Thèse n° 10 208

Reduced order models for coupled systems

Présentée le 22 septembre 2023

Faculté des sciences de base
Chaire de mathématiques computationnelles et science de la simulation
Programme doctoral en mathématiques

pour l'obtention du grade de Docteur ès Sciences

par

Niccolò DISCACCIATI

Acceptée sur proposition du jury

Prof. D. Kressner, président du jury
Prof. J. S. Hesthaven, directeur de thèse
Prof. A. Manzoni, rapporteur
Dr T. Taddei, rapporteur
Prof. S. Deparis, rapporteur

École
polytechnique
fédérale
de Lausanne

2023

To my parents

Acknowledgements

This work would not have been possible without the help and support of all the people who surrounded me in the last years. Everyone had an important role, but there are a few people who deserve a special mention.

First, I would like to thank my advisor Prof. Jan S. Hesthaven for welcoming me in the MCSS group as a master student first, and for giving me the invaluable opportunity to pursue my PhD studies after. I am extremely grateful for his continuous support and scientific guidance, and for sharing his thoughts and expertise on an endless list of mathematical subjects. He always allowed me to develop my ideas independently, while providing constructive feedback and new challenging topics. In parallel, he allowed me to grow not only as a researcher, but also as a person. This is not always given, and I will for sure keep him as a role model in the future.

I am also thankful to all the members of the jury, starting from Prof. D. Kressner, who kindly accepted the role of President. My sincere gratitude goes to the examiners, Prof. A. Manzoni, Dr. T. Taddei and Prof. S. Deparis, who not only took the time to examine my thesis, but also provided extremely valuable feedback, which led to a stimulating and inspiring discussion. Special thanks go to Prof. Deparis (well, Simone) for his kindness and availability throughout these years, especially when he was the Deputy Director of the Section of Mathematics.

Spending a few years in the same place means sharing experiences with several people. I start with Federico and Ricardo, who had the patience to help me clarify my doubts, especially when the submission deadline was approaching. Thank you Mariella for being office mates for several months and sharing the ups and downs of a PhD journey. Huge thanks to the best SIAM Chapter P+VP pair Riccardo and Andrea, for always being available for a quick break whenever I needed that. I would also like to thank Fabio, with whom I shared a few weekends in the office, for his continuously positive attitude. Merci Delphine pour ta gentillesse et ta disponibilité et pour avoir résolu rapidement tous les problèmes bureaucratiques qui se sont posés. I am also grateful to Agathe, whose master project results are partially included in this thesis.

Special thanks go to the ‘Limoncello and hotpot?’ members Deep, Nicolò, Qian and Zhen, for being not only colleagues but also wonderful friends. There’s an endless amount of memories

I will keep with me, including our trips to Naples, Paris and India, the ‘Matcha Tiramisù’, the hotpot dinners, the Champions League nights and the walks around Chavannes. A particular mention goes to Deep, who also had the patience to co-supervise my master thesis.

This journey would not have been possible without my flatmates. Sundar, I was lucky to share ‘my’ first apartment with you. There are many moments I will keep with me, including the IKEA furniture assembly, the discussions on the balcony while complaining about the neighbors, the Luigia and Chinese dinners, and the F1 or video games nights. Grazie Vanessa per essere sempre stata disponibile per una chiacchierata o una risata e per ascoltare le mie lamentele, specialmente durante il delirio pre-thesis-submission. I of course forgot some of the moments we spent together, but I am extremely grateful to have spent my first years of adulthood with both of you.

Merci Adeline de faire partie de ma vie. Merci pour tous les moments que nous avons vécus ces dernières années, les voyages, les week-ends, les rires, les appels et pour me rendre heureux chaque jour. Je sais que je peux compter sur toi chaque fois que j’en ai besoin. Je me réjouis déjà de commencer un nouveau chapitre de ma vie avec toi.

Infine, il più grande ringraziamento va a tutta la mia famiglia, ed in particolare a Paola, Tino, Fabio, Maria Grazia e Camilla. Anche se alcune volte non ci siamo visti o sentiti con la frequenza che avrei voluto, so che posso sempre contare su di voi e che siete sempre pronti a supportarmi. Grazie infinite per il vostro affetto e per ogni momento che abbiamo passato e passeremo insieme.

Abstract

Efficient numerical simulations of coupled multi-component systems can be particularly challenging. This is mostly due to the complexity of their solutions, as mutual interactions may cause emergent behaviors, including synchronization and instabilities. Variations in the physical parameters and the multi-query simulations arising from, e.g., control problems, pose additional challenges. In parallel, to reduce the computational complexity while preserving the accuracy of the underlying numerical discretization scheme, model order reduction techniques have proven to be effective for a plethora of models and problems. The goal of this thesis is to design model order reduction methods specifically targeted to coupled systems. This allows one to detect (and possibly discover) emergent behaviors in multi-component systems without the need to simulate the original, possibly expensive model. This work is divided into three main parts.

In the first part, we assume that the coupled system is known and the full model can be exploited to construct projection-based surrogate models. Variations in the constitutive parameters lead to qualitatively different system behaviors, that can be explored at a reduced level. In this setting, we propose and numerically validate an efficient reduction method for systems exhibiting synchronization, including phase dynamics equations and a model for circadian oscillators.

In the second part, we consider scenarios in which the full coupled model cannot be used to construct the surrogate. This is the case when one does not know how the components will be assembled at a later stage or repeated simulations of the full system are prohibitive due to the high degree of complexity of the problem. The reduced models are constructed by simulating the system components separately with a suitable (artificial) parametrization of the boundary conditions and projecting the local discretization operators. The surrogates can subsequently be used to recover the main features of the coupled system of interest under parameter variations. We first apply these techniques to an oscillatory mechanical system consisting of pendulum clocks hanging on a wooden structure (the Huygens' experiment), and we subsequently extend them to more general cases, including diffusion-reaction models and fluid-structure interaction problems.

In the third part, we aim to construct reduced models for systems whose solvers are available only as black boxes, i.e., with no access to the (local) discretization operators. We use purely data-driven interpolation methods combined with data obtained from the coupled model or an artificial parametrization, in the same spirit of the first two parts of this thesis. We show the efficiency of our method in a variety of problems, including highly heterogeneous and multi-physics models.

Keywords: Model order reduction, Coupled systems, Reduced basis method, Domain decomposition, Kernel methods, Circadian oscillators, Huygens' experiment, Multi-physics problems, Fluid-structure interaction.

Résumé

Les simulations numériques efficaces de systèmes couplés à composants multiples peuvent s'avérer particulièrement difficiles. Cela est principalement dû à la complexité de leurs solutions, car les interactions mutuelles peuvent entraîner des comportements émergents, y compris la synchronisation et les instabilités. Les variations des paramètres physiques et les simulations multi-requêtes découlant, par exemple, de problèmes de contrôle, posent des défis supplémentaires. En parallèle, pour réduire la complexité computationnelle tout en préservant la précision du schéma de discrétisation numérique sous-jacent, les techniques de réduction d'ordre de modèle se sont révélées efficaces pour une multitude de modèles et de problèmes. L'objectif de cette thèse est de concevoir des méthodes de réduction d'ordre de modèle spécifiquement destinées aux systèmes couplés. Cela permet ainsi de détecter (et éventuellement de découvrir) des comportements émergents dans des systèmes à composants multiples sans avoir besoin de simuler le modèle original, qui peut être coûteux. Ce travail est divisé en trois parties principales.

Dans la première partie, nous supposons que le système couplé est connu et que le modèle complet peut être exploité pour construire des modèles de substitution basés sur des projections. Les variations des paramètres constitutifs conduisent à des comportements qualitativement différents du système, qui peuvent être explorés à un niveau réduit. Dans ce cadre, nous proposons et validons numériquement une méthode de réduction efficace pour des systèmes présentant une synchronisation, y compris des équations de dynamique des phases et un modèle pour des oscillateurs circadiens.

Dans la deuxième partie, nous considérons des scénarios dans lesquels le modèle couplé complet ne peut pas être utilisé pour construire le substitut. C'est le cas lorsqu'on ne sait pas comment les composants seront assemblés à un stade ultérieur ou lorsque des simulations répétées du système complet sont prohibitives en raison du degré élevé de complexité du problème. Les modèles réduits sont construits en simulant les composants du système séparément avec une paramétrisation (artificielle) appropriée des condition aux limites et en projetant les opérateurs de discrétisation locaux. Les substituts peuvent ensuite être utilisés pour retrouver les principales caractéristiques du système couplé d'intérêt en fonction des variations des paramètres. Nous appliquons d'abord ces techniques à un système mécanique oscillatoire composé de pendules suspendues à une structure en bois (l'expérience de Huygens), puis nous les étendons à des cas plus généraux, y compris des modèles de diffusion-réaction et des problèmes d'interaction fluide-structure.

Dans la troisième partie, nous visons à construire des modèles réduits pour les systèmes dont les solveurs sont disponibles uniquement sous forme de boîtes noires, c'est-à-dire sans accès aux opérateurs de discrétisation (locaux). Nous utilisons des méthodes d'interpolation entièrement guidées par les données, combinées avec des données obtenues à partir du modèle couplé ou d'une paramétrisation artificielle, dans le même esprit que les deux premières parties de cette thèse. Nous montrons l'efficacité de notre méthode dans une variété de problèmes, y compris des modèles hautement hétérogènes et multi-physiques.

Mots-clés : Réduction d'ordre de modèle, Systèmes couplés, Méthode des bases réduites, Décomposition de domaine, Méthodes à noyaux, Oscillateurs circadiens, Expérience de Huygens, Problèmes multi-physiques, Interaction fluide-structure.

Contents

Acknowledgements **i**

Abstract (English/Français) **iii**

1 Coupled systems **1**

1.1 The importance of coupled systems 1

1.2 Model reduction of coupled systems 4

1.3 Outline 6

2 An introduction to reduced order modeling **9**

2.1 The reduced basis method 9

2.1.1 Formulation 10

2.1.2 Galerkin approximation 10

2.1.3 Basis construction 11

2.1.4 Computational efficiency 12

2.1.5 Beyond reduced basis 14

2.2 Non-intrusive surrogate modeling 15

2.2.1 Kernel interpolation methods 16

2.2.2 Artificial Neural Networks 18

2.2.3 Data preprocessing 20

3 Globally coupled oscillatory systems **21**

3.1 Globally coupled oscillatory systems 22

3.1.1 Phase dynamics 22

3.1.2 Circadian oscillators 24

3.2 Model reduction 26

3.3 Numerical results 30

3.3.1 Phase dynamics 30

3.3.2 Circadian oscillators 41

3.3.3 A control problem 46

3.4 Extension: sensitivity analysis 47

3.5 Conclusion 50

4 The Huygens' problem **53**

4.1 Mathematical model 54
  4.1.1 Structure 54
  4.1.2 Pendula 56
  4.1.3 The Huygens' model 57
  4.1.4 More pendula 59
4.2 Localized model reduction 59
  4.2.1 Component Mode Synthesis 59
  4.2.2 Proper Orthogonal Decomposition 60
4.3 Numerical results 61
4.4 Extensions 66
  4.4.1 Parameter-dependent structure 66
  4.4.2 Beyond Huygens: parametrized coupling 67
4.5 Conclusion 68
4.A Finite Element discretization 69

5 Localized model reduction and domain decomposition methods 73
5.1 Problem formulation 75
5.2 Localized model reduction 77
  5.2.1 Offline phase 78
  5.2.2 Online phase 80
  5.2.3 Error analysis 81
  5.2.4 Computational cost 83
5.3 Numerical results 85
  5.3.1 Advection-diffusion-reaction 85
  5.3.2 Checkerboard diffusion 89
  5.3.3 Multiple subdomains 93
  5.3.4 FitzHugh-Nagumo 95
  5.3.5 Stokes-Darcy coupling 98
  5.3.6 Fluid-Structure Interaction I 102
  5.3.7 Fluid-Structure Interaction II 105
5.4 Conclusion 110

6 Non-intrusive approximations of the boundary response maps 113
6.1 Problem formulation 115
6.2 Non-intrusive model reduction 118
  6.2.1 Approximating boundary maps 118
  6.2.2 Error analysis 121
  6.2.3 Computational cost 122
6.3 Numerical results 122
  6.3.1 Diffusion on a Cartesian geometry 122
  6.3.2 Diffusion on a circular geometry 127
  6.3.3 Fluid-Structure Interaction 132
  6.3.4 A time-dependent problem 137

6.4 Conclusion 142
6.A A detailed error analysis 143

7 Conclusion and future perspectives **147**

Bibliography **151**

Curriculum Vitae **163**

1 Coupled systems

1.1 The importance of coupled systems

Several natural and artificial processes can be described as coupled systems, as they are a result of the interactions among a number of different components.

Concretely, one could find examples of such systems in every scientific field. Arguably, the main one is biology, in which networks of coupled oscillators are at the core of human life. Neuronal cells in the human brain can be thought as simple oscillatory units, which can start to interact and behave in a coherent way. As a result, our body adapts its functions to the external day-night cycle and is able to optimize many processes accordingly. Without this, the growth of human beings, as well as the correct functioning of the entire body, could be severely impacted. At a larger scale, one of the most important coupled systems in the human body is the cardiorespiratory one. The heart and the lungs have physiological functions of great importance for human health, and it is crucial to correctly understand their interactions from a biological and a mathematical level, especially to predict diseases.

Similar interacting phenomena can be found in physical systems, both at micro and macro scales. For example, Josephson junctions can be thought as combinations of superconducting and nonsuperconducting materials, in which electrons can create a tunneling effect [136, 96]. A coherent motion can be observed even if the junctions are not equal, especially if an array-like structure is considered. On the other side of the spectrum, an example of a large-scale coupled system is the weather. The interactions between the atmosphere, the oceanic waters and surface lands give rise to, e.g., winds and waves, and ultimately lead to a chaotic system.

Large-scale mechanical structures can be viewed as a combination of smaller blocks of similar nature, which can be coupled in a variety of ways to construct buildings. The coupling propagates the load and the stresses throughout the system and makes the entire construction (un)stable. Similarly, multi-component mechanical systems are quite common. Any vehicle, including cars, airplanes and rockets, can be viewed as the result of the interactions between its constitutive components, that are governed by different physical phenomena. For instance, the mechanical structure interacts with the engine and the electrical components, and responds to external conditions in a variety of ways.

Similar to the biological world, chemical oscillations can be observed in a certain class of reactions [127, 92]. A complex mixture of reacting chemical compounds interact and can give rise to limit-cycle, periodic-like behaviors.

Finally, coupled systems can be found at the level of social sciences. Social networks are the result of people communicating at different levels and scales, including local interactions and large-scale connections. Recently, it has been shown [120] that there is a nontrivial interaction between a number of social variables, democracy and higher emancipation of citizens. Ultimately, this has economic implications, as all these variables can be related to economic growth and the GDP.

The common denominator among all such systems is that the interactions lead to possibly unexpected behaviors, that can be hardly deduced by looking at the single, individual components in a separate, decoupled way. Indeed, certain properties can only be retrieved or predicted by appropriately studying the interaction mechanism. In a broad sense, this is the definition of *emergent behavior*, as *the whole is greater than the sum of the parts*.

Several phenomena fall under the umbrella of emergent properties and once again they appear in a variety of fields and applications. The most common example of emergent behavior is the synchronization. This can be defined as the adjustment of rhythms of oscillating objects due to their (weak) interactions [101]. Historically, the first example of synchronization is the sympathy of two pendulum clocks observed by C. Huygens in 1665 [72]. After hanging two pendula on the same wooden beam with an initial, possibly random, phase shift, he observed that their oscillations ultimately coincided perfectly, with a movement in opposite directions. He correctly conjectured that the driving factor for synchronization was an imperceptible motion of the beam. More recently, there has been a significant number of attempts to recreate the conditions observed by Huygens [100]. This led to the discovery of other types of synchronization patterns, including in-phase synchronization and clustering [34].

Going beyond the adjustment of rhythms, other emergent behaviors can be observed in oscillating systems. Examples include oscillations death, in which a portion of the oscillators entirely dissipates the energy, and chimera states, in which coherent and incoherent states coexist [132].

Although we introduced these phenomena in the context of mechanical oscillators, a qualitatively similar behavior is observed in networks of neuronal oscillators. This has even more concrete applications, as synchronization in the human cells is responsible for the *circadian rhythm*. Even if the single neurons can have a different dynamics, their interaction leads to a synchronous motion and the adaptation to the sleep-wake cycle that repeats roughly every 24 hours. Synchronization at a neuronal level and among the different systems, including the cardiovascular and respiratory ones, is crucial for the correct functioning of our body.

Deviations from this behavior can cause imbalance and, in extreme cases, heart failures and deaths.

Similarly, many natural phenomena are a result of synchronization among interacting agents. For instance, fireflies are known to emit rhythmic light pulses to attract females. Synchrony is often observed, and it is conjectured that they flash in unison possibly to optimize courtship communication with grounded females [115, 116]. Another example can be found in bird flocking, in which any bird within the flying murmuration is perfectly synced with its seven closest birds [142].

Synchronization and adjustment of rhythms are not the only examples of emergent behavior. Any macroscopic phenomenon resulting from microscopic interactions falls into this category. For instance, the laws of thermodynamics can be deduced from modeling the local interactions among particles. Such microscopic laws are rather complex but reversible in time, whereas the macroscopic behavior can be described with relatively simple models and it is typically irreversible. This difference is at the core of emergent behavior, as the global system possesses different properties than its constituents.

Sticking to the physics framework, a number of instabilities can be cast as emergent behavior. For instance, superimposing two fluids with different velocities at the interface can result in the formation of wave-like rolling shapes, known as Kelvin-Helmholtz instability. Looking at the two fluids individually, there would be no reason to expect such a behavior, as the two components are naturally stable. It is the interactions and the physical properties of the fluids that cause the instability of the interface and the entire system. Another typical example is the resonance, in which an increase in the amplitude of an oscillating system is observed, provided that the interaction with the environment through an applied load happens at a certain frequency. Again, in the absence of external loads, the system is stable, and the unexpected behavior is induced by the forcing term.

Similarly, fluid-structure interaction problems can have properties that can hardly be observed if the local systems are considered independently. For instance, the interactions between a confined fluid and an elastic object cause a deformation of the latter, which can evolve towards nontrivial shapes. In certain cases, a complex dynamic behavior is observed, with large oscillations of the object [129].

We finally mention the Turing patterns arising mostly in biological settings [130]. In reaction-diffusion systems, nontrivial spatial patterns can emerge from random conditions, ranging from simple shapes, such as squares or stripes, to more complex ones such as labyrinths [4]. This behavior is caused by the interplay between diffusion and reaction effects at the chemical level, where the former is typically, and unexpectedly, responsible for the instability.

Naturally, one wonders what are the reasons for these nontrivial solution behaviors. The general answer is both the coupling functions and the interaction mechanisms, although this happens in several ways.

Firstly, through the functional form of the coupling term. Depending on its properties, including symmetries and zeros, different phenomena can emerge. As a notable example, in the case of phase dynamics equations [91], an odd function may induce synchronization, with different patterns observed according to the number of harmonics included [79].

Secondly, through the connectivity. The components may interact in a global way, meaning that there is no preferential coupling direction, and the coupling term is the same for all the components. This is typically the case in neuronal oscillators and circadian clock models [80]. Another option includes a local coupling, in which only a certain number of neighboring components are connected to each other. Here, local attractors, intermittency and a chaotic behavior can appear on top of more classical synchronized states [126]. Similarly, modular or community structures can lead to incoherence, local synchrony, and global synchrony [118]. In the partial differential equations framework, a particular type of local interaction is the coupling through an interface. The exchange of information happens only through the boundary of a given domain, which can act as both an input and an output source for the system. This coupling type is quite common in practical applications, including structural dynamics, multi-component mechanical systems and multi-physics problems. It also arises when a global problem is decomposed into local subproblems defined on smaller nonoverlapping subdomains.

Thirdly, and arguably the most relevant one for this work, through the values of the physical parameters of the system. Depending on both the local parameter values and the coupling strengths, which govern component-dependent physical phenomena and the intensity of the interactions, respectively, different physical phenomena can be observed. Emergent behaviors often appear after a suitable coupling threshold is reached, after which new equilibrium states appear or the stability patterns change. In this sense, emergent behavior can be related to bifurcating phenomena.

1.2 Model reduction of coupled systems

Given the intrinsic complexity of coupled systems, determining the existence of emergent behavior at an analytical level can be prohibitive, and a numerical approach should be preferred. At this numerical level, in order to well approximate the solution of a system of interest, a high level of accuracy is essential. This can typically be addressed by simulating large-scale networks of ordinary differential equations or performing a spatial discretization of partial differential equations with a large number of degrees of freedom. Additionally, in the context of parametrized systems, several practical problems require repeated simulations of the same differential model, albeit with different parameter values. This is the case when one wishes to predict and discover emergent behaviors, as they often appear for certain parameter values only. Other typical examples include optimization, control, design and uncertainty quantification. Consequently, as the associated computational cost scales proportionally to the size of the system and the number parameter instances, running multi-query simulations becomes critical. Although the increase in computing power of modern computers can mitigate this burden, the real-time results often required in practical applications cannot be obtained, especially if the system has a high level complexity.

A possible solution aims at designing a surrogate model of a given system, which can be simulated in place of its original counterpart. This approximation should satisfy two main desiderata:

- Accuracy. The surrogate should provide an approximation which is sufficiently close to the true model output, ideally with a control on the error.
- Speedup. Simulations of the approximated model need to be significantly faster than the original one, in order to run multi-query simulations within a reasonable computational time.

This is the main goal of (data-driven) *model order reduction* techniques, which aim to control the trade-off between these two points. Recently, there has been an explosion in the number of such techniques, which can be broadly divided into *intrusive* and *non-intrusive* methods. The former aim to approximate the differential problem by projecting the original model onto a lower-dimensional space which retains the main features of the solution and the dynamics. The latter use only input-output pairs combined with interpolation or regression methods to accurately approximate the system response. The joining link is the use of the original, expensive model to construct the surrogate, which can then be queried for a much larger set of parameter values with a high level of accuracy and a controlled computational cost.

Coupled systems clearly fall into the framework of large-scale parametrized models. Being networks of multiple interacting components, the computational cost can quickly become intractable. This is particularly true when the number of components is large, the physical nature of the problem is complex, or the discretization algorithm has a high computational complexity. Moreover, parameters can play a significant role, as they are often responsible for the emergence of a particular pattern. Motivated by the need to develop accurate and computationally fast methods for parameterized coupled systems, the main goal of this work is to design model reduction techniques specifically targeted to such models. These surrogates can be used, for instance, to detect and discover emergent phenomena and to speedup optimization problems. Three scenarios are considered.

- Firstly, we assume that the coupled problem is available, i.e., we have complete knowledge of the mathematical model of the entire system. Additionally, we have full access to the underlying implementation and the numerical software. The associated solver can either consist of a single global solver (in the spirit of monolithic schemes) or be given as a combination of local solvers (in the spirit of segregated/partitioned approaches), depending on the specific application. Typically, the former is used for single-physics problems while the latter is mostly targeted to multi-physics applications, but many exceptions can be found.

Arguably, most of the classical model reduction techniques are developed in this framework, aiming to replace the original model with a surrogate obtained by projecting the operators onto a lower-dimensional space. We borrow ideas from these methods to construct the reduced models for our problems of interest, taking particular care of ensuring the desired computational speedup. We show applications of this technique to networks of limit-cycle oscillators using both the phase dynamics equations and a model of circadian oscillators.

- However, in many cases one might not have access to the coupled problem when constructing the surrogate model. Indeed, one may only have access to local, independent, component-wise models, without knowing how the components will be assembled in future stages. On a similar line, even if one has access to the coupled system, its repeated simulations required to generate the reduced model may not be computationally feasible. This is particularly true in the case of highly nonlinear models, high-dimensional parameter spaces or large-scale systems, as the resulting complexity can be extremely large.

  Thus, our goal is to build local, component-wise reduced models that are robust with respect to the possible ways to construct the coupled system. This is done by considering the components independently and introducing an artificial parametrization of the interface boundary conditions. We show applications to a wide range of scenarios, including nonlinear oscillatory systems inspired by the Huygens' experiment, diffusion-reaction models, and fluid-structure interaction problems.

- Finally, inspired by multi-physics applications and partitioned approaches, we consider the case in which the components are modeled with local solvers, that we assume to be given as black boxes only. The reduced model can be constructed by relying on the coupled problem, i.e., the combination of the local solvers, or by artificial parametrization, when the coupled problem cannot be simulated.

  As the implementation is hidden to the user, we can only rely on input-output pairs to construct the surrogate. This is mainly done using interpolation methods, although regression techniques are also employed. We again show the applicability of our method to a variety of problems featuring nonlinear components or a high degree of heterogeneity.

1.3 Outline

The remainder of this thesis is mainly structured following the subdivision that we have just presented, and it is summarized in Table 1.1. In Chapter 2 we provide a general introduction to the model reduction techniques that are used in this work. Then, in Chapter 3 we focus on model reduction of (globally coupled) oscillatory systems, in which the full coupled model is available. Chapter 4 and Chapter 5 present two classes of spatially localized model reduction techniques, targeted to the Huygens' problem and general partial differential equations coupled through interfaces, respectively. Finally, in Chapter 6 we discuss a non-intrusive approach

	Coupled model available	Coupled model unavailable (only local, independent models available)
Intrusive model reduction	Chapter 3	Chapter 4 and Chapter 5
Non-intrusive model reduction	Chapter 6	

Table 1.1 – Outline of this thesis.

based on the boundary response maps and its application to coupled heterogeneous systems. Some concluding remarks and future perspectives can be found in Chapter 7.

2 An introduction to reduced order modeling

This chapter introduces the main model order reduction techniques that we employ in this work. The first part is devoted to intrusive methods, and specifically the reduced basis method. The second part contains instead a discussion on data-driven non-intrusive techniques. The focus is on the features of these methods and on the notation, while concrete applications will be discussed in the dedicated chapters. Moreover, we focus on time-dependent problems, although the discussion holds similarly for the static case.

Consider a parametrized dynamical system of the form

$$\begin{cases} \frac{d}{dt}\mathbf{x}(t,\mu) = \mathbf{f}(t,\mathbf{x},\mu), \quad t \in [0,T], \\ \mathbf{x}(0,\mu) = \mathbf{x}_0(\mu), \end{cases} \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^N$ is the state vector, $\mu \in \mathscr{P} \subset \mathbb{R}^P$ is a vector containing the physical parameters, and $\mathbf{f} \in \mathbb{R}^N$ is the function characterizing the system dynamics. Such systems typically arise in mathematical models of time-varying phenomena, where the solution rate-of-change in time is a function of the system variables. In the case of partial differential equations (PDEs), (2.1) can be viewed as the spatially semi-discretized model obtained from classical discretization methods, such as finite differences or finite elements.

2.1 The reduced basis method

This section summarizes the reduced basis method and its variants in a concise and practical way. More details can be found in, e.g., [103, 70].

2.1.1 Formulation

We define the solution manifold $\mathscr{M}$ as the set of solutions of (2.1) under variation of the parameters in $\mathscr{P}$ and time, i.e.,

$$\mathscr{M} = \left\{\mathbf{x}(t,\mu) | \mu \in \mathscr{P}, t \in [0,T]\right\} \subset \mathbb{R}^N. \tag{2.2}$$

The reduced basis method relies on the assumption that $\mathscr{M}$ is low dimensional and can be well approximated by the span of an appropriately chosen basis $\{\mathbf{u}_i\}_{i=1}^k$ of cardinality k. This notion of reducibility is rigorously encoded by the Kolmogorov k-width, which quantifies the worst-case error arising from the projection onto the best-possible linear subspace of dimension k. If it decays rapidly with k, only a few basis functions will suffice to get a high level of accuracy, and the reduced basis methods will exhibit its full potential. Given such a basis, we approximate the solution as

$$\mathbf{x}(t,\mu) \approx \hat{\mathbf{x}}(t,\mu) = U\alpha(t,\mu) = \sum_{i=1}^{k} \alpha_i(t,\mu)\mathbf{u}_i, \tag{2.3}$$

where $\alpha \in \mathbb{R}^k$ denotes the vector of the latent coordinates $\{\alpha_i\}_{i=1}^k$, and

$$U = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \in \mathbb{R}^{N\times k}$$

is a matrix that contains the basis vectors as columns. The value of k will identify the dimension of the reduced model, consistent with (2.3). Now, we need to specify how to compute the basis and the coefficients in (2.3), whose choice is at the core of the different model reduction techniques.

2.1.2 Galerkin approximation

To construct an evolution equation for α, we replace $\mathbf{x}$ in (2.1) with its approximation $\hat{\mathbf{x}}$ to obtain

$$U\frac{d}{dt}\alpha(t,\mu) = \mathbf{f}(t, U\alpha(t,\mu), \mu) + \mathbf{r}, \tag{2.4}$$

where $\mathbf{r}$ denotes the residual due to the approximation (2.3). As the system (2.4) is overdetermined, its dynamics can be closed by a Petrov-Galerkin projection. Considering a basis $W \in \mathbb{R}^{n\times k}$ such that the matrix $W^T U$ is invertible, the reduced equation is found by imposing the residual vector to be orthogonal to W. This yields

$$\frac{d}{dt}\alpha(t,\mu) = (W^T U)^{-1} W^T \mathbf{f}(t, U\alpha, \mu). \tag{2.5}$$

Unless stated otherwise, in this work we restrict to the Galerkin framework, where an orthogonal basis U is considered, and W is equal to U. Thus, (2.5) becomes

$$\begin{cases} \frac{d}{dt}\alpha(t,\mu) = U^T\mathbf{f}(t,U\alpha,\mu), & t \in [0,T], \\ \alpha(0,\mu) = \alpha_0(\mu) = U^T\mathbf{x}_0. \end{cases} \tag{2.6}$$

By far, the Galerkin projection is the most popular choice in the reduced basis community, and works well in a very large number of cases, including the ones presented in this work. A Petrov-Galerkin strategy is typically used when the standard Galerkin projection fails, e.g., in terms of loss of stability or accuracy.

2.1.3 Basis construction

To construct the basis $\{\mathbf{u}_i\}_{i=1}^k$ different approaches can be considered, among which one of the most widely used is the Proper Orthogonal Decomposition (POD). Assuming that the solution $\mathbf{x}$ of (2.1) is available at appropriately chosen times $\{t_i\}_{i=1}^p$ and parameter instances $\{\mu_j\}_{j=1}^q$, we define the snapshot matrix as the matrix having such solutions as columns, i.e.,

$$S = \left[\mathbf{x}(t_1,\mu_1),\ldots,\mathbf{x}(t_p,\mu_1),\ldots,\mathbf{x}(t_1,\mu_q),\ldots,\mathbf{x}(t_p,\mu_q)\right] \in \mathbb{R}^{N\times pq}. \tag{2.7}$$

The POD basis U minimizes the projection error of the snapshots onto the reduced space, i.e., it is the solution to the optimization problem

$$\begin{aligned} &\underset{V\in\mathbb{R}^{N\times k}}{\arg\min} \left\| S - VV^TS \right\|_F, \\ &\text{subject to } V^TV = \mathbb{I}_k, \end{aligned} \tag{2.8}$$

where $\|\cdot\|_F$ is the Frobenius norm and $\mathbb{I}_k$ is the identity matrix of dimension k. The Eckart-Young theorem states that the solution of (2.8) is given by the first k left singular vectors of S. In particular, if

$$S = U_S\Sigma_S V_S^T, \quad \text{with } \Sigma_S = \operatorname{diag}(\sigma_i) \in \mathbb{R}^{N\times pq}$$

is the Singular Value Decomposition (SVD) of S, then

$$\begin{aligned} U = U_S(:,1:k) &= \underset{V\in\mathbb{R}^{N\times k}}{\arg\min} \left\| S - VV^TS \right\|_F, \\ \left\| S - UU^TS \right\|_F^2 &= \sum_{i=k+1}^{\min(N,pq)} \sigma_i^2, \\ U^TU &= \mathbb{I}_k. \end{aligned} \tag{2.9}$$

A practical problem arising in many applications is the choice of k, whose value represents a compromise between accuracy and computational efficiency. A simple criterion, inspired by

(2.9), relies on the relative energy retained by the first k modes, defined as

$$E(k) = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{\min(N,pq)} \sigma_i^2}. \tag{2.10}$$

Here, one selects the reduced dimension as the smallest value of k satisfying $E(k) \geq 1 - \epsilon_{POD}$, where ϵ_{POD} is a user-defined tolerance.

Several alternatives exist to construct the reduced basis. A valid one is based on a greedy algorithm, with the goal of iteratively adding basis functions by selecting the regions in which the reconstruction error or an appropriate estimator is maximal [16, 14]. Combinations of POD and greedy strategies are also used [68, 67].

2.1.4 Computational efficiency

The presented description is formally sufficient to construct the reduced model, but particular care should be given to achieving the desired speedup. Naive simulations of (2.6) require the reconstruction of the solution $U\alpha$, the evaluation of the function $\mathbf{f}$, and the projection onto the low-dimensional space $U^T\mathbf{f}(U\alpha)$. The resulting computational cost is larger than its full order counterpart, destroying all the benefits of the dimensionality reduction. The solution to this problem relies on a splitting into a computationally expensive offline phase, to be done once and for all, and a cheap online evaluation, which gives the desired speedup under parameter variations.

The basis computation can be done in the former, independently of (2.6). Thus, we can focus on the efficient time integration of the reduced system with a given U. Assume that the function $\mathbf{f}$ can be decomposed in its linear and nonlinear parts as

$$\mathbf{f}(t,\mathbf{x},\mu) = \mathbf{f}_{\text{lin}}(t,\mathbf{x},\mu) + \mathbf{f}_{\text{nonlin}}(t,\mathbf{x},\mu) = \sum_{q=1}^{Q} \Theta_q(t,\mu)\mathscr{L}_q\mathbf{x} + \mathbf{f}_{\text{nonlin}}(t,\mathbf{x},\mu), \tag{2.11}$$

where $\mathscr{L}_q$ are parameter-independent, constant-in-time, linear operators, $\Theta_q(t,\mu)$ are suitable functions of the parameters and time, and $\mathbf{f}_{\text{nonlin}}$ is a general nonlinear term. Using (2.11), the right-hand-side of (2.6) reads

$$U^T\mathbf{f}(t,U\alpha,\mu) = \sum_{q=1}^{Q} \Theta_q(t,\mu)U^T\mathscr{L}_qU\alpha + U^T\mathbf{f}_{\text{nonlin}}(t,U\alpha,\mu). \tag{2.12}$$

The operators $U^T\mathscr{L}_qU \in \mathbb{R}^{k\times k}$ can be precomputed in the offline phase, and the online evaluation of the reduced linear part becomes independent of the full dimension N. This concludes the discussion related to the linear term, as the desired speedup is obtained, at least when the problem is reducible ($k \ll N$). Conversely, the evaluation of the nonlinear part does not allow this splitting, and the reduced model suffers again from a high computational cost.

To overcome this, several methods have been developed [10, 5, 123, 50, 141, 107], usually referred to as *hyper-reduction* techniques. Unless stated otherwise, in this work we rely on the Discrete Empirical Interpolation Method (DEIM) [26], whose main idea is to evaluate only a selected number of components of the nonlinearity. We seek to approximate the nonlinear function as a linear combination of suitable basis vectors $\{\mathbf{v}_i\}_{i=1}^{\tilde{k}}$ as

$$\mathbf{f}_{\text{nonlin}}(t, U\alpha, \mu) \approx \hat{\mathbf{f}}(t, U\alpha, \mu) = V\mathbf{c}(t, U\alpha, \mu), \tag{2.13}$$

where the vector $\mathbf{c} \in \mathbb{R}^{\tilde{k}}$ contains the coordinates of the approximation in the basis $V = \left[\mathbf{v}_1, \dots, \mathbf{v}_{\tilde{k}}\right] \in \mathbb{R}^{N \times \tilde{k}}$. To compute it, we select $\tilde{k}$ rows from the overdetermined system (2.13) and solve the resulting square linear system. Specifically, consider the matrix

$$P = \left[e_{p_1}, \dots, e_{p_{\tilde{k}}}\right],$$

where $\left\{p_1, \dots, p_{\tilde{k}}\right\}$ are $\tilde{k}$ indices selected from $\{1, \dots, N\}$ and e_{p_i} is the p_i-th column of $\mathbb{I}_N$. Assuming that the matrix $P^T V$ is invertible, the vector $\mathbf{c}$ is found by solving the linear system

$$P^T \mathbf{f}_{\text{nonlin}}(t, U\alpha, \mu) = (P^T V)\mathbf{c}(t, U\alpha, \mu).$$

The basis V can be constructed by solving a problem similar to (2.8), replacing the solution snapshots with the evaluation of the nonlinear function $\mathbf{f}_{\text{nonlin}}$ for each sample included in S. The indices $\left\{p_i\right\}_{i=1}^{\tilde{k}}$ correspond to the components of the largest magnitude of an appropriately defined residual vector, as proposed in [26] and summarized in Algorithm 2.1. After replacing $\mathbf{f}_{\text{nonlin}}$ with $\hat{\mathbf{f}}$ in (2.12), we obtain

Algorithm 2.1 DEIM selection procedure

```
Input: Projection basis $V = [\mathbf{v}_1, \dots, \mathbf{v}_{\tilde{k}}] \in \mathbb{R}^{n \times \tilde{k}}$ of the nonlinear snapshots
Output: DEIM indices $\boldsymbol{p}$, DEIM matrix $P$
  Select $p_1$ as the index corresponding to the largest absolute component of $\mathbf{v}_1$.
  $W = [\mathbf{v}_1]$, $P = [e_{p_1}]$, $\boldsymbol{p} = [p_1]$.
  for $i = 2, \dots, \tilde{k}$ do
      Solve $(P^T W)\boldsymbol{c} = P^T \mathbf{v}_i$ for $\boldsymbol{c}$.
      Define the residual as $\boldsymbol{r} = \mathbf{v}_i - W\boldsymbol{c}$.
      Select $\rho_i$ as the index corresponding to the largest absolute component of $\mathbf{r}$.
      Update $W \leftarrow [W\ \mathbf{v}_i]$, $P \leftarrow [P\ e_{p_i}]$, $\boldsymbol{p} \leftarrow [\boldsymbol{p}^T\ p_i]^T$.
  end for
```

$$U^T \mathbf{f}(t, U\alpha, \mu) = \sum_{q=1}^{Q} \Theta_q(t, \mu) U^T \mathscr{L}_q U\alpha + U^T V (P^T V)^{-1} P^T \mathbf{f}_{\text{nonlin}}(t, U\alpha, \mu). \tag{2.14}$$

Computing (2.14) requires evaluating the nonlinear term only in the rows identified by P. If $\mathbf{f}_{\text{nonlin}}$ depends sparsely on $\mathbf{x}$, the evaluation of $P^T \mathbf{f}_{\text{nonlin}}$ requires the reconstruction of a number $\mathscr{O}(\tilde{k})$ of components, in contrast to the N required by (2.12). Although the sparsity assumption typically holds for standard finite difference or finite element discretization of

PDEs, for globally coupled problems or spectral methods, its validity is at stake. However, if the coupling function belongs to a specific class, one could still achieve the desired splitting. We refer to Chapter 3 for a detailed discussion on this subject. As the operator $U^T V(P^T V)^{-1} \in \mathbb{R}^{k\times\tilde{k}}$ can be precomputed during the offline phase, the online computational cost becomes independent of N once again.

Combining (2.6) and (2.14), the POD-DEIM approximation of (2.1) reads

$$\begin{cases} \frac{d}{dt}\alpha(t,\mu) = \sum_{q=1}^{Q}\Theta_q(t,\mu)U^T\mathscr{L}_q U\alpha + U^T V(P^T V)^{-1}P^T\mathbf{f}_{\text{nonlin}}(t, U\alpha,\mu), \quad t\in[0,T], \\ \alpha(0,\mu) = \alpha_0(\mu) = U^T\mathbf{x}_0. \end{cases} \tag{2.15}$$

Generalizations of (2.11) and (2.15) can be easily derived in the presence of vector-valued differential equations. The easiest option consists in constructing a basis for each unknown independently, leading to a block-diagonal matrix U. This choice avoids missing relevant features of the dynamics when the variables have different magnitudes and enhances the interpretability of both the reduced model and the latent coordinates. However, there is nothing specific about that, and a construction based on the entire set of unknowns can also be used. In this setting, the use of weighted inner products can be helpful from a numerical perspective, especially when the variables are characterized by different scales. Although in this work we mostly rely on the first strategy, specific choices are discussed in the related chapters.

In the following, we refer to (2.1) as the Full Order Model (FOM), while (2.6) and (2.15) constitute the Reduced Order Models (ROMs). For a practical implementation of the POD-DEIM reduced basis method we refer to Algorithm 2.2 and Algorithm 2.3.

2.1.5 Beyond reduced basis

Although the POD-Galerkin reduction, possibly equipped with the DEIM, works well for a large number of differential problems, we briefly mention some of its issues. Firstly, possible stability issues should be appropriately addressed. In the context of saddle-point problems, this can be done by enriching the reduced space with suitably designed functions [8], see Chapter 5 for a concrete application. More general alternatives rely on Petrov-Galerkin projections [21], closure modeling [131] or structure-preserving model reduction methods [1]. Secondly, the efficiency of reduced basis method and its variants are at stake if the Kolmogorov width has a slow decay rate, as it is the case with advection-dominated problems [94]. Even if the manifold (2.2) is low dimensional, the linear nature of the decomposition (2.3) introduces a strong restriction on the reduced order approximation. Alternative methods seek to relax this assumption in a variety of ways, including nonlinear manifolds [84] and local-in-time methods [69].

Algorithm 2.2 POD-DEIM: offline phase

```
Input: Full Order Model
Output: Reduced Order Model
  Select parameters $\mu_j$ and time instants $t_i$.
  Compute the solution $\mathbf{x}(t_i,\mu_j)$ of (2.1) and assemble the matrix $S$ as in (2.7).
  Compute the nonlinear terms $\mathbf{f}_{\text{nonlin}}(t_i,\mathbf{x}(t_i,\mu_j),\mu_j)$ and assemble the matrix $\tilde{S}$.
  Compute $[U_S,\Sigma_S,V_S]=\texttt{SVD}(S)$ and $[U_{\tilde{S}},\Sigma_{\tilde{S}},V_{\tilde{S}}]=\texttt{SVD}(\tilde{S})$.
  Find $k$,$\tilde{k}$ using, e.g., (2.10), and set $U=U_S(:,1:k)$ and $V=U_{\tilde{S}}(:,1:\tilde{k})$.
  Compute the matrix $P$ using Algorithm 2.1 with input $V$.
  Compute the reduced linear operators $U^T\mathscr{L}_qU$ and the operator $U^TV(P^TV)^{-1}$.
```

Algorithm 2.3 POD-DEIM: online phase

```
Input: Reduced Order Model
Output: Reduced solution
  Select parameters $\mu_j$ and time instants $t_i$, possibly different from the ones used in Algorithm 2.2.
  Solve (2.15) using $\sum_{q=1}^{Q}\Theta_q(t,\mu_j)U^T\mathscr{L}_qU$ and $P^T\mathbf{f}_{\text{nonlin}}(t,U\alpha(t,\mu_j),\mu_j)$ computed efficiently as in [26].
  Reconstruct the solution $\hat{\mathbf{x}}(t_i,\mu_j)=U\alpha(t_i,\mu_j)$ for visualization and postprocessing.
```

2.2 Non-intrusive surrogate modeling

The reduced basis method, as well as its nonlinear variants, can be successfully applied in a variety of problems and usually give a good compromise between accuracy and computational cost. However, their nature requires the access to the discretization operators, from which one can construct their reduced counterparts. This might not always be possible, especially when proprietary software is used, as the solvers are available as black boxes only. Thus, an alternative class of methods aims to construct non-intrusive, data-driven surrogates. From this point of view, this reduces to a function approximation problem, which can be formulated as follows:

Given an unknown and/or computationally expensive function $f:\mathbb{R}^n\to\mathbb{R}^m$ and a collection of $\mathscr{N}$ input-output pairs

$$\{x_i,y_i=f(x_i)\}_{i=1}^{\mathscr{N}}, \tag{2.16}$$

find an approximation $\hat{f}:\mathbb{R}^n\to\mathbb{R}^m$ of f such that

$$\hat{f}(x)\approx f(x) \tag{2.17}$$

for all $x\in\mathbb{R}^n$.

In the context of parametrized dynamical systems of the form (2.1), one could think of $f(\cdot)$ as the parameter-to-solution map, which associates to a given instance of the parameters (t,μ) the corresponding solution $\mathbf{x}(t,\mu)$ of the problem. However, (2.17) is a more general problem, which goes beyond model order reduction. In the following, we describe two possible approaches to solve it.

2.2.1 Kernel interpolation methods

The first technique is based on kernel interpolation methods [6]. Assuming a scalar output ($m = 1$) for simplicity, we seek the surrogate function of the form

$$\hat{f}(x) = \sum_{j=1}^{\mathcal{N}} \alpha_j K(x, x_j), \tag{2.18}$$

where $K(x, x')$ is the so-called kernel function. The coefficients α_j can be obtained by imposing the interpolation property

$$\hat{f}(x_i) = \sum_{j=1}^{\mathcal{N}} \alpha_j K(x_i, x_j) = f(x_i) \tag{2.19}$$

for each training point $i = 1, \dots, \mathcal{N}$ and solving the resulting linear system

$$\mathbf{K}\alpha = \mathbf{F}, \quad \text{with } [\mathbf{K}]_{ij} = K(x_i, x_j),\ [\alpha]_i = \alpha_i,\ [\mathbf{F}]_i = f(x_i). \tag{2.20}$$

The choice of the kernel is crucial for the theoretical guarantees and numerical efficiency of the method. In this work, we restrict to the class of Matérn kernels [106]

$$K(x, x') = K(r) = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{\sqrt{2\nu}}{l} r \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}}{l} r \right), \quad \text{where } r = \left\| x' - x \right\|_{\ell_2}, \tag{2.21}$$

which ensures, among other properties, the well-posedness of (2.19). Here, $K_\nu(\cdot)$ is a modified Bessel function and ν, l are two hyperparameters controlling the smoothness and the length scale of the kernel, respectively. A practical implementation is reported in Algorithm 2.4 and Algorithm 2.5, which focus on the construction of the interpolator and its evaluation in concrete applications. The main points in favor of kernel interpolation are its strong theoretical foundations, including error estimates, optimality and uniqueness statements, which enables us to analyze the approximation properties of the kernel-based surrogate models [66]. However, its computational complexity scales linearly (resp. superlinearly) with the number of training points in the online (resp. offline) phase, which can quickly become unpractical. Moreover, numerical instabilities arise from the ill-conditioning of the system (2.20).

Algorithm 2.4 Kernel interpolation: offline phase

```
Input: Training set $\{x_i, y_i = f(x_i)\}_{i=1}^{\mathcal{N}}$, kernel $K$.
Output: Interpolation coefficients $\alpha_j$, training points $X_{\mathcal{N}} = \{x_i\}_{i=1}^{\mathcal{N}}$.
  Compute the kernel matrix $[\mathbf{K}]_{ij} = K(x_i, x_j)$.
  Compute the interpolation coefficients $\alpha_j$ by solving the linear system (2.20).
```

Algorithm 2.5 Kernel surrogate: online phase

```
Input: Effective training points $\{x_i\}_{i=1}^{\mathcal{N}}$ selected in the offline phase, interpolation coefficients $\alpha_j$, test points $\{x_i^{test}\}_{i=1}^{\mathcal{N}_{test}}$, kernel $K$.
Output: Evaluation of the surrogate $\hat{f}(x^{test})$.
  Compute the kernel matrix $[\mathbf{K}]_{ij} = K(x_i^{test}, x_j)$.
  Evaluate the surrogate (2.18), i.e., $\hat{f}(x^{test}) = \mathbf{K}\alpha$.
```

To mitigate this effect and to enforce sparsity in the model, different methods have proven to be effective [25, 110]. Here, we rely on greedy methods, whose idea is to select a subset of (2.16) and solve the corresponding interpolation problem there. This selection procedure can be efficiently done using greedy strategies, with the goal of iteratively choosing the points that maximize a suitable error estimator or selection rule η [113, 135]. Once the new dataset is constructed, a new interpolation problem (2.19) has to be solved. These steps are summarized in Algorithm 2.6. Although not explicitly reported here, a more efficient training phase can be carried out by reusing the interpolant at the previous iteration and an appropriate change of basis [114]. The online evaluation of the surrogate model is the same as Algorithm 2.5, the only difference being that the effective training points will be the ones selected in the offline phase, which are typically a subset of the available data. These methods are known as Vectorial Kernel Orthogonal Greedy Algorithms (VKOGA).

The choice of the selection criterion has been shown to affect the efficiency of the method, and different strategies can be considered. For a set of N points X_N, we define the power function as

$$P_N(x) = P_{X_N}(x) = \left\| K(\cdot, x) - \Pi_{V(X_N)}(K(\cdot, x)) \right\|_{\mathcal{H}_K(\mathcal{D})}, \tag{2.22}$$

where $V(X_N) = \text{span}\{k(\cdot, x_i), x_i \in X_N\}$, $\mathcal{D}$ is the domain of f and $\mathcal{H}_K(\mathcal{D})$ is the Reproducing Kernel Hilbert Space associated to the kernel K, and the residual as

$$r_N(x) = f(x) - \Pi_{V(X_N)}(f)(x) = f(x) - \sum_{j=1}^{N} \alpha_j K(x, x_j).$$

Then, one can consider:

- The P-greedy selection: $\eta_P(x) = P_N(x)$. This aims at minimizing the power function, thereby providing a uniform upper bound on the error for any function.
- The f-greedy selection: $\eta_f(x) = |r_N(x)|$. This aims at minimizing the residual, making use of the data at hand to select the points.
- The f/P-greedy selection: $\eta_{f/P}(x) = |r_N(x)|/P_N(x)$.
- The $f \cdot P$-greedy selection: $\eta_{f \cdot P}(x) = |r_N(x)| \cdot P_N(x)$.

Unless stated otherwise, in this work we use the P-greedy strategy, as it is found to be a good compromise between accuracy and stability for our problems of interest. More details on each strategy and the mathematical foundations of (2.22), as well as a more general discussion on VKOGA, can be found in [114, 135]. Additionally, it can be shown [135] that VKOGA methods share the theoretical and numerical advantages of more classical interpolation methods, and additionally improve their stability and sparsity properties. However, the computational complexity can still be prohibitive. Even if greedy strategies are used, this class of methods is subject to the *curse of dimensionality*. The number of points needed to achieve a given interpolation error grows exponentially with the input dimension n, making kernel methods potentially intractable for high input dimensions [6].

Finally, we mention that an alternative formulation of kernel methods is based on a minimization problem, from which one can naturally introduce an additional regularization term. We refer to [66, 114] for further details on this subject.

Algorithm 2.6 Kernel greedy interpolation: offline phase

```
Input: Training set $(X_{\mathcal{N}}, Y_{\mathcal{N}}) = \left(\{x_i, y_i\}_{i=1}^{\mathcal{N}}\right)$, kernel $K$, selection rule $\eta$, tolerance $\tau$.
Output: Interpolation coefficients $\alpha_j$, effective training points $X_N = \{x_i\}_{i=1}^{N}$.
  Initialize $N \leftarrow 0$, $X_0 \leftarrow \emptyset$.
  repeat
      $N \leftarrow N+1$.
      Select $x_N = \arg\max_{x \in X_{\mathcal{N}} \setminus X_{N-1}} \eta(x, N, X_{\mathcal{N}}, Y_{\mathcal{N}})$.
      Update $X_N = X_{N-1} \cup \{x_N\}$.
      Compute the coefficients $\alpha_j$ with dataset $(X_N, Y_N)$ using, e.g., Algorithm 2.4.
  until $\eta(x_N, N, X_{\mathcal{N}}, Y_{\mathcal{N}}) \le \tau$
```

2.2.2 Artificial Neural Networks

The curse of dimensionality motivates our second choice of function approximation methods, Artificial Neural Networks (ANNs). We consider a simple multi-layer perceptron, a particular type of deep network in which the single neurons are organized in a sequence of layers [63, 45]. The first and the last ones are named input and output layers, respectively, whereas the internal ones are called hidden layers. The depth of the network is denoted by L, corresponding to a total of $L+1$ layers whose sizes are denoted by $k^0 = n$, $k^L = m$ and $\{k^l\}_{l=1}^{L-1}$ in the case of the input, output, and hidden layers, respectively. The resulting surrogate can be written as

$$\hat{f}(x) = \sigma_L \circ G_L \circ \sigma_{L-1} \circ G_{L-1} \circ \cdots \circ \sigma_1 \circ G_1(x),$$

where

$$G_l(z) = W_l z + b_l$$

is the affine transformation of level $l \in \{1, \ldots, L\}$, while $W_l \in \mathbb{R}^{k_l \times k_{l-1}}$ and $b_l \in \mathbb{R}^{k_l}$ are the weights and biases at the same level l. The functions $\sigma_l(\cdot)$ are called activation functions and they play a key role in the performance of the method. Typically, one chooses $\sigma_1 = \cdots = \sigma_{L-1} = \sigma$, whereas σ_L, also known as output function, can be chosen differently according to the problem of interest, and we choose it to be the identity map. The network parameters $\theta = \{(W_l, b_l)\}_l$ can be learned by minimizing an appropriate loss function, which in our case takes the form

$$\texttt{loss}(\theta) = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \left\| f(x_i) - \hat{f}(x_i) \right\|_{\ell_2}^2 + \lambda \sum_{l=1}^{L} \| W_l \|_F^2, \tag{2.23}$$

where $\lambda \geq 0$ is a small enough regularization parameter. The minimization problem can be solved using variants of the gradient descent method, in which the derivatives can be efficiently computed thanks to the backpropagation algorithm. This procedure is summarized in Algorithm 2.7 and Algorithm 2.8, whereas more details can be found in, e.g., [63]. The properties of ANNs, compared to classical interpolation algorithms, are somehow complementary. The main advantage is their compositional structure, which makes them very attractive in high dimensions, as they are shown to alleviate the curse of dimensionality in a number of cases [71]. However, theoretical guarantees can only be proven in specific cases, where simplifying assumptions are made either on the architecture of the network, the class of functions that one seeks to approximate, or the considered activation functions [90, 53]. Additionally, such results are often not constructive or can hardly be retrieved in practice.

Algorithm 2.7 Artificial Neural Networks: offline phase

```
Input: Training set $\{x_i, y_i = f(x_i)\}_{i=1}^{\mathcal{N}}$, network architecture (depth $L, \dots$), convergence criterion $c$.
Output: Optimized network parameters $\theta = \{(W_l, b_l)\}_l$.
  Split the training set into the actual training set $\mathcal{T}$ and validation set $\mathcal{V}$.
  Randomly initialize $(W_l, b_l)$, $n \leftarrow 0$.
  repeat
      $n \leftarrow n + 1$.
      Update weights and biases using, e.g., an iteration of gradient descent on the loss($\theta$) (2.23) using $\mathcal{T}$.
  until $c(\mathcal{V}, n)$ is satisfied
```

Algorithm 2.8 Artificial Neural Networks: online phase

```
Input: Test points $\{x_i^{test}\}_{i=1}^{\mathcal{N}_{test}}$, network architecture (depth $L, \dots$), optimized network parameters $\theta = \{(W_l, b_l)\}_l$.
Output: Evaluation of the surrogate $\hat{f}(x^{test})$.
  $\hat{f}(x^{test}) \leftarrow x^{test}$.
  repeat
      $\hat{f}(x^{test}) \leftarrow W_l \hat{f}(x^{test}) + b_l$.
      $\hat{f}(x^{test}) \leftarrow \sigma_l\left(\hat{f}(x^{test})\right)$.
  until $l == L$
```

2.2.3 Data preprocessing

Both the presented techniques can be directly applied to (2.17), provided that a set (2.16) is available. However, it is good practice to preprocess the data in order to facilitate the training phase.

In many cases the input and the output dimensions n and m can be very large, and potentially infinite. Constructing an accurate surrogate model of f can be challenging, even if ANNs are used [11, 13]. Dimensionality reduction techniques allow us to map the data to lower-dimensional spaces while retaining their main features, so that the approximation problem (2.17) can be formulated at this reduced level. We denote such reduced dimensions with N and M, respectively. Sticking to linear methods, we seek two (orthogonal) projection operators $\Pi_n : \mathbb{R}^n \to \mathbb{R}^n$ and $\Pi_m : \mathbb{R}^m \to \mathbb{R}^m$ such that

$$f(x) \approx \Pi_m f(\Pi_n x).$$

This reduces to finding two orthogonal matrices U_x and U_y such that the projection error of the input (resp. output) training data onto the span of U_x (resp. U_y), is minimized in a suitable sense. Choosing the Frobenius norm, the optimal choice is given by the SVD, similar to (2.8) and (2.9). Gathering the input data into a matrix $X \in \mathbb{R}^{n\times\mathcal{N}}$, whose SVD is given by

$$X = U_X \Sigma_X V_X^T, \tag{2.24}$$

then the optimal matrix is $U_x = U_X(:, 1:N)$. Arguing similarly for the output data, we have

$$f(x) \approx \Pi_m f(\Pi_n x) = U_y U_y^T f(U_x U_x^T x), \tag{2.25}$$

so that the function approximation problem reduces to finding an approximation $\hat{f} : \mathbb{R}^N \to \mathbb{R}^M$ of $U_y^T \circ f \circ U_x$ given the projected training data. Here, the symbol $\circ$ denotes function composition.

Moreover, the training samples (x_i, y_i) or their projected counterparts $(U_x^T x_i, U_y^T y_i)$ may have components characterized by different magnitudes. This is enhanced if the projected data are used, as modal coefficients often have a hierarchical structure, in which the last coefficients typically have a low magnitude. In practice, these different input or output scales can result in a slow or unstable learning, ultimately causing the entire process to fail. Therefore, it is good practice to add a scaling step in which the data are appropriately normalized. Considering for simplicity the unprojected samples, the idea is to recast the training problem using scaled data and do the inverse transformation to retrieve the original scales. We then have

$$f(x) = \texttt{y\_inv\_scale}(\texttt{y\_scale}(f(\texttt{x\_inv\_scale}(\texttt{x\_scale}(x; X)); X); Y); Y),$$

and the approximation problem becomes to find an approximation $\hat{f} : \mathbb{R}^n \to \mathbb{R}^m$ of $\texttt{y\_scale} \circ f \circ \texttt{x\_inv\_scale}$ given the scaled training data. Selecting a good scaling method is problem-dependent, and we leave the details of our choice to Chapter 6.

3 Globally coupled oscillatory systems

In this chapter[1] we construct reduced models for networks of coupled oscillators, taken as prototypes of large-scale systems exhibiting synchronization phenomena. Interactions among a large number of simple components can cause an adjustment of the local rhythms, that we wish to explore at the reduced level. As discussed in Chapter 1, we assume that the coupled problem is available, also in terms of numerical implementation. Moreover, the full order solver is given in a monolithic form, which treats the entire system as a whole. We focus on intrusive projection-based methods, taking particular care of the efficient treatment of the coupling term.

The first system of interest is the Kuramoto model [82], which is a typical model in the context of synchronization. It naturally arises when modeling chemical and biological oscillators [15, 101], synchronization of metronomes [97] or Josephson junctions [35]. It describes the phase dynamics of a network of limit-cycle oscillators through component-specific parameters, the natural frequencies, and a coupling term which depends on the phase differences among the oscillators. In the context of the Kuramoto model, diverse model reduction techniques have already been proposed. Watanabe and Strogatz [133] derived an exact two-dimensional reduced model for identical oscillators by applying a nonlinear transformation to the phases. Ott and Antonsen [95] constructed a two-dimensional model for the thermodynamic limit of the Kuramoto model with a Lorentz distribution of the natural frequencies, by making an ansatz on the oscillator density. A different viewpoint is offered by Gottwald [64], who proposed a collective coordinate approach, relying on a basis that depends on the natural frequencies. Although all these techniques are effective, they rely on model-specific assumptions and/or analytical tools, which prevent their generalization to more complex coupled problems. Effectively, these are model-driven, problem-dependent techniques, in contrast to our main goal of constructing data-driven reduced models.

The second system we analyze is a model of biological oscillators [80, 62, 12] used to study the circadian rhythm. It aims at modeling synchronization and entrainment of neuronal

---

[1] Large parts of this chapter are based on our research work [44]. Section 3.4 is extracted from the MSc thesis of A. van Lamsweerde [83], that this author co-supervised.

cells subject to mutual interactions and a common external forcing, representing the day-night cycle. To the best of our knowledge, no previous attempt has been done to construct a surrogate for similar models.

The remainder of this chapter is structured as follows. In Section 3.1 we present both the phase dynamics equations, particularly the Kuramoto model, and the circadian clock model, together with the quantities of interest that are used to analyze synchronization. The application of the reduced basis method to the models of interest is described in Section 3.2. Section 3.3 presents diverse numerical results, including a study of the computational performance of the proposed models. An extension to this work conducts a parameter sensitivity analysis, which is discussed in Section 3.4. A few concluding remarks are found in Section 3.5.

3.1 Globally coupled oscillatory systems

In this section we present the two classes of systems that are studied in this chapter.

3.1.1 Phase dynamics

Consider a network of N limit-cycle oscillators with weak mutual interactions. Using phase reduction theory, its dynamics can be analyzed by means of the phase variables $\{\phi_i(t)\}_{i=1}^N$, leading to the well-known phase dynamics equations [91]. In the absence of coupling, each phase is independent of the others, and it grows linearly at a rate ω_i, the natural frequency of each oscillator. Introducing the mutual interactions, the following model, based on a system of N ordinary differential equations, has been proposed [102, 91]

$$\frac{d\phi_i(t)}{dt} = \omega_i + \frac{K}{N}\sum_{j=1}^{N} H(\phi_j(t) - \phi_i(t)), \tag{3.1}$$

where $H(\cdot)$ is a 2π-periodic function that depends only on the phase difference $\phi_j(t) - \phi_i(t)$ between oscillators j and i. The coefficient $K \geq 0$ is the coupling strength, and is assumed to be equal for all oscillators. We remark that coupling term in (3.1) is global, meaning that all N oscillators interact, independently of their mutual distance. A simple choice for H yields the so-called Kuramoto model [82]

$$\frac{d\phi_i(t)}{dt} = \omega_i + \frac{K}{N}\sum_{j=1}^{N} \sin\big(\phi_j(t) - \phi_i(t)\big). \tag{3.2}$$

In the spirit of the original model, we assume the natural frequencies to be independent and randomly distributed according to a unimodal and Ω-symmetric distribution $g(\omega)$, for a fixed value $\Omega \in \mathbb{R}$. We recall that, although (3.1) and (3.2) are rather general models, they can be modified to take into account local interactions only [47], high-order harmonics of $H(\cdot)$ [85, 79, 139], or more general distributions of the natural frequencies [146].

Denoting the imaginary unit with $\imath$ and introducing the *complex order parameter*

$$Z(t) = R(t)e^{\imath\Psi(t)} = \frac{1}{N}\sum_{j=1}^{N} e^{\imath\phi_j(t)}, \tag{3.3}$$

the level of synchronization can be measured by its absolute value, usually referred to as the *order parameter* or the *coherence parameter*

$$R(t) = |Z(t)| = \left|\frac{1}{N}\sum_{j=1}^{N} e^{\imath\phi_j(t)}\right| \in [0,1]. \tag{3.4}$$

In the case of incoherent dynamics, the phase variables are scattered in the interval $[0,2\pi]$, and their average (3.3) is characterized by low absolute values, leading to an order parameter close to zero. Vice versa, in the case of coherent dynamics, the order parameter approaches one, as all phases have a limited variability. In practice, one is interested in the asymptotic synchronization properties of the system after the transient phase. Thus, (3.4) can be estimated only for temporal instants t greater than a sufficiently large time T_0 [64]. A good approximation of this asymptotic value is found by simply evaluating (3.4) at the final simulation time $T > T_0$, i.e.,

$$R = R(t = T). \tag{3.5}$$

We refer to (3.5) as the *asymptotic order parameter*. As a function of the coupling strength, (3.5) undergoes a supercritical bifurcation [124]. In the limit of an infinite number of oscillators, there exists a critical coupling strength $K_c = \frac{2}{\pi g(\Omega)}$ such that

- If $K < K_c$, then $R = \mathcal{O}(1/\sqrt{N}) \simeq 0$. This denotes incoherent dynamics.

- If $K \gtrsim K_c$ and $g(\omega)$ is sufficiently smooth, then

$$R \simeq \sqrt{\frac{16}{\pi K_c^3}}\sqrt{\frac{K-K_C}{-K_C g''(\Omega)}} \propto \sqrt{K-K_c}.$$

- If $K \gg K_c$, then $R \simeq 1$. This denotes complete synchronization.

This confirms the intuition that a sufficiently large coupling strength induces synchronization, and additionally shows that the transition follows a square-root scaling law.

By construction, (3.2) can be viewed as a description of the dynamics in polar coordinates with a unitary value of the radial coordinate. Introducing the change of variables

$$x_i(t) = \cos\phi_i(t), \quad y_i(t) = \sin\phi_i(t), \tag{3.6}$$

the Kuramoto model can also be written as

$$\begin{aligned}\frac{dx_i(t)}{dt} &= -y_i(t)\left(\omega_i + \frac{K}{N}\sum_{j=1}^{N}(y_j(t)x_i(t) - x_j(t)y_i(t))\right),\\ \frac{dy_i(t)}{dt} &= x_i(t)\left(\omega_i + \frac{K}{N}\sum_{j=1}^{N}(y_j(t)x_i(t) - x_j(t)y_i(t))\right).\end{aligned} \tag{3.7}$$

This change of variables does not prevent us from analyzing the synchronization properties as above. The complex order parameter (3.3) can be computed by introducing the inverse transformation of (3.6)

$$\phi_i(t) = \mathrm{atan}\left(\frac{y_i(t)}{x_i(t)}\right),$$

to reconstruct the phases. Alternatively, one could compute $Z(t)$ by applying the change of coordinates (3.6) in its definition (3.3). As the differences between the two approaches are limited, we adopt the first strategy, which additionally ensures that $R(t)$ does not exceed one. The set of coordinates (3.6) has additional advantages in the framework of model order reduction, as discussed in Section 3.2.

3.1.2 Circadian oscillators

Practically, we might not have access to the coupling function $H(\cdot)$ in the phase dynamics equation (3.1), or it can be difficult to estimate [122, 128]. Thus, oscillatory dynamical systems are usually modeled with a more general set of coordinates, in the spirit of (3.7). The phase variable is reconstructed from the oscillating signal at the limit cycle, after the transient period.

Here, we study a specific model for globally coupled circadian oscillators [80], comprising a network of N neuronal oscillators. In each oscillator $i \in \{1, \ldots, N\}$, the clock gene mRNA $X_i(t)$ produces a clock protein $Y_i(t)$ which activates a transcriptional inhibitor $Z_i(t)$, and this in turn inhibits the transcription of the clock gene. The mRNA also excites the production of a neurotransmitter $V_i(t)$. The network dynamics is influenced by two additional factors. The external light, modeled as a sinusoidal signal $L(t) = (L_0/2)(1 + \sin(\omega t))$, and the intercellular coupling, which depends on the average cellular neurotransmitter

$$F(t) = \frac{1}{N}\sum_{i=1}^{N} V_i(t).$$

The resulting model becomes

$$
\begin{aligned}
\tau_i \frac{dX_i(t)}{dt} &= v_1 \frac{K_1^4}{K_1^4 + Z_i(t)^4} - v_2 \frac{X_i(t)}{K_2 + X_i(t)} + v_c \frac{KF(t)}{K_c + KF(t)} + \frac{L_0}{2}(1 + \sin(\omega t)),\\
\tau_i \frac{dY_i(t)}{dt} &= k_3 X_i(t) - v_4 \frac{Y_i(t)}{K_4 + Y_i(t)},\\
\tau_i \frac{dZ_i(t)}{dt} &= k_5 Y_i(t) - v_6 \frac{Z_i(t)}{K_6 + Z_i(t)},\\
\tau_i \frac{dV_i(t)}{dt} &= k_7 X_i(t) - v_8 \frac{V_i(t)}{K_8 + V_i(t)},
\end{aligned}
\tag{3.8}
$$

where $v_1, v_2, v_4, v_6, v_8, K_1, K_2, K_4, K_6, K_8, k_3, k_5,, k_7, v_c, K_c, \omega, K, L_0$ are global network parameters, while the τ_i's model the intrinsic heterogeneity among oscillators. The latter are assumed to be independent and identically distributed according to a unitary-mean law $g(\tau)$.

Synchronization properties can be analyzed by means of four different indicators [80]. We introduce the synchronization variable

$$
Q(t) = \frac{F(t)^2}{\frac{1}{N}\sum_{i=1}^{N} V_i(t)^2},
\tag{3.9}
$$

and the parameter of synchrony

$$
\rho = \sqrt{\langle Q(t) \rangle} \in [0, 1],
\tag{3.10}
$$

where $\langle \cdot \rangle$ denotes the time average in the asymptotic state. Incoherent dynamics and complete synchronization are characterized by a parameter ρ close to zero or one, respectively. Defining the average gene concentration

$$
X(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t)
\tag{3.11}
$$

allows us to introduce the spectral amplification factor

$$
S = \frac{4}{L_0^2} \left| \left\langle e^{-\imath \omega t} X(t) \right\rangle \right|^2
\tag{3.12}
$$

and the average period $\bar{T}$ as the period of the average variable $X(t)$. We note that, given a general sinusoidal signal, multiple techniques can be used to estimate its average period. In this work, we adopt a simple approach, by defining the period as the mean length of the time intervals between two consecutive maxima of the signal. This procedure is applied exclusively for times larger than a fixed instant T_*, as the period can be defined only once the limit cycle is reached. Finally, the phase ϕ_i of each oscillator can be estimated by recalling that it increases by 2π in every period. Using the signal $X_i(t)$ for $t > T_*$, we first find its maximum points, i.e., the instants where the phase is a multiple of 2π. Then, the phase is defined in the remaining instants by means of linear interpolation. As it is the case with the average period, alternative techniques can be used to reconstruct the phase. We note that the choice of X_i as a driving signal is arbitrary, and one could use the time series of any of the system variables. Similar to the Kuramoto model, we can define the order parameter

$$R(t) = \left| \frac{1}{N} \sum_{j=1}^{N} e^{\imath \phi_j(t)} \right| \in [0,1], \tag{3.13}$$

which has an analogous interpretation. Its asymptotic value, i.e., its value at the final simulation time, plays the role of the asymptotic order parameter (3.5). Depending on the parameter values, all four quantities undergo substantial changes. In particular, it has been observed [80] that

- If the intercellular coupling strength K increases, both the synchrony parameter (3.10) and the average period $\bar{T}$ increase.
- If L_0 is sufficiently large, the system is forced to oscillate with a period $2\pi/\omega$, together with full synchronization, measured by the order parameter (3.13).

3.2 Model reduction

We now describe how the reduced basis techniques presented in Chapter 2 can be applied to the dynamical systems introduced in Section 3.1. To ease the notation, we omit the dependence on the parameter vector μ, unless explicitly needed.

In a network of N oscillators, the evolution equation of each oscillator $i \in \{1,\ldots,N\}$ is modeled by an m-dimensional system of first-order differential equations with state vector $\mathbf{x}_i = [x_i^1,\ldots,x_i^m]^T \in \mathbb{R}^m$. The uncoupled dynamics is governed by a function $\mathbf{h}_i(\mathbf{x}_i) \in \mathbb{R}^m$. The global model is constructed by stacking the local states as

$$\mathbf{x} = [\mathbf{x}_1^T,\ldots,\mathbf{x}_N^T]^T = [x_1^1,\ldots,x_1^m,\ldots,x_N^1,\ldots x_N^m]^T \in \mathbb{R}^{Nm}. \tag{3.14}$$

Assembling the functions $\mathbf{h}_i(\mathbf{x}_i)$ in a similar fashion yields the global uncoupled system. We assume that the coupling terms contribute to the dynamics in an additive way, i.e., they do not directly modify the terms $\mathbf{h}_i(\mathbf{x}_i)$, but they simply add a component resulting from nonlocal contributions. We also assume that their effect is identical for all oscillators, i.e., the form of the coupling term does not change within the network. Thus, without loss of generality, interactions can be modeled by means of a function $\mathscr{H} = \mathscr{H}(\mathbf{x}) \in \mathbb{R}^m$, resulting in a global system described by a function

$$\mathbf{f} = \mathbf{f}(\mathbf{x}) = \begin{pmatrix} \mathbf{h}_1(\mathbf{x}_1) + \mathscr{H}(\mathbf{x}) \\ \vdots \\ \mathbf{h}_N(\mathbf{x}_N) + \mathscr{H}(\mathbf{x}) \end{pmatrix} \in \mathbb{R}^{Nm}.$$

Specific forms of the coupling function will be provided in the following.

As already mentioned in Chapter 2, the most natural way to reduce systems of differential equations consists in treating each coordinate $j = 1,\dots,m$ separately. In our framework, this is related to the similar structure of the mathematical models of each oscillator and the interest in easily capturing the different features of the dynamics induced by each variable. Other options can be considered, and would be necessary in the case of more heterogeneous models, in which each oscillator is modeled with a different number of variables. Sticking to the first strategy, we construct a separate basis for each coordinate $j = 1,\dots,m$, and project each variable independently of the others. Gathering the components of the state vector (3.14) associated to the j-th variable in the vector $\mathbf{x}^j = [x_1^j,\dots,x_N^j]^T \in \mathbb{R}^N$ and collecting the corresponding snapshots, we construct m separate matrices

$$S_j = \left[\mathbf{x}^j(t_1,\mu_1),\dots,\mathbf{x}^j(t_p,\mu_1),\dots,\mathbf{x}^j(t_1,\mu_q),\dots,\mathbf{x}^j(t_p,\mu_q)\right] \in \mathbb{R}^{N\times pq}.$$

Computing their singular value decomposition and selecting the first k left singular vectors, we obtain m approximations

$$\mathbf{x}^j \approx \hat{\mathbf{x}}^j = U_j\alpha_j,$$

where $U_j \in \mathbb{R}^{N\times k}$ is the orthogonal basis associated to variable j and $\alpha_j \in \mathbb{R}^k$ is the corresponding vector of latent coordinates. Globally, we can write

$$\mathbf{x} \approx \hat{\mathbf{x}} = U\alpha = \mathscr{P}\,\mathrm{blkdiag}(\{U_j\}_{j=1}^m)[\alpha_1^T,\dots,\alpha_m^T]^T, \tag{3.15}$$

where $U \in \mathbb{R}^{Nm\times km} = \mathbb{R}^{n\times r}$ and $\alpha \in \mathbb{R}^{km} = \mathbb{R}^r$. The matrix $\mathscr{P}$ is defined as the permutation matrix such that

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{pmatrix} = \mathscr{P}\begin{pmatrix} \mathbf{x}^1 \\ \vdots \\ \mathbf{x}^m \end{pmatrix}.$$

Similarly, instead of applying the DEIM to the entire vector associated to the nonlinearity, we again rely on a splitting of the coordinates. Specifically, we apply the DEIM using the basis associated to a single variable. Thus, the selected indices identify $\tilde{k} \leq N$ oscillators. Then, all variables corresponding to the selected oscillators are considered. If pre-multiplied by the matrix $\mathscr{P}^T$, the global DEIM matrix P has a block-diagonal structure, similar to (3.15).

Moreover, we numerically observed that good reduction properties are found by selecting the same basis for the Galerkin projection and the DEIM, i.e, $V = U$ and $\tilde{k} = k$. Although there is nothing specific about it and the original DEIM formulation can be used instead, we noticed that the latter implies that very large values of $\tilde{k}$ have to be chosen to ensure accuracy, with a consequent increase in the computational cost. We conjecture that our choice gives stronger stability properties to the reduced model, especially around synchronized states, possibly dictated by the orthogonality property $U^T V = \mathbb{I}_k$ in the model (2.15). The stability is also enhanced by recalling that the models depend on random parameters, and the solution is more stable under variations in such parameters compared to its time derivative. From an accuracy perspective, our choice is naturally suboptimal, but is still able to achieve low projection errors, at least on the training set. This could be explained by the oscillatory nature of the systems, at least in the circadian clock model, for which the dynamics of the solution and its time derivative are similar. Therefore, this should be intended as a problem-dependent observation that leads to an efficient reduction of the models studied in this chapter, but it may not generalize well if more general problems are considered.

To ensure computational efficiency, the evaluation of the reduced model (2.15) should depend on the total number of oscillators N weakly. For general globally coupled systems this is not possible. The evaluation of $\mathscr{H}(\mathbf{x})$ requires the reconstruction of the state $\mathbf{x}$ for all N oscillators, resulting in a total cost proportional to N at each time step. A sufficient condition to overcome this bottleneck assumes a mean-field coupling. If $\mathscr{H}$ depends on $\mathbf{x}$ through an average over the N oscillators of each variable $j = 1,\dots,m$, the dependency on N can be shifted to the offline phase. Specifically,

$$\begin{aligned}\mathscr{H}(\mathbf{x}) &= \mathscr{H}\left(\frac{1}{N}\sum_{i=1}^{N} x_i^1,\dots,\frac{1}{N}\sum_{i=1}^{N} x_i^m\right)\\ &\approx \mathscr{H}\left(\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k}(U_1)_{ij}(\alpha_1)_j,\dots,\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k}(U_m)_{ij}(\alpha_m)_j\right)\\ &= \mathscr{H}(M_1\cdot\alpha_1,\dots,M_m\cdot\alpha_m),\end{aligned} \tag{3.16}$$

where

$$(M_j)_l = \frac{1}{N}\sum_{i=1}^{N}(U_j)_{il}, \quad j = 1,\dots,m,\ l = 1,\dots,k \tag{3.17}$$

can be computed in the offline phase. The online evaluation requires at most m inner products of vectors of size k, independently of N.

For small values of m, the reduced system can be written explicitly. This helps us to show that both the modified Kuramoto model (3.7) and the circadian oscillators model (3.8) fall in the framework of mean-field coupling. In the original coordinate system, the Kuramoto model (3.2) has local dynamics governed by $m = 1$ variables, and its reduced counterpart becomes

$$\begin{aligned}\frac{d\alpha}{dt} &= (P^TU)^{-1}P^T\left(\omega + K\left(\frac{1}{N}\sum_{j=1}^{N}\sin(U\alpha)_j\right)\cos(U\alpha) - K\left(\frac{1}{N}\sum_{j=1}^{N}\cos(U\alpha)_j\right)\sin(U\alpha)\right)\\ &= (P^TU)^{-1}\left(P^T\omega + K\left(\frac{1}{N}\sum_{j=1}^{N}\sin(U\alpha)_j\right)P^T\cos(U\alpha) - K\left(\frac{1}{N}\sum_{j=1}^{N}\cos(U\alpha)_j\right)P^T\sin(U\alpha)\right),\end{aligned} \tag{3.18}$$

where we assume that both the sine and cosine functions are evaluated componentwise. The global coupling term has a nonlinear dependence on the reduced variable. It does not fit in the framework of (3.16), and evaluating (3.18) has a cost that depends on N. Using the change of coordinates proposed in (3.6), we recover the modified Kuramoto model (3.7). The local dynamics depends on $m = 2$ variables, and the reduced model can be written as

$$
\begin{aligned}
\frac{d\alpha_1}{dt} &= (\tilde{P}^T U_1)^{-1}\tilde{P}^T \left[-(U_2\alpha_2)\odot\left(\omega + K\left(\frac{1}{N}\sum_{j=1}^{N}(U_2\alpha_2)_j\right)U_1\alpha_1 - K\left(\frac{1}{N}\sum_{j=1}^{N}(U_1\alpha_1)_j\right)U_2\alpha_2\right)\right]\\
&= (\tilde{P}^T U_1)^{-1}\tilde{P}^T\left[-(U_2\alpha_2)\odot(\omega + K(M_2\cdot\alpha_2)U_1\alpha_1 - K(M_1\cdot\alpha_1)U_2\alpha_2)\right]\\
&= (\tilde{P}^T U_1)^{-1}\left[-(\tilde{P}^T U_2\alpha_2)\odot\left(\tilde{P}^T\omega + K(M_2\cdot\alpha_2)\tilde{P}^T U_1\alpha_1 - K(M_1\cdot\alpha_1)\tilde{P}^T U_2\alpha_2\right)\right],\\
\frac{d\alpha_2}{dt} &= (\tilde{P}^T U_2)^{-1}\tilde{P}^T \left[(U_1\alpha_1)\odot\left(\omega + K\left(\frac{1}{N}\sum_{j=1}^{N}(U_2\alpha_2)_j\right)U_1\alpha_1 - K\left(\frac{1}{N}\sum_{j=1}^{N}(U_1\alpha_1)_j\right)U_2\alpha_2\right)\right]\\
&= (\tilde{P}^T U_2)^{-1}\tilde{P}^T\left[(U_1\alpha_1)\odot(\omega + K(M_2\cdot\alpha_2)U_1\alpha_1 - K(M_1\cdot\alpha_1)U_2\alpha_2)\right]\\
&= (\tilde{P}^T U_2)^{-1}\left[(\tilde{P}^T U_1\alpha_1)\odot\left(\tilde{P}^T\omega + K(M_2\cdot\alpha_2)\tilde{P}^T U_1\alpha_1 - K(M_1\cdot\alpha_1)\tilde{P}^T U_2\alpha_2\right)\right],
\end{aligned}
\tag{3.19}
$$

where $\tilde{P}$ contains k indices selected by applying the DEIM algorithm using either U_1 or U_2 and the symbol $\odot$ denotes the element-wise product. As it clearly fits into the mean-field coupling framework, in (3.19) we already applied definition (3.17). The circadian clock model (3.8) has a similar structure. As the expression of its reduced counterpart is a bit cumbersome, we do not report it. We simply note that each oscillator is described by $m = 4$ variables, and the intercellular coupling term reads

$$\mathcal{H}(\mathbf{x}) = [\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4]^T = [\mathcal{H}_1, 0, 0, 0]^T,$$

with

$$
\begin{aligned}
\mathcal{H}_1(\mathbf{x}) &= \nu_c \frac{KF}{K_c + KF} = \nu_c \frac{K\frac{1}{N}\sum_{i=1}^{N} V_i}{K_c + K\frac{1}{N}\sum_{i=1}^{N} V_i}\\
&= \nu_c \frac{K\frac{1}{N}\sum_{i=1}^{N}(U_4\alpha_4)_i}{K_c + K\frac{1}{N}\sum_{i=1}^{N}(U_4\alpha_4)_i} = \nu_c \frac{K(M_4\cdot\alpha_4)}{K_c + K(M_4\cdot\alpha_4)},
\end{aligned}
$$

where we use the definition of M_4 as in (3.17).

The final step aims at selecting the parameters of interest. A reasonable assumption is a complete knowledge of the uncoupled model, so that all the deterministic parameters of the independent oscillators can be considered as fixed. This would correspond to isolating the constitutive components and determining their parameters, which could possibly be done at an experimental level. Additionally, this has the advantage of keeping the total number of physical parameters controlled. Note that this applies only to the circadian clock model, as the natural frequencies in the phase dynamics equation are not deterministic, and it is consistent with similar numerical experiments [80]. Thus, the variable parameters are

- The coupling strengths. In the phase dynamics equation, they are encoded by the parameter K. In the circadian clock model, they are encoded in both the intercellular coupling strength K and the maximal external light L_0.
- The random parameters governing the intrinsic heterogeneity of the oscillators. These

> are the natural frequencies of the phase dynamics equations ω_i and the period heterogeneity parameter τ_i in the circadian clock model. There is nothing unique about this, and they can be considered fixed. However, allowing this flexibility makes the model better suited for practical applications, as it is independent of the realization of the random variables. Moreover, for large-scale systems, assuming a complete knowledge of the natural frequency or the intrinsic heterogeneity of all oscillators seems unrealistic. However, the independent nature of the parameters and their sampling makes the reduced models moderately less efficient, as shown at a numerical level.

However, in more general problems, it might be of interest to assume variations in all the physical parameters. A possible approach to tackle this case is discussed in Section 3.4.

3.3 Numerical results

In this section we present a number of numerical results, showing the strong reduction properties of the systems of interest, the computational efficiency of the reduced models, and an application to a simple control problem.

3.3.1 Phase dynamics

Unless stated otherwise, the numerical simulations of the phase dynamics equation have been obtained with a 3rd-order explicit Runge-Kutta time integrator, with time step $\Delta t = 0.1$ and a final time $T = 200$. We assume that the phases of the oscillators are uniformly spaced in $[0, \pi]$ at the initial time. As the model is invariant under phase shifts [64], the natural frequencies are pre-processed by subtracting their mean value.

We consider the Kuramoto model (3.2) with $N = 100$ oscillators and a uniform distribution $\mathscr{U}([a, b])$ of the natural frequencies, with $a = 0.97, b = 1.03$. To show the reduction properties of the system, we initially fix the instance of the random variables, considering a single realization. Besides time, the only parameter of interest is the coupling strength K. We choose $K_{max} = 0.15$ and $q = 15$ values of K, uniformly spaced in the range $[0, K_{max}]$. As the critical value is $K_c = 2(b-a)/\pi \approx 0.04$, the snapshot matrix includes both synchronized and incoherent states. The singular values of the snapshot matrix are shown in Figure 3.1. Their exponential decay is an indicator of the large degree of reduction of the system. The reduced model (3.18) is simulated with $k = 10$.

The numerical solutions of the full and reduced order models with $K = K_{max}$ are shown in Figures 3.2(a), 3.2(b) and 3.2(c) for 5 different oscillators. This confirms the high accuracy of the approximated solutions, at least visually. As we aim to study the synchronization properties of the system, we are interested in the evolution of the coherence parameter (3.4), rather than the numerical solution. The order parameter is shown as a function of time in Figure 3.2(d), demonstrating that the synchronization properties are captured by the reduced models. The asymptotic order parameter (3.5) for 10 uniformly spaced parameter values is reported in Figure 3.2(e). Good agreement is found among all approaches. The reduced models reproduce the trend obtained with the full model, which is in turn consistent with the theoretical results outlined in Section 3.1. We remark that the 10 values of the coupling strengths used in these simulations are not part of the ones selected to construct the snapshot matrix. Thus, Figure 3.2(e) confirms that the reduced model generalizes well to new parameter values.

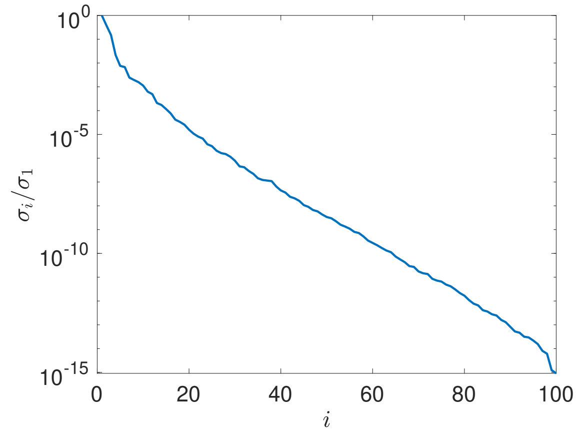
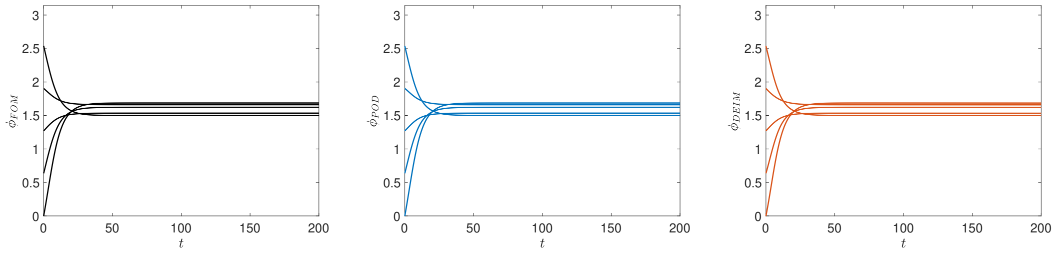
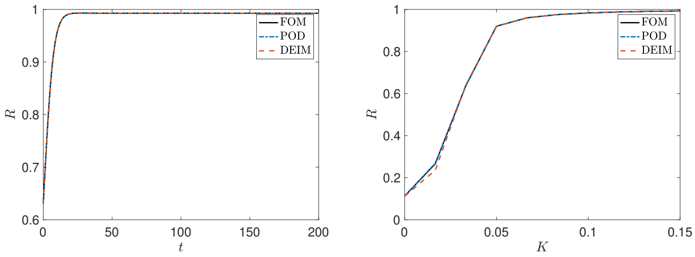


Figure 3.1 – Singular values for the Kuramoto model.



(a) Solution for $K = K_{max}$, FOM. (b) Solution for $K = K_{max}$, POD. (c) Solution for $K = K_{max}$, DEIM.



(d) Order parameter vs time for $K = K_{max}$.

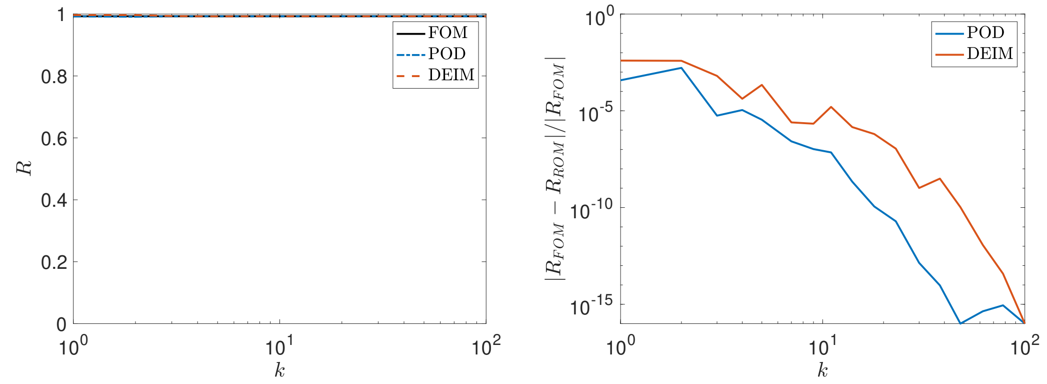
(e) Asymptotic order parameter vs coupling strength.

Figure 3.2 – Numerical results for the Kuramoto model with fixed random parameters.

The results show that using a reduced dimension $k = 10$, the quantity of interest is well approximated by the reduced models, at least qualitatively. This argument can be strengthened by considering a variable reduced dimension k. We study the approximation of the asymptotic order parameter (Figures 3.3(a) and 3.3(b)) and the computational cost (Figure 3.3(e)) when changing the latent dimension k, fixing the total number of oscillators $N = 100$ and the coupling strength $K = K_{max}$. The relative error in the order parameter decreases with the increasing latent dimension, as expected. We observe that the accuracy is high even when choosing $k = 1$, with an error of around 10^{-3}. This shows that for large coupling strengths, the dynamics is well approximated by a small number of basis functions, confirming the intuition that the behavior near synchronization is low-dimensional. However, to accurately capture the transition to synchronization and the asymptotic order parameter for small coupling strengths, a larger number of basis functions is needed. This constitutes the main motivation behind the choice of $k = 10$, as confirmed by Figures 3.3(c) and 3.3(d), which show the asymptotic order parameter for 3 different values of K used in the test phase and the corresponding error averaged over the entire test set. A very low reduced dimension struggles to accurately recover the asymptotic order parameters, especially in the region characterized by an incoherent dynamics. However, the increasing accuracy as a function of k is still observed, and the error decay is similar to Figure 3.3(b). The computational cost of the POD model is larger than the full model, as both the reconstruction and projection operations have to be added to the evaluation of the N components of the nonlinear function. Considering the DEIM model, for sufficiently small values of k we observe computational advantages, which are gradually lost for larger latent dimensions. These results are combined in Figure 3.3(f), which shows the error as a function of the computational cost. Using DEIM and a value of $k = 10$, we observe a computational saving of approximately 35%, with an error of 10^{-6} in the asymptotic order parameter.

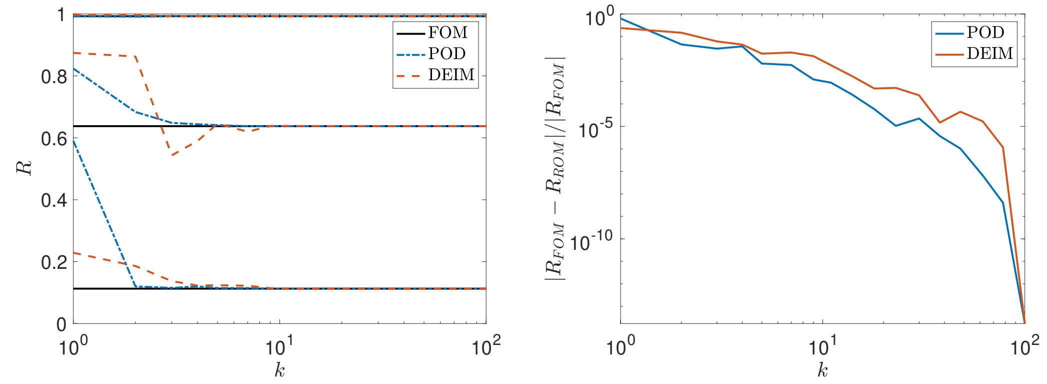
Similarly, we analyze the approximation properties when the full dimension N is changed. All other parameters are kept as in the previous simulations. The asymptotic order parameter and its relative error for $K = K_{max}$ are shown in Figures 3.4(a) and 3.4(b), whereas the asymptotic order parameter for 3 values of K and the corresponding relative error averaged over the entire test set are shown in Figures 3.4(c) and 3.4(d). The computational cost is reported in Figure 3.4(e). The reduced model does not significantly suffer from a variation in the size of the full model, denoting a mild dependence of k with respect to N. For all models, the computational cost increases with N, preventing the reduced model to reach its full potential in terms of computational efficiency.

We recall that in all the simulations the reduced model is generated by selecting $q = 15$ samples of the coupling strength and a time step $\Delta t = 0.1$. The former has to be sufficiently large to include both incoherent and synchronized states in the snapshot matrix, while the latter has to be sufficiently small to ensure stability of the numerical scheme both at the full and the reduced level. However, there is nothing specific about their values, and a sensitivity analysis can be carried out. We assume that $K_{max} = 0.15$ and $T = 200$ are kept fixed, as a change in their value does not provide new insights into the dynamics. Figure 3.5(a) shows the singular values



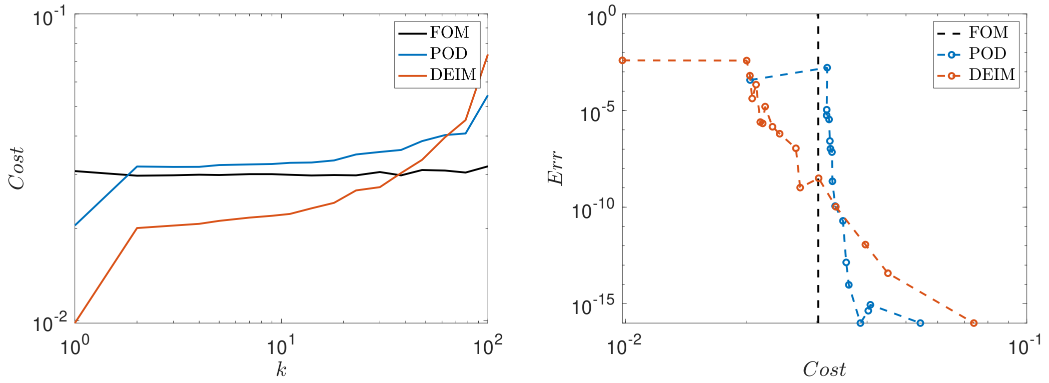
(a) Asymptotic order parameter ($K = K_{max}$).

(b) Relative error ($K = K_{max}$) of the asymptotic order parameter.



(c) Asymptotic order parameter for 3 values of K.

(d) K-averaged relative error of the asymptotic order parameter.

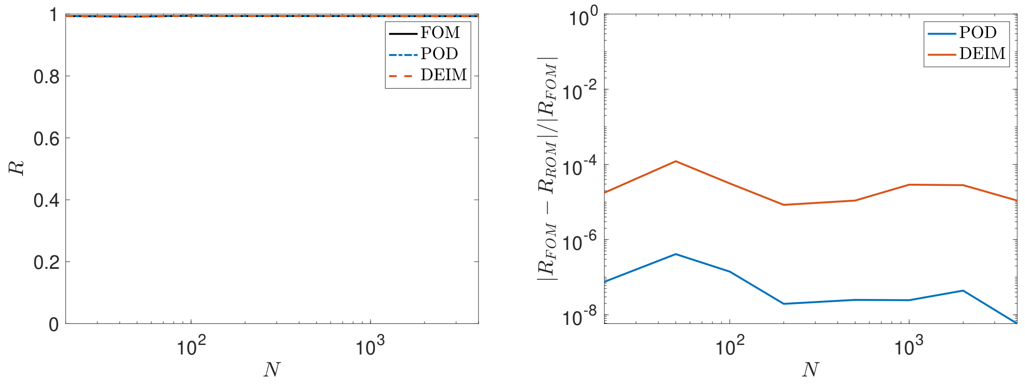


(e) Computational cost.

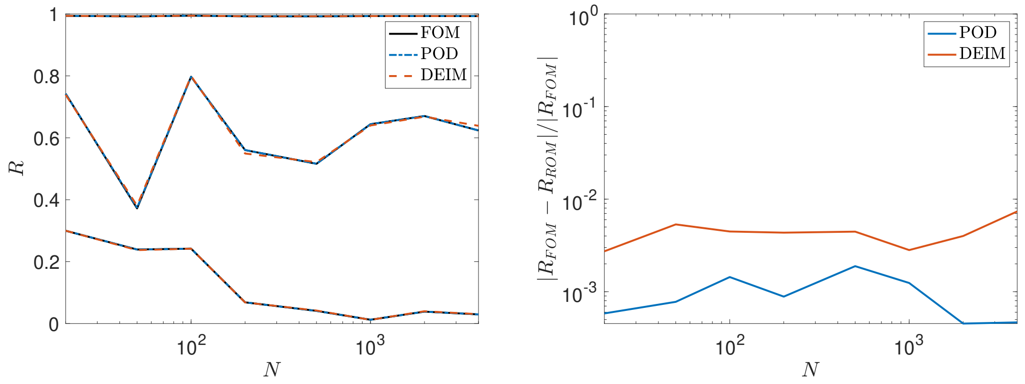
(f) Error vs computational cost.

Figure 3.3 – Numerical results for the Kuramoto model with fixed random parameters and variable reduced dimension k.

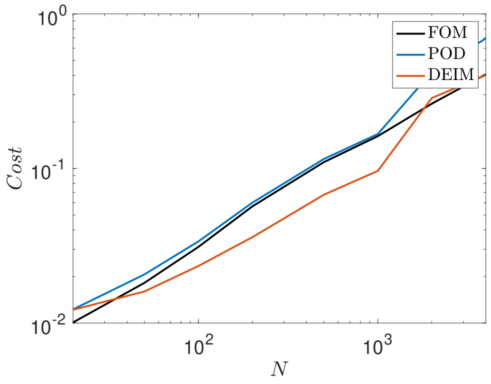
of the snapshot matrix for different values of the time step. As the trend is independent of Δt, we conclude that the time step has a very mild influence on the construction of the reduced model. Indeed, Δt is sufficiently small to accurately capture the system dynamics. On the



(a) Asymptotic order parameter ($K = K_{max}$). (b) Relative error ($K = K_{max}$) of the asymptotic order parameter.



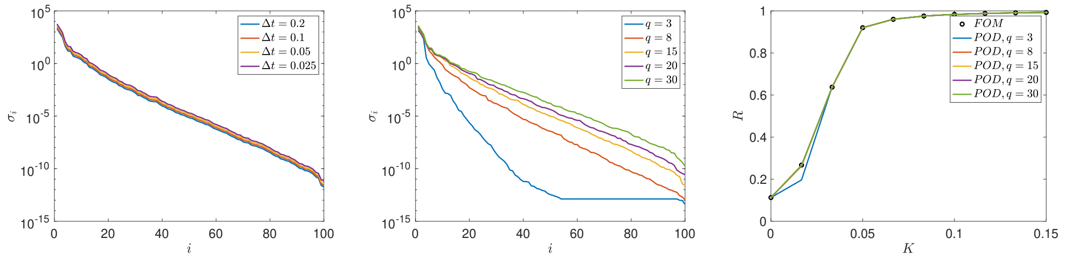
(c) Asymptotic order parameter for 3 values of K. (d) K-averaged relative error of the asymptotic order parameter.



(e) Computational cost.

Figure 3.4 – Numerical results for the Kuramoto model with fixed random parameters and variable full dimension N.

other hand, the number of parameter values q has a larger influence on the performance of the reduced model. For small values of q, the snapshots are not sufficient to completely capture the dynamics of the system. The singular values exhibit a fast decay (Figure 3.5(b)) as the phases have limited variability, but the generalization to values of the coupling strength not included in the snapshot matrix is relatively poor (Figure 3.5(c)). This is particularly evident for small coupling strengths. Increasing q, a more complete representation of the dynamics is ensured, leading to a smaller generalization error and a higher accuracy of the reduced model.



(a) Vs Δt.

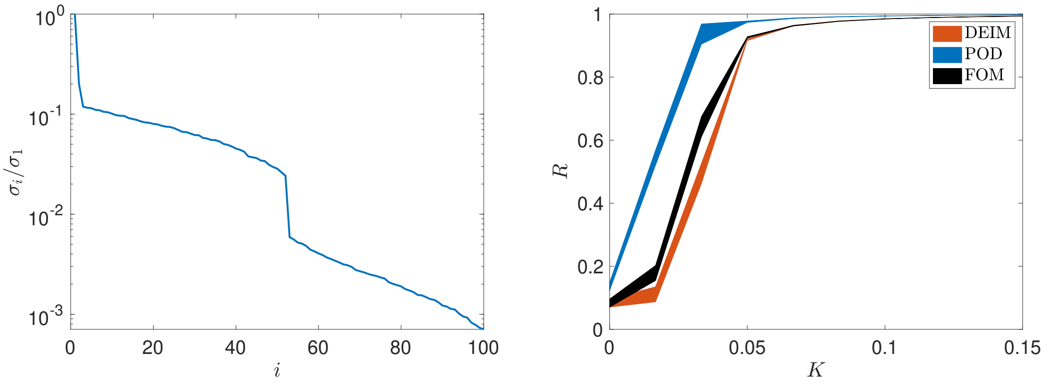
(b) Vs q.

(c) Asymptotic order parameter vs coupling strength for variable q.

Figure 3.5 – Sensitivity analysis for the Kuramoto model with fixed random parameters.

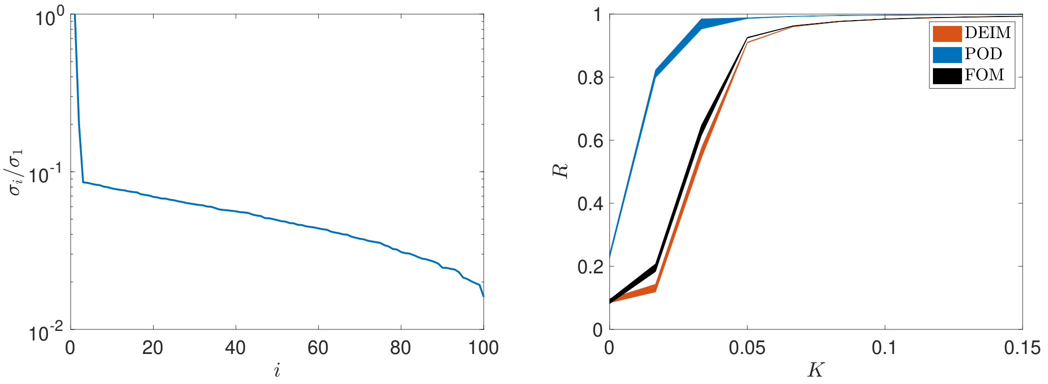
We consider now a variation of the natural frequencies of the system. The high dimensionality of the random vector $\omega = [\omega_1, \dots, \omega_N]^T$ prevents the generation of samples using stochastic collocation methods. The only practical option is a Monte Carlo simulation, with N_{MC} independent instances of the random vector. For simplicity, we use the same value of N_{MC} in both the offline and the online phase. Let us initially choose $N_{MC} = 50$, which leads to the results shown in Figures 3.6(a) and 3.6(b) for $k = 10$. The exponential decay of the singular values is lost. The first mode contains most of the energy, and a slow decay is observed for the first N_{MC} singular values. After a second jump, the decay is slow once again. This is due to the large heterogeneity of the uncoupled model, which is barely reducible for large values of N_{MC}. A positive coupling strength mitigates this phenomenon, but it is not sufficient to recover an optimal decay. Although a slow decay of the singular values indicates low reduction properties, synchronized states are approximated with high accuracy. The growth of the asymptotic order parameter, averaged over the Monte Carlo realizations, is retained, albeit with a relatively large error for small coupling strengths. As shown in Figure 3.6(b), the amplitude of the 95% confidence interval of the asymptotic order parameter is controlled. This suggests that using $N_{MC} = 50$ is sufficient to ensure convergence of the sampling error. Changing the number of Monte Carlo samples to $N_{MC} = 200$ leads to the results in Figures 3.6(c) and 3.6(d). The amplitude of the confidence interval is reduced, but the trend of the order parameter is not significantly altered. This confirms that the reconstruction error is due to the intrinsic variability of the systems, and not to sampling. We remark that a further increase in N_{MC} leads to even smaller confidence intervals, but the higher degree of heterogeneity of the system might lead to poor reduction properties.

In the Kuramoto model, an important role is played by the distribution $g(\omega)$ of the natural



(a) Singular values ($N_{MC} = 50$).

(b) Confidence interval ($N_{MC} = 50$).

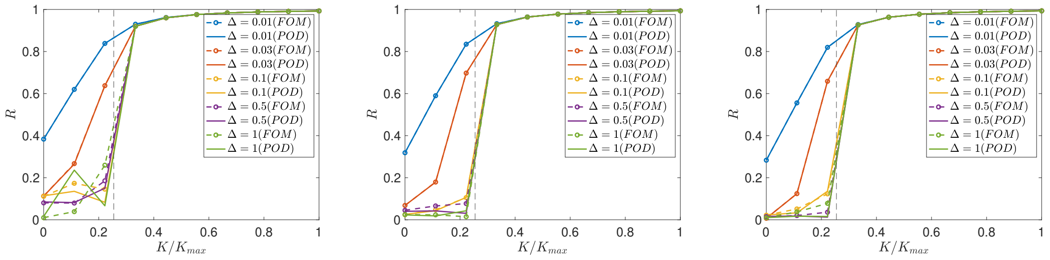


(c) Singular values ($N_{MC} = 200$).

(d) Confidence interval ($N_{MC} = 200$).

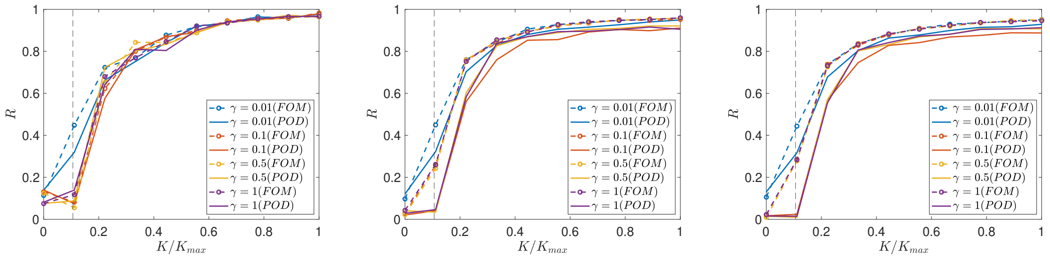
Figure 3.6 – Numerical results for the Kuramoto model with Monte Carlo simulation. Singular values and 95% confidence interval of the asymptotic order parameter vs coupling strength.

frequencies. Thus, we investigate how its choice affects the accuracy of the reduced models. We begin our analysis by considering a uniform distribution with parameters $a = 1-\Delta, b = 1+\Delta$ for different values of Δ. The critical value is given by $K_c = 2(b-a)/\pi = 4\Delta/\pi$. To ensure that both synchronized and incoherent states are captured, we select $K_{max} = 2.5(b-a) = 5\Delta$, while the other parameter values are kept as in the previous simulations. The results for different values of N are reported in Figure 3.7. As expected, in the synchronization region the evolution of the asymptotic order parameter is well retained at the reduced level for all values of Δ and N. In the incoherence region, the accuracy of the reduced models suffers from an increase in the variability Δ. The snapshots exhibit a larger degree of heterogeneity, which negatively affects the reduction potential of the model. However, the accuracy of the reduced models benefits from an increase in the value of N. The quality of the reduced basis increases, as the heterogeneity in the natural frequencies and snapshots remains controlled. Similar conclusions can be drawn in the case of other finitely-supported distributions. On the other hand, the reduced models appear to be moderately less effective in the case of infinitely-supported densities. We simulate the Kuramoto model with a Lorentz distribution centered in $\Omega = 1$ for different values of the scaling γ. We select $K_{max} = 6\pi\gamma$, while $K_c = 2\gamma$. The results for different values of N are reported in Figure 3.8. The trend of the asymptotic order parameter is well captured, although the error is larger as compared to the uniform distribution. The larger variability in the natural frequencies, enhanced by the presence of samples in the tails of the distribution, leads to a higher degree of heterogeneity in the snapshots and a lower reduction potential. In this case, to increase the accuracy of the reduced models, a larger value of the reduced dimension k has to be chosen.



(a) $N = 100$.   (b) $N = 1000$.   (c) $N = 4000$.

Figure 3.7 – Asymptotic order parameter vs coupling strength for the Kuramoto model with a uniform distribution with variable parameters.



(a) $N = 100$.   (b) $N = 1000$.   (c) $N = 4000$.

Figure 3.8 – Asymptotic order parameter vs coupling strength for the Kuramoto model with a Lorentz distribution with variable parameters.

The main drawback of the reduced Kuramoto model (3.2) is its computational cost, as the dependence of the coupling term on the full dimension N prevents large speedups. This can be solved by simulating the Kuramoto model with the change of coordinates (3.6). We keep most of the parameter values as in the original model. The maximum coupling strength is $K_{max} = 0.15$, the number of samples is $q = 15$, and the natural frequencies are drawn from a uniform distribution with parameters $a = 0.97, b = 1.03$. The number of oscillators is still $N = 100$, while the dimension of the system is $n = 2N$, and we choose a reduced dimension of $r = 2k = 20$. We construct $m = 2$ bases U_1 and U_2 of dimension $k = 10$, and apply DEIM on the matrix U_1, selecting $k = r/2 = 10$ oscillators. This choice of r allows us to compare the results obtained with the original coordinate system. With fixed natural frequencies, the singular values exhibit once more a fast decay, as shown in Figure 3.9 for the set of variables x_i in (3.7).

Fixing the coupling strength to $K = K_{max}$, Figures 3.10(a), 3.10(b) and 3.10(c) report the trajectories of the variable x_i of 5 different oscillators obtained with the full and the reduced models, while Figure 3.10(d) shows the time evolution of the order parameter. The transient behavior is well captured, and the synchronization properties are retained by the reduced model. If K varies, the asymptotic value of the order parameter obtained with the full model is well approximated for all values of the coupling strength, as shown in Figure 3.10(e) for 10 uniformly spaced values of K, different from the ones used in the offline phase. These results are similar to those obtained using (3.2), although minor differences are present, especially using DEIM.

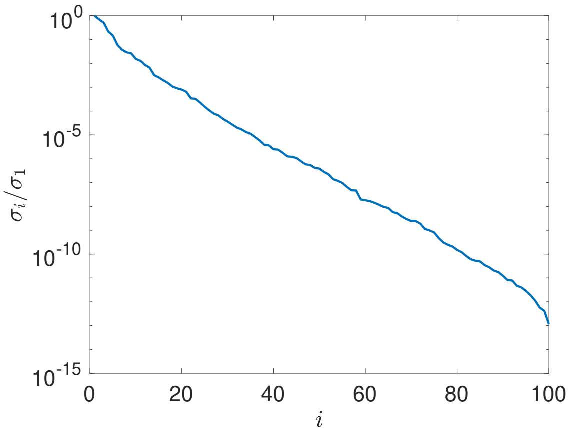
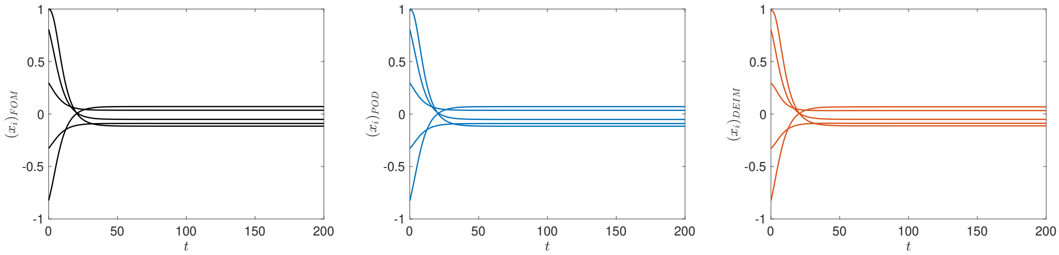


Figure 3.9 – Singular values for the modified Kuramoto model using the variables x_i.

We now consider a variation of the latent dimension k with $N = 100$ oscillators. As we are mostly interested in studying synchronization, we take the coupling strength equal to $K = K_{max}$, although the behavior for different values of K does not significantly change. The asymptotic order parameter, its relative error and the computational cost are shown in Figures 3.11(a), 3.11(b) and 3.11(c), respectively, as a function of the number of selected oscillators $k = r/2$. Small latent dimensions lead to poor approximations of the order parameter, although for $k \gtrsim 10$ the synchronization properties are well captured. This is in partial contrast to Figure 3.3(a), where the synchronized states are retained even with an extremely small latent dimension, but in agreement with Figure 3.3(c). The computational cost of the reduced models increases with k, and DEIM loses its efficiency after $k \gtrsim 30$. These results are combined in Figure 3.11(d), which shows the error as a function of the required computational cost. Although the approximation error is larger compared to Figure 3.3(f), it is still controlled. In particular, using DEIM with a value of $k = 10$, we observe a computational saving of approximately 30% with an error magnitude of 10^{-4}.

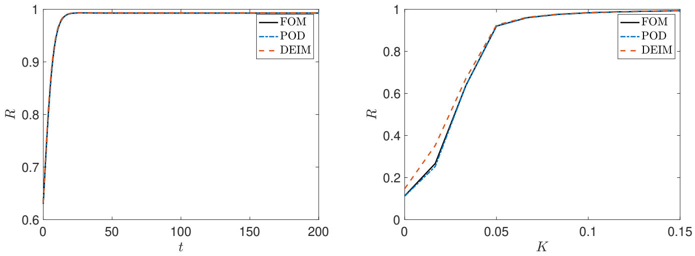
The potential of the proposed change of coordinates is fully triggered when changing the dimension of the full model $n = 2N$, while fixing $r = 2k = 20$, together with the other parameters. The asymptotic order parameter, its relative error and the computational cost are



(a) Solution (x_i) for $K = K_{max}$, FOM.

(b) Solution (x_i) for $K = K_{max}$, POD.

(c) Solution (x_i) for $K = K_{max}$, DEIM.



(d) Order parameter vs time for $K = K_{max}$.

(e) Asymptotic order parameter vs coupling strength.

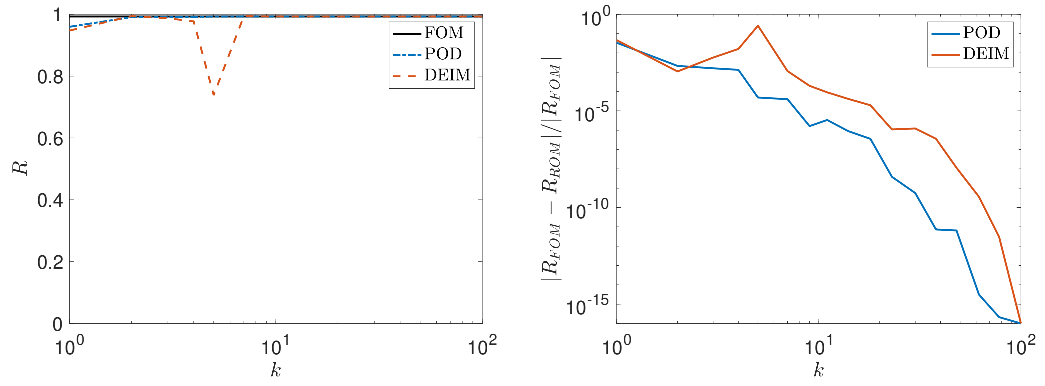
Figure 3.10 – Numerical results for the modified Kuramoto model with fixed random parameters.

reported in Figures 3.12(a), 3.12(b) and 3.12(c), respectively, as a function of the total number of oscillators N. The asymptotic order parameter is well approximated even for large values of N. As expected, the computational costs of the full and the POD model increase with N, with the latter being slightly larger. Given the mean-field nature of the coupling term, the cost of the DEIM model does not depend on N, guaranteeing a larger speedup as the number of oscillators increases, while maintaining a reasonable accuracy. The mild dependence on $N = n/2$ in Figure 3.12(c) is due to the reconstruction step at the end of the time integration, which is done for the solution at all time steps. This can be avoided by only reconstructing it at the final simulation time.

It is instructive to observe that, at the continuous level, (3.7) ensures a unitary amplitude for all times, i.e.,

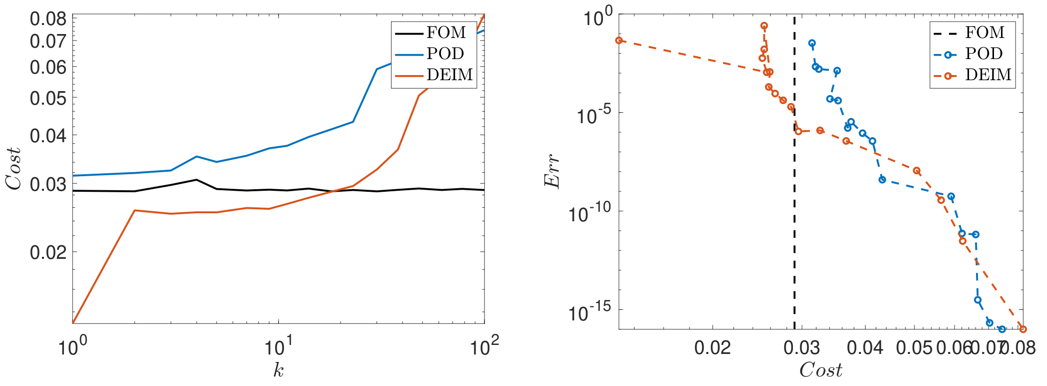
$$x_i(t)^2 + y_i(t)^2 = 1, \quad i \in \{1, \dots, N\},\ t \in [0, T].$$

Although the adopted time integrator does not guarantee the exact preservation of this constraint at the discrete level, the amplitude error in the full model is controlled to the order of the scheme. In turn, there is no guarantee that the same holds at the reduced level, as the constraint is enforced neither in the model formulation nor in the construction of the POD basis. Thus, the unitary amplitude is preserved up to the accuracy of the reduced model. This implies that for a sufficiently large dimension k, the error in the amplitude is controlled,



(a) Asymptotic order parameter ($K = K_{max}$).

(b) Relative error ($K = K_{max}$) of the asymptotic order parameter.



(c) Computational cost.

(d) Error vs computational cost.

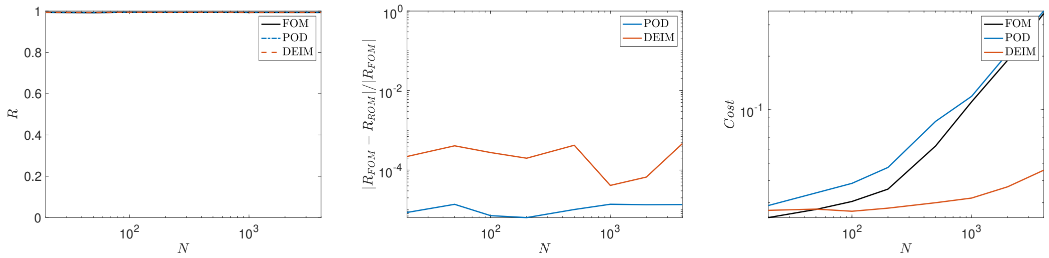
Figure 3.11 – Numerical results for the modified Kuramoto model with fixed random parameters and variable reduced dimension k.

similarly to the full model. This is confirmed by computing

$$\mathrm{Err_A} = \max_{t\in[0,T]} \max_{i=1,\ldots,N} \left| \sqrt{x_i(t)^2 + y_i(t)^2} - 1 \right|$$

for different values of $k = r/2$, as reported in Figure 3.13 for $K = K_{max}$. Consistent with Figure 3.9, the error of the reduced model decays exponentially with k, and it saturates at the value of the full model. We note that for values of $k \simeq 10$, the error is less than 10%.

To conclude the analysis, we consider different instances of the natural frequencies of the oscillators. We adopt a Monte Carlo sampling of the random parameters, choosing both $N_{MC} = 50$ and $N_{MC} = 200$ samples. The singular values decay slower (Figures 3.14(a) and 3.14(c)), but the synchronization properties are not impacted (Figures 3.14(b) and 3.14(d)). After averaging over the realizations, the order parameter reaches its asymptotic value in time, and the correct trend is retrieved when varying the coupling strength. As in the standard Kuramoto model, the error for small coupling strengths is due to the large degree of heterogeneity of the system.



(a) Asymptotic order parameter ($K = K_{max}$).

(b) Relative error ($K = K_{max}$) of the asymptotic order parameter.

(c) Computational cost.

Figure 3.12 – Numerical results for the modified Kuramoto model with fixed random parameters and variable full dimension N.

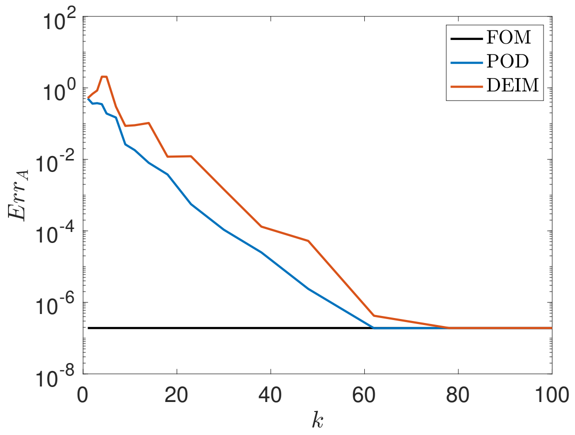
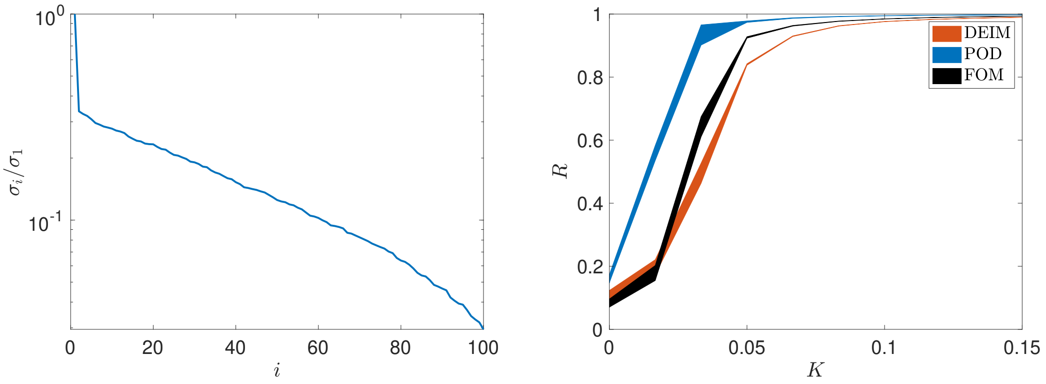


Figure 3.13 – Amplitude error vs latent dimension.

3.3.2 Circadian oscillators

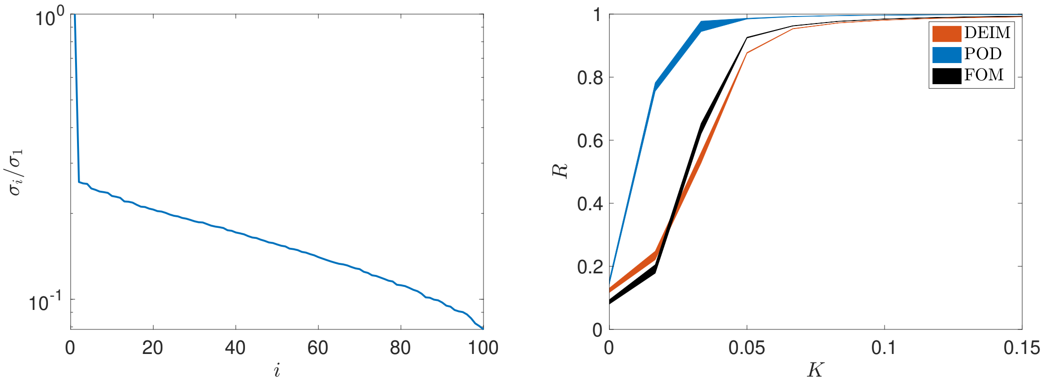
Numerical simulations of (3.8) with a general initial condition show a recurrent trend. The solution initially exhibits a long transient phase, characterized by a large spike of the variable Z_i, while asymptotically a limit cycle is reached. Constructing a reduced model which is able to accurately approximate the solution in both time ranges is rather challenging. However, a large part of the transient phase can be avoided by selecting an initial state sufficiently close to the limit cycle. As we assume a complete knowledge of the full, uncoupled model, we first identify a state that belongs to the limit cycle of a single oscillator. This is done by simulating the single-oscillator model for a sufficiently large time, and picking a state on the limit cycle. We then assume that all oscillators start in this synchronized state, which is found to be close to the limit cycle of the global model for all choices of the parameters.

The parameters determining the dynamics of each oscillator are fixed to $v_1 = 0.7$, $v_2 = v_4 = v_6 = 0.35$, $v_8 = 1$, $K_1 = K_2 = K_4 = K_6 = K_8 = 1$, $k_3 = k_5 = 0.7$, $k_7 = 0.35$, similar to [80]. We also



(a) Singular values ($N_{MC} = 50$).

(b) Confidence interval ($N_{MC} = 50$).



(c) Singular values ($N_{MC} = 200$).

(d) Confidence interval ($N_{MC} = 200$).

Figure 3.14 – Numerical results for the modified Kuramoto model with Monte Carlo simulation. Singular values and 95% confidence interval of the asymptotic order parameter vs coupling strength.

fix $v_c = 0.4$, $K_c = 1$ and $\omega = 2\pi/24$. The initial condition for the following experiments is

$$\mathbf{x}_0 = [0.0998, 0.2468, 2.0151, 0.0339, \dots, 0.0998, 0.2468, 2.0151, 0.0339]^T,$$

and we use a 3rd-order explicit Runge-Kutta time integrator, with time step $\Delta t = 0.5$ and final time $T = 500$. We consider $N = 100$ and a uniform distribution $\mathscr{U}([a, b])$ of the parameters τ_i, with $a = 0.8, b = 1.2$. First, we consider a single realization of the random variables, all of which are fixed a priori. Thus, the only non-constant parameters are the two coupling strengths, with a corresponding two-dimensional parameter space. Each variable is sampled uniformly in the intervals $[0, K_{max}]$ and $[0, L_{0,max}]$, and the two-dimensional sampling points are obtained by a tensor product of the one-dimensional counterparts. We choose $K_{max} = 0.6$ and $L_{0,max} = 0.02$, and 5 (resp. 3) points in the first (resp. second) dimension, for a total of $q = 15$ different parameter instances used for the construction of the snapshot matrix. The singular values are shown in Figure 3.15 for the variables V_i. Although moderately slow, at least when compared to Figure 3.1 and Figure 3.9, the decay is sufficient to obtain a good model reduction with

$r = 4k = 40$. To construct the reduced model, we compute a separate basis for each of the $m = 4$ variables, and the DEIM algorithm is applied to the basis U_4, selecting $k = 10$ oscillators. The full and the approximated solution variables V_i are shown, for 5 different oscillators, in Figure 3.16. Here, the coupling strengths are fixed to $K = K_{max}, L_0 = L_{0,max}$. As expected, the reconstruction error is controlled. To study synchronization phenomena, we focus on the evolution of the quantities of interest we introduced in Section 3.1. Only a subset of them are time-dependent, and are reported in Figure 3.18. Specifically, Figure 3.17(a), Figure 3.17(b), and Figure 3.17(c) show the synchrony variable (3.9), the average gene concentration (3.11), and the order parameter (3.13), respectively, as time varies. In all cases, the errors between the full and the reduced models are small. The remaining quantities of interest are shown as the coupling strengths vary. We consider 10 different values of the intercellular coupling K and 5 of the external source L_0, uniformly spaced in the same intervals as in the offline phase. We remark that the resulting parameter instances are different from the ones used to construct the snapshots. We report the synchrony parameter (3.10) in Figures 3.18(a), 3.18(b) and 3.18(c), the spectral amplification factor (3.12) in Figures 3.18(d), 3.18(e) and 3.18(f), the average period in Figures 3.18(g), 3.18(h) and 3.18(i), and the asymptotic value of

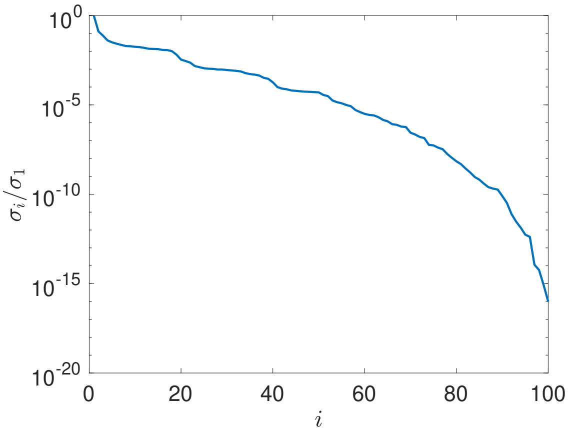
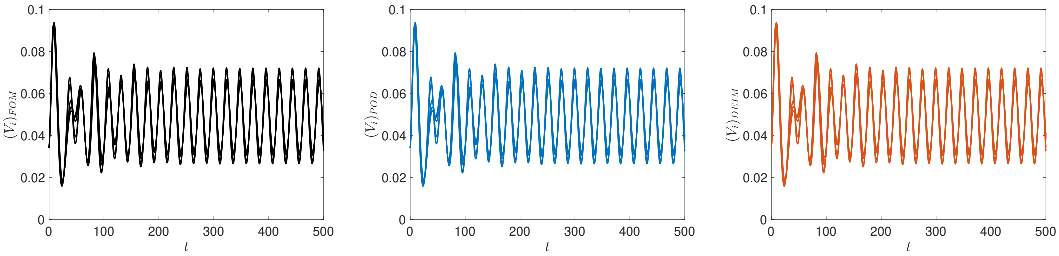
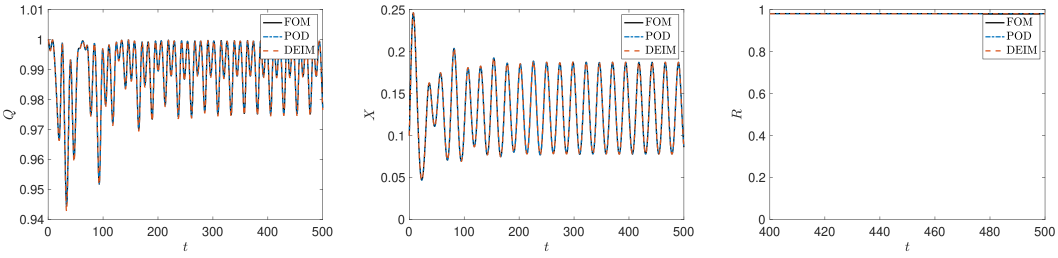


Figure 3.15 – Singular values for the circadian oscillators model using the variables V_i.



(a) FOM. (b) POD. (c) DEIM.

Figure 3.16 – Approximated solution (V_i) of the circadian oscillators model for $K = K_{max}, L_0 = L_{0,max}$.



(a) Synchrony variable. (b) Average gene concentration. (c) Order parameter.

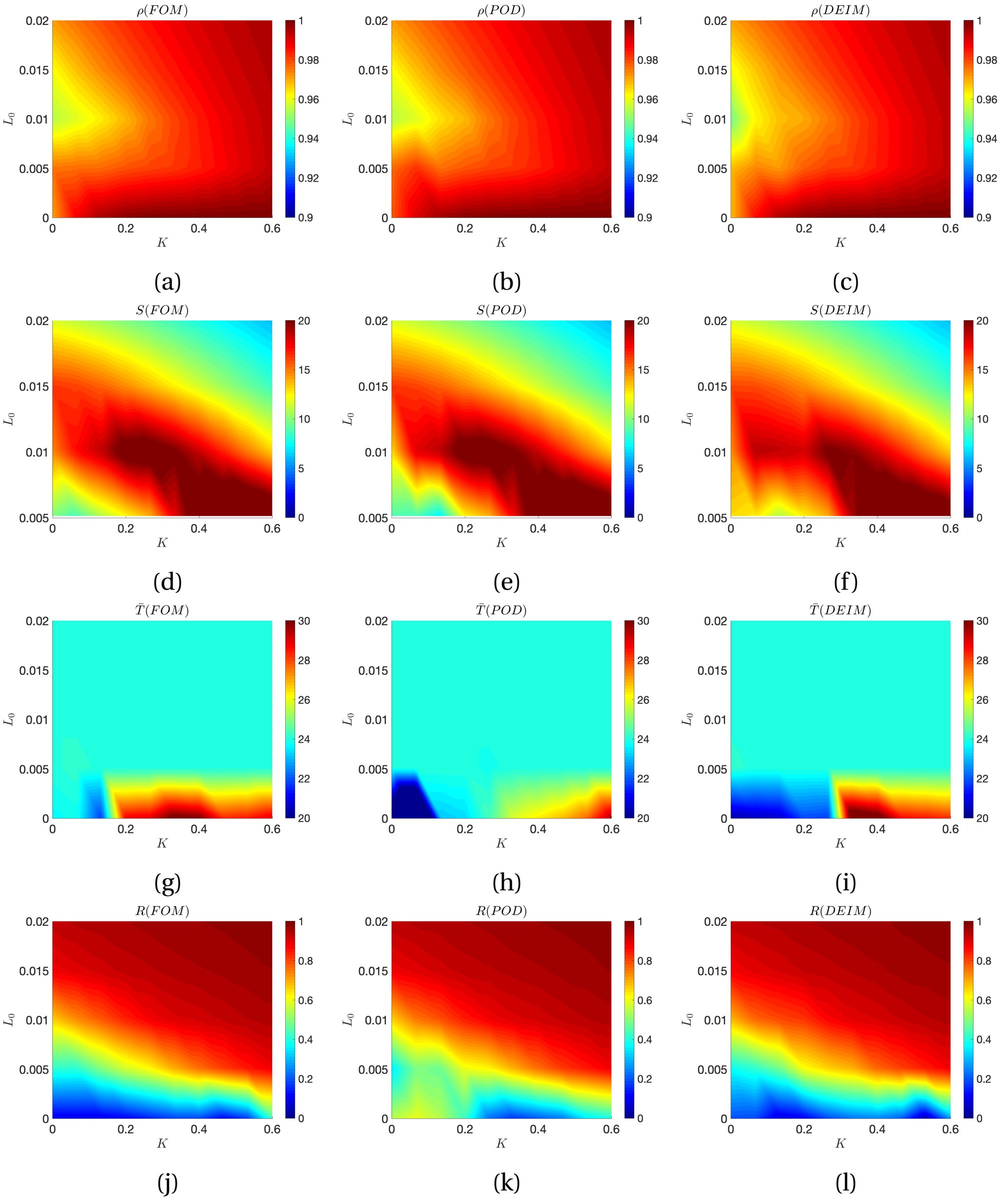
Figure 3.17 – Numerical results for the circadian oscillators model with fixed random parameters as a function of time for $K = K_{max}, L_0 = L_{0,max}$.

the order parameter in Figures 3.18(j), 3.18(k) and 3.18(l). The quantities of interest are well approximated by the reduced model, with higher accuracy for larger coupling strengths. The synchrony parameter and the spectral amplification factor are well recovered at the reduced level, as minor differences among all models are found. The transition to synchronization measured by the order parameter is also well retained, with minor differences in the regions with incoherent dynamics. The average period for large external sources is retained, while for low intensities, the differences among the models are moderate.

As for the Kuramoto model, we study the reduction potential when the latent dimension is varied. In the spirit of the modified Kuramoto model, we consider $N = 100$, $K = K_{max}$ and $L_0 = L_{0,max}$. We report the asymptotic order parameter, its relative error and the computational cost in Figures 3.19(a), 3.19(b) and 3.19(c), respectively, as a function of the number of selected oscillators $k = r/4$. The approximation error reduces as k increases, and even with a small reduced dimension this quantity of interest is approximated with high accuracy. As expected, the computational cost increases with k. These results are combined in Figure 3.19(d), which shows the error as a function of the required computational cost.

A similar analysis can be carried out when fixing the latent dimension $r = 40$, while varying the dimension of the full model n. The results are reported in Figure 3.20 as a function of the number of oscillators $N = n/4$. The reduced model is able to approximate the quantity of interest with a satisfactory accuracy even for large values of N. This indicates the mild dependence of the reduced dimension on the full dimension. The advantages of the reduced model are enhanced when analyzing the computational performance. Using the DEIM approximation, the simulation cost does not depend strongly on N, ensuring a significant speedup. As for the Kuramoto model, the mild dependence on N is due to the reconstruction step, and can be avoided.

Our analysis is completed by constructing a reduced model based on multiple instances of the random parameters. We consider a Monte Carlo sampling, using $N_{MC} = 50$ instances of the vector $\tau = [\tau_1, \ldots, \tau_N]^T$. The relevant quantities of interest are shown as a function of the coupling strengths in Figure 3.21, after averaging over the realizations. Their most significant



(a) (b) (c)

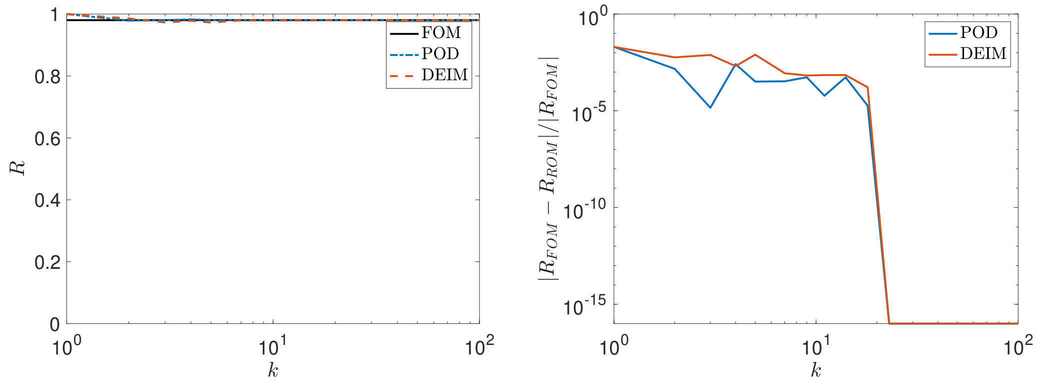
(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

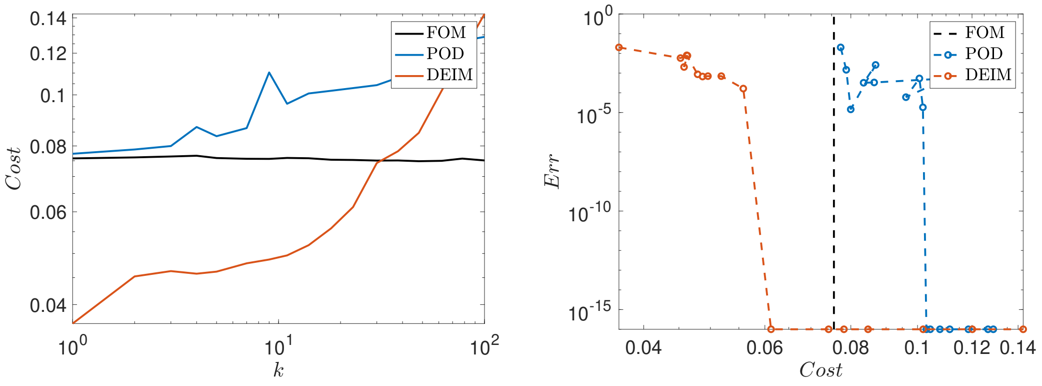
Figure 3.18 – Numerical results for the circadian oscillators model with fixed random parameters as a function of the coupling strengths. Contour plots of the quantities of interest. Synchrony parameter: (a) FOM, (b) POD, (c) DEIM. Spectral amplification factor: (d) FOM, (e) POD, (f) DEIM. Average period: (g) FOM, (h) POD, (i) DEIM. Asymptotic order parameter: (j) FOM, (k) POD, (l) DEIM.

variations are captured by the reduced models, at least from a visual perspective. However, compared to Figure 3.18 the errors are significantly larger. Although the synchrony parameter and average period are slightly underestimated, particularly with small intercellular coupling strenghts, their values have a similar order of magnitude as the full model. The order parameter captures the transition to synchronization well, with larger differences for weak external



(a) Asymptotic order parameter ($K = K_{max}, L_0 = L_{0,max}$).

(b) Relative error ($K = K_{max}, L_0 = L_{0,max}$) of the asymptotic order parameter.



(c) Computational cost.

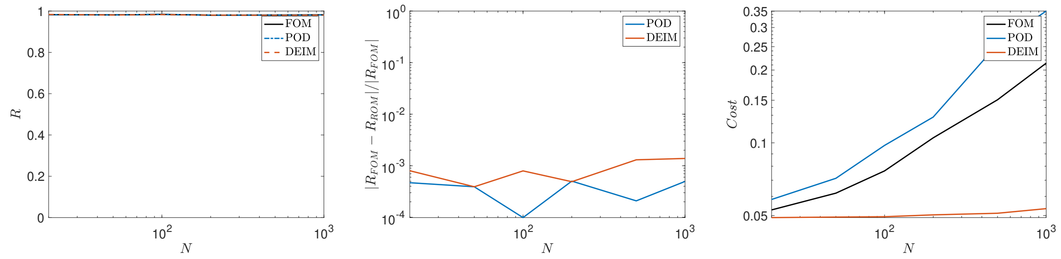
(d) Error vs computational cost.

Figure 3.19 – Numerical results for the circadian oscillators model with fixed random parameters and variable reduced dimension k.

sources. As already observed in the Kuramoto model, the loss in the accuracy of the reduced models is caused by the heterogeneity of the model, as the random variables are independently sampled. One could indeed verify that convergence with respect to sampling is reached with a satisfactory accuracy. Consistent with Figures 3.6(b) and 3.14(b), the discrepancies between the full and the reduced model are enhanced for small coupling strengths.

3.3.3 A control problem

Finally, we solve a control problem based on the Kuramoto model. We seek to show that the proposed technique finds application in more general problems, and is not limited to recovering the synchronization patterns of the system of interest. Extensions to more complex problems, e.g., topology inference, suppression of the oscillations, control problems with the circadian clock model, go beyond the scope of this work. We define the optimization problem



(a) Asymptotic order parameter ($K = K_{max}, L_0 = L_{0,max}$).

(b) Relative error ($K = K_{max}, L_0 = L_{0,max}$) of the asymptotic order parameter.

(c) Computational cost.

Figure 3.20 – Numerical results for the circadian oscillators model with fixed random parameters and variable full dimension N.

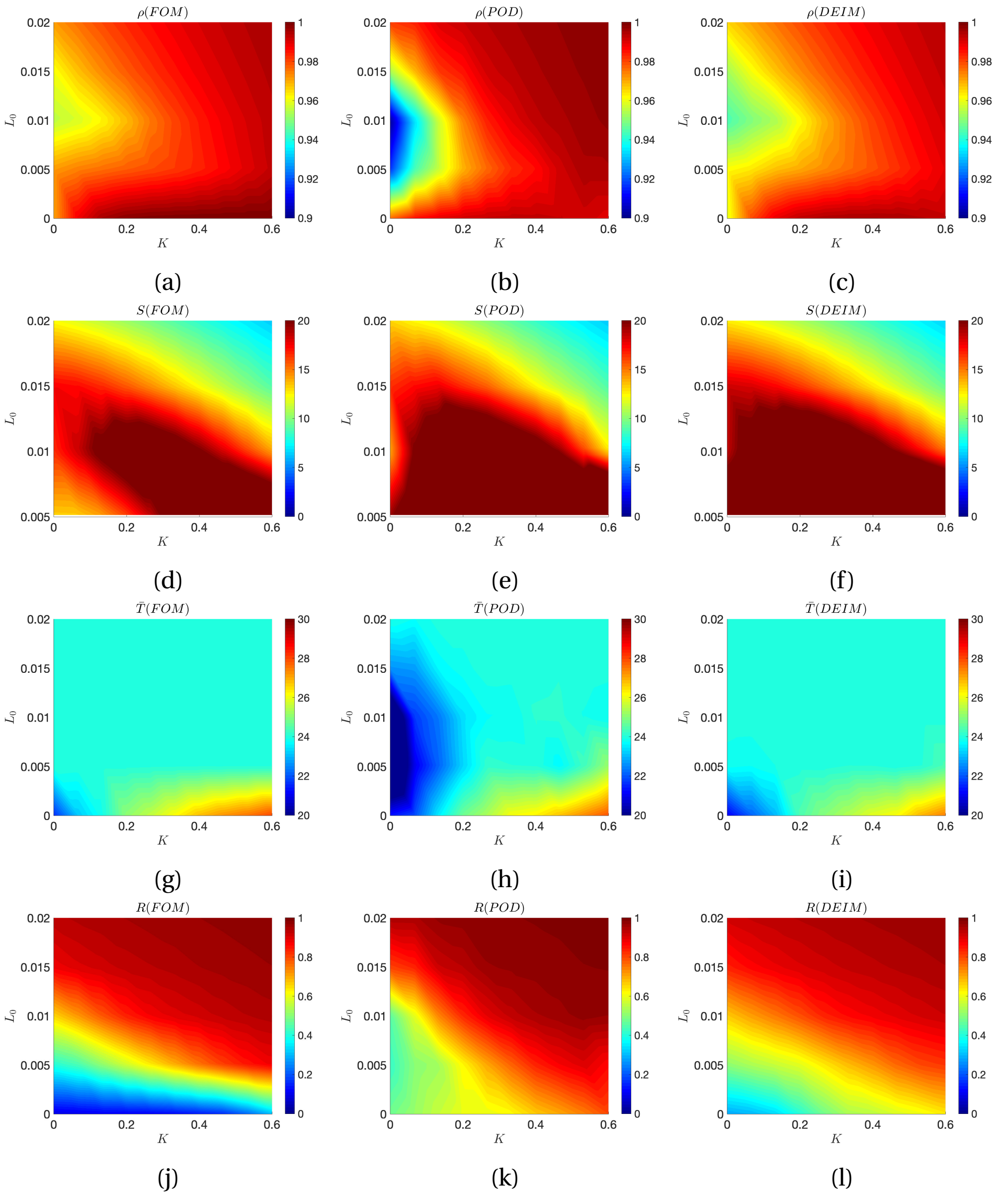
as

$$\min_{K\in[0,K_{max}]} \mathscr{L}(K) = \sqrt{\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\phi_j(T;K)-\phi_i(T;K))^2} + 200K^2, \tag{3.20}$$

where we stress the dependence of the phases ϕ_i on the optimization parameter K according to the Kuramoto model (3.2). The loss (3.20) can be interpreted as a minimization of the phase variability at time T, with a penalty term on the value of the coupling strength. We optimize (3.20) using gradient descent with learning rate $\eta = 10^{-4}$, where the derivatives are estimated using first order forward finite differences with spacing $\epsilon = 10^{-3}$. We select the number of gradient descent iteration as $N_{\text{iter}} = 50$. The other hyperparameters related to the Kuramoto model are kept as in Section 3.3.1. Figures 3.22(a), 3.22(b), 3.22(c) and 3.22(d) show the results obtained with starting point $K(0) = 0$. The full and reduced models converge to the same optimum. Consistent with Figure 3.2(e), the error obtained with the DEIM is slightly larger as compared to the POD model, which in turn shows good agreement with the full model. Similar to Figures 3.3(e) and 3.4(e), the DEIM ensures a computational saving of approximately 30%. These conclusions are not altered if the starting point of the optimization algorithm is changed. Specifically, Figures 3.22(e), 3.22(f), 3.22(g) and 3.22(h) show the results starting from $K(0) = K_{max}$. All models converge to the same optimum, with minor differences between the full and the reduced models. The computational saving is again around 30%. We recall that the global nature of the coupling term in the Kuramoto model prevents larger computational savings. Indeed, the full potential of the reduced models is triggered by the change of coordinates (3.6), consistent with Section 3.3.1. Thus, defining a control problem based on the modified Kuramoto model would lead to larger speedups, as already observed in Figure 3.12(c).

3.4 Extension: sensitivity analysis

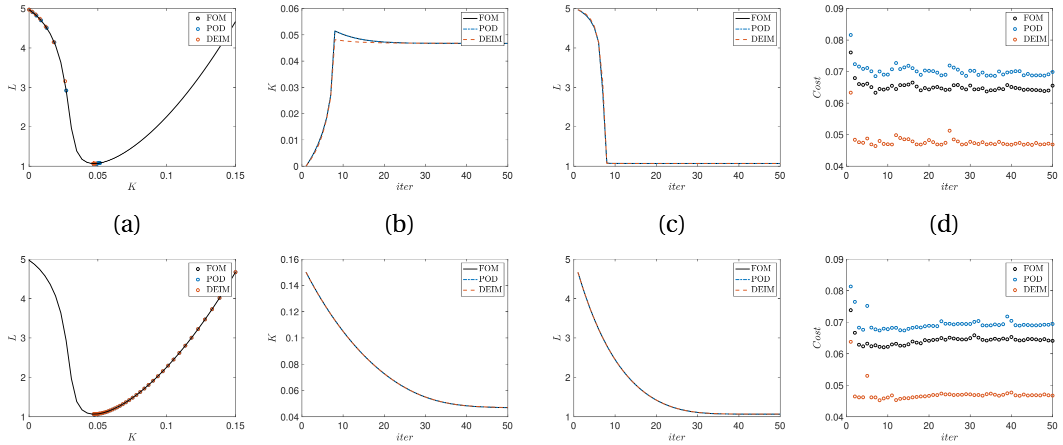
To construct the reduced order model of the circadian clock system (3.8), we decided a-priori that our parameters of interest were the coupling strengths. This choice was dictated by the fact



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Figure 3.21 – Numerical results for the circadian oscillators model with Monte Carlo simulation as a function of the coupling strengths. Contour plots of the quantities of interest. Synchrony parameter: (a) FOM, (b) POD, (c) DEIM. Spectral amplification factor: (d) FOM, (e) POD, (f) DEIM. Average period: (g) FOM, (h) POD, (i) DEIM. Asymptotic order parameter: (j) FOM, (k) POD, (l) DEIM.

that emergent phenomena are a consequence of the interactions, and a complete knowledge of the local models can be assumed. However, one might be interested in studying a case in which no a-priori assumption is made, and all parameters are allowed to vary. In principle, the same procedure described in (3.2) could be repeated, provided that an appropriate sampling technique of the parameters of (3.8) is available. However, the resulting reduced model



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.22 – Numerical results for the control problem based on the Kuramoto model. Loss landscape and optimization path ((a) $K(0) = 0$, (e) $K(0) = K_{max}$). Optimal parameter vs number of iterations ((b) $K(0) = 0$, (f) $K(0) = K_{max}$). Loss function vs number of iterations ((c) $K(0) = 0$, (g) $K(0) = K_{max}$). Computational cost ((d) $K(0) = 0$, (h) $K(0) = K_{max}$).

would not be accurate, as capturing variations in the entire parameter space can become challenging due to the high dimensionality of the parameter space. A wiser approach consists in performing a sensitivity analysis on the parameters to understand which are the most relevant ones for the dynamics of the problem and the synchronization patterns of the system. The results would be used to rank the parameters according to their importance, and vary only these in the training phase. We refer to [83] for more details on the subject.

We consider a variance-based sensitivity analysis based on first-order Sobol' indices. Take a parametrized model

$$Y = \mathscr{F}(\mu), \tag{3.21}$$

where Y is the model output of interest, $\mathscr{F}(\cdot)$ is a possibly unknown function, and $\mu = (\mu_1, \ldots, \mu_d)$ is a vector of input parameters chosen in a d-dimensional hypercube. Denoting with $\mathbb{V}[\cdot]$ and $\mathbb{E}[\cdot]$ the variance and the expectation operators, one can verify that

$$\mathbb{V}(Y) = \sum_i V_i + \sum_i \sum_{j>i} V_{ij} + \ldots + V_{1,2\ldots,d}, \tag{3.22}$$

where

$$V_i = \mathbb{V}_{\mu_i}\left[\mathbb{E}_{\mu_{\sim i}}\left[Y \mid \mu_i\right]\right],$$
$$V_{ij} = \mathbb{V}_{\mu_{ij}}\left[\mathbb{E}_{\mu_{\sim ij}}\left[Y \mid \mu_i, \mu_j\right]\right] - V_i - V_j,$$

and so on. Here, $\mu_{\sim i}$ denotes the set of all parameters except μ_i. The decomposition (3.22) shows that the variance of the model output can be decomposed into terms attributable to each input, as well as the interaction effects between them. This leads to the definition of the

μ_i	Baseline	Interval
ν_1	0.7	$[0.6, 0.8]$
ν_2	0.35	$[0.3, 0.4]$
ν_4	0.35	$[0.3, 0.4]$
ν_6	0.35	$[0.3, 0.4]$
ν_8	1	$[0.85, 1.15]$
K_1	1	$[0.85, 1.15]$
K_2	1	$[0.85, 1.15]$
K_4	1	$[0.85, 1.15]$
K_6	1	$[0.85, 1.15]$
K_8	1	$[0.85, 1.15]$
k_3	0.7	$[0.6, 0.8]$
k_5	0.7	$[0.6, 0.8]$
k_7	0.35	$[0.3, 0.4]$
ν_c	0.4	$[0.3, 0.5]$
K_c	1	$[0.9, 1.1]$
K	0.5	$[0.4, 0.6]$
L_0	0.01	$[0.005, 0.02]$
$\Omega = 2\pi/\omega$	24	$[20, 28]$

Table 3.1 – Variation of the parameters of the circadian clock model.

first-order indices

$$S_i = \frac{V_i}{\mathbb{V}(Y)} \in [0, 1], \tag{3.24}$$

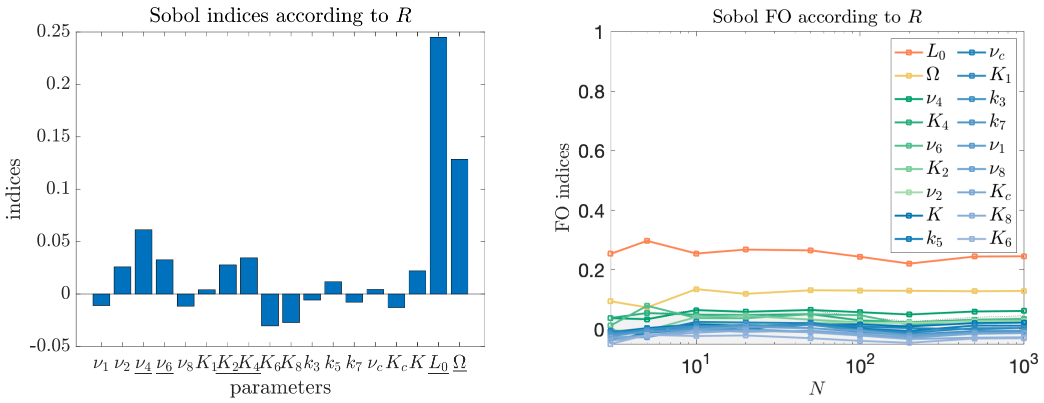
which quantify the effect of μ_i averaged over the other input parameters, ignoring higher-order interactions. Thus, the parameters with higher sensitivity index (3.24) can be classified as the more relevant ones for the model (3.21), as they explain a larger part of the total variance.

The circadian clock system (3.8) fits into this framework. The model parameters are reported in Table 3.1, whereas the model output of interest is the asymptotic order parameter (3.13). The Sobol' sensitivity analysis gives the results shown in Figure 3.23. Out of the 18 model parameters, L_0 and Ω are the most influential ones, with a large gap with the others. Recalling (3.8), this implies that the order parameter is mostly influenced by the external coupling, both through the strength and the frequency. This suggests that only the resulting two-dimensional space should be sampled to construct the reduced model. Moreover, Figure 3.23(b) shows that this discrepancy does not depend on the size of the problem, so that the sensitivity analysis can be done with small network sizes at a controlled computational cost with good generalization properties. The presence of small negative values in Figure 3.23 is caused by the estimation procedure [112], but their values are found not to be statistically significantly different from zero.

Constructing the corresponding reduced model, its accuracy is confirmed by Figure 3.24, at least visually.

3.5 Conclusion

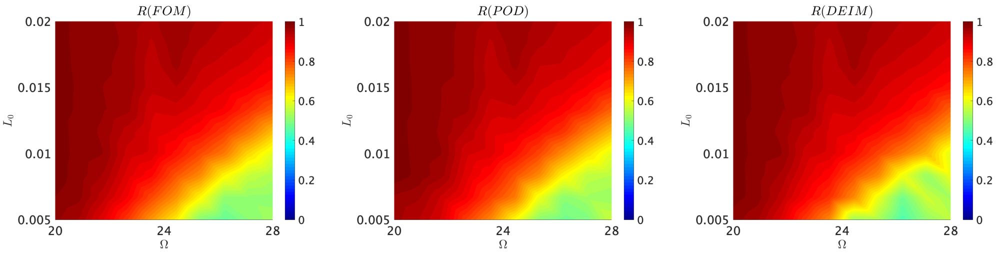
In this chapter we constructed reduced order models of globally coupled oscillatory systems exhibiting synchronization. Although computational efficiency can be challenging to achieve for globally coupled models, the assumption of a mean-field coupling ensures that the online simulations of the reduced models have a computational cost independent of the full



(a) $N = 10^3$.

(b) Vs N.

Figure 3.23 – First-order Sobol' indices of the circadian clock model.



(a) FOM.

(b) POD.

(c) DEIM.

Figure 3.24 – Numerical results (asymptotic order parameter) for the circadian oscillators model with a sensitivity analysis. Contour plots of the quantity of interest.

dimension.

Numerical results are presented for two classes of prototype systems in the context of synchronization. The Kuramoto model is studied both in the original formulation and using a suitable change of coordinates. In both cases, the system possesses significant reduction properties. Transition to synchronization is captured at the reduced level with sufficient accuracy, even for a small number of basis functions. Although a moderate computational speedup is achieved in most cases, the proposed change of coordinates leads to a reduced model whose cost is independent of the dimension of the full model. This triggers the full potential of the reduced model, at least after using DEIM. The circadian clock model shows a similar behavior. All quantities of interest are well approximated, and computational efficiency is achieved even in its original formulation. The chapter is concluded with a control problem based on the Kuramoto model, showing the applicability of model reduction techniques in optimization problems, and a sensitivity analysis for the circadian clock model, showing that the parameter space is effectively two-dimensional.

4 The Huygens' problem

In this chapter we focus on synchronization phenomena arising from a coupled mechanical system. The considered model is designed to recreate the Huygens' experiment [72] discussed in Chapter 1 by coupling pendulum clocks with a wooden structure. Similar to Chapter 3, the main goal is to recover the synchronization patterns at the reduced level. Moreover, by going beyond the original physical model and modifying the coupling functions, we also aim at possibly discovering new patterns. As discussed in Chapter 1, we aim at constructing a reduced model without simulating the coupled problem. This avoids solving a high-dimensional nonlinear system for a large number of time steps and, possibly, different physical parameter values. The main idea is to decouple the structure and the pendula, and construct a surrogate for the former using either a model-based approach inspired by the underlying differential model or an artificial parametrization of the load. Again, we restrict ourselves to intrusive methods.

Before discussing the mathematical model and a number of model reduction methods that could be considered, we point out that the unexpected behavior observed by Huygens led to several follow-up works. At an experimental level, we mention the observations by W. Ellis [49], who noticed a loss of synchronization if more than two pendula are considered. At a theoretical level, several models have been constructed with the goal of both explaining Huygens' observations and suggesting other types of synchronization [41, 33, 77, 32]. These studies have been followed by numerical simulations, in which the simplifying modeling assumptions were relaxed [34, 100] and even more patterns have been discovered, including amplitude death and intermittency.

Most of these models introduce problem-dependent parameters, whose variations lead to different patterns. The willingness to discover them motivates the need of model order reduction. In structural dynamics contexts, several techniques have been proposed in this regard [52, 31, 51, 137, 30]. The main idea is to construct a local basis that captures both the static and the dynamic behavior, often by splitting internal and interface degrees of freedom. This can be done by, e.g., studying both the system response in the presence of unitary boundary displacements and internal eigenvalue problems. Additionally, interface reduction methods [81] aim to further reduce the dimensionality at the component interfaces by using basis functions related to a secondary interface eigenvalue analysis. These techniques rely on a model-driven approach, as they use the model at hand or some simplifications to construct the basis. Moreover, they are prone to theoretical studies and their numerical efficiency is well established. We apply one of these techniques to our problem of interest.

However, the generalization of these methods to more complex problems, including the models discussed in Chapter 5, can be challenging. This is particularly true when multi-physics problems, possibly characterized by nonlinearities and a nontrivial parameter dependence, are considered. Thus, we propose an alternative data-driven approach based on an artificial parametrization of the boundary data. This shares some similarities with [75], but handles problems in which the dynamical part plays a significant role. Another similar approach is proposed in [140], although targeted to a different class of problems. A more complete discussion on these methods, which are often targeted to coupled partial differential equations, can be found in Chapter 5.

The remainder of this chapter is organized as follows. Section 4.1 introduces the mathematical formulation of the full order model. Section 4.2 presents two model reduction techniques for the structural part, which are numerically tested in Section 4.3. The cases of a parameter-dependent structure and a generalized model are discussed in Section 4.4. A brief final discussion is added in Section 4.5.

4.1 Mathematical model

The model described in this section is inspired by [100, 77]. We first consider two pendula, and leave the extension to a larger number to the end of the section. Unless stated otherwise, we use the International System of Units (SI) for all the physical variables.

4.1.1 Structure

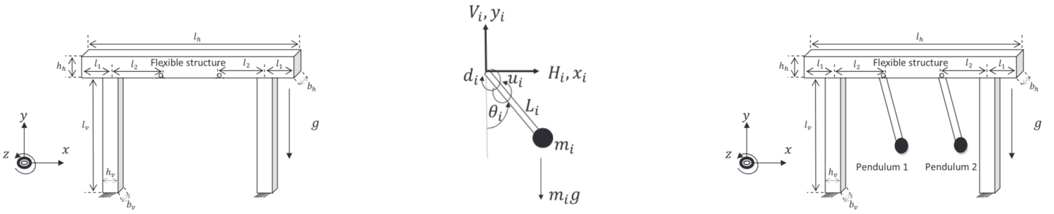
The structure is sketched in Figure 4.1(a) and it is characterized by the physical parameters reported in Table 4.1. The lengths l_1 and l_2 are related to the horizontal offsets of the vertical beams and the location of the points where external loads can be placed, and their values are set to

$$l_1 = 0.0247, \quad l_2 = 0.8161,$$

respectively. The structure is modeled using the Euler-Bernoulli beam theory, so that the governing equations of each beam read

$$\rho A \frac{\partial^2 u}{\partial t^2} - \frac{\partial}{\partial \xi}\left(EA\frac{\partial u}{\partial \xi}\right) = p_u(\xi), \tag{4.1a}$$

$$\rho A \frac{\partial^2 w}{\partial t^2} + \frac{\partial^2}{\partial \xi^2}\left(EI\frac{\partial^2 w}{\partial \xi^2}\right) = p_w(\xi), \tag{4.1b}$$



(a) Structure. (b) Pendula. (c) Coupled model.

Figure 4.1 – The setup of the Huygens' problem [100].

Property	Horizontal beam	Each vertical beam
Length	$l_h = 2.4484$	$l_v = 1$
Width	$b_h = 0.550$	$b_v = 0.040$
Thickness	$h_h = 0.0762$	$h_v = 0.030$
Density	$\rho_h = 560$	$\rho_v = \rho_h$
Young's modulus	$E_h = 8.963 \cdot 10^9$	$E_v = E_h$

Table 4.1 – Physical parameters of the structure.

where ξ is the spatial coordinate along each beam, ρ is the density, A is the cross-sectional area, E is the Young's modulus, and I is the area moment of inertia, while u and w denote the axial and transversal displacement of the beam, and p_u, p_w represent the external loads. Equations (4.1a) and (4.1b) are discretized using Lagrange (linear) and Hermite (cubic) finite elements, respectively, and we choose $N_h = 99$ and $N_v = 100$ beam elements for the horizontal and each vertical beam. As the finite element formulation entails some intrinsic complexities, we provide more details in Appendix 4.A. Here, we simply remark that our choice implies that each node is associated with three degrees of freedom, modeling the axial displacement, the transversal displacement, and the rotation $\partial w / \partial \xi$. Moreover, we assume that the vertical beams are fixed to the ground at the bottom, whereas at the two contact points between the horizontal beam and the two vertical beams, the axial (reps. transversal) displacement of the horizontal beam correspond to the transversal (resp. axial) displacement of the vertical ones. Inspired by the original Huygens' setup, no coupling between the rotational degrees of freedom of the horizontal and vertical beams are imposed. Thus, we obtain a total number of degrees of freedom equal to $N = 3 \cdot (100 + 2 \cdot 101) - 6 - 4 = 896$. The corresponding finite element model can be written as

$$\mathbf{M}\ddot{q} + \mathbf{B}\dot{q} + \mathbf{K}q = f, \tag{4.2}$$

where $q \in \mathbb{R}^N$ contains the degrees of freedom, and $\mathbf{M} \in \mathbb{R}^{N \times N}, \mathbf{B} \in \mathbb{R}^{N \times N}$, $\mathbf{K} \in \mathbb{R}^{N \times N}$ are the global mass, damping and stiffness matrices. The source term $f \in \mathbb{R}^N$ contains the external loads. Unlike (4.1), the discretized model (4.2) already includes damping, whose specific expression is discussed at the end of this section. Without loss of generality, we assume that the degrees of freedom are ordered in such a way that the first $2 \cdot 2$ entries represent the horizontal and vertical displacements of the nodes at which the pendula are attached, denoted by x_i and y_i, respectively, i.e.,

$$q = \left[x_1, y_1, x_2, y_2, q_I^T\right]^T, \tag{4.3}$$

where q_I contains the remaining internal degrees of freedom. Similarly, f takes the form

$$f = \left[f_{x1}, f_{y1}, f_{x2}, f_{y2}, \mathbf{0}^T\right]^T, \tag{4.4}$$

where f_{xi} and f_{yi} denote the horizontal and vertical forces exerted at the boundary point i.

A modal analysis is performed in order to determine the undamped eigenfrequencies and corresponding eigenmodes. This is achieved by solving the eigenvalue problem

$$\left[\mathbf{K} - \omega_i^2 \mathbf{M}\right] \varphi_i = 0, \quad i = 1, \ldots, N, \tag{4.5}$$

where $\omega_i = 2\pi f_i$ is the i-th undamped angular eigenfrequency and f_i is the corresponding frequency. The angular frequencies and the modes can be gathered in the matrices $\mathbf{\Omega} = \mathrm{diag}(\omega_1, \ldots, \omega_N)$ and $\mathbf{\Phi} = [\varphi_1, \ldots, \varphi_N]$, respectively, where an increasing ordering of the frequencies is assumed. The normalization is set according to $\mathbf{\Phi}^T \mathbf{M} \mathbf{\Phi} = \mathbf{I}$, so that $\mathbf{\Phi}^T \mathbf{K} \mathbf{\Phi} = \mathbf{\Omega}^2$. Assuming proportional damping, the damping matrix $\mathbf{B}$ in (4.2) is computed according to

$$\mathbf{B} = 2\mathbf{\Phi}^{-T} \mathbf{Z} \mathbf{\Omega} \mathbf{\Phi}^{-1},$$

where $\mathbf{Z} = \mathrm{diag}(\zeta_1, \ldots, \zeta_N)$, with ζ_i being the damping coefficient for eigenmode i. Here, we choose $\zeta_i = \zeta = 0.05$.

4.1.2 Pendula

Following Figure 4.1(b), each pendulum i is modeled by a point mass of mass m_i attached at the lower end of a massless rigid bar of length L_i. The damping in each pendulum is assumed to be viscous, linear, and concentrated in the revolute joint which couples the pendulum to the coupling structure. The rotational damping coefficient is denoted by d_i. The dynamical behavior of each pendulum, if it would be coupled to the fixed world, is therefore described by

$$\ddot{\theta}_i = -\frac{g}{L_i} \sin\theta_i - \frac{d_i}{m_i L_i^2} \dot{\theta}_i + \frac{u_i}{m_i L_i^2},$$

where $g = 9.81$ is the gravitational acceleration and θ_i is the rotation angle of pendulum i. The function $u_i = u_i(\theta_i, \dot{\theta}_i)$ models the so-called escapement mechanism. In spirit, this represents an additional source of energy provided to the pendula in order to compensate for the loss of energy due to friction and maintain its oscillatory nature. Its design can pose challenges from a modeling and a numerical perspective [108], and can highly affect the dynamics of the system [41]. Inspired by [77], we model it using a simple Van der Pol-like term that admits a limit cycle,

$$u_i(\theta_i, \dot{\theta}_i) = e_i(\gamma_i^2 - \theta_i^2)\dot{\theta}_i, \tag{4.6}$$

where e_i measures the strength of the escapement and γ_i is the critical angle, at which the escapement switches from boosting to damping. Although (4.6) does not model well the impulsive nature of the escapement, on average it gives a good approximation when the pendula are close to their limit cycles [97]. We consider $e_i = 2.1043$ and $\gamma_i = 0.122$ for both pendula.

4.1.3 The Huygens' model

The coupled structure-pendula model can be derived using balance equations for the forces and moments. The pendula are attached to the structure using revolute joints located at $l_1 + l_2$ from the endpoints of the horizontal beam as in Figure 4.1. The horizontal motion is described by

$$m_i \frac{d^2}{dt^2}\left(x_i + L_i \sin\theta_i\right) = H_i, \tag{4.7}$$

where H_i is the horizontal force exerted by the structure to the upper side of the pendulum and x_i is defined in (4.3) and Figure 4.1(b). Similarly, the vertical motion is described by

$$m_i \frac{d^2}{dt^2}\left(y_i - L_i \cos\theta_i\right) + m_i g = V_i, \tag{4.8}$$

where V_i is the vertical force exerted by the structure to the upper side of the pendulum and y_i is defined in (4.3) and Figure 4.1(b). Equilibrium of moments with respect to the out-of-plane axis going through point mass m_i results in

$$0 = -H_i L_i \cos\theta_i - V_i L_i \sin\theta_i - d_i \dot{\theta}_i + u_i, \tag{4.9}$$

By combining (4.7), (4.8) and (4.9), we obtain

$$m_i L_i^2 \ddot{\theta}_i = -m_i L_i \ddot{x}_i \cos\theta_i - m_i \ddot{y}_i L_i \sin\theta_i - m_i g L_i \sin\theta_i - d_i \dot{\theta}_i + u_i.$$

Therefore, the dynamical model of the system depicted in Figure 4.1 is obtained by equating the forces exerted by the structure to the pendula (H_i, V_i) and the load exerted by the pendula to the structure $(-f_{xi}, -f_{yi})$ appearing in the vector f in (4.2). It reads

$$\mathbf{M}\ddot{q} + \mathbf{B}\dot{q} + \mathbf{K}q = f, \tag{4.10a}$$
$$m_i L_i^2 \ddot{\theta}_i = -m_i L_i \ddot{x}_i \cos\theta_i - m_i \ddot{y}_i L_i \sin\theta_i - m_i L_i \sin\theta_i - d_i \dot{\theta}_i + u_i,$$

where $f = \left[-H_1, -V_1, -H_2, -V_2, \mathbf{0}^T\right]^T$ and (H_i, V_i) are defined in (4.7) and (4.8). Considering the explicit expressions of (H_i, V_i) and rearranging the terms, we obtain

$$\mathbf{M}_C(\bar{\mathbf{q}})\ddot{\bar{\mathbf{q}}} + \mathbf{B}_C\dot{\bar{\mathbf{q}}} + \mathbf{K}_C\bar{\mathbf{q}} = \mathbf{F}_C(\bar{\mathbf{q}}), \tag{4.11}$$

where we introduced the global unknown vector

$$\bar{\mathbf{q}} = \left[q^T, \theta_1, \theta_2\right]^T. \tag{4.12}$$

The specific expression of the terms in (4.11) are

$$\mathbf{M}_C = \begin{bmatrix} \mathbf{M} & 0 & 0 \\ 0 & m_1 L_1^2 & 0 \\ 0 & 0 & m_2 L_2^2 \end{bmatrix} + \mathbf{M}_c,$$

where

$$\mathbf{M}_c = \begin{bmatrix} m_1 & 0 & 0 & 0 & \mathbf{0}^T & m_1 L_1 \cos\theta_1 & 0 \\ 0 & m_1 & 0 & 0 & \mathbf{0}^T & m_1 L_1 \sin\theta_1 & 0 \\ 0 & 0 & m_2 & 0 & \mathbf{0}^T & 0 & m_2 L_2 \cos\theta_2 \\ 0 & 0 & 0 & m_2 & \mathbf{0}^T & 0 & m_2 L_2 \sin\theta_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{00}^T & \mathbf{0} & \mathbf{0} \\ m_1 L_1 \cos\theta_1 & m_1 L_1 \sin\theta_1 & 0 & 0 & \mathbf{0}^T & 0 & 0 \\ 0 & 0 & m_2 L_2 \cos\theta_2 & m_2 L_2 \sin\theta_2 & \mathbf{0}^T & 0 & 0 \end{bmatrix}$$

for the mass matrix,

$$\mathbf{B}_C = \begin{bmatrix} \mathbf{B} & 0 & 0 \\ 0 & d_1 & 0 \\ 0 & 0 & d_2 \end{bmatrix}$$

for the damping matrix,

$$\mathbf{K}_C = \begin{bmatrix} \mathbf{K} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

for the stiffness matrix, and

$$\mathbf{F}_C = \begin{bmatrix} m_1 L_1 \sin\theta_1 \dot{\theta}_1^2 \\ -m_1 L_1 \cos\theta_1 \dot{\theta}_1^2 - m_1 g \\ m_2 L_2 \sin\theta_2 \dot{\theta}_2^2 \\ -m_2 L_2 \cos\theta_2 \dot{\theta}_2^2 - m_2 g \\ \mathbf{0} \\ -m_1 L_1 g \sin\theta_1 \\ -m_2 L_2 g \sin\theta_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{0} \\ u_1 \\ u_2 \end{bmatrix}$$

for the nonlinear source term.

4.1.4 More pendula

This setup can be easily extended to construct a physical model with $N_p > 2$ pendula. In spirit, the physical description remains unchanged, while the vector of unknowns (4.12) becomes

$$\bar{\mathbf{q}} = \left[x_1, y_1, \dots, x_{N_p}, y_{N_p}, q_I^T, \theta_1, \dots, \theta_{N_p}\right]^T.$$

The length l_1 remains unchanged, while the pendula are equally spaced with a relative distance of $l_2 = \frac{l_h - 2\cdot l_1}{N_p+1}$. If l_2 does not correspond to a discretization node, the closest one is picked.

4.2 Localized model reduction

Although the problem has a component-wise structure, only the structural part requires a dimensionality reduction, as the each pendulum is modeled with a scalar ordinary differential equation. As a prototype test case for localized reduced models, we compare different techniques that construct the surrogate using the decoupled problem (4.2) instead of its coupled counterpart (4.10a). We stick to the case $N_p = 2$, with straightforward generalization if this number is increased.

4.2.1 Component Mode Synthesis

The first method, inspired by [100, 31, 52], is a component mode synthesis (CMS) technique based on free-interface eigenmodes and residual flexibility modes. Recalling (4.3), the state vector is partitioned as

$$q = \begin{bmatrix} q_B \\ q_I \end{bmatrix},$$

where $q_B = [x_1, y_1, x_2, y_2]^T \in \mathbb{R}^{n_B}$ contains the boundary dofs and $q_I \in \mathbb{R}^{n_I}$ contains the internal dofs. Clearly, we have $n_B = 4$ and $n_I = 892$.

Firstly, a reduction from the N original physical dofs to $n_B + n_K$ generalized dofs is performed by

$$q = \mathbf{T}_1 p_1,$$

where

$$p_1 = \begin{bmatrix} p_B \\ p_K \end{bmatrix}$$

and

$$\mathbf{T}_1 = \begin{bmatrix} \boldsymbol{\phi}_B & \boldsymbol{\phi}_K \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}_{BB} & \boldsymbol{\phi}_{BK} \\ \boldsymbol{\phi}_{IB} & \boldsymbol{\phi}_{IK} \end{bmatrix}.$$

Here, the matrix $\boldsymbol{\phi}_K = \boldsymbol{\phi}(:, 1 : n_K) \in \mathbb{R}^{N\times n_K}$ contains the smallest n_k elastic eigenmodes of the undamped problem (4.5), which capture frequencies up to ω_{n_K}. The matrix $\boldsymbol{\phi}_B \in \mathbb{R}^{n\times n_B}$ contains residual flexibility modes, which provide a static correction for the neglected higher frequency eigenmodes. These are defined as

$$\boldsymbol{\phi}_B = \left[\mathbf{K}^{-1} - \boldsymbol{\phi}_K \boldsymbol{\Omega}_{KK}^{-2} \boldsymbol{\phi}_K^T\right] \begin{bmatrix} \mathbf{I}_{BB} \\ \mathbf{O}_{IB} \end{bmatrix},$$

where $\boldsymbol{\Omega}_{KK} = \boldsymbol{\Omega}(1:n_K, 1:n_K)$.

Secondly, for the sake of easy coupling between the structure and the pendula, the generalized dofs p_1 are replaced by the dofs p according to

$$p_1 = \mathbf{T}_2 p,$$

where

$$p = \begin{bmatrix} q_B \\ p_K \end{bmatrix}$$

and

$$\mathbf{T}_2 = \begin{bmatrix} \boldsymbol{\phi}_{BB}^{-1} & -\boldsymbol{\phi}_{BB}^{-1}\boldsymbol{\phi}_{BK} \\ \mathbf{0}_{KB} & \boldsymbol{I}_{KK} \end{bmatrix}.$$

Finally, the projection matrix is obtained by

$$\mathbf{T} = \mathbf{T}_1\mathbf{T}_2 = \begin{bmatrix} \boldsymbol{I}_{BB} & \mathbf{0}_{BK} \\ \boldsymbol{\phi}_{IB}\boldsymbol{\phi}_{BB}^{-1} & -\boldsymbol{\phi}_{IB}\boldsymbol{\phi}_{BB}^{-1}\boldsymbol{\phi}_{BK} + \boldsymbol{\phi}_{IK} \end{bmatrix} \tag{4.13}$$

and the reduced equations of motion for the structure become

$$\underbrace{\mathbf{T}^T\mathbf{M}\mathbf{T}}_{\mathbf{M}_r}\ddot{p} + \underbrace{\mathbf{T}^T\mathbf{B}\mathbf{T}}_{\mathbf{B}_r}\dot{p} + \underbrace{\mathbf{T}^T\mathbf{K}\mathbf{T}}_{\mathbf{K}_r}p = \underbrace{\mathbf{T}^T f}_{f_r}. \tag{4.14}$$

Unless stated otherwise, we keep only the first five eigenmodes, i.e., $n_k = 5$. For the considered parameter values, the reduced model will be accurate up to approximately $\omega_c = 2\pi \cdot 90$, which is clearly above the average pendulum frequency over the considered configurations, as it lies at around 2π. The reduced model (4.14) of the structure will therefore have $n_B + n_K = 9$ dimensions.

4.2.2 Proper Orthogonal Decomposition

The second technique is based on a reduced basis, data-driven approach. Looking at the coupled model, (4.10a) has formally the same structure as its decoupled counterpart (4.2), the only difference being the dependence of the forcing term on the pendula. The idea is to consider (4.2) and artificially parametrize the load in order to collect snapshots of the solution of the structure, and compute the basis using the Proper Orthogonal Decomposition.

Thus, we consider

$$\mathbf{M}\ddot{q} + \mathbf{B}\dot{q} + \mathbf{K}q = f \tag{4.15}$$

for a number of instances of the load term. Similar to (4.4), only the first four components of the load can take nonzero values. As we expect an oscillatory solution, the load terms are chosen as

$$f_i = f_i(t) = \sum_{j=1}^{N_f} \alpha_{j,i} \sin\big(\omega_j t\big), \quad i = 1, \dots, n_B,$$

where ω_j are suitably defined angular frequencies, that are chosen a priori in an interval $[\omega_{\min}, \omega_{\max}]$. The coefficients $\alpha_{j,i}$ model the strength of each sinusoidal coupling, whereas N_f is the number of selected frequencies. Taking S samples of the parameters $\alpha_{j,i}$ gives the instances of the loads f in (4.15).

Applying the POD to the obtained snapshots as discussed in Chapter 2 gives the reduced basis $\boldsymbol{T}$ and the Galerkin reduced model

$$\underbrace{\mathbf{T}^T\mathbf{M}\mathbf{T}}_{\mathbf{M}_r}\ddot{p} + \underbrace{\mathbf{T}^T\mathbf{B}\mathbf{T}}_{\mathbf{B}_r}\dot{p} + \underbrace{\mathbf{T}^T\mathbf{K}\mathbf{T}}_{\mathbf{K}_r}p = \underbrace{\mathbf{T}^T f}_{f_r}. \tag{4.16}$$

Note that the formal structure of (4.16) is the same as (4.14), the only difference being the way in which the projection matrix is constructed. We remark that, unlike (4.13) and (4.14), we do not enforce a specific structure to the reduced basis $\mathbf{T}$, i.e., we do not explicitly split the boundary and internal degrees of freedom and we do not impose q_B to be exactly preserved at the reduced level. This does not represent a concern, as confirmed at the numerical level.

Unless stated otherwise, we decide to take $N_f = 3$ sinusoidal modes logarithmically spaced in the range $[1, 10]$ and $S = 13$ instances of the coefficients, sampled based on Sobol' sequences. We also set the final time to $T = 10$. Because of the oscillatory nature of the problem, varying such hyperparameters does not significantly change the efficiency of the proposed technique, provided that the relevant frequencies are included in the snapshot matrix. In order to have comparable results with the CMS technique, we select $k = 10$ modes.

4.3 Numerical results

We first validate the methods by fixing the parameter values of both the structure and the pendula. The most relevant parameters have already been defined in Section 4.1, and only a number of physical parameters related to the pendula are yet to be specified. We take equal pendula, characterized by lengths $L_1 = L_2 = 0.2286$, damping coefficients $d_1 = d_2 = 0.01$ and masses $m_1 = m_2 = 0.25$. The time integration of (4.11) and its reduced counterparts is performed by recasting the problems as first-order-in-time systems and using the implicit Euler method until $T = 200$ with a time step of $\Delta t = 0.02$. The nonlinear systems are handled using the MATLAB in-built routine `fsolve` combined with the analytic expression of the Jacobian matrix. A homogeneous initial condition is chosen for the structure displacement and velocity, whereas the pendula start at

$$\theta_i(t=0) = A\sin\big(\phi_i\big), \quad \dot{\theta}_i(t=0) = A\sqrt{\frac{g}{L_i}}\cos\big(\phi_i\big),$$

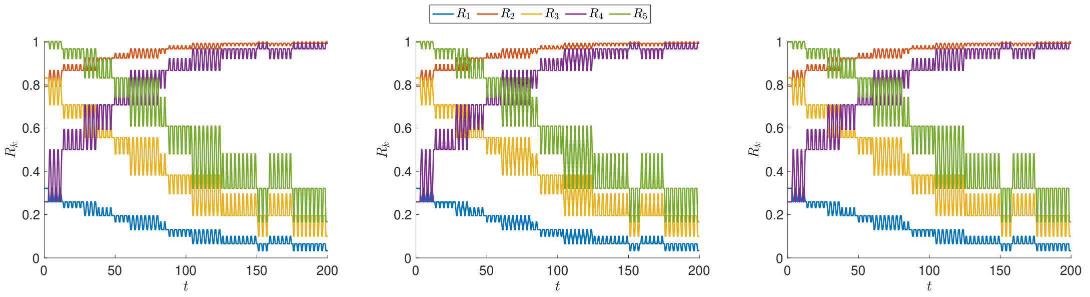
with $(\phi_1, \phi_2) = (\pi/2, -\pi/3)$ and $A = 0.2$. In order to analyze the synchronization patterns, we define K order parameters as

$$R_k = \left|\sum_{i=1}^{N_p} \exp\big(\imath k \varphi_i\big)\right| \in [0,1], \quad k = 1,\ldots,K. \tag{4.17}$$

Here, φ_i denotes the phase of each pendulum, which can be computed by estimating the local maxima of the signal θ_i and linear interpolation as in Chapter 3. The interpretation of (4.17) is similar to the Kuramoto model presented in Chapter 3, meaning that high values of the order parameters are associated to a coherent dynamics. The order parameters with $k > 1$ allow us to analyze more complex synchronization patterns, going beyond a simple distinction between full synchronization and incoherent dynamics. In this regard, they can be viewed as the higher-order counterparts of R_1. In this work we take $K = 5$.

In Figure 4.2 we report the order parameters as a function of time for the full and reduced models. For the selected parameter values, we see that the even order parameters R_2 and R_4 asymptotically take high values, whereas the odd order parameters R_1, R_3 and R_5 converge to values close to zero. This is a clear indication of anti-phase synchronization, which was the behavior observed by Huygens. Comparing the full and the reduced models, no differences are present, meaning that the surrogates are able to retrieve the behavior observed at a full level. This can be quantitatively confirmed by computing the corresponding reduction errors, which typically attain extremely low values, say around $\mathcal{O}(10^{-8})$. This high accuracy can be explained by the fact that, provided that the reduced basis captures the oscillatory modes of the structure, no differences are present between the full and the reduced models. Thus, only the computational cost changes, and a speedup of order $\mathcal{O}(10^2) - \mathcal{O}(10^3)$ is observed for the presented test case. As a side remark, we note that the trajectory of the order parameters has some spurious oscillations. These are due to numerical effects arising from the estimation of the phase and the time discretization, and they do not entail any physical meaning. It is not surprising that these effects are more evident if high values of k in (4.17) are taken. These oscillations can be smoothed by a simple filtering technique or by further reducing the time step. As we are mostly interested in the asymptotic properties of the system, we do not make further effort on this matter.

Arguably, the major advantage of reduced order modeling is the possibility to efficiently solve the systems under variations in its parameters. Here, the focus is on the long-term behavior and stability regimes that can be observed if the characteristics of the pendula are changed. Firstly, we consider the same physical parameters, whereas we vary the initial condition of one of the two pendula. Specifically, we set $\phi_1 = \pi/2$ and vary ϕ_2 in the interval $[\pi/2, 5\pi/2]$. The results are shown in Figure 4.3. If the pendula start in an (anti-)symmetric configuration,



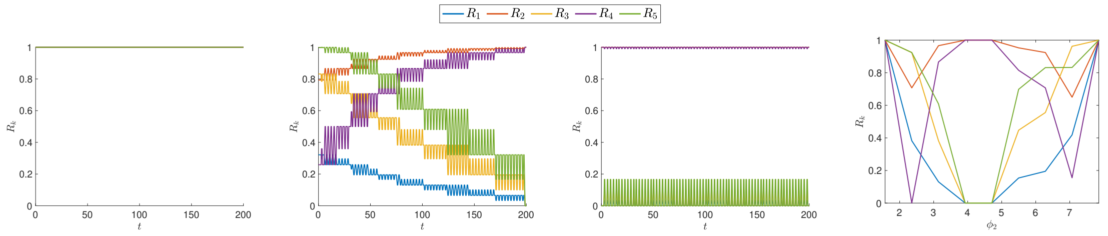
(a) FOM.

(b) CMS.

(c) POD.

Figure 4.2 – Order parameters vs time for the full and reduced models, $N_p = 2$.

this is maintained in the entire simulation (Figures 4.3(a) and 4.3(c)). These correspond to in-phase and anti-phase synchronization. If the initial condition lies between these two states, they asymptotically synchronize in anti-phase, with a transient period that increases with the distance with the perfect anti-phase condition (Figure 4.3(b)). These comments are confirmed by Figure 4.3(d), which shows the asymptotic order parameters as a function of the initial condition ϕ_2. The two regimes can clearly be identified, and the intermediate regions are characterized by intermediate values of R_k, suggesting that the synchronized state has not been reached yet.



(a) $\phi_2 = \pi/2$.

(b) $\phi_2 = 5\pi/4$.

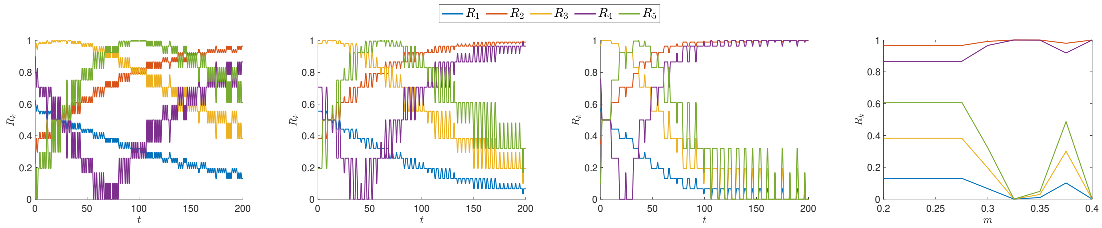
(c) $\phi_2 = 3\pi/2$.

(d) Asymptotic order parameters vs ϕ_2.

Figure 4.3 – Order parameters under variations of ϕ_2.

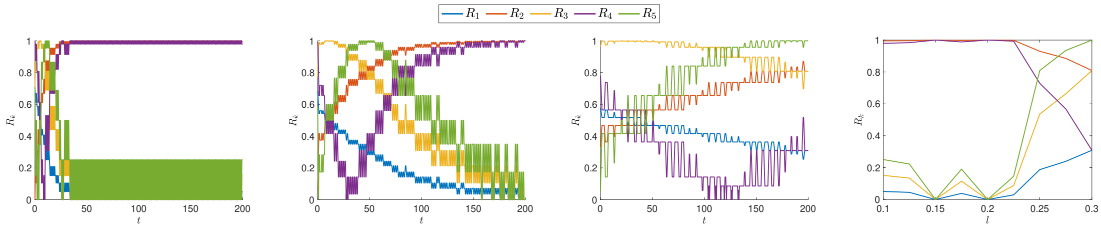
Considering variations in the mass of the two pendula $m_1 = m_2 = m$, we obtain the results shown in Figure 4.4. Changes in the mass do not modify the asymptotic state, which is of anti-phase synchronization. However, they affect the transient phase, as the equilibrium state is reached faster as the mass increases. This is consistent with theoretical analyses and numerical experiments with similar models [100, 77], in which the ratio between the mass of the pendula and the coupling structure plays the role of a coupling strength. The inconsistency around $m \simeq 0.375$ is again due to numerical effects, as one could verify by, e.g., looking at the time evolution of the order parameters.

A somehow opposite behavior is observed when the length $l_1 = l_2 = l$ is varied. Smaller lengths lead to shorter transient phases, as the interactions between the pendula and the structure tend to be stronger.



(a) $m = 0.2$. (b) $m = 0.3$. (c) $m = 0.4$. (d) Asymptotic order parameters vs m.

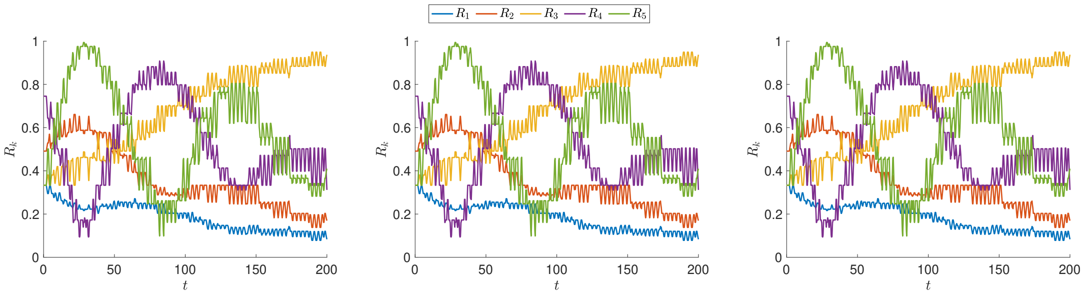
Figure 4.4 – Order parameters under variations of m.



(a) $l = 0.1$. (b) $l = 0.2$. (c) $l = 0.3$. (d) Asymptotic order parameters vs l.

Figure 4.5 – Order parameters under variations of l.

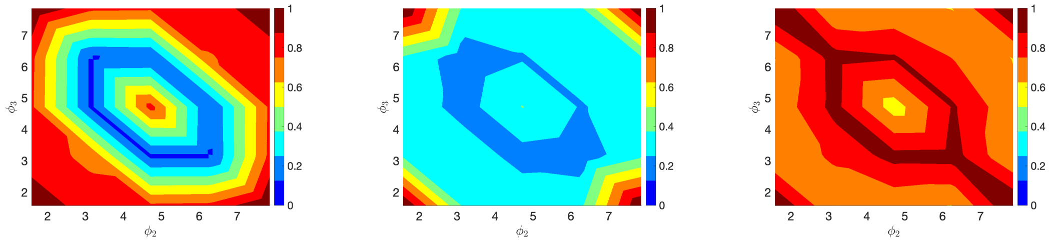
A similar discussion can be made if more than two pendula are considered. We start with the case $N_p = 3$. We first validate the model reduction techniques using the same physical parameters of its corresponding counterpart with $N_p = 2$. The results with initial conditions $(\phi_1, \phi_2, \phi_3) = (\pi/2, 11\pi/6, -\pi/2)$ are reported in Figure 4.6. Although an asymptotic state is not fully reached yet, it appears that R_3 approaches one, whereas all the other order parameters take low values. This is an indication of a symmetric configuration, in which the pendula oscillate with phase shifts of $2\pi/N_p$ [33]. Again, there is virtually no loss in accuracy if the reduced models are used, whereas a computational speedup larger than two orders of magnitude is observed.



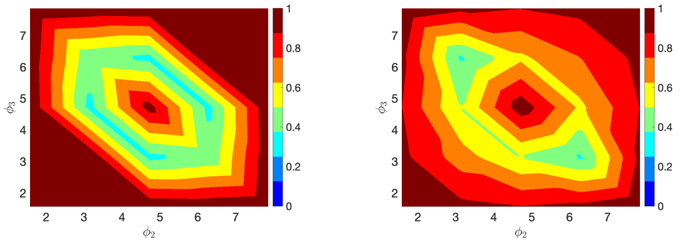
(a) FOM. (b) CMS. (c) POD.

Figure 4.6 – Order parameters vs time for the full and reduced models, $N_p = 3$.

As variations in the mass and length of the pendula do not modify the conclusion we drew for the two-pendulum case, we focus on varying the initial conditions. We consider $\phi_1 = \pi/2$, while ϕ_2 and ϕ_3 vary in the interval $[\pi/2, 5\pi/2]$. The obtained asymptotic order parameters are shown in Figure 4.3 as a function of ϕ_2 and ϕ_3. At the corners of the parameter domain, in-phase synchronization is observed ($R_1 \simeq 1$), as the pendula start in equal or very close conditions. In a symmetric region around the center, a symmetric configuration with phase shifts of $2\pi/3$ is obtained ($R_3 \simeq 1$). Finally, a clustering configuration in which two of the pendula are in-phase and a third one is shifted by an angle of approximately $\pi/2$ is observed for certain initial conditions ($R_4 \simeq 1$).



(a) R_1. (b) R_2. (c) R_3.

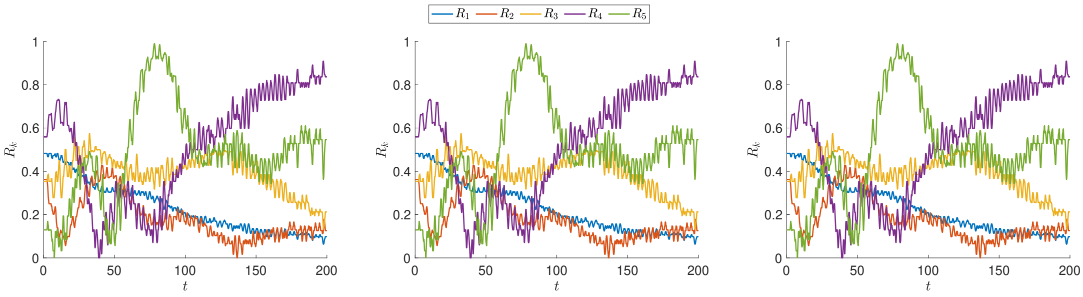


(d) R_4. (e) R_5.

Figure 4.7 – Order parameters under variations of ϕ_2, ϕ_3.

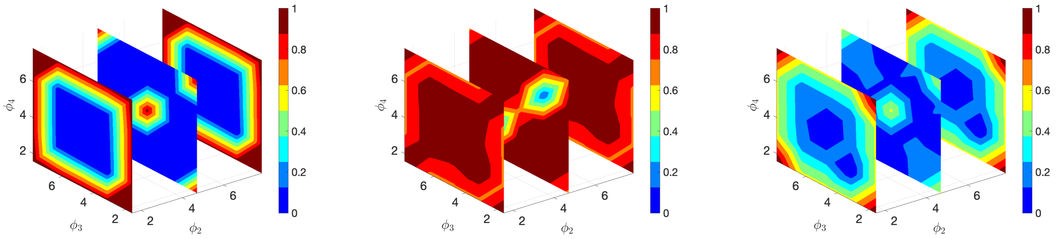
Finally, we consider the case of $N_p = 4$ pendula. To validate the model reduction techniques, we consider initial conditions defined by $(\phi_1, \phi_2, \phi_3, \phi_4) = (\pi/2, 7\pi/6, -\pi/2, \pi)$. The results are reported in Figure 4.8. Again, no differences between the values of R_k obtained with the full and the reduced order models are observed. Synchronization is not reached yet, but a symmetric configuration with phase shifts of $2\pi/N_p$ starts to appear, as R_4 seems to converge to one.

If the initial conditions ϕ_2, ϕ_3 and ϕ_4 are varied in the interval $[\pi/2, 5\pi/2]$, we obtain the asymptotic order parameters shown in Figure 4.9. Again, there exist regions in which the in-phase synchronization ($R_1 \simeq 1$) and symmetric $2\pi/4$ configurations ($R_4 \simeq 1$) are observed. However, more complex clustering configurations are also present. An example includes a 2-2 clustering characterized by two groups of in-phase oscillators which are mutually in anti-phase ($R_2 \simeq R_4 \simeq 1$).

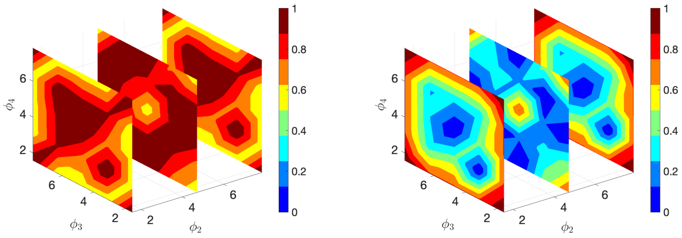


(a) FOM. (b) CMS. (c) POD.

Figure 4.8 – Order parameters vs time for the full and reduced models, $N_p = 4$.



(a) R_1. (b) R_2. (c) R_3.



(d) R_4. (e) R_5.

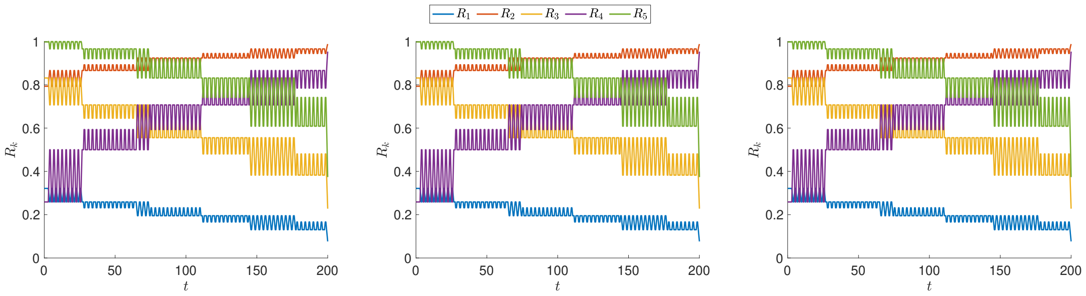
Figure 4.9 – Order parameters under variations of ϕ_2, ϕ_3, ϕ_4.

4.4 Extensions

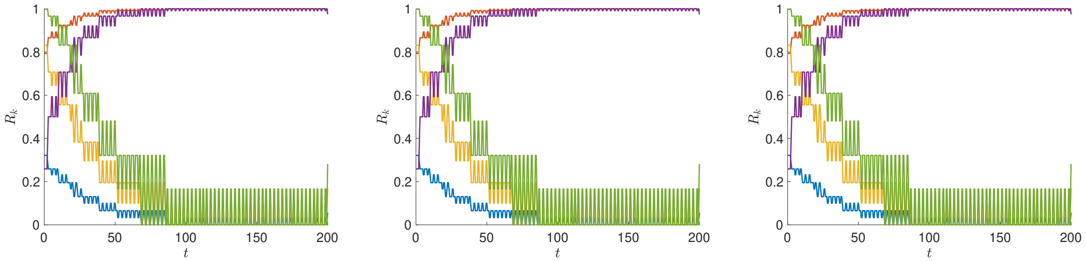
4.4.1 Parameter-dependent structure

Both the presented reduction techniques can naturally handle any parametrization related to the pendula. The question that arises is whether a parametrized structure can also be considered. With the CMS approach, one should solve a different eigenproblem for each instance of the structural parameters and suitably combine the resulting modes. With the POD, one should instead include snapshots generated with different parameters. However, we observed that the bases generated with a single instance of the parameters are robust with respect to parameter changes, and there is no need to re-construct them. This is possibly due to the simple and oscillatory nature of the problem, in which a variation of the parameters does not affect the oscillation modes strongly, consistent with [100]. As an example, in Figure 4.10 we consider a variation in the width b_h of the horizontal beam, whose baseline value is

$b_h = 0.55$, as reported in Table 4.1. The accuracy of the reduced model remains very high, despite the change in the parameter. This allows to potentially study the effect of different structural properties on the synchronization using the reduced model. In this case, an increase in the thickness b_h leads to a stronger coupling and a shorter transient phase.



(a) FOM, $b_h = 0.3$. (b) CMS, $b_h = 0.3$. (c) POD, $b_h = 0.3$.



(d) FOM, $b_h = 0.8$. (e) CMS, $b_h = 0.8$. (f) POD, $b_h = 0.8$.

Figure 4.10 – Order parameters vs time for the full and reduced models for different values of b_h.

4.4.2 Beyond Huygens: parametrized coupling

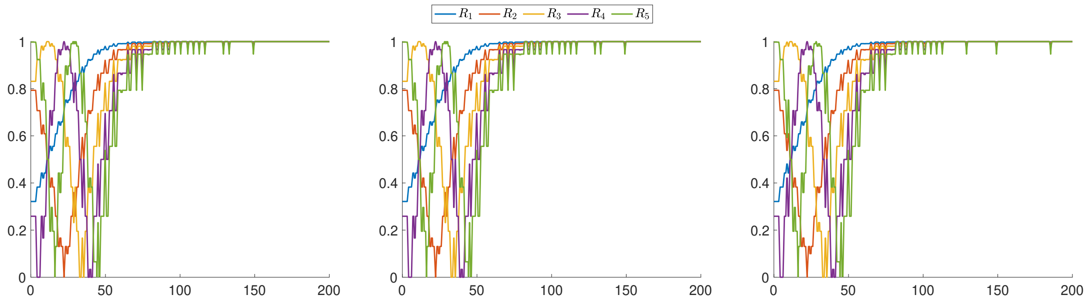
As the reduced models are generated independently of the coupled problem, more general coupling terms can possibly be considered, beyond the standard Huygens' model. This would be necessary, e.g., when one is interested in understanding the possible behaviors of the coupled system in the lack of physical laws to design the interactions. The idea is to consider a variation of (4.10) which retains the same formal structure, but in which all the interaction terms appearing in the right-hand-side are replaced by (artificially) parametrized terms. The modified coupled system reads

$$\mathbf{M}\ddot{q} + \mathbf{B}\dot{q} + \mathbf{K}q = \left[g^s_{x1}, g^s_{y1}, g^s_{x2}, g^s_{y2}, \mathbf{0}^T\right]^T, \tag{4.18a}$$
$$m_i L_i^2 \ddot{\theta}_i = -m_i L_i \sin\theta_i - d_i \dot{\theta}_i + u_i + g^p_{xi} + g^p_{yi},$$

where each of the coupling functions g can be arbitrarily designed. Assuming that they depend on the states and first-order derivatives only, and in order to enforce sparsity in the functions, we consider

$$g^s_{xi} = \sum_{\boldsymbol{p}} \beta_i^{\boldsymbol{p}} x_i^{p_1} \dot{x}_i^{p_2} \theta_i^{p_3} \dot{\theta}_i^{p_4}, \tag{4.19}$$

where $\boldsymbol{p} = (p_1, p_2, p_3, p_4)$ is a multi-index with each entry belonging to $\{0, \dots, P\}$, where P denotes an a-priori fixed maximum polynomial order, and $\beta_i^{\boldsymbol{p}} \in \mathbb{R}^4$ are the expansion coefficients. Expressions similar to (4.19) hold for all the coupling terms in (4.18). Simulating the system for different instances of $\beta_i^{\boldsymbol{p}}$ allows us to predict all the possible patterns that the system can exhibit. As an example, with a specific instance of these parameters one obtains the results shown in Figure 4.11. Although very mild differences errors appear if a POD technique is used, the accuracy of both reduced models is high. With our choice of parameters, an in-phase synchronization is obtained. Excluding peculiar initial conditions, this behavior was never observed in the classical Huygens' setup.



(a) FOM.

(b) CMS.

(c) POD.

Figure 4.11 – Order parameters vs time for the full and reduced models for a specific instance of (4.19).

4.5 Conclusion

In this chapter we constructed reduced order models of a locally coupled oscillatory system that exhibits synchronization. Compared to Chapter 3, we study higher-order synchronization patterns, similar to the Huygens' experiment. We constructed and compared both model-based and data-driven methods, whose offline phase is performed in the absence of the coupled model.

The presented results confirm the efficiency of both these methods, which are able to retrieve the synchronization patterns observed at the full level with a significant computational saving. Variations in constitutive parameters of both the structure and the pendula are considered, which lead to in-phase, anti-phase and clustering synchronization types.

4.A Finite Element discretization

Here, we aim at providing a more detailed description of the finite element model used to discretize the structural model

$$\rho A\frac{\partial^2 u}{\partial t^2}-\frac{\partial}{\partial\xi}\left(EA\frac{\partial u}{\partial\xi}\right)=p_u(\xi), \tag{4.20a}$$

$$\rho A\frac{\partial^2 w}{\partial t^2}+\frac{\partial^2}{\partial\xi^2}\left(EI\frac{\partial^2 w}{\partial\xi^2}\right)=p_w(\xi).$$

The meaning of each term and variable is discussed in Section 4.1. Without loss of generality, here we focus on a single beam $\Omega=[0,L]$ and we assume that the equations are completed with homogeneous boundary conditions for the horizontal displacement u, the vertical displacement w and the rotation $\partial w/\partial\xi$, so that no boundary terms are present in the following integral formulations.

The weak form of (4.20) consists in finding $u(t)\in V=H_0^1(\Omega)$ and $w(t)\in Z=H_0^2(\Omega)$ such that

$$\int_\Omega \rho A\frac{\partial^2 u}{\partial t^2}v+\int_\Omega EA\frac{\partial u}{\partial\xi}\frac{\partial v}{\partial\xi}=\int_\Omega p_u v, \tag{4.21a}$$

$$\int_\Omega \rho A\frac{\partial^2 w}{\partial t^2}z+\int_\Omega EI\frac{\partial^2 w}{\partial\xi^2}\frac{\partial^2 z}{\partial\xi^2}=\int_\Omega p_w z,$$

for all $v\in V$ and $z\in Z$. This can be compactly written as

$$m\left(\left(\frac{\partial^2 u}{\partial t^2},\frac{\partial^2 w}{\partial t^2}\right),(v,z)\right)+a\left((u,w),(v,z)\right)=f\left((v,z)\right),$$

where the definition of $m(\cdot,\cdot)$, $a(\cdot,\cdot)$ and $f(\cdot)$ can be directly inferred from (4.21). Introducing suitable finite-dimensional spaces $V_h\subset V$ and $Z_h\subset Z$ that depend on a positive parameter h, the Galerkin finite element formulation consists in finding $u_h(t)\in V_h$ and $w_h(t)\in Z_h$ such that

$$m\left(\left(\frac{\partial^2 u_h}{\partial t^2},\frac{\partial^2 w_h}{\partial t^2}\right),(v_h,z_h)\right)+a\left((u_h,w_h),(v_h,z_h)\right)=f\left((v_h,z_h)\right), \tag{4.22}$$

for all $v_h\in V_h$ and $z_h\in Z_h$. Particular care has to be given to the choice of the spaces V_h and Z_h, to ensure the conformity property and, ultimately, the convergence of the discrete solution to the continuous one [105]. We first introduce a partition $\mathcal{T}_h$ of $[0,L]$ in N intervals $K_j=[\xi_{j-1},\xi_j]$ with width h_j such that $0=\xi_0<\cdots<\xi_N=L$, and we set $h=\max_j h_j$. The simplest choice for V_h consists in

$$V_h=\left\{v_h\in C^0(\overline{\Omega})\ :\ v_h\big|_{K_j}\in\mathbb{P}^1\ \ \forall K_j\in\mathcal{T}_h\right\}\cap V,$$

whereas for Z_h we adopt

$$Z_h=\left\{z_h\in C^1(\overline{\Omega})\ :\ z_h\big|_{K_j}\in\mathbb{P}^3\ \ \forall K_j\in\mathcal{T}_h\right\}\cap Z.$$

Note that the dimensions of V_h and Z_h are $N+1$ and $2(N+1)$, respectively. As in classical finite elements, as basis functions for V_h we select the linear Lagrange polynomials ℓ_j. As they satisfy $\ell_j(\xi_i) = \delta_{ij}$, this implies that the degrees of freedom are the nodal values of u_h, i.e.,

$$u_h(\xi, t) = \sum_{j=1}^{N+1} u_h(\xi_j, t)\ell_j(\xi).$$

Conversely, for Z_h we select the cubic Hermite polynomials ϕ_j. As they satisfy $\phi_j(\xi_i) = \delta_{ij}$ and $\frac{d\phi_j}{d\xi}(\xi_i) = \delta_{(i+N+1),j}$ for $i \in \{1,\ldots,N+1\}$ and $j \in \{1,\ldots,2(N+1)\}$, this implies that the degrees of freedom are the nodal values of w_h and its first derivative $\partial w_h/\partial\xi$, i.e.,

$$w_h(\xi) = \sum_{j=1}^{N+1} w_h(\xi_j, t)\phi_j(\xi) + \sum_{j=1}^{N+1} \frac{\partial w_h}{\partial \xi}(\xi_j, t)\phi_{j+N+1}(\xi).$$

Evaluating (4.22) for each basis function separately allows us to write the algebraic formulation as

$$\begin{aligned} \mathbf{M}_u\ddot{\mathbf{u}} + \mathbf{K}_u\mathbf{u} &= \mathbf{f}_u \\ \mathbf{M}_w\ddot{\mathbf{w}} + \mathbf{K}_w\mathbf{w} &= \mathbf{f}_w \end{aligned}$$

where $\mathbf{u} \in \mathbb{R}^{N+1}$ and $\mathbf{w} \in \mathbb{R}^{2(N+1)}$ are vectors gathering the degrees of freedom associated to $u_h(t)$ and $w_h(t)$, the *dot* symbol denotes the time derivative, $\mathbf{M}_u$ and $\mathbf{M}_w$ are the mass matrices, $\mathbf{K}_u$ is the stiffness matrix associated to the first derivatives, $\mathbf{K}_w$ is the stiffness matrix associated to the second derivatives, and $\mathbf{f}_u$ and $\mathbf{f}_w$ are the source terms. In a compact form, we can finally write

$$\begin{bmatrix} \mathbf{M}_u & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_w \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}} \\ \ddot{\mathbf{w}} \end{bmatrix} + \begin{bmatrix} \mathbf{K}_u & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_w \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_u \\ \mathbf{f}_w \end{bmatrix},$$

which corresponds to the undamped version of (4.2), up to an appropriate reordering of the degrees of freedom.

For the sake of completeness, we report the explicit expression of the selected basis functions, both in the reference element $[0,1]$,

$$\begin{gathered} \ell_1(\xi) = (1-\xi), \quad \ell_2(\xi) = \xi, \\ \phi_1(\xi) = (\xi-1)^2(2\xi+1), \quad \phi_2(\xi) = \xi^2(-2\xi+3) = \xi^2(1+2(1-\xi)), \\ \phi_3(\xi) = \xi(\xi-1)^2, \quad \phi_4(\xi) = \xi^2(\xi-1), \end{gathered}$$

for $\xi \in [0,1]$, and in a generic interval $[a,b]$,

$$\begin{gathered} \ell_1(\xi) = \frac{b-\xi}{b-a}, \quad \ell_2(\xi) = \frac{\xi-a}{b-a} \\ \phi_1(\xi) = \left(\frac{b-\xi}{b-a}\right)^2 \left(2\frac{\xi-a}{b-a}+1\right), \quad \phi_2(\xi) = \left(\frac{\xi-a}{b-a}\right)^2 \left(1+2\frac{b-\xi}{b-a}\right), \end{gathered}$$

$$\phi_3(\xi) = \frac{\xi - a}{b - a}\left(\frac{b - \xi}{b - a}\right)^2, \quad \phi_4(\xi) = \left(\frac{\xi - a}{b - a}\right)^2\left(\frac{\xi - b}{b - a}\right),$$

for $\xi \in [a, b]$. Setting $b - a = h$ and assuming unitary physical parameters, the corresponding local mass and stiffness matrices read

$$\mathbf{M}_u = \frac{h}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{K}_u = \frac{1}{h}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix},$$

$$\mathbf{M}_w = \frac{h}{420}\begin{bmatrix} 156 & 54 & 22h & -13h \\ 54 & 156 & 13h & -22h \\ 22h & 13h & 4h^2 & -3h^2 \\ -13h & -22h & -3h^2 & 4h^2 \end{bmatrix}, \quad \mathbf{K}_w = \frac{1}{h^3}\begin{bmatrix} 12 & -12 & 6h & 6h \\ -12 & 12 & -6h & -6h \\ 6h & -6h & 4h^2 & 2h^2 \\ 6h & -6h & 2h^2 & 4h^2 \end{bmatrix}.$$

5 Localized model reduction and domain decomposition methods

In this chapter[1] we consider coupled systems modeled through partial differential equations. Similar to Chapter 4, the focus is still on localized reduced models, meaning that we wish to construct approximations of local problems without simulating the fully coupled problem.

As a global, monolithic solver is often not available, a first task is to design efficient methods to find the solution of a given coupled system of interest using local models only. Given the component-wise structure of the system, domain decomposition, partitioned-like methods have proven to be effective in this direction [104]. Here, the idea is to solve the local problems independently through iterated exchanges of boundary data.

However, the dimensionality of the problem might still be large, and reduced order models need to be designed. As discussed in Chapter 1 and Chapter 4, the a-priori unavailability of the coupled problem prevents one from directly using classical model reduction methods. This is also the case when the model is available, but its computational complexity is very large. This includes scenarios in which the complexity of the underlying differential model is high, the number of components is large, or when the parameter space is high-dimensional. Additional complexities arise if the components are modeled using different simulation tools, as they can be difficult to integrate. On the lines of Chapter 4, we propose a model reduction technique that relies on local surrogates, in the form of reduced models constructed in a component-wise, decoupled fashion. In the offline phase, the components are considered separately, and the snapshots are collected through an artificial parametrization of the interface data. These are subsequently used to construct a local reduced basis. During the online phase, the reduced models are combined using a domain decomposition approach.

Localized model reduction of multi-component systems of partial differential equations has been investigated in a number of works [17]. A first approach related to the one we propose in this work, is the Reduced basis, Domain decomposition, Finite elements (RDF) method [75]. The authors construct local surrogates using parametrized Lagrange or Fourier basis functions and a greedy algorithm. This method shares many features with ours, but the use of

---

[1] Large parts of this chapter are based on our research work [43].

the full model in a few regions of the computational domain and the application to a single steady-state diffusion problem limit its generalization potential.

Similar methods rely on Lagrange multipliers [51, 89, 88, 24, 23] to impose the coupling conditions. Although they can be easily constructed and efficient solvers can be designed, the introduction of additional variables can affect the computational performance of the method and might require modifications of the solvers to accommodate this technique. The local bases can be computed from solutions of the local problems defined on each subdomain and small parts of the contiguous ones with fixed Dirichlet conditions [89, 87]. This might require further modifications to the local solvers, computational domains and meshes, that might be challenging in general geometric settings or even impossible if the numerical implementation does not allow it. These methods have been successfully applied in structural mechanics, fluid dynamics, and, more recently, in haemodynamics [98].

Similarly, static condensation methods [48, 73, 74] aim to solve the coupled problem through an interface equation obtained by a splitting of internal and boundary degrees of freedom and a Schur complement method. Then, separate reduced bases are constructed to capture the internal and the interface features, respectively. Snapshots obtained from simulation of the local problems and interface eigenvalue problems are typically used for this purpose [73, 74]. Although they are not well suited for nonlinear problems, they have been proven extremely effective in large-scale structural dynamics problems. In spirit, our approach relies on similar ideas to construct the artificial parametrization of the interface conditions. However, we construct the reduced basis without an a-priori separation between internal and boundary degrees of freedom or the assembly of a Schur complement. Thus, our method requires a minimal set of modifications in both the offline and online phase. Moreover, we focus mostly on time-dependent heterogeneous problems, we rely on partitioned schemes and we support mixed interface conditions.

Alternative approaches rely on randomization and oversampling [18, 119, 40]. Similar to [87], the local problems are solved in a domain slightly larger than its physical one with random boundary conditions, and their solutions, once restricted to the physical domain, are used to construct the local bases. Exponential convergence in the reduced dimension can be proven for linear problems and the low-rank properties of the problem are strengthened, but the extension to complex nonlinear problems, possibly characterized by a high degree of heterogeneity among the components, can be challenging. Moreover, the training procedure requires once more a modification of the local computational domain. However, they share with our method the notion of parametrized boundary conditions.

A final approach, based on domain truncation, constructs a local reduced model for a combustion problem [140]. This method is once again related to the one presented here, and notably analyses a complex time-dependent problem. However, the artificial parametrization is based on a theoretical knowledge of the problem of interest, which potentially prevents the method to generalize other classes of multi-physics problems. Moreover, the final problem is not fully reduced, as the surrogate model is constructed for a single subproblem only.

The remainder of this chapter is structured as follows. In Section 5.1 we present the coupled system of interest and an overview of domain decomposition methods. Section 5.2 describes our method, highlighting its steps in the offline and the online phase, an a-priori error estimate, and a discussion on the computational cost. Section 5.3 presents several numerical results, showing the potential of the proposed approach in terms of accuracy, computational efficiency and robustness. A few concluding remarks are found in Section 5.4.

5.1 Problem formulation

In the following, we introduce the mathematical formulation of the problems of interest. We consider multi-component systems of partial differential equations defined on different non-overlapping subdomains, so that the local problems correspond to the components of the system. To simplify the notation, we focus on the case of two time-dependent problems, denoted by $\mathscr{P}_1$ and $\mathscr{P}_2$, defined on as many subdomains. We consider a time interval $[0,T]$ and a spatial domain $\Omega \subseteq \mathbb{R}^d$ such that $\Omega = \Omega_1 \cup \Omega_2$, where Ω_i is the domain on which $\mathscr{P}_i$ is defined. The interface between the two, where the coupling conditions are enforced, is denoted by $\Gamma := \bar{\Omega}_1 \cap \bar{\Omega}_2$.

Thus, the coupled system can be written as

$$\begin{aligned} \frac{\partial u_1}{\partial t} + \mathscr{F}_1(u_1) &= 0, \quad \text{in } \Omega_1, \\ u_1(t=0) &= u_{1,0}, \quad \text{in } \Omega_1, \\ h_1(u_1) &= 0, \quad \text{on } \partial\Omega_1 \setminus \Gamma, \\ f_1(u_1) &= g_1(u_2), \quad \text{on } \Gamma, \end{aligned} \tag{5.1a}$$

$$\begin{aligned} \frac{\partial u_2}{\partial t} + \mathscr{F}_2(u_2) &= 0, \quad \text{in } \Omega_2, \\ u_2(t=0) &= u_{2,0}, \quad \text{in } \Omega_2, \\ h_2(u_2) &= 0, \quad \text{on } \partial\Omega_2 \setminus \Gamma, \\ f_2(u_2) &= g_2(u_1), \quad \text{on } \Gamma, \end{aligned} \tag{5.1b}$$

where $\mathscr{F}_i$ is a generic second order differential operator modeling the dynamics of problem $\mathscr{P}_i$. The initial conditions in each subdomain are denoted by $u_{i,0}$. The boundary conditions on the physical domain boundaries are encoded by the operators $h_i(u_i)$, whereas the interactions between the components are modeled through suitable functions f_i and g_i.

Typically, the coupling functions are designed to enforce constraints on physical quantities among the components, such as continuity of the solutions and the normal fluxes. Instead of enforcing them separately as in more classical formulations, one can consider a linear combination of these constraints [104]. Among other advantages, this addresses concerns of well-posedness of the local problems and allows to design faster solution algorithms [56]. Denoting the fluxes and the unit normal vectors as $\boldsymbol{\sigma}_i = \boldsymbol{\sigma}_i(u_i)$ and $\boldsymbol{n}_i$, respectively, we consider coupling conditions of the form

$$\boldsymbol{\sigma}_1(u_1)\cdot\boldsymbol{n}_1 + \lambda_1 u_1 = \boldsymbol{\sigma}_2(u_2)\cdot\boldsymbol{n}_1 + \lambda_1 u_2, \quad \boldsymbol{\sigma}_2(u_2)\cdot\boldsymbol{n}_2 + \lambda_2 u_2 = \boldsymbol{\sigma}_1(u_1)\cdot\boldsymbol{n}_2 + \lambda_2 u_1, \tag{5.2}$$

where $\lambda_i \in \mathbb{R}_+$ are tunable parameters.

Remark 5.1 *More sophisticated coupling conditions can be considered, at least when simulating the coupled problem. Examples include adding tangential derivatives [56] or time derivatives [59] in* (5.2)*. However,* (5.2) *is a sufficiently general functional form to model the interactions in coupled systems and it is commonly used in practical applications. Therefore, in this work we employ this functional representation in both the offline and the online phase, as shown in Section 5.2. We conjecture that the reduced models are robust enough such that other boundary conditions can be employed in the online phase without re-running the offline phase entirely.*
Additionally, we mention that higher order problems require more general coupling conditions, so that a careful definition of (5.2) *is needed. An interesting example is the biharmonic operator, for which several options can be considered [58]. However, we believe that the method proposed in Section 5.2 can be extended to such cases while keeping the same spirit.*

As our goal is to efficiently solve (5.1) promoting component independence, we simulate the problem using a non-overlapping Schwarz method. This allows us to decouple the local problems, for which surrogate models can be constructed. We consider the implicit Euler scheme with time step Δt and $N_{steps} = T/\Delta t$ steps, and we denote with u_i^n the numerical solution of $\mathscr{P}_i$ at time $n\Delta t$. The idea is to use fixed-point iterations, solving the local problems $\mathscr{P}_i$ with educated guesses on the solutions of $\mathscr{P}_j$, $j \neq i$, which are in turn updated until a suitable convergence criterion is met. Specifically, at each iteration of the temporal loop, we proceed as follows:

1. Take a guess on u_i^{n+1}, e.g., $u_i^{n+1,0} = u_i^n$ and set $s = 0$.
2. Solve
$$u_1^{n+1,s+1} - u_1^n + \Delta t \mathscr{F}_1\left(u_1^{n+1,s+1}\right) = 0 \tag{5.3}$$
with boundary conditions $h_1\left(u_1^{n+1,s+1}\right) = 0$ and
$$f_1\left(u_1^{n+1,s+1}\right) = g_1\left(u_2^{n+1,s}\right). \tag{5.4}$$
3. Solve
$$u_2^{n+1,s+1} - u_2^n + \Delta t \mathscr{F}_2\left(u_2^{n+1,s+1}\right) = 0 \tag{5.5}$$

with boundary conditions $h_2\left(u_2^{n+1,s+1}\right) = 0$ and

$$f_2\left(u_2^{n+1,s+1}\right) = g_2\left(u_1^{n+1,s+1}\right). \tag{5.6}$$

4. Compute error, e.g.,

$$\epsilon^{s+1} = \frac{\left(\left\|u_1^{n+1,s+1} - u_1^{n+1,s}\right\|^2_{L^2(\Omega_1)} + \left\|u_2^{n+1,s+1} - u_2^{n+1,s}\right\|^2_{L^2(\Omega_2)}\right)^{1/2}}{\left(\left\|u_1^{n+1,s}\right\|^2_{L^2(\Omega_1)} + \left\|u_2^{n+1,s}\right\|^2_{L^2(\Omega_2)}\right)^{1/2}}. \tag{5.7}$$

5. Repeat Steps 2, 3, 4 until convergence, e.g., $\epsilon^{s+1} < \text{tol}$.
6. Set $u_i^{n+1}=u_i^{n+1,s+1}$.

Convergence of the fixed point algorithm has been proven in a number of cases, including steady [86, 46, 104] and unsteady [59, 7, 57] problems. We remark that, to ensure and accelerate convergence, a relaxation step might be needed [104]. As it is found not to be necessary for the problems studied in this chapter, and more generally when Robin-type conditions (5.2) are used, we decided not to explicitly include it in the algorithm. We also note that in Step 3 the interface condition (5.6) is based on the updated solution of $\mathscr{P}_1$, so that the proposed algorithm can be interpreted as a block Gauss-Seidel method. Alternatively, one could consider its Jacobi counterpart, that relies on $u_1^{n+1,s}$ instead. A Jacobi implementation is easier to implement and is embarrassingly parallel, but the convergence is slower. As our current implementation is purely serial, we prefer the Gauss-Seidel strategy. A detailed discussion on the two approaches can be found in, e.g., [22].

Extending the presented formulation to more than two subdomains is rather straightforward. Denoting by $\mathscr{I}_i$ the index set of the problems $\mathscr{P}_j$ that share an interface Γ_{ij} with $\mathscr{P}_i$, the coupling conditions of each $\mathscr{P}_i$ can be written as

$$\boldsymbol{\sigma}_i(u_i)\cdot\boldsymbol{n}_i + \lambda_{ij}u_i = \boldsymbol{\sigma}_j(u_j)\cdot\boldsymbol{n}_i + \lambda_{ij}u_j,$$

for each $j \in \mathscr{I}_i$. The non-overlapping Schwarz method can be generalized in a similar way. A Jacobi implementation can be directly inferred from the two-subdomain case, whereas in the Gauss-Seidel case one needs to take particular care of the ordering in which the elements are processed. This problem is strictly related to the generation of a directed acyclic graph and graph coloring [22].

5.2 Localized model reduction

We now adapt the general technique described in Chapter 2 to our problem of interest (5.1), recalling that our main assumption is that the fully coupled model is not available when constructing the reduced model.

5.2.1 Offline phase

The main objective of the offline phase is the generation of the basis functions. As the coupled problem (5.1) is not available, we consider the components separately and we build a basis for each of them. Specifically, each problem $\mathscr{P}_i$ in (5.1) has the form

$$\begin{aligned} \frac{\partial u}{\partial t} + \mathscr{F}(u) &= 0, \quad \text{in } \Omega, \\ u(t=0) &= u_0, \quad \text{in } \Omega, \\ h(u) &= 0, \quad \text{on } \partial\Omega \setminus \Gamma, \\ f(u) = \boldsymbol{\sigma} \cdot \boldsymbol{n} + \lambda u &= g, \quad \text{on } \Gamma, \end{aligned} \tag{5.8a}$$

where the subscripts i have been omitted for simplicity and g is a generic function of the form

$$g = \sum_{j=1}^{N_{exp}} \beta_j \varphi_j. \tag{5.9}$$

In (5.9), φ_j are functions used to expand the boundary datum, N_{exp} is their number, and β_j are the expansion coefficients. The idea is to collect snapshots of (5.8) from simulations with different parameter values. We have two classes of parameters:

- Physical parameters and time, analogous to standard model reduction approaches. Samples can be collected using, e.g., quasi-random sampling.
- The artificial parameters defining the boundary condition (5.8a) and (5.9). Samples are collected using the following method:
  - The Robin parameter λ is fixed a-priori. The only condition on its value is the well posedness of (5.8), which is typically ensured if $\lambda > 0$.
  - The functions φ_j are also fixed. Inspired by similar works [2, 73], we choose them as the smallest eigenfunctions of the Laplace-Beltrami operator on the interface Γ, i.e.,

    $$-\Delta\varphi_j = \omega_j \varphi_j, \quad \text{on } \Gamma, \tag{5.10}$$

    with suitable boundary conditions. In order not to be restricted to a specific set of interface profiles, to increase the level of generality of the basis, and to have a larger insight on the local dynamics, we consider both homogeneous Dirichlet and homogeneous Neumann conditions. Moreover, we additionally found that including the zero function in (5.9) is beneficial for the reduced model, as it contributes to capture the dynamics of the completely decoupled problem. Thus, we select $N_{exp} = 2N_{basis} = N_{Dir} + N_{Neu} + 1$, where $N_{basis} = N_{Dir} + 1 = N_{Neu}$ is the number of the smallest eigenvalues to be kept. If the interface is one-dimensional with length L, this reduces to

$$\varphi_j \in \left\{1, \ldots, \cos\left(\frac{(N_{basis}-1)\pi}{L}\xi\right), 0, \ldots, \sin\left(\frac{(N_{basis}-1)\pi}{L}\xi\right)\right\},$$

where ξ is a curvilinear coordinate defined on the interface.

- The coefficients used to generate the snapshots can vary. Assuming that we wish to simulate the problem (5.8) N_{sampl} times with coefficients β_{ij}, $i \in \{1,\ldots,N_{\text{sampl}}\}$, we choose $N_{\text{sampl}} = N_{exp}$ and

  $$\beta_{ij} = A\delta_{ij}. \tag{5.11}$$

  This implies that (5.8) is simulated with each eigenfunction as a boundary condition, including the zero function. The choice of the scaling factor A is arbitrary. We only require A to scale as 1 (resp. λ) in the limits of small (resp. large) λ, corresponding to Dirichlet and Neumann conditions. Thus, we consider $A = \max\{1,\lambda\}$.

The main steps of the offline phase follow, i.e., for each instance of the physical parameters, we simulate (5.8) using the parametrized Robin datum, collect the solution snapshots, and construct the reduced model as in Chapter 2. These are summarized as in Algorithm 5.1. The extension to vector-valued differential equations is rather straightforward, and can be done by parametrizing the boundary condition of each variable independently. Similarly, the extension to coupled problems with more than two subdomains can be easily derived. As $\mathscr{P}_i$ can have multiple interfaces Γ_{ij} and neighboring subdomains $j \in \mathscr{I}_i$, we repeat the proposed procedure for each Γ_{ij}, assigning homogeneous coupling conditions on the remaining $\Gamma_{i\tilde{j}}$, $\tilde{j} \neq j$.

Algorithm 5.1 Offline phase

```
Input: Local problems
Output: Local bases $U_i$ and local reduced operators
  for all system components $i$ do
      for all instance of the physical parameter vector do
          for all boundary data (5.9) using (5.10) and (5.11) do
              Simulate the problem and collect the snapshots.
          end for
      end for
      Construct the reduced basis $U_i$ for the given component.
      Compute the reduced operators.
  end for
```

Remark 5.2 *We choose to treat the physical and the artificial parameters in an independent, separate way. Although this can lead to a large offline computational cost, this is not prohibitive, especially in the case of a parallel implementation. Other choices can be considered, with the goal of jointly sampling their values and reducing the associated offline cost. An example can be found in [75], where a greedy algorithm is used.*

Remark 5.3 *We stress that the choice* (5.11) *implies that the local problems are simulated using the eigenfunctions φ_j individually with constant-in-time coefficients. Different choices can be considered, with the goal of improving the sampling of the boundary data* (5.9)*. An option could be a random sampling of the coefficients β_{ij}, possibly combined with a suitable modal decay rate. More challenging would be the introduction of a time-dependence, although theoretically possible. In the absence of information on the coupled problem, an appropriate selection of the time interval and/or a basis expansion in the time domain can be difficult. Both these options would increase the offline cost, introduce additional hyperparameters, and impact on the reduction potential, and are found not to be necessary to achieve a satisfactory level of accuracy for our problems of interest. We note that, given the non-intrusive nature of the approach presented in Chapter 6, a more thorough sampling, on the lines that we have just discussed, is necessary.*

Remark 5.4 *Effectively, the proposed method introduces a number of additional hyperparameters. The first is the number of interface basis functions N_{basis}. In practical applications, this can be decided a-priori by fixing the number of frequencies that should be retained in the representation of the boundary data. The second is the Robin parameter λ. If available, its value is determined by the coupled problem. Otherwise, at least in the offline phase it can be chosen arbitrarily, as long as the local problem* (5.8) *is well-posed. As shown in Section 5.3, we numerically observed that its choice does not strongly affect our conclusions, as long as well posedness of the local problems and convergence of the fixed point iterations are guaranteed. Moreover, if these conditions are ensured, there is nothing that prevents one to choose $\lambda \to 0$ and $\lambda \to \infty$. Other hyperparameters include the number of samples of the boundary condition N_{sampl} (see Remark 5.3) and the scaling factor A in* (5.11).

5.2.2 Online phase

Once the local reduced models have been constructed, we combine them to approximate the solution of a given problem of interest. Mimicking the algorithm presented in Section 5.1, the general formulation of the domain-decomposition-based reduced order model reads as follows:

1. Take a guess on α_i^{n+1}, e.g., $\alpha_i^{n+1,0} = \alpha_i^n$ and set $s = 0$.

2. Solve
$$\alpha_1^{n+1,s+1} - \alpha_1^n + \Delta t U_1^T \mathscr{F}_1 \left(U_1 \alpha_1^{n+1,s+1} \right) = 0 \tag{5.12}$$
with boundary conditions $h_1 \left(U_1 \alpha_1^{n+1,s+1} \right) = 0$ and
$$f_1 \left(U_1 \alpha_1^{n+1,s+1} \right) = g_1 \left(U_2 \alpha_2^{n+1,s} \right). \tag{5.13}$$

3. Solve
$$\alpha_2^{n+1,s+1} - \alpha_2^n + \Delta t U_2^T \mathscr{F}_2\left(U_2\alpha_2^{n+1,s+1}\right) = 0 \tag{5.14}$$
with boundary conditions $h_2\left(U_2\alpha_2^{n+1,s+1}\right) = 0$ and
$$f_2\left(U_2\alpha_2^{n+1,s+1}\right) = g_2\left(U_1\alpha_1^{n+1,s+1}\right). \tag{5.15}$$
4. Compute error, e.g.,
$$\epsilon^{s+1} = \frac{\left(\left\|U_1\alpha_1^{n+1,s+1} - U_1\alpha_1^{n+1,s}\right\|^2_{L^2(\Omega_1)} + \left\|U_2\alpha_2^{n+1,s+1} - U_2\alpha_2^{n+1,s}\right\|^2_{L^2(\Omega_2)}\right)^{1/2}}{\left(\left\|U_1\alpha_1^{n+1,s}\right\|^2_{L^2(\Omega_1)} + \left\|U_2\alpha_2^{n+1,s}\right\|^2_{L^2(\Omega_2)}\right)^{1/2}}. \tag{5.16}$$
5. Repeat Steps 2, 3, 4 until convergence, e.g., $\epsilon^{s+1} < \text{tol}$.
6. Set $\alpha_i^{n+1}=\alpha_i^{n+1,s+1}$.

A few notes are in order. Firstly, enforcing the interface boundary conditions requires reconstructing a functional of the solution at the boundary nodes only. In the case of nonconforming meshes, this is followed by a projection/interpolation step. The reduced problem is then solved by a projection of the interface condition onto the reduced space. If the relevant operators are linear and can be accessed in the offline phase, their reduced counterparts can be precomputed, and the online phase does not require the aforementioned reconstruction step. Secondly, the norms required in Step 4 can be computed in practice without reconstructing the solution. The online phase is summarized in Algorithm 5.2. Note that, if the nonlinearities are treated appropriately, the computational cost of online phase becomes in principle independent of the full dimension, consistent with the discussion in Chapter 2.

Algorithm 5.2 Online phase

```
Input: Local reduced problems
Output: Approximated solutions $u_i$
  for all instance of the physical parameter vector do
      Simulate the reduced coupled problem to find $\alpha_i$
      Reconstruct the solution $u_i$ when required
  end for
```

5.2.3 Error analysis

We provide a simple but effective a-priori error analysis, identifying the different contributions to the total numerical error. It follows and generalizes the estimate derived for the RDF method [75]. Focusing again on the two-subdomain case, we denote with $u = u_1\mathbb{1}_{\Omega_1} + u_2\mathbb{1}_{\Omega_2}$ the exact solution and with $u_{ROM} = u_{1,ROM}\mathbb{1}_{\Omega_1} + u_{2,ROM}\mathbb{1}_{\Omega_2}$ the reduced solution, i.e., the solution of the reduced coupled problem. A similar notation is used for the other variables. Here, $\mathbb{1}_{\Omega_i}$ denotes the characteristic function of Ω_i. Focusing on the space-time L^2-norm for simplicity, by the triangular inequality, we have

$$\|u - u_{ROM}\| \leq \|u - u_h\| + \|u_h - \hat{u}\| + \|\hat{u} - u_{ROM}\|, \tag{5.17}$$

where u_h is the discrete solution and $\hat{u}$ is the discrete solution whose interface boundary conditions can be written as a linear combination of the selected Laplace-Beltrami eigenfunctions, i.e., are of the form (5.9). The first term in (5.17) behaves as

$$\|u - u_h\| \leq C_1 h^p,$$

where h is a measure of the mesh size of the coupled problem. The convergence rate p can be derived using standard error estimates [103, 39] and its value depends on the regularity of the problem, the selected discretization method, and the accuracy in computing the Robin data when enforcing the interface conditions. The second term in (5.17) is related to the approximation of a generic function using selected Laplace-Beltrami eigenfunctions. Splitting the error on the different subdomains and assuming that standard stability and trace estimates hold for each subproblem [75, 39], we have

$$\|u_h - \hat{u}\| = \left(\sum_i \|u_{i,h} - \hat{u}_i\|^2_{\Omega_i}\right)^{1/2} \leq C\left(\sum_i \|g_i - \hat{g}_i\|^2_{\Gamma_i}\right)^{1/2},$$

where $\hat{g}_i$ is the projection of g_i using N_{exp} functions as in (5.9). This projection error typically behaves [3, 20] as

$$C_2 N_{basis}^{-r}.$$

The rate r depends on the regularity of the problem, its spatial dimension and the compatibility of the selected Laplace-Beltrami eigenfunctions with the boundary conditions at the physical boundaries. An exponential decay

$$C_3 \exp(-C_4 N_{basis})$$

can appear, e.g., when the problem is sufficiently regular and a boundary compatibility of the high-order derivatives is satisfied [3, 20]. The third term in (5.17) is the reduction error and highly depends on the complexity of the local problems. This can be quantified with the Kolmogorov k-width as discussed in Chapter 2. In many cases, including diffusion equations [16], it behaves as

$$C_5(N_{basis}) \exp(-C_6(N_{basis})k),$$

where k is the reduced dimension as in Chapter 2, at least assuming that the parameter space is appropriately sampled. For transport-dominated problems, polynomial rates appear instead [94].

Hence, the accuracy is mainly affected by the mesh size h, the number of boundary basis functions N_{basis} and the reduced dimension k. The constants $C_{2,\dots,6}$ can additionally depend on the hyperparameters defining the boundary conditions, including λ, A, N_{sampl}. However, as we assume these to be fixed, we have not explicitly included them in the analysis.

5.2.4 Computational cost

We conclude this section by providing an estimate of the computational cost of our method, considering again Robin boundary conditions and two subdomains only. For simplicity, we focus on the linear case, so that no hyper-reduction is required. We assume that $\mathscr{P}_i$ has size N_i and that the meshes are conforming at the interface, which in turn has N_Γ degrees of freedom. We denote by $\mathscr{C}$ the cost of each step of the algorithm.

Simulations of the coupled full order problem require, for each parameter sample, time step, and iteration of the domain decomposition algorithm:

- Computing the Robin boundary condition for $\mathscr{P}_1$, i.e., assembling the right-hand-side of (5.4). $\mathscr{C} = \mathscr{O}(N_\Gamma)$.
- Solving $\mathscr{P}_1$ with a given Robin datum, i.e., computing the solution of (5.3) with the boundary condition computed above. $\mathscr{C} = \mathscr{O}(N_1^{\gamma_1})$.
- Computing the Robin boundary condition for $\mathscr{P}_2$, i.e., assembling the right-hand-side of (5.6). $\mathscr{C} = \mathscr{O}(N_\Gamma)$.
- Solving $\mathscr{P}_2$ with a given Robin datum, i.e., computing the solution of (5.5) with with the boundary condition computed above. $\mathscr{C} = \mathscr{O}(N_2^{\gamma_2})$.
- Computing the error (5.7). $\mathscr{C} = \mathscr{O}(N_1^{\kappa_1}) + \mathscr{O}(N_2^{\kappa_2})$.

The values of γ_i and $\kappa_i \leq \gamma_i$ depend on the local underlying problems and discretization algorithms. Typical values, at least for classical finite element discretizations, are $\gamma_i = 2$ and $\kappa_i = 1$.

At the reduced level, both the offline and the online phases contribute to the computational cost, although the former has to be done only once. The offline phase requires, for each system component, parameter sample, instance of the interface boundary datum, and time step:

- Solving $\mathscr{P}_i$ with a given Robin datum, i.e., computing the solution of (5.3) or (5.5) with (5.8a) as interface condition for a given value of the coefficients β_j in (5.9). $\mathscr{C} = \mathscr{O}(N_i^{\gamma_i})$.

After all the snapshots of each component have been collected, the reduced bases have to be computed, as well as the linear reduced operators. This has a cost $\mathscr{C} = \mathscr{O}(N_i^2 k_i + N_i k_i^2)$. We assumed here that the Laplace eigenfunctions can be computed analytically. If that is not the case, an additional computational cost of $\mathscr{C} = \mathscr{O}(N_{basis}^2 N_\Gamma)$ should be added for each system component. The online phase requires, for each parameter sample, time step, and iteration of the domain decomposition algorithm:

- Computing the Robin boundary condition for $\mathscr{P}_1$, i.e., assembling the right-hand-side of (5.13), and project on basis U_1. $\mathscr{C} = \mathscr{O}(N_\Gamma + k_1 N_\Gamma + k_2 N_\Gamma)$. Similar to Section 5.2.2, if the boundary operators can be precomputed, the reconstruction step is not required and the computational cost is $\mathscr{C} = \mathscr{O}(k_1 k_2)$.
- Solving $\mathscr{P}_1$ with a given projected Robin datum, i.e., computing the solution of (5.12) with the projected boundary condition computed above. $\mathscr{C} = \mathscr{O}(k_1^{\delta_1})$.
- Computing the Robin boundary condition for $\mathscr{P}_2$, i.e., assembling the right-hand-side of (5.15), and project on basis U_2. $\mathscr{C} = \mathscr{O}(N_\Gamma + k_1 N_\Gamma + k_2 N_\Gamma)$. As above, if the boundary operators can be precomputed, the reconstruction step is not required, and the computational cost is $\mathscr{C} = \mathscr{O}(k_1 k_2)$.
- Solving $\mathscr{P}_2$ with a given projected Robin datum, i.e., computing the solution of (5.14) with the projected boundary condition computed above. $\mathscr{C} = \mathscr{O}(k_2^{\delta_2})$.
- Computing the error (5.16). $\mathscr{C} = \mathscr{O}(k_1^{\theta_1}) + \mathscr{O}(k_2^{\theta_2})$.

As in the full model, the values of δ_i and $\theta_i \leq \delta_i$ depend on the local underlying problems and discretization algorithms. Typical values are $\delta_i = 3$ and $\theta_i = 2$.

To summarize, assuming that the boundary operators can be precomputed, the computational cost is:

$$\begin{aligned}\mathscr{C}_{\text{FOM}} &\approx N_{\text{online}} N_{\text{steps,online}} N_{\text{iter,FOM,avg}}\, \mathscr{O}(N_1^{\gamma_1} + N_2^{\gamma_2} + N_\Gamma),\\ \mathscr{C}_{\text{ROM}} &\approx N_{\text{online}} N_{\text{steps,online}} N_{\text{iter,ROM,avg}}\, \mathscr{O}(k_1^{\delta_1} + k_2^{\delta_2} + k_1 k_2)\\ &\quad + \frac{1}{C}\left(N_{\text{offline}} N_{exp} N_{\text{steps,offline}}\, \mathscr{O}(N_1^{\gamma_1} + N_2^{\gamma_2}) + N_1^2 k_1 + N_1 k_1^2 + N_2^2 k_2 + N_2 k_2^2\right).\end{aligned}$$

Here, the number of online parameter instances and time steps are denoted by N_{online} and $N_{\text{steps,online}}$, respectively, whereas the average number of domain decomposition iterations for the reduced order model is denoted by $N_{\text{iter,ROM,avg}}$. The meaning of the remaining quantities can be directly inferred, except for the constant C, which acts as a normalization. Indeed, the offline cost should be either amortized with the number of online evaluations of the reduced model, i.e., $C = N_{\text{online}} N_{\text{steps,online}} N_{\text{iter,ROM,avg}}$ or simply ignored, i.e., $C \to \infty$. In this case, which is the one that is often considered in practice, the resulting speedup is:

$$\frac{\mathscr{C}_{\text{FOM}}}{\mathscr{C}_{\text{ROM}}} \approx \frac{N_{\text{iter,FOM,avg}}\, \mathscr{O}(N_1^{\gamma_1} + N_2^{\gamma_2} + N_\Gamma)}{N_{\text{iter,ROM,avg}}\, \mathscr{O}(k_1^{\delta_1} + k_2^{\delta_2} + k_1 k_2)}.$$

Although it is influenced by both the dimensionality reduction factor and the ratio in the number of domain decomposition iterations, we numerically observed that the former typically dominates. In the case of equal subdomains ($N_1 = N_2 = N$, $\gamma_1 = \gamma_2 = \gamma$) and reduced dimensions ($k_1 = k_2 = k$, $\delta_1 = \delta_2 = \delta$), the expected speedup scales approximately as $\mathcal{O}(N^\gamma + N_\Gamma)/\mathcal{O}(k^\delta + k^2) \approx \mathcal{O}(N^\gamma)/\mathcal{O}(k^\delta)$.

5.3 Numerical results

We now present a number of applications. For each coupled problem, we show that the reduced model efficiently recovers the full solution, supporting this by analyzing relevant quantities of interest, including the reduction error, the computational cost and the number of domain decomposition iterations.

For each parameter value and time step, the errors between the full solution u and the reduced one u_{ROM} are measured using the relative L^2-norm in the appropriate spatial domain. This reads

$$\epsilon = \frac{\|u - u_{ROM}\|}{\|u\|} = \frac{\left(\sum_i \|u_i - u_{i,ROM}\|^2_{L^2(\Omega_i)}\right)^{1/2}}{\left(\sum_i \|u_i\|^2_{L^2(\Omega_i)}\right)^{1/2}}, \tag{5.19}$$

if a global error is considered, and

$$\epsilon_i = \frac{\|u_i - u_{i,ROM}\|_{L^2(\Omega_i)}}{\|u_i\|_{L^2(\Omega_i)}}, \tag{5.20}$$

if the local error of problem $\mathscr{P}_i$ is considered instead. When necessary, (5.19) and (5.20) are averaged over time and and a suitable parameter test set. Unless stated otherwise, the tolerance for the Schwarz iterations is $\mathrm{tol} = 10^{-6}$.

5.3.1 Advection-diffusion-reaction

Our first system of interest is a coupled time-dependent advection-diffusion-reaction problem, in which the coefficients are allowed to vary between the components. The main goal of this test case is to validate our technique in a relatively simple framework, yet allowing the constituents to be characterized by different physical phenomena. We consider $\Omega = [0,1]^2$, divided into two subdomains by a vertical interface $\Gamma = \{0.6\} \times [0,1]$. The time domain is $[0,T] = [0,0.05]$. The governing equation for $\mathscr{P}_i$ is

$$\frac{\partial u_i}{\partial t} - \mu_i \Delta u_i + \boldsymbol{\beta}_i \cdot \nabla u_i + \sigma_i u_i = 0, \quad \text{in } \Omega_i,$$

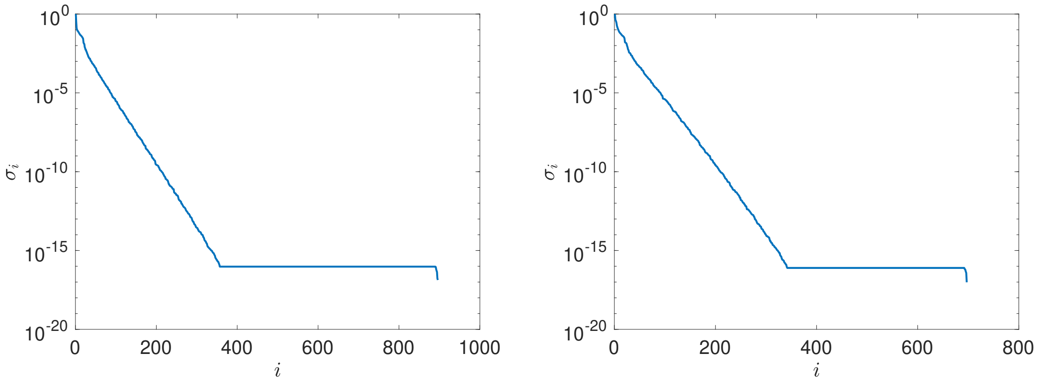
The coefficients μ_i, $\boldsymbol{\beta}_i$, σ_i control diffusion, advection and reaction, and are assumed to be constant in space and time. Homogeneous Neumann conditions are considered at the physical domain boundaries, whereas the initial condition is set as $u_0(x,y) = \cos(2\pi x)\cos\big(2\pi y\big)$ in the entire domain. We consider Robin coupling conditions at the interface, defined by

$$\mu_1 \nabla u_1 \cdot \boldsymbol{n}_1 + \lambda_1 u_1 = \mu_2 \nabla u_2 \cdot \boldsymbol{n}_1 + \lambda_1 u_2, \quad \mu_2 \nabla u_2 \cdot \boldsymbol{n}_2 + \lambda_2 u_2 = \mu_1 \nabla u_1 \cdot \boldsymbol{n}_2 + \lambda_2 u_1,$$

for suitable parameters λ_i.

Firstly, we assume all relevant parameters to be fixed. We consider $\mu_1 = 1/3$, $\boldsymbol{\beta}_1 = [3,2]^T$, $\sigma_1 = 1$, $\mu_2 = 1$, $\boldsymbol{\beta}_2 = [0,0]^T$, $\sigma_2 = 0$, while the Robin coefficients are set to $\lambda_1 = 10$, $\lambda_2 = 10$ in the offline and the online phase. The snapshots are generated from numerical simulations of the decoupled problems with $N_{basis} = 15$ and a final time $T = 0.05$. The discretization is done using a second-order finite difference method in space with grid spacing $\Delta x = \Delta y = 1/40$, and implicit Euler in time, with a time step obtained using a CFL condition with $C = 1$. The reduced dimension is set to $k = 80$ for each problem. The singular values of the local problems are reported in Figure 5.1, whereas the first four corresponding POD modes are shown in Figure 5.2. The exponential decay of the singular values indicates that both problems can be efficiently reduced, at least at a decoupled level. The most relevant modes capture the main features of the dynamics. The first mode appears to be highly influenced by the initial condition, whereas the successive ones capture additional low-frequency characteristics. The solution of the coupled problem at the final time is shown in Figure 5.3 for both the full and the reduced model, together with the reduction error. This shows the high accuracy of the reduced model, and it qualitatively validates our method. Although the bases are generated independently of the coupling, the surrogates are able to capture the features of the dynamics induced by the interaction terms. Additionally, in Figure 5.4 we report relevant quantities of interest as a function of time. We observe that, for the selected parameter values, the relative error between the full and the reduced solutions is $\mathcal{O}(10^{-4})$. The number of iterations of the domain decomposition algorithm does not vary between the full and the reduced model, consistent with the high accuracy observed in Figure 5.3. This suggests that the computational speedup is uniquely determined by the dimensionality reduction factor, and it is in this case around 8. Although this is not extremely large, at this point we didn't aim to optimize the hyperparameters, as we simply wanted to validate the method. A lower reduced dimension could be used as well, without significantly compromising the accuracy. However, we should remark that the construction proposed in Section 5.2.1 is not specific to a given coupled problem, and the reduced basis needs to be robust with respect to different values of the coupling functions. This implies that the speedup will be lower compared to more standard reduced basis approaches, as we need to select a larger value of k to achieve a given error value. We mention that, consistent with a theoretical analysis [57], the number of iterations depends on Δt in both the full and the reduced model. As our focus is on model reduction, we used the same timestep in the two cases. However, there is nothing specific about that, and it can be changed if required by a specific application.

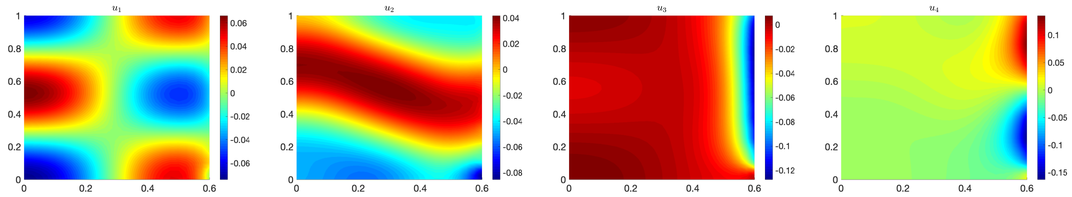
Now, we study the effects of varying hyperparameters on the reduction. Figure 5.5 reports the relative error, the computational cost and the number of iterations, averaged over time, for different values of the reduced dimension and boundary basis functions used to generate the reduced basis in the offline phase. Looking at both Figure 5.5(a) and Figure 5.5(d), we observe that an increase in the reduced dimension leads to a smaller reduction error, as expected. For a sufficiently large k, the reduction error saturates to a value related to the projection error of the Robin data onto the space spanned by the boundary basis functions. In turn,



(a) $\mathscr{P}_1$.

(b) $\mathscr{P}_2$.

Figure 5.1 – Singular values of the advection-diffusion-reaction problem.

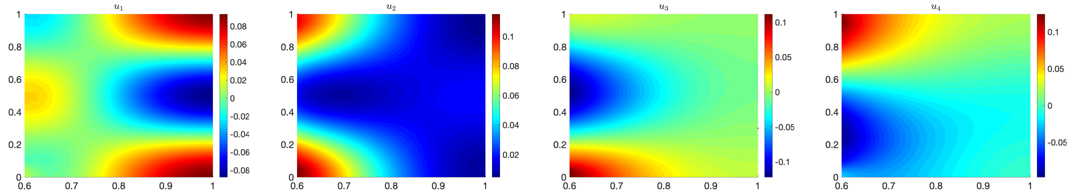


(a) $\mathscr{P}_1$, u_1.

(b) $\mathscr{P}_1$, u_2.

(c) $\mathscr{P}_1$, u_3.

(d) $\mathscr{P}_1$, u_4.



(e) $\mathscr{P}_2$, u_1.

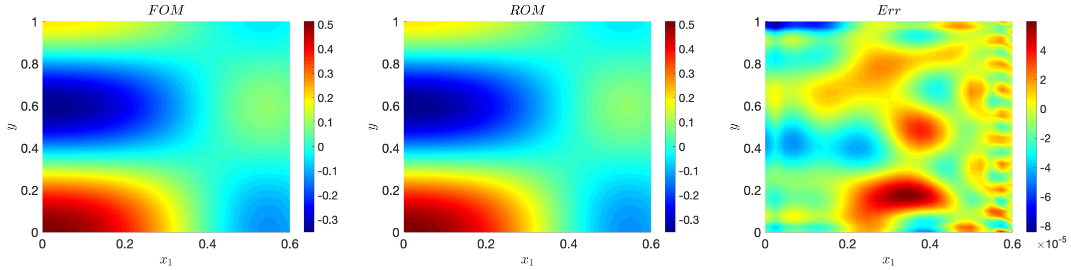
(f) $\mathscr{P}_2$, u_2.

(g) $\mathscr{P}_2$, u_3.

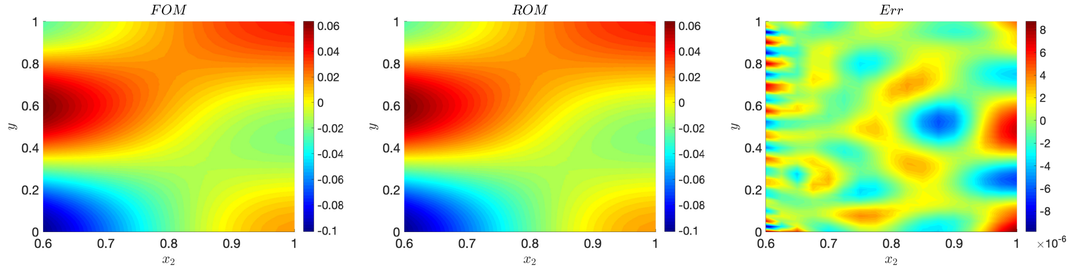
(h) $\mathscr{P}_2$, u_4.

Figure 5.2 – POD modes of the advection-diffusion-reaction problem.

increasing N_{basis}, we observe a trade-off. On one hand, as more snapshots are available, the dynamics of the full system can be better captured, and the accuracy of the reduced model improves. On the other, the heterogeneity in the snapshot matrix increases, leading to a larger variability in the reduced basis and an increase in the number of modes needed to accurately represent the solution. For a fixed number of latent variables, the former effect dominates when N_{basis} is small, and adding more snapshots improves the reduced model. When too many basis functions are added, the reduced model starts to be affected by the increasing variability, resulting in a lower level of accuracy. For these reasons, in Figure 5.5(a) we can identify two regions. For large N_{basis} and small k, the reduced basis error dominates, similar to more standard approaches. Instead, for small N_{basis} and large k the boundary projection error is the dominant component, as the basis is sufficiently rich to capture the system dynamics but is unaware of the high-frequency components induced by higher boundary harmonics. The behavior of the computational cost (Figures 5.5(b) and 5.5(e)) and number of



(a) $\mathscr{P}_1$, FOM. (b) $\mathscr{P}_1$, ROM. (c) $\mathscr{P}_1$, error.



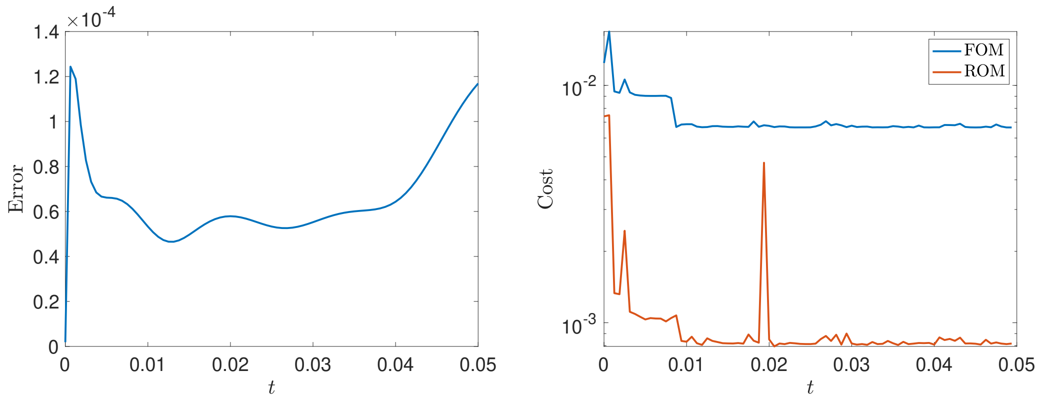
(d) $\mathscr{P}_2$, FOM. (e) $\mathscr{P}_2$, ROM. (f) $\mathscr{P}_2$, error.

Figure 5.3 – Numerical solutions and reduction error of the advection-diffusion-reaction problem.

iterations (Figures 5.5(c) and 5.5(f)) can be interpreted in a more straightforward way. The number of boundary basis functions does not impact the computational cost per number of iterations, as N_{basis} does not play a role in the online phase. As the number of domain decomposition iterations is not significantly altered when varying the hyperparameters, the total computational cost is only affected by k, up to minor variations.

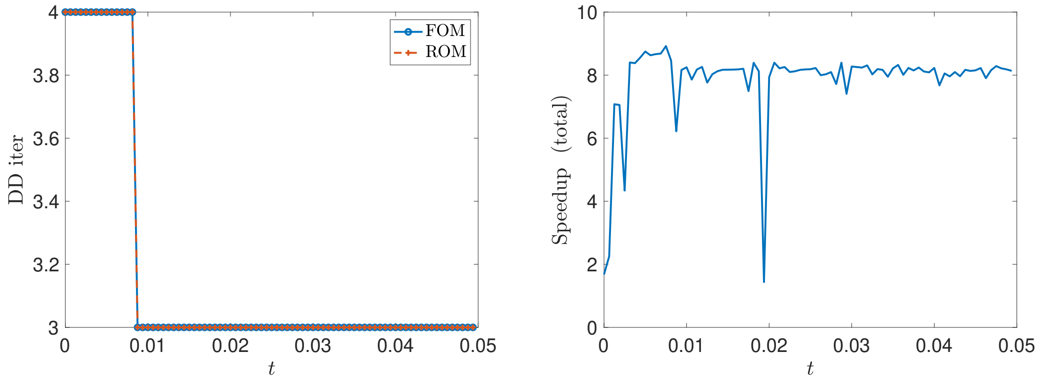
Although the offline phase is based on a parametrization of Robin data with a fixed λ, the reduced basis is able to capture the dynamics of the system with other coupling functions, including, e.g., Robin data with different coefficients. Figure 5.6 shows a comparison of different quantities of interest under variations in the Robin parameter used in the online phase. The reduction error is rather robust with respect to variations of λ, although minor differences are present. These are unavoidable, but the features of the dynamics are well captured even with a single offline instance of λ. Notably, the error decay as a function of k is not affected by changes in λ. The number of domain decomposition iterations, and consequently the computational cost, exhibit more significant changes. The former is related to the optimization of the interface conditions in domain decomposition methods, in which the value of λ is shown to impact the convergence rate [59]. The latter varies accordingly, as it is proportional to the number of iterations.

Additionally, in Figure 5.7 we report the trend observed when the Robin parameter used in the training phase is varied, while keeping the λ used in the testing phase constant. The reduction error is slightly affected, with lower accuracy for extreme values of λ. However, this does not have a significant impact on the error behavior as a function of k. The number of



(a) Reduction error.

(b) Computational cost.



(c) Iterations.

(d) Speedup.

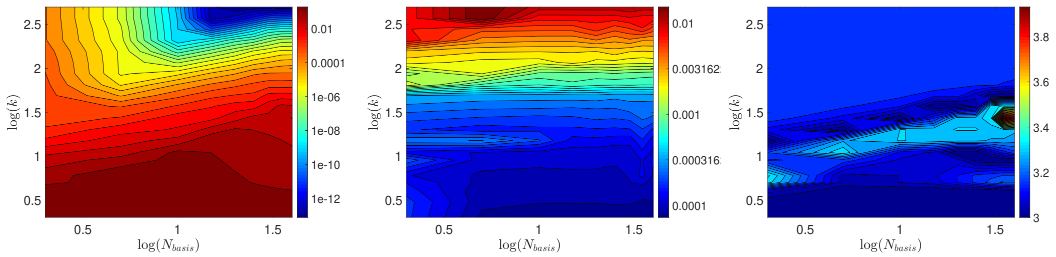
Figure 5.4 – Quantities of interest of the advection-diffusion-reaction problem.

iterations and the computational cost are not strongly influenced by a change in λ, as they mostly depend on the parameters used in the online phase.

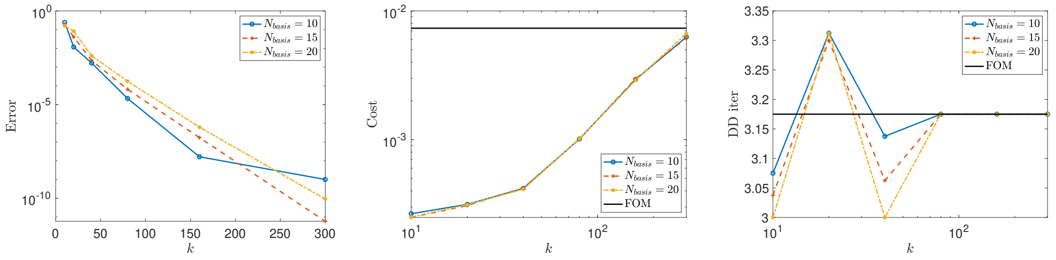
To conclude, we mention that the reduced model can be used to speedup optimization problems. For instance, one could be interested in tuning the Robin parameters in order to minimize the number of domain decomposition iterations. Although constructing an appropriate control problem goes beyond the scope of this work, we argue that the surrogate model could be used in such a context. Figure 5.8 shows the iterations of the domain decomposition method under variations in the Robin parameters λ_1 and λ_2 for the two problems. The differences between the full and reduced model are mild. In turn, the associated computational cost, although not reported here, exhibits significant changes, similar to Figure 5.4.

5.3.2 Checkerboard diffusion

We now focus on another problem, a steady parametrized diffusion equation. We seek to show that the method can be efficiently applied to parametrized problems, proving its effectiveness also in high-dimensional parameter spaces. The computational domain is $\Omega = [0,1]^2$, divided



(a) Relative error. (b) Computational cost. (c) Iterations.



(d) Relative error. (e) Computational cost. (f) Iterations.

Figure 5.5 – Comparison (number of basis functions and reduced dimension) of the time-averaged quantities of interest of the advection-diffusion-reaction problem.

into two subdomains by a vertical interface $\Gamma = \left\{\frac{2}{3}\right\} \times [0,1]$. Problem $\mathscr{P}_i$ is governed by

$$-\nabla\cdot(\mu_i \nabla u_i) = 0, \qquad \text{in } \Omega_i.$$

We take the diffusion coefficients to be piecewise constant in the domain in a checkerboard fashion, i.e., we assume

$$\mu_1 = \mu_1^1 \mathbb{1}_{[0,\frac{1}{3}]\times[0,\frac{1}{3}]} + \mu_1^2 \mathbb{1}_{[0,\frac{1}{3}]\times[\frac{1}{3},\frac{2}{3}]} + \mu_1^3 \mathbb{1}_{[0,\frac{1}{3}]\times[\frac{2}{3},1]} + \mu_1^4 \mathbb{1}_{[\frac{1}{3},\frac{2}{3}]\times[0,\frac{1}{3}]} + \mu_1^5 \mathbb{1}_{[\frac{1}{3},\frac{2}{3}]\times[\frac{1}{3},\frac{2}{3}]} + \mu_1^6 \mathbb{1}_{[\frac{1}{3},\frac{2}{3}]\times[\frac{2}{3},1]}$$

and

$$\mu_2 = \mu_2^1 \mathbb{1}_{[\frac{2}{3},1]\times[0,\frac{1}{3}]} + \mu_2^2 \mathbb{1}_{[\frac{2}{3},1]\times[\frac{1}{3},\frac{2}{3}]} + 1\mathbb{1}_{[\frac{2}{3},1]\times[\frac{2}{3},1]}.$$

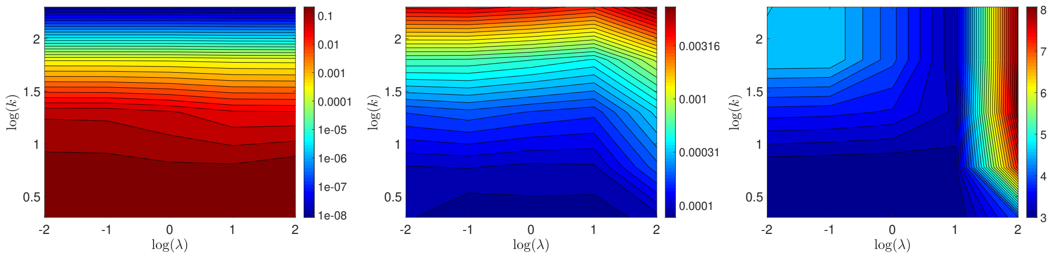
We consider parametrized non-homogeneous Dirichlet boundary conditions at the left boundary of Ω_1 and at the right boundary of Ω_2, i.e.,

$$u_1 = g_1 \text{ on } \{0\}\times[0,1], \qquad u_2 = g_2 \text{ on } \{1\}\times[0,1],$$

whereas homogeneous Dirichlet conditions are assumed on all the other physical boundaries. As in the previous example, at the interface we impose Robin boundary conditions of the form

$$\mu_1\nabla u_1\cdot\boldsymbol{n}_1 + \lambda_1 u_1 = \mu_2\nabla u_2\cdot\boldsymbol{n}_1 + \lambda_1 u_2, \quad \mu_2\nabla u_2\cdot\boldsymbol{n}_2 + \lambda_2 u_2 = \mu_1\nabla u_1\cdot\boldsymbol{n}_2 + \lambda_2 u_1.$$

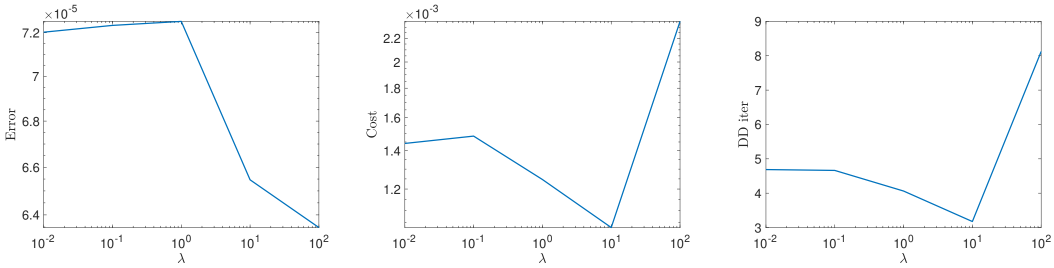
The parameters of interest are $\mu_1^i \in [0.1,10]$ for $i = 1,\dots,6$, $\mu_2^i \in [0.1,10]$ for $i = 1,2$, $g_1 \in [1,5]$, $g_2 \in [1,5]$, so that the global parameter space has dimension 10. This splits into two local



(a) Relative error.

(b) Computational cost.

(c) Iterations.



(d) Relative error ($k = 80$).

(e) Computational cost ($k = 80$).

(f) Iterations ($k = 80$).

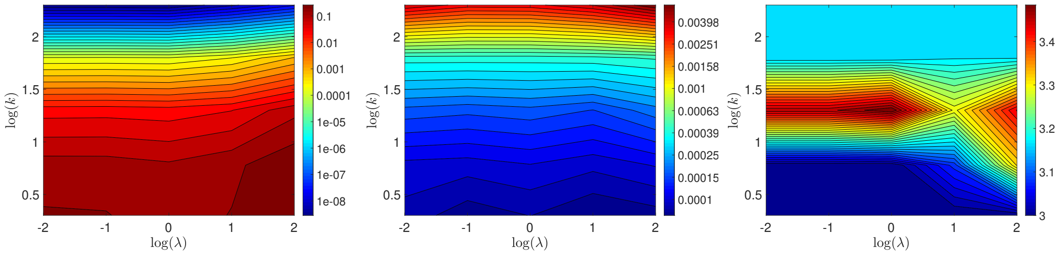
Figure 5.6 – Comparison (Robin parameter of the online phase) of the time-averaged quantities of interest of the advection-diffusion-reaction problem.

spaces of dimensions 7 and 3, respectively.

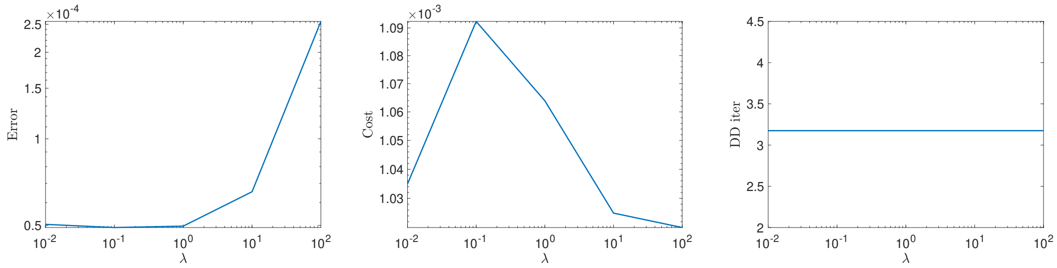
The reduced models are generated by sampling 50 parameter values using a Latin Hypercube Sampling in each subdomain and $N_{basis} = 15$. The discretization is done using $\mathbb{P}_1$ finite elements with grid spacing $h \approx \sqrt{2}/60$. The reduced dimension is set to $k = 40$ for each subdomain. The singular values of the local problems are reported in Figure 5.9, whereas the solution of the coupled problem and the corresponding reduction error for a single parameter instance not included in the training set are reported in Figure 5.10. As in the previous test case, the reduced models accurately approximate the high-fidelity solution.

We now study the reduction error, the computational cost and the number of iterations as a function of the reduced dimension. They are reported in Figures 5.11(a), 5.11(b) and 5.11(c), and Figures 5.11(d), 5.11(e) and 5.11(f) for 20 and 200 offline parameter samples per subdomain, respectively. In both cases, they have been computed by averaging the relative error (5.19), the cost and the number of iterations over 100 parameter values generated again using a Latin Hypercube Sampling, but different from the training samples. Increasing the reduced dimension improves the accuracy, until a saturation level, related to the projection of the boundary datum onto the set spanned by the Laplace-Beltrami eigenfunctions, is reached. In turn, increasing N_{basis} improves the accuracy by reducing the saturation value, without impacting on the online computational cost.

Furthermore, Figure 5.11 allows us to compare the performance of the reduced bases generated using either the decoupled problems or the coupled problem. The second basis is generated



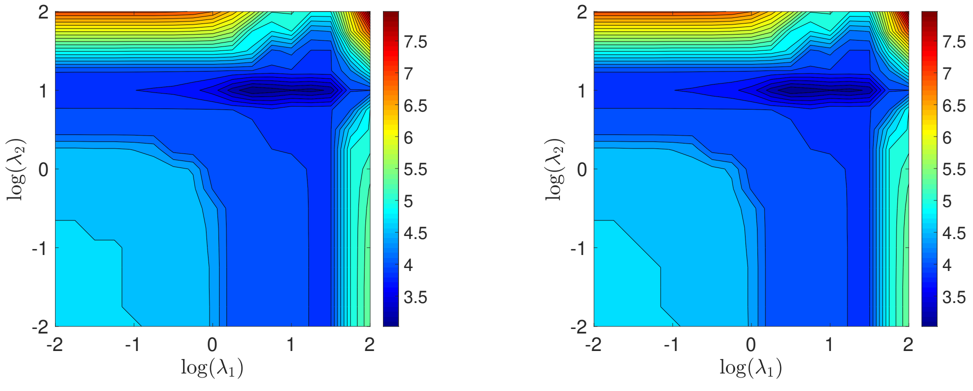
(a) Relative error. (b) Computational cost. (c) Iterations.



(d) Relative error ($k = 80$). (e) Computational cost ($k = 80$). (f) Iterations ($k = 80$).

Figure 5.7 – Comparison (Robin parameter of the offline phase) of the time-averaged quantities of interest of the advection-diffusion-reaction problem.

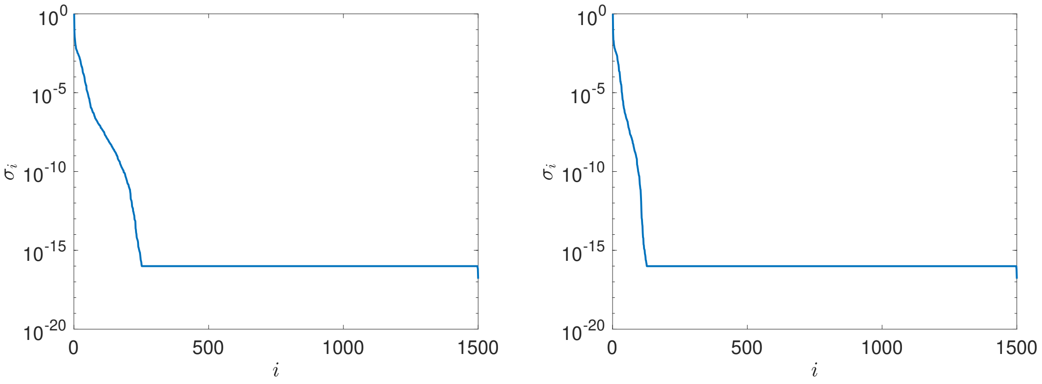
from localized snapshots obtained from simulations of the coupled problem, keeping the other hyperparameters unchanged. In this way, the basis functions remain localized to each component, but they are aware of the dynamics of the coupled problem. We observe that if the number of parameter samples is sufficiently large or a relatively low level of accuracy is sufficient for the specific application, the basis generated with the coupled problem is more effective. This is tailored for that specific system, so it is expected to perform better than a general, system-independent, model. However, when the online parameters are chosen in such a way that the dynamics is very different from the one of the snapshots, using the decoupled systems allows one to achieve a low reduction error, thanks to the generality of the model. This is particularly evident when the number of offline parameter samples is small, as in Figure 5.11(a). The sampling error in the coupled model is large, limiting the accuracy of the reduced model. This effect is mitigated by considering the systems independently, as the local parameter spaces can be sampled more effectively, despite the need to parametrize the interface boundary conditions. As the dimension of the parameter space grows, this effect becomes more significant. Finally, we recall that in many applications the global coupled problem is not available when constructing the basis. Thus, although in some cases the basis generated from localized snapshots outperforms the proposed method, it might not be possible to construct it.



(a) FOM.

(b) ROM.

Figure 5.8 – Comparison (Robin parameters λ_1 and λ_2) of the time-averaged number of iterations of the advection-diffusion-reaction problem.



(a) $\mathscr{P}_1$.

(b) $\mathscr{P}_2$.

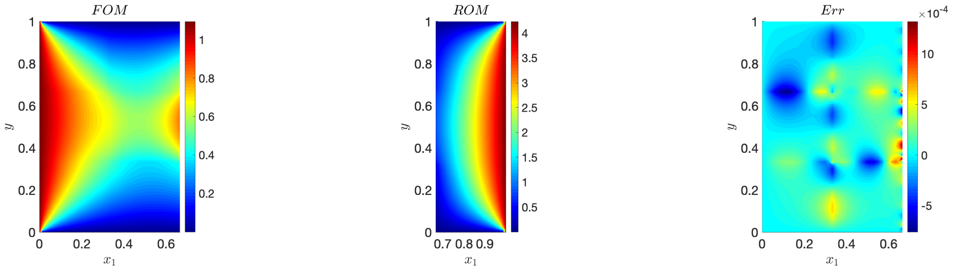
Figure 5.9 – Singular values of the checkerboard diffusion problem.

5.3.3 Multiple subdomains

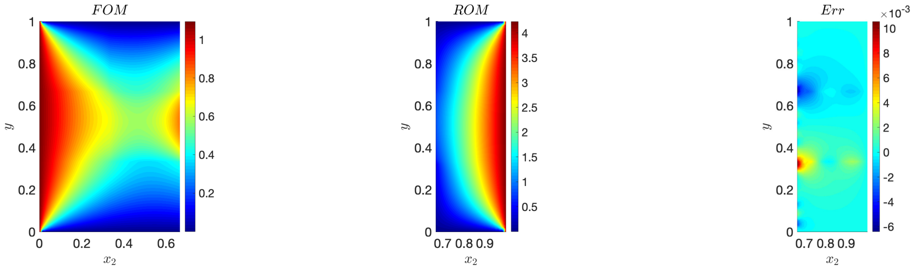
To further increase the complexity of the problem, we consider a variation in the number of subdomains. The global spatial domain is $\Omega = [0,1]^2$, which is split into M^2 square subdomains, obtained by dividing each direction into M intervals. Here, we focus on $M = 2$ and $M = 3$ only, as they already retain the properties that larger problems would feature. The problem we consider is a time-dependent diffusion equation, where the viscosity is assumed to be constant in each subdomain. Thus, problem $\mathscr{P}_i$ is governed by

$$\frac{\partial u_i}{\partial t} - \mu_i \Delta u_i = f_i, \quad \text{in } \Omega_i.$$

The diffusion coefficients are chosen as shown in Figure 5.12, whereas $f_i = 1$ in each subproblem. Moreover, homogeneous Neumann conditions are considered at the physical boundaries, the initial condition is $u_0(x,y) = \cos(2\pi x)\cos\big(2\pi y\big)$, and the final time is $T = 0.2$. At each



(a) $\mathscr{P}_1$, FOM. (b) $\mathscr{P}_1$, ROM. (c) $\mathscr{P}_1$, error.



(d) $\mathscr{P}_2$, FOM. (e) $\mathscr{P}_2$, ROM. (f) $\mathscr{P}_2$, error.

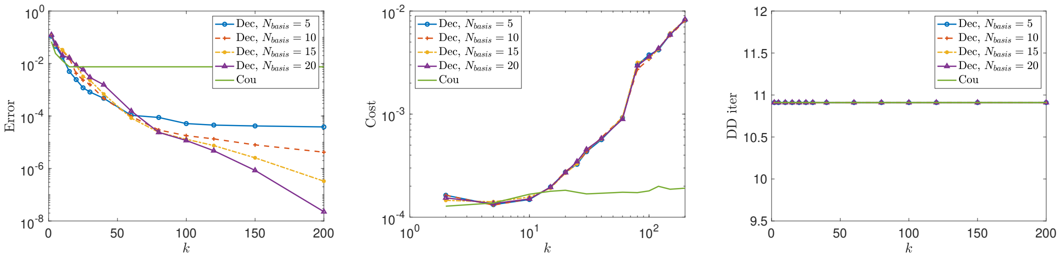
Figure 5.10 – Numerical solutions and reduction error of the checkerboard diffusion problem.

interface Γ_{ij} of problem $\mathscr{P}_i$, we impose Robin boundary conditions of the form

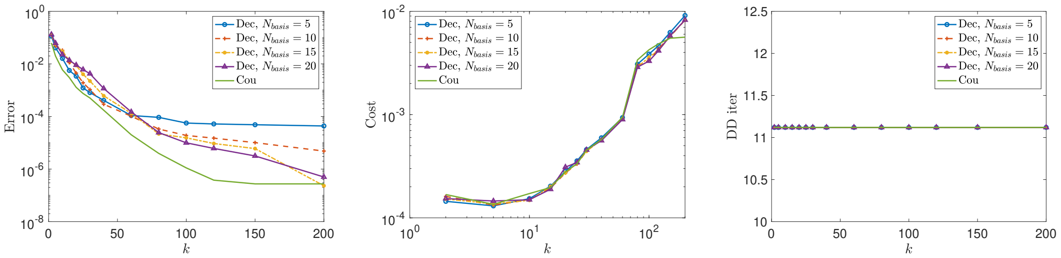
$$\mu_i \nabla u_i \cdot \boldsymbol{n}_i + \lambda_{ij} u_i = \mu_j \nabla u_j \cdot \boldsymbol{n}_i + \lambda_{ij} u_j,$$

where u_i and u_j denote the local and neighboring solution, respectively. We choose all the Robin parameters λ_{ij} equal to 1. The discretization is done using $\mathbb{P}_1$ finite elements with grid spacing $h \approx \sqrt{2}/(20M)$ and a time step $\Delta t = 0.01$. This implies that the local number of degrees of freedom is independent of M, in the spirit of a weak scaling analysis [22].

As a visual comparison of the full and reduced solution gives satisfactory results, in Figure 5.13 we directly report the relevant quantities of interest as a function of the reduced dimension for different N_{basis}. The results are similar to the previous problems, to which we refer for a more detailed discussion. We note that increasing the number of subdomains and keeping the other parameter values unchanged leads to a slight increase in the error, especially for large N_{basis}. This is expected, as the complexity of the local solutions increase due to the larger number of interfaces. An increase in the number of subdomains also leads to an increase in the number of the DD iterations. This is again consistent with our expectations, as more iterations are needed for information to propagate throughout the entire domain when it is decomposed into more components. To conclude, we mention that in our simulations we processed the elements using a two-color, checkerboard pattern. A different choice may lead to a different number of iterations and a different number of sequential step required in a parallel implementation [22], both at the full and the reduced level.



(a) Relative error (20 samples). (b) Computational cost (20 samples). (c) Iterations (20 samples).



(d) Relative error (200 samples). (e) Computational cost (200 samples). (f) Iterations (200 samples).

Figure 5.11 – Comparison (number of basis functions and reduced dimension) of the parameter-averaged quantities of interest of the checkerboard diffusion problem. The terms 'Dec' and 'Cou' refer to the bases generated using the decoupled problems or the coupled problem, respectively.

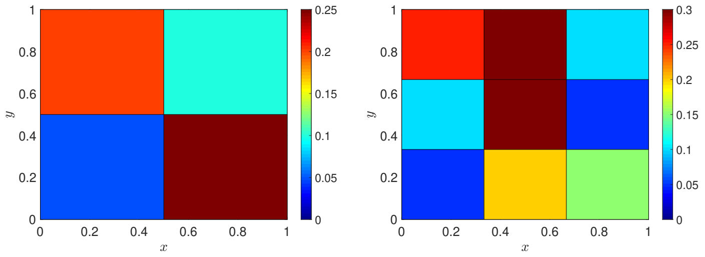
5.3.4 FitzHugh-Nagumo

We now consider the two-dimensional FitzHugh-Nagumo model, as an example of a reaction-diffusion problem. Because of the nonlinear reaction terms, the solution develops Turing patterns [55]. The main goal is to extend the presented approach to both vector-valued differential equations and nonlinear problems. We consider $\Omega = [0,\pi]^2$, divided into two subdomains by a vertical interface $\Gamma = \left\{\frac{\pi}{2}\right\} \times [0,\pi]$. The time domain is $[0,T] = [0,0.2]$. The governing equation for $\mathscr{P}_i$ is

$$\begin{cases} \dfrac{\partial u_i}{\partial t} - \Delta u_i - \Gamma\left(-u_i^3 + u_i - v_i\right) & = 0, \quad \text{in } \Omega_i, \\ \dfrac{\partial v_i}{\partial t} - d\Delta v_i - \Gamma\left(\beta\left(u_i - \alpha v_i\right)\right) & = 0, \quad \text{in } \Omega_i. \end{cases}$$

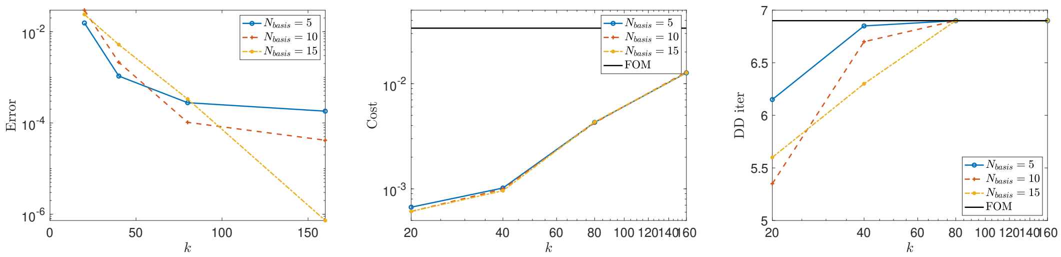
The variables u_i and v_i represent the electric potential and a recovery variable, respectively, while the coefficients β, α, d, Γ govern different physical phenomena. They are chosen to have the same value in the two subdomains. Homogeneous Neumann conditions are considered at the physical domain boundaries, whereas the initial condition is set as

$$\begin{cases} u_0(x,y) = \epsilon \exp\left(-\frac{(x-\pi/2)^2}{2\sigma^2} - \frac{(y-\pi/2)^2}{2\sigma^2}\right), \\ v_0(x,y) = \epsilon \exp\left(-\frac{(x-\pi/2)^2}{2\sigma^2} - \frac{(y-\pi/2)^2}{2\sigma^2}\right), \end{cases}$$

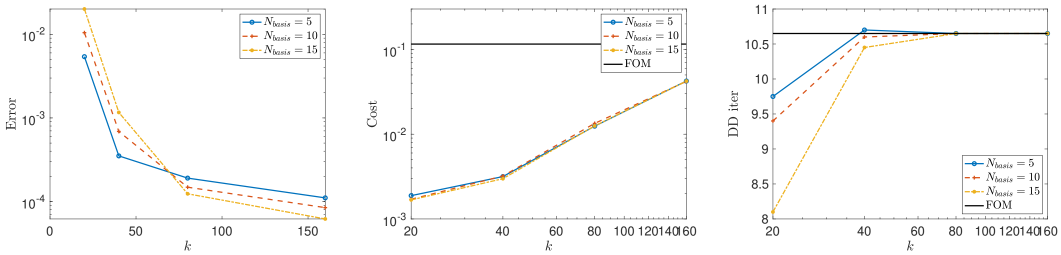


(a) $M = 2$.   (b) $M = 3$.

Figure 5.12 – Viscosity distribution of the multiple subdomains problem.



(a) Relative error ($M = 2$).   (b) Computational cost ($M = 2$).   (c) Iterations ($M = 2$).



(d) Relative error ($M = 3$).   (e) Computational cost ($M = 3$).   (f) Iterations ($M = 3$).

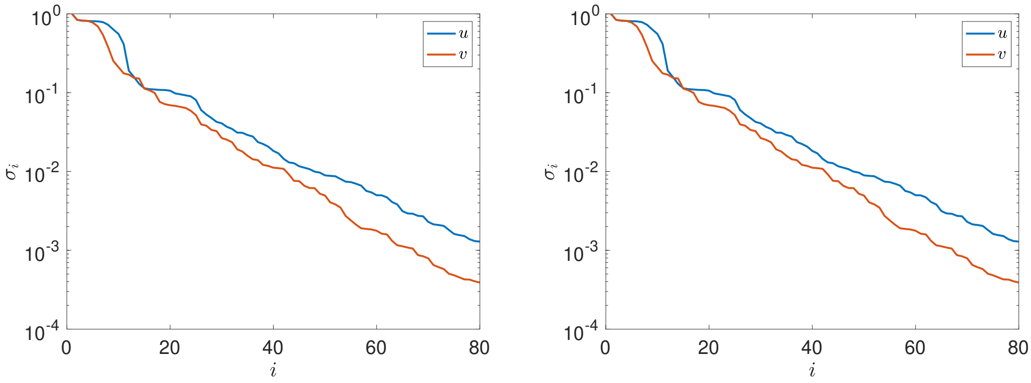
Figure 5.13 – Comparison (number of basis functions and reduced dimension) of the time-averaged quantities of interest of the multiple subdomains problem.

where ϵ controls the amplitude. We consider Robin coupling conditions at the interface, defined by

$$\begin{cases} \nabla u_1 \cdot \boldsymbol{n}_1 + \lambda_1 u_1 = \nabla u_2 \cdot \boldsymbol{n}_1 + \lambda_1 u_2, \quad \nabla u_2 \cdot \boldsymbol{n}_2 + \lambda_2 u_2 = \nabla u_1 \cdot \boldsymbol{n}_2 + \lambda_2 u_1, \\ d\nabla v_1 \cdot \boldsymbol{n}_1 + \lambda_1 v_1 = d\nabla v_2 \cdot \boldsymbol{n}_1 + \lambda_1 v_2, \quad d\nabla v_2 \cdot \boldsymbol{n}_2 + \lambda_2 v_2 = d\nabla v_1 \cdot \boldsymbol{n}_2 + \lambda_2 v_1, \end{cases}$$

for suitable parameters λ_i.

Firstly, we focus again on a specific setting, by fixing the relevant parameters. Following a theoretical analysis [55], they are chosen as $\alpha = 0.1$, $\beta = 11$, $\epsilon = 0.1$, $d = \beta\left(2\sqrt{1-\alpha} + 2 - \alpha\right)(\epsilon^2 + 1) \simeq 42.1887$, $\Gamma = -64d/(\alpha\beta - d) \simeq 65.731$. The Robin coefficients are set to $\lambda_1 = 10$, $\lambda_2 = 10$ in the offline and the online phase. The snapshots are generated from numerical simulations of the decoupled problems with a number of basis $N_{basis} = 10$ for both u and v and a final time $T = 0.2$. The discretization is done using second-order finite differences in space with grid spacing $\Delta x = \Delta y = 1/40$, and implicit Euler in time, with a time step obtained using a CFL condition with $C = 40$. The reduced dimension is set to $k = 80$ for each problem and variable. The nonlinear term is treated efficiently in the reduced model using DEIM, with a corresponding basis size $\tilde{k} = k$. The first k singular values of the local problems are reported in Figure 5.14 for each variable. Again, they have a large reduction potential. Similarly, in Figure 5.15 we show the singular values associated to the nonlinear term and the indices selected by the DEIM as described in Section 5.2. The decay is again quite fast, and the effect of the nonlinearity appears to be stronger closer to the interface. This is not surprising, as in the offline phase we impose a large variability of the coupling conditions, which seems to affect the nonlinearity in a stronger way. The solution u of the coupled problem at the final time is reported in Figure 5.16 for both the full and the reduced model, together with the reduction error. Although the nonlinear term increases the complexity of the dynamics and the reduced model, the solution is well approximated with the reduced basis. Relevant indicators are shown in Figure 5.17 as a function of time. We observe that the reduced model accurately reproduces the solution (the relative error is around 1%), resulting in a speedup factor of 15. Again, the number of local iterations is practically independent of the dimension of the problem.

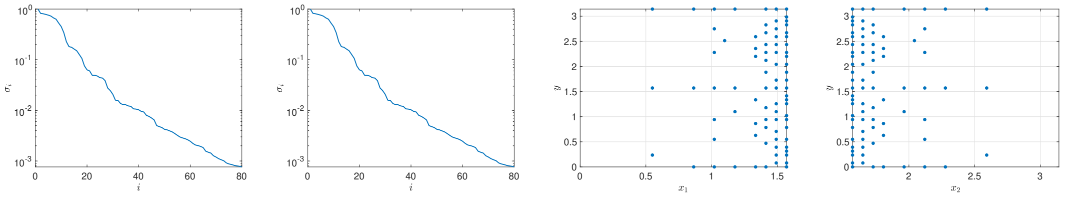


(a) $\mathscr{P}_1$.

(b) $\mathscr{P}_2$.

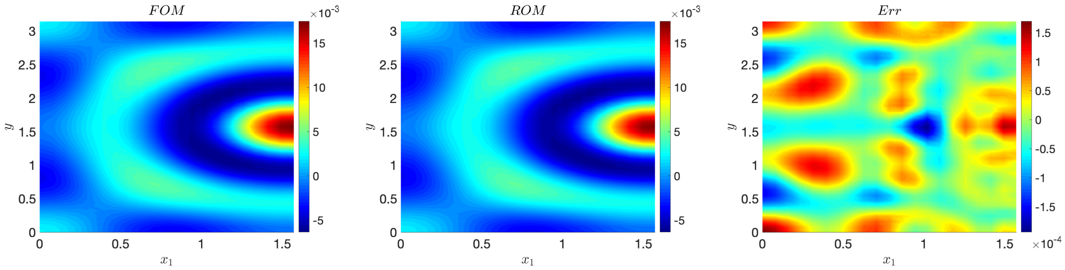
Figure 5.14 – Singular values of the FitzHugh-Nagumo problem.

The effect of varying hyperparameters on the reduction is studied in Figure 5.18. We report the main performance indicators, averaged over time, for different values of the reduced dimension and boundary basis functions used in the offline phase. In terms of reduction error, the behavior as a function of the reduced dimension reflects our expectations, as the trend is exponentially decreasing. When the number of boundary basis functions is varied, we still experience the trade-off between higher accuracy and larger variability, at least when the number of latent variables is sufficiently large. The number of domain decomposition iterations has a larger variation compared to the previous test cases, and it is influenced by both the reduced dimension and the number of boundary basis functions. This reflects on the

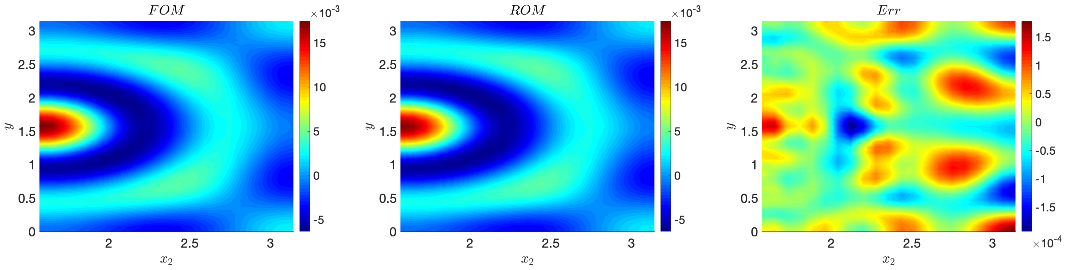


(a) Singular values of the nonlinearity, $\mathscr{P}_1$. (b) Singular values of the nonlinearity, $\mathscr{P}_2$. (c) DEIM indices, $\mathscr{P}_1$. (d) DEIM indices, $\mathscr{P}_2$.

Figure 5.15 – Singular values of the nonlinear term and DEIM indices of the FitzHugh-Nagumo problem.



(a) $\mathscr{P}_1$, FOM. (b) $\mathscr{P}_1$, ROM. (c) $\mathscr{P}_1$, error.



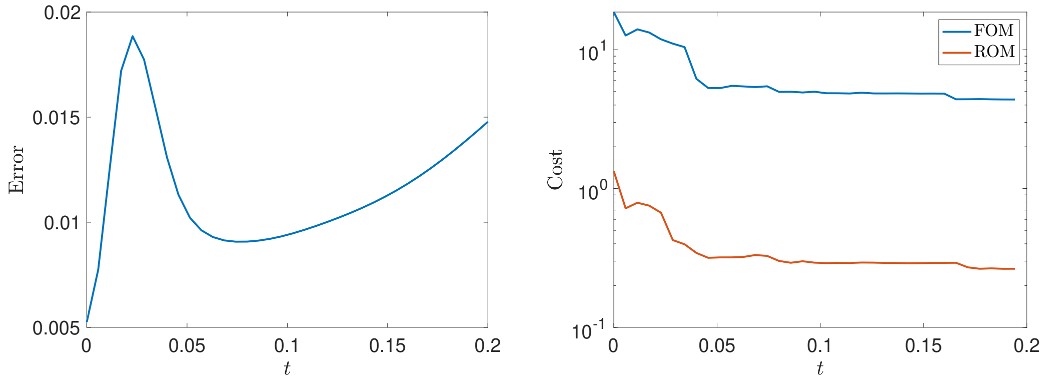
(d) $\mathscr{P}_2$, FOM. (e) $\mathscr{P}_2$, ROM. (f) $\mathscr{P}_2$, error.

Figure 5.16 – Numerical solutions and reduction error (first component) of the FitzHugh-Nagumo problem.

computational cost, which depends on the dimensionality reduction factor and the varying number of iterations.

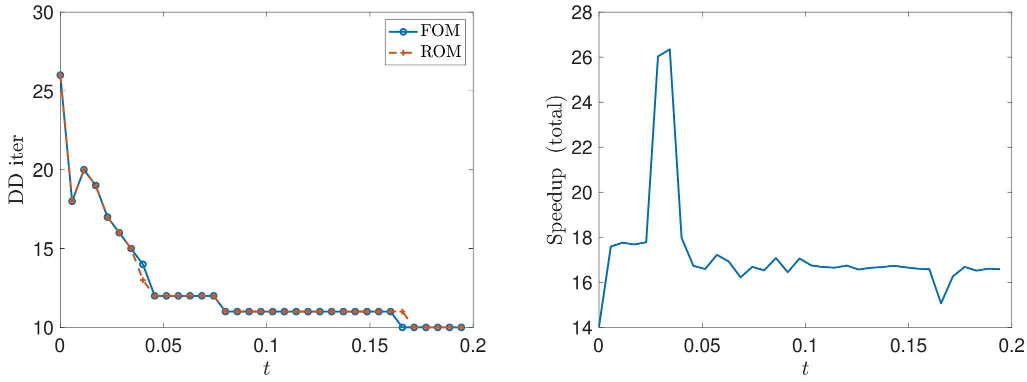
5.3.5 Stokes-Darcy coupling

We now switch to a simple multi-physics problem, modeling the interaction between an incompressible flow governed by the unsteady Stokes equations and a porous medium described by the unsteady Darcy equation [2]. Although in a relatively simple context, this problem retains the main features of a multi-physics system, and is used to validate our technique with a more heterogeneous coupled system. We consider $\Omega = [0,1]^2$, divided into two subdomains by a horizontal interface $\Gamma = [0,1] \times \{0.9\}$. The time domain is $[0,T] = [0,10]$. The governing



(a) Reduction error.

(b) Computational cost.



(c) Iterations.

(d) Speedup.

Figure 5.17 – Quantities of interest of the FitzHugh-Nagumo problem.

equation for $\mathscr{P}_1$ is

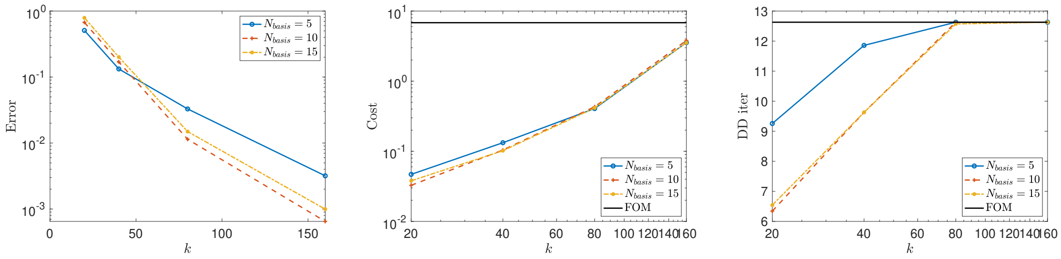
$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} - \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}, p) & = 0, \quad \text{in } \Omega_1, \\ \nabla \cdot \boldsymbol{u} & = 0, \quad \text{in } \Omega_1, \end{cases} \tag{5.21}$$

where $\boldsymbol{\sigma}\left(\boldsymbol{u}, p\right) = \nu\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right) - p\boldsymbol{I}$ is the Cauchy stress tensor, $\nu = 0.04$ the fluid viscosity and $\boldsymbol{u}$ and p are the fluid velocity and pressure. The model for $\mathscr{P}_2$ is

$$\frac{\partial q}{\partial t} - \nabla \cdot (K \nabla q) = 0, \quad \text{in } \Omega_2,$$

where $K = 0.2$ is the permeability coefficient. The fluid problem is completed by no-slip conditions at the lateral boundaries $\{0,1\} \times [0.9,1]$ and a constant velocity profile $\boldsymbol{u} = (0, -10x(1-x))^T$ at the top boundary $[0,1] \times \{1\}$, while the Darcy problem is solved with homogeneous Neumann conditions at the lateral boundaries $\{0,1\} \times [0,0.9]$ and homogeneous Dirichlet at the bottom boundary $[0,1] \times \{0\}$. The initial condition is set to zero for both problems. At the interface we consider

$$\boldsymbol{\sigma}(\boldsymbol{u}, p)\boldsymbol{n}_1 \cdot \boldsymbol{n}_1 = -q, \quad K \nabla q \cdot \boldsymbol{n}_2 = \boldsymbol{u} \cdot \boldsymbol{n}_1,$$



(a) Relative error.

(b) Computational cost.

(c) Iterations.

Figure 5.18 – Comparison (number of basis functions and reduced dimension) of the time-averaged quantities of interest of the FitzHugh-Nagumo problem.

together with a zero tangential velocity component for the fluid problem. We note that Neumann interface conditions are used to define the coupled problem. This is common practice for this problem, as they naturally arise in the weak form of the coupled problem and lead to a symmetric system [42]. By introducing suitable coefficients, they can be combined in a Robin form. As our main goal is to show that we can efficiently recover the solution of the coupled problem by constructing local reduced models, rather than optimizing the coupling conditions, we stick to this formulation for both the offline and online phase. Moreover, as shown for simpler problems, we can expect that a nonzero Robin parameter would not significantly alter the results.

Our numerical setting comprises a number of basis $N_{basis} = 10$ and a final time $T = 10$. The spatial discretization is done using $\mathbb{P}_2$-$\mathbb{P}_1$, respectively $\mathbb{P}_2$, finite elements with a characteristic mesh size $h \approx \sqrt{2}/40$ for the Stokes, respectively Darcy, problem. An implicit Euler method with a timestep $\Delta t = 0.5$ is used for time integration. As this leads to a saddle point problem for the fluid component, particular care has to be given to the construction of the reduced model, as the vanilla reduced-basis model introduced in Chapter 2 may not be (inf-sup) stable. Specifically, integrating (5.21) requires solving a system of the form

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}. \tag{5.22}$$

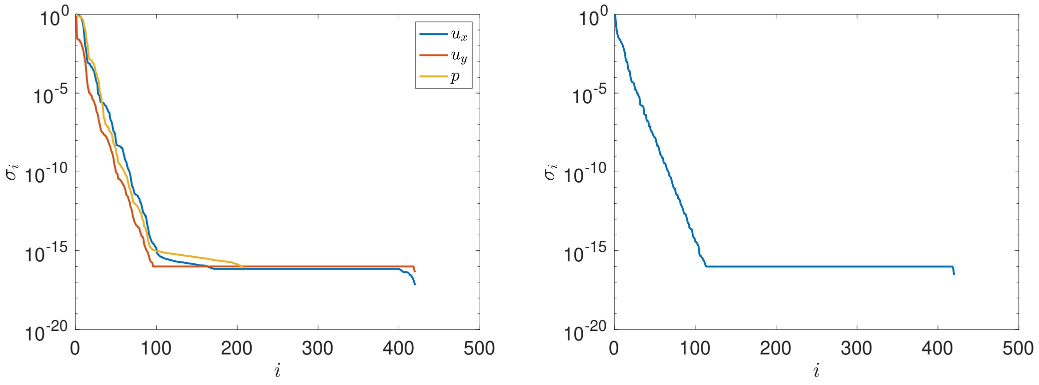
With our choice of the finite element spaces for the approximation of $\mathbf{u}$ and p, (5.22) is well-posed. The reduced counterpart, constructed by computing the velocity and pressure bases $U_\mathbf{u}$ and U_p as described in Chapter 2,

$$\begin{bmatrix} U_\mathbf{u}^T \mathbf{A} U_\mathbf{u} & U_\mathbf{u}^T \mathbf{B}^T U_p \\ U_p^T \mathbf{B} U_\mathbf{u} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \alpha_\mathbf{u} \\ \alpha_p \end{bmatrix} = \begin{bmatrix} U_\mathbf{u}^T \mathbf{f} \\ U_p^T \mathbf{g} \end{bmatrix} \tag{5.23}$$

does not enjoy the same property. For this reason, we enrich the velocity basis $U_\mathbf{u}$ with properly chosen (approximated) supremizer solutions [8]. The idea is to map each element of the pressure space to a suitable velocity element ensuring the inf-sup condition, and to include these in the velocity space. This leads to an enriched reduced space, making (5.23) stable. Consequently, we solve a problem of the form

$$\mathbf{K}\,\mathbf{s}(p_j) = \mathbf{B}^T p_j \tag{5.24}$$

for each pressure snapshot p_j, collect the solutions $\mathbf{s}(p_j)$, and apply the POD to obtain a number of supremizer modes that are added to the matrix $U_{\mathbf{u}}$. The matrix $\mathbf{K}$ in (5.24) can be chosen as the one arising from the discretization of the Laplace operator for $\mathbf{u}$. We remark that the supremizer enrichment is not the only technique that ensures stability of the reduced system. Examples are the pressure Poisson formulation of the Navier-Stokes equations [121] and the artificial compressibility method [36], which ensure stability at the full and the reduced level for more general velocity-pressure spaces without the need to enrich the velocity space. For this particular problem, the reduced dimension is set to $k = 40$ for each problem and variable, and the velocity space is enriched with 40 supremizer modes. The singular values of the local problems are reported in Figure 5.19 for each variable. The decay is quite fast, meaning that the intrinsic dimension of the local problems is quite small. The solution of the coupled problem at the final time is reported in Figure 5.20, together with the reduction error. All components are accurately reproduced at a reduced level. Relevant indicators are shown in Figure 5.21 as a function of time. For the selected hyperparameters, both the level of accuracy and the computational speedup reach high values. We note that, although the introduction of the supremizers negatively affects the computational cost, the speedup is still significant. The number of iterations is found to be independent of the dimension of the problem.

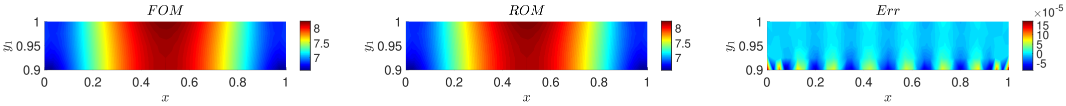


(a) $\mathscr{P}_1$.

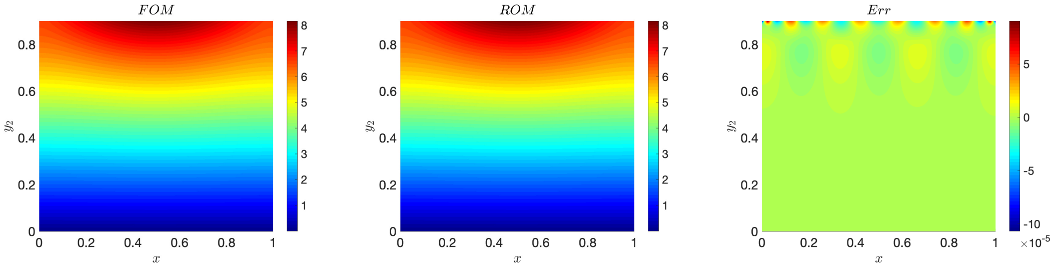
(b) $\mathscr{P}_2$.

Figure 5.19 – Singular values of the Stokes-Darcy problem.

Concerning variations in the hyperparameters of the reduced model, Figure 5.22 reports the relevant quantities of interest for different values of the reduced dimension and basis functions used to generate the snapshots in the offline phase. Most of the comments related to the previous test cases remain valid, even though the trend of the reduction error is less smooth as compared to Figure 5.5. This is possibly due to the larger heterogeneity and complexity of the problem, the addition of the supremizer modes, or a higher level of noise introduced by higher frequency components.



(a) $\mathscr{P}_1$, p, FOM. (b) $\mathscr{P}_1$, p, ROM. (c) $\mathscr{P}_1$, p, error.



(d) $\mathscr{P}_2$, FOM. (e) $\mathscr{P}_2$, ROM. (f) $\mathscr{P}_2$, error.

Figure 5.20 – Numerical solutions and reduction error of the Stokes-Darcy problem.

5.3.6 Fluid-Structure Interaction I

Our next test case involves a more complex, heterogeneous multi-physics problem, selected as a prototype of general fluid-structure interaction problems. This is used to show the applicability of our reduction method in a scenario which is even closer to practical multi-physics problems. We consider $\Omega = [0, 1.2] \times [0, 0.5]$, divided into two subdomains by a curved interface as depicted in Figure 5.23. The time domain is $[0, T] = [0, 0.5]$. The governing equations for $\mathscr{P}_1$ are again the Stokes equations

$$\begin{cases} \rho^f \frac{\partial \mathbf{u}}{\partial t} - \nabla \cdot \boldsymbol{\sigma}^f(\boldsymbol{u}, p) & = 0, \quad \text{in } \Omega_1, \\ \nabla \cdot \boldsymbol{u} & = 0, \quad \text{in } \Omega_1, \end{cases}$$

where $\boldsymbol{\sigma}^f(\boldsymbol{u}, p) = \mu^f \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right) - p\boldsymbol{I}$ is the Cauchy stress tensor, $\rho^f = 1.0$ and $\mu^f = 1.0$ are the fluid density and viscosity, respectively, and $\boldsymbol{u}$ and p are the fluid velocity and pressure. Conversely, the governing equation for $\mathscr{P}_2$ is the linear elasticity equation

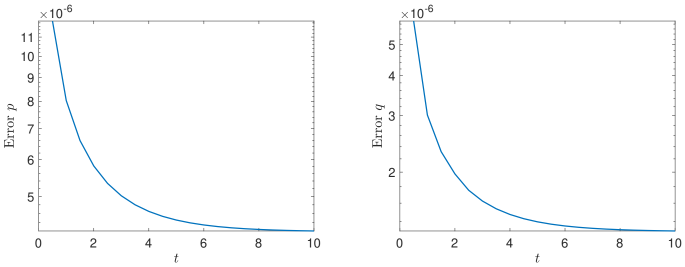
$$\rho^s \frac{\partial^2 \boldsymbol{\eta}}{\partial t^2} - \nabla \cdot \boldsymbol{\sigma}^s(\boldsymbol{\eta}) = 0, \quad \text{in } \Omega_2, \tag{5.25}$$

where

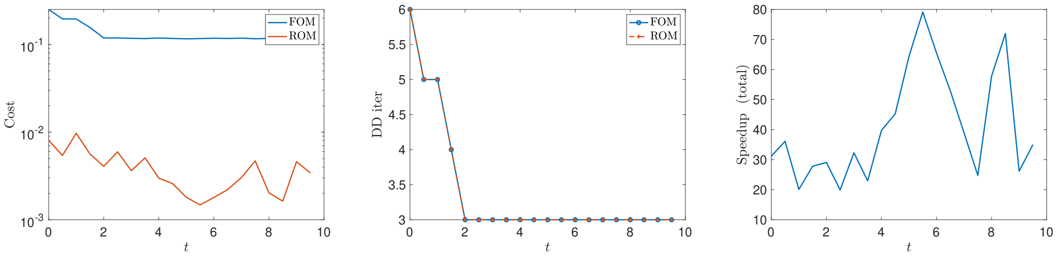
$$\boldsymbol{\sigma}^s(\boldsymbol{\eta}) = \frac{E\nu}{(1+\nu)(1-2\nu)} \left(\nabla \cdot \boldsymbol{\eta}\right) \boldsymbol{I} + \frac{E}{2(1+\nu)} \left(\nabla \boldsymbol{\eta} + \nabla \boldsymbol{\eta}^T\right)$$

is the elastic stress tensor. The physical parameters, ρ^s, E and ν represent the density of the solid, the Young's modulus and the Poisson's ratio, respectively. For this test we have used [2] $\rho_s = 1.0$, $E = 10^4$, $\nu = 0.48$. Equation (5.25) is split into two first-order systems in time by defining

$$\boldsymbol{\xi} = \frac{\partial \boldsymbol{\eta}}{\partial t}.$$



(a) Relative error (p). (b) Relative error (q).



(c) Computational cost. (d) Iterations. (e) Speedup.

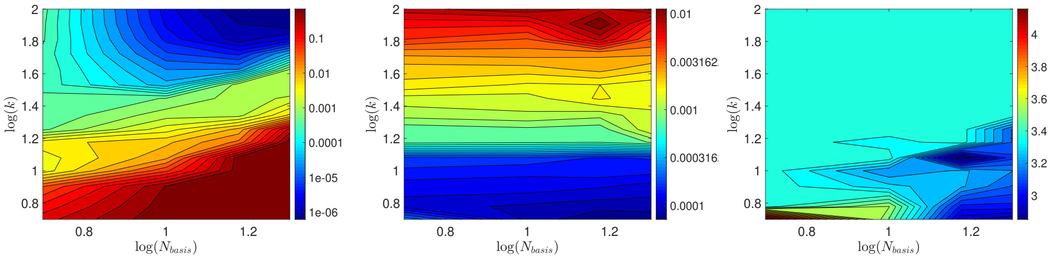
Figure 5.21 – Quantities of interest of the Stokes-Darcy problem.

The fluid problem is completed by no-slip conditions at the top and bottom boundaries $([0,1.2]\times\{0,0.5\})\setminus([0.3,0.5]\times\{0\})$, a constant velocity profile $\boldsymbol{u}=(10y(0.5-y),0)^T$ at the inlet boundary $\{0\}\times[0,0.5]$, and no-stress at the outlet $\{1.2\}\times[0,0.5]$. For the elastic problem we assume homogeneous Dirichlet at the bottom boundary $[0.3,0.5]\times\{0\}$. The initial conditions are set to zero for both problems. At the interface we consider

$$\boldsymbol{u}=\boldsymbol{\xi},\quad \boldsymbol{\sigma}^s(\boldsymbol{\eta})\boldsymbol{n}_s=-\boldsymbol{\sigma}^f\left(\boldsymbol{u},p\right)\boldsymbol{n}_f,$$

corresponding to a Dirichlet-Neumann coupling. As in the Stokes-Darcy problem, they can be generalized to Robin conditions, but for the present problem this is not necessary. Thus, we maintain this formulation, parametrizing the Dirichlet and Neumann data respectively. We consider a number of basis $N_{basis}=10$ and a final time $T=0.5$. The discretization of the fluid, respectively elastic, problem is done using $\mathbb{P}_2$-$\mathbb{P}_1$, respectively $\mathbb{P}_2$, finite elements in space based on the mesh reported in Figure 5.23, and implicit Euler in time with $\Delta t=0.02$. The reduced dimension is set to $k=80$ for each problem and variable, and the velocity space is enriched with 80 supremizer modes.

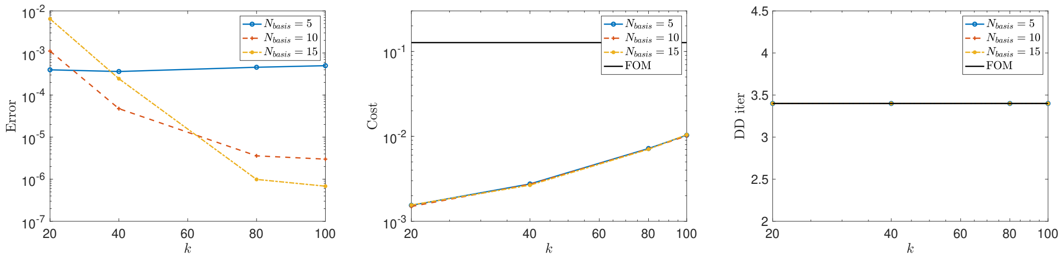
The singular values of the local problems are reported in Figure 5.24 for each variable. The decay for the fluid problem, although exponential, is slower when compared to the previous test case. This is not surprising, as the current problem exhibits more complex features. In the elastic problem, the differences among the x- and y- component are minor, at least in terms of singular values decay. The solution of the coupled problem and the corresponding reduction error at the final time are reported in Figure 5.25 and Figure 5.26 for the fluid and elastic subdomain, respectively. All the solution components are accurately reproduced at a



(a) Relative error (**u**).

(b) Computational cost.

(c) Iterations.

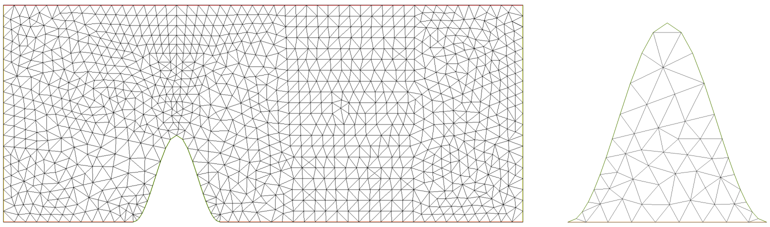


(d) Relative error (**u**).

(e) Computational cost.

(f) Iterations.

Figure 5.22 – Comparison (number of basis functions and reduced dimension) of the time-averaged quantities of interest of the Stokes-Darcy problem.



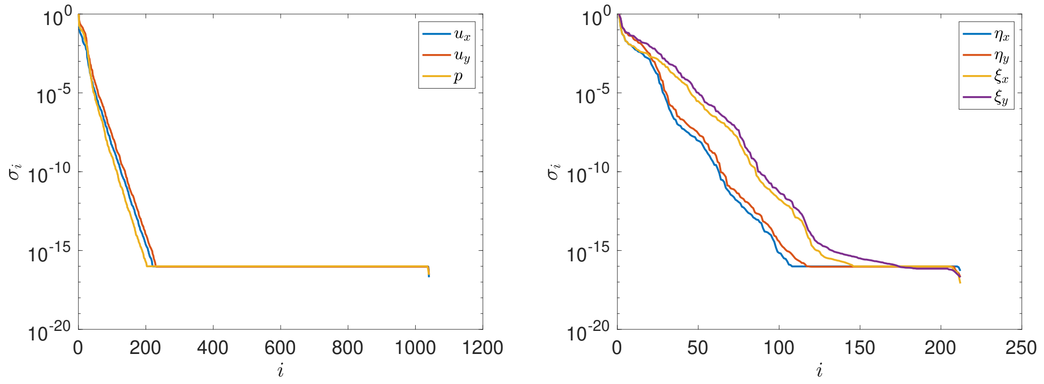
(a) Fluid.

(b) Structure.

Figure 5.23 – Physical domain and computational mesh of the fluid-structure interaction problem.

reduced level. Relevant quantities of interest are shown in Figure 5.27 as a function of time, for which most of the comments made for the previous test cases remain valid. The complexity of the problem, especially for the fluid component, leads to a lower efficiency of the reduced model in comparison to the Stokes-Darcy problem, at least in terms of relative error.

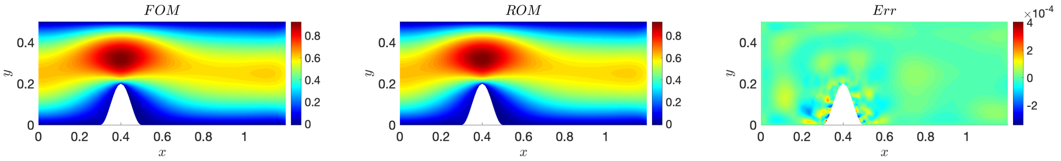
Concerning variations in the hyperparameters of the reduced model, Figure 5.28 reports the relevant quantities of interest for different values of k and N_{basis}. The results are consistent with the previous test cases, although the error decay is negatively impacted by the high degree of heterogeneity and complexity of the problem.



(a) $\mathscr{P}_1$.

(b) $\mathscr{P}_2$.

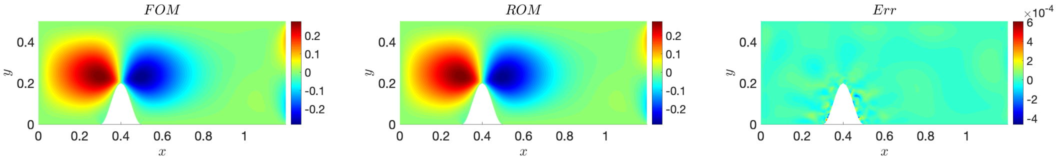
Figure 5.24 – Singular values of the fluid-structure interaction problem.



(a) $\mathscr{P}_1$, u_x, FOM.

(b) $\mathscr{P}_1$, u_x, ROM.

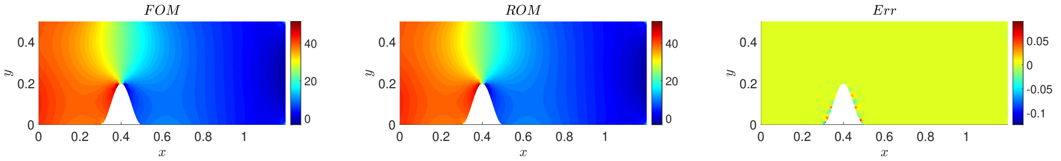
(c) $\mathscr{P}_1$, u_x, error.



(d) $\mathscr{P}_1$, u_y, FOM.

(e) $\mathscr{P}_1$, u_y, ROM.

(f) $\mathscr{P}_1$, u_y, error.



(g) $\mathscr{P}_1$, p, FOM.

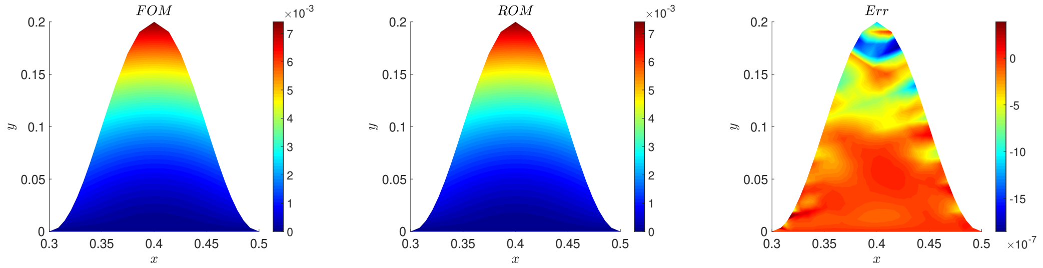
(h) $\mathscr{P}_1$, p, ROM.

(i) $\mathscr{P}_1$, p, error.

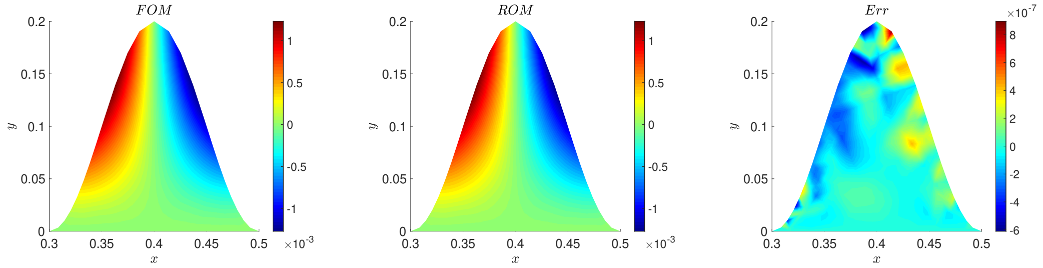
Figure 5.25 – Numerical solutions and reduction error (fluid) of the fluid-structure interaction problem.

5.3.7 Fluid-Structure Interaction II

With the aim of further increasing the complexity of the problem and reducing the simplifying assumptions on the model and the discretization, we present a final test case in the fluid-structure interaction framework. Indeed, the movement of the structure modifies the fluid domain, which should be appropriately taken into account at a numerical level. This can be done by means of the Arbitrary Lagrangian Eulerian (ALE) formulation, which introduces the mesh displacement in the fluid domain as an additional variable. Performing a pull–back of



(a) $\mathscr{P}_2$, η_x, FOM.
(b) $\mathscr{P}_2$, η_x, ROM.
(c) $\mathscr{P}_2$, η_x, error.



(d) $\mathscr{P}_2$, η_y, FOM.
(e) $\mathscr{P}_2$, η_y, ROM.
(f) $\mathscr{P}_2$, η_y, error.

Figure 5.26 – Numerical solutions and reduction error (structure) of the fluid-structure interaction problem.

the fluid equations onto the fluid reference configuration allows us to re-cast the problem in our framework [93]. The test case we consider is the flow past two leaflets located in symmetric positions in a confined channel, inspired by [93], and is characterized by three subproblems. The geometrical setup is shown in Figure 5.29, in which the fluid domain has a dimension of 10×2.5 and each leaflet is 0.2×1.1 large and is located at $x = 1$ downstream. The governing equations for the fluid problem $\mathscr{P}_1$ are the Navier-Stokes equations coupled with the mesh displacement equation

$$\begin{cases} -\Delta \boldsymbol{d}_f & = 0, \quad \text{in } \Omega_1, \\ \rho_f J \frac{\partial \boldsymbol{u}_f}{\partial t} + \rho_f J \nabla \boldsymbol{u}_f \boldsymbol{F}^{-1} \left(\boldsymbol{u}_f - \frac{\partial \boldsymbol{d}_f}{\partial t} \right) - \nabla \cdot \left(J \boldsymbol{\sigma}^f (\boldsymbol{u}_f, p_f) \boldsymbol{F}^{-T} \right) & = 0, \quad \text{in } \Omega_1, \\ \nabla \cdot \left(J \boldsymbol{F}^{-1} \boldsymbol{u}_f \right) & = 0, \quad \text{in } \Omega_1, \end{cases} \tag{5.26}$$

where $\boldsymbol{u}_f$, p_f and $\boldsymbol{d}_f$ are the fluid velocity, pressure and the mesh displacement, respectively. Moreover,

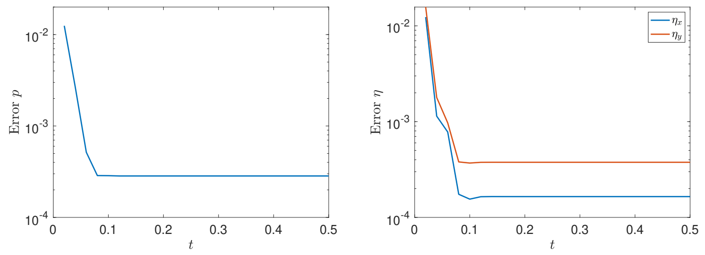
$$\boldsymbol{\sigma}^f \left(\boldsymbol{u}_f, p_f \right) = \mu^f \left(\nabla \boldsymbol{u}_f \boldsymbol{F}^{-1} + \left(\nabla \boldsymbol{u}_f \boldsymbol{F}^{-1} \right)^T \right) - p_f \boldsymbol{I}$$

is the Cauchy stress tensor, $\boldsymbol{F} = \boldsymbol{I} + \nabla \boldsymbol{d}_f$ is the gradient of the ALE map, $\boldsymbol{I}$ is the identity tensor and $J = \det \boldsymbol{F}$. Conversely, the governing equations for $\mathscr{P}_2$ and $\mathscr{P}_3$ are the linear elasticity equation

$$\rho^s \frac{\partial^2 \boldsymbol{\eta}}{\partial t^2} - \nabla \cdot \boldsymbol{\sigma}^s (\boldsymbol{\eta}) = 0, \quad \text{in } \Omega_2,\ \Omega_3, \tag{5.27}$$

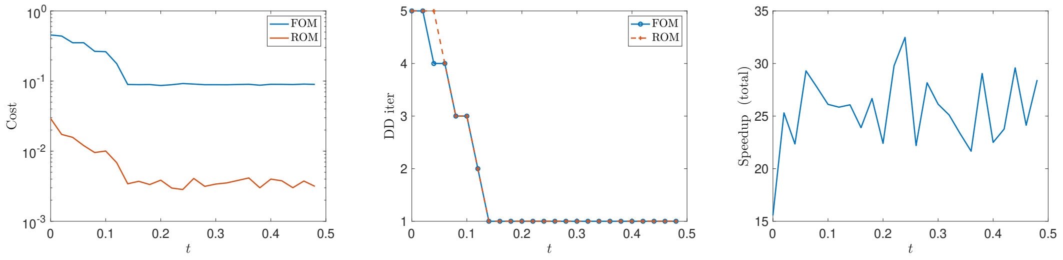
where

$$\boldsymbol{\sigma}^s (\boldsymbol{\eta}) = \frac{E \nu}{(1+\nu)(1-2\nu)} \left(\nabla \cdot \boldsymbol{\eta} \right) \boldsymbol{I} + \frac{E}{2(1+\nu)} \left(\nabla \boldsymbol{\eta} + \nabla \boldsymbol{\eta}^T \right)$$



(a) Relative error (p).

(b) Relative error (η).

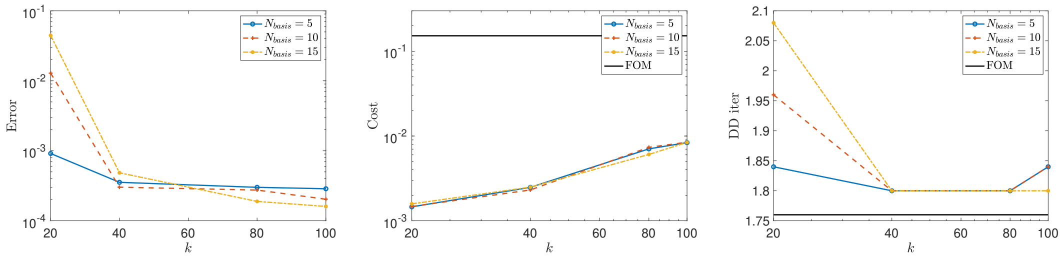


(c) Computational cost.

(d) Iterations.

(e) Speedup.

Figure 5.27 – Quantities of interest of the fluid-structure interaction problem.



(a) Relative error ($\mathbf{u}$).

(b) Computational cost.

(c) Iterations.

Figure 5.28 – Comparison (number of basis functions and reduced dimension) of the time-averaged quantities of interest of the fluid-structure interaction problem.

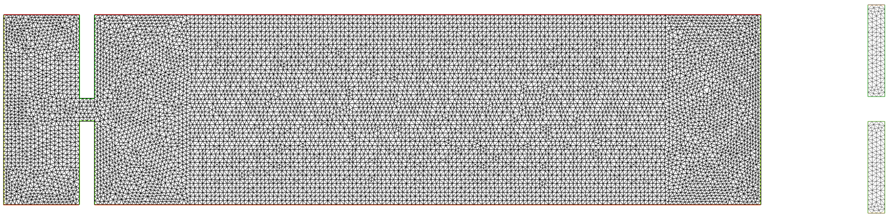
is the elastic stress tensor. The physical variables are the fluid density $\rho_f = 1.0$ and viscosity $\mu_f = 0.035$, the solid density $\rho_s = 1.1$ and the Lamé parameters $\mu_s = \frac{E}{2(1+\nu)} = 10^5$ and $\lambda_s = \frac{E\nu}{(1+\nu)(1-2\nu)} = 8 \cdot 10^5$. As before, (5.27) is split into two first-order systems in time by defining

$$\boldsymbol{\xi} = \frac{\partial \boldsymbol{\eta}}{\partial t}.$$

The fluid problem is completed by no-slip conditions at the top and bottom boundaries, a constant pressure $p = 200$ at the inlet boundary and no-stress at the outlet. For the elastic problems we assume homogeneous Dirichlet at the corresponding physical boundaries. The initial conditions are set to zero for both problems. At the interfaces $\mathscr{P}_1$-$\mathscr{P}_2$ and $\mathscr{P}_1$-$\mathscr{P}_3$ we consider

$$\boldsymbol{u} = \boldsymbol{\xi}, \quad \boldsymbol{d}_f = \boldsymbol{\eta}, \quad \boldsymbol{\sigma}^s(\boldsymbol{\eta})\boldsymbol{n}_s = -J\boldsymbol{\sigma}^f\left(\boldsymbol{u}_f, p_f\right)\boldsymbol{F}^{-T}\boldsymbol{n}_f,$$

corresponding to a Dirichlet-Neumann coupling. We consider a number of basis $N_{basis} = 5$



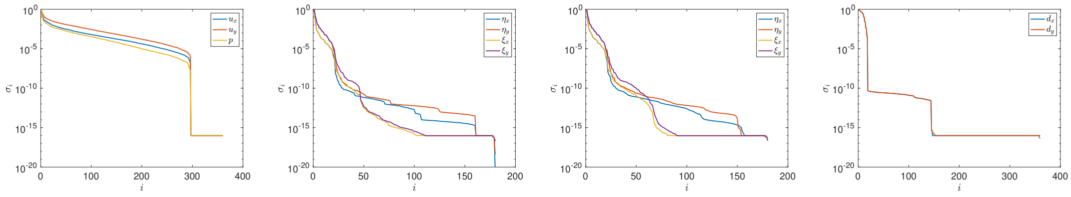
(a) Fluid.

(b) Leaflets.

Figure 5.29 – Physical domain and computational mesh of the fluid-structure interaction problem with ALE formulation.

for each variable and a final time $T = 0.4$. The discretization of the fluid, elastic, and geometric problem is done using $\mathbb{P}_2$-$\mathbb{P}_1$, $\mathbb{P}_2$ and $\mathbb{P}_2$ finite elements in space based on the mesh reported in Figure 5.29. Concerning the time discretization, in order to avoid solving a nonlinear system at each iteration, we use the semi-implicit scheme proposed in [7] with a time step $\Delta t = 0.05$. For illustration purposes, we set the reduced dimension of the fluid problem to $k = 180$ for each variable, with the velocity space enriched with an equal number of supremizer modes, while the reduced dimension of the elastic and the geometric problem are set to $k = 40$ and $k = 10$, respectively.

The singular values of the local problems are reported in Figure 5.30 for each variable. The decay for the fluid problem is quite slow, consistent with the increasing complexity of the fluid problem. This is enhanced by the nonlinear convective term in (5.26). Conversely, due to the diffusion-dominated nature of the elastic and the geometric problem, high reduction potential is expected for both. The solution of the coupled problem and the corresponding reduction error at the final time are reported in Figure 5.31 and Figure 5.32 for the fluid and elastic subdomain, respectively. The largest errors are observed around the interfaces and close to the bottleneck, but with the reduced model we are still able to recover the main components of the dynamics. Concerning variations in the hyperparameters of the reduced model, Figure 5.34 reports the relevant quantities of interest for different reduced dimensions. This test case being mostly a proof of concept, we did not make any effort in implementing a hyper-reduction method, and we simply focus on the accuracy and the domain decomposition iterations. Moreover, as the fluid problem is the most challenging component, we consider



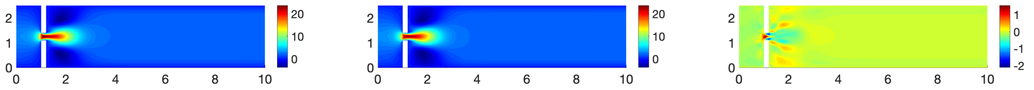
(a) $\mathscr{P}_1$.

(b) $\mathscr{P}_2$.

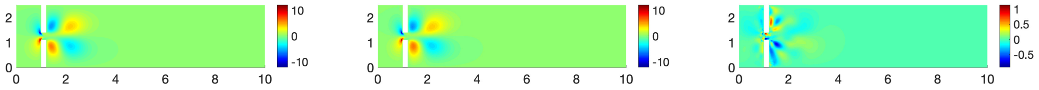
(c) $\mathscr{P}_3$.

(d) Mesh displacement.

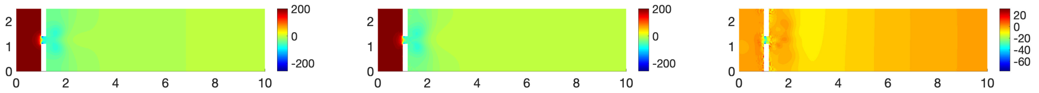
Figure 5.30 – Singular values of the fluid-structure interaction problem with ALE formulation.



(a) $\mathscr{P}_1$, u_x, FOM. (b) $\mathscr{P}_1$, u_x, ROM. (c) $\mathscr{P}_1$, u_x, error.

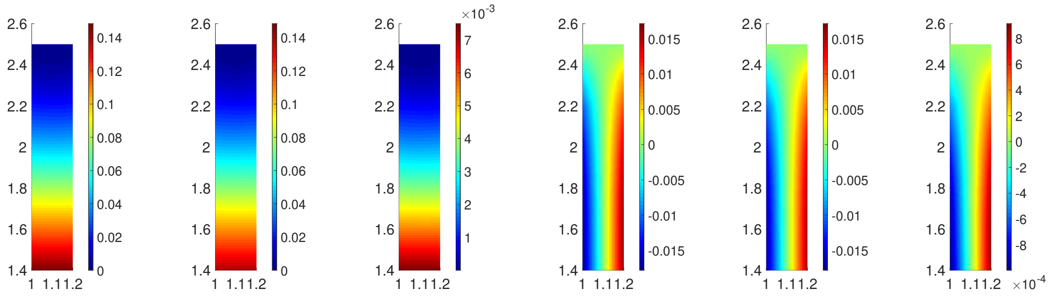


(d) $\mathscr{P}_1$, u_y, FOM. (e) $\mathscr{P}_1$, u_y, ROM. (f) $\mathscr{P}_1$, u_y, error.

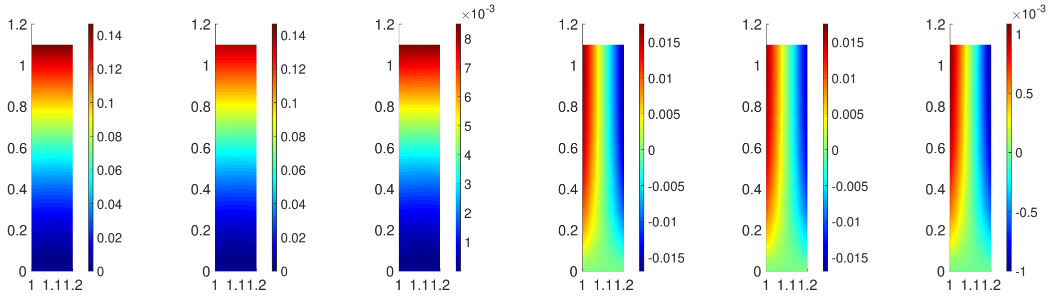


(g) $\mathscr{P}_1$, p, FOM. (h) $\mathscr{P}_1$, p, ROM. (i) $\mathscr{P}_1$, p, error.

Figure 5.31 – Numerical solutions and reduction error (fluid) of the fluid-structure interaction problem with ALE formulation.



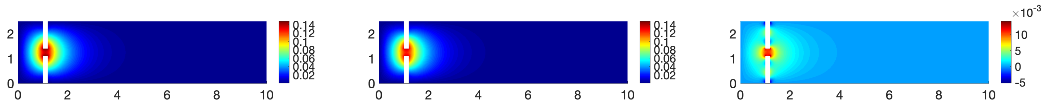
(a) η_x, FOM. (b) η_x, ROM. (c) η_x, error. (d) η_y, FOM. (e) η_y, ROM. (f) η_y, error.



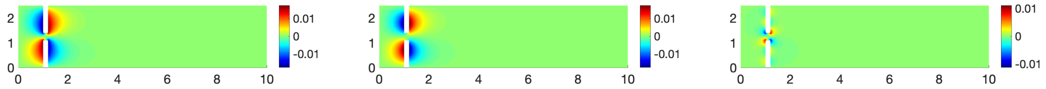
(g) η_x, FOM. (h) η_x, ROM. (i) η_x, error. (j) η_y, FOM. (k) η_y, ROM. (l) η_y, error.

Figure 5.32 – Numerical solutions and reduction error (leaflets) of the fluid-structure interaction problem. Top: second leaflet ($\mathscr{P}_3$). Bottom: first leaflet ($\mathscr{P}_2$).

variations in the reduced dimension of this problem only. The error magnitude is larger compared to the previous test cases, but increasing the size of the reduced problem appears to have a beneficial effect for all the variables.

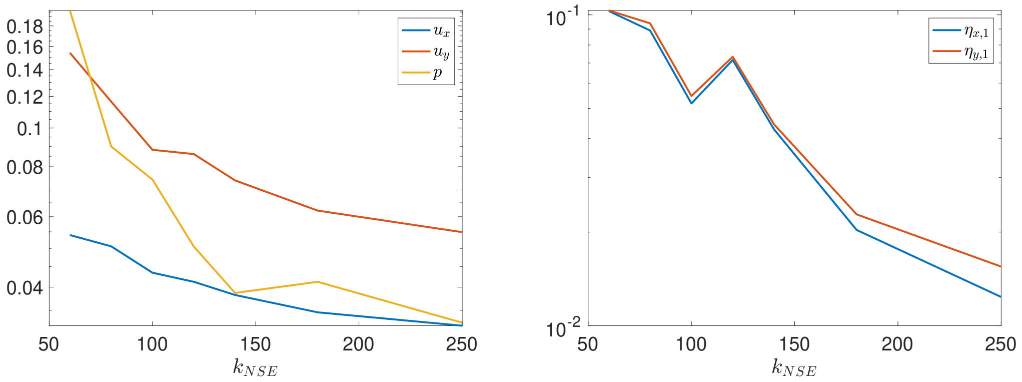


(a) d_x, FOM. (b) d_x, ROM. (c) d_x, error.

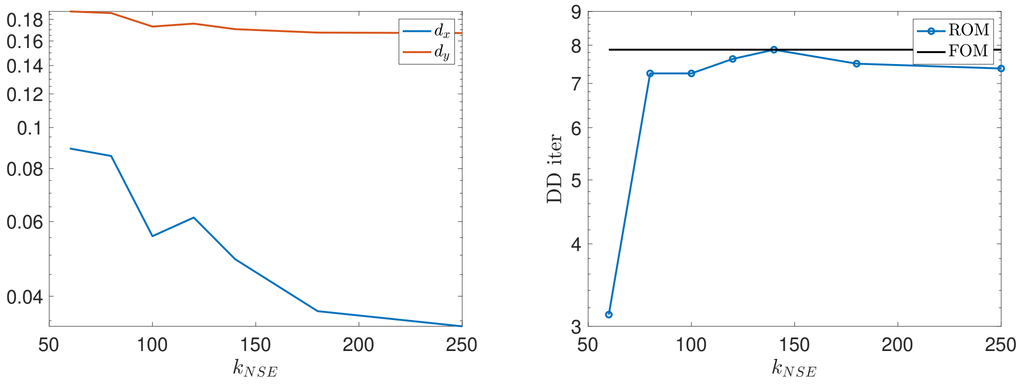


(d) d_y, FOM. (e) d_y, ROM. (f) d_y, error.

Figure 5.33 – Numerical solutions and reduction error (geometry) of the fluid-structure interaction problem with ALE formulation.



(a) Relative error (fluid). (b) Relative error (first leaflet).



(c) Relative error (geometry). (d) Iterations.

Figure 5.34 – Comparison (number of basis functions) of the time-averaged quantities of interest of the fluid-structure interaction problem with ALE formulation.

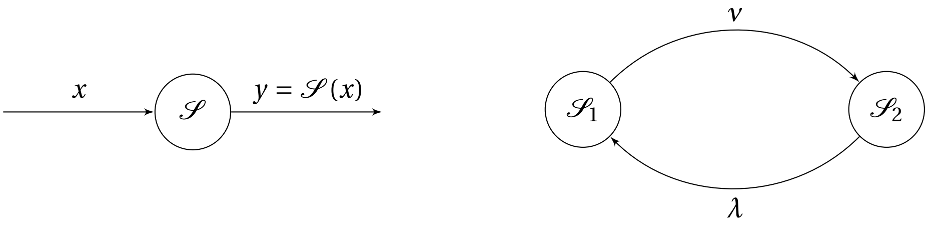
5.4 Conclusion

In this chapter, we proposed a reduced basis method for coupled heterogeneous systems. Our approach requires no offline simulations of the coupled problem, as it relies on local solvers only. The modifications to the local models are minimal, making the approach straightforward and easily generalizable to other problems of interest. The offline phase introduces an artificial parametrization of the boundary conditions at the interface, which is used to collect snapshots of a given subsystem. The online phase couples the reduced models using a domain decomposition strategy inherited from the full order problem.

We showed that the solution of a given coupled, possibly multi-physics, problem can be efficiently approximated by the reduced model. The reduction error depends on both the number of boundary basis functions used in the offline phase and the reduced dimension, and a *sweet spot* clearly appears. The computational cost is mainly influenced by the dimensionality reduction factor, as the number of domain decomposition iterations is only mildly affected by the reduction. Finally, the reduced models are found to be robust with respect to changes in the coupling functions and parameters.

6 Non-intrusive approximations of the boundary response maps

This chapter targets differential models similar to Chapter 5. We again assume that only local solvers are available, meaning that each component possesses its own numerical implementation. This is often the case in multi-physics problems, as discussed in Chapter 1. Additionally, we assume that we not have access to the discretization operators and software implementation, i.e., the local models are given as black boxes only. Each component can therefore be thought as an input-output map, whereas the coupled system consists of the interaction among them, as in Figure 6.1. Although our approach can potentially be applied to any coupled system of this form, in the spirit of Chapter 5, we restrict ourselves to the PDE case, meaning that each component is modeled through a differential equation. Although the final goal would be to construct reduced models that are local in both the offline and the online phase as in Chapter 5, we consider two different training strategies, depending on whether the coupled model, i.e., the combination of the local solvers, is used or not.



(a) A single component.

(b) The coupled system.

Figure 6.1 – Coupled system modeled with black boxes.

This non-intrusive formulation requires recasting the domain decomposition algorithm introduced in Chapter 5 in an operator form, modeling the boundary response of the system [109]. Among other advantages, this formulation also incorporates the cases in which one is interested in approximating the effect of a system onto another without particular reasons to compute the solution [2], or when only boundary functionals can be accessed [111]. Then, after collecting input-output pairs of the relevant operators, the reduced models can be constructed and combined as described in Chapter 2 and Chapter 5.

In the intrusive framework, a variety of surrogate modeling techniques to construct local reduced models, at least in the online phase, has already been proposed. If the coupled problem is known, the problem boils down to reduced basis methods applied to multi-physics [78, 93] or multi-component contexts [76, 144, 145, 9]. If unknown, one could apply the ideas from Chapter 5, to which we refer for a detailed discussion. We take the main ideas from those methods, but we wish to construct purely data-driven variants.

In the context of non-intrusive reduced models, several methods have been developed [143]. The most common examples include using Radial Basis Function interpolation [138], artificial neural networks [71, 28, 54, 29], Gaussian Process Regression [65], operator inference [99] and Dynamic Mode Decomposition [117], but many other alternatives exist. The learning process is often split into a dimensionality reduction phase, which can be done using linear or nonlinear methods, and the actual function approximation problem, similar to Chapter 2. Depending on the specific application, the goal is to construct data-driven approximations of the solution or the underlying differential operators. Such methods are usually easy to design and they have proven their efficiency and robustness in a large number of complex problems. However, they are not local, as their input typically includes the set of parameters of the coupled problem, and they are not explicitly targeted to approximate the operators arising from domain decomposition methods.

An alternative approach [2] constructs an approximation of the boundary response map, also referred to as the *Poincaré-Steklov* operator, by simulating a local problem with a specific set of input data. This method has been successfully applied in a number of multi-physics problems, and can be viewed as a simpler version of the one that we present here. Indeed, this method considers reduction of one subdomain only, in which the underlying differential problem is linear, steady and non-parametrized, thereby significantly restricting its range of applicability. A recent work [27] constructs an approximation of the local system response through artificial neural networks. This method shares many features with our approach, notably on the local and non-intrusive nature of the reduction technique. However, the use of overlapping domain decomposition requires a careful definition of the system response maps, for which their boundary nature is partially lost, and an a-priori knowledge of the size of the overlap. Although it targets a complex multiscale problem, no physical parameters or multi-physics systems are considered, which is in turn the focus of this work.

Thus, our method can be viewed as a combination of these classes of reduced models, as it is local, data-driven and in principle agnostic to the nature of the underlying equations. Notably, it can handle cases in which the coupled problem is not available in the offline phase, does not require any knowledge of the discretization operators, and relies on the boundary response of each subsystem.

The remainder of this chapter is structured as follows. In Section 6.1 we introduce the problem of interest and its solution algorithm. Then, Section 6.2 presents our method, highlighting its features in the offline and the online phase, an a-priori error estimate, and a discussion on the computational cost. Section 6.3 presents different numerical results, showing the efficiency of our method in a number of heterogeneous multi-physics problems. A few concluding remarks are found in Section 6.4.

6.1 Problem formulation

We introduce a general mathematical formulation of coupled heterogeneous systems [61], leaving the discussion on specific instances to Section 6.3. Although the focus is on parametrized problems, at this stage we omit the parameter dependence of the relevant operators to avoid complicating the notation. Moreover, as the majority of this chapter is devoted to steady problems, we restrict here to this framework. Similarly, we consider two components only, defined on as many subdomains. Appropriate comments concerning the extension to such cases can be found at the end of this section.

Let Ω be a domain, and let $\partial\Omega$ be its boundary. Moreover, let Ω_1, Ω_2 be two non-overlapping subdomains such that $\Omega = \Omega_1 \cup \Omega_2$. The interface between the two is denoted by Γ. We are interested in coupled systems of the form

$$\begin{aligned} \mathscr{L}_i(u_i) &= f_i, \quad \text{in } \Omega_i, && (6.1a)\\ h_i(u_i) &= 0, \quad \text{on } \partial\Omega_i \setminus \Gamma, \\ \Phi_1(u_1) &= \Phi_2(u_2), \quad \text{on } \Gamma, \\ \Psi_2(u_2) + \Psi_1(u_1) &= 0, \quad \text{on } \Gamma, \end{aligned}$$

for $i = 1, 2$, where $\mathscr{L}_i$ is a generic second order differential operator modeling the dynamics of problem $\mathscr{P}_i$ and f_i is a given source term. Moreover, the boundary conditions on the physical domain boundaries are encoded by the operators $h_i(u_i)$, whereas the interactions between the components are modeled through suitable functions Φ_i and Ψ_i [61]. Although the specific form of these functions is problem-dependent, they are typically designed to enforce compatibility constraints across the domains, including continuity of the solution and the normal fluxes or stresses. Therefore, the functions Φ_i model Dirichlet-like conditions, that can be typically enforced by choosing appropriate functional spaces, whereas the functions Ψ_i model Neumann-like conditions, that enter directly in the weak formulation of the problems. More general coupling terms can be considered, including Robin-like conditions as in Chapter 5, but the general form (6.1) remains unchanged.

Remark 6.1 *The formulation considered in* (6.1) *is equivalent to the one introduced in Chapter 5, i.e.,* (5.1)*, up to a change in the notation. This helps to highlight the different nature of the Dirichlet-like and Neumann-like conditions, rather than the similar structure of the local problems and the artificial parametrization as was done in Chapter 5.*

As we aim to compute the solution of (6.1) using local solvers only, the non-overlapping Schwarz iterative algorithm discussed in Chapter 5 can still be used. This allows us to decouple the local problems, for which ad-hoc independent models are available, that can in turn be replaced by appropriate surrogates. The idea is to use fixed-point iterations, solving the local problems $\mathscr{P}_i$ with educated guesses on the respective boundary conditions, which are in turn iteratively corrected using the updated local solutions. Denoting such iterations with s, and assuming that a suitable guess u_i^s on the local solutions u_i is given, one can proceed as follows:

1. Solve $\mathscr{L}_1(u_1^{s+1}) = f_1$ with boundary conditions $h_1\left(u_1^{s+1}\right) = 0$ and
$$\Phi_1\left(u_1^{s+1}\right) = \Pi_{12}\Phi_2\left(u_2^s\right).$$
2. Solve $\mathscr{L}_2(u_2^{s+1}) = f_2$ with boundary conditions $h_2\left(u_2^{s+1}\right) = 0$ and
$$\Psi_2\left(u_2^{s+1}\right) + \Pi_{21}\Psi_1\left(u_1^{s+1}\right) = 0.$$
3. Do a relaxation step, i.e., $\Phi_2\left(u_2^{s+1}\right) \leftarrow \omega\Phi_2\left(u_2^{s+1}\right) + (1-\omega)\Phi_2\left(u_2^s\right)$, for some $\omega \in (0,1]$.

These steps are repeated until a suitable convergence criterion is met. Unless stated otherwise, here we adopt
$$\epsilon^{s+1} = \left\|\Phi_2\left(u_2^{s+1}\right) - \Phi_2\left(u_2^s\right)\right\|_\Gamma.$$
Unlike the general formulation (6.1), in Steps 1 and 2 we explicitly introduced the mesh projection operators Π_{ij}. They allow us to exchange interface data from problem $\mathscr{P}_j$ to problem $\mathscr{P}_i$, which can be discretized with two independent meshes that may not feature interface conformity [145]. We note that the use of a Gauss-Seidel algorithm exhibits faster convergence rates and simplifies the description of our method. Additional details on the procedure can be found in Chapter 5.

In this form, the algorithm requires an explicit computation of the local solutions at each iteration. However, only specific boundary functionals are required to propagate information throughout the fixed-point iterations [109, 2]. Therefore, we can introduce the following boundary operators:
$$\mathscr{A} : \Pi_{12}\Phi_2(u_2) \mapsto \Psi_1(u_1), \tag{6.2}$$
and
$$\mathscr{B} : -\Pi_{21}\Psi_1(u_1) \mapsto \Phi_2(u_2). \tag{6.3}$$
They encode the boundary response of each local problem when subject to a given boundary condition. In practice, this stands for: *take an interface datum that depends on the neighboring solution, solve the local problem with the appropriate boundary condition, compute a boundary functional of the solution itself.* As the functions Φ_i and Ψ_i are associated to Dirichlet-like and Neumann-like conditions, the operators (6.2) and (6.3) can be respectively interpreted as the Dirichlet-to-Neumann and Neumann-to-Dirichlet maps, which play a crucial role in domain decomposition methods [37]. They are also referred to as *Poincaré-Steklov* operators, although this terminology has been historically introduced for a Laplace equation only. Critically, (6.2)

and (6.3) are local to a given subdomain, and they can in principle be defined independently of the coupled system. Using them, the iterative Schwarz procedure can be recast, assuming that a suitable guess $\Phi_2(u_2^s)$ is given, as:

1. Compute $\Psi_1(u_1^{s+1}) = \mathscr{A}(\Pi_{12}\Phi_2(u_2^s))$.
2. Compute $\Phi_2(u_2^{s+1}) = \mathscr{B}(-\Pi_{21}\Psi_1(u_1^{s+1}))$.
3. Do a relaxation step, i.e., $\Phi_2\left(u_2^{s+1}\right) \leftarrow \omega\Phi_2\left(u_2^{s+1}\right) + (1-\omega)\Phi_2\left(u_2^s\right)$, for some $\omega \in (0,1]$.

Introducing the variable

$$\lambda = \Phi_2\left(u_2\right), \tag{6.4}$$

the algorithm further simplifies to a single fixed-point equation as

$$\lambda^{s+1} = \omega\mathscr{B}\left(-\Pi_{21}\mathscr{A}\left(\Pi_{12}\lambda^s\right)\right) + (1-\omega)\lambda^s. \tag{6.5}$$

Remark 6.2 *In this work, we always deal with parametrized problems. This implies that all the operators in* (6.1)*, as well as the solutions u_i, are parameter-dependent. Thus, the Dirichlet-to-Neumann and Neumann-to-Dirichlet maps would look like*

$$\mathscr{A} : \left(\Pi_{12}\Phi_2(u_2(\mu_1,\mu_2),\mu_2),\mu_1\right) \mapsto \Psi_1(u_1(\mu_1,\mu_2),\mu_1), \tag{6.6}$$

and

$$\mathscr{B} : \left(-\Pi_{21}\Psi_1(u_1(\mu_1,\mu_2),\mu_1),\mu_2\right) \mapsto \Phi_2(u_2(\mu_1,\mu_2),\mu_2), \tag{6.7}$$

where μ_i is a vector containing the parameters of problem $\mathscr{P}_i$. Therefore, the input of the operators will always include the values of μ_i in addition to the boundary datum. This does not add any significant change to the proposed procedure, which is mainly designed to deal with the input boundary condition, i.e., the first input of (6.6) *and* (6.7)*. This is why we omit this dependence, unless strictly necessary.*

Formally, the algorithm does not involve internal solution values, as the operators are defined on Γ. Thus, if one is interested in retrieving the full solution, an option is to solve the local problems (6.1a) with appropriate boundary conditions once convergence is reached. An alternative consists of constructing a local interpolation method similar to [71], but this is left as an extension of our method, which instead focuses on propagating information across the boundary. However, we mention that in several cases there is no specific need to reconstruct the entire solution as only the effect of a subsystem is necessary [2], and the local solvers might give access to boundary quantities only [111].

Remark 6.3 *We assume that the black-box solvers can be queried to retrieve the output of the operators* (6.2) *and* (6.3) *directly, in the spirit of [111]. If this is not the case, and only the solution* *values can be accessed, a pre-processing step can be added in order to compute the boundary functionals associated to the operators.*

Remark 6.4 *Assuming that the solution values can be accessed, one could construct a reduced model to approximate the local solution at each step of the fixed point method, similar to [71], and reconstruct the boundary functional in a post-processing step to propagate the relevant information. There is nothing that prevents one from doing this, and it would arguably be needed if physics-informed variants of the method have to be designed [28] or if one wishes to reconstruct the solution at convergence without solving an additional problem or constructing another surrogate. However, in its naive version, we conjecture that learning the full solution would lead to less accurate methods, as one would have to approximate the solution in the entire domain instead of a boundary functional only.*

Before concluding this section, we briefly discuss the case of more complex problems. If more than two subdomains are considered, the main ideas carry over, up to a few aspects. Firstly, one needs to take particular care in assuring well-posedness of the local problems and the iterative scheme, which can be at stake if Dirichlet-Neumann iterations are considered. Secondly, the boundary maps are defined on the entire interface of each subproblem, say $\Gamma_i = \bigcup_{j \in \mathscr{I}_i} \Gamma_{ij}$, where $\mathscr{I}_i$ is the index set of the subproblems that share an interface Γ_{ij} with problem $\mathscr{P}_i$. In this regard, combinations of Dirichlet and Neumann data may appear both at the input and the output. Thirdly, an appropriate processing of the elements should be designed if a Gauss-Seidel procedure is used [22]. Extending the method to unsteady cases can potentially be more challenging. An option is to embed the time loop into the fixed point iterations. This implies that at each step of the domain decomposition loop, a parabolic-like problem is solved with appropriate time-dependent boundary conditions, similar to (6.2) and (6.3) [104]. The dual option, consisting of solving an interface equation at each time step, could also be considered [37]. Although the latter appears to be more natural, in view of the reduction procedure proposed in Section 6.2, we stick to the former.

6.2 Non-intrusive model reduction

Now, we present how to construct data-driven approximations of (6.2), (6.3) and ultimately (6.5). We discuss features, training procedures, accuracy and computational complexity.

6.2.1 Approximating boundary maps

As our operator approximation problem falls in the class described in Chapter 2, we refer to that for more details. Here, we highlight a few additional problem-specific aspects.

As discussed in Remark 6.2, the problems are parameter-dependent. For the Dirichlet-to-Neumann map (6.6), this implies that:

- The input consists of instances of the boundary input $\Pi_{12}\Phi_2(u_2)$ and the physical parameter values μ_1. The dimensionality reduction discussed in Chapter 2 is applied only to the former, as the dimensionality of the parameter space is typically small, and adding an additional reduction step might lead to a loss in the physical interpretability of the values of μ_1. The scaling is instead kept as discussed in Chapter 2. We numerically observed that a min-max scaling appears to give the best results.

- The output consists of instances of the boundary output $\Psi_1(u_1)$. The dimensionality reduction and the scaling are carried out as in Chapter 2, and a standard scaler is chosen.

A similar discussion holds for the Neumann-to-Dirichlet map (6.7).

The generation of the training set plays a significant role, and can be done in two alternative ways:

- If the coupled problem is available and its computational cost is not prohibitive, training data can be generated by repeatedly simulating it. After appropriately sampling the physical parameter space, we run the non-overlapping Schwarz method for each parameter value. From the local problems, we extract the Dirichlet- and Neumann-like data, and we add them to the training set.

- If the coupled problem is available but computationally expensive or is not known in the training phase, the local problems need to be considered independently and artificial data need to be constructed, similar to Chapter 5. Focusing on $\mathscr{P}_1$ for simplicity, an option is to solve

$$\begin{aligned} \mathscr{L}_1(u_1) &= f_1, \quad \text{in } \Omega_1, && (6.8a)\\ h_1(u_1) &= 0, \quad \text{on } \partial\Omega_1 \setminus \Gamma, \\ \Phi_1(u_1) &= g, \quad \text{on } \Gamma, && (6.8b) \end{aligned}$$

  for different instances of the datum g. To generate physically relevant boundary conditions, we first expand g as

$$g = \sum_{j=1}^{N_{exp}} \beta_j \varphi_j, \tag{6.9}$$

  where $\{\varphi_j\}_{j=1}^{N_{exp}}$ are boundary modes stemming from the eigenfunctions of the interface Laplace-Beltrami operator

$$-\Delta\varphi_j = \omega_j \varphi_j, \quad \text{on } \Gamma. \tag{6.10}$$

  The boundary conditions of (6.10) are in general problem-dependent, and specific choices will be discussed in Section 6.3. As a general rule, both Dirichlet and Neumann can be used as in Chapter 5, or an appropriate lifting procedure should be designed. However, in a number of cases, including closed interfaces or when the interface shares its boundary with the physical one, more specific choices can be made, with the goal of

keeping the value of N_{exp} controlled.
The sampling of the coefficients β_j in (6.9) plays a key role, as we need to capture a variety of possible inputs. Although many options can be considered [119], a simple choice consists of a (quasi)-random sampling in the N_{exp}-dimensional hypercube $[-A, A]^{N_{exp}}$ for a suitable constant $A > 0$. Additionally, to take into account a certain smoothness of the input data, we re-scale the coefficients to impose a certain decay rate r as the eigenfrequency increases. Both A and r are problem-dependent parameters, but the input and output scaling appears to reduce their impact at a numerical level. For each instance of the coefficients β_j, the local problem is solved for multiple values of the physical parameters, that can be obtained using again a random sampling. The resulting input-output pairs constitute our training set.

Following Chapter 2, we can construct two approximations $\hat{\mathscr{A}}$ and $\hat{\mathscr{B}}$ of the reduced boundary maps such that

$$\mathscr{A}(x) \approx U_y^{\mathscr{A}} \hat{\mathscr{A}}([U_x^{\mathscr{A}}]^T x),$$

and

$$\mathscr{B}(x) \approx U_y^{\mathscr{B}} \hat{\mathscr{B}}([U_x^{\mathscr{B}}]^T x),$$

where $U_y^{\mathscr{A}}, U_x^{\mathscr{A}}, U_y^{\mathscr{B}}, U_x^{\mathscr{B}}$ perform the dimensionality reduction as in (2.24) and (2.25). The fixed-point algorithm becomes then:

1. Compute $\widehat{\Psi_1(u_1^{s+1})} = U_y^{\mathscr{A}} \hat{\mathscr{A}}([U_x^{\mathscr{A}}]^T \Pi_{12} \widehat{\Phi_2(u_2^s)})$.
2. Compute $\widehat{\Phi_2(u_2^{s+1})} = U_y^{\mathscr{B}} \hat{\mathscr{B}}(-[U_x^{\mathscr{B}}]^T \Pi_{21} \widehat{\Psi_1(u_1^{s+1})})$.
3. Do a relaxation step, i.e., $\widehat{\Phi_2\left(u_2^{s+1}\right)} \leftarrow \omega \widehat{\Phi_2\left(u_2^{s+1}\right)} + (1-\omega)\widehat{\Phi_2\left(u_2^s\right)}$, for some $\omega \in (0, 1]$.

Mimicking (6.4) and (6.5), we can write it as

$$\hat{\lambda}^{s+1} = \omega U_y^{\mathscr{B}} \hat{\mathscr{B}}\left(-[U_x^{\mathscr{B}}]^T \Pi_{21} U_y^{\mathscr{A}} \hat{\mathscr{A}}\left([U_x^{\mathscr{A}}]^T \Pi_{12} \hat{\lambda}^s\right)\right) + (1-\omega)\hat{\lambda}^s. \tag{6.11}$$

The proposed methodology can be extended to more complex problems. If more than two subdomains are considered, the method generalizes in a straightforward way. Note that the functions φ_j in (6.9) can be obtained by solving (6.10) at each subinterface. In the case of time-dependent problems, both the input and the output are space-time functions. The dimensionality reduction can be done considering a space-time basis, constructed as the Kronecker product between a spatial and a temporal basis obtained by two SVDs on the appropriately reshaped training data [125]. The space-time analogous of (6.9) is

$$g(\xi, t) = \sum_{j=1}^{N_{exp,\xi}} \sum_{k=1}^{N_{exp,t}} \beta_j \varphi_j(\xi) \ell_k(t), \tag{6.12}$$

where ξ is a curvilinear coordinate defined on the interface, t is time, and $N_{exp,\xi}$, $N_{exp,t}$ denote the number of functions used to expand the datum in space and time, respectively. For the latter, we use a Taylor expansion.

Remark 6.5 *This method is agnostic to the underlying differential problem, and can in principle be applied to any system of interest without major modifications. However, if additional knowledge of the problem is available, other model reduction strategies could also be designed, on the line of operator inference approaches [99]. An example concerns linear problems, in which the boundary operators are affine in the input for each parameter value. Exploiting this, one could solve a least-square problem to retrieve the reduced operators for each physical parameter, and do the interpolation in the physical parameter space only. This has few advantages, including a higher accuracy and a lower computational complexity, but extending it to general nonlinear problems might be prohibitive.*

6.2.2 Error analysis

Qualitatively, the quality of the approximation will depend on both the dimensionality reduction error and the data-driven function approximation error. Here, we make this argument more rigorous. For simplicity, we consider a steady problem with two subdomains only, and we assume that the meshes are conforming at the interface and that no relaxation is necessary for convergence. We denote with (n, m) and (N, M) the input and output dimensions before and after the dimensionality reduction, respectively, while $\mathscr{N}$ is the number of training samples. We add appropriate subscripts when necessary.

We aim at estimating the error between the solution λ of true fixed point mapping

$$\lambda = \mathscr{F}(\lambda) = \mathscr{B}(-\mathscr{A}(\lambda)), \tag{6.13}$$

and its surrogate $\hat{\lambda}$, which solves the reduced problem

$$\hat{\lambda} = \hat{\mathscr{F}}(\hat{\lambda}) = U_y^{\mathscr{B}} \hat{\mathscr{B}}\left(-[U_x^{\mathscr{B}}]^T U_y^{\mathscr{A}} \hat{\mathscr{A}}\left([U_x^{\mathscr{A}}]^T \hat{\lambda}\right)\right),$$

under appropriate regularity assumptions on the true operators $\mathscr{A}$ and $\mathscr{B}$.

Following the derivation proposed in Appendix 6.A, we obtain

$$\left\|\lambda - \hat{\lambda}\right\| \leq \frac{K_1(\texttt{params}, \lambda, \mathscr{A}, \mathscr{B})}{1 - L_{\mathscr{F}} - K_2(\texttt{params}, \lambda, \mathscr{A}, \mathscr{B})}, \tag{6.14}$$

where

$$\texttt{params} = (M_{\mathscr{B}}, N_{\mathscr{B}}, \mathscr{N}_{\mathscr{B}}, M_{\mathscr{A}}, N_{\mathscr{A}}, \mathscr{N}_{\mathscr{A}})$$

is the collection of the model hyperparameters and $L_{\mathscr{F}} < 1$ is the Lipschitz constant of (6.13). The functions K_1 and K_2 are appropriate functions of `params`, the true operators $\mathscr{A}$ and $\mathscr{B}$, and the true solution λ, see (6.24). Note that (6.14) holds when $K_2 < 1 - L_{\mathscr{F}}$, which is the case for sufficiently large values of the model parameters.

6.2.3 Computational cost

We conclude this section with a brief remark on the computational complexity of our method in the online phase. Let κ_i be the number of the degrees of freedom of problem $\mathscr{P}_i$ and let κ_Γ be the number of interface degrees of freedom for both problems. The true fixed-point algorithm has a complexity of

$$\mathscr{C}_{\mathscr{F}} = \mathcal{O}(\kappa_1^\alpha) + \mathcal{O}(\kappa_2^\alpha)$$

at each iteration, up to terms depending on κ_Γ. The exponent α depends on the discretization algorithm at hand, and typically $\alpha = 2$ for a classical finite element discretization. Instead, the approximated mapping costs

$$\mathscr{C}_{\hat{\mathscr{F}}} = \mathcal{O}(\kappa_\Gamma N_{\mathscr{A}} + \mathscr{C}_{\hat{\mathscr{A}}} + \kappa_\Gamma M_{\mathscr{A}}) + \mathcal{O}(\kappa_\Gamma N_{\mathscr{B}} + \mathscr{C}_{\hat{\mathscr{B}}} + \kappa_\Gamma M_{\mathscr{B}}), \tag{6.15}$$

at least in its naive implementation (6.11). The costs of evaluating the approximated mappings are denoted by $\mathscr{C}_{\hat{\mathscr{A}}}$ and $\mathscr{C}_{\hat{\mathscr{B}}}$, respectively. For the VKOGA, we have

$$\mathscr{C}_{\hat{i}} = \mathcal{O}(N_i \mathscr{N}_i + M_i \mathscr{N}_i), \quad i \in \{\mathscr{A}, \mathscr{B}\}, \tag{6.16}$$

whereas for the ANN implementation with $L+1$ layers of size $\{k_i^l\}_{l=0}^{L}$ we have

$$\mathscr{C}_{\hat{i}} = \mathcal{O}\left(\sum_{l=1}^{L_i} k_i^l k_i^{l-1}\right) = \mathcal{O}\left(N_i k_i^1 + \sum_{l=2}^{L_i-1} k_i^l k_i^{l-1} + M_i k_i^{L_i-1}\right), \quad i \in \{\mathscr{A}, \mathscr{B}\}.$$

The dependence of (6.15) on κ_Γ can be eliminated if the mesh projection operators are linear and one can precompute the linear terms in (6.11). To maintain a full independence between the components, we consider its standard version.

6.3 Numerical results

We now present a number of applications, aiming to show the efficiency of our method. Unless stated otherwise, the errors are measured using the relative L^2-norm in the appropriate spatial domain, the tolerance for the Schwarz iterations is $\mathrm{tol} = 10^{-6}$, and a zero initial guess is used for both the full and the reduced model.

6.3.1 Diffusion on a Cartesian geometry

Our first system of interest consists of a parametrized diffusion problem, with possibly different diffusion coefficients between the components. The parameters are designed to control the strength of the dissipation terms, similar to [60]. The main goal of this test case is to validate our technique in a relatively simple framework, mostly dictated by the elliptic and linear nature of the problem. We consider $\Omega = [0,1] \times [0,2]$, divided into two subdomains by a horizontal interface Γ located at $y = 1$. The governing equation for $\mathscr{P}_i$ is

$$-\nabla \cdot \left(\mu_i \nabla u_i\right) = f_i, \quad \text{in } \Omega_i,$$

where $f_i = 20 + 10x - 5y$ and

$$\mu_i = \mu + \sum_{j=1}^{p} w_j^i j^{\sigma} (1 + a_j^i), \quad \text{where } a_j^i = \sin\left(\left\lfloor \frac{j+2}{2} \right\rfloor \pi (x - x_0^i)\right) \sin\left(\left\lceil \frac{j+2}{2} \right\rceil \pi (y - y_0^i)\right).$$

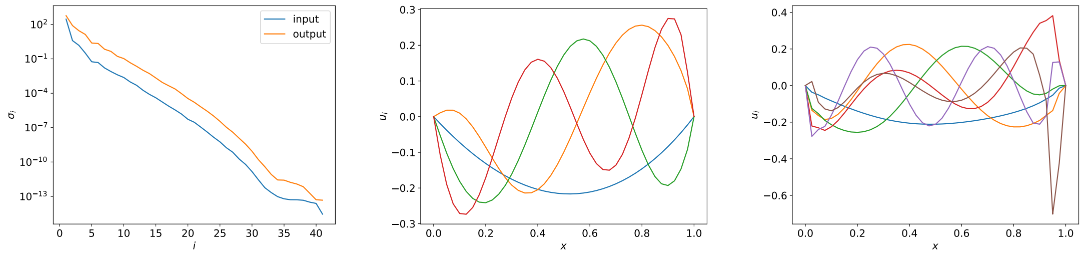
The shift μ and decay σ are constant and equal to 1 and -1, respectively, whereas the number of harmonics p in each subdomain is fixed to a value of 1, 2 or 3, depending on the specific application. The sinusoidal terms are centered at $(x_0^1, y_0^1) = (0,1)$ and $(x_0^2, y_0^2) = (0,0)$, respectively, and they are weighted with problem-dependent parameters $w_j^i \in [0,1]$. At the physical boundaries, we consider homogeneous Dirichlet conditions, whereas the interface coupling conditions are of a standard Dirichlet-Neumann form, i.e.,

$$u_1 = u_2, \quad \text{on } \Gamma,$$
$$\mu_2 \nabla u_2 \cdot \boldsymbol{n}_2 + \mu_1 \nabla u_1 \cdot \boldsymbol{n}_1 = 0, \quad \text{on } \Gamma.$$

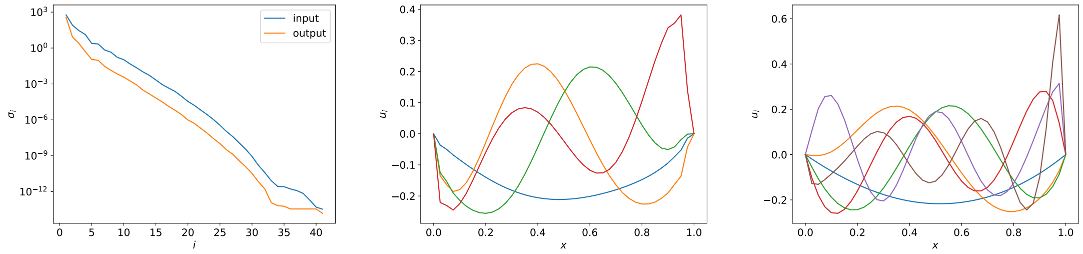
We discretize the problem using $\mathbb{P}_1$ finite elements in both subdomains, defined on two meshes with approximate sizes of $h \approx \sqrt{2}/40$.

Firstly, we consider training data stemming from the coupled problem and we fix the relevant hyperparameters. We consider $p = 2$ and 247 parameter samples generated with a Latin Hypercube Sampling technique. The singular vectors and the first 4 (resp. 6) modes obtained with the SVD on the input (resp. output) data are reported in Figure 6.2. The singular values decay exponentially and the dominant modes encode low frequency components, which indicate an efficient dimensionality reduction. As expected, the differences between the input of $\mathscr{P}_1$ and the output of $\mathscr{P}_2$, as well as between the output of $\mathscr{P}_1$ and the input of $\mathscr{P}_2$, are mild or zero. After selecting $N_{\mathscr{A}} = N_{\mathscr{B}} = 4$ and $M_{\mathscr{A}} = M_{\mathscr{B}} = 6$, we use the VKOGA with a Matérn kernel ($\nu = 2.5$) to construct the interpolator, whose performance is subsequently tested for a parameter value outside the training set. The fixed point iterations ($\omega = 0.5$) give the results reported in Figure 6.3. Visually, the difference between the full (λ) and the reduced ($\hat{\lambda}$) fixed point variable at convergence is mild. The fixed point error slightly deviates from the true line, especially when small values are reached, but convergence is not compromised. By solving the local problems with the converged boundary values, we obtain the solutions shown in Figure 6.4, which confirms the high accuracy of our method and qualitatively validates its efficiency.

We now study the effect of varying hyperparameters on the reduction. Specifically, we are interested in how the number of input and output modes affect the accuracy of the method. To have a more comprehensive analysis, we consider the cases with $p = 1$, $p = 2$ and $p = 3$ parameters per subdomain, with the other hyperparameters unchanged. The results are



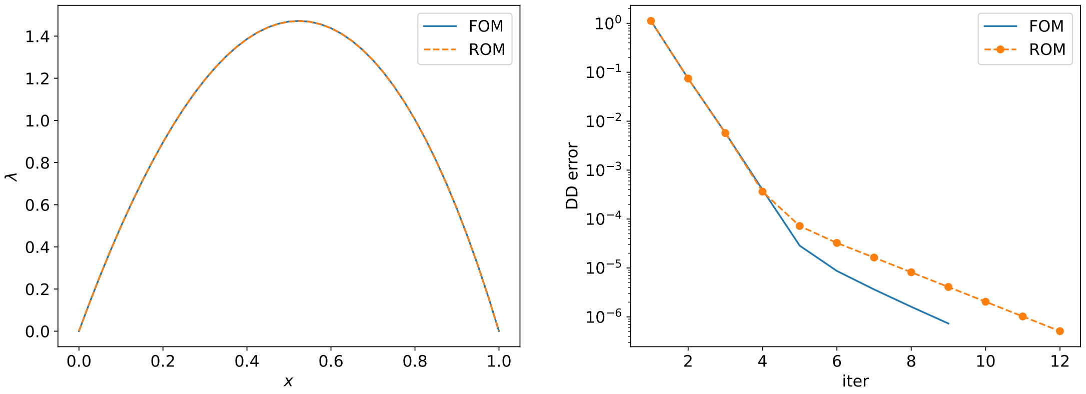
(a) $\mathscr{P}_1$, Singular values.
(b) $\mathscr{P}_1$, input singular vectors.
(c) $\mathscr{P}_1$, output singular vectors.



(d) $\mathscr{P}_2$, Singular values.
(e) $\mathscr{P}_2$, input singular vectors.
(f) $\mathscr{P}_2$, output singular vectors.

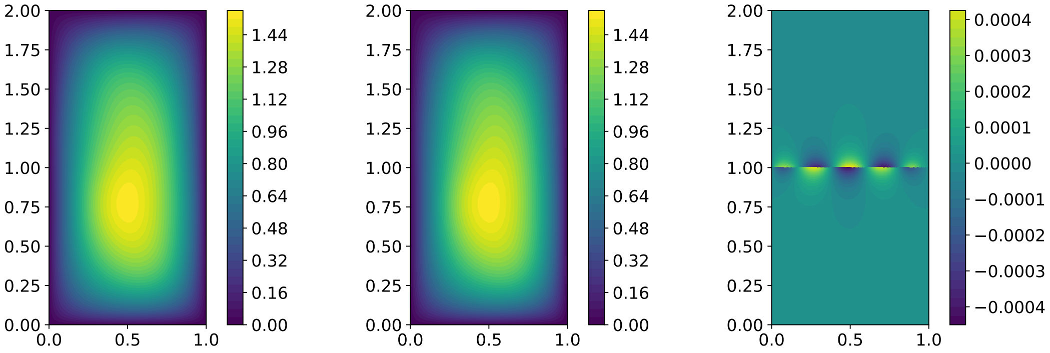
Figure 6.2 – Singular values and singular vectors associated to the diffusion problem, training done with the coupled model.

reported in Figure 6.5. Consistent with the theoretical analysis provided in Section 6.2, an initial increase in the output dimension has a beneficial effect, as the dimensionality reduction error decreases. This holds true until a plateau is reached, and increasing the output dimension has no effect on the accuracy. This saturation level is due to the input projection and interpolation errors. Here, the former dominates, as an increase in the input dimension lowers this threshold, at least until very small error magnitudes are reached. Qualitatively, no significant differences appear if the number of physical parameters is changed. However, the error magnitude is negatively affected by an increase in p, as the larger snapshots variability gives a larger dimensionality reduction error.



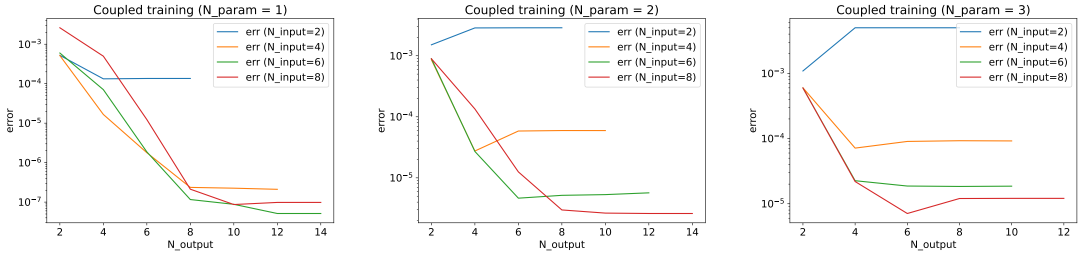
(a) Fixed point variable λ.
(b) Fixed point error.

Figure 6.3 – Results of the diffusion problem, training done with the coupled model.



(a) FOM. (b) ROM. (c) Pointwise error.

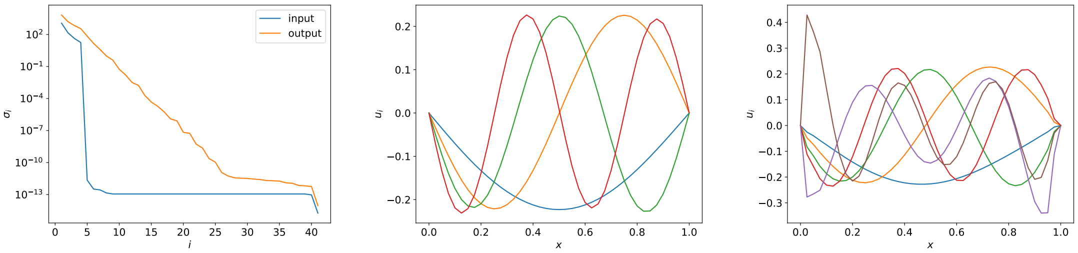
Figure 6.4 – Solution of the diffusion problem, training done with the coupled model.



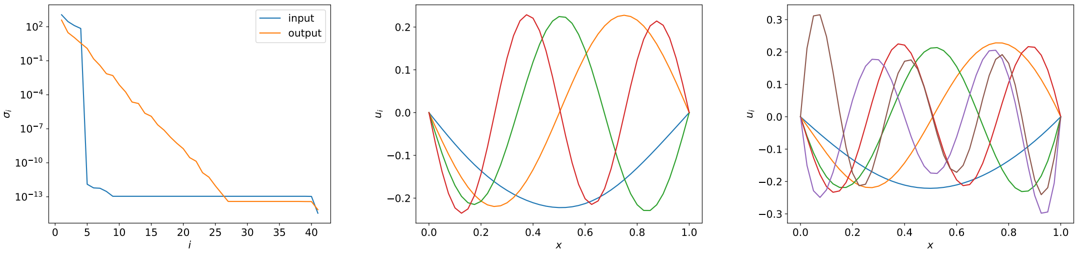
(a) $p = 1$. (b) $p = 2$. (c) $p = 3$.

Figure 6.5 – Reduction errors of the diffusion problem for varying number of SVD modes, training done with the coupled model.

We now repeat the analysis using the alternative training strategy we proposed, i.e., we generate snapshots using the two subproblems separately. All the relevant hyperparameters are kept unchanged, except those related to the training set, which is now generated with 10 samples of the physical parameters and 200 samples of the artificial parameters for each subproblem. The interface Laplace problems are solved analytically with Dirichlet boundary conditions, and only the first $N_{\mathscr{A}} = N_{\mathscr{B}} = 4$ modes are kept. A scaling factor of $A = 10$ and a decay rate of $r = 3$ (resp. $r = 2$) is selected for $\mathscr{P}_1$ (resp. $\mathscr{P}_2$). Looking at the input singular values and modes in Figure 6.6, we observe that only the first $N_{\mathscr{A}}$, $N_{\mathscr{B}}$ have a nonzero value. As the input data are a linear combinations of $N_{\mathscr{A}}$ eigenfunctions, this is not surprising, and the corresponding singular vectors clearly resemble the Laplace modes. This does not hold true for the output, as the varying physical parameters prevent a similar exact reconstruction. However, the decay appears to be exponential, so that one can conjecture that the reduction is not severely compromised despite the different training strategy. This is confirmed by looking at both the fixed point variables and numerical solutions obtained with a VKOGA interpolator, reported in Figure 6.7 and Figure 6.8 respectively. Most of the comments related to the previous analysis remain valid. However, albeit at a qualitative level, the deviations from the full order model appear to be more evident compared to Figure 6.3 and Figure 6.4. The training data are generated independently of the coupled problem, and they need to be robust with respect to any incoming datum. This increased level of robustness negatively affects the accuracy.

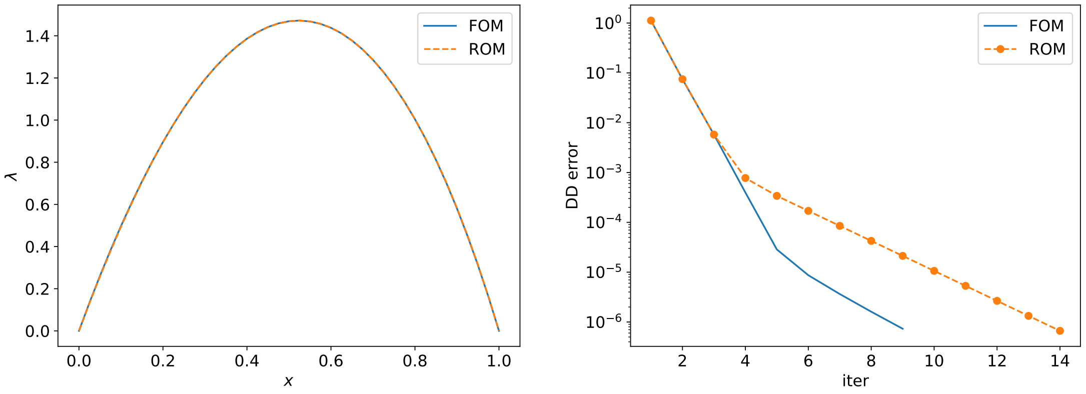


(a) $\mathscr{P}_1$, Singular values. (b) $\mathscr{P}_1$, input singular vectors. (c) $\mathscr{P}_1$, output singular vectors.



(d) $\mathscr{P}_2$, Singular values. (e) $\mathscr{P}_2$, input singular vectors. (f) $\mathscr{P}_2$, output singular vectors.

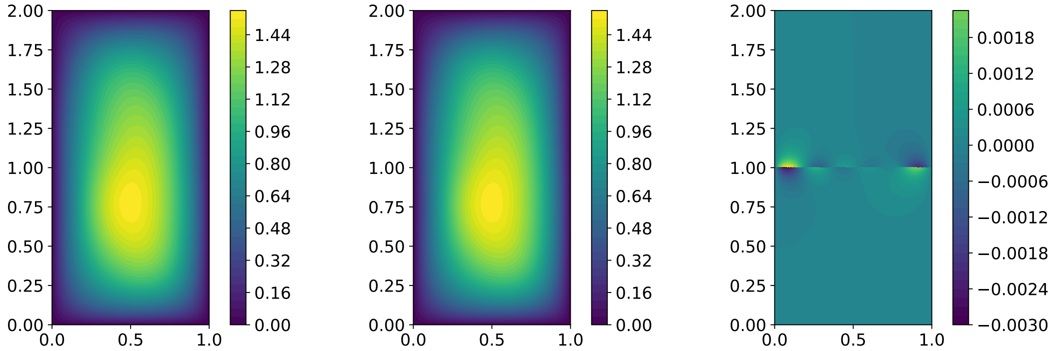
Figure 6.6 – Singular values and singular vectors associated to the diffusion problem, training done with the decoupled model.



(a) Fixed point variable λ. (b) Fixed point error.

Figure 6.7 – Results of the diffusion problem, training done with the decoupled model.

This argument can be made more rigorous by looking at the model accuracy as a function of the hyperparameters, reported in Figure 6.9. Here, we use 200 artificial parameter samples for $N_{\mathscr{A}} \leq 5$ and 800 samples when $N_{\mathscr{A}} > 5$. Again, the total error is affected by the dimensionality reduction and interpolation error, but its magnitude is generally larger when compared to Figure 6.5. Given that there is no problem-dependent bias on the input data, samples need to



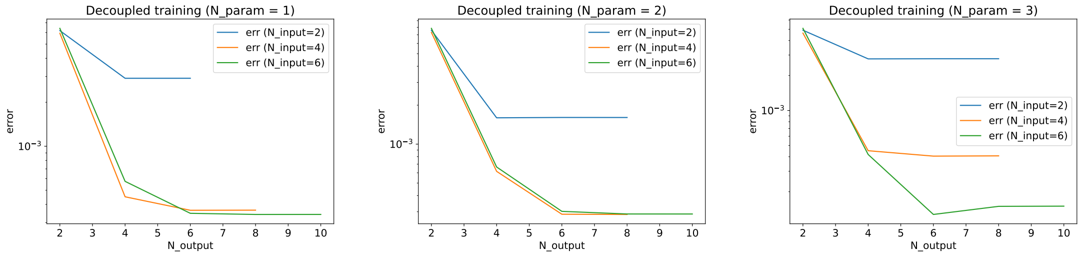
(a) FOM.

(b) ROM.

(c) Pointwise error.

Figure 6.8 – Solution of the diffusion problem, training done with the decoupled model.

be taken in a possibly high-dimensional hypercube. The curse of dimensionality has a clear impact, and it is the main responsible for the suboptimal behavior of the error as the input dimension is increased.



(a) $p = 1$.

(b) $p = 2$.

(c) $p = 3$.

Figure 6.9 – Reduction errors of the diffusion problem for varying number of SVD modes, training done with the decoupled model.

6.3.2 Diffusion on a circular geometry

Our second problem, adapted from [145], has a similar physical nature to the previous one, as it still consists of a parametrized diffusion equation with a discontinuous viscosity profile. However, it is defined on a circular domain and features a circular interface, and the resulting meshes possess an interface nonconformity. Thus, this test case has the main objective to show the potential of our technique in a more geometrically challenging setting. We take Ω to be the circle of radius $R_{\text{ext}} = 2$, which is separated into an internal circular and an external annular region, denoted by Ω_1 and Ω_2 respectively, by a circular interface at $R_{\text{int}} = 1$. The governing equation for $\mathscr{P}_i$ is

$$-\nabla \cdot \left(\mu_i \nabla u_i\right) = f_i, \quad \text{in } \Omega_i,$$

where $f_i = 10x(\exp(y) - 1.0)\sin(\pi y)$ and

$$\mu_i = \mu + 5A_i x^2,$$

in which $\mu = 5$, and A_i are parameters chosen in the interval $[0,1]$. Only the external problem has a physical boundary, on which we set a Dirichlet condition $u_2 = 0.1$. The interface conditions are again of the Dirichlet-Neumann form, i.e.,

$$\begin{aligned} u_1 &= u_2, && \text{on } \Gamma, \\ \mu_2 \nabla u_2 \cdot \boldsymbol{n}_2 + \mu_1 \nabla u_1 \cdot \boldsymbol{n}_1 &= 0, && \text{on } \Gamma. \end{aligned}$$

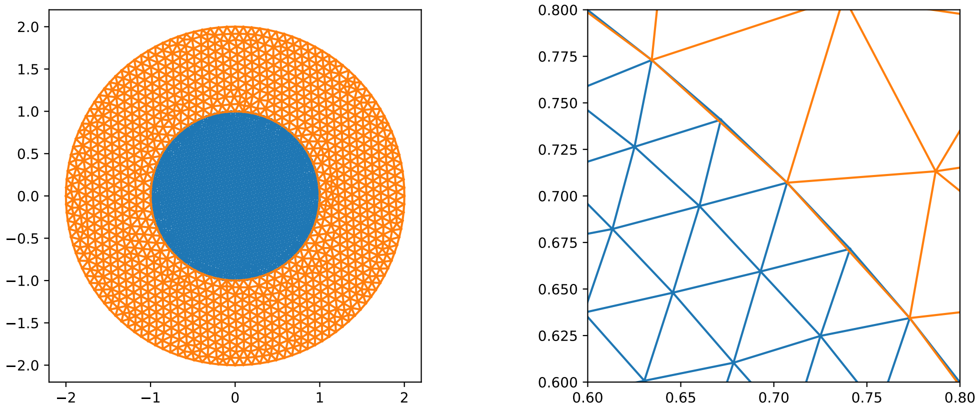
We discretize both problems using $\mathbb{P}_1$ finite elements, defined on meshes with respective sizes of $h_1 \approx 0.05$ and $h_2 \approx 0.1$. The resulting meshes are reported in Figure 6.10, which clearly shows the interface nonconformity. To handle this, suitable mesh transfer operators need to be constructed. Here, we use the Radial Basis Functions [38], so that

$$(\Pi_{12} g)(x_i^{\Gamma_1}) = \sum_{j=1}^{N_{\Gamma_2}} \phi\left(\left\| x_i^{\Gamma_1} - x_j^{\Gamma_2} \right\|\right) \alpha_j.$$

Here, $x_j^{\Gamma_i}$ refers to the j-th point on the interface Γ_i, and α_j are the interpolation coefficients, that can be found by solving

$$\sum_{j=1}^{N_{\Gamma_2}} \phi\left(\left\| x_k^{\Gamma_2} - x_j^{\Gamma_2} \right\|\right) \alpha_j = g(x_k^{\Gamma_2}).$$

Unlike the original version [38], we select $\phi(r) = \exp(-r)$ and we do not re-scale the operator.

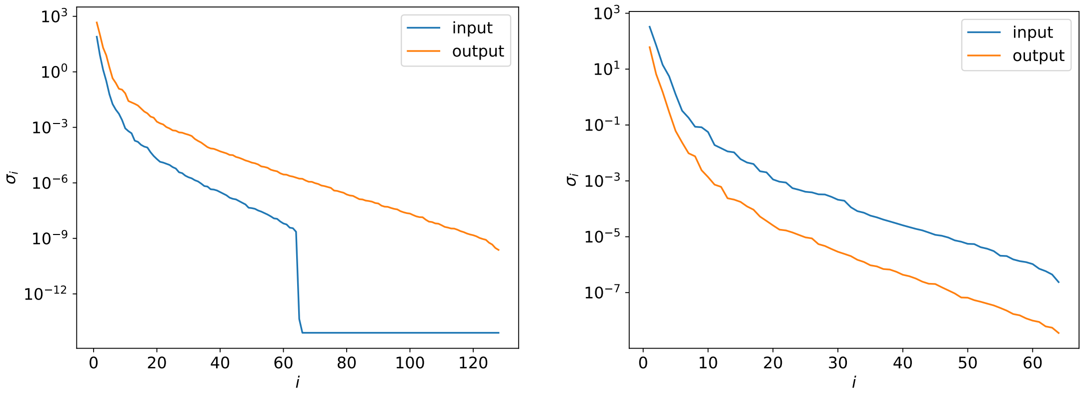


(a) Mesh (entire domain).

(b) Mesh (zoom).

Figure 6.10 – Meshes of the diffusion problem on a circular geometry.

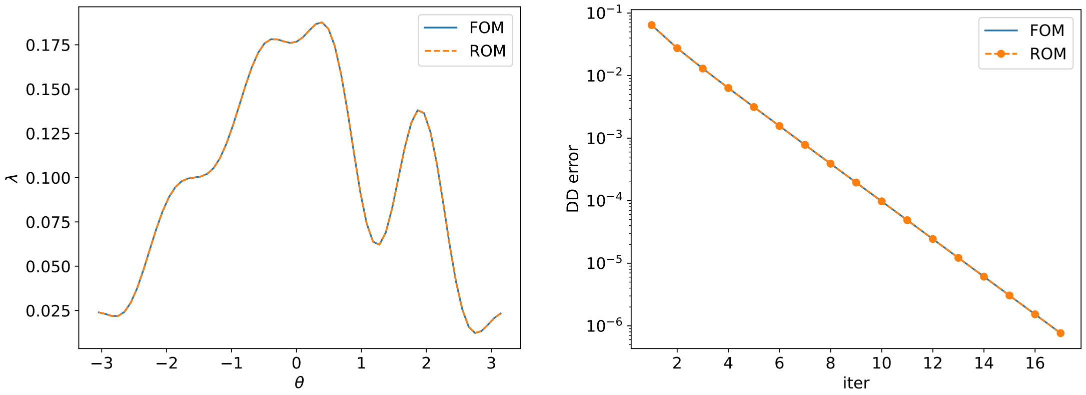
As before, we first validate our methodology using training data stemming from the coupled problem and fixing the relevant hyperparameters. We consider and 247 diffusion samples, which result in the singular values reported in Figure 6.11. A relatively fast decay is observed, thereby motivating the choice of $N_{\mathscr{A}} = N_{\mathscr{B}} = 6$ and $M_{\mathscr{A}} = M_{\mathscr{B}} = 16$, and the use of the VKOGA interpolator ($\nu = 2.5$). The domain decomposition iterations ($\omega = 0.5$) lead to the results reported in Figure 6.12 and the corresponding solutions are shown in Figure 6.13. The circular geometry introduces, at least for this particular problem, a more complex solution profile at the interface. Higher frequencies are present, so that a larger number of interface modes is required to achieve a high accuracy compared to its Cartesian counterpart. If that is taken into account, the interpolation error does not destroy the efficiency of the method, and the resulting reduced order solution visually resembles the full order one.



(a) $\mathscr{P}_1$.

(b) $\mathscr{P}_2$.

Figure 6.11 – Singular values of the diffusion problem on a circular geometry, training done with the coupled model.

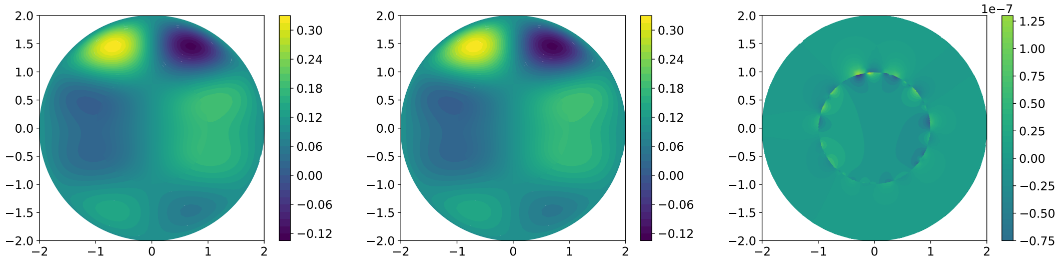


(a) Fixed point variable λ.

(b) Fixed point error.

Figure 6.12 – Results of the diffusion problem on a circular geometry, training done with the coupled model.

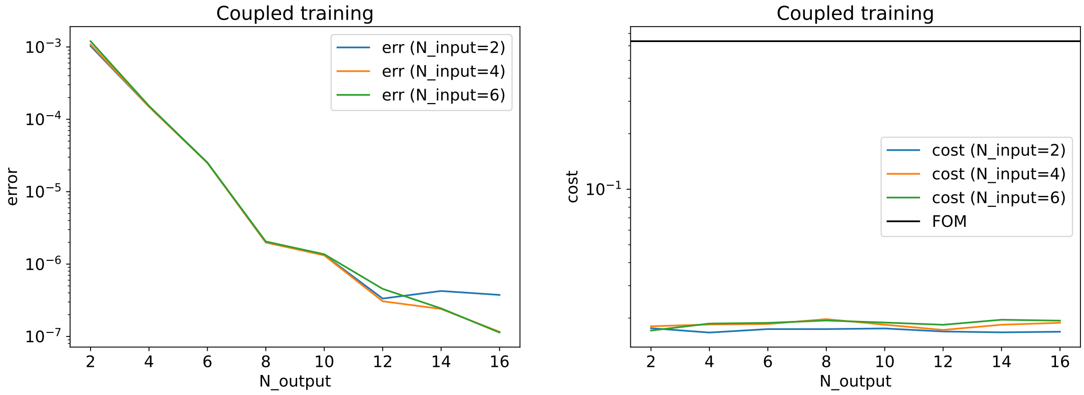
More quantitatively, we now vary the number of input and output modes. In parallel, we also use this test case to assess the computational performance of our method. The results



(a) FOM. (b) ROM. (c) Pointwise error.

Figure 6.13 – Solution of the diffusion problem on a circular geometry, training done with the coupled model.

of both analyses are reported in Figure 6.14. The error saturation observed in the previous test case does not appear here, at least for the hyperparameter values that we considered here. This is possibly due to the very simple nature of the problem, which possesses a high reduction potential. A similar trend is observed in Figure 6.5(a). The surrogate model is shown to be faster than its full order counterpart, with speedup values of around 25. Given the linear nature of the full problem and its relatively mild number of degrees of freedom, at least when compared to the training samples, this value is quite satisfactory. A further increase in the problem complexity and a higher mesh resolution contribute to obtain larger values. No significant changes are observed if more input or output modes are taken, suggesting that one could take relatively large values without significantly compromising the computational benefits.

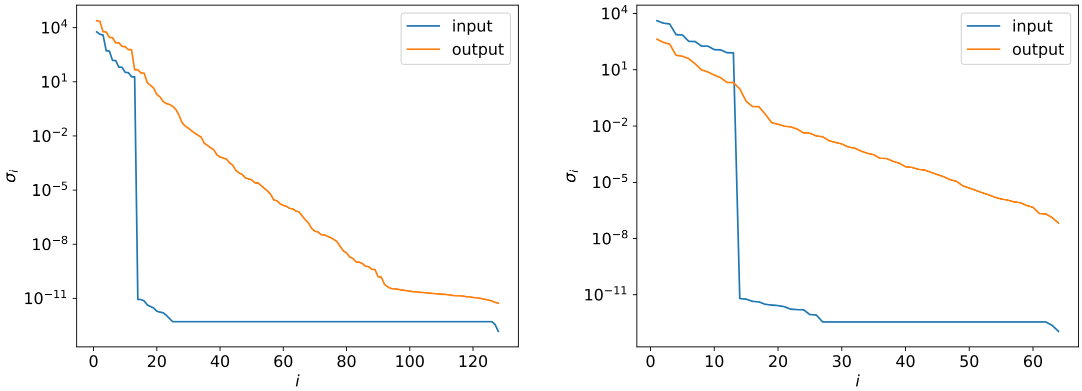


(a) Reduction error. (b) Computational cost.

Figure 6.14 – Reduction errors and computational cost of the diffusion problem on a circular geometry, training done with the coupled model.

Alternatively, one could also consider a decoupled training strategy. The snapshots are generated with 10 physical and 200 artificial parameter values, unless stated otherwise. As the interface is a closed curve, we consider periodic boundary conditions to solve the interface

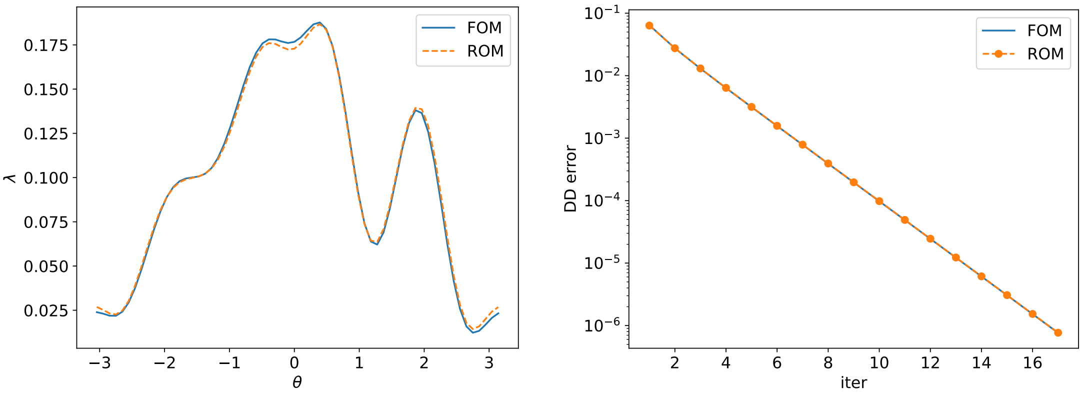
Laplace problem, in the spirit of a classical Fourier expansion. The values of the scaling $A = 10$ and the decay rates $r = 3$ and $r = 2$ are kept as in the previous test case. Considering $N_{\mathscr{A}} = N_{\mathscr{B}} = 13 = 6 \cdot 2 + 1$ basis functions, corresponding to sinusoidal, cosinusoidal, and the constant function, and simulating the local problems, we obtain the singular values shown in Figure 6.15. Solving the coupled problem using fixed point iterations at the full and the reduced level ($M_{\mathscr{A}} = M_{\mathscr{B}} = 16$), we obtain the solution trace and the corresponding solutions reported in Figure 6.16 and Figure 6.17. A larger error is observed with respect to its counterpart shown in Figure 6.13, as the training strategy is designed to trade efficiency for robustness.



(a) $\mathscr{P}_1$.

(b) $\mathscr{P}_2$.

Figure 6.15 – Singular values of the diffusion problem on a circular geometry, training done with the decoupled model.

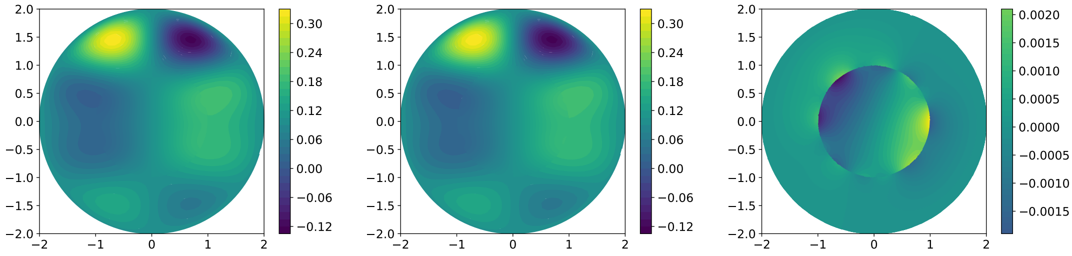


(a) Fixed point variable λ.

(b) Fixed point error.

Figure 6.16 – Results of the diffusion problem on a circular geometry, training done with the decoupled model.

To conclude the analysis, in Figure 6.18 we assess the efficiency of the method in terms of accuracy and computational cost when the hyperparameters are varied. Here, we use 200 artificial parameter samples for $N_{\mathscr{A}} \leq 5$ and 800 samples when $N_{\mathscr{A}} > 5$. The error components



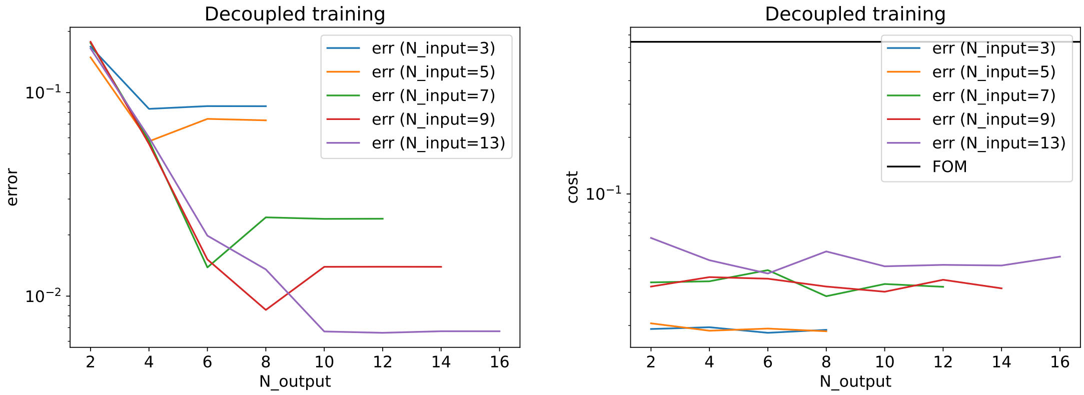
(a) FOM.

(b) ROM.

(c) Pointwise error.

Figure 6.17 – Solution of the diffusion problem on a circular geometry, training done with the decoupled model.

are clearly visible, as increasing the output dimension improves the accuracy until the input projection error starts to play a role. Due to the high frequency components of the fixed point variable, a quite large number of basis functions are required to obtain an accurate solution. The computational speedup obtained with the surrogate iterations ranges from 10 to 30, which is quite satisfactory, given that the training is completely agnostic to the coupled problem. The noticeable difference between the different values of the input dimension is due to the choice of a different number of artificial training samples, which has an approximately linear impact on the cost as in (6.16). Otherwise, no significant variations are present.



(a) Reduction error.

(b) Computational cost.

Figure 6.18 – Reduction errors and computational cost of the diffusion problem on a circular geometry, training done with the decoupled model.

6.3.3 Fluid-Structure Interaction

We move now to a more physically complex problem, which models the interaction between a fluid and an elastic structure [19]. This is designed to test our methodology in a more realistic scenario, characterized by a high degree of heterogeneity between the components. Additionally, the problem involves vector-valued differential equations and coupling conditions.

Initially, the fluid domain is $\Omega_1 = [0,5] \times [0,0.5]$, on top of which lies an elastic structure with domain $\Omega_2 = [0,5] \times [0.5,0.6]$. Tracking the change of the domain induced by the structural displacement using the ALE formulation, we can re-cast the fluid-structure problem as a coupled system of the form introduced in Section 6.1 [78]. The governing equations for $\mathscr{P}_1$ are the Navier-Stokes equations coupled with the mesh displacement equation

$$\begin{cases} -\Delta \boldsymbol{d}_f & = 0, \quad \text{in } \Omega_1, \\ \rho_f J \nabla \boldsymbol{u}_f \boldsymbol{F}^{-1} \boldsymbol{u}_f - \nabla \cdot \left(J \boldsymbol{\sigma}^f(\boldsymbol{u}_f, p_f) \boldsymbol{F}^{-T} \right) & = 0, \quad \text{in } \Omega_1, \\ \nabla \cdot \left(J \boldsymbol{F}^{-1} \boldsymbol{u}_f \right) & = 0, \quad \text{in } \Omega_1, \end{cases}$$

where $\boldsymbol{u}_f$, p_f and $\boldsymbol{d}_f$ are the fluid velocity, pressure and the mesh displacement, respectively. Moreover,

$$\boldsymbol{\sigma}^f \left(\boldsymbol{u}_f, p_f \right) = \mu^f \left(\nabla \boldsymbol{u}_f \boldsymbol{F}^{-1} + \left(\nabla \boldsymbol{u}_f \boldsymbol{F}^{-1} \right)^T \right) - p_f \boldsymbol{I}$$

is the Cauchy stress tensor, $\boldsymbol{F} = \boldsymbol{I} + \nabla \boldsymbol{d}_f$ is the gradient of the ALE map, $\boldsymbol{I}$ is the identity tensor and $J = \det \boldsymbol{F}$. The governing equation for $\mathscr{P}_2$ is instead the elasticity equation defined by

$$-\nabla \cdot \boldsymbol{P}^s(\boldsymbol{\eta}_s) = 0, \quad \text{in } \Omega_2,$$

where $\boldsymbol{P}^s$ is the first Piola tensor, whose expression depends on the constitutive relation of the solid. Here, we consider a Saint Venant-Kirchhoff model [78], which gives

$$\boldsymbol{P}^s = \boldsymbol{F}_s \left(2\mu_s \boldsymbol{E} + \lambda_s \operatorname{tr}\{\boldsymbol{E}\} \boldsymbol{I} \right), \quad \text{where } \boldsymbol{E} = \frac{1}{2} \left(\boldsymbol{F}_s^T \boldsymbol{F}_s - \boldsymbol{I} \right) \text{ and } \boldsymbol{F}_s = \boldsymbol{I} + \nabla \boldsymbol{\eta}_s.$$

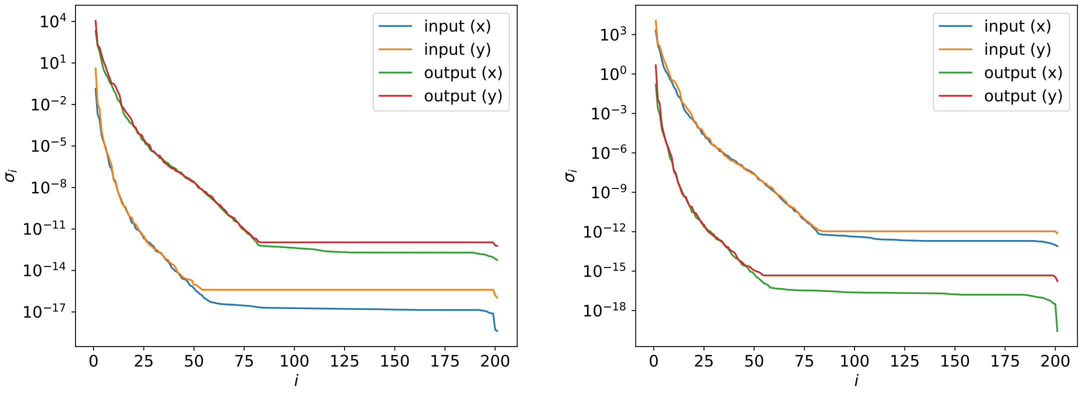
The physical variables are the fluid density $\rho_f = 1.1$ and viscosity $\mu_f = 0.1 + 0.4\mu_0$ and the Lamé parameters $\mu_s = \frac{E}{2(1+\nu)}$ and $\lambda_s = \frac{E\nu}{(1+\nu)(1-2\nu)}$, which in turn depend on the Young modulus $E = 10^7 + 5 \cdot 10^7 E_0$ and the Poisson ratio $\nu = 0.31$. The physical parameters of interest are μ_0 and E_0, both of which vary in the unit interval. As boundary conditions on the fluid physical domain, we impose a pressure $p_{\text{in}} = 10^4 \cdot 5 \cdot 10^{-3}$ at the inlet boundary, a zero-stress condition at the outlet boundary, and a slip condition at the bottom boundary [19]. Homogeneous Dirichlet and Neumann conditions are set at the left/right and top boundary of the structure, respectively. The coupling conditions read

$$\boldsymbol{d}_f = \boldsymbol{\eta}_s \quad \text{and} \quad \boldsymbol{u}_f = \boldsymbol{0}, \quad \text{on } \Gamma, \tag{6.17}$$

$$\boldsymbol{P}_s \boldsymbol{n}_s + \left(J \boldsymbol{\sigma}_f \boldsymbol{F}^{-T} \right) \boldsymbol{n}_f = \boldsymbol{0}, \quad \text{on } \Gamma. \tag{6.18}$$

To ensure accuracy and stability at a full level, we discretize the fluid, structural, and geometrical problem using $\mathbb{P}_2$-$\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbb{P}_2$ finite elements, respectively, on meshes with approximate sizes of $h \approx 0.1 \cdot \sqrt{2}$.

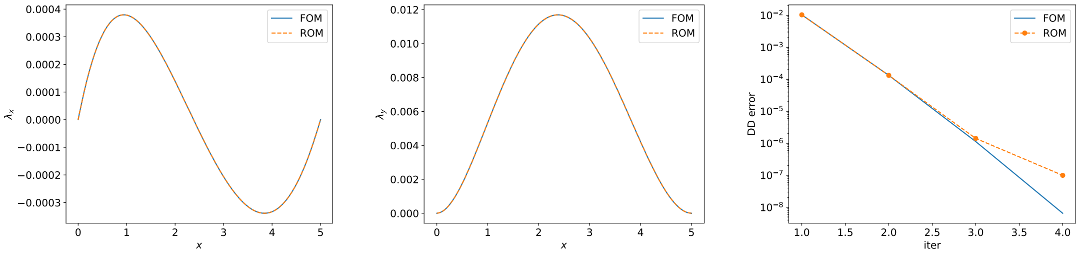
We first consider training data generated using the coupled problem. We take 247 parameter samples, which result in the singular values reported in Figure 6.19. As the coupling conditions (6.17) and (6.18) have a two-dimensional nature, the input and output variables naturally have an x and a y component, that are treated independently. This does not impact the fast decay of the singular values. To approximate the fixed-point mapping, we project each variable using 4 modes, so that the input and output dimensions of the corresponding VKOGA interpolators are $N_{\mathscr{A}} = N_{\mathscr{B}} = 4 \cdot 2$ and $M_{\mathscr{A}} = M_{\mathscr{B}} = 4 \cdot 2$. The smoothness of the Matérn kernel is $\nu = 1.5$. The domain decomposition iterations ($\omega = 1$) lead to the results reported in Figure 6.20, and the corresponding solutions are shown in Figure 6.21. Despite the input and output dimensions being rather high, our surrogate model still gives very accurate results.



(a) Fluid.

(b) Structure.

Figure 6.19 – Singular values of the fluid-structure interaction problem, training done with the coupled model.



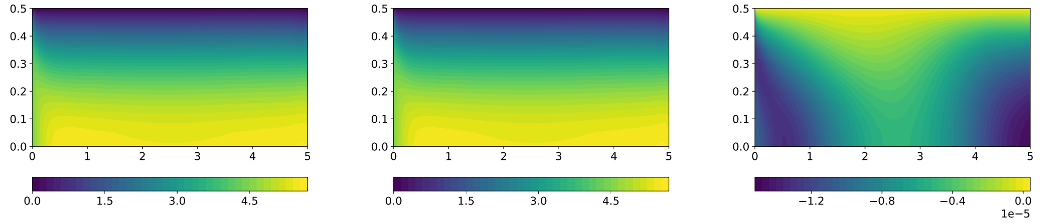
(a) Fixed point variable λ, x component.

(b) Fixed point variable λ, y component.

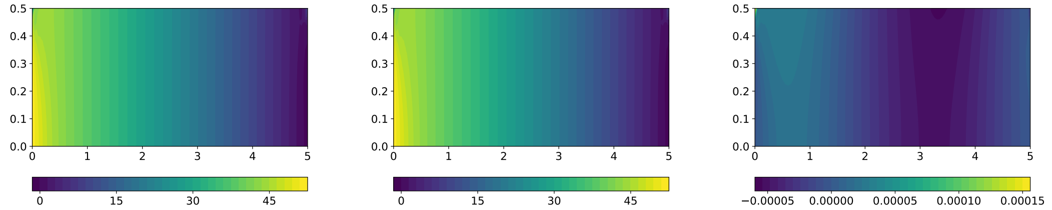
(c) Fixed point error.

Figure 6.20 – Results of the fluid-structure interaction problem, training done with the coupled model.

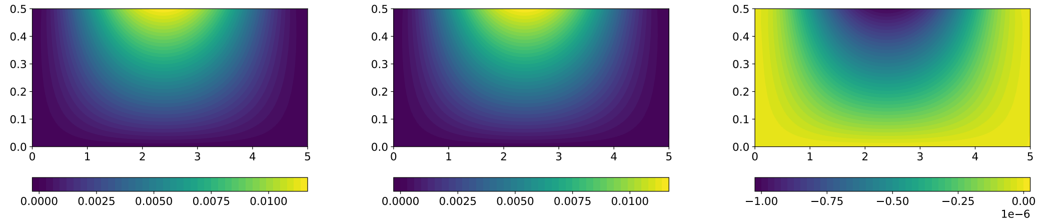
We can study more quantitatively the accuracy if the number of input and output functions are varied. Moreover, given the complex nature of the underlying system, this is a good test case to assess the capabilities of an ANN. This should be viewed as a proof of concept, as we did not aim to find the best hyperparameters for this test case, but only to show that an ANN could be used in such cases, especially for high-dimensional input values. The selected architecture consists of 3 hidden layers with 20 neurons each. The errors obtained with both methods are reported in Figure 6.22. The VKOGA gives good accuracy, although



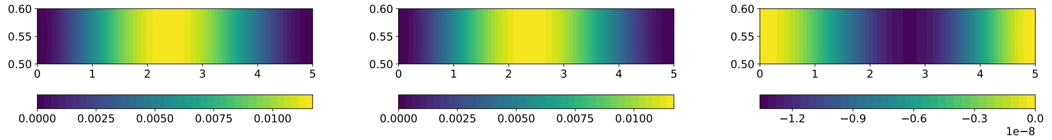
(a) $|\boldsymbol{u}_f|$, FOM. (b) $|\boldsymbol{u}_f|$, ROM. (c) $|\boldsymbol{u}_f|$, Pointwise error.



(d) p_f, FOM. (e) p_f, ROM. (f) p_f, Pointwise error.



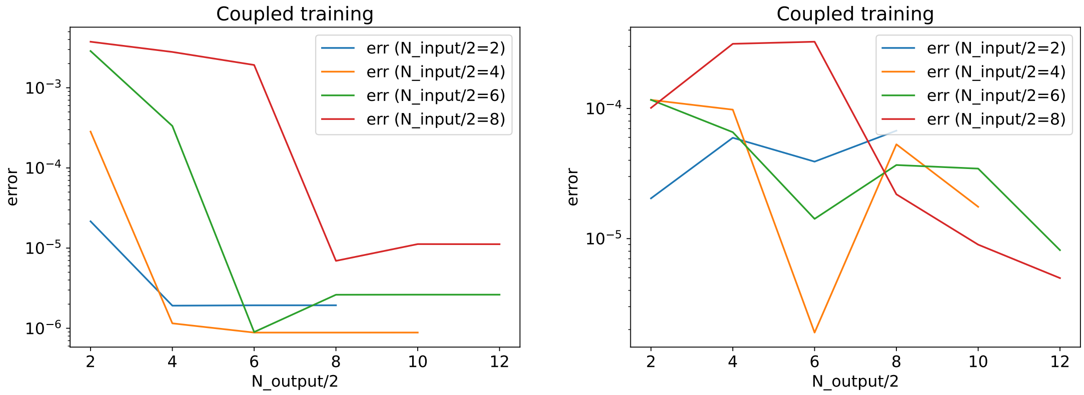
(g) $|\boldsymbol{d}_f|$, FOM. (h) $|\boldsymbol{d}_f|$, ROM. (i) $|\boldsymbol{d}_f|$, Pointwise error.



(j) $|\boldsymbol{\eta}_s|$, FOM. (k) $|\boldsymbol{\eta}_s|$, ROM. (l) $|\boldsymbol{\eta}_s|$, Pointwise error.

Figure 6.21 – Solution of the fluid-structure interaction problem, training done with the coupled model.

for high dimensions a suboptimal behavior seems to appear, as a result of the complexity of the problem, numerical instabilities and the curse of dimensionality. The ANN achieves competitive accuracy values, and seems to surpass the VKOGA for high dimensions. The multi-layer, compositional structure of the network can indeed reduce the curse of dimensionality and achieve better generalization properties than a more classical interpolation method. However, this is quite sensitive to the training algorithm and its intrinsic randomicity, which make the trend less smooth compared to the VKOGA.



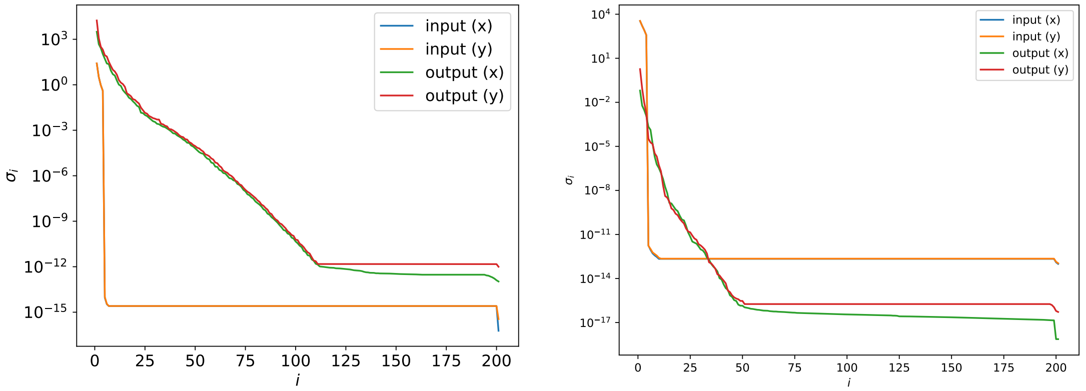
(a) Reduction error (VKOGA).

(b) Reduction error (ANN).

Figure 6.22 – Reduction errors of the fluid-structure interaction problem, training done with the coupled model.

If a decoupled training strategy is considered, the results and comments of the previous test cases do not vary significantly. However, at least for the structural component, the boundary conditions of the input variable are not homogeneous. To capture them, we enrich the basis of the Dirichlet eigenfunctions with two lifting functions, in the spirit of classical approximation theory. This is not the only option, as one could consider instead both Dirichlet and Neumann eigenfunctions or more general bases, but this choice allows us to easily control both the basis size and the projection error. We again consider with 10 physical and 200 artificial parameter values. The values of the scaling factors are chosen as $A = 0.1$ and $A = 10$ and the decay rates $r = 3$ and $r = 2$ for the fluid and elastic problems, respectively. Considering $N_{\mathscr{A}} = N_{\mathscr{B}} = M_{\mathscr{A}} = M_{\mathscr{B}} = 4 \cdot 2$ basis functions as before, we obtain the singular values shown in Figure 6.23. Solving the coupled problem using fixed point iterations, we obtain the trace and the corresponding solutions reported in Figure 6.24 and Figure 6.25. Once more, larger deviations from the true model are observed with respect to their counterparts Figure 6.20 and Figure 6.21.

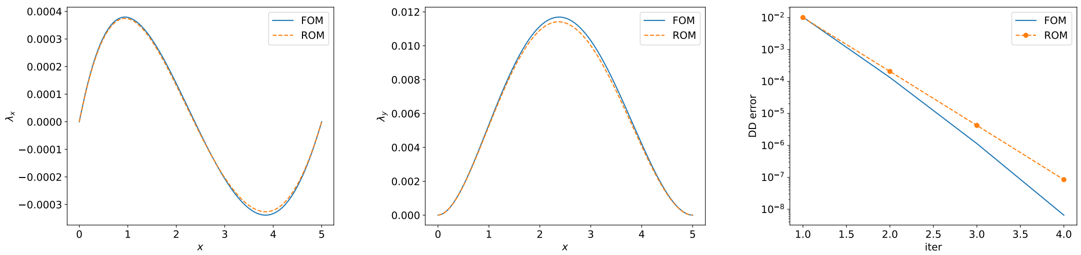
Finally, we consider a variation in the number of input and output functions and a comparison between the two model reduction techniques. We use 200 artificial parameter samples for $N_{\mathscr{A}} \leq 5 \cdot 2$ and 800 samples when $N_{\mathscr{A}} > 5 \cdot 2$. The results are reported in Figure 6.26, which confirm the competitiveness of the ANN method, although the error decay suffers from oscillations caused by the training phase. Moreover, in this problem the output projection error appears to be small compared the other error components, as in Figure 6.26(a) the error saturation is observed even for small output dimensions, at least with the VKOGA.



(a) Fluid.

(b) Structure.

Figure 6.23 – Singular values of the fluid-structure interaction problem, training done with the decoupled model.



(a) Fixed point variable λ, x component.

(b) Fixed point variable λ, y component.

(c) Fixed point error.

Figure 6.24 – Results of the fluid-structure interaction problem, training done with the decoupled model.

6.3.4 A time-dependent problem

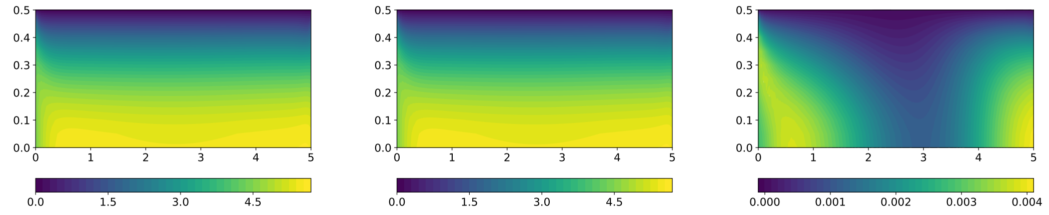
Our final test case is designed to assess the capabilities of our method with a time-dependent problem. We consider a parametrized heat equation, with possibly discontinuous diffusion coefficients at the subdomain interface. This should be regarded as a proof of concept, as a more detailed investigation and the extension to more complex time-dependent problems go beyond the scope of this work. We consider $\Omega = [0,1] \times [0,2]$, divided into two subdomains by a horizontal interface Γ located at $y = 1$. The time domain is $[0,T] = [0,1]$. The governing equation for $\mathscr{P}_i$ is

$$\frac{\partial u_i}{\partial t} - \nabla \cdot \left(\mu_i \nabla u_i\right) = f_i, \quad \text{in } \Omega_i \times [0,T],$$

where we set $f_i = 10x\sin\left(y/2\right)$ and

$$\mu_i = \mu + y_i,$$

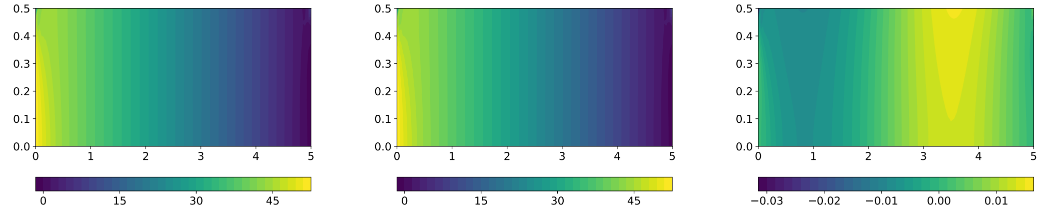
with a shift $\mu = 1$ and problem-dependent parameters $y_i \in [0,1]$. The problem is completed by homogeneous Dirichlet conditions at the physical boundary and an initial condition



(a) $|\boldsymbol{u}_f|$, FOM.

(b) $|\boldsymbol{u}_f|$, ROM.

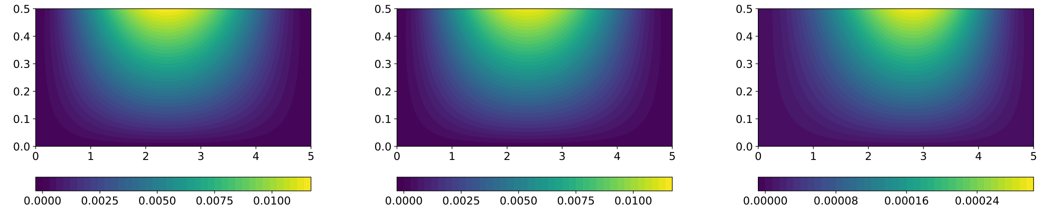
(c) $|\boldsymbol{u}_f|$, Pointwise error.



(d) p_f, FOM.

(e) p_f, ROM.

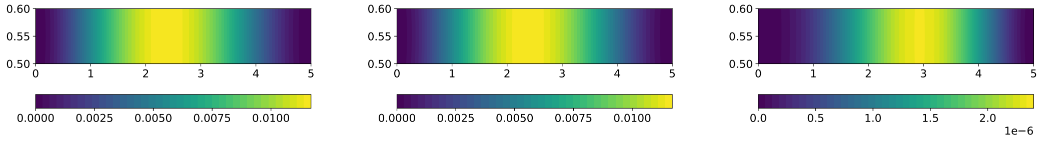
(f) p_f, Pointwise error.



(g) $|\boldsymbol{d}_f|$, FOM.

(h) $|\boldsymbol{d}_f|$, ROM.

(i) $|\boldsymbol{d}_f|$, Pointwise error.



(j) $|\boldsymbol{\eta}_f|$, FOM.

(k) $|\boldsymbol{\eta}_f|$, ROM.

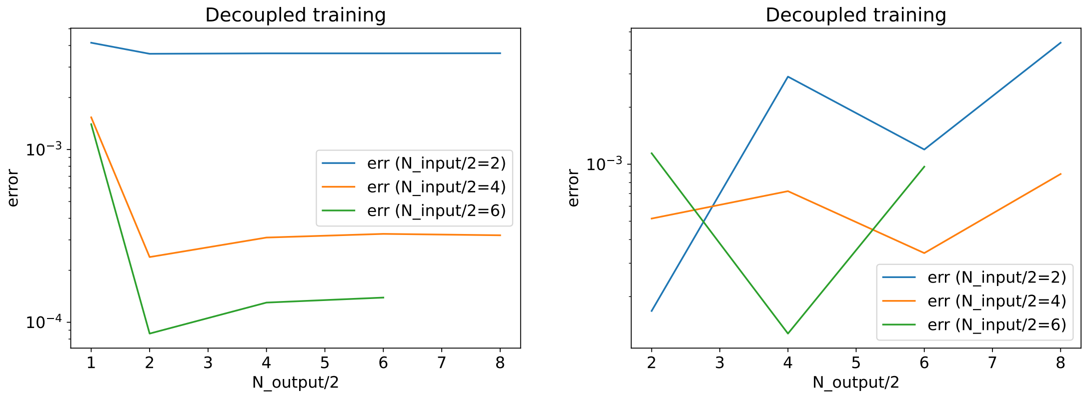
(l) $|\boldsymbol{\eta}_f|$, Pointwise error.

Figure 6.25 – Solution of the fluid-structure interaction problem, training done with the decoupled model.

$u_i = \sin(2\pi x)\sin(\pi y)$ in both subdomains, whereas the interface coupling conditions read

$$\begin{aligned} u_1 &= u_2, \quad \text{on } \Gamma \times [0,T], \\ \mu_2 \nabla u_2 \cdot \boldsymbol{n}_2 + \mu_1 \nabla u_1 \cdot \boldsymbol{n}_1 &= 0, \quad \text{on } \Gamma \times [0,T]. \end{aligned}$$

As in the previous test cases, they are of a standard Dirichlet-Neumann form, but they are imposed in the entire time interval. The problem is discretized using implicit Euler in time



(a) Reduction error (VKOGA).

(b) Reduction error (ANN).

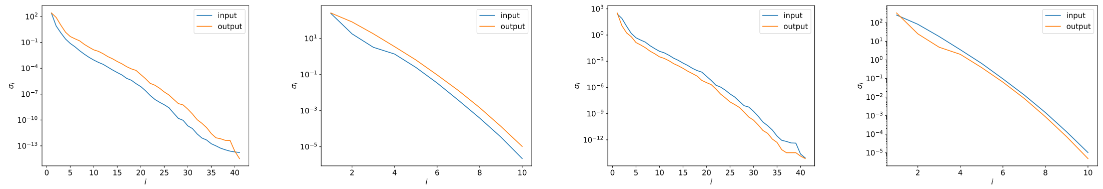
Figure 6.26 – Reduction errors of the fluid-structure interaction problem, training done with the decoupled model.

with a time step $\Delta t = 0.1$ and $\mathbb{P}_1$ finite elements in space for both subdomains, defined on two meshes with approximate sizes of $h \approx \sqrt{2}/40$.

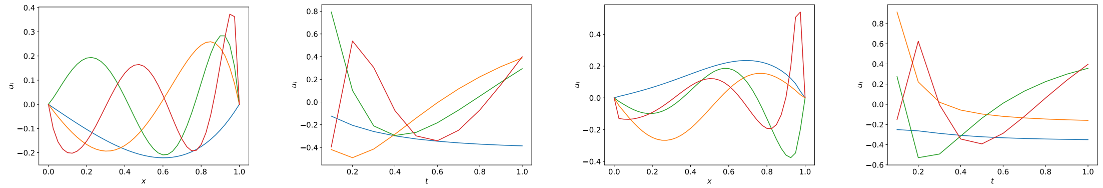
As usual, we focus firstly on training data generated with the coupled problem. We consider 97 parameter samples obtained with a Latin Hypercube Sampling technique. The singular values and the first 4 spatial modes obtained for both the input and the output spatial data, as well as their temporal counterparts, are reported in Figure 6.27. Even if a time-dependent problem is considered, the exponential decay of the singular values is not impacted, which indicates good reduction properties of the training sets. After selecting $N_{\mathscr{A}} = N_{\mathscr{B}} = 4 \cdot 4$ and $M_{\mathscr{A}} = M_{\mathscr{B}} = 4 \cdot 4$, we use the VKOGA to construct the interpolator ($\nu = 1.5$). Taking a parameter value outside the training set, the fixed point iterations ($\omega = 0.2$) give the results reported in Figure 6.28. Compared to its steady counterparts, the deviations from the true error decay rate are slightly larger, but the true and the approximated fixed point variables visually look similar. By solving the local problems with the converged trace values, we can reconstruct the local solutions. Their values at time $t = T$, together with the corresponding pointwise error, are reported in Figure 6.29, confirming the satisfactory accuracy of our method.

We now study the effect of varying input and output dimensions on the reduction. As function approximation methods, we consider both the VKOGA interpolator and an ANN. For the latter, we select 2 hidden layers with 80 neurons each. Taking the same number of spatial and temporal modes, the results are reported in Figure 6.30. Given the tensor product nature of the space-time basis, the input dimension quickly increases. This is the main reason why we cannot reach extremely high accuracy in the selected range of hyperparameter values, together with possible numerical instabilities in the training algorithm. However, we are still able to reach errors of order $\mathcal{O}(10^{-4})$, which remains quite satisfactory. The ANN gives similar results, and appears to outperform the VKOGA for large input dimensions.

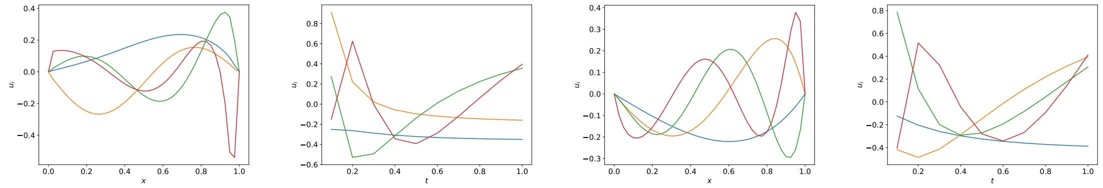
We now repeat the analysis using data obtained from the two subproblems separately. The



(a) $\mathscr{P}_1$, space. (b) $\mathscr{P}_1$, time. (c) $\mathscr{P}_2$, space. (d) $\mathscr{P}_2$, time.



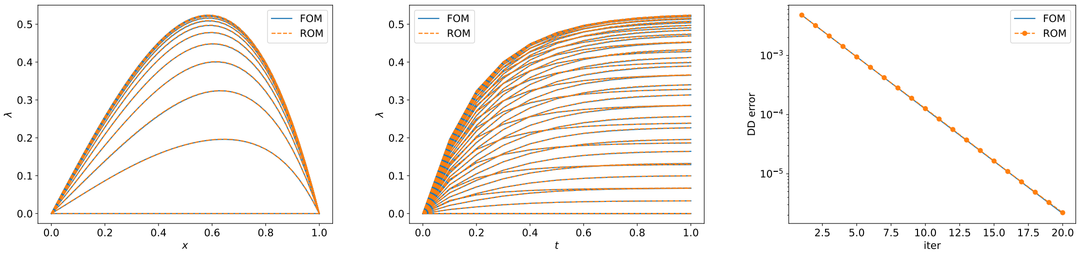
(e) $\mathscr{P}_1$, space. (f) $\mathscr{P}_1$, time. (g) $\mathscr{P}_2$, space. (h) $\mathscr{P}_2$, time.



(i) $\mathscr{P}_1$, space. (j) $\mathscr{P}_1$, time. (k) $\mathscr{P}_2$, space. (l) $\mathscr{P}_2$, time.

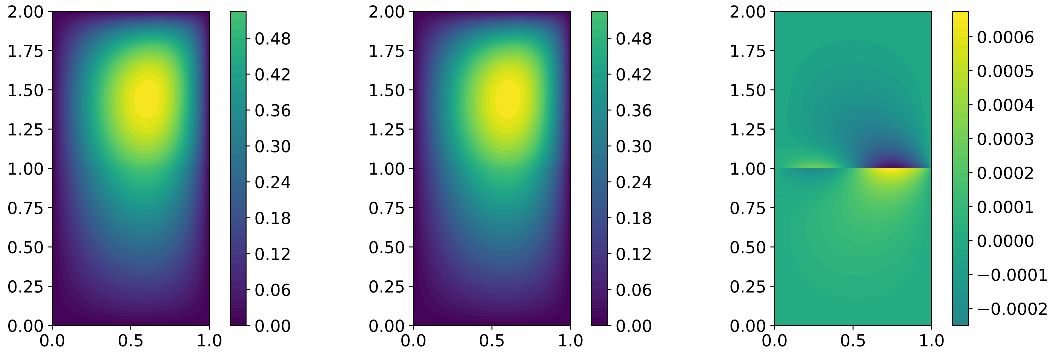
Figure 6.27 – Singular values and singular vectors associated to the time-dependent problem, training done with the coupled model, spatial and temporal data. From top to bottom: singular values, input singular vectors, output singular vectors.

training set is generated with $4 \cdot 4$ basis functions in the expansion (6.12), 10 samples of the physical parameters and $20 \cdot 20$ space-time samples of the artificial parameters. We consider a scaling factor of $A = 1$ (resp. $A = 10$) with no decay rate for $\mathscr{P}_1$ (resp. $\mathscr{P}_2$). The results of the SVD are reported in Figure 6.31, for which the comments related to the steady counterpart can be extended. Testing the corresponding VKOGA interpolator gives the results reported in Figures 6.7 and 6.8, which show the fixed point variables and numerical solutions at the final time, respectively. Given the high dimension of the input and the output space and the relatively small number of training samples, the error between the full and the reduced models becomes more evident. However, the main features of the dynamical behavior are captured,



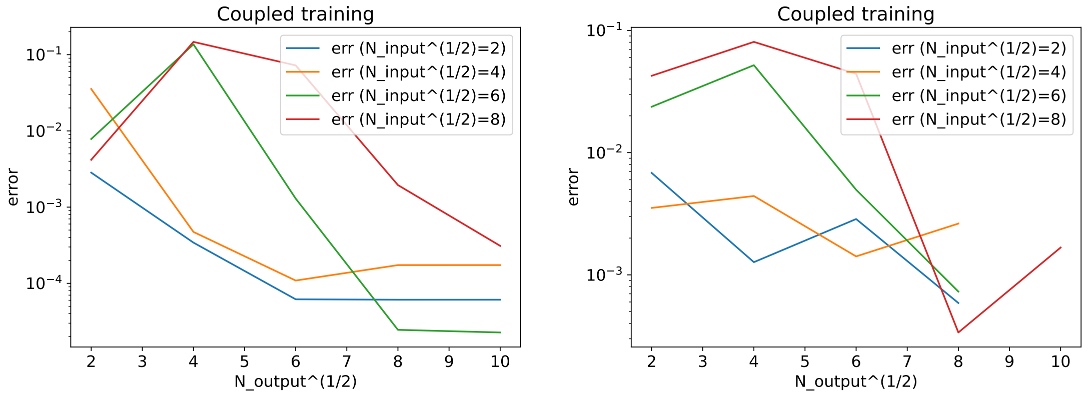
(a) Fixed point variable λ. (b) Fixed point variable λ. (c) Fixed point error.

Figure 6.28 – Results of the time-dependent problem, training done with the coupled model.



(a) FOM. (b) ROM. (c) Pointwise error.

Figure 6.29 – Solution of the time-dependent problem, training done with the coupled model.

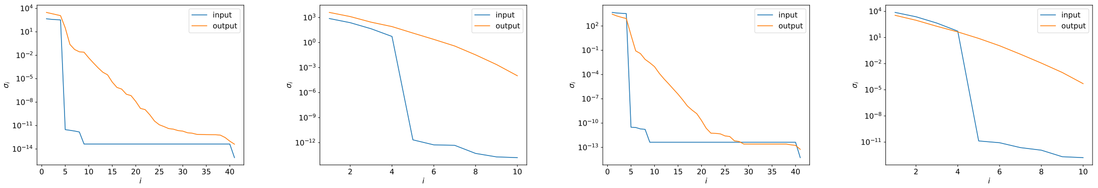


(a) Reduction error (VKOGA). (b) Reduction error (ANN).

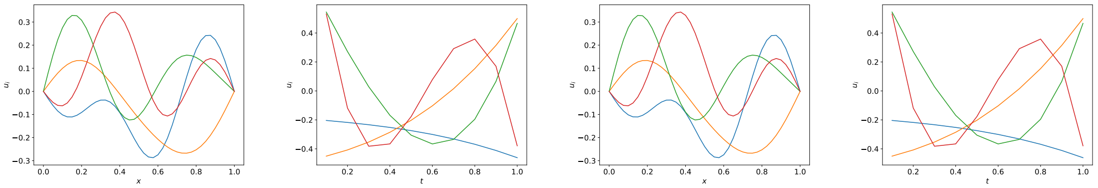
Figure 6.30 – Reduction errors of the time-dependent problem, training done with the coupled model.

suggesting that the chosen surrogate modeling technique can still provide a satisfactory approximation of the solutions without the need to solve a time-dependent problem at each iteration.

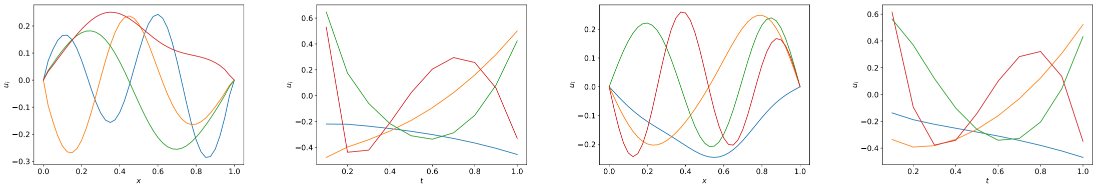
This argument can be made more rigorous by looking at the model accuracy as a function of the hyperparameters, reported in Figure 6.34. We still use $20 \cdot 20$ artificial parameter samples. The magnitude of the reduction error lies at around $\mathcal{O}(10^{-2})$ and seems to suffer from the curse of dimensionality for large input dimensions. As no information from the coupled problem is given and that the number of samples is low compared to the involved dimensions, the obtained degree of accuracy remains acceptable.



(a) $\mathscr{P}_1$, space.
(b) $\mathscr{P}_1$, time.
(c) $\mathscr{P}_2$, space.
(d) $\mathscr{P}_2$, time.



(e) $\mathscr{P}_1$, space.
(f) $\mathscr{P}_1$, time.
(g) $\mathscr{P}_2$, space.
(h) $\mathscr{P}_2$, time.



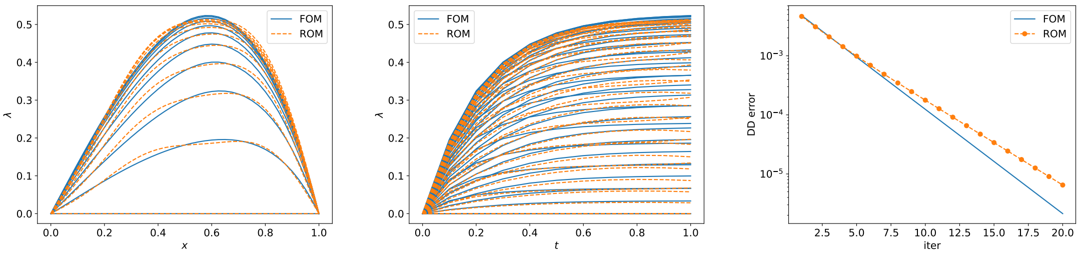
(i) $\mathscr{P}_1$, space.
(j) $\mathscr{P}_1$, time.
(k) $\mathscr{P}_2$, space.
(l) $\mathscr{P}_2$, time.

Figure 6.31 – Singular values and singular vectors associated to the time-dependent problem, training done with the decoupled model, spatial and temporal data. From top to bottom: singular values, input singular vectors, output singular vectors.

6.4 Conclusion

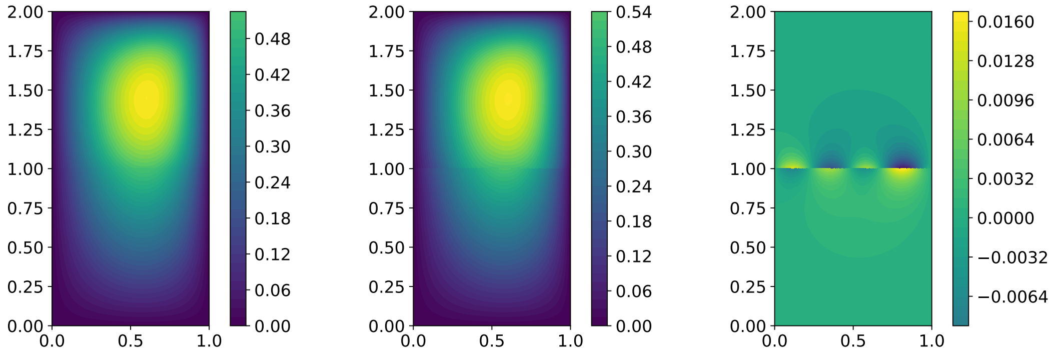
In this chapter, we proposed a local, non-intrusive method to accelerate the multi-query simulations of coupled heterogeneous systems. The approximated input-output maps are mostly constructed based on kernel methods, although artificial neural networks constitute a valid alternative. We propose two training strategies, which make use of the coupled problem or an artificial parametrization of the interface data, respectively. In this regard, the latter can be viewed as the non-intrusive counterpart of Chapter 5.

We show that the solution of a given coupled problem can be efficiently retrieved at the reduced



(a) Fixed point variable λ.
(b) Fixed point variable λ.
(c) Fixed point error.

Figure 6.32 – Results of the time-dependent problem, training done with the decoupled model.

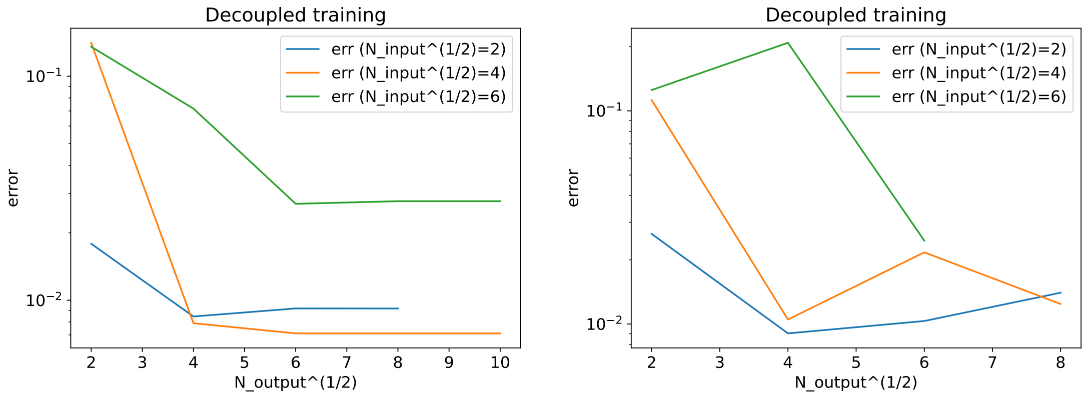


(a) FOM.

(b) ROM.

(c) Pointwise error.

Figure 6.33 – Solution of the time-dependent problem, training done with the decoupled model.



(a) Reduction error (VKOGA).

(b) Reduction error (ANN).

Figure 6.34 – Reduction errors of the time-dependent problem, training done with the decoupled model.

level. The accuracy depends on both dimensionality reduction errors and the quality of the interpolation, and appears not to be strongly affected by the complexity of the underlying differential model. Training with data from a specific coupled problem gives more accurate results than an artificial parametrization, provided that the parameter space is sufficiently explored in the offline phase. Consistent with any interpolation technique, the computational cost is mostly affected by the number of training samples, at least when kernel methods are used.

6.A A detailed error analysis

Here, we aim at quantifying the error contributions in a more detailed way. In the framework presented in Section 6.2.2, our goal is to quantify the error between the solutions of true fixed point mapping

$$\lambda = \mathscr{F}(\lambda) = \mathscr{B}(-\mathscr{A}(\lambda))$$

and its surrogate

$$\hat{\lambda} = \hat{\mathscr{F}}(\hat{\lambda}) = U_y^{\mathscr{B}} \hat{\mathscr{B}}\left(-[U_x^{\mathscr{B}}]^T U_y^{\mathscr{A}} \hat{\mathscr{A}}\left([U_x^{\mathscr{A}}]^T \hat{\lambda}\right)\right).$$

By subtracting the two and using the triangular inequality, we get

$$\left\|\lambda - \hat{\lambda}\right\| = \left\|\mathscr{F}(\lambda) - \hat{\mathscr{F}}(\hat{\lambda})\right\| \leq \left\|\mathscr{F}(\lambda) - \hat{\mathscr{F}}(\lambda)\right\| + \left\|\hat{\mathscr{F}}(\lambda) - \hat{\mathscr{F}}(\hat{\lambda})\right\|, \tag{6.19}$$

which allows us to study the two terms separately.

The first term quantifies the error between the two mappings with same input datum. Applying the triangular inequality multiple times, we get

$$\begin{aligned}
&\left\|\mathscr{F}(\lambda) - \hat{\mathscr{F}}(\lambda)\right\| = \left\|\mathscr{B}(-\mathscr{A}(\lambda)) - U_y^{\mathscr{B}} \hat{\mathscr{B}}\left(-[U_x^{\mathscr{B}}]^T U_y^{\mathscr{A}} \hat{\mathscr{A}}\left([U_x^{\mathscr{A}}]^T \lambda\right)\right)\right\| \\
&\leq \left\|\mathscr{B}(-\mathscr{A}(\lambda)) - U_y^{\mathscr{B}}[U_y^{\mathscr{B}}]^T \mathscr{B}(-\mathscr{A}(\lambda))\right\| \qquad \text{(6.20a)} \\
&+ \left\|U_y^{\mathscr{B}}[U_y^{\mathscr{B}}]^T \mathscr{B}(-\mathscr{A}(\lambda)) - U_y^{\mathscr{B}}[U_y^{\mathscr{B}}]^T \mathscr{B}\left(-U_x^{\mathscr{B}}[U_x^{\mathscr{B}}]^T \mathscr{A}(\lambda)\right)\right\| \qquad \text{(6.20b)} \\
&+ \left\|U_y^{\mathscr{B}}[U_y^{\mathscr{B}}]^T \mathscr{B}\left(-U_x^{\mathscr{B}}[U_x^{\mathscr{B}}]^T \mathscr{A}(\lambda)\right) - U_y^{\mathscr{B}} \hat{\mathscr{B}}\left(-[U_x^{\mathscr{B}}]^T \mathscr{A}(\lambda)\right)\right\| \qquad \text{(6.20c)} \\
&+ \left\|U_y^{\mathscr{B}} \hat{\mathscr{B}}\left(-[U_x^{\mathscr{B}}]^T \mathscr{A}(\lambda)\right) - U_y^{\mathscr{B}} \hat{\mathscr{B}}\left(-[U_x^{\mathscr{B}}]^T U_y^{\mathscr{A}}[U_y^{\mathscr{A}}]^T \mathscr{A}(\lambda)\right)\right\| \qquad \text{(6.20d)} \\
&+ \left\|U_y^{\mathscr{B}} \hat{\mathscr{B}}\left(-[U_x^{\mathscr{B}}]^T U_y^{\mathscr{A}}[U_y^{\mathscr{A}}]^T \mathscr{A}(\lambda)\right) - U_y^{\mathscr{B}} \hat{\mathscr{B}}\left(-[U_x^{\mathscr{B}}]^T U_y^{\mathscr{A}}[U_y^{\mathscr{A}}]^T \mathscr{A}\left(U_x^{\mathscr{A}}[U_x^{\mathscr{A}}]^T \lambda\right)\right)\right\| \qquad \text{(6.20e)} \\
&+ \left\|U_y^{\mathscr{B}} \hat{\mathscr{B}}\left(-[U_x^{\mathscr{B}}]^T U_y^{\mathscr{A}}[U_y^{\mathscr{A}}]^T \mathscr{A}\left(U_x^{\mathscr{A}}[U_x^{\mathscr{A}}]^T \lambda\right)\right) - U_y^{\mathscr{B}} \hat{\mathscr{B}}\left(-[U_x^{\mathscr{B}}]^T U_y^{\mathscr{A}} \hat{\mathscr{A}}\left([U_x^{\mathscr{A}}]^T \lambda\right)\right)\right\|. \qquad \text{(6.20f)}
\end{aligned}$$

Each of the terms is associated to a different error component:

- projection due to $U_y^{\mathscr{B}} \in \mathbb{R}^{m_{\mathscr{B}} \times M_{\mathscr{B}}}$,
- projection due to $U_x^{\mathscr{B}} \in \mathbb{R}^{n_{\mathscr{B}} \times N_{\mathscr{B}}}$,
- approximation due to $\hat{\mathscr{B}}$ constructed with $\mathscr{N}_{\mathscr{B}}$ samples,
- projection due to $U_y^{\mathscr{A}} \in \mathbb{R}^{m_{\mathscr{A}} \times M_{\mathscr{A}}}$,
- projection due to $U_x^{\mathscr{A}} \in \mathbb{R}^{n_{\mathscr{A}} \times N_{\mathscr{A}}}$,
- approximation due to $\hat{\mathscr{A}}$ constructed with $\mathscr{N}_{\mathscr{A}}$ samples.

The behavior of each term can be quantified in a more detailed way.

The projection errors depend on the training strategy and the complexity of the problem. If data from the coupled problem are used and a diffusion-like equation is solved, an exponential decay can be expected for both the input ((6.20b) and (6.20e)) and the output ((6.20a) and (6.20d)) projection errors, similar to classical estimates on the Kolmogorov width. Otherwise, if the problems are treated separately, the input projection error will depend on the accuracy of the expansion (6.9). In turn, this depends on regularity of the problem, its spatial dimension and the compatibility of the selected Laplace-Beltrami eigenfunctions with the boundary conditions at the physical boundaries. This is typically polynomial, although an exponential decay can appear in a number of cases [3, 20]. The output projection error will again depend on the complexity of the local problems, and an exponential decay can be expected for diffusion-like equations.

The operator approximation errors ((6.20c) and (6.20f)) depend on the chosen technique, the learning algorithm, the training strategy and the number of training data. If the VKOGA method is used, one can show a polynomial decay rate in the number of training points, with an exponent that depends on the input spatial dimension and the regularity of the kernel [113, 135]. In the case of the Matérn kernels (2.21), this is typically of the form

$$\mathscr{N}^{-\nu/N}, \tag{6.21}$$

where N is the input dimension, ν is the smoothness of the kernel and $\mathscr{N}$ is the number of training points selected in the offline phase.

In order to appropriately bound each term in (6.20), we need suitable regularity assumptions on the full order operators, that we assume to hold, and the approximated operator $\hat{\mathscr{B}}$ to be Lipschitz continuous. This also holds, provided that the kernel is sufficiently smooth. More quantitatively, a bound for the derivative $\mathscr{D}$ of $\hat{\mathscr{B}}$ can be obtained. Using again the triangular inequality, we get

$$\left\|\mathscr{D}_x\hat{\mathscr{B}}(x)\right\| \leq \|\mathscr{D}_x\mathscr{B}(x)\| + \left\|\mathscr{D}_x\left(\hat{\mathscr{B}}(x) - \mathscr{B}(x)\right)\right\|. \tag{6.22}$$

The first term is bounded with constant $L_{\mathscr{B}}$ as the true operator is regular enough. The second term quantifies the error between the true and the approximated derivative. A VKOGA method with a Matérn kernel will give a rate [134]

$$\mathscr{N}^{-\nu/N+1/N}, \tag{6.23}$$

consistent with the fact that a differentiation decreases the convergence rate by a factor that scales as the fill distance $h \simeq \mathscr{N}^{-1/N}$.

Assuming an exponential decay for all the projection errors and the polynomial decays (6.21) and (6.23), we can write

$$\begin{aligned}\left\|\mathscr{F}(\lambda) - \hat{\mathscr{F}}(\lambda)\right\| \leq{}& C_1\exp\{-C_2M_{\mathscr{B}}\} + C_3\exp\{-C_4N_{\mathscr{B}}\} + C_5\mathscr{N}_{\mathscr{B}}^{-\nu/N_{\mathscr{B}}}\\ &+ C_6\exp\{-C_7M_{\mathscr{A}}\} + C_8\exp\{-C_9N_{\mathscr{A}}\} + C_{10}\mathscr{N}_{\mathscr{A}}^{-\nu/N_{\mathscr{A}}},\end{aligned} \tag{6.24}$$

where all the constants C_i depend on the true solution λ and the operators $\mathscr{A}$ and $\mathscr{B}$. Combining them, we can formally write

$$\left\|\mathscr{F}(\lambda) - \hat{\mathscr{F}}(\lambda)\right\| \leq K_1(\texttt{params}, \lambda, \mathscr{A}, \mathscr{B}),$$

where K_1 is an appropriate function of the relevant model parameters

$$\texttt{params} = (M_{\mathscr{B}}, N_{\mathscr{B}}, \mathscr{N}_{\mathscr{B}}, M_{\mathscr{A}}, N_{\mathscr{A}}, \mathscr{N}_{\mathscr{A}}),$$

the true solution λ and the operators $\mathscr{A}$ and $\mathscr{B}$. This bound is sufficient for our purposes, as it quantifies all the error contributions. As expected, provided that the kernel is sufficiently regular, the error asymptotically vanishes.

The second term in (6.19) is related to the application of the approximate mapping with two different inputs. Here, we aim to show that $\hat{\mathscr{F}}$ is a contraction mapping, which can be done by verifying that $\hat{\mathscr{F}}$ has bounded derivative $\mathscr{D}$, with a bound strictly inferior to 1. The following bound holds

$$\left\|\mathscr{D}_{\lambda}\hat{\mathscr{F}}(\lambda)\right\| \leq L_{\mathscr{F}} + K_2(\texttt{params}, \lambda, \mathscr{A}, \mathscr{B}),$$

so that

$$\left\|\hat{\mathscr{F}}(\lambda) - \hat{\mathscr{F}}(\hat{\lambda})\right\| \leq (L_{\mathscr{F}} + K_2)\left\|\lambda - \hat{\lambda}\right\|.$$

Here, $L_{\mathscr{F}} < 1$ as the true mapping is a contraction, otherwise (6.5) would not converge, and K_2 is an appropriate function of the parameters, the true solution λ and the operators $\mathscr{A}$ and $\mathscr{B}$. An explicit expression for K_2 can be obtained using arguments similar to (6.20), (6.22) and (6.23). Moreover, if the values of the hyperparameters are sufficiently large and the kernel is sufficiently smooth, the constant K_2 can be made arbitrarily small, similar to (6.24). Therefore, at least asymptotically, we have that $K_2 < 1 - L_{\mathscr{F}}$, so that the approximated mapping is a contraction.

Recalling (6.19), we get

$$\left\|\lambda - \hat{\lambda}\right\| \leq K_1(\texttt{params}, \lambda, \mathscr{A}, \mathscr{B}) + \big(L_{\mathscr{F}} + K_2(\texttt{params}, \lambda, \mathscr{A}, \mathscr{B})\big)\left\|\lambda - \hat{\lambda}\right\|,$$

and, in the case $K_2 < 1 - L_{\mathscr{F}}$, we obtain

$$\left\|\lambda - \hat{\lambda}\right\| \leq \frac{K_1(\texttt{params}, \lambda, \mathscr{A}, \mathscr{B})}{1 - L_{\mathscr{F}} - K_2(\texttt{params}, \lambda, \mathscr{A}, \mathscr{B})},$$

which is the desired estimate, as it depends on the relevant hyperparameters and the true solution λ.

7 Conclusion and future perspectives

This work was mainly motivated by recognizing the importance of efficient numerical simulations of coupled systems. This requires not only realistic differential models and accurate discretization schemes, but also the ability to perform multi-query simulations at a reasonable computational cost without compromising accuracy. This led to the interest in designing and applying model order reduction techniques specifically targeted to systems possessing a multi-component structure. The problem has been tackled from different angles, and this work provides contributions in each of the considered frameworks. The main conclusion that can be drawn is that, despite the complexity of the models and numerical solutions, the emergent behaviors observed with the high-fidelity models can be retrieved at the reduced level. The reduction does not severely compromise accuracy, and on average we observed a speedup of one to two orders of magnitude compared to the full order model. More specifically, we considered three scenarios.

Firstly, we assumed that the coupled problem can be used to construct the reduced models. In Chapter 3, we applied the POD-DEIM reduced basis method to the phase dynamics equations and a neuronal model of circadian oscillators. Because of the global interactions, a careful treatment of the coupling term is necessary to achieve the expected speedup. This is different from many differential models arising from a spatial discretization of PDEs, which typically enjoy a certain level of spatial locality. We showed that the reduced models are able to retrieve the synchronization induced by mutual coupling or external forcing. The observed computational speedup is satisfactory, at least if a large number of oscillators is considered and the coupling is treated appropriately. We also applied the reduction methods to a control problem, showing that the reduced models can be used in contexts that go beyond state reconstruction.

Secondly, we considered cases in which the coupled problem cannot be used to construct the reduced model, as it is not a-priori available (i.e., one does not know how the components will be assembled in practical applications) or it is too complex to simulate (i.e., large parameter spaces, large-scale models, nonlinearities, slow convergence of domain decomposition iterations). In turn, one has complete access to the local models. In Chapter 4 and Chapter 5 we extensively relied on an artificial parametrization of the boundary conditions, which allowed us to construct reduced models that are robust with respect to the input data. In Chapter 4 we showed that the surrogates are able to recover the synchronization patterns in a system that mimics the Huygens' setup involving coupled pendulum clocks. These include in-phase, anti-phase and clustering-like synchronization. We compared our technique with more classical model-driven approaches in structural dynamics, showing similar performances. Going beyond oscillatory systems, in Chapter 5 we showed that our approach can be generalized to a larger class of interface-coupled heterogeneous systems. We considered Robin coupling conditions, that usually outperform classical Dirichlet-Neumann methods, showing the accuracy, computational efficiency and robustness of the reduced models. Notably, we tackled problems ranging from linear and nonlinear diffusion-reaction to fluid-structure interaction models. We also provided a theoretical discussion on the accuracy and the computational complexity of our approach.

Finally, with a similar spirit, in Chapter 6 we considered a nonintrusive approach. Relying on component-wise black-box solvers and boundary response maps, we constructed local data-driven methods combining dimensionality reduction techniques with general-purpose interpolation and regression methods. We proposed two strategies to generate the training data, along the lines of the first two scenarios we described above, showing the potential and the drawbacks of both approaches. After a theoretical discussion on the reduction error and the computational cost, we applied these methods to steady and unsteady problems, which include a high degree of heterogeneity or geometrical complexity.

Although this work considers different problems and proposes various solutions, we believe that it can also represent the starting point for a number of future research directions.

The most natural extension would be a study of the applicability of the proposed methods in larger-scale, more heterogeneous problems. This could be viewed as a further step towards a potential final goal of incorporating the methods in scientific libraries or at an industrial level. In the localized model reduction context discussed mostly in Chapter 5, one could consider systems characterized by a larger number of components, higher spatial dimensions or a geometrical parametrization. Although our method can be extended to such cases in a relatively straightforward way, a different sampling of the artificial parameters could also be considered, including time-dependent boundary conditions and greedy methods. Similar extensions could also be considered in the nonintrusive context presented in Chapter 6. Besides these, applications to more complex time-dependent problems could be investigated in order to thoroughly test the nonintrusive method in such cases. We remark that these might require more sophisticated choices not only on the artificial parametrization, but also on the interpolation and regression methods, mostly to ensure a better generalization and reduce the curse of dimensionality. In both cases, although the problems we considered possess a significant level of heterogeneity, this can be further increased. Examples could be coupled electro-mechanical systems or thermo-hydrodynamical problems.

The efficiency of the methods has been mostly shown in terms of the reconstruction of the full-order solutions. In the spirit of Chapter 3, it would be interesting to use the techniques in applications and problems that go beyond a plain state reconstruction. Examples could be optimization problems and uncertainty quantification. In fluid-structure interaction models, one may be interested in studying the system response (e.g., lift/drag forces, pressure drop caused by an obstacle) under uncertainties in constitutive parameters. Additionally, optimization problems could be designed in order to find the optimal parameter values that minimize this response. An appropriate reduction of the adjoint problems should also be considered, which appears to be challenging in the context of localized models.

Effectively, all these problems add a higher level of complexity to the ones we considered. For this reason, the reduced basis method and the kernel interpolation methods might need to be replaced by more sophisticated techniques. An example would be the use of autoencoders in place of the SVD, which could be beneficial in the localized context, as the high dimensionality of the artificial parameter space can potentially reduce the efficiency of linear projection methods. Additionally, it might allow to bridge the gap between linear data-driven methods and nonlinear model-driven approaches, which are widely used in the Kuramoto model and its variants. An alternative option, lying at the intersection between Chapter 5 and Chapter 6, would be the use of interpolation or regression methods to approximate the local solution or an appropriate reduced representation in the entire subdomains, and reconstruct the boundary functional in a post-processing step. Such approaches have already been applied in a variety of problems, but the application in a fixed-point iterative scheme, where an additional boundary reconstruction is required, has not been considered yet, to the best of our knowledge. In turn, it has been shown that they can naturally be extended to incorporate the physics of the model in the training phase, which is arguably required in complex heterogeneous problems or when a limited number of training samples is available.

Bibliography

[1] B. M. Afkham, N. Ripamonti, et al. “Conservative Model Order Reduction for Fluid Flow”. In: *Quantification of Uncertainty: Improving Efficiency and Technology: QUIET selected contributions*. Ed. by M. D’Elia, M. Gunzburger, and G. Rozza. Cham: Springer International Publishing, 2020, pp. 67–99. DOI: 10.1007/978-3-030-48721-8_4.

[2] M. Aletti and D. Lombardi. “A Reduced Order representation of the Poincaré-Steklov operator: an application to coupled multi-physics problems”. In: *International Journal for Numerical Methods in Engineering* (2017).

[3] M. C. Aletti, S. Perotto, and A. Veneziani. “HiMod Reduction of Advection–Diffusion–Reaction Problems with General Boundary Conditions”. In: *Journal of Scientific Computing* 76.1 (2018), pp. 89–119. DOI: 10.1007/s10915-017-0614-5.

[4] A. Alla, A. Monti, and I. Sgura. *Adaptive POD-DEIM correction for Turing pattern approximation in reaction-diffusion PDE systems*. arXiv preprint arXiv:2203.05998. 2022.

[5] P. Astrid, S. Weiland, et al. “Missing Point Estimation in Models Described by Proper Orthogonal Decomposition”. In: *IEEE Transactions on Automatic Control* 53.10 (2008), pp. 2237–2251.

[6] F. R. Bach. *Learning theory from first principles*. 2020.

[7] S. Badia, F. Nobile, and C. Vergara. “Fluid–structure partitioned procedures based on Robin transmission conditions”. In: *Journal of Computational Physics* 227.14 (2008), pp. 7027–7051. DOI: https://doi.org/10.1016/j.jcp.2008.04.006.

[8] F. Ballarin, A. Manzoni, et al. “Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations”. In: *International Journal for Numerical Methods in Engineering* 102.5 (2015), pp. 1136–1161. DOI: https://doi.org/10.1002/nme.4772.

[9] J. Barnett, I. Tezaur, and A. Mota. *The Schwarz alternating method for the seamless coupling of nonlinear reduced order models and full order models*. arXiv preprint arXiv:2210.12551. 2022.

[10] M. Barrault, Y. Maday, et al. “An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations”. In: *Comptes Rendus Mathematique* 339.9 (2004), pp. 667 –672. DOI: https://doi.org/10.1016/j.crma.2004.08.006.

[11] V. Barthelmann, E. Novak, and K. Ritter. “High dimensional polynomial interpolation on sparse grids”. In: *Advances in Computational Mathematics* 12.4 (2000), pp. 273–288. DOI: 10.1023/A:1018977404843.

[12] S. Bernard, D. Gonze, et al. “Synchronization-Induced Rhythmicity of Circadian Oscillators in the Suprachiasmatic Nucleus”. In: *PLOS Computational Biology* 3.4 (Apr. 2007), pp. 1–13. DOI: 10.1371/journal.pcbi.0030068.

[13] K. Bhattacharya, B. Hosseini, et al. *Model Reduction and Neural Networks for Parametric PDEs.* arXiv preprint arXiv:2005.03180. 2021.

[14] P. Binev, A. Cohen, et al. “Convergence Rates for Greedy Algorithms in Reduced Basis Methods”. In: *SIAM Journal on Mathematical Analysis* 43.3 (2011), pp. 1457–1472. DOI: 10.1137/100795772.

[15] M. Breakspear, S. Heitmann, and A. Daffertshofer. “Generative Models of Cortical Oscillations: Neurobiological Implications of the Kuramoto Model”. In: *Frontiers in Human Neuroscience* 4 (2010), p. 190. DOI: 10.3389/fnhum.2010.00190.

[16] A. Buffa, Y. Maday, et al. “*A priori* convergence of the greedy algorithm for the parametrized reduced basis method”. In: *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique* 46.3 (2012), pp. 595–603. DOI: 10.1051/m2an/2011056.

[17] A. Buhr, L. Iapichino, et al. *Localized model reduction for parameterized problems.* arXiv preprint arXiv:1902.08300. 2019.

[18] A. Buhr and K. Smetana. “Randomized Local Model Order Reduction”. In: *SIAM Journal on Scientific Computing* 40.4 (2018), A2120–A2151. DOI: 10.1137/17M1138480.

[19] E. Burman and M. A. Fernández. “Stabilization of explicit coupling in fluid–structure interaction involving fluid incompressibility”. In: *Computer Methods in Applied Mechanics and Engineering* 198.5 (2009), pp. 766–784. DOI: https://doi.org/10.1016/j.cma.2008.10.012.

[20] C. Canuto, M. Hussaini, et al. *Spectral Methods: Fundamentals in Single Domains.* Scientific Computation. Springer Berlin Heidelberg, 2007.

[21] K. Carlberg, M. Barone, and H. Antil. “Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction”. In: *Journal of Computational Physics* 330 (2017), pp. 693–734. DOI: https://doi.org/10.1016/j.jcp.2016.10.033.

[22] K. Carlberg, S. Guzzetti, et al. *The network uncertainty quantification method for propagating uncertainties in component-based systems.* arXiv preprint arXiv:1908.11476. 2019.

[23] A. de Castro, P. Bochev, et al. *Explicit synchronous partitioned scheme for coupled reduced order models based on composite reduced bases.* arXiv preprint arXiv:2306.05531. 2023.

[24] A. de Castro, P. Kuberry, et al. *A Novel Partitioned Approach for Reduced Order Model – Finite Element Model (ROM-FEM) and ROM-ROM Coupling.* arXiv preprint arXiv:2206.04736. 2022.

[25] R. Cavoretto, A. De Rossi, and E. Perracchione. “Efficient computation of partition of unity interpolants through a block-based searching technique”. In: *Computers & Mathematics with Applications* 71.12 (2016), pp. 2568–2584. DOI: https://doi.org/10.1016/j.camwa.2016.04.021.

[26] S. Chaturantabut and D. C. Sorensen. “Nonlinear Model Reduction via Discrete Empirical Interpolation”. In: *SIAM Journal on Scientific Computing* 32.5 (2010), pp. 2737–2764. DOI: 10.1137/090766498.

[27] S. Chen, Z. Ding, et al. “A Reduced Order Schwarz Method for Nonlinear Multiscale Elliptic Equations Based on Two-Layer Neural Networks”. In: *Journal of Computational Mathematics* (Mar. 2023). DOI: 10.4208/jcm.2204-m2021-0311.

[28] W. Chen, Q. Wang, et al. “Physics-informed machine learning for reduced-order modeling of nonlinear problems”. In: *Journal of Computational Physics* 446 (2021), p. 110666. DOI: https://doi.org/10.1016/j.jcp.2021.110666.

[29] L. Cicci, S. Fresca, and A. Manzoni. “Deep-HyROMnet: A Deep Learning-Based Operator Approximation for Hyper-Reduction of Nonlinear Parametrized PDEs”. In: *Journal of Scientific Computing* 93.2 (2022), p. 57. DOI: 10.1007/s10915-022-02001-8.

[30] R. Craig. “Coupling of substructures for dynamic analyses - An overview”. In: *41st Structures, Structural Dynamics, and Materials Conference and Exhibit.* 2000. DOI: 10.2514/6.2000-1573.

[31] R. Craig and A. Kurdila. *Fundamentals of Structural Dynamics.* Wiley, 2006.

[32] K. Czolczynski, P. Perlikowski, et al. “Clustering and synchronization of n Huygens’ clocks”. In: *Physica A: Statistical Mechanics and its Applications* 388.24 (2009), pp. 5013–5023. DOI: https://doi.org/10.1016/j.physa.2009.08.033.

[33] K. Czolczynski, P. Perlikowski, et al. “Synchronization of the self-excited pendula suspended on the vertically displacing beam”. In: *Communications in Nonlinear Science and Numerical Simulation* 18.2 (2013), pp. 386–400. DOI: https://doi.org/10.1016/j.cnsns.2012.07.007.

[34] K. Czołczyński, P. Perlikowski, et al. “Clustering of Non-Identical Clocks”. In: *Progress of Theoretical Physics* 125.3 (Mar. 2011), pp. 473–490. DOI: 10.1143/PTP.125.473.

[35] B. C. Daniels, S. T. M. Dissanayake, and B. R. Trees. “Synchronization of coupled rotators: Josephson junction ladders and the locally coupled Kuramoto model”. In: *Phys. Rev. E* 67 (2 Feb. 2003), p. 026216. DOI: 10.1103/PhysRevE.67.026216.

[36] V. DeCaria, T. Iliescu, et al. “An Artificial Compression Reduced Order Model”. In: *SIAM Journal on Numerical Analysis* 58.1 (2020), pp. 565–589. DOI: 10.1137/19M1246444.

[37] S. Deparis, M. Discacciati, et al. “Fluid–structure algorithms based on Steklov–Poincaré operators”. In: *Computer Methods in Applied Mechanics and Engineering* 195.41 (2006). John H. Argyris Memorial Issue. Part II, pp. 5797–5812. DOI: https://doi.org/10.1016/j.cma.2005.09.029.

[38] S. Deparis, D. Forti, and A. Quarteroni. “A Rescaled Localized Radial Basis Function Interpolation on Non-Cartesian and Nonconforming Grids”. In: *SIAM Journal on Scientific Computing* 36.6 (2014), A2745–A2762. DOI: 10.1137/130947179.

[39] S. Deparis, A. Iubatti, and L. Pegolotti. “Coupling non-conforming discretizations of PDEs by spectral approximation of the Lagrange multiplier space”. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 53.5 (Sept. 2019), pp. 1667–1694. DOI: 10.1051/m2an/2019030.

[40] P. Diercks, K. Veroy, et al. *Multiscale modeling of linear elastic heterogeneous structures based on a localized model order reduction approach.* arXiv preprint arXiv:2201.10374. 2022.

[41] R. Dilão. “Antiphase and in-phase synchronization of nonlinear oscillators: The Huygens’s clocks system”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 19.2 (2009), p. 023118. DOI: 10.1063/1.3139117.

[42] M. Discacciati. “Domain decomposition methods for the coupling of surface and groundwater flows”. PhD thesis. École Polytechnique Fédérale de Lausanne, 2004.

[43] N. Discacciati and J. S. Hesthaven. “Localized model order reduction and domain decomposition methods for coupled heterogeneous systems”. In: *International Journal for Numerical Methods in Engineering* (2023). To appear. DOI: https://doi.org/10.1002/nme.7295.

[44] N. Discacciati and J. S. Hesthaven. “Modeling synchronization in globally coupled oscillatory systems using model order reduction”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.5 (2021), p. 053127. DOI: 10.1063/5.0031142.

[45] N. Discacciati, J. S. Hesthaven, and D. Ray. “Controlling oscillations in high-order Discontinuous Galerkin schemes using artificial viscosity tuned by neural networks”. In: *Journal of Computational Physics* 409 (2020), p. 109304. DOI: https://doi.org/10.1016/j.jcp.2020.109304.

[46] O. Dubois. “Optimized Schwarz Methods with Robin Conditions for the Advection-Diffusion Equation”. In: *Domain Decomposition Methods in Science and Engineering XVI.* Ed. by O. B. Widlund and D. E. Keyes. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 181–188.

[47] F. Dörfler and F. Bullo. “Synchronization in complex networks of phase oscillators: A survey”. In: *Automatica* 50.6 (2014), pp. 1539 –1564. DOI: https://doi.org/10.1016/j.automatica.2014.04.012.

[48] J. L. Eftang and A. T. Patera. “A port-reduced static condensation reduced basis element method for large component-synthesized structures: approximation and A Posteriori error estimation”. In: *Advanced Modeling and Simulation in Engineering Sciences* 1.1 (2014), p. 3. DOI: 10.1186/2213-7467-1-3.

[49] E. Ellis William. “On Sympathetic Influence between Clocks”. In: *Monthly Notices of the Royal Astronomical Society* 33.8 (June 1873), pp. 480–484. DOI: 10.1093/mnras/33.8.480.

[50] C. Farhat, P. Avery, et al. “Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency”. In: *International Journal for Numerical Methods in Engineering* 98.9 (2014), pp. 625–662. DOI: https://doi.org/10.1002/nme.4668.

[51] C. Farhat and F.-X. Roux. “A method of finite element tearing and interconnecting and its parallel solution algorithm”. In: *International Journal for Numerical Methods in Engineering* 32.6 (1991), pp. 1205–1227. DOI: https://doi.org/10.1002/nme.1620320604.

[52] R. H. B. Fey, D. H. van Campen, and A. de Kraker. “Long Term Structural Dynamics of Mechanical Systems With Local Nonlinearities”. In: *Journal of Vibration and Acoustics* 118.2 (Apr. 1996), pp. 147–153. DOI: 10.1115/1.2889642.

[53] N. R. Franco, S. Fresca, et al. *Approximation bounds for convolutional neural networks in operator learning*. arXiv preprint arXiv:2207.01546. 2023.

[54] S. Fresca, L. Dede’, and A. Manzoni. “A Comprehensive Deep Learning-Based Approach to Reduced Order Modeling of Nonlinear Time-Dependent Parametrized PDEs”. In: *Journal of Scientific Computing* 87.2 (2021), p. 61. DOI: 10.1007/s10915-021-01462-7.

[55] G. Gambino, M. C. Lombardo, et al. “Pattern selection in the 2D FitzHugh–Nagumo model”. In: *Ricerche di Matematica* 68.2 (2019), pp. 535–549. DOI: 10.1007/s11587-018-0424-6.

[56] M. J. Gander and O. Dubois. “Optimized Schwarz methods for a diffusion problem with discontinuous coefficient”. In: *Numerical Algorithms* 69.1 (2015), pp. 109–144. DOI: 10.1007/s11075-014-9884-2.

[57] M. J. Gander and L. Halpern. “Méthodes de relaxation d’ondes (SWR) pour l’équation de la chaleur en dimension 1”. In: *Comptes Rendus Mathematique* 336.6 (2003), pp. 519–524. DOI: https://doi.org/10.1016/S1631-073X(03)00009-8.

[58] M. J. Gander and Y. Liu. “On the Definition of Dirichlet and Neumann Conditions for the Biharmonic Equation and Its Impact on Associated Schwarz Methods”. In: *Domain Decomposition Methods in Science and Engineering XXIII*. Ed. by C.-O. Lee, X.-C. Cai, et al. Cham: Springer International Publishing, 2017, pp. 303–311.

[59] M. J. Gander, L. Halpern, and F. Nataf. “Optimal convergence for overlapping and non-overlapping Schwarz waveform relaxation”. In: 11th international conference on domain decomposition methods. 1999, pp. 27–36.

[60] M. Geist, P. Petersen, et al. *Numerical Solution of the Parametric Diffusion Equation by Deep Neural Networks*. arXiv preprint arXiv:2004.12131. 2020.

[61] P. Gervasio and A. Quarteroni. “INTERNODES for Heterogeneous Couplings”. In: *Domain Decomposition Methods in Science and Engineering XXIV*. Ed. by P. E. Bjørstad, S. C. Brenner, et al. Cham: Springer International Publishing, 2018, pp. 59–71.

[62] D. Gonze, S. Bernard, et al. “Spontaneous synchronization of coupled circadian oscillators”. In: *Biophysical journal* 89.1 (July 2005), pp. 120–129. DOI: 10.1529/biophysj.104.058388.

[63] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[64] G. A. Gottwald. “Model reduction for networks of coupled oscillators”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 25.5 (2015), p. 053111. DOI: 10.1063/1.4921295.

[65] M. Guo and J. S. Hesthaven. “Reduced order modeling for nonlinear structural analysis using Gaussian process regression”. In: *Computer Methods in Applied Mechanics and Engineering* 341 (2018), pp. 807–826. DOI: https://doi.org/10.1016/j.cma.2018.07.017.

[66] B. Haasdonk, H. Kleikamp, et al. *A new certified hierarchical and adaptive RB-ML-ROM surrogate model for parametrized PDEs*. arXiv preprint arXiv:2204.13454. 2022.

[67] Haasdonk, Bernard. “Convergence Rates of the POD-Greedy Method”. In: *ESAIM: M2AN* 47.3 (2013), pp. 859–873. DOI: 10.1051/m2an/2012045.

[68] Haasdonk, Bernard and Ohlberger, Mario. “Reduced basis method for finite volume approximations of parametrized linear evolution equations”. In: *ESAIM: M2AN* 42.2 (2008), pp. 277–302. DOI: 10.1051/m2an:2008001.

[69] J. S. Hesthaven, C. Pagliantini, and N. Ripamonti. *Rank-adaptive structure-preserving reduced basis methods for Hamiltonian systems*. arXiv preprint arXiv:2007.13153. 2020.

[70] J. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. SpringerBriefs in Mathematics. Springer International Publishing, 2015.

[71] J. Hesthaven and S. Ubbiali. “Non-intrusive reduced order modeling of nonlinear problems using neural networks”. In: *Journal of Computational Physics* 363 (2018), pp. 55–78. DOI: https://doi.org/10.1016/j.jcp.2018.02.037.

[72] C. Huygens. *Oeuvres complètes de Christiaan Huygens. Publiées par la Société hollandaise des sciences*. Vol. 1. https://www.biodiversitylibrary.org/bibliography/21031. La Haye, M. Nijhoff, 1888-1950, p. 658.

[73] D. B. P. Huynh, D. J. Knezevic, and A. T. Patera. “A static condensation reduced basis element method: Complex problems”. In: *Computer Methods in Applied Mechanics and Engineering* 259 (2013), pp. 197–216. DOI: https://doi.org/10.1016/j.cma.2013.02.013.

[74] D. B. P. Huynh, D. J. Knezevic, and A. T. Patera. “A Static condensation Reduced Basis Element method : approximation and a posteriori error estimation”. In: *ESAIM: M2AN* 47.1 (2013), pp. 213–251. DOI: 10.1051/m2an/2012022.

[75] L. Iapichino, A. Quarteroni, and G. Rozza. “Reduced basis method and domain decomposition for elliptic problems in networks and complex parametrized geometries”. In: *Computers and Mathematics with Applications* 71.1 (2016), pp. 408–430. DOI: https://doi.org/10.1016/j.camwa.2015.12.001.

[76] A. Iollo, G. Sambataro, and T. Taddei. “A one-shot overlapping Schwarz method for component-based model reduction: application to nonlinear elasticity”. In: *Computer Methods in Applied Mechanics and Engineering* 404 (2023), p. 115786. DOI: https://doi.org/10.1016/j.cma.2022.115786.

[77] V. Jovanovic and S. Koshkin. “Synchronization of Huygens' clocks and the Poincaré method”. In: *Journal of Sound and Vibration* 331.12 (June 2012), pp. 2887–2900. DOI: 10.1016/j.jsv.2012.01.035.

[78] M. Khamlich, F. Pichi, and G. Rozza. “Model order reduction for bifurcating phenomena in fluid-structure interaction problems”. In: *International Journal for Numerical Methods in Fluids* 94.10 (2022), pp. 1611–1640. DOI: https://doi.org/10.1002/fld.5118.

[79] M. Komarov and A. Pikovsky. “The Kuramoto model of coupled oscillators with a biharmonic coupling function”. In: *Physica D: Nonlinear Phenomena* 289 (2014), pp. 18 –31. DOI: https://doi.org/10.1016/j.physd.2014.09.002.

[80] N. Komin, A. C. Murza, et al. “Synchronization and entrainment of coupled circadian oscillators”. In: *Interface Focus* 1.1 (2011), pp. 167–176. DOI: 10.1098/rsfs.2010.0327.

[81] D. Krattiger, L. Wu, et al. “Interface reduction for Hurty/Craig-Bampton substructured models: Review and improvements”. In: *Mechanical Systems and Signal Processing* 114 (2019), pp. 579–603. DOI: https://doi.org/10.1016/j.ymssp.2018.05.031.

[82] Y. Kuramoto. “Self-entrainment of a population of coupled nonlinear oscillators”. In: *International Symposium on Mathematical Problems in Theoretical Physics, Lecture Notes in Physics, Vol. 39,* ed. by H. Araki. New York, NY, USA: Springer, 1975, pp. 420–422.

[83] A. van Lamsweerde. “Parametric model order reduction and sensitivity analysis for the dynamics of circadian clocks”. MSc thesis. Katholieke Universiteit Leuven, 2021.

[84] K. Lee and K. T. Carlberg. “Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders”. In: *Journal of Computational Physics* 404 (2020), p. 108973. DOI: https://doi.org/10.1016/j.jcp.2019.108973.

[85] K. Li, S. Ma, et al. “Transition to synchronization in a Kuramoto model with the first- and second-order interaction terms”. In: *Phys. Rev. E* 89 (3 Mar. 2014), p. 032917. DOI: 10.1103/PhysRevE.89.032917.

[86] P.-L. Lions. “On the Schwarz alternating method. III: a variant for nonoverlapping subdomains”. In: Third international symposium on domain decomposition methods for partial differential equations. SIAM Philadelphia. 1990, pp. 202–223.

[87] Løvgren, Alf Emil, Maday, Yvon, and Rønquist, Einar M. “A reduced basis element method for the steady Stokes problem”. In: *ESAIM: M2AN* 40.3 (2006), pp. 529–552. DOI: 10.1051/m2an:2006021.

[88] Y. Maday and E. M. Ronquist. “The Reduced Basis Element Method: Application to a Thermal Fin Problem”. In: *SIAM Journal on Scientific Computing* 26.1 (2004), pp. 240–258. DOI: 10.1137/S1064827502419932.

[89] Y. Maday and E. M. Rønquist. “A Reduced-Basis Element Method”. In: *Journal of Scientific Computing* 17.1-4 (2002), pp. 447–459. DOI: 10.1023/a:1015197908587.

[90] H. Montanelli and Q. Du. “New Error Bounds for Deep ReLU Networks Using Sparse Grids”. In: *SIAM Journal on Mathematics of Data Science* 1.1 (2019), pp. 78–92. DOI: 10.1137/18M1189336.

[91] H. Nakao. “Phase reduction approach to synchronisation of nonlinear oscillators”. In: *Contemporary Physics* 57.2 (2016), pp. 188–214. DOI: 10.1080/00107514.2015.1094987.

[92] S. Nkomo. *Synchronization Behavior in Coupled Chemical Oscillators.* 2014.

[93] M. Nonino, F. Ballarin, et al. “Projection Based Semi-Implicit Partitioned Reduced Basis Method for Fluid-Structure Interaction Problems”. In: *Journal of Scientific Computing* 94.1 (2022), p. 4. DOI: 10.1007/s10915-022-02049-6.

[94] M. Ohlberger and S. Rave. *Reduced Basis Methods: Success, Limitations and Future Challenges.* arXiv preprint arXiv:1511.02021v2. 2015.

[95] E. Ott and T. M. Antonsen. “Low dimensional behavior of large systems of globally coupled oscillators”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 18.3 (2008), p. 037113. DOI: 10.1063/1.2930766.

[96] Y. N. Ovchinnikov and V. Z. Kresin. “Networks of Josephson junctions and their synchronization”. In: *Physical Review B* 88.21 (Dec. 2013). DOI: 10.1103/physrevb.88.214504.

[97] J. Pantaleone. “Synchronization of metronomes”. In: *American Journal of Physics* 70.10 (2002), pp. 992–1000. DOI: 10.1119/1.1501118.

[98] L. Pegolotti, M. R. Pfaller, et al. “Model order reduction of flow based on a modular geometrical approximation of blood vessels”. In: *Computer Methods in Applied Mechanics and Engineering* 380 (2021), p. 113762. DOI: https://doi.org/10.1016/j.cma.2021.113762.

[99] B. Peherstorfer and K. Willcox. “Data-driven operator inference for nonintrusive projection-based model reduction”. In: *Computer Methods in Applied Mechanics and Engineering* 306 (2016), pp. 196–215. DOI: https://doi.org/10.1016/j.cma.2016.03.025.

[100] J. Peña Ramirez, R. Fey, et al. “An improved model for the classical Huygens’ experiment on synchronization of pendulum clocks”. In: *Journal of Sound and Vibration* 333.26 (2014), pp. 7248 –7266. DOI: https://doi.org/10.1016/j.jsv.2014.08.030.

[101] A. Pikovsky, M. G. Rosenblum, and J. Kurths. *Synchronization, A Universal Concept in Nonlinear Sciences.* Cambridge: Cambridge University Press, 2001.

[102] A. Pikovsky and M. Rosenblum. “Dynamics of globally coupled oscillators: Progress and perspectives”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 25.9 (2015), p. 097616. DOI: 10.1063/1.4922971.

[103] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction.* UNITEXT. Springer International Publishing, 2015.

[104] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations.* Numerical Mathematics and Scie. Clarendon Press, 1999.

[105] A. Quarteroni. *Numerical Models for Differential Problems.* Springer Milan, 2014. DOI: 10.1007/978-88-470-5522-3.

[106] C. E. Rasmussen. “Gaussian Processes in Machine Learning”. In: *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures.* Ed. by O. Bousquet, U. von Luxburg, and G. Rätsch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 63–71. DOI: 10.1007/978-3-540-28650-9_4.

[107] I. Rocha, F. van der Meer, and L. Sluys. “An adaptive domain-based POD/ECM hyper-reduced modeling framework without offline training”. In: *Computer Methods in Applied Mechanics and Engineering* 358 (2020), p. 112650. DOI: https://doi.org/10.1016/j.cma.2019.112650.

[108] A. Roup, D. Bernstein, et al. “Limit cycle analysis of the verge and foliot clock escapement using impulsive differential equations and Poincare maps”. In: *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148).* Vol. 4. 2001, 3245–3250 vol.4. DOI: 10.1109/ACC.2001.946422.

[109] F.-X. Roux. “Domain Decomposition Methodology with Robin Interface Matching Conditions for Solving Strongly Coupled Problems”. In: *Computational Science – ICCS 2008.* Ed. by M. Bubak, G. D. van Albada, et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 311–320.

[110] A. Rudi, L. Carratino, and L. Rosasco. *FALKON: An Optimal Large Scale Kernel Method.* arXiv preprint arXiv:1705.10958. 2018.

[111] B. Rüth, B. Uekermann, et al. “Quasi-Newton waveform iteration for partitioned surface-coupled multiphysics applications”. In: *International Journal for Numerical Methods in Engineering* 122.19 (Aug. 2020), pp. 5236–5257. DOI: 10.1002/nme.6443.

[112] A. Saltelli, M. Ratto, et al. “Variance-Based Methods”. In: *Global Sensitivity Analysis. The Primer.* John Wiley & Sons, Ltd, 2007. Chap. 4, pp. 155–182. DOI: https://doi.org/10.1002/9780470725184.ch4.

[113] G. Santin and B. Haasdonk. “Convergence rate of the data-independent P-greedy algorithm in kernel-based approximation”. In: *Dolomites Research Notes on Approximation* 10.06/2017 (2017), 68–78. DOI: 10.14658/pupj-drna-2017-Special_Issue-9.

[114] G. Santin and B. Haasdonk. “Kernel methods for surrogate modeling”. In: *System- and Data-Driven Methods and Algorithms*. De Gruyter, Oct. 2021, pp. 311–354. DOI: 10.1515/9783110498967-009.

[115] R. Sarfati, J. C. Hayes, and O. Peleg. “Self-organization in natural swarms of *Photinus carolinus* synchronous fireflies”. In: *Science Advances* 7.28 (2021), eabg9259. DOI: 10.1126/sciadv.abg9259.

[116] R. Sarfati and O. Peleg. “Chimera states among synchronous fireflies”. In: *Science Advances* 8.46 (2022), eadd6690. DOI: 10.1126/sciadv.add6690.

[117] P. J. Schmid. “Dynamic mode decomposition of numerical and experimental data”. In: *Journal of Fluid Mechanics* 656 (2010), 5–28. DOI: 10.1017/S0022112010001217.

[118] P. S. Skardal and J. G. Restrepo. *Synchronization of Kuramoto oscillators in networks of networks*. arXiv preprint arXiv:1206.3822. 2012.

[119] K. Smetana and T. Taddei. *Localized model reduction for nonlinear elliptic partial differential equations: localized training, partition of unity, and adaptive enrichment*. arXiv preprint arXiv:2202.09872. 2022.

[120] V. Spaiser, S. Ranganathan, et al. “The Dynamics of Democracy, Development and Cultural Values”. In: *PLOS ONE* 9.6 (June 2014), pp. 1–11. DOI: 10.1371/journal.pone.0097856.

[121] G. Stabile and G. Rozza. “Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations”. In: *Computers & Fluids* 173 (2018), pp. 273–284. DOI: https://doi.org/10.1016/j.compfluid.2018.01.035.

[122] T. Stankovski, T. Pereira, et al. “Coupling functions: Universal insights into dynamical interaction mechanisms”. In: *Reviews of Modern Physics* 89.4 (Nov. 2017). DOI: 10.1103/revmodphys.89.045001.

[123] R. Ştefănescu, A. Sandu, and I. M. Navon. “Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations”. In: *International Journal for Numerical Methods in Fluids* 76.8 (Aug. 2014), 497–521. DOI: 10.1002/fld.3946.

[124] S. H. Strogatz. “From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators”. In: *Physica D: Nonlinear Phenomena* 143.1–4 (2000), pp. 1 –20. DOI: 10.1016/S0167-2789(00)00094-4.

[125] R. Tenderini, N. Mueller, and S. Deparis. *Space-time reduced basis methods for parametrized unsteady Stokes equations*. arXiv preprint arXiv:2206.12198. 2022.

[126] P. F. Tilles, H. A. Cerdeira, and F. F. Ferreira. “Local attractors, degeneracy and analyticity: Symmetry effects on the locally coupled Kuramoto model”. In: *Chaos, Solitons & Fractals* 49 (2013), pp. 32–46. DOI: https://doi.org/10.1016/j.chaos.2013.02.008.

[127] M. Toiya, H. O. González-Ochoa, et al. “Synchronization of Chemical Micro-oscillators”. In: *The Journal of Physical Chemistry Letters* 1.8 (Apr. 2010), pp. 1241–1246. DOI: 10.1021/jz100238u.

[128] I. T. Tokuda, Z. Levnajic, and K. Ishimura. “A practical method for estimating coupling functions in complex dynamical systems”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 377.2160 (2019), p. 20190015. DOI: 10.1098/rsta.2019.0015.

[129] S. Turek and J. Hron. “Proposal for Numerical Benchmarking of Fluid-Structure Interaction between an Elastic Object and Laminar Incompressible Flow”. In: *Fluid-Structure Interaction*. Ed. by H.-J. Bungartz and M. Schäfer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 371–385.

[130] “Turing patterns, 70 years later”. In: *Nature Computational Science* 2.8 (2022), pp. 463–464. DOI: 10.1038/s43588-022-00306-0.

[131] Q. Wang, N. Ripamonti, and J. S. Hesthaven. “Recurrent neural network closure of parametric POD-Galerkin reduced-order models based on the Mori-Zwanzig formalism”. In: *Journal of Computational Physics* 410 (2020), p. 109402. DOI: https://doi.org/10.1016/j.jcp.2020.109402.

[132] Z. Wang and Z. Liu. “A Brief Review of Chimera State in Empirical Brain Networks”. In: *Frontiers in Physiology* 11 (2020). DOI: 10.3389/fphys.2020.00724.

[133] S. Watanabe and S. H. Strogatz. “Integrability of a globally coupled oscillator array”. In: *Phys. Rev. Lett.* 70 (16 Apr. 1993), pp. 2391–2394. DOI: 10.1103/PhysRevLett.70.2391.

[134] H. Wendland. *Scattered Data Approximation*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511617539.

[135] T. Wenzel, G. Santin, and B. Haasdonk. “Analysis of Target Data-Dependent Greedy Kernel Algorithms: Convergence Rates for f-, $f \cdot P$- and f/P-Greedy”. In: *Constructive Approximation* 57.1 (2023), pp. 45–74. DOI: 10.1007/s00365-022-09592-3.

[136] K. Wiesenfeld, P. Colet, and S. H. Strogatz. “Synchronization Transitions in a Disordered Josephson Series Array”. In: *Phys. Rev. Lett.* 76 (3 Jan. 1996), pp. 404–407. DOI: 10.1103/PhysRevLett.76.404.

[137] L. Wu, P. Tiso, and F. van Keulen. “Interface Reduction with Multilevel Craig–Bampton Substructuring for Component Mode Synthesis”. In: *AIAA Journal* 56.5 (2018), pp. 2030–2044. DOI: 10.2514/1.J056196.

[138] D. Xiao, P. Yang, et al. “Non-intrusive reduced order modelling of fluid–structure interactions”. In: *Computer Methods in Applied Mechanics and Engineering* 303 (2016), pp. 35–54. DOI: https://doi.org/10.1016/j.cma.2015.12.029.

[139] C. Xu, H. Xiang, et al. “Collective dynamics of identical phase oscillators with high-order coupling”. In: *Scientific Reports* 6.1 (2016), p. 31133. DOI: 10.1038/srep31133.

[140] J. Xu, C. Huang, and K. Duraisamy. “Reduced-Order Modeling Framework for Combustor Instabilities Using Truncated Domain Training”. In: *AIAA Journal* 58.2 (2020), pp. 618–632. DOI: 10.2514/1.J057959.

[141] M. Yano and A. T. Patera. “An LP empirical quadrature procedure for reduced basis treatment of parametrized nonlinear PDEs”. In: *Computer Methods in Applied Mechanics and Engineering* 344 (2019), pp. 1104–1123. DOI: https://doi.org/10.1016/j.cma.2018.02.028.

[142] G. F. Young, L. Scardovi, et al. “Starling Flock Networks Manage Uncertainty in Consensus at Low Cost”. In: *PLOS Computational Biology* 9.1 (Jan. 2013), pp. 1–7. DOI: 10.1371/journal.pcbi.1002894.

[143] J. Yu, C. Yan, and M. Guo. “Non-intrusive reduced-order modeling for fluid problems: A brief review”. In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 233.16 (2019), pp. 5896–5912. DOI: 10.1177/0954410019890721.

[144] E. Zappon, A. Manzoni, and A. Quarteroni. “Efficient and certified solution of parametrized one-way coupled problems through DEIM-based data projection across non-conforming interfaces”. In: *Advances in Computational Mathematics* 49.2 (2023), p. 21. DOI: 10.1007/s10444-022-10008-w.

[145] E. Zappon, A. Manzoni, et al. *A reduced order model for domain decompositions with non-conforming interfaces.* arXiv preprint arXiv:2206.09618. 2022.

[146] J. Zhang, S. Boccaletti, et al. “Synchronization of phase oscillators under asymmetric and bimodal distributions of natural frequencies”. In: *Chaos, Solitons & Fractals* 136 (2020), p. 109777. DOI: https://doi.org/10.1016/j.chaos.2020.109777.

Curriculum Vitae

Personal data

Name	Niccolò Discacciati
Date of birth	May 13, 1993
Nationality	Italian

Education

2019 - 2023	**PhD in Mathematics** École polytechnique fédérale de Lausanne, Switzerland Thesis advisor: Prof. J. S. Hesthaven
2015 - 2018	**MSc in Computational Science and Engineering** École polytechnique fédérale de Lausanne, Switzerland **MSc in Mathematical Engineering** Politecnico di Milano, Italy Thesis advisors: Prof. J. S. Hesthaven, Prof. N. Parolini, Dr. D. Ray
2012 - 2015	**BSc in Mathematical Engineering** Politecnico di Milano, Italy Thesis advisor: Prof. M. Verri

Work experience

2017	**Internship in Scientific Computing** Swiss National Supercomputing Centre (CSCS), Lugano, Switzerland Supervisor: Dr. W. Sawyer

Publications

- N. Discacciati and J. S. Hesthaven. *Model reduction of coupled systems based on non-intrusive approximations of the boundary response maps*. In preparation. 2023.

- N. Discacciati and J. S. Hesthaven. "Localized model order reduction and domain decomposition methods for coupled heterogeneous systems". In: *International Journal for Numerical Methods in Engineering* (2023). To appear. DOI: https://doi.org/10.1002/nme.7295.

- K. Szenes, N. Discacciati, L. Bonaventura, and W. Sawyer. *Domain-specific implementation of high order Discontinuous Galerkin methods in spherical geometry*. arXiv preprint arXiv:2303.11767. 2023.

- N. Discacciati and J. S. Hesthaven. "Modeling synchronization in globally coupled oscillatory systems using model order reduction". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.5 (2021), p. 053127. DOI: 10.1063/5.0031142.

- N. Discacciati, J. S. Hesthaven, and D. Ray. "Controlling oscillations in high-order Discontinuous Galerkin schemes using artificial viscosity tuned by neural networks". In: *Journal of Computational Physics* 409 (2020), p. 109304. DOI: https://doi.org/10.1016/j.jcp.2020.109304.

Conferences

- *Predicting emergent behaviors in coupled systems using localized reduced order models*, SIAM Conference in Computational Science and Engineering, Amsterdam (The Netherlands), 26 February - 03 March 2023.

- *Localized model order reduction for coupled heterogeneous systems*, Young Mathematicians in Model Order Reduction Conference, Münster (Germany), 18 - 22 July 2022.

- *Localized model order reduction for coupled systems*, 17th Copper Mountain Conference on Iterative Methods, Virtual Event, 04 - 08 April 2022.

- *Model order reduction and sensitivity analysis for the dynamics of circadian clocks*, Swiss Numerics Day, Lausanne (Switzerland), 13 September 2021.

- *Modeling synchronization in coupled oscillatory systems using model order reduction*, VI ECCOMAS Young Investigators Conference, Virtual Event, 07 - 09 July 2021.

- *Modeling synchronization in globally coupled oscillatory systems using model order reduction* (poster), Model Order Reduction Summer School, Virtual Event, 07 - 10 September 2020.

Distinctions

- Secretary of the EPFL Chapter of SIAM (2019 - 2023).
- SIAM Student Travel Award for the 2023 SIAM Conference on Computational Science and Engineering.
- Junior organizer of the Model Order Reduction Summer School 2020.
- ‘Carlo Cercignani’ award issued by Politecnico di Milano, 2018.
- ‘Best freshmen’ award issued by Politecnico di Milano, 2013.

Teaching

Co-supervised MSc Master Project:

- A. van Lamsweerde, *Parametric model order reduction and sensitivity analysis for the dynamics of circadian clocks*, Spring 2021.

Co-supervised MSc Semester Projects:

- L. Crijns, *Efficient Approximations of the Poincaré-Steklov Operator*, Spring 2022.
- B. Rodriguez Carrillo, *Effect of random parameters on model order reduction of coupled oscillatory systems*, Fall 2021.
- P. Vuillecard, *Detecting discontinuity orientation and location on two-dimensional grids using neural networks*, Spring 2020.

Teaching Assistant for the courses:

- Probabilités et statistique, BSc at EPFL/UNIL, Fall 2019, Spring 2020, Fall 2020, Spring 2021, Fall 2021, Fall 2022.
- Analyse IV, BSc at EPFL, Spring 2022.
- Analyse Numérique, BSc at EPFL, Spring 2019.