A few non linear approaches in model order reduction

Nicolas Cagniart

École Doctorale de Sciences Mathématiques de Paris Centre

Thèse de Doctorat

en vue de l'obtention du grade de

Docteur ès Sciences de l'Université Pierre et Marie Curie

Discipline: Mathématiques Appliquées

présentée par

Nicolas Cagniart

Quelques approches non linéaires en réduction de complexité

---

dirigée par Yvon Maday

Rapportée par

Francisco CHINESTA ENSAM
Bernard HAASDONK IANS

Soutenue le 05 Novembre 2018 devant le jury composé de

Francisco CHINESTA	ENSAM	Rapporteur
Virginie EHRLACHER	ENPC	Examinatrice
Edwige GODLEWSKI	LJLL	Examinatrice
Yvon MADAY	LJLL	Directeur de thèse
Tommaso TADDEI	INRIA	Examinateur

Quelques approches non linéaires en réduction de complexité

Nicolas CAGNIART

Remerciements

Cette section va être assez courte, non pas que je n'ai personne à remercier, mais parce qu'il faut que je prépare la soutenance !

Je remercie Prof. Chinesta et Prof. Haasdonk d'avoir accepté de rapporter la thèse. Je remercie Yvon de m'avoir permis d'effectuer cette thèse dans de bonnes conditions. Je le remercie aussi pour ses bonnes idées qui, distillées régulièrement, m'ont permis de progresser dans le sujet. Merci à (Dr) Roxana Crisovan pour la collaboration, et merci à Prof. Abgrall de l'avoir facilitée.

Je termine par le plus important assez sobrement. Merci à la famille , aux copains/copines du labo, aux copains/copines d'ailleurs pour les bons moments et pour la rigolade, indispensables pour finir une thèse.

Quelques approches non linéaires en réduction de complexité

Résumé

Les méthodes de réduction de modèles offrent un cadre général permettant une réduction de coûts de calculs substantielle pour les simulations numériques. Dans cette thèse, nous proposons d'étendre le domaine d'application de ces méthodes. Le point commun des sujets discutés est la tentative de dépasser le cadre standard «bases réduites» linéaires, qui ne traite que les cas où les variétés solutions ont une petite épaisseur de Kolmogorov. Nous verrons comment tronquer, translater, tourner, étirer, comprimer etc. puis recombiner les solutions, peut parfois permettre de contourner le problème qui se pose lorsque cette épaisseur de Kolmogorov n'est pas petite. Nous évoquerons aussi le besoin de méthodes de stabilisation sur-mesure pour le cadre réduit.

Mots clés : Réduction de modèles, décomposition de domaine, épaisseur de Kolmogorov, méthode de freezing, calibration, hyper-reduction, control optimal, stabilité de schémas numériques pour la mécanique des fluides, décomposition orthogonale en modes propres

A few non linear approaches in model order reduction

Abstract

Model reduction methods provide a general framework for substantially reducing computational costs of numerical simulations. In this thesis, we propose to extend the scope of these methods. The common point of the topics discussed here is the attempt to go beyond the standard linear "reduced basis" framework, which only deals with cases where the solution manifold have a small Kolmogorov width. We shall see how truncate, translate, rotate, stretch, compress etc. and then recombine the solutions, can sometimes help to overcome the problem when this Kolmogorov width is not small. We will also discuss the need for tailor-made stabilisation methods for the reduced frame.

Keywords Model order reduction, domain decomposition, Kolmogorov n-width, freezing method, calibration, hyper-reduction, optimal control, stabilization, proper orthogonal decomposition, Dynamic Mode Decomposition

Table of Contents

List of Figures

Avant-Propos

Cette thèse s'est effectuée au laboratoire Jacques Louis Lions (LJLL), à l'Université Pierre et Marie Curie (UPMC). Elle a été financée dans le cadre du project MECASIF, dont l'objectif était de rapprocher industriels et universitaires sur des problèmatiques de réduction de modèles. La thèse devait répondre à un problème soulevé par Bertin Technologies et s'est faite sous la direction d'Yvon MADAY.

Préambule

Le chapitre introductif commence par une description du problème cible, l'objectif que nous nous sommes fixé en début de thèse. Après avoir décrit brièvement une méthode type, empirique, utilisée aujourd'hui pour résoudre ce problème cible, nous motivons le besoin pour une approche plus rigoureuse, qui tienne compte de la très grande complexité du problème due aux nombreuses échelles spatiales impliquées. Dans cette direction, nous décrivons la méthode de bases réduites, qui répond simultanément aux deux contraintes: un cadre théorique rigoureux ainsi que des coûts de calculs réduits. La fin du chapitre introductif insiste sur le fait que certaines questions liées à l'application de méthodes type «bases réduites» au problème cible n'ont pas encore été résolues. C'est à quelques unes de ces questions que nous tentons de répondre dans les autres chapitres.

Dans le chapitre 2, nous étudions la facette «variabilité géometrique» du problème cible, et nous le faisons indépendamment des autres difficultés soulevées dans le chapitre introductif. Il apparait assez clair que cette problématique est proche des méthodes de décomposition de domaine, et que notre méthode devra être adaptée au contexte qui est celui de la réduction de modèles. Après avoir rappelé les quelques approches de la littérature proches de nos besoins, nous concluons sur la nécessité du développement d'une nouvelle méthode, plus flexible et proposons un cahier des charges. Le reste du chapitre est consacré à la description et à l'analyse d'une méthode répondant aux besoins fixés.

Dans le chapitre 3, nous cherchons de nouveaux outils à ajouter à la réduction de modèles standard et qui permettent de traiter des problèmes pour lesquels les variétés solutions ont une grande épaisseur de Kolmogorov. Nous commençons par décrire la méthode de Freezing, disponible dans la litterature et qui est une réponse possible, mais pas entièrement satisfaisante à la problématique du chapitre. Le reste du chapitre est consacré à la description d'une méthode alternative, la calibration. Des tests numériques sur l'équation de Burgers visqueux dans le cas périodique, viennent confirmer la viabilité de l'approche.

Dans le chapitre 4, nous appliquons la méthode de calibration à un problème plus réaliste: un écoulement autour d'un profil NACA. L'application de la calibration à ce problème qui est un problème en dimension deux, hyperbolique, et non périodique, pose de nouvelles difficultés que nous tentons de résoudre. De nombreux ingrédients sont nécessaires pour la résolution complète de ce problème. Par conséquent, la section numérique se concentre sur quelques aspects, ceux qui nous paraissent les plus importants.

La calibration dans les chapitres 3 et 4 est utilisée pour réduire l'épaisseur de Kolmogorov de variétés solutions. Le chapitre 5 part du constat que la calibration amène une propriété supplémentaire: elle ajoute à la régularité des solutions, comme fonction des paramètres. Ceci est vrai quelque soit le problème étudié. Un exemple particulièrement parlant, et celui que nous

avons choisi de traiter dans ce chapitre, est celui de problèmes hyperboliques, avec une forte dépendence de la position du choc aux paramètres. Nous décrivons quelques idées préliminaires vers le calcul des dérivées des solutions par rapport aux paramètres, une première étape vers la résolution de problèmes de control optimal dans ce contexte.

Les deux derniers petits chapitres de cette thèse sont un peu à part, mais sont le résultats de réflexions connexes à celles des chapitres principaux. Pour le calcul de la décomposition en modes propres orthogonaux pour des problèmes instationnaires et/ou des espaces de paramètres qui ne sont pas de toute petite dimension, il est intuitif de vouloir utiliser une méthode «diviser pour régner». C'est à la question de l'erreur due à une telle approche que nous tentons de répondre dans le chapitre 6.

Le point de départ du chapitre 7 est le constat que les méthodes inspirées par le big data rencontrent un intérêt croissant dans la communauté «bases réduites». Nous mentionnons quelques travaux récents et prometteurs dans cette direction. Nous nous arrêtons ensuite plus longuement sur la méthode DMD (Decomposition en Modes Dynamiques) et utilisons son analyse pour mettre en garde sur le fait qu'il existe des situations dans lesquelles une phase d'apprentissage «force brute» ne remplace pas une bonne modélisation, et ce quelle que soit la quantité de données assimilées.

Preamble[1]

The introductive chapter starts with a description of the target problem, the initial goal of the thesis. After briefly discussing existing empirical models, we motivate the need for an approach with a stronger theoretical background: we choose to use reduced order modeling. We give a quick overview of how ROM is usually performed, and insist on the properties a problem should satisfy in order to be a good ROM candidate. We then provide evidences showing that the target problem is not, at first glance, suitable for ROM, and that additional steps need to be performed.

In chapter 2, we study difficulties due to the geometry variability of the target problem. We work with a model problem, that helps isolating this one issue from the other mentioned in the introductive chapter. We then give a quick overview of the ROM methods available in the literature that are designed to handle geometry variations. We show that there is a need for a method more flexible than the existing ones. The remainder of the chapter is devoted to the development of such a method.

In chapter 3, we try to develop a method that allows for the use of ROM in the context of solution manifolds with large Kolmogorov n-widths. We describe the method of Freezing that is a interesting answer to this specific problem, but not entirely satisfactory in our opinion. The remainder of the chapter focuses on the description of an alternative method: the so-called calibration. Numerical experiments are performed on the periodic viscous Burgers equation, and tend to confirm the viability of the method.

In chapter 4, we apply the calibration method to a more realistic problem: the flow around a NACA airfoil. This leads to additional challenges, as the solutions present shocks, whose positions are strongly parameter dependent. Also, the domain is now two dimensional, and we are not in the favorable periodic setting. A complete numerical scheme is quite involved, and we have rather chosen to focus on a reduced number of issues, the ones we consider the most important.

The calibration is used in chapter 3 and 4 to diminish the Kolmogorov n-width of solution manifolds. Chapter 5 starts by noticing that calibration adds a bonus property: it improves the regularity of the dependency of the solutions with respect to the parameters. This is true whatever the problem studied. An especially enlightening example, and the one we have chosen to study in this chapter, is the one of hyperbolic problems, where the shock positions' sensitivity to parameter variations is high. We end the chapter by deriving a few preliminary results, towards the development of numerical schemes for the computation of parameter derivatives of the solutions.

[1] This a translation of Préambule.

The last two small chapters of this manuscript are a little different, but discuss connected topics. For the computation of proper orthogonal modes for time dependent problems and/or for parameter spaces with moderate dimensions, it is natural to apply a divide and conquer strategy. Chapter 6 gives bounds on the errors due to this kind of approach.

The starting point of chapter 7 was to notice that big data related ideas were receiving more and more attention in the reduced basis community. We start by mentioning some recent and promising results in that direction. We then discuss the DMD (Dynamic Mode Decomposition) method, and use its analysis to warn for inconsiderate use of such approaches. There are some cases, where brute force can not replace a good model, whatever the amount of data assimilated.

Chapter 1

Introduction

This first chapter is an extended introductory chapter. We start with a short description of the target problem, the original objective of this thesis. After briefly discussing existing empirical models, we motivate the need for an approach with a stronger theoretical background: we choose to use reduced order modeling. We give a quick overview on how ROM is usually performed, and insist on the properties a problem should satisfy in order to be a good ROM candidate. We then provide evidences showing that the target problem is not, at first glance, suitable for ROM, and that additional steps need to be performed. We finally detail how each of the chapters of this thesis is a new step towards the objective.

The first section states the starting point of the thesis. We briefly discuss the context in which this study takes place. The introductory section ends with a motivation for the use of Reduced Order Modeling (ROM). We then give an overview of the ROM framework. Instead of a general, heavy presentation, we have chosen to illustrate it on a simple heat equation. The end of this chapter is devoted to showing that there were (and still are) many ingredients missing to provide a complete ROM based solution to solve the initial objective. Following this remark, we have focused on simpler, more reasonable objectives. Some of them will seem far away from the initial goal. we consider them as steps, or elementary bricks, towards a sensible and robust solution to the initial objective.

1.1 Vertical axis wind turbine placement optimization

Wind farms have started catching attention for clean energy production for a few decades now. To get a grasp of their growing importance, one can for instance take a peak into the list of major European offshore wind farms, in [103]. The infatuations of the beginning have opened up to a more mature market where optimization of cost, production and maintenance hold a bigger place.

Whatever the precise objective function, these optimization problems have many layers of complexity. Indeed, for such problems, many length scales are involved, each having interacting influences. The scales identified a priori are: the scale of the boundary layer, the scale of the wing (shape, material etc.), the scale of the farm (for instance, the relative placement of the turbines), and finally the atmospheric scale (required for boundary conditions). The first two scales have already received some attention because of the many different applications among which aeronautics. It also benefits the fact that experiments in wind tunnels can easily be

programmed and conducted. The last two scales are more difficult to study, from a theoretical stand point as well as numerically and experimentally.

In this thesis, the focus has been put on optimization at the scale of the farm. The analysis of the measurements of working wind farms have shown that the average power loss due to the influence of upstream turbines averages 10% to 20%. It is also shown that in the wake, one finds higher level of turbulence, and thus higher loads on the turbines, which means higher maintenance costs. Many different configurations have been and are currently being tested. We mention two of them: in [15], the authors describe a bow shaped windfarm. A more natural 'grid' type farm is studied in [56]. The question that has naturally risen is: is there a way to diminish the wake effect as well as the load imposed on the turbines by adjusting the placement of the turbines ?

The (very ambitious) objective of the thesis was to give an optimized wind farm layout given the shape (or model) of a turbine, some constraints on the positioning of turbines and some other physical parameters (such as boundary and initial conditions). We have restricted the study to offshore windfarms and vertical wind turbines. This provides with useful assumptions: we do not have to take into account the topography. Also, vertical wind turbines allow for the use of a cylindrical symmetry. The turbines are not sensible to the inflow direction, and the only parameters for a given turbine shape are the direction and speed of rotation.

We note right away that even if the optimization only takes place at the scale of the farm, the behavior at other scales aforementioned still need to be modeled. In other words, the computation for one specific configuration is already a challenging task. But the situation is even worse: the objective being optimization, we are in a many-query context. This means that a focus needs to be put on the computational cost of the method.

The majority of the literature on the computation of the power output of wind farms uses rough empirical models[1]. We start our journey by giving a very quick overview on how these models are constructed, and on the theoretical results they rely on. We then give the major drawbacks of these approaches. This serves as a justification for the central discussion of this thesis: how to use the theory of Partial Differential Equations (PDE) and Reduced Order Modeling (ROM) to solve the optimization problem at hand.

A complete state of the art of the engineering methods currently being used to estimate a priori the output of a wind turbine farm is out of the scope of this thesis. We refer to [38, 55] for surveys, and have rather chosen to briefly present one 'generic engineering method'. We underline some of its flaws and thus motivate the need for more advanced models. We also take this opportunity to once again highlight the overall complexity of the task at hand.

The first component, common to all rough engineering models, is a wake deficit to power output relation. We describe one possible route. Let some cylindrical control volume around the rotor, with the longitudinal axis in the direction of the flow. The process is illustrated in Figure 1.1. We apply the conservation of momentum equation and suppose that the viscous term, the pressure term, basically all terms except the convective term and the force due to the turbine are negligible. Some hypotheses can be justified by physical considerations and we refer to the aforementioned articles for more details. We denote with u the velocity, with T a source term that represents the force acting on the turbine, with ρ the density and with ω the cylindrical control volume of interest. We end up with a relation such as:

$$\int_{\omega} \nabla \cdot (\rho u \otimes u) \approx T.$$

Denote u^{in} and u^{out} respectively the inflow and the outflow. We apply the divergence theorem

[1] Big-data approaches to solve CFD problems will be discussed in chapter 7.

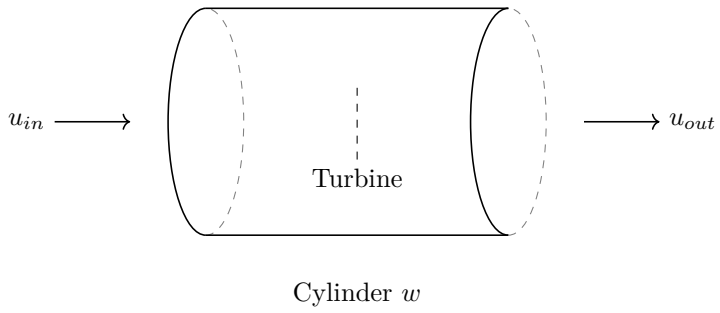


Figure 1.1: One possible starting point for engineering models

and assume rotational invariance. Let $\partial\omega^{sec}$ be one section of ω. At first order in $u^{in} - u^{out}$, we have:

$$\int_{\partial\omega^{sec}} \rho u^{out}(u^{in} - u^{out}) \approx T. \tag{1.1}$$

We can see how, by placing such models end to end, one can derive a first naive way of estimating the power output of wind farms with aligned turbines.

To get more advanced models and results for more general configurations, one needs to add other components to try and account for other various physical phenomenon. We mention a few of them, but there are as many variants as there are paper on the topic. For instance, one needs to model the wake geometry, or equivalently a procedure to select the downstream turbines that are being impacted by a given wake. In [82] they try using a reasoning similar to the one that lead to equation (1.1) to account for the interaction between wakes. Other models have been constructed to estimate the power loss due to an increase in the turbulence intensities, see for instance [15]. This type of rough modeling is currently being used on real life data. The numerical results presented in the literature often compare the energy output of the model with experimental data. Thanks to the extra degrees of freedom given by parameters in the model, a reasonable match is (most of the time) found between the two.

The examples of more advanced modeling just mentioned illustrate something important: there are many effects that need to be taken into account. The propagation of the wake, the wake interactions, and the change in the nature of the flow (increasing turbulence levels) have important effects on the output. One can not conclude, a priori, on the effects that can be neglected. For instance, a model that accounts for wake interaction, without discussing turbulence levels may be dubious.

I draw two important conclusions from this short analysis. The first one is that there is a need for more advanced modeling, to better understand the different mechanisms and their influences. The second conclusion is that it will be hard to find intermediate steps between the rough calculations resulting in relations such as in (1.1), and the full solution of the problem. The objective of this thesis is to find such steps (that might look like sideway steps) towards a complete resolution of the problem, or at least towards a model with rigorously justified assumptions.

1.2 PDE models

As a more realistic approach, we have chosen to use the theory of Partial Differential Equations (PDE). We take advantage of this section to define some notations that will be used throughout

this manuscript. Ω denotes a smooth domain in $\mathbb{R}^d$ where $d \in 1, 2, 3$. The problems considered will often be time dependent PDEs and the model equation that we have chosen to work with is:

$$\forall t \in [0, T],\ \frac{\partial u}{\partial t} + \mathcal{L}(u) = f \text{ on } \Omega. \tag{1.2}$$

To complete the system, and have a chance to have a well posed problem, one needs to provide appropriate initial $u(t = 0) = u_0$ and boundary conditions for u or $\frac{\partial u}{\partial n}$ on $\partial\Omega$. $\mathcal{L}$ will be throughout this manuscript a first or second order partial differential operator.

We make a constant use of the following functional spaces: $L^2(\Omega)$ the space of square integrable functions over Ω; $H^1(\Omega) := \left\{u \in L^2(\Omega),\ \nabla u \in \left(L^2(\Omega)\right)^d\right\}$. Also, denote $H_0^1(\Omega)$, the elements of $H^1(\Omega)$ with zero trace. That is $H_0^1(\Omega) := \left\{u \in H^1(\Omega),\ \text{ and } u = 0 \text{ a.e on } \partial\Omega\right\}$.

Let Y be any real Hilbert space. We will denote by $< \cdot, \cdot >_Y$ and $\| \cdot \|_Y := \sqrt{< \cdot, \cdot >_Y}$ the scalar product and the norm respectively. The dual space of Y will be denoted Y'. For instance, $\left(H_0^1\right)'(\Omega) = H^{-1}(\Omega)$. For an introduction on Sobolev spaces, we refer to the reference book on the subject [4].

In this thesis, our focus will be put mainly on three equations. A rigorous theoretical presentation of each of them is not in the scope of this thesis, nor is it its topic. We briefly state here some of their characteristics, and detail the chapters of the thesis and the particular contexts in which each of them appears.

Burgers We will encounter in this manuscript both the viscous $\epsilon > 0$ and inviscid cases $\epsilon = 0$. The viscous case is a one dimensional non linear but simplified model of the Navier-Stokes equations. It allows for simple numerical experiments, as will be conducted in chapter 3.

The inviscid case is a simple hyperbolic problem, and is often chosen as a first test case for hyperbolic solvers in the literature. We will throughout this manuscript focus on the viscosity, physically meaningful solutions. We will use it in chapter 5 as a model for Euler equation.

Euler We will study this equation in a two dimensional setting in chapters 4 and 5. It is, as the inviscid Burgers equation, an hyperbolic problem. Our main focus will be put on the development of shocks.

Navier-Stokes This is the target equation. Because of the issues that will be raised in section 1.5, we will actually not work with it a lot in this manuscript. We present in Figure 1.2 the model problem we work with in chapter 2. The wind turbines are modeled by cylinders, to focus on the optimization of the geometry of the farm, rather than on the realistic computation of a CFD flow around a turbine. Navier-Stokes simulations are also performed in the numerical experiments of section 3.9.

1.2.1 Numerical methods

A discussion about the (many) different numerical schemes used for CFD computations is not in the scope of this manuscript. This is the topic of many books in the literature, for instance [8] or the more recent [34]. Also, a more specific overview of PDE solvers' specificities for the computation of flows around wind turbines can be found in [134]. We only briefly come back one aspect. Instead of considering realistic physical blade bodies, one can model the wind turbines with equivalent source terms. The most commonly used methods in that direction are the actuator disk and actuator line models. A presentation of both methods can be found in [97]. The form of the source term is chosen using basic physical considerations in the same

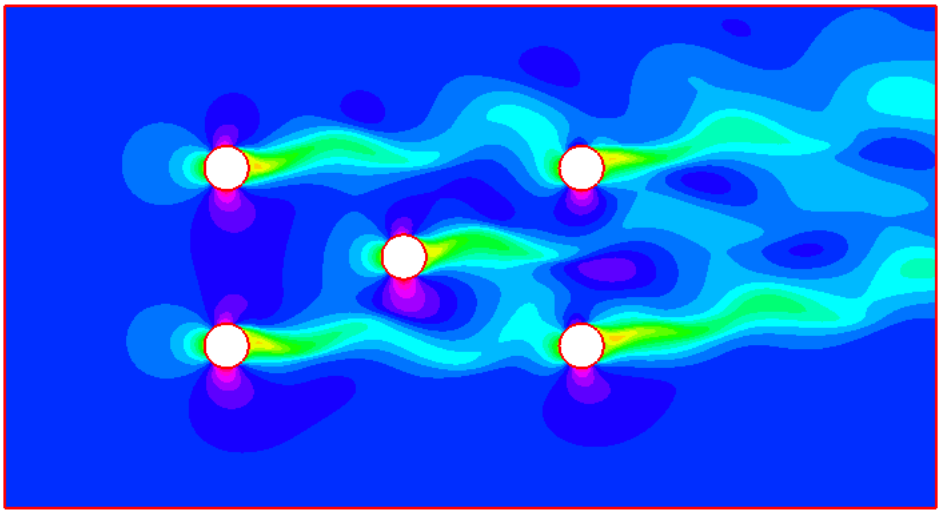


Figure 1.2: The chosen model problem for chapter 2

vein as the ones that were used in section 1.1. The numerical results obtained for wind farm flow computations are promising. But as for the engineering models, they require parameter fitting. More precisely, one has to tweak the parameters of the modeled blade for each numerical simulation. One can argue that the quality of the results presented is mitigated by this parameter fitting. Nevertheless, this method can be seen as an intermediate method between the complete numerical solution, and the basic engineering models such as the one presented in section 1.1.

This type of methods will not be further studied in this manuscript, and this for a very simple reason: we use even coarser approximations in the course of this thesis. In chapter 2, we use cylinders to model the windturbines. The chapters that follow use simpler shapes or even different equations, to focus on very specific issues.

We conclude this short section by stating one important assumption, that we will have to keep in mind all through this manuscript. It actually explains why we do not further discuss the standard numerical CFD methods. We assume that we have a fine solver that exactly captures the true physical solution. This fine solver can for instance be of Finite Elements (FE) type, of Finite Volumes (FV) type or a Discontinuous Galerkin (DG) scheme. To insist on the fact that the resulting solution cannot be distinguished from the continuous solution, it is often referred to as 'truth approximation'. The reasons for this point of view is easily understandable. Behavior/structures that can not be captured by fine schemes, are out of reach of the desired cheap approximate solvers. The best thing the latter can do is try and match the 'best known' solutions, which are the ones obtained with fine schemes.

Remark 1 *Only a few references do not use this premise. In [93, 127], the idea is to use data fitting in addition to some learned model. Instead of trying to match the fine solutions as best as*

possible, we allow for a bias in the model.

We denote from now on $\mathcal{N}$ the number of degrees of freedom of the chosen fine solver providing the truth solution. This gives us a reference computational cost. More precisely, any computational cost of order $\mathcal{N}$ will be considered a high cost.

1.3 A first try at optimization

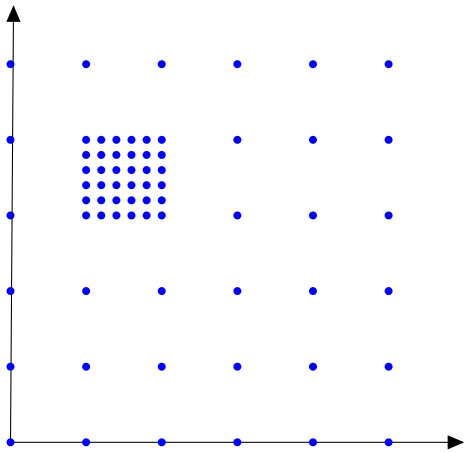


Figure 1.3: One possible empirical optimization algorithm: the surface method

We present one more method before entering the core of this manuscript. It is intermediate between the engineering models of section 1.1 and the full reduced model framework of the remainder of this thesis. Let $\mathcal{D}$ be a parameter space and let J be a functional defined on $\mathcal{D}$ that we are trying to minimize. What we describe is a typical surface type method, an empirical answer to parametrized problems that can be easily implemented. We present a possible implementation in Algorithm 1 below.

Data: Parameter space: $\mathcal{D}$
Result: μ^{opt}
Define Ξ_0 some coarse sampling of the parameter space;
Compute the solution $u(\mu)$ for each $\mu \in \Xi_0$;
repeat
  Find $\omega_k \subset \mathcal{D}$ such that $J(\mu)$ is small for $\mu \in \Xi_k \cap \omega_k$;
  Sample more finely $\mathcal{D}$ in w_k: Ξ_{k+1};
  Compute the solution $u(\mu)$ for each $\mu \in \Xi_{k+1}$;
  $k \leftarrow k + 1$;
until some accuracy/computational cost condition;

Algorithm 1: Surface type method

An illustration of this algorithm for a parameter space embedded into $\mathbb{R}^2$ is shown in Figure 1.3. The blue dots represent parameters for which the truth solution has been computed. This method requires the estimation of J in between sampled points. This is achieved using an interpolation procedure. The latter is an important component of the method and has a big impact on the overall accuracy. But as there is no theoretical results associated, it is done

empirically. The direct consequence is that in order to achieve a decent accuracy, one needs to perform many fine scheme computations. It is obvious that this is not in accordance with the goal of computational cost reduction. Moreover, this issue is made even worse because of the curse of dimensionality. Indeed, as the parameter space is a subspace $\mathbb{R}^{\dim(\mathcal{D})}$, for a fixed discretization step in the parameter space (and so equivalently for a fixed accuracy) the number of fine computations required grows exponentially with dimension of the parameter space. The model order reduction framework gives a theoretically more sound and computationally more reasonable answer to this kind of parametrized problem.

1.4 Model order reduction

We start with a macroscopic overview of the framework as well as its main requirements. We then detail the steps and key results. For a more thorough presentation, we refer to the two recent books on the subject, [68] and [109]. We denote from now on with μ a generic parameter and with $\mathcal{D}$ a generic parameter space.

The objective is to construct a method that allows for many queries of the type $\mu \to u(\mu)$. Let $\mu \in \mathcal{D}$. The solutions $u(\mu)$ are typically solutions to some PDE:

$$\begin{cases} \frac{\partial u(\cdot,t;\mu)}{\partial t} + \mathcal{L}(u(\cdot,t;\mu)) &= f & \text{in } \Omega \\ u(\mu)(t=0) &= u_0 & \text{in } \Omega \\ u \text{ or } \frac{\partial u}{\partial n} &= g & \text{on } \partial\Omega. \end{cases} \tag{1.3}$$

ROM assumes no a priori parametric dependence. f, u_0, g, Ω and $\mathcal{L}$ may depend on the parameter μ. We will always explicitly state the parametric dependency. The question that model order reduction tries to answer is: is there some regularity, whatever the precise meaning, of

$$\begin{cases} \mathcal{D} &\to & X \\ \mu &\mapsto & u(\mu) \end{cases} \tag{1.4}$$

and if so, can we take advantage of it to accelerate the computation of (1.4). Our motivation in this manuscript is optimization. Note that this can also be used to solve inverse problems. In the course of this thesis, the regularity of (1.4) will take various forms. It ranges from differentiable with an explicit derivative, as in section 1.4.1, to cases that require a preconditioning step, see section 1.5.1.

The fundamental notion for ROM that will be used throughout this manuscript, is the concept of solution manifold. Several definitions are possible depending on the context, see for instance the introduction of chapter 3. The most common one[2], that will be used unless specified, is given by:

$$\mathcal{M} := \{u(\cdot,t;\mu),\ \mu \in \mathcal{D},\ t \in [0,T]\}\,. \tag{1.5}$$

The name manifold is not chosen loosely. The premise of ROM is to think of $\mathcal{M}$ as a smooth manifold, embedded in a well chosen Hilbert space X, even if no theoretical results regarding regularity are available for most real life problems.

As we are aiming for the reduction of the computational cost of solutions in $\mathcal{M}$, one sensible first step is to try to capture and compress the characteristics of this manifold. One adapted

[2]One sometimes considers a space time formulation, which results in a different solution manifold, see for instance [20].

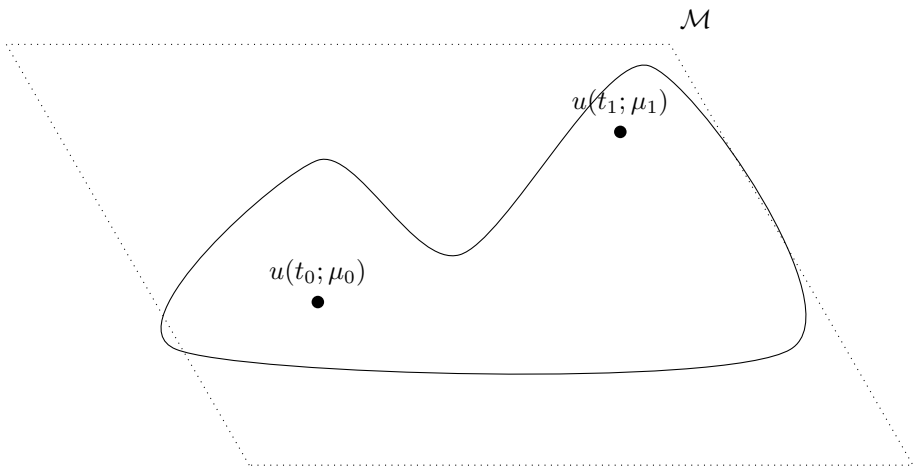


Figure 1.4: Solid line: Graphical representation of the solution manifold $\mathcal{M}$. Dotted line: A possible trial space

theoretical tool is the notion of Kolmogorov n-width. This quantity measures the 'linear width' of any subset of normed spaces. More precisely, for any manifold $\mathcal{M}$ it is defined as:

$$d_n(\mathcal{M}, X) := \inf_{E_n} \sup_{f \in \mathcal{M}} \inf_{g \in E_n} \|f - g\|_X. \tag{1.6}$$

The first infimum is taken over all linear spaces of dimension n embedded in X. A graphical example is presented in Figure 1.4. The Kolmogorov n-width returns the worst approximation, on the best linear space of dimension n. A few theoretical results are available to estimate a priori this n-width. In [99], they prove n-width estimates for solutions to elliptic problems where the parameter dependence is on the source term. The latter is taken in some compact of a high order Sobolev space. A more recent result gives estimates of the Kolmogorov n-width under holomorphic mappings [35]. More precisely, they show that the exponential decay of the n-width is conserved through the image of a Frechet differentiable function.

We conclude this overview of ROM by introducing the offline/online paradigm. The capture and modeling of the characteristics of the solution manifold $\mathcal{M}$ can be seen as a learning phase, referred to as offline in the ROM community [3]. It often amounts to creating a linear space of moderate dimension, that represents well $\mathcal{M}$. The target error, for a fixed basis size, is given by the Kolmogorov n-width. This learning phase is expensive from a computational cost, as it involves the computation of a moderate number of fine solutions. The online phase uses this learning phase, and provides a way of computing approximations of members of $\mathcal{M}$ at a reduced cost[4]. The analytical example of the next section will help understand this concept of offline/online stages.

[3] A tabular expliciting other similarities between ROM and machine learning algorithms is presented in chapter 7.

[4] Recall that the reference high cost, is the cost of a fine computation, with $\mathcal{N}$ degrees of freedom.

1.4.1 Analytical example

As stated in the preamble, we have chosen to tackle the description of ROM by focusing on toy examples. The first one we study is a favorable case, as the regularity of the problem with respect to the parameter is explicit. Let the following heat equation:

$$\left\{ \begin{array}{rcl} -\nabla \cdot (\mu \nabla u) & = & f \text{ in } \Omega \\ u & = & 0 \text{ on } \partial\Omega. \end{array} \right. \tag{1.7}$$

We choose a parameter space $\mathcal{D}$ that satisfies:

$$\left\{ \begin{array}{l} \mathcal{D} \text{ is a compact subspace of } L^\infty(\Omega) \\ \exists \mu_{min} \in \mathbb{R}^{+*},\ \forall \mu \in \mathcal{D},\ \mu > \mu_{min} \text{ a.e in } \Omega. \end{array} \right.$$

How do the solutions $u(\mu)$ behave when we change the diffusion coefficients ? Existence and uniqueness of the solution in $H^1(\Omega)$ are guaranteed by classic Lax-Milgram theory for coercive and continuous operators.

We study the limit of $u(\mu + h\nu) - u(\mu)$ for a fixed direction ν contained in the unit ball of tangent space of $\mathcal{D}$ and $h \in \mathbb{R}$. Let $u^h := u(\mu + h\nu)$ and let $v^h := u^h - u(\mu)$. Using the linearity of the equation, the latter is solution of:

$$\left\{ \begin{array}{rcll} -\nabla \cdot (\mu \nabla v^h) & = & -h\nabla \cdot (\nu \nabla u^h) & \text{in } \Omega \\ v^h & = & 0 & \text{on } \partial\Omega. \end{array} \right.$$

We can compute the standard a priori estimates for v^h. For this, multiply the first equation by v^h and integrate by parts:

$$\int_\Omega \mu \nabla v^h \cdot \nabla v^h = h \int_\Omega \nu \nabla u^h \nabla v^h.$$

We have used v^h zero on $\partial\Omega$. We use the hypotheses on $\mathcal{D}$ and on $\|\nu\|_{L^\infty}$ to show the a priori estimate:

$$\mu_{\min} |v^h|_{H^1} \leq h |u^h|_{H^1}.$$

To conclude, we use the a priori estimates on u^h:

$$\mu_{\min} |u^h|^2_{H^1} \leq \|f\|_{H^{-1}} \|u^h\|_{H^1},$$

with the Poincare inequality:

$$\exists C \in \mathbb{R}, \quad \text{s.t} \quad \mu_{\min} |u^h|_{H^1} \leq C \|f\|_{H^{-1}},$$

to finally obtain the following a priori estimates on v^h:

$$|v^h|_{H^1} \leq \frac{1}{\mu^2_{min}} C h \|f\|_{H^{-1}}. \tag{1.8}$$

Let $w(\mu, \nu) \in H^1_0(\Omega)$ be the solution to

$$\left\{ \begin{array}{rcll} -\nabla \cdot (\mu \nabla w) & = & -\nabla \cdot (\nu \nabla u(\mu)) & \text{in } \Omega \\ w & = & 0 & \text{on } \partial\Omega. \end{array} \right.$$

To prove Gateaux differentiability, we will prove estimates on:

$$\eta : \left\{ \begin{array}{rcl} \mathcal{D}, \mathcal{D}, \mathbb{R} & \to & H^1(\Omega) \\ (\mu, \nu, h) & \mapsto & u(\mu) - u(\mu + h\nu) - h w(\mu, \nu). \end{array} \right.$$

We know:

$$\begin{aligned} -\nabla \cdot (\mu \nabla \eta(\mu, \nu, h)) &= -\nabla \cdot (\mu \nabla \left(u(\mu) - u(\mu + h\nu) - hw(\mu, \nu)\right) \\ &= \nabla \cdot ((h\nu) u(\mu + h\nu)) + h\nabla \cdot (\mu w(\mu, \nu)) \\ &= \nabla \cdot ((h\nu) u(\mu + h\nu)) + h\nabla \cdot (\nu u(\mu)) \\ &= \nabla \cdot ((h\nu) \nabla (u(\mu + h\nu) - u(\mu)). \end{aligned}$$

This gives a priori estimates on η:

$$\mu_{min} |\eta(\mu, \nu, h)|^2_{H^1} \leq h |\eta(\mu, \nu, h)|_{H^1} |u(\mu + h\nu) - u(\mu)|^2_{H^1}.$$

We use the estimates on $v^h = u(\mu + h\nu) - u(\mu)$, see equation (1.8), and obtain:

$$\mu_{min} |\eta(\mu, \nu, h)|_{H^1} \leq \frac{1}{\mu^2_{min}} C h^2 \|f\|_{H^{-1}}. \tag{1.9}$$

This concludes on the Gateaux differentiability of the application, in the direction ν, with derivative $w(\mu, \nu)$. Moreover, since for all $\mu \in \mathcal{D}$,

$$w(\mu, \cdot) : \left\{ \begin{array}{ccc} \mathcal{D} & \rightarrow & H^1_0 \\ \nu & \mapsto & w(\mu, \nu) \end{array} \right.$$

is linear and continuous, we conclude on the Frechet differentiability of the parametrized problem over $\mathcal{D}$.

From this analysis, we have one natural way of developing a method to approximate $\mu \rightarrow u(\mu)$ at reduced cost. The first step is construct a good representation of the solution manifold $\mathcal{M}$. The Frechet differentiability provides an easy solution, that we illustrate in Figure 1.5. Define

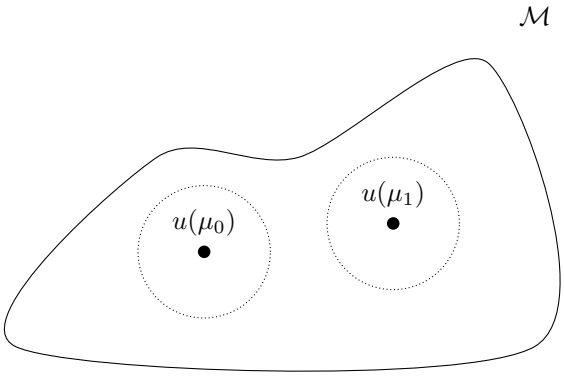


Figure 1.5: One way of constructing a reduced basis: the Frechet differentiable case

some threshold ϵ, the maximum approximation error that we want on our solution manifold $\mathcal{M}$. Sample the parameter space such that:

$$\forall \mu \in \mathcal{D},\ \exists \mu_i \in \mathcal{D},\ \ \text{s.t}\ \ \|u(\mu) - u(\mu_i) - w_{\mu_i}(\mu - \mu_i)\|_{H^1} \leq \epsilon.$$

The a priori estimates on η, see equation (1.9), allow us to compute a rigorous distribution of $\{\mu_i\}_i$ and the corresponding Frechet derivatives $\{w_{\mu_i}\}_i$. This ends the offline phase as introduced

in the previous section. The online phase is even easier in this setting. For any $\mu \in \mathcal{D}$, we just have to pick μ_0 in the pre computed set the closest to μ. We then have an explicit formula for an approximation of $u(\mu)$. The guaranteed error estimate is the chosen threshold ϵ. Because of the Frechet differentiability, independent of the direction, we feel that we are starting to overcome the curse of dimensionality.

We note the main differences with a realistic case. First of all, the smoothness with respect to the parameter does not often translate into an explicit formulation. Sometimes, the manifold even needs some preconditioning to enforce smoothness, see for instance chapter 5. Also, it feels like this type of construction neglects a lot of redundancy. For instance, we expect redundant pieces of information to be found in solutions for parameters far away in the parameter space. Nevertheless, the methodology developed on this analytic example is exactly the same as what is being done for ROM for real life problems, and is enlightening in that respect. The next section describes a typical offline phase.

1.4.2 ROM, the offline phase

As already stated, the first necessary step is to construct a basis that captures most of $\mathcal{M}$. More precisely, we are looking for Ψ, a set of N functions in X the underlying Hilbert space, such that:

$$\forall u \in \mathcal{M},\ \exists u^N \in \text{ span } \Psi,\ \|u - u^N\|_X \text{ small } .$$

This first necessary step is the sampling of the parameter space. That is, we need to choose a representative set:

$$\Xi := \{\mu_k,\ k = [1 \dots N]\} \subset \mathcal{D}.$$

It is easy to see that the size of Ξ influences the offline computational cost as well as the online accuracy of the method. The selection of the set Ξ is almost always [5] done empirically. The reduced basis is then picked as a subspace of:

$$\text{span } \{u(\mu_k),\ \mu_k \in \Xi\} \subset X.$$

The way the compression of the information contained in $\{u(\mu),\ \mu \in \Xi\}$ is done, varies among the many methods available. We mention here a few of them, the ones we feel the closest to our objective. A geometric approach close to the Centroidal Voronoi Tesselation (CVT) has been developed in [46, 25]. The parameter space is splitted using a CVT algorithm. In solid mechanics, the most commonly used algorithm is the Proper Generalized Decomposition (PGD) [83]. This greedy approach results in a separable approximation of the solution, along the different dimensions: time, space and parameter for instance. It is empirical for most applications. We mention some of the theoretical results are available. If the problem actually is a separable problem, then the PGD algorithm reduces to a POD algorithm (that we will detail below), see [102]. Also, for the Poisson equation in the 2 dimensional unit square, it can be shown that each greedy iteration is well posed, and that the algorithm converges, see [85]. Finally, we mention the Balanced Truncation method [118], which is built on a 'Linear Time Invariant' form of the system to solve.

We now give more details on the two methods that will be used in the rest of the manuscript. They are dominating in the ROM community, especially for CFD computations. We underline the up and downsides of both methods and conclude on the purposes they should be used for.

[5] The only example using a different sampling strategy we are aware of is presented in [81].

1.4.2.1 Reduced Basis (RB)

In this section, we give an overview of the reduced basis method. It relies heavily on the existence of a cheap, rigorous error estimator. That is, for each reduced basis $\Psi := \{\psi_i\}$, it requires an application:

$$\Delta_\Psi : \left\{ \begin{array}{ccc} \mathcal{D} & \rightarrow & \mathbb{R} \\ \mu & \mapsto & \Delta_\Psi(\mu), \end{array} \right.$$

that is an upper bound on the actual error made on the reduced basis approximation. More precisely, for $u(\mu)$ the truth approximation and $u^N(\mu)$ the reduced basis Galerkin approximation, it should satisfy:

$$\forall \mu \in \mathcal{D},\ \|u(\mu) - u^N(\mu)\|_X \leq \Delta_\Psi(\mu).$$

The construction of such an error estimator is described in chapter 3. The greedy algorithm follows naturally:

Data: Fine sampling of the parameter space $\mathcal{D}$: Ξ
Threshold ϵ
Maximum size of the basis N^{max}
Result: Reduced basis Ψ^*
$k \leftarrow 0$;
$\mu_0 \leftarrow$ random parameter in $\mathcal{D}$;
$\Psi_0 := \{u(\mu_0)\}$;
while $k < N^{max}$ and $\Delta_{\psi_k} > \epsilon$ **do**
  $\mu_{k+1} := \underset{\mu}{\text{argsup}}\ \Delta_{\Psi_k}(\mu)$;
  $\Psi_{k+1} := \Psi_k \cup u(\mu_{k+1})$;
  $k \leftarrow k + 1$;
end
$\Psi^* := \Psi_k$;

Algorithm 2: The greedy RB method

Recent results [23] show that if the Kolmogorov n-width of the solution manifold $\mathcal{M}$ decays exponentially, i.e if:

$$\exists c, C \in \mathbb{R}^2 \text{ s.t } \forall n\ d_n(\mathcal{M}, X) \leq ce^{-Ck}, \tag{1.10}$$

then the basis obtained using the greedy algorithm described above inherits this property:

$$\begin{array}{ll} \exists (c', C') \in \mathbb{R}^2, \quad \text{depending on } (c, C) & \text{s.t } \forall N, \\ & \sup_{u \in \mathcal{M}} \inf_{u^N \in \Psi_N} \left\| u - u^N \right\|_X \leq c' e^{-C' N} \end{array} \tag{1.11}$$

Similar results are available if the argsup during the greedy algorithm is not done exactly [43].

This method's main advantage is obviously its computational cost. Indeed, only a moderate number fine computations need to be performed, thanks to the error estimator. Another interesting property is that we have a guaranteed error bound on a fine sample of $\mathcal{D}$. The major downside is that that for non linear problems, the error estimators available are not reliable, as the bounds are not tight. This reduces a lot the range of application of this method. It will be discussed in section 1.4.3.1 where we explain how to construct the error estimator. We will also discuss it in 3 when discussing a posteriori error estimations for the one dimensional viscous Burgers equation.

1.4.2.2 Proper Orthogonal Decomposition (POD)

In the RB algorithm, the objective is to mimic the search for the optimal space in the sense given by the Kolmogorov n-width. The only difference is the replacement of the true error by an error estimator. The Proper Orthogonal Decomposition (POD) uses a different objective function. Let N be some prescribed size for the reduced basis and let J be the following functional:

$$J : \left\{ \begin{array}{lcl} X^N & \to & \mathbb{R} \\ (\psi_1, ..., \psi_N) & \mapsto & \sum_{\mu_j \in \Xi} \|u(\mu_j) - \Pi u(\mu_j)\|_X^2 \end{array} \right. \tag{1.12}$$

where Π is the orthogonal projection[6] onto $\text{span } \{\psi_i,\ i = [1 \dots N]\}$. The objective of the POD method is to minimize J over all orthogonal basis of cardinality N in X. We will prove in the course of this section that J has a unique minimizer and that the resulting basis is in fact in $\text{span } \{u(\mu_j),\ \mu_j \in \Xi\}$.

Remark 2 *When using discretized solutions (i.e when X is finite dimensional), the POD reduces to a (correctly reweighted) Singular Value Decomposition (SVD), see for instance [63].*

Remark 3 *This problem has a solution even when considering continuous snapshots in X (instead of the sampled case described in this section). With our notation, we can replace the discrete $\sum_{\mu_j \in \Xi}$ by the continuous $\int_{\mathcal{D}} d\mu$, for an appropriate measured parameter space $\mathcal{D}$.*

We start by deriving the first order optimality conditions, a set of necessary conditions on an hypothetical optimal basis $\Psi := \{\psi_i,\ i = [1, \dots, N]\}$. For this, we formulate the optimal problem as:

$$\min_{\Psi \in X^N} J(\psi) \text{ s.t } \ < \psi_i, \psi_j >_X = \delta_{ij},\ \forall i, j.$$

To avoid redundant constraints, define the following:

$$e_j : \left\{ \begin{array}{lcl} X^j & \to & \mathbb{R}^j \\ (\psi_k)_{k \in [1,...,j]} & \mapsto & ((\psi_k, \psi_j)_Y - \delta_{kj}) \, . \end{array} \right.$$

For all k, denote $\lambda_k \in \mathbb{R}^k$ the lagrange multiplier associated with the constraint e_k and construct the Lagrangian $\mathcal{L}$:

$$\mathcal{L} : \left\{ \begin{array}{lcl} X^N \times \Pi_{k=1}^N \mathbb{R}^k & \to & \mathbb{R} \\ \{\psi_i\},\ \{\lambda_k\} & \mapsto & J(\Psi) + \sum_{k=1}^N < e_k, \lambda_k >_{\mathbb{R}^k} = 0 \end{array} \right.$$

We compute $\frac{\partial \mathcal{L}}{\partial \psi_i}$, for some $i \in [1, \dots, N]$ in the direction $\delta\psi$:

$$\begin{array}{rl} \frac{\partial \mathcal{L}}{\partial \psi_i} \delta\psi = & - \sum_{\mu_j \in \Xi} < \delta\psi, u(\mu_j) >< \psi_i, u(\mu_j) > \\ & + \sum_{p=1}^{i-1} (< \delta\psi, \psi_p >_Y) \lambda_i^p \\ & + 2\lambda_i^i < \delta\psi, \psi_i > - \sum_{k=i+1}^N (< \psi_k, \delta\psi >_Y) \lambda_k^i. \end{array}$$

First order optimality conditions state that for the basis Ψ to be optimal, it needs to satisfy:

$$\forall i \in [1, \dots, N],\ 2 \sum_{\mu_j \in \Xi} < \psi_i, u(\mu_j) > u(\mu_j) = \sum_{p=1}^{i-1} \psi_p \lambda_i^p + 2\lambda_i^i \psi_i - \sum_{k=i+1}^{N} \psi_k \lambda_k^i. \tag{1.13}$$

[6] We note that POD uses the Hilbert space structure, where RB can be performed in a any Banach space.

Define the following functional on X:

$$\mathcal{R} : \left\{ \begin{array}{ccc} X & \to & X \\ \psi & \mapsto & \sum_{\mu_j \in \Xi} < u(\mu_j), \psi >_X u(\mu_j). \end{array} \right.$$

We show that for each N, size of the basis, the first order optimality conditions, (1.13), are equivalent to:

$$\{\forall i \in [1, \dots, N],\ \psi_i \text{ is an eigenfunction of } \mathcal{R}\}.$$

The proof is done by induction on N. For $N = 1$, equation (1.13) becomes:

$$\sum_{\mu_j \in \Xi} < \psi_1, u(\mu_j) > u(\mu_j) = \lambda_1^1 \psi_1,$$

which concludes. Suppose it is true for some N. We know that the set of ψs satisfy:

$$\forall i \in [1, \dots, N+1],\ 2\mathcal{R}(\psi_i) = \sum_{p=1}^{i-1} \psi_p \lambda_i^p + 2\lambda_i^i \psi_i - \sum_{k=i+1}^{N+1} \psi_k \lambda_k^i \tag{1.14}$$

Let $i < N + 1$. We use the orthogonality of the basis to show that:

$$\left\{ \begin{array}{l} 2 < \mathcal{R}(\psi_i), \psi_{N+1} >_X = -\lambda_{N+1}^i \\ 2 < \mathcal{R}(\psi_{N+1}), \psi_i >_X = \lambda_{N+1}^i. \end{array} \right.$$

As $\mathcal{R}$ is symmetric, we easily conclude $\forall i < N + 1,\ \lambda_{N+1}^i = 0$. The first order optimality conditions are thus equivalent to:

$$\left\{ \begin{array}{lll} \forall i \leq N,\ 2 \quad \mathcal{R}(\psi_i) & = & \sum_{p=1}^{i-1} \psi_p \lambda_i^p + 2\lambda_i^i \psi_i - \sum_{k=i+1}^{N} \psi_k \lambda_k^i \\ \mathcal{R}(\psi_{N+1}) & = & \lambda_{N+1}^{N+1} \psi_{N+1} \end{array} \right.$$

We conclude using the induction hypothesis.

Remark 4 *In the literature, the inductive proof is not always conducted properly. It sometimes takes for granted that the optimal basis is hierarchical, i.e that:*

$$\underset{rank\ N+1}{argmin}\ J(\psi) = \underset{rank\ N}{argmin}\ J(\psi) \cup \psi^{n+1}$$

where ψ^{n+1} is the solution of some other minimization problem [7]*. This is not trivial, and should be proved (using the same simple steps above).*

To prove that this necessary condition is in fact sufficient requires more work. The first thing is to prove the existence of such eigenfunctions/eigenvectors. For finite dimensional scalar products, this is a simple consequence of the spectral theorem. The extension to $X = L^2(\Omega)$ is presented in appendix and uses the Hilbert-Schmidt theorem. The last ingredient is to show that the basis Ψ constructed using the first N (when ordered with decreasing eigenvalues) eigenfunctions of $\mathcal{R}$ is effectively minimizing the functional J. The proof is done by a direct argument, i.e by showing that:

$$\forall\ \tilde{\Psi} \text{ orthogonal basis of cardinality } N \text{ in } X,\ J(\Psi) \leq J(\tilde{\Psi}).$$

The proof can be found for instance in [136].

[7] The rank 1 POD of the projection of the original set onto the POD basis of rank N.

Remark 5 *In the literature, it is not unusual to work with centered solution manifold in this context. More precisely, one can subtract the mean field or the stationary solution to the snapshot set before performing POD compression. A motivation for this additional procedure can be found in [138, 70].*

The major downside of the POD method is its computational cost. We discuss this issue in a small chapter in this thesis see 6. There is one other downside. As we are optimizing the mean projection error, it is possible that at the end of the algorithm, there exists a subset of $\mathcal{D}$ for which the basis behaves very poorly. This issue is solved by the variant of the standard POD described in the next section.

1.4.2.3 Localized RB

To complete this small introduction on the construction of reduced basis, we mention adaptive RB methods. Adaptive is here to be understood in the following sense: instead of one single basis used on the whole parameter space $\mathcal{D}$, we construct a small number of reduced bases, each of them with a domain of validity. The initial work in that direction was done in [49]. They define the notion of trust region in the parameter space. A more involved version has later been proposed in [96]. Their idea is to construct offline a metric in the parameter space. We present in Figure 1.6 a graphical illustration. Ideas related to hp can be implemented in the same spirit, see

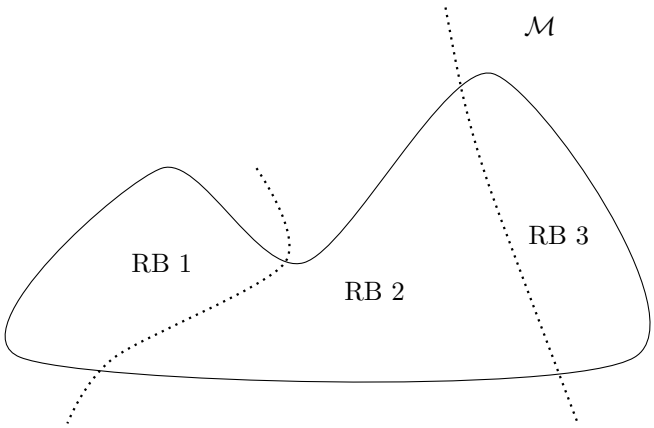


Figure 1.6: Localized Reduced Basis

for instance [48]. We also mention a related method which uses interpolation between reduced basis [7].

1.4.3 ROM, the online phase

In this section, we present how the basis constructed in the previous section are being used. This corresponds, using the ROM vocabulary, to the online phase. We once again work with the heat equation, see (1.7). We modify the setting compared to section 1.4.1, and put ourselves in a more classical ROM framework. The parameter dependency is now characterized by some function g over $\mathcal{D}$:

$$g : \left\{ \begin{array}{ccc} \mathcal{D} & \to & L^\infty \\ \mu & \mapsto & g(\cdot;\mu). \end{array} \right.$$

The objective is to propose a method that allows for an efficient computation of an approximation of $u(\mu) \in X := H^1(\Omega)$, the solution to:

$$\left\{ \begin{array}{rcl} -\nabla \cdot (g(\cdot;\mu)\nabla u) & = & f \text{ in } \Omega \\ u & = & 0 \text{ on } \partial\Omega \end{array} \right. \tag{1.15}$$

when μ varies in $\mathcal{D}$. As usual, we denote with $C(\mu)$ and $\alpha(\mu)$ the continuity and coercivity constants of the bilinear form associated with the weak form of the equation. Suppose that we have managed, through POD or RB method for instance, to find an appropriate basis, that is, a linear space

$$X^N := \text{ span } \{\phi_i,\ i \in [1 \dots N]\} \subset H^1_0(\Omega),$$

that is almost as good as the optimal Kolmogorov n-width basis. This hypothesis can be formulated as:

$$\forall u \in \mathcal{M},\ \exists u^N \in X^N,\ \|u - u^N\|_X \approx d_N(\mathcal{M}, X).$$

The optimal representant is here the orthogonal projection of the true solution onto X^N. As the true solution is not known, we need another way of choosing a good representant in X^N. Good in the sense that the error should be controlled by the best projection error.

Remark 6 *This question is the same as the one that appears when one looks for a finite element solution.*

The choice is almost always the Galerkin method[8], as in the FE context. Pick $u^N(\mu) \in X^N$ that cancels the projection of the residual onto X^N. In other words, pick $u^N(\mu)$ in X^N such that:

$$\left\{ \begin{array}{rcll} -\nabla \cdot (g(\cdot;\mu)\nabla u^N(\mu)) & = & f & \text{in the dual space of } X^N \\ u^N(\mu) & = & 0 & \text{on } \partial\Omega. \end{array} \right. \tag{1.16}$$

We can put (1.16) in variational form:

$$\forall v^N \in X^N,\ \int_\Omega g(\cdot;\mu)\nabla u^N(\mu)\nabla v^N = \int_\Omega f v^N$$

Cea's Lemma guarantees, for this elliptic coercive problem the following estimate on the Galerkin approximation:

$$\forall \mu,\ \|u(\mu) - u^N(\mu)\|_X \leq \frac{C(\mu)}{\alpha(\mu)} \inf_{v^N \in X^N} \|u(\mu) - v^N\|_X.$$

That is, the error is controlled by the best approximation error.

Remark 7 *More general cases such as saddle problems, objective oriented problems and problems with non compliant outputs can be treated. See for instance [65] for a review.*

How do we implement the resolution of problem (1.16) ? We use the fact that both v^N and $u^N(\mu)$ lie on a finite dimensional space. The search for $u^N(\mu)$ can thus be reduced to the search of a set $\{\alpha_i\}_{i\in[1\dots N]} \in \mathbb{R}^N$, such that $u^N(\mu) = \sum_{i=1}^N \alpha_i \phi_i$ and thus:

$$\forall j \in [1, \dots, N],\ \sum_{i=1}^N \alpha_i \int_\Omega g(\cdot;\mu)\nabla\phi_i\nabla\phi_j = \int_\Omega f\ \phi_j.$$

[8] Other choices such as the Petrov-Galerkin method can also be used in a ROM context.

The desired set of coordinates is the solution to a system of small size, as problem (1.16) can be expressed as:

$$A(\mu)\alpha = F, \quad \text{where} \begin{cases} A_{i,j}(\mu) & := & \int_\Omega g(\cdot;\mu)\nabla\phi_i\nabla\phi_j \\ F_j & := & \int_\Omega f\ \phi_j. \end{cases}$$

We now make explicit the offline/online paradigm introduced section 1.4 in this specific case. The objective is to pre compute as many quantities as possible in the offline phase. The hope is that during the online phase, whatever the parameter μ considered, no quantity with a complexity dependent on $\mathcal{N}$ the number of degrees of freedom of the truth solver has to be performed. In our example for instance, the vector F can be computed once and for all, as it is independent of the parameter μ. The cheap computation of $\mu \to A(\mu)$ requires additional properties on $\mu \to g(\cdot;\mu)$. This will be discussed in section 1.5.2.1.

We end this quick overlook of reduced order modeling by mentioning a key feature, the a posteriori error estimators.

1.4.3.1 A posteriori error estimator

One of the problems with the engineering methods of section 1.1, and similarly with the method of section 1.3 is the fact that we have no way of certifying (or at least assessing) the answer given. Fortunately, this desired feature is available in the ROM framework. Let $\mu \in \mathcal{D}$. For simplicity we drop the μ dependency and denote u the continuous solution and u^N its RB approximation. Define the following residual operator r in the dual space of X:

$$r : \begin{cases} X & \to & \mathbb{R} \\ v & \mapsto & \int_\Omega g(\cdot;\mu)\nabla u^N\nabla v - \int_\Omega f\ v \\ & \to & \int_\Omega g(\cdot;\mu)(\nabla u^N - \nabla u)\nabla v. \end{cases}$$

Because we are using a Galerkin method, we know that the restriction of r onto X^N cancels, but of course, r does not cancel over X. The objective is to give an estimation of the RB error $\|u - u^N\|_X$ as a function of the norm of the residual $\|r\|_{X'}$. Using the coercivity of the initial problem, whose property is passed to both finite element discretization and reduced basis approximation, we have:

$$\left\|u - u^N\right\|_{H^1}^2 \leq \frac{1}{\alpha}\left|\int_\Omega g(\cdot;\mu)(\nabla u - \nabla u^N)^2\right| = \frac{1}{\alpha}\left|r(u^N - u)\right|$$

This gives us directly the desired upper bound on the RB error:

$$\left\|u - u^N\right\|_{H^1} \leq \frac{1}{\alpha}\|r'\|_{X'}$$

The resulting question concerns the computational cost of this residual norm. It is often achieved by computing r's Riesz representant, i.e $\hat{e} \in X$, and by using the well known property: $\|\hat{e}\|_X = \|r\|_{X'}$. A typical offline/online decomposition of such an error estimator will be presented in chapter 3.

Remark 8 *The resulting error bound guarantees a maximum between the reduced solution and the truth solution. Again, the underlying assumption is that the truth solution and the continuous solution are indistinguishable.*

Remark 9 *The a posteriori error estimations heavily relies on the structure of the continuous partial differential operator. We have presented one simple case, but the formulation is well understood for a large class of problems. A review can be found in [111].*

1.5 Specificities of the problem at hand

The previous section could have ended the introductive chapter. The roadmap is clear: construct an adapted reduced basis offline. Then build a computationally efficient scheme that estimates the power output of a wind farm. Finally use a posteriori error estimators to certify the answer. Unfortunately, things are not so easy.

The previous ROM framework does not apply directly to our problem, for several reasons. The geometric variability is the most visible issue. It makes the raw notion of Kolmogorov n-width obsolete, and thus the rest of the analysis of section 1.4 unusable. This is the topic of chapter 2. Other, bigger, problems, inherent to ROM for fluid simulations, will be discussed in the remainder of this section. We start by presenting some toy examples leading to continuous solution manifolds with large Kolmogorov n-width. We then present examples where the continuous solution manifold is well behaved, but where standard numerical schemes involve terms with large n-widths. As the smallness of the n-width is a necessary premise of any ROM method, we end this section by discussing two classes of methods to deal with these issues.

1.5.1 Continuous solution manifold with large Kolmogorov n-width

One can easily construct examples inspired by realistic fluid dynamics problems, where the raw reduced order modeling presented in section 1.4 fails. We mention two of them.

1.5.1.1 Strong influence of the inflow direction

Let us take a wind turbine (or obstacle) fixed in Ω. Suppose that the main inflow direction varies, either in time or with respect to some parameter. We can propose a few reasons for this specific behavior. One can for instance think of the modification of the outside boundary conditions, or of the deviation of the flow due to the presence of an upstream turbine. In any case, a basis reproducing flow structures propagating in all directions will necessarily be of big cardinality. As a consequence, instead of the standard solution manifold, see equation (1.5), it seems more adapted to consider a 'transformed' solution manifold. A natural choice is:

$$\tilde{\mathcal{M}} := \{u(r_{\theta(t;\mu)}(x), t; \mu),\ \mu \in \mathcal{D}\},$$

where r_θ is the 2 dimensional rotation matrix. Intuitively, we expect that

$$\forall n,\ d_n(\mathcal{M}) \gg d_n(\tilde{\mathcal{M}}),$$

for a well chosen $(t, \mu) \mapsto \theta(t; \mu)$. It is clear that this choice of modified solution manifold causes many problems, among which the choice of θ during the simulation, the matching of this rotating domain with an outside solution. These two issues will be discussed respectively in chapter 2 and chapter 3.

1.5.1.2 A propagating front

Propagating fronts/shocks are known to be a limiting case for ROM, see for instance [2]. In this section, we provide some quantitative evidences using a model example. We then try to give an explanation of why such strategies are still being used in the literature, and why the numerical results are not as poor as expected. This discussion is closely related to the one we will have

in the next section about numerical stabilization. The specific toy example we discuss here is inspired by [126]. We consider the following continuous solution manifold:

$$\mathcal{M} = \left\{ u(\cdot;\mu) \in L^2(0,1), \mu \in (0,1) \right\}, \tag{1.17}$$

where

$$u(x;\mu) = \begin{cases} 0 \text{ if } x < \mu \\ 1 \text{ otherwise.} \end{cases} \tag{1.18}$$

This can be thought of as the solution manifold of the transport of a step function. Its n-width can be explicitly computed. Let $\{\mu_j,\ j = [1,\dots,N]\}$ be a set of ordered parameters in $\mathcal{D}$, and the corresponding snapshots: $\{u(\cdot;\mu_j),\ j = [1,\dots,N]\}$. Let X^N be the space spanned by these snapshots. We start by computing the distance from X^N to one member of $\mathcal{M}$. Let $\mu \in \mathcal{D}$:

$$\inf_{v \in X^N} \|u(\cdot;\mu) - v\|_{L^2} = \begin{cases} \left(\frac{(\mu-\mu_j)(\mu_{j+1}-\mu)}{\mu_{j+1}-\mu_j} \right)^{\frac{1}{2}} & \text{on} \quad [\mu_j, \mu_{j+1}] \\ |\mu - \mu_1|^{\frac{1}{2}} & \text{on} \quad [0, \mu_1] \\ |\mu - \mu_N|^{\frac{1}{2}} & \text{on} \quad [\mu_N, 1]. \end{cases} \tag{1.19}$$

We estimate the n-width as follows:

$$\inf_{\{\mu_j\} \in \mathcal{D}} \max \begin{pmatrix} \sup\limits_{\mu \in [\mu_j, \mu_{j+1}]} & \left(\frac{(\mu-\mu_j)(\mu_{j+1}-\mu)}{\mu_{j+1}-\mu_j} \right)^{\frac{1}{2}} \\ \sup\limits_{\mu \in [0, \mu_1]} & |\mu - \mu_1|^{\frac{1}{2}} \\ \sup\limits_{\mu \in [\mu_N, 1]} & |\mu - \mu_N|^{\frac{1}{2}}. \end{pmatrix} \tag{1.20}$$

It is easy to see why the first infimum is obtained for a parameter distribution $\{\mu_j,\ j \in [1,\dots,N]\}$ such that the three quantities appearing in (1.20) balance. That is, the optimal distribution satisfies:

$$\begin{cases} \exists K \in \mathbb{R}, \quad \text{s.t } \forall j \in [1,\dots,N-1],\ \mu_{j+1} - \mu_j = K \\ \mu_1 \text{ and } \mu_N \text{ are such that } \sup\limits_{\mu \in [0,\mu_1]} |\mu - \mu_1|^{\frac{1}{2}} = \sup\limits_{\mu \in [\mu_N, 1]} |\mu - \mu_N|^{\frac{1}{2}} = \sup\limits_{\mu \in [\mu_j, \mu_{j+1}]} \left(\frac{(\mu-\mu_j)(\mu_{j+1}-\mu)}{\mu_{j+1}-\mu_j} \right)^{\frac{1}{2}} \end{cases}$$

The second condition is equivalent to $\mu_1 = \frac{K}{4} = (1 - \mu_N)$. With this new condition, we can compute explicitly the distance between the optimal parameters: $K = \frac{1}{N - \frac{1}{2}}$, and the corresponding distance between X^N and $\mathcal{M}$:

$$\text{dist } (X^N, \mathcal{M}) = \sqrt{\frac{1}{N - \frac{1}{2}}}$$

This is obviously a bad convergence rate, and it jeopardizes the framework of section 1.4[9].

The real life situation is even worse, because the output of a numerical scheme is not as good as the optimal error given above. Also, it is known that compression algorithms (such as POD or RB) would result in oscillating basis, causing stability issues. Nevertheless, one can found in the literature such strategies being used, and they are associated with decent numerical results and convergence rates. I suggest one explanation. Suppose that instead of trying to reproduce the

[9] We insist that for problems where the shock/front's position is constant in time or parameter wise, the current discussion does not apply, and the n-width of the solution manifold is not an issue.

full solution manifold $\mathcal{M}$ given equation (1.18), you are trying to reproduce the discrete training set:

$$\mathcal{M}^{train} := \left\{ u(\cdot;\mu^k),\ k \in [1,\dots,N^{snap}] \right\}.$$

We can see right away that the new n-width at hand is better behaved, as $d_n(\mathcal{M}^{train}) = 0$ for $n > N^{snap}$, and independent of the dimension $\mathcal{N}$ of the underlying truth approximation space. One possibility to detect such behavior, without adding to much complexity to the implementation, is to modify the time step/parameter step being used in the online phase. In the case described above, one would keep in the online phase the parameter range of the offline phase $[\mu_0, \mu_{N^{snap}}]$, but with a different sampling strategy.

1.5.2 Numerical schemes involving large n-widths

The causes of the failure of ROM for the two previous examples are visible and easily understandable. This section is devoted to the presentation of another class of sets with less obvious non decreasing n-widths. Numerical schemes for CFD computations often involve some sort of numerical stabilizer. In this manuscript, the latter have two major origins: they are either related to the closure of Navier-Stokes equation or pure numerical tools that help enforcing physical properties on the solutions. Among the second class of methods, we will focus on methods that aim at preserving the Total Variation Diminishing (TVD) nature of entropic solutions to conservation laws. Even though these two classes have different natures, we can extract common properties:

- they are working at a different, smaller scale than the approximation error , i.e the n-width of the solution manifold measured in X norm. More precisely, the stabilization terms should be bounded by the desired consistency error
- most of them have a highly non linear dependence on the state variable.

We start by saying a few things on how non linearities are usually handled in the ROM framework. We then describe one example of TVD stabilization that is not a proper candidate for standard ROM. We then discuss the closure of Navier-Stokes equation. We compare our conclusions with some numerical examples in the literature. We end the introductive chapter by giving possible methods to deal with this specific issue.

1.5.2.1 Handling non linear terms in ROM

In this section, we discuss the computational costs incurred by non linear terms. This is one of the main topic of research in the ROM community at the moment. The most commonly used method is the Empirical Interpolation Method (EIM). It was originally developed in [14] for a restricted set of applications among which the case where one of the physical parameters of the problem has a non trivial dependency on the parameter μ. We work once again with the model equation (1.15). In order to follow computational steps described in section 1.4, we need a good online/offline decomposition. A sufficient condition on $g(x;\mu)$ for this is its so called 'affine decomposability'. More precisely, we need a set of cardinality M of functions $\{g_k, k = 1,\dots,M\} \in X^M$, such that

$$\forall \mu \in \mathcal{D},\ \exists \{h_k(\mu), k = 1,\dots,M\} \in \mathbb{R}^M,\quad \text{s.t } g(\cdot;\mu) = \sum_{k=1}^{M} g_k(x) h_k(\mu).$$

The EIM method provides a way of constructing an approximate decomposition of such a type for situations where g is not rigorously affinely decomposable. It also provides a priori error estimates on the resulting approximation. Note that the strength of this method is that it does not solve the computational cost related problems by resorting to linearization or local linearization[10].

The simplicity of EIM have lead authors to widen its scope of application. It is now used when there is a non linear dependency on the solution itself. Denote $R : X \to X$ some generic non linear [11] function of u and $\{\phi_i,\ i = 1, \dots, N\}$ some well chosen reduced basis. A standard Galerkin method, see section 1.4, requires the efficient computation of terms such as:

$$< R(\sum_{j=1}^{N} \alpha_j \phi_j), \phi_i >_X,\ \forall i \in [1, \dots, N] \text{ for reasonable sets } \{\alpha_j, j = 1, \dots, N\}.$$

How is this case being handled by EIM ? A sufficient condition is again the existence of an approximate affine decomposition of $\mu \to R(u(\cdot;\mu))$, that is the existence of a set of cardinality M of functions $\{g_k(x), k = 1, \dots, M\} \in X^M$ such that:

$$\forall \mu \in \mathcal{D},\ \exists \{h_k(\mu), k = 1, \dots, M\} \in \mathbb{R}^M,\ R\left(u(\cdot;\mu)\right) \approx \sum_{k=1}^{M} g_k(x) h_k(\mu). \tag{1.21}$$

We can extract two key ingredients that impact the overall accuracy of the method. The first one the is the existence of such a family $\{g_k(x), k = 1, \dots, M\} \in X^M$. A necessary condition is for

$$R(\mathcal{M}) := \{R\left(u(\cdot;\mu)\right),\ \mu \in \mathcal{D}\}$$

to have a small n-width, in the norm of the underlying Hilbert space, the X norm. As we will see, this property can be hard to satisfy for non linear stabilization mechanisms.

The second ingredient is less important but is still worth noting. It is related to the procedure that selects the representant on the EIM basis, i.e the set $\{h_k(\mu), k = 1, \dots, M\}$ in equation (1.21). In the EIM framework, the latter are chosen using pointwise estimates. As a consequence, the a priori estimates available are given in L^∞ norm. Moreover, the upper bound appearing in the error estimate, the Lebesgue constant, depends on the size of the EIM basis. This is less favorable than its counterpart in the standard Galerkin method[12]. We draw one important conclusion from the previous discussion for standard non linearities: in order not to deteriorate the overall accuracy, we either need $R(\mathcal{M})$ to have a faster decaying Kolmogorov n-width than $\mathcal{M}$ or to work with an EIM basis bigger than the POD or RB basis.

1.5.2.2 Constructing a reduced TVD scheme

One needs to be careful when trying to apply an EIM type method to stabilization terms. As already mentioned above, these non linear terms act at the consistency error scale, and not at the approximation error scale. Denote R^{lim} some operator acting on X and serving as stabilization mechanism for the truth numerical scheme. The following inequality will often hold:

$$\|R^{lim}\|_{X'} \ll d_N\left(\mathcal{M}, X\right).$$

[10] In the bonus chapter 7, we present one such method, namely Dynamic Mode Decomposition.

[11] Note that this non linearity can originate from the continuous differential form or from the numerical scheme.

[12] We have seen in section 1.4 that Galerkin gives a priori estimates in X norm, independent of the size of the reduced space.

Thus, when performing an EIM (or any other reduction) algorithm on such terms, taking as threshold some estimation of $d_N(\mathcal{M}, X)$ does not guarantee a correct approximation R^{lim}. Nevertheless, discussing the question of the n-width of $d_n(R^{lim}(\mathcal{M}))$ is still relevant: we need to fit the computation of this term into the ROM framework. We make a few preliminary remarks:

- this term is inherently linked to the underlying mesh and corresponding truth approximation space. The n-width at hand is thus a "discrete" n-width
- its action is local
- the solutions we are trying to stabilize live in the projection of the solution manifold onto some reduced space $\mathcal{M}^N := \Pi_{X^N}\mathcal{M}$ and not in $\mathcal{M}$

As a result, the natural n-width to estimate is the following:

$$d_n(R^{lim}(\mathcal{M}^N), l^\infty),$$

and for these terms to correctly enter the ROM framework, we need something like:

$$\exists n \ll \mathcal{N}, \quad \text{s.t } d_n\left(R^{lim}(\mathcal{M}^N), l^\infty\right) \ll d_0\left(R^{lim}(\mathcal{M}^N), l^\infty\right), \tag{1.22}$$

or for some other right hand side that correctly characterizes the size of the limitation term. We provide now one example that illustrates the fact that for some standard discretization scheme, and their associated stabilization methods, a property such as equation (1.22) is not verified. We will deduce that trying to use such methods in a ROM framework is not a good strategy.

Let a one dimensional transport equation. The parameter is chosen as the convection parameter:

$$u_t + \mu u_x = 0.$$

Denote $u(\cdot, t; \mu)$ the fine solution transported at speed μ, at time t. The initial condition is chosen to be smooth to ensure the smallness of the n-width of the continuous solution manifold $\mathcal{M} := \{u(\cdot, t; \mu),\ t,\ \mu \in \mathcal{D}\}$. Let ϵ be some prescribed accuracy. One can find some N dimensional vector space X^N such that:

$$\forall u \in \mathcal{M},\ \exists u^N \in X^N, \quad \text{s.t } \|u - u^N\| \leq \epsilon.$$

The question that arises is how to construct a stable procedure $u(\cdot, t; \mu) \mapsto u(\cdot, t + \delta t; \mu)$. The situation is illustrated in Figure 1.7. The solid line is the trajectory of the continuous (or equivalently stabilized truth) solution. The plane represents the low dimensional linear space associated to some RB or POD basis: X^N. The dotted line is the best projection error. A well conducted offline phase guarantees that the error between dotted and solid line remains small, here bounded by the chosen ϵ. The problem is that the solution of a numerical scheme without proper stabilization might diverge. This is illustrated by the dashed line and we have denoted u^w the reduced solution when no proper stabilization is provided.

We now describe an hypothetical situation, that will help us understand the issue at hand. Say that the truth numerical scheme we are using uses forward Euler time discretization and a space discretization operator denoted $\mathcal{L}$. For each time step t^n, we compute the truth solution u^{n+1} as:

$$u^{n+1} = u^n + \Delta t \mathcal{L}(u^n).$$

A standard way of guaranteeing the TVD nature of numerical approximations to conservation laws, is to use flux limiters. We roughly state the goal, and refer to specialized literature for

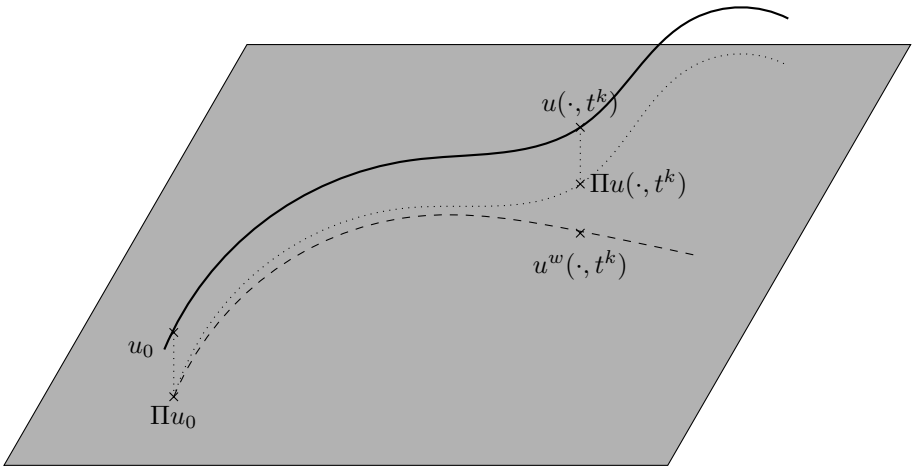


Figure 1.7: Solid line: truth solution; Dotted line: Best projection onto some reduced basis; Dashed line: Output of a system with no proper stabilization

a more precise description of this problem, see for instance [62]. Godunov's theorem state that linear high order[13] schemes for the resolution of conservation laws are not monotone, and thus subject to stability issues. A reasonable choice is thus to use the high order scheme everywhere, except where it behaves poorly, i.e where solutions exhibit sharp gradients or reach a local extremum. At those locations, the flux limiters enforce the use of a first order scheme. For our current problem, as we are dealing with smooth solutions, we consider that our truth discretization differential operator $\mathcal{L}$ is composed of a sum of one linear component, the high order spatial discretization operator, denoted A^{truth}, plus a non linear correction term, a local ingredient that only acts at local extrema, denoted R^{lim}. We have:

$$u^{n+1} = u^n + \Delta t \left(A^{truth}(u^n) + R^{lim}(u^n) \right).$$

For simplicity, suppose that the solutions in $\mathcal{M}$ only posses one local maximum. The stabilization term is then given by:

$$\forall \mu \in \mathcal{D},\ \forall t,\ R^{lim}(u(\cdot, t; \mu)) \sim \alpha \left(u(\cdot, t; \mu) \right) \delta_{p_{max}(t;\mu)},$$

where $p_{max}(t; \mu)$ is the index of the control volume where the maximum of the solution for parameter μ and time t is located and δ is the standard Kronecker delta. Denote by $N^{lim}(\mathcal{D})$ the cardinality of $\{p_{max}(t; \mu),\ t,\ \mu \in \mathcal{D}\}$. We expect $N^{lim}(\mathcal{D})$ to be of order $\mathcal{N}$, the size of the underlying truth discretization space.

Let us try to construct a reduced scheme mimicking this stabilization procedure. For each time step t^n, we compute the reduced solution $u^{N,n+1}$ as:

$$u^{N,n+1} = u^{N,n} + \Delta t \Pi_{X^N} \mathcal{L}(u^{N,n}).$$

Denote $A^{RB} = \Pi_{X^N} A^{truth}$. We have:

$$u^{N,n+1} = u^{N,n} + \Delta t A^{RB}(u^{N,n}) + \Delta t \Pi_{X^N} \left(R^{lim}(u^{N,n}) \right). \tag{1.23}$$

[13] High order means here more than 2nd order

Because of the structure of R^{lim}, there is no hope in reducing its computational cost. Indeed, we can explicitly compute its n-width:

$$d_n(R^{lim}(X^N), l^\infty) = \begin{cases} \max_{t;\mu}\{\alpha(u(\cdot,t;\mu)\} \text{ if } n < N^{lim}(\mathcal{D}) \\ 0 \text{ if } n > N^{lim}(\mathcal{D}). \end{cases} \tag{1.24}$$

We thus expect no decay before $n \approx \mathcal{N}$.

One can argue that trying to reproduce the corrective term alone is a naive approach, and that a more reasonable one is to reproduce directly the limited term over Ω. That is, one can try and perform an EIM procedure on $\mathcal{L}(X^N)$. We provide a partial answer in what follows. The assumption of the form of the scheme is that:

$$\mathcal{L}(X^N) = A^{truth}(X^N) + R^{lim}(X^N).$$

For all p, denote E_p a generic linear space of dimension p embedded in X. The n-width of the manifold of intereset, is given by:

$$d_n(\mathcal{L}(X^N), l^\infty) := \inf_{E_n} \sup_{f\in\mathcal{L}(X^N)} \inf_{g\in E_n} \|f-g\|_{l^\infty},$$

and it satisfies for all n:

$$\begin{aligned} d_n(\mathcal{L}(X^N), l^\infty) &\geq \inf_{E_n} \sup_{f\in A^{truth}(X^N)+R^{lim}(X^N)} \inf_{g\in E_n+A^{truth}(X^N)} \|f-g\|_{l^\infty} \\ &\geq \inf_{E_n} \sup_{f\in R^{lim}(X^N)} \inf_{g\in E_n+A^{truth}(X^N)} \|f-g\|_{l^\infty} \\ &\geq \inf_{E_{n+N}} \sup_{f\in R^{lim}(X^N)} \inf_{g\in E_{n+N}} \|f-g\|_{l^\infty} \\ &= d_{n+N}(R^{lim}(X^N), l^\infty). \end{aligned}$$

The first inequality comes from the fact than $E_n \subset E_n + A^{truth}(X^N)$. The last inequality comes from the fact that the infimum over all spaces of dimension $(n+N)$ is better than the infimum over all spaces of dimension n plus $A^{truth}(X^N)$. For our one dimensional transport equation, this means that the Kolmogorov n-width, measured in the l^∞ norm, of the limited flux is constant for $n < \mathcal{N} - N$. This is not satisfactory from a ROM point of view.

Remark 10 *The previous analysis of course does not constitute a proof of the impossibility to create a stable, reduced scheme for this class of problems. For instance, it is very easy to taylor a reduced scheme that can handle the transport of smooth quantities. Nevertheless, we have provided evidences that tend to show that we can not guarantee the reconstruction of local, stabilization terms. This will be further discussed in section 1.5.5.*

1.5.3 Instability for Navier-Stokes equation

In this section we discuss another common source of instability in CFD codes, the one associated with the numerical approximation of solutions to the Navier-Stokes equation. Let $\bar{\cdot}$ be some filtering operation. It can be:

- a spatial filtering operation, as in the Large Eddy Simulation (LES) methods
- a time filtering operation as in the Reynolds Averaged Navier Stokes (RANS) simulations
- the orthogonal projection onto some well chosen reduced basis, still denoted X^N, for RB type methods

For LES and RANS methods, the filtering operation can be done using some convolution kernel. More precisely, denote for all state variable w:

$$\bar{w} := G * w,$$

where G any convolution kernel and $*$ is the standard convolution operator in time (RANS) or in space (LES).

Whatever the filtering operation chosen, the pair $(\bar{u}, \bar{p})$ is solution to the following system:

$$\left\{ \begin{array}{rcl} \frac{\partial \bar{u}}{\partial t} + \bar{u} \cdot \nabla \bar{u} - \mu \Delta \bar{u} + \nabla \bar{p} & = & \bar{u} \cdot \nabla \bar{u} - \overline{(u \cdot \nabla u)} \\ \nabla \cdot \bar{u} & = & 0. \end{array} \right. \tag{1.25}$$

In all three cases, (LES, RANS and RB), the system (1.25) is underdetermined. This is due to the filtering of the quadratic term $u \cdot \nabla u$. The right hand side of the first equation is called the residual stress tensor. It represents the interaction between the resolved and unresolved scales, and needs to be modeled in order to close the system.

1.5.3.1 Closure for LES and RANS methods

One of the approaches that can be considered is to set the residual stress tensor to zero. For sufficiently fine meshes, this leads to stable schemes, as all scales are resolved: this is called Direct Numerical Simulation (DNS) in the literature. For reasonable (computational wise) mesh size, these schemes are often instable. Figure 1.7 is also a good illustration of this type of instability. The dashed line u^w is in this context the output of a reduced scheme that uses zero residual stress tensor.

To my knowledge, there is consensus to interpret this instability, at least for LES type methods. The starting point is the existence of an energy cascade between the different length scales coexisting, see for instance [86, 79]. More precisely, the conjecture is that there is creation of energy at large scales and that the mechanism ensuring global stability is the dissipation occurring at small scales. Because of the filtering operation, the small scales are not resolved. The conclusion is that the methods with zero residual stress tensor lack a dissipation mechanism. To our knowledge, the numerical study of this cascade mechanism has received little attention. We mention here one of the results we are aware of. In [79], they study of the spectral decomposition of Navier-Stokes solutions and show that the energy transfers[14] are local, in the wave number space.

Following these considerations, the most common option is to model the residual stress tensor with a viscous term. A profusion of closure models are available in the literature. We mention a few of them:

- for RANS, common choices are the $k - \epsilon$ and $k - \omega$ models
- for LES: Smagorinsky, Variational Multiscale (VMS), Dynamic Subgrid Scale etc.

For the most advanced LES models, the eddy viscosity is a highly non linear function of the state variables and needs to be recomputed at each time step and in each cell of the physical domain. In the Dynamic Subgrid Scale, it is even more complicated, since you are using two different filters corresponding to different cut off length scales, each of them with its own eddy viscosities model.

[14] These results depend of course of the way you define the energy transfer between modes.

1.5.3.2 Theoretical results

We now mention another class of method, not directly applicable to Navier-Stokes equation, but that is backed by firm theoretical results. These have been developed studied in [129] and related references. We briefly present the results, but stay at a formal level since the rigorous proofs are quite involved and far from the scope of this thesis.

The class of problems considered is the class of dissipative partial differential equations. That is, the differential operator $\mathcal{L}$, see (1.2), can be splitted into two contributions, one linear viscous term and one non linear term. We also require the linear part to be dominating, in some sense. Let $\mathcal{L}$ such a generic differential operator. Denote A the linear contribution and R the non linear part. Equation (1.2) becomes:

$$\frac{\partial u}{\partial t} + A(u(t)) + R(u(t)) = F. \tag{1.26}$$

The starting point of the method is to consider the (complete) set of eigen functions of A in X. Denote $\{\phi_i,\ i = 1, \dots, \infty\}$ this basis, ordered in decreasing eigenvalues. In the remainder of this section, we denote:

- P_N the orthogonal projection onto span $\{\phi_i,\ i = 1 \dots N\}$
- Q_N the projection onto span $\{\phi_i,\ i > N\}$
- $p = P_N u$ and $q = Q_N u$

With this notation, the objective appears clearly. We want to solve for the high energy modes components, denoted p. The other components, stored in q, are the so-called unresolved scales. We project (1.26) onto the two orthogonal spaces $P_N X$ and $Q_N X$ and get a system equivalent to the initial problem:

$$\left\{ \begin{array}{rcrcrcl} P_N \frac{\partial u}{\partial t} & + & P_N A(u(t)) & + & P_N R(u(t)) & = & P_N F \\ Q_N \frac{\partial u}{\partial t} & + & Q_N A(u(t)) & + & Q_N R(u(t)) & = & Q_N F. \end{array} \right.$$

We use the fact that the basis $\{\phi_i\}_i$ is independent of time, and that the projection operators P_N and Q_N commute with A. We get a coupled system equivalent to (1.26):

$$\left\{ \begin{array}{rcrcrcl} \frac{\partial p}{\partial t} & + & A(p(t)) & + & P_N R(p+q) & = & P_N F \\ \frac{\partial q}{\partial t} & + & A(q(t)) & + & Q_N R(p+q) & = & Q_N F. \end{array} \right.$$

To get familiar with the notation, we can reformulate some standard methods in this framework:

- a zero residual stress tensor method uses $P_N R(p+q) = P_N R(p)$
- eddy viscosity closure models assume that:

$$P_N R(p+q) = P_N R(p) + \nu_{eddy}(p) A(p).$$

We now describe the ideal case. Suppose you are able to construct a function $\Phi : X \to X$ such that the system's dynamics are exactly captured by

$$\frac{\partial p}{\partial t} + Ap + P_N R(p + \Phi(p)) = P_N F. \tag{1.27}$$

Then, the knowledge of Φ gives us a well posed problem on the high energy modes p. In [54], it is shown that when the diffusion operator A has a spectral gap, such a function Φ exists, at least asymptotically in time. The graph of Φ is in that situation called inertial manifold.

There is no spectral gap for the two dimensional Navier-Stokes equation, and there is yet no proof of any similar property. To my knowledge, the strongest results are the existence of determining forms, see for instance [53, 52] and the references therein. Again, these results are true asymptotically in time. To put it simply, they say that the state of the two dimensional Navier-Stokes system is completely determined by a finite (small) number of degrees of freedom. It is obvious that this is not as strong as (1.27), and not sufficient to construct self sufficient schemes.

1.5.4 Closure models for RB methods

The closure of system (1.25) is an open question for RB type methods, and no standard procedure has yet taken over. The first idea is to take a zero residual stress tensor, and hope that the situation is better than for LES. One reason for that would be the existence of an intrinsic stability due to the fact that the reduced solution manifold only contains physically relevant solutions/structures. The corresponding well behaved trial and test spaces should thus give a more stable numerical scheme than say a full Finite Element method. This is true to some extent. For instance, in one of the numerical experiments of chapter 3, we show that the CFL condition is much less stringent for the reduced scheme, than the one of the fine scheme. Unfortunately, this intrinsic stabilization mechanism does not seem to be enough. It is now well accepted in the ROM community that a scheme with zero residual stress tensor generally fails for two and three dimensional Navier-Stokes problems, see for instance [138].

The concept of energy cascade introduced for LES also applies in this context. In [37], they numerically study the energy transfers between POD modes for the backward facing step problem. They show that for integer pairs (n, m) such that $|m - n| > 25$, the energy transfers are neglectable. The underlying idea is that high energy POD modes contain large structures, whereas low energy modes contain small structures, and so that the discussion about energy cascade for LES also applies here.

The specific form of the filtering operation in the RB case has lead to new interpretations of the instability. For instance, in [113, 84] they construct an enlightening example that we have chosen to include here. It is a simple finite dimensional example that tries to give a geometrical interpretation of the instability[15] The high dimensional complete system considered here is three dimensional, and possess a stable limit cycle, captured by a low dimensional subspace (here two dimensional). We have plotted typical trajectories in Figure 1.8. This stable limit cycle is contained in the $z = 0$ plane and is represented as a thick line. The problem has been chosen such that the projected system onto the $z = 0$ subspace is unstable. We show in Figure 1.9 a graphical illustration of the mechanism at hand. The plane on which the scheme is projected can be seen as the space spanned by the high energy POD mods (or first RB modes). The projection of the scheme is unstable, as there is creation of energy. The physical stabilization mechanism, here the diffusion that occurs at small scales, is materialized by the z direction. This point of view have lead to the development of geometrical answers such as the one proposed in [40]. They are interested in steady state solutions and they advise adding transient snapshots to the basis. This amounts to adding a neighborhood of the low dimensional subspace.

[15] Note that the instability of the toy example discussed in this section is not of the same nature as the one of a Galerkin method for Navier-Stokes equation. The system described here is not linearly stable.

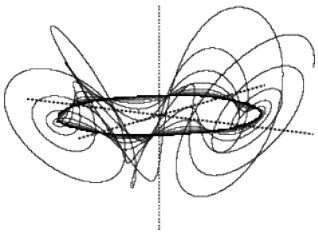


Figure 1.8: Trajectories for the finite dimensional model problem

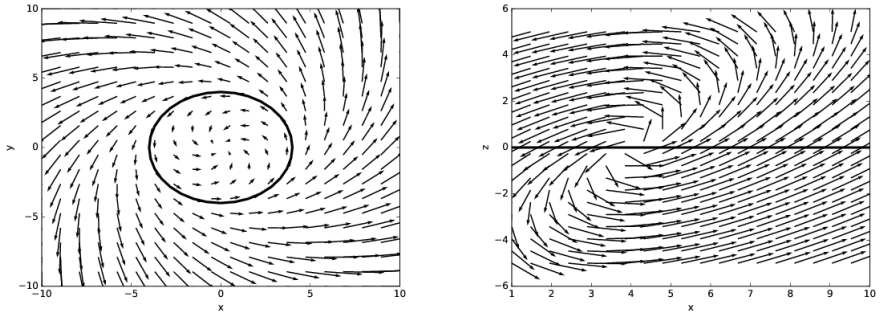


Figure 1.9: Analysis of the model problem. Left: projection onto the $x - y$ plane. Right: projection onto the $x - z$ plane

We cite one more result related to this subject. It can not be applied to the resolution of Navier-Stokes equation, but is a theoretical result on a closely related issue. In [112], they give sufficient conditions on $\mathcal{L}$ so that the error of the projected system is controlled by the best projection error. Unfortunately, the constraints are far from any real life application. For instance, one of the requirements is the Lipschitz continuity of the system in the direction orthogonal to the space of projection.

We conclude this section with the most common methods. As already mentioned eddy viscosity methods are by far the most widely used approaches in LES and RANS methods. They have naturally been transposed to ROM. We mention a few references [138, 18, 71], but note that there is an extensive literature on the subject.

1.5.5 Study of reduced basis for CFD numerical results in the literature

In this section we present a sample of numerical results taken from the literature. These have been carefully chosen to illustrate some of the hypotheses/conclusions that we have drawn in the previous sections. The first class of problems that we have chosen to discuss are the ones for which the truth scheme uses zero residual stress tensor, and no additional numerical stabilization. This happens when the underlying mesh is fine enough to capture all scales. In this situation, we expect that a reduced scheme which also uses zero residual stress tensor and no additional stabilization mechanism to both give satisfactory results and be computationally efficient. Recent, preliminary results tend to confirm this conjecture, see [98, 124].

The next class of examples we have chosen are situations for which the truth schemes uses some sort of stabilization that correctly enters the ROM framework. One example can be found in a recent paper [33]. They study the backward facing step for moderate Reynolds, 50 to 450, and use a Smagorinsky eddy viscosity model. This term is linked to the underlying discretization space. In each cell K, one adds an eddy viscosity ν^K_{eddy} such that:

$$\nu^K_{eddy} \propto h_K |\nabla u|_K$$

where h_K is the characteristic size of mesh element K and $|\nabla \cdot |_K$ is the Frobenius norm of the gradient in element K. We are in the situation of equation (1.23) and we need to keep in mind that the Smagorinsky contribution is acting at the consistency error scale, and not at the approximation error scale. The Frobenius norm of the gradient of a state variable is rougher than the state variable itself. We thus expect its **relative** n-width to decay slower. Nevertheless, because of the specific problem studied (there are no complex structures propagating), the situation is not as bad as the one described in 1.5.2.2. The numerical results show that an EIM basis of size 73 is enough to guarantee a small relative error for the Smagorinsky term. By comparison, the RB needed to represent the solution manifold is of size 17. As the reduced scheme correctly reproduces all the contributions of the fine solver, the resulting reduced scheme gives good results.

The next step it to solve problems with more complex stabilization mechanisms. This has been done for instance in [29, 11, 31]. More complex can in this context mean the use of local/directional quantities such as upwinded flux, utilization of approximate Riemann solvers, directional gradient reconstruction (ENO or WENO schemes for instance) in a FV context, characteristics-Galerkin method (see [22]) in the FE context. We once again use the notation introduced in 1.5.2.2. The first step in all the references mentioned is to compute the contribution $\mathcal{L}(u^{N,n})$ exactly at each time step. Such a method has a computational complexity of the same order as the original fine scheme. This make the usage of the RB framework dubious. Nevertheless, the fact that the resulting schemes give good results guarantees that the cumulative approximation error is not too big of an issue, even for very challenging CFD problems. At this point, the problem seems to have narrowed down to the reduction of the computational cost of $\mathcal{L}(u^{N,n})$ at each time step.

To study this issue, we focus on the final numerical experiment presented in [31], but similar considerations could be derived by looking at [29, 11]. The objective is to compute the flow around the Ahmed body, for one fixed set of parameters. The fine mesh they used has several million nodes. The fine scheme used to produce the training set involves several local ingredients. For instance, it uses a Roe scheme to discretize convective fluxes. The unsteady simulation they are trying to reproduce has around 1200 snapshots. We denote by $\mathcal{M}$ the solution manifold $\{u^n,\ n \in [1, \dots, 1200]\}$. They show that a basis of size 283 captures 99.99% of the energy. In order to reduce the complexity of $\mathcal{L}(u^{N,n})$ at each time step, they have applied a reduction algorithm (here gappy POD) directly. As $\mathcal{L}$ involves local ingredients, and following the discussion of 1.5.2.2, we expect the n-width of stabilization terms appearing in the numerical scheme to reach a plateau for $n < \mathcal{N} - N$, for $\mathcal{N}$ in the order of millions. They find that using a subsample of 1500 nodes in the gappy POD is enough to obtain a stable scheme. At first, this does not seem to match our conclusions.

Before going further, we repeat something that has already been been mentioned in section 1.5.1.2, when discussing solution manifolds with large n-widths. When an RB type method is used only to reconstruct the training set of small cardinality, the overall complexity is not in $\mathcal{N}$ the number of spatial degrees of freedom, but rather in N^{snap} the number of snapshots in the training set. For instance, for the TVD example of section 1.5.2.2, this means that $N^{lim}(\mathcal{D})$ is

of order N^{snap}, and not of order $\mathcal{N}$.. We thus expect the plateau behavior to stop at $N^{snap} - N$ instead at $\mathcal{N} - N$.

Back to the Ahmed body problem. My interpretation is the following: as the space $\{\mathcal{L}(u^n),\ n \in [1, \dots, 1200]\}$ is roughly of dimension 1200, one needs to find enough sampling points $\{x_i\}$ so that the matrix $\{u^n(x_i),\ (i,n)\}$ is of rank 1200. For this three dimensional challenging example, because of conditioning, the actual number of points needed for the gappy POD is 1500. The gappy POD can then be understood as the following application:

$$\begin{cases} \{u^n,\ k \in [1, \dots, 1200]\} & \to & X \\ \{u(x_i),\ i \in [1, \dots, 1500]\} & \mapsto & \mathcal{L}(u). \end{cases}$$

If this interpretation is correct, this is not satisfactory from a ROM point of view, as this method would not work for a multiparameter setting, and because this is very sensible to the time discretization.

1.6 A few methods to deal with the issues

We now propose two classes of methods that are partial answers to the problems mentioned in section 1.5. The first one uses the fact that the reduced scheme does not need to be the same as the fine scheme. The second class of methods is of a different nature. It handles manifolds with large n-widths by adding a pre-conditioning step.

1.6.1 A new class of stabilization mechanisms

One possible direction is to use in the online section a different scheme than the one used in the offline, learning phase. This constitutes a natural approach after the discussion of section 1.5.2. One can for instance use, in the reduced scheme, a different, simpler, eddy viscosity model than the one used for the truth scheme. We mention a few references that perform such procedure [73, 138, 107, 140] and the references therein. In a recent paper [50], another strategy is advised. They enforce stability by using a constrained Galerkin method. The constraints are chosen in the offline phase. The initial numerical results are promising, and additional tests need to be performed to get a deeper understanding of the mechanism at hand.

Whatever the ROM stabilization mechanism chosen, one important ingredient was missing. It has been recently provided in [92]. It explicits the key hypothesis when using this kind of strategy. Their setting is the following: during the offline phase, they use a fine scheme with SUPG type stabilization, unfit for ROM. On the other hand, their reduced scheme uses a simpler stabilization, adapted to ROM. More precisely, they add viscosity increasingly with the mode number, a variant of the Spectral Vanishing Viscosity (SVV) method [128]. As expected, the solution obtained with the reduced scheme happens to be different from the one obtained with the fine discretization. They reconcile the two by stating that there is a one to one correspondence between the two stabilized solution manifolds. The truth solution is then retrieved using a so called rectifier, a one to one mapping between the SUPG solution manifold and the SVV solution manifold[16]. This rectifier is constructed during the offline phase.

In my opinion, one ingredient could be added to the offline stage, to give a more physically sound stabilization mechanism. Suppose that you know that you want/need to add diffusion at

[16] An interesting open question is whether this concept of 'underlying truth solution, independent of the stabilization mechanism', can be interpreted in terms of determining forms, see [52]

some prescribed scales. The following algorithm could be implemented:

Data: Set of snapshots $\{u(\mu_i)\}$, spatial scale ϵ
Result: Adapted Reduced Basis Ψ^ϵ
Filter the snapshots: $\{u(\mu_i)\} \to \{\bar{u}^\epsilon(\mu_i)\}$;
$\bar{\Psi}^\epsilon_k :=$ POD $\{\bar{u}^\epsilon(\mu_i)\}$;
$\forall i,\ \tilde{u}^\epsilon(\mu_i) := u(\mu_i) - \Pi_{\bar{\Psi}^\epsilon} u(\mu_i)$;
$\tilde{\Psi}^\epsilon_k :=$ POD $\{\tilde{u}^\epsilon(\mu_i)\}$;
$\Psi^\epsilon := \bar{\Psi}^\epsilon \cup \tilde{\Psi}^\epsilon$;

Algorithm 3: Construction of basis adapted to SVV

The resulting basis Ψ^ϵ is perfectly adapted to the Spectral Vanishing Viscosity method. Indeed, by playing with the parameter ϵ, one can choose the scales that need to be damped and how they should be damped. It is obvious that Algorithm 3 can be extended to a version with multiple spatial scales using different damping coefficients. This method should add less numerical viscosity than a standard SVV to the reduced solutions, and should thus lead to a better behaved rectification step.

I mention one other approach that fits the ROM framework, and that could be numerically investigated. It follows the presentation of section 1.5.3.2, and we will use the notations defined there. Even if no 'inertial manifold type' results are available for the two dimensional Navier Stokes system, this sound framework can inspire new numerical stabilization algorithms. We propose one here, inspired by the one developed in [130]. Let $M > N$. We replace Q_N the projection onto all unresolved scales by the projection P_M onto the largest unresolved scales. We are looking for Φ^M:

$$\Phi^M : \begin{cases} P_N X & \to & (P_M - P_N)X \\ p & \mapsto & \Phi^M(p), \end{cases}$$

that mimics the behavior of the application Φ in the inertial manifold case. We can for instance take it as the limit of a fixed point algorithm:

$$\Phi^M : p \to A^{-1}\left(P_M f - P_M R(p + \Phi^M(p))\right). \tag{1.28}$$

This involves taking the projection of the residual onto the largest non resolved scales and inverting the dominant part of the operator. There are no major obstacles to fit those computations into a ROM framework. Also, it matches the energy cascade principle. Indeed, this method models the interaction between the resolved scales and the largest unresolved scales. The other unresolved scales (for $n > M$) should not impact the stability of the method. This method is not backed by any rigorous results and, of course, we have no guarantee that this would be more stable than for instance a raw Galerkin scheme using the first M modes. I nevertheless believe that it is worth some numerical investigation. A version of this algorithm was tested numerically on a one dimensional reaction diffusion equation in [51].

1.6.2 Calibration

Another direction is to transform the original problem, in order to enforce the smallness of the n-width of the problematic terms. For discontinuous solutions, see section 1.5.1.2, this means forcing the position in the mesh of the singularity. The same idea can be applied for stabilization terms. For instance, for flux limitation, this means forcing that the position of the maximum, so that it remains always positioned in the same mesh element.

In this introductive chapter, we stick to a quick glance at the method, as it is actually the main contribution of this thesis, and will thus be extensively described in chapters 4 and 5. We

consider the transport equation again. This time, say we are able to recenter the maximum at each time step. We denote $\tilde{\mathcal{M}}$ the resulting modified solution manifold. The corrective term that needs to be added to the scheme is now concentrated in the same cell p_0, whatever the member $\tilde{u}$ of $\tilde{\mathcal{M}}$ considered: $R^{lim}(\tilde{u}) \propto \delta_{p_0}$. The Kolmogorov n-width is (greatly) reduced, whatever the norm $\|\cdot\|$ considered:

$$d_n(R^{lim}(\tilde{\mathcal{M}}), \|\cdot\|) = 0 \text{ for } n \geq 1.$$

This can be compared to the n-width in the original case, see equation (1.24). Moreover, we have a bonus property, R^{lim} is now a linear function of u.

Remark 11 *The linearity does not appear like a useful property for flux limitation. If you transpose this discussion to upwinding, or any directional procedure, then it becomes a valuable property. This will be discussed in chapter 5.*

Appendix

Let Ξ be some proper sampling of $\mathcal{D}$, of cardinality N^{snap}. The objective is to extend the proof of the existence of an optimal POD basis to the case $X = L^2(\Omega)$. Define r as:

$$r : \left\{ \begin{array}{ccc} \Omega \times \Omega & \to & \mathbb{R} \\ (x,y) & \mapsto & \sum_{k=1}^{N^{snap}} u(x;\mu_k)u(y;\mu_k) \end{array} \right.$$

The optimal basis, if it exists, is given by eigen functions of the following operator :

$$\mathcal{R} : \left\{ \begin{array}{ccc} X & \to & X \\ \psi & \mapsto & x \mapsto (\int_\Omega \psi(y) r(x,y) dy). \end{array} \right.$$

The existence of a complete eigenfunction set of $\mathcal{R}$ is a consequence of the Hilbert-Schmidt theorem, see for instance [114]. The first thing is to prove that the image of $\mathcal{R}$ is included in $L^2(\Omega)$. We then need to show that $\mathcal{R}$ is a bounded, self-adjoint and compact operator $L^2(\Omega) \to L^2(\Omega)$.

$$\forall \psi \in L^2(\Omega),\ \|\mathcal{R}\psi\|_{L^2}^2 = \int_\Omega \left[\int_\Omega \psi(y) r(x,y) dy \right]^2 dx$$

Cauchy-Schwarz's inequality in L^2 gives

$$\forall \psi \in L^2(\Omega),\ \|\mathcal{R}\psi\|_{L^2}^2 \leq \|\psi\|_{L^2}^2 \int_\Omega \int_\Omega r(x,y)^2 dy dx$$

We then use the fact that r is separable, and the Cauchy Schwarz inequality in $\mathbb{R}^{N^{snap}}$:

$$\forall \psi \in L^2(\Omega),\ \|\mathcal{R}\psi\|_{L^2} \leq \|\psi\|_{L^2} \sum_{k=1}^{N^{snap}} \|u(\mu_k)\|_{L^2}^2$$

This concludes that $\mathcal{R}$ is a continuous function $X \to X$. Using Fubini, we get the self-adjoint property. Compactness is trivial, as the image of $\mathcal{R}$ is a finite dimensional subspace of X.

Chapter 2

Domain decomposition in ROM context, for configurations with varying structures

---

The objective of this chapter is to propose a ROM based method to solve problems with challenging geometry variations, in the context of CFD. Our target application has been discussed in the introduction of this thesis, chapter 1, and the corresponding illustration can be found in Figure 1.2. In this chapter, we start with a quick overview of the ROM methods available in the literature that are designed to handle geometry variations. We then show that there are important properties missing before being able to solve the target problem. This analysis leads us to propose a variation of the RBEM method, both more flexible and computationally more challenging than the original version. After discussing the a priori estimates, we conclude by discussing possible implementations and the overall computational complexity.

2.1 Introduction

The objective of this chapter is to tackle one of the issues raised when discussing the initial goal of this thesis, see the introductive chapter 1. We show in Figure 2.1 the target problem that we will be focusing on. It can be seen as a model version of the original problem. The wind turbines have been replaced by cylinders. Also, we replace Navier-Stokes equation with the following model convection diffusion equation, with Dirichlet boundary conditions

$$\left\{ \begin{array}{rcll} -\alpha\Delta u + \beta \cdot \nabla u & = & 0 & \text{in } \Omega \\ u & = & u_D & \text{on } \partial\Omega. \end{array} \right. \tag{2.1}$$

These simplifications will allow us to focus on the unusual parameter dependency. It is the number N_{obs} and position of the obstacles inside Ω:

$$\mathcal{D} = \{x_k \in \Omega,\ k \in [1 \dots N_{obs}]\}\,.$$

The physical domain will be denoted from now on $\Omega(\mu)$. The solutions to (2.1) onto $\Omega(\mu)$ will be denoted $u(\mu)$. With this unusual parameter dependency comes a new set of questions.

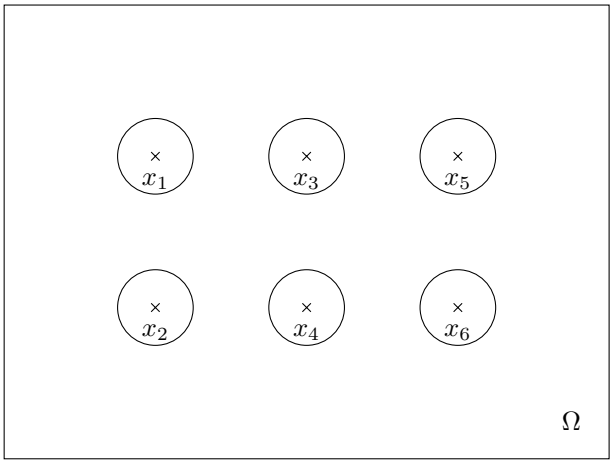


Figure 2.1: The chosen model problem. Here, $N_{obs} = 6$

What is the correct reduced space ? Indeed, the standard solution manifold $\mathcal{M}$ defined as:

$$\mathcal{M} := \{u(\mu), \mu \in \mathcal{D}\}$$

is not embedded in any obvious Hilbert space. We can reduce this question to a simpler one: how do you compare solutions with different number of obstacles?

The intuition is that the correct space to look at is the manifold of the restriction of solutions on subdomains of Ω. One can for instance study $\hat{\mathcal{M}}$ defined as:

$$\hat{\mathcal{M}} := \{u(\mu)|_{\Omega_k}, \mu \in \mathcal{D},\ \Omega_k \text{ some subdomain of } \Omega\}.$$

An illustration is depicted in Figure 2.2. Following this initial remark, we expect that solving the

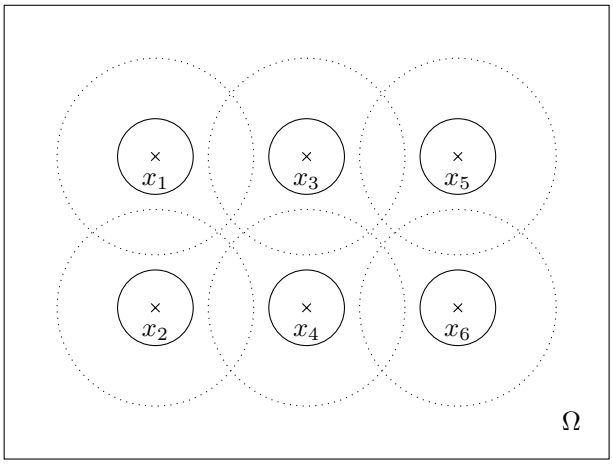


Figure 2.2: An illustration of the concept of solution partitioning

target problem will involve matching solutions defined on neighboring domains. This is exactly the purpose of domain decomposition methods.

The first section will be a discussion on existing domain decomposition methods. More specifically we will give a short review of the intersection between domain decomposition and reduced order modeling. We will explicit the reasons why none of the existing methods is adapted for our specific need, and conclude on specifications required to solve the target problem. The rest of the chapter will be devoted to the analysis of the proposed method, as well as computational considerations.

2.2 ROM and domain decomposition

This section will start as most talks and lectures on domain decomposition methods, by presenting their origin. Suppose that you know how to solve a PDE separately on elementary domains, given any boundary conditions. Can you solve the same PDE on a combination of the elementary domains ? The situation where there are two elementary domains, one circle and one square is presented in Figure 2.3. Many variations were developed around the method proposed initially

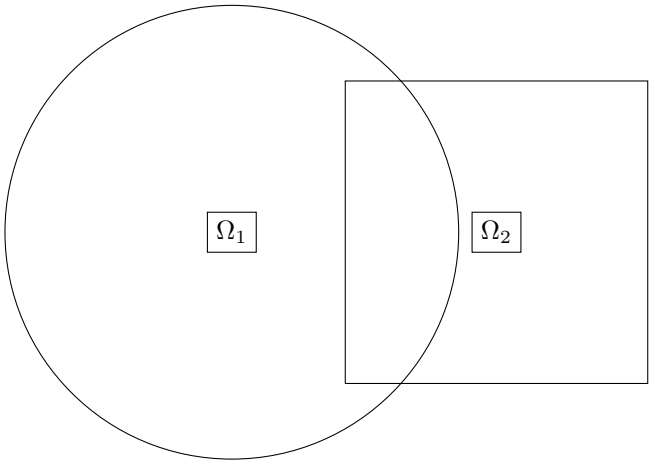


Figure 2.3: The original Schwarz problem

by Schwarz. As our purpose is not to review all existing methods, we will (very) briefly describe one of them, taken from [87].

Let u_1^0 and u_2^0 be some 'guessed solution' inside Ω_1 and Ω_2. The following iterative algorithm

$$\begin{array}{rcll} -\alpha\Delta u_i^{n+1} + \beta \cdot \nabla u^{n+1} & = & 0 & \text{in } \Omega_i \\ u_i^{n+1} & = & u_{(i+1)\%2}^n & \text{on } \Omega \bigcap \partial\Omega_i \\ u_i^{n+1} & = & u_D & \text{on } \partial\Omega \bigcap \partial\Omega_i \end{array}$$

converges towards the solution to the full problem (2.1) under conditions on the overlap $\Omega_1 \cap \Omega_2$. These ideas have found many applications and are widely used. They are the building block of the design of parallel PDE solvers, see for instance [45] and the references therein. As we will see, we use them here in a different setting.

To continue exploring the specificities of the problem at hand, we present a version of the Schwarz problem, closer to the target problem of this chapter. It will be used in the next section to illustrate the fact that existing methods are not suited to solve our problem. Let

a parametrized version of the problem depicted in Figure 2.3. The parameter is taken to be the distance between the barycenters of the elementary domains. We can put this into a ROM framework:

- denote μ the distance between the barycenter of the subdomains $\hat{\Omega}_1$ and $\hat{\Omega}_2$. Denote $\Omega_1 := \hat{\Omega}_1$ and $\Omega_2 := \hat{\Omega}_2(\cdot - \mu)$
- denote $\Omega(\mu)$ the global domain, for a parameter μ:

$$\Omega(\mu) = \Omega_1 \cup \Omega_2$$

- denote $\mathcal{D}$, the parameter space in which μ lives. It is some interval of $\mathbb{R}$ such that Ω_1 and Ω_2 overlap

This problem resembles the one that lead to the first domain decomposition method by Schwarz. We know how the restriction of the solution to both subdomains $\hat{\Omega}_1$ and $\hat{\Omega}_2$ behave. Can we propose a ROM strategy to quickly solve the combined parametrized problem ?

Remark 12 *The computational difficulties due to this specific parameter dependence already appear on this simple toy example. We expect this to be even worse when multiple domains are intersecting, as in the target problem. We will see in the numerical section how to mitigate these issues.*

We now recall some of the methods used in the ROM community to solve problems with geometric variations. Some of them will seem far from the target problem, but they will help understand the context in which this work takes place. We discuss the method in section 2.2.1 because it is the most widely used ROM method for variation of geometries. We try to apply it to the toy problem (Figure 2.3) and show that it is not adapted. The reasons why we discuss the method in section 2.2.3 will become clear at the end of this chapter.

2.2.1 Domain Deformation

One option to deal with geometric variations is to use domain deformation. More precisely, one can build an equivalent problem for which the parameter dependency appears in the variational form. A complete description can be found for instance in [110, 42]. It assumes the existence of a reference domain $\hat{\Omega}$ and of a family of smooth mappings $\{F_\mu\}$ such that the physical domains in which we want to solve the problem $\Omega(\mu)$ can be expressed as

$$\Omega(\mu) = F_\mu(\hat{\Omega}).$$

The mappings can be:

- affine: $x = G(\mu)\hat{x} + g$ where G is a $\mathbb{R}^{d\times d}$ matrix.
- non affine: $x = T(\mu, \hat{x})$ where T is a smooth mapping $\mathbb{R}^d \to \mathbb{R}^d$.

The geometric variability is then handled as follows: in an offline stage, perform the fine computation for several values of μ. Define the mapped solution manifold $\hat{\mathcal{M}}$ as:

$$\hat{\mathcal{M}} := \{u(\mu) \circ F_\mu,\ \mu\}.$$

Use standard compression algorithm on $\hat{\mathcal{M}}$. In an online stage, modify the variational form to make the problems on $\Omega(\mu), \mu \in \mathcal{D}$ equivalent to the problem on $\hat{\Omega}$.

Remark 13 *When the mapping is non affine, the additional terms in the variational form have non affine parameter dependence. The EIM method [14] can be used to reduce the computational cost.*

Is this idea adapted to solve the toy problem depicted in Figure 2.3 ? We need to choose a reference domain $\hat{\Omega}$ and smooth mappings F_μ such that $\forall \mu \in \mathcal{D}$, $\Omega(\mu) = F_\mu(\hat{\Omega})$. Let $\hat{\mu}$ be some parameter in $\mathcal{D}$. We take $\hat{\Omega} = \Omega(\hat{\mu})$. We take the affine by parts mapping that is the identity in some neighborhood of the circle and that linearly stretches the rectangle, away form the overlap. Let $\hat{x}_0$ be some abscissa, away from the overlap. Let $x_r(\mu)$ be the abscissa of the right edge of the rectangle. This situation is illustrated in Figure 2.4. We take the following mapping

$$\hat{x} \to \begin{cases} \hat{x} & \text{for } \hat{x} < \hat{x}_0 \\ \hat{x}_0 + (\hat{x} - \hat{x}_0)\frac{x_r(\mu) - \hat{x}_0}{x_r(\hat{\mu}) - \hat{x}_0} & \text{for } \hat{x} > \hat{x}_0. \end{cases}$$

This method is far-fetched in this situation; it is not using the underlying structure of the

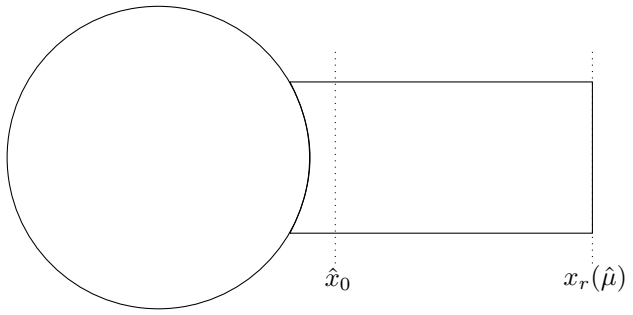


Figure 2.4: As possible reference mesh $\hat{\Omega}$

problem. Also, the extension to the target problem, with multiple intersections is not reasonable.

2.2.2 Reduced basis element method

The reduced basis element method (RBEM) is using ingredients that better match the target problem. It follows the following recipe:

- decompose the domain into non overlapping elementary blocks
- eventually use domain deformation onto well chosen reference blocks
- perform standard reduced order modeling to get a reduced basis onto each reference block
- glue solutions together using domain decomposition methods. The matching constraints on the interfaces can for instance be enforced using reduced basis, coarse finite element spaces or low order spectral spaces

This method was initially developed and tested on the heat equation [95]. Results for the two dimensional Navier-Stokes equation are presented in [90]. We also mention the static condensation reduced basis element method [133, 47] which is an 'industrial' version of the RBEM, as well as ArbiLoMod [24], a recently developed variant that allows for online local basis enrichment. All of these methods require a structured geometric decomposition. A quick look at Figure 2.1 will

convince the reader that the target problem lacks this property: we have no a priori idea of the geometry of the overlaps. The method we will propose is an extension of the RBEM, that deals with this issue.

2.2.3 Rotating obstacle

[131] handles another geometry related problem. The stages of the method proposed are:

- build local (reduced) basis around objects
- rotate the objects during the simulation
- match these local solutions to an 'outside solution'

The exact relation of this method to the target problem is not obvious. We refer to the discussion in the concluding section of this chapter for more details.

in Figure 2.5, we present a simplified version of their problem. The global domain Ω is decomposed into three non overlapping subdomains, Ω_0, outside, Ω_{int}, a rotating domain and Ω_{tampon} which makes the junction between the two other domains. Define the rotation operator

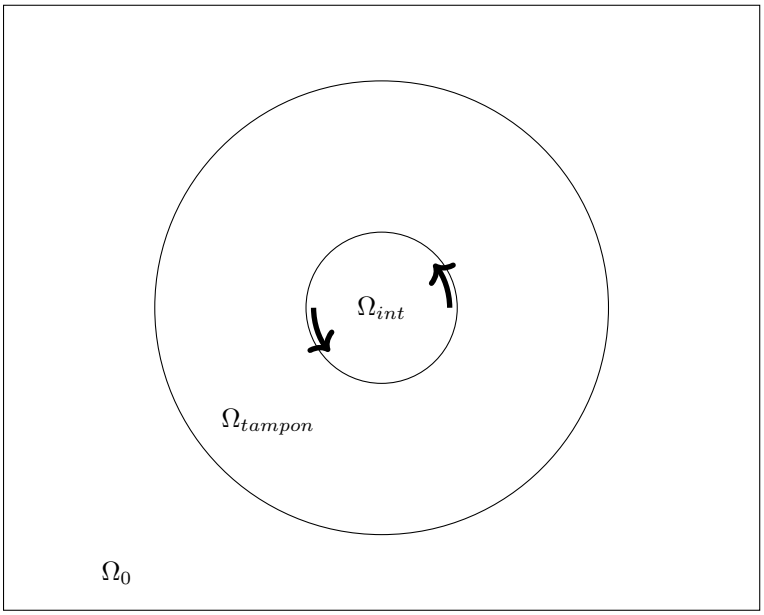


Figure 2.5: Method developed in [131]

r_θ as:

$$\forall \theta,\ r_\theta : \left\{ \begin{array}{lcl} \Omega & \rightarrow & \Omega \\ (x,y) & \mapsto & \left(\begin{array}{cc} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{array} \right) \left(\begin{array}{c} x \\ y \end{array} \right) \end{array} \right.$$

Let θ be some prescribed rotation parameter. They suggest using the following mapping:

$$x = F_\theta(\hat{x}) = \left\{ \begin{array}{ll} \hat{x} & \text{if } \hat{x} \in \Omega_0 \\ r_{\theta'}(\hat{x}) & \text{if } \hat{x} \in \Omega_{tampon}, \text{ where } \theta' = \theta \frac{r_{ext} - \|\hat{x}\|_2}{r_{ext} - r_{int}} \\ r_\theta(\hat{x}) & \text{if } \hat{x} \in \Omega_{int} \end{array} \right.$$

Let $\hat{\Omega}$ be some reference mesh such as the one presented Figure 2.6. The effect of the mapping

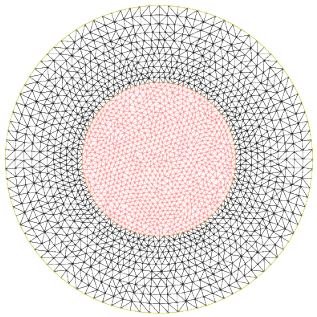


Figure 2.6: The chosen reference mesh. In red: $\hat{\Omega}_{int}$; in black: $\hat{\Omega}_{tampon}$

F_θ, for various values of the rotation angle θ is shown in Figure 2.7. This specific mapping

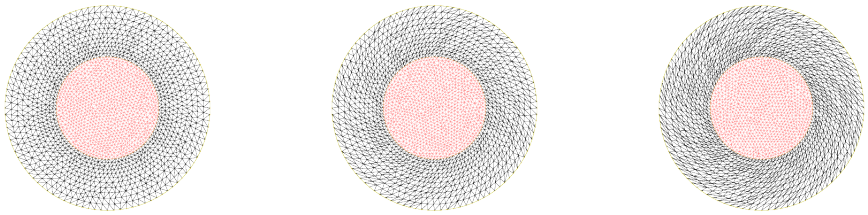


Figure 2.7: $F_\theta(\hat{\Omega})$, for various values of θ

insures global continuity of the mapping. To get higher regularity, we could choose something more involved than linearly going from rotation of angle θ to identity.

Remark 14 *Trying to build mappings with smooth transitions between neighboring domains is one of the topic discussed in chapter 4.*

With our application to fluid dynamics simulation in mind, suppose that we have Karman vortices following a mean flow direction, taken here as the parameter. We model this situation by considering the following solution manifold:

$$\mathcal{M} := \{u_0 \circ r_\theta, \theta \in \mathcal{D}\}$$

where u_0 is some reference solution. A few snapshots taken from this manifold are presented in Figure 2.8. The inflow has a $\frac{\pi}{4}$ amplitude variation, and the structures are modeled by 2 dimensional gaussian functions. We present in Figure 2.9 the snapshots mapped back onto the reference domain, that is $\{F_\theta^{-1}(u_0 \circ r_\theta)\}$. Inspecting the results, we can see that the structures are distorted in the patching domain. The method does not resolve the complexity of the directional problem. It brings it out of the direct vicinity of the blade, to the patching domain. The underlying hope is that the complexity is much smaller away from the objects.

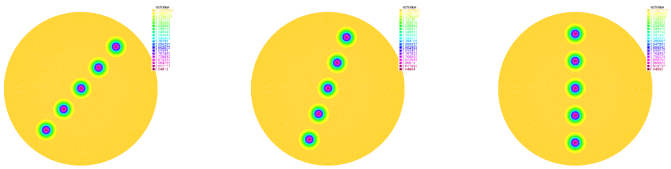


Figure 2.8: Three snapshots taken from $\mathcal{M}$

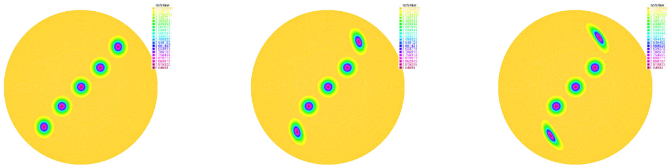


Figure 2.9: Three snapshots mapped back onto the reference domain

2.2.4 One last example

In [59] is studied a electromagnetic field problem where the geometric variability resembles the one we are dealing with. What 'saves' them, is that they transform their problem into a surfacic one. The distance between the obstacles, which is the parameter, then appears explicitly in the modelisation of the interactions.

From this analysis of the literature, we can draw some conclusions:

- RBEM type methods require a structured geometry. We would like to have something more general.
- because of the applications we have in mind, we have to be careful on how we choose to impose the matching conditions, see section 2.2.3
- the number of basis functions in each domain is small. The number of matching constraints that we can impose on each overlap is thus reduced

We will propose two versions of an OverlappingRBEM (ORBEM) method.

The first one is a conforming method that is close the partition of unity method introduced by Babuska et al [100]. This method uses a priori knowledge of the solution inside a localized subdomain. For instance, you can pick as trial space some coarse finite element function on Ω, plus some sharp gradient where you know it will appear. The a priori error estimations show that the global error depends on:

- local errors: the error in each subdomain using the local approximation space
- terms that depend on the smoothness of the partition of unity functions

We will see in subsection 2.4.1 that despite the fact that this approach matches some of the objectives of our target problem, it does not fit well into a reduced order modeling context.

The second one is a non conforming method, close the Arlequin method described in [44] and for which a mathematical analysis can be found in [16]. It was developed in these references in a solid mechanics context. We add to this approach the reduced order modeling aspect, as well

as a more rigorous mathematical analysis. The latter is also close to the analysis of the mortar with overlapping, see for instance [3].

2.3 The set up

We are still solving the parametric PDE defined in equation (2.1), on domains such as the one displayed in Figure 2.1. We will denote by a the associated bilinear form. Suppose that we have identified offline

- a set of 'generic' subdomains denoted by $\hat{\Omega}_j,\ j = 1 \dots J$
- for each $\hat{\Omega}_j$, a reduced basis $\left\{\hat{\phi}_{j,l}, l \in [1 \dots N_j^{red}]\right\}$ of small cardinality.

For all μ, we assume that the physical domain $\Omega(\mu)$ can be decomposed into an overlapping set

$$\Omega(\mu) = \Omega_0 \cup \left(\bigcup_{k=1}^{N_{dom}} \Omega_k \right).$$

such that

- $\forall k > 0,\ \Omega_k$ is the image through some linear transformation of a generic $\hat{\Omega}_j$. One can for instance think of rotation or translation. In the case where only translation is considered, this means that we have a satisfactory reduced basis noted $\{\phi_{k,l}\}$, given as the translated version of the generic reduced basis:

  $$\forall k,\ \exists j \text{ s.t } ,\ \phi_{k,l} = \hat{\phi}_{j,l}(\cdot - x_k),\ l \in [1, \dots N_j^{red}]$$

  For the rest of this chapter, $j(k)$ denotes the application that gives the index of the corresponding generic domain. x_k is the translation parameter.

- there are no complex structures on Ω_0. That is, $u|_{\Omega_0}$ can be represented by a coarse FE space
- $\forall k > 0,\ \text{dist}(\partial\Omega_k, \partial\Omega) > 0$.

These assumptions can be used to define a new notion. Our parameterized initial problem is assumed to have a small 'local Kolmogorov n-width'. It is

- small in the $\Omega_k, k > 0$ even if there are small scales represented. Indeed, we have found a good basis given by a translated version of $\{\hat{\phi}_{j(k),l},\ l \in [1, \dots, N_j^{red}]\}$
- small in Ω_0 because we only need coarse scales.

We show in Figure 2.10 an illustration of possible generic subdomains. The different color here represents the fact there can exist different basis for the same shape. We have two applications in mind. Either a very sparse collection of interesting subdomains, but with no underlying regularity in the positioning. This situation is depicted in Figure 2.11. Or a more dense collection of interesting subdomains, but with some kind of underlying organization. A typical situation is presented in Figure 2.12. Most of the rest of the analysis will be common to these two type of applications. The only difference will be in the numerical section.

Both methods require the construction of a partition of unity. This family of functions defined on Ω will be denoted $\{\chi_i\}_{i \in [0..N_{dom}]}$. It needs to satisfy the following properties:

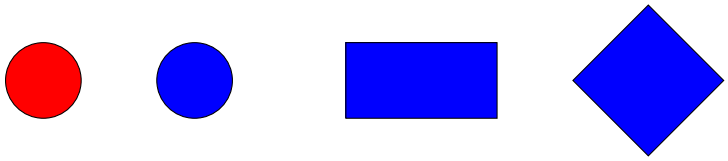


Figure 2.10: Possible generic subdomains $\hat{\Omega}_j$

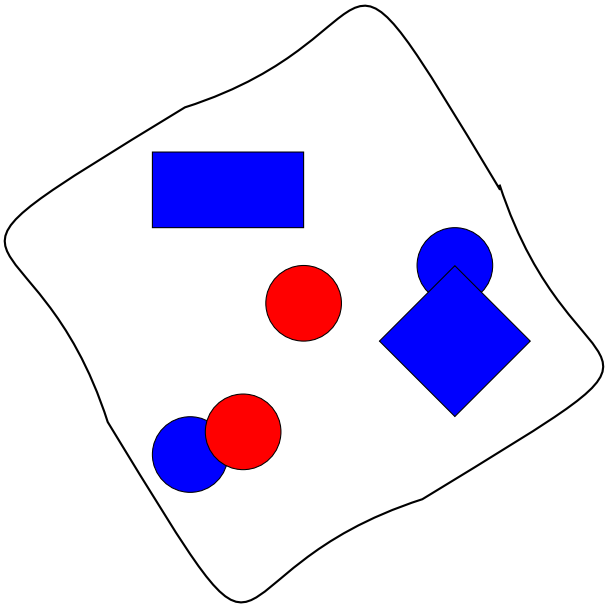


Figure 2.11: First application of ORBEM: a sparse collection of interesting subdomains

- $\forall i,\ \chi_i$ is smooth on Ω
- $\sum_{i=0}^{N_{dom}} \chi_i = 1$ on Ω
- $\forall i,\ \chi_i^{-1}(1) \subset \Omega$ is a closed domain, with non empty interior
- supp $\chi_i \subset \Omega_i$

The characteristics of the set $\{\chi_i\}$ will appear explicitly in the numerical analysis of the method. Until the end of the chapter, we will denote

- $\Omega_{int} = \bigcup_{i=1...N_{dom}}$ supp (χ_i)

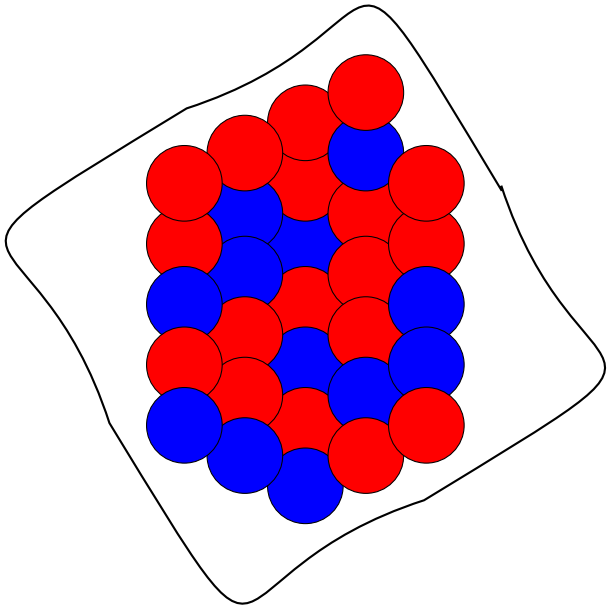


Figure 2.12: Second application of ORBEM: a dense collection of subdomains, following a pattern

- $\Omega_0 =$ supp (χ_0)

2.4 The ORBEM method

In both conforming and non conforming methods, we will follow the same steps:

- make sure that at the continuous level, the method is equivalent to the initial problem
- prove the well posedness of the discretized problem
- compute a priori error estimation of the discretized solutions
- give computational details

2.4.1 Conforming method

There is not much to say about the analysis of the continuous problem. We are simply solving problem (2.1) using a particular subspace of $H^1(\Omega)$.

2.4.1.1 Numerical approximation

To approximate $H_0^1(\Omega_{int})$, we will use $X_{int,\delta}$ defined as:

$$X_{int,\delta} = \text{ span } \{\chi_k \phi_{k,l}\} \subset H_0^1(\Omega_{int}) \tag{2.2}$$

where $k \in [1 \dots N^{dom}]$, $l \in [1, \dots N_k^{red}]$. On Ω_0, we take the following approximation space

$$X_{0,\delta} = \chi_0 \text{ span}\{h_k\}, \tag{2.3}$$

where $\{h_k\}$ is a coarse finite element space on Ω_0. With an abuse of notation, we still denote with $X_{int,\delta}$ and $X_{0,\delta}$ the spaces of functions extended by zero to the whole domain Ω. The discretized space for the whole problem is then given by:

$$X_\delta = X_{int,\delta} + X_{0,\delta}. \tag{2.4}$$

Wrapping things up, we are looking for an approximation of the solution $u \in H^1(\Omega)$ as:

$$\begin{cases} u_\delta &= \sum_k \sum_l \chi_k \alpha_{kl} \phi_{k,l} + \sum_n \chi_0 \alpha_{0n} h_n \\ u_\delta &= u_D \text{ on } \partial\Omega \end{cases} \tag{2.5}$$

2.4.1.2 Well posedness

As we are dealing with a conforming approximation of a coercive elliptic problem, the well posedness of the discretized problem is a direct consequence of the well posedness of the original problem.

2.4.1.3 Best approximation error

This is a conforming method for an elliptic coercive problem. Let α be the coercive constant and C the continuity constant of the bilinear form a. Let u be the truth solution to problem (2.1). Let u_δ be the solution obtained with the Galerkin method in X_δ. Cea's Lemma gives

$$\|u - u_\delta\|_{H^1} \leq \frac{C}{\alpha} \inf_{v_\delta \in X_\delta} \|u - v_\delta\|_{H^1}$$

We want to link this global error to local error estimates, that is we would like to have an inequality such as

$$\inf_{v_\delta \in X_\delta} \|u - v_\delta\|_{H^1} \lesssim \sum_k \inf_{v_\delta \in \text{ span } \{\phi_{k,l}\}} \|u|_{\Omega_k} - v_\delta\|_{H^1(\Omega_k)}.$$

We follow the lines of [100]. For $v_\delta \in X_\delta$, denote $\{v_{\delta,k}\}_k$ the decomposition onto the basis. That is,

$$v_\delta := \sum_k \chi_k v_{\delta,k}.$$

We have

$$\forall v_\delta \in X_\delta,\ \|\nabla(u - v_\delta)\|_{L^2}^2 = \left\| \nabla \left(\sum_k \chi_k (u - v_{\delta,k}) \right) \right\|_{L^2}^2$$

As the χ_k are assumed to be smooth, we have:

$$\forall v_\delta \in X_\delta,\ \|\nabla(u - v_\delta)\|_{L^2}^2 \leq 2 \left\| \sum \chi_k \nabla(u - v_{\delta,k}) \right\|_{L^2}^2 + 2 \left\| \sum \nabla\chi_k (u - v_{\delta,k}) \right\|_{L^2}^2$$

Let $n_{overlap}$ be the maximum number of subdomains that contain any given point in Ω. We have:

$$\forall v_\delta \in X_\delta,\ \|\nabla(u-v_\delta)\|_{L^2}^2 \leq 2\ n_{overlap} \left(\sum_k \|\chi_k \nabla(u-v_{\delta,k})\|_{L^2}^2 + \sum_k \|\nabla\chi_k (u-v_{\delta,k})\|_{L^2}^2 \right) \quad (2.6)$$

We conclude that there exists a constant K, function of C, α, $n_{overlap}$, $\|\chi_k\|_{L^\infty}$, $\|\nabla\chi_k\|_{L^\infty}$,such that:

$$\|u-u_\delta\|_{H^1} \leq K \sum_k \inf_{v_{\delta,k}} \|u-v_{\delta,k}\|_{H^1(\Omega_k)}$$

Forgetting about computational errors, the overall error depends on

- the parameters of the original problem
- the smoothness of the partition of unity
- the local approximation errors

This is exactly the property that we wanted. Let's look at the computational side.

2.4.1.4 Computational cost

In practice, this conforming method reduces to finding α_{kl} and α_{0n} such that

$$\left\{ \begin{array}{rcll} \sum_{\Omega_k} \sum_l \alpha_{kl} a\left(\chi_k\phi_{k,l}, \chi_p\phi_{p,q}\right) + \sum_n \alpha_{0n} a(\chi_0 h_n, \chi_p\phi_{p,q}) & = & 0 & \forall p \in [1\dots N_{dom}], \forall q \in [1\dots N_{red}] \\ \sum_{\Omega_k} \sum_l \alpha_{kl} a\left(\chi_k\phi_{k,l}, \chi_0 h_m\right) + \sum \alpha_{0n} a(\chi_0 h_n, \chi_0 h_m) & = & 0 & \forall h_m \in \text{ coarse Finite Element space} \\ u_\delta & = & u_D & \text{on } \partial\Omega \end{array} \right. \quad (2.7)$$

This formulation requires the computation of coupled terms such as:

$$K_{k,l,p,q} := \int_\Omega \nabla(\chi_k\phi_{k,l}) \nabla(\chi_p\phi_{p,q}).$$

We would like to get rid of the coupled terms, coming from $K_{k,l,p,q}$ when p is not equal to k. We will see that, in that respect, a non conforming approach makes more sense. We will insist on this property in the numerical section.

2.4.2 Non conforming method

Unlike the conforming method, we will work a little bit at the continuous level. For simplicity, we will assume throughout this section that u_0 is known, having in mind a fluid dynamics problem, where inflow and outflow would be taken as uniform flow. The case with u_0 in a coarse finite element space would be treated the same way.

We work in the following Hilbert space

$$X = \left\{ (u_k)_{k=1..N_{dom}} \in \prod_{k=1}^{N_{dom}} H^1(\Omega_k) \right\}. \quad (2.8)$$

X is induced with the following broken norm:

$$\|u\|_*^2 := \sum_k \|u\|_{H^1(\Omega_k)}^2. \quad (2.9)$$

The appropriate bilinear form is given by:

$$a : \left\{ \begin{array}{lcl} X \times X & \rightarrow & \mathbb{R} \\ (u,v) & \mapsto & \sum_k \int \chi_k \nabla u_k \nabla v_k + \sum_k \int \chi_k \beta \cdot \nabla u_k \ v_k. \end{array} \right. \tag{2.10}$$

The non conforming formulation is to look for u in X such that

$$\left\{ \begin{array}{rcll} -\Delta u + \beta \cdot \nabla u & = & 0 & \text{in each } \Omega_k \text{ in the distributional sense} \\ < (u_k - u_l), \Psi >_{L^2(\Omega_k \cap \Omega_p)} & = & 0 & \forall \Psi \in L^2(\Omega_k \cap \Omega_p), \\ < u_k, \Psi >_{L^2(\Omega_k \cap \Omega_0)} & = & < u_0, \Psi >_{L^2(\Omega_k \cap \Omega_0)} & \forall \Psi \in L^2(\Omega_k \cap \Omega_0) \end{array} \right. \tag{2.11}$$

For k, p, denote $M_{k,p} := L^2(\Omega_k \cap \Omega_p)$ and $M_{k,0} := L^2(\Omega_k \cap \Omega_0)$. We define the following bilinear forms

- $b_{k,p}(u, \Psi)$ the set of 'internal' matching conditions, i.e for $u \in X$ and $\Psi \in M_{k,p}$. These correspond to the second equation in (2.11).
- $b_{k,0}(u, \Psi)$ the set of matching conditions with Ω_0, i.e for $u \in X$ and $\Psi \in M_{k,0}$. These correspond to the third equation in (2.11).

For all $\{g_k \in L^2(\Omega_k \cap \Omega_0)\}$, define $V(\{g_k\})$:

$$V(\{g_k\}) = \left\{ u \in X, \ \text{ s.t } \begin{array}{rcl} \forall k, p, \ \forall \psi \in M_{k,p}, \ b_{k,p}(u, \Psi) & = & 0 \\ \forall k, \ \forall \psi \in M_{k,0}, \ b_{k,0}(u, \Psi) & = & < g_k, \Psi >_{L^2(\Omega_k \cap \Omega_0)} \end{array} \right\} \tag{2.12}$$

By considering the isomorphism between $V(0)$ and $H_0^1(\Omega \setminus \Omega_0)$, it is easy to see that our initial problem is equivalent to the following: find $u \in V(u_0)$ such that

$$\forall v \in V(0), \ a(u, v) = 0. \tag{2.13}$$

2.4.2.1 Discrete formulation/ well posedness

We need to pick a discretization space for X. It is natural to pick

$$\forall k, \ W_k := \text{ span } \left\{ \phi_{k,l}, \ l = 1 \dots N_k^{red} \right\}. \tag{2.14}$$

The global discretization space is then given by:

$$X_\delta = \left\{ (u_k)_{k=1..N_{dom}} \in \prod_{k=1}^{N_{dom}} W_k \right\} \tag{2.15}$$

It is more tricky to choose a suitable discretization space for $M_{k,p}$ and $M_{k,0}$. We do not specify them yet, and denote with $M_{\delta,k,p}$ and $M_{\delta,k,0}$ some generic discretization space of $M_{k,p}$ $M_{k,0}$, respectively. As in the continuous case, define, for $\{g_k \in L^2(\Omega_k \cap \Omega_0)\}$:

$$V_\delta(\{g_k\}) = \left\{ u \in X_\delta, \ \text{ s.t } \begin{array}{rcl} \forall \Psi_{k,p} \in M_{\delta,k,p}, \ b_{k,p}(u, \Psi_{k,p}) & = & 0 \\ \forall \Psi_{k,0} \in M_{\delta,k,0}, \ b_{k,0}(u, \Psi_{k,0}) & = & < g_k, \Psi_{k,0} >_{L^2(\Omega_k \cap \Omega_0)} \end{array} \right\}. \tag{2.16}$$

With this notation, the discretized problem becomes: find $u_\delta \in V_\delta(u_0)$ such that:

$$\forall v_\delta \in V_\delta(0), \ a(u_\delta, v_\delta) = 0$$

In order to prove the well posedness of the resulting discrete problem, we need to prove the $V_\delta(0)$ ellipticity of a.

Remark 15 *Unlike in the conforming case, $V_\delta(0)$ is not a subspace of $V(0)$. The ellipticity is not trivial.*

Let v in $V_\delta(0)$. We have:

$$a(v,v) = \sum_{k=1}^{N_{dom}} \int_{\Omega_k} \chi_k |\nabla v_k|^2 + \sum_{k=1}^{N_{dom}} \int_{\Omega_k} \chi_k \beta \cdot \nabla v_k \; v_k$$

A rigorous proof will not be presented here, and we will stick to formal arguments. Let some domain Ω_k. Away from the boundary $\partial\Omega_k$, χ_k is bounded from below. The problem occurs near the boundary as $\chi_k|\nabla v_k|^2$ from the bilinear form can not compete with $|\nabla v_k|^2$ of the broken norm. The hope comes from the fact that where χ_k is small, there exist $p \in [0, N_{dom}]$ such that χ_p is big, and because of the matching conditions (v is in $V_\delta(0)$), v_p is close to v_k. The ellipticity constant should thus be close to the ellipticity constant of the continuous non conforming case, if the matching is good enough. More quantitative arguments can be found in the next subsubsection.

2.4.2.2 A priori error estimations

We start with an application of Strang's 2nd lemma [125], which is the standard tool for a priori estimation for non conforming methods. With an abuse of notation, we also denote by u the representant of the truth solution in X:

$$u := \{u_k, \; k \in [1 \dots N_{dom}]\}.$$

Let $u_\delta := \{u_{\delta,k}\}_k$ be the solution to the discretized problem. Strang's 2nd lemma states:

$$\|u - u_\delta\|_* \leq \frac{C}{\alpha} \left(\inf_{v_\delta \in V_\delta(u_0)} \|u - v_\delta\|_* + \sup_{v_\delta \in V_\delta(0)} \frac{a(u, v_\delta)}{\|v_\delta\|_*} \right). \tag{2.17}$$

The best approximation error will depend on the offline phase and on the choice of the reduced basis. Our hypothesis in this whole chapter is that this error is small, see the definition of local Kolmogorov n-width section 2.3. We will thus focus on the consistency error. This error measures 'how far' we are from having a global H^1 solution.

Define the following linear form on X_δ:

$$K : \left\{ \begin{array}{ccc} X_\delta & \to & \mathbb{R} \\ v_\delta & \mapsto & \sum_{k=1}^{N_{dom}} \int_{\Omega_k} \chi_k \nabla u_k \nabla v_{\delta,k} + \sum_{k=1}^{N_{dom}} \int_{\Omega_k} \chi_k \beta \cdot \nabla u_k \; v_{\delta,k} \end{array} \right.$$

As u is in $H^1(\Omega)$ and thus consistent over overlapping subdomains, we have:

$$\forall v_\delta \in X_\delta, \; K(v_\delta) = \int_\Omega \nabla u \cdot \left(\sum_{k=1}^{N_{dom}} \chi_k \nabla v_{\delta,k} \right) + \int_\Omega \left(\sum_{k=1}^{N_{dom}} \chi_k v_{\delta,k} \right) \beta \cdot \nabla u$$

We use $\sum_{k=1}^{N_{dom}} \chi_k v_{\delta,k} \in H_0^1(\Omega)$ and that u is the solution to the initial problem, to show that:

$$\int_\Omega \nabla u \nabla \left(\sum_{k=1}^{N_{dom}} \chi_k v_{\delta,k} \right) + \int_\Omega \left(\sum_{k=1}^{N_{dom}} \chi_k v_{\delta,k} \right) \beta \cdot \nabla u = 0.$$

Combining the two previous relations, we have:

$$\forall v_\delta \in X_\delta,\ K(v_\delta) = \sum_{k=1}^{N_{dom}} \int_\Omega \nabla u \cdot \nabla \chi_k v_{\delta,k} \tag{2.18}$$

On domains that do not overlapp, the previous quantity is zero. Let's handle a situation with two subdmains overlapping. Let Ω_k, Ω_p these subdomains. Define K_{kp} the restriction of K on this intersection, that is:

$$K_{kp} : v_\delta \to \int_{\Omega_k \cap \Omega_p} \nabla u \nabla \chi_p v_{\delta,p} + \int_{\Omega_k \cap \Omega_p} \nabla u \nabla \chi_k v_{\delta,k} = \int_{\Omega_k \cap \Omega_p} \nabla u \nabla \chi_p \left(v_{\delta,p} - v_{\delta,k}\right)$$

In order to get an idea of the consistency error, we need to bound $K_{k,p}$ on the subspace $V_\delta(0)$.

We first follow the lines of the RBEM method. As we will see, this path does not match all of the objectives set up for the method, but will help get a better understanding of the differences with RBEM. A natural choice for $M_{\delta,k,p}$ is to pick some reduced basis that represents well the space spanned by:

$$\left\{\nabla \chi_k \nabla \phi_{k,l},\ l = 1, \dots, N_k^{red}\right\} \bigcup \left\{\phi_{k,l},\ l = 1, \dots, N_k^{red}\right\}. \tag{2.19}$$

As $v_\delta \in V_\delta(0)$, we have:

$$v_{\delta,k} - v_{\delta,p} \in M_{\delta,k,p}^\perp.$$

We deduce the following estimate for K_{kp}:

$$\forall v_\delta \in V_\delta(0),\ K_{kp}(v_\delta) \leq \left\|\Pi^\perp_{M_{\delta,k,p}}(\nabla \chi_p \cdot \nabla u)\right\|_{L^2(\Omega_k \cap \Omega_p)} \left\|\Pi^\perp_{M_{\delta,k,p}}\left(v_{\delta,p} - v_{\delta,k}\right)\right\|_{L^2(\Omega_k \cap \Omega_p)}$$

As for each overlap, the set given in (2.19) is of small n-width, we know that we can find a reduced space $M_{\delta,k,p}$ of small dimension that guarantees a satisfactory upper bound on $K_{k,p}$, and thus on the consistency error. Nevertheless, this strategy does not entirely fulfill our objectives. Indeed, it requires the knowledge of the sets given in (2.19) and thus the knowledge of the overlaps a priori. We will see in the next section an alternative, weaker but more flexible matching.

2.5 Implementation details

The offline section does not cause specific issues. The first step is to identify the generic subdomains. These can be given for instance by the geometry of the mesh (hole with a specific shape, object of complex shape etc.), or by localized singularities of some physical parameter. We then restrict the solutions to these generic subdomains, and perform standard reduced order modeling. This involves some mesh interpolation, but nothing to dangerous.

For the online section and unlike for the conforming method, we only need to compute the bilinear form a in each subdomain independently. Indeed, the quantities appearing during the online phase are given by:

$$\forall k,\ \forall (l,m),\ \int_{\Omega_k} \chi_k \nabla \phi_{k,p} \nabla \phi_{k,m} = \int_{\hat{\Omega}_{j(k)}} \chi_k(\cdot + x_k) \nabla \hat{\phi}_{j(k),l} \nabla \hat{\phi}_{j(k),m}. \tag{2.20}$$

Two big issues remain:

- the partition of unity $\{\chi_k(\mu), k \in 1, \dots, N_{dom}\}$ carries all the geometric variability. Its construction is a challenging task. We will propose two ways of mitigating the associated online computational cost.
- the matching constraints are very involved numerically. Both the stability and accuracy of the method depend on the properties of the discretization spaces $M_{\delta,k,p}$ and $M_{\delta,k,0}$.

2.5.1 Partition of unity

We can follow two routes in order to solve the complexity implied by the geometric variability. We can either limit the geometric variability, or accept a high set up cost, at the beginning of the simulation.

2.5.1.1 High set up cost

Suppose that we are solving a time dependent PDE. There will be a set up process before the actual resolution of the problem. During this set up, we compute the set of $\{\chi_i,\ i = 0, \dots, N_{dom}]\}$ and the associated quantities, see (2.20). This will be computationally expensive, but will only be done once.

2.5.1.2 Structured problem

If we are dealing with a problem such as depicted in Figure 2.12, the pattern in the geometry will induce a pattern in the partition of unity functions $\{\chi_i, i = 1, \dots, N_{dom}\}$. That is, there will be a limited number of generic partition of unity functions $\{\hat{\chi}_k\}$, such that:

$$\forall i \in [1, \dots, N],\ \exists k(i),\quad \text{s.t } \chi_i = \hat{\chi}_{k(i)}(\cdot - x_i).$$

This means a highly reduced online computational cost.

2.5.1.3 Start from a non overlapping decomposition

The idea of this section was taken from [104]. This method starts with a non-overlapping decomposition of Ω:

$$\{Q_k,\ k = 0 \dots N_{dom}\}.$$

Remark 16 *This premise restricts the number of situations that can be handled.*

Different shapes can coexist in the family $\{Q_k\}_k$. The only requirement is that it covers Ω.

Let Θ a 2 dimensional smooth 'window function', with a support included in a small neighborhood of the origin, and such that $\int_\Omega \Theta = 1$. We define our partition of unity functions $\{\chi_k,\ k = 0 \dots N_{dom}\}$, as:

$$\forall k \in [1, \dots, N_{dom}],\ \chi_k = \Theta * 1_{Q_k} \tag{2.21}$$

where the operator $*$ denotes the standard convolution operator. Such a family of function satisfy:

- $\forall x \in \Omega,\ \sum_{k=1}^{N_{dom}} \chi_k(x) = 1$
- $\text{supp}(\chi_k) = Q_k + \text{supp}(\Theta)$

- the resulting partition of unity functions are translated version of a generic function: $\chi_k = \hat{\chi}_{j(k)}(\cdot - x_k)$

The first point is easily checked:

$$\forall x \in \Omega,\ \sum_k \chi_k(x) = \sum_k \int_{\mathbb{R}^2} \Theta(x-y) 1_{Q_k}(y) dy = \int_\Omega \Theta(x-y)dy = 1 \tag{2.22}$$

as long as $\forall k > 0,\ dist(\partial\Omega_k, \partial\Omega) >$ characteristic size of the support of Θ.

With this choice, the partition of unity functions are independent of the non overlapping cover of Ω. It thus involves no additional online computational cost.

2.5.2 Matching

We will give a possible computationally reasonable choice for $M_{\delta,k,p}$, subspace of $M_{k,p} = L^2(\Omega_k \cap \Omega_p)$. As usual, we focus on one particular overlap, say $\Omega_k \cap \Omega_p$.

2.5.2.1 One possible choice for constraints

As in the previous section, we use Θ some two dimensional 'window' function, of small support. We propose the following matching constraints:

$$\forall m \in [1 \dots M],\ \int_{\Omega_k \cap \Omega_p} \eta(v_k)(\cdot)\Theta(\cdot - x_m) = \int_{\Omega_k \cap \Omega_p} \eta(v_p)(\cdot)\Theta(\cdot - x_m)$$

for some well chosen $\{x_m,\ m \in [1, M]\}$ in $\Omega_k \cap \Omega_p$, and for η some linear operator $H^1(\Omega_k \cap \Omega_p) \to L^2(\Omega_k \cap \Omega_p)$. For instance, when η is the identity, this imposes that the average in some predefined neighborhood of x_m of v_k and v_p match.

First question to answer is to see if it fits into the framework of section 2.4.2.1. Define $\Gamma_{k,p}$ as:

$$\Gamma_{k,p}(x_m) : \left\{ \begin{array}{rcl} H^1(\Omega_k \cap \Omega_p) & \to & \mathbb{R} \\ v & \mapsto & (v * \Theta)(x_m) = \int_{\Omega_k \cap \Omega_p} \eta(v)(\cdot)\Theta(\cdot - x_m) \end{array} \right.$$

and take $M_{\delta,k,p} :=$ span $\{\Gamma_{k,p}(x_m),\ m \in [1, \dots, M]\}$. The associated bilinear form $b_{k,p}$ is given by:

$$\forall v \in X_\delta,\ b_{k,p}(v, \Gamma_{k,p}(x_m)) = \int_{\Omega_k \cap \Omega_p} \Theta(\cdot - x_m)(\eta(v_k) - \eta(v_p)).$$

With this particular choice of $M_{\delta,k,p}$, what can we say about the consistency error? Recall that for two subdomains overlaping, it is given by:

$$\sup_{v_\delta \in V_\delta(0)} \left(\frac{1}{\|v_\delta\|_*} \int_{\Omega_k \cap \Omega_p} \nabla u \nabla \chi_p \left(v_{\delta,p} - v_{\delta,k} \right) \right).$$

As one of our premises is that for all k, W_k is of small n-width, we know that:

$$\left\{ w_{k,p} := (v_{\delta,k} - v_{\delta,p})|_{\Omega_k \cap \Omega_p},\ v_{\delta,k} \in W_k,\ v_{\delta,p} \in W_p \right\}$$

also has a small n-width, whatever the particular overlap $\Omega_k \cap \Omega_p$ considered. Thus, if we manage to find enough independent matching conditions, this should be sufficient to guarantee the smallness of $\|v_{\delta,p} - v_{\delta,k}\|_{L^2(\Omega_k \cap \Omega_p)}$, and thus of the global consistency error. One can for instance think that by equating enough local averages of the solution and local averages of the

vorticity, one can correctly match the solutions of a CFD problem, even if this is not done in an optimal fashion as in the RBEM version presented above. We note the difference with equation (2.19). Where we were explicitly constructing a taylored reduced basis for a specific overlap, we now hope to find local (averaged) quantities that characterize the solution on any overlap. The precise quantities to consider and the size of the support of the window function Θ are of course problem dependent, and should be chosen in the offline section.

2.5.2.2 Offline, online decomposition

We show how to efficiently impose the matching condition proposed in the previous section. For this, we need a procedure to select a set of $\{x_m\}$ in the overlap as well as a way of efficiently computing the corresponding $\Gamma_{k,p}(x_m)(v)$ defined in the previous section. For simplicity, we will use η the identity mapping, but note that the same procedure could be performed for any linear operator acting on $H^1(\Omega_k \cap \Omega_p)$.

Define $\Lambda_{k,l}$ as:

$$\Lambda_{k,l} : \left\{ \begin{array}{lcl} \Omega_k \cap \Omega_p & \rightarrow & \mathbb{R} \\ x & \mapsto & \int_{\Omega_k \cap \Omega_p} \phi_{k,l}(\cdot)\Theta(\cdot - x) \end{array} \right.$$

Let $v \in X_\delta$ and $\{\alpha_{k,l},\ l = 1 \dots N_k^{red}\}$ be the coordinates of v on W_k. We know that

$$\forall x,\ \Gamma_{k,p}(x)(v) = \sum_{l=1}^{N_k^{red}} \alpha_{k,l} \Lambda_{k,l}(x) \tag{2.23}$$

We have replaced the problem of estimating $\Gamma_{k,p}(x_m)(v)$ by the estimation of $\Lambda_{k,l}(x_m)$, using linearity.

Let $j(k)$ in J and $\hat{\Omega}_{j(k)}$ be the generic obstacle corresponding to Ω_k. Select offline some points in $\hat{\Omega}_{j(k)}$ denoted $\{\hat{x}_{j(k),n}\}$. We pre compute the local quantities:

$$\forall j \in J,\ \hat{\Lambda}_{j,l,n} := \int_{\hat{\Omega}_j} \hat{\phi}_{j,l} \Theta(\cdot - \hat{x}_{j,n}).$$

We use the translation invariance to show that:

$$\forall \hat{x} \in \hat{\Omega}_{j(k)},\ \Lambda_{k,l}(x_k + \hat{x}) = \int_{\Omega_k \cap \Omega_p} \phi_{k,l} \Theta\left(\cdot - (x_k + \hat{x})\right) = \int_{\hat{\Omega}_{j(k)}} \hat{\phi}_{j(k),l} \Theta(\cdot - \hat{x})$$

and thus that:

$$\Lambda_{k,l}(x_k + \hat{x}_{j(k),n}) = \hat{\Lambda}_{j(k),l,n}.$$

Finally, $\Lambda_{k,l}$ is a function defined on Ω_k, smooth, whose value is known at a discrete set of points, namely $\{x_k + \hat{x}_{j(k),n},\ n\}$.

This ends the matching procedure. Indeed, let $\{x_m\} \in \Omega_k$ be a set of points chosen online. We use some interpolation method to approximate the $\Lambda_{k,l}(x_m)$. Then, with equation (2.23) we have an estimation of $b_{k,p}(v, \Gamma_{k,p})$. The accuracy of the interpolation procedure needs to be assessed in the offline section.

2.6 Conclusion

We have proposed in this chapter an overlapping version of the RBEM method. This method has been constructed to handle situations with a high geometric variability. The main difference with the RBEM method is the way the matching constraints are imposed.

What has been presented is a preliminary work and a lot of questions still need to be answered. We have sketched a procedure to efficiently build partition of unity functions. This should be numerically investigated. Also, I have the feeling that the matching procedure could be improved. We have to keep its adaptability, but try to find an alternative resulting in a stronger matching.

We have one bonus property with the non conforming version of ORBEM. It matches the objective of the method presented in section 2.2.3. Each local basis can be modified independently online through non linear transformations. For instance, suppose that we are dealing with problems where the inflow direction varies in time. We would like the 'local' basis to rotate accordingly. This is illustrated in Figure 2.13. This can be handled simply by changing the

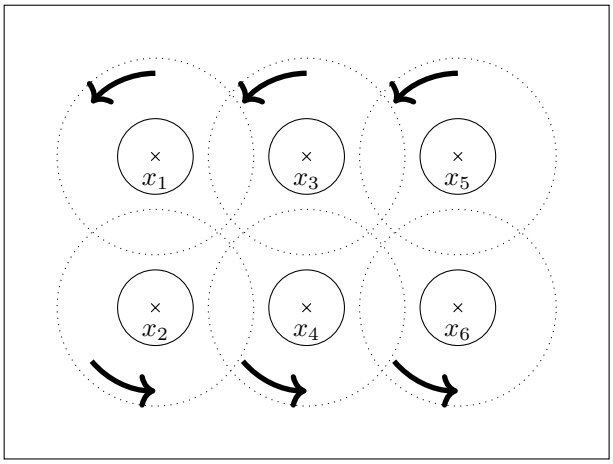


Figure 2.13: The ORBEM method allows for the rotation of the local basis

mapping between the generic domains $\hat{\Omega}_j$ and the instances of these subdomains Ω_k, as they will now involve a rotation. We do not need more evaluation of the bilinear form a, as it only involves local terms and is rotation invariant. Also, the matching process can be used as such. Summing up, for a prescribed rotation parameter on each local basis, the algorithm is exactly the same.

How to pick, online, a 'good' rotation parameter, such that the flow is best represented by the local basis is the topic of the next chapter 3.

Chapter 3

On the construction of a calibration procedure

The reduced basis method allows to propose accurate approximations for many parameter dependent partial differential equations, almost in real time, at least if the Kolmogorov n-width of the set of all solutions, under variation of the parameters, is rapidly decaying. The idea is that any solutions may be well approximated by the linear combination of some well chosen solutions that are computed offline once and for all (by another, more expensive, discretization) for some well chosen parameter values. In some cases however, such as problems with large convection effects, the linear representation is not sufficient and, as a consequence, the set of solutions needs to be transformed/twisted so that the combination of the proper twist and the appropriate linear combination recovers an accurate approximation. This chapter presents a simple approach towards this direction, preliminary simulations support this approach. There is an article version of this chapter available, see [27]. Note that we used Freefem++ [67] for some of the numerical simulations used in this chapter.

3.1 Introduction

Fast reliable solutions to many queries parametric Partial Differential Equations (PDE) have many applications among which real time systems, optimization problems and optimal control. Many different methods for reducing the complexity of the computations when such many queries are required have blossomed for answering this specific need. One of the approaches that have emerged is reduced order modeling (ROM). Methods in this category have been developed and are now well understood and set on firm grounds, both for steady cases or time dependent problems where time can be considered as another parameter.

The reduced basis method, which is the method that we focus on in this chapter, enters in this frame and consists in, i) defining a sequence of low dimensional spaces for the approximation of the whole set of the solutions to the parametric PDE when the parameters vary (called hereafter the solution manifold associated to our problem); ii) once such a sequence of low dimensional spaces (known as reduced basis spaces) is determined, an approximate solution is sought in such a chosen reduced space to the PDE for the values of the parameter we are interested in. The

approximation is often based on a Galerkin formulation. For such reduced basis methods, both the variety of applications and the theory are now quite sound. For instance, reliable algorithms with a priori estimates and certified a posteriori errors have been developed for elliptic and parabolic problems, with or without so-called affine parameter dependence, see e.g. the two recent books on the subject [68] and [109] and, of course, the publications therein.

Reduced basis methods, classically, consider the solution manifold associated to the parametrized problem as outlined above and are appropriate if this manifold can be approximated accurately by a sequence of finite dimensional spaces. The mathematical frame for this is inherently linked to the notion of *Kolmogorov width* of solution manifolds, i.e. on how well the solution manifold can be approximated by a finite dimensional linear space. More precisely, let $\mathcal{M}$ be a manifold embedded in some normed linear space X. The Kolmogorov n-width of $\mathcal{M}$ is defined as:

$$d_n(\mathcal{M}, X) = \inf_{E_n} \sup_{f \in \mathcal{M}} \inf_{g \in E_n} \|f - g\|_X \tag{3.1}$$

The first infimum being taken over all linear subspaces E_n of dimension n embedded in X.

Even if, from the practical point of view, there are various ways for checking that $\mathcal{M}$ can be approximated by a series of reduced spaced with small dimension, the first natural mathematical question is to provide an estimation of the Kolmogorov n-width of $\mathcal{M}$. Second, the question of an applied mathematician is if one can actually build an optimal, or close to optimal sequence of basis sets for these spaces?

Of course, in the vast majority of real cases, there is no analytical expression for this dimension but there are some papers giving bounds for some restricted classes of problems in the literature. For instance, in [99] bounds on d_n are found for solution manifolds corresponding to regular elliptic problems and where the parameter dependence is on the forcing term. More general cases can be handled using the results in [35]. The hypothesis therein is on the regularity of the solution with respect to the parameter dependence, it is proven that, under analyticity assumption on the behavior of the parameters in the PDE, the small Kolmogorov n-width of the manifold of parameters $\mathcal{D}$ ($\leq cn^{-t}$, $t > 1$) implies the smallness of the Kolmogorov n-width of the associated solutions manifold $\mathcal{M}_{\mathcal{D}}$ ($\leq cn^{-s}$, $s < t - 1$).

In practice, instead of the "optimal" linear subspace of dimension n in the sense described earlier, we build a "good" linear subspace. In the literature, the two most classical algorithms are the greedy method based on a certified (or at least fair enough) a posteriori estimator, and the Proper Orthogonal Decomposition (POD). We proceed assuming that the chosen algorithm has given a "good" basis "close" to the optimal one, that is, we assume that our reduced family of spaces $\{X_n\}_n$ satisfies:

$$d_n(\mathcal{M}, X) \approx \sup_{f \in \mathcal{M}} \inf_{g \in X_n} \|f - g\|_X \tag{3.2}$$

A first paper on this subject is [94], where the authors derived error bounds on the error for the Reduced Basis Method (RBM) approximation in case of a single parameter dependent elliptic PDE. More general results have been obtained more recently for the greedy approach of the RBM [21, 43]. The optimality considered in the case of POD is slightly different. The POD focuses on minimizing the average error (parameter wise), in some norm. More precisely, we have the well known relation

$$\int_{\mathcal{D}} \|u(\mu) - \Pi_{POD} u(\mu)\|^2 \, d\mu = \sum_{i > N_{POD}} \lambda_i \tag{3.3}$$

where Π_{POD} is the orthogonal projection onto the POD reduced space of dimension N_{POD} and the λ_i are the eigenvalues of the associated correlation operator, in decreasing order. The faster

the decay of the eigenvalues, the fewer modes are needed for a good (in average) reconstruction of the solution manifold.

Up to now, most of the literature on the subject, deals with problems where one can expect/check/prove/ or hope, that the solution manifold $\mathcal{M}_{\mathcal{D}}$ has a small Kolmogorov n-width. There are however cases where the plain approach does not work and some transformation of $\mathcal{M}_{\mathcal{D}}$ needs to be done. An example is for instance the use of the Piola transform in the processing of the velocity field when the PDE is the Stokes or Navier Stokes problem and the parameter includes the geometry of the computational problem (see e.g. [90]). The choice of the Piola transform indeed provides better reduction than a simple change of variables.

The most classical and simple example illustrating limitations of reduced models due to large Kolmogorov n-width is the pure transport equation, with constant speed $c > 0$. Formally, we consider the following parametric PDE over the domain $\Omega = (a, b) \subset I\!R$

$$\left\{ \begin{array}{lcll} u_t(x,t) + cu_x(x,t) & = & 0, & \text{in } \Omega\times]0,T[\\ u(x,0) & = & u_0(x), & \text{in } \Omega \\ c \in \mathcal{D} := [c_{min}, c_{max}]. & & & \end{array} \right. \tag{3.4}$$

The analytic solution is given by

$$u(x,t;c) = u_0(x - ct). \tag{3.5}$$

We can consider two solution manifolds. Either the space time solution manifold

$$\mathcal{M}_{\mathcal{D}}^{x,t} = \{u(\cdot,\cdot;c),\ c \in \mathcal{D}\}, \tag{3.6}$$

or a more natural solution manifold in our context is the snapshot solution manifold

$$\mathcal{M}_{\mathcal{D}}^{x} = \{u(\cdot,t;c),\ t \in [0,T],\ c \in \mathcal{D}\}. \tag{3.7}$$

We will first give an illustrative idea of $d_n(\mathcal{M}_c^x)$, i.e for a fixed convection parameter. Thus, the only "parameter" left is time and $d_n(\mathcal{M}_c^x)$ is, of course, smaller than $d_n(\mathcal{M}_{\mathcal{D}}^x)$.

Suppose now that our initial solution is compactly supported and let ℓ denote the Lebesgue measure of its support. Let us assume in addition that its support is included in $]a, a + \ell[$. Then, there are at least $(b-a)/\ell$ snapshots $\{u(\cdot,t^k;c)\}_k$ obtained for $t_k = k\ell/c$ that are two by two orthogonal proving that a lower bound of the Kolmogorov n-width is $(b-a)/\ell$. For a given accuracy, reducing ℓ, we can make the size of the reduced basis needed arbitrarily large. Another example of badly behaved manifold space can also be found in [126], or in chapter 1 of this manuscript.

The objective here is to give a proper framework and to introduce notations generalizing the following observation: apart from translation, the solution manifold for the whole time simulation can be represented by a unique basis. However, let us stress that this translation is not a linear process hence the Kolmogorov process cannot capture it. An additional ingredient to existing reduced order methods has thus to be added so as to capture this very simple problem structure.

Most of the literature in the reduced order modeling community on convection dominated problem focus on the stabilization issue, and not on the reduction of the Kolmogorov n-wdith. For instance, the authors in [39] have proven that using, as usual, the residual of the PDE as a surrogate for the true error, is not adapted if convection is dominating as the relative a posteriori estimator is not fair enough. Their method involves other norms than the natural ones, and increases the stability at each iteration by enriching the trial space. Once again, their method improves the stability of the construction of a reduced basis, but does not handle the fact that the solution manifold can have a large Kolmogorov n-width.

In the same direction let us quote the papers related to the so called GNAT approach [30, 31] where the authors propose also an alternative reduction approach for these type of problems.

In [2], the authors address the stability issue in another direction. They give ideas and show numerical examples illustrating the fact that using L^1-minimisation, instead of the — more classical — L^2-minimisation (corresponding to a Galerkin scheme, which is natural in the reduced modeling context), does a better job for handling shocks (as appears in non linear convection problems) and provides more stable results. However this approach does not cure the problem that we have indicated above related to the large dimension of the solution manifold.

Let us also mention at this level, as an intermediate approach, the paper [28]. As standard reduced order modeling fails, the author chooses, in a preprocessing step, to "chop off" the reduced basis functions resulting in a kind of adaptive coarse enriched finite element method.

Very few papers tackle the n-width issue directly. In [126], the authors propose a method that is the first attempt to use shock fitting related ideas in the context of reduced order modeling. The idea is to decompose the spatial domain into zones separated by shocks. In each zone, classical reduced order modeling is performed, and the shocks dynamic is handled using another equation. For them, it is given by Rankine-Hugoniot conditions. This method, just as any other shock fitting method, is somehow limited to one dimensional problems.

In [60], the authors develop a method where the POD basis is reconstructed at each time step to follow the propagation of the phenomenon. More precisely, by referring to Lax–pairs, they choose as reduced basis the modes of the Schrödinger operator where the potential is taken as the solution at the previous time step. Even if no theoretical proof of this ansatz is presented, the numerical results presented in that paper illustrate the interest of the approach for selecting the reduced space and adding stability to the process without curing however the large increase of the dimension of the reduced space when the accuracy requirement increases.

The method presented in [74] is similar to our work in many aspects, in particular in looking for a change of variable for better representing the solution manifold. Their approach relies on the existence of a main mode u_0 that, by convection, represents most of the solution. The proper change of variable (written as a sum of advection modes) is fitted by evaluating Wasserstein distances between the snapshots in $\mathcal{M}^x_{\mathcal{D}}$, with modes being obtained by solving Monge-Kantorovich optimal transport problems w.r.t. the reference mode u_0. Various numerical results illustrate the approach, however only in cases where the solution exhibits indeed such a main mode u_0 which is doubtful in nonlinear processes. We will come back on their ideas in the following sections.

In the first section, we introduce the notion of "calibration" of a solution manifold based on our knowledge of the process (differing here somehow from the optimal transport problem approach in [74]). We detail the steps of a procedure using the calibration manifold, the so-called freezing method [19, 105]. As will quickly appear, the mathematical ingredients needed for a rigorous analysis of the freezing method are very involved. In the end of the section, we motivate the need for a lighter, more natural framework. The rest of this chapter is devoted to the development of this alternative method. We extensively describe its specificities on the one dimensional unsteady viscous Burger equation. We propose a self sufficient reduced scheme, with efficient offline/online decomposition. We then present some numerical simulations confirming the overall feasibility of the method. The last sections of this chapter describe possible extensions to the method.

3.2 Formal presentation

Let us consider a general time dependent parametric PDE in some physical space $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$

$$\left\{ \begin{array}{lcll} u_t + \mathcal{L}_\mu(u) & = & 0 & \text{in } [0,T] \times \Omega \\ u(\cdot, t=0; \mu) & = & u_0(\cdot, \mu) & \text{in } \Omega \\ B(u; \mu) & = & 0 & \text{on } \partial\Omega \end{array} \right. \tag{3.8}$$

where μ varies in some compact parameter space $\mathcal{D}$. Our approach considers the corresponding snapshot solution manifold $\mathcal{M}_{\mathcal{D}}^x$ as defined in (3.7), that we assume embedded in some Hilbert space X.

Let us assume that the solution manifold has a simple structure, not reflected though by the Kolmogorov n-width but hidden by a transformation of the solution manifold. As stated in the introduction, we can think of the transport equation as being the simplest example for which this is occurring. The objective is, through a "calibration" step, to recover the simple structure of the solution manifold.

3.2.1 Specifications

This first section will be formal. We will try to dress up a list of specifications that we would like our calibration process to satisfy. We are looking for a family of applications

$$\mathcal{F} := \{\gamma : X \to X\},$$

and for

$$\gamma : \left\{ \begin{array}{lcl} ([0,T], \mathcal{D}) & \to & \mathcal{F} \\ (t, \mu) & \mapsto & \gamma(t; \mu) \end{array} \right.$$

such that

1. the transformed solution manifold,

$$\tilde{\mathcal{M}} = \left\{ \gamma(t;\mu)^{-1} \left(u(\cdot, t; \mu) \right),\ \mu \in \mathcal{D},\ t \in [0,T] \right\}$$

   has a faster decaying n-width

2. for all μ, we can find a well posed equation satisfied by $\gamma(\cdot; \mu)^{-1} \left(u(\cdot, \cdot; \mu) \right)$. This equation will be called 'calibrated equation'. A necessary condition is the smoothness of:

$$\forall \mu,\ t \to \gamma(t; \mu)^{-1}.$$

   The sense in which this should be understood can vary depending on the framework considered. Namely, it depends on the way the family $\mathcal{F}$ acts on the solution manifold. It will be explicited for one example in the next section.

3. the following application:

$$\forall \gamma \in \mathcal{F},\ \gamma : \left\{ \begin{array}{lcl} X & \to & X \\ u & \mapsto & \gamma(u) \end{array} \right.$$

   is smooth. This property will be needed to derive the calibrated equation. It also naturally appears when studying the stability and accuracy of numerical schemes.

4. the application $\gamma \circ \mathcal{L}_\mu$ should have an explicit form such as:

$$\forall \mu \in \mathcal{D},\ \forall \gamma \in \mathcal{F},\ \exists \mathcal{M}_{\mu,\gamma},\ \mathcal{L}_\mu\left(\gamma(v)\right) = \gamma\left(\mathcal{M}_{\mu,\gamma}(v)\right)$$

   where $\mathcal{M}_{\mu,\gamma}$ is some differential operator.

5. from the knowledge of the calibrated solution $\gamma(t;\mu)^{-1}\left(u(\cdot,\cdot;\mu)\right)$, and of the current calibration parameter $\gamma(t;\mu)$, we need to be able to go back to the true, 'uncalibrated' solution. This means that the applications $\gamma \in \mathcal{F}$ should be invertible.

Remark 17 *The smoothness with respect to μ is never discussed here, as it is not required. We refer to the chapter 5 of this thesis, where we will try to adapt optimal control methods to this particular framekwork.*

Following [19], we call such a process a decomposition of a solution $u(\cdot,t;\mu)$ into two components:

- phase: the calibration function $\gamma(t;\mu) \in \mathcal{F}$
- shape: the solution "calibrated" $v := \gamma(t;\mu)^{-1}\left(u(\cdot,t;\mu)\right)$.

We will now present one framework that satisfies the previous specifications.

3.2.2 One possible framework

This section describes the freezing method, introduced in [19]. The mathematical analysis is based on Lie group theory. We refer to [78] and the references therein for a rigorous presentation of Lie groups and Lie group actions, and restrict ourselves to formal arguments. From now on, let G be a Lie group and let us denote by a some group action on X:

$$\begin{array}{rccc} a: & G \times X & \to & X \\ & (\gamma, u) & \mapsto & a(\gamma, u). \end{array} \tag{3.9}$$

If we compare to the previous section, that means that we parametrize our family of applications $\mathcal{F}$ by some group G. The Lie group setting makes sense for two reasons. First, the applications should be invertible. Second, the parameters should lie on a smooth manifold. The problem is now to find appropriate

- Lie group: G
- group action: a
- application γ:

$$\begin{array}{ccc} [0,T] \times \mathcal{D} & \to & G \\ (t,\mu) & \mapsto & \gamma(t;\mu) \end{array} \tag{3.10}$$

satisfying the list of specifications of the previous section. The most important one, and the sole reason of this whole chapter, is that the calibrated solution manifold

$$\tilde{\mathcal{M}} = \{a(\gamma(t;\mu)^{-1}, u(\cdot,t;\mu))\} \subset X \tag{3.11}$$

has a Kolmogorov n-width with good decay properties.

We restrict ourselves to group actions acting linearly on X. That is, we choose

$$\forall \gamma \in G,\ a(\gamma, \cdot) : \left\{ \begin{array}{ccc} X & \to & X \\ v & \mapsto & a(\gamma, v) \end{array} \right.$$

as a linear application. This solves right away the 3rd point in the specifications section. Other choices are in theory possible, but the analysis of the calibrated equations would require more involved ingredients than the one that will be discussed here. The 5th point of the specifications is also trivially satisfied in this Lie group framework, as the group action is invertible.

Throughout this section, we explicit the different notions on one simple example. We consider the pure transport equation with constant speed:

$$\left\{ \begin{array}{rcl} u_t + c u_x & = & 0 \text{ on } \mathbb{R} \\ u(t=0) & = & u_0 \end{array} \right. \tag{3.12}$$

where the parameter space is $\mathcal{D} = [c_{min}, c_{max}]$. Is natural to pick $G = \mathbb{R}^d$ and the following group action:

$$a : \left\{ \begin{array}{ccc} \mathbb{R}^d \times X & \to & X \\ (\gamma, u) & \mapsto & u(\cdot + \gamma) \end{array} \right. \tag{3.13}$$

It is clear that this group action, for fixed γ, acts linearly on X. The only sensible choice for γ is:

$$\gamma(c, t) := ct. \tag{3.14}$$

With these choices, we can check the remaining specifications:

1. the calibrated solution manifold has a (very) satisfactory n-width:

$$d_n(\tilde{\mathcal{M}}) = \left\{ \begin{array}{l} 1 \text{ if } n = 0 \\ 0 \text{ else .} \end{array} \right. \tag{3.15}$$

   That is, one shape function is enough to capture the whole simulation.

2. γ is a smooth function of t

3. as $\mathcal{L} : u \to c u_x$, it is trivial to check

$$\forall \gamma \in G,\ \forall u \in X,\ a(\gamma, \mathcal{L}(u)) = \mathcal{L}\left(a(\gamma, u)\right).$$

   So we have $\mathcal{M}_\gamma = \mathcal{L}$, for all γ.

The question is now to find a calibrated equation, whose solution would be in the calibrated manifold, equivalent to the initial problem.

3.2.3 The freezing method

We go back to the general case. We start by a **formal** derivation of an equivalent formulation on the shape component. Let

$$\{g(t;\mu) \in G,\ (t,\mu) \in ([0,T] \times \mathcal{D})\}$$

be some fixed, well chosen, phase component function. The corresponding calibrated solution manifold is:

$$\tilde{\mathcal{M}} := \left\{ v(\cdot, t; \mu) := a\left(g(t;\mu)^{-1}, u(\cdot, t; \mu)\right),\ (t;\mu) \in [0,T] \times \mathcal{D} \right\} \tag{3.16}$$

The PDE satisfied by v is given by:

$$\left(a(g(t;\mu),v(\cdot,t;\mu))\right)_t + \mathcal{L}_\mu\left(a(g(t;\mu),v(\cdot,t;\mu))\right) = 0. \tag{3.17}$$

We need some smoothness hypothesis on the group action. This corresponds to the second specification. We restrict ourselves to the group actions such that:

$$\forall v \in X,\ a(\cdot,v) : \left\{ \begin{array}{ccc} G & \to & X \\ g & \mapsto & a(g,v) \end{array} \right. \tag{3.18}$$

is continuously differentiable. Denote a_1 its derivative.

$$\forall v \in X,\ \gamma \in G,\ a_1(\gamma,v) : \left\{ \begin{array}{ccc} T_\gamma G & \to & X \\ \lambda & \mapsto & a_1(\gamma,v)\lambda \end{array} \right. \tag{3.19}$$

a_1 is supposed to be a continuous linear operator and $T_\gamma G$ is the tangent space of G at γ.

Remark 18 *For details on Lie algebras, see [78].*

In [19], they assume that $\mathcal{L}$ is equivariant under the group action, that is:

$$\forall \gamma \in G,\ a\left(\gamma, \mathcal{L}_\mu\left(u\right)\right) = \mathcal{L}_\mu a\left(\gamma, u\right) \tag{3.20}$$

We rather use a weaker assumption. We suppose that there exists (a possibly different) differential operator $\mathcal{M}_{\mu,\gamma}$ such that

$$\forall \gamma \in G,\ \mathcal{L}_\mu\left(a\left(\gamma,u\right)\right) = a\left(\gamma, \mathcal{M}_{\mu,\gamma}\left(u\right)\right)$$

We want to insist on the fact that equivariance is not a core requirement for the method. This is discussed in section 3.8.

The equation satisfied by the calibrated solution, equation (3.17) is equivalent to:

$$a\left(g(t;\mu), v_t(\cdot,t;\mu)\right) + a_1(g(t;\mu),v(\cdot,t;\mu))g_t(t;\mu) + a\left(g(t;\mu), \mathcal{M}_{\mu,\gamma}(v(\cdot,t;\mu))\right) = 0$$

Using the fact that the group action is invertible, we get:

$$v_t(\cdot,t;\mu) + a\left(g(t;\mu)^{-1},\ \left[(a_1(g(t;\mu),v(\cdot,t;\mu))g_t(t;\mu))\right]\right) + \mathcal{M}_{\mu,\gamma}(v(\cdot,t;\mu)) = 0 \tag{3.21}$$

To better grasp the new ingredients, we go back to our model translation problem. The tangent space can be identified with $\mathbb{R}^d$. What does the smoothness of the group action, equation (3.18), imply in this simple case ? Let $\lambda \in \mathbb{R}^d$.

$$\forall h \in \mathbb{R},\ a(\gamma + h\lambda, v) - a(\gamma, v) = v(\cdot + (\gamma + h\lambda)) - v(\cdot + \gamma) \tag{3.22}$$

If we suppose that X is embedded in $C^1(\mathbb{R})$, then a is differentiable in the sense of (3.18) and the corresponding derivative a_1 is given by:

$$a_1(\gamma, v)\lambda = a(\gamma, \nabla v \cdot \lambda). \tag{3.23}$$

3.2.4 Phase component

The previous section assumed no restriction on the choice of the phase function, apart from its smoothness. We now have to show how to add a constraint on the phase function that actually reduces the n-width. They propose in [19] the following generic coupled problem:

$$\left\{ \begin{array}{rcl} v_t(\cdot,t;\mu) + a\left(g(t;\mu)^{-1}, (a_1(g(t;\mu), v(\cdot,t;\mu))g_t(t;\mu))\right) + \mathcal{M}_\mu v(\cdot,t;\mu) & = & 0 \\ \Phi(v,g) & = & 0 \text{ in } \mathcal{A}^* \end{array} \right. \tag{3.24}$$

where $\mathcal{A}$ is the tangent space of G at g and $\mathcal{A}^*$ the dual space of $\mathcal{A}$.

We give one example of constraint that enters this frameowkr. Let u^* be some reference solution. Suppose that the calibration is chosen so that solutions in the calibrated solution manifold are as close to u^* as possible, in X norm. That is, we choose $g \in G$ as:

$$g : v \rightarrow \underset{g \in G}{\text{arginf}} \, \|a(g, u^*) - v\|_X^2.$$

First of all, it is easy to see why this constraint goes towards a quicker decay of the n-width. First order optimality constraint gives:

$$\forall \delta g \in \mathcal{A}, \; < a_1(g, u^*)\delta g \; , a(g, u^*) - v >_X = 0,$$

where $< \cdot, \cdot >_X$ denotes the standard scalar product in X. We are thus interested in the zeros of the following application:

$$\forall v \in X, \; \forall g \in G, \; \Phi_{v,g} : \left\{ \begin{array}{rcl} \mathcal{A} & \rightarrow & \mathbb{R} \\ \delta g & \mapsto & < a_1(g, u^*)\delta g, a(g, u^*) - v >_X \; . \end{array} \right.$$

This enters the framework of equation (3.24).

The existence of solutions to the coupled system depends on the properties of the constraint equations. Given v, a necessary condition is for the constraint equation to be well posed, and to result in a smooth calibration parameter g. We refer to [19] for more details.

3.2.5 Conclusions on the freezing method

This is it, we have a method to overcome the difficulties associated with the development of ROM for (a restricted class of) solution manifolds with large n-width. Corresponding numerical example for can be found in [105]. The purpose of this section is to motivate the need for an alternative.

One disadvantage of the freezing method has appeared from the beginning: it is its complexity. The development of a theoretically sound framework combining the study of PDE's with Lie group and Lie algebra theory is not only (far) from the scope of this thesis, it is also (to my knowledge) not available in the literature. To my understanding, the strongest general result in [19] concerns smooth solutions (say continuously differentiable), and is only local in time.

We now discuss another property of the freezing method. Let the inviscid Burgers equation:

$$u_t + \left(\frac{u^2}{2}\right)_x = 0.$$

Taking translation as the calibration process, it is easy to see why the calibrated equation is given by:

$$v_t + (v - \gamma(t))v_x = 0$$

for some function $t \mapsto \gamma(t)$, that can for instance be the shock's speed. To numerially solve this equation on the shape function, one needs to construct a scheme taylored for this task. Using a scheme designed for convection dominated problems to solve for v is not a viable strategy. The fact that the calibration procedure has made us lost track of the physical intuition can make the construction of such a scheme a complex task.

What we do in this chapter is less ambitious. We construct a method that whilst still using the calibrated solution manifold, manages to keep the properties of the original, physically meaningful equation.

3.2.6 Road Map

We are ready to set up the notations. They will be used until the end of the chapter. $\mathcal{F}$ denotes a family of applications:

$$\mathcal{F} = \left\{ F : \overline{\Omega} \to \overline{\Omega} \right\}. \tag{3.25}$$

Note right away that we have restricted the choice of $\mathcal{F}$ to family of smooth mappings. The elements of $\mathcal{F}$ will be parametrized by t and μ, that is, we consider:

$$\begin{array}{ccccc} [0,T] & \times & \mathcal{D} & \to & \mathcal{F} \\ (t,\mu) & & & \mapsto & F_{t;\mu} \end{array}$$

Finally, we take the action group resulting in the following calibrated solution manifold:

$$\mathcal{M}^x_{\mathcal{F},\mathcal{D}} := \left\{ u(F^{-1}_{t;\mu}(\cdot), t; \mu),\ \mu \in \mathcal{D},\ t \in [0,T] \right\} \tag{3.26}$$

As for the freezing method, the set $\mathcal{F}$ is based on a priori expertise on the behavior of the solution. The offline section is also similar as we pick elements in $\mathcal{F}$ to make the n-width of $\mathcal{M}^x_{\mathcal{F},\mathcal{D}}$ as small as possible. The novelty appears on the way we use the calibrated solution manifold in the online stage.

3.3 Algorithm

For simplicity, let us assume that we are using an explicit Euler scheme for the time discretization. Extensions to implicit, higher order time discretization, or more involved conservative numerical scheme, is straightforward[1]. Our semi-discretized PDE then becomes

$$\left\{ \begin{array}{lcll} \frac{u^{n+1}-u^n}{dt} + \mathcal{L}(u^n;\mu) & = & 0 & \text{in } \Omega \\ u(\cdot, t=0;\mu) & = & u_0(\cdot,\mu) & \text{in } \Omega \\ B(u^n;\mu) & = & 0 & \text{on } \partial\Omega \end{array} \right. \tag{3.27}$$

Here, as is classical, dt denotes the time step, and u^n an approximation for the solution to (3.8) at time ndt.

Assume that we have a basis $\{\phi_i\}$ such that $\text{span}\{\phi_i\}$ approaches the calibrated solution manifold $\mathcal{M}^x_{\mathcal{F},\mathcal{D}}$ defined in (3.26) to a given accuracy. Since $\mathcal{M}^x_{\mathcal{F},\mathcal{D}}$ is assumed to be of small Kolmogorov n-width, we expect that we can find such a basis of moderate size. At each time

[1] Note that, of course, this choice of an explicit scheme involves a limitation on the time step due to a CFL condition that can be severe for an accurate finite element or finite difference scheme but reveals to be moderate in the reduced basis framework.

step, we look for coordinates $(\alpha_i^{n+1})_i$ on the reduced basis and an application $F_{n+1} \in \mathcal{F}$ such that $u(\cdot, t^{n+1}; \mu)$ is well approximated by:

$$u^{n+1} := \sum_{i=1}^{M} \alpha_i^{n+1} \phi_i \circ F_{n+1}. \tag{3.28}$$

In order to expect the search for F_{n+1} be computationally tractable, let us assume that our family $\mathcal{F}$ can be parametrized by a few parameters: that is

$$\forall F_{t;\mu} \in \mathcal{F}, \quad \exists (\gamma_j)_j, \quad \text{such that} \quad F_{t;\mu} = F\left[\gamma_1(t;\mu), \ldots, \gamma_m(t;\mu)\right]. \tag{3.29}$$

In the discrete setting, the search for F_{n+1} then reduces to the search for $(\gamma_j^{n+1})_j$, and we set $F_{n+1} = F\left[\gamma_1^{n+1}, \ldots, \gamma_m^{n+1}\right]$.

We are thus simultaneously looking for a proper appropriate reduced space (defined as the span of the $(\phi_i \circ F_{n+1})_i$) and for coordinates on this reduced space. We have chosen to derive our solution from some minimization problem of the form:

$$(\gamma_j^{n+1}, \alpha_i^{n+1}) = \underset{(\gamma_j, \alpha_i)}{\operatorname{argmin}} \left\| \sum_i \alpha_i \ \phi_i \circ F([\gamma_j]_j) - u^n + dt\mathcal{L}(u^n; \mu) \right\| \tag{3.30}$$

for some appropriate norm $\|\cdot\|$ on X.

Remark 19 *It is interesting to note that our approach, in this context, may be presented as a shock fitting method, and thus one may fear that it will suffer from the classical drawback of this class of approach, especially the difficulty to generalize to multidimensional framework. One reassuring element is that the position of the fitting F_n is not defined through the Rankine-Hugoniot conditions but through the minimization process (3.30), and the evolution in time can be chosen to follow any appropriate conservative numerical scheme.*

Several choices are possible for the sense in which we will minimize this quantity. One example will be given in the next section. We propose the following generic algorithm.

Initialize α_i and γ_j

$$(\alpha_i^{n+1,0}, \gamma_j^{n+1,0}) = (\alpha_i^{ini}, \gamma_j^{ini}) \tag{3.31}$$

α_i^{ini} and γ_j^{ini} will depend on the previous timesteps, namely on $(\alpha_i^k)_i$ and $(\gamma_j^k)_j$ for $k \leq n$.

Then, assuming that $(\alpha_i^{n+1,q}, \gamma_j^{n+1,q})$ are known for some internal iteration $q \geq 0$, we proceed

Fit the α_i given $[\gamma_j^{n+1,q}]_j$

Find $(\alpha_i^{n+1,q+1})_i$ that minimizes the following quantity (in some sense):

$$\sum_i \alpha_i^{n+1,q+1} \phi_i \circ F\left(\left[\gamma_j^{n+1,q}\right]_j\right) - u^n + dt\mathcal{L}(u^n; \mu) \tag{3.32}$$

Fit the γ_j given $(\alpha_i^{n+1,q+1})_i$

Find $(\gamma_j^{n+1,q+1})_j$ that minimizes the following quantity (in some sense):

$$\sum_i \alpha_i^{n+1,q+1}\phi_i \circ F\left([\gamma_j^{n+1,q+1}]_j\right) - u^n + dt\mathcal{L}(u^n;\mu) \tag{3.33}$$

until convergence (for which, say $q = q^*$). Then, we set

$$(\alpha_i^{n+1}, \gamma_j^{n+1}) = (\alpha_i^{n+1,q^*+1}, \gamma_j^{n+1,q^*+1}). \tag{3.34}$$

3.4 Illustration on the viscous Burger's equation in one dimension

The viscous Burger's equation has already received some attention in the reduced modeling context. We mention [135] for the stationary case and when the solution manifolds can be well represented by a small finite dimensional linear space, without any calibration.

We consider $\Omega = (-1, 1)$, and solve for the time dependent viscous Burger equation with no forcing term and periodic boundary conditions (we will see later why these are important in our analysis):

$$\begin{cases} u_t + \nu u u_x - \epsilon u_{xx} = 0 \text{ in } [0,T]\times\Omega \\ u|_{t=0} = u_0 \\ u \text{ periodic} \end{cases} \tag{3.35}$$

The parameters of this problem are the triplets: $\mu = (u_0, \nu, \epsilon)$. We want to choose a parameter domain $\mathcal{D}$ in order that the problem is

- convection dominated so that the solution manifold has a large Kolmogorov n-width
- not too stiff so as not to be bothered by stabilization issues as mentioned in the introduction, hence, we shall only consider the cases $\epsilon \geq \epsilon_0 > 0$ (see the recent paper [92] that tackles this problem).

We have identified the following parameter range:

$$\mathcal{D} = \begin{cases} \lambda \in [0.5, 1.3], \\ \nu \in [4., 6.], \\ \epsilon \in [0.04, 0.2]. \end{cases} \tag{3.36}$$

3.4.1 Variational formulation and truth approximation

For the truth approximation to the solution of problem (3.35), let us consider a semi implicit scheme (so as not to be bothered by a two stringent stability constraint) with time step dt_{truth}. Let

$$X = H^1_{per}(\Omega) \tag{3.37}$$

and let us denote by $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$ the usual L^2 inner product and norm. For each $\mu = (u_0, \nu, \epsilon) \in \mathcal{D}$, for the semi-discrete (in time) truth problem, we are looking for $u^{n+1} \in X$ (approximation of $u(., (n+1)dt_{truth}, \mu)$) such that: $\forall v \in X$

$$\langle u^{n+1}(\mu), v\rangle + dt_{truth}\epsilon a(u^{n+1}(\mu), v) = \langle u^n(\mu), v\rangle - dt_{truth}\nu c(u^n(\mu), u^n(\mu), v) \tag{3.38}$$

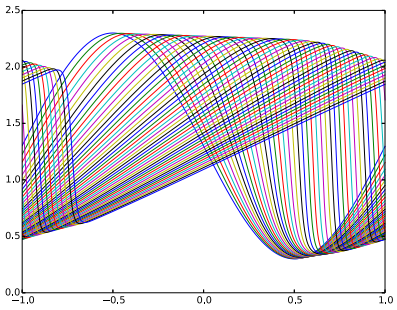


Figure 3.1: Snapshots of the solution to the unsteady viscous Burger equation with $u_0 = \lambda + sin(x)$, $\lambda = 1.3$, $\nu = 4$, $\epsilon = 0.04$

where

$$c(w, z, v) = \int_\Omega w\ z_x\ v \quad \text{and} \quad a(w, v) = \int_\Omega w_x\ v_x. \tag{3.39}$$

This semi-discretized problem is trivially well posed. In order to finalize the discretization, let us introduce an appropriate finite element discretization, the truth approximation space, $X^{\mathcal{N}}$. We pick it fine enough so that, with the chosen time step dt_{truth}, it is able to represent well our solution manifold. From now on, we will consider that the exact solution $u(\cdot, t; \mu)$ and the "truth" solution $u^{\mathcal{N}}(\cdot, t, \mu)$ cannot be distinguished.

3.4.2 Model order reduction — offline stage

As mentioned earlier, the first question we need to answer is: does our solution manifold $\mathcal{M}^x_{\mathcal{D}}$ (in practice represented by $\mathcal{M}^{x,truth}_{\mathcal{D}}$) have a large Kolmogorov n-width? And if so, can we find better behaved "calibrated" manifold solution? Figure 3.1 shows some snapshots $\left\{u(\cdot, t^k; \mu),\ k \in 1 \dots K\right\}$ taken in $\mathcal{M}^x_{\mathcal{D}}$ for some parameters.

From basic expertise on the Burger's equation, we choose the following mapping family: $\mathcal{F} = \{F_{t;\mu}\}$, where $F_{t;\mu}$ are defined as translation operators:

$$\begin{array}{rccl} F_{t;\mu}: & \Omega & \to \Omega \\ & x & \mapsto x - \gamma(t; \mu) \end{array} \tag{3.40}$$

with $\gamma(t; \mu) \in \mathbb{R}$. With this choice, our family of mappings is a one parameter family, i.e:

$$\mathcal{F} = \{F(\gamma),\ \gamma \in \mathbb{R}\}. \tag{3.41}$$

Unlike in the pure translation problem of the introduction (3.4), our parameter γ is not constant (it is a function of μ and time) and has no analytical expression. One possible calibration (and the most natural one) is presented in Figure 3.2, where we pick γ manually so that all steepest points coincide. Our calibrated solution manifold is then

$$\mathcal{M}^x_{\mathcal{F},\mathcal{D}} = \{u(\cdot - \gamma(t; \mu), t; \mu),\ t \text{ in } [0, T], \mu \in \mathcal{D}\}, \tag{3.42}$$

that is represented in Figure 3.2 where we understand that the Kolmogorov n-width of $\mathcal{M}^x_{\mathcal{F},\mathcal{D}}$ is smaller than the original one represented in Figure 3.1.

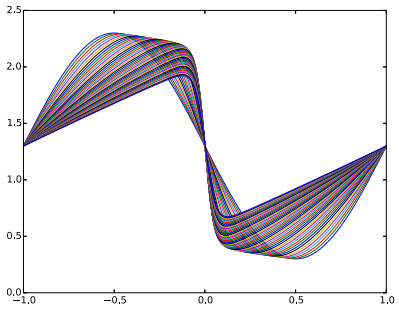


Figure 3.2: Calibrated set of the above snapshots for $u_0 = \lambda + sin(x)$, $\nu = 4$, $\epsilon = 0.04$

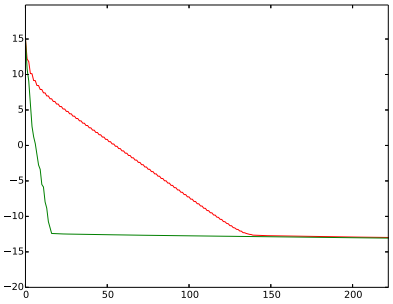


Figure 3.3: Eigenvalues of the POD decomposition of the original set of snapshots (in red) and of the calibrated set of snapshots (in green)

This is confirmed in Figure 3.3 which presents the decay of the POD eigenvalues in logarithmic scale for $\mathcal{M}^x_{\mathcal{D}}$ and $\mathcal{M}^x_{\mathcal{F},\mathcal{D}}$. As we could have expected, to achieve a fixed accuracy, the number of POD modes needed to represent the calibrated manifold is much smaller than the number of modes needed for the original solution set. To confirm this, we present in Figure 3.4 the 3rd and 6th POD modes of the calibrated and non calibrated simulations. As we can see, in the calibrated case, with just 3 modes, our L^2-projection focuses on reproducing the shock, whereas in the non calibrated case, the modes desperately try to represent shocks centered anywhere in Ω. We mention again the fact that even in the calibrated case, our algorithm could be improved using L^1-minimization. We present in Figure 3.5 the projection of one of the snapshot on the first three POD modes. With 10 POD modes in the uncalibrated case, the projection shown in Figure 3.5 exhibit the oscillatory behavior as described in [2].

At this stage, we suppose that we have found a "calibrated" solution manifold, with nice Kolmogorov n-width decay. That is, we have calibrated an original dataset, and obtained a reduced orthonormal basis:

$$\text{span}\,\{\phi_i,\ i = 1 \dots M\} \subset X \tag{3.43}$$

that approximates well the calibrated solution manifold $\mathcal{M}^x_{\mathcal{F},\mathcal{D}}$.

We now need to explicit the algorithm presented in the section 3.3. The biggest question is

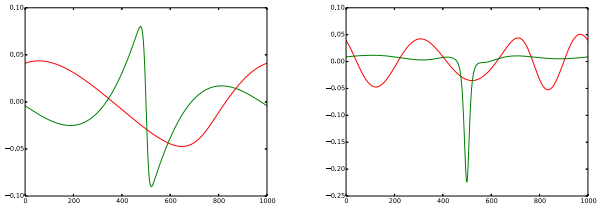


Figure 3.4: 3rd (left) and 6th (right) POD modes for the calibrated (green) and original (red) simulations

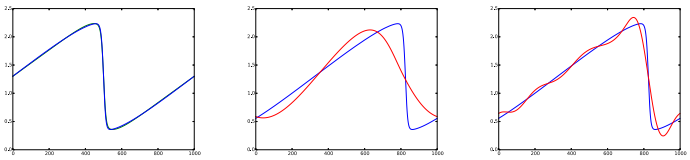


Figure 3.5: Projection of a snapshot (blue) on:

- left: 3 POD modes in the calibrated case
- center: 3 POD modes in the non calibrated case
- right: 10 POD modes in the non calibrated case

how do we pick the $F \in \mathcal{F}$ at each time step?

3.4.3 Model order reduction — online stage

As was introduced in the previous section (see (3.27)), for the time semi-discretization of the RBM approach, we use a forward Euler discretization with a time step dt that may be different from dt_{truth}, at each time step we are looking for the solution to the following elliptic problem [2]:

$$u^{n+1} = u^n - dt\nu u^n u_x^n + dt\epsilon u_{xx}^n \tag{3.44}$$

with periodic boundary conditions over $(-1, 1)$, which leads to the following variational formulation that will be used to provide the Galerkin formulation of the RBM: knowing u^n, compute $u^{n+1} \in X$ such that

$$\forall v \in X,\ \langle u^{n+1}(\mu), v\rangle = \langle u^n(\mu), v\rangle - dt\nu c(u^n(\mu), u^n(\mu), v) - dt\epsilon a(u^n(\mu), v). \tag{3.45}$$

One could fear that a problem with this discretization is the stringent CFL condition on the time-step. Our reduced basis formulation will allow for very fast computation, which will mitigate this issue on which we shall dwell upon later. As said already, we could also consider an implicit Euler scheme. We refer to [135] (stationary) and [101] (non stationary), for the development of reduced order model in that case.

[2] Indeed there is no reason why using the same discretization in time for the truth solution and for the reduced basis scheme

The full RBM discretization starts from the knowledge of the (supposedly accurate) approximation of u^n as an expansion

$$u^n := \sum_{i=1}^{M} \alpha_i^n \phi_i \circ F_n, \tag{3.46}$$

where the $\{\phi_i\}_i$ are the reduced basis elements of the good approximation of the calibrated solution manifold that have been introduced in (3.43) as a result of the offline process. F_n is here $F(\gamma^n)$ where γ^n is the current translation value. In order to deduce the next approximation,

$$u^{n+1} := \sum_{i=1}^{M} \alpha_i^{n+1} \phi_i \circ F_{n+1}, \text{ where } F_{n+1} = F(\gamma_{n+1}) \tag{3.47}$$

as described in the previous section, we iterate between the search for the reduced coordinates $(\alpha_i^{n+1})_i$ and for the mapping F_{n+1} i.e. for the translation parameter γ^{n+1}, we initialize these entities as follows:

$$\begin{aligned} \alpha_i^{n+1,0} &= \alpha_i^n \\ \gamma^{n+1,0} &= \gamma^n + \left(\gamma^n - \gamma^{n-1}\right). \end{aligned} \tag{3.48}$$

In the first part of the iterative step indexed by q, assuming we know $((\alpha_i^{n+1,q})_i, \gamma^{n+1,q})$ we fit the α_i for a fixed translation parameter γ, i.e.we are looking for $(\alpha_i^{n+1,q+1})_i$ that satisfy

$$\{\alpha_i^{n+1,q+1}\} = \underset{(\alpha_i)_i \in \mathbb{R}^{\mathbb{N}}}{argmin} \left\| \sum_i \alpha_i \phi_i \circ F(\gamma^{n+1,q}) - u^n - dt\nu u^n u_x^n + dt\epsilon u_{xx}^n \right\|_2^2 \tag{3.49}$$

The nice feature with the chosen norm is that we pick our reduced coordinates such that our residual is orthogonal to the translated reduced space, the space spanned by the $\{\phi_i \circ F(\gamma^{n+1,q})\}_i$. Using u^n's expansion on its reduced basis, the coefficients $\{\alpha_i^{n+1,q+1}\}_i$ are given by the first-order optimality condition:

$$\begin{aligned} \alpha_i^{n+1,q+1} = & \quad \textstyle\sum_j \alpha_j^n \langle \phi_j \circ F(\gamma^n), \phi_i \circ F(\gamma^{n+1,q}) \rangle \\ - & \quad dt\nu \quad \textstyle\sum_j \sum_p \alpha_j^n \alpha_p^n \langle \phi_j \circ F(\gamma^n) \left(\phi_p \circ F(\gamma^n)\right)_x, \phi_i \circ F(\gamma^{n+1,q}) \rangle \\ - & \quad dt\epsilon \quad \textstyle\sum_j \alpha_j^n \langle \left(\phi_j \circ F(\gamma^n)\right)_x, \left(\phi_i \circ F(\gamma^{n+1,q})\right)_x \rangle. \end{aligned} \tag{3.50}$$

In order to evaluate this expression, we need to compute the following integrals:

$$\begin{cases} \forall i,j, & \int_\Omega \phi_j \circ F(\gamma^n)(x) \phi_i \circ F(\gamma^{n+1,q})(x) \\ \forall i,j,p, & \int_\Omega \phi_j \circ F(\gamma^n)(x) \left(\phi_p \circ F(\gamma^n)\right)_x (x) \phi_i \circ F(\gamma^{n+1,q})(x) \\ \forall i,j, & \int_\Omega \phi_j \circ F(\gamma^n)_x(x) \phi_i \circ F(\gamma^{n+1,q})_x(x) \end{cases} \tag{3.51}$$

We will see in the next subsection how to achieve efficient offline/online decomposition for these quantities.

Once this is done, we fit the γ. Let us define first the residual function $r(\gamma)$:

$$r(\gamma) = \left\| \sum_i \alpha_i^{n+1,q+1} \phi_i \circ F_\gamma - u^n - dt\nu u^n u_x^n + dt\epsilon u_{xx}^n \right\|_2^2, \tag{3.52}$$

then we choose $\gamma^{n+1,q+1}$ as the "best", i.e residual minimizing, translation parameter. It is given by:

$$\gamma^{n+1,q+1} = \underset{\gamma}{\text{argmin}}\ r(\gamma) \tag{3.53}$$

Next we develop $r(\gamma)$:

$$\begin{array}{rl} r(\gamma) & = \left\| \sum_i \alpha_i^{n+1,q+1} \phi_i \circ F(\gamma) \right\|_2^2 + \| u^n - dt\nu u^n u_x^n + dt\epsilon u_{xx}^n \|_2^2 \\ & \quad -2\langle \sum_i \alpha_i^{n+1,q+1} \phi_i \circ F(\gamma), u^n - dt\nu u^n u_x^n + dt\epsilon u_{xx}^n \rangle. \end{array} \tag{3.54}$$

The second term is independent of γ. The first one, using periodicity, happens also to be independent of γ. We can thus replace the minimization of r by the minimisation of the following quantity $\tilde{r}$:

$$\tilde{r}(\gamma) = -\langle \sum_i \alpha_i^{n+1,q+1} \phi_i \circ F(\gamma), u^n - dt\nu u^n u_x^n + dt\epsilon u_{xx}^n \rangle. \tag{3.55}$$

Here again, we we need to evaluate the quantities

$$\begin{cases} \forall i,j, & \int_\Omega \phi_j \circ F(\gamma^n)(x) \phi_i \circ F(\gamma)(x) \\ \forall i,j,p, & \int_\Omega \phi_j \circ F(\gamma^n)(x) \left(\phi_p \circ F(\gamma^n)\right)_x (x) \phi_i \circ F(\gamma)(x) \\ \forall i,j, & \int_\Omega \phi_j \circ F(\gamma^n)_x(x) \phi_i \circ F(\gamma)_x(x) \end{cases} \tag{3.56}$$

for various values of γ in order to derive the value of γ that minimizes r (or $\tilde{r}$).

3.4.4 Offline/Online decomposition of the expressions depending on γ.

In both the search for γ (see (3.55)) and $(\alpha_i)_i$ (see (3.51)), we need to compute scalar products of the form:

$$\langle \psi_i \circ F(\gamma^n), \psi_j \circ F(\gamma) \rangle \tag{3.57}$$

where ψ can be one of the POD basis or one of its x-derivatives. γ^n and γ can take any value in Ω. Our key ingredient here is that, due to translation invariance (because we are in a periodic settings), we can replace the previous terms by

$$\langle \psi_i \circ F(\gamma^n - \gamma), \psi_j \rangle = \langle \psi_i \circ F(\Delta\gamma), \psi_j \rangle. \tag{3.58}$$

We have plotted in Figure 3.6 these quantities (after rescaling) as a function of $\Delta\gamma$ for some pairs of chosen ψ's and we notice that, as can be expected because we are essentially using a primitive function of the integrant, these are regular functions of $\Delta\gamma$.

For a sufficiently small time step, we expect $\Delta\gamma$ to be of order $dt * c$ where c is some local characteristic velocity. We have chosen the following method:

- precompute the scalar products for a predefined set of values of $\Delta\gamma$
- using some regularity hypothesis, use spline interpolation to get approximated values for all γ in $[-dt * c_{max}, dt * c_{max}]$, where c_{max} is the maximum expected shock speed during the simulation.

Remark 20 *For the optimization of $\tilde{r}$ we have also tested to linearize our problem around γ^n which leads to a doable method but does not work better that the above.*

Remark 21 *A common comment about this method is about mesh interpolation, and the related numerical errors. In the offline part, we have indeed to interpolate between meshes. But, as the computational time is not much an issue, this can be done as precisely as required. During the online section, the only thing that is required is the interpolation between the discrete quantities computed in (3.58). This error can be quantified offline. See figure 3.6 for an idea of the quantities that we are interpolating. This point will be discussed in section 3.8 of this chapter.*

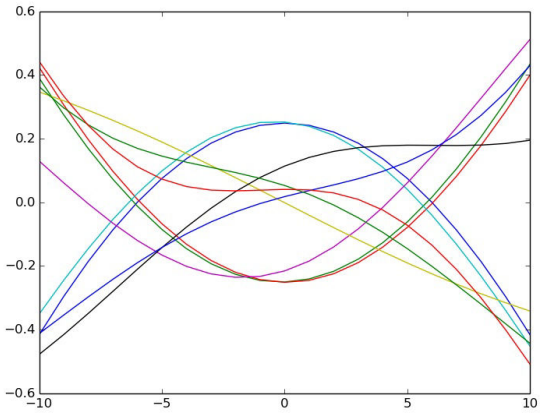


Figure 3.6: A few values of the quantities (3.57) as a function of $\Delta\gamma$. The x axis is scaled to multiples of $c * \Delta t$.

3.5 Numerical results

3.5.1 About the CFL Condition

We represent in Figure 3.7 the value of the CFL condition of our reduced scheme using the space calibrated $\mathcal{M}^x_{\mathcal{F},\mathcal{D}}$ as a function of the dimension M of the discrete space expressed in Equation (3.46). Of course, the bigger the reduced basis, the smaller the time step required for stability. We remark that there is a plateau for large values of M that is above the CFL-condition for the truth solver. More importantly, for $M = 5$, we can use a discrete time step $3,000$ times bigger than the one of the fine (finite element) scheme (that was $dt_{truth} \leq 10^{-6}$).

3.5.2 Convergence Illustration

On the next Figure 3.8, we have plotted the L^2-error of the solution of (3.30) in case of problem (3.35) as a function of time for different values of the reduced basis for $dt = 2.5\ 10^{-4}$. The different colors represent various values of M used in (3.46) (Note that on the same figure, the plots close the x axis represent the projection errors (best approximation) of the solution onto the set $\mathcal{M}^x_{\mathcal{F},\mathcal{D}}$ with the exact value of the translation γ). We see that our numerical scheme is convergent, as a function of M. The final accuracy is somehow difficult to grasp since it is a function of Δt and the number of degrees of freedom used in the spatial direction (here M) as for any discretization of an evolution problem. It is also a function of the way the value of γ is found as each time step as a solution of the full minimization problem (3.30). These results are not entirely satisfactory, as there is an order of magnitude difference between error of the reduced scheme and the best approximation error. We will propose one explanation in the next section.

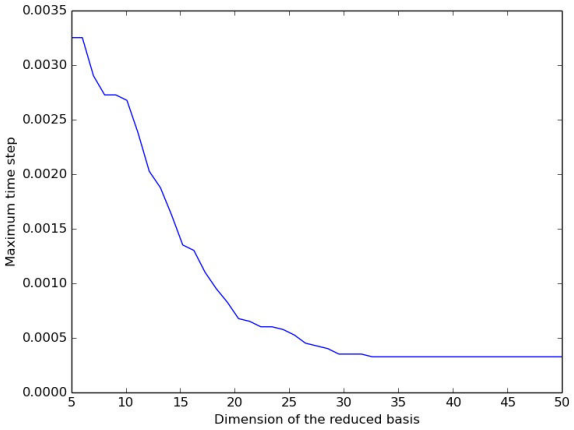


Figure 3.7: A few values of the quantities (3.57) as a function of $\Delta\gamma$. The x axis is scaled to multiples of $c * \Delta t$.

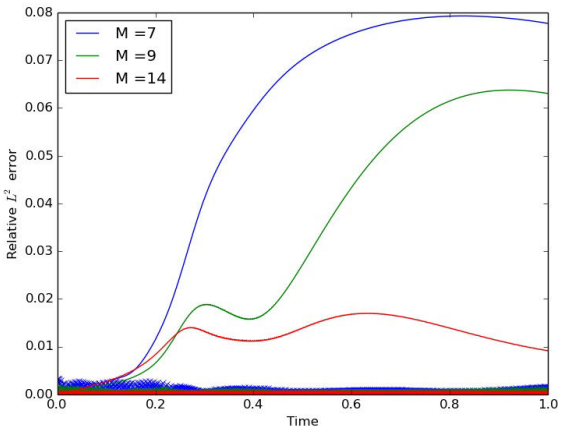


Figure 3.8: Relative L^2-error of the solution as a function of time for different values of the reduced basis. The three curves close to the x-axis (almost overlapping at this scale) are the associated best approximation errors.

3.5.3 Interpretation

In order to explain the order of magnitude difference between the errors obtained with the reduced scheme and the best approximation errors, we compare the results for various parameters. We start by constructing a basis that reproduces well solutions for parameters in $\mathcal{D}$, chosen in (3.36). The first step is to discretize $\mathcal{D}$. We randomly pick the following:

$$\begin{aligned} \lambda \in & \quad \{1.3, 1, 0.8, 0.5\} \\ \nu \in & \quad \{4, 5, 6\} \\ \epsilon \in & \quad \{0.04, 0.08, 0.16, 0.2\} \end{aligned} \tag{3.59}$$

We compute the truth approximation solutions of each of these triplets. Because of the computational cost, we have chosen to use the two level POD presented in chapter 6. That is, we perform a POD on each of the 48 time simulations and then perform a weighted POD on these POD basis. The error incurred by this divide and conquer strategy is controlled.

We present in Figure 3.9, the mean (in time) relative L^2 best projection error, using a basis of size 10, at each sampled parameter. As we can see, the mean projection error is slightly worse

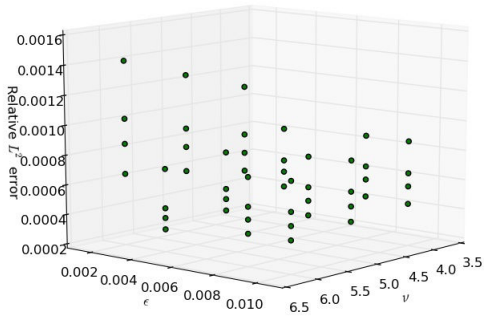


Figure 3.9: Relative L^2 best projection errors, on the training set

for small viscosities. One argument is that lower the viscosity, the stiffer the propagating front. Since our calibration procedure is not exact, the calibrated solution manifold is more difficult to capture.

We now want to compare the previous plot with the errors on the reduced solutions. Their relative L^2 error are displayed in Figure 3.10. Similar to what we have seen on the convergence plot, there is an order of magnitude of difference between the best projection error and the output of the reduced scheme. More importantly, the scatter plot is much more spread out. The low viscosities behave much worse, compared to the other simulations. My interpretation is that the order of magnitude difference is due to the raw Galerkin scheme used in the online section. We can not reproduce the solutions of the truth scheme, that are obtained using an upwinding scheme. This could be solved either using rectification ideas [92], or using the ideas developed in chapter 5.

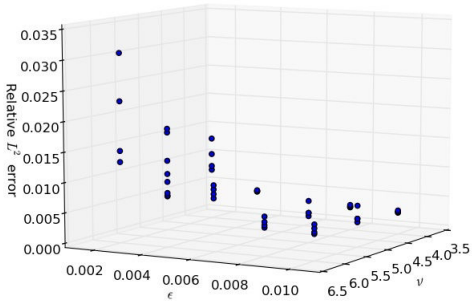


Figure 3.10: Relative L^2 error of the reduced solutions, on the training set

This concludes the analysis of the one dimensional Burgers viscous equation. In the remaining sections, we propose variations around the calibration procedure.

3.6 Extension to non periodic setting

The periodicity has held an important role in the analysis of the calibration method so far. We try in this short section to extend the latter to a non periodic setting. The model equation that we work with is:

$$\begin{cases} \frac{u^{n+1}-u^n}{dt} + \mathcal{L}(u^n;\mu) &= 0 & \text{in } \Omega \\ u(\cdot, t=0;\mu) &= u_0(\cdot,\mu) & \text{in } \Omega \\ B(u^n;\mu) &= g^n & \text{on } \partial\Omega. \end{cases} \tag{3.60}$$

We start by presenting a typical problem entering this framework. Suppose that there is a complex phenomenon/shock/front moving inside Ω. We want to use the ideas of the previous section, that is, we want to use the fact that only the relative positions of the phenomenon/shock/front between two successive time steps is relevant. The absolute position in Ω has not a big influence. How can we translate this into a reduced basis framework ?

3.6.1 The method

We suppose that there exists an 'interior' domain Ω_{int} and a calibrating function γ,

$$\gamma : \begin{cases} [0,T]\times\mathcal{C} & \to & \mathbb{R} \\ (t;\mu) & \mapsto & \gamma(t;\mu), \end{cases}$$

such that the calibrated solution manifold $\mathcal{M}_{int}$ defined as

$$\mathcal{M}_{int} := \{u(\cdot - \gamma(t;\mu), t;\mu)|_{\Omega_{int}},\ t\in[0\dots T],\ \mu\in\mathcal{C}\}$$

has a small n-width. To put it in other words, because we have removed periodicity, we need to truncate around the phenomenon/shock/front.

We require one more assumption in order to make this method relevant. Let $\{\phi_i\}_i$ be some well chosen reduced basis of $\mathcal{M}_{int}$, and Π_Ψ the orthogonal projection onto any orthogonal basis Ψ embedded in X. We suppose that

$$\mathcal{M}_0 := \left\{ u(\cdot - \gamma(t;\mu), t; \mu) - \Pi_{\phi_i \circ F(\gamma(t;\mu))} u(\cdot - \gamma(t;\mu), t; \mu),\ t \in [0 \dots T],\ \mu \in \mathcal{C} \right\} \tag{3.61}$$

has also a small n-width.

Remark 22 *This resembles the notion of 'local Kolmogorov n-width' introduced in chapter 2.*

We can rephrase this using the notations of chapter 2. We decompose Ω into two overlapping subdomains : one outside domain, that is supposed to handle the boundary conditions $\Omega_0(\gamma)$ and one inside domain, supposed to handle the complex structure moving : $\Omega_{int}(\gamma)$. The difference with chapter 2 is that these two subdomains are function of the translation parameter γ. One possible setting is presented in Figure 3.11. We need a few assumptions on Ω_0 and Ω_{int}:

$$\begin{aligned}
& \forall \gamma,\ \Omega = \Omega_0(\gamma) \cup \Omega_{int}(\gamma) \\
& \forall \gamma,\ \partial\Omega \subset \overline{\Omega_0(\gamma)} \\
\exists \alpha, \quad & \forall \gamma,\ \mathrm{dist}(\Omega_{int}(\gamma), \partial\Omega) && > \alpha \\
\exists \beta, \quad & \forall \gamma,\ \mathrm{mes}(\Omega_{int}(\gamma) \cap \Omega_0(\gamma)) && > \beta
\end{aligned}$$

Using assumption (3.61), we will now denote by $\{h_k\}$ a good reduced basis of $\mathcal{M}_0$. In the course

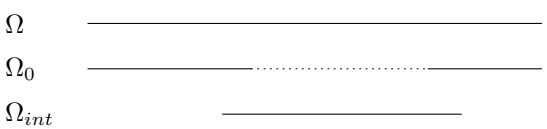


Figure 3.11: This method requires an overlapping decomposition of Ω

of this section, it will be thought of as Fourier/polynomial, coarse Finite Element, and RB type.

We take the following conforming approximation space of X :

$$X^N(\gamma) := \mathrm{span}\,\{\phi_i,\ i = 1 \dots M\} \circ F(\gamma) + \mathrm{span}\{h_k\}.$$

We could also have used a non conforming method, such as the one presented in chapter 2. The reasons why the non conforming approach was necessary do not apply here.

Remark 23 *We need to multiply our 'inside basis' ϕ_i by a smooth indicator function. This is necessary so that the 'outside' solution is smooth. It is also necessary so that we can 'safely' compute the scalar products between $\{\phi_i\}$ and $\{\phi_i \circ F(\delta)\}$. We will now assume that the basis functions $\{\phi_i\}_i$ have zero trace on $\partial\Omega_{int}(\gamma)$.*

3.6.2 Conditioning

One key element of such a method is, of course, the conditioning of the resulting mass/stiffness matrices. It is closely related to the linear dependency of the basis $X^N(\gamma)$. For standard ROM methods, the conditioning can be controlled in the offline phase by using procedures such as Gram-Schmidt orthogonalization. Here, the approximation space depends continuously on the translation parameter γ, and so does the condition number of the mass/stiffness matrix.

For polynomial or Fourier basis, one way of having a better conditioned system, is to subtract to the fine basis $\{\phi_i\}$ the projection on the first few modes $\{h_k\}$. Indeed, these basis are invariant by translations. Thus, imposing

$$\forall i,\ \forall k < N,\ < \phi_i, h_k >_{\Omega_{int}} \text{ small}$$

guarantees

$$\forall i,\ \forall k < N,\ < \phi_i \circ F(\gamma), h_k >_{\Omega_{int}(\gamma)} \text{ small .}$$

Another option is to use constrained optimization in the online phase, as done in [9].

3.6.3 Computational details

Just as for the periodic setting, at each time step, we are looking simultaneously for the coordinates on the reduced space $X^N(\gamma)$ and for the correct calibrating parameter γ. We have chosen to use the same iterative process as the one described in section 3.3. The only difference is the discretization space.

Let γ^n be the calibrating parameter at time step n. We need to evaluate the following quantities, for γ in a small neighborhood of γ^n:

$$\begin{cases} \forall i, j,\ \int_\Omega \phi_i \circ F(\gamma^n)\ \phi_j \circ F(\gamma) \\ \forall k, l,\ \int_\Omega h_k h_l \\ \forall i, k,\ \int_\Omega \phi_i \circ F(\gamma)\ h_k \end{cases} \tag{3.62}$$

As we have chosen the $\{\phi_i\}$ to have zero trace on $\partial\Omega_{int}$, the first term will be handled just as in the periodic case as it only involves relative translations. The second term will not cause any problem, as the basis $\{h_k\}$ is independent of the translation parameter and is as a premise supposed to be of small cardinality. The tricky part is the matching term, as it does not only involve the relative translation. We detail in the next section a few options available.

3.6.3.1 Coarse Finite element space

We will present one possible solution, when $\{h_k\}$ is a coarse Finite Element space, on a regular grid. Let

- $\mathcal{K} = \{k,\ supp(h_k) \cap supp(\phi_i) \text{ not empty }\}$.
- $\mathcal{K}(\gamma) = \{k,\ supp(h_k) \cap supp(\phi_i) \circ F(\gamma) \text{ not empty }\}$.
- l the Lebesgue measure of the support of the ϕ_i
- Δx the characteristic size of the coarse mesh.

Online, we need to compute

$$\forall k \in \mathcal{K}(\gamma),\ \int_{\Omega_{int}(\gamma)} \phi_i \circ F(\gamma) h_k$$

Let $\gamma_0 = \lfloor \frac{\gamma}{\Delta x} \rfloor$ and $\delta = \gamma - \gamma_0$. Using the fact that the mesh is uniform, it is clear that the previous quantity is equal to:

$$\forall k \in \mathcal{K},\ \int_{\Omega_{int}} \phi_i h_k \circ F(\delta).$$

One just needs to be careful while assembling the global system. An interpolation such as the one proposed in section 3.4.4 concludes: let $\{\delta_p\}_p$ be some discretization of $[0, \Delta x]$ and compute offline

$$\forall p,\ \forall k \in \mathcal{K},\ \int_{\Omega_{int}} \phi_i h_k \circ F(\delta_p). \tag{3.63}$$

Interpolate for estimations of $\int_{\Omega_{int}} \phi_i h_k \circ F(\delta x)$ for $\delta x \leq \Delta x$.

3.6.3.2 Polynomial/Fourier basis

Suppose that we have a coarse basis, globally defined on Ω, that has an explicit expression through calibration. More precisely, suppose that

$$\forall \gamma,\ h_k \circ F(\gamma) \in \text{ span } \{h_p(\cdot)\}_p \tag{3.64}$$

and that the coordinates on the basis have an explicit form. By a simple change of variable, we have:

$$\int_{\Omega_{int}(\gamma)} \phi_i(\cdot - \gamma) h_k(\cdot) = \int_{\Omega_{int}} \phi_i(\cdot) h_k(\cdot + \gamma). \tag{3.65}$$

This method works for instance for a global polynomial or Fourier basis.

3.6.3.3 A third idea

This method presents similarities with the adaptive empirical projection method presented in [123]. We focus on the 'coarse solution' manifold : $\mathcal{M}_0$ defined on equation (3.61). Construct the basis of the restrictions of the functions in $\mathcal{M}_0$ to subdomains of measure mes (Ω_{int}) :

$$\Xi := \{v \circ F(\gamma)|_{\Omega_{int}},\ v \in \mathcal{M}_0,\ \gamma \in \Omega\}.$$

A natural discretization, is to take the γs as multiple of δx^{fine}, the characteristic size of the fine mesh on Ω_{int}. Ξ is then of cardinality $\text{card}(\mathcal{M}_0) * \text{mes}(\Omega)/\delta x^{fine}$. We use a compression algorithm on Ξ to get a reduced basis of small cardinality, $\{\theta_j\}_j$ with support Ω_{int}.

We then learn offline the following scalar products :

$$\forall i, j,\ < \phi_i, \theta_j >_{\Omega_{int}} \tag{3.66}$$

Online, we need to estimate terms such as :

$$\int_{\Omega_{int}} \phi_i h_k \circ F(\gamma)$$

We can use an EIM type method to express $h_k \circ F(\gamma)$ on the basis $\{\theta_j\}_j$.

3.6.4 Numerical results

What is presented here has to be taken as a proof of concept. More involved numerical simulations should be performed. We have chosen the following test case:

- a gaussian initial condition
- homogeneous Neumann boundary condition

- a small viscosity such that the classical ROM fails, but big enough so that our centered reduced scheme gives decent results, see section 3.4.

in Figure 3.12, we display the first offline steps of the method. The left Figure shows the original snapshots. We use an offline calibration process, such as the one described in section 3.4.2 to obtain the calibrated solution manifold displayed on the central picture. The choice for the

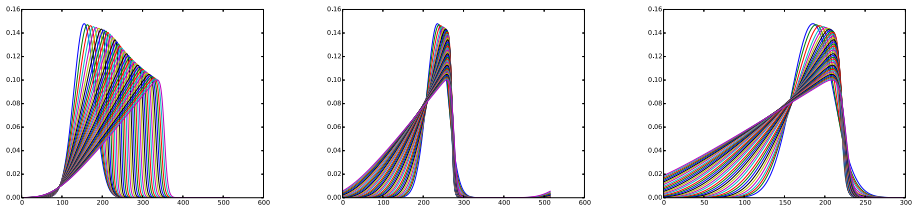


Figure 3.12: Snapshot sets at different steps of the offline stage. From left to right: snapshots with no calibration; centered snapshots; truncated snapshots

truncation window should be studied as it has a big influence on the condition number and overall feasibility of the method. Here, it has been done empirically. The resulting snapshots are presented on the right picture of Figure 3.12.

Following remark 23, we taylor an indicator function, as presented in Figure 3.13. We have

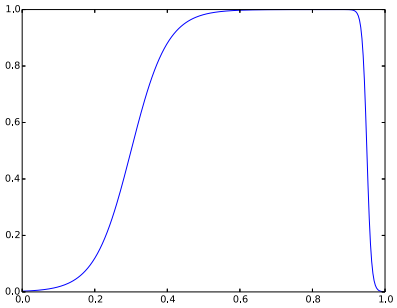


Figure 3.13: The indicator function, that depends strongly on the solution manifold at hand

chosen this indicator function to be non symmetric, as our snapshot set is not symmetric. Taking it stiff on the right makes sense to capture the shock. Taking it smooth on the left makes sense, as we want the complement to be well represented by a low dimensional space. These complements are members of the manifold $\mathcal{M}_0$ introduced above. Some snapshots are represented in Figure 3.14. We have reduced the propagation of a shock, in a non periodic setting into two parts: one stiff part that can be handled exactly as in the previous section. The other part is the transport of a smooth quantity, which should be doable using a coarse basis. Indeed, what is being transported is much more regular that what we had initially.

We present numerical evidence backing this remark in Figure 3.15. It compares the projection onto a non calibrated basis of cardinality 15, with the projection onto the union of a calibrated

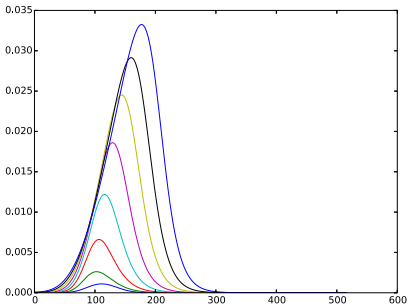


Figure 3.14: Snapshots in the resulting "coarse" solution manifold $\mathcal{M}_0$

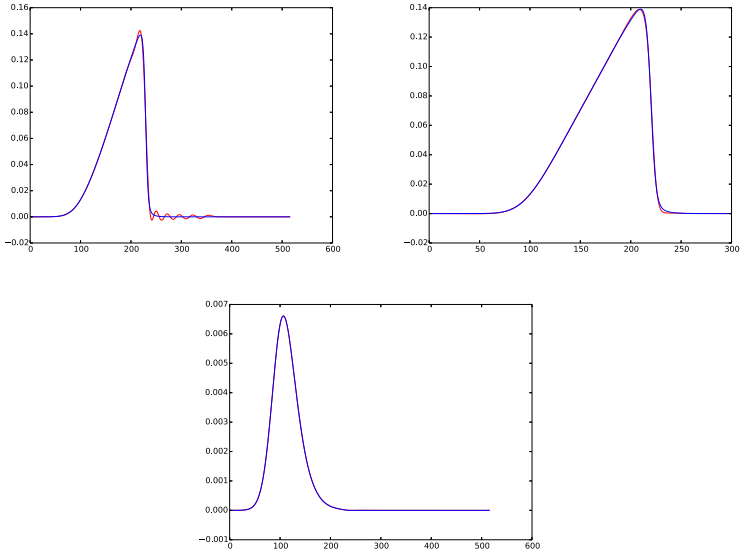


Figure 3.15: Top left : projection of one snapshot, with no calibration, on a basis of cardinality 15; Top right : projection of the truncated snapshot onto an adapted basis; Bottom : projection of the complement on a small Fourier basis

basis of cardinality 5 and a coarse global basis of cardinality 10. We mention one last argument. To obtain the same accuracy as the calibrated $5 + 10$ basis, the uncalibrated case requires 40 Fourier modes.

This concludes this small section on an extension of calibration to non periodic settings. The most important thing left to do is to conduct numerical investigations to make sure that the conditioning problem discussed in section 3.6.2 can be overcome. The different offline-online decomposition strategies should also be assessed.

3.7 Calibrating step

In this section, we give some insight on how to chose the calibration parameters in the offline stage. More precisely, say we start with some raw solution manifold $\mathcal{M}$, and suppose that some a priori knowledge of the problem has given us some family $\mathcal{F}$ of transformations. As usual, we are looking for:

$$F : \left\{ \begin{array}{ccc} [0,T] \times \mathcal{D} & \rightarrow & \mathcal{F} \\ t, \mu & \mapsto & F_{t;\mu} \end{array} \right.$$

so that the calibrated solution manifold is better behaved. For most of the examples we have studied (and we will study) in this thesis, an a priori knowledge is enough to select the $F_{t;\mu}$ offline. In this section, we handle the situation when they are out of reach and propose an alternative, generic algorithm.

Let Ξ be a representative snapshot set of $\mathcal{M}$ of cardinal say N^{snap}. We introduce the problematic by presenting in Algorithm 4 a simple greedy algorithm.

Data: Representative snapshot set Ξ of $\mathcal{M}_{\mathcal{D}}$
Result: Calibration parameters $\{F_v, \text{ for } v \in \Xi\}$
$\mathbb{V}_0 :=$ span u_0;
k = 1;
while $k < N^{snap}$ **do**
  $\forall F \in \mathcal{F},\ I(F) := \|u_k \circ F - \Pi_{\mathbb{V}_{k-1}}(u_k \circ F)\|$;
  $F_k \leftarrow \underset{F \in \mathcal{F}}{\text{argmin}}\ I(F)$;
  $\mathbb{V}_k :=$ span $(\mathbb{V}_{k-1} + u_k \circ F_k)$;
  $k \leftarrow k + 1$;
end

Algorithm 4: Finding offline calibration parameters

3.7.1 Optimal method

Here, we rather use ideas close to the optimal snapshot location developed in [81]. Instead of a greedy algorithm, the resulting basis is optimal, in a sense that will be made clear later. As in the description of the POD method, in the introductive chapter 1, we start by defining the correlation operator:

$$R_{\mathcal{F}} : \{F_p\}_p \in \mathcal{F}^{N^{snap}} \rightarrow \left\{ \begin{array}{ccc} X & \rightarrow & X \\ \psi & \mapsto & \sum_{k=1}^{N^{snap}} < u_k \circ F_k, \psi > u_k \circ F_k. \end{array} \right.$$

This operator is well defined, and we denote $\{\lambda_i(\{F_p\}_p)\}_i$ the eigenvalue set of $R(\{F_p\}_p)$, for each $\{F_p\}_p \in \mathcal{F}^{N_{snap}}$ and by $\{\phi_i(\{F_p\}_p)\}_i$ the corresponding eigen vector/ eigen function set. The existence of the previous quantities for general scalar products is discussed in chapter 1.

We know that the set of λs is related to the average approximation error. More precisely, we know that:

$$\forall \{F_p\}_p \in \mathcal{F}^{N_{snap}},\ \forall M \in \mathbb{N},\ \sum_{i>M} \lambda_i(\{F_p\}_p) = \sum_{k=1}^{N_{snap}} \left\| u_k \circ F_k - \sum_{i=1}^{M} < u_k \circ F_k, \phi_i(\{F_p\}_p) > \phi_i(\{F_p\}_p) \right\|^2 .$$

We are starting get a sense in which we are looking for an optimal calibration: we look for the set of $\{F_p\}_p$ that minimizes the average projection error.

Let $M \in \mathbb{N}$. Let J_M be the following functional:

$$J_M : \left\{ \begin{array}{lll} \mathcal{F}^{N^{snap}} \times \mathbb{R}^{N^{snap}} \times X^{N^{snap}} & \rightarrow & \mathbb{R} \\ \{F_p\}_p, \{\lambda_p\}_p, \{\phi_p\}_p & \mapsto & \sum_{i>M} \lambda_i \end{array} \right. \tag{3.67}$$

We are trying to minimise J_M, subject to:

$$\left\{ \begin{array}{lll} \forall i,\ (R_{\mathcal{F}}(\{F_p\}_p) - \lambda_i)\phi_i & = & 0 \\ \forall i,\ (1 - \|\phi_i\|^2) & = & 0 \end{array} \right. \tag{3.68}$$

We can show the existence of a minimzer. We will only sketch the proof and refer to [81] for a complete description. Let $\left(\{F_p^n\}_p, \{\lambda_p^n\}_p, \{\phi_p^n\}_p\right)_n$ be a minimizing sequence in $\mathcal{F}^{N^{snap}} \times \mathbb{R}^{N^{snap}} \times X^{N^{snap}}$. Suppose that $\mathcal{F}$ is compact. This will be the case in all the examples we will be using. For instance, in the translation case in dimension d, $\mathcal{F}$ is a bounded domain in $\mathbb{R}^d$. We can extract a converging subsequence in $\mathcal{F}^{N^{snap}}$. For simplicity, we still denote $(\{F_p^n\}_p)_n$ this subsequence. Denote $\{F_p^*\}_p$ the limit.

We can conclude using arguments given in [77]. The sequence of operators $(R_{\mathcal{F}}(\{F_p^n\}_p))_n$ are close in a sense which ensures convergence of eigenvalues and eigenvectors towards the eigenvalues and eigenvectors of the limit operator $R_{\mathcal{F}}(\{F_p^*\}_p)$. This concludes on the existence of a minimizer of $R_{\mathcal{F}}$ in $\mathcal{F}^{N^{snap}}$.

How do we approach this minimizer ? Define a modified objective equation

$$\tilde{J}_M : \left\{ \begin{array}{lll} \mathcal{F}^{Nsnap} & \rightarrow & \mathbb{R} \\ \{F_p\}_p & \mapsto & \sum_{i>M} \lambda_i(\{F_p\}_p) \end{array} \right. \tag{3.69}$$

This problem is of course equivalent to problem (3.67), (3.68). We want to compute the Gateaux derivative of J_M. Let $j \in [1, N^{snap}]$, we start with a formal computation of $\frac{\partial J_M}{\partial F_j}$. The arguments for the existence of these derivatives are given below. We have:

$$\frac{\partial J_M}{\partial F_j} = \sum_{i>M} \frac{\partial \lambda_i(\{F_p\}_p)}{\partial F_j}.$$

We use the fact that the $\lambda(\{F_p\}_p)$ are eigenvalues of the operator $R_{\mathcal{F}}(\{F_p\}_p)$ and the product rule:

$$\forall i,\ \forall p,\ \left(\frac{\partial R_{\mathcal{F}}}{\partial F_j} - \frac{\partial \lambda_i}{\partial F_j}\right)\phi_i(\{F_p\}_p) + (R_{\mathcal{F}}(\{F_p\}_p) - \lambda_i(\{F_p\}_p))\frac{\partial \phi_i}{\partial F_j} = 0$$

We take the scalar product of the previous quantity with $\phi_i(\{F_p\}_p)$:

$$\forall i,\ \forall p,\ < \left(\frac{\partial R_{\mathcal{F}}}{\partial F_j} - \frac{\partial \lambda_i}{\partial F_j}\right)\phi_i(\{F_p\}_p), \phi_i(\{F_p\}_p) > + < (R_{\mathcal{F}}(\{F_p\}_p) - \lambda_i(\{F_p\}_p))\frac{\partial \phi_i}{\partial F_j}, \phi_i(\{F_p\}_p) >= 0.$$

As $R_{\mathcal{F}}$ is an autoadjoint operator, and $(\phi_i(\{F_p\}_p), \lambda_i(\{F_p\}_p))$ is an associated eigenfunction/eigenvalue pair, the second term cancels. We thus have

$$\forall i,\ \forall p,\ < \left(\frac{\partial R_{\mathcal{F}}}{\partial F_j}\right)\phi_i(\{F_p\}_p), \phi_i(\{F_p\}_p) >= \frac{\partial \lambda_i}{\partial F_j} \tag{3.70}$$

We use the definition of $R_{\mathcal{F}}$:

$$\frac{\partial R_{\mathcal{F}}}{\partial F_j} : \left\{ \begin{array}{lll} X & \rightarrow & X \\ \psi & \mapsto & < \frac{\partial (u_j \circ F_j)}{\partial F_j}, \Psi > u_j \circ F + < u_j \circ F, \Psi > \frac{\partial (u_j \circ F_j)}{\partial F_j} \end{array} \right.$$

Plugging this into (3.70), we have:

$$\forall i,\ \forall p,\ \frac{\partial \lambda_i}{\partial F_j} = 2 < \frac{\partial (u_j \circ F_j)}{\partial F_j}, \phi_i(\{F_p\}_p) >< u_j \circ F_j, \phi_i(\{F_p\}_p) >$$

We can then express the gradient of our functional J as

$$\nabla J = \sum_{i>M} 2 < \frac{\partial (u_j \circ F_j)}{\partial F_j}, \phi_i(\{F_p\}_p) >< u_j \circ F_j, \phi_i(\{F_p\}_p) > \tag{3.71}$$

All of the previous derivatives with respect components of $F \in \mathcal{F}$ exist if and only if the solutions in the original solution manifold have a smooth dependence on F.

We will explicit it in the case of translations, in the periodic one dimensional case. $\mathcal{F}$ is the family of translations, with parameter $\gamma \in \Omega$.

$$\frac{\partial u \circ F(\gamma)}{\partial \gamma} = -u_x \circ F(\gamma)$$

So the previous derivation is rigorous if and only if the original solution manifold $\mathcal{M}$ is embedded in $C^1(\Omega)$. For discontinuous solution, one can use a similar algorithm, but restricting the computations to Ω_d a subdomain of Ω away from the discontinuity. This idea is used in chapter 4 of this manuscript.

3.7.2 Algorithm

We are looking for a minimum of the functional J_M. A standard quasi Newton algorithm goes a follows:

Data: Representative snapshot set Ξ of $\mathcal{M}_{\mathcal{D}}$
Result: Calibration parameters F_k for $u_k \in \Xi$
Let $\{F_k^0\}_k \in \mathcal{F}^{N^{snap}}$ be an initial guess ;
$n := 0$;
repeat
  Compute $\lambda(\{F_k^n\})$ and $\phi(\{F_k^n\})$;
  Compute $\nabla J(\{F_k^n\})$ using equation (3.71);
  Approximate the Hessian B, using BFGS for instance ;
  $\delta F := -B^{-1} \nabla J(\{F_k^n\})$;
  $\forall k,\ F_{k+1}^{n+1} \leftarrow$ line search $(F_k^n, \delta F)$;
  $n \leftarrow n + 1$;
until some computational/accuracy threshold;

Algorithm 5: Finding offline calibration parameters

This method is computationally very expensive. We propose a method to mitigate this issue. It is a divide a conquer type of method. We first decompose our snapshot set Ξ into smaller batches $\{\Xi_p\}$. We can then perform the previous algorithm onto each batch independently. The fusion step can be done using the same algorithm.

Data: Representative snapshot set Ξ of $\mathcal{M}_{\mathcal{D}}$
Result: Calibration parameters F_k for $u_k \in \Xi$
Let $\{\Xi_p\}$ be some decomposition of the original Ξ ;
$\forall p,\ \{F_{p,k}\}_k \leftarrow$ Algorithm 5($\{\Xi_p\}$);
Let $\phi_0^p(\{F_{p,k}\}_k)$ be the first POD basis of the pth batch using the calibration parameter $\{F_{k,p}\}_k$;
$\{G_p\}_p \leftarrow$ Algorithm 5 $\left(\{\phi_0^p(\{F_{p,k})\}_k\}_p\right)$;
$\forall p,\ \forall k,\ F^*_{p,k} \leftarrow G_p \circ F_{p,k}$;

Algorithm 6: Finding offline calibration parameters

An iterative version with several layers follows easily. We could also chose other type of representants of the 'batch' POD basis. Another computationally cheaper version of algorithm 5 is discussed in section 3.9.

Remark 24 *This method can seem scary, as it involves many mesh interpolations at each iteration. The computational cost set aside, some may worry about the accuracy of the resulting method. The answer to this, is that we are trying to compress the information contained in $\mathcal{M}$. That is, even if one specific calibration loses some detail, it may still be in the final reduced basis. Also, as we are in the offline setting, we can afford to interpolate as precisely as we want. A final argument is that we can, instead of minimizing J, we can accept less precise calibration that preserves more informations, without degrading the resulting calibrated manifold too much.*

Of course, we do not have any theoretical convergence results on the quasi-newton algorithm for this specific functional. A possible issue is then to get stuck at a local minimum. This has happened to us when applying the optimization algorithm on the dataset used in section 3.4 above. The output of the algorithm is presented in Figure 3.16. We have reached a local minimum where the shock fronts have clustered at two distinct locations. Among the many ways

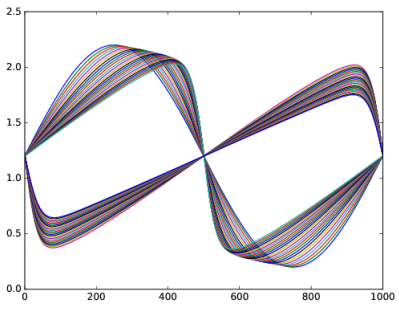


Figure 3.16: On possible ouput of an offline calibration procedure such as algorithm 5

of handling this situation, we have chosen one coming from the machine learning community: a clustering algorithm, and more precisely a hierarchical clustering algorithm. This empiric approach is easy to implement and seems adapted for our particular need. The output for the dataset presented in Figure 3.16 is presented in Figure 3.17. We will briefly describe how to read this so-called dendogram and refer to [5] for more details on this method. This algorithm's purpose is to decompose sets into clusters. The starting point is some user-defined metric on the set. To obtain the previous Figure, we have simply chosen the norm of the underlying Hilbert

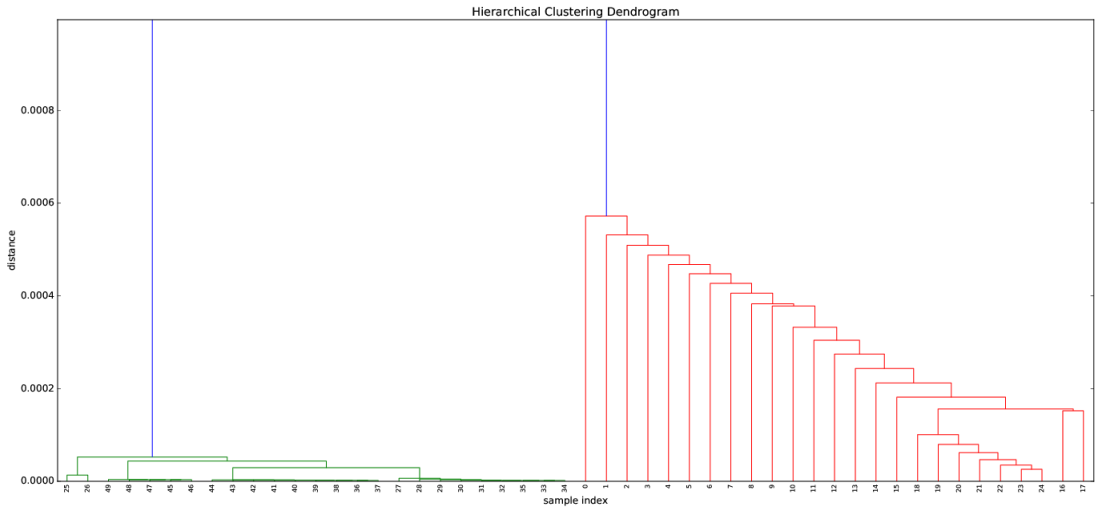


Figure 3.17: The output of the hierarchical clustering algorithm, for the snapshot set presented Figure 3.16

space. Then, the lines' length represent the distance between two members of the set. Here, green lines correspond to the cluster of the snapshots of the end of the simulations. The associated distance are really small as the solution manifold is almost a constant propagating front. The beginning of the simulations, when the front has not yet formed, shows a bigger 'inner cluster' distance, in red in Figure 3.17. The distance between members of different clusters is much bigger than any 'inner-cluster' distance, and corresponds to the blue line in Figure 3.16.

This algorithm should be able to spot any situations such as the one presented in Figure 3.16. One option is then to split the dataset into two disjoints sets: the outputed clusters and then to perform a compression algorithm on each subset. Finally, just as in algorithm 6, we can find a good 'transformation' between representative basis of the two subsets. The global offline calibration parameter can be taken as a composition of a global calibration (to make the clusters match) and of a relative calibration (inter cluster).

This method for offline calibration has potential. We now have a way of calibrating sets for which the calibration parameters are not obvious. We have raised concerns on the associated computational cost but have proposed small improvements to mitigate the issue. What is left to do is to try this method on more challenging numerical examples, to assess its performances.

3.8 Two dimensional example

We briefly present a two dimensional extension for the calibration method. This section does not introduce any major novelty compared to what has been done before. It serves two purposes. The first one is to illustrate the fact that the equivariant property, see section 3.2.3, which was stated as necessary requirement in [19], can be replaced by a less stringent hypothesis. In other words, the calibrated differential operator can be different from the original one. The second

purpose here is to interpret our calibration procedure as a reduced order modeling version of mesh adaptation.

Remark 25 *For a true, challenging, two dimensional example, we refer to chapter 4.*

We choose the following favorable setting: we are given some parabolic equation in some two dimensional domain Ω with periodic boundary conditions such that the solution manifold $\mathcal{M}_{\mathcal{D}}$ benefits from translation. $\mathcal{F}$ is thus chosen as a subspace of two dimensional translations. We skip the description of the offline section, and assume right away that we have a calibrated basis:

$$\{\phi_i,\ i = 1 \dots N_{red}\}$$

such that:

$$\forall u(\cdot, t; \mu) \in \mathcal{M}_{\mathcal{D}},\ \exists F_{t;\mu} \in \mathcal{F} \text{ s.t } \ u(\cdot, t; \mu) \in \text{ span } \{\phi_i \cdot F_{t;\mu}\}.$$

3.8.1 On the equivariance of $\mathcal{L}_\mu$

The objective of this section is to show that equivariance is not a necessary property for the application of the calibration procedure. For this, we have chosen a family of mapping $\mathcal{F}$ for which the parameter dependency is of the following form:

$$F_{t;\mu} : \begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} x - \theta_{t;\mu}(y) \\ y \end{pmatrix}. \tag{3.72}$$

The underlying idea is to be able to handle transformations around the affine mapping:

$$F_\mu : \begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} x - \mu y \\ y \end{pmatrix}$$

which transforms the lines $x = cte$ into $x - \mu y = cte$. For this restricted class of mappings, the determinant of the Jacobian is constant over Ω, and equal to one. We also have an explicit form of the inverse mappings:

$$\forall \mu \in \mathcal{D},\ F_{t;\mu}^{-1} : \begin{pmatrix} u \\ v \end{pmatrix} \to \begin{pmatrix} u + \theta_{t;\mu}(v) \\ v \end{pmatrix}.$$

Suppose that the differential operator $\mathcal{L}_\mu$ involves a Laplace operator. It is clear, with our choice of mappings, that $\Delta(u \circ F_{t;\mu}) \neq (\Delta u) \circ F_{t;\mu}$ (say in the sense of distributions). Nevertheless, we will show that the calibration procedure can still be applied.

A standard online section requires the efficient computation of the variational form. We focus on two standard terms:

$$\begin{cases} \int_\Omega \phi_i(F_{t;\mu}(x, y))\ \phi_j(F_{t';\mu}(x, y))dxdy \\ \int_\Omega \nabla(\phi_i \circ F_\mu)(x, y) \cdot \nabla(\phi_j \circ F_{t';\mu})(x, y)dxdy. \end{cases}$$

L^2 scalar products are directly handled by a simple change of variables:

$$\int_\Omega \phi_i(F_{t;\mu}(x, y))\ \phi_j(F_{t';\mu}(x, y))dxdy = \int_\Omega \phi_i(u + \theta_{t';\mu}(v) - \theta_{t;\mu}(v), v)\ \phi_j(u, v)dudv. \tag{3.73}$$

As for all the examples in this chapter, we suppose the smoothness of $t \mapsto F_{t;\mu}$. Thus $\theta_{t';\mu}(v) - \theta_{t;\mu}(v)$ is close to 0 and these terms can be estimated using an interpolation method, see section 3.4.4.

There is a little bit more work for the H^1 scalar products. For clarity, we will denote $\frac{\partial}{\partial 1}$ and $\frac{\partial}{\partial 2}$ the derivatives with respect to first and second variables. With our specific form of mapping, we know that:

$$\frac{\partial(\phi_i \circ F_{t;\mu})}{\partial x} = \frac{\partial \phi_i}{\partial 1} \circ F_{t;\mu}$$

and that:

$$\frac{\partial(\phi_i \circ F_{t;\mu})}{\partial y} = \frac{\partial \phi_i}{\partial 2} \circ F_{t;\mu} - \theta'_{t;\mu} \frac{\partial \phi_i}{\partial 1} \circ F_{t;\mu}.$$

The evaluation of the H^1 scalar product between two consecutive time steps will involve the computation of terms such as:

$$\begin{aligned}
\int_\Omega \tfrac{\partial(\phi_i \circ F_{t;\mu})}{\partial y}(x,y) \tfrac{\partial(\phi_j \circ F_{t';\mu})}{\partial y}(x,y) &= \int_\Omega \tfrac{\partial \phi_i}{\partial 2}(x - \theta_{t;\mu}(y), y) \tfrac{\partial \phi_j}{\partial 2}(x - \theta_{t';\mu}(y), y) \\
&- \int_\Omega \theta'_{t;\mu}(y) \tfrac{\partial \phi_i}{\partial 1}(x - \theta_{t;\mu}(y), y) \tfrac{\partial \phi_j}{\partial 2}(x - \theta_{t';\mu}(y), y) \\
&- \int_\Omega \tfrac{\partial \phi_i}{\partial 2}(x - \theta_{t;\mu}(y), y) \theta'_{t';\mu}(y) \tfrac{\partial \phi_j}{\partial 1}(x - \theta_{t';\mu}(y), y) \\
&+ \int_\Omega \theta'_{t;\mu}(y) \tfrac{\partial \phi_i}{\partial 1}(x - \theta_{t;\mu}(y), y) \theta'_{t';\mu}(y) \tfrac{\partial \phi_j}{\partial 1}(x - \theta_{t';\mu}(y), y)
\end{aligned}$$

With the proper change of variables, and because of periodicity, we can express these quantities in terms of relative variations:

$$\begin{aligned}
\int_\Omega \tfrac{\partial(\phi_i \circ F_{t;\mu})}{\partial y}(x,y) \tfrac{\partial(\phi_j \circ F_{t';\mu})}{\partial y}(x,y) &= \int_\Omega \tfrac{\partial \phi_i}{\partial 2}(u + \theta_{t';\mu}(v) - \theta_{t;\mu}(v), v) \tfrac{\partial \phi_j}{\partial 2}(u,v) \\
&- \int_\Omega \theta'_{t;\mu}(v) \tfrac{\partial \phi_i}{\partial 1}(u + \theta_{t';\mu}(v) - \theta_{t;\mu}(v), v) \tfrac{\partial \phi_j}{\partial 2}(u,v) \\
&- \int_\Omega \tfrac{\partial \phi_i}{\partial 2}(u + \theta_{t';\mu}(v) - \theta_{t;\mu}(v), v) \theta'_{t';\mu}(v) \tfrac{\partial \phi_j}{\partial 1}(u,v) \\
&+ \int_\Omega \theta'_{t;\mu}(v) \tfrac{\partial \phi_i}{\partial 1}(u + \theta_{t';\mu}(v) - \theta_{t;\mu}(v), v) \theta'_{t';\mu}(v) \tfrac{\partial \phi_j}{\partial 1}(u,v)
\end{aligned}$$

Remark 26 *For these terms to be properly defined, $\forall(\mu,t),\ y \mapsto \theta_{t;\mu}(y)$ needs to be smooth.*

The offline/online decomposition of the global method depends on the offline/online decomposition of θ_μ. A sufficient conditions is for θ_μ to be affine (or close to) affine decomposable, see the description of the EIM section 1.5.2.1.

3.8.2 Calibration seen as mesh adaptation

The idea of this section could have been discussed earlier in this chapter, and is not specific to the two dimensional problem at hand. I have chosen to mention it here, as the current setting provides good and yet easily understandable illustrations. This interpretation of calibration in terms of mesh adaptation was not in the first version of this work. It is indeed not necessary, and it does not provide any additional results. I have nevertheless chosen to add it here because calibration, as presented so far, is not being seen by many as a reasonable option from a numerical point of view. On the other hand, mesh adaptation in finite volume and finite element communities, for similar problems, is considered standard/mandatory. We thus hope to convince skeptical readers of the validity of calibration.

We have chosen to illustrate this simple setting with a two dimensional version of Burgers equation. For simplicity, we work with a superposition of 1D problem. More precisely, we are solving for the scalar function $u \in H^1(\Omega)$ such that:

$$u_t + c(\frac{u^2}{2})_x - \epsilon u_{xx} = 0,$$

where forall μ, $c(\mu)$ is a smooth function over some two dimensional rectangle Ω. We plot in Figure 3.18 some snapshots for various times, and various parameters μ. The initial condition is the same as for the 1D problem, see section 3.4, and is plotted on the top left picture. The top right figure shows a solution for $c(x, y) \mapsto 1 + \alpha y$. Third figure shows a solution for $c(x, y) \mapsto 1 + \beta(y - \bar{y})^2$, where $\bar{y}$ denotes the mid height of the rectangle Ω. This results in solutions presenting various front shapes, and it is exactly fitted for the $F_{t;\mu}$ introduced in equation (3.72).

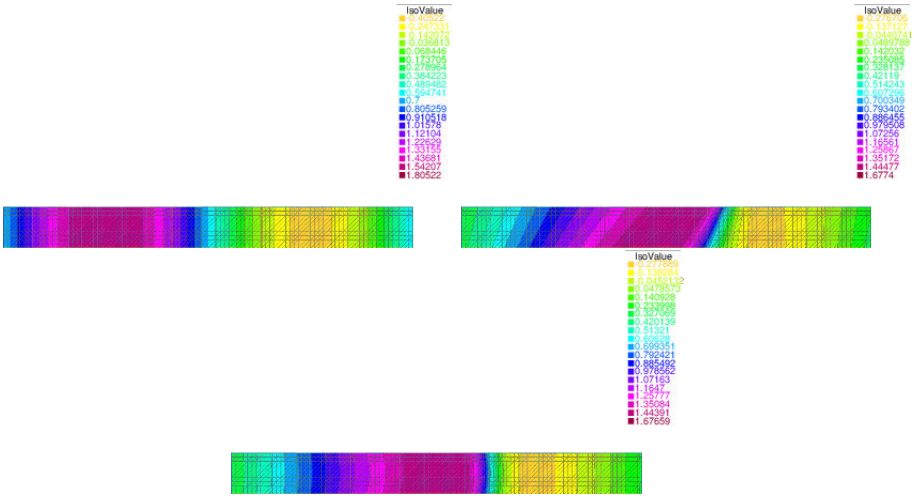


Figure 3.18: Snapshots for various times, and various convection parameters $c(\mu)$

We display in Figure 3.19 one reference mesh on Ω, on which the truth, calibrated, solutions $\tilde{\mathcal{M}}$ are defined. If the calibration procedure is successfully performed, the shock of such calibrated solutions should always be located in the refined portion of the mesh. Now, what happens in

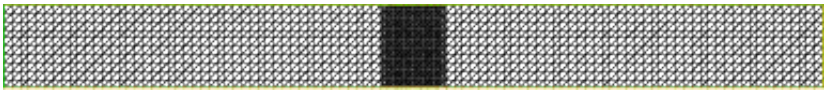


Figure 3.19: Reference mesh

the online phase ? Say we are at time t^n, with a well chosen calibrating function F^n and the corresponding calibrated solution $\hat{u}^n$. Equivalently, that means that we have a physical solution $u^n := \hat{u}^n \circ F^n$ defined on an adapted mesh. Meshes for front such as the one presented in Figure 3.18 are presented in Figure 3.20. The iterative algorithm of section 3.3 can be seen as an adaptive mesh procedure, where we are looking for the best mesh to represent u^{n+1}. In FE or FV contexts, this adaptation is done using some inter cell computations, as no more information is available, see for instance [57]. In our reducing context, we do have some additional pieces of information. More precisely, the iterative algorithm selects a small variation around the identity, such that the modified basis (or mesh) best represents u^{n+1}. As already stated, because our calibration functions in $\mathcal{F}$ are easily invertible, all the computations can be made on the same reference mesh (for instance the one depicted Figure 3.19) whatever the parameter/time considered. For ideas on the numerical errors due to this process, we refer to remark 21. We want to insist on one

Figure 3.20: Meshes on which the physical solutions $u^n = \hat{u}^n \circ F^n$ are defined

point: all the critics made on the calibration method can be exactly transposed to the standard mesh interpolation methods. We even mitigate some of the issues:

- only a limited number of mesh interpolation have to be computed, and all the computations are done in the offline phase (and thus can be done as precisely as we want, with methods as costly as we want)
- the underlying interpolation error in the online phase is controlled: for this, we refer to Figure 3.7 and the associated discussion
- the counterpart to the error indicator used in the FE/FV context, is here perfectly adapted to the reduced framework, and can be assessed in an offline phase

This will again be discussed in the concluding chapter of this manuscript. One last remark on a related subject. It seems like the calibration method could be applied to problems with moving boundaries or moving interfaces. Such problems have received some attention in the reducing community recently, see for instance [106]. It seems like the raw solution manifolds for such problems would suffer the same n-width issue as propagating fronts. I believe that these problems could be treated as the two dimensional Burgers problem with varying front shapes. This idea has not been further investigated in this manuscript.

3.9 Rotating obstacle

The example we build in this section is a little different from the ones we have seen so far, since the calibration process is not a translation. It as already been mentioned in the introductive chapter, and in chapter 2. The setting is a rotating obstacle, fixed inside Ω. The mesh used for the computation of the fine/truth solutions is presented in Figure 3.21. We solve the incompressible Navier-Stokes equation in Ω. For a certain range of Reynolds number, we know that this particular problem leads to the creation of a Karman vortex street. We also know that the direction of the vortex street is roughly the same as the main inflow direction. Following this remark, we try to construct an example on which standard ROM fails and that would benefit from calibration. We choose as parameter the inflow direction:

$$u^{in}(\cdot, t; \mu) = \begin{pmatrix} cos(\theta(t; \mu)) \\ sin(\theta(t; \mu)) \end{pmatrix}.$$

For sufficiently big parameter range for θ, we expect that the raw solution manifold $\mathcal{M}$ is not an adapted candidate to standard ROM.

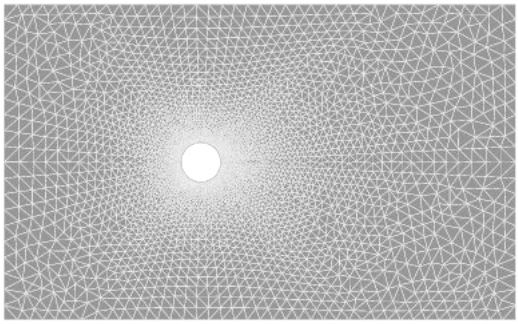


Figure 3.21: Fine mesh on which the offline stage is performed

3.9.1 Offline phase

We propose a more adapted procedure. Decompose the domain Ω into one circular domain around the obstacle Ω_{int} and one outside domain Ω_0. Just as in section 3.6, that means that instead of considering the global raw solution manifold $\mathcal{M}$, we study two disjoints manifolds:

$$\mathcal{M}_{int} := \{u(\cdot, t; \mu)|_{\Omega_{int}},\ t \in [0, T],\ \mu \in \mathcal{D}\}$$

and

$$\mathcal{M}_0 := \{u(\cdot, t; \mu)|_{\Omega_0},\ t \in [0, T],\ \mu \in \mathcal{D}\}\,.$$

$\mathcal{M}_0$ is far away from the obstacle, and for a 'proper' choice of Ω_{int}, it should deal with less complex structures, and not be as heavily direction dependent, as the initial problem. This translates into a better behaved Kolmogorov n-width. We use the following calibration family for $\mathcal{M}_{int}$:

$$F_{t;\mu} : \left\{ \begin{array}{ccc} \Omega & \to & \Omega \\ \begin{pmatrix} x \\ y \end{pmatrix} & \mapsto & \begin{pmatrix} cos(\theta(t;\mu)) & -sin(\theta(t;\mu)) \\ sin(\theta(t;\mu)) & cos(\theta(t;\mu)) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \end{array} \right.$$

Note that even if the problem is two dimensional, the family $\mathcal{F}$ is a one parameter family. The definition of the calibration manifold $\mathcal{M}_{int,\mathcal{F},\mathcal{D}}$ naturally follows:

$$\mathcal{M}_{int,\mathcal{F},\mathcal{D}} := \left\{u(F_{t;\mu}^{-1}(\cdot, t; \mu),\ t \in [0, T],\ \mu \in \mathcal{D}\right\}.$$

We expect this manifold to have a smaller n-wdith that the one of the original problem. This concludes the preliminary analysis. We have the proper candidates for ROM procedure, $\mathcal{M}_0$ and $\mathcal{M}_{int,\mathcal{F},\mathcal{D}}$.

As an introductive example, we have chosen to consider a problem where the only parameter is time. Also, θ is chosen as a linear function with $\theta(0) = -\frac{\pi}{4}$ and $\theta(T) = \frac{\pi}{4}$. As an illustration, we

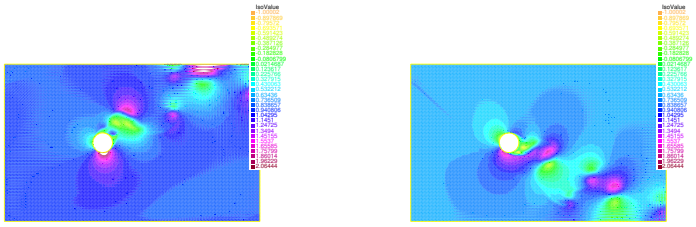


Figure 3.22: Left: A snapshot near the beginning of the simulation; Right: a snapshot near the end of the simulation

present in Figure 3.22 the horizontal velocity component for some $t \approx 0$ and one with $t \approx T$. By inspecting these two snapshots, it appears clearly that standard ROM is not adapted. Indeed, the two solutions are close to orthogonal for all reasonable scalar products. Another way of putting this is that a compression algorithm such as POD or RB would not find much redundancy of information in such a solution manifold.

The offline section continues smoothly. We start by truncating our solutions around the obstacles, on Ω_{int}. The size of the latter is problem dependent, and should be investigated. Our choice of mesh for Ω_{int} is presented in Figure 3.23. The restriction of the solutions of Figure

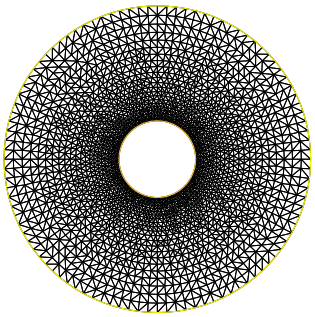


Figure 3.23: The truncated mesh

3.22 on Ω_{int} are presented in Figure 3.24.

We have to calibrate the snapshots offline. For each snapshot $u(\cdot, t^n)$, the parameters F_n can be chosen, as for the other examples, using some a priori knowledge. Here, it is the known inflow direction at time t^n. A more realistic possibility is to use the algorithm proposed in section 3.7. The computational cost issues that were raised are still valid. It is even worse as we are dealing with a two dimensional problem. In order to make the method of Algorithm 5 computationally tractable, we propose a small variation. It relies on the use of filtering. For instance, let η be some smooth indicator function of some small neighborhood of 0. One can smooth any function defined on Ω, say u, by looking at $\bar{u}$ defined as:

$$\bar{u} : x \to (u * \eta)(x) = \int_{\Omega} u(\cdot)\eta(x - \cdot).$$

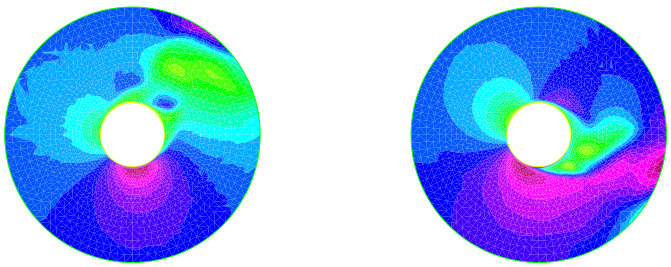


Figure 3.24: Truncated versions of the solutions presented in Figure 3.22

Data: Uncalibrated solution manifold $\mathcal{M}_{\mathcal{D}}$
Result: Calibration parameters $\theta^*(t;\mu),\ t \in [0,T], \mu \in, \mathcal{D}$
Define G_0 some coarse filter and $*$ the standard convolution operator ;
Let $G_0 * \mathcal{M}_0$ be a filtered version of the solution manifold ;
$\theta^0(t;\mu) \leftarrow$ Algorithm5 $(G_0 * \mathcal{M}_0)$;
$\hat{\mathcal{M}}_{\theta^0} := \left\{u(t;\mu) \circ F_{\theta^0(t;\mu)},\ t, \mu\right\}$;
repeat
  Let G_k be a filter finer than G_{k-1} ;
  $\theta^k(t;\mu) \leftarrow$ Algorithm5 $\left(G_k * \hat{\mathcal{M}}_{\theta^{k-1}}\right)$;
  $\hat{\mathcal{M}}_{\theta^k} := \left\{u(t;\mu) \circ F_{\sum_{p=1}^{k} \theta^p(t;\mu)},\ t, \mu\right\}$;
  $k \leftarrow k+1$;
until some accuracy/computational cost condition;
$\theta^*(t;\mu) := \sum_{p=0}^{k} \theta^p$;

Algorithm 7: Finding offline calibration parameters

The idea is that as k increases, the filter are becoming finer. At the same time, algorithm 5 is being run on roughly calibrated solutions. To put it in other words, the range in which to look for better calibration parameters θ in algorithm 5 diminishes when k increases.

To highlight the gain of this preconditioning process, we present on Figure 3.25 three POD modes (2nd, 5th and 10th) for both original (left) and rotated (right) sets. Just as in the $1D$ case, without calibration, the reduced basis cannot reproduce the small scales of the flow, but rather has to deal with the inflow direction. The latter plays the exact same role as the position of the propagating front, in the one dimensional Burgers case.

3.9.2 Online Phase

The complete construction of a reduced scheme is not in the scope of this small section. We refer to chapter 2 for details on how to glue the reduced solution in Ω_{int} to the outside domain Ω_0. We rather focus here on the novelty: finding a 'good' rotation parameter online. It is not easy to find a simple problem that asseses the search for a calibration parameter. The method we have come up with can seem far-fetched, but it will nevertheless help us draw some preliminary conclusions on the feasibility of the method. This method starts with a fine, truth approximation of the solution $\{u(\cdot,t^n)\}_n$. We insist on the fact that in the remaining of this section, the full simulation is known.

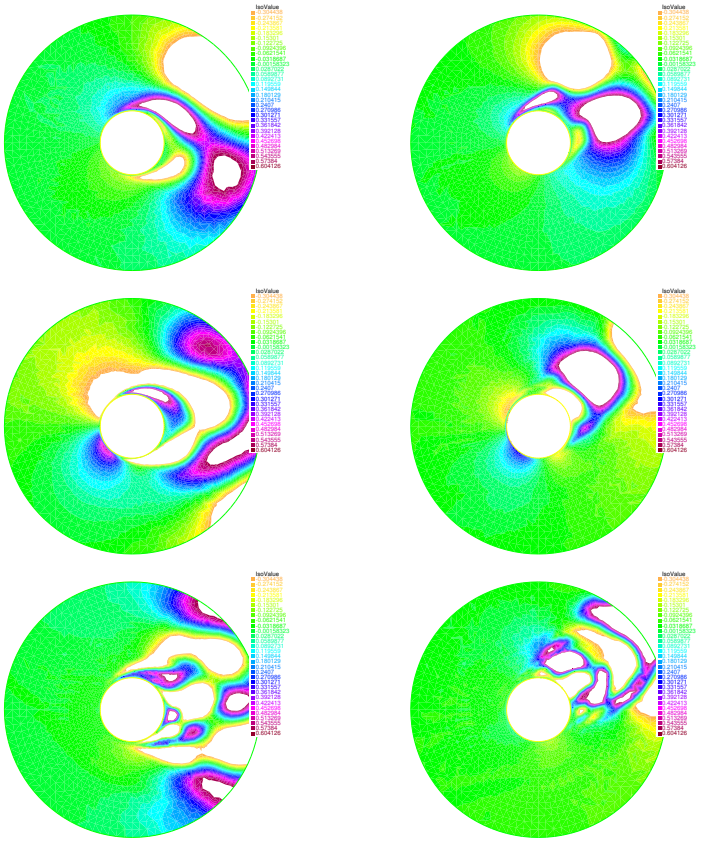


Figure 3.25: Top: 2nd POD mode; Middle: 5th POD mode; Bottom: 10th POD mode

Left: Uncalibrated case; Right: Calibrated case

Let $\{\phi_i,\ i = 1, \dots, N\}_i$ be the calibrated reduced space on which the reduced solutions in Ω_{int} are living. Following the discussion in the offline section, it is assumed to be of small cardinality and to represent well $\mathcal{M}_{int,\mathcal{F},\mathcal{D}}$:

$$\forall u \in \mathcal{M}_{int,\mathcal{D}},\ \exists \theta \in [-\pi, \pi],\quad \text{s.t}\quad u \in\ \text{span}\{\phi_i \circ F(\theta),\ i = 1, \dots, N\}.$$

We need another reduced basis, $\{\psi_j,\ j = 1, \dots, M\}_j$, 'bigger' than $\{\phi_i,\ i = 1, \dots, N\}$. We have $M > N$. We choose it such that it represents well the calibrated solution manifold, with rotations around the identity

$$\forall \hat{u} \in \mathcal{M}_{int,\mathcal{F},\mathcal{D}},\ \forall |\delta\theta| \leq \epsilon,\ \hat{u} \circ F(\delta\theta) \in\ \text{span}\{\psi_j,\ j = 1 \dots, M\}.$$

We reassure the worried reader. This is not suppose to be the description of a real online scheme. It is just an attempt to asses the reconstruction of calibration parameter.

As we know the truth solution $u(\cdot, t^{n+1})$, we can compute its coordinates on the big basis at

time t^n, i.e its coordinates on $\{\psi_j \circ F(\theta^n),\ j = 1, \dots, M\}$:

$$\begin{array}{rll} u(\cdot, t^{n+1}) := & \sum_{j=1}^{M} < u(\cdot, t^{n+1}), \psi_j \circ F(\theta^n) > \psi_j \circ F(\theta^n) & \\ = & & \sum_{j=1}^{M} \beta_j^{n+1} \psi_j \circ F(\theta^n). \end{array} \tag{3.74}$$

θ^n is not the optimal parameter to represent $u(t^{n+1})$. As for Burgers, we assume smoothness of the calibration parameter and look for θ^{n+1} in a small neighborhood of θ^n. This and the assumption on $\{\psi_j,\ j = 1, \dots, M\}$ guarantees that the projection error remains small. In a way, they play the same role as the iterative algorithm of section 3.3.

The second step is to look for a rotation angle denoted θ^{n+1} such that

$$u(\cdot, t^{n+1}) \in \text{ span } \{\phi_i \circ F(\theta^{n+1}),\ i = 1, \dots, N\}. \tag{3.75}$$

Just as in the translation case, we compute the norm of the orthogonal projection of $u(\cdot, t^{n+1})$ on rotated versions of the calibrated basis, trying to minimize the projection error. That is, we are trying to minimize J defined as:

$$J : \left\{ \begin{array}{ccc} [-\pi, \pi] & \to & \mathbb{R} \\ \theta & \mapsto & \left\| u(\cdot, t^{n+1}) - \Pi_{\phi_i \circ F(\theta)} u(\cdot, t^{n+1}) \right\|^2 . \end{array} \right. \tag{3.76}$$

Using the orthogonality of $\{\phi_i, i = 1, \dots, N\}$, and thus the orthogonality of $\{\phi_i \circ F(\theta),\ i = 1, \dots, N\}$, it is equivalent to maximizing $\tilde{J}$:

$$\forall \theta,\ \tilde{J}(\theta) = \sum_{i=1}^{N} \left(< u(\cdot, t^{n+1}), \phi_i \circ F(\theta) > \right)^2 .$$

Using the rotation invariance, $\tilde{J}$ can be expressed as:

$$\forall \theta,\ \tilde{J}(\theta) = \sum_{i=1}^{N} \left(< u(t^{n+1}) \circ F(-\theta^n), \phi_i \circ F(\theta - \theta^n) > \right)^2$$

This is where we use the coordinates of $u(t^{n+1})$ on the basis $\{\psi_j \circ F(\theta^n)\}$:

$$\forall \theta,\ \tilde{J}(\theta) = \sum_{i=1}^{N} \left(\sum_{j=1}^{M} \beta_j^{n+1} < \psi_j, \phi_i \circ F(\theta - \theta^n) > \right)^2$$

We once again use the assumption on the smoothness of $t \to \theta(t)$ and look for θ^{n+1} in a small neighborhood of θ^n. Let $(\delta\theta)_{max}$ be the maximum rotation angle possible between two successive time steps. Let Ξ be some fine discretization of $[-\delta\theta_{max}, \delta\theta_{max}]$. We pre compute the following quantities

$$\forall i \in [1, N]\ , j \in [1, M],\ \forall \delta\theta \in \Xi,\ < \psi_j, \phi_i \circ F(\delta\theta) >_{\Omega_{int}} \tag{3.77}$$

We find the 'optimal' new rotation parameter, at a cheap cost, using interpolation or polynomial fitting, see section 3.4.4.

This specific setting was tested on our initial example with rotating inflow condition. The results are presented in Figure 3.26. In green is plotted the true inflow direction, that changes linearly with time. In blue is the 'cumulative' optimal angle.

With this simple numerical test, we have proven two things. First, that the optimization procedure proposed in (3.76), that was proven to give descent results for viscous Burgers, can extend to more challenging problems. We have also shown that the interpolations of terms such as (3.77) make sense. This is just a first result, and we do not pretend that it concludes on the feasibility of the whole method. It is nevertheless a necessary first step. A computationally viable matching method needs to be implemented before a full reduced scheme can be constructed.

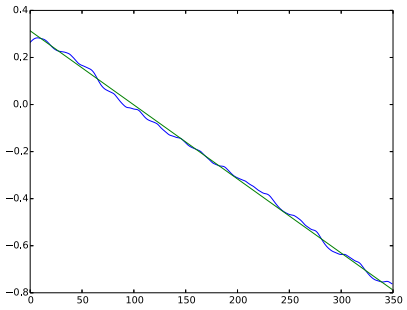


Figure 3.26: Green: the 'true' inflow direction; Blue: the guessed angle

3.10 A posteriori error estimation

In this section, we show that the tools developed in the ROM community for a posteriori estimates directly apply to our calibration framework. We remind the reader that rigorous a posteriori estimates are functions:

$$\Delta : \left\{ \begin{array}{ccccc} [0,T] & \times & \mathcal{D} & \rightarrow & \mathbb{R} \\ (t^k, \mu) & & & \mapsto & \Delta(t^k;\mu) \end{array} \right. \tag{3.78}$$

that satisfy an inequality such as:

$$\forall t^k,\ \forall \mu \in \mathcal{D},\ \|u(t^k,\mu) - u^k(\mu)\| \leq \Delta(t^k;\mu), \tag{3.79}$$

where u is the truth approximation solution, and u^k is the reduced solution at time t^k.

The common step, when trying to construct Δ, whatever the equation considered, is to find a relation between the norm of the error (that we want to estimate), and the norm of the residual. It has been done for a simple model elliptic equation in the introductive chapter, section 1.4.3.1. More generally, it is standard for coercive elliptic PDEs, see for instance [65]. It has also been done for the unsteady viscous Burgers in [101]. We will thus focus here on the computation of the norm of the residual.

We work with a general time dependent equation, say (3.27). For simplicity, we choose an explicit time discretization. The residual in that setting is defined as:

$$r : \left\{ \begin{array}{ccc} X & \rightarrow & \mathbb{R} \\ v & \mapsto & < \frac{u^{n+1}-u^n}{dt}, v > + \mathcal{L}(u^n;\mu)(v) \end{array} \right. \tag{3.80}$$

Let $\hat{e}$ be the Riesz representant in X of r, i.e

$$\forall v \in X,\ < \hat{e}, v >= r(v).$$

$\hat{e}$ is such that $\|\hat{e}\|_X = \|r\|_{X'}$. For all n, let $g(u^n;\mu) \in X$ such that

$$\forall v \in X\ \ < g(u^n;\mu), v >_X = \mathcal{L}(u^n;\mu)(v)$$

We know that:

$$\|\hat{e}\|_X = \left\| \frac{u^{n+1}-u^n}{dt} + g(u^n;\mu) \right\|_X .$$

We can sum up what we have so far. To compute the norm of the residual, we need to compute $\|u^n\|_X$, $\|u^{n+1}\|_X$, $\|g(u^n;\mu)\|_X$, $< u^{n+1}, g(u^n;\mu) >_X$, $< u^{n+1}, u^n >_X$ and $< u^n, g(u^n;\mu) >_X$. To make the error estimation useful, these terms need to be computed efficiently. We thus need to have a proper offline/online decomposition. For this, we write the decomposition of each term on their respective basis. For the first two terms, there is no problem. Indeed, as $\{\phi_i \circ F^n\}$ is an orthonormal basis in X, we have:

$$\exists \alpha_i,\ \|u^n\|_X^2 = \left\| \sum_i \alpha_i \phi_i \circ F^n \right\|_X^2 = \sum_i \alpha_i^2 \|\phi_i \circ F^n\|_X^2 = \sum_i \alpha_i^2$$

The offline/online decomposition of the terms involving $g(u^n;\mu)$ depends on the nature of the operator $\mathcal{L}$. It can eventually be handled using an EIM type method. Anyway, it is not impacted by calibration, and will thus not be further discussed. The only novelties are the quantities involving scalar products of terms not defined on the same basis, i.e terms for which we have no orthogonality property. For instance,

$$< u^{n+1}, u^n >_X = \sum_i \sum_j \alpha_i^{n+1} \alpha_j^n < \phi_i \circ F^{n+1}, \phi_j \circ F^n >_X$$

Just as in the previous sections, the terms only involve the relative transformation $(F^{n+1})^{-1} \circ F^n$. As usual, we pre compute a few terms for δF in the neighborhood of the identity in $\mathcal{F}$, and interpolate.

Unfortunately, we do not present any numerical experiments for the test case of section 3.4. There are two major reasons why. First of all, the error estimator's tightness is strongly dependent on the relative influence between convective and diffusive terms [101] . Our convection dominated Burgers test case is not adapted. Also, as we are solving a time dependent problem, the error estimate grows exponentially (cumulative error). To mitigate this issue, a space-time formulation has recently been developed in [142] for viscous Burgers.

We have shown in this subsection that the a posteriori estimates naturally extend to our calibration method. Nevertheless, our method inherits the flaws of the standard method: the a posteriori error estimators are not adapted to convection dominated problems and are tedious for time dependent ones.

3.11 Conclusion

In this chapter, we have started by showing that standard reduced basis method (or actually most model reduction methods) were not adapted for convection dominated phenomenon. We have then described the freezing method, which was designed to answer this specific problem. After highlighting its undesirable properties, we have proposed a simpler, alternative, method. Unlike the freezing method, it does not act at the continuous level, but works with a semi-discretized scheme. We have extensively discussed this method on one favorable example, the one dimensional periodic viscous Burgers equation. We have shown that the resulting self-sufficient reduced scheme could be efficiently implemented in the standard online/offline paradigm.

The last sections of this chapter have been devoted to presenting extensions to this generic algorithm. We have started by showing how to extend this method to non periodic problems. We have then proposed one method that would help for problems where the offline calibration parameters can not be picked using some a priori knowledge, but rather have to be numerically approximated. In section 3.8, we have given ideas for the extension to two dimensional problems.

The last 'bonus' section shows how to extend the a posteriori error estimators to the calibration framework.

The next chapter of this thesis will be devoted to using the calibration idea for a more involved problem: the two dimensional flow around a NACA airfoil.

Chapter 4

Calibration for a challenging two dimensional example

The objective of this chapter is to use the calibration procedure introduced in the previous chapter, to solve a more challenging problem: the two dimensional Euler equation around a NACA airfoil. We start by showing that standard ROM is not fitted to solve such problem, as the shock's position is parameter dependent and thus that the Kolmogorov n-width of the solution manifold is large. We then propose an adapted calibration procedure, that uses Gordon-Hall (or transfinite) mappings. We conclude the offline phase by showing that the resulting calibrated solution manifold is better behaved. We then derive an online phase, that follows the lines of the previous chapter, by making use of the fact that only the relative (between two successive time step) shock's position are relevant. The computational complexity is controlled thanks to hyper-reduction ideas. We conclude with preliminary numerical results, that are strong evidences that a complete reduced scheme is within reach.

This work is the result of a collaboration with R. Crisovan and R. Abgrall from UZH Zurich. An article version of this chapter is available, see [26]. Some of the numerical experiments of this chapter have been done using Freefem++ [67].

4.1 Introduction

The objective of this chapter is to apply the calibration idea developed in the previous chapter, to more realistic problems than the one dimensional Burgers equation. We have decided to focus on the steady two dimensional Euler equation around an airfoil. The precise setting will be discussed in Section 4.2. To motivate the calibration idea in this specific setting, we refer to the illustrative Figure 4.1. The coloured lines are going through the barycenters of the mesh elements in which the gradient of the solution is the largest, for various pair of parameters: Mach number and angle of attacks (AoA). Each black line is a fitted line through the position of these barycenters. It is obvious that this example suffers the same problem as the one dimensional Burgers case. Because of the moving shock, the Kolmogorov n-width of the raw data set $\mathcal{M}_{\mathcal{D}}$ will not have the good decay properties required for standard ROM. We thus need a preconditioning step, and will propose an appropriate calibration.

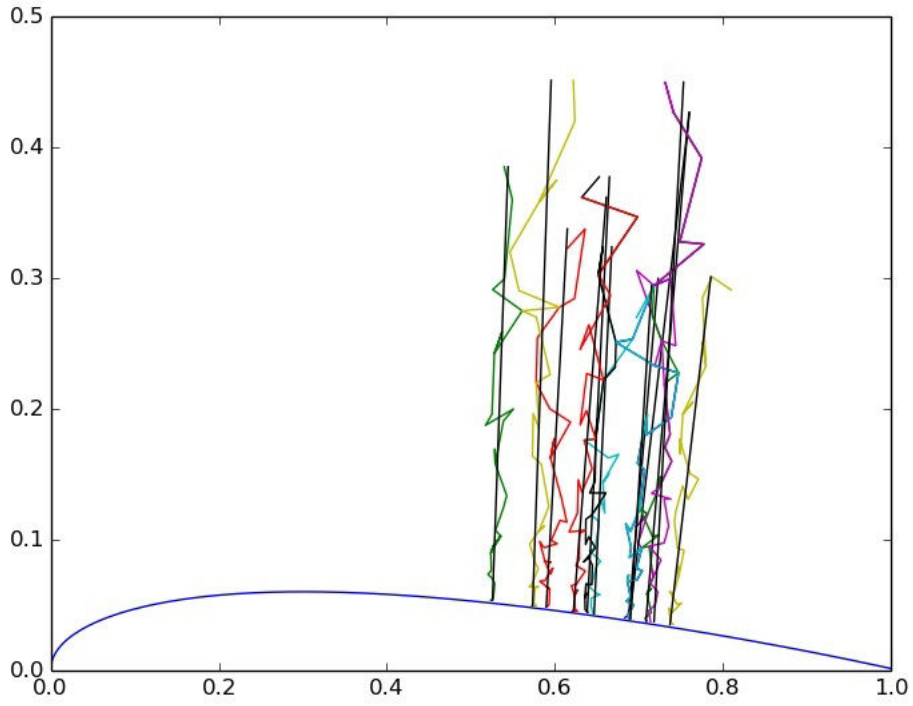


Figure 4.1: Position of the shock for various AoA and Mach numbers.
Coloured lines: Barycenters of the cells in which the shock is located;
Black lines: Fitted line through these barycenters

The choice in this note is to follow the steps of chapter 3, for this 2D hyperbolic problem. That is, we want to:

- calibrate the offline computed solution, to get a reduced basis as small as possible;
- have an online scheme that builds a "calibrated problem", making use of the calibrated reduced basis.

We highlight the differences with chapter 3:

- we were solving Burgers' equation in 1D with one propagating front i.e, there was only one calibration parameter. The shock's position and shape might require more calibration parameters;
- we were using periodic boundary conditions, so there was no matching with any exterior domain. As a result, the calibrated problem was just a translated version of the initial problem;
- we were using a standard Galerkin scheme in the online phase. This scheme is easily put in a reduced framework and, although different from the one used in the offline phase,

was stable for the parabolic problem studied. As we are now working in an hyperbolic setting, it seems like we will require some form of numerical stabilization, which can lead to difficulties in a reduced setting. This issue has already been introduced in chapter 1.

In the first section of this chapter, we completely describe the problem we want to solve. We then give details on the 'truth' scheme we are using. In the second section, we describe our choice of family of mappings $\mathcal{F}$, as well as one possible choice for $\mu \to F_\mu$. We use this to perform the 'offline phase'. We make sure that the calibration procedure leads to a better behaved solution manifold. In the third section, we propose a cheap 'online' algorithm. This is the central part of this chapter, as most related work simply perform the offline calibration, and do not propose any numerical scheme actually using the calibrated manifold $\mathcal{M}_{\mathcal{F},\mathcal{D}}$, see [74, 141, 115]. In the online phase, we propose a standard L^2 minimization algorithm and a L^1 extension, as was advised in [2]. In order to make the overall method computationally efficient, we describe how one could adapt hyper-reduction ideas [119]. The final section is devoted to numerical experiments. We present different mappings and we show the importance of the smoothness of the mappings in $\mathcal{F}$. We conclude this chapter by presenting some ideas that could be further investigated and implemented.

4.2 Problem setting

4.2.1 Naca0012 test case

We have chosen to perform our calibration ideas on the following well documented external flow test-case: the two-dimensional, inviscid, transonic flow past the NACA 0012 airfoil. The explicit form of the wing is given as:

$$y = w(x) := 0.6 \cdot \Big(0.2969 \cdot \sqrt{x} - 0.1260 \cdot x - 0.3516 \cdot x^2 + 0.2843 \cdot x^3 - 0.1015 \cdot x^4 \Big), \text{ for } x \in [0,1]. \quad (4.1)$$

We are using subsonic boundary conditions on the outside boundary and slip boundary conditions on the wing. The latter is a Neumann type boundary condition that imposes that the velocity of the fluid is tangent to the wing.

It is commonly known, that from a certain threshold of Mach number, a shock appears. Both the position and the form of the shock depend on many parameters among which the Mach number and the angle of attack (AoA), i.e the inflow mean direction.

4.2.2 2 dimensional Euler equation

We are interested in the numerical approximation of the two dimensional Euler equations. Let us denote by Ω some domain around the airfoil described in the previous section, W the state vector of conserved variables and $f = (f_x, f_y)$ the flux:

$$\begin{aligned} W =& (\rho, \rho u, \rho v, E)^T \\ f_x(W) =& (\rho u, \rho u^2 + p, \rho uv, u(E+p))^T \\ f_y(W) =& (\rho v, \rho uv, \rho v^2 + p, v(E+p))^T, \end{aligned}$$

ρ is the density, u and v are the components of the velocity, $E = \rho\epsilon + \frac{1}{2}\rho(u^2+v^2)$ is the total energy and ϵ is the specific internal energy. The system is closed by the equation of state relating

the pressure p to the conserved variables:

$$p = (\gamma - 1)\big(E - \frac{1}{2}\rho(u^2 + v^2)\big) = (\gamma - 1)\rho\epsilon,$$

where the ratio of the specific heat γ is constant, with $\gamma = 1.4$ in our applications.

We are interested in the steady solutions. We will take them as the steady limit of the following evolution equation:

$$\begin{cases} \frac{\partial W}{\partial t} + div f(W) &= 0, & t > 0, x \in \Omega \\ W(x,0) &= W_0(x), & x \in \Omega. \end{cases} \tag{4.2}$$

This problem is supplemented with the boundary conditions specified in the previous subsection.

We will take a quick glance at the fine computational method we are using, the Residual Distribution (RD) method. It is a second order oscillation free method. A complete description of this method for steady problems can be found, for example, in [1, 41].

4.2.3 Residual distribution scheme

This short presentation of the RD scheme follows the lines of [41]. In order to approximate the solutions (4.2), we are using a conforming mesh with triangular elements. We will denote with T some generic element in the mesh, with $\mathcal{N}$ the number of elements in the mesh and by M a generic vertex. In the RD schemes, the data are stored at the vertices. W_i will denote an approximation of $W(M_i)$. The scheme also requires a continuous approximation of the flux $f(W)$ over elements. It will be denoted $(f(W))^h$.

Definition 1 *Let W_i be some current state, and $(f(W))^h$ the corresponding continuous approximation of the flux.*

1. $\forall T \in [1, \dots, \mathcal{N}]$ compute the residual

$$\Phi^T := \int_T div((f(W))^h) dx = \int_{\partial T} (f(W))^h \cdot \vec{n}\ d\tilde{x}. \tag{4.3}$$

2. $\forall T \in [1, \dots, \mathcal{N}]$ distribute the functions of Φ^T to each node of T. Denote by Φ_i^T the local nodal residual for the node $M_i \in T$. By construction one must have

$$\sum_{M_i \in T} \Phi_i^T = \Phi^T. \tag{4.4}$$

Equivalently, denoting by β_i^T the distribution coefficient of node M_i:

$$\beta_i^T = \frac{\Phi_i^T}{\Phi^T} \tag{4.5}$$

with

$$\sum_{M_i \in T} \beta_i^T = 1. \tag{4.6}$$

3. Possibly add some numerical stabilization

$$\forall T \in [1, \dots, \mathcal{N}],\ \forall M_i \in T,\ \beta_i^{T,stab} := \beta_i^T + \epsilon_i^T \tag{4.7}$$

4. Assemble the contribution for all vertices M, and solve:

$$\sum_{T \ s.t \ M \in T} \beta_M^{T,stab} \Phi^T = 0. \tag{4.8}$$

Note that as we are dealing with a system, the previous equality is to be understood in $\mathbb{R}^4$. The resolution uses an iterative process (pseudo time-stepping) to get to the solution $\{W_i\}_i$.

This is a very general formulation and many classical schemes can be formulated within this framework. First, one can modify the way the the residual of each triangle is distributed among nodes, that is, the choice of the β_i. For instance, distributing the residual evenly among nodes corresponds to a Lax-Friedrichs type of scheme. One can achieve upwindng by taking into account the transport direction when distributing the residual. Second, many stabilization mechanism can be implemented with the use of specific ϵ_i in (4.7). We have chosen a Lax-Friedrichs type of scheme, with an SUPG stabilization. The consequences of these particular choices will be discussed in the online section.

Remark 27 *Reduced Order Modeling does not necessarily require a deep understanding of the underlying truth solver. We give these details about the CFD code because we intend to use it as part of our online scheme.*

The used fine CFD mesh has 4510 grid points which corresponds to a total of 18040 unknowns. Snapshots in this solution manifold can be visualized in Figure 4.2. We have identified a range of parameters, for which the sensitivity of the shock position to Mach and AoA is high :

$$\mathcal{D} := \begin{cases} \text{Mach} & \in & [0.81, 0.83] \\ \text{AoA} & \in & [0.0^\circ, 3.0^\circ]. \end{cases}$$

The positions of the shock for sample parameters in $\mathcal{D}$ are depicted in Figure 4.1. This problem has been already studied in [2] in the context of model reduction using $L1$-norm minimization. It was shown the existence of discrepancies in the reduced solution, for problems with shocks. This is actually the motivation of this work.

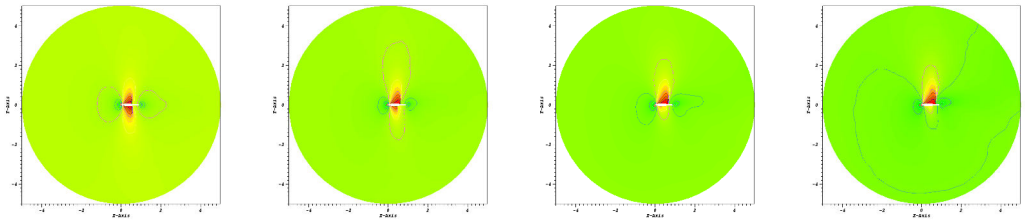
In the rest of this chapter, we will denote u a generic component of the state vector W. For instance, one component of the output of the CFD code for parameter μ will be denoted $u(\cdot; \mu)$. This choice of notation is not made to confuse the reader, but rather to match the standard notation in the ROM community.

4.3 Offline phase

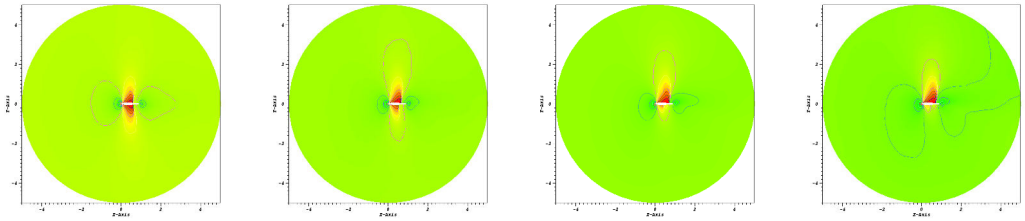
As we will use a POD method to construct a reduced basis, we first need to select a moderate but representative snapshot set inside $\mathcal{M}_\mathcal{D}$. We have chosen the following set of cardinality 12:

$$\begin{array}{ll} \text{Mach} & \in \{0.81, 0.82, 0.83\} \\ \text{AoA} & \in \{0.0^\circ, 1.0^\circ, 2.0^\circ, 3.0^\circ\}. \end{array}$$

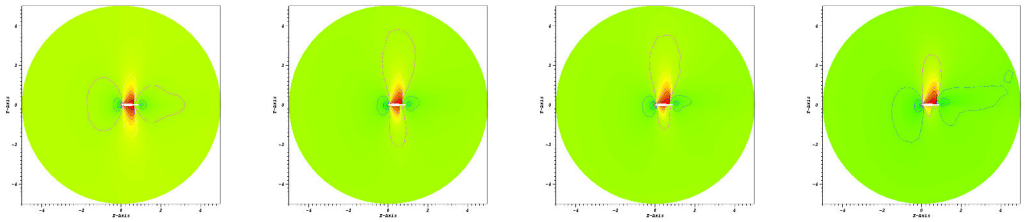
These snapshots are presented in Figure 4.2. We plot a few basis resulting from the application of POD to this data set in Figure 4.4. One can observe that just as in the 1D Burgers' case, in order to take into account the variability of the shocks' position and shape, the reduced basis tend to oscillate. This behavior is even clearer when looking at the restriction of the POD basis at the wing, see Figure 4.3. The first objective of this section is to propose a calibration procedure to mitigate this issue. For this, we construct in the next section a family of mappings $\mathcal{F}$ as well as an application $\mu \to F_\mu$.



(a) Mach=0.81,AoA=0.0° (b) Mach=0.81,AoA=1.0° (c) Mach=0.81,AoA=2.0° (d) Mach=0.81,AoA=3.0°



(e) Mach=0.82,AoA=0.0° (f) Mach=0.82,AoA=1.0° (g) Mach=0.82,AoA=2.0° (h) Mach=0.82,AoA=3.0°



(i) Mach=0.83,AoA=0.0° (j) Mach=0.83,AoA=1.0° (k) Mach=0.83,AoA=2.0° (l) Mach=0.83,AoA=3.0°

Figure 4.2: The solutions of the problem for AoA=$\{0.0°, 1.0°, 2.0°, 3.0°\}$ and Mach=$\{0.81, 0.82, 0.83\ \}$

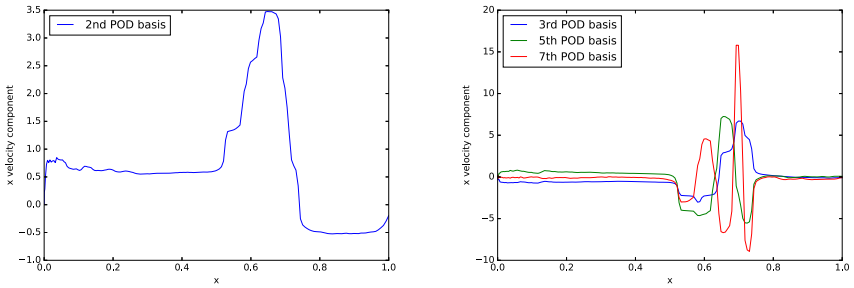


Figure 4.3: The x velocity component at the wing in the uncalibrated case : a few POD basis

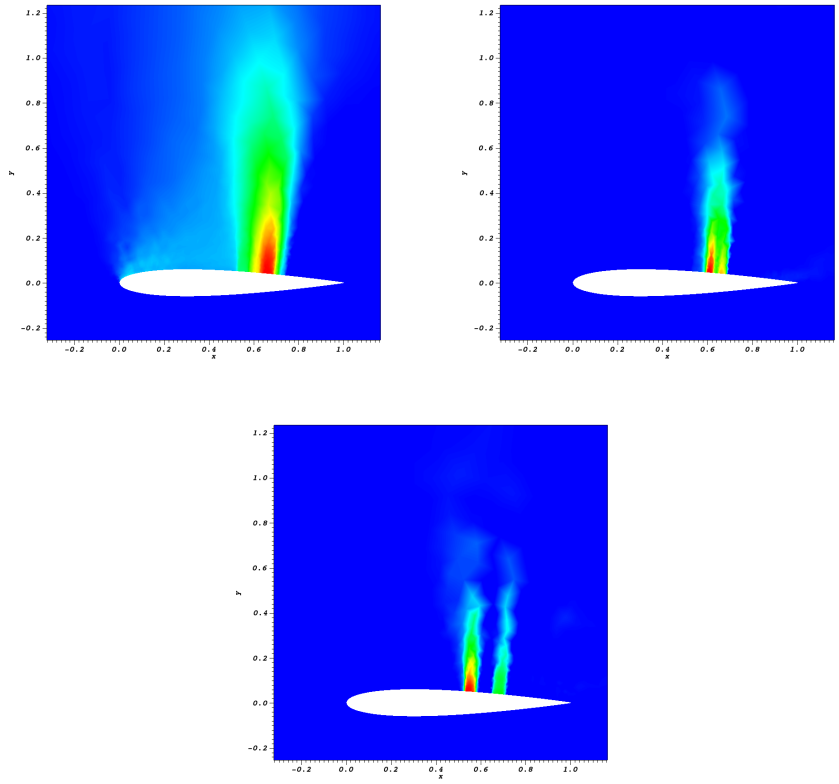


Figure 4.4: 1st, 3th and 5th POD basis at the wing in the uncalibrated case for the full domain Ω

4.3.1 Preliminary remarks

As mentioned in the introduction, calibration starts with some a priori knowledge of the solution manifold. By analogy with the first dimensional Burgers' case, we choose the following calibration: let $\hat{\Omega}$ be some reference domain and $\hat{x}_0$ some abscissa in $\hat{\Omega}$. Construct $\mathcal{F}$ a family of mappings from $\Omega \to \hat{\Omega}$ such that

$$\forall \mu \in \mathcal{D},\ \exists F_\mu \in \mathcal{F},\ \left\{ (\hat{x}, \hat{y}) \in \hat{\Omega} \text{ s.t } u\left(F_\mu^{-1}(\cdot); \mu\right) \text{ is discontinuous } \right\} \subset \{(\hat{x}_0, \hat{y})\}$$

To put it in other words, with this choice of calibration, the solutions in the calibrated manifold

$$\mathcal{M}_{\mathcal{F},\mathcal{D}} := \left\{ u\left(F_\mu^{-1}(\cdot); \mu\right),\ \mu \in \mathcal{D} \right\}$$

have vertical shocks, at position $\hat{x}_0$. Again, using the analogy with the one dimensional Burgers case, we expect that the POD representation of the calibrated manifold would be more representative of the shape of the solutions and not try to catch the moving discontinuity.

How do we achieve this calibration ? The first task is to locate the position of the shock. We have chosen the following simple strategy: first find the boundary element (on the wing) where

the quantity of interest has the highest gradient. Then look at neighboring elements and pick the one with the highest gradient. Iterate until the end of the shock (i.e some condition on the gradient) or until one reaches some predefined distance to the wing. One can use other methods in order to locate more precisely the shock. For instance, in [121], they use ENO related ideas to locate the inner-cell position of the shock.

We denote as $x = s(y;\mu), \mu \in \mathcal{D}$ the true shape of the shock and we will make the following assumption :

$$\exists k \text{ small }, \ \forall \mu \in \mathcal{D}, \exists P_\mu \in \mathcal{P}_k(\mathbb{R}), \ s(y;\mu) = \mathcal{P}_\mu(y). \tag{4.9}$$

That is, the shock can be represented by a low order polynomial. All numerical experiments presented in this chapter have been done using a polynomial of degree 1:

$$P_\mu(y) = a_0(\mu) + a_1(\mu)\ y. \tag{4.10}$$

In Figure 4.1, the colored lines are the barycenters of the control volumes with the highest gradient. In black, is the fitted polynomial, characterized by two parameters, $a_0(\mu)$ and $a_1(\mu)$.

Second step now, we need to construct the family $\mathcal{F}$. The global picture is presented in Figure 4.5. We decompose Ω into three subdomains

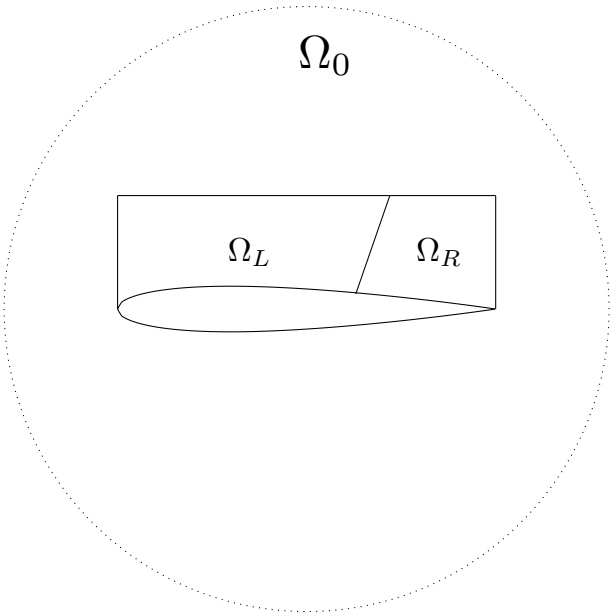


Figure 4.5: Physical domain Ω

- Ω_0 where we will use the identity mapping
- Ω_L and Ω_R, where we will perform the calibration

We have chosen to use a Gordon-Hall (G-H) type mapping [64]. Its properties have been studied in [89]. Examples in fluid dynamics have been numerically studied in [90]. There are multiple reasons for this choice. First, for the offline part, what is important is its simplicity and its flexibility. Second, we will give computational cost related arguments in the online section

below. The rest of this section will detail the application of the Gordon-Hall method onto Ω_L. Similar work is, of course, performed on the right subdomain Ω_R.

The reference domain has to be a rectangle in the original G-H algorithm. This fits in our framework, as we want the calibrated shock to be a vertical line. The situation is depicted in Figure 4.6, where we have plotted one possible instance of $F_\mu^{-1}(\hat{\Omega})$. Contrary to most examples using Gordon-Hall type method in the literature, our domain of interest is embedded in a bigger domain. The mapping thus needs to be (at least) continuous on $\partial\Omega_L$, and $\partial\Omega_R$. More precisely, we need

$$\begin{aligned}(x_1, y_1) &= (\hat{x}_1, \hat{y}_1)\\ (x_3, y_1) &= (\hat{x}_3, \hat{y}_1)\\ (x_3, y_2) &= (\hat{x}_3, \hat{y}_2)\\ (x_1, y_2) &= (\hat{x}_1, \hat{y}_2)\end{aligned}$$

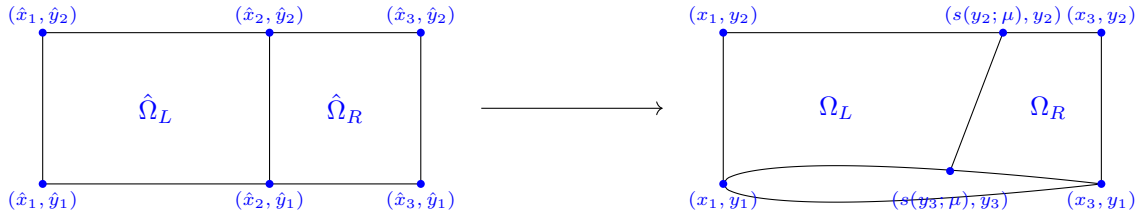


Figure 4.6: The reference domain $\hat{\Omega}$, and one possible instance $\Omega(\mu) := F_\mu^{-1}(\hat{\Omega})$

4.3.2 The actual G-H method

The G-H method is conceptually easy to understand. We denote with Γ_i the edges of Ω_L. We choose a clockwise numbering, starting from the left boundary. Their counterparts on $\hat{\Omega}_L$ are denoted $\hat{\Gamma}_i$. The steps are the following:

- map each edge of $\hat{\Omega}_L$ onto its counterpart on Ω_L. That is define f such that :

$$\forall i,\ f_\mu|_{\hat{\Gamma}_i} = \Gamma_i$$

- define the weights functions ϕ_i:

$$\begin{array}{lcl}\hat{\Omega}_L & \to & [0,1]\\ (\hat{x}, \hat{y}) & \mapsto & \phi_i\end{array}$$

  satisfying the following necessary conditions :

$$\forall i \in [1, \dots, 4],\ \begin{cases}\phi_i + \phi_{i+2} = 1\\ \phi_i|_{\hat{\Gamma}_i} = 1\end{cases}$$

  These functions represent the relative positioning between the opposing edges.

- define the projection functions π_i;

$$\begin{array}{lcl} \hat{\Omega}_L & \to & [0,1] \\ (\hat{x},\hat{y}) & \mapsto & \pi_i \end{array}$$

satisfying the following necessary condition :

$$\forall i \in [1,\dots,4],\ \begin{cases} \pi_i|_{\hat{\Gamma}_{i+1}} & = & 1 \\ \pi_i|_{\hat{\Gamma}_{i-1}} & = & 0 \\ \pi_i|_{\hat{\Gamma}_i} & \in & [0,1]. \end{cases}$$

These functions define a new coordinate system in $\hat{\Omega}_L$.

- for any point $(\hat{x},\hat{y})$ on $\hat{\Omega}_L$, compute the projection on each edge $\pi_i(\hat{x},\hat{y})$. Then, use a weighted combination of the $f_\mu(\pi_i(\hat{x},\hat{y}))$. The weights are the $\phi_i(\hat{x},\hat{y})$.

Remark 28 *The conditions on the sets $\{\phi_i\}$ and $\{\pi_i\}$ stated above are necessary conditions. We have no explicit sufficient conditions to ensure the bijectivity of the G-H mapping.*

As a first easy step, we have chosen to linearly stretch/shrink the domain. That is, we choose the following parametrization of the Γ_i :

$$\begin{array}{l} f_\mu|_{\hat{\Gamma}_1} : (\hat{x}_1,\hat{y}) \to (\hat{x}_1,\hat{y}) \\ f_\mu|_{\hat{\Gamma}_2} : (\hat{x},\hat{y}_2) \to (\hat{x}_1+\hat{x}\cdot(s(\hat{y}_2;\mu)-\hat{x}_1),\hat{y}_2) \\ f_\mu|_{\hat{\Gamma}_3} : (\hat{x}_2,\hat{y}) \to (s(\hat{y}_2+\hat{y}\cdot(\hat{y}_3-\hat{y}_2);\mu),\hat{y}_2+\hat{y}\cdot(\hat{y}_3-\hat{y}_2)) \\ f_\mu|_{\hat{\Gamma}_4} : (\hat{x},\hat{y}_1) \to (s(\hat{y}_3;\mu)+\hat{x}\cdot(\hat{x}_1-s(\hat{y}_3;\mu)), w(s(\hat{y}_3;\mu)+\hat{x}\cdot(\hat{x}_1-s(\hat{y}_3;\mu)))), \end{array} \tag{4.11}$$

where w is defined in (4.1) and s is given in (4.9). For example, take the left edge of the reference domain $\hat{\Gamma}_1$, the set $\left\{(\hat{x},\hat{y}) \in \hat{\Omega},\ \text{ s.t } \hat{y} \in [\hat{y}_1,\hat{y}_2] \text{ and } \hat{x}=\hat{x}_1\right\}$. The vector valued function $f_\mu|_{\hat{\Gamma}_1}$ chosen above is one possible parametrization of Γ_1.

We use, for now, the same weight and same projection functions as in the original G-H formulation:

$$\phi_1(\hat{x},\hat{y}) = \frac{\hat{y}-\hat{y}_1}{\hat{y}_2-\hat{y}_1} \quad \phi_3(\hat{x},\hat{y}) = 1-\frac{\hat{y}-\hat{y}_1}{\hat{y}_2-\hat{y}_1}$$

$$\phi_2(\hat{x},\hat{y}) = \frac{\hat{x}-\hat{x}_1}{\hat{x}_2-\hat{x}_1} \quad \phi_4(\hat{x},\hat{y}) = 1-\frac{\hat{x}-\hat{x}_1}{\hat{x}_2-\hat{x}_1}.$$

and

$$\pi_1(\hat{x},\hat{y}) = \frac{\hat{y}-\hat{y}_1}{\hat{y}_2-\hat{y}_1} \quad \pi_3(\hat{x},\hat{y}) = \frac{\hat{y}-\hat{y}_2}{\hat{y}_3-\hat{y}_2}$$

$$\pi_2(\hat{x},\hat{y}) = \frac{\hat{x}-\hat{x}_2}{\hat{x}_3-\hat{x}_2} \quad \pi_4(\hat{x},\hat{y}) = \frac{\hat{x}_3-\hat{x}}{\hat{x}_3-\hat{x}_1}.$$

The standard G-H mapping is given by :

$$\begin{aligned} GH(\hat{x},\hat{y};\mu) = {} & \phi_1(\hat{x},\hat{y})\cdot f_\mu(\hat{x}_1,\hat{y}) + \phi_2(\hat{x},\hat{y})\cdot f_\mu(\hat{x},\hat{y}_2) \\ & + \phi_3(\hat{x},\hat{y})\cdot f_\mu(\hat{x}_2,\hat{y}) + \phi_4(\hat{x},\hat{y})\cdot f_\mu(\hat{x},\hat{y}_1) \\ & - \sum_{i=1}^{4} \phi_i(\hat{x},\hat{y})\cdot\phi_{i+1}(\hat{x},\hat{y})\cdot f_{i;\mu}, \end{aligned} \tag{4.12}$$

where $f_{i;\mu}$ is the value of f_μ in the corner between Γ_i and Γ_{i+1}. Here, we have

$$f_{1;\mu} = (x_1, y_1), \quad f_{2;\mu} = (x_1, y_2)$$
$$f_{3;\mu} = (x_3, y_2), \quad f_{4;\mu} = (x_3, y_1).$$

We will use, in the course of this chapter, the following notation :

$$\begin{array}{lcl} \mathbb{R}^2 & \to & \mathcal{F} \\ (a_0, a_1) & \mapsto & \text{G-H}(\cdot; a_0, a_1) \end{array} \tag{4.13}$$

This application takes as argument a shock position, and returns the corresponding G-H mapping in $\mathcal{F}$.

Remark 29 *It is important to know that the π's, the ϕ's and f_μ can be chosen independently from each other. This will be made clearer in Section 4.7.2 when we try to improve the method.*

It is clear that this mapping suffers from major drawbacks :

- this mapping is continuous at the boundary, but has discontinuous derivatives;
- this mapping linearly stretches/shrinks the domain; this is not the best choice to diminish the Kolmogorov n-width;
- in x_1 and x_3, the boundary $\partial\hat{\Omega}$ is not C^1.

These issues will be fixed in the numerical section 4.7.2. They are not a problem for the offline section. Thus, for simplicity, we will illustrate the usefulness of calibration using this rough mapping. We have computed separate POD basis on $\hat{\Omega}_L$ and $\hat{\Omega}_R$. We present in Figure 4.7 the counterpart of Figure 4.3, that is, the x component of the velocity on the left part of the wing. As one can see, using calibration we got rid of the oscillations. We present in Figure

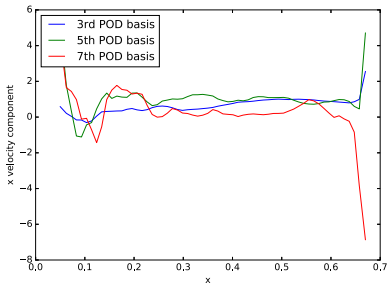


Figure 4.7: The x velocity component at the wing in the calibrated case : a few POD basis

4.8 the first, third and fifth POD basis in the calibrated case, as a counterpart of Figure 4.4. As expected, the calibrated POD captures most of the information in the first 4 basis. The 5th basis only contains numerical noise. The first objective of this chapter has been solved, we know how to build a better behaved solution manifold.

We present in the next section a reduced scheme with a computational complexity independent of the size of the truth problem, based on the calibrated basis that we have just constructed.

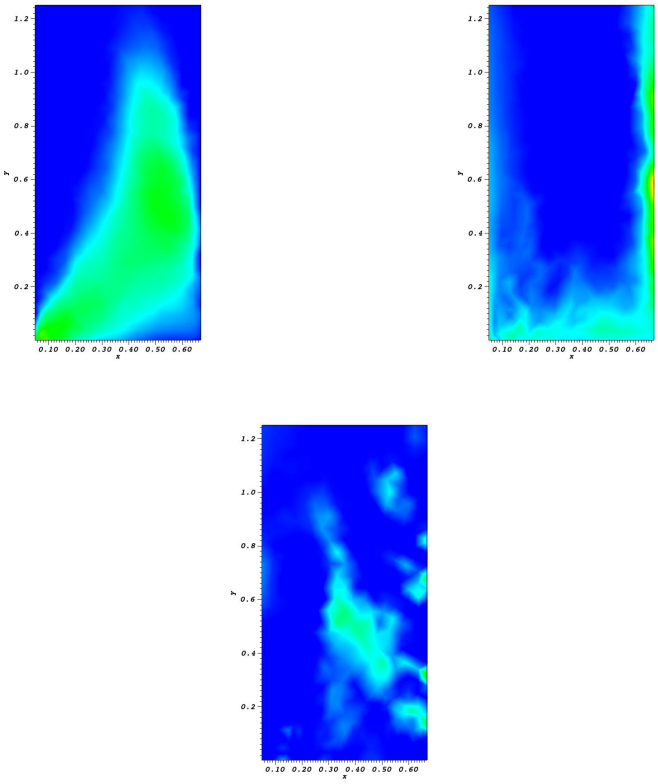


Figure 4.8: 1st, 3th and 5th POD basis in the calibrated case for the left subdomain

4.4 Online phase

For the remaining of this section, we drop out the μ dependency, as we are focused on reducing one particular simulation. It will reappear in the offline/online decomposition section. Also, we use the following notation:

- t^n is the discrete pseudo time;
- F_n the mapping chosen at time step n. It maps Ω onto $\hat{\Omega}$. The inverse mapping will be denoted F_n^{-1};
- $\{\phi_i\}$ is some reduced basis on the reference mesh, of cardinality N^{red}. The one constructed in section 4.3.

We denote by u^n the solution at pseudo time step t^n and $\hat{u}^n$ it's counterpart on the reference mesh. That is, we have

$$\hat{u}^n = u^n \circ F_n^{-1} \text{ on } \hat{\Omega}.$$

We see three paths, that we present with increasing difficulty:

- The easiest method we can think of is the following:

  - suppose we have some reduced solution at iteration n, $\hat{u}^n$, defined on the reference domain $\hat{\Omega}$, and a "well chosen" mapping F_n;
  - map this reduced solution onto the real mesh, using F_n;
  - use the CFD code, on Ω, using $\hat{u}^n \circ F_n$ as initial condition, to get u^{n+1};
  - map u^{n+1} back onto $\hat{\Omega}$. This implies finding a "good" (in some sense) mapping F_{n+1}, and the corresponding reduced coordinates.

- The second method is smarter, and more in the spirit of what has been done previously, in chapter 3
  - just as for the first method, suppose we have some reduced solution at iteration n, $\hat{u}^n$, defined on the reference domain $\hat{\Omega}$, and a "well chosen" mapping F_n;
  - use a CFD code on $\hat{\Omega}$ using $\hat{u}^n$ as initial condition. This implies of course the modification of flux and boundary conditions to make this "non physical problem" equivalent to the initial one Denote $\tilde{u}^{n+1}$ the output. By construction, we have:
$$\tilde{u}^{n+1} \approx u(\cdot, t^{n+1}) \circ F_n^{-1};$$
  - deduce a new "relative" mapping: $F_{n+1} \circ F_n^{-1}$ best suited to represent $\tilde{u}^{n+1}$. From this, compute a better calibrated solution $\hat{u}^{n+1}$ and the corresponding mapping F_{n+1} such that
$$u(\cdot, t^{n+1}) \approx \hat{u}^{n+1} \circ F_{n+1};$$

- a third approach is to construct a self sufficient reduced scheme. As mentioned in chapter 1, standard CFD codes often imply numerical stabilization unfit for the reduced setting. The self sufficient scheme will thus necessarily rely on new ingredients, such as the one introduced in section 1.6.1.

The first method will not be further discussed here, as the numerous mesh interpolations imply very high computational costs, as well as numerical errors. The third method is out of the scope of this chapter. We have chosen to prove the feasibility of the second method. It assumes the existence of a fully functioning CFD code. In the lines of what has been done [31], the idea is to keep the stability and accuracy properties of the existing code. The computational savings would be obtained using EIM/hyper reduction ideas.

The objective is to recast the original problem defined on Ω, onto an equivalent problem defined on $\hat{\Omega}$. This is a well studied problem in the elliptic and parabolic communities, see for instance [110, 88]. It relies on the variational form of the PDE at hand. A similar procedure for our hyperbolic problem could be performed on a non conservative formulation. There are two issues with this approach in our setting. The first one is that this derivation is not rigorous as some of the quantities appearing are not properly defined for discontinuous solutions. Also, this formulation is not suited for our purpose, as the resulting problem is no longer posed as a conservation law, and thus require some intrusion into the CFD code. The intent here is to find a mapping procedure fitted for conservation laws. We will see that it involves a modifications of both flux and boundary conditions.

We start with a step common to Finite Volume schemes and Residual Distribution schemes. Let $\{\omega_i,\ i \in [1, \dots, \mathcal{N}]\}$ be the set of control volumes in Ω and let u be any state variable. The integration of the conservation law in space and time, in control volume i gives:

$$\int_{\omega_i} u(w, t^{n+1}) dw - \int_{w_i} u(w, t^n) dw + \int_{\omega_i} \int_{t^n}^{t^{n+1}} \nabla \cdot f(u) dt dw = 0. \tag{4.14}$$

It is known that equation (4.14) is equivalent to:

$$\int_{\hat{\omega}_i} \hat{u}(\hat{w}, t^{n+1})|J_{F_n^{-1}}|d\hat{w} - \int_{\hat{\omega}_i} \hat{u}(\hat{w}, t^n)|J_{F_n^{-1}}|d\hat{w} + \int_{\hat{\omega}_i}\int_{t^n}^{t^{n+1}} \nabla_{\hat{w}} \cdot (N_n^T f(\hat{u}))dtd\hat{w} = 0 \quad (4.15)$$

where $\hat{u} := u \circ F_n$, $\hat{\omega}_i = F_n^{-1}\omega_i$ and $N_n^T f$ is the correct modified flux with

$$N_n^T = \begin{bmatrix} (J_{F_n^{-1}})_{22} & -(J_{F_n^{-1}})_{12} \\ -(J_{F_n^{-1}})_{21} & (J_{F_n^{-1}})_{11} \end{bmatrix}_n,$$

where J_F denotes the Jacobian of any mapping F. This equality is known as the Piola transform, which is usually used in a different context. For more details, we refer for instance to [90]. We will make the assumption that the determinant of the Jacobian is sufficiently smooth and the mesh is fine enough so that we can consider N_n^T constant per element. The error due to this approximation will not be investigated in this chapter.

Remark 30 *Some more rigorous approaches could be developed, but would lead to more intrusion into the CFD code. In [36] for instance, they choose to work with the average of $\hat{u}|J_{F_n^{-1}}|$ over control volumes, instead of $\hat{u}$.*

We arrive to the following equation in each control volume $\hat{w}_i$.

$$\int_{\hat{\omega}_i} \hat{u}(\hat{w}, t^{n+1})d\hat{w} - \int_{\hat{\omega}_i} \hat{u}(\hat{w}, t^n)d\hat{w} + \frac{1}{|J_{F_n^{-1}}|_i}\int_{\hat{\omega}_i}\int_{t^n}^{t^{n+1}} \nabla_{\hat{w}} \cdot (N_n^T f(\hat{u}))dtd\hat{w} = 0.$$

We have all the ingredients to feed the CFD code:

- a mesh: here it is the reference mesh, over $\hat{\Omega}$;
- the average of the solution over control volumes:

$$\hat{\mathbf{u}}_i = \frac{1}{\text{mes}(\hat{w}_i)}\int_{\hat{w}_i} \hat{u}(\hat{w}, t^n)$$

- a flux, in a closed form: with the Piola transform, here it just amounts to

$$N_n^T f$$

  where the N^T term will depend on the time step and is not constant over $\hat{\Omega}$. We will see in Section 4.6.3 that using G-H type mapping allows for a proper offline/online decomposition

- boundary conditions: we do not need to worry about the outside boundary conditions, as they are not be affected by the mapping. The slip boundary conditions for the original problem are given by

$$u \cdot \vec{n} = 0 \text{ on the wing.}$$

  In our case, these are imposed as follows: treat the boundary nodes as any other node, and add the correct quantity to impose the slip boundary condition. More precisely, let $n = (n_1, n_2)$ be the norm at the boundary . The flux at nodes on the boundary are given by:

$$(f_x, f_y) \cdot n = \begin{pmatrix} \rho\Big((u,v)\cdot n\Big) \\ \rho u\Big((u,v)\cdot n\Big) + pn_1 \\ \rho v\Big((u,v)\cdot n\Big) + pn_2 \\ \Big((u,v)\cdot n\Big)(E+p) \end{pmatrix}$$

We enforce the slip boundary condition by subtracting the following quantity:

$$(\tilde{f}_x, \tilde{f}_y)\cdot n = \begin{pmatrix} \rho\Big((u,v)\cdot n\Big) \\ \rho u\Big((u,v)\cdot n\Big) \\ \rho v\Big((u,v)\cdot n\Big) \\ \Big((u,v)\cdot n\Big)(E+p) \end{pmatrix}$$

We can use the Piola transform again for these terms. The subtracted quantity formulated in terms of the reference variables is simply given by

$$\int_{\partial\hat{K}} (\tilde{f}_x(\hat{u}), \tilde{f}_y(\hat{u}))\cdot(N^T\cdot n).$$

The conclusion from this analysis is that under the assumption that the determinant of the Jacobian is constant per element, changing the normals in the CFD code is enough to compute the total residual in each triangle. This is the first part of the RD scheme, see section 4.2.3. What follows is the distribution of the residual among nodes, in each element. As mentioned in the offline section, the CFD code is of Lax-Friedriechs type. The residual is evenly distributed. This procedure is independent of the mesh and of the solution. There is no additional work. For an upwinding scheme, this is a much more difficult problem to tackle, not in the scope of this chapter.

As mentioned in section 4.2.3, the truth scheme uses SUPG type stabilization. We have not studied in this chapter how to modify this term in order to have an equivalent stabilization procedure on $\hat{u}$. We will discuss this approximation in the numerical experiment section.

We now assume that we have performed the $n+1$th iteration with the CFD code. The output is denoted $\tilde{u}^{n+1}$ and by construction, $\tilde{u}^{n+1}\circ F_n \approx u^{n+1}$. As F_n is not, a priori, the right mapping for u^{n+1}, we are looking simultaneously for:

- a better suited mapping F_{n+1}
- the corresponding $\hat{u}^{n+1}$ expressed in terms of the reduced basis defined on $\hat{\Omega}$.

Following the lines of chapter 3, define the following objective function, for $p\in\{1,2\}$:

$$J^p : \left\{ \begin{array}{lcl} \mathcal{F}\times\mathbb{R}^{N^{red}} & \to & \mathbb{R} \\ F, \{\alpha_k\}_k & \mapsto & \left\| \tilde{u}^{n+1}\circ F_n - \sum_{k=1}^{N^{red}} \alpha_k\phi_k\circ F \right\|_{L^p(\Omega)} . \end{array} \right. \tag{4.16}$$

4.5 Finding the coordinates, for a fixed mapping

In this section, we are working for fixed $\tilde{u}^{n+1}$ and F_n. We first propose an optimization procedure when the mapping F in (4.16) is assumed to be known. Fix $F\in\mathcal{F}$ and define J^p_F as:

$$J^p_F : \left\{ \begin{array}{lcl} \mathbb{R}^{N^{red}} & \to & \mathbb{R} \\ \{\alpha_k\}_k & \mapsto & \left\| \tilde{u}^{n+1}\circ F_n - \sum_{k=1}^{N^{red}} \alpha_k\phi_k\circ F \right\|_{L^p(\Omega)} \end{array} \right. \tag{4.17}$$

We are going to discuss 2 particular cases. First the $p=2$ case, standard in ROM and then an extension to $p=1$ minimization, which was advised in [2].

4.5.1 L^2 minimization, standard Galerkin projection

The objective functional is thus given by:

$$J_F^2 : \{\alpha_k, k \in [1, \dots, N^{red}]\} \to \left\| \tilde{u}^{n+1} \circ F_n - \sum_{k=1}^{N^{red}} \alpha_k \phi_k \circ F \right\|_{L^2(\Omega)}$$

First order optimality condition gives us the αs. One needs to take into account the fact that the basis $\{\phi_k \circ F\}_k$ will most probably not be an orthogonal basis.

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{N^{red}} \end{pmatrix} = A \begin{pmatrix} < \tilde{u}^{n+1} \circ F_n, \phi_1 \circ F >_{L^2(\Omega)} \\ < \tilde{u}^{n+1} \circ F_n, \phi_2 \circ F >_{L^2(\Omega)} \\ \vdots \\ < \tilde{u}^{n+1} \circ F_n, \phi_{N^{red}} \circ F >_{L^2(\Omega)} \end{pmatrix}$$

where $A_{i,j} :=< \phi_i \circ F, \phi_j \circ F >_X$ is a symmetric invertible square matrix of size N^{red}. Define $\delta_F := F \circ F_n^{-1}$. We have

$$\begin{array}{rcl} \forall i,\ < \tilde{u}^{n+1} \circ F_n, \phi_i \circ F >_{L^2(\Omega)} & = & \int_{\hat{\Omega}} \tilde{u}^{n+1} \phi_i \circ \delta_F |J_{F_n^{-1}}| \\ \forall i, j,\ < \phi_i \circ F, \phi_j \circ F >_{L^2(\Omega)} & = & \int_{\hat{\Omega}} \phi_i \phi_j |J_{F^{-1}}| \end{array}$$

As in the previous chapter, we have replaced the expensive problem involving the absolute mapping by a problem where mappings are close to the identity. We will see in section 4.6.3 how to achieve efficient offline/online decomposition.

4.5.2 L^1 minimization

The objective functional is here given by:

$$\forall \alpha \in \mathbb{R}^{N^{red}},\ J_F^1(\alpha) = \sum_{i=1}^{\mathcal{N}} \int_{\hat{\omega}_i} \left| \tilde{u}^{n+1} \circ F_n - \sum_{k=1}^{N^{red}} \alpha_k \phi_k \circ F \right|.$$

Once again, a standard change of variable gives:

$$\forall \alpha \in \mathbb{R}^{N^{red}},\ J_F^1(\alpha) = \sum_{i=1}^{\mathcal{N}} \int_{\hat{\omega}_i} \left| \tilde{u}^{n+1} \circ \delta_F^{-1} - \sum_{k=1}^{N^{red}} \alpha_k \phi_k \right| |J_{F^{-1}}| \tag{4.18}$$

In each control volume i, we choose a set of N_i^{quad} quadrature points $\{\hat{x}_{i,j},\ j \in [1, \dots, N_i^{quad}]\}$ and the corresponding weights $\{\gamma_{i,j},\ j \in [1, \dots, N_i^{quad}]\}$. We have:

$$J_F^1(\alpha) = \sum_{i=1}^{\mathcal{N}} \sum_{j=1}^{N_i^{quad}} \gamma_{i,j} \left| \tilde{u}^{n+1}(\delta_F^{-1}(\hat{x}_{i,j})) - \sum_{k=1}^{N^{red}} \alpha_k \phi_k(\hat{x}_{i,j}) \right| |J_{F^{-1}}(\hat{x}_{i,j})|. \tag{4.19}$$

This is handled as in [2] by recasting it as a linear programming problem. For now, the size of the problem is of order $\mathcal{N}$, the number of control volumes of the mesh. We will see in section 4.6.3 how to reduce the computational cost.

4.6 Finding the mapping

One important remark, similar to the one made in chapter 3 is that the shock's position evolves smoothly in time. This is rigorously justified by Rankine-Hugoniot conditions. Let $\hat{A}_0$ and $\hat{A}_1$ be the maximum absolute values for the variation between two successive pseudo time steps of respectively position and slope of the shock. These are roughly given by:

$$\forall i \in \{0,1\},\ \hat{A}_i \approx W^{1,\infty} \text{ (maximum shock speed)}.$$

We use these values to define the following neighborhood of the identity in $\mathcal{F}$:

$$\mathcal{F}^{rel} := \left\{ \text{G-H}(\hat{a}_0, \hat{a}_1),\ |\hat{a}_i| \leq \hat{A}_i \right\},$$

where the application G-H has been defined in equation (4.13).

4.6.1 Alternative differentiable objective function

Let $\hat{u} \in \mathcal{M}_{\mathcal{F},\mathcal{D}}$. It is clear that for solutions with shocks, the following application is not smooth:

$$\begin{array}{lll} \mathcal{F}^{rel} & \to & X \\ \delta_F & \mapsto & \hat{u}(\delta_F(\cdot)). \end{array}$$

More precisely, the derivative in the sense of distributions has a Dirac mass at the shock. We give here a formal proof, and refer to [13, 6] for a rigorous one. Denote

$$\Sigma_0 := \left\{ (\hat{x}, \hat{y}) \in \hat{\Omega},\ \text{ s.t } \hat{u} \text{ is discontinuous } \right\} \text{ and } v \to [v] \text{ the standard jump operator.}$$

By construction, Σ_0 is independent of $\hat{u} \in \mathcal{M}_{\mathcal{F},\mathcal{D}}$. Each solution in the calibrated solution manifold can be decomposed into a smooth component and one discontinuity:

$$\forall \hat{u} \in \mathcal{M}_{\mathcal{F},\mathcal{D}},\ \exists \hat{u}_{smooth} \text{ and } \hat{u}_j,\ \text{ s.t } \hat{u} = \hat{u}_{smooth} + [\hat{u}_j]|_{\Sigma_0}$$

$$\hat{u} - \hat{u} \circ \delta_F = \hat{u}_{smooth} - \hat{u}_{smooth} \circ \delta F + [\hat{u}_j]|_{\Sigma_0} - [\hat{u}_j \circ \delta F]|_{\delta F^{-1}(\Sigma_0)}$$

The derivative in the sense of distributions has thus also two components:

$$\hat{u} - \hat{u} \circ \delta_F \approx \partial \hat{u}_{smooth} + \delta|_{\Sigma_0} \partial\Sigma$$

where $\delta|_{\Sigma_0}$ is the Dirac mass at Σ_0.

We propose one option to circumvent this issue, an alternative and differentiable objective function. For $\tilde{u}^{n+1}$ the output of one iteration of the CFD code, denote

$$\Sigma(\tilde{u}^{n+1}) := \left\{ (\hat{x}, \hat{y}) \in \hat{\Omega} \text{ s.t, } \tilde{u}^{n+1} \text{ is discontinuous } \right\}$$

As already mentioned, because of R-H condition, $\Sigma(\tilde{u}^{n+1})$ will be close to Σ_0. We use the $p = 1$ notation, but the following approach can be directly transposed to the $p = 2$ case. It is easy to see why for $\hat{x}_{i,j}$ sufficiently far from the shock so that

$$\forall \delta_F \in \mathcal{F}^{rel},\ \delta_F(\hat{x}_{i,j}) \text{ is on the same side of } \Sigma_0 \text{ as } \hat{x}_{i,j} \tag{4.20}$$

the following application is differentiable:

$$\left\{ \begin{array}{lll} \mathcal{F}^{rel} & \to & \mathbb{R} \\ \delta_F & \mapsto & \tilde{u}^{n+1}(\delta_F(\hat{x}_{i,j})). \end{array} \right.$$

Following this remark, we denote $\hat{\Omega}_d$ the subdomain of $\hat{\Omega}$ where we have removed some neighborhood of the shock. More precisely, let

$$\hat{\Omega}_d := \bigcup \hat{\omega}_i, \text{ for } i \in [1, \dots, \mathcal{N}], \text{ s.t } \forall j \in [1, \dots, N_i^{quad}], \ \hat{x}_{i,j} \text{ satisfies condition (4.20)} .$$

We denote Ω_d it's counterpart in the physical domain.

Remark 31 *For the L^1 norm, the overall problem as presented is not differentiable. This can be solved using Huber type minimization instead of the raw L^1 [2].*

Following the previous discussion, we define smaller objective functions. For every Ω_{sub} subdomains of Ω, define the following $J_{\Omega_{sub}}$:

$$J^p_{\Omega_{sub},F} : \{\alpha_k\}_k \to \left\| \tilde{u}^{n+1} \circ F_n - \sum_{k=1}^{N^{red}} \alpha_k \phi_k \circ F \right\|_{L^p(\Omega_{sub})} .$$

With this new notation, we replace the original problem J^p_F with the differentiable objective function $J^p_{\Omega_d,F}$. We can now perform standard optimization algorithm to get the desired mapping δ_F, as

$$\begin{array}{lll} [-\hat{A}_0, \hat{A}_0] \times [-\hat{A}_1, \hat{A}_1] \times \mathbb{R}^{N^{red}} & \to & \mathbb{R} \\ \hat{a}_0, \hat{a}_1, \{\alpha_k\} & \mapsto & J_{\Omega_d, \text{G-H}(\hat{a}_0, \hat{a}_1)}(\alpha) \end{array} \tag{4.21}$$

is a smooth application.

4.6.2 One possible algorithm

We now present one way of performing in practice the optimization of the quantity defined in (4.21).

- discretize the set $[-\hat{A}_0, \hat{A}_0]$ and $[-\hat{A}_1, \hat{A}_1]$: $\{\hat{a}_0^k, \ k \in [1, \dots, N_0]\}$ and $\{\hat{a}_1^k, \ k \in [1, \dots, N_1]\}$
- denote $\Psi_{\mathcal{F}^{rel}}$ the following sample of $\mathcal{F}^{rel}$:

$$\Psi_{\mathcal{F}^{rel}} := \{\text{G-H}(\hat{a}_0^k, \ \hat{a}_1^p), \ k \in [1, \dots, N_0], \ p \in [1, \dots, N_1]\}.$$

- compute the coordinates for all mappings in $\Psi_{\mathcal{F}^{rel}}$ using section 4.5 and deduce the corresponding value of the objective function:

$$\forall \delta_F \in \Psi_{\mathcal{F}^{rel}}, \quad \text{compute} \inf_{\alpha \in \mathbb{R}^{N^{red}}} J_{\Omega_d, \delta_F}(\alpha).$$

- interpolate the previously computed quantities to get an estimate of

$$\inf_{\alpha \in \mathbb{R}^{N^{red}}} J_{\Omega_d, \delta_F}(\alpha) \text{ over } \mathcal{F}^{rel}.$$

  Deduce the value of the optimal coefficients $\hat{a}_0^{opt}$ and $\hat{a}_1^{opt}$, as in chapter 3

- deduce the reduced coordinates for the corresponding mapping G-H($\hat{a}_0^{opt}$, $\hat{a}_1^{opt}$) using section 4.5

Remark 32 *Other ideas to find F_{n+1} can be implemented. They are however less natural in our framework.*

- ***Shock fitting:*** *close to what has been described in the offline section. Find the control volumes such that $\tilde{u}^{n+1}$ has highest gradient and fit a polynomial. This is made computationally efficient because we do not need to look for highest gradient all over Ω: $\Sigma(\tilde{u}^{n+1})$ is close to Σ_0.*

- ***RH condition:*** *update the shock's position and slope using the shock's speed's explicit form given by Rankine-Hugoniot.*

4.6.3 Online/offline decomposition

We have not yet discussed the computational complexity of our full algorithm. For now, at each time step, we need to run the full CFD code to get $\tilde{u}^{n+1}$ over $\hat{\Omega}$. Until we manage to build a self sufficient reduced scheme, see the third method described in section 4.4 , this computational time is not easily reducible. The only ideas available in the literature are hyper reduction [119].

In the previous section, we have restricted the problem from Ω to Ω_d because of differentiability. Here, we replace Ω_d by an even smaller, denoted generically Ω_{sub} because of computational cost. Of course, we will look for Ω_{sub} subsets of Ω_d to keep the differentiability property. The hyper-reduction method is an empirical procedure that aims at selecting a "good" Ω_{sub}.

We present here a version of the hyper-reduction procedure that uses a different objective function than $J^p_{\Omega_{sub},F}$ defined in the previous section. Note that many different variants around the algorithm we propose here are possible. For $\hat{u} \in \mathcal{M}_{\mathcal{F},\mathcal{D}}$, define

$$I^p_{\hat{\Omega}_{sub}}(\hat{u}) : \left\{\alpha_k,\ k \in [1,\dots,N^{red}]\right\} \mapsto \left\| \hat{u} - \sum_{k=1}^{N^{red}} \alpha_k \phi_k \right\|_{L^p(\hat{\Omega}_{sub})} .$$

During the hyper-reduction procedure, we try to find $\hat{\Omega}_{sub}$ such that:

$$\forall \hat{u} \in \mathcal{M}_{\mathcal{F},\mathcal{D}},\quad \underset{\{\alpha_k\}\in\mathbb{R}^{Nred}}{\operatorname{arginf}} I^p_{\hat{\Omega}_{sub}}(\hat{u})\left(\left\{\alpha_k,\ k \in [1,\dots,N^{red}]\right\}\right) \approx \underset{\{\alpha_k\}\in\mathbb{R}^{Nred}}{\operatorname{arginf}} I^p_{\hat{\Omega}_d}(\hat{u})\left(\left\{\alpha_k,\ k \in [1,\dots,N^{red}]\right\}\right).$$

That is, we want the optimization not to be affected too much by the reduction of the size of the problem. Of course, we do not know the continuous set $\mathcal{M}_{\mathcal{F},\mathcal{D}}$. Let us denote $\Xi_{\mathcal{M}_{\mathcal{F},\mathcal{D}}}$ a representative set of the continuous manifold, and let ϵ be some threshold. We perform the following greedy algorithm.

Data: $\Xi_{\mathcal{M}_{\mathcal{F},\mathcal{D}}}$,$\{\phi_k,\ k \in [1,\dots,N^{red}]\}$
Result: $\hat{\Omega}_{hyper}$,N^{hyper}
Initialize $\hat{\Omega}_{hyper} := \bigcup_{i\in I_{ini}} \hat{\omega}_i$;
repeat
  $\forall \hat{u} \in \Xi_{\mathcal{M}_{\mathcal{F},\mathcal{D}}},\ \{\beta_k(\hat{u}), k \in [1,\dots,N^{red}]\} :=$
  $\underset{\{\alpha_k,\ k\in[1,\dots,N^{red}]\}}{\operatorname{arginf}} I^p_{\hat{\Omega}_{hyper}}(\hat{u})\left(\{\alpha_k,\ k \in [1,\dots,N^{red}]\}\right)$;
  $i := \underset{p\in\mathcal{N}}{\operatorname{argsup}} \underset{\hat{u}\in\Xi_{\mathcal{M}_{\mathcal{F},\mathcal{D}}}}{\sup} \| \sum_{k=1}^{N^{red}} \beta_k(\hat{u})\phi_k - \hat{u}\|_{L^p(\hat{w}_p)}$;
  $\hat{\Omega}_{hyper} := \hat{\Omega}_{hyper} \bigcup \hat{\omega}_i$
until convergence;
$N^{hyper} := \operatorname{card}(\hat{\Omega}_{hyper})$;

Algorithm 8: One possible algorithm to select $\hat{\Omega}_{sub}$

The idea of hyper reduction is that on the solution manifold there is a one to one correspondence between the restriction of the solution on $\hat{\Omega}^{hyper}$ and the full solution. For the problem at hand, we expect that the solutions in the solution manifold are characterized by their behavior in the vicinity of the shock. Because of calibration, the knowledge of the solutions in a reduced number of control volumes around Σ_0, independent of μ, should thus be enough to completely characterize the solution.

4.6.4 Implementation details

For our choice of online implementation, the computation of the N^T terms is not a pressing issue, as these are only required in a moderate number of cells, denoted by N^{hyper}. We will nevertheless emphasize that these terms, because of the choice of Gordon-Hall type mapping, would not be a computational problem even with no hyper reduction. By inspecting the structure of the G-H mapping, see equation (4.12), we can see that the weights and the projection functions are not parameter dependent. Also, in our problem, $\mu \to \psi_i(\cdot;\mu)$ for $i \in \{1,2,3\}$ are linear function of μ. Most of the terms appearing in equation (4.12) are thus trivially affinely decomposable. The fact that the computation of terms involving ψ_4 also fall into the offline/online decomposition paradigm requires more work. We do not enter the details, but one could show it by using a variation of the G-H method, see section 4.7.2, and the fact that away from Γ_1, the wing can be approximated by a polynomial.

4.7 Numerical Experiments

The framework presented in this chapter present many similarities with the method described in chapter 3. The choice in this chapter has thus been to focus the numerical effort on the real novelty: the resolution of an equivalent calibrated problem on a reference mesh using the Piola transform, see section 4.4. Of course, the overall performance of such an approach relies on the ability to construct a smooth family of mappings $\mathcal{F}$. This has been challenging and is a big part of the numerical experiments presented below.

4.7.1 Mapping on a flat domain

The first experiment we discuss is a preliminary, alpha test: we try to reproduce one snapshot, using the Piola transform and a reference mesh. We are running the CFD code for Mach = 0.81

and AoA = 3.0°. The truth solution that we are trying to recover is presented in Figure 4.9. We

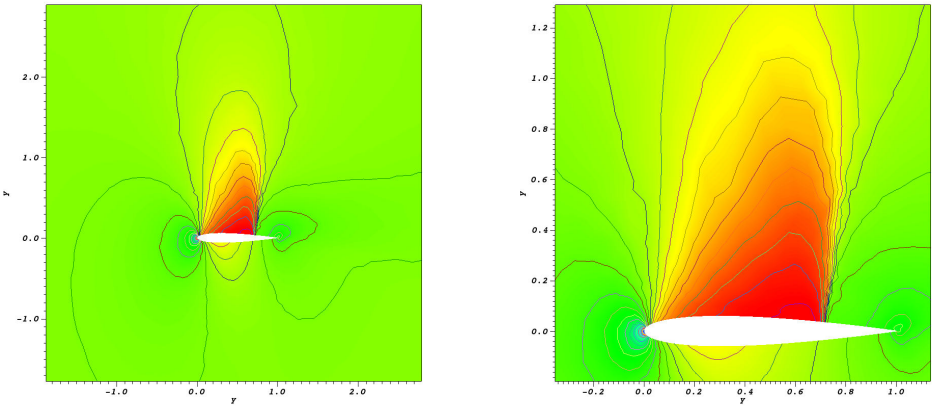


Figure 4.9: Truth solution for velocity component with Mach=0.81 and AoA=3.0°

first perform a 'control sample' test. We run the original CFD code on the reference mesh of section 4.3. The output solution is presented in Figure 4.10. As expected, it is not comparable with the truth solution: $u(\mu)$. Indeed, the problem solved is not equivalent to the original one. We need to modify fluxes and boundary conditions, as presented in section 4.4. As the steady

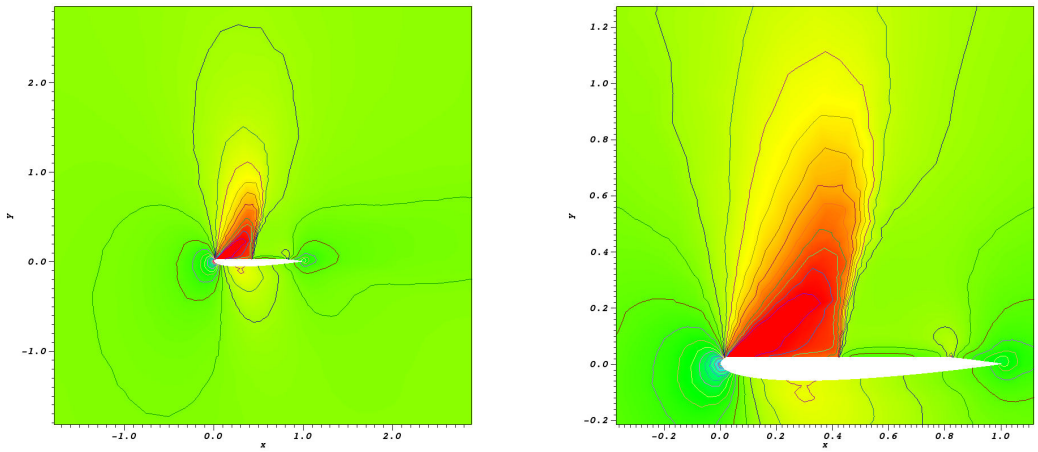


Figure 4.10: The identity mapping velocity component on a flat domain

solution $u(\cdot;\mu)$ is known, we can compute its shock position and slope: $a_0(\mu)$ and $a_1(\mu)$. We use the G-H mapping, see equation (4.12) on both $\hat{\Omega}_R$ and $\hat{\Omega}_L$, and run the modified CFD code. Note that with this preliminary approach, the N^T term is not updated at each pseudo time step. Figure 4.11 shows the resulting solution, that we denote $\hat{u}(\mu)$. One can observe that the general behavior is correct. The shock is more or less located at the correct position and it

has been straightened. In other words, quantitatively we have $u(\mu) \circ \mathrm{GH}\left(a_0(\mu), a_1(\mu)\right) \approx \hat{u}(\mu)$ on $\hat{\Omega}$. This preliminary result is a first answer to the viability of using the Piola transform to construct equivalent problems on reference meshes. Nevertheless, we can see that we have some non physical behavior close to the wing. This could have been anticipated, as the mapping constructed in Section 4.3 suffers major flaws. The biggest problem seems to be at the wing and a consequence of the high gradient at the bottom left corner of the domain of $\hat{\Omega}_L$. We conclude

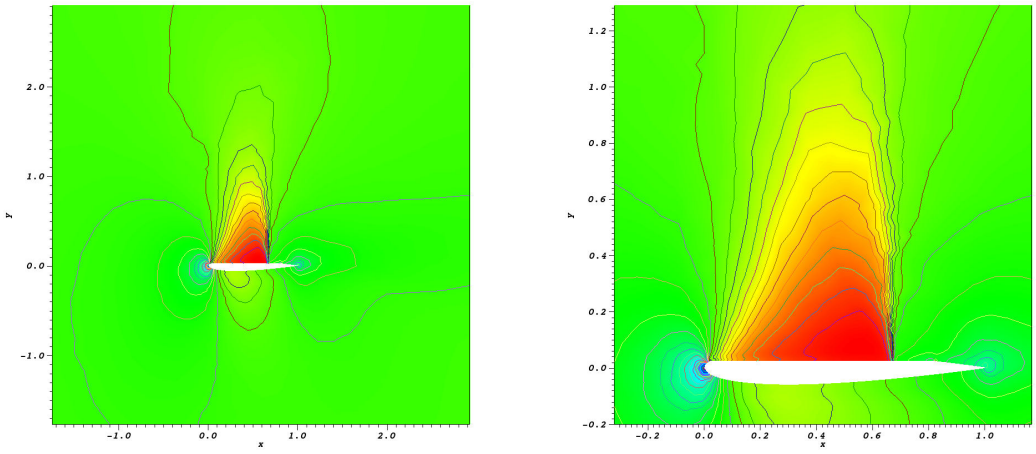


Figure 4.11: The mapped solution for velocity component on a flat domain

that we need a smoother mapping, more than continuous on $\partial\hat{\Omega}_L$ and $\partial\hat{\Omega}_R$.

4.7.2 Mapping on a curved domain

The choice of a flat wing of the previous sections was intentional in order to remind that the reference domain on which we are solving the problem is not the physical one. Nevertheless, because of the lack of smoothness of the resulting mapping, we have decided to use a more advanced mapping than the raw Gordon-Hall. Our starting point is the method developed in [89], applied to the domains depicted in Figures 4.12 and 4.13. As in section 4.3, to enforce continuity of the global mapping, we require that the four corners of reference and physical domain match, i.e we require:

$$\begin{aligned}(x_1, y_1) &= (\hat{x}_1, \hat{y}_1)\\ (x_5, y_1) &= (\hat{x}_5, \hat{y}_1)\\ (x_4, y_2) &= (\hat{x}_4, \hat{y}_2)\\ (x_2, y_2) &= (\hat{x}_2, \hat{y}_2)\end{aligned}$$

4.7.2.1 Original formulation

This extension of the G-H mapping, also called generalized transfinite extension in the literature, has the same structure as the original G-H. For each boundary on the reference domain, $\hat{\Gamma}_i$, we

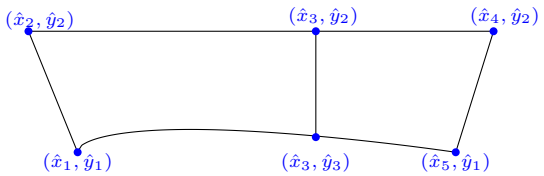


Figure 4.12: Reference domain $\hat{\Omega}$

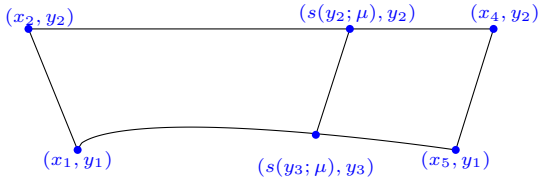


Figure 4.13: Physical domain Ω

need one parametrization of the physical counterpart Γ_i, that is

$$\psi_i \circ \pi_i|_{\hat{\Gamma}_i} : \begin{cases} \hat{\Gamma}_i & \to & \Gamma_i \\ (\hat{x}, \hat{y}) & \mapsto & (x, y) \end{cases}$$

The mapping is then taken as a weighted combination of these mapped boundaries :

$$GH(\hat{x}, \hat{y}) = \sum_{i=1}^{4} [\phi_i(\hat{x}, \hat{y})\psi_i(\pi_i(\hat{x}, \hat{y}), \mu) - \phi_i(\hat{x}, \hat{y})\phi_{i+1}(\hat{x}, \hat{y})\psi_i(1, \mu)]. \tag{4.22}$$

where ϕ_i and π_i are respectively the weight and projection functions associated to $\hat{\Gamma}_i$, see section 4.3.2. The linear weights and projection functions are not an option any more, as the reference domain $\hat{\Omega}$ is not a rectangle.

We will first present the choice of weights and projections proposed in the original version [89]. This was done in a very general case, and the focus was put on the smoothness of the overall mapping. The weights functions are taken as the solutions of the following Laplace problems :

$$\forall i \in [1, \ldots, 4], \begin{cases} -\Delta\phi_i & = & 0 & \text{in } \hat{\Omega} \\ \phi_i & = & 1 & \text{on } \hat{\Gamma}_i \\ \phi_i & = & 0 & \text{on } \hat{\Gamma}_{i+2} \\ \frac{\partial \phi_i}{\partial n} & = & 0 & \text{on } \hat{\Gamma}_{i-1} \cup \hat{\Gamma}_{i+1}. \end{cases} \tag{4.23}$$

The projection functions are also chosen as solutions to a Laplace problem :

$$\forall i \in [1,\dots,4], \begin{cases} -\Delta\pi_i &= 0 & \text{in } \hat{\Omega} \\ \pi_i &= t & \text{on } \hat{\Gamma}_i, t \text{ monotone and smooth} \\ \pi_i &= 1 & \text{on } \hat{\Gamma}_{i+1} \\ \pi_i &= 0 & \text{on } \hat{\Gamma}_{i-1} \\ \frac{\partial \pi_i}{\partial n} &= 0 & \text{on } \hat{\Gamma}_{i+2}. \end{cases} \tag{4.24}$$

Remark 28 on the bijectivity of the resulting mapping still holds in this extended version of the Gordon-Hall method.

4.7.2.2 Additional ingredients

We will now deal with the issues mentioned in Section 4.3 one by one. The smoothness of $\partial\hat{\Omega}$ is solved, with the new choice of $\hat{\Omega}$. Also, we had noticed in our flawed flat approximation that missing to take into account the curvature of the wing represents a too rough approximation. We thus need to chose π_4 and ψ_4 accordingly. The proper way of dealing with this curved boundary is to use the standard arclength definition. For instance, the projection function on the wing π_4 is chosen as :

$$\pi_4|_{\Gamma_4} : (\hat{x},\hat{y}) \rightarrow \int_0^{\hat{x}} \sqrt{1+\left(\frac{\partial w}{\partial \hat{x}}\right)^2}.$$

The same holds for ψ_4.

We also need to be closer to the identity mapping on the left boundary. In the original formulation, homogeneous Neumann boundary conditions are imposed on neighboring edges when computing the projection function, see (4.24). This choice is not the right one for our particular problem. We present in Figure 4.14 on the left, the projection function π_3 in the transfinite version of [89]. Remember, π_3 is the projection onto the edge $\hat{\Gamma}_3$. It is clear that this particular choice deforms the coordinate system. This is one of the causes of the lack of smoothness of the mapping on the left edge $\hat{\Gamma}_1$. The right picture in Figure 4.14 presents π_3 for a better suited boundary condition.

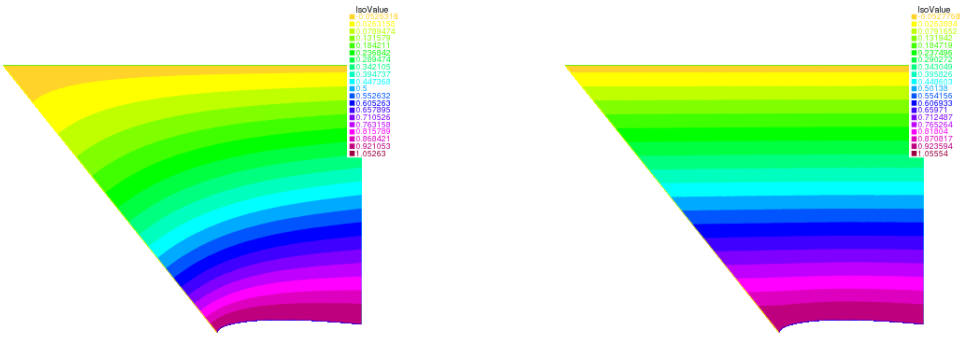


Figure 4.14: Left: π_3 in the original formulation, with homogeneous Neumann boundary condition; Right: π_3 for a more suitable boundary condition

Towards the same objective, we do not want any stretching of the solution around the left boundary and close to the shock. Indeed, we need smooth transitions to neighboring domains.

In order to enforce this, one necessary step is to modify ψ_2 and ψ_4 from the original version. Denote with $H(x)$ some smoothed Heaviside step function. We write it for ψ_2, but note that the same can be done for ψ_4. We pick the following :

$$\tilde{\psi}_2(\hat{x},\hat{y},\mu) = \pi_2(\hat{x},\hat{y})\cdot\frac{\hat{x}_3-\hat{x}_2}{s(y_2;\mu)-x_2}\cdot(1-H(\pi_2(\hat{x},\hat{y})))+\left(1+(\pi_2(\hat{x},\hat{y})-1)\cdot\frac{\hat{x}_3-\hat{x}_2}{s(y_2;\mu)-x_2}\right)\cdot H(\pi_2(\hat{x},\hat{y})).$$

That is, we want no stretching for $\pi_2(\hat{x},\hat{y}) \approx 0$ or 1. A graphical illustration is presented on the left picture of Figure 4.15 for an hypothetical stretching of 4/3. The dashed red lines correspond to a non stretched mapping.

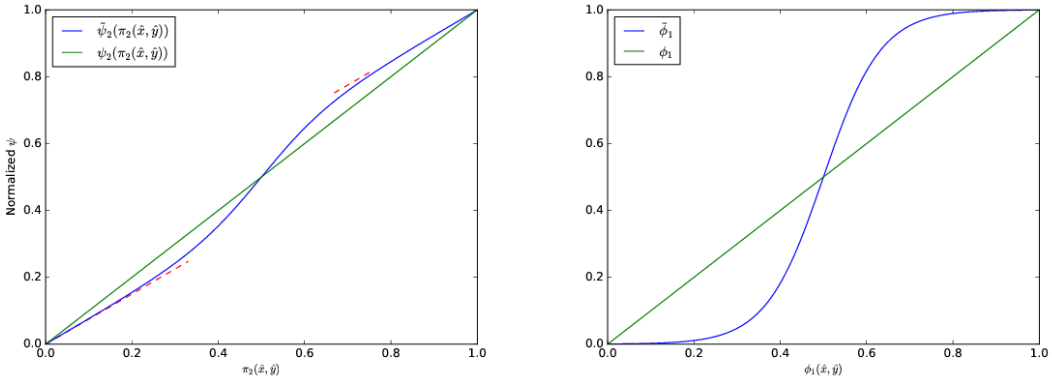


Figure 4.15: Modification of weights and projection functions to get smoother transitions on $\hat{\Gamma}_1$ and $\hat{\Gamma}_3$

We will modify one more ingredient. We take steeper weight functions for boundaries 1 and 3. For instance, we can pick:

$$\tilde{\phi_1}(\hat{x},\hat{y}) = H\left(\phi_1(\hat{x},\hat{y})\right),$$

where the ϕ_1 is the solution the the Laplace problem, see (4.23). This is presented on the right picture of Figure 4.15. What this achieves is that close to left boundary, the exact shape of the right physical boundary has no influence, and the converse.

Finally, we choose the following set of $\{\psi_i,\ i \in [1,\dots,4]\}$:

$$\begin{aligned}
\psi_1(\pi_1(\hat{x},\hat{y}),\mu) &:= \Big(x_1+\pi_1(\hat{x},\hat{y})\cdot(x_2-x_1), y_1+\pi_1(\hat{x},\hat{y})\cdot(y_2-y_1)\Big)\\
\psi_2(\pi_2(\hat{x},\hat{y}),\mu) &:= \Big(x_2+\pi_2(\hat{x},\hat{y})\cdot(s(y_2;\mu)-x_2), y_2\Big)\\
\psi_3(\pi_3(\hat{x},\hat{y}),\mu) &:= \Big(s(y_2+\pi_3(\hat{x},\hat{y})\cdot(y_3-y_2);\mu), y_2+\pi_3(\hat{x},\hat{y})\cdot(y_3-y_2)\Big)\\
\psi_4(\pi_4(\hat{x},\hat{y}),\mu) &:= \Big(arclen^{-1}(\pi_4(\hat{x},\hat{y})), y_3\Big)
\end{aligned}$$

Remark 33 *The offline/online decomposition of the global method will strongly depend on the way we pick the set of ϕ_i's, π_i's and ψ_i's.*

Remark 34 *This smarter choice of functions not only makes the G-H mapping smoother but it also makes*

$$\begin{array}{ccc} \mathcal{D} & \to & \mathcal{F} \\ \mu & \mapsto & F_\mu \end{array}$$

smoother. This can be an interesting property in a optimal control context (see chapter 5).

To assess the gain of this more advanced mapping, we perform the same test as in subsection 4.7.1, this time with the new and improved G-H mapping, given by equation (4.22). The output solution $\hat{u}(\mu)$ is presented in Figure 4.16. As for the results obtained with the original G-H, the overall behaviour is correct as $\hat{u}(\mu)$ has a similar shape as $u(\mu) \circ \mathrm{GH}(a_0(\mu), a_1(\mu))$. The novelty is that we have managed to remove the non physical behavior at the boundary that we had in the raw G-H scenario, Figure 4.11.

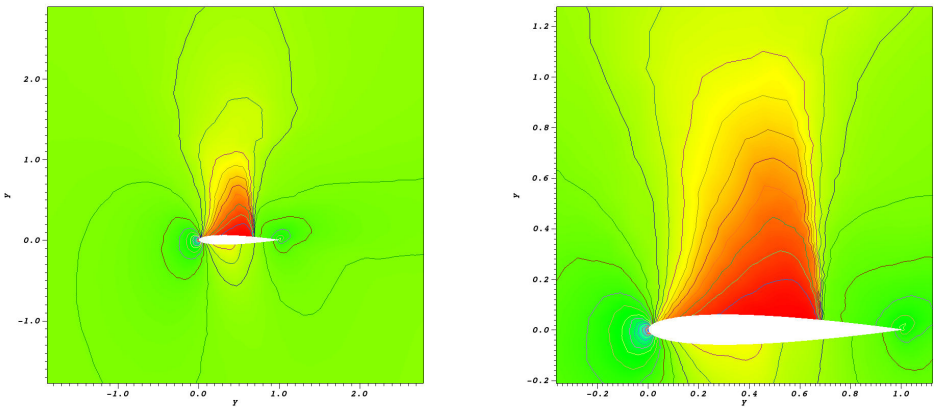


Figure 4.16: The mapped solution for velocity component on a curved domain

Remark 35 *One must not forget that this case is no different from the flat boundary scenario of subsection 4.7.1. The fact that the reference domain has the same body as the physical domain is required for smoothness purposes only.*

Before a more involved test run, we present yet another improvement. This goes one step further in building a smooth mapping at the boundaries. One recent development on transfinite maps is defined in [75] and is called boundary displacement dependent transfinite map (BDD TM). The idea is not to construct the whole mapping, but to construct a relative displacement with respect to the identity. Most of the method is the same, the only difference is that instead of ψ_i function, which represent the position on the physical domain, a new function $d_i : [0,1] \times \mathcal{D} \to \mathbb{R}$ is introduced and it will represent the displacement:

$$d_i(t, \mu) = \psi_i(t, \mu) - \hat{\psi}_i(t).$$

Each of the boundaries in the reference domain is parametrized by $\hat{\psi}_i : [0,1] \to \mathbb{R}$. Like this, the mapping will take into account the original positions of the points in the reference domain $\hat{\Omega}$ and will move them by weighting only the difference between the original boundaries and the deformed ones. Let $(\hat{x}, \hat{y})$ be a point in the reference domain $\hat{\Omega}$, the idea of BDD TM is

to displace it through the quantity $(\hat{x},\hat{y}) + \sum_{i=1}^{n} \phi_i(\hat{x},\hat{y}) d_i(\pi_i(\hat{x},\hat{y}),\mu)$. In the end, the BDD transfinite mapping is defined as:

$$GH_{BDDTM}(\hat{x},\hat{y}) = (\hat{x},\hat{y}) + \sum_{i=1}^{n} \Big(\phi_i(\hat{x},\hat{y}) d_i(\pi_i(\hat{x},\hat{y}),\mu) - \phi_i(\hat{x},\hat{y})\phi_{i+1}(\hat{x},\hat{y}) d_i(1,\mu) \Big) \tag{4.25}$$

This has one major effect, on the left boundary for instance, where we want zero displacement. The resulting mapping restricted to a neighborhood of this boundary will be the identity, which guarantees overall smoothness.

Remark 36 *The improvements on ϕ's and ψ's presented for the TM method still apply to the BDD TM.*

After this long preamble, we are ready to illustrate numerically the gain obtained from the methods just described. We present in Figure 4.17 the comparaison between the original method, section 4.7.2.1 and the pimped one, taylored for our specific application. We show one of the entries of N^T. We have picked the one varying the most i.e $\frac{\partial x}{\partial \hat{x}}$. It is obvious that the preivous improvements to G-H have helped for the smoothness between neighboring domains.

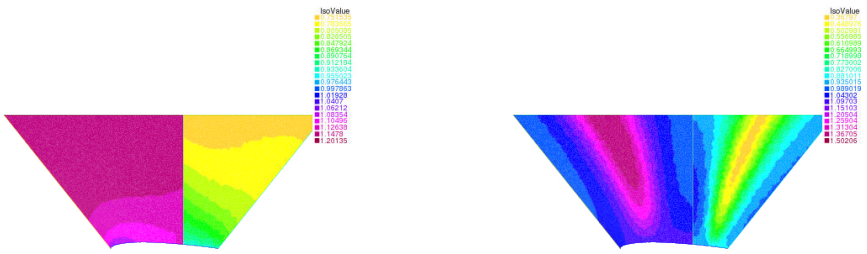


Figure 4.17: One of the entries of the Jacobian matrix, namely $(J_{F_n^{-1}})_{11}$. Left: with no additional smoothing ingredients; Right: with some smoothing ingredients

4.7.3 Final experiment

What is presented in this section does not correspond to any actual step of the online section. The purpose is to provide a more quantitative result on the utilization of the Piola transform for resolution of a problem on a reference mesh. For this, we have chosen to perform the following test:

- pick a small number of pairs $\{(a_0,a_1)\}$ and construct the corresponding G-H mappings: $\{\text{G-H}(a_0,a_1)\}$
- as in the previous subsection, launch the CFD code, using the modified flux and boundary conditions. Denote the output $\hat{u}(a_0,a_1)$ for each mapping G-H(a_0,a_1). Once again, the mapping is not updated at each pseudo time step

- compare the output with the mapped 'truth' solution, i.e compare

$$u \circ (\text{G-H}(a_0, a_1)) \text{ with } \hat{u}(a_0, a_1)$$

We have chosen a simple comparison criteria: the position and slope of the shock. We present the results for two pairs (a_0, a_1) in Figure 4.18. Blue represents the shock of the truth solution mapped onto the reference domain, $u \circ \text{GH}(a_0, a_1)$. Red is the shock of $\hat{u}(a_0, a_1)$. Green is the position of the shock of u, and has been plotted for control purposes. We have fitted one degree polynomials through each shock. The discrepancy on the left picture, between the output of the

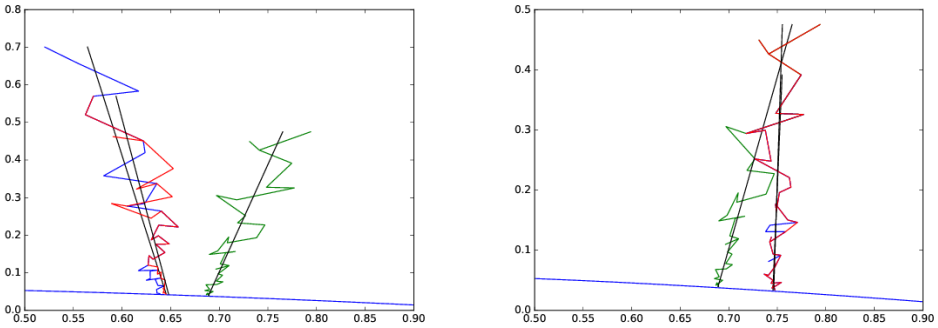


Figure 4.18: Comparison of the outputs

modified CFD code and the mapped truth scheme can be due to many factors:

- numerical errors on the computation of the N^T terms;
- the SUPG stabilization has not been touched, to avoid too much intrusion in the code. This means that we are not using the same stabilization procedure as the truth scheme. We refer to [92] for a study of this situation. They advise using an a posteriori procedure, called rectification;
- our method to locate the shock is basic. We would need something more involved to quantify the error

4.8 Conclusion

The purpose of this chapter was to propose a complete calibration procedure to make standard ROM methods fitted for solving the two dimensional Euler equation around an airfoil. We have proposed an offline calibration procedure, have shown that it leads to better behaved, non oscillatory basis. We have then proposed a fully functioning reduced scheme. The computational complexity and the optimization procedures have been theoretically studied. We have finally developed numerical experiments that serve as a proof of concept for the global method.

Most of the stages in this chapter can be further investigated. Future work could involve :

- a deeper study of the offline calibration and its effect on the Kolmogorov n-width. We have proposed online some advanced mappings, where we impose no stretching in the vicinity of the shock. A numerical investigation has to be conducted

- the construction of fully reduced scheme, with the procedure advised in Section 4.4. This could lead to an interesting comparison between L^1 and L^2 minimization. Also, the optimization procedure can be studied. The differentiability issues can be also tested.
- the hyper reduction procedure can be numerically investigated. The conjectures made on the resulting $\hat{\Omega}_{hyper}$, namely that the interesting control volumes are close to the shock, which is fixed in $\hat{\Omega}$ can be tested
- the smoothness of the entries of the N^T matrix has to be studied. All the smoothing ingredients proposed in section 4.7 could be further investigated.
- a more long term objective could be to use this method to try new airfoils shapes. As mentioned, the G-H is a very flexible algorithm. We could use the NACA 00012 as reference domain, but use fantasist choices for physical domains. This is a well suited framework for optimal control (for more details, see chapter 5)

Chapter 5

Calibration for optimal control problems: illustration on solutions to hyperbolic problems with shocks

---

In this chapter, we have chosen to use the calibration ideas introduced in chapter 3 to help solving optimal control problems. The starting point was to notice that calibration was increasing in some sense the regularity of the solution's dependency to the parameter at hand. We have chosen to illustrate this idea by studying the optimal control of an hyperbolic equation, for solutions with shocks. We start this chapter by showing how optimal control is usually performed. Of course, this procedure depends on the regularity of the solution with respect to the parameter, and is dubious for solutions with shocks. We thus propose an optimal control method that uses calibration, and illustrate it on the one dimensional inviscid Burgers equation. We conclude by sketching a procedure to solve the optimal design of a NACA airfoil.

The title of this chapter should not mislead the reader. The ambition of this chapter on the optimal control side is modest. The point of view adopted here is rather to try and use the calibration ideas developed in the previous chapters, as well as reduced order modeling, in an optimal control context. As already mentioned several times in this manuscript, optimal control is a natural application of ROM and there is an extensive literature on the subject, see for instance [80, 108]. In this chapter, we add a calibration process to the resolution of an optimal control problem, for an hyperbolic equation with solutions with shocks. In that setting, it serves two purposes. The first one has has already been discussed in the introductive chapter 1. The calibration helps enforce regularity to both the solution and some numerical schemes with respect to the parameter, see section 1.5.2. The second one is that the domains where the calibrated solution remains smooth are constant in time and constant parameter wise. We can thus build reduced basis on each domain and perform the classical model order reduction.

We start by introducing optimal control on a simple elliptic example. We insist on all the properties required to get to a well posed problem. We then follow these steps for an hyperbolic problem. We show the major differences caused by the existence of discontinuities whose position is function of the parameter. We finally use calibration and show that it is an adapted framework as it enforces smoothness with respect to the parameter in a way that can be handled by ROM.

Most of the ideas are illustrated using the inviscid one dimensional Burgers equation with periodic boundary conditions. We end this section by roughly stating a way of using this idea to perform optimal design of an airfoil.

5.1 What is optimal control

Let $\mathcal{U}_{ad}$ be some admissible parameter set. We consider the following time dependent parametrized PDE

$$\frac{\partial u(\cdot,t;\mu)}{\partial t} + \mathcal{L}(u(\cdot,t;\mu)) = f(\mu), \tag{5.1}$$

supplemented with proper boundary and initial condition. Suppose also that we are given some objective functional

$$\left\{ \begin{array}{lll} X \times \mathcal{U}_{ad} & \rightarrow & \mathbb{R} \\ (u,\mu) & \mapsto & J(u,\mu). \end{array} \right.$$

We are interested in the solution of an optimization problem of the form:

$$\inf_{\mu \in \mathcal{U}_{ad}} J(u(\mu),\mu)$$

where $u(\mu)$ is the solution to equation (5.1).

What are the formal steps ?

- make sure that equation (5.1) has good properties. The problem has to be well posed and there needs to be some form of regularity on $\mu \rightarrow u(\mu)$
- pick some minimizing sequence μ_n. Use hypothesis on $\mathcal{U}_{ad}$ to show that μ_n converges, in some sense, towards a limit $\mu^* \in \mathcal{U}_{ad}$
- use the regularity $\mu \rightarrow u(\mu)$ to prove that $u(\mu_n)$ converges to $u(\mu^*)$ in some sense
- use hypothesis on J to conclude on the existence of minimizers:

  $$J(u(\mu^*),\mu^*) = \inf_{\mu \in \mathcal{U}_{ad}} J(u(\mu),\mu).$$

  A bonus property is the uniqueness of the minimizer
- look for (one of) the minimizers. This often is done using some gradient descent algorithm, and involves differentiating $\mu \rightarrow J(u(\mu),\mu)$

In this chapter, we focus solely on the last step.

5.1.1 Introductive elliptic case

To give a better understanding of these formal steps, we apply them to a simple example. Let Ω be some domain and $\Gamma := \partial\Omega$. We are interested by the solutions of the following elliptic problem:

$$\left\{ \begin{array}{lll} -\Delta u + u & = & f(\mu) \text{ in } \Omega \\ \frac{\partial u}{\partial n} & = & g(\mu) \text{ on } \Gamma. \end{array} \right.$$

The objective functional is chosen as

$$J(u,\mu) = \frac{1}{2}\int_\Omega (u-u^d)^2 + \frac{\kappa}{2}\int_\Gamma g(\mu)^2.$$

Suppose that $\mathcal{U}_{ad}$ is embedded in some normed linear space, that

$$\begin{array}{l} \forall \mu \in \mathcal{U}_{ad}, \quad f(\mu) \in L^2(\Omega) \\ \forall \mu \in \mathcal{U}_{ad}, \quad g(\mu) \in L^2(\Gamma) \end{array}$$

and that f and g have continuous dependence on μ. From Lax-Milgram, we have:

$$\begin{array}{lll} \mathcal{U}_{ad} & \rightarrow & H^1(\Omega) \\ \mu & \mapsto & u(\mu) \end{array}$$

is well defined. The hypothesis on $\mu \rightarrow f(\mu)$ and $\mu \rightarrow g(\mu)$, and the linearity are sufficient to show the continuity of this application.

Let $\{\mu_n\}$ be a minimizing sequence in $\mathcal{U}_{ad}$. We suppose that $\mathcal{U}_{ad}$ is chosen such that there is a subsequence converging to μ^*. From the continuity, we know that $u(\mu_n) \rightarrow u(\mu^*)$ in $H^1(\Omega)$. To conclude that μ^* is a minimizer of J, we need: $J(u(\mu_n), \mu_n) \rightarrow J(u(\mu^*), \mu^*)$. We develop:

$$\begin{array}{rl} \forall n, \ J(u(\mu_n), \mu_n) - J(u(\mu^*), \mu^*) = & \frac{1}{2} \int_\Omega (u(\mu_n) - u(\mu^*))(u(\mu_n) + u(\mu^*) - 2u^d) \\ & + \frac{\kappa}{2} \int_\Gamma (g(\mu_n) - g(\mu^*))(g(\mu_n) + g(\mu^*)) \end{array}$$

This concludes the existence of a minimizer μ^* of J. The strict convexity of $u \rightarrow J(u)$ guarantees the uniqueness.

The next topic on the line is the computation of $\frac{\partial J}{\partial \mu}$. Formally,

$$\frac{\partial J}{\partial \mu} = \int_\Omega (u - u^d) \frac{\partial u}{\partial \mu} + \kappa \int_\Gamma \frac{\partial g}{\partial \mu} g.$$

Can we give a meaning to $\frac{\partial u}{\partial \mu}$? This requires new assumptions on the regularity of $\mu \rightarrow f(\mu)$ and $\mu \rightarrow g(\mu)$. Namely, we assume that f and g are Frechet differentiable and denote A_μ and B_μ their derivative. Denote $\partial_\mu u$ the application from $\mathcal{U}_{ad} \rightarrow H^1(\Omega)$ given by:

$$\partial_\mu u : \delta\mu \rightarrow \text{ the solution to } \left\{ \begin{array}{lll} -\Delta w + w & = & A_\mu \delta\mu \\ \frac{\partial w}{\partial n} & = & B_\mu \delta\mu \end{array} \right.$$

We can easily show that $\partial_\mu u$ is the Frechet derivative of u. We have now a rigorous definition of the Frechet derivative $\partial_\mu J$:

$$\partial_\mu J : \left\{ \begin{array}{lll} \mathcal{U}_{ad} & \rightarrow & \mathbb{R} \\ \delta_\mu & \mapsto & \int_\Omega (u - u^d) \partial_\mu u \delta\mu + \kappa \int_\Gamma g B_\mu \delta\mu \end{array} \right.$$

The standard way of looking for the minimizer, is to express $\partial_\mu J$ as a function of the solution, denoted p, to the dual equation. This avoids having to solve for $\partial_\mu u$ directly. The dual equation is here given by:

$$\left\{ \begin{array}{lll} -\Delta p + p & = & u - u^d \quad \text{in } \Omega \\ \frac{\partial p}{\partial n} & = & 0 \quad \text{on } \partial\Omega. \end{array} \right.$$

We start with the first term of $\partial_\mu J$:

$$\begin{array}{rl} \int_\Omega (u - u^d) \partial_\mu u \delta\mu & = \int_\Omega (-\Delta p + p) \partial_\mu u \delta\mu \\ & = \int_\Omega p(-\Delta \partial_\mu u \delta\mu + \partial_\mu u \delta\mu) + \int_\Gamma p B_\mu \delta\mu \\ & = \int_\Omega p A_\mu \delta\mu + \int_\Gamma p B_\mu \delta\mu. \end{array}$$

Wrapping everything up, we get the following expression for $\partial_\mu J$:

$$\partial_\mu J(\delta_\mu) = \int_\Omega pA_\mu \delta\mu + \int_\Gamma (p + \kappa g) B_\mu \delta\mu$$

This gives a direction for a standard gradient descent algorithm. For instance, taking $A_\mu \delta\mu$ to be 'close' to p over Ω is a natural strategy.

As we will see in the next sections, some of the steps above are not as easy for more challenging problems, and different parameter dependencies. Next section is devoted to the study of some hyperbolic equations in the presence of shocks. In that case, the derivative of $\mu \to u(\mu)$ in the sense of distributions has a dirac mass. Of course, this has consequence on the smoothness of $\mu \to J(\mu)$. Lastly, we have to be more careful with the dual equation.

5.2 Burgers equation

We first focus on one favorable case: Burgers equation with periodic boundary conditions. We have chosen to restrict the study to a parameter dependency on the initial condition. That is, we are solving

$$\left\{ \begin{array}{lcll} \partial_t u + \partial_x(\frac{u^2}{2}) & = & 0 & \text{on } \Omega \times [0,T] \\ u(t=0,\cdot) & = & u_0(\mu) & \text{on } \Omega \\ u \text{ periodic} & & & \end{array} \right. \tag{5.2}$$

for μ varying in some parameter space once again denoted $\mathcal{U}_{ad}$.

At the core of both optimal control and ROM, there is some form of regularity on $\mu \to u(\cdot,\cdot;\mu)$. It will not be as obvious as in the elliptic case. We assume that:

- $\mathcal{U}_{ad}$ and $\mu \to u_0(\mu)$ are chosen so that the solutions are $L^2(\Omega)$
- $\mu \to u_0(\mu)$ is smooth
- for all μ and all times, the solutions considered have at most one discontinuity. Its position for time t and parameter μ will be denoted $\phi(t;\mu)$
- $(\mu, t) \to \phi(t;\mu)$ is smooth. We will see in a later section that this property is guaranteed by the Rankine Hugoniot condition and by the smoothness of the solution away from the shock.

5.2.1 Objective function

We work with the following objective functional:

$$J : \left\{ \begin{array}{lcl} \mathcal{U}_{ad} & \to & \mathbb{R} \\ \mu & \mapsto & \frac{1}{2}\int_\Omega \left| u(x,T;\mu) - u^d(x) \right|^2 dx \end{array} \right. \tag{5.3}$$

where $u(\cdot,\cdot;\mu)$ is the entropic solution of (5.2) with initial condition $u_0(\mu)$.

Remark 37 *The precise form of the objective function will have a big importance in the following section. The new ingredient that will be introduced in section 5.3 will allow us to consider a wider range of J.*

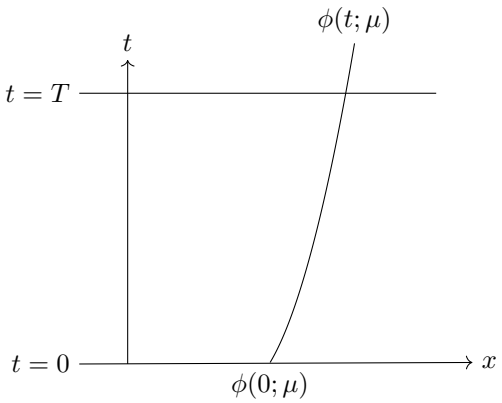


Figure 5.1: We restrict ourselves to solutions with at most one shock

We give the steps of the proof of the existence of a minimizer in that setting, and refer to [32] for a complete description. The ingredients are quite different from the elliptic case, as the solutions are not in the same functional spaces. Let $\{\mu_n\}$ be a minimizing sequence, and $\{u(\cdot,\cdot;\mu^n)\}$ the corresponding entropic solutions,

- suppose $\mathcal{U}_{ad}$ is chosen such that we can extract a subsequence that converges towards a μ^*, in some sense.
- we use Oleinik one sided Lipschitz condition:

$$\forall \mu, \forall t, \ \frac{u(x,t;\mu) - u(y,t;\mu)}{x-y} \leq \frac{1}{t}$$

  to extract from $\{u(x,t;\mu^n)\}_n$ a subsequence that converges in $L^2(\Omega)$
- proving that the limit is $u(\cdot,\cdot;\mu^*)$ concludes the proof.

The focus of this chapter is put on studying the differentiability of J. For simplicity, we restrict ourselves to the case where $\mathcal{U}_{ad}$ is a compact subset of $\mathbb{R}^M$ for some M. That is, we suppose that the variations on u_0 can be described by a moderate number of parameters. More general parameter dependence would not impact the method proposed here, but would just complicate the notation. The generic $\partial_\mu J$ that will be used in the rest of the chapter has to be understood as the derivative with respect to one of the M parameters.

Formally differentiating the objective function, see equation (5.3), gives:

$$\partial_\mu J = \int_\Omega \partial_\mu u(x,T;\mu) \left(u(x,T;\mu) - u^d(x)\right) dx.$$

Questions arise:

- can we give a meaning to $\partial_\mu u$ away from the shock ?
- as $u(\cdot,T;\mu)$ is discontinuous, we can anticipate that the derivative of u in the sense of distributions will involve a dirac mass. How does this influence $\partial_\mu J$?
- how do we compute $\partial_\mu J$ efficiently ? is the dual method described in the simple elliptic equation still relevant ?

5.2.2 Smoothness away from the shock

We show in this section that the derivative of the solution with respect to the parameter is smooth, away from the shock. This is an important property and will be used constantly in this chapter. We use the following notation:

$\forall s,\ t,\ \forall x,\ Z^{\mu}_{s,t}(x)$ is the position at time t of the characteristics that passes through (x, s).

For instance, for fixed $x \in \Omega$, $Z^{\mu}_{0,t}(x)$ is the position at time t of the characteristics that started at time 0 at x. For classical solutions to Burgers equation, i.e for solutions with no shocks, it is well known that the characteristics are straight lines:

$$\forall \mu \in \mathcal{U}_{ad},\ \forall t \in [0,T],\ Z^{\mu}_{0,t} : \left\{ \begin{array}{ccc} \Omega & \to & \Omega \\ x & \mapsto & x + t u_0(\mu)(x). \end{array} \right.$$

For simplicity, we start by restricting the analysis to the set of classic solutions. Fix some $(x,t) \in \Omega \times [0,T]$. The first step is to prove that $\mu \to Z^{\mu}_{t,0}(x)$ is smooth. This function returns the position of the foot of the characteristics that gets to (x,t) when the parameter μ varies. The situation is illustrated in Figure 5.2.

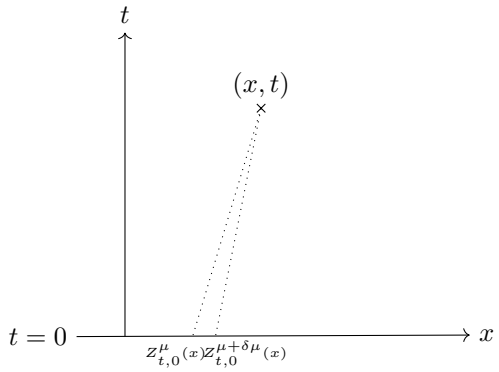


Figure 5.2: Away from the shock, the foot of the characteristics are close

Let $\delta\mu \in \mathbb{R}$. As we are only considering classical solutions, we know:

$$\left\{ \begin{array}{l} x = Z^{\mu}_{t,0}(x) + t\ u_{0,\mu}(Z^{\mu}_{t,0}(x)) \\ x = Z^{\mu+\delta\mu}_{t,0}(x) + t\ u_{0,\mu+\delta\mu}(Z^{\mu+\delta\mu}_{t,0}(x)). \end{array} \right.$$

So we have

$$Z^{\mu+\delta\mu}_{t,0} - Z^{\mu}_{t,0} = t u_{0,\mu}(Z^{\mu}_{t,0}) - t u_{0,\mu+\delta\mu}(Z^{\mu+\delta\mu}_{t,0}). \tag{5.4}$$

Decompose the right hand side of (5.4) into two quantities:

- the foot of the characteristics are not the same:

$$t u_{0,\mu+\delta\mu}(Z^{\mu}_{t,0}) - t u_{0,\mu+\delta\mu}(Z^{\mu+\delta\mu}_{t,0})$$

- the velocities of the characteristics are not the same:

$$u_{0,\mu}(Z^{\mu}_{t,0}) - u_{0,\mu+\delta\mu}(Z^{\mu}_{t,0}) = \delta\mu \partial_{\mu} u^{\mu}_0(Z^{\mu}_{t,0}) + O(\delta\mu)^2$$

Equation (5.4) becomes

$$Z_{t,0}^{\mu+\delta\mu} - Z_{t,0}^{\mu} = tu_{0,\mu+\delta\mu}(Z_{t,0}^{\mu}) - tu_{0,\mu+\delta\mu}(Z_{t,0}^{\mu+\delta\mu}) + t\delta\mu\partial_\mu u_{0,\mu}(Z_{t,0}^{\mu}) + O(\delta\mu)^2.$$

This gives:

$$Z_{t,0}^{\mu+\delta\mu} - Z_{t,0}^{\mu} = -t(Z_{t,0}^{\mu+\delta\mu} - Z_{t,0}^{\mu})\partial_x u_{0,\mu}(Z_{t,0}^{\mu}) + O(Z_{t,0}^{\mu+\delta\mu} - Z_{t,0}^{\mu})^2 + t\delta\mu\partial_\mu u_{0,\mu}(Z_{t,0}^{\mu}) + O(\delta\mu)^2. \quad (5.5)$$

We need one more argument before we can conclude on the differentiability of $\mu \to Z_{t,0}^{\mu}(x)$, and it will be given by the fact that no shock is created in the vicinity of the characteristics considered. This is illustrated in Figure 5.3. The thick line is the characteristic for $u(\cdot,\cdot;\mu)$ that passes through (x,t). The foot of the characteristic is thus $Z_{t,0}^{\mu}(x)$. The dashed line is the characteristic of $u(\cdot,\cdot;\mu+\delta\mu)$, emitted from $Z_{t,0}^{\mu}(x)$. The dotted line is the parallel of this characteristic that passes through (x,t). What can we say about $Z_{t,0}^{\mu+\delta\mu}(x)$? It cannot be one the left of the dashed line. Otherwise, two characteristics of $u(\cdot,\cdot;\mu+\delta\mu)$ would intersect, thus leading to the formation of a shock. A similar arugment prevents it from beeing on the right of the dotted line. The foot of the dotted line Z_{dotted} is given by:

$$x = Z_{dotted} + tu_{0;\mu+\delta\mu}(Z_{t,0}^{\mu}(x)).$$

With this, we can bound $Z_{t,0}^{\mu+\delta\mu} - Z_{t,0}^{\mu}$:

$$\begin{aligned} |Z_{t,0}^{\mu+\delta\mu}(x) - Z_{t,0}^{\mu}(x)| &\leq |Z_{t,0}^{\mu}(x) - Z_{dotted}| \\ &\leq \left|tu_{0;\mu}(Z_{t,0}^{\mu}(x)) - tu_{0;\mu+\delta\mu}(Z_{t,0}^{\mu}(x))\right| \\ &\leq T|\delta\mu\ \partial_\mu u_{0;\mu}| + O(\delta\mu)^2. \end{aligned}$$

This hypothesis that no shock is created is necessary in order to prove the differentiability of $\mu \mapsto Z_{t,0}^{\mu}(x)$.

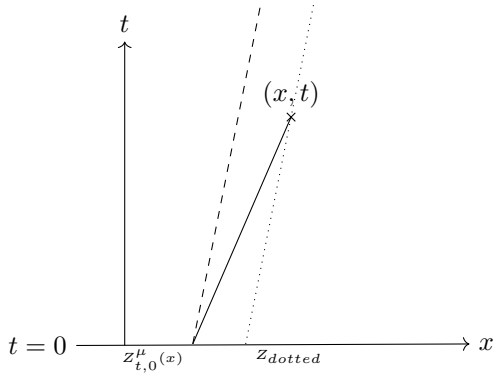


Figure 5.3: Size of $\left|Z_{t,0}^{\mu+\delta\mu} - Z_{t,0}^{\mu}\right|$

We can now go back to equation (5.5). For fixed $x \in \Omega$ and $t \in [0,T]$, we have: $\mu \to Z_{t,0}^{\mu}(x) \in C^1(\mathcal{U}_{ad})$ and its derivative is given by:

$$\partial_\mu Z_{t,0}^{\mu}(x) = \frac{t\partial_\mu u_{0,\mu}(Z_{t,0}^{\mu}(x))}{1 + t\partial_x u_{0,\mu}(Z_{t,0}^{\mu}(x))}. \quad (5.6)$$

Remark 38 *The upper bound on t suggested by the equation (5.6) is the same as the one appearing when discussing the breakdown of classical solutions.*

We go back to the initial objective: the smoothness of

$$\begin{array}{rcl} \mathcal{U}_{ad} & \to & L^2(\Omega \times [0,T]) \\ \mu & \mapsto & u(\cdot,\cdot;\mu) \end{array}$$

for classical solutions. We use the fact that the solutions considered are constant on the characteristics:

$$u(x,t;\mu+\delta\mu) - u(x,t;\mu) = u_{0,\mu+\delta\mu}(Z^{\mu+\delta\mu}_{t,0}) - u_{0,\mu}(Z^{\mu}_{t,0},0)$$

Once again, we decompose the right hand side into two quantities:

$$\begin{cases} u_{0,\mu+\delta\mu}(Z^{\mu+\delta\mu}_{t,0}) - u_{0,\mu+\delta\mu}(Z^{\mu}_{t,0}) & \text{the foot of the two characteristics are close} \\ u_{0,\mu+\delta\mu}(Z^{\mu}_{t,0}) - u_{0,\mu}(Z^{\mu}_{t,0}) & \text{the transported informations are close} \end{cases}$$

We use the previous result on the smoothness of $\mu \to Z^{\mu}_{t,0}$:

$$\frac{u(x,t;\mu+\delta\mu) - u(x,t;\mu)}{\delta\mu} = \partial_\mu Z^{\mu}_{t,0}\ \partial_x u_{0,\mu}(Z^{\mu}_{t,0}) + \partial_\mu u_{0,\mu}(Z^{\mu}_{t,0}) + O(\delta\mu)$$

We conclude that for fixed (x,t), $\mu \to u(x,t;\mu)$ is smooth. By inspecting the actual form of the derivative, we conclude that $\mu \mapsto u(\cdot,\cdot;\mu)$ is a smooth function $\mathcal{U}_{ad} \to L^2(\Omega \times [0,T])$.

The previous analysis has been done for classical solutions. We provide formal arugments to show that this result extends to solutions with shock, but away from the shock. Let $(x,t) \in \Omega \times [0,T]$ away from the shock for parameter μ, i.e such that $x \neq \phi(t;\mu)$. As $(t,\mu) \mapsto \phi(t;\mu)$ is a smooth function, we know that there is a neighborhood of μ, say $\mathcal{V}(\mu)$, such that:

$$\forall \mu' \in \mathcal{V}(\mu),\ u(\cdot,\cdot;\mu')\text{'s shock is away from } (x,t).$$

The argument illustrated in Figure 5.3 then applies just as for classical solutions. We conclude that:

$$\begin{array}{rcl} \mathcal{U}_{ad} & \to & L^2\left((\Omega \setminus \text{neighborhood of the shock }) \times [0,T]\right) \\ \mu & \mapsto & u(\cdot,\cdot;\mu) \end{array}$$

is smooth, for solutions satisfying the hypothesis described in the begining of section 5.2.

5.2.3 Control with no calibration

We need to introduce new notations before going further in the analysis. For all μ, denote $\sigma(\mu)$ the locus of the $\{(\phi(t;\mu), t \in [0,T]\}$. It is a smooth curve. Let $\Omega^-(\mu)$ and $\Omega^+(\mu)$ be a non overlapping cover of Ω such that $\sigma(\mu)$ is one of their common boundaries. For instance, for $\Omega = (0,1)$, one can take $\Omega^-(\mu) := \big((\phi(t;\mu) - \frac{1}{2})\%1, \phi(t;\mu)\big) \times [0,T]$, and $\Omega^+(\mu) := \big(\phi(t;\mu), (\phi(t;\mu) + \frac{1}{2})\%1\big) \times [0,T]$. We also need the following notations:

- $Q^- = \bigcup_\mu \Omega^-(\mu)$
- $Q^+ = \bigcup_\mu \Omega^+(\mu)$
- $\Sigma = \bigcup_\mu \sigma(\mu)$. By hypothesis, it is a smooth surface in $\mathbb{R}^3$.

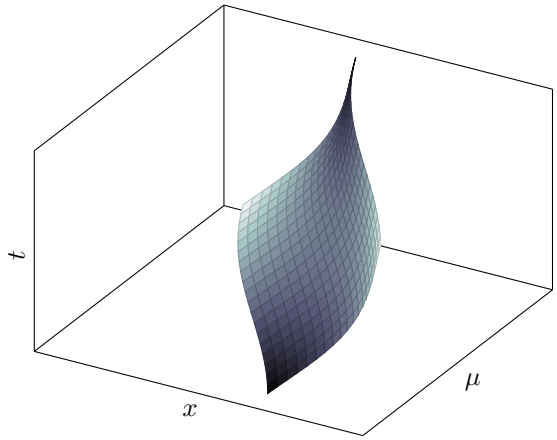


Figure 5.4: $\Omega \times [0,T] \times \mathcal{U}_{ad}$ in the non calibrated case

One graphical illustration is presented in Figure 5.4. As we are dealing with discontinuous solutions, we will use:

- $[u]_\sigma = u^+|_\sigma - u^-|_\sigma$ the standard jump operator
- $\bar{u}_\sigma = \frac{1}{2}(u^+|_\sigma + u^-|_\sigma)$ the average operator

For fixed μ, the normals to the smooth curve $\sigma(\mu)$ in $\Omega \times [0,T]$ will be denoted:

$$\nu := (\nu_x, \nu_t) = \frac{1}{\sqrt{1+(\partial_t \phi)^2}}(1, -\partial_t \phi).$$

We want to find the derivative in the sense of distributions of $\partial_\mu u(x,t;\mu)$. Let $Q = \Omega \times [0,T] \times U_{ad}$ and $w \in \mathcal{D}(Q)$:

$$\int_Q u(x,t;\mu)\partial_\mu w = -\int_{Q^+} \partial_\mu u(x,t;\mu)w - \int_{Q^-} \partial_\mu u(x,t;\mu)w + \int_\Sigma [u]_{\phi(t;\mu)}\partial_\mu \phi(t;\mu)w.$$

The derivative of u with respect to μ can thus be decomposed into one function, smooth by parts, denoted $\partial_\mu u^p$, and a measure concentrated on the surface Σ. More precisely, it is given by:

$$\partial_\mu u(x,t;\mu) = \partial_\mu u^p(x,t;\mu) - \partial_\mu \phi(t;\mu)[u]_{\phi(t;\mu)}\delta_\Sigma \tag{5.7}$$

in the sense of distributions. Can we find an equation satisfied by $\partial_\mu u(x,t;\mu)$? It is a weak solution if and only if, for all test function $w \in \mathcal{D}(Q)$:

$$I = \int_Q u\partial_\mu \partial_t w + \int_Q \frac{u^2}{2}\partial_\mu \partial_x w + \int_{\Omega \times \mathcal{U}_{ad}} u^0(\mu)\partial_\mu w(x,0,\mu).$$

When integrating by parts, one must take into account the fact that all these quantities are discontinuous on Σ. We just cite the result here, and refer to [13] for the full derivation. $\partial_\mu u$ is the solution of:

$$\begin{cases} \partial_t \partial_\mu u^p + \partial_x(u\partial_\mu u^p) &= 0 & \text{on } Q^- \bigcup Q^+ \\ \partial_\mu u(x,0) &= \partial_\mu u^0(x) & \\ (\partial_t + \bar{u}\partial_x)(\partial_\mu \phi(t;\mu)[u]_{\phi(t;\mu)}) &= \nu_x[u\partial_\mu u^p]_{\phi(t;\mu)} + \nu_t[\partial_\mu u^p]_{\phi(t;\mu)} & \text{on } \sigma(\mu) \end{cases} \tag{5.8}$$

Remark 39 *The proof is given in the calibrated case in the appendix.*

First equation of (5.8) is a linear equation on $\partial_\mu u^p$. It needs to be solved pointwise away from the shock. Unlike for Burgers equation, we have no physical Rankine Hugoniot condition at the shock. The third equation of (5.8) is the equation solved by $\partial_\mu \partial_t \phi$. It determines the measure component in $\partial_\mu u$. We can see another interesting feature. The characteristics for $\partial_\mu u$ are the same as the characteristics for u. We will see how to take advantage of that numerically in a ROM context.

5.2.3.1 Computation of $\partial_\mu J$

Using the same kind of calculations as in the previous subsection, and assuming for now that u^d is smooth at $\phi(T;\mu)$, we get:

$$\partial_\mu J = \int_\Omega \partial_\mu u^p(x,T;\mu)\left(u(x,T;\mu) - u^d(x)\right) dx - \partial_\mu \phi(T;\mu)[u]_{\phi(T;\mu)}(\bar{u}_{\phi(T;\mu)} - u^d).$$

Remark 40 *The case where the shocks of u^d and $u(\cdot,T;\mu)$ are aligned needs a different treatment, as it involves the product of a dirac mass and a discontinuous function.*

Fix $\mu \in \mathcal{U}_{ad}$. In [13], they propose one possible approach to compute $\partial_\mu J$, inspired by the elliptic case. Let p the solution of the dual equation:

$$\begin{cases} \frac{\partial p}{\partial t} + u(\cdot,\cdot;\mu)\frac{\partial p}{\partial x} &= 0 \\ p(\cdot,T) &= p_T(\cdot). \end{cases} \tag{5.9}$$

The idea is to re write the objective function's derivative in terms of the solution to the dual equation, just as in the introductive elliptic case in section 5.1.1. This implies finding a set of boundary and initial conditions for the dual equation such that $\partial_\mu J$ can be easily expressed in terms of p. Inspecting the form of (5.9), it appears that the characteristics are the same as for u. Thus, solving this backward equation will involve going back up the characteristics. For solutions with shocks, some characteristics enter the shock before $t = T$. We thus expect to have to define a boundary condition for p on $\sigma(\mu)$. Figure 5.5 illustrates this process. The dotted lines are the characteristics from Ω^- and Ω^+ subdomains entering the shock at $t = T$. The value of p inside the dotted lines depends on the boundary condition on $\sigma(\mu)$.

Keeping in mind these preliminary remarks, we follow the steps of the elliptic case. We start with the equation satisfied by p, see (5.9). Multiply by $\partial_\mu u^p$ and integrate over Ω and $[0,T]$:

$$\int_{\Omega\times[0,T]} \partial_\mu u^p(x,t;\mu)\left(\frac{\partial p}{\partial t} + u\frac{\partial p}{\partial x}\right) = 0.$$

We integrate by parts and get:

$$\begin{aligned}
\textstyle\int_{\Omega\times[0,T]} \partial_\mu u^p(x,t;\mu)\frac{\partial p}{\partial t} = \quad & -\textstyle\int_{\Omega\times[0,T]} \partial_t\left(\partial_\mu u^p(x,t;\mu)\right)p \\
& +\textstyle\int_{\sigma(\mu)} \nu_t\left[\partial_\mu u^p(x,t;\mu)p(x,t)\right]_{\phi(t;\mu)} \\
& +\textstyle\int_\Omega \partial_\mu u^p(x,T;\mu)p(x,T) \\
& -\textstyle\int_\Omega \partial_\mu u^p(x,0;\mu)p(x,0) \\
\textstyle\int_{\Omega\times[0,T]} \partial_\mu u^p(x,t;\mu)u\frac{\partial p}{\partial x} = \quad & -\textstyle\int_{\Omega\times[0,T]} \partial_x\left(u\partial_\mu u^p(x,t;\mu)\right)p \\
& +\textstyle\int_{\sigma(\mu)} \nu_x\ \left[u\partial_\mu u^p(x,t;\mu)p(x,t)\right]_{\phi(t;\mu)}.
\end{aligned}$$

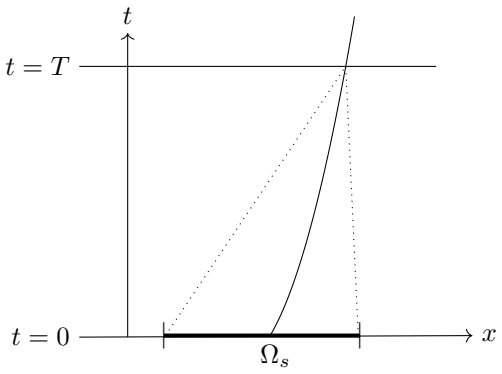


Figure 5.5: Solving for $p(x,0)$

We choose to impose the same boundary condition on p on both sides of $\sigma(\mu)$: $[p]|_{\sigma(\mu)} = 0$. This and the first equation of (5.8) give:

$$\begin{aligned} 0 \quad &= \int_{\sigma(\mu)} \nu_x \ [u\partial_\mu u^p(x,t;\mu)]_{\phi(t;\mu)}\, p(x,t) + \int_{\sigma(\mu)} \nu_t \ [\partial_\mu u^p(x,t;\mu)]_{\phi(t;\mu)}\, p(x,t) \\ &+ \int_\Omega \partial_\mu u^p(x,T;\mu) p(x,T) - \int_\Omega \partial_\mu u^p(x,0;\mu) p(x,0) \end{aligned}$$

We choose p constant over $\sigma(\mu)$ and use the third equation in (5.8):

$$\int_{\sigma(\mu)} (\partial_t + \bar{u}\partial_x)(\partial_\mu \phi(t;\mu)[u]_{\phi(t;\mu)})p = \partial_\mu \phi(T;\mu)[u]_{\phi(T;\mu)}\, p(\phi(T;\mu),T) - \partial_\mu \phi(0;\mu)[u]_{\phi(0;\mu)}\, p(\phi(0;\mu),0).$$

Finally:

$$\begin{aligned} 0 \quad &= \int_\Omega p(x,T)\partial_\mu u^p(x,T;\mu) - \int_\Omega p(x,0)\partial_\mu u_0 \\ &+ \partial_\mu \phi(T;\mu)[u]_{\phi(T;\mu)}\ p(\phi(T;\mu),T) - \partial_\mu \phi(0;\mu)[u]_{\phi(0;\mu)}\ p(\phi(0;\mu),0) \end{aligned}$$

The inspection of $\partial_\mu J$ suggests choosing $p(x,T) := u(x,T;\mu) - u^d(x)$, and $p|_{\sigma(\mu)} = \bar{u}_{\phi(T;\mu)} - u^d$. This choice fits the objective, as the derivative of the objective functional can now be computed using the solution of the dual equation:

$$\partial_\mu J = \int_\Omega p(x,0)\partial_\mu u_0^p + \partial_\mu \phi(0;\mu)[u]_{\phi(0;\mu)} \ \left(\bar{u}_{\phi(T;\mu)} - u^d\right). \tag{5.10}$$

This requires the value of p on the $t = 0$ line. Let Ω_s as depicted in Figure 5.5:

$$\Omega_s := \left\{ x \in \Omega, \quad \text{s.t } t \to Z_{0,t}^\mu(x) \text{ enters the shock before } T \ \right\}.$$

Following the discussion on the characteristics, we know that the value of $p(x,0)$ outside Ω_s will depend on $p(\cdot,T)$, and that the value of $p(x,0)$ for $x \in \Omega_s$ will depend on the boundary condition p on $\sigma(\mu)$.

Remark 41 *By construction, $p(\cdot,0)$ will be discontinuous at $\partial\Omega_s$.*

We sum up the conditions on p:

- p constant over $\sigma(\mu)$

- $\forall t,\ p(\phi(t;\mu),t) = \bar{u}_{\phi(T;\mu)} - u^d$
- $p(\cdot,T) = u(x,T;\mu) - u^d(x)$ away from $\phi(T;\mu)$

At that point, a few questions remain unanswered:

- if u^d and $u(\cdot,T;\mu)$ have aligned shocks, then $\partial_\mu J$ is not properly defined.
- equation (5.10) suggests taking $\partial_\mu u_0^p = p(x,0)$ which is discontinuous. This is very likely to create solutions with more than one shock

These issues are partially solved in [32]. By inspecting the resulting form of the derivative of J, equation (5.10), they choose to decompose the variation of the initial condition into two independent infinitesimal variations:

- the variation of the smooth part of the initial condition; this corresponds to the first term in (5.10)
- the variation of the position and intensity of the shock at $t = 0$; this corresponds to the second term in (5.10)

They alternate between these two optimizations and manage to mitigate the previous issues. Nevertheless, the method they propose relies on the exact form of the objective function beeing used. Also, their method does not fulfill one of our objectives here: fit this control problem in the presence of shock into a ROM framework. This is the topic of the rest of this chapter. We will use the calibration ideas developped in chapters 3 and 4

5.3 Calibration

The work done until now has highlighted the difficulties due to the non smoothness of $\mu \to u(x,t;\mu)$. This makes both the development of standard optimal control strategy, and of standard ROM methods tedious. The central aspect of the method proposed in this chapter is calibration. As in chapter 3, denote v the calibrated solution:

$$v := \left\{ \begin{array}{lll} \Omega \times [0,T] \times \mathcal{U}_{ad} & \to & \mathbb{R} \\ x,t;\mu & \mapsto & u(x-\phi(t;\mu),t;\mu). \end{array} \right.$$

The point of view adopted until the end of this chapter is to consider that u and ϕ are known. One can for instance think that we have solved for u using the method of chapter 3. v is then just a translated version of the known u. The focus is now put on computing the derivatives of v and ϕ with respect to the parameter. Note that this idea of using calibrated solutions for conservation laws is not new. For instance, it was used in [61] to study the linearized stability of solutions. The main difference with what is presented here, is that the authors eventually go back to the initial equation, as it is its stability that is beeing studied. Calibration was in that context used a detour to get properties on the solutions to the original problem. In this section, we use calibration to modify the objective function, and work solely with calibrated solutions.

Remark 42 *Recall that the method proposed in chapter 3 does not necessarily capture the exact position of the shock. Unlike shock fitting methods, calibration allows for a discrepancy between the calibration parameter and the true shock position. The consequences of this discrepancy are unclear, and will not be discussed in this chapter. In consequence, we will make the strong assumption that the calibration method captures the exact position of the shock.*

We will re use the notation of section 5.2.3. This time Ω^-,Ω^+,Q^+ and Q^- are independent of μ.

5.3.1 Smoothness away from the shock

This will be a very short section. We simply want to insist on the fact that the calibration has not destroyed the smoothness property away from the shock. We can follow the steps of section 5.2.2.

Let $(x,t) \in \Omega$. We need to make sure that $v(x,t;\mu+\delta\mu)$ and $v(x,t;\mu)$ are close. Thanks to calibration, we now know that they both stand on the same side of the shock. Then, the associated characteristics are emitted from close points on the $t=0$ line. We use the smoothness of $v(x,t=0;\mu)$ as a function of μ to conclude. A more rigorous proof would use the precise form of the characteristics for v and the smoothness hypothesis on $\mu \to \phi(t;\mu)$, see next section.

5.3.2 Smoothness at the shock

The surface in $\Omega \times [0,T] \times \mathcal{U}_{ad}$ where the calibrated solutions v are discontinuous is now a plane: $(\phi_0, t; \mu)$. An illustration is presented in Figure 5.6. For fixed μ, the normals to the calibrated

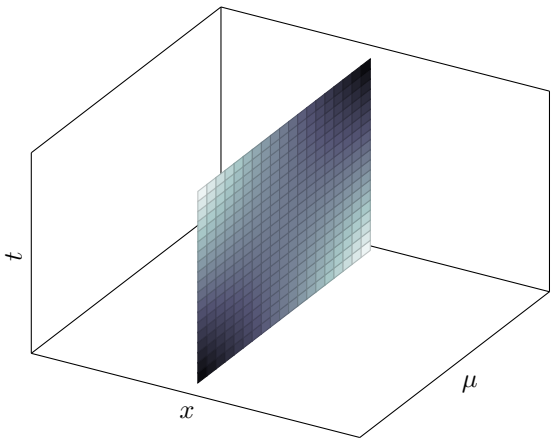


Figure 5.6: $\Omega \times [0,T] \times \mathcal{U}_{ad}$ in the calibrated case

$\sigma(\mu)$ are defined as:

$$\nu := (\nu_x, \nu_t) = (1, 0).$$

The key property of the calibrated solution is that the measure component in the derivative is set to zero. Its derivative is a smooth by part function. To put it in the notation of the uncalibrated case, we have:

$$\partial_\mu v(x,t;\mu) = \partial_\mu v^p(x,t;\mu).$$

The proof is trivial, we just have to follow the derivation of subsection 5.2.3.

$$\int_Q v(x,t;\mu)\partial_\mu w = -\int_{Q^+} \partial_\mu v(x,t;\mu) w - \int_{Q^-} \partial_\mu v(x,t;\mu) w.$$

We insist a little bit. Usually, the derivative of a discontinuous function makes a dirac mass pop. Here, even if $(x,t) \to v(x,t;\mu)$ is discontinuous, we have :

$$\begin{array}{ccc} \mathcal{U}_{ad} & \to & L^2(\Omega \times (0,T)) \\ \mu & \mapsto & v(\cdot,\cdot;\mu) \end{array}$$

in $C^1(\mathcal{U}_{ad})$. This is it, we have found a better candidate for optimal control, than the original state variable.

5.3.3 The calibrated solution

It is easy to derive the equation satisfied by v. We know:

$$\begin{aligned} \partial_t u(x,t;\mu) &= \partial_t v(x+\phi(t;\mu),t;\mu) + \partial_t \phi(t;\mu)\partial_x v(x+\phi(t;\mu),t;\mu) \\ \partial_x u(x,t;\mu) &= \partial_x v(x+\phi(t;\mu),t;\mu) \end{aligned}$$

The calibrated Burgers equation is thus given by:

$$\partial_t v + \partial_t \phi(t;\mu)\ \partial_x v + \partial_x \left(\frac{v^2}{2}\right) = 0. \tag{5.11}$$

Remark 43 *This is the freezing method of section 3.2.2, in the simple translation case.*

As for Burgers equation, we can interpret this problem in terms of characteristics. Equation (5.11) can be written in what is usually called non conservative form as:

$$\partial_t v + (\partial_t \phi(t;\mu) + v)\,\partial_x v = 0.$$

Let $\mu \in \mathcal{U}_{ad}$. Let $X_{cal;\mu}$ be the solution of the following simple ODE:

$$\partial_t X_{cal;\mu}(t) = v\left(X_{cal;\mu}(t),t;\mu\right) + \partial_t \phi(t;\mu).$$

Until $X_{cal;\mu}$ enters the shock, we have:

$$v(X_{cal;\mu}(t),t;\mu) = v(X_{cal;\mu}(0),0;\mu).$$

A simple illustration is presented in Figure 5.7. As the original solution, v is constant on characteristics. The difference is that the latter are no longer are straight lines.

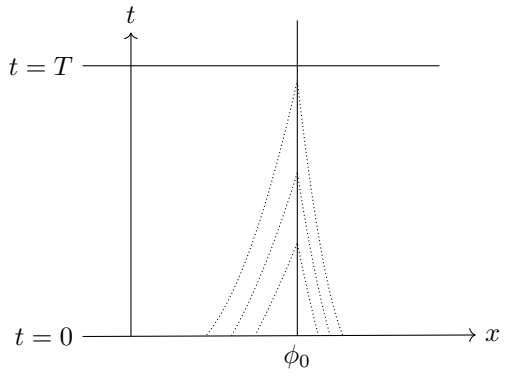


Figure 5.7: Characteristics in the calibrated case

$\partial_\mu v$ is solution of the following system:

$$\begin{cases} \partial_t \partial_\mu v + \partial_x(\partial_\mu v v) + \partial_\mu \partial_t \phi \ \partial_x v + \partial_t \phi \ \partial_x \partial_\mu v = 0 & \text{in } Q^- \cup Q^+ \\ \partial_\mu v(t=0) = \partial_\mu v^0 & \text{on } \Omega \times \mathcal{U}_{ad} \\ \partial_\mu \partial_t \phi = \overline{\partial_\mu v} \end{cases} \tag{5.12}$$

The first two equations are a consequence of the smoothness of $\mu \to v(\cdot, \cdot; \mu)$ away from the shock. The last equation can be rigorously derived by following the lines of [13]. This is presented in the appendix.

Once again, we can use characteristics to better understand the structure of this system. The first equation of (5.12) is equivalent to:

$$\partial_t \partial_\mu v + (v + \partial_t \phi)\, \partial_x \partial_\mu v = -\partial_\mu \partial_t \phi \ \partial_x v - \partial_x v \partial_\mu v.$$

Note that the characteristics are the same as the ones of the equation involving v: $X_{cal;\mu}$. But unlike v, $\partial_\mu v$ is not constant on the characteristics, but is solution to a first order ODE. More precisely, denote h the value of $\partial_\mu v$ over some characteristic:

$$h : t \to \partial_\mu v(X_{cal;\mu}(t), t; \mu),$$

it is a solution of:

$$\partial_t h = -\partial_x v h, -\partial_\mu \partial_t \phi \ \partial_x v.$$

This equation can be explicitly solved.

Remark 44 *Again, because of calibration, $\partial_x v$ is a well defined term outside of the fixed shock. The integration of the ODE does not cause any particular problem.*

This interpretation in terms of characteristics will be used in the numerical section.

5.3.4 Objective function

With no calibration, the derivative of the objective function was given by something like:

$$\partial_\mu J(\mu) \approx \int_\Omega \left(u(x,T;\mu) - u^d(x)\right) \partial_\mu u(x,T;\mu) + \text{ some pointwise value at the shock}$$

We propose a few possible objective functions adapted to the calibrated framework:

- the calibrated solution should look like some prescribed solution u^d
- the shock should be located at some prescribed location ϕ^{obj} at $t = T$
- some combination of the two previous objectives

Let $K \in \mathbb{R}$. We choose the last option:

$$J_K(\mu) = \int_\Omega |v\,(x,T;\mu) - u^d(x)|^2 dx + K(\phi(T;\mu) - \phi^{obj})^2$$

where K is the weight given to the shock positioning objective. The computation of the derivative is straightforward:

$$\partial_\mu J_K(\mu) = \begin{array}{l} \int_\Omega \left(v(x,T;\mu) - u^d(x)\right) \partial_\mu v(x,T;\mu) \\ +K \ \partial_\mu \phi(T;\mu) \left(\phi(T;\mu) - \phi^{obj}\right). \end{array}$$

Unlike in the uncalibrated case, there is no problem in interpreting the first term as both quantities are functions, smooth by parts. The measure component has been set to zero thanks to calibration.

We have solved the first objective: we have modified the state variable as well as the objective function to get smooth dependencies $\mu \to v(\cdot, \cdot; \mu)$ and $\mu \to J_K(\mu)$. The next topic on the line is finding an efficient way of computing $\partial_\mu J_K(\mu)$. This involves the numerical estimation of $\partial_\mu \partial_t \phi$.

Remark 45 *The computation of $\partial_\mu \phi(T; \mu)$ will not be an issue, as it has a simple explicit form:*

$$\partial_\mu \phi(T; \mu) = \partial_\mu \phi(\mu, 0) + \int_0^T \partial_t \partial_\mu \phi(t; \mu).$$

because of the smoothness in time discussed in section 5.2.

5.3.5 The estimation of $\partial_\mu \partial_t \phi$

We could use the derivative of the R-H condition with respect to the parameter at the shock, see third equation of (5.12). Instead, we propose a method that can also be used for classical solutions. More precisely, we will show how the problem of finding $\partial_\mu \partial_t \phi$ can be treated as was the problem of finding $(\gamma^{n+1} - \gamma^n)$ in chapter 3 and $F_{n+1} \circ F_n^{-1}$ in chapter 4. Let $\{\Phi_i\}$ be some reduced basis of the calibrated solution manifold, computed offline. The ROM premise is that:

$$\forall \mu \in \mathcal{U}_{ad},\ \forall t \in [0, T],\ \exists \{\alpha_i\},\quad \text{s.t } u(\cdot, t; \mu) = \sum_i \alpha_i(t; \mu) \Phi_i(\cdot - \phi(t; \mu)). \tag{5.13}$$

The smoothness of the calibrated solution with respect to time and to parameter can be expressed as:

$$\forall i,\ (t, \mu) \to \alpha_i(t; \mu) \in C^1(\mathcal{U}_{ad} \times [0, T], \mathbb{R}).$$

With this assumption, we have an explicit form for $\partial_\mu v$:

$$\partial_\mu v(\cdot, t; \mu) : (t; \mu) \to \sum_i \partial_\mu \alpha_i(t; \mu) \Phi_i(\cdot).$$

We want to construct a procedure to estimate $\partial_\mu \partial_t \phi$ mimicking the methods of chapter 3 and 4. Towards this objective, for each smooth functional

$$\psi : \left\{ \begin{array}{lll} [0, T] \times U_{ad} & \to & \mathbb{R} \\ (t, \mu) & \mapsto & \psi(t; \mu), \end{array} \right. \tag{5.14}$$

define

$$v^\psi : \left\{ \begin{array}{lll} \mathcal{U}_{ad} \times [0, T] & \to & X \\ (\mu, t) & \mapsto & u(\cdot - \psi(t; \mu), t; \mu). \end{array} \right.$$

This corresponds to a badly calibrated solution. With this notation, the true calibrated solution v is denoted v^ϕ. A wrong estimation for $\partial_\mu \partial_t \phi$ while solving (5.12) is equivalent to working with v^ψ where ψ is such that

$$\left\{ \begin{array}{lll} \psi(\mu, t) & = & \phi(\mu, t) \\ \partial_t \psi(\mu, t) & = & \partial_t \phi(\mu, t) \\ \partial_\mu \psi(\mu, t) & \neq & \partial_\mu \phi(\mu, t). \end{array} \right.$$

Away from the shock (i.e where v is smooth), we have:

$$v^{\psi} = v - (\psi - \phi)\partial_x v. \tag{5.15}$$

Thus, away from the shock:

$$\partial_\mu v^{\psi} = \underbrace{\partial_\mu v}_{\in \text{ span } \Phi_i} - \partial_\mu(\psi - \phi) \underbrace{\partial_x v}_{\in \text{ span } \partial_x \Phi_i} - (\psi - \phi) \underbrace{\partial_x \partial_\mu v}_{\in \text{ span } \partial_x \Phi_i}.$$

A wrong calibration means a bigger component onto the derivative of the underlying ROM basis.

We are in the proper setting to propose a method with the desired properties. Let $\{\tilde{\Phi}_i\}_i$ be some orthogonal basis. We choose it such that:

$$\begin{cases} \sum_i \sum_j < \tilde{\Phi}_i, \Phi_j >^2 \text{ not too small} \\ \tilde{\Phi}_i \notin \text{ span } \{\partial_x \Phi_j\}, \text{ away from the shock .} \end{cases} \tag{5.16}$$

The first property ensures that $\{\tilde{\Phi}_i\}$ captures enough of the calibrated solution manifold. The quality of the estimation on $\partial_\mu \partial_t \phi$ will depend on the second property.

A thorough investigation of the construction of the basis $\{\tilde{\Phi}_i\}$ is out of the scope of this chapter. We just mention the simplest method we can think of. One could use some variant of the Gram-Schmidt orthogonalization procedure on the basis $\{\Phi_j\}$ to remove the components parallel to span $\{\partial_x \Phi_j\}$. We have no guarantee that the resulting basis would still satisfy the first property of (5.16).

Remark 46 *A rough offline calibration procedure makes the construction of $\{\tilde{\Phi}_i\}$ complicated, as the x derivatives are in the basis.*

We now assume the existence of such a basis. We have everything we need to construct an algorithm similar to the one used in chapter 3. Start by computing $(\partial_\mu v)^{n+1}$ for several guesses for $\partial_\mu \partial_t \phi$ and compute the corresponding norm of the orthogonal projection onto $\{\tilde{\Phi}_i\}$. Then interpolate to get an optimal estimate of $\partial_\mu \partial_t \phi$. More precisely, we are choosing ϕ as

$$\phi := \underset{\psi}{\text{argsup}} \ \left\| \Pi_{\tilde{\Phi}} \partial_\mu v^{\psi}(\cdot, t; \mu) \right\|_{L^2(\Omega_{sub})} \tag{5.17}$$

where $\Pi_{\tilde{\Phi}}$ denotes the orthogonal projection onto the basis $\left\{\tilde{\Phi}_i\right\}$, and Ω_{sub} is some domain away from the shock. This idea of restricting the physical domain before an optimization procedure has already been used in chapter 4.

Remark 47 *This method, unlike R-H conditions, works for solutions where the shock has not yet appear. We just have to define $(t; \mu) \to \phi(t; \mu)$ as the line of highest gradient that eventually forms a shock. This ensures the required smoothness in μ and t.*

5.3.6 On the computation of $\partial_\mu v$

The last objective is to propose a well behaved numerical method to solve equation (5.12) at time t^{n+1}, for a fixed value of $\partial_\mu \partial_t \phi$. For simplicity, one can think of an explicit Euler time discretization:

$$(\partial_\mu v)^{n+1} - (\partial_\mu v)^n + dt \ \partial_x \left((\partial_\mu v)^n v^n\right) + dt \ (\partial_\mu \partial_t \phi)^n \partial_x v^n + dt \ (\partial_t \phi)^n (\partial_x (\partial_\mu v)^n) = 0. \tag{5.18}$$

The purpose of the following sections is to try to extend the properties of a scheme used to compute v, to the computation of $\partial_\mu v$. This is the most natural choice in order to have a

'well behaved' scheme in this context, but note that other choices are possible. This discussion resembles the one we had in chapter 1 about the fact that reduced scheme and truth scheme could be different. Similarly here, tailored scheme for the computation of $\partial_\mu v$ could be constructed. Nevertheless, adapting the numerical tools used for the computation of v is the simplest way to guarantee stability.

5.3.7 Upwinding

To solve numerically a convection dominated problem with a finite volume method, it is advised to use upwinded flux instead of centered flux, to enforce stability. It appears obvious that whatever the parameter dependency, the upwinding direction used to compute $\partial_\mu v$ should be the same as the one used for v. This appears clearly when the parameter is on the initial condition, see section 5.3.2. An example with varying boundary conditions is mentioned in section 5.4.3. This particular fact, that pieces of information about the computations of the state variables could be used to compute the derivative with respect to some parameter, was already used in [61].

For a non calibrated problem, the upwinding procedure is trivially not smooth with respect to the parameter. We have no guarantee that the resulting scheme on $\partial_\mu v$ will be stable and we can expect that terms with irreductible n-widths would appear, as described in section 1.5.2. Calibration is helping in that respect. For instance, in the simple viscous Burgers numerical examples of chapter 3, the upwinded flux are linear functions of the calibrated solution considered. Using the premise that $\mu \to v(\cdot, \cdot; \mu)$ is a smooth function, we conclude that the upwinding procedure has a smooth parameter dependence, and can thus be handle by ROM. In the next section, we explicit this idea for another standard stabilization mechanism: slope limitation.

5.3.8 Slope limiters

The development in this section will be closely related to the discussion about flux limiters in the introductive chapter, section 1.5.2. We denote $\mathcal{T}_h$ some mesh in Ω and $P^0(\mathcal{T}_h)$ some constant per element discretization space. Γ denotes a generic boundary in $\mathcal{T}_h$.

High order finite volume method require the estimation of the solution at the boundaries of control volumes, using reconstructed quantities. The most common procedure is the reconstruction of the gradient over a control volume, using the average of the solution in neighboring cells. Let v_i be the average of the state variable v in control volume i. The values at some interface Γ are then estimated using the first order expansion :

$$v|_\Gamma = v_i + \nabla_i \cdot (x|_\Gamma - x_{\text{ centroid }}). \tag{5.19}$$

∇_i is a function of the average of v in the neighboring cells, $\{v_j\}$. The cells actually used are often referred to as stencil. ∇_i needs to be chosen so that:

- the scheme is stable
- the scheme is accurate
- the added computational cost is reasonable

The rest of this section takes place in a one dimension context, where $\mathcal{T}_h$ is an uniform mesh. This favorable setting offers rigorous theoretical results. Indeed, it is known [17] that a scheme using a reconstructed gradient such that:

- $\nabla_i = 0$ at local maximas

- the reconstruction process does not create new local maximas

is a TVD scheme.

The most common gradient limitation procedure is the usage of a modified gradient:

$$\tilde{\nabla}_i := \phi_i \nabla_i,$$

where ∇_i is a standard gradient reconstruction, ENO or LS for instance and where the limiter ϕ is chosen as a function of the ratio of forward and backward differences:

$$\forall i,\ \phi_i = \phi(R_i) \text{ where } R_i = \frac{v_{i+1} - v_i}{v_i - v_{i-1}}.$$

Many different choices are possible for ϕ. We will restrict ourselves to the smooth ones that guarantee the TVD property. One example is the Van-Leer limitor given by:

$$\phi : R \to \begin{cases} \frac{2R}{R+1} \text{ for } R > 0 \\ 0 \text{ for } R < 0. \end{cases}$$

We denote by ∇ some standard gradient reconstruction process. In control volume i:

$$\nabla_i : \begin{cases} \mathcal{U}_{ad} & \to & \mathbb{R} \\ \mu & \mapsto & \nabla_i(\{v_j(\mu)\}). \end{cases}$$

We denote by ∇_i' the derivative of ∇_i with respect to μ, when it exists:

$$\nabla_i' := \frac{1}{\delta\mu} \nabla_i(\{v_j(\mu + \delta\mu)\}) - \nabla_i(\{v_j(\mu)\}).$$

For central differences, least squares (or any process independent of the solution), we know that this application is smooth. For instance, for central difference, the reconstructed gradient is chosen as: $\nabla_i : \mu \to \frac{v_{i+1}(\mu) - v_{i-1}(\mu)}{2}$. This application is trivially smooth, and the derivative is given by:

$$\nabla_i' : \mu \to \frac{(\partial_\mu v(\mu))_{i+1}}{2} - \frac{(\partial_\mu v(\mu))_{i-1}}{2}.$$

For ENO, WENO and other scheme more involved (with a notion of direction) the smoothness of the gradient reconstruction process is not guaranteed in the general case. We need stronger hypothesis, that will be satisfied because of the calibration process. These hypothesis are:

- the local maximas do not move with the parameters
- the direction of the flow, in the calibrated domain, does not change with the parameters

A direct consequence of the first point, is that the limiting process is smooth, even at local maximas. A direct consequence of the second point is that ENO type gradient reconstructions will be smooth with respect to the parameter.

We now have ∇ a smooth application. Let's show ϕ is also a smooth function of the parameter:

$$\phi_i : \begin{cases} \mathcal{U}_{ad} & \to & \mathbb{R} \\ \mu & \mapsto & \phi_i(R_i) \end{cases}$$

Away from local extremums, we have:

$$\begin{aligned} \partial_\mu R_i &= \frac{\partial_\mu(v_{i+1} - v_i)(v_i - v_{i-1}) - \partial_\mu(v_i - v_{i-1})(v_{i+1} - v_i)}{(v_i - v_{i-1})^2} \\ &= \frac{1}{v_i - v_{i-1}} \left[\partial_\mu(v_{i+1} - v_i) - R\partial_\mu(v_i - v_{i-1})\right]. \end{aligned}$$

At local extremums, we have $\partial_\mu \phi_i = 0$.

We go back to the first order expansion, see equation (5.19). All the quantities involved are smooth with respect to μ. We can thus use the product rule to get an explicit form for the derivative of v_Γ with respect to μ:

$$\begin{aligned}\partial_\mu v_\Gamma := \partial_\mu v_i(\mu) \quad & +\phi'(R_i)\ \partial_\mu R_i \nabla_i \cdot (x|_\Gamma - x_{\text{centroid}}) \\ & +\phi(R_i)\ \nabla_i' \cdot (x|_\Gamma - x_{\text{centroid}})\end{aligned}$$

For VanLeer limitor $\phi' : R \to \frac{2}{(R+1)^2}$.

We have shown in the two previous sections how to extend the properties of numerical schemes for convection dominated problems to the computation of the derivatives of the state variables. Once again, this is not the only viable strategy. Nevertheless, this perfectly fits within the calibration framework.

5.4 Euler 2d

We will apply the ideas of the previous sections to the problem studied in chapter 4. Suppose that we have some NACA airfoil inside a domain. It is standard in that context to take as objective functional J, the integral over the body of some function of the pressure and of the normal at the boundary:

$$J(\mu) := \int_{\partial\Omega} j(P(\mu), n_w) ds.$$

For simplicity, one can think of a parameter dependency on physical parameters such as Mach number, angle of attack etc. But using what has been done in chapter 4, recall that we can also handle parametrized wing shapes.

5.4.1 Initial remarks

From chapter 4, we know that the shock's existence as well as its position and shape depend on the angle of attack, the mach number and the shape of the wing. As for inviscid Burgers, we need hypothesis on the type of solutions considered:

- $\forall \mu \in \mathcal{U}_{ad}$, there is one shock. Denote $\phi(\mu) \in \mathbb{R}$ its position on the wing
- the solutions are smooth with respect to parameter variation away from the shock. The main arguments, just as for one dimensional Burgers, are the finite speed of propagation of the information, as well as the smoothness of initial and boundary conditions
- $\mu \to \phi(\mu)$ is smooth

Without any calibration, we expect the derivative of J to contains a dirac mass at the shock's position:

$$\partial_\mu J = \text{some smooth by part function of } \partial_\mu j + \delta(x - \phi(\mu))[j(P(\mu, n_w)]|_{\phi(\mu)} \partial_\mu \phi.$$

This exact setting has been studied in [10].

5.4.2 Equation for the derivative

Far from the shock, the μ derivatives of the state variables are solution of the linearized equation, see [10]. We will use the notation of chapter 4 directly. Mimicking what has been done for Burgers, we do not try to solve for the original state variables $\partial_\mu u$, but rather for the calibrated ones: $\partial_\mu \hat{u} := \partial_\mu (u \circ F_\mu)$. A rigorous derivation of the equation solved by $\partial_\mu \hat{u}$ is not in the scope of this paper. Nevertheless, we expect that the work done in the previous sections can be transposed exactly. We will highlight a few aspects.

5.4.3 Computational details

Just as for Burgers, one has to be careful on the numerical method used to solve for $\partial_\mu u$ (or $\partial_\mu \hat{u}$). For instance, suppose that one of the parameters of the problem is the angle of attack. That is, on Γ_{ext}, we have a Dirichlet boundary condition on the velocity of the form

$$(u, v) = (cos(\mu), sin(\mu)).$$

The derivative of the solution will satisfy the linearized equation, with the following boundary condition on Γ_{ext}:

$$(\partial_\mu u, \partial_\mu v) = (-sin(\mu), cos(\mu)).$$

Intuitively, we know that giving this boundary conditions to a black box solver will give non satisfactory results. Indeed, some of the inflow and outflow boundaries are exchanged. This is not reasonable physically. As for Burgers, we need to take as upwinding direction for the derivatives the same as the upwinding direction for the original variables. The calibration helps with the smoothness of the upwinding directions.

5.4.4 Estimation of $\partial_\mu N$

This is the equivalent to the estimation of $\partial_\mu \phi$ in the previous section. Once again, we could use RH conditions. We rather propose a ROM oriented method. Let $\mathcal{F}$ be a family of mapping and $F_\mu \in \mathcal{F}$ the correct calibration. An error on $\partial_\mu F_\mu$ while solving for $\partial_\mu (u \circ F_\mu)$ corresponds to a badly calibrated solution $\tilde{u}(\mu) := u \circ G$ with $G \neq F_\mu$. For Burgers, we were looking for an orthogonal basis satisfying $\tilde{\Phi}_j \notin \text{ span } \{\partial_x \Phi_i\}$. The counterpart is here:

$$\tilde{\Psi}_j \notin \text{ span } \{\Psi_i \circ \delta_F\}$$

where δ_F are variations around the identity, among the family $\mathcal{F}$ of mappings considered. The optimization procedure can be conducted as follows: first compute the solution of the equation on the derivative for several values of $\partial_\mu F_\mu$. Then, interpolate to get the $\partial_\mu F_\mu$ that maximizes the projection of $\partial_\mu (u \circ F_\mu)$ onto the basis $\{\tilde{\Psi}_j\}$.

5.4.5 Optimal control on the shape of the wing

We can combine the work that has been done in chapter 4 and in this one to propose a procedure to solve the optimal control problem of the design of airfoils. We give here a roadmap.

First, the offline phase

Data: Continuous collection of wing shapes
Result: Calibrated basis
Select a moderate number of wing shapes ;
Compute the solutions using a fine solver ;
Restrict the parameter range so that all considered candidates satisfy the hypothesis of section 5.4.1 ;
Pick some reference wing shape, as in chapter 4 ;
Map the fine solutions to the reference mesh ;
Store scheme informations, such as upwinding direction or gradient limitation ;

Algorithm 9: Offline

Then, the online phase

Data: Continuous collection of wing shapes, calibrated reduced basis
Result: Optimal wing shape
Set up, as in 4, the fine solver with modified flux boundary conditions, gradient limitations etc. ;
Start the optimization algorithm, using the method developed in this chapter;
The output is some 'optimal mapping' ;
Deduce an optimal wing shape;

Algorithm 10: Online

5.5 Conclusion

In this chapter, we have constructed a method to solve optimal control problems in the context of solutions with shocks. With the proper calibration and an appropriate choice of objective function, we have shown that we recover the smoothness of $\mu \to J(\mu)$ which is at the core of any optimal control problem. Using toy examples, we have shown that some specific care had to be put in the construction of numerical schemes for the computation of the derivatives of the solutions (calibrated and non calibrated). We have shown how to use calibration to extend the stability properties of standard numerical scheme to the computation of the derivatives.

Another objective of this chapter was to use ROM. This task is completely solved by calibration, see section 5.3.5. We have adapted the ideas of chapter 3 and 4 to propose an alternative of the R-H conditions for the computation of $\partial_\mu \phi$.

All the ideas developed in this chapter need to be numerically studied. The first step is to check the premise for an utilization of ROM, see equation (5.13). It could for instance be done in the context of the numerical section in chapter 3. Also, the method proposed to estimate $\partial_\mu \partial_t \phi$ has to be investigated. It relies on the construction of what we have denoted $\{\tilde{\Phi}_i\}$. Finally we need to test using the characteristics of the fine solver used for v to compute $\partial_\mu v$.

The ultimate application of what has been done in chapter 4 and this one is to perform wing shape optimization, such as described in section 5.4.5. There is still a long way to go, as all ingredients have only been skimmed over. Nevertheless, I believe that this is an adapted framework to solve this real-life problem.

Appendix

We will prove the result given in equation (5.12). The proof is just a variation of the one given in the calibrated case in [13]. $\partial_\mu v$ is a weak solution if and only if:

$$\forall w \in \mathcal{D}(Q),\ \int_Q v\ \partial_t\partial_\mu w + \int_Q \partial_t\phi(t;\mu)v\ \partial_x\partial_\mu w + \int_Q \left(\frac{v^2}{2}\right)\ \partial_x\partial_\mu w + \int_{\Omega\times\mathcal{U}_{ad}} v^0\partial_\mu w = 0.$$

Let $w \in \mathcal{D}(Q)$. We will first compute the new terms coming from calibration. We start by integrating by parts with respect to μ.

$$\begin{aligned}\textstyle\int_Q \partial_t\phi\ v\partial_\mu\partial_x w &= -\textstyle\int_Q \partial_\mu\left(\partial_t\phi\ v\right)\partial_x w + \int_\Sigma \partial_\mu\phi\partial_t\phi[v]\partial_x w\\ &= -\textstyle\int_Q \partial_\mu\partial_t\phi\ v\partial_x w - \int_Q \partial_t\phi\ \partial_\mu v\partial_x w + \int_\Sigma \partial_\mu\phi\partial_t\phi[v]\partial_x w.\end{aligned}$$

Now, we integrate by parts with respect to x:

$$\begin{aligned}\textstyle\int_Q \partial_t\phi\ v\partial_\mu\partial_x w &= \textstyle\int_Q \partial_\mu\partial_t\phi\ \partial_x v w + \int_Q \partial_t\phi\ \partial_x\partial_\mu v w + \int_\Sigma \partial_\mu\phi\partial_t\phi[v]\partial_x w\\ &\quad - \textstyle\int_\Sigma \nu_x\left(\partial_\mu\partial_t\phi[v]w + \partial_t\phi[\partial_\mu v]w\right).\end{aligned}$$

The others three terms (already dealt with in [13]) are:

$$\begin{aligned}\textstyle\int_Q v\partial_t\partial_\mu w &= -\textstyle\int \partial_\mu v\partial_t w + \int_\Sigma \partial_\mu\phi[v]\partial_t w\\ &= \textstyle\int_Q \partial_t\partial_\mu v w + \int_\Sigma \nu_t[\partial_\mu v]w + \int_\Sigma \partial_\mu\phi[v]\partial_t w - \int_{\Omega\times\mathcal{U}_{ad}} \partial_\mu v(t=0)w\\ \textstyle\int_Q \left(\frac{v^2}{2}\right)\ \partial_x\partial_\mu w &= -\textstyle\int_Q \left(v\ \partial_\mu v\right)\partial_x w + \int_\Sigma \partial_\mu\phi[v^2]\partial_x w\\ &= \textstyle\int_Q \partial_x\left(v\ \partial_\mu v\right)w + \int_\Sigma \nu_x[v\partial_\mu v]w + \int_\Sigma \partial_\mu\phi[v^2]\partial_x w\\ \textstyle\int_{\Omega\times\mathcal{U}_{ad}} v^0\partial_\mu w &= -\textstyle\int_{\Omega\times\mathcal{U}_{ad}} \partial_\mu v^0 w + \int_{\Sigma_0\times\mathcal{U}_{ad}} \partial_\mu\phi[v^0]w\end{aligned}$$

We regroup all volume contributions:

$$\int_Q \partial_\mu\partial_t\phi\ \partial_x v w + \partial_t\phi\ \partial_x\partial_\mu v w + \int_Q \partial_t\partial_\mu v w + \int_Q \partial_x\left(v\ \partial_\mu v\right)w - \int_{\Omega\times\mathcal{U}_{ad}} \partial_\mu v^0 w - \int_{\Omega\times\mathcal{U}_{ad}} \partial_\mu v(t=0)w = 0.$$

For $\partial_\mu v$ to be weak solution, the previous equation needs to be true for all $w \in \mathcal{D}(Q)$. As usual, we start with w such that $w(\cdot, t=0;\mu) = 0$, this gives us the linear PDE solved by $\partial_\mu v$:

$$\partial_t\partial_\mu v + \partial_x(\partial_\mu v v) + \partial_\mu\partial_t\phi\ \partial_x v + \partial_t\phi\ \partial_x\partial_\mu v = 0.$$

Taking $w(\cdot, t=0;\mu)$ non zero, we have the expected initial condition:

$$\partial_\mu v(\cdot, t=0;\cdot) = \partial_\mu v^0(\cdot;\cdot).$$

The surface contributions now. Because of $\partial_\mu\phi = 0$ and $\nu_t = 0$, most of the terms cancel. We are left with:

$$\begin{gathered}\textstyle\int_\Sigma \partial_\mu\partial_t\phi[v]w\\ \textstyle\int_\Sigma \partial_t\phi[\partial_\mu v]w\\ \textstyle\int_\Sigma [v\partial_\mu v]w\end{gathered}$$

We then just have to use: $[v\partial_\mu v] = [v]\overline{\partial_\mu v} + [\partial_\mu v]\bar{v}$, and the classical Rankine Hugoniot: $\bar{v} = \partial_t \phi$. We are left with:

$$\partial_\mu \partial_t \phi [v] = [v] \partial_\mu \bar{v}$$

which is the same as result as the one obtained by formally differentiating the RH condition.

Summing up, $\partial_\mu v$ is solution of the following PDE:

$$\begin{cases} \partial_t \partial_\mu v + \partial_x(\partial_\mu v v) + \partial_\mu \partial_t \phi \ \partial_x v + \partial_t \phi \ \partial_x \partial_\mu v = 0 & \text{in } Q^- \cup Q^+ \\ \partial_\mu v(t=0) = \partial_\mu v^0 & \text{on } \Omega \times \mathcal{U}_{ad} \\ \partial_\mu \partial_t \phi = \overline{\partial_\mu v} \end{cases}$$

which is the desired result.

Chapter 6

Analysis of the two level POD method

The problem we propose to deal with in this section is the computation of POD basis of very large sets. One reviewer has brought to my attention that this issue has already been recently solved in a more general setting in [69]. Suppose that we have a set of N_{snap} snapshots, all living in some $\mathcal{N}$ dimensional truth discretized space (say finite element space). We know that the computational cost of the POD method is roughly given by:

$$min(\mathcal{N}^2 N_{snap}, \mathcal{N} N_{snap}^2).$$

This is pretty clear when looking at the method of snapshots, see [122].

There are several ways to circumvent this issue. The first one is to use a POD greedy type method such as the one developed in [65]. The associated convergence rate has recently been studied in [66]. These methods, as all RB type methods, need a cheap, tight upper bound on the error. It is well known that this task can be difficult when dealing with non linear PDEs. Another option available in the literature is to perform the POD in parallel in the $\mathcal{N}$ direction, as in [139]. Here, we follow a third route, see [12], and propose a two level POD method.

Let Ξ be some fine discretization of a solution manifold denoted $\mathcal{M}$ embedded in some Hilbert space X. Start by creating subsets of Ξ, that we will call batches from now on and denote them $\{\Xi_p\}_p$. The idea is to parallelize the computation of the POD basis of Ξ using a divide and conquer approach. More precisely, we start by computing the POD basis of each batch. We then re combine the 'sub POD' basis, as best as we can. The rest of this section is devoted to the analysis of this recombination.

One particular application we have in mind is time-parameter solution manifolds. We will use this specific notations, as it will make our calculations a little more intuitive: each batch corresponds to a time simulation for one parameter μ_p:

$$\Xi_p := \left\{ u(\cdot, t^k; \mu_p),\ t^k \in [0, T] \right\}.$$

From now on, we will denote:

- P the number of batches
- N_{snap}^p the number of time-snapshots for the simulation for parameter $\mu = \mu_p$
- N_{red}^p the size of the POD basis corresponding to the simulation $\mu = \mu_p$

- N_{tot} the size of the final basis
- $\phi_i^{\mu_p}$ the ith basis of the batch Ξ_p
- ϕ_j^{tot} the jth basis, after re combination of the $\phi_i^{\mu_p}$.
- ϕ_j^{stand} the jtth basis returned by the standard POD algorithm on Ξ.

What do we want to control ? let N_{tot} be some prescribed size of the final basis. We want to minimize the following quantity:

$$J_{N_{tot}}\left(\{\phi_j\}_j\right) := \sum_p \sum_{k=1}^{N_{snap}^p} \left\| u(\cdot,t^k;\mu^p) - \sum_{j=1}^{N_{tot}} \left\langle u(\cdot,t^k;\mu^p),\phi_j\right\rangle_X \phi_j \right\|_X^2 . \tag{6.1}$$

As already mentioned, minimizing J over all orthogonal basis in X is the usual POD algorithm, see the introductive chapter, section 1. The resulting optimal basis is in fact in span $\left\{u(\cdot,t^k;\mu_p),\ p\in[1\dots P],\ k\in[1\dots N_{snap}^p]\right\}$.

Here we rather look for a global basis $\{\phi_j^{tot}\}$ embedded in span $\left\{\phi_i^{\mu_p},\ p\in[1\dots P],\ i\in[1\dots N_{red}^p]\right\}$. This is natural in the divide and conquer strategy. We refer to Figure 6.1 for an illustration of this process.

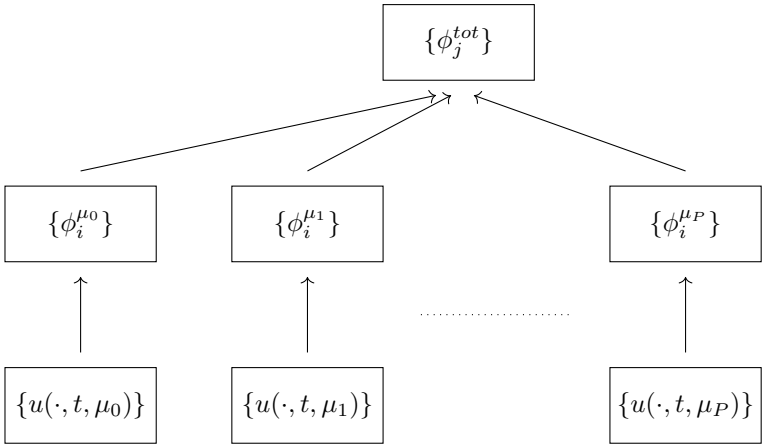


Figure 6.1: Divide and conquer

Several different methods could be implemented here. We choose to look for $\{\phi_j^{tot}\}$ as a weighted POD of $\left\{\phi_i^{\mu_p},\ p\in[1\dots P],\ i\in[1\dots N_{red}^p]\right\}$. This is the most natural way to go in this particular setting. We model this easier problem by defining the following functional $\tilde{J}$:

$$\tilde{J}:\begin{cases} \mathbb{R}^{\sum_p N_{red}^p}\times X^{N_{tot}} & \to \quad \mathbb{R} \\ \left\{\alpha_i^{\mu_p}\right\}_{i,p},\{\phi_j\}_j & \mapsto \quad \tilde{J}(\alpha,\phi_j) \quad = \sum_p \sum_{i=1}^{N_{red}^p} \left\| \sqrt{\alpha_i^p}\phi_i^{\mu_p} - \sum_{j=1}^{N_{tot}} \left\langle \sqrt{\alpha_i^p}\phi_i^{\mu_p},\phi_j\right\rangle_X \phi_j \right\|_X^2 \\ & \qquad\qquad\quad = \sum_p \sum_{i=1}^{N_{red}^p} \alpha_i^p \left\| \phi_i^{\mu_p} - \sum_{j=1}^{N_{tot}} \left\langle \phi_i^{\mu_p},\phi_j\right\rangle_X \phi_j \right\|_X^2 \end{cases}$$

We will denote by $\{\phi_j^{tot}(\alpha)\}$ the optimal basis for a specific choice of weights and by $\{\lambda_j^{tot}(\alpha)\}$ the corresponding eigenvalue set. Recall that they are the eigenfunctions and eigenvalues of the

operator $\tilde{R}(\alpha)$ defined as:

$$\tilde{R}(\alpha) : \left\{ \begin{array}{ccc} X & \rightarrow & X \\ \psi & \mapsto & \sum_p \sum_{i=1}^{N_{red}^p} \alpha_i^p \left\langle \phi_i^{\mu_p}, \psi \right\rangle_X \phi_i^{\mu_p}, \end{array} \right. \tag{6.2}$$

and that we have the well known POD property:

$$\forall \alpha,\ \tilde{J}\left(\alpha, \phi_j^{tot}(\alpha)\right) = \sum_{j > N_{tot}} \lambda_j^{tot}(\alpha).$$

Remark 48 *Of course, we do not have the standard ROM property between the decay of the* $\{\lambda_j^{tot}\}_j$ *and the global error. This is actually the purpose of this note.*

We will prove two things:

- we can control the global error by a combination of the local 'in batch' errors and of the error made when 'combining' the PODs
- the loss incurred by this divide an conquer strategy is controlled: there exists a constant C (data set independent) such that

$$\forall N_{tot},\ \exists \{N_{red}^p\}_p,\ \exists \{\alpha_i^{\mu_p}\}_{i,p} \in \mathbb{R}^{\sum_p N_{red}^p} \text{ s.t } J_{N_{tot}}(\{\phi_j^{tot}(\alpha)\}) \leq C J_{N_{tot}}(\{\phi_j^{stand}\}).$$

For the sake of readability, we will from now on denote:

- $\{\Pi_{\phi^{\mu_p}}\}$ the orthogonal projection onto $\{\phi_i^{\mu_p},\ i \in [1 \dots N_{red}^p\}$
- $\{\Pi_{\phi^{tot}(\alpha)}\}$ the orthogonal projection onto $\{\phi_j^{tot}(\alpha),\ j \in [1 \dots N_{tot}]\}$
- the projections onto orthogonal sets will be denoted $\Pi^{\perp}$

6.1 A posteriori error bound

Let $(p,k) \in [1 \dots P] \times [1 \dots N_{snap}^p]$. We have:

$$\begin{array}{rcl} \left\| u(\cdot,t^k;\mu_p) - \Pi_{\phi^{tot}(\alpha)} u(\cdot,t^k;\mu_p) \right\| \leq & \left\| u(\cdot,t^k;\mu_p) - \Pi_{\phi^{\mu_p}} u(\cdot,t^k;\mu_p) \right\| \\ + & \left\| \Pi_{\phi^{\mu_p}} u(\cdot,t^k;\mu_p) - \Pi_{\phi^{tot}(\alpha)} \Pi_{\phi^{\mu_p}} u(\cdot,t^k;\mu_p) \right\| \\ + & \left\| \Pi_{\phi^{tot}(\alpha)} u(\cdot,t^k;\mu_p) - \Pi_{\phi^{tot}(\alpha)} \Pi_{\phi^{\mu_p}} u(\cdot,t^k;\mu_p) \right\|. \end{array}$$

We know that the operator norm of $\Pi_{\phi^{tot}}$ is smaller than 1. We thus have:

$$\begin{array}{rcl} \left\| u(\cdot,t^k;\mu_p) - \Pi_{\phi^{tot}(\alpha)} u(\cdot,t^k;\mu_p) \right\| \leq & 2 \left\| u(\cdot,t^k;\mu_p) - \Pi_{\phi^{\mu_p}} u(\cdot,t^k;\mu_p) \right\| \\ + & \left\| \Pi_{\phi^{\mu_p}} u(\cdot,t^k;\mu_p) - \Pi_{\phi^{tot}(\alpha)} \Pi_{\phi^{\mu_p}} u(\cdot,t^k;\mu_p) \right\|. \end{array}$$

As $\left\| u(\cdot,t^k;\mu_p) - \Pi_{\phi^{\mu_p}} u(\cdot,t^k;\mu_p) \right\|$ depends only on the in batch approximations, we have:

$$\forall p \in [1 \dots P],\ \sum_{k=1}^{N_{snap}^p} \left\| u(\cdot,t^k;\mu_p) - \Pi_{\phi^{\mu_p}} u(\cdot,t^k;\mu_p) \right\|^2 \leq \sum_{i > N_{red}^p} \lambda_i^p.$$

To handle the second term, we consider the following dataset:

$$\Xi_{proj} := \left\{ \Pi_{\phi^{\mu_p}} u(\cdot, t^k; \mu_p),\ p \in [1 \dots P],\ k \in [1 \dots N^p_{snap}] \right\}. \tag{6.3}$$

Let $R_{\Xi_{proj}}$ be the standard POD operator, defined in the introductive chapter 1, applied to the previous dataset:

$$R_{\Xi_{proj}} : \left\{ \begin{array}{ccc} X & \to & X \\ \psi & \mapsto & \sum_p \sum_{k=1}^{N^p_{snap}} < \Pi_{\phi^{\mu_p}} u(\cdot, t^k; \mu_p), \psi >_X \Pi_{\phi^{\mu_p}} u(\cdot, t^k; \mu_p). \end{array} \right.$$

For (p, k), denote $\{x^p_{k,i}\}_i$ the coordinates of $u(\cdot, t^k; \mu_p)$ onto the basis $\{\phi^{\mu_p}_i\}$. We have:

$$\forall \psi \in X,\ R_{\Xi_{proj}}(\psi) = \sum_p \sum_{k=1}^{N^p_{snap}} \sum_{i=1}^{N^p_{red}} \sum_{j=1}^{N^p_{red}} x^p_{k,i} x^p_{k,j} < \phi^{\mu_p}_i, \psi > \phi^{\mu_p}_j.$$

We use another key property of the POD method:

$$\forall i, j \in [1 \dots N^p_{red}],\ \sum_{k=1}^{N^p_{snap}} x^p_{k,i} x^p_{k,j} = \lambda^p_i \delta_{i,j}.$$

The proof of this well known result is recalled in the appendix. We thus have:

$$\forall \psi \in X,\ R_{\Xi_{proj}}(\psi) = \sum_p \sum_{i=1}^{N^p_{red}} \lambda^p_i < \phi^{\mu_p}_i, \psi >_X \phi^{\mu_p}_i.$$

This is the same operator as $\tilde{R}(\lambda)$ defined equation (6.2) with the following choice of weights:

$$\forall p \in [1 \dots P],\ \forall i \in [1 \dots N^p_{red}],\ \alpha^p_i := \lambda^p_i.$$

Denote $\left\{\phi^{tot}_j(\lambda)\right\}$ and $\{\lambda^{tot}(\lambda)\}$ the corresponding eigenvector and eigenvalue sets. We use standard property:

$$\sum_p \sum_{k=1}^{N^p_{snap}} \left\| \Pi_{\phi^{\mu_p}} u(\cdot, t^k; \mu_p) - \Pi_{\phi^{tot}(\lambda)} \Pi_{\phi^{\mu_p}} u(\cdot, t^k; \mu_p) \right\|^2 = \sum_{j > N_{tot}} \lambda^{tot}_j(\lambda),$$

and thus reach the first desired result:

$$\begin{array}{rcl} J_{N_{tot}}(\{\phi^{tot}_j(\lambda)\}) \leq 8 & \sum_p \sum_{i > N^p_{red}} \lambda^p_i \\ + & 2 \sum_{j > N_{tot}} \lambda^{tot}_j(\lambda). \end{array} \tag{6.4}$$

We can now safely use the two level POD, as we have a rigorous upper bound on the global error. One possible algorithm is presented below:

Data: Fine discretization Ξ of some solution manifold $\mathcal{M}$, ϵ some threshold
Result: Reduced basis such that $\sum_{u \in \Xi} \|u - \Pi_{\phi^{tot}} u\|^2_X \leq \epsilon$
Decompose Ξ into batches $\{\Xi_p\}$;
Compute $\{\phi^{\mu_p}_i, \lambda^p_i\} =$ POD (Ξ_p) ;
Choose N^p_{red} such that $8 \sum_p \sum_{i > N^p_{red}} \lambda^p_i \leq \frac{\epsilon}{2}$;
Compute $\{\phi^{tot}_j(\lambda), \lambda^{tot}_j(\lambda)\} =$ POD $(\sqrt{\lambda^p_i} \phi^{\mu_p}_i)$;
Choose N_{tot} such that $2 \sum_{j > N_{tot}} \lambda^{tot}_j(\lambda) \leq \frac{\epsilon}{2}$;

Algorithm 11: Performing two level POD

6.2 Comparison with the standard POD

We now turn to the second objective. We want to guarantee that for a fixed accuracy, the basis resulting from a two level POD is not much bigger than the one resulting from a standard POD. We fix N_{tot}. In this section, we restrict ourselves to the case $N^p_{red} \geq N_{tot}$. Indeed, to have a rigorous bound, we need to be able to handle the case $P = 1$, and the case with identical batches. We will denote: $\Pi_{\phi^{stand}}$ the orthogonal projection onto $\{\phi^{stand}_j,\ j \in [1 \dots N_{tot}]\}$. We have one easy inequality:

$$\begin{aligned} J_{N_{tot}}(\{\phi^{stand}_j\}) &= \sum_p \sum_{k=1}^{N^p_{snap}} \|u(\cdot, t^k; \mu_p) - \Pi_{\phi^{stand}} u(\cdot; t^k; \mu_p)\|^2_X \\ &\geq \sum_p \left(\sum_{k=1}^{N^p_{snap}} \|u(\cdot, t^k; \mu_p) - \Pi_{\phi^p} u(\cdot; t^k; \mu_p)\|^2_X \right) \\ &\geq \sum_p \sum_{n > N_{tot}} \lambda^p_n. \end{aligned}$$

This uses the standard POD equality and the fact that for a fixed size, the spaces chosen on each subbasis are better adapted than a global one as well as $N^p_{red} \geq N_{tot}$.

The natural follow up is to compare $\{\lambda^{stand}\}$ and $\{\lambda^{tot}(\lambda)\}$. Recall that $\lambda^{tot}(\lambda)$ is the output of a standard POD algorithm applied to the dataset Ξ_{proj} defined in (6.3). Intuitively, Ξ_{proj} should be 'smaller' than the original dataset, so we expect the $\{\lambda^{tot}(\lambda)\}$ to be not far from the original $\{\lambda^{stand}\}$.

Following this initial remark, we start with a simpler problem. Let Ξ be any dataset in X and let $\{\psi_n\}_n$ be some orthogonal basis in X. We want to compare the decay of the eigenvalues of the POD algorithm applied to Ξ versus the one applied to $\{\Pi_\psi v,\ v \in \Xi\}$. Denote

- $\{\phi^{orig}\}$ and $\{\lambda^{orig}\}$ the eigenvector/eigenvalues sets for the original dataset
- $\{\phi^{proj}\}$ and $\{\lambda^{proj}\}$ the eigenvector/eigenvalues sets of the projected dataset
- Π the projection on the first N basis in both original and projected cases

We know that:

$$\sum_{k \in \Xi} \|u_k - \Pi_{\phi^{orig}} u_k\|^2_X = \sum_{n > N} \lambda^{orig}_n$$

$$\sum_{k \in \Xi} \|\Pi_\psi u_k - \Pi_{\phi^{proj}} \Pi_\psi u_k\|^2_X = \sum_{n > N} \lambda^{proj}_n$$

The optimal property of the basis $\{\phi^{proj}\}$ gives:

$$\sum_{k \in \Xi} \|\Pi_\psi u_k - \Pi_{proj} \Pi_\psi u_k\|^2 \leq \sum_{k \in \Xi} \|\Pi_\psi u_k - \Pi_{origin} \Pi_\psi u_k\|^2 .$$

$$\sum_{n > N} \lambda^{proj}_n \leq 2 \sum_{k \in \Xi} \|\Pi_\psi \left(u_k - \Pi_{origin} u_k\right)\|^2 + 2 \left\|\left(\Pi_\psi \Pi_{origin} - \Pi_{origin} \Pi_\psi\right) u_k\right\|^2 .$$

Using the fact that the orthogonal projector has unit norm, we have

$$\sum_{n > N} \lambda^{proj}_n \leq 2 \sum_{n > N} \lambda^{origin}_n + 2 \left\|\left(\Pi_\psi \Pi_{origin} - \Pi_{origin} \Pi_\psi\right) u_k\right\|^2$$

As $\{\phi^{proj}\}$ is the POD basis of the projected set, we know that $\phi^{proj} \in$ span $\{\psi_n\}$, and thus that:

$$\Pi_\psi \Pi_{\phi^{proj}} = \Pi_{\phi^{proj}} \Pi_\psi = \Pi_{\phi^{proj}} .$$

Unfortunately, this is not true for the original basis $\{\phi^{orig}\}$. We need to use something else:

$$(\Pi_\psi \Pi_{origin} - \Pi_{origin}\Pi_\psi) = \left(\Pi_\psi^\perp \Pi_{origin}^\perp - \Pi_{origin}^\perp \Pi_\psi^\perp\right)$$

This concludes this digression and we go back to our original problem. As $\{\phi_j^{tot}\}$ is optimal to represent the set Ξ_{proj} defined in (6.3), we have:

$$\begin{aligned}
\sum_{j>N_{tot}} \lambda_j^{tot}(\lambda) &\leq \sum_p \sum_{k=1}^{N_{snap}^p} \left\|\Pi_{\phi^{\mu_p}} u(\cdot,t^k;\mu_p) - \Pi_{\phi^{stand}}\Pi_{\phi^{\mu_p}} u(\cdot,t^k;\mu_p)\right\|^2 \\
&\leq 2\sum_p \sum_{k=1}^{N_{snap}^p} \left\|\Pi_{\phi^{\mu_p}} \left(u(\cdot,t^k;\mu_p) - \Pi_{\phi^{stand}} u(\cdot,t^k;\mu_p)\right)\right\|^2 \\
&\quad +2\sum_p \sum_{k=1}^{N_{snap}^p} \left\|\Pi_{\phi^{\mu_p}}\Pi_{\phi^{stand}} u(\cdot,t^k;\mu_p) - \Pi_{\phi^{stand}}\Pi_{\phi^{\mu_p}} u(\cdot,t^k;\mu_p)\right\|^2 \\
\sum_{j>N_{tot}} \lambda_j^{tot}(\lambda) &\leq 2\sum_{n>N_{tot}} \lambda_n^{stand} + 2\sum_p \sum_{k=1}^{N_{snap}^p} \left\|\left(\Pi_{\phi^p}^\perp \Pi_{\phi^{stand}}^\perp - \Pi_{\phi^{stand}}^\perp \Pi_{\phi^p}^\perp\right) u(\cdot,t^k;\mu_p)\right\|^2 \\
&\leq 2\sum_{n>N_{tot}} \lambda_n^{stand} + 4\sum_p \sum_{k=1}^{N_{snap}^p} \left(\left\|\Pi_{\phi^{stand}}^\perp u(\cdot,t^k;\mu_p)\right\|^2 + \left\|\Pi_{\phi^p}^\perp u(\cdot,t^k;\mu_p)\right\|^2\right) \\
&\leq 10\sum_{n>N_{tot}} \lambda_n^{stand}
\end{aligned}$$

We have once again used $N_{red}^p \geq N_{tot}$.

As a conclusion, we went from

$$\sum_p \sum_{k=1}^{N_{snap}^p} \left\| u(\cdot,t^k;\mu_p) - \sum_{j=1}^{Ntot} \left\langle u(\cdot,t^k;\mu_p), \phi_j^{stand}\right\rangle_X \phi_j^{stand} \right\|_X^2 \leq \sum_{n>Ntot} \lambda_n^{stand} \tag{6.5}$$

to

$$\sum_p \sum_{k=1}^{N_{snap}^p} \left\| u(\cdot,t^k;\mu_p) - \sum_{j=1}^{Ntot} \left\langle u(\cdot,t^k;\mu_p), \phi_j^{tot}(\lambda)\right\rangle_X \phi_j^{tot}(\lambda) \right\|_X^2 \leq 28 \sum_{n>Ntot} \lambda_n^{stand}. \tag{6.6}$$

For manifolds for which the standard POD gives quickly decaying eigenvalues, the size of the final basis with a two level POD method is guaranteed not to be much bigger than the one obtained with the standard POD algorithm.

6.3 Computational cost

What have we gained ? For simplicity, let's say that each batch has size $\frac{N_{snap}}{p}$. We need to perform

- P sub POD computations, of size $\frac{N_{snap}}{P}$. That amounts to a complexity of $P\mathcal{N}(\frac{N_{snap}}{P})^2$
- Let N_{red} be a typical sub basis side (one of the N_{red}^p). The recombination requires $\mathcal{N}(PN_{red})^2$ computations

What is the tradeoff then ? The bigger P, the cheaper the sub POD basis computations. But in order to have decent accuracy, one must take enough basis at the recombination step. Typically, one must take for each batch the same number of basis as in the final basis, see section 6.2.

Hypothetical situation. We have 20 parameters. Each time simulation consists in 300 time steps, and we are expecting a final basis of size 40. Using standard POD costs $(20*300)^2\mathcal{N}$. This double POD costs $(20*300^2 + (40*20)^2)\mathcal{N}$. We have reduced the total computational cost by 20.

Appendix

Let Ξ_p be one batch and R_{Ξ_P} the standard POD operator on this set:

$$R_{\Xi_p} : \left\{ \begin{array}{lcl} X & \rightarrow & X \\ \psi & \mapsto & \sum_{k=1}^{N_{snap}^p} < u(\cdot, t^k; \mu_p), \psi >_X u(\cdot, t^k; \mu_p). \end{array} \right.$$

Using the fact that $\{\phi_i^{\mu_p}\}$ is a set of eigenfunctions of R_{Ξ_p} and is orthogonal, we have:

$$\begin{array}{rcl} \forall i, j, < R_{\Xi_p}(\phi_i^{\mu_p}), \phi_j^{\mu_p} >_X & = & \lambda_i^p < \phi_i^{\mu_p}, \phi_j^{\mu_p} >_X \\ & = & \delta_{i,j} \lambda_i^p. \end{array}$$

Denote for all triplet (i, k, p), $x_{k,i}^p :=< u(\cdot, t^k; \mu_p), \phi_i^{\mu_p} >_X$. We have the desired result:

$$\forall i, j \in [1 \dots N_{red}^p], \sum_{k=1}^{N_{snap}^p} x_{k,i}^p x_{k,j}^p = \lambda_i^p \delta_{i,j}.$$

Chapter 7

ROM and big data: a common methodology

In this short bonus chapter we will discuss the new trend in the reduced order modeling community that consists in adapting 'big data' methods. The underlying idea is to replace part/all of a PDE resolution by some learning algorithm inspired by big data.

We do not intend to present a thorough list of applications but have chosen a small set of examples, that illustrate the broad spectrum of possibilities. In a recent paper [58], they propose an original way to combine ROM and big data. The physical domain is decomposed, using some a priori knowledge of the solution, into disjoint subdomains. The reduced basis in each subdomain is then chosen among a set of reduced basis using some learning process. In [72], they adapt the Local Linear Embedding (LLE), that was originally developed for image processing, to the numerical resolution of PDEs. More precisely, they build a 'model free' method for elasticity problems. The modeling of the relation between displacement field and stress tensor is replaced by a learning algorithm. Finally, in [137] they have chosen to evolve the POD coefficients of a Navier-Stokes simulation, using a neural network.

The focus in this short chapter is put on a method with growing importance in a part of the reduced order modeling community. This method is referred to as Dynamic Mode Decomposition (DMD) [120, 132, 117]. It uses a learning algorithm in order to deal with the computational complexity incurred by the presence of non linear functions of the solution, in a ROM context. We end this chapter, by presenting a fully 'model free' solver that has been developed in the computer graphics community [76].

7.1 DMD

DMD aims at accelerating the online computation of non linear functions of the solution, while solving some PDE. Figure 7.1 shows a typical situation. We have computed snapshots over some solution manifold $\mathcal{M}$, $\{u_k\}_k$ and have computed their image through some non linear application F, $\{F(u_k)\}_k$. The objective is then to find a 'cheap' way of computing the image of any member of $\mathcal{M}$ through F. We have presented in the introductive chapter, section 1.5.2.1 the most common way of dealing with this in a ROM context, the Empirical Interpolation Method (EIM).

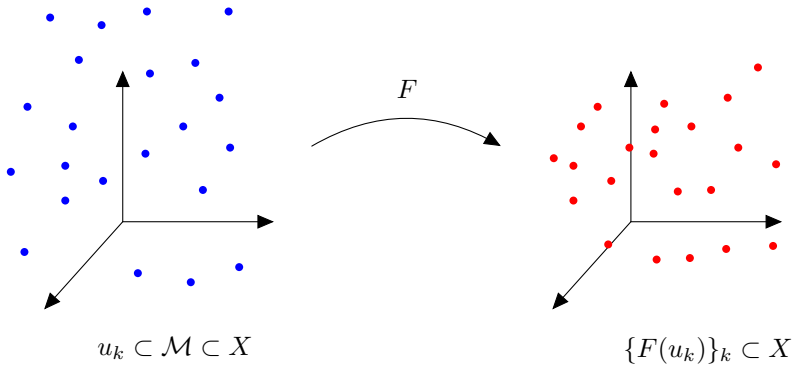


Figure 7.1: Non linear operator acting on the solution manifold

DMD method, as the LLE method mentioned earlier, is of a different type. It handles the non linear dependency by building an equivalent linear model. The associated computational gain is obvious. The novelty compared to other linearization methods is its theoretical foundation. The method supposes the existence of a space, somehow related to $\mathcal{M}$, on which the possibly highly non linear application F, is linear. A graphical representation can be found Figure 7.2

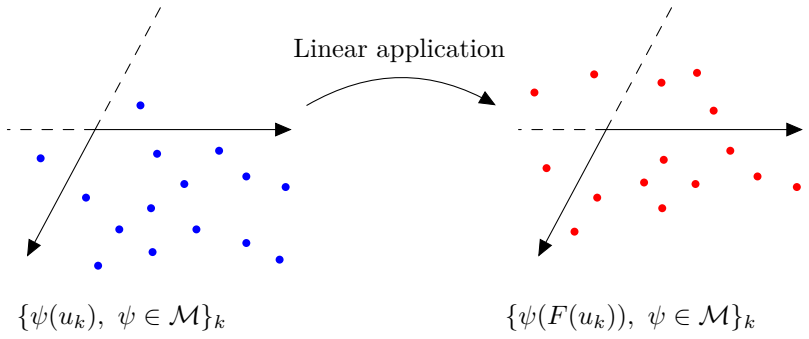


Figure 7.2: The action of F, on the subspace, is simpler

The method relies on the Koopman theory, and more precisely on the construction of a so-called Koopman Operator. In this section, we denote:

- X either the manifold of continuous solutions, or equivalently a 'truth' discretized space
- a generic non linear application $X \to X$, denoted by F
- $\mathcal{O}$ some observable space over X, that is, a set of functionals over X. Note that these are not necessarily linear

In that framework, the dots of Figure 7.2 are the images of the elements of X through the observables $\mathcal{O}$. The latter can for instance be thought of as the scalar product with some reduced basis $\{\phi_i\}_i$:

$$\mathcal{O} := \left\{ \begin{array}{ccc} X & \to & \mathbb{R} \\ u & \mapsto & < u, \phi_i >_X,\ i \in [1, \dots, N^{red}], \end{array} \right.$$

or, in the spirit of EIM, as the evaluation of a function X at N^{red} discretization points in Ω:

$$\mathcal{O} := \left\{ \begin{array}{rcl} X & \rightarrow & \mathbb{R} \\ u & \mapsto & u(x_k),\ x_k \in \Omega, k \in [1, \dots, N^{red}]. \end{array} \right.$$

Whatever the space of observables, the idea is to replace the non linear transformation on X by a linear transformation on $\mathcal{O}$. For that, the key component is the Koopman operator $\mathcal{K}$, a linear operator acting on the vector space spanned by the elements of $\mathcal{O}$ as:

$$\forall g \in \text{ span } \mathcal{O},\ \mathcal{K}g = g \circ F. \tag{7.1}$$

Note that the previous equality is an equality between functionals over X. We can now reformulate the premise of the DMD method: it supposes that the complicated non linear behavior, restricted to the reduced basis (or some other observables space) is well represented by a linear transformation.

This is precisely the assumption that we discuss in this section. To get a little more context, I have chosen to reproduce a paragraph found in [132], which is among the most cited paper in the DMD literature:

> Without these connections, the use of DMD to analyze nonlinear dynamics appears dubious, since there seems to be an underlying assumption of (approximately) linear dynamics (see Section 3.1), as in (1). One might well question whether such an approximation would characterize a nonlinear system in a meaningful way. However, so long as DMD can be interpreted as an approximation to Koopman spectral analysis, there is a firm theoretical foundation for applying DMD in analyzing nonlinear dynamics.

It is the firm theoretical foundation that we will try to understand.

7.1.1 The Koopman operator

We show in this section how the DMD community intends to use the knowledge of the eigenfunctions of $\mathcal{K}$ to solve the target objective. Suppose that there exists a complete set of eigenfunction/eigenvalue pairs $\{(\psi_j, \lambda_j), j = 1 \dots N^{red}\}$ for the operator $\mathcal{K}$. In other words, we suppose that:

$$\forall g \in \text{ span } \mathcal{O},\ \exists \{c_j(g)\} \in \mathbb{R}^{N^{red}},\ \text{ s.t },\ g = \sum_{j=1}^{N^{red}} c_j(g)\psi_j.$$

Let $g \in \mathcal{O}$. The objective is to be able to compute efficiently $g \circ F$ over X. With the previous assumption, we have:

$$\exists \{c_j(g)\} \in \mathbb{R}^{N^{red}},\ g \circ F = \sum_{j=1}^{N^{red}} c_j(g)\left(\psi_j \circ F\right). \tag{7.2}$$

Using the Koopman definition, and the fact that $\{\psi_j\}$ is an eigenfunction set, we have:

$$g \circ F = \sum_{j=1}^{N^{red}} c_j(g)\left(\mathcal{K}\psi_j\right) = \sum_{j=1}^{N^{red}} \lambda_j c_j(g)\psi_j. \tag{7.3}$$

The rest of the derivation will be done in the case where the observables are taken as the scalar products with some reduced basis $\{\phi_i\}_i$, but note that there would be no difference for other choices of observables spaces. To avoid confusion, we denote by $\tilde{\phi}_i$ the observable in $\mathcal{O}$ that corresponds to the scalar product with ϕ_i.

Offline, start by computing the couples $\{(\psi_j, \lambda_j)\}_j$. We show in the next section how this is performed. Then, compute the coordinates of the observables onto the space spanned by the eigenfunctions of $\mathcal{K}$: the set $\{c_j(\tilde{\phi}_i)\}_{i,j}$.

Online, evaluate the observables applied to the non linear term as:

$$\forall u \in \mathcal{M},\ \forall i,\ \tilde{\phi}_i(F(u)) =< F(u), \phi_i >_X = \sum_{j=1}^{N^{red}} \lambda_j c_j(\tilde{\phi}_i)\psi_j(u) \tag{7.4}$$

From this discussion, we need a priori N^{red} eigenfunctions of $\mathcal{K}$. We will come back on this in section 7.1.4.

In order to fit DMD into a standard offline/online paradigm, we need more assumptions on the set of eigenfunctions $\{\psi_j\}$. By inspecting the form of the quantities needed online, see (7.4), the functionals in the eigenfunction set either need to have a computationally cheap action on X (linear or quadratic functionals for instance) or to involve a reduced number of spatially localized estimates. This of course restricts the observable spaces $\mathcal{O}$ that can be considered. The next section is devoted to the search for approximate eigenfunctions.

7.1.2 Details on the offline stage

The objective of this section is to propose a method to compute some[1] eigenfunctions of the Koopman operator. We are still in the situation where the observables $\{\tilde{\phi}_i\}$ are the projection onto some reduced basis $\{\phi_i\}$. Let $\{u_j\}_j$ be a representative/well chosen snapshot set of $\mathcal{M}$ in X, of cardinality M. Define U as the following $N^{red} \times M$ matrix:

$$U_{ij} := \tilde{\phi}_i(u_j) =< u_j, \phi_i >_X \tag{7.5}$$

Compute the image through the non linear application F, and store the result into Y:

$$Y_{ij} := \tilde{\phi}_i(F(u_j)) =< (F(u_j)), \phi_i >_X . \tag{7.6}$$

Denote $A \in \mathbb{R}^{N^{red} \times N^{red}}$ any linear model fitting X to Y. The most common choice is to use the least squares approximations:

$$A = \underset{B \in \mathbb{R}^{N^{red} \times N^{red}}}{\operatorname{argmin}} \|BU - Y\|_2^2$$

given by:

$$A = YU^+ \tag{7.7}$$

where $\cdot^+$ denotes the Moore Penrose pseudo inverse.

There is only one result in the DMD literature on the computation of eigenfunctions of $\mathcal{K}$. It links eigenvectors of A with eigenfunctions of $\mathcal{K}$ under very stringent conditions. Let ψ be an eigenfunction of $\mathcal{K}$, and λ its eigenvalue. As it is in $\operatorname{span}\{\tilde{\phi}_i\}$, the set of observables, we have:

$$\exists \{\beta_i\} \in \mathbb{R}^{N_{red}},\ \psi = \sum_{i=1}^{N^{red}} \beta_i \tilde{\phi}_i.$$

[1] Note that we have no results on the existence nor on the number of eigenfunctions of $\mathcal{K}$.

Using the linearity of the Koopman operator, we know:

$$\mathcal{K}\psi = \sum_{i=1}^{N^{red}} \beta_i \mathcal{K}\tilde{\phi}_i = \sum_{i=1}^{N^{red}} \beta_i \tilde{\phi}_i \circ F \tag{7.8}$$

Let

- $u_k \in X$ a solution in the training set (used to build A)
- U_k the N^{red} dimensional vector $\left\{\tilde{\phi}_i(u_k) =< \phi_i, u_k >_X\right\}_i$
- Y_k the vector $\left\{\tilde{\phi}_i(F(u_k)) =< \phi_i, F(u_k) >_X\right\}_i$

We know that

$$\forall k,\ AU_k = Y_k + \mathbf{r}_k \tag{7.9}$$

where $\mathbf{r}_k \in \mathbb{R}^{N^{red}}$ is residual of the linear fit. We can think of it as the least square residual. Now,

$$\begin{aligned} \forall k,\ \mathcal{K}\psi(u_k) &= \beta^T Y_k = \beta^T A U_k - \beta^T \mathbf{r}_k \\ &= \lambda \beta^T U_k. \end{aligned} \tag{7.10}$$

The first equality comes from equation (7.8). Second equality comes from equation (7.9). The last equality comes from the fact that ψ is an eigenfunction of $\mathcal{K}$. If the residual of the least squares approximation vanishes, we have:

$$\forall k,\ \beta^T A U_k = \lambda \beta^T U_k. \tag{7.11}$$

Equation (7.11) is equivalent to:

$$\beta^T A U = \lambda \beta^T U.$$

We are ready to state the result: if ψ is an eigenfunction of $\mathcal{K}$ such that:

- ψ is in the span of the observables
- the linear fit (say least squares approximation) is exact on the training set $\{u_k\}_k$
- U is of rank N^{red}. This is equivalent to UU^T invertible, as we expect $M \gg N^{red}$. The underlying idea is that $\{u_k\}$ needs to be a big enough sample

then ψ can can be found by computing the eigenvectors of A^T. More precisely, there exists β eigenvector of A^T such that

$$\psi := \sum_{i=1}^{N^{red}} \beta_i \tilde{\phi}_i.$$

Before moving on to a simple illustrative example, we discuss the hypotheses made in this section. First of all, as the residual will most likely not vanish, we have no result guaranteeing that the error made on the approximate eigenfunction is controlled by the residual error. More importantly, we have no proof of the existence of eigenfunctions of $\mathcal{K}$ in the space spanned by the observables, for a general function $F : X \to X$.

7.1.3 Discrete system

The theory of Koopman operator was first developed for discrete systems. One two dimensional system example is developed in [116]. The model non linearity is chosen as:

$$F := \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \mapsto \begin{bmatrix} \lambda z_1 \\ \mu z_2 + (\lambda^2 - \mu) c z_1^2 \end{bmatrix} \tag{7.12}$$

for some λ, μ and c. Suppose that we take z_1, z_2 and z_1^2 as observables. Define ϕ_1 and ϕ_2 as:

$$\begin{aligned} \phi_1(z) &= z_1 \\ \phi_2(z) &= z_2 - c z_1^2. \end{aligned} \tag{7.13}$$

Easy computations show that both these functions are eigenfunctions of the Koopman operator:

$$\phi_1 \circ F \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \lambda z_1 = \lambda \phi_1 \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$$\phi_2 \circ F \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \mu z_2 + (\lambda^2 - \mu) c z_1^2 - c\lambda^2 z_1^2 = \mu \phi_2 \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

The associated eigenvalues are λ and μ respectively. In this carefully tailored example, ϕ_1 and ϕ_2 are the desired Koopman eigenfunctions in the span of the observables. Unfortunately, even in this toy example, we do not really solve the computational cost issue, as the evaluation of $F \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$ still requires the evaluation of z_1, z_2 and z_1^2. We now go back to a more reasonable example, using the usual reduced order modeling framework.

7.1.4 A more realistic example

Let $\mathcal{M}$ be some continuous solution manifold to some PDE, embedded in a well chosen Hilbert space X. An eigenfunction ψ in the span of the observables satisfies:

$$\begin{cases} \exists \{\alpha_i\}_i, \ \text{s.t} \quad \psi & = & \sum_{i=1}^{N^{red}} \alpha_i \tilde{\phi}_i \\ \forall u \in \mathcal{M}, \qquad \mathcal{K}\psi(u) & = & \lambda \psi(u) = \psi \circ F(u). \end{cases}$$

When $\mathcal{O}$ is the space of scalar products with a reduced basis, this condition becomes:

$$\exists \lambda, \ \exists \{\alpha_i\} \in \mathbb{R}^{N^{red}}, \ \ \text{s.t} \ \ \forall u \in \mathcal{M}, \ < \lambda u - F(u), \sum_{i=1}^{N^{red}} \alpha_i \phi_i >_X = 0. \tag{7.14}$$

Recall that in order to have a proper offline/online decomposition, we need enough of such eigenfunctions. We can state a necessary and sufficient condition: there exists $N \in \mathbb{N}$, $\{\psi_j \in \text{ span } \mathcal{O}, j \in [1, \dots, N^{red}]\}$ and $\{\lambda_j \in \mathbb{R}, j \in [1, \dots, N]\}$ such that:

$$\forall u \in \mathcal{M}, \ \begin{cases} \forall j \in [1, \dots, N], & < F(u), \psi_j > & = & \lambda_j < u, \psi_j > \\ \forall j > N, & < F(u), \psi_j > & = & 0. \end{cases} \tag{7.15}$$

This is a very stringent condition, and we cannot expect it to be satisfied for general F. I will try to make my point a little clearer: in my opinion, the DMD method does not constitute a proof of the validity of the linear model. It is merely a way of finding the components with a 'close to linear' behavior in a non linear system.

7.1.5 Link with EIM

Using as observable a small set of pointwise evaluations, say $\{u(x_i),\ u \in \mathcal{M}\}_i$ can lead to think that this method is some kind of generalization of EIM, allowing for a larger choice of observable space $\mathcal{O}$. This is not the case, as DMD linearly fits:

$$\{u(x_i), i \in [1, \dots, N^{red}]\} \to \{F(u)(x_i),\ i \in [1, \dots, N^{red}]\}.$$

The power of EIM is that it does not reduce a non linearity to a linear (or locally linear) problem, as do DMD and LLE methods.

7.1.6 Conclusions on DMD

I believe that the DMD's theoretical analysis forgets that the eigenfunction of the Koopman operator, if any, are out of reach. My intuition is that the eigenvectors coming from the least squares approximation performed offline have nothing to do with eigenfunctions of the Koopman operator.

I have chosen to present this method in the big data section because it is often implied in the DMD literature that if you feed enough data, you will eventually find these Koopman eigenfunctions. My opinion on this method is that it is a 'best linear fit' approximation of a non linear model. In the literature, they advise using the DMD linear proxy only locally in X. DMD can then be assimilated to a local linearization, with all the limitations that this implies. We end this bonus chapter by peaking into the reduction of complexity as imagined by another community.

7.2 Machine learning

We have chosen to end this chapter by presenting a model free method developed in the machine learning community. Indeed, as stated many times in this manuscript, ROM and machine learning have a lot in common, in terms of methodology. Tabular 7.1 recalls the equivalence between notions. The strength of reduced order modeling is the strong theoretical background

Community	Reduced Order Modeling	Machine Learning
Stages	Offline	Training stage
	Online	Prediction/Classification
Offline	Reduced basis construction	Feature selection
Online	Reduced basis	Feature vector
	Reduced Scheme	Regressor
Error estimation	A priori error estimation	Validation set
	A posteriori error estimation	??

Table 7.1: Equivalence between ROM and machine learning

that ensures among other things, stability results and rigorous error estimations.

In this section, we follow the lines of [76]. Instead of working at a macroscopic scale on averaged quantities, we use a microscopic description and study the behavior of individual particles. From now on, we fix a set of $\mathcal{N}$ particles characterized by their positions and velocities.

$$\forall t,\ \{\mathbf{X}_k(t) = (x_k(t), v_k(t)),\ k \in [1 \dots \mathcal{N}]\}.$$

How does a standard machine learning method goes ?

- find, in a training step, a correct set of features characterizing the state and environment of a particle
- in a training set still, find a good regressor. It returns the quantity of interest from the feature vectors
- online, for each evaluation, compute the feature vectors then use the regressor to classify/-cluster/predict

A first naive idea, is to use brute force, with no modeling whatsoever. Let k be one particle. Let $J(k)$ be the index of some 'relevant' neighbors. Use as feature vector the relative positions and velocity of particles in $J(k)$ to update the velocity and the speed of particle k. In compact form, we want to build a regressor that does the following:

$$\{x_j(t^n), v_j(t^n),\ j \in J(k)\} \to \left(x_k(t^{n+1}), v_k(t^{n+1})\right).$$

This specific choice of feature vector uses translation invariance, through the choice of 'local neighbors'. The main problem here is that with no modeling imposed, we are not using any inertia invariance property. This problem is too complex, and as a consequence, we go from 'no model' to 'rough model' still in the lines of [76]. The method chosen is derived from the Smoothed Particle Hydrodynamics (SPH) method, see for instance [91].

The features are taken as averaged quantities around each particle. For instance, let A be some quantity of interest. It is evaluated at each point x of Ω as:

$$A(x) = \sum_{j \in J(x)} \frac{m_j}{\rho_j} A_j W(x - x_j)$$

where W is a smooth 'window' function, (m_j, ρ_j) are respectively the mass and the density associated to particle j and $J(x)$ is some neighborhood of x. The reason for this specific form is clear when you evaluate the density at x:

$$\tilde{\rho}(x) = \sum_{j \in J(x)} m_j W(x - x_j).$$

Now, let some particle $\mathbf{X}(t) = (x(t), v(t))$. The time derivative of the velocity of this particle is given by

$$\frac{d}{dt} v\left(x(t), t\right) = \frac{\partial}{\partial t} v + v \cdot \nabla v$$

We know that this time derivative will depend on viscous and pressure forces. The choice of features follows naturally:

- $\Phi^{visc} \approx -\Delta v(x)$
- $\Phi^{pressure} \approx \nabla p(x)$

More advanced models could consider more forces such as surface tension for instance.

The regressor in that situation is chosen as a function that estimates the acceleration of particle j from the local feature vectors. It is constructed, using some learning algorithm, during the training stage:

$$\left(\Phi^{visc}(x_j, t^n), \Phi^{pressure}(x_j, t^n)\right) \to a_j(t^{n+1})$$

The last thing to do in order to close the method is to choose an approach to update the full state vector:

$$\begin{cases} \{v_j(t^n), a_j(t^n), a_j(t^{n+1})\} & \rightarrow \quad v_j(t^{n+1}) \\ \{x_j(t^n), v_j(t^n), v_j(t^{n+1}), a_j(t^n), a_j(t^{n+1})\} & \rightarrow \quad x_j(t^{n+1}) \end{cases}$$

When using this 'model free' approach, we are of course far from solving the Navier-Stokes equation. As mentioned in the introduction, in this framework we loose all error estimations, as well as key physical properties such as incompressibility, positive pressure etc.

7.3 Conclusion

This concludes this short chapter. We have presented big data ideas, sometimes applied directly in the ROM framework, or at least that follow a similar methodology and computational cost reduction objective. We have also, by dissecting the DMD method, presented evidence on the fact that, when facing a challenging problem, feeding data to an empiric model is not (always) enough. Where EIM correctly preserves the non linear nature of a transformation and provides rigorous error estimates, the DMD does linearly fits, whatever the amount of data assimilated.

Chapter 8

Conclusion

Looking back at the topics discussed in this thesis, it seems, at first glance, like we have ended up pretty far from the initial objective stated in the introductory chapter. We take advantage of this concluding chapter to try once more to give some coherence to the work that has been presented, and to conclude on the next major steps towards the objective.

In the introductive chapter, I have highlighted the fact that a raw ROM approach, directly applied to the target problem, was doomed to fail. The first problem, the geometric variability, has been discussed in chapter 2 and in section 3.6. To solve it, we have introduced the notion of local solution manifolds and local Kolmogorov n-width. The offline section was pretty straightforward, and the focus has been put on the construction of an efficient matching method. The second problem is not specific to the target objective, and the issues raised can be transposed to the resolution of a wide variety of problems. The starting point was to notice that because of the profusion of ROM related methods, and the fact that they can be combined, and used one of top of the other, some of the literature tends to forget some of the key, mandatory, requirements for ROM. For instance, if the solution manifold has a large n-width, no POD/RB/EIM combo will ever give satisfactory results. We have discussed the illustrative case of a steep convection problem in chapters 1 and 3. We have shown how advanced numerical stabilization mechanisms for CFD problems also fall into this category. Thus, this issue needs to be dealt with to solve the target problem. We have proposed in the end of the introductive chapter two classes of methods. Chapters 3, 4 and 5 are devoted to the analysis of one of them, the so-called calibration.

Calibration is sometimes being criticized because of its supposed lack of robustness. From what I have understood, these comments are often made because calibration can be seen, at first glance, as a variation around the shock fitting method. Indeed, the calibration function can be seen as a 'best known shock position'. In that setting, the physically relevant speed, given by the RH condition, is replaced by an empiric optimization procedure. Of course, seen as such, this method does not seem very appealing. It suffers the same drawbacks as shock fitting (namely the difficulty to extend to higher dimensions), and is even worse because it uses a non physical shock speed.

In section 3.8, we propose an alternative interpretation: calibration as an h or p adaptive method, tailored for ROM. We try to show here why this is a more suited analogy than shock fitting, and answer at the same time some of the robustness concerns. First of all, the calibrated function does not need to capture the exact the position of the shock. Using a larger calibrated

manifold (with space derivatives for instance) would allow for some error in the calibration function. Also, calibration can be used for more general problems than hyperbolic problems with shocks, see section 3.9.

Shed in that light, we have one common property of all the methods proposed in this thesis. They are a transposition of h or p adaptivity, that end up either in tailored domain decomposition, calibration, or both. Instead of refining the computational mesh or of increasing the degree of the (polynomial or Fourier) underlying basis, as usually performed in standard adaptive methods, we directly tailor a basis. This fits perfectly with the ROM framework, as the premise is that we have an a priori idea of the (local or global) solution manifolds at hand. We are not seeing the underlying computational mesh, but look directly for an adapted basis at a continuous level.

The important ingredient of h and p adaptive methods is the construction of a refinement function. In the most simple cases, the latter can be taken as the zones where the solutions encounters large variations. This can for instance be measured by zones with large gradients, or where a reconstructed Hessian has a large component. Other, more involved, methods require the computation of error estimators. As the output of say a FE method is the best one can say for a given computational mesh, and given trial space and test space, these error estimators necessarily involve subcell computations. The learning phase of the ROM framework helps us in that respect. To avoid any additional expensive, online, procedure, we rather precompute and store offline quantities that help estimate the quality of a mesh (or more precisely a basis) to represent a given solution manifold. The error estimator we have chosen in this thesis is the most natural one and is given by the best projection error. Other choices could be numerically investigated.

With this interpretation, the robustness issues that can be (and have been) raised for the methods developed in this thesis have their exact counterpart in standard adaptive methods, and thus, although relevant, these issues are certainly not sufficient to reject calibration as a numerically viable procedure. What is true though, is that more numerical experiments should be performed to assess the overall method. The intuition is that one should not perform adaptation (and thus should not update the calibration parameter) at each time step, as this implies numerical errors. There is a compromise to be found between reducing the n-width of the calibrated solution manifold, but not having to update the calibration parameter at each time step. One natural option would be to balance the estimation of the numerical errors due to the underlying mesh interpolation, with the best projection error on the reduced basis.

I now discuss a (pretty big) missing ingredient in this thesis: the lack of numerical experiments in some of the chapters. The domain decomposition method proposed in chapter 2 needs to be numerically investigated, and this has not been done in this manuscript. The reason is that I have tried to solve the matching problem and the stability issues inherent to CFD computations simultaneously. A more realistic next step is to apply this method to simpler, smoother problems. One evidence showing that this is a reasonable method is the fact that the ORBEM method shares many components with the Arlequin method [44], and that the latter has been successfully implemented on solid mechanics examples.

The optimal control method for solutions with shock proposed in chapter 5 also requires numerical investigation. Optimal control for the one dimensional Burgers equation is a well studied problem, and the comparison of the results obtained with the reduced method we propose is an objective within hand's reach. In the end of the chapter, we extend the method to the optimal design of an airfoil. There is still a long way to go before a complete framework and numerical experiments. We refer to the conclusion of the chapter for the unresolved issues. In any case, the roadmap has been set up and the scariest part, which is the resolution of an equivalent calibrated problem has been successfully tested in the numerical section of chapter 4.

We can now conclude on what's left to be done in order to solve the initial objective. We split its resolution into two disjoint components:

- a robust local reduced model around one wind turbine. For this, we can either transpose the stability and accuracy features of a fine solver to a ROM model: this will most likely involve some form of calibration. The other option is to construct a self sufficient reduced scheme, with a tailored stabilization mechanism, see section 1.6.1.
- a robust and accurate matching method. As discussed in chapter 2, it should allow for independent rotations in each domains and translations between domains

Unfortunately, this thesis has not lead to firm conclusions on either of these aspects. Nevertheless, I believe that some of the ideas and methods discussed have paved the way to a new class of ROM methods, that could be used to solve the target problem, and more generally to solve problems that were previously out of reach.

Bibliography

[1] Rémi Abgrall. Residual distribution schemes: current status and future trends. Computers & Fluids, 35(7):641–669, 2006.

[2] Rémi Abgrall, David Amsallem, and Roxana Crisovan. Robust model reduction by L1-norm minimization and approximation via dictionaries: application to nonlinear hyperbolic problems. Advanced Modeling and Simulation in Engineering Sciences, 3(1), 2016.

[3] Yves Achdou. The mortar element method with overlapping subdomains. SIAM Journal on Numerical Analysis, 40(2):601–628, 2002.

[4] Robert A Adams and John JF Fournier. Sobolev spaces, volume 140. Academic press, 2003.

[5] Charu C Aggarwal and Chandan K Reddy. Data clustering: algorithms and applications. CRC press, 2013.

[6] Navid Allahverdi, Alejandro Pozo, and Enrique Zuazua. Numerical aspects of large-time optimal control of burgers equation. ESAIM: Mathematical Modelling and Numerical Analysis, 50(5):1371–1401, 2016.

[7] David Amsallem and Charbel Farhat. Interpolation method for adapting reduced-order models and application to aeroelasticity. AIAA journal, 46(7):1803–1813, 2008.

[8] John David Anderson and J Wendt. Computational fluid dynamics, volume 206. Springer, 1995.

[9] JP Argaud, B Bouriquet, H Gong, Y Maday, and O Mula. Stabilization of (g) eim in presence of measurement noise: application to nuclear reactor physics. arXiv preprint arXiv:1611.02219, 2016.

[10] Antonio Baeza, Carlos Castro, Francisco Palacios, and Enrique Zuazua. 2-d euler shape design on nonregular flows using adjoint rankine-hugoniot relations. AIAA journal, 47(3):552, 2009.

[11] Joan Baiges, Ramon Codina, and Sergio Idelsohn. Explicit reduced-order models for the stabilized finite element approximation of the incompressible navier–stokes equations. International Journal for Numerical Methods in Fluids, 72(12):1219–1243, 2013.

[12] Francesco Ballarin. Reduced-order models for patient-specific haemodynamics of coronary artery bypass grafts. 2015.

[13] Claude Bardos and Olivier Pironneau. Derivatives and control in the presence of shocks. Computational Fluid Dynamics Journal, 11(4):383–391, 2003.

[14] Maxime Barrault, Yvon Maday, Ngoc Cuong Nguyen, and Anthony T Patera. An âĂŸempirical interpolation'method: application to efficient reduced-basis discretization of partial differential equations. Comptes Rendus Mathematique, 339(9):667–672, 2004.

[15] Rebecca Jane Barthelmie, Sten Tronæs Frandsen, MN Nielsen, SC Pryor, P-E Rethore, and Hans Ejsing Jørgensen. Modelling and measurements of power losses and turbulence intensity in wind turbine wakes at middelgrunden offshore wind farm. Wind Energy, 10(6):517–528, 2007.

[16] Hachmi Ben Dhia and Guillaume Rateau. Analyse mathématique de la méthode arlequin mixte. Comptes Rendus de l'Académie des Sciences-Series I-Mathematics, 332(7):649–654, 2001.

[17] Marsha Berger, Michael J Aftosmis, and Scott M Murman. Analysis of slope limiters on irregular grids. AIAA paper, 490(2005):1–22, 2005.

[18] Michel Bergmann, C-H Bruneau, and Angelo Iollo. Enablers for robust pod models. Journal of Computational Physics, 228(2):516–538, 2009.

[19] Wolf-Jürgen Beyn and Vera Thümmler. Freezing solutions of equivariant evolution equations. SIAM Journal on Applied Dynamical Systems, 3(2):85–116, 2004.

[20] Marie Billaud-Friess and Anthony Nouy. Dynamical model reduction method for solving parameter-dependent dynamical systems. arXiv preprint arXiv:1604.05706, 2016.

[21] Peter Binev, Albert Cohen, Wolfgang Dahmen, Ronald DeVore, Guergana Petrova, and Przemyslaw Wojtaszczyk. Convergence rates for greedy algorithms in reduced basis methods. SIAM journal on mathematical analysis, 43(3):1457–1472, 2011.

[22] K Boukir, Y Maday, B Métivet, and E Razafindrakoto. A high-order characteristics/finite element method for the incompressible navier-stokes equations. International Journal for Numerical Methods in Fluids, 25(12):1421–1454, 1997.

[23] Annalisa Buffa, Yvon Maday, Anthony T Patera, Christophe Prud'homme, and Gabriel Turinici. A priori convergence of the greedy algorithm for the parametrized reduced basis method. ESAIM: Mathematical Modelling and Numerical Analysis, 46(03):595–603, 2012.

[24] Andreas Buhr, Christian Engwer, Mario Ohlberger, and Stephan Rave. Arbilomod, a simulation technique designed for arbitrary local modifications. SIAM Journal on Scientific Computing, 39(4):A1435–A1465, 2017.

[25] John Burkardt, Max Gunzburger, and Hyung-Chun Lee. Centroidal voronoi tessellation-based reduced-order modeling of complex systems. SIAM Journal on Scientific Computing, 28(2):459–484, 2006.

[26] N Cagniart, R Crisovan, Y Maday, and R Abgrall. Model order reduction for hyperbolic problems: a new framework. 2017.

[27] Nicolas Cagniart, Yvon Maday, and Benjamin Stamm. Model order reduction for problems with large convection effects. In Contributions to Partial Differential Equations and Applications, pages 131–150. Springer, 2019.

[28] Kevin Carlberg. Adaptive h-refinement for reduced-order models. International Journal for Numerical Methods in Engineering, 102(5):1192–1210, 2015.

[29] Kevin Carlberg, Matthew Barone, and Harbir Antil. Galerkin v. least-squares petrov–galerkin projection in nonlinear model reduction. Journal of Computational Physics, 330:693–734, 2017.

[30] Kevin Carlberg, Charbel Bou-Mosleh, and Charbel Farhat. Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compressive tensor approximations. International Journal for Numerical Methods in Engineering, 86(2):155–181, 2011.

[31] Kevin Carlberg, Charbel Farhat, Julien Cortial, and David Amsallem. The gnat method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. Journal of Computational Physics, 242:623–647, 2013.

[32] Carlos Castro, Francisco Palacios, and Enrique Zuazua. An alternating descent method for the optimal control of the inviscid burgers equation in the presence of shocks. Mathematical Models and Methods in Applied Sciences, 18(03):369–416, 2008.

[33] Tomás Chacón Rebollo, Enrique Delgado Ávila, and Macarena Gómez Mármol. Reduced basis method for the smagorinsky model. Recent developments in numerical methods for model reduction (2016),, 2016.

[34] TJ Chung. Computational fluid dynamics. Cambridge university press, 2010.

[35] Albert Cohen and Ronald DeVore. Kolmogorov widths under holomorphic mappings. IMA Journal of Numerical Analysis, 36(1):1–12, 2015.

[36] Phillip Colella, Milo R Dorr, Jeffrey AF Hittinger, and Daniel F Martin. High-order, finite-volume methods in mapped coordinates. Journal of Computational Physics, 230(8):2952–2976, 2011.

[37] M Couplet, P Sagaut, and C Basdevant. Intermodal energy transfers in a proper orthogonal decomposition–galerkin representation of a turbulent separated flow. Journal of Fluid Mechanics, 491:275–284, 2003.

[38] A Crespo, J Hernandez, and S Frandsen. Survey of modelling methods for wind turbine wakes and wind farms. Wind energy, 2(1):1–24, 1999.

[39] Wolfgang Dahmen, Christian Plesken, and Gerrit Welper. Double greedy algorithms: Reduced basis methods for transport dominated problemsâĹŮ. ESAIM: Mathematical Modelling and Numerical Analysis, 48(3):623–663, 2014.

[40] AE Deane, IG Kevrekidis, G Em Karniadakis, and SA Orszag. Low-dimensional models for complex geometry flows: Application to grooved channels and circular cylinders. Physics of Fluids A: Fluid Dynamics (1989-1993), 3(10):2337–2354, 1991.

[41] Herman Deconinck and Mario Ricchiuto. Residual distribution schemes: foundations and analysis. Encyclopedia of computational mechanics, 2007.

[42] Simone Deparis and A Emil Løvgren. Stabilized reduced basis approximation of incompressible three-dimensional navier-stokes equations in parametrized deformed domains. Journal of Scientific Computing, 50(1):198–212, 2012.

[43] Ronald DeVore, Guergana Petrova, and Przemyslaw Wojtaszczyk. Greedy algorithms for reduced bases in banach spaces. Constructive Approximation, 37(3):455–466, 2013.

[44] Hachmi Ben Dhia. Approches locales-globales méthode arlequin. In Proceedings du 7ème Colloque National de Calcul des Structures, volume 1, pages 21–32, 2005.

[45] Victorita Dolean, Pierre Jolivet, and Frédéric Nataf. An introduction to domain decomposition methods: algorithms, theory, and parallel implementation. SIAM, 2015.

[46] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: applications and algorithms. SIAM review, 41(4):637–676, 1999.

[47] Jens L Eftang and Anthony T Patera. A port-reduced static condensation reduced basis element method for large component-synthesized structures: approximation and a posteriori error estimation. Advanced Modeling and Simulation in Engineering Sciences, 1(1):3, 2014.

[48] Jens L Eftang, Anthony T Patera, and Einar M Rønquist. An" hp" certified reduced basis method for parametrized elliptic partial differential equations. SIAM Journal on Scientific Computing, 32(6):3170–3200, 2010.

[49] Marco Fahl. Trust-region methods for flow control based on reduced order modelling. PhD thesis, Universitätsbibliothek, 2001.

[50] Lambert Fick, Yvon Maday, Anthony T Patera, and Tommaso Taddei. A reduced basis technique for long-time unsteady turbulent flows. arXiv preprint arXiv:1710.03569, 2017.

[51] C Foias, MS Jolly, IG Kevrekidis, GR Sell, and ES Titi. On the computation of inertial manifolds. Physics Letters A, 131(7):433–436, 1988.

[52] Ciprian Foias, Michael S Jolly, Rostyslav Kravchenko, and Edriss S Titi. A determining form for the two-dimensional navier-stokes equations: The fourier modes case. Journal of Mathematical Physics, 53(11):115623, 2012.

[53] Ciprian Foias, Michael S Jolly, Rostyslav Kravchenko, and Edriss S Titi. A unified approach to determining forms for the 2d navier-stokes equations–the general interpolants case. Russian Mathematical Surveys, 69(2):359, 2014.

[54] Ciprian Foias, George R Sell, and Roger Temam. Inertial manifolds for nonlinear evolutionary equations. Journal of Differential Equations, 73(2):309–353, 1988.

[55] Sten Frandsen, Rebecca Barthelmie, Sara Pryor, Ole Rathmann, Søren Larsen, Jørgen Højstrup, and Morten Thøgersen. Analytical modelling of wind speed deficit in large offshore wind farms. Wind energy, 9(1-2):39–53, 2006.

[56] Sten Tronæs Frandsen, Rebecca Jane Barthelmie, Ole Rathmann, Hans Ejsing Jørgensen, Jake Badger, Kurt Schaldemose Hansen, Søren Ott, Pierre-Elouan Mikael Rethore, Søren Ejling Larsen, and LE Jensen. the shadow effect of large wind farms: measurements, data analysis and modelling: Risø-r-1615 (en). Technical report.

[57] Pascal-Jean Frey and Frédéric Alauzet. Anisotropic mesh adaptation for cfd computations. Computer methods in applied mechanics and engineering, 194(48):5068–5082, 2005.

[58] Patrick Gallinari, Yvon Maday, Maxime Sangnier, Olivier Schwander, and Tommaso Taddei. Reduced basis' acquisition by a learning process for rapid on-line approximation of solution to pdes': Laminar flow past a backstep. Archives of Computational Methods in Engineering, pages 1–11, 2017.

[59] M Ganesh, JS Hesthaven, and B Stamm. A reduced basis method for multiple electromagnetic scattering in three dimensions. Preprint submitted to Journal of Computational Physics, 2011.

[60] Jean-Frédéric Gerbeau and Damiano Lombardi. Approximated lax pairs for the reduced order integration of nonlinear evolution equations. Journal of Computational Physics, 265:246–269, 2014.

[61] Edwige Godlewski and Pierre-Arnaud Raviart. The linearized stability of solutions of nonlinear hyperbolic systems of conservation laws: A general numerical approach. Mathematics and Computers in Simulation, 50(1):77–95, 1999.

[62] Edwige Godlewski and Pierre-Arnaud Raviart. Numerical approximation of hyperbolic systems of conservation laws, volume 118. Springer Science & Business Media, 2013.

[63] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. Numerische mathematik, 14(5):403–420, 1970.

[64] William J Gordon and Charles A Hall. Transfinite element methods: blending-function interpolation over arbitrary curved element domains. Numerische Mathematik, 21(2):109–129, 1973.

[65] Martin A Grepl and Anthony T Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. ESAIM: Mathematical Modelling and Numerical Analysis, 39(01):157–181, 2005.

[66] Bernard Haasdonk. Convergence rates of the pod–greedy method. ESAIM: Mathematical Modelling and Numerical Analysis, 47(3):859–873, 2013.

[67] F. Hecht. New development in freefem++. J. Numer. Math., 20(3-4):251–265, 2012.

[68] Jan S Hesthaven, Gianluigi Rozza, Benjamin Stamm, et al. Certified reduced basis methods for parametrized partial differential equations. Springer, 2016.

[69] Christian Himpe, Tobias Leibner, and Stephan Rave. Hierarchical approximate proper orthogonal decomposition. SIAM Journal on Scientific Computing, 40(5):A3267–A3292, 2018.

[70] Philip Holmes, John L Lumley, and Gal Berkooz. Turbulence, coherent structures, dynamical systems and symmetry. Cambridge university press, 1998.

[71] Thomas JR Hughes, Luca Mazzei, and Kenneth E Jansen. Large eddy simulation and the variational multiscale method. Computing and Visualization in Science, 3(1-2):47–59, 2000.

[72] Ruben Ibañez, Domenico Borzacchiello, Jose Vicente Aguado, Emmanuelle Abisset-Chavanne, Elías Cueto, Pierre Ladevèze, and Francisco Chinesta. Data-driven non-linear elasticity: constitutive manifold construction and problem discretization. Computational Mechanics, 60(5):813–826, 2017.

[73] Traian Iliescu and Zhu Wang. Variational multiscale proper orthogonal decomposition: Navier-stokes equations. Numerical Methods for Partial Differential Equations, 30(2):641–663, 2014.

[74] Angelo Iollo and Damiano Lombardi. Advection modes by optimal mass transfer. Physical Review E, 89(2):022923, 2014.

[75] Christoph Jäggli, Laura Iapichino, and Gianluigi Rozza. An improvement on geometrical parameterizations by transfinite maps. Comptes Rendus Mathématique, 352(3):263–268, 2014.

[76] SoHyeon Jeong, Barbara Solenthaler, Marc Pollefeys, Markus Gross, et al. Data-driven fluid simulations using regression forests. ACM Transactions on Graphics (TOG), 34(6):199, 2015.

[77] Tosio Kato. Perturbation theory for linear operators, volume 132. Springer Science & Business Media, 2013.

[78] Alexander Kirillov. An introduction to Lie groups and Lie algebras, volume 113. Cambridge University Press, 2008.

[79] Robert H Kraichnan. Eddy viscosity in two and three dimensions. Journal of the Atmospheric Sciences, 33(8):1521–1536, 1976.

[80] Karl Kunisch and Stefan Volkwein. Proper orthogonal decomposition for optimality systems. ESAIM: Mathematical Modelling and Numerical Analysis, 42(1):1–23, 2008.

[81] Karl Kunisch and Stefan Volkwein. Optimal snapshot location for computing pod basis functions. ESAIM: Mathematical Modelling and Numerical Analysis, 44(3):509–529, 2010.

[82] Andrew Kusiak and Zhe Song. Design of wind farm layout for maximum wind energy capture. Renewable Energy, 35(3):685–694, 2010.

[83] Pierre Ladevèze, J-C Passieux, and David Néron. The latin multiscale computational method and the proper generalized decomposition. Computer Methods in Applied Mechanics and Engineering, 199(21):1287–1296, 2010.

[84] Toni Lassila, Andrea Manzoni, Alfio Quarteroni, and Gianluigi Rozza. Model order reduction in fluid dynamics: challenges and perspectives. In Reduced Order Methods for Modeling and Computational Reduction, pages 235–273. Springer, 2014.

[85] Claude Le Bris, Tony Lelievre, and Y Maday. Results and questions on a nonlinear approximation approach for solving high-dimensional partial differential equations. Constructive Approximation, 30(3):621–651, 2009.

[86] A Leonard. Energy cascade in large-eddy simulations of turbulent fluid flows. Advances in geophysics, 18:237–248, 1975.

[87] Pierre-Louis Lions. On the schwarz alternating method. iii: a variant for nonoverlapping subdomains. In Third international symposium on domain decomposition methods for partial differential equations, volume 6, pages 202–223. SIAM, Philadelphia, PA, 1990.

[88] A.E. Løvgren, Y. Maday, and E.M Rønquist. The reduced basis element method: Offline-online decomposition in the nonconforming, nonaffine case. In Spectral and High Order Methods for Partial Differential Equations, pages 247–254. Springer, 2011.

[89] Alf Emil Løvgren, Yvon Maday, and Einar M Rønquist. Global c1 maps on general domains. Mathematical Models and Methods in Applied Sciences, 19(5):803–832, 2009.

[90] Alf Emil Løvgren, Yvon Maday, and EinarM Rønquist. A reduced basis element method for the steady stokes problem. ESAIM: Mathematical Modelling and Numerical Analysis, 40(03):529–552, 2006.

[91] Leon B Lucy. A numerical approach to the testing of the fission hypothesis. The astronomical journal, 82:1013–1024, 1977.

[92] Yvon Maday, Andrea Manzoni, and Alfio Quarteroni. An online intrinsic stabilization strategy for the reduced basis approximation of parametrized advection-dominated problems. Comptes Rendus Mathematique, 354(12):1188–1194, 2016.

[93] Yvon Maday, Anthony T Patera, James D Penn, and Masayuki Yano. A parameterized-background data-weak approach to variational data assimilation: formulation, analysis, and application to acoustics. International Journal for Numerical Methods in Engineering, 102(5):933–965, 2015.

[94] Yvon Maday, Anthony T Patera, and Gabriel Turinici. A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations. Journal of Scientific Computing, 17(1-4):437–446, 2002.

[95] Yvon Maday and Einar M Ronquist. The reduced basis element method: application to a thermal fin problem. SIAM Journal on Scientific Computing, 26(1):240–258, 2004.

[96] Yvon Maday and Benjamin Stamm. Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces. SIAM Journal on Scientific Computing, 35(6):A2417–A2441, 2013.

[97] Luis A Martinez, Stefano Leonardi, Matthew J Churchfield, and Patrick J Moriarty. A comparison of actuator disk and actuator line wind turbine models and best practices for their use. AIAA Paper, 900, 2012.

[98] Immanuel Martini, Bernard Haasdonk, and Gianluigi Rozza. Certified reduced basis approximation for the coupling of viscous and inviscid parametrized flow models. Journal of Scientific Computing, 74(1):197–219, 2018.

[99] Jens M Melenk. On n-widths for elliptic problems. Journal of mathematical analysis and applications, 247(1):272–289, 2000.

[100] Jens Markus Melenk and Ivo Babuška. The partition of unity finite element method: basic theory and applications. Computer methods in applied mechanics and engineering, 139(1):289–314, 1996.

[101] Ngoc-Cuong Nguyen, Gianluigi Rozza, and Anthony T Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous burgers' equation. Calcolo, 46(3):157–185, 2009.

[102] Anthony Nouy. A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations. Computer Methods in Applied Mechanics and Engineering, 199(23):1603–1626, 2010.

[103] List of offshore wind farms. List of offshore wind farms— Wikipedia, the free encyclopedia, 2017. [Online; accessed 12-June-2017].

[104] Hae-Soo Oh, June G Kim, and Won-Tak Hong. The piecewise polynomial partition of unity functions for the generalized finite element methods. Computer Methods in Applied Mechanics and Engineering, 197(45):3702–3711, 2008.

[105] Mario Ohlberger and Stephan Rave. Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing. Comptes Rendus Mathematique, 351(23):901–906, 2013.

[106] Mario Ohlberger and Kathrin Smetana. Approximation of skewed interfaces with tensor-based model reduction procedures: application to the reduced basis hierarchical model reduction approach. Journal of Computational Physics, 321:1185–1205, 2016.

[107] Jan Östh, Bernd R Noack, Siniša Krajnović, Diogo Barros, and Jacques Borée. On the need for a nonlinear subscale turbulence term in pod models as exemplified for a high-reynolds-number flow over an ahmed body. Journal of Fluid Mechanics, 747:518–544, 2014.

[108] Elizabeth Qian, Martin Grepl, Karen Veroy, and Karen Willcox. A certified trust region reduced basis approach to pde-constrained optimization, 2016.

[109] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. Reduced basis methods for partial differential equations, vol. 92 of unitext, 2016.

[110] Alfio Quarteroni and Gianluigi Rozza. Numerical solution of parametrized navier–stokes equations by reduced basis methods. Numerical Methods for Partial Differential Equations, 23(4):923–948, 2007.

[111] Alfio Quarteroni, Gianluigi Rozza, and Andrea Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. Journal of Mathematics in Industry, 1(1):3, 2011.

[112] Muruhan Rathinam and Linda R Petzold. A new look at proper orthogonal decomposition. SIAM Journal on Numerical Analysis, 41(5):1893–1925, 2003.

[113] D Rempfer. On low-dimensional galerkin models for fluid flow. Theoretical and Computational Fluid Dynamics, 14(2):75–88, 2000.

[114] Michael Renardy and Robert C Rogers. An introduction to partial differential equations, volume 13. Springer Science & Business Media, 2006.

[115] Donsub Rim, Scott Moe, and Randall J. LeVeque. Transport reversal for model reduction of hyperbolic partial differential equations, January 2017.

[116] Clancy Rowley. Connections between koopman and dynamic mode decomposition. Oberwolfach workshop, 2016.

[117] Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. Spectral analysis of nonlinear flows. Journal of fluid mechanics, 641:115–127, 2009.

[118] CW Rowley. Model reduction for fluids, using balanced proper orthogonal decomposition. International Journal of Bifurcation and Chaos, 15(03):997–1013, 2005.

[119] David Ryckelynck. Hyper-reduction of mechanical models involving internal variables. International Journal for Numerical Methods in Engineering, 77(1):75–89, 2009.

[120] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. Journal of fluid mechanics, 656:5–28, 2010.

[121] Kaleem Siddiqi, Benjamin B Kimia, and Chi-Wang Shu. Geometric shock-capturing eno schemes for subpixel interpolation, computation, and curve evolution. In Computer Vision, 1995. Proceedings., International Symposium on Computer Vision - ISCV, pages 437–442. IEEE, 1995.

[122] Lawrence Sirovich. Turbulence and the dynamics of coherent structures. i-coherent structures. ii-symmetries and transformations. iii-dynamics and scaling. Quarterly of applied mathematics, 45:561–571, 1987.

[123] Kathrin Smetana and Mario Ohlberger. Hierarchical model reduction of nonlinear partial differential equations based on the adaptive empirical projection method and reduced basis techniques. ESAIM: Mathematical Modelling and Numerical Analysis, 51(2):641–677, 2017.

[124] Giovanni Stabile and Gianluigi Rozza. Finite volume pod-galerkin stabilised reduced order methods for the parametrised incompressible navier-stokes equations. arXiv preprint arXiv:1710.11580, 2017.

[125] Gilbert Strang and George J Fix. An analysis of the finite element method, volume 212. Prentice-hall Englewood Cliffs, NJ, 1973.

[126] T Taddei, S Perotto, and A Quarteroni. Reduced basis techniques for nonlinear conservation laws. ESAIM: Mathematical Modelling and Numerical Analysis, 49(3):787–814, 2015.

[127] Tommaso Taddei. An adaptive parametrized-background data-weak approach to variational data assimilation. ESAIM: Mathematical Modelling and Numerical Analysis.

[128] Eitan Tadmor. Convergence of spectral methods for nonlinear conservation laws. SIAM Journal on Numerical Analysis, 26(1):30–44, 1989.

[129] Roger Temam. Infinite-dimensional dynamical systems in mechanics and physics, volume 68. Springer Science & Business Media, 2012.

[130] Edriss S Titi. On approximate inertial manifolds to the navier-stokes equations. Journal of mathematical analysis and applications, 149(2):540–557, 1990.

[131] Timo Tonn. Reduced-Basis Method (RBM) for Non-Affine Elliptic Parametrized PDEs:(Motivated by Optimization in Hydromechanics). PhD thesis, Ulm, UniversitaÌĹt Ulm, Diss., 2012, 2012.

[132] Jonathan H Tu, Clarence W Rowley, Dirk M Luchtenburg, Steven L Brunton, and J Nathan Kutz. On dynamic mode decomposition: theory and applications. arXiv preprint arXiv:1312.0041, 2013.

[133] Sylvain Vallaghe and Anthony T Patera. The static condensation reduced basis element method for a mixed-mean conjugate heat exchanger model. SIAM Journal on Scientific Computing, 36(3):B294–B320, 2014.

[134] LJ Vermeer, Jens Nørkær Sørensen, and A Crespo. Wind turbine wake aerodynamics. Progress in aerospace sciences, 39(6):467–510, 2003.

[135] Karen Veroy, Christophe Prud'Homme, and Anthony T Patera. Reduced-basis approximation of the viscous burgers equation: rigorous a posteriori error bounds. Comptes Rendus Mathematique, 337(9):619–624, 2003.

[136] Stefan Volkwein. Proper orthogonal decomposition: Theory and reduced-order modelling. Lecture Notes, University of Konstanz, 4(4), 2013.

[137] Z Wang, D Xiao, F Fang, R Govindan, CC Pain, and Y Guo. Model identification of reduced order fluid dynamics systems using deep learning. International Journal for Numerical Methods in Fluids, 2017.

[138] Zhu Wang, Imran Akhtar, Jeff Borggaard, and Traian Iliescu. Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison. Computer Methods in Applied Mechanics and Engineering, 237:10–26, 2012.

[139] Zhu Wang, Brian McBee, and Traian Iliescu. Approximate partitioned method of snapshots for pod. Journal of Computational and Applied Mathematics, 307:374–384, 2016.

[140] D Wells, Z Wang, X Xie, and T Iliescu. An evolve-then-filter regularized reduced order model for convection-dominated flows. International Journal for Numerical Methods in Fluids, 2017.

[141] G. Welper. Interpolation of functions with parameter dependent jumps by transformed snapshots. SIAM Journal on Scientific Computing, 39(4):A1225âĂŞA1250, 2017.

[142] Masayuki Yano, Anthony T Patera, and Karsten Urban. A space-time hp-interpolation-based certified reduced basis method for burgers' equation. Mathematical Models and Methods in Applied Sciences, 24(09):1903–1935, 2014.