These behaviors differ slightly from the ones presented in Figure 4.2 because of the heterogeneity.

#### 1. Almost-in-Phase synchronization (AiP). For $c_\beta \in [0.1, 1]$ .

In this range of the parameter, each cell reaches asymptotically an orbit similar to the cycle existing for the relaxation oscillation of the uncoupled case without Mixed Mode Oscillations. All the 3D oscillators are synchronized with a small

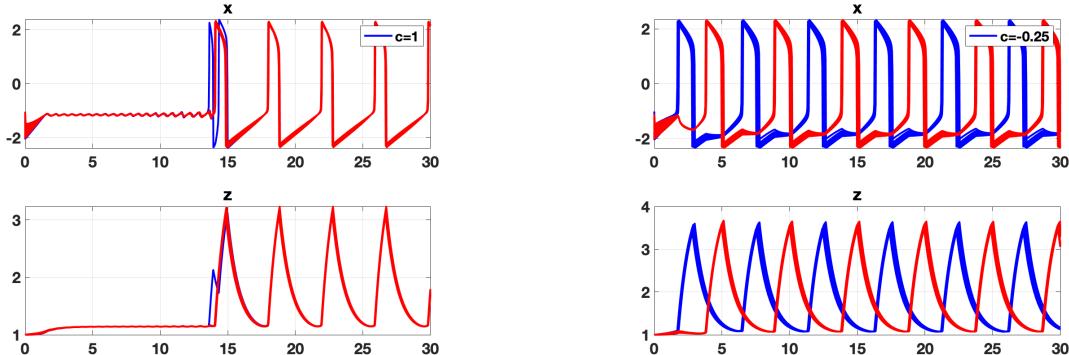
phase-shift or change of frequency depending on its heterogeneity parameter. This behavior is shown in Figure 4.7a.

2. **Antiphase Synchronization (AS).** For  $c_\beta \in [-0.45, -0.05]$ .

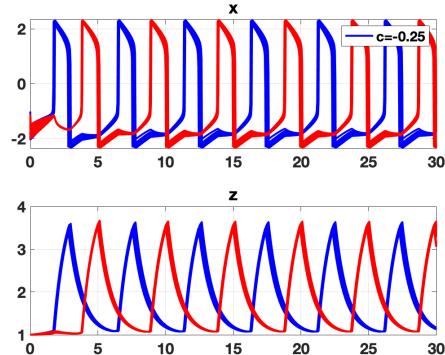
In this range of the parameter, each cell within each cluster reach asymptotically the same orbit, and the phase-shift between the clusters is close to half of the period with a small phase-shift or change of frequency depending on its heterogeneity parameter. This behavior is shown in Figure 4.7b.

3. **Total Oscillation Death (TOD).** For  $c_\beta \in [-1, -0.55]$ .

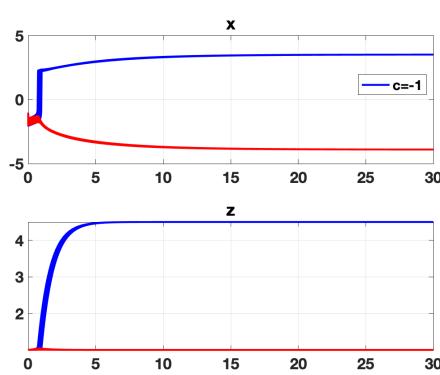
In this range of the parameter, each cell within each cluster reaches asymptotically a stable equilibrium. This behavior is shown in Figure 4.7c.



(a) AiP synchronization behavior of system (25) for  $c_\beta = 1$ .



(b) AS behavior of system (25) for  $c_\beta = -0.25$ .



(c) TOD behavior of system (25) for  $c_\beta = -1$ .

**Figure 4.7:** Representation of the different behaviors to study. In blue, the cells belonging to the first cluster and in red, the cells belonging to the second cluster. We only represent  $x$  and  $z$ , for the sake of clarity.

### *Blocks of variables*

As this test will be the first one performed with the techniques developed in Chapter 3, we need to consider different configurations for them, in order to have a good setup for comparisons with the previous methods. As we have two clusters with three types of variables, we will perform the POBD with 3 and 6 blocks, the first POBD will have one block per variable ( $x, y, z$ ), and we will name it POBD3, and the second POBD will have one block per cluster variable ( $x_I, x_{II}, y_I, y_{II}, z_I, z_{II}$ ), where subindex  $I$  denotes the variables relatives to the first cluster and subindex  $II$  denotes the variables relatives to the second cluster. We will name this method POBD6. As APOBD groups the blocks automatically, we start with the maximum number of possible blocks, in our case 6, one per cluster and variable, the same as for POBD6.

We have now all the ingredients needed for the study of the techniques developed in Chapter 3 for this test case.

#### 4.1.3 Numerical Results

In this section, we will perform various numerical tests for different number of cells, namely,  $N = 200, 500, 1000, 2000$ .

In every test case, we will let the original system (25) compute from  $T = 0$  to  $T = 50$  via a fourth order Runge-Kutta solver with fixed time step of  $h = 10^{-3}$ . We perform every method with the data from  $T = 35$  to  $T = 50$  as we need that the system is almost at an asymptotic behavior.

In the construction of reduced systems, the tolerance criteria for the collinearity analysis in the APOBD method is set at 0.2, that give us two blocks in AiP behavior, one for the fast variables ( $x$ ) and another one for the slow ones ( $y, z$ ), four blocks in AS behavior,  $(x_I)$ ,  $(x_{II})$ ,  $(y_I, z_I)$  and  $(y_{II}, z_{II})$ , that is, one block per timescale and cluster, and just one block for TOD behavior, that is, all the variables mixed together.

The tolerance criteria of the selection of the reduced dimension in POD methods is set to  $10^{-9}$  for AiP behavior and  $10^{-6}$  for AS and TOD behaviors. Nonlinear terms of system (25) are addressed through the Empirical Interpolation Method (EIM, see Section 1.4.1 and Barrault et al. (2004) and Grepl et al. (2007)) with a tolerance of  $10^{-6}$ .

In the following, we compare the number of equations, the sparsity of the reduced model via the sparsity index, and the computational time spent to solve the reduced system from  $T = 50$  to  $T = 70$ , via the speed-ups, defined as the quotient between

the computational time of the original system (25) and the computational time of the reduced system.

The objectives of this test case are the following:

- First, to validate the new proposed methods. In order to do so, we compare all the methods with  $N = 200$  cells in the three behaviors mentioned above.
- Then, in order to compare in more detail the methods, we will test them with a more large amount of cells, that is,  $N = 500, 1000, 2000$  for AiP and AS behaviors, as the TOD behavior has not a major interest, as it reproduces the system in a steady state and the reduction is straightforward, as we will see in the following.

In Table 4.1, we show the simulation data of the original system (25) for every test case.

Test Case	$N = 200$	$N = 500$	$N = 1000$	$N = 2000$
Number of equations	600	1500	3000	6000
Sparsity index	0.8867	0.8880	0.8884	0.8887
CPU time	$\approx 6.2$ s	$\approx 14$ s	$\approx 126$ s	$\approx 475$ s

Table 4.1: Data obtained for system (25) for different test cases depending on the number of cells considered. We show the number of equations, the corresponding sparsity index and the CPU time.

### *Validation of the methods for $N = 200$*

The results of the reductions are summarized in Tables 4.2-4.4.

In Table 4.2, we see that for this test, POD performs better in terms of speed-up just for AiP behavior, but in terms of sparsity, POBD6 provides the best index in the three cases, that is, the best separation of variables. We see that both POBD3 and APOBD provide a well trade-off between sparsity and speed-up.

Method	POD	POBD3	POBD6	APOBD
Number of equations	33	$26 + 12 + 10$	$21 + 23 + 10 + 11 + 8 + 9$	$26 + 16$
Sparsity index	0	0.6510	<b>0.7518</b>	0.3685
Speed-up	<b>6.04</b>	4.20	2.70	4.24
Error ( $\cdot 10^{-3}$ )	5.06	4.46	6.95	<b>4.35</b>

Table 4.2: AiP behavior. Comparison of the data for the reduced system for the original system (25) with  $N = 200$ . Notation  $a+...+c$  indicates the number of equations in each block. In bold fashion, the best results.

In Table 4.3, we can see the same indexes, but now for AS behavior, POBD6 and APOBD are the methods that provide the best speed-ups and sparsity indexes.

Method	POD	POBD3	POBD6	APOBD
Number of equations	57	$53 + 14 + 8$	$27 + 26 + 7 + 7 + 4 + 4$	$27 + 26 + 9 + 8$
Sparsity index	0	0.6874	<b>0.7739</b>	0.6937
Speed-up	2.02	1.67	<b>2.20</b>	<b>2.17</b>
Error ( $\cdot 10^{-3}$ )	<b>3.59</b>	7.80	6.24	6.98

Table 4.3: AS behavior. Comparison of the data for the reduced system for the original system (25) with  $N = 200$ . Notation  $a+...+c$  indicates the number of equations in each block. In bold fashion, the best results.

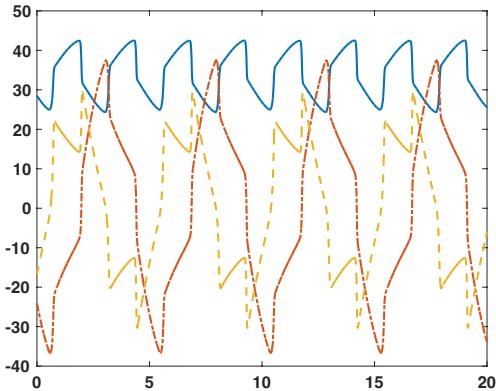
Finally, in Table 4.4, we can see that for TOD behavior, APOBD provides the best speed-up, while POBD6 provides the best separation of variables and error.

Method	POD	POBD3	POBD6	APOBD
Number of equations	1	$1 + 1 + 1$	$1 + 1 + 1 + 1 + 1 + 1$	1
Sparsity index	0	0.4444	<b>0.6667</b>	0
Speed-up	34	15.32	9.29	<b>40.06</b>
Error ( $\cdot 10^{-4}$ )	3.09	0.40	<b>0.21</b>	3.09

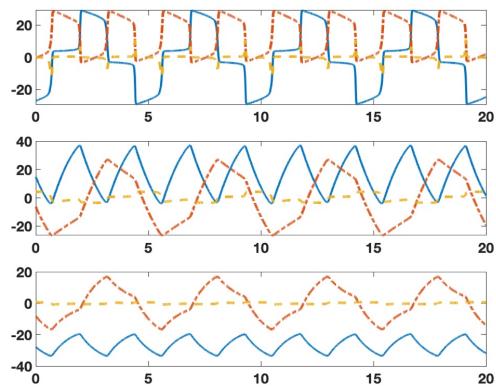
Table 4.4: TOD behavior. Comparison of the data for the reduced system for the original system (25) with  $N = 200$ . Notation  $a+...+c$  indicates the number of equations in each block. In bold fashion, the best results.

For illustration purposes, in Figure 4.8 we show the first 3 basis obtained for each method and each block in AS behavior. Note that the basis for POD (Figure 4.8a) and POBD3 (Figure 4.8b) are much more erratic, and we cannot identify neither the separation between variables nor clusters, while for POBD6 (Figure 4.8c) and APOBD (Figure 4.8d) this separation is clearer, due to the higher sparsity index. This means that, we could not need to go back to the full-dimensional space to analyze the data, allowing us to save more computational time.

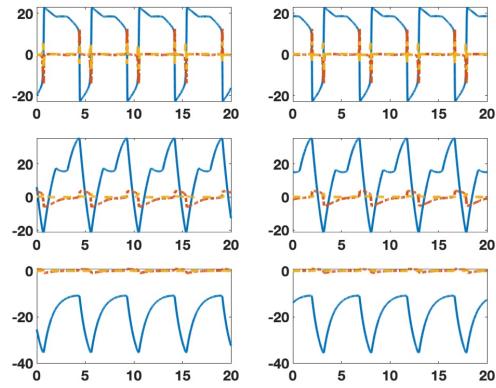
At this point, we can affirm that this validation test is finished successfully. The reduced methods proposed are able to reproduce the solution of the original problem, while performing a reduction in computational times. It should be noted that while all the methods are comparable in terms of sparsity and error, the novel methods also provide us with a new feature, the perseverance of the structure of the original system. Now, we are able to continue our study. In next sections, we will focus just on the AiP and AS behaviors. As mentioned in the previous, the TOD behavior is



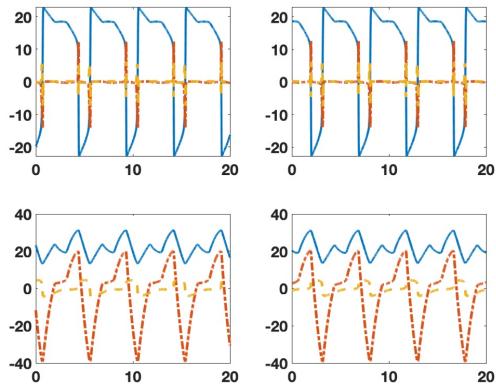
(a) First three basis for the POD method for  $N = 200$  cells in AS behavior.



(b) First three basis for each block of the POBD3 method for  $N = 200$  cells in AS behavior.



(c) First three basis for each block of the POBD6 method for  $N = 200$  cells in AS behavior.



(d) First three basis for each block of the APOBD method for  $N = 200$  cells in AS behavior.

**Figure 4.8:** Comparison of the basis for each method. In all the cases, the first basis is in blue continuous fashion, the second in red dash-dotted fashion and the third basis in yellow dashed fashion.  
a) POD. b) POBD3. c) POBD6. d) APOBD.

not of significant interest as it simply replicates the system in a steady state, and the reduction process is straightforward.

#### Comparison for larger number of cells in AiP behavior

In this section, we study in detail the reduction in the AiP behavior of system (25) for larger numbers of cells,  $N = 500, 1000, 2000$ .

The results are summarized in Table 4.5, which shows the number of variables of each cells number case, and figures 4.9a-4.9c, which represent the evolution of sparsity, speed-up and error, respectively.

Number of cells	POD	POBD3	POBD6	APOBD
500	36	$29 + 14 + 11$	$25 + 27 + 12 + 13 + 10 + 11$	$29 + 18$
1000	39	$32 + 14 + 12$	$28 + 31 + 12 + 14 + 10 + 12$	$32 + 18$
2000	42	$36 + 15 + 12$	$35 + 34 + 15 + 14 + 12 + 11$	$36 + 18$

Table 4.5: Comparison of the number of equations of the reduced systems for the original system (25) in AiP behavior.

From Table 4.5, we can see that APOBD minimizes the number of equations with respect to the POBD methods, while maintaining the structure of the system.

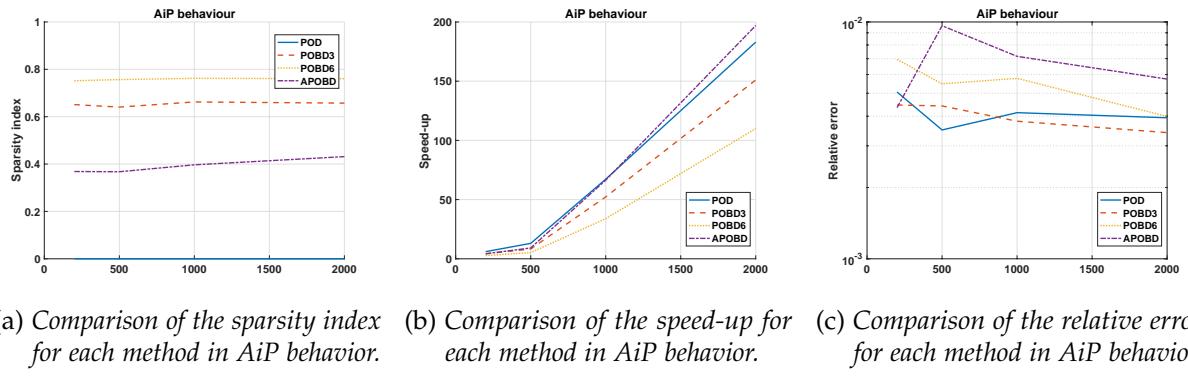


Figure 4.9: Comparisons of the results in sparsity index (left), speed-up (center), and relative error (right), for each method in AiP behavior of system (25).

The best method in terms of sparsity is POBD6 in every case, see Figure 4.9a, however, a better sparsity index does not imply a better speed-up, as it is also the worst method in terms of speed-up, see Figure 4.9b. It seems that the best method in terms of the trade-off between sparsity and speed-up is APOBD. We can observe that the relative errors for every method are comparable, as they have the same order, see Figure 4.9c.

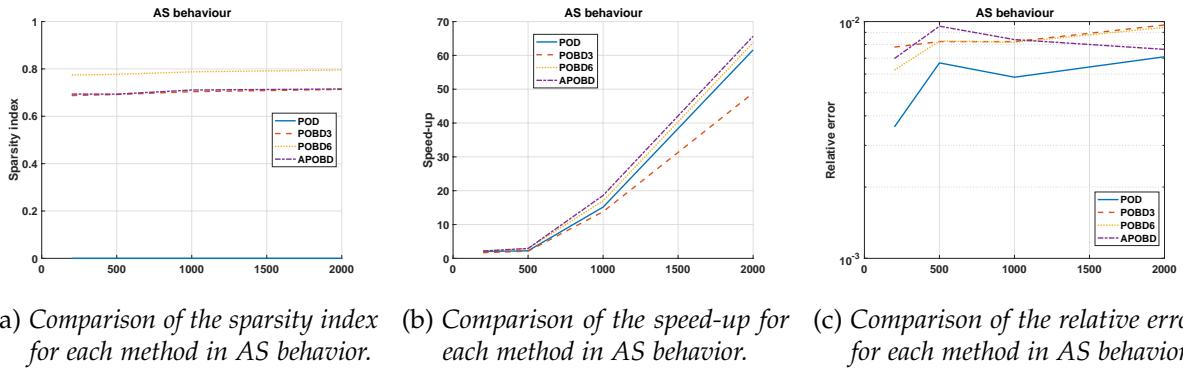
#### Comparison for larger number of cells in AS behavior

In this section, we study in detail the reduction in the AS behavior of system (25) for larger numbers of cells,  $N = 500, 1000, 2000$ .

The results are summarized in Table 4.6, which shows the number of variables of each cell number case, and figures 4.10a-4.10c, which represent the evolution of sparsity, speed-up and error, respectively. From Table 4.6, we can see again that APOBD minimizes the number of equations with respect to the POBD methods, while maintaining the structure of the system.

Number of cells	POD	POBD3	POBD6	APOBD
500	66	$61 + 16 + 9$	$31 + 30 + 8 + 8 + 5 + 4$	$31 + 30 + 10 + 9$
1000	71	$68 + 26 + 8$	$33 + 35 + 8 + 8 + 4 + 5$	$33 + 35 + 9 + 10$
2000	75	$72 + 16 + 8$	$36 + 37 + 8 + 8 + 5 + 4$	$36 + 37 + 10 + 10$

Table 4.6: Comparison of the number of equations of the reduced systems for the original system (25) in AS behavior.



(a) Comparison of the sparsity index for each method in AS behavior. (b) Comparison of the speed-up for each method in AS behavior. (c) Comparison of the relative error for each method in AS behavior.

**Figure 4.10:** Comparisons of the results in sparsity index (left), speed-up (center), and relative error (right), for each method in AS behavior of system (25).

The best method in terms of sparsity is POBD6 in every case, see Figure 4.10a, however, we see again that a better sparsity index does not imply a better speed-up, as it is not the best method in terms of speed-up, see Figure 4.10b. We can observe that methods POBD3 and APOBD provide a similar sparsity index, but APOBD is much better in terms of speed-up. Again, it seems that the best method in terms of the trade-off between sparsity and speed-up is APOBD. Again, the relative errors are comparable between the methods, see Figure 4.10c.

It is clear that our proposed method APOBD gives the best results in terms of the trade-off between model reduction, saving of computational time and explainability of the reduced variables, as it needs a number of equations that is very low in comparison with the original system, and the computational times reductions and errors are competitive for all the behaviors studied. We can also note that APOBD performs better the larger the number of equations of the original problem is. Furthermore, it also gives us a new feature, explainability of the reduced variables, that could also help the post-processing of the results as we do not need to go back to the full-dimensional space.

## 4.2 APPLICATION TO BURSTERS MODELS

In this section, our aim is to apply the APOBD technique to other models with multiple timescales. More precisely, we will apply the APOBD technique to two bursters models. Bursters display a clear alternating pattern between periods of activity and silence. In the case of slow-fast bursters, we observe two distinct rhythms: one functioning during the active phase (fast) and another between active and silent phases (slow). This phenomenon is widespread across a spectrum of physical and biological systems, carrying substantial importance across multiple fields of study.

On the one hand, in Section 4.2.1 we study the Hindmarsh-Rose model, that exhibit two fast variables and one slow. On the other hand, we study in Section 4.2.2 pancreatic  $\beta$ -cells models, that exhibit three different timescales.

### 4.2.1 Hindmarsh-Rose Network model

In this section, we will consider another type of single-cell dynamics used for building a network model. The Hindmarsh-Rose model formalizes the dynamics of neuronal activity in a single neuron and was initially introduced in Hindmarsh and Rose (1984). The model involves three state variables: the membrane potential  $x$ , a recovery variable  $y$  controlling the neuron ability to recover from depolarization and a gating variable  $z$  regulating the dynamics of the neuron's spiking activity. Moreover, one actually considers the gating variable dynamics to be slow compared to the membrane potential and recovery variable dynamics, which is materialized by parameter  $\varepsilon$ , assumed to be small and positive, in the  $z$ -dynamics. Thus, the model is the following:

$$\begin{cases} \dot{x} = l(x) + y - z + I, \\ \dot{y} = c + m(x) - y, \\ \dot{z} = \varepsilon(k(x - x_0) - z), \end{cases}$$

where  $l(x) = -ax^3 + bx^2$  and  $m(x) = -dx^2$ . The timescale separation parameter  $\varepsilon$  fulfills  $0 < \varepsilon \ll 1$ , parameter  $k > 0$  has been introduced so that the outputs fit realistic biological evolution patterns and parameter  $I$  stands for the external exciting current. Finally,  $a, b, c$  and  $d$  are positive parameters, and  $x_0 < 0$ .

The membrane potential is mainly driven by a nonlinear voltage-dependent conductance term that captures the neuron's response to input currents: a cubic polynomial

function introduces bistability and the potential for oscillations. Recovery and gating variables are driven by linear intrinsic terms. Additionally, the coupling term of  $y$  upon  $x$ -dynamics is linear and influences the excitability by modulating the rate at which the recovery variable affects the membrane potential: an increase in the recovery variable  $y$  leads to an increase in the membrane potential  $x$ , enhancing the excitability of the neuron. The coupling term of the gating variable onto the potential dynamics is also linear, but inhibitory. We have considered a quadratic term for the feedback of  $x$  upon  $y$ , ensuring that the feedback is always inhibitory, and a linear one upon  $z$ .

We will test the POD and the APOBD on the following network of Hindmarsh-Rose Network system that has been previously introduced and studied in the literature (Barrio & Shilnikov, 2011; Bonaventura et al., 2022; Etémé et al., 2017)

$$O_i \begin{cases} \dot{x}_i = l(x_i) + y_i - z_i + I + \frac{1}{N-1} \sum_{j=1}^N c_{ij}(x_j - x_i), \\ \dot{y}_i = c + m(x_i) - y_i, \\ \dot{z}_i = \varepsilon(k_i(x_i - x_0) - z_i), \end{cases} \quad (26)$$

where  $i = 1, 2, \dots, N$ . In this test, we consider a normalized version of the coupling, with  $c_{ij} = 1, \forall i \neq j$ , and  $c_{ii} = 0$ . We also add some heterogeneity in the slow equation via  $k_i \in [3.9, 4.1]$  following a uniform distribution.

The Hindmarsh-Rose model is different from the Intracellular Calcium Concentration Model as it has two fast variables and just one slow variable. It also exhibits a wide range of dynamical behaviors depending on the parameter values and therefore is able to capture a wide panel of neuronal activity features, e.g. sustained oscillations (or regular spiking), various types of bursting oscillations, chaotic oscillations, etc. As a further difference, we will not consider the cells separated into two clusters, here we will suppose that all the cells belongs to the same cluster.

### *Preliminary study*

In this section, we build the algebraic version of the original network problem (26), then, we will present the behaviors that we want to study, and, finally, we will determine the *a priori* number of blocks and their configuration for the APOBD technique.

### *Algebraic version*

In order to build the algebraic version of model (26), we begin by simplifying the  $x$  equations of the model. We can consider a matrix  $\mathbb{D} = (d_{ij})$ , with

$$d_{ij} = \frac{1}{N-1} \left( \delta_{ij} \sum_{s=1}^N c_{sj} - c_{ij} \right),$$

where  $\delta_{ij}$  is the Kronecker delta. Therefore, the  $x$  equations can be rewritten as

$$\dot{x}_i = l(x_i) + y_i - z_i + I + \sum_{j=1}^N d_{ij} x_j.$$

Now, denoting by  $\gamma$  the parameter vector, the previous allows us to rewrite the system (26) in vector notation as in the following

$$\mathbf{w} = (\mathbb{A}_p(\gamma) + \mathbb{A}_{np})\mathbf{w} + \mathbf{b}(\gamma) + \mathbf{b}_{nl}(\mathbf{w}; \gamma), \quad (27)$$

where  $\mathbb{A}_{np}$  denotes the non-parametric part,  $\mathbb{A}_p$  and  $\mathbf{b}$  the matrix and vector parametric parts, respectively, and  $\mathbf{b}_{nl}$  stands for the nonlinear part of system (26). They can be expressed as follows

$$\begin{aligned} \mathbb{A}_{np} &= \left( \begin{array}{c|c|c} \mathbb{O}_N & \mathbb{I}_N & -\mathbb{I}_N \\ \hline \mathbb{O}_N & -\mathbb{I}_N & \mathbb{O}_N \\ \hline \mathbb{O}_N & \mathbb{O}_N & \mathbb{O}_N \end{array} \right), \quad \mathbb{A}_p(\gamma) = \left( \begin{array}{c|c|c} \mathbb{D} & \mathbb{O}_N & \mathbb{O}_N \\ \hline \mathbb{O}_N & \mathbb{O}_N & \mathbb{O}_N \\ \hline \varepsilon \mathbb{K} & \mathbb{O}_N & -\varepsilon \mathbb{I}_N \end{array} \right), \\ \mathbf{b}(\gamma) &= \begin{pmatrix} I\mathbf{1}_N \\ c\mathbf{1}_N \\ -\varepsilon x_0 \mathbf{k} \end{pmatrix}, \quad \mathbf{b}_{nl}(\mathbf{w}; \gamma) = \begin{pmatrix} \mathbf{l}(\mathbf{x}) \\ \mathbf{m}(\mathbf{x}) \\ \mathbf{0}_N \end{pmatrix}. \end{aligned}$$

With a quick look, we can see that model (27) depends on several parameters. In the next epigraph, we will set the behaviors that we are trying to model along with its parameters values.

### *Behaviors to study*

We focus now on the explanation of the behaviors that will be our object of study for the reduction of problem (27). In this case, we will focus on two different behaviors.

### 1. Sustained Oscillation (SO).

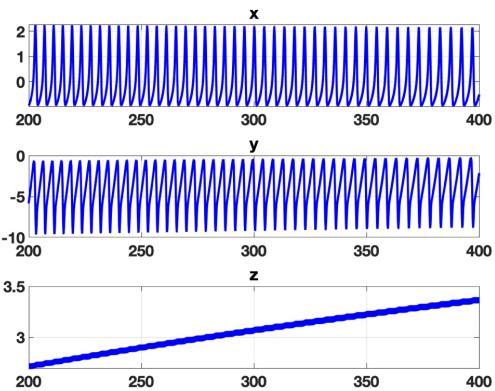
This behavior was considered in Bonaventura et al. (2022). It is obtained for parameters with classical values  $a = 1$ ,  $b = 3$ ,  $c = 1$  and  $d = 5$ . The separation between timescales is set to  $\varepsilon = 10^{-3}$  and the external exciting current is set to  $I = 5$ .

For this selection of the parameters, each cell is spiking with no quiescent phase, and all 3D oscillators are strongly synchronized with a slight phase shift. This behavior is shown in Figure 4.11a.

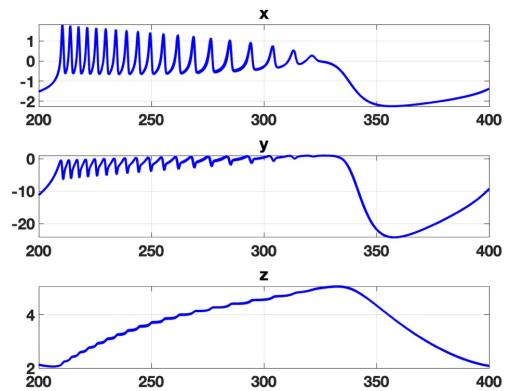
### 2. Plateau-like Bursting (PB).

This behavior was studied in Barrio and Shilnikov (2011). It is obtained for parameter values  $a = 1$ ,  $b = 2.52$ ,  $c = 1$  and  $d = 5$ . The separation between timescales is set to  $\varepsilon = 10^{-2}$  and the external exciting current is set to  $I = 4$ .

This behavior is characterized by repeated cycles of spiking followed by a prolonged depolarized state known as plateau. All the cells starts their spiking regime at the same time, gradually desynchronize their spikes during the spiking regime, and resynchronize during the quiescence phase. This behavior is shown in Figure 4.11b.



(a) SO behavior of system (26) for  $N = 100$  cells.



(b) PB behavior of system (26) for  $N = 100$  cells.

**Figure 4.11:** Representation of the different behaviors to study of system (26). Left panel shows SO behavior, and right panel shows PB behavior.

### Blocks of variables

In this case, we will only consider the POD and APOBD techniques. For the latter, we select one block per variable ( $x, y, z$ ) for both behaviors, as all the cells belong to the same cluster.

### Numerical Results

In the following study, we will consider various instances for the number of cells  $N = 100, 500, 1000$ . In every test case, we will let the original system (27) compute from  $T = 0$  to  $T = 400$  via a fourth order Runge-Kutta solver with fixed time step of  $h = 10^{-2}$ . We perform both reduction methods with the data from  $T = 200$  to  $T = 400$  as we need that the system is almost at an asymptotic behavior. This will be enough to capture the main behavior of the system, as shown in Figure 4.11.

In the construction of reduced systems, the tolerance criteria for the collinearity analysis in the APOBD method is set at 0.2, that gives us three blocks for the SO and PB behaviors. The tolerance criteria of the selection of the reduced dimension in POD methods is set to  $10^{-12}$  for every behavior. Again, nonlinear parts of (27) are addressed by means of the EIM (Barrault et al., 2004; Grepl et al., 2007), now with a tolerance of  $10^{-12}$ .

Table 4.7 summarizes the simulation data of the original system (27) for both test cases.

Test Case	$N = 100$	$N = 500$	$N = 1000$
Number of equations	300	1500	3000
Sparsity index	0.8833	0.8878	0.8883
CPU time	$\approx 5.5$ s	$\approx 100$ s	$\approx 415$ s

Table 4.7: Simulation data for system (27) for different test cases depending on the number of cells considered. We show the number of equations, the corresponding sparsity index and the CPU time.

As done for the previous tests, we compare the number of equations, the sparsity index, and the speed-ups, from  $T = 400$  to  $T = 1000$ , for the original model and the reduced ones. Furthermore, as an additional way to compare the reduced solutions with the original one, we compute the following *a posteriori* error indicator,

$$\Delta_j = \max_i \min_k |t_j^{(i)} - \tilde{t}_k^{(i)}|, \quad (28)$$

where  $t_j^{(i)}$  is the  $j$ -th time when the original model has a local maximum in the  $x$  variable for the cell  $i$ , and  $\tilde{t}_k^{(i)}$  is the  $k$ -th time when the reduced model has a local maximum in the  $x$  variable for the cell  $i$ . This estimator tells us how shifted are the reduced peaks in comparison to the original ones. Next, we summarize the results obtained for each behavior.

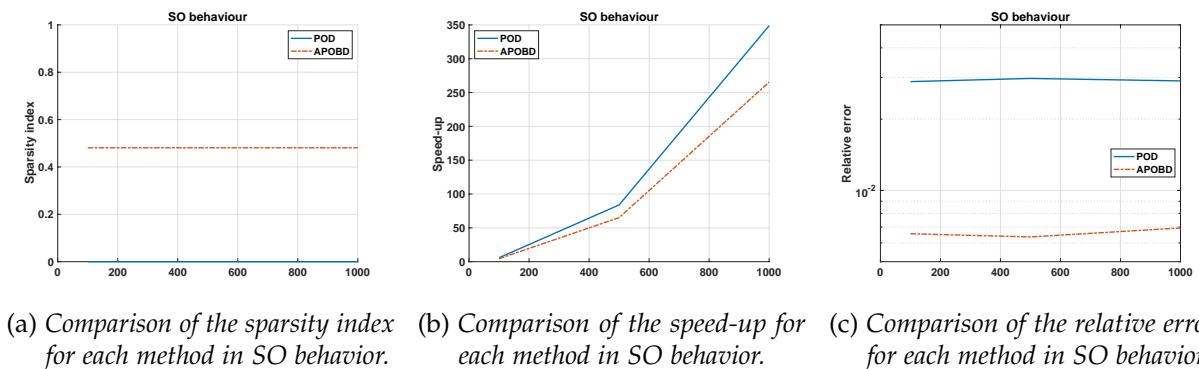
#### Numerical results for SO behavior

In this section, we study in detail the SO behavior of system (27) for  $N = 100, 500, 1000$ .

Number of cells	POD	APOBD
100	15	$7 + 7 + 3$
500	15	$7 + 7 + 3$
1000	16	$7 + 7 + 3$

Table 4.8: Comparison of the number of equations of the reduced systems for the original system (27) in SO behavior.

The results are summarized in Table 4.8, which shows the number of variables of each cell number case, and figures 4.12a-4.12c, which represent the evolution of sparsity, speed-up and error, respectively. From Table 4.8, we can see that the reduction is stronger when the number of original equations rises, as the number of reduced equations remains almost the same.

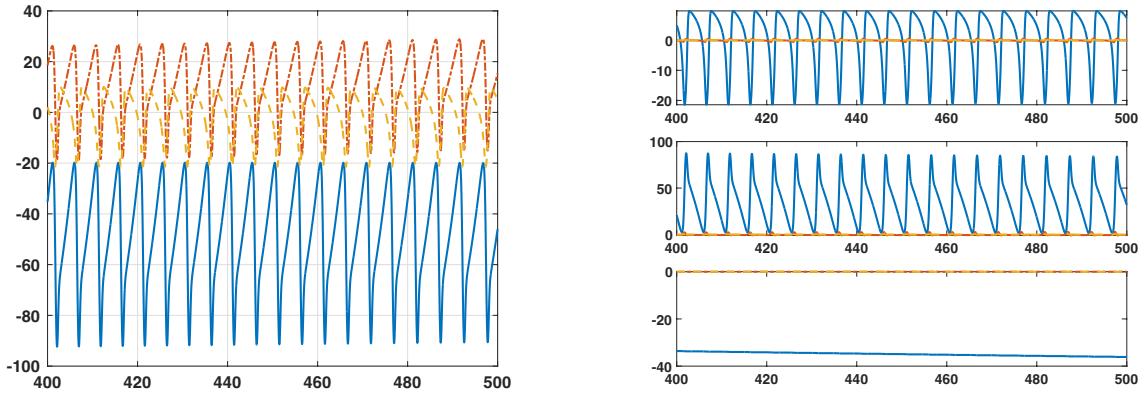


(a) Comparison of the sparsity index for each method in SO behavior. (b) Comparison of the speed-up for each method in SO behavior. (c) Comparison of the relative error for each method in SO behavior.

**Figure 4.12:** Comparisons of the results in sparsity index (left), speed-up (center), and relative error (right), for each method in SO behavior of system (27).

We can see that although the speed-up is better for the POD method (see Figure 4.12b), APOBD provides an additional order in the relative error, Figure 4.12c.

For illustration purposes, in Figure 4.13, we show the first 3 basis obtained for each method and each block in SO behavior. The basis for POD has been represented in



(a) First basis for the POD method for  $N = 100$  cells  
in sustained oscillation behavior.

(b) First basis for each block of the APOBD method for  
 $N = 100$  cells in sustained oscillation behavior.

**Figure 4.13:** Comparisons of the basis for each method obtained in the sustained oscillation behavior. In all the cases, the first basis is in blue continuous fashion, the second in red dash-dotted fashion and the third basis in yellow dashed fashion. a) POD. b) APOBD with 3 blocks.

Figure 4.13a, where we cannot see a clear separation in the timescales. The basis for APOBD has been represented in Figure 4.13b, where this separation is clearer, due to the higher sparsity index.

#### Numerical results for PB behavior

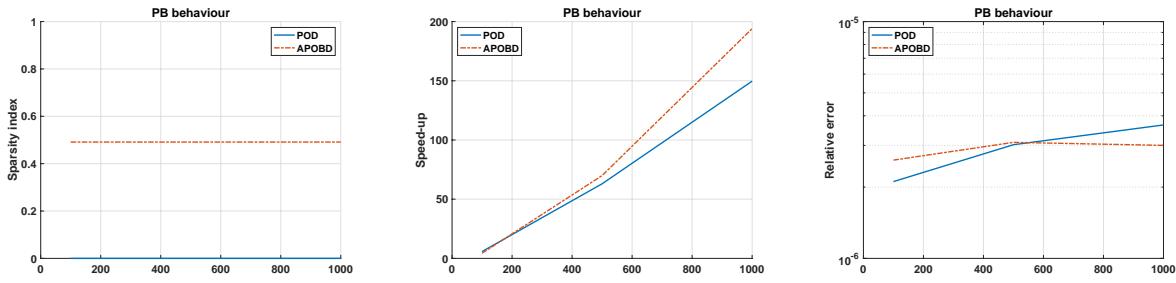
In this section, we study in detail the reduction in the PB behavior of system (27) for  $N = 100, 500, 1000$ .

Number of cells	POD	APOBD
100	15	$7 + 6 + 4$
500	15	$7 + 6 + 4$
1000	15	$7 + 6 + 4$

Table 4.9: Comparison of the number of equations of the reduced systems for the original system (27) in PB behavior.

The results are summarized in Table 4.9, which shows the number of variables of each cell number case, and figures 4.14a-4.14c, which represent the evolution of sparsity, speed-up and error, respectively. From Table 4.9, we can see again that the reduction is stronger when the number of original equations rises, as the number of reduced equations remains almost the same.

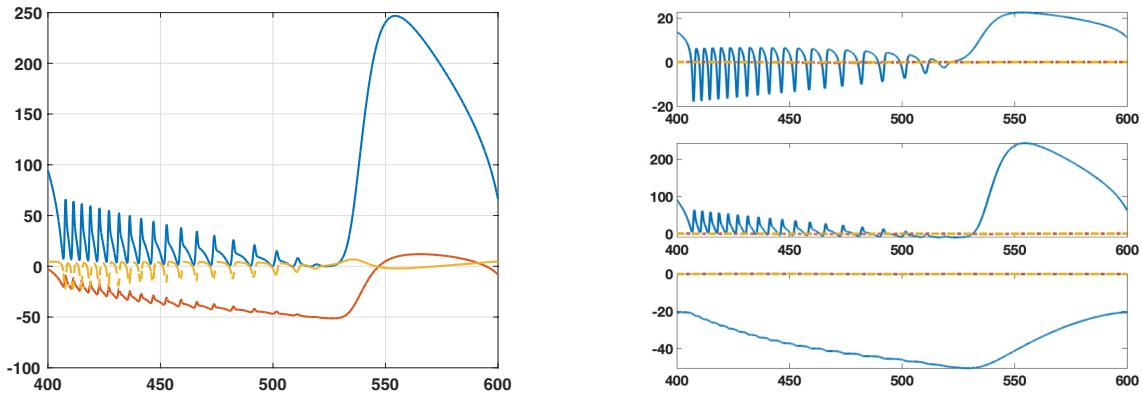
It is clear that the APOBD method provides a better speed-up in comparison with the POD for this behavior, maintaining an error of the same order as the one of the original POD method.



(a) Comparison of the sparsity index for each method in PB behavior. (b) Comparison of the speed-up for each method in PB behavior. (c) Comparison of the relative error for each method in PB behavior.

**Figure 4.14:** Comparisons of the data obtained in sparsity index (left), speed-up (center), and relative error (right), for each method in PB behavior of system (27).

For illustration purposes, we show in Figure 4.15 a comparison between the first 3 basis obtained for each method and block in the PB behavior for POD (Figure 4.15a) and APOBD (Figure 4.15b). In the POD case, we cannot observe a separation in the timescales, while in the APOBD case this separation is clearer, due to the higher sparsity index.

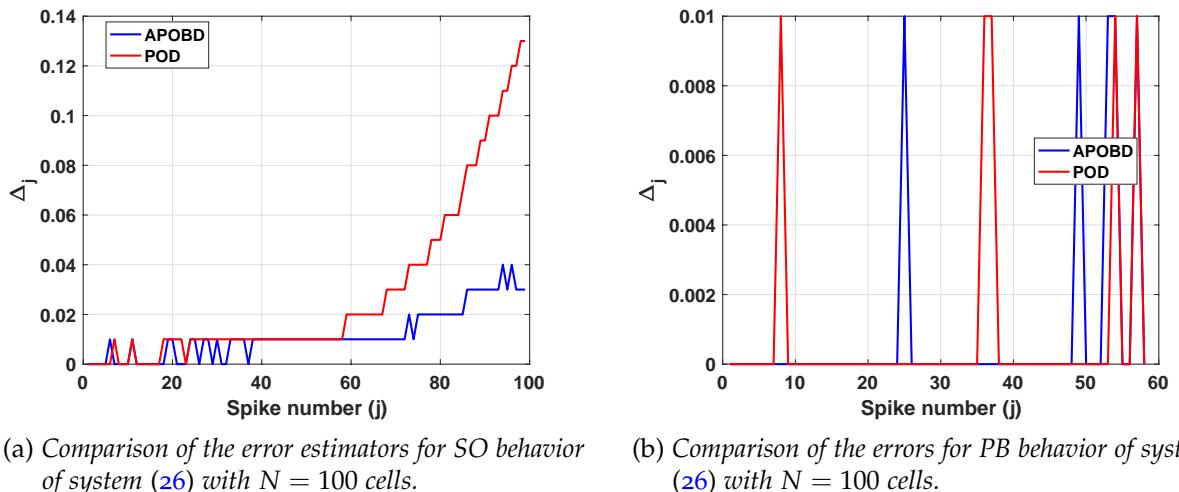


(a) First basis for the POD method for  $N = 100$  cells in PB behavior. (b) First basis for each block of the APOBD method for  $N = 100$  cells in PB behavior.

**Figure 4.15:** Comparisons of the basis for each method obtained in the PB behavior. In all the cases, the first basis is in blue continuous fashion, the second in red dash-dotted fashion and the third basis in yellow dashed fashion. a) POD. b) APOBD.

In Figure 4.16, we show the comparison of the error estimators (28) for both behaviors in the case with  $N = 100$ , as it is similar for  $N = 500$  and  $N = 1000$ . We see that for

the integrating time, APOBD and POD have a great precision in terms of reproducing the spiking times correctly for the PB behavior as the error is, at most, the integrating time step  $h$ , see Figure 4.16b. However, for the SO behavior, as the time increases this precision is lost. This could be due to the transition to a qualitatively different behavior after a stable transient, as it will be discussed in Section 4.3. Nevertheless, according to Figure 4.16a, APOBD performs a more accurate approximation of the spiking times and bursting duration, which is consistent with the relative error shown in Figure 4.12c.



**Figure 4.16:** Comparison of errors of the reduced solutions of system (26) with  $N = 100$ , in red fashion, POD, and in blue fashion, APOBD. a) SO. b) PB.

It seems that APOBD method performs better when there is a switch between the fast behavior and the slow transient. To test it more in detail, we will compare it with the POD method for another model that exhibits this kind of behavior.

#### 4.2.2 Fold/Hopf Bursting Network model

In this section, we test the POD and APOBD methods in a case of "Fold/Hopf" (also called "tapered") bursting studied, for instance, in Izhikevich (2000). As reported in Izhikevich (2000), this type of bursting has been found in models of insulin-producing

pancreatic  $\beta$ -cells (Pernarowski, 1994; Smolen et al., 1993) and in models of some enzymatic systems (Holden & Erneux, 1993a, 1993b). The model reads:

$$\begin{cases} \dot{x} = x - x^3/3 - y, \\ \dot{y} = \varepsilon_1(a + x - S(y, z)), \\ \dot{z} = \varepsilon_2 k x, \end{cases}$$

where  $S(y, z) = b/(1 + \exp((z - y)/d))$ ,  $a = 1.55$ ,  $b = 2.5$ ,  $d = 0.1$ , and the timescale separation parameters are set so that the system has three timescales,  $\varepsilon_1 = 0.5$  and  $\varepsilon_2 = 0.01$ . Note that, then, this model is different from the previous ones in terms of separation timescales as is the only one featuring three timescales  $1, \varepsilon_1$  and  $\varepsilon_2$ .

In this type of burster, the stable equilibrium corresponding to the resting state disappears through a fold bifurcation and the limit cycle attractor corresponding to the spiking state shrinks to a point though a supercritical Hopf bifurcation. Moreover, the fast subsystem undergoes another bifurcation while in the excited state, a fold bifurcation corresponding to the transition from the excited equilibrium to the resting equilibrium, closing the hysteresis loop.

The Fold/Hopf Bursting network model can be written as follows,

$$O_i \begin{cases} \dot{x}_i = x_i - x_i^3/3 - y_i + \frac{1}{N-1} \sum_{j=1}^N c_{ij}(x_j - x_i), \\ \dot{y}_i = \varepsilon_1(a + x_i - S(y_i, z_i)), \\ \dot{z}_i = \varepsilon_2 k_i x_i, \end{cases} \quad (29)$$

where  $i = 1, 2, \dots, N$ . In this test, we consider again a normalized version of the coupling, with  $c_{ij} = 1, \forall i \neq j$ , and  $c_{ii} = 0$ . We also add some heterogeneity in the slow equation via  $k_i \in [3.9, 4.1]$  following a uniform distribution.

### *Preliminary study*

In this section, we will build the algebraic version of the original network problem (29), then, we will present the behavior that we want to study, and finally, we will determine the *a priori* number of blocks and their configuration for the APOBD technique.

*Algebraic version*

In order to build the algebraic version of model (29), we begin, as in the previous case, by simplifying the  $x$  equations of the model. We can consider  $\mathbb{D} = (d_{ij})$ , with

$$d_{ij} = \frac{1}{N-1} \left( \delta_{ij} \sum_{s=1}^N c_{sj} - c_{ij} \right),$$

so, the  $x$  equations can be rewritten as

$$\dot{x}_i = x_i - x_i^3/3 - y_i + \sum_{j=1}^N d_{ij} x_j.$$

Now, denoting by  $\gamma$  the parameter vector, the previous simplification allows us to rewrite system (29) as follows

$$\mathbf{w} = (\mathbb{A}_{np} + \mathbb{A}_p)\mathbf{w} + \mathbf{b}(\gamma) + \mathbf{b}_{nl}(\mathbf{w}; \gamma), \quad (30)$$

where  $\mathbb{A}_{np}$  denotes the non-parametric part,  $\mathbb{A}_p$  and  $\mathbf{b}$  the matrix and vector parametric parts, respectively, and  $\mathbf{b}_{nl}$  stands for the nonlinear part of the system (29). They can be expressed as follows

$$\begin{aligned} \mathbb{A}_p(\gamma) &= \left( \begin{array}{c|c|c} \mathbb{D} & \mathbb{O}_N & \mathbb{O}_N \\ \hline \mathbb{O}_N & \mathbb{O}_N & \mathbb{O}_N \\ \hline \varepsilon_2 \mathbb{K} & \mathbb{O}_N & \mathbb{O}_N \end{array} \right), \quad \mathbb{A}_{np} = \left( \begin{array}{c|c|c} \mathbb{I}_N & -\mathbb{I}_N & \mathbb{O}_N \\ \hline \varepsilon_1 \mathbb{I}_N & \mathbb{O}_N & \mathbb{O}_N \\ \hline \mathbb{O}_N & \mathbb{O}_N & \mathbb{O}_N \end{array} \right), \\ \mathbf{b}(\gamma) &= \left( \begin{array}{c} \mathbf{0}_N \\ \varepsilon_1 a \mathbf{1}_N \\ \mathbf{0}_N \end{array} \right), \quad \mathbf{b}_{nl}(\mathbf{w}; \gamma) \left( \begin{array}{c} x^3/3 \\ 1 \\ \hline 1 + \exp((z - y)/d) \\ \mathbf{0}_N \end{array} \right). \end{aligned}$$

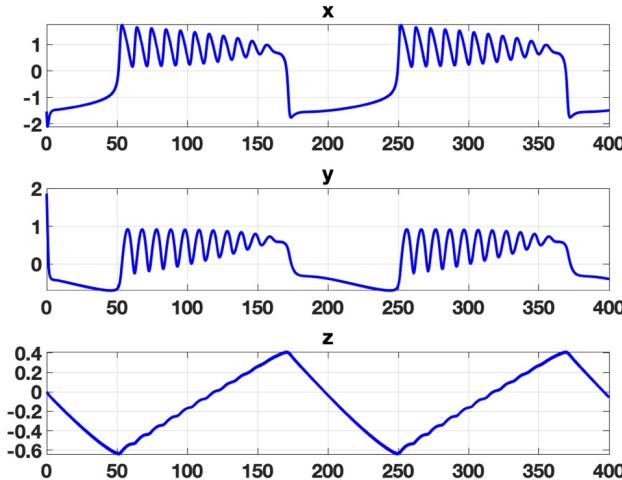
Again, we can easily see that model (30) depends on several parameters. In the next epigraph, we will set the behavior that we are trying to model along with its parameters values.

### *Behaviors to study*

We focus now on the explanation of the behavior that will be our object of study for the reduction of problem (29). In this case, we will focus on one behavior already described:

#### 1. Fold/Hopf Bursting (FHB).

The resting state disappears through a fold bifurcation, and the spiking state disappears through a supercritical Hopf bifurcation. Moreover, the hysteresis loop is closed through another fold bifurcation. See Figure 4.17.



**Figure 4.17:** FHB behavior of system (30) for  $N = 100$  cells.

### *Blocks of variables*

In this case, we will only consider the POD and APOBD techniques. For the latter, we select one block per variable ( $x, y, z$ ) for both behaviors, as done in the previous test.

### *Numerical Results*

In the following study, we will consider various instances for the number of cells  $N = 100, 500, 1000$ . In every test case, we will let the original system (30) compute from  $T = 0$  to  $T = 400$  via a fourth order Runge-Kutta solver with fixed time step of  $h = 10^{-2}$ . We perform both reduction methods with the data from  $T = 200$  to  $T = 400$

as we need that the system is almost at an asymptotic behavior. This will be enough to capture the main behavior of the system as shown in Figure 4.17.

In the construction of reduced systems, the tolerance criteria for the collinearity analysis in the APOBD method is set at 0.2, that gives us three blocks in the behavior of the study. The tolerance criteria of the selection of the reduced dimension in POD methods is set to  $10^{-6}$  for every behavior, and nonlinear parts of system (30) are addressed by means of the EIM with a tolerance of  $10^{-6}$ .

Table 4.10 summarizes the simulation data of the original system (30) for the test case.

Test Case	$N = 100$	$N = 500$	$N = 1000$
Number of equations	300	1500	3000
Sparsity index	0.8833	0.8867	0.8887
CPU time	$\approx 4$ s	$\approx 65$ s	$\approx 280$ s

Table 4.10: Simulation data for system (30) for different test cases depending on the number of cells considered. We show the number of equations, the corresponding sparsity index and the CPU time.

Now, we compare the number of equations, the sparsity index, and the speed-ups, from  $T = 400$  to  $T = 800$ , for the original model and the reduced ones. The results are summarized in Table 4.11, which shows the number of variables of each cell number case, and figures 4.18a-4.18c, which represent the evolution of sparsity, speed-up and error, respectively.

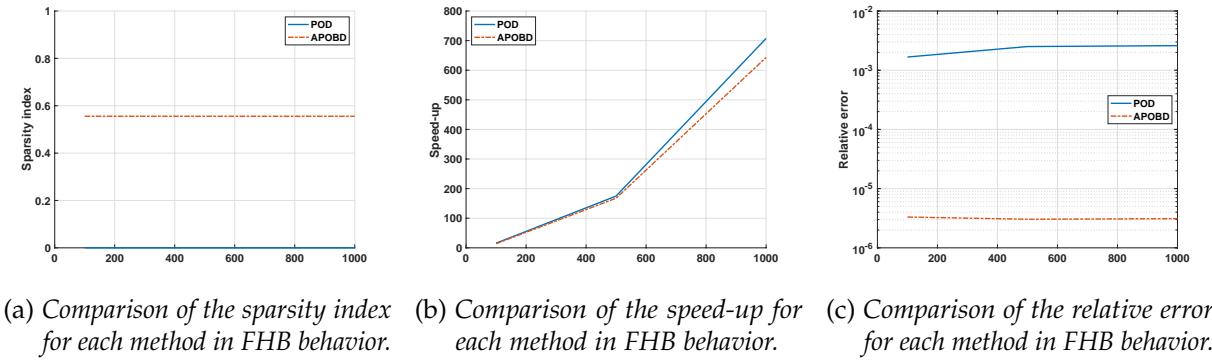
Number of cells	POD	APOBD
100	3	$2 + 2 + 2$
500	3	$2 + 2 + 2$
1000	3	$2 + 2 + 2$

Table 4.11: Comparison of the number of equations of the reduced systems for the original system (30) in FHB behavior.

From Table 4.11, we can see again that the reduction is stronger when the number of original equations rises, as the number of reduced equations remains the same.

We can again see, that although the speed-up is better for the POD method (see Figure 4.18b), APOBD provides three additional orders in the relative error, Figure 4.18c. So, for almost the same computational time, we obtain a much more precise approximation of the solution of the original problem (30).

For illustration purposes, in Figure 4.19, we show the first basis obtained for each method and each block in the studied behavior. The basis for POD are shown in Figure



**Figure 4.18:** Comparisons of the results in sparsity index (left), speed-up (center), and relative error (right), for each method in FHB behavior of system (30).

4.19a, where we cannot see a clear separation in the timescales. The basis for APOBD are shown in Figure 4.19b, where this separation is clearer, thanks to the higher sparsity index.

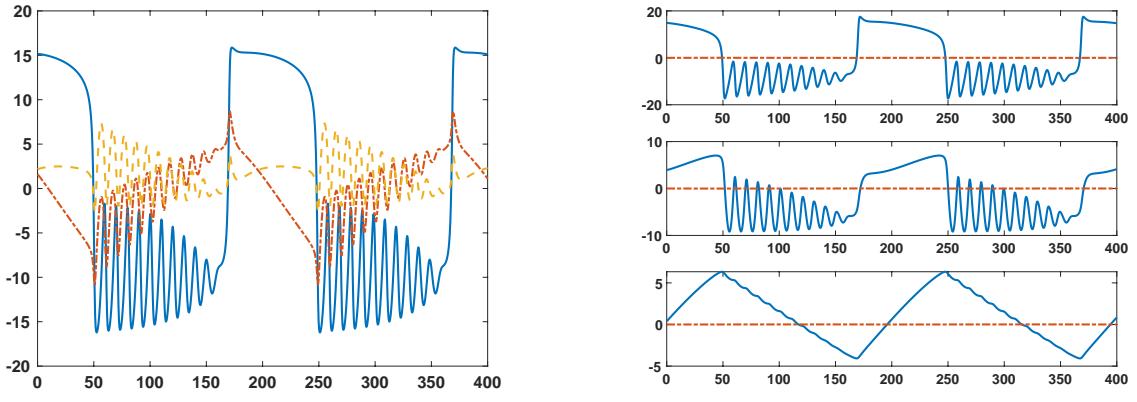
### 4.3 LIMITATIONS

The methods presented in the previous sections have been proven to provide competitive results in terms of maintaining the structure of the original problem, studied via the sparsity index, providing a reduction in the computational time, studied via the speed-up and being true to the original solution, studied via the relative error. However, there are some limitations to these techniques, both older and novel. Here, we will discuss these limitations with examples from Sections 4.1 and 4.2.

#### 4.3.1 Ill-located blocks

As said in the presentation of the APOBD method, we could check if the block were well located, just checking if the elements of the diagonal of the matrix were close to one. A natural question arises: What happens in the case where there is no correlation inside each block of variables? In order to illustrate the results in this case, we consider system (25) for  $N = 100$  with  $c_\alpha = -0.5$ , that is, inhibitory coupling within each cluster, and  $c_\beta \in [-0.25, 0.25]$ , see zone D in Figure 4.2.

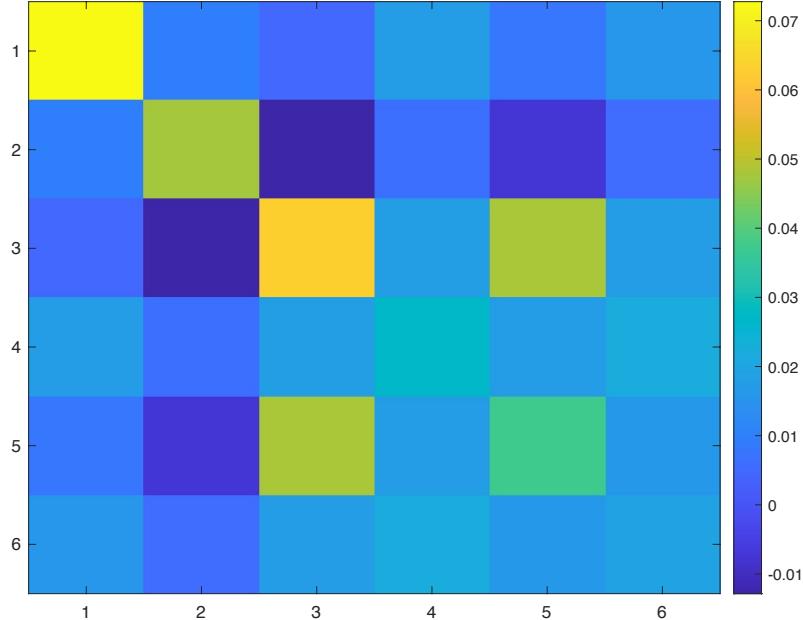
For the study, we consider the same parameters test as the ones used for behavior AS, and we compare the methods POD, POBD3, POBD6 and APOBD. The APOBD pre-analysis provides the matrix  $\mathbb{L}$  shown in Figure 4.20. It is easy to see that the



(a) First basis for the POD method for  $N = 100$  cells in FHB.

(b) First basis for each block of the APOBD method for  $N = 100$  cells in FHB.

**Figure 4.19:** Comparisons of the basis for each method obtained in the studied behavior. In all the cases, the first basis is in blue continuous fashion, the second in red dash-dotted fashion and the third basis in yellow dashed fashion. The latter only for the POD method. a) POD. b) APOBD.



**Figure 4.20:** Matrix  $\mathbb{L}$  arising after computing the mean correlation of each block of variables of system (25) for  $N = 100$  with  $c_\alpha = -0.5$  and  $c_\beta \in [-0.25, 0.25]$ .

elements of the diagonal are far from one, so the POBD methods and the APOBD

method, would not provide a faithful approximation. We checked that this is also the case for the POD method.

Table 4.12 shows the computational data obtained for the reductions. We can clearly see that the number of equations of the reduced models are higher than the ones obtained for the behaviors studied in the previous, as well no speed-up is obtained and the errors are not assumable.

	POD	POBD3	POBD6	APOBD
Number of equations	109	$73 + 33 + 31$	$37 + 36 + 19 + 16 + 17 + 15$	109
Sparsity index	0	0.6797	0.7676	0
Speed-ups	$< 1$	$< 1$	$< 1$	$< 1$
Relative error	0.96	0.93	0.90	0.96

Table 4.12: Comparison of the results obtained in sparsity index (left), speed-up (center), and relative error (right), considering system (25) for  $N = 100$  with  $c_\alpha = -0.5$  and  $c_\beta \in [-0.25, 0.25]$ .

Figure 4.21 shows a comparison of the solutions obtained for the methods presented above. It is clear that the reduced solutions do not provide a good approximation of the behavior studied, and that the reduced behavior is very erratic.

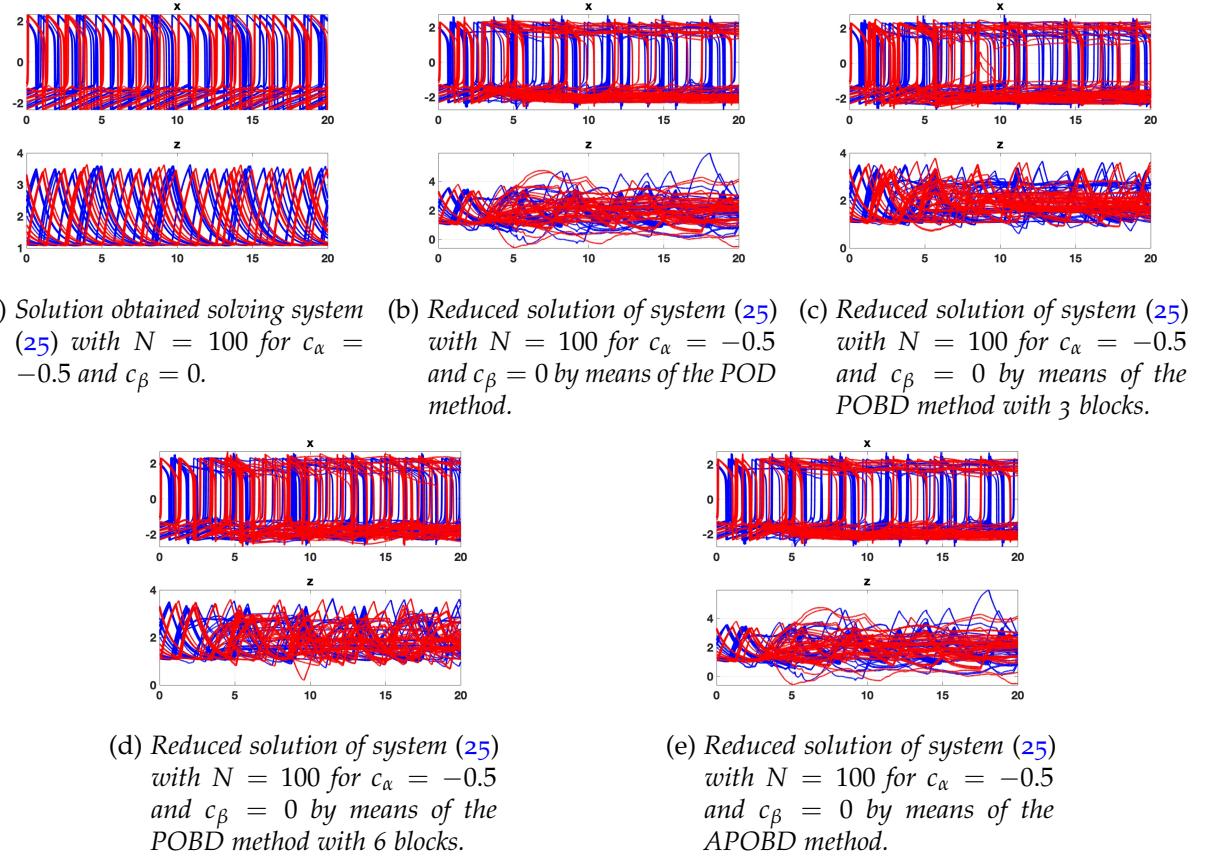
This loss in accuracy could be due to the fact that the original system (25) exhibits chaotic-like behavior for this range of parameters, where each cell is independent of the others and its behavior cannot be expressed as a linear combination of the behaviors of the other cells for the whole length of the integrating range.

### 4.3.2 Limitations in the case of nonlinear coupling

Another limitation could be faced if the original model exhibits too many nonlinearities. In that case, the EIM could not be able to reproduce all the variability of the nonlinear terms, and we should need to go back to the original space to compute the nonlinearities.

In the following example, we present a modified version of model (23) with nonlinear coupling:

$$O_i \begin{cases} \dot{x}_i = \tau(-y_i + f(x_i) - \phi_f(z_i)), \\ \dot{y}_i = \tau\epsilon k_i(a_0 x_i + a_1 y_i + a_2 + \frac{1}{N/2} \sum_{j=1}^N c_{ij} \sigma(x_i - x_j)), \\ \dot{z}_i = \tau\epsilon \left( \phi_r(x_i) - \frac{z_i - z_b}{\tau_z} \right), \end{cases} \quad (31)$$



**Figure 4.21:** Representation of the solutions obtained for the original system and the reduced ones. a) Original, b) POD, c) POBD with 3 blocks, d) POBD with 6 blocks and e) APOBD.

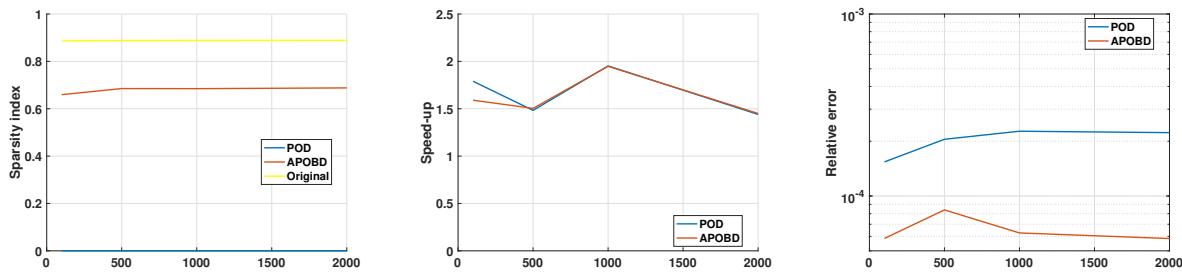
with the same parameters as for (23), and  $\sigma(x) = \frac{2}{1 + \exp(-2x)} - 1$ .

The selection of the nonlinear coupling has been made in order to keep qualitatively similar behaviors as for the original model (23). In particular, we will be focusing on the AiP behavior with  $c_\beta = -0.25$ .

As done for the previous test cases, we present in Table 4.13 the number of equations for the reduced problem obtained with each method. Furthermore, in Figure 4.22, we represent the evolution of sparsity, speed-up and error, respectively.

Number of cells	POD	APOBD
100	47	$19 + 16 + 13 + 12$
500	55	$21 + 22 + 13 + 13$
1000	58	$23 + 23 + 14 + 14$
2000	60	$23 + 24 + 14 + 14$

Table 4.13: AiP behavior with nonlinear coupling. Comparison of the number of equations of the reduced systems for the original system (31).



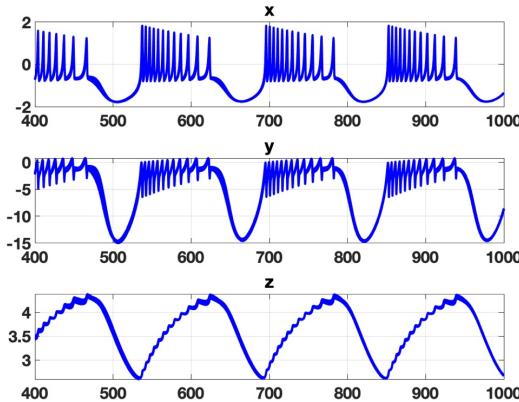
**Figure 4.22:** AiP behavior with nonlinear coupling. Comparisons of the results obtained for each method for model (31). The left panel shows the sparsity index, the central panel shows the speed-ups, and the right panel shows the relative error.

We can see that although the fact that we obtain better results in terms of the sparsity index and relative error in comparison with the POD method, the speed-up is similar to the POD but very low in comparison with the previous tests. This is due to the bottleneck of computing the nonlinear coupling in the original space.

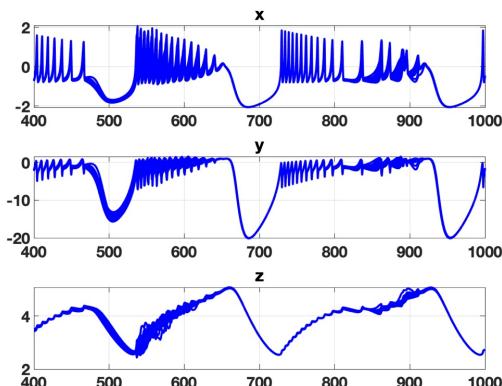
#### 4.3.3 Close to chaotic behavior

In the case of the Hindmarsh-Rose Network model, we have a limitation when trying to reduce the system in the Square-Wave Bursting (SW) behavior ( $I = 3, b = 2.6, \varepsilon = 0.01$ ), see Figure 4.23a. In this type of bursters, the stable equilibrium corresponding to the resting state disappears through a fold bifurcation and the limit cycle attractor corresponding to the spiking state disappears through a homoclinic bifurcation. The presence of this homoclinic bifurcation makes the system to be near to chaotic behavior.

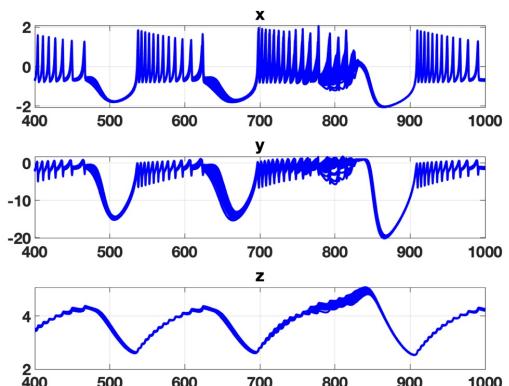
In this case, the reduced models are not able to truly reproduce the behavior of the original system (26) as they exhibit some erratic behavior after a certain amount of time, see figures 4.23b and 4.23c, due to the presence of the homoclinic bifurcation.



(a) Solution for the original system (27) for  $N = 500$  cells in SW behavior.



(b) Solution for the POD reduced system (27) for  $N = 500$  cells in SW behavior.



(c) Solution for the APOBD reduced system (27) for  $N = 500$  cells in SW behavior.

**Figure 4.23:** Comparisons of the solutions obtained in the SW behavior. a) Original. b) POD. c) APOBD with 3 blocks.



---

## CONCLUSIONS AND OPEN PROBLEMS

---

In this part, we have developed new ROM techniques based on the POD method (Section 1.3.1) for systems that present unknowns with different behaviors, for instance, dynamical systems with multiple timescales. These new techniques have been proven to maintain the structure of the original model, while producing a competitive speed-up. Furthermore, we have presented an *a priori* analysis over the variables of the original model, that could be used in order to group them into blocks. Then, we can apply a data-driven analysis based on the correlation between these group of variables in order to reduce the number of blocks to consider, while maintaining the most part of the structure of the original problem.

In the field of applications, we have tested these methods for several biological neural models with multiple timescales that exhibit a rich variety of different qualitative behaviors. These new methods have been proven to produce an accurate approximation of the original solution in much less computational time with respect to solving the original model. The obtained results in speed-up and error are better, or at least, as good as the ones obtained if we build the reduced model with the POD procedure. These new methods also provide a new feature, they keep the structure of the original model. This feature could be useful as we can see the different timescales in the new basis of the reduced model, and could allow the exploitation of this separation of the timescales. Furthermore, we have studied the evolution of the performance of these methods augmenting the number of cells in the original models. The new methods seem to perform better the more cells are considered. We have also noticed that these new techniques seem to perform better when there are more fast variables than slow variables, and where there is a switch between the fast and slow behaviors.

Therefore, the proposed techniques based on the POD method and an *a priori* study of the variables have been proven to be a useful tool for the numerical study of network models. We have shown that they are able to reproduce complex behaviors that these models may exhibit.

There are different ways to continue the research presented in this part. We present now some open problems related to them:

- Extension of the presented techniques considering various biological parameters, for example, the coupling and the heterogeneity parameters in ICC model (23).
- Use these reduced order models to perform a numerical analysis of the distribution of the behaviors of the models presented with respect to different biological parameters.
- Study the reduction of the presented models in the transition between different behaviors.
- Application of these techniques to different problems arising from other fields, like physical problems and computational fluid dynamics problems.

The research presented in this part of the thesis is an updated version of the research paper Bandera Moreno et al. (2022). And we also foresee the publication of another research paper Bandera Moreno et al. (2024c).

## Part II

### REDUCED BASIS MODELLING OF TURBULENCE WITH WELL-DEVELOPED INERTIAL RANGE



---

## INTRODUCTION

---

In this part, our emphasis is the study and simulation of turbulent flows, modelled by Partial Differential Equations (PDEs). In particular, we focus on the Reduced Basis (RB) method, with a specific goal of addressing one of the limitations that it presents, namely, the difficulty of obtaining an *a posteriori* error estimator by means of mathematics-based theories, and its dependence on the numerical discretization. In particular, we study the development of an *a posteriori* error estimator based upon the Kolmogórov's energy cascade theory (Kolmogórov, 1941). This part aligns with the Spanish national projects ROAD and HYROM, and the European project ARIA. The research presented in this part of the thesis is an updated version of the results published in the research paper Bandera Moreno et al. (2024a).

In order to evaluate the errors between the original solution and the reduced one, the *a posteriori* error estimates provide a powerful tool to compute estimations of this magnitude in terms of the energy carried by the neglected modes. This technique has been extensively applied to the solution of incompressible flows with success (Busto et al., 2020; Hijazi et al., 2023; Kunisch & Volkwein, 2002; Negri et al., 2015; Stabile & Rozza, 2018).

Yet, a joint Proper Orthogonal Decomposition (POD)-Galerkin procedure encounters a significant challenge. To achieve errors below a prescribed level, computing the solution of the original model for multiple parameter values becomes imperative. However, in scenarios concerning incompressible flows, this computation might quickly escalate to an unmanageable scale. Hence, a procedure was developed in order to obtain errors below a certain threshold while minimizing the number of needed solutions of the original model.

This issue is addressed by the certified RB method, in which the snapshots are recursively computed by a Greedy Algorithm to progressively reduce the errors. This procedure needs a fast *a posteriori* error estimation (Grepl et al., 2007; Hesthaven et al., 2016; Quarteroni et al., 2016). It has been applied to incompressible flows with increasing complexity: Stokes, Navier-Stokes and multi-parametric flows (Ali et al., 2020; Ballarin et al., 2015; Deparis, 2008; Deparis & Rozza, 2009; Manzoni, 2014). Moreover, some error indicators based on the dual norm of the discrete time-averaged

residual have been used, for example in Fick et al. (2018) and in Tsai and Fischer (2022), for the parameter selection in the Greedy Algorithm.

Kolmogorov's energy cascade theory (also known as K41 theory) is a fundamental concept in the field of fluid mechanics (Chacón Rebollo & Lewandowski, 2014; Kolmogórov, 1941; Richardson, 1922), and it provides a theoretical framework for understanding the energy transfer between different scales in a turbulent flow. In a fully developed turbulent flow, the kinetic energy is transferred from the large-scale eddies to smaller and smaller scales through a series of nonlinear interactions until it reaches the smallest scales, where it is dissipated into heat. This process is known as the energy cascade, and it is characterized by a self-similar scaling behavior in the inertial subrange of scales. Kolmogorov's theory provides a statistical description of turbulence that has been widely used to guide the development of turbulence models and numerical simulations of turbulent flows.

The energy cascade concept has lead to the use of the high-order modes of the POD expansion (see Section 1.3.1) to model the small scales within the inertial range. However, a very large number of modes is needed to achieve a good precision, making the Reduced Order Model (ROM) need computing times similar to a direct numerical solver (Hijazi et al., 2020).

An alternative strategy is to build ROMs for well-established turbulence models, actually here we address Large Eddy Simulation (LES) models. In this approach no turbulence modelling procedure by a ROM is intended, instead the purpose is to build surrogate LES models that can be run in highly reduced computing times. Certified RB LES Smagorinsky models for steady flows have been built with high computing speed-ups for single and multi-parametric configurations (Ballarin et al., 2020; Caravaca García, 2022; Chacón Rebollo et al., 2023; Chacón Rebollo et al., 2017). The error certification is based upon *a posteriori* error estimation procedures, that in their turn are built by the Brezzi-Rappaz-Raviart theory of approximation of regular branches of non-linear PDEs (Brezzi et al., 1980), thanks to the enhanced regularity of the solutions of the Smagorinsky model.

However, this procedure faces severe limitations. Indeed, the building of *a posteriori* error indicators is closely linked to the actual turbulence model and the actual FOM used, as well as to the nature of the parameters (either physical or geometrical) to deal with in the ROM. Also, the extension of this error estimation procedure to transient flows requires very high computational costs, thus limiting the advantage of reduced modeling.

In this part, we tackle the construction of a RB method for general transient LES parametric turbulence models, regardless of the FOM used. The basic tool is an error indicator based upon the physics of turbulent flows, rather than on the mathematics used to analyze the models. This error indicator measures the deviation of the energy spectrum of a trial solution versus the theoretical  $k^{-5/3}$  spectrum predicted by the K41 theory of turbulence in statistical equilibrium.

In practice, this indicator leads to a stagnation of the Greedy Algorithm, that selects again a parameter already chosen. When stagnation occurs, we use a correction of this indicator, now measuring the deviation of the spectrum of the trial solution versus the spectrum of the reduced solution obtained in the current reduced basis. Indeed, the ROM solution is intended to approximate the FOM solution, whose spectrum already may have some error with respect to the theoretical K41 spectrum.

To validate this indicator, we address some academic tests for the Smagorinsky turbulence model. We consider 2D periodic flows, yet meaningful, as they are designed to present the inertial spectrum predicted by the K41 theory. We assume that the Smagorinsky flow already has a part of its spectrum within the inertial range. The Empirical Interpolation Method (EIM) is used to compute reduced approximations of the non-linear eddy viscosity term, see Section 1.4.1. We use various different POD filtering strategies to reduce the time snapshots. The Fast Fourier Transform (FFT) is used to compute the spectrum of the trial solutions. We compare the errors obtained with this strategy (using the error indicator) versus those obtained with the best error indicator, that is, the exact error. As a result, we obtain errors between the ROM and the FOM solutions quite close to the optimal ones. Furthermore, we obtain a spectral decay of the error as the dimension of the reduced basis space increases, quite close to the one obtained if the exact error is used as indicator. We obtain speed-up rates of computing times of nearly 23.

This part is structured as follows. In Chapter 5, we introduce an error indicator based upon the K41 theory and state the combined Greedy Algorithm-POD strategy to build the reduced basis from this indicator. We also present in this chapter some error bounds estimates for Kolmogórov's indicator. The numerical tests are presented in Chapter 6. In particular, we address some academic test for the Smagorinsky turbulence model.



---

## REDUCED BASIS MODELLING OF TURBULENT FLOWS BASED ON THE STATISTICAL TURBULENCE THEORY

---

The main purpose behind this chapter is the introduction of an *a posteriori* error indicator for turbulent flows. This *a posteriori* error indicator will allow us to select the parameter in the Greedy procedure for the construction of the reduced basis model.

For the development of this indicator, we have used general results from the statistical turbulence theory, attributed to Kolmogórov (Kolmogórov, 1941), also known as K41 theory. This theory presents an expression for the energy cascade (Chacón Rebollo & Lewandowski, 2014; Richardson, 1922) and the main assumption behind this indicator is that the difference between the energy spectrum of a certain solution and the theoretical spectrum predicted by this theory is small for accurate solutions.

The structure of the Chapter is as follows. In Section 5.1, we provide error estimates based on the differences of the energy spectrum of two velocity fields, and we introduce an error indicator based upon the K41 theory. On the other hand, in Section 5.2, we state the combined Greedy Algorithm-POD strategy to build the reduced basis from this indicator.

### 5.1 A POSTERIORI ERROR INDICATOR BASED ON THE STATISTICAL TURBULENCE THEORY

Andrei Kolmogórov stated in Kolmogórov (1941) that under suitable similarity and isotropy assumptions for turbulent flow in statistical equilibrium, there exists an inertial range  $[k_1, k_2]$ , in which, the physical magnitude, called energy spectrum  $E(k)$ , admits an expression in terms of the turbulent dissipation  $\varepsilon$ , that is, an indicator of the energy dissipated through viscosity, and the wavenumber  $k$ .

### 5.1.1 A brief introduction to the Kolmogórov $-5/3$ law

One of the most notorious features in turbulence flow is the variability of eddies induced by the fluid motion. Each eddy could be classified by its size  $r$ , and also by its wavenumber  $k = 1/r$ . In this section, we aim to express the energy in terms of the turbulent dissipation and the wavenumber. Furthermore, this will allow us to express the energy spectrum in their terms and present the energy cascade.

In the following, we assume that the flow hypotheses of similarity, that is, invariance under changes of scale, and isotropy, invariance under rotations, are fulfilled. We name  $\mathbf{k} = k\mathbf{n}$ , the wavevector, with  $\mathbf{n}$  a unit vector in  $\mathbb{Z}^d$ .

We present now the so-called energy spectrum, a function  $E(k)$  for  $k \in (0, +\infty)$  that express the energy associated to the eddies of wavenumber  $k$ . With a slight abuse of notation, it can be written as

$$E(k) = \sum_{k=|\mathbf{k}|} E(\mathbf{k}),$$

where  $E(\mathbf{k})$  represents the kinetic energy part on the Fourier mode  $\mathbf{k}$ .

Furthermore, we assume that the total kinetic energy  $E_{kin}$  can be written as

$$E_{kin} = \int_0^{+\infty} E(k) dk, \quad (32)$$

that is, the energy spectrum allows to express the total kinetic energy as an integral over the wavenumber.

Andréi Kolmogórov stated in 1941 (Kolmogórov, 1941) that under the hypothesis of similarity and isotropy there exists an inertial range  $[k_1, k_2]$  where the energy spectrum can be expressed by the wavenumber  $k$  and the turbulent dissipation  $\varepsilon$ , and where the latter is constant. Therefore, the inertial range is determined by two wavenumbers,  $k_1$ , associated to the largest scale of the problem, and  $k_2$ , associated to the smaller scale under which the viscosity plays an important role.

From these assumptions, we will deduce the expression for the energy spectrum in the inertial range by means of the dimensional analysis. In order to do so, we will denote by  $L$  the spatial dimension and by  $T$  the temporal dimension. We know the units of the kinetic energy  $[E_{kin}] = L^2/T^2$  and the wavenumber  $[k] = L^{-1}$ . Then, thanks to equation (32),  $[E] = L^3/T^2$ . We also know the dimensions of the turbulent dissipation  $[\varepsilon] = L^2/T^3$ .

Finally, from Kolmogórov's theory, we can express the energy spectrum as a function of the wavenumber and the turbulent dissipation as  $E(k) \sim \varepsilon^\alpha k^\beta$ . Thanks to dimensional analysis,

$$\frac{L^3}{T^2} = \left(\frac{L^2}{T^3}\right)^\alpha \left(\frac{1}{L}\right)^\beta \iff \begin{cases} 3 = 2\alpha - \beta \\ 2 = 3\alpha \end{cases} \iff \begin{cases} \alpha = 2/3 \\ \beta = -5/3 \end{cases}$$

So, we finally obtain that  $E(k) \sim \varepsilon^{2/3} k^{-5/3}$  in the inertial range where  $\varepsilon$  is constant, and that is why it is also known as Kolmogórov's  $-5/3$  law.

### 5.1.2 Error estimates based on the energy spectrum

In this section, we offer a series of arguments that seek to substantiate, based on physical and mathematical principles, the ability of energy-spectrum-based estimators to yield accurate Reduced Order Model (ROM) solutions for the Smagorinsky turbulence models. In the following, we assume that  $\mathbf{u}$  is the average field in time of the velocity, and we drop the dependence with respect to  $x, t$  and  $Re$  for the sake of clarity.

In Chacón Rebollo and Lewandowski (2014, Chapter 5), it is proved that the turbulent dissipation  $\varepsilon$  satisfies

$$\varepsilon = \frac{1}{\gamma} \int_0^\infty k^2 E(k) dk, \quad |D\mathbf{u}|^2 = \frac{\gamma}{2} \varepsilon,$$

where  $D\mathbf{u}$  denotes the symmetric gradient, that is,  $D\mathbf{u} = 1/2(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$ .

From the previous equation, we can also assume that the following relationship holds for the resolved (large scale) velocity, that we also note by  $\mathbf{u}$ .

$$|D\mathbf{u}|^2 = \frac{1}{2} \int_{k_1}^{k_2} k^2 E(k) dk, \tag{33}$$

where  $[k_1, k_2]$  is the inertial spectrum. In practice, we substitute  $k_2$  by  $k_c < k_2$ , where  $k_c$  is the smallest scale we can solve numerically by means of the turbulence model considered. In most cases,  $k_c$  is related to the mesh size  $\delta$ , that is, there exist two non-negative constants  $c < C$  such that  $ch^{-1} \leq k_c \leq Ch^{-1}$ .

In Section 2.2, equation (13), we have stated that the eddy viscosity can be expressed in terms of the used triangulation and the gradient of the resolved velocity  $\mathbf{u}$ . By Chacón Rebollo and Lewandowski (2014, Chapter 5), we have the following relationship

$$\nu_T(\mathbf{u}) = Ch^2 |D\mathbf{u}|, \tag{34}$$

where we have dropped the dependence of  $\mathbf{u}$  with respect to space and time.

In the following proposition, we give an upper bound for the difference of the turbulent viscosities of two velocity fields of locally isotropic flows.

### Proposition 5.1

Assuming (33) holds, if  $\mathbf{u}_1$  and  $\mathbf{u}_2$  are the velocity fields of two locally isotropic flows, denoting by  $E(k; \mathbf{u}_i)$  the energy spectrum corresponding to the velocity field  $\mathbf{u}_i$ , it follows that

$$|\nu_T(\mathbf{u}_1) - \nu_T(\mathbf{u}_2)| \leq Ch^2 \left( \int_{k_1}^{k_c} k^2 |E(k; \mathbf{u}_1) - E(k; \mathbf{u}_2)| dk \right)^{1/2}. \quad (35)$$

Furthermore, if  $|E(k; \mathbf{u}_1) - E(k; \mathbf{u}_2)| \leq \delta, \forall k \in [k_1, k_c]$ , then

$$|\nu_T(\mathbf{u}_1) - \nu_T(\mathbf{u}_2)| \leq C(\delta h)^{1/2}. \quad (36)$$

**Proof:** Let  $\mathbf{u}_1$  and  $\mathbf{u}_2$  be the velocity fields of two locally isotropic flows, and taking into account equations (33) and (34), then we obtain the following upper bound

$$|\nu_T(\mathbf{u}_1) - \nu_T(\mathbf{u}_2)| \leq Ch^2 \left| \left( \int_{k_1}^{k_c} k^2 E(k; \mathbf{u}_1) dk \right)^{1/2} - \left( \int_{k_1}^{k_c} k^2 E(k; \mathbf{u}_2) dk \right)^{1/2} \right|,$$

Now, noting that for two non-negative numbers  $a, b \geq 0$ , the following inequality holds,  $|a - b| \leq |a^2 - b^2|^{1/2}$ , and that  $\int_{k_1}^{k_c} k^2 E(k; \mathbf{u}_i) dk \geq 0, i = 1, 2$ , we have

$$|\nu_T(\mathbf{u}_1) - \nu_T(\mathbf{u}_2)| \leq Ch^2 \left| \int_{k_1}^{k_c} k^2 (E(k; \mathbf{u}_1) - E(k; \mathbf{u}_2)) dk \right|^{1/2}.$$

Now, we focus on the integral term,

$$\left| \int_{k_1}^{k_c} k^2 (E(k; \mathbf{u}_1) - E(k; \mathbf{u}_2)) dk \right| \leq \int_{k_1}^{k_c} k^2 |E(k; \mathbf{u}_1) - E(k; \mathbf{u}_2)| dk,$$

after applying the usual integral bound  $|\int f(k) dk| \leq \int |f(k)| dk$ , and that  $k > 0$ . This yields the first result of the proposition, estimate (35).

In order to obtain estimate (36), we let  $|E(k; \mathbf{u}_1) - E(k; \mathbf{u}_2)| \leq \delta, \forall k \in [k_1, k_c]$ , and the previous integral becomes

$$\int_{k_1}^{k_c} k^2 |E(k; \mathbf{u}_1) - E(k; \mathbf{u}_2)| dk \leq \delta \int_{k_1}^{k_c} k^2 dk = \frac{1}{3} \delta (k_c^3 - k_1^3) \leq \frac{1}{3} \delta k_c^3 \approx \frac{1}{3} \delta h^{-3},$$

where we have used that  $k_1 \geq 0$ , and that  $k_c \approx h^{-1}$ . Including this result in estimate (35), we obtain

$$|\nu_T(\mathbf{u}_1) - \nu_T(\mathbf{u}_2)| \leq Ch^2(\delta h^{-3})^{1/2} \leq C(h\delta)^{1/2},$$

and that is the desired second estimate (36).  $\blacksquare$

Thanks to Proposition 5.1, we can state the following theorems about the error bounds of the solutions of turbulent problems. We shall denote by  $\nu_e$  the "effective" viscosity for a velocity field  $\mathbf{u}$  as

$$\nu_e = \nu + (C_S h)^2 \inf_{\Omega \times [0,T]} |\nabla \mathbf{u}|.$$

In the sequel, we denote by  $C$  a constant that may vary from line to line, but that is always independent on  $\nu$ , the discretization parameters  $h$  and  $\Delta t$ . Moreover, we denote by  $\|\cdot\|_{n,p}$  the norm of the  $W^{n,p}(\Omega)$  space and, for simplicity, we denote  $\|\cdot\|_p = \|\cdot\|_{0,p}$ .

With base on Proposition 5.1, we obtain the following theorem that states an error bound between two velocity fields of locally isotropic flows, solution of the Smagorinsky turbulence model (12), that is just driven by the deviation between their energy spectra.

### Theorem 5.2

*Let  $\mathbf{u}_1, \mathbf{u}_2$  be two velocity fields of locally isotropic flows, solution of the Smagorinsky turbulence model (12). Assuming that  $|E(k; \mathbf{u}_1) - E(k; \mathbf{u}_2)| \leq \delta \forall k \in [k_1, k_c]$ , and that  $\mathbf{u}_1, \mathbf{u}_2 \in L^\infty(W^{1,3})$ , we obtain the following error bound*

$$\|\mathbf{r}(t)\|_2^2 + \nu_e \int_0^t \|\nabla \mathbf{r}(s)\|_2^2 ds \leq e^{\frac{C}{\nu_e} t} \|\mathbf{r}(0)\|_2^2 + \frac{C}{\nu_e} h \delta (e^{\frac{C}{\nu_e} t} - 1),$$

*where  $\mathbf{r} = \mathbf{u}_1 - \mathbf{u}_2$ ,  $\nu_e$  is the maximum effective viscosity between both velocity fields, and  $C$  is a constant independent of  $\nu_e$ .*

**Proof:** For the sake of clarity, we will be dropping the dependence of the velocity fields on space and time.

We consider the weak formulation arising from equation (12) for  $\mathbf{u}_1, \mathbf{u}_2$  and the same test function  $\mathbf{v}$  belonging to  $W^{1,3}(\Omega)$ . If we subtract one from the other, and denoting  $\mathbf{e} = \mathbf{u}_1 - \mathbf{u}_2$ , we get to

$$\begin{aligned} & (\partial_t \mathbf{e}, \mathbf{v})_\Omega + a(\mathbf{e}, \mathbf{v}; \gamma) + c(\mathbf{u}_1, \mathbf{u}_1, \mathbf{v}; \gamma) - c(\mathbf{u}_2, \mathbf{u}_2, \mathbf{v}; \gamma) \\ & + a_S(\mathbf{u}_1; \mathbf{u}_1, \mathbf{v}; \gamma) - a_S(\mathbf{u}_2; \mathbf{u}_2, \mathbf{v}; \gamma) = 0, \quad \forall \mathbf{v}. \end{aligned} \tag{37}$$

If we consider  $v = e$ , then we have the first and second term become

$$(\partial_t e, e)_\Omega + a(e, e; \gamma) = \frac{1}{2} \partial_t \|e\|_2^2 + \nu \|\nabla e\|_2^2.$$

Also, the third and forth term can be written as

$$c(u_1, u_1, e; \gamma) - c(u_2, u_2, e; \gamma) = c(e, u_1, e; \gamma) - c(u_2, e, e; \gamma) = c(e, u_1, e; \gamma),$$

where we have added and subtracted  $c(u_2, u_1, e; \gamma)$  and applied equality (16) of the convective term. Finally, the eddy viscosity terms can be transformed into

$$a_S(u_1; u_1, e; \gamma) - a_S(u_2; u_2, e; \gamma) = a_S(u_1; e, e; \gamma) + (a_S(u_1; u_2, e; \gamma) - a_S(u_2; u_2, e; \gamma)),$$

after adding and subtracting  $a_S(u_1; u_2, e; \gamma)$ .

Introducing these equivalences into equation (37) and rearranging the terms, we have

$$\frac{1}{2} \partial_t \|e\|_2^2 + \nu \|\nabla e\|_2^2 + a_S(u_1; e, e; \gamma) = -c(e, u_1, e; \gamma) - (a_S(u_1; u_2, e; \gamma) - a_S(u_2; u_2, e; \gamma)). \quad (38)$$

The second and third terms on the left-hand side of equation (38) can be bounded by the maximum effective viscosity as  $\nu \|\nabla e\|_2^2 + a_S(u_1; e, e; \gamma) \geq \nu_e \|\nabla e\|_2^2$ .

After taking absolute value on the right-hand side of equation (38), we finally obtain

$$\frac{1}{2} \partial_t \|e\|_2^2 + \nu_e \|\nabla e\|_2^2 \leq |c(e, u_1, e; \gamma)| + |a_S(u_1; u_2, e; \gamma) - a_S(u_2; u_2, e; \gamma)|. \quad (39)$$

Now, we focus on the terms of the right-hand side of equation (39). First, we study the convective term, if we use the continuous Sobolev's embedding  $H^1 = W^{1,2} \subset L^6$ , and Hölder's and Young's inequalities, we can obtain

$$|c(e, u_1, e; \gamma)| \leq C \|e\|_2 \|\nabla u_1\|_3 \|e\|_6 + C \|e\|_2 \|\nabla e\|_2 \|u_1\|_\infty \leq \frac{C\epsilon}{2} \|\nabla e\|_2^2 + \frac{C(u_1, u_2)}{2\epsilon} \|e\|_2^2, \quad (40)$$

where we have also used the regularity of  $u_i \in W^{1,3}(\Omega) \cap L^\infty(\Omega)$ .

Following a similar procedure for the eddy viscosity term, using Hölder's and Young's inequalities, we obtain

$$\begin{aligned} |a_S(u_1; u_2, e; \gamma) - a_S(u_2; u_2, e; \gamma)| &\leq C |\nu_T(u_1) - \nu_T(u_2)| \|\nabla u_2\|_2 \|\nabla e\|_2 \\ &\leq \frac{C(u_1, u_2)}{2\epsilon} |\nu_T(u_1) - \nu_T(u_2)|^2 + \frac{C\epsilon}{2} \|\nabla e\|_2^2. \end{aligned} \quad (41)$$

Taking equations (40) and (41) into account in equation (39), we arrive at

$$\frac{1}{2} \partial_t \|e\|_2^2 + (\nu_e - C\epsilon) \|\nabla e\|_2^2 \leq \frac{C(\mathbf{u}_1, \mathbf{u}_2)}{2\epsilon} (\|e\|_2^2 + |\nu_T(\mathbf{u}_1) - \nu_T(\mathbf{u}_2)|^2).$$

Now, we can select  $\epsilon = \nu_e / (2C)$  and multiply both terms by 2 to obtain

$$\partial_t \|e\|_2^2 + \nu_e \|\nabla e\|_2^2 \leq \frac{C(\mathbf{u}_1, \mathbf{u}_2)}{\nu_e} (\|e\|_2^2 + |\nu_T(\mathbf{u}_1) - \nu_T(\mathbf{u}_2)|^2) \leq \frac{C(\mathbf{u}_1, \mathbf{u}_2)}{\nu_e} (\|e\|_2^2 + \delta h),$$

where we have also used Proposition 5.1 in the last inequality.

Finally, using Grönwall's inequality in its differential form, in the standard way, we obtain the desired result.  $\blacksquare$

In the next theorem, we provide an estimation of the error between the solution of the FOM Smagorinsky model (15), that we name  $\mathbf{u}_h$ , and the solution of ROM Smagorinsky model (18), that we note as  $\mathbf{u}_N$ . Naming  $\mathbf{w}_N$ , the projection of  $\mathbf{u}_h$  into the reduced space, and  $e_N = \mathbf{w}_N - \mathbf{u}_N$ , the error between  $\mathbf{w}_N$ , and the reduced solution  $\mathbf{u}_N$ , the theorem can be stated as

### Theorem 5.3

Let  $\mathbf{u}_h$  be the solution of the FOM Smagorinsky model (15) and  $\mathbf{u}_N$  the solution of ROM Smagorinsky model (18). Assume that  $|E(k; \mathbf{u}_h) - E(k; \mathbf{u}_N)| \leq \delta \forall k \in [k_1, k_c]$ . Then, the following error estimate holds

$$\begin{aligned} \|e_N(t)\|_2^2 + \nu_e \int_0^t \|\nabla e_N(s)\|_2^2 ds &\leq \\ \|e_N(0)\|_2^2 \exp(\sigma(t)) + \int_0^t P_N(s) \exp(\sigma(t) - \sigma(s)) ds & \\ + \frac{C}{\nu_e} h \delta \int_0^t \|\nabla \mathbf{u}_h(s)\|_2^2 \exp(\sigma(t) - \sigma(s)) ds, \end{aligned}$$

with

$$P_N = C \|\partial_t \boldsymbol{\rho}_N\|_2^2 + \frac{C}{\nu_e} \left( \nu^2 + \|\nabla \mathbf{u}_h\|_2^2 + \|\nabla \mathbf{w}_N\|_2^2 \right) \|\nabla \boldsymbol{\rho}_N\|_2^2 + \frac{C}{\nu_e} \|\nu_T(\mathbf{u}_h) \nabla \boldsymbol{\rho}_N\|_2^2,$$

where we have noted  $\boldsymbol{\rho}_N = \mathbf{w}_N - \mathbf{u}_h$ , and

$$\sigma_N(t) = C \left( t + \frac{1}{\nu_e} \left( \|\nabla \mathbf{w}_N(t)\|_{L^2(L^3)}^2 + \|\mathbf{w}_N(t)\|_{L^2(L^\infty)}^2 \right) \right).$$

**Proof:** We start from equation (15), taking  $q_h = p_h$ , we add and subtract  $(\partial_t \mathbf{w}_N, \mathbf{v}_h)_\Omega + \nu(\nabla \mathbf{w}_N, \nabla \mathbf{v}_h)$ . Naming  $\rho_N = \mathbf{w}_N - \mathbf{u}_h$ , we obtain the following

$$\begin{aligned} & (\partial_t \mathbf{w}_N, \mathbf{v}_h)_\Omega + a(\mathbf{w}_N, \mathbf{v}_h; \mu) + a_S(\mathbf{u}_h; \mathbf{u}_h, \mathbf{v}_h; \mu) + c(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h; \mu) \\ &= \langle \mathbf{f}, \mathbf{v}_h \rangle + (\partial_t \rho_N, \mathbf{v}_h)_\Omega + a(\rho_N, \mathbf{v}_h; \mu), \quad \forall \mathbf{v}_h \in X_h. \end{aligned}$$

Now, we consider  $\mathbf{v}_h = \mathbf{v}_N \in X_N$ , and we subtract equation (18) considering  $q_N = p_N$ . If we name  $\mathbf{e}_N = \mathbf{w}_N - \mathbf{u}_N$ , then, we obtain

$$\begin{aligned} & (\partial_t \mathbf{e}_N, \mathbf{v}_N)_\Omega + a(\mathbf{e}_N, \mathbf{v}_N; \mu) + (a_S(\mathbf{u}_h; \mathbf{u}_h, \mathbf{v}_N; \mu) - a_S(\mathbf{u}_N; \mathbf{u}_N, \mathbf{v}_N; \mu)) \\ &+ (c(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_N; \mu) - c(\mathbf{u}_N, \mathbf{u}_N, \mathbf{v}_N; \mu)) = (\partial_t \rho_N, \mathbf{v}_N)_\Omega + a(\rho_N, \mathbf{v}_N; \mu). \end{aligned}$$

Considering  $\mathbf{v}_N = \mathbf{e}_N$  and  $\nu = 1/\mu$ , and following the same ideas as in the first part of the proof of Theorem 5.2 (see computation of equation (38)), we get to

$$\begin{aligned} & \frac{1}{2} \partial_t \|\mathbf{e}_N\|^2 + \nu \|\nabla \mathbf{e}_N\|_2^2 + a_S(\mathbf{u}_h; \mathbf{e}_N, \mathbf{e}_N; \mu) \\ &= (\partial_t \rho_N, \mathbf{e}_N)_\Omega + a(\rho_N, \mathbf{e}_N; \mu) + a_S(\mathbf{u}_h; \rho_N, \mathbf{e}_N; \mu) \\ &+ c(\mathbf{u}_h, \rho_N, \mathbf{e}_N; \mu) + c(\rho_N, \mathbf{w}_N, \mathbf{e}_N; \mu) - c(\mathbf{e}_N, \mathbf{w}_N, \mathbf{e}_N; \mu) \\ &+ (a_S(\mathbf{u}_h; \mathbf{u}_h, \mathbf{e}_N; \mu) - a_S(\mathbf{u}_N; \mathbf{u}_h, \mathbf{e}_N; \mu)). \end{aligned}$$

As done for equation (38) in the proof of Theorem 5.2, we can use the effective viscosity in order to bound the last two terms on the left-hand side.

After taking absolute value on the right-hand side and applying the triangular inequality, we obtain the following bound

$$\begin{aligned} \frac{1}{2} \partial_t \|\mathbf{e}_N\|^2 + \nu_e \|\nabla \mathbf{e}_N\|_2^2 &\leq |(\partial_t \rho_N, \mathbf{e}_N)_\Omega + a(\rho_N, \mathbf{e}_N; \mu) + a_S(\mathbf{u}_h; \rho_N, \mathbf{e}_N; \mu)| \\ &+ |c(\mathbf{u}_h, \rho_N, \mathbf{e}_N; \mu) + c(\rho_N, \mathbf{w}_N, \mathbf{e}_N; \mu) - c(\mathbf{e}_N, \mathbf{w}_N, \mathbf{e}_N; \mu)| \\ &+ |a_S(\mathbf{u}_h; \mathbf{u}_h, \mathbf{e}_N; \mu) - a_S(\mathbf{u}_N; \mathbf{u}_h, \mathbf{e}_N; \mu)|. \end{aligned} \tag{42}$$

The first term on the right-hand side can be upper bounded using Hölder's and Young's inequalities as follows

$$\begin{aligned}
& |(\partial_t \rho_N, e_N)_\Omega + a(\rho_N, e_N; \mu) + a_S(u_h; \rho_N, e_N; \mu)| \\
& \leq \|\partial_t \rho_N\|_2 \|e_N\|_2 + \nu \|\nabla \rho_N\|_2 \|\nabla e_N\|_2 + \|\nu_T(u_h) \nabla \rho_N\|_2 \|\nabla e_N\|_2 \\
& \leq C \|\partial_t \rho_N\|_2^2 + C \|e_N\|_2^2 + \frac{\nu}{2\epsilon_*} \|\nabla \rho_N\|_2^2 + \frac{\nu \epsilon_*}{2} \|\nabla e_N\|_2^2 + \frac{C}{2\epsilon} \|\nu_T(u_h) \nabla \rho_N\|_2^2 + \frac{C\epsilon}{2} \|\nabla e_N\|_2^2.
\end{aligned} \tag{43}$$

For the second term on the right-hand side, we need to take into account expression (17), and Hölder's and Young's inequalities and the Sobolev's embedding  $H^1 = W^{1,2} \subset L^6$ , as we did on equation (40), to obtain

$$\begin{aligned}
& |c(u_h, \rho_N, e_N; \mu) + c(\rho_N, w_N, e_N; \mu) - c(e_N, w_N, e_N; \mu)| \\
& \leq \|\nabla u_h\|_2 \|\nabla \rho_N\|_2 \|\nabla e_N\|_2 + \|\nabla w_N\|_2 \|\nabla \rho_N\|_2 \|\nabla e_N\|_2 \\
& \quad + \|\nabla w_N\|_3 \|\nabla e_N\|_6 \|\nabla e_N\|_2^2 + \|w_N\|_\infty \|e_N\|_2 \|\nabla e_N\|_2^2 \\
& \leq \frac{C}{2\epsilon} \|\nabla u_h\|_2^2 \|\nabla \rho_N\|_2^2 + \frac{C\epsilon}{2} \|\nabla e_N\|_2^2 + \frac{C}{2\epsilon} \|\nabla w_N\|_2^2 \|\nabla \rho_N\|_2^2 + \frac{C\epsilon}{2} \|\nabla e_N\|_2^2 \\
& \quad + \frac{C}{2\epsilon} (\|\nabla w_N\|_3^2 + \|w_N\|_\infty^2) \|e_N\|_2^2 + \frac{C\epsilon}{2} \|\nabla e_N\|_2^2.
\end{aligned} \tag{44}$$

In order to bound the last term on the right-hand side, we simply need to apply Hölder's and Young's inequalities to get to

$$|(a_S(u_h, u_h, e_N; \mu) - a_S(u_N, u_h, e_N; \mu))| \leq \frac{C}{2\epsilon} \|\nu_T(u_h) - \nu_T(u_N)\|_\infty^2 \|\nabla u_h\|_2^2 + \frac{C\epsilon}{2} \|\nabla e_N\|_2^2. \tag{45}$$

Applying bounds (43)-(45) in equation (42), and arranging the terms we get to the following bound

$$\begin{aligned}
& \frac{1}{2} \partial_t \|e_N\|^2 + \left( \nu_e - \frac{\nu \epsilon_*}{2} - \frac{5C\epsilon}{2} \right) \|\nabla e_N\|_2^2 \leq C \left( 1 + \frac{C}{2\epsilon} (\|\nabla w_N\|_3^2 + \|w_N\|_\infty^2) \right) \|e_N\|_2^2 \\
& \quad + C \|\partial_t \rho_N\|_2^2 + \left( \frac{\nu}{2\epsilon_*} + \frac{C}{2\epsilon} (\|\nabla u_h\|_2^2 + \|\nabla w_N\|_2^2) \right) \|\nabla \rho_N\|_2^2 + \frac{C}{2\epsilon} \|\nu_T(u_h) \nabla \rho_N\|_2^2 \\
& \quad + \frac{C}{2\epsilon} \|\nu_T(u_h) - \nu_T(u_N)\|_\infty^2 \|\nabla u_h\|_2^2.
\end{aligned}$$

Selecting  $\epsilon_* = \nu_e/(2\nu)$  and  $\epsilon = \nu_e/(10C)$  in the previous equation, and multiplying both terms by 2, we obtain

$$\begin{aligned} \partial_t \|\boldsymbol{e}_N\|^2 + \nu_e \|\nabla \boldsymbol{e}_N\|_2^2 &\leq C \left( 1 + \frac{\|\nabla \boldsymbol{w}_N\|_3^2 + \|\boldsymbol{w}_N\|_\infty^2}{\nu_e} \right) \|\boldsymbol{e}_N\|_2^2 \\ &+ C \|\partial_t \boldsymbol{\rho}_N\|_2^2 + \frac{C}{\nu_e} (\nu^2 + \|\nabla \boldsymbol{u}_h\|_2^2 + \|\nabla \boldsymbol{w}_N\|_2^2) \|\nabla \boldsymbol{\rho}_N\|_2^2 + \frac{C}{\nu_e} \|\nu_T(\boldsymbol{u}_h) \nabla \boldsymbol{\rho}_N\|_2^2 \\ &+ \frac{C}{\nu_e} (\delta h) \|\nabla \boldsymbol{u}_h\|_2^2, \end{aligned}$$

where we have used Proposition 5.1 in the last term, and all the constants are included in  $C$ . Applying Grönwall's inequality, we get to desired estimate. ■

This supports the use of an energy spectrum-based error estimator in the building of a reduced basis model for the Smagorinsky model.

### 5.1.3 Kolmogórov's *a posteriori* error indicator

In the section 5.1.1, we have seen that the energy spectrum admits a separate expression in terms of the wavenumber  $k$ . In this section, we will exploit that property in order to build an *a posteriori* error indicator for turbulent flows.

Let  $\boldsymbol{u}(\boldsymbol{x})$  be a velocity field on a certain domain  $\Omega \in \mathbb{R}^d$ , with  $d = 2, 3$ . Thanks to the Fourier transform (Fourier, 1888; Stein & Weiss, 1971), we are able to decompose the velocity field in its Fourier modes

$$\hat{\boldsymbol{u}}(\boldsymbol{k}) = \int_{\Omega} \boldsymbol{u}(\boldsymbol{x}) \bar{\phi}_{\boldsymbol{k}}(\boldsymbol{x}) d\boldsymbol{x},$$

where  $\bar{\phi}_{\boldsymbol{k}} = \exp(-2\pi i \boldsymbol{k} \cdot \boldsymbol{x})$ . Furthermore, we can recover the velocity field from its Fourier modes thanks to the inverse Fourier transform,

$$\boldsymbol{u}(\boldsymbol{x}) = \int_{\mathbb{R}^d} \hat{\boldsymbol{u}}(\boldsymbol{k}) \phi_{\boldsymbol{k}}(\boldsymbol{x}) d\boldsymbol{k},$$

with  $\phi_{\boldsymbol{k}}$  the complex conjugate of  $\bar{\phi}_{\boldsymbol{k}}$ .

Now, assuming that the velocity field is ergodic and regular enough, that is  $u(x) \in L^2(\Omega)$ , we can apply Parseval's identity to obtain an expression of the kinetic energy with respect to its Fourier modes

$$E_{kin} = \int_{\mathbb{R}^d} |\hat{u}(\mathbf{k})|^2 d\mathbf{k}.$$

Therefore, in the applications we can approximate the total kinetic energy by

$$E_{kin} \approx \sum_{k=0}^L E(k) = \sum_{k=0}^L \sum_{k \leq |\mathbf{k}| < k+1} |\hat{u}(\mathbf{k})|^2 d\mathbf{k},$$

where  $L$  is related to the maximum wavenumber that can be solved numerically, and  $\hat{u}(\mathbf{k})$  can be computed by means of the discrete Fourier transform.

For a high fidelity solution, we can assume that, in the inertial subrange, this physical property accurately follows the expression

$$E(t, k; \gamma) = \alpha(t; \gamma) k^{-5/3}, \quad (46)$$

where  $\alpha(t; \gamma) > 0$  is a positive function depending on time and parameters  $\gamma$ . One of those parameters can be the turbulent dissipation  $\varepsilon$  of the solution for the Reynolds number,  $Re$ . For the sake of clarity, we will suppose that the only parameter acting on  $\alpha$  is the Reynolds number, that is,  $\gamma = Re$ , so we can write the energy spectrum as  $E_N(t, k; Re)$ .

It is fair to assume that the spectrum of the solution of the original (full-order) model closely follows (46), as the full-order model is assumed to approximate the continuous problem faithfully in the resolved part of the inertial spectrum.

Let us denote  $E_N(t, k; Re)$  be the energy spectrum associated to the reduced solution that belongs to a reduced space of dimension  $N$  for a certain  $Re$  and at a certain time  $t$ . We define the *a posteriori* error indicator as follows,

$$\Delta_N(t; Re) = \min_{\alpha \in \mathbb{R}} \left( \int_{k_1}^{k_2} |E_N(t, k; Re) - \alpha(t; Re) k^{-5/3}| dk \right)^{1/2}. \quad (47)$$

This quantity measures the difference between the spectrum of a reduced solution at a given time and the theoretical spectrum in the inertial range  $[k_1, k_2]$  described by the theory. The main advantage of this error indicator is that it can be applied no matter the numerical discretization and the turbulence model we are working with, as

it only involves the spectrum of the trial solution. That means that thanks to the fact that this indicator is build using only physical properties will allow us to avoid the construction of ad hoc *a posteriori* error estimators for the Reduced Basis discretization, which can be technically demanding, as the ones developed in Chacón Rebollo et al. (2023), Chacón Rebollo et al. (2017), and Deparis (2008), Manzoni (2014), based upon the Brezzi-Rappaz-Raviart theory (Brezzi et al., 1980) and that necessarily are specific for each actual turbulence model and numerical discretization.

In practice, we substitute  $k_2$  by  $k_c < k_2$ , where  $k_c$  is the smallest scale we can solve numerically by means of the turbulence model considered. In most cases,  $k_c$  is related to the mesh size  $\delta$ . Since this indicator is valid whenever the turbulence model solves a range of the inertial spectrum, special care should be taken in the construction of the mesh  $\mathcal{T}_h$ , so we are effectively solving in a suitable range, that is,  $k_c \in (k_1, k_2)$ , in practice  $k_c \ll k_2$ .

## 5.2 GENERAL ALGORITHM TO BUILD THE REDUCED BASIS SPACES

In this section, we describe a general algorithm based on the *a posteriori* error indicator  $\Delta_N(Re)$ , presented in (47), for the construction of the reduced basis model of LES turbulent model. We describe it here in a general framework, as it is independent of the actual LES model considered.

We intend to solve parametric turbulent flows depending on a parameter vector  $\gamma \in \mathcal{D} \subset \mathbb{R}^P$ , for any integer  $P \geq 1$ . In the applications that follow this parameter will be the Reynolds number, although it can also be a different physical or geometrical parameter, o several of these. We shall suppose that  $\gamma$  ranges on a compact set  $\mathcal{D} \subset \mathbb{R}^P$ .

Given a partition,  $\mathcal{D}_{train}$ , of the parameter space,  $\mathcal{D}$ , and a given starting parameter vector  $\gamma_0$ , the general algorithm for the Reduced Basis functions selection can be summarized as follows:

1. In iteration  $i \geq 1$ , solve the original full-order turbulence model for  $\gamma_{i-1}^* \in \mathcal{D}_{train}$  in some temporal interval, so the turbulence has reached the statistical equilibrium. Store the results of the velocity and pressure, separately.
2. Apply some technique to compress the snapshots for the velocity and pressure (for instance, one of the POD techniques explained in Section 1.3.3), and add them to the previous velocity and pressure Reduced Basis spaces. This will give us the new reduced dimension  $N_i$ .

3. In order to assure that the inf-sup condition (14) is satisfied for the pair of reduced velocity and pressure spaces, compute the inner pressure *supremizer* for the basis of the pressure resulting from the previous step (Ballarin et al., 2015), and append it to the basis of the reduced velocity space (see Subsection 2.2.2 for a detailed description). If some stabilization techniques are considered (Ali et al., 2020; Chacón Rebollo et al., 2023) this step would be no longer needed to be taken into account.
4. Compute the *a posteriori* error indicator  $\Delta_{N_i}^{(i)}(\gamma)$  for each  $\gamma \in \mathcal{D}_{train}$  associated to the reduced basis of dimension  $N_i$ . The error indicator should be computed for some discrete time sets to be determined.
5. Select the new parameter  $\gamma_i^*$  according to some criteria based upon the results of the error indicator  $\Delta_{N_i}^{(i)}$ . We will focus on this point in Section 5.2.1.
6. Check the stopping criteria. If the criteria are fulfilled, then stop the algorithm, otherwise, repeat from step 1.

As said before, this procedure does not depend on the way we obtain the full-order solution, so it is suitable for any turbulence model that satisfies the Kolmogorov's law and allows us to avoid the construction of *a posteriori* error estimators specific for each problem, that could involve some technical difficulties.

### 5.2.1 Parameter selection

Taking into account that we build the reduced solutions from the numerical approximations of the original solutions, the indicator  $\Delta_N(t; \gamma)$  is not expected to vanish as  $N$  grows. But, it is expected to converge to the indicator for the solution of the full order model  $\Delta_h(t; \gamma)$ , with

$$\Delta_h(t; \gamma) = \min_{\alpha \in \mathbb{R}} \left( \int_{k_1}^{k_c} |E_h(t, k; \gamma) - \alpha(\gamma)k^{-5/3}|^2 dk \right)^{1/2}$$

where  $E_h(t, k; \gamma)$  denotes the spectrum of the full-order solution at time  $t$  for  $k \in [k_1, k_c]$ , that is, the resolved inertial range. This spectrum should be close to the theoretical  $k^{-5/3}$  law if the full-order model is accurate, however, some error is to be expected as the full-order model is not exact and real-life flows do not follow strictly the theoretical law.

Then, the usual criterion for the selection of the best parameter in the Greedy algorithm, that is  $\gamma^* = \max_{\gamma \in \mathcal{D}} \Delta_N(\gamma)$ , is no longer useful. Instead, we propose in Algorithm 1 new criteria for the selection of the next parameter at each step.

---

**Algorithm 1:** Parameter selection criteria

---

**Require:**  $S_i$ , the set of previously selected parameters.  $\Delta_{N_{i-1}}^{(i-1)}$  and  $\Delta_{N_i}^{(i)}$ , the reduced energy spectrum of the previous and current iteration, respectively.

**Goal:** Obtain the next parameter  $\gamma^*$  for the Greedy algorithm.

Compute  $\gamma_i^* = \arg \max_{\gamma \in \mathcal{D}_{train}} \Delta_{N_i}^{(i)}(\gamma)$ ;

**if**  $\gamma_i^* \in S_i$  **then**

$\gamma_i^* = \arg \max_{\gamma \in \mathcal{D}_{train} \setminus S_i} |\Delta_{N_i}^{(i)}(\gamma) - \Delta_{N_{i-1}}^{(i-1)}(\gamma)|$

**end**

---

Algorithm 1 first aims to add to the basis the information of the solutions whose energy spectrum is farther away from the theoretical one, then, if the parameter selected that way has already been selected in a previous iteration, we select between the non-selected parameters, the one that provides the largest difference between the energy spectrums of the previous and the current reduced solutions. In the latter case, we are assuming that the reduced error indicator  $\Delta_N(\gamma)$  is getting close to  $\Delta_h(\gamma)$ , and the previous solution plays the role of the full-order one to compare with it.

# 6

---

## APPLICATION TO SMAGORINSKY TURBULENCE MODEL

---

In this chapter, we aim to validate the behavior of the indicator  $\Delta_N$ , equation (47), developed in the previous chapter. In particular, we address some academic tests for the Smagorinsky turbulence model. We consider 2D periodic flows, yet meaningful, as they are designed to present the inertial spectrum predicted by Kolmogórov's theory. We assume that the Smagorinsky flow already has a part of its spectrum within the inertial range.

The structure of the chapter is the following. First, we will perform a preliminary study in Section 6.1. In this section, we will focus on the presentation of the space-time discretization and the data of the problem. Then, we will select the POD+Greedy procedure and study the treatment of the eddy viscosity term and the time variability in the error indicator (47). Section 6.2 will be devoted to the presentation of the validation tests and numerical results.

### 6.1 REFINEMENT OF THE CHARACTERISTICS OF THE METHOD

This section is aimed to performing a preliminary study on the application of the Kolmogórov indicator (47) to the Smagorinsky turbulence model. In particular, we will tune up the method characteristics and play close attention to the selection of the POD+Greedy procedure.

#### 6.1.1 *Space-time discretization*

We consider the domain  $[0, T_f] \times \Omega$ , with  $\Omega = [-1/2, 1/2]^2$  and  $T_f = 15$ , and we solve the Smagorinsky model (15) for 2D periodic flows, with periodic boundary conditions and assume no external sources, thus  $f = 0$ .

In order to obtain the approximate solution, we use a semi-discretization in time using the Crank-Nicolson method with  $dt = 0.3125$ , so the number of time steps is  $L = 48$ , and for the resolution of the resulting problem we use stable finite elements for the pressure velocity, specifically Taylor-Hood i.e.,  $\text{P}2 - \text{P}1$ . The mesh on which it is solved is a structured mesh, with  $m = 64$  intervals in each of the directions, obtaining a mesh with 8192 triangles and 4225 vertices.

### 6.1.2 Data of the problem and initial Condition

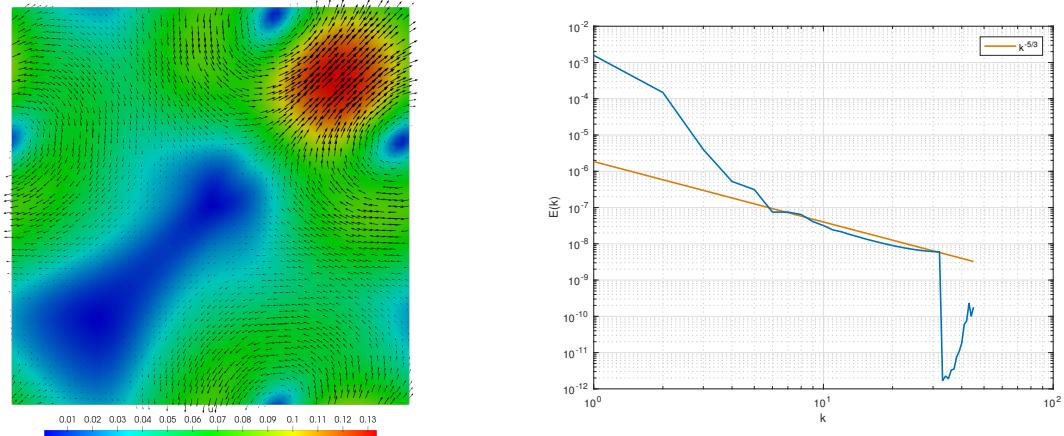
The parameter considered is the Reynolds number  $\gamma$ , which varies in the range  $\mathcal{D} = [1000, 16000]$ . For this range, approximately a time greater than 10 is needed to reach the profile of the Kolmogorov spectrum. In this case there is no inertial spectrum  $k^{-3}$  of the 2D turbulence, as a consequence of no forcing at large wavenumbers (Kraichnan, 1967).

For the initial condition, we need to consider a velocity field whose energy spectrum follows the form (46). In order to build this velocity field, we will start from its Fourier transform, so we can impose the desired energy spectrum. In particular, we take  $w_h^0 = (v, v)$  where  $v$  is such that its Fourier transform is given by

$$\hat{v}(k) = \begin{cases} k^{-(5/3+1)/2} & \text{if } 0 < k \leq m/4, \\ 0 & \text{other case.} \end{cases}$$

However, this velocity field is completely artificial, so it may not correspond to a realistic physical setting. In order to overcome this issue, we will define our initial condition  $u_h^0$  as the solution of the Smagorinsky model (15), with stating point  $w_h^0$ , for the mean value of the parameter,  $\gamma = 8500$ , at time  $T_f = 15$ . Thanks to this, the resulting initial condition  $u_h^0$  has an energy spectrum in statistical equilibrium, as required by our theory.

In Figure 6.1, we can see the initial condition  $u_h^0$  constructed and its energy spectrum. We can observe that for wavenumbers  $k$  between 5 and 32 a good approximation of the initial spectrum is obtained. For values larger than  $k_c = 32$ , we see a sharp decay of the energy, this is due to the fact that these wavenumbers are outside the circle of the largest radius completely contained in the unit square.



**Figure 6.1:** Velocity field of the initial condition  $\mathbf{u}_h^0$ , (left panel) and the corresponding energy spectrum (right panel).

### 6.1.3 Selection of the POD+Greedy procedure

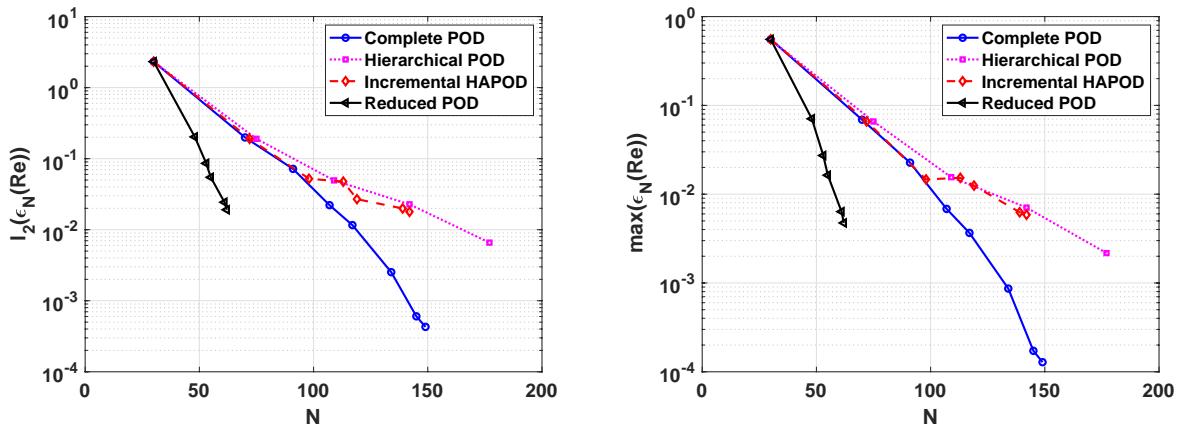
In this subsection, we aim to select the method that we will be using in the following numerical study. More precisely, we are comparing the four approaches we presented in Section 1.3.3. For this selection, and in the following numerical results, we will use the following combined velocity-pressure relative error

$$\epsilon_N(t; \gamma) = \left( \frac{\|\mathbf{u}_h(t; \gamma) - \mathbf{u}_N(t; \gamma)\|_{H^1}^2 + \|p_h(t; \gamma) - p_N(t; \gamma)\|_2^2}{\|\mathbf{u}_h(t; \gamma)\|_{H^1}^2 + \|p_h(t; \gamma)\|_2^2} \right)^{1/2}.$$

From this definition, we focus on two quantities of interest: the error at the final time step, i.e.  $\epsilon_N(T_f; \gamma)$ , and the time-integrated error, i.e.  $\int_{t_0}^{T_f} \epsilon_N(t; \gamma) dt$ .

In Figure 6.2, we can see the comparison between the methods presented in Section 1.3.3 in terms of the  $l_2$  and  $l_\infty$  norms over the parameter, for the same tolerance of the POD procedures.

The behavior is similar in both norms, and we can conclude various things from this comparison. On the one hand, we can see that the Reduced POD approach outperforms the others but stagnates in a low dimension of the reduced basis with no further improvement. On the other hand, the complete POD provides a better approximation of the full-order solution for a higher dimension of the reduced basis, achieving almost two more orders of magnitude in the relative error with respect to the Hierarchical POD (Haasdonk, 2017) and Incremental POD (Himpe et al., 2018). For the sake of completeness, we present in Table 6.1 a comparison of the evolution of the  $l_2$ -



**Figure 6.2:** Comparison of the performance of different POD+Greedy approaches with the use of the Kolmogorov error indicator. Left panel shows a comparison in parametric  $l_2$ -norm and right panel shows the comparison in parametric  $l_\infty$ -norm. For both panels, the x-axis stands for the dimension of the reduced space  $N$ , and the y-axis stands for the relative error w.r.t. the full-order solution in logarithmic scale.

norm of the relative errors for the Complete POD and Reduced POD approaches. In this table, we can clearly see that the Complete POD approach outperforms the Reduced POD at each POD+Greedy iteration, and that after the sixth iteration the Reduced POD approach stagnates with a  $l_\infty$ -norm of the relative error equal to  $4.75 \cdot 10^{-3}$ .

It.	Complete POD	Reduced POD
1	$5.54 \cdot 10^{-1}$	$5.54 \cdot 10^{-1}$
2	$6.90 \cdot 10^{-2}$	$7.07 \cdot 10^{-2}$
3	$2.27 \cdot 10^{-2}$	$2.72 \cdot 10^{-2}$
4	$6.83 \cdot 10^{-3}$	$1.63 \cdot 10^{-2}$
5	$3.66 \cdot 10^{-3}$	$6.36 \cdot 10^{-3}$
6	$8.66 \cdot 10^{-4}$	$4.75 \cdot 10^{-3}$
7	$1.72 \cdot 10^{-4}$	$4.75 \cdot 10^{-3}$
8	$1.28 \cdot 10^{-4}$	$4.75 \cdot 10^{-3}$

**Table 6.1:** Comparison of the  $l_\infty$ -norm of the relative error at each iteration of the POD+Greedy algorithm for the Complete POD and Reduced POD approaches. We can see that after the sixth iteration, the  $l_\infty$ -norm for the Reduced POD approach stagnates.

The stagnation of the Reduced POD could be explained by the fact that the Reduced POD extracts the dominant features of the FOM until the grid size scale. The remaining POD procedures give smaller errors with respect to the FOM solution, although possibly the scales smaller than the grid size are spurious, with no physical meaning,

as the Smagorinsky model only accurately reproduces the large scales that can be represented in the grid.

Taking all the above into account, we will be using the Complete POD procedure in the following studies and numerical tests.

#### 6.1.4 Treatment of the eddy viscosity term by EIM

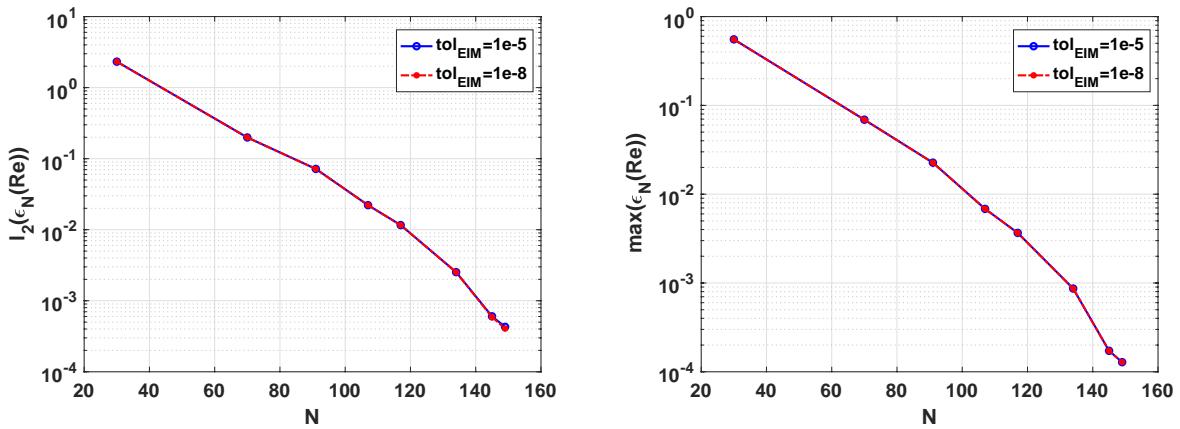
In this subsection, we study the tensorization of the Smagorinsky turbulent viscosity (13) with respect to the parameter, for which we will use the EIM, explained in Section 1.4.1. To test the accuracy of this procedure, we will compare two settings:

- On the one hand, we choose a set of 4 equispaced values of the parameter  $\gamma = \{1000, 6000, 11000, 16000\}$ , and we calculate the finite element solution  $(\mathbf{u}_h^n(\gamma), p_h^n(\gamma))$  for different time instants  $n = 1, \dots, L$ . With these snapshots we compute an approximation of the viscosity function by the EIM, taking a tolerance of  $\epsilon_{EIM} = 10^{-5}$ , resulting in 186 basis functions.
- On the other hand, we choose a set of 25 equispaced values of the parameter, instead of 4 as in the previous setting, and reduce the tolerance to  $\epsilon_{EIM} = 10^{-8}$ . Following the same procedure as for the previous point, we obtain 723 basis functions. Therefore, this latter should be more precise in computing the eddy viscosity.

The objective is to compare the results obtained by means of the EIM, considering that the second one is a highly accurate representation of the Smagorinsky turbulent viscosity term (13) and would act as if we were computing the Smagorinsky turbulent viscosity term in the original full-order space. We follow this approach, as computing the exact eddy viscosity would lead to very high computing times in the ROM.

In Figure 6.3, we show a comparison of the error decay in parametric  $l_2$  and  $l_\infty$  norms. It can be clearly seen that both decays are almost identical in both norms. Therefore, the first proposed EIM setting does not result on an augmentation of the error. So, in the following study, we will consider the first EIM setting.

For the sake of completeness, we present on the left panel of Figure 6.4 the convergence error of EIM as a function of the number of bases  $M$ , while on the right panel of Figure 6.4, we see the error over the parameter set  $\mathcal{D}$ , where each line corresponds to a time instant  $t_n$  for  $n = 1, \dots, L$ .



**Figure 6.3:** Comparison of the performance of different EIM approaches with the use of the Kolmogorov error indicator. Left panel shows a comparison in parametric  $l_2$ -norm and right panel shows the comparison in parametric  $l_\infty$ -norm. For both panels, the x-axis stands for the dimension of the reduced space  $N$ , and the y-axis stands for the relative error w.r.t the full-order solution in logarithmic scale.

### 6.1.5 Time variability in the error indicator

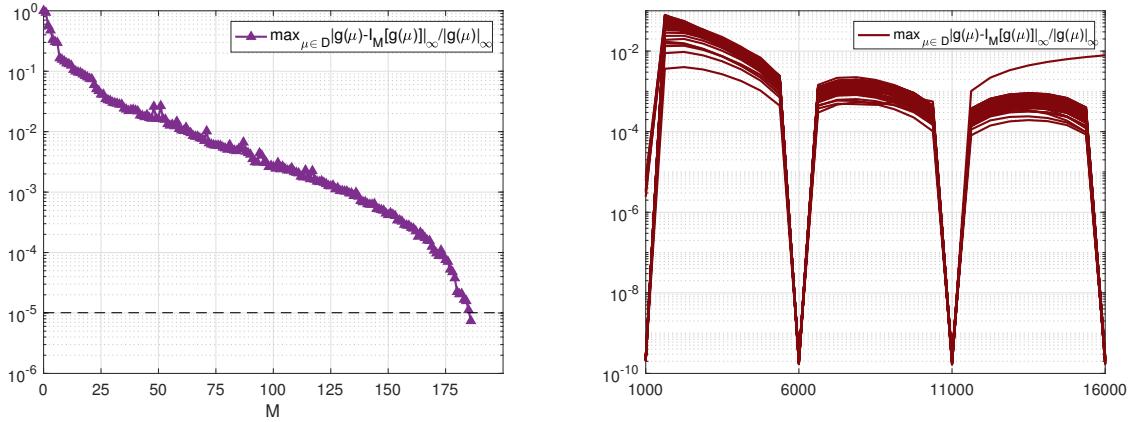
The final matter that we need to address before the validation is whether the use of the Kolmogórov error indicator at the final time step is good enough or we need to use a time-integrated version of it. In order to tackle this question, we will perform a comparison between the use of the indicator at the final step, i.e.  $\Delta_N(T_f; \gamma)$ , and the use of the time-integrated indicator, i.e.  $\int_{t_0}^{T_f} \Delta_N(t; \gamma) dt$ .

In Figure 6.5, we can see the comparisons of error decay for both approaches. We can see that in both cases the error decay is quite similar, although the time-integrated approach provides a slightly better relative error. However, the gain of using the time-integrated version does not compensate the computational effort that has to be done in the offline stage. So, for the rest of the study, we will keep the use of the indicator at the final step.

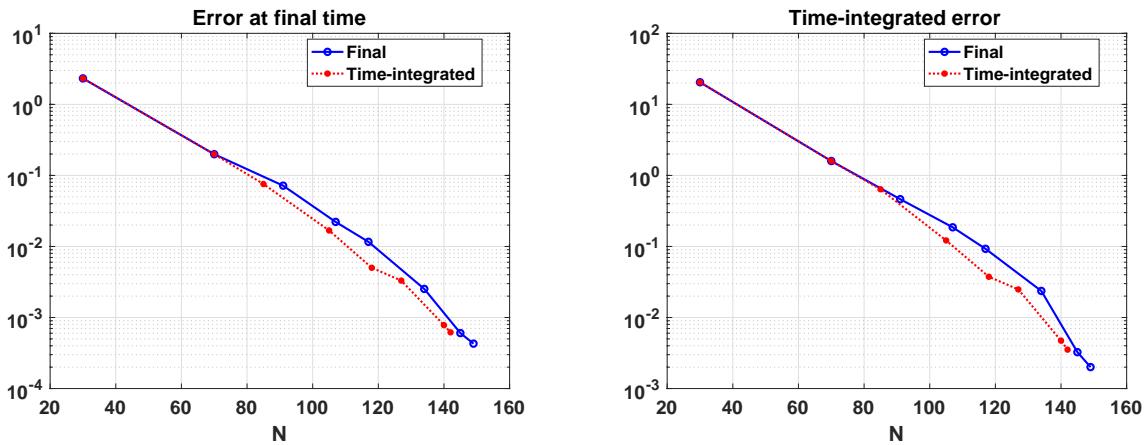
## 6.2 NUMERICAL RESULTS

In the light of the results of the preliminary study, we will use the following set for the numerical tests:

- We use  $u_h^0$  as the initial condition as it has a well-developed turbulence that follows Kolmogórov's law.



**Figure 6.4:** EIM error evolution. Left panel shows the convergence of the EIM error w.r.t. the number of EIM basis  $M$ , and right panel shows the EIM error at each time step  $t_n$  for  $n = 1, \dots, L$ .



**Figure 6.5:** Comparison of the performance of different approaches for the Kolmogorov error indicator. Left panel shows a comparison of the relative error measured at the final time. Right panel shows the comparison of the parametric  $l_1$ -norm of the relative errors in time. For both panels, the errors obtained by the use of the indicator at the final step is shown in blue fashion, while the time-integrated indicator is shown in red fashion, the x-axis stands for the dimension of the reduced space  $N$ , and the y-axis is in logarithm scale.

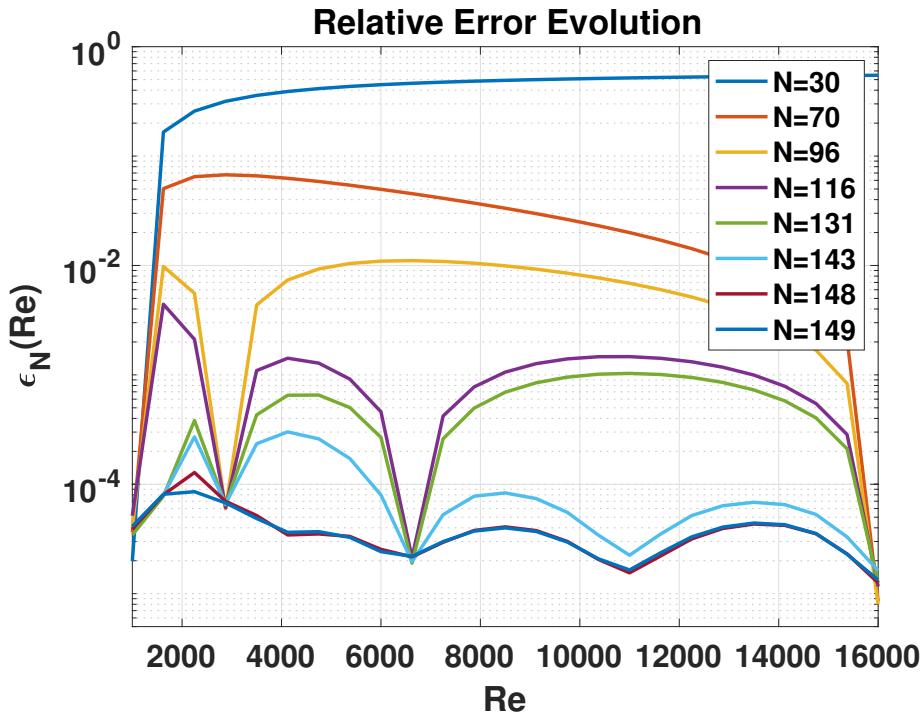
- For the POD+Greedy procedure, we will use the Complete POD approach, presented in Section 1.3.3, as it provides the best results in comparison with the rest of approaches.
- In order to approximate the eddy viscosity term, we will use the EIM with 186 basis functions, as adding more functions to the basis has been proven to not result in a major improvement in the error.
- We will be using the error indicator at the final time step, as using the time-integrated version requires much more computational effort and does not translate in a great improvement of the error.

### 6.2.1 Comparison against actual relative error

Now that we have set the numerical setting for the experiments, we aim to validate the *a posteriori* error indicator  $\Delta_N(\gamma)$  introduced in (47). We will be using the parameter values provided by the POD+Greedy procedure (Section 1.3.3) with the selection criteria presented in Algorithm 1 of Section 5.2.1, that provides the best value of the parameter at each iteration. We are going to compare the use of Kolmogórov's indicator and the use of the actual relative error at the final time  $T_f = 30$ .

In Figure 6.6, we show the evolution of the relative errors for the reduced models. It is clear that after the last iteration, all the relative errors in the considered range are below  $10^{-4}$ .

Also, in Table 6.2, we show the evolution of the maximum relative error if we were using the actual relative error as the indicator for the parameter selection. In the table, the selected values of the parameter are shown, as well as the dimension,  $N$ , of the reduced space  $X_N \times M_N$ . We can see that after the eighth iteration, the maximum relative error is below  $10^{-4}$ .



**Figure 6.6:** Relative error  $\epsilon_N(\gamma)$  in the range of the parameter, for each iteration of the POD+Greedy algorithm with the Complete POD procedure. Here, we are using the actual relative error for the parameter selection.

The main objective would be to obtain similar results when we use Kolmogórov's indicator (47) instead of the actual relative error. In Table 6.3, we present the results obtained in this case. As we can see, the resulting relative errors are comparable with respect to those obtained using the actual relative error (see Table 6.2). We can see that the maximum value for  $\Delta_N(\gamma)$ , stagnates after the second iteration at  $3.53 \cdot 10^{-1}$ , that is, the value of Kolmogórov's indicator for  $\gamma = 16000$ . Starting from the third iteration, we need to use the second criterion of Algorithm 1, explained in Section 5.2.1.

We also show in Figure 6.7 the evolution of the relative errors for the different reduced dimensions. It is clear to see that after the eighth iteration, the relative errors are below  $10^{-4}$  for the whole range of the parameters, as in the case where we were using the actual relative error.

It.	$\gamma$	N	$\max_{\gamma} \epsilon_N(\gamma)$
1	1000	30	$5.54 \cdot 10^{-1}$
2	16000	70	$6.74 \cdot 10^{-2}$
3	2875	96	$1.11 \cdot 10^{-2}$
4	6625	116	$4.43 \cdot 10^{-3}$
5	1625	131	$1.03 \cdot 10^{-3}$
6	11000	143	$3.01 \cdot 10^{-4}$
7	4125	148	$1.28 \cdot 10^{-4}$
8	2250	149	$8.55 \cdot 10^{-5}$

Table 6.2: Convergence story of the POD+Greedy algorithm with the Complete POD approach, using the actual relative error,  $\epsilon_N(\gamma)$ , for the parameter selection.

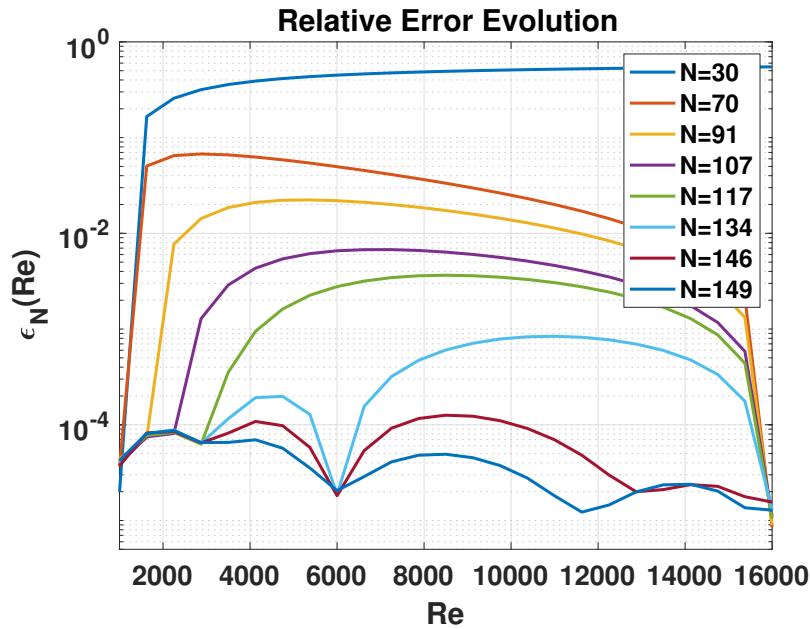
It.	$\gamma$	N	$\max_{\gamma} \Delta_N(\gamma)$	$\max_{\gamma}  \Delta_N(\gamma) - \Delta_{N^*}(\gamma) $	$\max_{\gamma} \epsilon_N(\gamma)$
1	1000	30	$7.92 \cdot 10^{-1}$	-	$5.54 \cdot 10^{-1}$
2	16000	70	$5.13 \cdot 10^{-1}$	-	$6.74 \cdot 10^{-2}$
3	1625	91	$3.53 \cdot 10^{-1}$	$2.11 \cdot 10^{-1}$	$2.23 \cdot 10^{-2}$
4	2250	107	"	$4.32 \cdot 10^{-2}$	$6.76 \cdot 10^{-3}$
5	2875	117	"	$1.68 \cdot 10^{-2}$	$3.65 \cdot 10^{-3}$
6	6000	134	"	$1.72 \cdot 10^{-3}$	$8.41 \cdot 10^{-4}$
7	12875	146	"	$1.46 \cdot 10^{-3}$	$1.26 \cdot 10^{-4}$
8	11625	149	"	$5.93 \cdot 10^{-5}$	$8.70 \cdot 10^{-5}$

Table 6.3: Convergence story of the POD+Greedy algorithm with the Complete POD approach, using Kolmogórov's indicator,  $\Delta_N(\gamma)$ , for the parameter selection. Here,  $\Delta_{N^*}(\gamma)$  denotes the error indicator in the previous iteration.

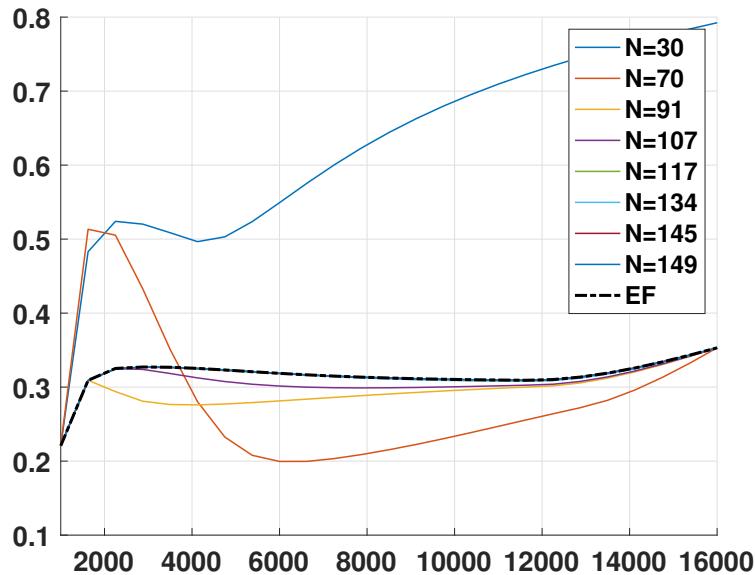
As we mentioned in Section 5.2.1, the indicator  $\Delta_N(\gamma)$  is not expected to converge to zero when the dimension  $N$  of the reduced space grows, rather it should converge to  $\Delta_h$ , since the reduced solution is constructed from the finite element approximation. This convergence is shown in the Figure 6.8, where we can easily see that the larger value for the indicator is indeed achieved for  $\gamma = 16000$ , and that after the third iteration, the rest of the values of the indicator are below that threshold. This fact supports the use of Algorithm 1 for the selection of the parameter.

From Figure 6.8, it can also be seen easily that after the fifth iteration, both, reduced and finite element indicators are very similar. In order to provide a better insight into this matter, we show in Figure 6.9, the absolute difference between the reduced indicators and the finite elements one. After the last iteration, we can see that the major difference between them is of order  $10^{-4}$ .

In Table 6.4, we show for several values of the parameter, which some do not correspond to those chosen by the Greedy algorithm, the calculation times of the finite

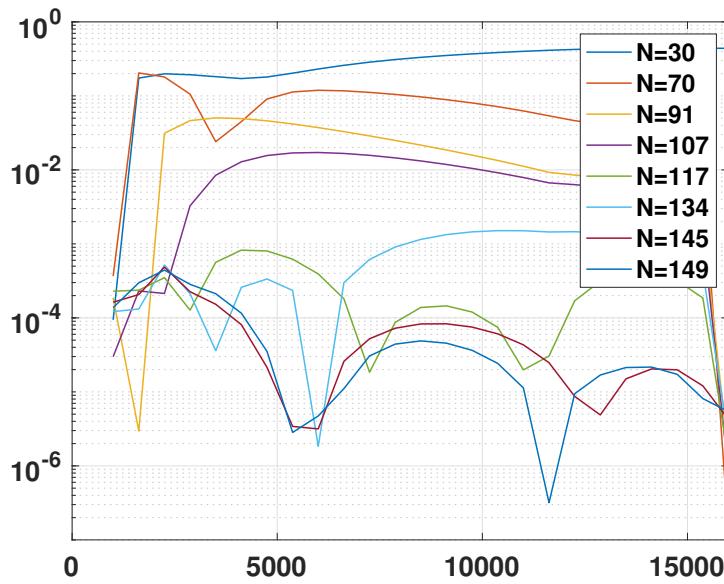


**Figure 6.7:** Relative error  $\epsilon_N(\gamma)$  in the range of the parameter, for each iteration of the POD+Greedy algorithm with the Complete POD procedure. Here, we are using Kolmogórov's indicator (47) for the parameter selection.



**Figure 6.8:** Convergence of Kolmogórov's indicator (47) for the reduced solutions to the finite elements Kolmogórov's indicator. Indicator from reduced solutions are shown in continuous fashion, while the one from the finite elements methods is shown in dash-dotted black fashion.

element method  $T_{FE}$  and of the reduced method  $T_{RB}$ , as well as the corresponding



**Figure 6.9:** Absolute difference between the Kolmogórov's indicator (47) for the reduced solutions and the finite elements Kolmogórov's indicator.

speed-up. The lower part shows the values corresponding to the indicators  $\Delta_N(\gamma)$ ,  $\Delta_h(\gamma)$  and the actual relative error  $\epsilon_N(\gamma)$  committed for the different cases.

$\gamma$	3000	6000	9000	12000	15000
$T_{FE}(s)$	62.982	62.579	62.634	62.632	62.535
$T_{RB}(s)$	2.73086	2.74046	2.71848	2.72783	2.71082
Speed-up	23.06	22.84	23.04	22.96	23.07
$\Delta_h(\gamma)$	0.327244	0.318601	0.311539	0.309520	0.337642
$\Delta_N(\gamma)$	0.326974	0.318606	0.311492	0.309526	0.337656
$\epsilon_N(\gamma)$	$6.27 \cdot 10^{-5}$	$2.05 \cdot 10^{-5}$	$4.64 \cdot 10^{-5}$	$1.28 \cdot 10^{-5}$	$1.78 \cdot 10^{-5}$

Table 6.4: Validation of the reduced basis model obtained by means of a POD+Greedy approach using Kolmogórov's indicator (47) for the parameter selection.

All the presented data has been obtained from the eight iteration of the POD+Greedy algorithm, with a reduced dimension of  $N = 149$ . The speed-up obtained is close to 23 in each of the cases tested, which is satisfying for an evolution turbulence model. The values of the indicators  $\Delta_N(\gamma)$ ,  $\Delta_h(\gamma)$ , are very close to each other, having at least two coincident decimal places. We can also see that the relative errors for the selected parameters are below  $10^{-4}$ .

In Figure 6.10, we show the comparison between the FE (left) and RB (center) velocity field, with the absolute error (right) between them, for a Reynolds value  $\gamma = 15000$ ,

and times  $t = 20, 25, 30$  seconds (from top to bottom), where we can observe that the difference between both FE and RB velocity field is practically negligible, since the maximum of the absolute error between both fields is less than  $10^{-7}$ .

Moreover, in Figure 6.11 we show the temporal evolution of the velocity magnitude for some spatial locations, in  $(0, 0)$ ,  $(0.25, 0.25)$  and  $(0.5, 0.5)$  for both the high FE solution and the RB solution for a Reynolds parameter value of  $\gamma = 15000$ . We can observe the regularity of the flow for both the FE and RB solution.

Therefore, we can conclude that the error indicator has been successfully extended from parametric linear parabolic problems to LES parametric turbulence models, obtaining relative errors in the order of  $10^{-5}$ .

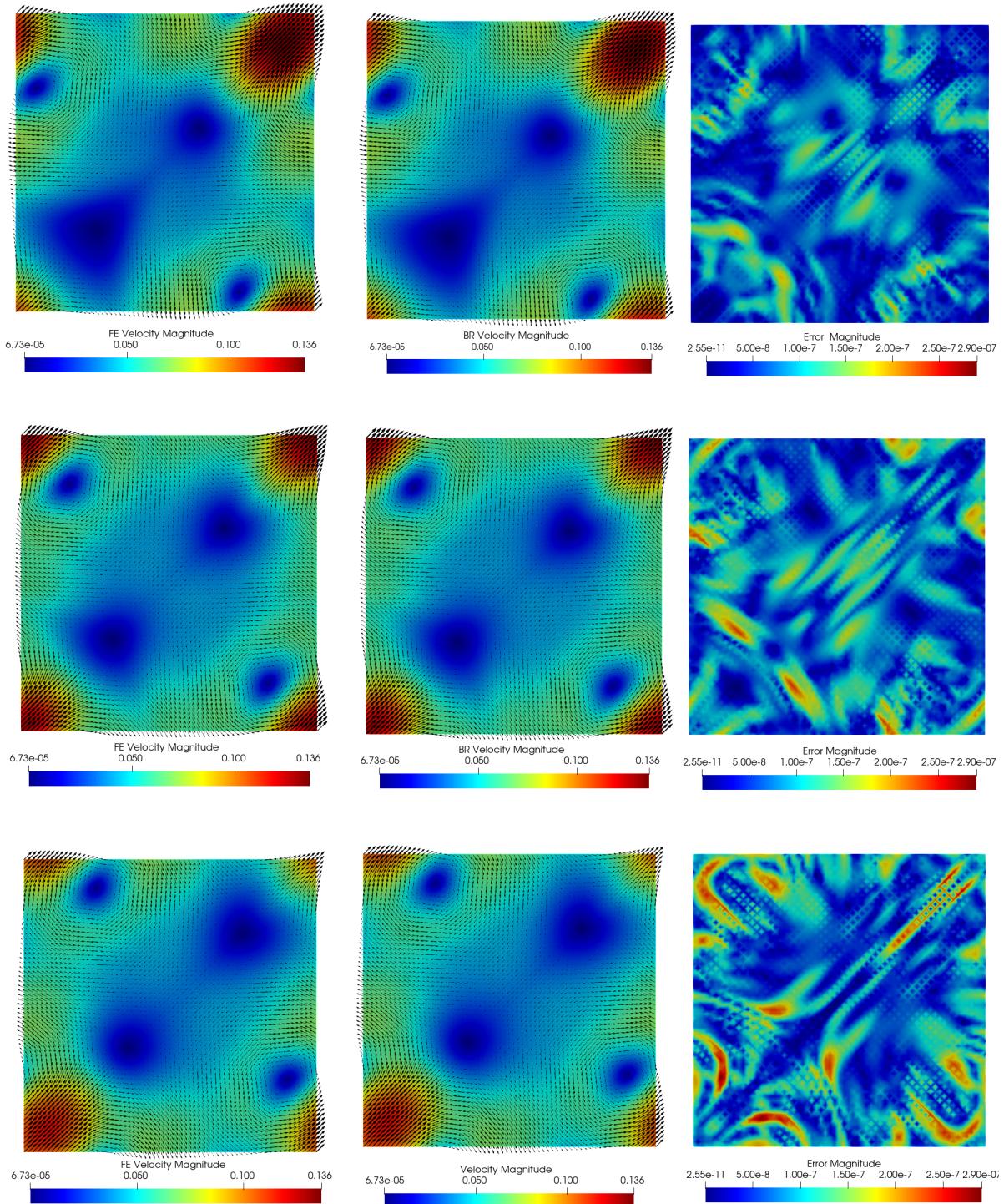
### 6.2.2 Comparison against equispaced sampling

The objective of this second test is to compare the results obtained in the previous test with the ones obtained when considering equispaced parameter values, instead of considering a Greedy algorithm for selecting them.

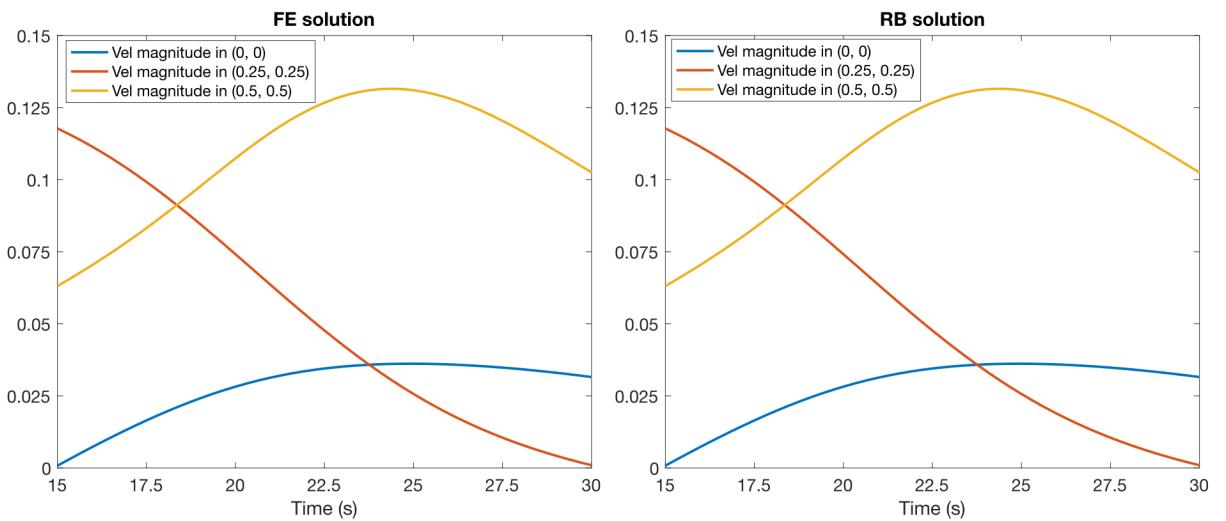
It.	N	$\max_{\gamma} \epsilon_N(\gamma)$	$l_2(\epsilon_N(\gamma))$
1	30	$5.54 \cdot 10^{-1}$	$5.53 \cdot 10^{-1}$
2	70	$6.74 \cdot 10^{-2}$	$1.95 \cdot 10^{-1}$
3	96	$2.63 \cdot 10^{-2}$	$4.95 \cdot 10^{-2}$
4	116	$1.27 \cdot 10^{-2}$	$1.96 \cdot 10^{-2}$
5	126	$6.96 \cdot 10^{-3}$	$9.18 \cdot 10^{-3}$
6	133	$3.81 \cdot 10^{-3}$	$4.49 \cdot 10^{-3}$
7	134	$2.85 \cdot 10^{-3}$	$3.20 \cdot 10^{-3}$
8	138	$1.96 \cdot 10^{-3}$	$2.12 \cdot 10^{-3}$

Table 6.5: Convergence of the algorithm with the Complete POD approach, using equispaced parameter values. Here, we present the maximum relative error and their  $l_2$ -norm.

In Table 6.5, we present the maximum relative error and the  $l_2$ -norm of the relative error in the parameter range. We are using the Complete POD approach, presented in Section 1.3.3, for the construction of the reduced basis. We can see that, after the second iteration, these values differ from the ones presented for the use of the actual relative error (Table 6.2) and those for the use of Kolmogórov's indicator (Table 6.3). While for the use of equispaced parameters after eight iterations the maximum error is above  $10^{-3}$ , the ones for the use of the actual error and Kolmogórov's error, after eight iterations the maximum error is below  $10^{-4}$ .



**Figure 6.10:** Comparison between FE (left) and RB (right) velocity field, along with the absolute error between them (right), for  $\gamma = 15000$ , at  $t = 20s, 25s, 30s$  from top to bottom.

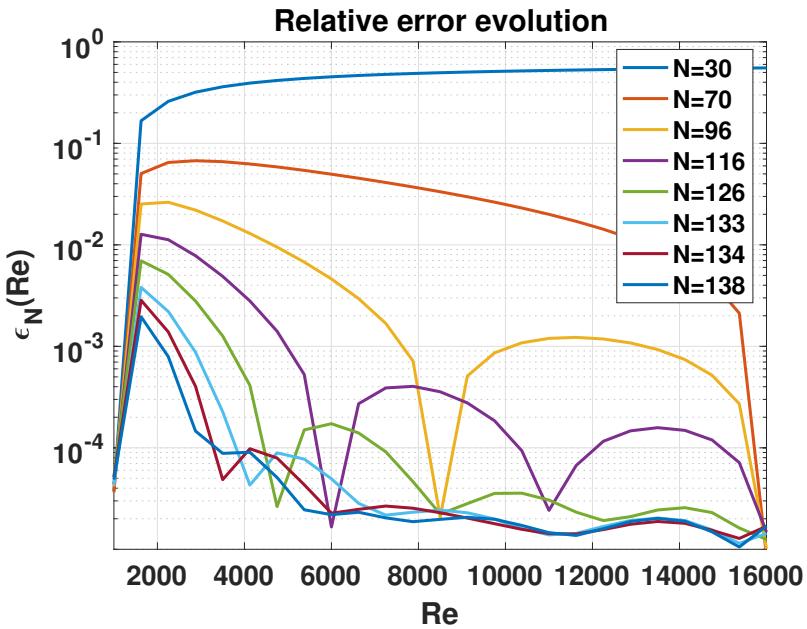


**Figure 6.11:** Temporal evolution of the FE (left) and RB (right) velocity magnitudes at three different points,  $x = (0, 0)$ ,  $x = (0.25, 0.25)$  and  $x = (0.5, 0.5)$ .

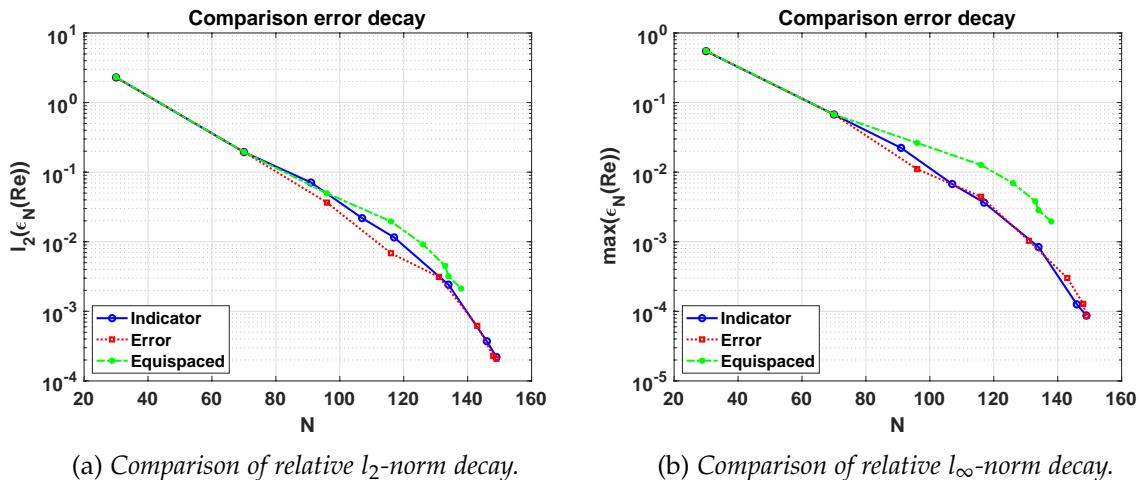
We also present in Figure 6.12 the evolution of the relative error in the range of the parameter. We can see that the largest errors appear for lower values of the parameter, decreasing slowly as the reduced basis dimension,  $N$ , increases. This is due to the fact that this approach to enrich the reduced basis is blind to the actual behavior of the solution. The approach involving Kolmogórov's indicator (47) has been proven to identify this kind of behavior, and, therefore, improving the enrichment of the reduced basis.

In order to have a better visual of the differences between each approach, we will compare their behaviors in two norms, the  $l_\infty$ -norm, and  $l_2$ -norm. We present, in Figure 6.13a, the comparison of the decay of the  $l_2$ -norm of the relative error over the parameter range and, in Figure 6.13b the comparison of the decay of the maximum relative error over the parameter range. We observe a spectral convergence with respect to the dimension of the reduced space  $N$ .

We also observe that the *a posteriori* error indicator based on Kolmogorov's law (47), provides very similar results compared to the ones obtained using the exact error, in both  $l_2$  and maximum relative error decay, when using the Complete POD procedure presented in Section 1.3.3. Also, we observe that it provides better results than just selecting equispaced parameters in the reduced basis construction, specially in the maximum relative error. This is due to the fact that for equispaced sampling the errors are localized in the lower Reynolds numbers, whereas thanks to our approach the information about these parameters are included in the construction of the reduced



**Figure 6.12:** Relative error  $\epsilon_N(\gamma)$  in the range of the parameter, for each iteration of the POD+Greedy algorithm with Complete POD procedure. Here, we are selecting equispaced parameter values.



**Figure 6.13:** Comparison of the relative error decay for the different methods. In both panels, the errors obtained by means of the a posteriori error indicator appear in continuous blue fashion with circles, the errors obtained using the real error appear in dashed red fashion with squares and the errors obtained using equispaced parameters are shown in dash-dotted green fashion with asterisks.

basis, therefore reducing the maximum error. Let us recall that the simple equispaced polynomial interpolation may present exponential decay if there is an analytic depen-

dence with respect to the parameter in a domain large enough. This fact possibly plays a role here, as the flow is quite smoothly changing as the Reynolds number changes.

Therefore, we conclude the validation tests for Kolmogórov's indicator (47) with very promising results. It has been shown that the application of this indicator provides a similar behavior in error decay as using the exact error, achieving relative errors of order  $10^{-5}$  after eight iterations. It has also been proven very useful the development of Algorithm 1, as it allows the enrichment of the reduced basis, taking into account that Kolmogórov's indicator (47) does not tend to 0 as  $N$  increases.



---

## CONCLUSIONS AND OPEN PROBLEMS

---

In this part, we have introduced a Reduced Basis model for turbulence at statistical equilibrium. This Reduced Basis model is based upon an *a posteriori* error estimation procedure that measures the distance from a trial solution to the theoretical  $k^{-5/3}$  energy spectrum provided by Kolmogórov's theory.

Upon this error estimation procedure, we have carried on an error analysis that clarifies the role of the energy-spectrum error estimator. We have stated that the error between the finite element solution and the reduced one is driven by this estimator, plus the projection error on the reduced space.

In order to build the reduced problem, we have presented a coupled POD+Greedy approach (Section 1.3.3). We have adapted the criteria for the selection of the new parameter in the Greedy algorithm, as this indicator is not intended to vanish as the dimensionality of the reduced model increases. To avoid the possible stagnation situations, we compare the energy spectrum of the trial solution with that of the current RB solution as a surrogate of the FOM solution.

In the field of the applications, we have tested the RB Smagorinsky model for some academic, yet meaningful, 2D flows, as these present the Kolmogorov's Theory energy spectrum profile. In the performed tests, the Reynolds number acts as the only parameter. We obtain speed-up ratios of computational time close to 23, which is very satisfying for an evolution turbulence model. In addition, we have obtained relative errors very similar to those provided by the reduced model built using the actual relative error for the parameter selection. Both errors present a similar exponential decay as the dimension of the reduced spaces increases, in both  $l^2$  and  $l^\infty$  parametric norms. We have also compared these errors against the ones obtained for the reduced model built using equispaced parameters, that present a much slower decay in  $L^\infty$  parametric norm.

One of the main interests of this novel *a posteriori* error indicator based upon physical properties of the models is that it applies to any kind of parameter (either geometrical or physical) and to any kind of numerical discretization. The only requirement is that the targeted flows have a well-developed inertial spectrum, and that the numerical

solver is able to reproduce a part of this inertial spectrum, what is well achieved by current solvers.

We present now some open problems related to this part:

- Extension of the academic test to three-dimensional turbulent flows and other space discretizations.
- Application to industry problems thanks to the low cost of the computation of Kolmogórov's indicator (47).
- Development of other error indicators based on specific laws of the energy spectrum, for instance, some kinds of non-equilibrium flows (Rubinstein & Clark, 2017).
- Reduced modelling of the turbulent limit layer, developing an estimator based upon wall-laws.

The research presented in this part of the thesis is an updated version of the research paper Bandera Moreno et al. (2024a).

### Part III

## INTRINSIC PROPER GENERALIZED DECOMPOSITION FOR PARAMETRIC ELLIPTIC PROBLEMS



---

## INTRODUCTION

---

In this part, our emphasis is on the resolution of symmetric elliptic Partial Differential Equations (PDEs), and the computation of the best subspace that approximates their solution. To that end, we focus on the Proper Generalized Decomposition (PGD) method, with the specific objective of addressing one of the limitations that it presents, namely, the computation of the optimal PGD modes. In particular, we aim to study the possible calculation of these modes directly in a Grassmann manifold by means of the Gradient Descent (GD) algorithm. This part aligns with the Spanish national projects ROAD and HYROM.

One of the most widely used tools in reduced order modelling for industrial problems is the Proper Orthogonal Decomposition (POD), presented in Section 1.3.1 (Pinna, 2008; Volkwein, 2013), which is commonly used along with Galerkin projection strategies in order to reduce the dimensionality of the problems (Chapelle et al., 2012; Kahlbacher & Volkwein, 2007; Kunisch & Volkwein, 2002), and the use of coupled Galerkin and POD strategies makes affordable the solution of many-query and on-the-fly simulations. However, as said in Section 1.3.1, the application of the POD technique requires some knowledge of the solution for a variety of parameters, large enough to capture the main behaviors of the real solution over a range of the parameters. This can be extremely expensive in terms of computational time or storage, as we have stated before.

In order to overcome this problematic, a new approach was introduced in Ammar et al. (2006), the so-called PGD. The idea behind this technique is to assume that the solution  $u(\gamma)$  admits a decomposition into a sum of basis functions, which do not need to be orthogonal between them, i.e.  $u = \sum_{i \geq 1} \alpha_i(\gamma) \phi_i$ . And to iteratively enrich the reduced solution by computing a tensorized representation of the parametric PDE that separates the problem into a parametric,  $\alpha_i(\gamma)$ , and a non-parametric part,  $\phi_i$ . The mathematical analysis and the application of the PGD has experienced an important development in the last years (Ammar, 2010; Ammar et al., 2006; Chinesta et al., 2010; Chinesta et al., 2013; Chinesta et al., 2011; Néron & Ladevèze, 2010).

We would like to highlight an extension of the PGD presented in Azaïez et al. (2018), that the authors call the intrinsic PGD. This technique is based on the construction

of recursive approximations on finite-dimensional optimal subspaces. These approximations are obtained by means of the minimization of the mean parametric error of the residual, similar to the procedure introduced in Falcó and Nouy (2011). In a latter work (Azaïez et al., 2020), this technique is applied along with a deflation algorithm and a Power Iterate (PI) method to compute the PGD modes for parametric elliptic partial differential equations both, symmetric and non-symmetric.

This part is structured as follows. In Chapter 7, we propose the resolution of the intrinsic PGD for parametric symmetric elliptic partial differential equations directly by means of the GD algorithm in a matrix framework to obtain the optimal subspaces. This matrix framework can be identified with the Grassmann manifold (Bendokat et al., 2020). Furthermore, we combine this method with a deflation algorithm in order to obtain one-dimensional subspaces and keep enriching the solution, simplifying the computations. Then, in Chapter 8, we compare our results with the ones obtained in Azaïez et al. (2020) as a validation of our method, and we also apply this novel technique to an elastostatics problem.

---

## COMPUTATION OF INTRINSIC PROPER GENERALIZED DECOMPOSITION MODES IN A MATRIX FRAMEWORK

---

In the introduction of this part, we have presented one of the problematics related to the high computational cost of obtaining a reduced order model of industrial problems by means of the POD technique, presented in Section 1.3.1. In order to overcome this issue, we will work with an extension of the PGD, the so-called intrinsic PGD (Azaïez et al., 2018; Azaïez et al., 2020). In this chapter, we develop a novel way to obtain the intrinsic PGD modes for parametric symmetric elliptic partial differential equations, directly by means of the GD algorithm in a matrix framework to obtain the optimal subspaces.

The chapter outline is the following. In Section 7.1, we present the main problem of the chapter and the intrinsic PGD in the continuous version. There, we will summarize several mathematical results and properties for the intrinsic PGD. Section 7.2 is devoted to the presentation of the algebraic form for the intrinsic PGD. Then, in Section 7.3, we present the general GD algorithm, and we explain the changes that are necessary to adapt the general GD algorithm to our framework. In order to end this chapter, we present in Section 7.4 the deflation algorithm used in Azaïez et al. (2020), which will allow us to obtain higher dimensional optimal subspaces by recursively obtaining one-dimensional optimal subspaces.

### 7.1 INTRINSIC PROPER GENERALIZED DECOMPOSITION

In this section, we will present the problem that we are trying to solve in its continuous version, along with some mathematical results and properties.

Throughout all this part, we assume that  $H$  is a separable Hilbert space with an associated norm denoted by  $\|\cdot\|$ ,  $B_S(H)$  is the space of bilinear, symmetric and continuous forms in  $H$ ,  $\langle \cdot, \cdot \rangle$  denotes the duality pairing between  $H'$  and  $H$ ,  $\mathcal{D}$  is a

given parametric space,  $(\mathcal{D}, \mathcal{B}, \mu)$  is a given measure space,  $a$  is a parametric coercive and continuous bilinear form defined over  $H \times H$  and  $f$  is the parametric data function belonging to  $H'$ .

Also, we will denote by  $u$  the parametric solution of the problem:

$$\text{Find } u(\gamma) \in H \text{ such that } a(u(\gamma), v; \gamma) = \langle f(\gamma), v \rangle, \quad \forall v \in H, d\mu\text{-a.e. } \gamma \in \mathcal{D}, \quad (48)$$

and by  $u_{\mathcal{Z}}$  the parametric solution of the Galerkin approximation of problem (48) on a closed subspace  $\mathcal{Z}$  of  $H$ , which can be expressed as:

$$\text{Find } u_{\mathcal{Z}}(\gamma) \in \mathcal{Z} \text{ such that } a(u_{\mathcal{Z}}(\gamma), z; \gamma) = \langle f(\gamma), z \rangle, \quad \forall z \in \mathcal{Z}, d\mu\text{-a.e. } \gamma \in \mathcal{D}. \quad (49)$$

We are interested in obtaining the subspace that better approximates the solution of a parametric PDE by solving the following minimization problem for a given  $k \in \mathbb{N}$ .

$$\left\{ \begin{array}{l} \text{Find the best subspace } \mathcal{W} \text{ of } H \text{ of dimension smaller than or equal to } k \text{ that solves} \\ \mathcal{W} := \arg \min_{\mathcal{Z} \in \mathbb{G}_{\leq k}} \bar{a}(u - u_{\mathcal{Z}}, u - u_{\mathcal{Z}}), \end{array} \right. \quad (50)$$

where  $\mathbb{G}_{\leq k} = \bigcup_{m \leq k} \mathbb{G}_m(H)$ , is the family of subspaces of  $H$  of dimension smaller or equal to  $k$ , and  $\bar{a} \in B_S(L^2(\mathcal{D}, H; d\mu))$  is a bilinear form given by

$$\bar{a}(v, w) = \int_{\mathcal{D}} a(v(\gamma), w(\gamma); \gamma) d\mu(\gamma), \quad \forall v, w \in L^2(\mathcal{D}, H; d\mu). \quad (51)$$

The following fundamental theorem for the intrinsic PGD was also presented in Azaïez et al. (2018, Theorem 4.1), where the interested reader could find the proof.

### Theorem 7.1

*For any given  $k \geq 1$ , the problem defined by (50) admits at least one solution.*

Furthermore, the authors stated that when  $\bar{a}$  is symmetric, an equivalent formulation can be provided, for which problem (50) does not depend on the knowledge of the solution  $u$  of (48), but only on the parametric data function  $f$ . The proof of this proposition can also be found in Azaïez et al. (2018, Proposition 2.6).

**Proposition 7.2**

Assuming that  $\bar{a}$  is symmetric, the subspace  $\mathcal{W} \in \mathbb{G}_{\leq k}$  solves problem (50) if and only if it is a solution of the problem,

$$\left\{ \begin{array}{l} \text{Find the best subspace } \mathcal{W} \text{ of } H \text{ of dimension smaller than or equal to } k \text{ that solves} \\ \mathcal{W} := \arg \max_{\mathcal{Z} \in \mathbb{G}_{\leq k}} \int_{\mathcal{D}} \langle f(\gamma), u_{\mathcal{Z}}(\gamma) \rangle d\mu(\gamma). \end{array} \right.$$

Also, in order to simplify the notation, we can define a real-valued function on  $\mathbb{G}_{\leq k}$ , as

$$\begin{aligned} F : \mathbb{G}_{\leq k} &\longrightarrow \mathbb{R} \\ \mathcal{Z} &\longmapsto - \int_{\mathcal{D}} \langle f(\gamma), u_{\mathcal{Z}}(\gamma) \rangle d\mu(\gamma), \end{aligned} \tag{52}$$

where we have kept the formulation as a minimization problem, changing the sign of the function. This allows us to express the continuous version of the intrinsic PGD for symmetric problems as,

$$\left\{ \begin{array}{l} \text{Find the best subspace } \mathcal{W} \text{ of } H \text{ of dimension smaller than or equal to } k \text{ that solves} \\ \mathcal{W} := \arg \min_{\mathcal{Z} \in \mathbb{G}_{\leq k}} F(\mathcal{Z}). \end{array} \right. \tag{53}$$

Thus, we are trying to solve a minimization problem of a real-valued function whose entries are subspaces of  $H$ . As one can suppose, at this state the problem cannot be tackled numerically. In order to do that, we need to build the algebraic representation of the intrinsic PGD problem (53).

## 7.2 ALGEBRAIC REPRESENTATION OF THE INTRINSIC PROPER GENERALIZED DECOMPOSITION

In this section, we will present the algebraic representation of the intrinsic PGD problem for symmetric PDEs (53).

We consider  $\{H_n\}_{n \in \mathbb{N}}$  an internal approximation of  $H$ , that is, a sequence of subspaces of finite dimension of  $H$  such that  $\lim_{n \rightarrow \infty} \inf_{\psi \in H_n} \|v - \psi\| = 0, \forall v \in H$ . This allows us to approximate the original infinite-dimensional space  $H$  by some finite

dimensional space  $H_n$ . The latter can also be identified with  $\mathbb{R}^n$ . We will be using this identification in the following.

In this framework, the parametric symmetric bilinear form  $a$  in problem (48) can be identified as a parametric matrix  $\mathbb{A}(\gamma)$  in  $\mathcal{M}_n$ , the set of real square matrices of dimension  $n$ , for all  $\gamma \in \mathcal{D}$ . Similarly, the parametric data function  $f(\gamma)$  in problem (48) can be identified as a vector  $\mathbf{b}(\gamma) \in \mathbb{R}^n$  for all  $\gamma \in \mathcal{D}$ . Finally, the duality pairing  $\langle \cdot, \cdot \rangle$  becomes the scalar product of two vectors in  $\mathbb{R}^n$ , thanks to the Riesz Representation Theorem.

Taking this into account, solving system (48) inside the finite dimensional space  $H_n$  is equivalent to solving the following linear system,

$$\mathbb{A}(\gamma)\mathbf{u}(\gamma) = \mathbf{b}(\gamma), \quad d\mu\text{-a.e. } \gamma \in \mathcal{D}.$$

### Remark 7.1

We note that the continuous solution  $u$  of problem (48) is identified now with a vector  $\mathbf{u} \in \mathbb{R}^n$ .

Also, we can identify the  $k$ -dimensional subspace  $\mathcal{Z}$ , that is now a subspace of  $\mathbb{R}^n$ , as a matrix  $\mathbb{Z} \in St(n, k) = \{\mathbb{Z} \in \mathbb{R}^{n \times k} : \mathbb{Z}^T \mathbb{Z} = \mathbb{I}_k\}$ , that is, belonging to the Stiefel manifold. More precisely,  $\mathcal{Z}$  is spanned by the columns of  $\mathbb{Z}$ , i.e.  $\mathcal{Z} = \text{col}(\mathbb{Z})$ .

Taking this into consideration, the bilinear form  $a(\cdot, \cdot; \gamma)$  and data function  $f(\gamma)$  in problem (49) can be identified as  $\mathbb{Z}^T \mathbb{A}(\gamma) \mathbb{Z} \in \mathcal{M}_k$  and  $\mathbb{Z}^T \mathbf{b}(\gamma) \in \mathbb{R}^k$ , respectively. Therefore, if we consider the ansatz  $u_{\mathbb{Z}}(\gamma)$ , defined as  $\mathbf{u}(\gamma) = \mathbb{Z} u_{\mathbb{Z}}(\gamma)$ , solving problem (49) is equivalent to solve the following linear system

$$(\mathbb{Z}^T \mathbb{A}(\gamma) \mathbb{Z}) \mathbf{u}_{\mathbb{Z}}(\gamma) = \mathbb{Z}^T \mathbf{b}(\gamma) \quad d\mu\text{-a.e. } \gamma \in \mathcal{D}. \quad (54)$$

Bearing equation (54) in mind, we can write the algebraic version of (52). With a slight abuse of notation,  $F : St(n, k) \rightarrow \mathbb{R}$  is a real-valued function defined as

$$F(\mathbb{Z}) = - \int_{\mathcal{D}} (\mathbf{b}(\gamma)^T \mathbb{Z}) (\mathbb{Z}^T \mathbb{A}(\gamma) \mathbb{Z})^{-1} (\mathbb{Z}^T \mathbf{b}(\gamma)) d\mu(\gamma). \quad (55)$$

Now that we know the algebraic form of all the elements involved in the definition of the intrinsic PGD problem (53), we can present its algebraic version.

**Proposition 7.3**

Let  $\mathbb{Z}$  be a matrix identified with the subspace  $\mathcal{Z}$ . Then, the algebraic form of problem (53) is

$$\left\{ \begin{array}{l} \text{Find the matrix } \mathbb{Z}_* \in St(n, k) \text{ that solves} \\ \mathbb{Z}_* := \arg \min_{\mathbb{Z} \in St(n, k)} F(\mathbb{Z}). \end{array} \right. \quad (56)$$

**Proof:** In order to obtain the algebraic form of problem (53), we work with  $F$  defined as (52), and we build its algebraic form.

Let  $\mathbb{Z}$  be a matrix identified with the subspace  $\mathcal{Z}$ . Then, from expression (54), the solution  $\mathbf{u}_{\mathcal{Z}}(\gamma)$  can be expressed as

$$\mathbf{u}_{\mathcal{Z}}(\gamma) = (\mathbb{Z}^T \mathbb{A}(\gamma) \mathbb{Z})^{-1} (\mathbb{Z}^T \mathbf{b}(\gamma)).$$

Now, explicitly writing the inner product,

$$\langle f(\mu), \mathbf{u}_{\mathcal{Z}} \rangle \approx (\mathbf{b}(\gamma)^T \mathbb{Z}) \mathbf{u}_{\mathcal{Z}}(\gamma) = (\mathbf{b}(\gamma)^T \mathbb{Z}) (\mathbb{Z}^T \mathbb{A}(\gamma) \mathbb{Z})^{-1} (\mathbb{Z}^T \mathbf{b}(\gamma)),$$

and integrating over  $\mathcal{D}$ , we obtain the desired algebraic form (55) of the objective function. Finally, changing the spaces for their representatives, we obtain the desired result. ■

We highlight now a property of this algebraic version of the intrinsic PGD that states the independence of function (55) with respect to the representative matrix of the subspace  $\mathcal{Z}$ .

**Proposition 7.4**

It is easy to see that the algebraic form of  $F$  does not depend on the matrix identified with the subspace  $\mathcal{Z}$ , that is,  $F(\mathbb{Z}) = F(\mathbb{Z}\mathbb{Q})$ ,  $\forall \mathbb{Q} \in \mathcal{O}_k$ , where  $\mathcal{O}_k$  is the set of orthonormal matrices of order  $k$ .

**Proof:** In this proof we will exploit the properties of orthonormal matrices in order to get the desired result.

$F(\mathbb{Z}\mathbb{Q})$  can be expressed explicitly as

$$F(\mathbb{Z}\mathbb{Q}) = - \int_{\mathcal{D}} (\mathbf{b}(\gamma)^T \mathbb{Z}\mathbb{Q}) (\mathbb{Q}^T \mathbb{Z}^T \mathbb{A}(\gamma) \mathbb{Z}\mathbb{Q})^{-1} (\mathbb{Q}^T \mathbb{Z}^T \mathbf{b}(\gamma)) d\mu(\gamma).$$

Noticing that  $(\mathbb{Q}^T \mathbb{Z}^T \mathbb{A}(\gamma) \mathbb{Z} \mathbb{Q})^{-1} = \mathbb{Q}^T (\mathbb{Z}^T \mathbb{A}(\gamma) \mathbb{Z})^{-1} \mathbb{Q}$  holds for every  $\mathbb{Q} \in \mathcal{O}_k$ , injecting this in the previous expression and taking into account that  $\mathbb{Q} \mathbb{Q}^T = \mathbb{I}_k$ , we obtain

$$\begin{aligned} F(\mathbb{Z}\mathbb{Q}) &= - \int_{\mathcal{D}} (\mathbf{b}(\gamma)^T \mathbb{Z})(\mathbb{Q}\mathbb{Q}^T)(\mathbb{Z}^T \mathbb{A}(\gamma) \mathbb{Z})^{-1}(\mathbb{Q}\mathbb{Q}^T)(\mathbb{Z}^T \mathbf{b}(\gamma)) d\mu(\gamma) = \\ &= - \int_{\mathcal{D}} (\mathbf{b}(\gamma)^T \mathbb{Z})(\mathbb{Z}^T \mathbb{A}(\gamma) \mathbb{Z})^{-1}(\mathbb{Z}^T \mathbf{b}(\gamma)) d\mu(\gamma) = F(\mathbb{Z}). \end{aligned}$$

That is the desired result. ■

Since  $F(\mathbb{Z}) = F(\mathbb{Z}\mathbb{Q})$  holds for all  $\mathbb{Q} \in \mathcal{O}_k$ , then we could express function (55) as a function on the Grassmann manifold, as follows

$$F : St(n, k)/\mathcal{O}_k \cong \mathbb{G}_k(\mathbb{R}^n) \rightarrow \mathbb{R}, \quad (57)$$

defined as  $F(\mathbb{Z}\mathcal{O}_k) = F(\mathbb{Z})$ , and where  $\mathcal{Z} = \text{col}(\mathbb{Z})$ . That means that we can use differential geometry tools of Grassmann manifolds in our case if we are able to find their respective algebraic representatives. This will prove very useful in the latter.

For numerical computations, it is natural to consider the discretized version. So, in the following, we will work in the matrix framework for the discrete version of the intrinsic PGD problem (56).

### 7.3 COMPUTATION OF INTRINSIC PROPER GENERALIZED DECOMPOSITION MODES VIA THE GRADIENT DESCENT ALGORITHM

In this section, we will develop a procedure to solve the intrinsic PGD in its algebraic form (56) through an adapted version of the GD Algorithm to our framework of matrix spaces.

The general GD Algorithm can be stated roughly as presented in Algorithm 2. Note that the time step  $t$  can be adapted for each iterate to speed up the convergence of the Algorithm.

---

**Algorithm 2:** Gradient Descent Algorithm

---

**Require:** Differentiable real-valued function  $f$ , step size  $t$ .

**Goal:** Find a local minimum of  $f$ .

**Input:** Initial point  $x_0$ .

**Output:** Sequence of iterates  $\{x_i\}_{i \geq 0}$ .

**for**  $i = 0, 1, \dots$  **do**

Compute the gradient of  $f$  at  $x_i$ ,  $\nabla f(x_i)$ .

Solve the Gradient Descent equation to obtain the next iterate

$$x_{i+1} = x_i - t \nabla f(x_i). \quad (58)$$

**end**

---

However, the general GD Algorithm 2 is not suitable to solve the intrinsic PGD in our matrix framework. On the one hand, the computation of the gradient of function (55) involves some technical difficulties that need to be cared about. On the other hand, if we were to update the iterates directly by the GD equation (58), the resulting matrix could no longer be inside our matrix manifold. Hence, in the following, we will study the necessary changes in Algorithm 2 in order to adapt it to the algebraic form of the intrinsic PGD problem (56).

### 7.3.1 Gradient of a function on a matrix manifold

First, we recall that the gradient of the function  $F$  defined in (55) on  $St(n, k)$  belongs to the tangent space of  $St(n, k)$ . It is known (Bendokat et al., 2020) that the tangent space of  $St(n, k)$  at the point  $Z$  can be expressed as,

$$T_Z St(n, k) = \{\Delta \in \mathbb{R}^{n \times k} \mid Z^T \Delta = -\Delta^T Z\} = \{\Delta \in \mathbb{R}^{n \times k} \mid \text{sym}(Z^T \Delta) = \mathbb{O}_k\}.$$

That is, the matrices in  $\mathbb{R}^{n \times k}$  such that after a left-multiplication by the transpose of matrix  $Z$  give the null square matrix of dimension  $k$ . Note that, if  $Z^T \Delta = \mathbb{O}_k$ , then  $Z^T \Delta = -\Delta^T Z$ , so it would be enough to prove that  $Z^T \Delta = \mathbb{O}_k$ , to check if  $\Delta$  belongs to  $T_Z St(n, k)$ .

Furthermore, following Bendokat et al. (2020), for a function  $F$  defined on  $St(n, k)$ , the gradient of  $F$  at  $Z$  can be expressed as

$$\nabla F(Z) = dF(Z) - ZZ^T dF(Z), \quad (59)$$

where  $dF : St(n, k) \rightarrow \mathbb{R}^{n \times k}$  is the matrix derivative of function  $F$  with respect to  $\mathbb{Z}$ .

### **Proposition 7.5**

*For a fixed  $\mathbb{Z} \in St(n, k)$ ,  $\nabla F(\mathbb{Z})$  belongs to the tangent space  $T_{\mathbb{Z}}St(n, k)$ .*

**Proof:** It is enough to left-multiply by  $\mathbb{Z}^T$  and take into account that  $\mathbb{Z}^T \mathbb{Z} = \mathbb{I}_k$ , as  $\mathbb{Z} \in St(n, k)$ . ■

Therefore, it is clear that in order to obtain the gradient of  $F$ , first, we need to compute the matrix derivative of function  $F$ .

In order to simplify some calculations in the following, we need to state two technical lemmas from Magnus and Neudecker (2019) and Petersen and Pedersen (2008), where their proofs can be found. They are related to matrix derivatives and computations.

The first lemma is devoted to the computation of the derivative of an inverse matrix.

### **Lemma 7.6**

*Let  $\mathbb{A}$  be a nonsingular matrix, then the derivative of its inverse with respect to  $\mathbb{A}$  can be expressed as follows,*

$$d(\mathbb{A}^{-1}) = -\mathbb{A}^{-1} d(\mathbb{A}) \mathbb{A}^{-1}.$$

And the other lemma is related to a vector multiplication property of the matrix derivative.

### **Lemma 7.7**

*Let  $\mathbf{a}$  and  $\mathbf{b}$  be two vectors and  $\mathbb{A}$  a matrix, then the following equality follows,*

$$\mathbf{a}^T d(\mathbb{A}) \mathbf{b} = \mathbf{a} \otimes \mathbf{b} = \mathbf{a} \mathbf{b}^T,$$

*where  $\otimes$  denotes the external product.*

In the following theorem, we give the algebraic form for the derivative of the intrinsic PGD function (55) with respect to the point  $\mathbb{Z}$  in our case of study.

### **Theorem 7.8**

*The algebraic form of the matrix derivative of the intrinsic PGD function  $F$  given in (55) with respect to  $\mathbb{Z}$  can be expressed as*

$$dF(\mathbb{Z}) = -2 \int_{\mathcal{D}} \left( \mathbb{I}_n - \mathbb{A}(\gamma) \mathbb{Z} (\mathbb{Z}^T \mathbb{A}(\gamma) \mathbb{Z})^{-1} \mathbb{Z}^T \right) \mathbf{b}(\gamma) \mathbf{b}(\gamma)^T \mathbb{Z} (\mathbb{Z}^T \mathbb{A}(\gamma) \mathbb{Z})^{-1} d\mu(\gamma). \quad (60)$$

**Proof:** For the sake of brevity, we omit the dependence of matrix  $\mathbb{A}$  and vector  $\mathbf{b}$  on the parameter vector  $\gamma$ , as it does not affect the following computations.

First, we define  $h$  as,

$$h(\mathbb{Z}) = \mathbf{b}^T \mathbb{Z} (\mathbb{Z}^T \mathbb{A} \mathbb{Z})^{-1} \mathbb{Z}^T \mathbf{b}.$$

So, using the product rule, the derivative can be expressed as follows,

$$dh(\mathbb{Z}) = \mathbf{b}^T d\mathbb{Z} (\mathbb{Z}^T \mathbb{A} \mathbb{Z})^{-1} \mathbb{Z}^T \mathbf{b} + \mathbf{b}^T \mathbb{Z} d((\mathbb{Z}^T \mathbb{A} \mathbb{Z})^{-1}) \mathbb{Z}^T \mathbf{b} + \mathbf{b}^T \mathbb{Z} (\mathbb{Z}^T \mathbb{A} \mathbb{Z})^{-1} d\mathbb{Z}^T \mathbf{b}.$$

Now, taking into account Lemma 7.6 and regrouping, we obtain,

$$\begin{aligned} dh(\mathbb{Z}) = & \mathbf{b}^T d\mathbb{Z} (\mathbb{Z}^T \mathbb{A} \mathbb{Z})^{-1} \mathbb{Z}^T \mathbf{b} + \mathbf{b}^T \mathbb{Z} (\mathbb{Z}^T \mathbb{A} \mathbb{Z})^{-1} d\mathbb{Z}^T \mathbf{b} \\ & - \mathbf{b}^T \mathbb{Z} [(\mathbb{Z}^T \mathbb{A} \mathbb{Z})^{-1} (d\mathbb{Z}^T \mathbb{A} \mathbb{Z} + \mathbb{Z}^T \mathbb{A} d\mathbb{Z}) (\mathbb{Z}^T \mathbb{A} \mathbb{Z})^{-1}] \mathbb{Z}^T \mathbf{b}. \end{aligned}$$

Noticing that  $\mathbb{A}$  is symmetric and that  $d\mathbb{Z}^T = (d\mathbb{Z})^T$ , we can rewrite the previous expression as,

$$dh(\mathbb{Z}) = 2[\mathbf{b}^T d\mathbb{Z} (\mathbb{Z}^T \mathbb{A} \mathbb{Z})^{-1} \mathbb{Z}^T \mathbf{b} - \mathbf{b}^T \mathbb{Z} (\mathbb{Z}^T \mathbb{A} \mathbb{Z})^{-1} \mathbb{Z}^T \mathbb{A} d\mathbb{Z} (\mathbb{Z}^T \mathbb{A} \mathbb{Z})^{-1} \mathbb{Z}^T \mathbf{b}].$$

Finally, applying Lemma 7.7 we obtain

$$\begin{aligned} dh(\mathbb{Z}) = & 2[\mathbf{b} \mathbf{b}^T \mathbb{Z} (\mathbb{Z}^T \mathbb{A} \mathbb{Z})^{-1} - \mathbb{A} \mathbb{Z} (\mathbb{Z}^T \mathbb{A} \mathbb{Z})^{-1} \mathbb{Z}^T \mathbf{b} \mathbf{b}^T \mathbb{Z} (\mathbb{Z}^T \mathbb{A} \mathbb{Z})^{-1}] \\ = & 2[\mathbb{I}_n - \mathbb{A} \mathbb{Z} (\mathbb{Z}^T \mathbb{A} \mathbb{Z})^{-1} \mathbb{Z}^T] \mathbf{b} \mathbf{b}^T \mathbb{Z} (\mathbb{Z}^T \mathbb{A} \mathbb{Z})^{-1}. \end{aligned}$$

And that is the desired result. ■

Now, according to expression (59), we can compute the gradient of the intrinsic PGD function (55) in the matrix framework. Nevertheless, in the following remark, we state that, actually, the derivative of the intrinsic PGD function (60) belongs to the tangent space  $T_{\mathbb{Z}} St(n, k)$ . Therefore, the derivative of the intrinsic PGD function (60) is indeed the gradient of the intrinsic PGD function (55), as stated in the following proposition.

### Proposition 7.9

Let  $F$  be the function defined in (54), and  $dF(\mathbb{Z})$  its derivative, computed as in Theorem 7.8.

Then,  $\mathbb{Z}^T dF(\mathbb{Z}) = \mathbb{O}_k$ , thus, following (59),  $\nabla F(\mathbb{Z}) = dF(\mathbb{Z})$  and, as a consequence,  
 $dF(\mathbb{Z}) \in T_{\mathbb{Z}} St(n, k)$ .

**Proof:** We focus on the first term inside the parenthesis in the integral of the intrinsic PGD function derivative (60), and we omit the parameter dependence.

First, we left-multiply it by  $\mathbb{Z}^T$ .

$$\begin{aligned}\mathbb{Z}^T(\mathbb{I}_n - \mathbf{A}\mathbb{Z}(\mathbb{Z}^T\mathbf{A}\mathbb{Z})^{-1}\mathbb{Z}^T) &= \\ &= \mathbb{Z}^T - (\mathbb{Z}^T\mathbf{A}\mathbb{Z})(\mathbb{Z}^T\mathbf{A}\mathbb{Z})^{-1}\mathbb{Z}^T = \mathbb{Z}^T - \mathbb{Z}^T = \mathbb{O}_{k \times n}.\end{aligned}$$

Now, taking into account expression (59), the right term vanishes, and we obtain the desired result. ■

In addition, we keep the independence with respect to the matrix representative of the subspace  $\mathcal{Z}$ . The proof is straightforward, as we can follow the same procedure as in the proof of Proposition 7.4.

### Proposition 7.10

*The derivative (60), and therefore the gradient, of the intrinsic PGD function (54) is invariant under right multiplication of matrix  $\mathbb{Z}$  by  $k$ -dimensional orthonormal matrices, that is,*

$$dF(\mathbb{Z}) = dF(\mathbb{Z}\mathbb{Q}), \quad \forall \mathbb{Q} \in \mathcal{O}_k.$$

Taking this into account, we could consider a map in the Grassmann manifold  $G_k(\mathbb{R}^n)$  in the same manner as we did in (57). This will prove very useful in order to overcome the issue studied in the following section, as we will be able to inherit some differential geometry tools from the Grassmann manifold.

Finally, we have an explicit expression for the matrix derivative of the intrinsic PGD function (60). We can move to the next necessary adaptation, the GD equation (58).

#### 7.3.2 Gradient Descent equation on a matrix manifold

The main point to note is that it is not possible to update the representative matrix  $\mathbb{Z}$  directly by means of the GD equation (58). We need that the new matrix arising from this equation  $\tilde{\mathbb{Z}} = \mathbb{Z} - t\nabla F(\mathbb{Z})$  belongs to  $St(n, k)$ , and we have no control over the orthonormality of  $\tilde{\mathbb{Z}}$ . In other words, if we try to update the matrix representative  $\mathbb{Z}$  directly by means of the GD equation (58), we almost surely leave the matrix manifold  $St(n, k)$ . In order to overcome this issue, we need to consider a tool to update the matrix representative  $\mathbb{Z}$ , similarly as the GD equation (58), while remaining in  $St(n, k)$ .

Here comes in handy that we can use differential geometry tools of Grassmann manifolds in our case, given we can develop their algebraic representative. In this work, we will consider a retraction on the matrix manifold, that is, a continuous mapping

from a topological space into a subspace that preserves the position of all points in that subspace.

More precisely, in the following and in the applications, we consider a map that transforms a tangent vector  $\Delta \in T_{\mathbb{Z}}St(n, k)$  to the endpoint  $g_{\mathbb{Z}}(t, \Delta) \in St(n, k)$ , for a give step  $t$ , of the unique geodesic  $g$  that emanates from  $\mathbb{Z}$  in the direction  $\Delta$ .

Thanks to this tool, we will be able to update the matrix representative  $\mathbb{Z}$  as done with the GD equation (58), while obtaining a result that belongs to the matrix manifold  $St(n, k)$ .

In the next proposition, we state how the geodesic on the matrix manifold can be expressed, that is, we give its algebraic representation. This proposition is a modification of Theorem 2.3 in Edelman et al. (1998), where the proof can be consulted.

### Proposition 7.11

Let  $\mathbb{Z} \in St(n, k)$ ,  $t \in (0, 1]$ , and  $\Delta \in T_{\mathbb{Z}}St(n, k)$ , that is,  $\mathbb{Z}^T \Delta = 0$ . Then,

$$g_{\mathbb{Z}}(t, \Delta) = (\mathbb{Z}\mathbb{V} \cos(t\Sigma) + \mathbb{U} \sin(t\Sigma))\mathbb{V}^T, \quad (61)$$

where  $\mathbb{U}\Sigma\mathbb{V}^T$  is the compact Singular Value Decomposition (SVD) of  $\Delta$ , so  $\mathbb{U} \in \mathbb{R}^{n \times r}$ ,  $\Sigma$  is a diagonal nonsingular matrix in  $\mathbb{R}^{r \times r}$ , and  $\mathbb{V} \in \mathbb{R}^{r \times k}$ , with  $r$  the number of non-zero singular values. Here,  $\cos(\mathbb{X})$  (resp.  $\sin(\mathbb{X})$ ) denotes the matrix whose entries are the cosine (sine) of the entries of matrix  $\mathbb{X}$ .

We state in the next proposition that the orthonormality is conserved.

### Proposition 7.12

Let  $\mathbb{Z} \in St(n, k)$ , and  $\tilde{\mathbb{Z}}$  such that,  $\tilde{\mathbb{Z}} = g_{\mathbb{Z}}(t, -\nabla F(\mathbb{Z}))$ . Then,  $\tilde{\mathbb{Z}}^T \tilde{\mathbb{Z}} = \mathbb{I}_k$ , that is,  $\tilde{\mathbb{Z}} \in St(n, k)$ .

**Proof:** First, we know that  $\mathbb{Z}^T \nabla F(\mathbb{Z}) = \mathbb{O}_k$  as  $\nabla F(\mathbb{Z}) \in T_{\mathbb{Z}}St(n, k)$ .

Now, let  $\mathbb{U}\Sigma\mathbb{V}^T$  be the compact SVD of  $\nabla F(\mathbb{Z})$ . So, if we name as  $r$  is the number of non-zero singular values, we have the following

- $\mathbb{U} \in \mathbb{R}^{n \times r}$  is a semi-unitary matrix,
- $\Sigma \in \mathbb{R}^{r \times r}$  is a diagonal nonsingular matrix,
- $\mathbb{V} \in \mathbb{R}^{k \times r}$  is a semi-unitary matrix.

Hence, it follows that,  $\mathbb{Z}^T \mathbb{U}\Sigma\mathbb{V}^T = \mathbb{O}_k$  implies  $\mathbb{Z}^T \mathbb{U} = \mathbb{O}_k$ , after two right-multiplications, one by  $\mathbb{V}$ , as  $\mathbb{V}^T \mathbb{V} = \mathbb{I}_r$ , and another one by  $\Sigma^{-1}$ , leading us to the previous result.

Then, if we left-multiply  $\tilde{\mathbb{Z}}$  by its transpose, we obtain the following

$$\begin{aligned}\tilde{\mathbb{Z}}^T \tilde{\mathbb{Z}} &= \mathbb{V}(\cos(t\Sigma)^T \mathbb{V}^T \mathbb{Z}^T + \sin(t\Sigma) \mathbb{U}^T)(\mathbb{Z} \mathbb{V} \cos(t\Sigma) + \mathbb{U} \sin(t\Sigma)) \mathbb{V}^T \\ &= \mathbb{V}(\cos(t\Sigma)^T \cos(t\Sigma) + \sin(t\Sigma)^T \sin(t\Sigma) + \\ &\quad + \cos(t\Sigma)^T \mathbb{V}^T \mathbb{Z}^T \mathbb{U} \sin(t\Sigma) + \sin(t\Sigma)^T \mathbb{U}^T \mathbb{Z} \mathbb{V} \cos(t\Sigma)) \mathbb{V}^T,\end{aligned}$$

where the last two terms cancel because  $\mathbb{Z}^T \mathbb{U} = \mathbb{O}_k$ , and we obtain

$$\tilde{\mathbb{Z}}^T \tilde{\mathbb{Z}} = \mathbb{V}(\cos(t\Sigma)^T \cos(t\Sigma) + \sin(t\Sigma)^T \sin(t\Sigma)) \mathbb{V}^T = \mathbb{I}_k.$$

Therefore, matrix  $\tilde{\mathbb{Z}}$  arising from the retraction (61) is orthonormal and therefore  $\tilde{\mathbb{Z}} \in St(n, k)$ , which is the desired result. ■

It is clear now that we can use the retraction (61) as a substitute of the gradient equation (58) in Algorithm 2, as it updates the matrix representative  $\mathbb{Z}$  with the information given by the gradient of equation (55) at  $\mathbb{Z}$ , and the resulting matrix to  $St(n, k)$ .

### 7.3.3 Adapted Gradient Descent algorithm to the matrix framework

At this point, we have all the necessary ingredients to adapt the general GD Algorithm 2 to our algebraic framework.

As discussed in the previous sections, we need to compute the gradient of the function  $F$  as stated in equation (60) and then, instead of solving the GD equation (58) directly, we would need to compute retraction (61) in order to conserve the orthonormality of the matrices.

Taking Theorem 7.8 and Proposition 7.11 into account, the GD Algorithm adapted to the general matrix framework can be stated as in Algorithm 3.

Indeed, Algorithm 3 is the Gradient Descent Algorithm expected when considering minimization in the Stiefel manifold. This adapted GD Algorithm to the matrix framework is suitable to compute the intrinsic PGD modes. However, the computation of the SVD of a matrix, even in the compact version, can be very demanding in terms of computational effort. In order to overcome this issue, in the following section, we will present the particular case where the dimension of the desired subspace is 1. In this particular case, we will be allowed to simplify the computation of the gradient (60), as well as the computation of the retraction (61), as we will not need to perform the

---

**Algorithm 3:** Gradient Descent Algorithm on the matrix framework

---

**Require:** Differentiable real-valued function  $F$ , step size  $t$ .

**Goal:** Find a local minimum of  $F$ .

**Input:** Initial point  $\mathbb{Z}_0 \in St(n, k)$ .

**Output:** Sequence of iterates  $\{\mathbb{Z}_i\}_{i \geq 0} \subset St(n, k)$ .

**for**  $i = 0, 1, 2, \dots$  **do**

Compute the gradient of  $F$  at  $\mathbb{Z}_i$ ,  $\nabla F(\mathbb{Z}_i)$ , via equation (60).

Compute the compact SVD of  $-\nabla F(\mathbb{Z}_i)$  to obtain the matrices  $\mathbb{U}, \Sigma$  and  $\mathbb{V}$ .

Compute the retraction (61)

$$\mathbb{Z}_{i+1} = g_{\mathbb{Z}}(t, -\nabla F_{\mathbb{Z}}(\mathbb{Z}_i)) = (\mathbb{Z}_i \mathbb{V} \cos(t\Sigma) + \mathbb{U} \sin(t\Sigma)) \mathbb{V}^T.$$

**end**

---

compact SVD at each step of Algorithm 3. Furthermore, we will present this particular case along with a deflation algorithm in order to use the one-dimensional tools to obtain optimal subspaces of higher dimensions, by enriching the solution.

#### 7.4 DEFLATION ALGORITHM FOR THE COMPUTATION OF INTRINSIC PROPER GENERALIZED DECOMPOSITION MODES VIA THE GRADIENT DESCENT ALGORITHM

A particular interesting case from the point of view of applications and computational simplification is when the subspace  $\mathcal{Z} \in H$  is one-dimensional, i.e.  $k = 1$ , for which  $\mathcal{Z} \in \mathbb{G}_1$ . This means that there exists a basis function  $\phi \in H \setminus \{0\}$  such that  $\mathcal{Z} = \text{span}\{\phi\}$ . Then, the solution  $u_{\mathcal{Z}}(\gamma)$  of problem (49) can be obtained as a parameter-dependent scalar multiplied by that basis function, that is,  $u_{\mathcal{Z}}(\gamma) = \alpha(\gamma)\phi$ ,  $d\mu$ -a.e.  $\gamma \in \mathcal{D}$ .

In this section, we start by presenting the one-dimensional particular cases of the intrinsic PGD problem in both forms, continuous (53) and algebraic (56). Then, we will present a particular case of the adapted GD Algorithm 3 in which we are looking to solve the intrinsic PGD problem (56) when  $k = 1$ , that is, we are looking for the optimal one-dimensional space. Finally, we will explain how this one-dimensional approach can be coupled with a deflation algorithm in order to obtain optimal subspaces of higher dimensions.

### 7.4.1 One-dimensional intrinsic Proper Generalized Decomposition

First, we start from the continuous formulation, as it will be helpful to introduce naturally some concepts that we will be using subsequently. Letting  $z = \phi$  in equation (49), after a short calculation using the bilinearity of  $a$ , we can see that,

$$\alpha(\gamma) = \frac{\langle f(\gamma), \phi \rangle}{a(\phi, \phi; \gamma)} d\mu\text{-a.e. } \gamma \in \mathcal{D},$$

so, thanks to problem (53),  $\phi$  can be characterized by the solution of the following minimization problem:

$$\left\{ \begin{array}{l} \text{Find the best one-dimensional subspace of } H, \text{ whose generating vector } \phi_* \text{ solves} \\ \phi_* := \arg \min_{\phi \in H \setminus \{0\}} F(\phi) \end{array} \right. \quad (62)$$

where  $F$  is a real-valued function on  $H \setminus \{0\}$  defined as

$$F(\phi) = - \int_{\mathcal{D}} \alpha(\gamma) \langle f(\gamma), \phi \rangle d\mu(\gamma) = - \int_{\mathcal{D}} \frac{\langle f(\gamma), \phi \rangle^2}{a(\phi, \phi; \gamma)} d\mu(\gamma).$$

Now that we have stated the particular one-dimensional case for the continuous intrinsic PGD function (52), we can start to develop its algebraic version. The first remark that we need to take into account in this particular algebraic framework is that when  $k = 1$ , then the matrix representative  $\mathbb{Z}$  identified with the subspace  $\mathcal{Z}$  becomes the column matrix  $\boldsymbol{\phi} \in \mathbb{R}^{n \times 1} \cong \mathbb{R}^n$ , that will allow us to identify the Galerkin discrete representatives by two scalars, as stated in the following remark.

#### Remark 7.2

*The Galerkin discrete representatives of the bilinear form  $\mathbb{Z}^T \mathbb{A}(\gamma) \mathbb{Z}$  and the data  $\mathbb{Z}^T \mathbf{b}(\gamma)$ , become the scalars  $\boldsymbol{\phi}^T \mathbb{A}(\gamma) \boldsymbol{\phi}$ , and  $\boldsymbol{\phi}^T \mathbf{b}(\gamma)$ , respectively.*

The identification of the Galerkin representatives with scalar instead of matrices and vectors will allow us to simplify various steps in the calculation of the gradient and the retraction presented in the multidimensional framework, and that will lead to a subsequent reduction in the computational effort required to obtain the intrinsic PGD modes. Therefore, if we consider the one-dimensional version, we can define an algebraic form of problem (62), presented in the following Proposition.

**Proposition 7.13**

In the one-dimensional case, subspace  $\mathcal{Z}$  can be identified with the basis vector  $\boldsymbol{\phi} \in \mathbb{R}^n$ . So, the coefficient  $\alpha(\gamma)$  can be computed as

$$\alpha(\gamma) = \frac{\boldsymbol{\phi}^T \mathbf{b}(\gamma)}{\boldsymbol{\phi}^T \mathbb{A}(\gamma) \boldsymbol{\phi}} d\mu\text{-a.e. } \gamma \in \mathcal{D}.$$

As a consequence, problem (62) becomes:

$$\left\{ \begin{array}{l} \text{Find the best one-dimensional subspace of } H_n, \text{ whose generating vector solves} \\ \min_{\boldsymbol{\phi} \in \mathbb{R}^n} F(\boldsymbol{\phi}), \end{array} \right.$$

where

$$F(\boldsymbol{\phi}) = - \int_{\mathcal{D}} \frac{(\boldsymbol{\phi}^T \mathbf{b}(\gamma))^2}{\boldsymbol{\phi}^T \mathbb{A}(\gamma) \boldsymbol{\phi}} d\mu(\gamma). \quad (63)$$

The proof of this proposition is straightforward, as it is a particular case of the previous results with  $k = 1$  in the matrix setting. One of the main advantages that we want to remark in this one-dimensional particular case, is that, as we have to work with scalars instead of matrices and vectors, we will avoid the computation of the inverses of matrices as we can just divide by the corresponding scalar.

Now that we have defined the algebraic representative of the one-dimensional version of the intrinsic PGD problem, we can move to present the one-dimensional version of the adapted Gradient Descent algorithm to the matrix framework.

#### 7.4.2 One-dimensional adapted Gradient Descent algorithm to the matrix framework

First, we study the particular one-dimensional version of the intrinsic PGD function derivative (60). Noticing that the matrix inverses can be substituted by a division by the respective scalar, we get to the following remark.

**Remark 7.3**

For  $St(n, 1)$ , the algebraic form of the partial derivative of  $F$  in Theorem 7.8 takes the form

$$dF(\boldsymbol{\phi}) = -2 \int_{\mathcal{D}} \left[ \mathbb{I}_n - \frac{\mathbb{A}(\gamma) \boldsymbol{\phi} \boldsymbol{\phi}^T}{\boldsymbol{\phi}^T \mathbb{A}(\gamma) \boldsymbol{\phi}} \right] \frac{\mathbf{b}(\gamma) \mathbf{b}(\gamma)^T \boldsymbol{\phi}}{\boldsymbol{\phi}^T \mathbb{A}(\gamma) \boldsymbol{\phi}} d\mu(\gamma). \quad (64)$$

As for the multidimensional version (see Proposition 7.9), we are able to identify the gradient of the one-dimensional intrinsic PGD function (63) with the one-dimensional intrinsic PGD function derivative (64), that is,  $\nabla F(\phi) = dF(\phi)$ . As after a left-multiplication by the transpose of the basis vector  $\phi$ , we obtain the null row vector,

$$\phi^T - \frac{\phi^T \mathbb{A}(\gamma) \phi \phi^T}{\phi^T \mathbb{A}(\gamma) \phi} = \mathbf{0}_n^T.$$

With this, we conclude the computation of the one-dimensional version of the gradient of function (63).

The main advantage that we obtain if we consider the one-dimensional particular case arrives at the computation of the retraction, as we will be able to avoid the computation of the compact SVD at each step of Algorithm 3. In the following proposition, we state the form of the retraction in the particular one-dimensional case.

#### **Proposition 7.14**

*For  $St(n, 1)$ , the geodesic that emanates from  $\phi$  in the direction  $\Delta$  of Proposition 7.11 takes the form*

$$g_\phi(t, \Delta) = \cos(t\|\Delta\|_2)\phi + \sin(t\|\Delta\|_2)\frac{\Delta}{\|\Delta\|_2}. \quad (65)$$

**Proof:** When computing the compact SVD of a non-zero vector  $v \in \mathbb{R}^n$ , we can easily check

that,  $v = \frac{v}{\|v\|_2}\|v\|_2\mathbf{1} = U\Sigma V^T$ , that is,

- Diagonal matrix  $\Sigma$  becomes a scalar equal to the  $l_2$ -norm of the vector  $v$ ,  $s = \|v\|_2$ .
- Left-singular matrix  $U$  becomes the normalized vector  $\delta = v/\|v\|_2$ .
- Right-singular matrix  $V$  becomes the scalar  $\mathbf{1}$ .

Substituting these changes in equation (61) and rearranging the terms, we obtain the desired result. ■

Hence, we have all the ingredients to state the one-dimensional particular case of the adapted GD Algorithm 3. Taking Remark 7.3 and Proposition 7.14 into account, in the particular case of  $St(n, 1)$ , the adapted GD Algorithm 3 is reduced to Algorithm 4.

---

**Algorithm 4:** Gradient Descent Algorithm on  $St(n, 1)$ 

---

**Require:** Differentiable real-valued function  $F$ , step size  $t$ .

**Goal:** Find a local minimum of  $F$ .

**Input:** Initial point  $\phi_0 \in St(n, 1)$ .

**Output:** Sequence of iterates  $\{\phi_i\}_{i \geq 0} \subset St(n, 1)$ .

**for**  $i = 0, 1, 2, \dots$  **do**

Compute the gradient of  $F$  at  $\phi_i$ ,  $\nabla F(\phi_i)$  via equation (64).

Compute  $s = \|\nabla F(\phi_i)\|_2$  and  $\delta = -\nabla F(\phi_i)/s$ .

Solve the Geodesic Descent equation via the retraction (65)

$$\phi_{i+1} = g_\phi(t, -\nabla F(\phi_i)) = \cos(ts)\phi_i + \sin(ts)\delta.$$

**end**

---

The algorithm cost of each step of the proposed technique goes like  $O(mn^2)$ , where  $m$  is the number of operations considered for evaluating the integral in equation (64) numerically by means of a suitable quadrature formula. One of the main drawback of this technique is that it only allows us to compute the optimal one-dimensional subspace, while in most applications it is more interesting to obtain optimal subspaces of higher dimension. With that objective in mind, we will present a deflation algorithm, in the same sense as in Azaïez et al. (2020), that we can couple with the one-dimensional tools presented in Algorithm 4, in order to obtain optimal subspaces of increasing dimension in an iterative manner by an enrichment procedure.

#### 7.4.3 Deflation algorithm

The main idea behind the deflation algorithm is to express the solution of problem (48) as a series,

$$u(\gamma) = \sum_{i \geq 1} \alpha_i(\gamma) \phi_i,$$

where  $\alpha_i \in L^2(\mathcal{D}, d\mu)$  and  $\phi_i \in H \setminus \{0\}$ .

Then, in the case in which we have computed  $p - 1$  basis functions, we can approximate the solution  $u$  as a sum of a known part and an unknown part, as

$$u(\gamma) \approx u_p(\gamma) = \sum_{i=1}^p \alpha_i(\gamma) \phi_i = \alpha_p(\gamma) \phi_p + \sum_{i=1}^{p-1} \alpha_i(\gamma) \phi_i = \alpha_p(\gamma) \phi_p + u_{p-1}(\gamma),$$

with

$$u_{p-1}(\gamma) = \sum_{j=1}^{p-1} \alpha_j(\gamma) \phi_j, \quad \text{for } p \geq 2. \quad (66)$$

Note that, this latter expression is the known part of the solution. Function  $u_k(\gamma)$  for  $k \geq 1$  it is the so-called PGD expansion. Injecting this PGD expansion for  $k = p - 1$  along with the unknown part  $\alpha_p(\gamma) \phi_p$  into equation (51), we obtain the following decomposition,

$$\bar{a}(u, v) = \bar{a}(\alpha_p \phi_p, v) + \bar{a}(u_{p-1}, v),$$

where the latter term is known at step  $p$  and can be treated as a change of the right term of the PDE.

That is, we will solve functions of the same kind as (52), replacing the parametric data function  $f$  by a parametric data function  $f_i$  that will change with the dimension, defined by  $f_1 = f$  and  $f_i$ , for  $i \geq 2$ , such that,

$$\int_{\mathcal{D}} \langle f_i(\gamma), v(\gamma) \rangle d\mu(\gamma) = \int_{\mathcal{D}} \langle f(\gamma), v(\gamma) \rangle d\mu(\gamma) - \bar{a}(u_{i-1}, v), \quad \forall v \in L^2(\mathcal{D}, H, d\mu).$$

The convergence of the deflation algorithm is ensured by the following theorem, whose proof can be seen in Theorem 5.3 of Azaïez et al. (2018).

### Theorem 7.15

*The sequence provided by the iterative deflation algorithm (66) strongly converges in  $L^2(\gamma, H; d\mu)$  to the parametrized solution  $\gamma \in \mathcal{D} \mapsto u(\gamma) \in H$  of problem (48).*

Bearing all the above in mind, we are able to develop an iterative algorithm, that enriches the PGD expansion of the solution by means of the one-dimensional tools, presented in Section 7.4.2 and the deflation algorithm presented in this section. Therefore, in Algorithm 5, we present the procedure to obtain a multidimensional subspace using this deflation method together with the one-dimensional tools presented in Algorithm 4.

Now that we have developed the procedure to obtain the intrinsic PGD modes for parametric symmetric elliptic differential equations by means of the GD algorithm, we proceed in the next chapter to validate and test our method.

---

**Algorithm 5:** Iterative Gradient Descent Algorithm by deflation

---

**Require:** Matrices  $\mathbb{A}(\gamma)$ , vectors  $\mathbf{b}(\gamma)$ , associated to the parametrized PDE, objective dimension  $k$ , step size  $t$ .

**Goal:** Find optimal matrix  $\mathbb{Z} \in St(n, k)$  that solves problem (55).

**Input:** Initial point  $\boldsymbol{\phi}_0 \in St(n, 1)$ .

**Output:** Sequence of matrices  $\{\mathbb{Z}_i\}_{i=1,\dots,k}$  with  $\mathbb{Z}_i \in \mathcal{M}_i(\mathbb{R}^{n \times i})$ ,  $\forall i = 1, \dots, k$ .

Set  $\mathbf{b}_1(\gamma) = \mathbf{b}(\gamma)$  and  $\mathbb{Z}_0 = [ ]$ .

**for**  $i = 1, \dots, k$  **do**

Define real-valued function  $F_i$  with matrices  $\mathbb{A}(\gamma)$  and vectors  $\mathbf{b}_i(\gamma)$ .

Apply Algorithm 4 to obtain  $\boldsymbol{\phi}_i$ .

Make  $\boldsymbol{\phi}_i$  orthonormal to the subset  $\mathbb{Z}_{i-1}$ .

Compute  $\alpha_i(\gamma) = \frac{\boldsymbol{\phi}_i^T \mathbf{b}_i(\gamma)}{\boldsymbol{\phi}_i^T \mathbb{A}(\gamma) \boldsymbol{\phi}_i} d\mu\text{-a.e. } \gamma \in \mathcal{D}$ .

Update  $\mathbf{b}_{i+1}(\gamma) = \mathbf{b}_i(\gamma) - \sum_{j=1}^i \alpha_j \mathbb{A}(\gamma) \boldsymbol{\phi}_j$ .

Set  $\mathbb{Z}_i = [\mathbb{Z}_{i-1} | \boldsymbol{\phi}_i]$ .

**end**

---



# 8

---

## APPLICATION TO SYMMETRIC ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS

---

This chapter is devoted to the validation and numerical study of the procedure presented in Algorithm 5. We will apply the procedure to two different parametric problems:

1. A problem of parametric diffusion, where two different materials coexist in a domain. Here, the parameter will be the relationship between the diffusion of both materials.
2. A classic elastostatics problem of a block of material fixed on one side, and which is under the action of gravity. Here, the parameter will be the relationship between the Lamé parameters of the material.

In both cases, we will present the modes obtained by means of Algorithm 5 and the error decay of the reduced solutions. Here,  $u(\gamma)$  stands for the reference numerical solutions obtained by means of the Finite Element Method with a fine mesh, and  $u_p(\gamma)$  denotes the intrinsic PGD expansion up to mode  $p$  as in expression (66).

### 8.1 DIFFUSION PROBLEM

In this section, we will test Algorithm 5 in a diffusion problem previously treated in Azaïez et al. (2020), and we will compare the results obtained with the ones obtained by means of the Power Iterate (PI) algorithm proposed in that paper. This test will serve as a validation of our procedure.

The diffusion problem that we are treating in this section, is related to a parametric second order elliptic partial differential equation, where the diffusion is variable in space, and depending on the parameter, this problem could model a certain block

composed by two materials with different diffusion parameters. The problem, as presented in Azaïez et al. (2020), reads

$$\begin{cases} -\nabla \cdot (\nu(\gamma) \nabla u) = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \quad (67)$$

where  $\Omega = (0, 1)^2$ ,  $f(x, y) = 1$  and

$$\nu(\gamma)(x, y) = \begin{cases} \gamma & \text{if } 0 \leq x < 1/4, \\ 1 & \text{if } 1/4 \leq x \leq 1. \end{cases} \quad \text{for all } (x, y) \in \overline{\Omega}.$$

Problem (67) admits the following weak formulation

$$\begin{cases} \text{Find } u(\gamma) \in H_0^1(\Omega) \text{ such that} \\ \int_{\Omega} \nu(\gamma) \nabla u(\gamma) \cdot \nabla v = \int_{\Omega} f v, \quad \forall v \in H_0^1(\Omega). \end{cases} \quad (68)$$

In the following, as done in Azaïez et al. (2020), we set  $\gamma \in \mathcal{D} = [0.01, 1]$ , with  $\mu$  the Lebesgue measure on  $\mathcal{D}$ . Thanks to the properties of  $\nu$  and  $f$ , and Lax-Milgram's Theorem, there exists a unique solution of problem (68). The integrals on  $\mathcal{D}$  that appear in function (63), as well as its gradient (64), need to be addressed by means of a quadrature formula constructed on a subdivision of  $\mathcal{D}$  in  $N$  subintervals, for instance, the midpoint, trapezoid or Simpson's rule, among others. In every case, we replace  $\mathcal{D}$  by the set formed by the quadrature formula nodes  $\mathcal{D}_N = \{\gamma_i\}_{i=1}^N$  and the Lebesgue measure by a discrete measure  $\mu_N$  concentrated at these nodes, with weights which actually are the weights  $w_i$  of the quadrature formula. That is, we can rewrite the integrals on  $\mathcal{D}$  as a sum

$$\begin{aligned} \int_{\mathcal{D}} \psi(\gamma) d\mu(\gamma) &\approx \hat{I}_N(\psi) = \sum_{i=1}^N w_i \psi(\gamma_i), \\ \int_{\mathcal{D}} \psi_1(\gamma) \psi_2(\gamma) d\mu(\gamma) &\approx I_N(\psi_1, \psi_2) = \sum_{i=1}^N w_i \psi_1(\gamma_i) \psi_2(\gamma_i) \end{aligned} \quad (69)$$

It is important to note that Algorithms 3 and 4, as well as the iterative procedure presented in Algorithm 5 do not change when replacing  $\mathcal{D}$  by  $\mathcal{D}_N$  and  $\mu$  by  $\mu_N$ .

Then, the partial sums  $u_i$  of the PGD expansion given by formula (66) will converge to the parametric solution  $u(\gamma)$  of problem (67) in the sense that

$$\lim_{i \rightarrow \infty} I_N(u - u_i, u - u_i) = 0. \quad (70)$$

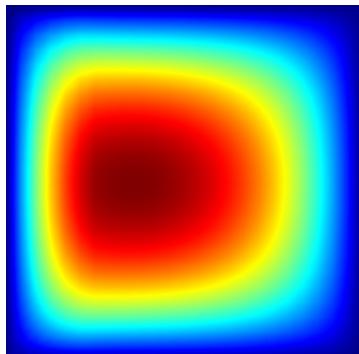
For this first test, in order to be consistent with the previous works, we have used the mid-point quadrature formula with  $N = 10$  equally-spaced subintervals. This corresponds to

$$\gamma_i = \frac{1}{N}(i - 1/2), \quad w_i = \frac{1}{N}, \quad i = 1, \dots, N.$$

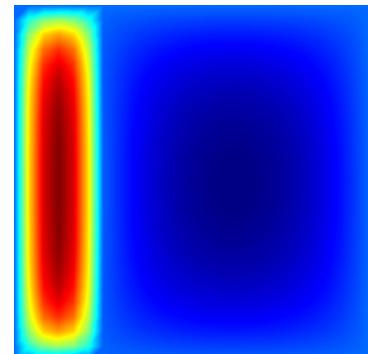
After applying the iterative Algorithm 5, we obtain the intrinsic PGD modes that can be seen in Figure 8.1. It is clear that the first modes keep the most important information about the solutions, and that the higher modes try to correct the differences that arise in the interface of the materials where the diffusion presents a discontinuity. A simple comparison with the intrinsic PGD modes obtained in Azaiez et al. (2020), shows that there are some qualitative differences between them, and they become more visible after the second mode.

Now, we are going to study the convergence of the truncated series  $u_i$  given by formula (66) to the parametric solution  $u(\gamma)$ , in the sense stated in expression (70). In Figure 8.2, we show the obtained truncation errors, measured in  $L^2(\mathcal{D}, L^2(\Omega), d\mu)$  and  $L^2(\mathcal{D}, H_0^1(\Omega), d\mu)$  norms and presented in logarithm scale, versus the dimension of the reduced subspace. We also compare the error decay with the one obtained in Azaiez et al. (2020) for the PI algorithm. We can easily see that the decay is pretty similar for the two methods until the third PGD mode, where the new procedure gives a much faster decay of the error, achieving up to four less orders of magnitude.

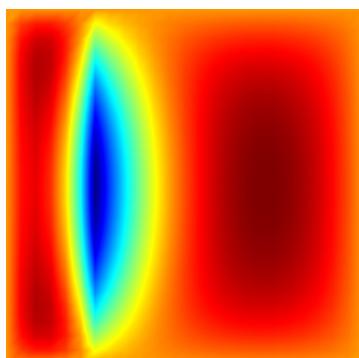
Finally, in order to present a better comparison of the results with those obtained in Azaiez et al. (2020), we show in figures 8.3-8.4, for each subspace dimension, a comparison of the true errors obtained in the whole domain between the reduced solutions and the parametric solution  $u(\gamma)$  of problem (67) for a fixed instance of the parameter  $\gamma = 0.3$ . In the left panels, we show the errors obtained by means of Algorithm 5, and the errors represented on the right panels are the ones obtained by means of the PI Algorithm presented in Azaiez et al. (2020). We can see that the errors are comparable in magnitude up to the second PGD mode, and that after the third mode the errors given by the iterative Algorithm 5 are much better in the whole domain than the ones obtained in Azaiez et al. (2020) by means of the PI method.



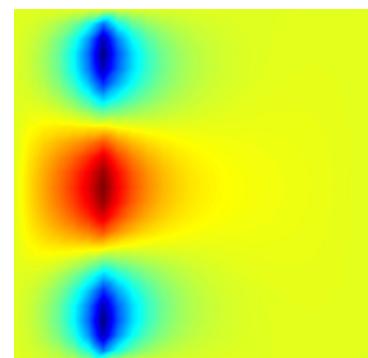
(a) Mode 1



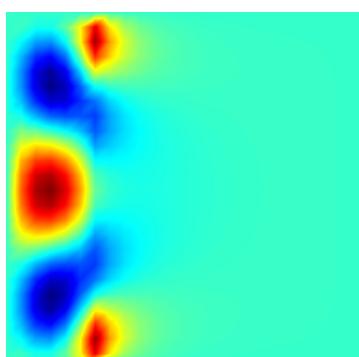
(b) Mode 2



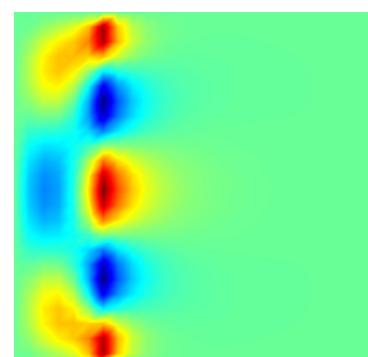
(c) Mode 3



(d) Mode 4

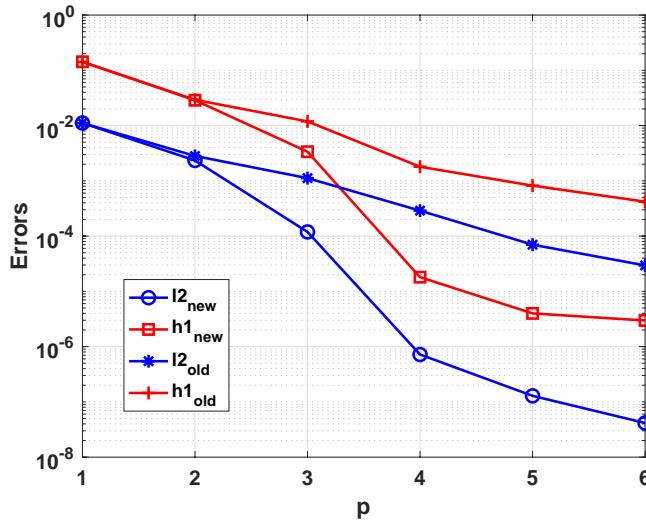


(e) Mode 5



(f) Mode 6

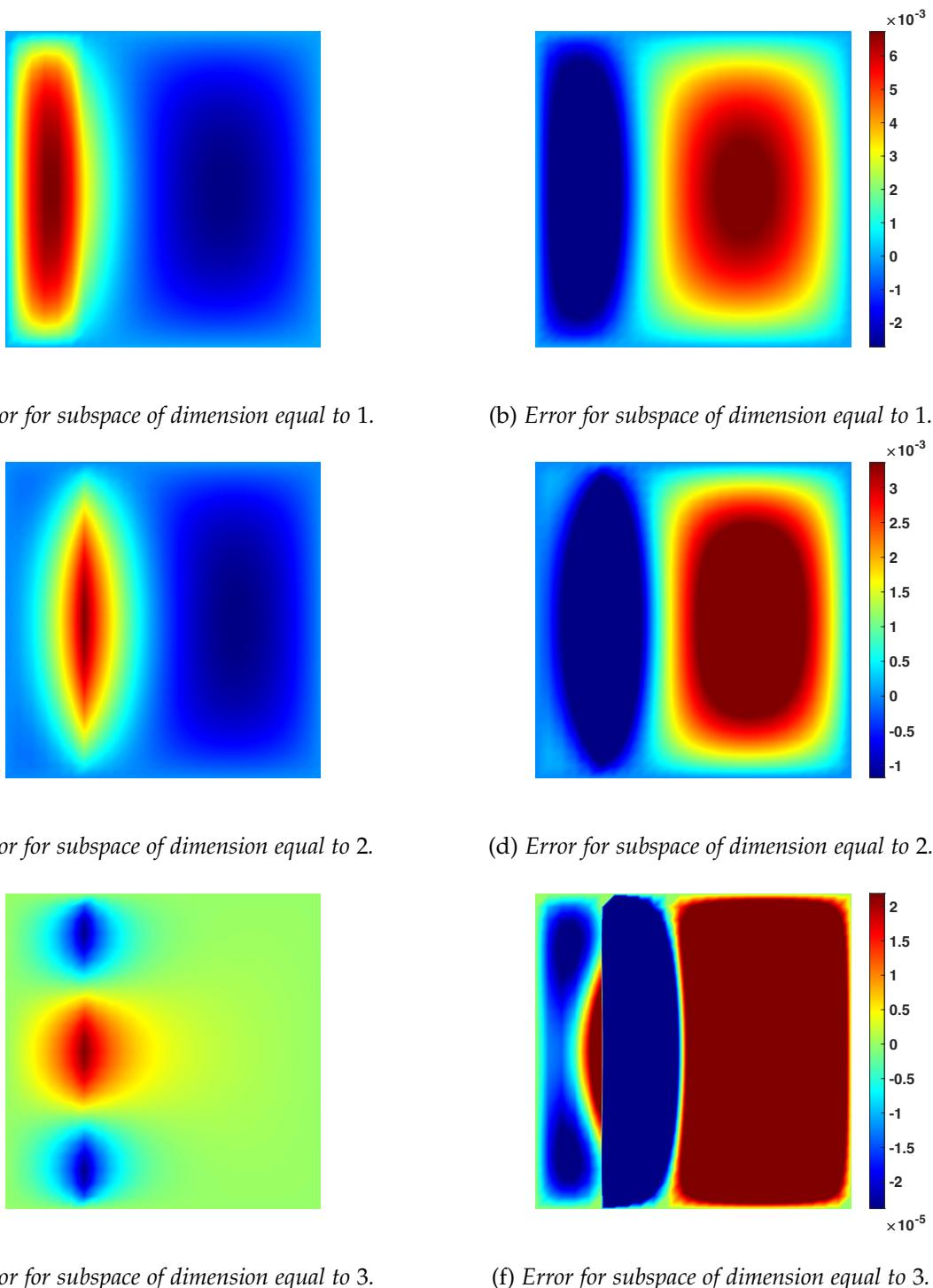
**Figure 8.1:** First 6 intrinsic PGD modes of problem (67) obtained by means of iterative Algorithm 5.



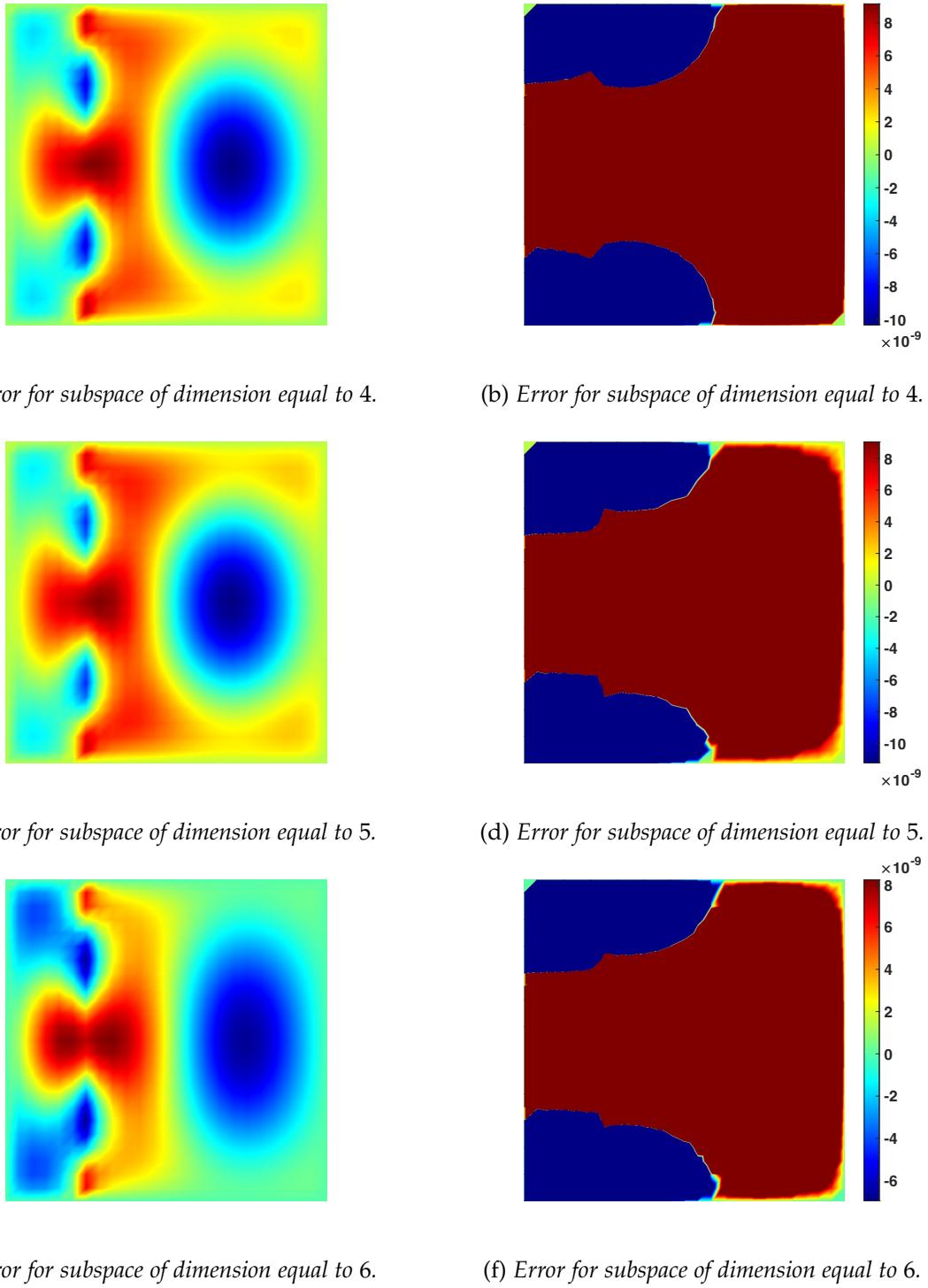
**Figure 8.2:** Convergence history of the modes  $u_i$  to  $u(\gamma)$  in the sense stated in limit (70). Comparison with the convergence history in Azaïez et al. (2020). Blue circles and red squares represent, respectively, the  $L^2$  and  $H_0^1$  norms of our methods, and blue asterisks and red crosses represent, respectively, the  $L^2$  and  $H_0^1$  norms of the PI method.

As presented before, our new procedure based on the GD to obtain the intrinsic PGD modes and a deflation algorithm to keep enriching the PGD expansion seems to perform better in terms of error decay than the procedure presented in Azaïez et al. (2020) based on the PI algorithm, achieving up to three more orders of magnitude for the error. This clearly supports the use of this procedure in order to obtain the intrinsic PGD modes of parametric symmetric problems.

This test concludes the validation of the procedure presented in Algorithm 5.



**Figure 8.3:** Comparison of the errors obtained between the reduced solutions and the parametric solution  $u(\gamma)$  for  $\gamma = 0.3$ . Errors in the left panels are the ones obtained by means of Algorithm 5 and the ones on the right panels are the ones obtained by means of the PI Algorithm presented in Azaïez et al. (2020). (Part 1: Dimensions 1 (top), 2 (middle), 3 (bottom)).



**Figure 8.4:** Comparison of the errors obtained between the reduced solutions and the parametric solution  $u(\gamma)$  for  $\gamma = 0.3$ . Errors in the left panels are the ones obtained by means of Algorithm 5 and the ones on the right panels the ones obtained by means of the PI Algorithm presented in Azaiez et al. (2020). (Part 2: Dimensions 4 (top), 5 (middle), 6 (bottom)).

## 8.2 ELASTOSTATICS PROBLEM

In this section, we will test Algorithm 5 in a linear and parametric elasticity problem, where the parameter is the relationship between the Lamé parameters. The main difference between this problem and the previous one, besides the governing PDE, is that, in this problem, the solution is vectorial, as the deformation of the blocks has two components, horizontal and vertical. The problem reads,

$$\begin{cases} -\nabla \sigma(\mathbf{u}) = \mathbf{f} & \text{on } \Omega, \\ \sigma(\mathbf{u}) \cdot \mathbf{n} = 0 & \text{on } \partial\Omega_1, \\ \mathbf{u} = \mathbf{0} & \text{in } \partial\Omega_2, \end{cases} \quad (71)$$

where  $\mathbf{f}$  is the external force,  $\mathbf{u} = (u_1, u_2)^T$  is the displacement vector,  $\sigma(\mathbf{u})$  is the stress tensor defined as

$$\sigma_{ij}(\mathbf{u}) = \delta_{ij} \nabla \cdot \mathbf{u} + \gamma \epsilon_{ij}(\mathbf{u}),$$

with  $\delta_{ij}$  the Kronecker delta,  $\gamma$  the parameter of our problem and  $\epsilon(\mathbf{u})$  the strain tensor, defined as,

$$\epsilon_{ij}(\mathbf{u}) = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right).$$

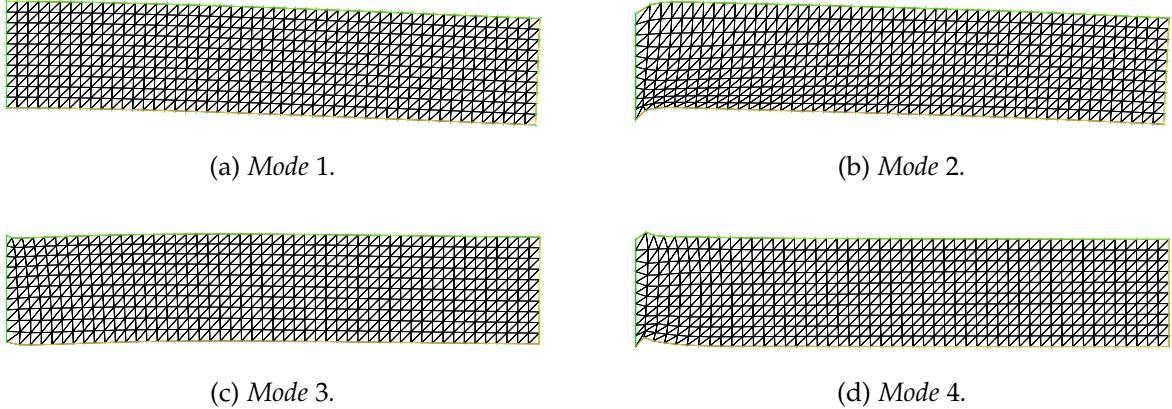
This problem could model blocks of different materials fixed on one side and under the action of gravity. It admits the following weak formulation:

$$\begin{cases} \text{Find } \mathbf{u}(\gamma) \in H_{\partial\Omega_2}^1(\Omega) \text{ such that} \\ \int_{\Omega} \sigma(\mathbf{u}) : \epsilon(\mathbf{v}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v}, \quad \forall \mathbf{v} \in H_{\partial\Omega_2}^1(\Omega), \end{cases} \quad (72)$$

where  $H_{\partial\Omega_2}^1(\Omega) := \{\mathbf{v} \in H^1(\Omega) \text{ such that } \mathbf{v}|_{\partial\Omega_2} = \mathbf{0}\}$ .

Once again, thanks to the properties of  $\sigma, \epsilon$  and  $\mathbf{f}$ , Lax-Milgram Theorem assures the existence and uniqueness of a solution of problem (72). For this test case, we set the domain as a rectangle,  $\Omega = (0, 10) \times (0, 2)$ , the only external force will be a normalized gravity  $\mathbf{f}(x, y) = (0, -10^{-3})$ , and the parameter  $\gamma \in \mathcal{D} = [0.5, 5]$  represents a relationship between the Lamé parameters. Boundary conditions impose that the rectangle is fixed on the left. As done in the previous test, the integrals on  $\mathcal{D}$  are addressed with the mid-point quadrature formula (69) for  $N = 10$  equally-spaced subintervals.

After applying Algorithm 4, we obtain the intrinsic PGD modes that can be seen in Figure 8.5. We can see that the first mode represents the main deformation of the block, the second mode represents some corrections in the borders of the block near the fixed side and modes three and four represent further corrections more localized.

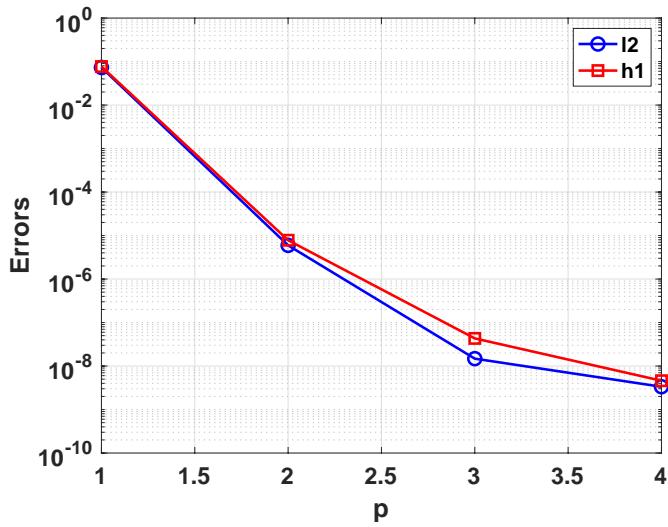


**Figure 8.5:** Intrinsic PGD modes of problem (71) obtained by means of Algorithm 5. We show the deformation of the mesh given by the modes.

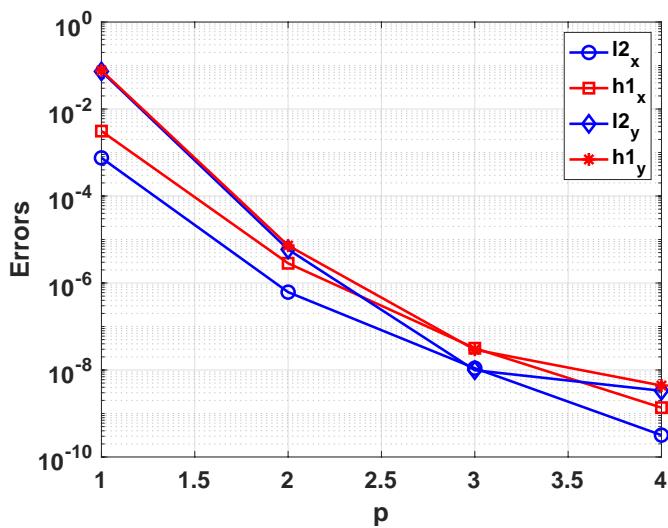
As done in the previous test, we also study the convergence of the truncated series  $u_i$  given by formula (66) to the parametric solution  $u(\gamma)$  in both  $L^2(\mathcal{D}, L^2(\Omega)^2, d\mu)$  and  $L^2(\mathcal{D}, H^1(\Omega)^2, d\mu)$  norms. We show in Figure 8.6 the truncation errors measured in those two norms in logarithm scale versus the dimension of the reduced space. As in the previous test, we can see that the error decay is very sharp and that we obtain a very good error in both norms after just a few modes.

We also present the same convergence of the truncated series for each component of the displacement vector in Figure 8.7 in both  $L^2(\mathcal{D}, L^2(\Omega), d\mu)$  and  $L^2(\mathcal{D}, H^1(\Omega), d\mu)$  norms. We can see that we seem to obtain a slightly better approximation of the  $x$  component than of the  $y$  component. Nevertheless, both components show a very efficient error decay in both norms.

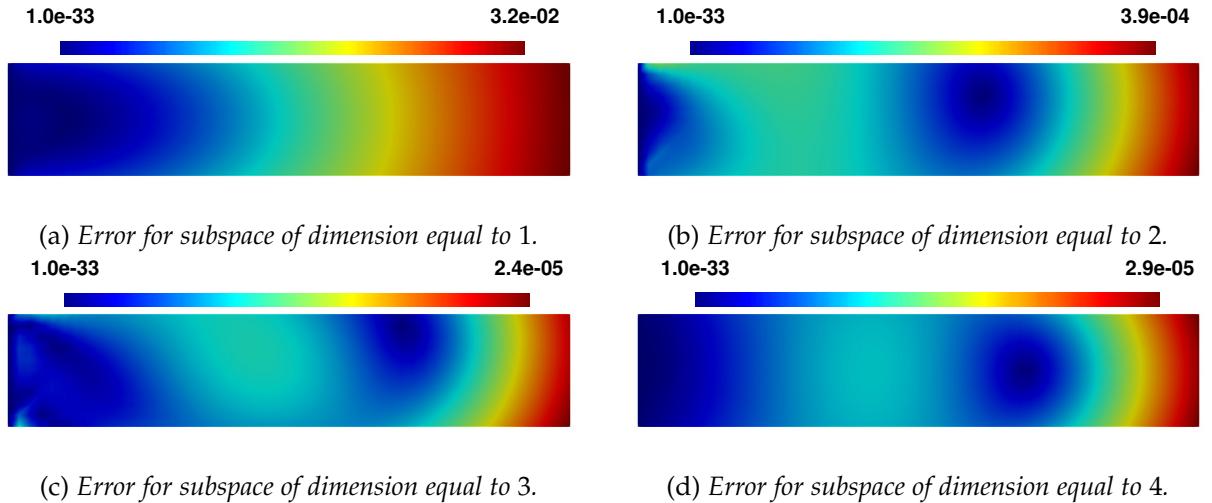
As well as done for the diffusion problem, we present in Figure 8.8, for each dimension of the subspace, the modulus of the truncation error between the reduced solution and the parametric solution of problem (71) obtained by means of Algorithm 5 in the whole domain for a fixed instance of the parameter  $\gamma = 1$ . We can clearly see that the errors are accumulated on the right part of the block, as it is the part under the most influence of the gravity. We can see that in the fourth iteration the numerical artifacts that appeared in the left part of the block after the second iteration disappear and that the maximum of the error is of order  $10^{-5}$  after the third iteration.



**Figure 8.6:** Convergence history of the modes  $u_i$  to  $u(\gamma)$  in the sense stated in limit (70). Blue circles represent the  $L^2$  norm, while red squares represent the  $H^1$  norm.



**Figure 8.7:** Convergence history of the  $x$  and  $y$  components of the modes  $u_i$  to the  $x$  and  $y$  components of  $u(\gamma)$  in the sense stated in limit (70). Blue circles and red squares represent respectively the  $L^2$  and  $H^1$  norm of the  $x$  component of the PGD expansion, and blue diamonds and red asterisks represent, respectively, the  $L^2$  and  $H^1$  norm of the  $y$  component.



**Figure 8.8:** Modulus of the errors obtained by means of Algorithm 5 for problem (71) for different optimal subspace dimensions between the reduced solution and the parametric solution for  $\gamma = 1$ . From left to right and from top to bottom, dimensions of the subspace are a) Dimension 1, b) Dimension 2, c) Dimension 3, d) Dimension 4.

These results support the use of this novel procedure presented in Algorithm 5 to obtain the intrinsic PGD modes of parametric symmetric problems.



---

## CONCLUSIONS AND OPEN PROBLEMS

---

In this part, we have developed a novel technique based on the GD on matrix spaces (Algorithm 3) to obtain the intrinsic PGD modes of parametric symmetric elliptic PDEs. Furthermore, we have also presented the one-dimensional variants of this technique (Algorithm 4) and coupled them with a deflation algorithm in order to obtain higher-dimensional optimal subspaces by enriching the subspaces (Algorithm 5). This allows us to avoid some numerical computations whose cost would be excessive.

We have also tested the proposed Algorithm 5 in two physical parametric symmetric problems. On the one hand, we have a diffusion problem that had been tested with the intrinsic PGD method before in the literature and that allowed us to validate our procedure. On the other hand, we have a classical elastostatics problem in which the solution has two components. The proposed method has been proven to produce very accurate approximations of the original parametric solutions of both problems, reaching relative errors of the order of  $10^{-8}$  after just few intrinsic PGD modes, and improving the error decay obtained by the procedures already present in the literature for the intrinsic PGD. Therefore, this novel technique based on the GD on matrix spaces is a very promising tool in the obtainment of the intrinsic PGD modes of parametric symmetric problems.

In the light of possible future areas of study, we present here some open problems that could be tackled with this procedure as a starting point:

- Extension of the presented techniques to parametric non-symmetric, and higher dimensional problems.
- Development of accelerated GD procedures, like Nesterov's algorithm and other optimization techniques, such as the Newton method in the intrinsic PGD framework.
- To couple the proposed technique with a total tensor decomposition of the solution. This would allow the calculation of high-dimensional solutions from a sequence of 1D problems.

From the research developed in this part of the thesis, we foresee the publication of a research paper Bandera Moreno et al. (2024b).



---

## BIBLIOGRAPHY

---

- Aharon, M., Elad, M., & Bruckstein, A. (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11), 4311–4322.
- Ahmed, N., Chacon Rebollo, T., John, V., & Rubino, S. (2017). A review of variational multiscale methods for the simulation of turbulent incompressible flows. *Archives of Computational Methods in Engineering*, 24, 115–164.
- Albu, L. L., Preda, C. I., Lupu, R., Dobrotă, C. E., Călin, G. M., Boghicevici, C. M., et al. (2020). Estimates of dynamics of the COVID-19 pandemic and of its impact on the economy. *Romanian Journal of Economic Forecasting*, 23(2), 5–17.
- Ali, S., Ballarin, F., & Rozza, G. (2020). Stabilized reduced basis methods for parametrized steady Stokes and Navier–Stokes equations. *Computers & Mathematics with Applications*, 80(11), 2399–2416.
- Almroth, B., Stehlin, P., & Brogan, F. (1981). Use of global functions for improvement in efficiency of nonlinear analysis. *22nd Structures, Structural Dynamics and Materials Conference*, 575.
- Almroth, B., Stern, P., & Brogan, F. A. (1978). Automatic choice of global shape functions in structural analysis. *Aiaa Journal*, 16(5), 525–528.
- Ammar, A. (2010). The proper generalized decomposition: A powerful tool for model reduction. *International Journal of Material Forming*, 3(2), 89–102.
- Ammar, A., Mokdad, B., Chinesta, F., & Keunings, R. (2006). A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. *Journal of non-Newtonian fluid Mechanics*, 139(3), 153–176.
- Anderson, S., White, C., & Farhat, C. (2022). Space-local reduced-order bases for accelerating reduced-order models through sparsity. *International Journal for Numerical Methods in Engineering*.
- Azaïez, M., Ben Belgacem, F., Casado-Díaz, J., Chacón Rebollo, T., & Murat, F. (2018). A new algorithm of proper generalized decomposition for parametric symmetric elliptic problems. *SIAM Journal on Mathematical Analysis*, 50(5), 5426–5445.

- Azaiez, M., Chacón Rebollo, T., & Gómez-Mármol, M. (2020). On the computation of Proper Generalized Decomposition modes of parametric elliptic problems. *SeMA Journal*, 77(1), 59–72.
- Bai, Z. (2002). Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Applied numerical mathematics*, 43(1-2), 9–44.
- Bak, P. (1986). The devil's staircase. *Physics Today*, 39(12), 38–45.
- Ballarin, F., Chacón Rebollo, T., Delgado Ávila, E., Gómez-Mármol, M., & Rozza, G. (2020). Certified Reduced Basis VMS-Smagorinsky model for natural convection flow in a cavity with variable height. *Computers & Mathematics with Applications*, 80(5), 973–989.
- Ballarin, F., Manzoni, A., Quarteroni, A., & Rozza, G. (2015). Supremizer stabilization of POD-Galerkin approximation of parametrized steady incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Engineering*, 102(5), 1136–1161.
- Bamani, A., & Al-Saddiki, G. (1991). Model reduction in the presence of parameter uncertainty. *Control Theory and Advanced Technology*, 7(3), 491–502.
- Bandera Moreno, A., Caravaca García, C., Chacón Rebollo, T., Delgado Ávila, E., & Gómez-Mármol, M. (2024a). Reduced Basis modelling of turbulence with well-developed inertial range. *Computer Methods in Applied Mechanics and Engineering*, 419, 116683.
- Bandera Moreno, A., Fernández-García, S., & Gómez-Mármol, M. (2024b). On the computation of intrinsic Proper Generalized Decomposition modes of parametric symmetric elliptic problems on grassmann manifolds. *Applied Mathematics and Computation*, 470, 128579.
- Bandera Moreno, A., Fernández-García, S., Gómez-Mármol, M., & Vidal, A. (2022). A multiple timescale network model of intracellular calcium concentrations in coupled neurons: Insights from ROM simulations. *Mathematical Modelling of Natural Phenomena*, 17, 11.
- Bandera Moreno, A., Fernández-García, S., Gómez-Mármol, M., & Vidal, A. (2024c). Automatic Proper Orthogonal Block Decomposition method for network dynamical systems with multiple timescales. *Communications in Nonlinear Science and Numerical Simulation*, 107844.
- Barrault, M., Maday, Y., Nguyen, N. C., & Patera, A. T. (2004). An ‘empirical interpolation’ method: Application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9), 667–672.

- Barrio, R., & Shilnikov, A. (2011). Parameter-sweeping techniques for temporal dynamics of neuronal systems: Case study of Hindmarsh-Rose model. *The Journal of Mathematical Neuroscience*, 1(1), 1–22.
- Bendokat, T., Zimmermann, R., & Absil, P.-A. (2020). A Grassmann manifold handbook: Basic geometry and computational aspects. *arXiv preprint arXiv:2011.13699*.
- Benoit, E., Callot, J. L., Diener, F., Diener, M., et al. (1981). Chasse au canard (première partie). *Collectanea Mathematica*, 37–76.
- Berkooz, G., Holmes, P., & Lumley, J. L. (1993). The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1), 539–575.
- Binev, P., Cohen, A., Dahmen, W., DeVore, R., Petrova, G., & Wojtaszczyk, P. (2011). Convergence rates for greedy algorithms in reduced basis methods. *SIAM journal on mathematical analysis*, 43(3), 1457–1472.
- Blocken, B. (2015). Computational Fluid Dynamics for urban physics: Importance, scales, possibilities, limitations and ten tips and tricks towards accurate and reliable simulations. *Building and Environment*, 91, 219–245.
- Bonaventura, L., Fernández-García, S., & Gómez-Mármol, M. (2022). Efficient implicit solvers for models of neuronal networks. *arXiv preprint arXiv:2210.01697*.
- Bordag, L. A. (2015). *Geometrical properties of differential equations: Applications of the Lie group analysis in financial mathematics*. World Scientific Publishing Company.
- Boyaval, S., Le Bris, C., Lelievre, T., Maday, Y., Nguyen, N. C., & Patera, A. T. (2010). Reduced basis techniques for stochastic problems. *Archives of Computational methods in Engineering*, 17, 435–454.
- Brezzi, F. (1974). On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *Publications des séminaires de mathématiques et informatique de Rennes*, (S4), 1–26.
- Brezzi, F., Rappaz, J., & Raviart, P.-A. (1980). Finite dimensional approximation of nonlinear problems: Part I: Branches of nonsingular solutions. *Numerische Mathematik*, 36(1), 1–25.
- Buffa, A., Maday, Y., Patera, A. T., Prud'homme, C., & Turinici, G. (2012). A priori convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM: Mathematical modelling and numerical analysis*, 46(3), 595–603.
- Busto, S., Stabile, G., Rozza, G., & Vázquez-Cendón, M. E. (2020). POD–Galerkin reduced order methods for combined Navier–Stokes transport equations based on a hybrid FV-FE solver. *Computers & Mathematics with Applications*, 79(2), 256–273.

- Byrne, Á., Ross, J., Nicks, R., & Coombes, S. (2022). Mean-field models for EEG/MEG: from oscillations to waves. *Brain topography*, 35(1), 36–53.
- Caravaca García, C. (2022). *Reduced Basis Method applied to the Smagorinsky Turbulence Model* (Doctoral dissertation). Universidad de Sevilla.
- Carrizosa, E., Guerrero, V., & Romero Morales, D. (2017). Visualizing proportions and dissimilarities by space-filling maps: A large neighborhood search approach. *Computers & Operations Research*, 78, 369–380.
- Chacón Rebollo, T., Delgado Ávila, E., & Gómez-Mármol, M. (2023). On a certified VMS-Smagorinsky reduced basis model with LPS pressure stabilisation. *Applied Numerical Mathematics*, 185, 365–385.
- Chacón Rebollo, T., Delgado Ávila, E., Gómez-Mármol, M., Ballarin, F., & Rozza, G. (2017). On a certified smagorinsky reduced basis turbulence model. *SIAM Journal on Numerical Analysis*, 55(6), 3047–3067.
- Chacón Rebollo, T., & Lewandowski, R. (2014). *Mathematical and numerical foundations of turbulence models and applications*. Springer.
- Chapelle, D., Gariah, A., & Sainte-Marie, J. (2012). Galerkin approximation with proper orthogonal decomposition : New error estimates and illustrative examples. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(4), 731–757.
- Chinesta, F., Ammar, A., & Cueto, E. (2010). Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models. *Archives of Computational methods in Engineering*, 17(4), 327–350.
- Chinesta, F., Keunings, R., & Leygue, A. (2013). *The proper generalized decomposition for advanced numerical simulations: A primer*. Springer Science & Business Media.
- Chinesta, F., Ladeveze, P., & Cueto, E. (2011). A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering*, 18(4), 395.
- Ciarlet, P. G. (2002). *The finite element method for elliptic problems*. SIAM.
- Coombes, S. (2023). Next generation neural population models. *Frontiers in Applied Mathematics and Statistics*, 9, 1128224.
- Coombes, S., & Wedgwood, K. C. (2023). Weakly coupled oscillator networks. In *Neurodynamics* (pp. 227–291). Springer.
- De Vico Fallani, F., Corazzol, M., Sternberg, J. R., Wyart, C., & Chavez, M. (2015). Hierarchy of neural organization in the embryonic spinal cord: Granger-causality graph analysis of in vivo calcium imaging data. *IEEE Trans Neural Syst Rehabil Eng*, 23 (3), 333–341.

- Delgado Ávila, E. (2018). *Development of reduced numeric models to aero-thermal flows in buildings* (Doctoral dissertation). Universidad de Sevilla.
- Deparis, S. (2008). Reduced basis error bound computation of parameter-dependent Navier–Stokes equations by the natural norm approach. *SIAM journal on numerical analysis*, 46(4), 2039–2067.
- Deparis, S., & Rozza, G. (2009). Reduced basis method for multi-parameter-dependent steady Navier–Stokes equations: Applications to natural convection in a cavity. *Journal of Computational Physics*, 228(12), 4359–4378.
- Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3), 211–218.
- Edelman, A., Arias, T. A., & Smith, S. T. (1998). The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2), 303–353.
- Egbert, G. D., & Erofeeva, S. Y. (2002). Efficient inverse modeling of barotropic ocean tides. *Journal of Atmospheric and Oceanic technology*, 19(2), 183–204.
- Erdem, T., & Keane, M. P. (1996). Decision-making under uncertainty: Capturing dynamic brand choice processes in turbulent consumer goods markets. *Marketing science*, 15(1), 1–20.
- Ermentrout, B., Pascal, M., & Gutkin, B. (2001). The effects of spike frequency adaptation and negative feedback on the synchronization of neural oscillators. *Neural computation*, 13(6), 1285–1310.
- Ermentrout, B., & Wechselberger, M. (2009). Canards, clusters, and synchronization in a weakly coupled interneuron model. *SIAM Journal on Applied Dynamical Systems*, 8(1), 253–278.
- Ersöz, E. K., Desroches, M., & Krupa, M. (2017). Synchronization of weakly coupled canard oscillators. *Physica D: Nonlinear Phenomena*, 349, 46–61.
- Etémé, A. S., Tabi, C. B., & Mohamadou, A. (2017). Long-range patterns in Hindmarsh–Rose networks. *Communications in Nonlinear Science and Numerical Simulation*, 43, 211–219.
- Falcó, A., & Nouy, A. (2011). A proper generalized decomposition for the solution of elliptic problems in abstract form by using a functional Eckart–Young approach. *Journal of Mathematical Analysis and Applications*, 376(2), 469–480.
- Farhat, C., Grimberg, S., Manzoni, A., & Quarteroni, A. (2020). Computational bottlenecks for proms: Precomputation and hyperreduction. *Model order reduction*, Berlin: De Gruyter, 181–244.

- Faugeras, O., Grimbert, F., & Slotine, J.-J. (2008). Absolute stability and complete synchronization in a class of neural fields models. *SIAM Journal on applied mathematics*, 69(1), 205–250.
- Fenichel, N. (1979). Geometric singular perturbation theory for ordinary differential equations. *Journal of differential equations*, 31(1), 53–98.
- Fernández-García, S., & Vidal, A. (2020). Symmetric coupling of multiple timescale systems with Mixed-Mode Oscillations and synchronization. *Physica D: Nonlinear Phenomena*, 401, 132129.
- Fick, L., Maday, Y., Patera, A. T., & Taddei, T. (2018). A stabilized POD model for turbulent flows over a range of Reynolds numbers: Optimal parameter sampling and constrained projection. *Journal of Computational Physics*, 371, 214–243.
- Fink, J. P., & Rheinboldt, W. C. (1983a). On the discretization error of parametrized nonlinear equations. *SIAM Journal on Numerical Analysis*, 20(4), 732–746.
- Fink, J. P., & Rheinboldt, W. C. (1983b). On the error behavior of the reduced basis technique for nonlinear finite element approximations. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 63(1), 21–28.
- FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6), 445–466.
- Fourier, J. B. J. (1888). *Théorie analytique de la chaleur*. Gauthier-Villars et fils.
- Freund, R. W. (2011). The SPRIM algorithm for structure-preserving order reduction of general RCL circuits. *Model reduction for circuit simulation*, 25–52.
- Germano, M., Piomelli, U., Moin, P., & Cabot, W. H. (1991). A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3(7), 1760–1765.
- Grepl, M. A., Maday, Y., Nguyen, N. C., & Patera, A. T. (2007). Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 41(3), 575–605.
- Gugercin, S., & Antoulas, A. C. (2004). A survey of model reduction by balanced truncation and some new results. *International Journal of Control*, 77(8), 748–766.
- Gunzburger, M. D. (1989). Finite element methods for viscous incompressible flows: A guide to theory, practice, and algorithms. *NASA STI/Recon Technical Report A*, 91, 30750.
- Haasdonk, B. (2013). Convergence rates of the POD–Greedy method. *ESAIM: Mathematical modelling and numerical Analysis*, 47(3), 859–873.

- Haasdonk, B. (2017). Reduced basis methods for parametrized PDEs—a tutorial introduction for stationary and instationary problems. *Model reduction and approximation: theory and algorithms*, 15, 65.
- Haasdonk, B., & Ohlberger, M. (2008). Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(2), 277–302.
- Hesthaven, J. S., Rozza, G., Stamm, B., et al. (2016). *Certified reduced basis methods for parametrized partial differential equations* (Vol. 590). Springer.
- Hiermaier, S. (2007). *Structures under crash and impact: Continuum mechanics, discretization and experimental characterization*. Springer Science & Business Media.
- Hijazi, S., Freitag, M., & Landwehr, N. (2023). POD-Galerkin reduced order models and physics-informed neural networks for solving inverse problems for the Navier–Stokes equations. *Advanced Modeling and Simulation in Engineering Sciences*, 10(1), 1–38.
- Hijazi, S., Stabile, G., Mola, A., & Rozza, G. (2020). Data-driven POD-Galerkin reduced order model for turbulent flows. *Journal of Computational Physics*, 416, 109513.
- Himpe, C., Leibner, T., & Rave, S. (2018). Hierarchical approximate proper orthogonal decomposition. *SIAM Journal on Scientific Computing*, 40(5), A3267–A3292.
- Hindmarsh, J. L., & Rose, R. (1984). A model of neuronal bursting using three coupled first order differential equations. *Proceedings of the Royal society of London. Series B. Biological sciences*, 221(1222), 87–102.
- Holden, L., & Erneux, T. (1993a). Slow passage through a Hopf bifurcation: From oscillatory to steady state solutions. *SIAM Journal on Applied Mathematics*, 53(4), 1045–1058.
- Holden, L., & Erneux, T. (1993b). Understanding bursting oscillations as periodic slow passages through bifurcation and limit points. *Journal of Mathematical Biology*, 31, 351–365.
- Ito, K., & Ravindran, S. S. (1998). A reduced-order method for simulation and control of fluid flows. *Journal of computational physics*, 143(2), 403–425.
- Izhikevich, E. M. (2000). Neural excitability, spiking and bursting. *International journal of bifurcation and chaos*, 10(06), 1171–1266.
- Izhikevich, E. M. (2001). Synchronization of elliptic bursters. *SIAM review*, 43(2), 315–344.

- John, V. (2003). *Large eddy simulation of turbulent incompressible flows: Analytical and numerical results for a class of LES models* (Vol. 34). Springer Science & Business Media.
- Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255–260.
- Kahlbacher, M., & Volkwein, S. (2007). Galerkin proper orthogonal decomposition methods for parameter dependent elliptic systems. *Discussiones Mathematicae, Differential Inclusions, Control and Optimization*, 27(1), 95–117.
- Kenway, G. K., & Martins, J. R. (2014). Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration. *Journal of Aircraft*, 51(1), 144–160.
- Kerschen, G., Golinval, J.-c., Vakakis, A. F., & Bergman, L. A. (2005). The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: An overview. *Nonlinear dynamics*, 41, 147–169.
- Kindratenko, V. V., Enos, J. J., Shi, G., Showerman, M. T., Arnold, G. W., Stone, J. E., Phillips, J. C., & Hwu, W.-m. (2009). GPU clusters for high-performance computing. *2009 IEEE International Conference on Cluster Computing and Workshops*, 1–8.
- Kolmogórov, A. N. (1941). The local structure of turbulence in incompressible viscous fluid for very large Reynolds number. *Doklady Akademii Nauk SSSR*, 30, 299–303.
- Komatitsch, D., Tsuboi, S., Ji, C., & Tromp, J. (2003). A 14.6 billion degrees of freedom, 5 teraflops, 2.5 terabyte earthquake simulation on the Earth Simulator. *Proceedings of the 2003 ACM/IEEE conference on Supercomputing*, 4.
- Kraichnan, R. H. (1967). Inertial ranges in two-dimensional turbulence. *The Physics of Fluids*, 10(7), 1417–1423.
- Krupa, M., Vidal, A., & Clément, F. (2013). A network model of the periodic synchronization process in the dynamics of calcium concentration in GnRH neurons. *The Journal of Mathematical Neuroscience*, 3, 1–24.
- Krupa, M., Popović, N., Kopell, N., & Rotstein, H. G. (2008). Mixed-mode oscillations in a three time-scale model for the dopaminergic neuron. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 18(1).
- Krupa, M., & Szmolyan, P. (2001). Relaxation oscillation and canard explosion. *Journal of Differential Equations*, 174(2), 312–368.

- Kunisch, K., & Volkwein, S. (2002). Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fluid Dynamics. *SIAM Journal on Numerical Analysis*, 40(2), 492–515.
- Ladyzenskaja, O. (1967). New equations for the description of motion of viscous incompressible fluids and solvability in the large of boundary value problems for them. *Proc. of the Steklov Institute of Mathematics*, 102.
- Lassila, T., Manzoni, A., Quarteroni, A., & Rozza, G. (2014). Model order reduction in fluid dynamics: Challenges and perspectives. *Reduced Order Methods for modeling and computational reduction*, 235–273.
- Lilly, D. K. (1967). The representation of small-scale turbulence in numerical simulation experiments. *Proc. IBM Sci. Comput. Symp. on Environmental Science*, 195–210.
- Maday, Y., Nguyen, N. C., Patera, A. T., & Pau, G. S. H. (2009). A general, multipurpose interpolation procedure: The magic points. *Commun. Pure Appl. Anal.*, 8(1), 383–404.
- Maday, Y., Patera, A. T., & Turinici, G. (2002). A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations. *Journal of Scientific Computing*, 17, 437–446.
- Magnus, J. R., & Neudecker, H. (2019). *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons.
- Majdabadi, M. M., Farhad, S., Farkhondeh, M., Fraser, R. A., & Fowler, M. (2015). Simplified electrochemical multi-particle model for LiFePO<sub>4</sub> cathodes in lithium-ion batteries. *Journal of Power Sources*, 275, 633–643.
- Manzoni, A. (2014). An efficient computational framework for reduced basis approximation and a posteriori error estimation of parametrized Navier–Stokes flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(4), 1199–1226.
- Mojgani, R., & Balajewicz, M. (2021). Low-rank registration based manifolds for convection-dominated PDEs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1), 399–407.
- Nagumo, J., Arimoto, S., & Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10), 2061–2070.
- Nagy, D. A. (1979). Modal representation of geometrically nonlinear behavior by the finite element method. *Computers & Structures*, 10(4), 683–688.
- Negri, F., Manzoni, A., & Rozza, G. (2015). Reduced basis approximation of parametrized optimal flow control problems for the stokes equations. *Computers & Mathematics with Applications*, 69(4), 319–336.

- Néron, D., & Ladevèze, P. (2010). Proper generalized decomposition for multiscale and multiphysics problems. *Archives of Computational Methods in Engineering*, 17, 351–372.
- Noor, A. K., & Peters, J. M. (1980). Reduced basis technique for nonlinear analysis of structures. *Aiaa journal*, 18(4), 455–462.
- Nouy, A. (2010). A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 199(23-24), 1603–1626.
- Pan, S., & Pal, J. (1995). Reduced order modelling of discrete-time systems. *Applied mathematical modelling*, 19(3), 133–138.
- Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11), 559–572.
- Pernarowski, M., Miura, R., & Kevorkian, J. (1992). Perturbation techniques for models of bursting electrical activity in pancreatic  $\beta$ -cells. *SIAM Journal on Applied Mathematics*, 52(6), 1627–1650.
- Pernarowski, M. (1994). Fast subsystem bifurcations in a slowly varying Liénard system exhibiting bursting. *SIAM Journal on Applied Mathematics*, 54(3), 814–832.
- Petersen, K. B., & Pedersen, M. S. (2008). The matrix cookbook. *Technical University of Denmark*, 7(15), 510.
- Peterson, J. S. (1989). The reduced basis method for incompressible viscous flow calculations. *SIAM Journal on Scientific and Statistical Computing*, 10(4), 777–786.
- Pinna, R. (2008). Model Reduction via Proper Orthogonal Decomposition. In W. Schilders, H. van der Vorst, & J. Rommes (Eds.), *Mathematics in industry* (pp. 95–109). Springer Berlin Heidelberg.
- Porsching, T., & Lee, M. L. (1987). The reduced basis method for initial value problems. *SIAM journal on numerical analysis*, 24(6), 1277–1287.
- Prud'Homme, C., Rovas, D. V., Veroy, K., & Patera, A. T. (2002). A mathematical and computational framework for reliable real-time solution of parametrized partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 36(5), 747–771.
- Quarteroni, A., Manzoni, A., & Negri, F. (2016). Reduced Basis Methods for Partial Differential Equations. Springer.

- Quarteroni, A., Rozza, G., & Manzoni, A. (2011). Certified reduced basis approximation for parametrized partial differential equations and applications. *Journal of Mathematics in Industry*, 1, 1–49.
- Richardson, L. F. (1922). *Weather prediction by numerical process*. University Press.
- Rubino, S. (2014). *Numerical modelling of turbulence by Richardson number-based and VMS models/Modelado numérico de la turbulencia mediante modelos de Richardson y de Multiescala Variacional* (Doctoral dissertation). Universidad de Sevilla.
- Rubinstein, R., & Clark, T. T. (2017). “Equilibrium” and “non-equilibrium” turbulence. *Theoretical and applied mechanics letters*, 7(5), 301–305.
- Ryckelynck, D. (2005). A priori hyperreduction method: An adaptive approach. *Journal of computational physics*, 202(1), 346–366.
- Sagaut, P. (2005). *Large eddy simulation for incompressible flows: An introduction*. Springer Science & Business Media.
- Semenov, D. M., & Fradkov, A. L. (2021). Adaptive synchronization in the complex heterogeneous networks of Hindmarsh–Rose neurons. *Chaos, Solitons & Fractals*, 150, 111170.
- Silva, J. M., Villena, J. F., Flores, P., & Silveira, L. M. (2007). Outstanding issues in model order reduction. *Scientific Computing in Electrical Engineering*, 139–152.
- Smagorinsky, J. (1963). General circulation experiments with the primitive equations: I. The basic experiment. *Monthly weather review*, 91(3), 99–164.
- Smolen, P., Terman, D., & Rinzel, J. (1993). Properties of a bursting model with two slow inhibitory variables. *SIAM Journal on Applied Mathematics*, 53(3), 861–892.
- Stabile, G., & Rozza, G. (2018). Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations. *Computers & Fluids*, 173, 273–284.
- Stein, E. M., & Weiss, G. (1971). *Introduction to Fourier analysis on Euclidean spaces* (Vol. 1). Princeton university press.
- Taubenblatt, M. A. (2011). Optical interconnects for high-performance computing. *Journal of Lightwave Technology*, 30(4), 448–457.
- Taylor, C. A., Hughes, T. J., & Zarins, C. K. (1998). Finite element modeling of three-dimensional pulsatile flow in the abdominal aorta: Relevance to atherosclerosis. *Annals of biomedical engineering*, 26, 975–987.
- Traub, R. D., Kopell, N., Bibbig, A., Buhl, E. H., LeBeau, F. E., & Whittington, M. A. (2001). Gap junctions between interneuron dendrites can enhance synchrony

- of gamma oscillations in distributed networks. *Journal of Neuroscience*, 21(23), 9478–9486.
- Tsai, P.-H., & Fischer, P. (2022). Parametric model-order-reduction development for unsteady convection. *Frontiers in Physics*, 10.
- Van der Pol, B. (1920). A theory of the amplitude of free and forced triode vibrations. *Radio Review*, 1, 701–710.
- Van der Pol, B. (1926). On “relaxation-oscillations”. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11), 978–992.
- Van der Pol, B., & Van Der Mark, J. (1927). Frequency demultiplication. *Nature*, 120(3019), 363–364.
- Veroy, K., Prud'Homme, C., Rovas, D., & Patera, A. (2003). A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. *16th AIAA Computational Fluid Dynamics Conference*, 3847.
- Vidal, A., Zhang, Q., Médigue, C., Fabre, S., & Clément, F. (2012). DynPeak: An algorithm for pulse detection and frequency analysis in hormonal time series. *PloS one*, 7(7), e39001.
- Volkwein, S. (2013). Proper orthogonal decomposition: Theory and reduced-order modelling. *Lecture Notes, University of Konstanz*, 4(4), 1–29.