SISSA

Scuola
Internazionale
Superiore di
Studi Avanzati

Mathematics Area - PhD course in
Mathematical Analysis, Modelling, and Applications

Data-driven parameter and model order reduction for industrial optimisation problems with applications in naval engineering

Candidate:
Marco Tezzele

Advisor:
Prof. Gianluigi Rozza

Industrial advisors:
Ing. Matteo Sidari
Ing. Mauro Sicchiero

Academic Year 2020-2021

Dedicated to the memory of Augusto Tezzele.

1937 – 1994

Higher education is the natural enemy of any type of nationalism, it spreads easily beyond the borders and it cannot be stopped.

— Paolo Budinich (1916-2013), founder of the Trieste system

ACKNOWLEDGMENTS

It has been quite a journey, but above all unexpected. Never I would have imagined how intense and gratifying would be the doctorate path. I went through a broken elbow, a pandemic, a marriage, and many conferences, and I had the chance of meeting a lot of great members of our community.

This work and personal path would have not been possible without the support of my wife, Alice, who gave me the strength and confidence to overcome any difficulties, and for helping me balance life and work in an healthy way. She is my best groupie. I owe her a lot.

A special thanks goes to Gian who taught me how to be a good researcher, but above all he passed on his passion for this work to me.

Many people of the mathLab family were there when I needed help. A particular mention goes to Demo for the many adventures we spent together around the world, Stabile for his perspectives, Romor for his availability, Mola for the patience and the dinners, Noselli for the humour, and many others even if we just met at a coffee break. You were more than colleagues to me.

The collaboration with Fincantieri S.p.A. gave me the chance of working with Matteo Sidari and Mauro Sicchiero. I would like to thank them for their patience in introducing me in the cruise ships design and manufacturing world, and for supporting me during the entire collaboration.

Finally I would like to thank Virgi and Sofi for the relaxing time spent together, and in particular Petra as new and best mascot of our group!

Uno speciale ringraziamento va alla mia famiglia che tanto ha portato pazienza in questi anni, spero di renderli orgogliosi almeno tanto quanto lo sono io di loro. Infine ringrazio anche Luciana, che ormai fa parte della nostra famiglia allargata insieme a Mario, per i suoi manicaretti che mi hanno accompagnato per tutto il dottorato e per la calorosa accoglienza che ha sempre nei miei confronti.

ABSTRACT

In this work we study data-driven reduced order models with a specific focus on reduction in parameter space to fight the curse of dimensionality, especially for functions with low-intrinsic structure, in the context of digital twins. To this end we proposed two different methods to improve the accuracy of responce surfaces built using the Active Subspaces (AS): a kernel-based approach which maps the inputs onto an higher dimensional space before applying AS, and a local approach in which a clustering induced by the presence of a global active subspace is exploited to construct localized regressors. We also used AS within a multi-fidelity nonlinear autoregressive scheme to reduced the approximation error of high-dimensional scalar function using only high-fidelity data. This multi-fidelity approach has also been integrated within a non-intrusive Proper Oorthogonal Decomposition (POD) based framework in which every modal coefficient is reconstructed with a greater precision.

Moving to optimization algorithms we devised an extension of the classical genetic algorithm exploiting AS to accelerate the convergence, especially for high-dimensional optimization problems.

We applied different combinations of such methods in a diverse range of engineering problems such as structural optimization of cruise ships, shape optimization of a combatant hull and a NACA airfoil profile, and the prediction of hydroacoustic noises. A specific attention has been devoted to the naval engineering applications and many of the methodological advances in this work have been inspired by them.

This work has been conducted within the framework of the IRONTH project, an industrial Ph.D. grant financed by Fincantieri S.p.A..

CONTENTS

II OPTIMIZATION METHODS

III APPLICATIONS IN NAVAL ENGINEERING

IV ACADEMIC CFD APPLICATIONS

ACRONYMS

AS	Active Subspaces
ASGA	Active Subspaces Genetic Algorithm
CFD	Computational Fluid Dynamics
DG	Discontinuous Galerkin
DMD	Dynamic Mode Decomposition
DyAS	Dynamic Active Subspaces
FFD	Free Form Deformation
FFT	Fast Fourier Transform
FOM	Full Order Model
FWH	Ffowcs Williams and Hawkings
GA	Genetic Algorithm
GP	Gaussian Processes
GPR	Gaussian Process Regression
HAS	Hierarchical top-down clustering with Active Subspaces
HF	High-fidelity
KAS	Kernel-based Active Subspaces
LAS	Local Active Subspaces
LES	Large Eddy Simulation
MF	Multi-fidelity
MOR	Model Order Reduction
NAND	Nested Analysis and Design
NARGP	Nonlinear Autoregressive Multi-fidelity Gaussian Process Regression
NARGPAS	Nonlinear Autoregressive Multi-fidelity Gaussian Process Regression with Active Subspaces
PDE	Partial Differential Equation
POD	Proper Orthogonal Decomposition
POD-GPR	Proper Orthogonal Decomposition with Gaussian Process Regression
PODI	Proper Orthogonal Decomposition with Interpolation
RANS	Reynolds Averaged Navier–Stokes
RBF	Radial Basis Functions
RKHS	Reproducing Kernel Hilbert Spaces
ROM	Reduced Order Method

RRMSE	Relative Root Mean Square Error
SGS	Sub-Grid Scales
SIR	Slice Inverse Regression
SV	Singular Values
SVD	Singular Value Decomposition
VOF	Volume of Fluid

1

INTRODUCTION AND MOTIVATIONS

Nowadays the spread of digital twins is ubiquitous in engineering, but still many improvements have to be made. A static digital twin is no more capable to provide an edge in the modern era of data. The model has to be able to evolve dynamically and to integrate in real-time informations coming from sensors, external services, and simulations. This leads to a vast amount of input parameters for every quantity of interest. Thus to construct reliable response surfaces and reduced order models the number of samples needed grows exponentially: this is the concept of curse of dimensionality.

Is it possible to reduce the environmental impact of the modern cruise ship manufacturing from a structural point of view? This is the main question we posed ourselves at the start of this project with Fincantieri S.p.A.. The minimization of the amount of steel was a natural choice to both reduce the raw materials needed to build the entire ship and to make the hull more efficient thus reducing also the operational costs in terms of fuel consumption. Of course, we cannot perform such minimization without considering some stability constraints to be satisfied. In fact, we aim at a more efficient hull preserving the structural behaviour of the ship.

Data-driven Model Order Reduction (MOR) allows a fast and accurate prediction of not only scalar quantities but also fields of interest originating from parametrized systems. This is especially useful for many-query problems such as optimization loops, inverse problems, and uncertainty quantification. The need of surrogate models for a real-time evaluation of the quantities of interest for optimization tasks is only the first step, we need to integrate also reduction in parameter space, due to the increased complexity of the industrial artifacts and computational capabilities, which enables more and more complex simulations with a high number of degrees of freedom.

There is a huge effort to fight the curse of dimensionality originating from high-dimensional inputs design, and parameter space reduction plays a key role in this regard. Active Subspaces (AS) is one of the most spread techniques for linear input space dimensionality reduction. It is able to identify linear combinations of all the parameters along which the function of interest varies the most, on average. This results in a ridge approximation.

In this thesis we present two very different approaches to improve the accuracy of response surfaces built using AS, which can be seen as a step towards nonlinear extension of AS. The first one relies on a nonlinear mapping of the inputs onto an higher dimensional feature space before computing the active subspace, and is called Kernel-based Active Subspaces (KAS). The second approach is built upon a localization idea, which exploits a clustering of the input parameters. Local Active Subspaces (LAS) uses the presence of a global AS to induce a transversal splitting of the response surface design. So even if we have local linear approximations, they encode nonlinear infomations. Both methods were developed in collaboration with Francesco Romor which I tutored during his master thesis.

We exploited AS also to create a multi-fidelity nonlinear autoregressive scheme which works using only high-fidelity simulations. The low-fidelity is built via a regression over the active subspace, thus we do not need coarser meshes or simplified models to run.

We also developed numerical and modular pipelines for data-driven non-intrusive optimization for parametrized systems, in an industrial context, with a special focus on naval engineering. We considered both structural and geometric parameterizations. This advanced numerical framework involves parameter space reduction at multiple steps of the optimization procedure: approximation of the quantities of interest, enhancement of the reduced order models, and efficient optimization of high-dimensional functions exploiting the presence of an active subspace, which led to the creation of a new extension of the classical genetic algorithm, called Active Subspaces Genetic Algorithm (ASGA).

The numerical results present in this thesis have been obtained thanks to several scientific software packages created within SISSA mathLab in the last years. We developed and used PyDMD [71] for all the computation regarding Dynamic Mode Decomposition, EZyRB [70] for the non-intrusive Proper Orthogonal Decomposition (POD)-based Reduced Order Methods (ROMs) for parametrized systems, PyGeM [269] for the geometrical parametrization and morphing. PyDMD in particular won the first prize in the Junior Faculty Category of the DSWeb 2019 Contest: Tutorials on Dynamical Systems Software. Finally, this project also led to the release of a Python package for parameter space reduction, called ATHENA [222]. We thank all the contributors and main developers.

The collaboration with Fincantieri S.p.A. has been fundamental to understand the needs and bottlenecks of current state of the art in industrial optimization. This pushed us to advance the state of the art for parameter space reduction due to the complexity of modern cruise ships in terms of design parameters.

1.1 THESIS OUTLINE

The thesis is divided in four parts. The first one is devoted to the reduced order methods, the second to optimization, the third to naval engineering applications, and the fourth to more academic applications.

In figure 1.1 we show an illustration of the methods used in the present work, which comprises the first two parts of the thesis. In the illustration we also emphasize the relationship and the connections between parameter space dimensionality reduction and reduced order modeling for parametrized systems. In particular these parts are organized as follows:

- Part 1: Data-driven reduced order modeling

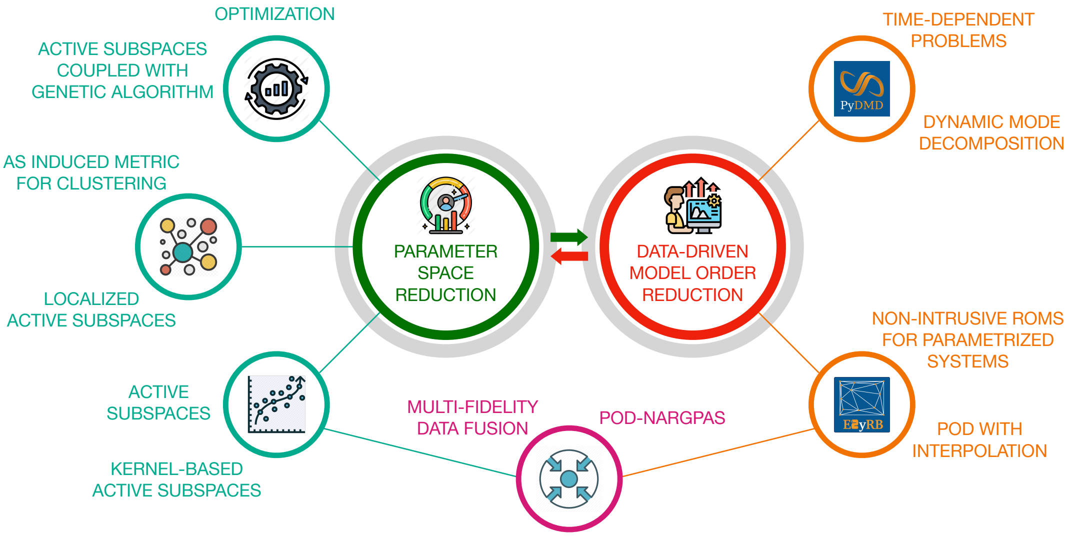

OPTIMIZATION
ACTIVE SUBSPACES COUPLED WITH GENETIC ALGORITHM
AS INDUCED METRIC FOR CLUSTERING
LOCALIZED ACTIVE SUBSPACES
ACTIVE SUBSPACES
KERNEL-BASED ACTIVE SUBSPACES
PARAMETER SPACE REDUCTION
DATA-DRIVEN MODEL ORDER REDUCTION
MULTI-FIDELITY DATA FUSION
POD-NARGPAS
TIME-DEPENDENT PROBLEMS
PyDMD
DYNAMIC MODE DECOMPOSITION
NON-INTRUSIVE ROMS FOR PARAMETRIZED SYSTEMS
EZyRB
POD WITH INTERPOLATION

Figure 1.1: Illustrative sketch of the methods used in this thesis for parameter space and model order reduction together with their relationship.

- In Chapter 2 we review the AS technique for linear parameter space reduction and we propose a nonlinear extension, called KAS, which exploits an intermediate mapping on a higher-dimensional feature space. The new method is tested on both scalar and vector-valued functions of interest and compared with classical AS.
- In Chapter 3 we present another approach to go beyond the linear limitations of AS, which we called Local Active Subspaces. This new technique uses a clustering of the input parameters and a supervised distance metric induced by the global AS. We prove the benefits of such localized approach over classical benchmarks.
- In Chapter 4 we introduce a multi-fidelity data fusion method built using the regression over the Active Subspaces as low-fidelity model. This results in a nonlinear autoregressive scheme with multi-fidelity Gaussian Process Regression (GPR) which does not need a simpler model or a corser mesh to create the low-fidelity model. The accuracy of the method is demonstrated for the approximation of high-dimensional scalar functions with low-intrinsic dimensionality.
- In Chapter 5 we present a general non-intrusive data-driven reduced order modeling framework based on POD for parametric problems, and we review Dynamic Mode Decomposition (DMD) for time-dependent equation-free MOR. Here we also propose a coupling with the multi-fidelity approach introduced in chapter 4. This is a cornerstone to enable a full coupling between reduction in input and output spaces.

- Part 2: Optimization methods
  - In Chapter 6 we extend the classical Genetic Algorithm (GA) exploiting Active Subspaces to accelerate the convergence of the method for the optimization of high-dimensional functions. We prove effectiveness of the

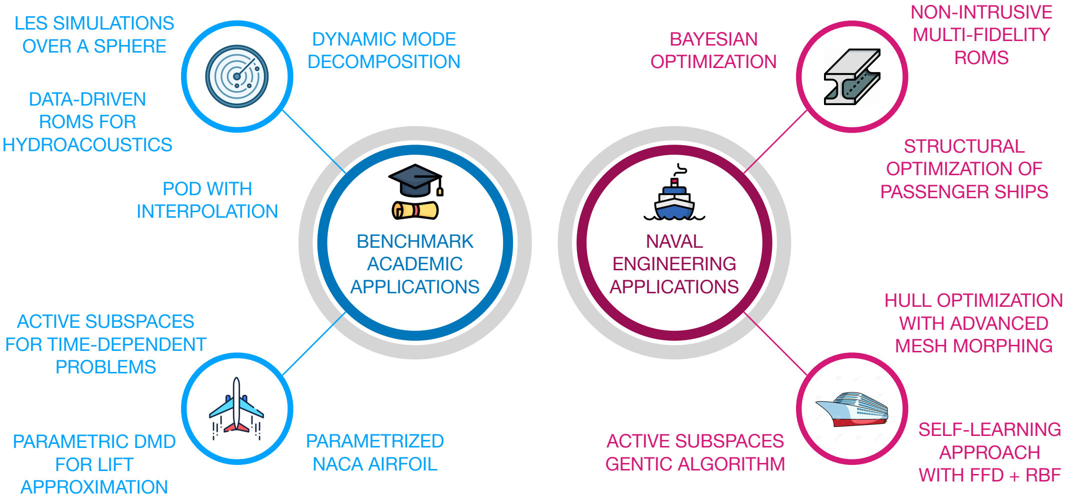

LES SIMULATIONS OVER A SPHERE
DYNAMIC MODE DECOMPOSITION
DATA-DRIVEN ROMS FOR HYDROACOUSTICS
POD WITH INTERPOLATION
BENCHMARK ACADEMIC APPLICATIONS
ACTIVE SUBSPACES FOR TIME-DEPENDENT PROBLEMS
PARAMETRIC DMD FOR LIFT APPROXIMATION
PARAMETRIZED NACA AIRFOIL
BAYESIAN OPTIMIZATION
NON-INTRUSIVE MULTI-FIDELITY ROMS
STRUCTURAL OPTIMIZATION OF PASSENGER SHIPS
NAVAL ENGINEERING APPLICATIONS
HULL OPTIMIZATION WITH ADVANCED MESH MORPHING
ACTIVE SUBSPACES GENTIC ALGORITHM
SELF-LEARNING APPROACH WITH FFD + RBF

Figure 1.2: Illustrative sketch of the part of the thesis regarding the academic and the naval engineering applications.

ASGA over different benchmark test functions and for the optimization of the lift-to-drag coefficient a parametrized NACA airfoil.

The last two parts are devoted to the naval engineering applications and to the more academic ones, respectively. In figure 1.2 we sketch the subdivision of part 3 and 4, highlighting the methods been used. In particular the chapters are organized as follows:

- Part 3: Applications in naval engineering
  - In Chapter 7 we present the main application which has led to the majority of the methodological development: the structural optimization of passenger ship hulls. Among the different ROMs we exploit the coupling between Proper Orthogonal Decomposition with Interpolation (PODI) and Nonlinear Autoregressive Multi-fidelity Gaussian Process Regression (NARGP) with AS, introduced in chapter 4. We apply the optimization framework to a Fincantieri's hull with two different parameterizations showing how the pipeline is able to efficiently deal with high-dimensional parametric problems.
  - In Chapter 8 we present a hull shape design optimization pipeline which exploits non-intrusive model order reduction, ASGA as optimizer, and an innovative self-learning mesh morphing. Geometrical deformations are propagated from surface modifications to volumetric meshes. We apply the proposed framework to a benchmark hull.
- Part 4: Academic CFD applications
  - In Chapter 9 we apply data-driven reduced order methods such as PODI and DMD to reconstruct and predict Large Eddy Simulation (LES) simulations of flow past a sphere for for hydroacoustic analysis. We show the computational savings for the evaluation of both linear and nonlinear

terms of hydroacoustic noise. This academic case unveils the potential application of such methodologies for a fast and accurate analysis of noise creation of marine propellers thus enabling more environmental friendly shape design.

- In Chapter 10 we introduce a first preliminary extension of DMD to handle parametric reduction of scalar output of interests evolving in time. This represents a first step towards parametric DMD for Computational Fluid Dynamics (CFD) predictions. We also employ parameter space reduction for time-dependent problems to discard unnecessary parameters and improve the prediction accuracy of the lift coefficient.

Part I

DATA-DRIVEN REDUCED ORDER MODELING

2

A KERNEL APPROACH TO PARAMETER SPACE REDUCTION

In this chapter, after reviewing Active Subspaces for parameter space linear reduction, we propose a new method to perform a nonlinear reduction in parameter spaces called Kernel-based Active Subspaces (KAS). By using a kernel approach it is possible to find active subspaces in high-dimensional feature spaces. A mathematical foundation of the method is presented, with several applications to benchmark model functions, both scalar and vector-valued. We also apply the kernel-based active subspaces extension to a Computational Fluid Dynamics (CFD) parametric problem using the Discontinuous Galerkin method. A full comparison with respect to the linear active subspaces technique is provided for all the applications, proving the better performances of the proposed method. Moreover we show how the new kernel method overcomes the drawbacks of the active subspaces application for radial symmetric model functions. All the results presented in this chapter appeared in [221].

2.1 LITERATURE REVIEW

Nowadays, in many industrial settings the simulation of complex systems requires a huge amount of computational power. Problems involving high-fidelity simulations are usually large-scale, moreover the amount of solutions required increases with the number of parameters. In this context we mention multidisciplinary analysis and optimization, inverse problems, optimal control, and uncertainty quantification; they all suffer from the curse of dimensionality, that is the complexity of the algorithms grows exponentially with the dimension of the input parameter space. Data-driven Reduced Order Methods (ROMs) [37, 227, 230] have been developed to deal with such complex problems but the limit for high dimensional parameter spaces remains.

One approach to alleviate the curse of dimensionality is to identify and exploit some notion of low-dimensional structure of the model function. A possible linear input coordinate transformation technique is the Sliced Inverse Regression (SIR) [161] approach and its extensions [59, 162, 289]. Sharing some characteristics with SIR, there is the Active Subspaces (AS) property [54, 55, 228, 291] which, in the last years, has emerged as a powerful linear data-driven technique to construct ridge approximations using gradients of the model function. AS has been successfully applied to quantify uncertainty in the numerical simulation of the HyShot II scramjet [58], and for sensitivity analysis of an integrated hydrologic model [129]. Reduction in parameter space has been coupled with model order reduction techniques [112, 212, 226] to enable more complex numerical studies without increasing the computational load. We mention the use of AS in cardiovascular applications with Proper Orthogonal Decomposition (POD)-Galerkin [266], in nonlinear structural analysis [106], in nautical and naval engineering [267, 268, 270, 272], coupled with Proper Orthogonal Decomposition with Interpolation for structural and CFD analysis [72], and with Dynamic Mode Decomposition in [271]. Advances in efficient global design optimization with surrogate modeling are presented in [168, 169] and applied to the shape design of the $N+2$ Supersonic Passenger Jet. Applications to enhance optimization methods have been developed in [73, 99, 275].

Possible extensions and variants of the active subspaces property are the Modified Active Subspace method [160], the Active Manifold method [35] which reduces the problem to the analysis of a 1D manifold by traversing level sets of the model function at the expense of high online costs, the shared Active Subspace method [131], the active subspaces property for multivariate functions [291], and more recently an extension of AS to dynamical systems [3]. Another method is the Nonlinear Level set Learning [294] which exploits RevNets to reduce the input parameter space with a nonlinear transformation.

The search for low dimensional structures is also investigated in machine learning with manifold learning algorithms. In this context the AS methodology can be seen as a supervised dimension reduction technique along with Kernel Principal Component Analysis [257] and Supervised Kernel Principal Component Analysis [15]. Other methods in the context of kernel-based ROMs are [110, 140, 180]. In [194] a non-linear extension of the active subspaces property based on Random Fourier

Features [163, 214] is introduced and compared with machine learning manifold learning algorithms for the construction of Gaussian Process Regression (GPR) [287].

From the ideas introduced in [194], we developed a new active subspaces extension called KASs. Our contribution with this work presents a solid mathematical foundation of the method, provides a generalization to vector valued model functions, algorithms for every component of the method, and the application to several test problems of increasing complexity. The KAS method is finally applied to a computational fluid dynamic problem and compared with the standard AS technique. We study the evolution of fluid flow past a NACA 0012 airfoil in a duct composed by an initialization channel and a chamber. The motion is modelled with the unsteady incompressible Navier-Stokes equations, and discretized with the Discontinuous Galerkin (DG) method [113]. Physical and geometrical parameters are introduced and sensitivity analysis of the lift and drag coefficients with respect to these parameters is provided.

The chapter is divided as follows: in section 2.2 we briefly present the active subspaces property of a model function with a focus on the construction of Gaussian process response surfaces. Section 2.3 illustrates the novel method called kernel-based active subspaces for both scalar and vector-valued model functions. Several tests to compare AS and KAS are provided in section 2.4 where we start from scalar functions with radial symmetry of increasing dimension, we analyse an epidemiology model and a vector-valued output generated from a stochastic elliptic Partial Differential Equation (PDE). A parametric CFD test case for the study of the flow past a NACA airfoil using the Discontinuous Galerkin method is presented in section 2.5. Finally we outline some perspectives and future studies in section 2.6.

2.2 ACTIVE SUBSPACES FOR PARAMETER SPACE REDUCTION

Active Subspaces approach proposed by Trent Russi [228] and developed by Paul Constantine [54] is a technique for dimension reduction in parameter space. In brief AS are defined as the leading eigenspaces of the second moment matrix of the model function's gradient (for scalar model functions) and constitutes a global sensitivity index more informative than coordinate-aligned derivative-based ones [291]. In the context of ridge regression, the choice of the active subspace corresponds to the minimizer of an upper bound of the mean square error obtained through Poincaré-type inequalities [291]. After performing dimension reduction in the parameter space through AS, the method can be applied to reduce the computational costs of different parameter studies such as inverse problems, optimization tasks and numerical integration [54]. In this work we are going to focus on the construction of response surfaces with Gaussian process regression.

Assumption 1 (Hypothesis on input and output spaces). *The quantities related to the input space are:*

- $m \in \mathbb{N}$ *the dimension of the input space,*
- $(\Omega, \mathcal{F}, P)$ *the probability space,*
- $\mathbf{X} : (\Omega, \mathcal{F}, P) \to \mathbb{R}^m$*, the absolutely continuous random vector representing the parameters,*

- *$\rho : \mathbb{R}^m \to \mathbb{R}$, the probability density of $\mathbf{X}$ with support $\mathcal{X} \subset \mathbb{R}^m$.*

The quantities related to the output are:

- *$d \in \mathbb{N}$ the dimension of the output space,*
- *$V = (\mathbb{R}^d, R_V)$ the Euclidean space with metric $R_V \in \mathcal{M}(d \times d)$ and norm*
$$\|\mathbf{x}\|_{R_V}^2 = \mathbf{x}^T R_V \mathbf{x},$$
- *$f : \mathcal{X} \subset \mathbb{R}^m \to V$, the quantity of interest.*

Let $\mathcal{B}(\mathbb{R}^m)$ be the Borel σ-algebra of $\mathbb{R}^m$. We will consider the Hilbert space $L^2(\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m), \rho; V)$ of the measurable functions $f : (\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m), \rho) \to (\mathbb{R}^d, R_V)$ such that

$$\|f\|_{L^2}^2 := \int_{\mathcal{X}} \|f(\mathbf{x})\|_{R_V}^2 \, d\rho(\mathbf{x}) \leq \infty;$$

and we will consider the Sobolev space $H^1(\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m), \rho; V)$ of measurable functions $f : (\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m), \rho) \to (\mathbb{R}^d, R_V)$ such that

$$\|f\|_{H^1}^2 := \|f\|_{L^2}^2 + \|\nabla f\|_{L^2}^2 = \|f\|_{L^2}^2 + |f|_{H^1}^2 \leq \infty \tag{2.1}$$

where ∇f is the weak derivative of f. We briefly recall how dimension reduction in parameter space is achieved in the construction of response surfaces. The first step involves the approximation of the model function with ridge approximation. We will follow [197, 291] for a review of the method.

The ridge approximation problem can be stated in the following way:

Problem 1 (Ridge approximation). *Let $\mathcal{B}(\mathbb{R}^m)$ be the Borel σ-algebra of $\mathbb{R}^m$. Given $r \in \mathbb{N}$, $r \ll d$ and a tolerance $\epsilon \geq 0$, find the profile $h : (\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m), \rho) \to V$ and the r-rank projection $P_r : \mathbb{R}^m \to \mathbb{R}^m$ such that*

$$\mathbb{E}_\rho[\|f(\mathbf{X}) - h(P_r\mathbf{X})\|_V^2] \leq \epsilon^2. \tag{2.2}$$

In particular we are interested in the minimization problem

$$\underset{P_r \in \mathcal{M}(m \times m)}{\operatorname{argmin}} \quad \mathbb{E}_\rho\left[\|f(\mathbf{X}) - \tilde{h}(P_r\mathbf{X})\|_V^2\right], \tag{2.3}$$

where $\tilde{h} \circ P_r = \mathbb{E}_\rho[f|\sigma(P_r)]$ is the conditional expectation of f under the distribution ρ given the σ-algebra $\sigma(P_r)$. The linear space $\ker(P_r) \subset \mathbb{R}^m$ is the reduced parameter space. The existence of $\tilde{h}$ is guaranteed by the Doob-Dynkin lemma [26]. The function $\tilde{h}$ is proven to be the optimal profile for each fixed P_r, as a consequence of the definition of the conditional expectation of a random variable with respect to a σ-algebra.

Dimension reduction is effective if the previous inequality in equation (2.2) is satisfied for a specific tolerance. The choice of r is certainly of central importance: the dimension of reduced parameter space can be chosen a priori for a specific parameter study (for example r-dimensional regression), it can be chosen in order to satisfy the inequality in equation (2.2) or it is determined to guarantee a good accuracy of the numerical method used to evaluate it [Corollary 3.10, [57]].

If $\epsilon^2 \ll \text{Var}(f(\mathbf{X})) = \mathbb{E}_\rho[\|f(\mathbf{X}) - \mathbb{E}_\rho[f(\mathbf{X}])\|_V^2]$ then $\mathbf{X}_r = P_r\mathbf{X}$ is an explanatory variable for $f(\mathbf{X})$. Dividing the left term of the inequality equation (2.2) with $\mathbb{E}_\rho[\|f(\mathbf{X}) - \mathbb{E}_\rho[f(\mathbf{X}])\|_V^2]$ we obtain the Relative Root Mean Square Error (RRMSE) and since it is a normalized quantity, we will use it to make comparisons between different models

$$\text{RRMSE} = \sqrt{\frac{\mathbb{E}_\rho[\|f(\mathbf{X}) - g(P_r\mathbf{X})\|_V^2]}{\mathbb{E}_\rho[\|f(\mathbf{X}) - \mathbb{E}_\rho[f(\mathbf{X})]\|_V^2]}}. \tag{2.4}$$

Remark 1 (P_r is not unique)**.** *We could also have used the r-rank projection $P_r : \mathbb{R}^m \to \mathbb{R}^r$ instead. It can be shown that if $\tilde{h}$ is the optimal profile, then P_r is not uniquely defined and can be chosen arbitrarily from the set $\{Q_r : \mathbb{R}^m \to \mathbb{R}^m | \ker Q_r = \ker P_r\}$, see [Proposition 2.2, [291]].*

The following lemma is the key ingredient in the proof of the existence of an active subspace. It is inherently linked to probability Poincaré inequalities of the kind

$$\int_{\mathcal{X}} \|h(\mathbf{x})\|_{L^2}^2 \, d\rho \leq C_P(\mathcal{X}, \rho) \int_{\mathcal{X}} \|\nabla h(\mathbf{x})\|_{L^2}^2 \, d\rho, \tag{2.5}$$

for zero-mean functions in the Sobolev space $h \in H^1(\mathcal{X})$, where $C_P(\mathcal{X}, \rho)$ is the Poincaré constant dependent on the domain $\mathcal{X}$ and on the probability density functions (p.d.f.), ρ. We need to make the following assumption to prove the next lemma and the next theorem.

Assumption 2. *The probability density function $\rho : \mathcal{X} \to \mathbb{R}^m$ belongs to one of the following classes:*

1. *$\mathcal{X}$ is convex and bounded, $\exists \delta, D > 0 : 0 < \delta \leq \|\rho(\mathbf{x})\|_{L^\infty} \leq D < \infty \, \forall \mathbf{x} \in \mathcal{X}$,*
2. *$\rho(\mathbf{x}) \sim \exp(-V(\mathbf{x}))$ where $V : \mathbb{R}^m \to (-\infty, \infty]$, $V \in \mathcal{C}^2$ is α-uniformly convex,*
$$\mathbf{u}^T \mathit{Hess}(V(\mathbf{x}))\mathbf{u} \geq \alpha\|\mathbf{u}\|_2^2, \quad \forall \mathbf{x}, \mathbf{u} \in \mathbb{R}^m \tag{2.6}$$
where $\mathit{Hess}(V(\mathbf{x}))$ is the Hessian of $V(\mathbf{x})$.
3. *$\rho(\mathbf{x}) \sim \exp(-V(\mathbf{x}))$ where V is a convex function. In this case we require also f Lipschitz continuous.*

In particular the uniform distribution belongs to the first class, the multivariate Gaussian distribution $\mathcal{N}(m, \Sigma)$ to the second with $\alpha = 1/(\sigma_{max}(\Sigma))$ and the exponential and Laplace distributions to the third. A complete analysis of the various cases is done in [197].

Lemma 1. *Let $(\Omega, \mathcal{F}, P)$ be a probability space, $\mathbf{X} : (\Omega, \mathcal{F}, P) \to \mathbb{R}^m$ an absolutely continuous random vector with probability density function ρ belonging to one of the classes from the assumption 2. Then the following inequality is satisfied*

$$\mathbb{E}_\rho\left[(h - \mathbb{E}_\rho[h|\sigma(P_r)])^2 \, |\sigma(P_r)\right] \leq C_P(P_r, \rho)\, \mathbb{E}_\rho\left[\|(I - P_r^T)\nabla h\|_2^2 | \sigma(P_r)\right] \tag{2.7}$$

for all scalar functions $h \in H^1(\mathcal{X})$ and for all r-rank orthogonal projectors, P_r, where $C_P(P_r, \rho)$ is the Poincaré constant depending on P_r and on the p.d.f. ρ.

A summary of the values of the Poincaré constant in relationship with the choice of the probability density function ρ is reported in [197].

Remark 2. *In the next theorem the projection P_r will depend on the output function f, so also the Poincaré constant $C_P(P_r, \rho)$ will depend in fact on f.*

We introduce the following notation for the matrix that substitutes the covariance matrix of the gradient ∇f in the case of the application of AS to scalar model functions [57]

$$H = \int_{\mathcal{X}} (D_{\mathbf{x}} f(\mathbf{x}))^T R_V(\rho) (D_{\mathbf{x}} f(\mathbf{x}))\, d\rho(\mathbf{x}). \tag{2.8}$$

where $D_{\mathbf{x}} f(\mathbf{x}) \in \mathcal{M}(d \times m)$ is the Jacobian matrix of f. The matrix $R_V(\rho)$ depends on the class which ρ belongs to.

Theorem 1. *(Existence of an active subspace) Under the previous hypothesis in assumption 1, let $f \in H^1(\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m), \rho\,;\, V)$ and let the p.d.f. ρ satisfy lemma 1 and assumption 2. In particular if ρ belongs to the third class of assumption 2, we also require f Lipschitz continuous. Then the solution $\tilde{P}_r$ of the ridge approximation problem 1 is the orthogonal projector to the eigenspace of the first r-eigenvalues of the eigenvalue problem for H ordered by magnitude*

$$Hv_i = \lambda_i v_i \qquad \forall i \in \{1, \dots, m\}, \qquad \tilde{P}_r = \sum_{j=1}^{r} v_j \otimes v_j,$$

with $r \in \mathbb{N}$ chosen such that

$$\mathbb{E}_\rho \left[\|f(\mathbf{X}) - h(\tilde{P}_r \mathbf{X})\|_V^2 \right] \leq C(C_P, \tau) \left(\sum_{i=r+1}^{m} \lambda_i \right)^{\frac{1}{1+\tau}} \leq \epsilon^2. \tag{2.9}$$

with $C(C_P, \tau)$ a constant depending on $\tau > 0$ related to the choice of ρ and on the Poincaré constant from lemma 1.

Proof of theorem 1: existence of an active subspace. This theorem summarizes the results from Proposition 2.5 and Proposition 2.6 of [291], and from Lemma 3.1, Lemma 4.2, Lemma 4.3, Lemma 4.4 and Theorem 4.5 of [197]. The proof is remodelled from [197, 291], and it is developed in five steps:

1. Since $R_V \in \mathcal{M}(d, d)$ is symmetric positive definite there exists a basis of eigenvectors $(\mathbf{w}_i)_{i \in \{1,\dots,d\}}$ and a corresponding set of positive eigenvalues $(\beta_i)_{i \in \{1,\dots,d\}}$ such that

$$R_V = \sum_{i=1}^{d} \beta_i\, \mathbf{w}_i \otimes \mathbf{w}_i. \tag{2.10}$$

2. Let us define the ridge approximation error as

$$\mathbf{e} = \|\mathbf{f} - \mathbf{h} \circ P_r\|_{L^2(\mathbb{R}^m, \mathcal{B}(\mathbb{R}^m), \rho; V)} = \mathbb{E}_\rho \left[\|(\mathbf{f}(\mathbf{X}) - \mathbf{h}(P_r(\mathbf{X}))\|_V^2 \right]. \tag{2.11}$$

Then we can decompose the error analysis for each component employing the spectral decomposition (2.10)

$$\begin{aligned}\mathbb{E}_{\rho}\left[\|\mathbf{e}(\mathbf{X})\|_V^2\right] &= \mathbb{E}_{\rho}\left[\operatorname{tr}((R_V\mathbf{e}(\mathbf{X}))\otimes\mathbf{e}(\mathbf{X}))\right] = \\ &= \sum_{i=1}^{d}\beta_i\,\mathbb{E}_{\rho}\left[\operatorname{tr}(((\mathbf{w}_i\otimes\mathbf{w}_i)\mathbf{e}(\mathbf{X}))\otimes\mathbf{e}(\mathbf{X}))\right] = \\ &= \sum_{i=1}^{d}\beta_i\,\mathbb{E}_{\rho}\left[(\mathbf{w}_i\cdot\mathbf{e}(\mathbf{X}))\operatorname{tr}(\mathbf{w}_i\otimes\mathbf{e}(\mathbf{X}))\right] = \\ &= \sum_{i=1}^{d}\beta_i\,\mathbb{E}_{\rho}\left[(\mathbf{w}_i\cdot\mathbf{e}(\mathbf{X}))^2\right], \end{aligned} \tag{2.12}$$

so we can define $e_i(\mathbf{X}) = \mathbf{w}_i\cdot\mathbf{e}(\mathbf{X}) = f_i(\mathbf{X}) - h_i(P_r(\mathbf{X}))$, $\forall i\in\{1,\dots,d\}$ and treat each component separately.

3. The next step involves the application of lemma 1 to the scalar functions $f_i(\mathbf{X}) - h_i(P_r(\mathbf{X}))$, $\forall i\in\{1,\dots,d\}$

$$\begin{aligned}&\mathbb{E}_{\rho}\left[(f_i(\mathbf{X}) - h_i(P_r(\mathbf{X})))^2\right] = \\ &= \mathbb{E}_{\rho}\left[\mathbb{E}_{\rho}\left[(f_i(\mathbf{X}) - h_i(P_r(\mathbf{X})))^2\,|\sigma(P_r)\right]\right] = \\ &= \mathbb{E}_{\rho}\left[C_p(\rho,P_r(\mathbf{X}))\mathbb{E}_{\rho}\left[\|(I-P_r^T)\nabla f_i(\mathbf{X})\|_2^2|\sigma(P_r)\right]\right] = \\ &= \mathbb{E}_{\rho}\left[C_p(\rho,P_r(\mathbf{X}))\right]^{\frac{1}{p}}\,\mathbb{E}_{\rho}\left[\|(I-P_r^T)\nabla f_i(\mathbf{X})\|_2^2\right]^{\frac{1}{q}}, \end{aligned} \tag{2.13}$$

where we used the Hölder inequality with indexes $(p,q) = (\infty,1)$ when ρ belongs to the first and second classes of assumption 2, and $(p,q) = (\frac{\tau+1}{\tau}, 1+\tau)$ when ρ belongs to the third class.

Then we can bound $\mathbb{E}_{\rho}\left[(C_p, P_r(\mathbf{X}))\right]^{\frac{1}{p}}$ with a constant $C(C_p(\rho,P_r(\mathbf{X})))$ which depends on the class of ρ (see Lemma 3.1, Lemma 4.2, Lemma 4.3, Lemma 4.4 and Theorem 4.5 of [197]) as follows

$$\begin{aligned}&\mathbb{E}_{\rho}\left[C_p(\rho,P_r(\mathbf{X}))\right]^{\frac{1}{p}}\,\mathbb{E}_{\rho}\left[\|(I-P_r^T)\nabla f_i(\mathbf{X})\|_2^2\right]^{\frac{1}{q}} \leq \\ &\leq C(C_p(\rho,P_r(\mathbf{X})))\operatorname{tr}(\mathbb{E}_{\rho}\left[(I-P_r^T)\nabla f_i(\mathbf{X})(\nabla f_i(\mathbf{X}))^T(I-P_r)\right]^{\frac{1}{q}}) = \\ &= C(C_p(\rho,P_r(\mathbf{X})))\operatorname{tr}((I-P_r^T)\mathbb{E}_{\rho}\left[\nabla f_i(\mathbf{X})(\nabla f_i(\mathbf{X}))^T\right](I-P_r))^{\frac{1}{q}}. \end{aligned} \tag{2.14}$$

4. The spectral decomposition in equation (2.10) is employed again and the covariance matrix H is introduced in the last equation below. We substitute $C(C_p(\rho,P_r(\mathbf{X})))$ with q for seek of compactness:

$$\begin{aligned}
&\mathbb{E}_\rho\left[\|\mathbf{e}(\mathbf{X})\|_V^2\right] \leq \\
&\leq \sum_{i=1}^{d} \beta_i\, q\, \mathrm{tr}((I-P_r^T)\mathbb{E}_\rho\left[((\nabla\mathbf{f}(\mathbf{X}))^T\mathbf{w}_i)\otimes((\nabla\mathbf{f}(\mathbf{X}))^T\mathbf{w}_i)\right](I-P_r))^{\frac{1}{q}} = \\
&= q\,\mathrm{tr}((I-P_r^T)\mathbb{E}_\rho\left[(\nabla\mathbf{f}(\mathbf{X}))^T\left(\sum_{i=1}^{d}\beta_i^q\,\mathbf{w}_i\otimes\mathbf{w}_i\right)\nabla\mathbf{f}(\mathbf{X})\right](I-P_r))^{\frac{1}{q}} = \\
&= q\,\mathrm{tr}((I-P_r^T)\mathbb{E}_\rho\left[(\nabla\mathbf{f}(\mathbf{X}))^T R_V(\rho)\nabla\mathbf{f}(\mathbf{X})\right](I-P_r))^{\frac{1}{q}} = \\
&= q\,\mathrm{tr}((I-P_r^T)H(I-P_r))^{\frac{1}{q}}, \qquad (2.15)
\end{aligned}$$

where $R_V(\rho)$ is the original metric matrix if ρ belongs to the first or second class of assumption 2 and is equal to

$$\sum_{i=1}^{d}\beta_i^{1+\tau}\,\mathbf{w}_i\otimes\mathbf{w}_i, \qquad (2.16)$$

if ρ belongs to the third class.

5. Finally the bound in the statement of the theorem is recovered solving the following minimization problem with classical model reduction arguments employing Singular Value Decomposition (SVD)

$$\tilde{P}_r = \underset{P_r\in\mathcal{O}(m,m)}{\mathrm{argmin}}\ \mathrm{tr}((I-P_r^T)H(I-P_r)). \qquad (2.17)$$

□

The eigenspace $\mathrm{span}\{v_1,\dots,v_r\}\subset\mathbb{R}^m$ is the active subspace and the remaining eigenvectors generate the inactive subspace $\mathrm{span}\{v_{r+1},\dots,v_m\}\subset\mathbb{R}^m$. The condition $f\in L^2(\mathbb{R}^m,\mathcal{B}(\mathbb{R}^m),\rho;V)$ is necessary for f to satisfy the error bound in equation (2.2). For the explicit procedure to compute the active subspace given its dimension r see algorithm 1. To have an idea of how the projection along the AS works on a simple example we can have a look at figure 2.1 where we show 3 projections along different subspaces.

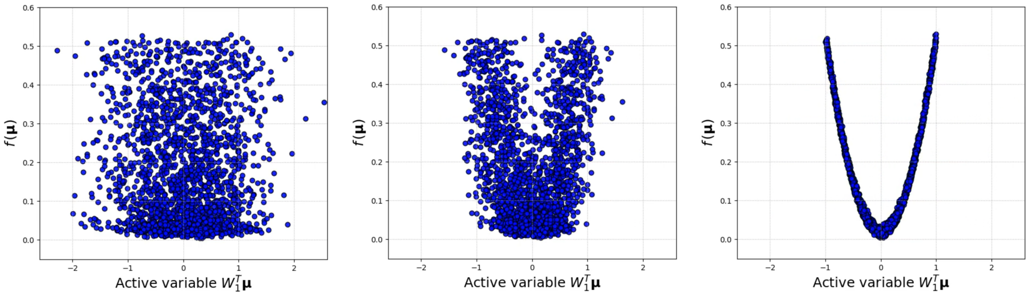


Figure 2.1: Sufficient summary plot for a 7-dimensional hyperparaboloid for three different rotation of the input parameter space. Last panel shows the model function along the active subspace.

Algorithm 1 Active subspace computation.

```
    Input:
      gradients dataset $dY = (dy_1, \dots, dy_M)^T,\ dy_i \in \mathcal{M}(d \times m)$
      spd metric matrix $R_V \in \mathcal{M}(d \times d)$
      active subspace dimension $r$
    Output:
      active eigenvectors $W_1 = (\mathbf{v}_1, \dots, \mathbf{v}_r),\ \mathbf{v}_i \in \mathbb{R}^m$
      inactive eigenvectors $W_2 = (\mathbf{v}_{r+1}, \dots, \mathbf{v}_m),\ \mathbf{v}_i \in \mathbb{R}^m$
      ordered eigenvalues $(\lambda_1, \dots, \lambda_m)$
1: Compute the covariance matrix with Monte Carlo:
      $\tilde{H} = \frac{1}{M} \sum_{j=1}^{M} dY[j,:,:]^T R_V dY[j,:,:].$
2: Solve the eigenvalue problem:
      $\tilde{H}\mathbf{v}_i = \lambda_i \mathbf{v}_i \qquad \forall i \in \{1, \dots, m\},$
      $W_1 = (\mathbf{v}_1, \dots, \mathbf{v}_r), \quad W_2 = (\mathbf{v}_{r+1}, \dots, \mathbf{v}_m).$
```

2.2.1 *Sampling the inactive variable*

We remark that the projection map onto the active subspace is a surjective map because $\mathbf{W}_1^T$ is defined as a linear projection onto a subspace. So the back-mapping from the active subspace onto Ω is not trivial. Let $\mathbf{x}_M^*$ be a point in the active subspace, we can find K points in the full parameter space which are mapped onto $\mathbf{x}_M^*$ by $\mathbf{W}_1^T$. Following from the decomposition introduced above we have

$$\mathbf{x} = \mathbf{W}_1\mathbf{W}_1^T\mathbf{x} + \mathbf{W}_2\mathbf{W}_2^T\mathbf{x} = \mathbf{W}_1\mathbf{x}_M + \mathbf{W}_2\mathbf{y} \qquad \forall \mathbf{x} \in \Omega, \tag{2.18}$$

with the additional constraint coming from the rescaling of the input parameters needed to apply AS: $-\mathbf{1} \leq \mathbf{x} \leq \mathbf{1}$. With $\mathbf{1}$ we denote the vector in $\mathbb{R}^k$ with all elements equal to 1. To sample the inactive variable $\mathbf{y}$ we need to satisfy

$$-\mathbf{1} \leq \mathbf{W}_1\mathbf{x}_M^* + \mathbf{W}_2\mathbf{y} \leq \mathbf{1}, \tag{2.19}$$

or equivalently

$$\mathbf{W}_2\mathbf{y} \leq \mathbf{1} - \mathbf{W}_1\mathbf{x}_M^* \qquad -\mathbf{W}_2\mathbf{y} \leq \mathbf{1} + \mathbf{W}_1\mathbf{x}_M^*. \tag{2.20}$$

These inequalities define a polytope in $\mathbb{R}^{k-M}$ from which we want to uniformly sample K points. The inactive variables are in general sampled from the conditional distribution $p(\mathbf{y}|\mathbf{x}_M^*)$, here we show how to perform it for the uniform distribution. For more general distributions one should use Hamiltonian Monte Carlo. Here we start with a rejection sampling scheme, which finds a bounding hyperbox for the polytope, draws points uniformly from it, and rejects points outside the polytope. In case this method does not return enough samples, we use a *hit and run* method [19, 166, 252] for sampling from the polytope. This method starts from the center of the largest hypersphere within the polytope, then selects a random direction and identifies the longest segment lying inside the polytope. The new sample is randomly drawn along this segment. This procedure continues by iterating the same steps starting from the last sample until K samples are found. If also that does not work, we select K copies of the Chebyshev center [32] of the polytope. In figure 2.2 we depicted the sampling strategy, at different stages.

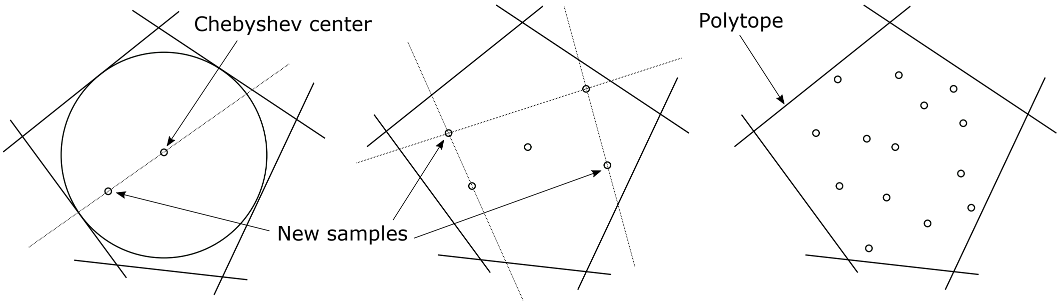


Figure 2.2: Illustration of the inactive variable sampling strategy. Successive steps are depicted at once. We highlight the Chebyshev center, the selection of the next sample using the hit and run method, and the polytope defined by equation (2.20).

2.2.2 *Response surfaces*

The term response surface refers to the general procedure of finding the values of a model function f for new inputs without directly computing it but exploiting regression or interpolation from a training set $\{\mathbf{x}_i, f(\mathbf{x}_i)\}$. We will follow [57, 291] for a review of the construction of response surfaces. The procedure for constructing a Gaussian process response is reported in algorithm 2, while in algorithm 3 we show how to exploit the response surface to predict the model function at new input parameters.

Directly applying the Monte Carlo method with N samples we get a reduced approximation of f as

$$(g \circ P_r)(\mathbf{X}) = \mathbb{E}_\mu \left[f(\mathbf{X}) | \sigma(P_r) \right] \approx \frac{1}{N} \sum_{i=1}^{N} f(\hat{P}_r \mathbf{X} + (I_d - \hat{P}_r)\mathbf{Y}_i) =: \hat{g}(\hat{P}_r \mathbf{X}), \quad (2.21)$$

where $\mathbf{Y}_1, \dots, \mathbf{Y}_N$ are independent and identically distributed samples of $\mathbf{Y} \sim \rho$, and $\hat{P}_r$ is an approximation of P_r obtained discretizing H with Monte Carlo. An intermediate approximation error is obtained employing the Poincaré inequality and the central limit theorem for the Monte Carlo approximation

$$\mathbb{E}_\mu \left[(f(\mathbf{X}) - \hat{g}(\hat{P}_r \mathbf{X}))^2 \right] \leq C_1 \left(1 + N^{-1/2} \right)^2 (\lambda_{n+1} + \cdots + \lambda_m), \quad (2.22)$$

where C_1 is a constant, and $\lambda_{n+1}, \dots, \lambda_m$ are the eigenvalues of the inactive subspace of H [Theorem 4.4, [57]].

In practice $\hat{g}(\hat{P}_r \mathbf{X})$ is approximated with a regression or an interpolation such that a response surface $\mathscr{R}$ satisfying $\mathbb{E}_\mu \left[(\hat{g}(\hat{P}_r \mathbf{x}) - \mathscr{R}(\hat{P}_r \mathbf{x})^2) \right] \leq C_2 \delta$ is built, where C_2

is a constant, and δ depends on the chosen method. An estimate for the successive approximations

$$f(\mathbf{X}) \approx g_\epsilon(P_r\mathbf{X}) \approx \hat{g}_{\epsilon,N}(\hat{P}_r\mathbf{X}) \approx \mathscr{R}_{\epsilon,N,\delta}(\hat{P}_r\mathbf{X}), \tag{2.23}$$

is given by

$$\begin{aligned} &\mathbb{E}_\mu\left[(f(\mathbf{X}) - \mathscr{R}(\hat{P}_r\mathbf{X}))^2\right] \\ &\quad\leq\ C_1(1+N^{-1/2})^2\left(\epsilon(\lambda_1+\cdots+\lambda_n)^{1/2} + (\lambda_{n+1}+\cdots+\lambda_m)^{1/2}\right)^2 + C_2\lambda \end{aligned}$$

where $\mathrm{dist}(\mathrm{Im}(P_r), \mathrm{Im}(\hat{P}_r)) \leq \epsilon$, and λ_i are the eigenvalues of H [Theorem 4.8, [57]].

In our numerical simulations we will build the response surface $\mathscr{R}$ with GPR [287].

Algorithm 2 Response surface construction with Gaussian process regression over the active subspace.

```
Input:
    normalized input dataset $X = (\mathbf{x}_1, \dots, \mathbf{x}_M)^T,\ \mathbf{x}_i \in \mathbb{R}^m$
    output dataset $Y = (y_1, \dots, y_M)^T,\ y_i \in \mathbb{R}$
    active eigenvectors $W_1 = (\mathbf{v}_1, \dots, \mathbf{v}_r),\ \mathbf{v}_i \in \mathbb{R}^m$
    kernel $k : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$
Output:
    trained Gaussian process
Project inputs in the active subspace:
    $XW_1 = \tilde{X} \in \mathcal{M}(M \times r).$
Evaluate the Gram matrix:
    $K_{ij} = k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j), \qquad 1 \leq i,j \leq r.$
Tune the hyperparameters maximizing the maximum likelihood:
    $\mathbf{f} \sim \mathcal{N}(Y, K), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma I_M), \quad \mathbf{y}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) + \boldsymbol{\epsilon},$
    $p(\mathbf{y}|\mathbf{x}, \mathbf{f}, \boldsymbol{\omega}, \sigma) = \mathcal{N}(\mathbf{0}, K + \sigma I_M),$
    $(\boldsymbol{\omega}, \sigma) = \underset{(\boldsymbol{\omega},\sigma)}{\mathrm{argmin}}(-\log(p(\mathbf{y}|\mathbf{x}, \mathbf{f}, \boldsymbol{\omega}, \sigma))).$
```

2.3 kernel-based active subspaces

The kernel-based extension of the AS property we present is based on the preliminary works in the context of supervised dimension reduction algorithms in machine learning in [194].

Keeping the notations of assumption 1, $\mathbf{X} : (\Omega, \mathcal{F}, P) \to \mathbb{R}^m$ is the absolutely continuous random vector representing the m-dimensional inputs with density $\rho : \mathcal{X} \subset \mathbb{R}^m \to \mathbb{R}$, and $f : \mathcal{X} \subset \mathbb{R}^m \to (V, R_V)$ is the model function that we assume to be continuously differentiable and Lipschitz continuous.

Algorithm 3 Prediction phase using the Gaussian process response surface over the active subspace.

```
Input:
    trained response surface, $\mathbf{y}(\mathbf{x})$
    active eigenvectors $W_1 = (\mathbf{v}_1, \dots, \mathbf{v}_r),\ \mathbf{v}_i \in \mathbb{R}^m$
    test samples $\bar{\mathbf{x}},\ \bar{\mathbf{x}}_i \in \mathbb{R}^m$
Output:
    prediction $t$
Map the test samples $\bar{\mathbf{x}}$ onto the active subspace: $\tilde{\mathbf{x}} = W_1 \bar{\mathbf{x}}$.
Evaluate the Gaussian process on $\tilde{\mathbf{x}}$:
    $t = \mathbf{f}(\tilde{\mathbf{x}}),$
    $\mathbb{E}[t] = k(\tilde{\mathbf{x}}, X) K^{-1} \mathbf{f},$
    $\sigma^2(t) = k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) - k(\tilde{\mathbf{x}}, X) K^{-1} k(X, \tilde{\mathbf{x}}).$
```

One drawback of sufficient dimension reduction with AS applied to ridge approximation is that if a clear linear trend is missing, projecting the inputs as $P_r\mathbf{X}$ represents a loss of accuracy on the approximation of the model f that may not be compensated even by the choice of the optimal profile $\tilde{h} \circ P_r = \mathbb{E}_\rho[f|\sigma(P_r)]$. In order to overcome this, non-linear dimension reduction to one-dimensional parameter space could be achieved discovering a curve in the space of parameters that cuts transversely the level sets of f, this variation is presented in [35] as Active Manifold. Another approach could consist in finding a diffeomorphism ϕ that reshapes the level sets such that subsequently applying AS dimension reduction to the new model function $\tilde{f} \circ \phi = f$ could be more profitable:

$$\begin{tikzcd}
\mathcal{X} \subset \mathbb{R}^m \arrow[r, "\phi"] \arrow[dr, "f"'] & \phi(\mathcal{X}) \subset \mathbb{R}^m \arrow[d, "\tilde{f}"] \\
 & V
\end{tikzcd}$$

Unfortunately constructing the Active Manifold or finding the right diffeomorphism ϕ could be a complicate matter. If we renounce to have a backward map and we weaken the bond of the method with the model, we can consider an immersion ϕ from the space of parameters $\mathcal{X}$ to an infinite-dimensional Hilbert space $\mathbb{H}$ obtaining

$$\begin{tikzcd}
\mathcal{X} \subset \mathbb{R}^m \arrow[r, "\phi"] \arrow[dr, "f"'] & \phi(\mathcal{X}) \subset \mathbb{H} \arrow[d, "\tilde{f}"] \\
 & V
\end{tikzcd}$$

This is a common procedure in machine learning in order to increase the number of features [287]. Then AS is applied to the new model function $\tilde{f} : \phi(\mathcal{X}) \subset \mathbb{H} \to V$ with parameter space $\phi(\mathcal{X}) \subset \mathbb{H}$. A response surface can be built with algorithm 2 remembering to replace every occurrence of the inputs $\mathbf{x}$ with their images $\phi(\mathbf{x})$. A synthetic scheme of the procedure is represented in figure 2.3.

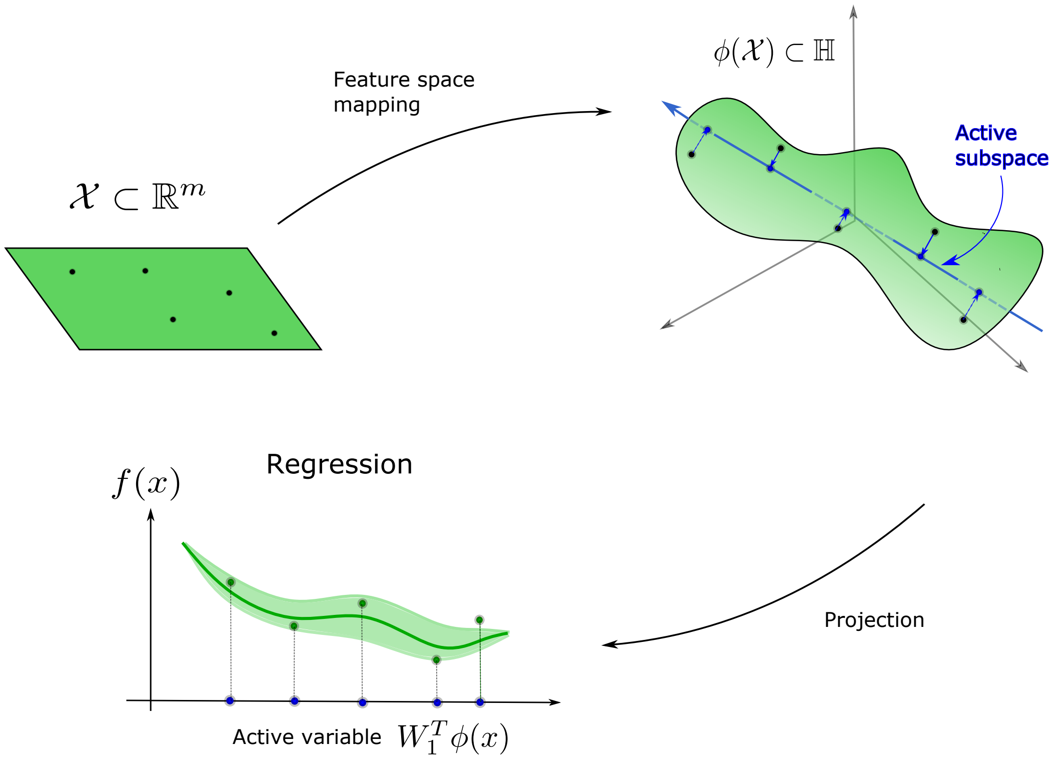

$\phi(\mathcal{X}) \subset \mathbb{H}$
Feature space mapping
Active subspace
$\mathcal{X} \subset \mathbb{R}^m$
Regression
$f(x)$
Projection
Active variable $W_1^T\phi(x)$

Figure 2.3: Illustration of the construction of a one-dimensional response surface with kernel-based active subspaces and Gaussian process regression.

Remark 3. *In this case the AS method may be theoretically formulated as an application of the spectral decomposition to the integral operator $T_{\mathcal{H}} : \mathbb{H} \to \mathbb{H}$*

$$\mathcal{H} = \int_{\phi(\mathcal{X})} (R_V \circ d_{\mathbf{z}}\tilde{f}(\mathbf{z})) \otimes d_{\mathbf{z}}\tilde{f}(\mathbf{z})\, d\mu(\mathbf{z}),$$

since it is adjoint and compact.

In practice we consider a discretization of the infinite-dimensional Hilbert space $\mathbb{R}^D \simeq \mathbb{H}$ with $D > m$. Sufficient dimension reduction with AS results in the choice of a r-rank projection in the much broader set of r-rank projections in $\mathbb{H}$.

Since for AS only the samples of the Jacobian matrix of the model function are employed, we can ignore the definition of the new map $\tilde{f} : \phi(\mathcal{X}) \subset \mathbb{H} \to (V, R_V)$ and focus only on the computation of the Jacobian matrix of $\tilde{f}$ with respect to the new input variable $\mathbf{z} := \phi(\mathbf{x})$. Taking the difference of the Jacobian $D_{\mathbf{z}}\tilde{f}$ with its mean in case it is not a centered random matrix [160], the covariance matrix becomes

$$\begin{aligned}\tilde{\mathcal{H}} &= \int_{\phi(\mathcal{X})} \left[(D_{\mathbf{z}}\tilde{f})^T(\mathbf{z})\right] R_V \left[(D_{\mathbf{z}}\tilde{f})(\mathbf{z})\right]\, d\mu(\mathbf{z}) \\ &= \int_{\mathcal{X}} \left[(D_{\mathbf{z}}\tilde{f})^T(\phi(\mathbf{x}))\right] R_V \left[(D_{\mathbf{z}}\tilde{f})(\phi(\mathbf{x}))\right]\, d\mathcal{L}_{\mathbf{X}}(\mathbf{x}),\end{aligned}$$

where $\mu := \phi \circ \mathcal{L}_{\mathbf{X}}$ is the pushforward probability measure of $\mathcal{L}_{\mathbf{X}}$, the law of probability of $\mathbf{X}$, with respect to the map ϕ. Simple Monte Carlo can be applied sampling from the distribution ρ in the input space $\mathcal{X}$

$$\begin{aligned}\tilde{\mathcal{H}} &= \int_{\mathcal{X}} \left[(D_{\mathbf{z}}\tilde{f})^T(\phi(\mathbf{x}))\right] R_V \left[(D_{\mathbf{z}}\tilde{f})(\phi(\mathbf{x}))\right] \, d\mathcal{L}_{\mathbf{X}}(\mathbf{x}) \\ &\approx \frac{1}{M}\sum_{i=1}^{M} \left[(D_{\mathbf{z}}\tilde{f})^T(\phi(\mathbf{x}_i))\right] R_V \left[(D_{\mathbf{z}}\tilde{f})(\phi(\mathbf{x}_i))\right].\end{aligned}$$

The gradients of $\tilde{f}$ with respect to the new input variable $\mathbf{Z}$ are computed from the known values $D_{\mathbf{x}}f$ with the chain rule.

Remark 4. *The application of the chain rule to the composition of functions $\tilde{f} \circ \phi : \mathbb{R}^m \to \mathbb{H} \to V$ is theoretically validated if $\tilde{f}$ is defined in an open set $U \supset \phi(\mathcal{X})$. If ϕ is non singular and also injective the new input space is a m-dimensional submanifold of $\mathbb{H}$. If ϕ is also smooth there exists a smooth extension of $\tilde{f} : \phi(\mathcal{X}) \subset \mathbb{H} \to V$, see [160].*

If the Hilbert space $\mathbb{H}$ has finite dimension $\mathbb{H} \sim \mathbb{R}^D$ this procedure leaves us with an underdetermined linear system to solve for $D_{\mathbf{z}}\tilde{f}$

$$\begin{aligned}&D_{\mathbf{z}}\tilde{f}(\phi(\mathbf{x}))D\phi(\mathbf{x}) = D_{\mathbf{x}}f(\mathbf{x}), \\ &D_{\mathbf{z}}\tilde{f}(\phi(\mathbf{x})) = D_{\mathbf{x}}f(\mathbf{x})(D\phi(\mathbf{x}))^{\dagger},\end{aligned} \tag{2.24}$$

where † stands for the right Moore-Penrose inverse of the matrix $D\phi(\mathbf{x})$ with rank r, that is

$$(D\phi(\mathbf{x}))^{\dagger} = V\Sigma^{\dagger}U^T,$$

with the usual notation for the SVD of $D\phi(\mathbf{x})$

$$D\phi(\mathbf{x}) = U\Sigma V^T, \tag{2.25}$$

and $\Sigma^{\dagger} \in \mathcal{M}(r \times r)$ equal to the diagonal matrix with the inverse of the singular values as diagonal elements.

Remark 5. *In the AS method we approximate the random variable $\mathbf{X}$ as*

$$P_r\mathbf{X} = \mathbf{v}_1(\mathbf{v}_1 \cdot \mathbf{X}) + \cdots + \mathbf{v}_r(\mathbf{v}_r \cdot \mathbf{X}), \tag{2.26}$$

with $\{\mathbf{v}_i\} \subset \mathbb{R}^m$ the active eigenvectors, whereas with KAS the reduced input space is contained in $\mathcal{H}$

$$P_r\mathbf{X} = \mathbf{v}_1(\mathbf{v}_1 \cdot \phi(\mathbf{X})) + \cdots + \mathbf{v}_r(\mathbf{v}_r \cdot \phi(\mathbf{X})), \tag{2.27}$$

with $\{\mathbf{v}_i\} \subset \mathcal{H}$ the active eigenvectors of KAS. In this case the model is enriched by the non-linear feature map ϕ.

Algorithm 4 Kernel-based active subspace computation.

```
Input:
    gradients dataset $dY = (dy_1, \dots, dy_M)^T,\ dy_i \in \mathcal{M}(d \times m)$
    spd metric matrix $R_V \in \mathcal{M}(d \times d)$
    feature subspace dimension $D$
    feature map $\phi : \mathbb{R}^m \to \mathbb{R}^D$
    active subspace dimension $r$
Output:
    active eigenvectors $W_1 = (\mathbf{v}_1, \dots, \mathbf{v}_r),\ \mathbf{v}_i \in \mathbb{R}^D$
    inactive eigenvectors $W_2 = (\mathbf{v}_{r+1}, \dots, \mathbf{v}_D),\ \mathbf{v}_i \in \mathbb{R}^D$
    ordered eigenvalues $(\lambda_1, \dots, \lambda_D)$
Evaluate gradients solving an overdetermined linear system:
    $\forall j \in \{1, \dots, M\}, \quad dY[j,:,:](D\phi)^\dagger = d\tilde{Y}[j,:,:] \in \mathcal{M}(d, D).$
Compute the covariance matrix with Monte Carlo:
    $\tilde{H} = \frac{1}{M} \sum_{k=1}^{M} d\tilde{Y}[k,:,:]^T R_V d\tilde{Y}[k,:,:].$
Solve the eigenvalue problem:
    $\tilde{H}\mathbf{v}_i = \lambda_i \mathbf{v}_i \qquad \forall i \in \{1, \dots, D\},$
    $W_1 = (\mathbf{v}_1, \dots, \mathbf{v}_r), \quad W_2 = (\mathbf{v}_{r+1}, \dots, \mathbf{v}_D).$
```

2.3.1 *Choice of the Feature Map*

The choice for the map ϕ suggested in [194] is linked to the theory of Reproducing Kernel Hilbert Spaces (RKHS) [23], and it is defined as

$$\mathbf{z} = \phi(\mathbf{x}) = \sqrt{\frac{2}{D}}\sigma_f \cos(W\mathbf{x} + \mathbf{b}), \tag{2.28}$$

$$\cos(W\mathbf{x} + \mathbf{b}) := \frac{1}{\sqrt{D}}(\cos(W[1,:] \cdot \mathbf{x} + b_1), \dots, \cos(W[D,:] \cdot \mathbf{x} + b_D))^T \tag{2.29}$$

where σ_f is an hyperparameter corresponding to the empirical variance of the model, $W \in \mathcal{M}(D \times m, \mathbb{R})$ is the projection matrix whose rows are sampled from a probability distribution μ on $\mathbb{R}^m$ and $\mathbf{b} \in \mathbb{R}^D$ is a bias term whose components are sampled independently and uniformly in the interval $[0, 2\pi]$. We remark that its Jacobian can be computed analytically as

$$\frac{\partial z^j}{\partial x^i} = -\sqrt{\frac{2}{D}}\sigma_f \sin\left(\sum_{k=1}^{D} W_{ik}\mathbf{x}_k + \mathbf{b}_k\right) W_{ij}, \tag{2.30}$$

for all $i \in \{1, \dots, m\}$, and for all $j \in \{1, \dots, D\}$.

Remark 6. *In order to guarantee the correctness of the procedure for evaluating the gradients we have to prove that the feature map is injective and non singular. In general* *however the feature map in equation (2.28) is not injective due to the periodicity of the cosine but at least it is almost surely non singular if the dimension of the feature space is high enough.*

Remark 7 (Other choices for the feature map)**.** *The feature map in equation (2.28) is not the only effective immersion that provides a kernel-based extension of the active subspaces. For example an alternative is the following composition of a linear map with a sigmoid*

$$\phi(\mathbf{z}) = \frac{C}{1 + \alpha\, e^{-W\mathbf{z}}},$$

where C is a constant, α is an hyperparameter to be tuned, and $W \in \mathcal{M}(D, m)$ is, as before, a matrix whose rows are sampled from a probability distribution on $\mathbb{R}^m$.

Other choices involve the use of Deep Neural Networks to learn the profile h and the projection function P_r of the ridge approximation problem [274].

Remark 8. *The tuning of the hyperparameters of the spectral measure chosen is the most computationally expensive part of the procedure. It consists in a global optimization problem where the dimension of the domain can vary between 1 and the dimension of the input space m. The object function to optimize is the RRMSE*

$$RRMSE(Y_{test}, T_{test}) = \sqrt{\frac{\sum_{i=1}^{N}(t_i - y_i)^2}{\sum_{i=1}^{N}(t_i - \bar{y})^2}}, \tag{2.31}$$

where $T_{test} = (t_i)_{i\in\{1,\dots,N\}}$ are the predictions obtained from the one-dimensional response surface built with KAS and associated to the test set, $Y_{test} = (y_i)_{i\in\{1,\dots,N\}}$ are the targets associated to the test set, and $\bar{y}$ is the mean value of the targets. We implemented a logarithmic grid-search in algorithm 5 making use of the SciPy library [280]. Another choice could be Bayesian stochastic optimization implemented in [91].

2.3.2 *Random Fourier Features*

The motivation behind the choice for the map in equation (2.28) comes from the theory on Reproducing Kernel Hilbert Spaces. The infinite-dimensional Hilbert space $(\mathbb{H}, \langle\cdot,\cdot\rangle)$ is assumed to be a RKHS with real shift-invariant kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with $k(0) = 1$ and feature map ϕ.

In order to get a discrete approximation of $\phi : \mathcal{X} \subset \mathbb{R}^m \to \mathbb{H}$, random Fourier features are employed [163, 214]. Bochner's theorem [183] guarantees the existence of a spectral probability measure μ such that

$$k(\mathbf{x}, \mathbf{y}) = \int_{\mathbb{R}^m} e^{i\boldsymbol{\omega}\cdot(\mathbf{x}-\mathbf{y})}\, d\mu(\boldsymbol{\omega}).$$

From this identity we can get a discrete approximation of the scalar product $\langle\cdot,\cdot\rangle$ with Monte Carlo method, exploiting the fact that the kernel is real

$$\langle\phi(\mathbf{x}), \phi(\mathbf{y})\rangle = k(\mathbf{x}, \mathbf{y}) \approx \frac{1}{D}\sum_{i=1}^{D} \cos(\boldsymbol{\omega}_i \cdot \mathbf{x} + b_i)\cos(\boldsymbol{\omega}_i \cdot \mathbf{y} + b_i) = \mathbf{z}^T\mathbf{z}, \tag{2.32}$$

$$\mathbf{z} = \frac{1}{\sqrt{D}}(\cos(\boldsymbol{\omega}_1 \cdot \mathbf{x} + b_1), \dots, \cos(\boldsymbol{\omega}_D \cdot \mathbf{x} + b_D)), \tag{2.33}$$

Algorithm 5 Tuning the feature map with logarithmic grid-search.

```
Input:
  normalized input dataset $X = (\mathbf{x}_1, \dots, \mathbf{x}_M)^T$, $\mathbf{x}_i \in \mathbb{R}^m$
  output dataset $Y = (y_1, \dots, y_M)^T$, $y_i \in \mathbb{R}$
  gradients dataset $dY = (dy_1, \dots, dy_M)^T$, $dy_i \in \mathcal{M}(d \times D)$
  spd metric matrix $R_V \in \mathcal{M}(d \times d)$
  feature subspace dimension $D$
  feature map $\phi : \mathbb{R}^m \to \mathbb{R}^D$
  spectral density with hyperparameter $\alpha$, $\mu$
  active subspace dimension $r$
  tolerance for the tuning procedure tol $\approx 0.8$
Output:
  projection matrix $W$
  bias $b$
Create the grid $G$ and set the variable BEST to 1.
for $\alpha \in G$ do
    Compute the feature map projection matrix $W$ associated to the
    hyperparameter $10^\alpha$, and the uniformly sampled bias $b$:
        $W[i,:]$ sampled from $\mu(10^\alpha)$,   $\forall i \in \{1, \dots, D\}$,
        $b[i] \sim \mathcal{U}(0, 2\pi)$.
    Compute score with $n$-fold cross validation:
    for $i \leftarrow 1$ to $n$ do
        Divide input, output, and gradients in train and test datasets.
        Compute $(W_1, W_2, (\lambda_1, \dots, \lambda_D))$ with KAS method in algorithm 4
        with inputs $(dY_{\text{train}}, R_V, D, \phi, r)$.
        Build the GPR response surface with inputs $(X_{\text{train}}, Y_{\text{train}}, W_1, k)$
        using algorithm 2.
        Predict the values $T_{\text{test}}$ using algorithm 3 with input $X_{\text{test}}$.
        Evaluate the score as score$[n]$ = RRMSE$(Y_{\text{test}}, T_{\text{test}})$.
        if score$[n] >$ tol then
            Stop cross validation and pass to the next value of $\alpha$.
        end if
    end for
    if mean(score) <BEST then
        Save $W$ and $b$, and set BEST to mean(score).
    end if
end for
```

and from this relation we obtain the approximation $\phi \approx \mathbf{z}$. The sampled vectors $\{\boldsymbol{\omega}_i\}_{i=1,\ldots,D}$ are called random Fourier features. The scalars $\{b_i\}_{i=1,\ldots,D}$ are bias terms introduced since in the approximation we have excluded some trigonometric terms from the following initial expression

$$\frac{1}{D}\sum_{i=1}^{D}\left(\cos(\boldsymbol{\omega}_i \cdot \mathbf{x})\cos(\boldsymbol{\omega}_i \cdot \mathbf{y}) - \sin(\boldsymbol{\omega}_i \cdot \mathbf{x})\sin(\boldsymbol{\omega}_i \cdot \mathbf{y})\right).$$

Random Fourier features are frequently used to approximate kernels. We consider only spectral probability measures which have a probability density, usually named spectral density. Under some regularity conditions on the kernel, an explicit probabilistic bound depending on the dimension of the feature space D can be proved [183]. This technique is used to scale up Kernel Principal Component Analysis [236, 238] and Supervised Kernel Principal Component Analysis [15], but in the case of KAS the resulting overdetermined linear system employed to compute the Jacobian matrix of the new model function increases in dimension instead.

Remark 9 (Radial basis function kernel)**.** *The most famous kernel is the squared exponential kernel also called Radial Basis Functions (RBF) kernel*

$$k_{RBF}(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2l^2}\right), \tag{2.34}$$

where l is the characteristic length-scale. The spectral density is Gaussian $\mathcal{N}(0,\ 1/4\pi^2 l^2)$:

$$S(\boldsymbol{\omega}) = (2\pi l^2)^{D/2} exp(-2\pi^2 l^2 \boldsymbol{\omega}^2). \tag{2.35}$$

Thanks to Bochner's theorem to every probability distribution that admits a probability density function corresponds a stationary positive definite kernel. So having in mind the definition of the feature map ϕ from equation (2.28), we can choose any probability distribution for sampling the random projection matrix $W \in \mathcal{M}(D \times m, \mathbb{R})$ without focusing on the corresponding kernel since it is not needed by the numerical procedure.

After the choice of the spectral measure the corresponding hyperparameters have to be tuned. This is linked to the choice of the hypothesis model in machine learning and it is usually carried out for the hyperparameters of the employed kernel. From the choice of the kernel and the corresponding hyperparameters some regularity properties of the model are implicitly assumed [287].

2.4 BENCHMARK TEST PROBLEMS

In this section we are going to present some benchmark tests to prove the potential gain of KAS over standard linear AS, for both scalar and vectorial model functions. In particular we test KAS on radial symmetric functions, with 2-dimensional and 8-dimensional parameter spaces, on the approximation of the reproduction number R_0 of the SEIR model, and finally on a vectorial output function that is the solution of a Poisson problem.

One dimensional response surfaces are built following the algorithm described in section 2.2.2. The tuning of the hyperparameters of the feature map is carried out with a logarithmic grid-search and 5-fold cross validation for the Ebola test case, while for the other cases we employed Bayesian stochastic optimization implemented in [91] with 3-fold cross validation. The score function chosen is the RRMSE. For the radial symmetric and Ebola test cases the inputs are sampled from a uniform distribution with problem dependent ranges. For the stochastic elliptic partial differential case the inputs are the coefficients of a Karhunen–Loève expansion and are sampled from a normal distribution.

2.4.1 *Radial symmetric functions*

Radial symmetric functions represent a class of model functions for which AS is not able to unveil any low dimensional behaviour. In fact for these functions any rotation of the parameter space produce the same model representation. Instead KAS is able to overcome this problem thanks to the mapping onto the feature space.

We present two benchmarks: an 8-dimensional hyperparaboloid defined as

$$f : [-1, 1]^8 \subset \mathbb{R}^8 \to \mathbb{R}, \qquad f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2, \tag{2.36}$$

and the surface of revolution in $\mathbb{R}^3$ with generatrix $g(x) = \sin(x^2)$ that reads as follows

$$f : [-3, 3]^2 \subset \mathbb{R}^2 \to \mathbb{R}, \qquad f(\mathbf{x}) = \sin(\|\mathbf{x}\|^2). \tag{2.37}$$

The gradients are computed analytically.

For the hyperparaboloid we used $N_s = 500$ independent, uniformly distributed training samples in $[-1, 1]^8$, while for the sine case the training samples are $N_s = 800$ in $[-3, 3]^2$. In both cases the test samples are 500. The feature space has dimension 1000 for the first case and for the second case both. The spectral distribution chosen is the multivariate normal with hyperparameter a uniform variance λI_d, and a product of Laplace distributions with μ and b as hyperparameters, respectively. The tuning is carried out with 3-fold cross validation. The results are summarized in table 2.1.

Table 2.1: Performance results for AS and KAS methods. For each case we report the parameter space dimension, the number of samples N_s used for the training, the chosen distribution, the dimension of the feature space, and the RRMSE mean and standard deviation for AS and KAS. In bold the best results.

Case	Dim	N_s	Spectral distribution	Feature space dim	RRMSE AS	RRMSE KAS
Hyperparaboloid	8	500	$\mathcal{N}(\mathbf{0}, \lambda I_d)$	1000	0.98 ± 0.03	**0.23** ± 0.02
Sine	2	800	Laplace(μ, b)	1000	1.011 ± 0.01	**0.31** ± 0.06
Ebola	8	800	Beta(α, β)	1000	0.46 ± 0.31	**0.31** ± 0.03
SPDE (2.42)	10	1000	$\mathcal{N}(\mathbf{0}, \Sigma)$	1500	0.611 ± 0.001	**0.515** ± 0.013

Looking at the eigenvalues of the uncentered covariance matrix of the gradients $\tilde{H}$ for the hyperparaboloid case in figure 2.4, we can clearly see how the decay for AS

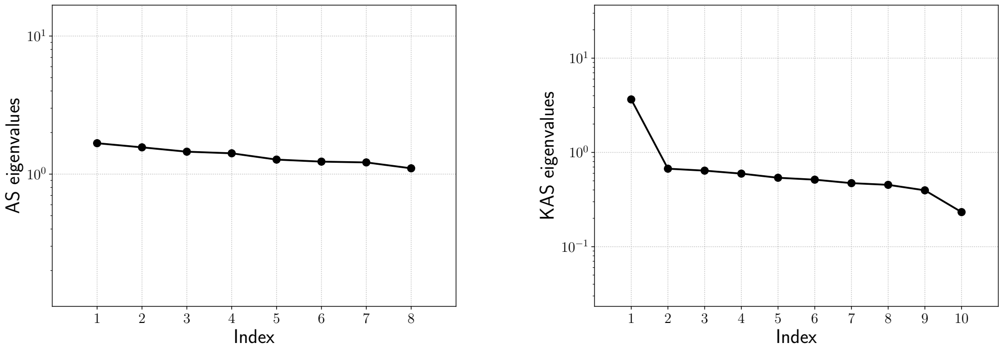


Figure 2.4: Eigenvalues of the covariance matrix $\tilde{H} \in \mathbb{R}^{8\times 8}$ applied to the hyperparaboloid case for the AS procedure on the left, and the first 10 eigenvalues of the covariance matrix $\tilde{H} \in \mathbb{R}^{1000\times 1000}$ for the KAS procedure applied to the same case on the right.

is almost absent, while using KAS the decay after the first eigenvalue is pronounced, suggesting the presence of an active subspace of dimension 1.

The one-dimensional sufficient summary plots, which are $f(\mathbf{x})$ against $W_1^T\mathbf{x}$ — in the AS case — or against $W_1^T\phi(\mathbf{x})$ — in the KAS case —, are shown in figure 2.5 and figure 2.6, respectively. On the left panels we present the Gaussian process response surfaces obtained from the active subspaces reduction, while on the right panels the ones obtained with the kernel-based AS extension. As we can see AS fails to properly reduce the parameter spaces, since there are no preferred directions over which the model functions vary the most. The KAS approach, on the contrary, is able to unveil the generatrix behaviour for the sine case, and a linear trend for the hyperparaboloid. This results in a reduction of the RMS of one order of magnitude, in both cases (see table 2.1).

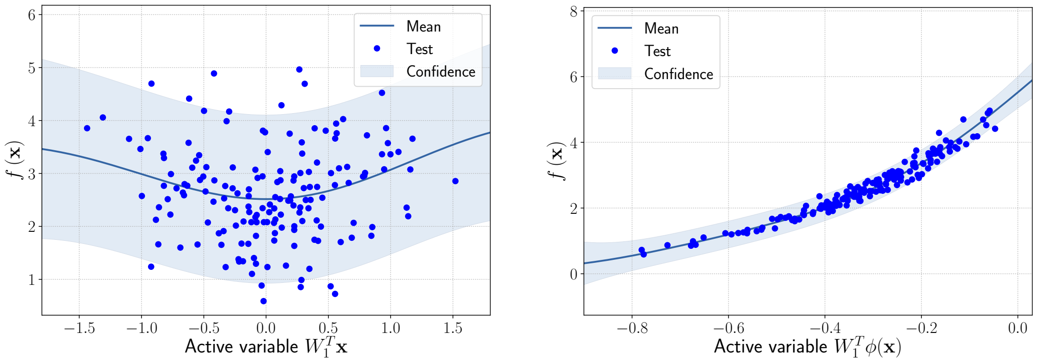


Figure 2.5: Comparison between the sufficiency summary plots obtained from the application of AS and KAS methods for the hyperparaboloid model function with domain $[-1,1]^8$, defined in equation (2.36). The left plot refers to AS, the right plot to KAS. With the blue solid line we depict the posterior mean of the GP, with the shadow area the confidence intervals, and with the blue dots the testing points.

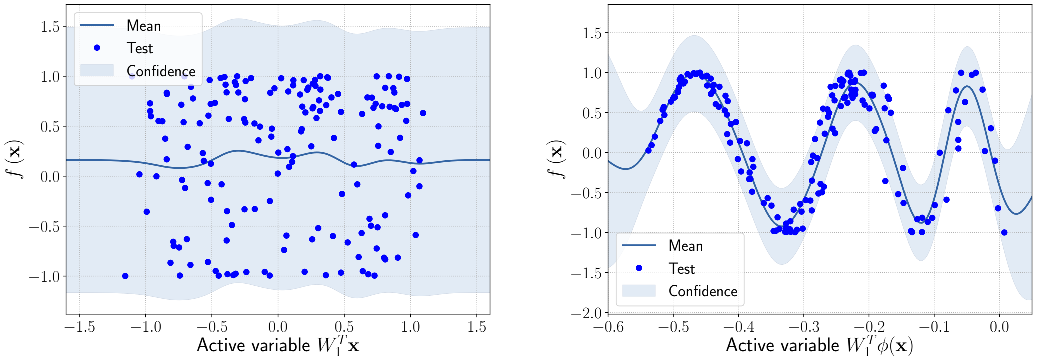


Figure 2.6: Comparison between the sufficiency summary plots obtained from the application of AS and KAS methods for the surface of revolution model function with domain $[-3,3]^2$, defined in equation (2.37). The left plot refers to AS, the right plot to KAS. With the blue solid line we depict the posterior mean of the GP, with the shadow area the confidence intervals, and with the blue dots the testing points.

2.4.2 *SEIR model for the spread of Ebola*

In most engineering applications the output of interest presents a monotonic behaviour with respect to the parameters. This means that, for example, the increment in the inputs produces a proportional response in the outputs. Rarely the model function has a radial symmetry, and in such cases the parameter space can be divided in subdomains, which are analyzed separately. In this section we are going to present a test case where there is no radial symmetry, showing that, even in this case the kernel-based AS presents better performance with respect to AS.

For the Ebola test case[1], the output of interest is the basic reproduction number R_0 of the SEIR model, described in [74], which reads

$$R_0 = \frac{\beta_1 + \frac{\beta_2\rho_1\gamma_1}{\omega} + \frac{\beta_3}{\gamma_2}\psi}{\gamma_1 + \psi}, \tag{2.38}$$

with parameters distributed uniformly in $\Omega \subset \mathbb{R}^8$. The parameter space Ω is defined by the lower and upper bounds summarized in table 2.2.

Table 2.2: Parameter ranges for the ebola model. Data taken from [74].

	β_1	β_2	β_3	ρ_1	γ_1	γ_2	ω	ψ
Lower bound	0.1	0.1	0.05	0.41	0.0276	0.081	0.25	0.0833
Upper bound	0.4	0.4	0.2	1	0.1702	0.21	0.5	0.7

We can compare the two one-dimensional response surfaces obtained with Gaussian process regression. The training samples are $N_s = 800$, and we use 1000 features. As spectral measure we use again the multivariate gaussian distribution $\mathcal{N}(\mathbf{0},\Sigma)$ with hyperparameters the elements of the diagonal of the covariance matrix. The tuning is carried out with 5-fold cross validation. Even in this case, the

[1] The dataset was taken from `https://github.com/paulcon/as-data-sets`.

KAS approach results in smaller RRMSE with respect to the use of AS (around 60% less), as reported in table 2.1. In figure 2.7 we report the comparison of the two approaches over an active subspace of dimension 1.

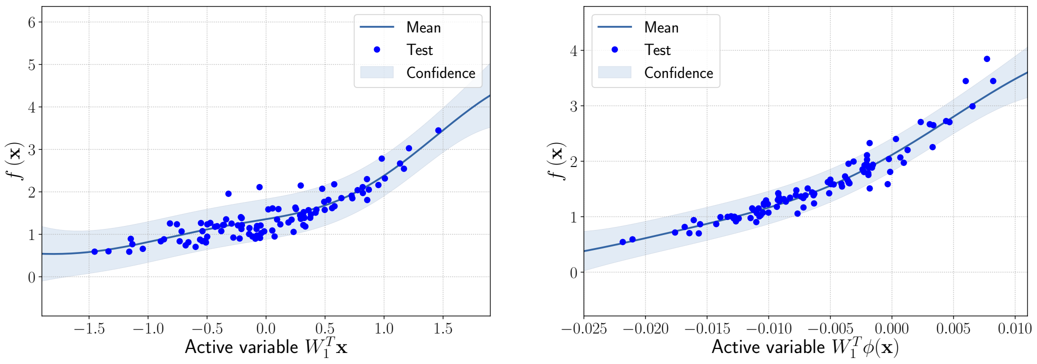


Figure 2.7: Comparison between the sufficiency summary plots obtained from the application of AS and KAS methods for the R_0 model function with domain Ω, defined in equation (2.38). The left plot refers to AS, the right plot to KAS. With the blue solid line we depict the posterior mean of the GP, with the shadow area the confidence intervals, and with the blue dots the testing points.

2.4.3 *Elliptic Partial Differential Equation with random coefficients*

In our last benchmark test case we apply the Kernel-based Active Subspaces to a vectorial model function, that is the solution of a Poisson problem with heterogeneous diffusion coefficient. We refer to [291] for an application, on the same problem, of the AS approach.

We consider the following stochastic Poisson problem on the square $\mathbf{x} = (x, y) \in \Omega := [0, 1]^2$:

$$\begin{cases} -\nabla \cdot (\kappa \ \nabla u) = 1, & \mathbf{x} \in \Omega, \\ u = 0, & \mathbf{x} \in \partial\Omega_{\text{top}} \cup \partial\Omega_{\text{bottom}}, \\ u = 10y(1-y), & \mathbf{x} \in \partial\Omega_{\text{left}}, \\ \mathbf{n} \cdot \nabla u = 0, & \mathbf{x} \in \partial\Omega_{\text{right}}, \end{cases} \tag{2.39}$$

with Neumann boundary condition on the right side of the domain, that is $\partial\Omega_{\text{right}}$, and Dirichlet boundary conditions on the remaining part of $\partial\Omega$. The diffusion coefficient $\kappa : (\Omega, \mathcal{A}, P) \times \Omega \to \mathbb{R}$, where $\mathcal{A}$ is a σ-algebra, is such that $\log(\kappa)$ is a Gaussian random field, with covariance function $C(\mathbf{x}, \mathbf{y})$ defined by

$$C(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{\beta^2} \right), \quad \forall \mathbf{x}, \mathbf{y} \in \Omega, \tag{2.40}$$

where $\beta = 0.03$ is the correlation length. This random field is approximated with the truncated Karhunen–Loève decomposition

$$\kappa(s, \mathbf{x}) \approx \exp\left(\sum_{i=0}^{m} X_i(s) \gamma_i \boldsymbol{\psi}_i(\mathbf{x}) \right) \qquad \forall (s, \mathbf{x}) \in \Omega \times \Omega, \tag{2.41}$$

where $(X_i)_{i\in 1,\ldots,m}$ are independent standard normal distributed random variables, and $(\gamma_i, \boldsymbol{\psi}_i)_{i\in 1,\ldots,d}$ are the eigenpairs of the Karhunen–Loève decomposition of the zero-mean random field κ.

In our simulation the domain Ω is discretized with a triangular unstructured mesh $\mathcal{T}$ with 3194 triangles. The parameter space has dimension $m = 10$. The simulations are carried out with the finite element method (FEM) with polynomial order one, and for each simulation the parameters $(X_i)_{\text{ß}=1,\ldots m}$ are sampled from a standard normal distribution. The solution u is evaluated at $d = 1668$ degrees of freedom, thus $(V, R_V) \approx (\mathbb{R}^d, S + M)$ where the metric R_V is approximated with the sum of the stiffness matrix $S \in \mathbb{R}^d \times \mathbb{R}^d$ and the mass matrix $M \in \mathbb{R}^d \times \mathbb{R}^d$. This sum is a discretization of the norm of the Sobolev space $H^1(\Omega)$. The number of features used in the KAS procedure is $D = 1500$, the number of different independent simulations is $M = 1000$.

Three outputs of interest are considered. The first target function $f : \mathbb{R}^m \to \mathbb{R}$ is the mean value of the solution at the right boundary $\partial\Omega_{\text{right}}$, which reads

$$f(\mathbf{X}) = \frac{1}{|\partial\Omega_{\text{right}}|} \int_{\partial\Omega_{\text{right}}} u(s)\, ds, \tag{2.42}$$

and it is used to tune the feature map minimizing the RRMSE of the Gaussian process regression, as described in algorithm 5. A summary of the results for the first output is reported in table 2.1. The plots of the regression are reported in figure 2.8. Even in this case both from a qualitative and a quantitative point of view, the kernel-based approach achieves the best results.

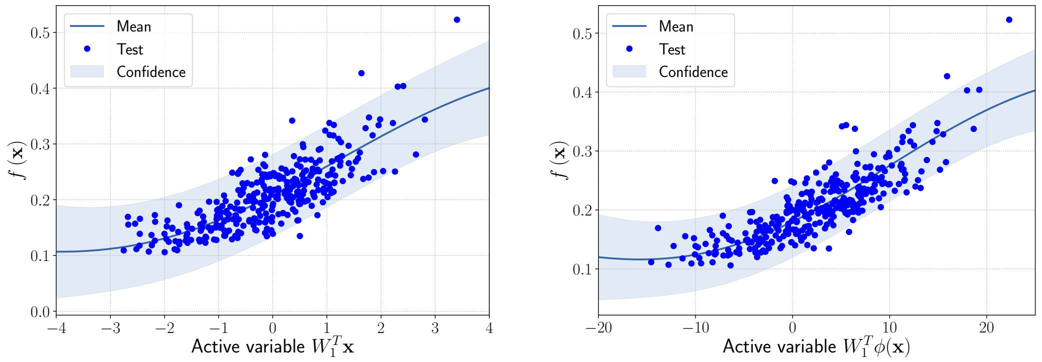


Figure 2.8: Comparison between the sufficiency summary plots obtained from the application of AS and KAS methods for the stochastic PDE model, defined in equation (2.39) and equation (2.42). The left plot refers to AS, the right plot to KAS. With the blue solid line we depict the posterior mean of the GP, with the shadow area the confidence intervals, and with the blue dots the testing points.

The second output we consider is the solution function

$$f : \mathbb{R}^m \to (V, R_V) \approx (\mathbb{R}^d, S), \qquad f(\mathbf{X}) = u \in \mathbb{R}^d. \tag{2.43}$$

This output can be employed as a surrogate model to predict the solution u given the parameters $\mathbf{X}$ that define the diffusion coefficient instead of carrying out the numerical simulation. It can be shown that the AS and KAS modes are distinguished but can detect some common regions of interest as shown in table 2.3.

The third output is the evaluation of the solution at a specific degree of freedom with index $\hat{\imath}$, that is

$$f : \mathbb{R}^m \to \mathbb{R}, \qquad f(\mathbf{X}) = u_{\hat{\imath}} \in \mathbb{R}, \tag{2.44}$$

in this case the dimension of the input space is $m = 100$. Since we use a Lagrangian basis in the finite element formulation and the polinomial order is 1, the node of the mesh associated to the chosen degree of freedom has coordinates $[0.27, 0.427] \in \Omega$. Qualitatively we can see from table 2.3 that the AS modes locate features in the domain which are relatively more regular with respect to the KAS modes. To obtain this result we increased the dimension of the input space, otherwise not even the AS modes could locate properly the position in the domain Ω of the degree of freedom.

In the second and third case the diffusion coefficient is given by

$$\kappa(\mathbf{x}) = \exp\left(\sum_{i=1}^{D} \mathbf{v}_j[i]\tilde{\boldsymbol{\psi}}_j(\mathbf{x})\right) \qquad \forall (s, \mathbf{x}) \in \Omega \times \Omega, \tag{2.45}$$

where $\mathbf{v}_j \in \mathbb{R}^D$, $j \in \{1, \dots, D\}$, is the j-th active eigenvector from the KAS procedure and the functions $\tilde{\Psi} := (\tilde{\boldsymbol{\psi}}_1, \dots, \tilde{\boldsymbol{\psi}}_D)$ are defined by

$$\tilde{\Psi} = \phi(\Psi), \tag{2.46}$$

where ϕ is the feature map defined in equation (2.28) with the projection matrix W and bias b, and $\Psi := (\gamma_1\boldsymbol{\psi}_1, \dots, \gamma_m\boldsymbol{\psi}_m)$.

Remark 10. *The gradients of the three outputs of interest considered are evaluated with the adjoint method.*

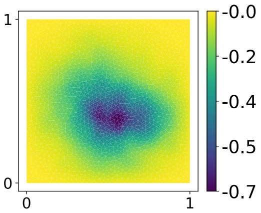
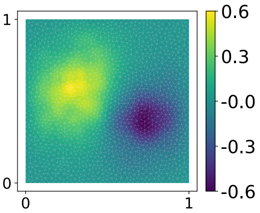
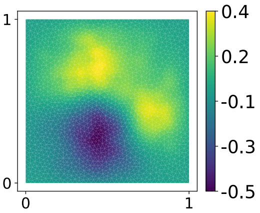
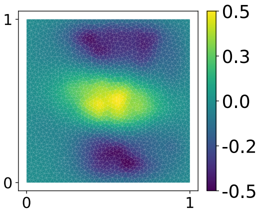
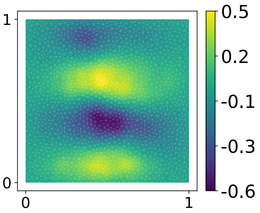
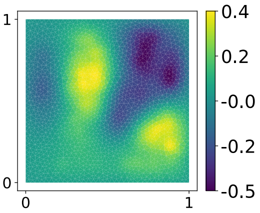
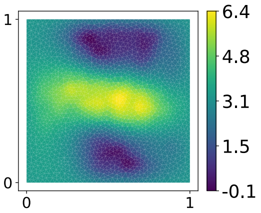
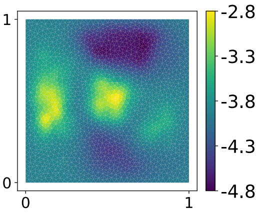
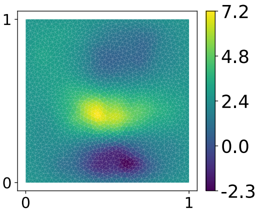
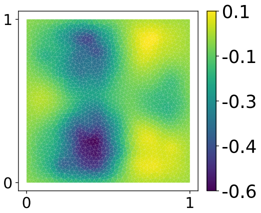
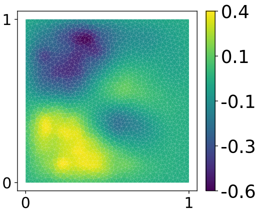
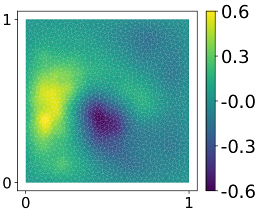
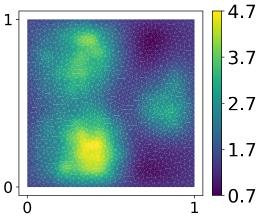
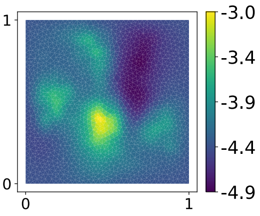
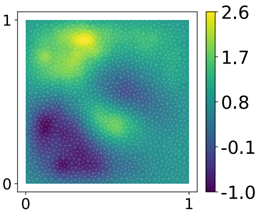
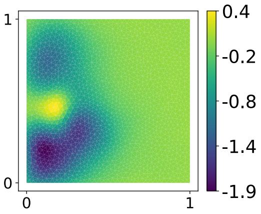
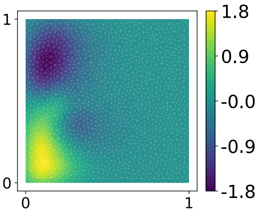
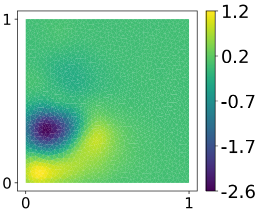
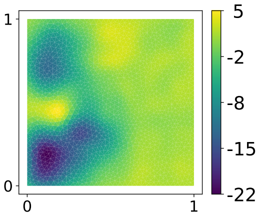
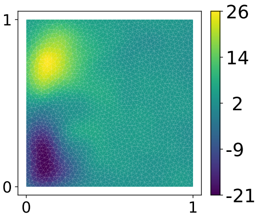
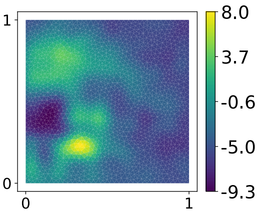
2.5 A CFD PARAMETRIC APPLICATION OF KAS USING DG METHOD

We want to test the kernel-based extension of the active subspaces in a computational fluid dynamics context. The lift and drag coefficients of a NACA 0012 airfoil are considered as model functions. Numerical simulations are carried out with different input parameters for quantities that describe the geometry and the physical conditions of the problem. The evolution of the model is protracted until a periodic regime is reached. Once the simulation data have been collected, sensitivity analysis is performed searching for an active subspace and response surfaces with GPR are then built from the application of AS and KAS techniques.

The fluid motion is modelled through the unsteady incompressible Navier-Stokes equations approximated through the Chorin-Temam operator-splitting method implemented in HopeFOAM [121]. HopeFOAM is an extension of OpenFOAM [284], an open source software for the solution of complex fluid flows problems, to variable higher order element method and it adopts a Discontinuous Galerkin Method, based on the formulation proposed by Hesthaven and Warburton [113].

The Discontinuous Galerkin method is a high-order method, which has appealing features such as the low artificial viscosity and a convergence rate which is optimal also on unstructured grids, commonly used in industrial frameworks. In addition to this, DG is naturally suited for the solution of problems described by conservative

Table 2.3: First 3 modes using Karhunen–Loève (K-L) decomposition, AS, and KAS, for the outputs defined in equation (2.42), equation (2.43), and equation (2.44).

Case	Mode 1	Mode 2	Mode 3
K-L			
AS (2.42)			
KAS (2.42)			
AS (2.43)			
KAS (2.43)			
AS (2.44)			
KAS (2.44)			

governing equations (Navier Stokes equations, Maxwell's equations and so on) and for parallel computing. All these properties are due to the fact that, differently from formulations based on standard finite elements, no continuity is imposed on the cell boundaries and neighboring elements only exchange a common flux. The major drawback of DG is its high computational cost with respect to continuous Galerkin methods, due to the need of evaluating fluxes during each time step and the presence of extra degree of freedoms in correspondence of the elemental edges.

Nowadays efforts are aimed at applying the DG in problems which involve deformable domains [293] and at improving the computational efficiency of the DG adopting techniques based on hybridization methods, matrix-free implementations, and massive parallelization [191, 201].

2.5.1 *Domain and mesh description*

The domain Ω of the fluid dynamic simulation is a two-dimensional duct with a sudden area expansion and a NACA 0012 airfoil is placed in the largest section. The inflow $\partial\Omega_I$ is placed at the beginning of the narrowest part of the duct, and here the fluid velocity, constant along all the boundary, is imposed. The outlet is placed on the right hand side and it is denoted with $\partial\Omega_O$. We refer with $\partial\Omega_W := \partial\Omega \backslash \{\partial\Omega_O \cup \partial\Omega_I\}$ to the boundaries of the airfoil and to the walls of the duct, where no slip boundary conditions are applied. The horizontal lengths of the sections of the channels are 0.6 m and 1.35 m, respectively. The vertical length of the duct after the area expansion is 0.4 m, while the width of the first one depends on two distinct parameters. The airfoil has a chord-length equal to 0.1 m but its position with respect to the duct and its angle of attack are described by geometric parameters. Further details about the geometric parameterization of the geometry are provided in the following section. A proper triangulation is introduced with the aid of the gmsh [103] tool and the domain is discretized with 4445 unstructured elements.

The evaluation of the Reynolds, the Mach number and the other adimensional magnitudes, commonly used for characterizing the fluid flow field, requires the definition of some reference magnitudes. For the problem at hand we consider the equivalent diameter of the channel in correspondence of the inlet as the reference lengthscale, while the reference velocity is the one imposed at the inlet.

2.5.2 *Parameter space description*

We chose 7 heterogeneous parameters for the model: 2 physical, and 5 geometrical which describe the width of the channel and the position of the airfoil. In table 2.4 are reported the ranges for the geometrical and physical parameters of the simulation. U is the first component of the initial velocity, ν is the kinematic viscosity, x_0 and y_0 are the horizontal and vertical components of the translation of the airfoil with respect to its reference position (see figure 2.9), α is the angle of the counterclockwise rotation and the center of rotation is located right in the middle of the airfoil, y^+ and y^- are the module of the vertical displacements of the upper and lower side of the initial conduct from a prescribed position.

Table 2.4: Parameter ranges for the NACA problem.

	ν	U	x_0	y_0	α	y^+	y^-
Lower bound	0.00036	0.5	-0.099	-0.035	0	-0.02	-0.02
Upper bound	0.00060	2	0.099	0.035	0.0698	0.02	0.02

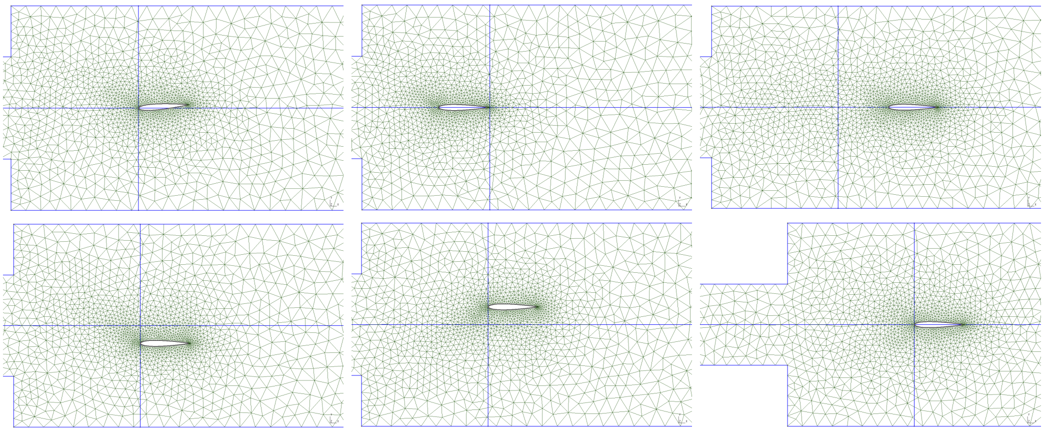


Figure 2.9: Domain configuration for minimum and maximum values of some geometric parameters. In order are represented the maximum angle of attack α, the ranges for the horizontal translation x_0, the ranges for the vertical translation y_0, and the minimum opening of the channel which depends on the parameters y^+ and y^- in table 2.4.

In figure 2.9 are reported different configurations of the domain for the minimum and maximum values of the parameters α, x_0, y_0, and the minimum opening of the channel.

We have considered only the counterclockwise rotation of the airfoil for symmetrical reasons. The range of the Reynolds number varies from 400 to 2000, still under the regime of laminar flow.

2.5.3 *Governing equations*

The CFD problem is modeled through the incompressible Navier-Stokes and the open source solver HopeFOAM has been employed for solving this set of equations [113].

Let $\Omega \subset \mathbb{R}^2$ be the two-dimensional domain introduced in section 2.5.1, and let us consider the incompressible Navier-Stokes equations. Omitting the dependence on $(\mathbf{x}, t) \in \Omega \times \mathbb{R}^+$ in the first two equations for sake of compactness, the governing equations are

$$\begin{cases} \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \nu \Delta \mathbf{u} & \mathbf{x} \in \Omega, \\ \nabla \cdot \mathbf{u} = 0 & \mathbf{x} \in \Omega, \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, \quad p(\mathbf{x}, 0) = 0 & \mathbf{x} \in \Omega, \\ \mathbf{u}(\mathbf{x}, t) = \mathbf{u}_0, \quad \mathbf{n} \cdot \nabla p(\mathbf{x}, t) = 0 & \mathbf{x} \in \partial\Omega_I, \\ \mathbf{u}(\mathbf{x}, t) = 0, \quad \mathbf{n} \cdot \nabla p(\mathbf{x}, t) = 0 & \mathbf{x} \in \partial\Omega_W, \\ \mathbf{n} \cdot \nabla \mathbf{u}(\mathbf{x}, t) = 0, \quad p(\mathbf{x}, t) = 1 & \mathbf{x} \in \partial\Omega_O, \end{cases} \tag{2.47}$$

where p is the scalar pressure field, $\mathbf{u} = (u, v)$ is the velocity field, ν is the viscosity constant and $\mathbf{u}_0$ is the initial velocity. In conservative form, the previous equations can be rewritten as

$$\begin{cases} \partial_t \mathbf{u} + \nabla \cdot \mathcal{F} = -\nabla p + \nu \Delta \mathbf{u}, \\ \nabla \cdot \mathbf{u} = 0, \end{cases} \tag{2.48}$$

with the flux $\mathcal{F}$ given by

$$\mathcal{F} = [\mathbf{F}_1, \mathbf{F}_2] = \begin{bmatrix} u^2 & uv \\ uv & v^2 \end{bmatrix}. \tag{2.49}$$

From now on, in order to have a more compact notation, the advection term is written as $\mathcal{N}(\mathbf{u}) = \nabla \cdot \mathcal{F}(\mathbf{u})$.

For each timestep the procedure is broken into three stages accordingly to the algorithm proposed by Chorin and adapted for a DG framework by Hesthaven et al. [113]: the solution of the advection dominated conservation law component, the pressure correction weak divergence-free velocity projection, and the viscosity update. The non-linear advection term is treated explicitly in time through a second order Adams-Bashforth method [96], while the diffusion term implicitly. The Chorin algorithm is reported in algorithm 6.

In order to recover the Discontinuos Galerkin formulation, the equations introduced by the Chorin method are projected onto the solution space by introducing a proper set of test functions and then the variables are approximated over each element as a linear combination of local shape functions. The DG does not impose the continuity of the solution between neighboring elements and therefore it requires the adoption of methods for the evaluation of the flux exchange between neighboring elements. In the present work the convective fluxes are treated accordingly to the Lax-Friedrichs splitting scheme, while the viscous ones are solved through the Interior Penalty method [7, 246].

The aerodynamic quantities we are interested in are the lift and drag coefficients in the incompressible case computed from the quantities $\mathbf{u}$, p, ν, A_{ref}, and $\mathbf{u}_0$ with a contour integral along the airfoil Γ as

$$f = \oint_{\Gamma} p\mathbf{n} - \nu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right) \mathbf{n} \, d\mathbf{s}. \tag{2.50}$$

Algorithm 6 Chorin Algorithm.

```
  Input:
    state variables $\mathbf{u}$ and $p$ at $t = 0$
    mesh
    boundary conditions
  Output:
    state variables $\mathbf{u}$ and $p$ at $t = t_{\text{final}}$
while $t < t_{\text{final}}$ do
    Update state variables $\mathbf{u}^{n-1} = \mathbf{u}^n$, $\mathbf{u}^n = \mathbf{u}^{n+1}$.
    Find a guess value for the velocity $\tilde{\mathbf{u}}$ by solving:
        $\frac{\gamma_0 \tilde{\mathbf{u}} - \alpha_0 \mathbf{u}^n - \alpha_1 \mathbf{u}^{n-1}}{\Delta t} = -\beta_0 \mathcal{N}(\mathbf{u}^n) - \beta_1 \mathcal{N}(\mathbf{u}^{n-1}).$
    Find the pressure at $n+1$ solving:
        $-\Delta \bar{p}^{n+1} = -\frac{\gamma_0}{\Delta t} \nabla \cdot \tilde{\mathbf{u}}.$
    Find the intermediate velocity $\tilde{\tilde{\mathbf{u}}}$ solving:
        $\gamma_0 \frac{\tilde{\tilde{\mathbf{u}}} - \tilde{\mathbf{u}}}{\Delta t} = \nabla \bar{p}^{n+1}.$
    Find the velocity at the $n+1$ time instant solving:
        $\gamma_0 \left( \frac{\mathbf{u}^{n+1} - \tilde{\tilde{\mathbf{u}}}}{\Delta t} \right) = \nu \Delta \mathbf{u}^{n+1}.$
    Update $t^n$.
end while
```

The vector $\mathbf{n}$ is the outward normal along the airfoil surface. The circulation in Γ is affected by both the pressure and stress distributions around the airfoil. The projection of the force along the horizontal and vertical directions gives the drag and lift coefficients respectively

$$C_D = \frac{f \cdot \mathbf{e}_1}{\frac{1}{2} |\mathbf{u}_0|^2 A_{\text{ref}}}, \tag{2.51}$$

$$C_L = \frac{f \cdot \mathbf{e}_2}{\frac{1}{2} |\mathbf{u}_0|^2 A_{\text{ref}}}, \tag{2.52}$$

where the reference area A_{ref} is the chord of the airfoil times a length of 1 m. For the aerodynamic analysis of the fluid flow past an airfoil see [146].

2.5.4 *Numerical results*

In this section a brief review of the procedure and some details about the numerical method and the computational domain will be presented along the results obtained.

For what concerns the DG the polynomial order chosen is 3. The total number of degrees of freedom is 133350. Small variations on the mesh are present in each of the 285 simulations due to the different configurations of the domain. Each simulation is carried out until a periodic behaviour is reached and for this reason the final times ranges between 3.5 and 5 s, depending on the specific configuration. The integration time intervals are variable and they are updated at the end of each step in order to satisfy the CFL condition. The 7 physical and geometrical parameters of the simulation are sampled uniformly from the intervals in table 2.4. In total we considered a dataset of 285 samples.

With the purpose of qualitatively visualizing the results, 4 different simulations are reported in figure 2.10 for the module of the velocity field and the scalar pressure field, respectively, both evaluated at the last time instant. These simulations were chosen from the 285 collected in order to show significant differences in the evolution of the fluid flow. In table 2.5 are reported the corresponding parameters. Depending on the position of the airfoil and the other physical parameters, different fluid flow patterns can be qualitatively observed.

Table 2.5: Parameters associated to the simulations plotted in figure 2.10.

#	ν	U	x_0	y_0	α	y^+	y^-
1	0.000405	1.99	-0.096	-0.00207	0.00282	0.00784	0.0188
2	0.000541	0.763	-0.084	0.00279	0.0260	-0.0108	0.0195
3	0.000406	0.533	-0.0503	-0.0327	0.0604	-0.0193	0.0068
4	0.000430	1.11	-0.0897	-0.0279	0.0278	-0.00624	0.0197

The lift (C_L) and drag (C_D) coefficients are evaluated when stationary or periodic regimes are reached, starting from the values of pressure and viscous stresses evaluated on the nodes close to the airfoil. After this sensitivity analysis is carried out. First the AS method is applied. The gradients necessary for the application of the AS method are obtained from the Gaussian process regression of the model functions C_L and C_D on the whole parameters domain. The eigenvalues of the uncentered covariance matrix for the lift and drag coefficients suggest the presence of a one-dimensional active subspace in both cases.

The plots of the first active eigenvector components are useful as sensitivity measures, see figure 2.11. The greater the absolute value of a component is, the greater is its influence on the model function. We observe that the lift coefficient is influenced mainly by the vertical position of the airfoil and the angle of attack, while the drag coefficient depends mainly on the initial velocity, and secondarily on the viscosity and on the angle of attack.

As one could expect from physical considerations, the angle of attack affects both drag and lift coefficients, while the viscosity, which governs the wall stresses, is relevant for the evaluation of the C_D. The vertical position of the airfoil with respect to the symmetric axis of the section of the duct after the area expansion also greatly affects both coefficients, and this is mainly due to the fact that the fluid flow conditions change drastically between the core, where the speed is higher, and the one close to the wall of the duct, where the speed tends to zero. On the other hand, the horizontal translation has almost no impact on the results, given the regularity

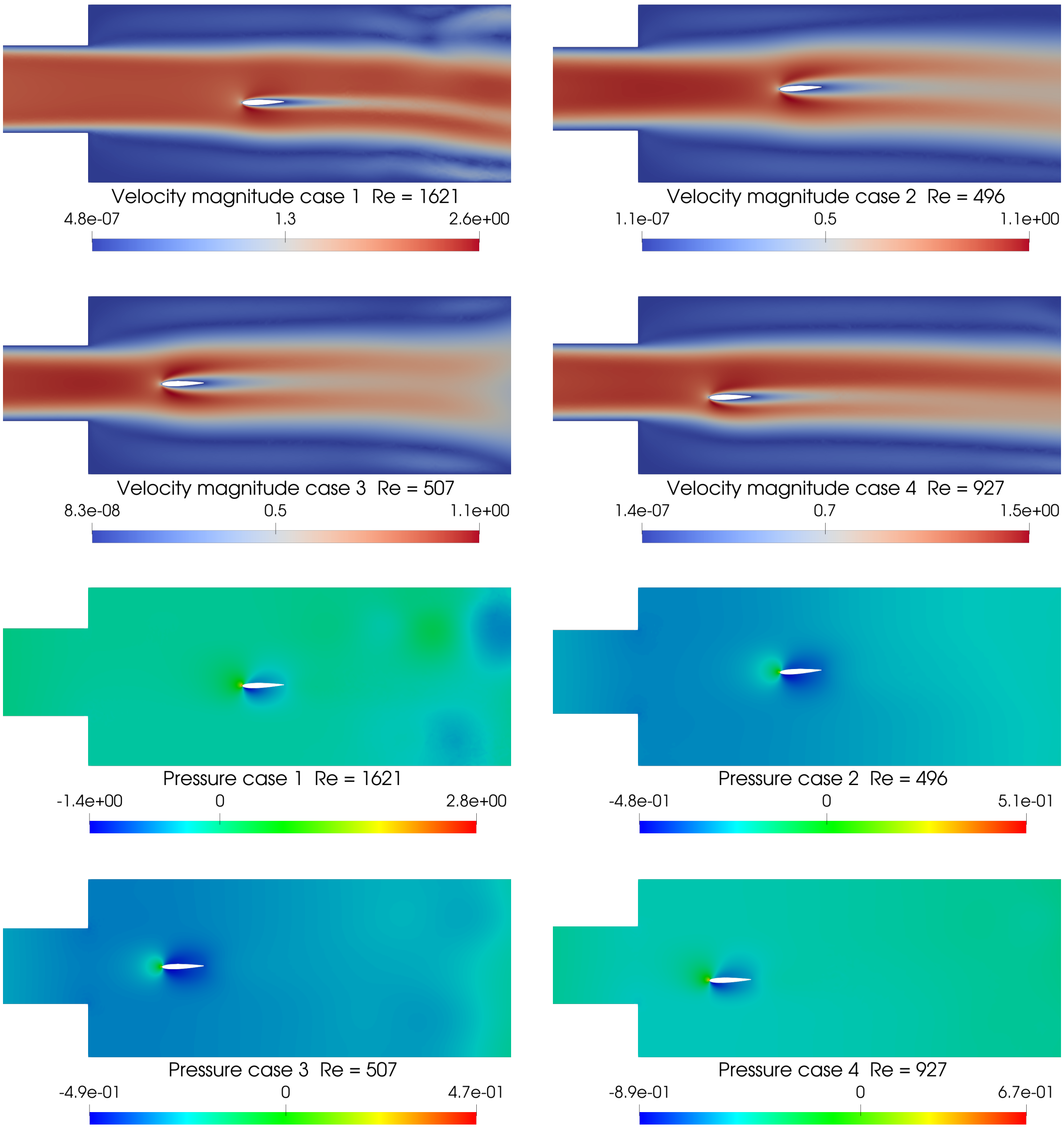


Figure 2.10: Module of the velocity fields (on the left) and pressure fields (on the right) evaluated at the last time instant of 4 different simulations. The corresponding parameters are reported in table 2.5.

of the fluid flow along the x-axis for the considered range of x_0. Moreover, the non-symmetric behaviour of the upper and lower parameters which determine the opening of the channel is due to the non-symmetric choice of the range considered for the angle of attack.

The KAS method was applied with 1500 features. In order to compare the AS and KAS methods 5-fold cross validation was implemented. The score of cross validation is the RRMSE defined in equation (2.31).

The Gaussian process regressions for the two methods are shown in figure 2.12 for the lift coefficient, and in figure 2.13 for the drag coefficient. They were obtained as a single step of 5-fold cross validation with one fifth of the 285 samples used as test set. The spectral distribution of the feature map is the Gaussian distribution

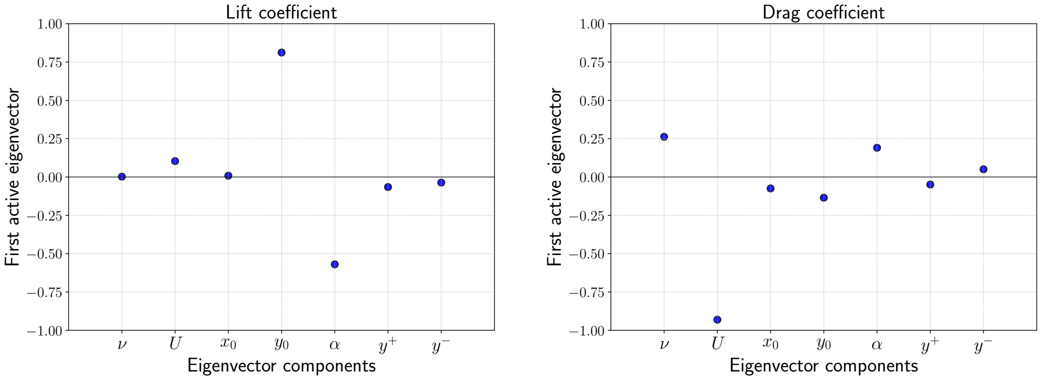


Figure 2.11: Components of the first active eigenvector for the lift coefficient (on the left), and for the drag coefficient (on the right). Values near 0 suggest little sensitivity for the target function.

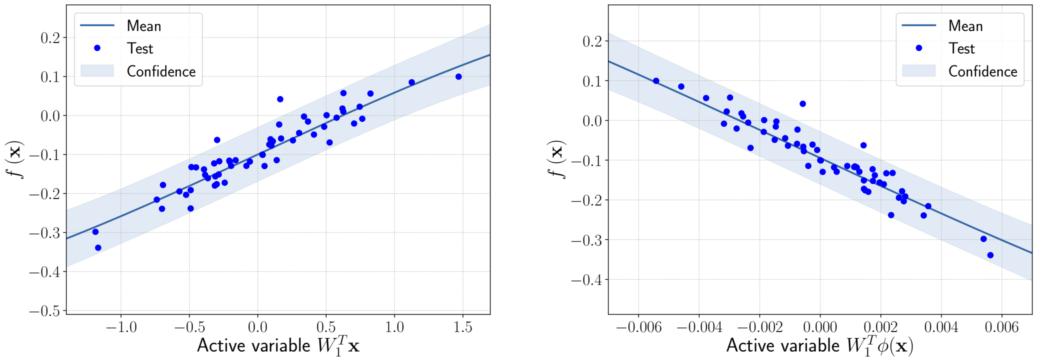


Figure 2.12: Comparison between the sufficiency summary plots obtained from the application of AS and KAS methods for the lift coefficient C_L defined in equation (2.52). The left plot refers to AS, the right plot to KAS. With the blue solid line we depict the posterior mean of the GP, with the shadow area the confidence intervals, and with the blue dots the testing points.

for the lift, and the Beta for the drag, respectively. The RRMSE mean and standard deviation from 5-fold cross validation, are reported for different active dimensions in table 2.6. The feature map in equation (2.28) was adopted. The hyperparameters of the spectral distributions were tuned with logarithmic grid-search with 5-fold cross validation as described in algorithm 5.

Regarding the drag coefficient, the relative gain using the KAS method reaches the 19.2% on average when employing the Beta spectral measure for the definition of the feature map. The relative gain of the one dimensional response surface built with GPR from the KAS method is 7% on average for the lift coefficient. This result could be due to the higher noise in the evaluation of the C_L. In this case the relative gain increases when the dimension of the response surface increases to 2 with a gain of 14.6%. A slight reduction of the AS RRMSE relative to the drag coefficient is ascertained when increasing the dimension of the response surface.

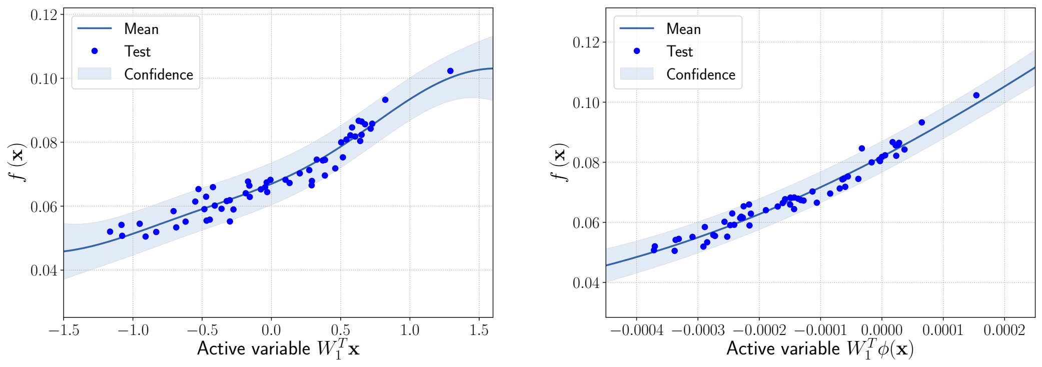


Figure 2.13: Comparison between the sufficiency summary plots obtained from the application of AS and KAS methods for the drag coefficient C_D defined in equation (2.51). The left plot refers to AS, the right plot to KAS. With the blue solid line we depict the posterior mean of the GP, with the shadow area the confidence intervals, and with the blue dots the testing points.

Table 2.6: Summary of the results for AS and KAS procedures. In bold the best results.

Method	Dim	Feature space dim	Lift spectral distribution	RRMSE Lift	Drag spectral distribution	RRMSE Drag
AS	1	-	-	0.37 ± 0.09	-	0.268 ± 0.032
KAS	1	1500	$\mathcal{N}(0, \lambda I_d)$	**0.344** ± 0.048	Beta(α, β)	**0.218** ± 0.045
AS	2	-	-	0.384 ± 0.073	-	0.183 ± 0.027
KAS	2	1500	$\mathcal{N}(0, \lambda I_d)$	**0.328** ± 0.071	Beta(α, β)	**0.17** ± 0.02

2.6 conclusions and perspectives

In this chapter we presented a new nonlinear extension of the active subspaces property called Kernel-based Active Subspaces. The method exploits random Fourier features to find active subspaces on high-dimensional feature spaces. We tested the new method over 5 different benchmarks of increasing complexity, and we provided pseudo-codes for every aspects of the proposed kernel-extension. The tested model functions range from scalar to vector-valued. We also provide a CFD application discretized by the Discontinuous Galerkin method. We compared the kernel-based active subspaces to the standard linear active subspaces and we observed in all the cases an increment of the accuracy of the Gaussian response surfaces built over the reduced parameter spaces. The most interesting results regard the possibility to apply the KAS method when an active subspace cannot be found theoretically. This was shown for radial symmetric model functions.

Future developments will involve the study of more efficient procedures for tuning the hyperparameters of the spectral distribution. Other possible advances could be done finding an effective back-mapping from the targets to the actual parameters in the full original space. This could promote the implementation of optimization algorithms or other parameter studies enhanced by the kernel-based active subspaces extension.

3

LOCAL ACTIVE SUBSPACES

Frequently, the parameter space, chosen for shape design or other applications that involve the definition of a surrogate model, present subdomains where the objective function of interest is highly regular or well behaved. So, it could be approximated more accurately if restricted to those subdomains and studied separately. The drawback of this approach is the possible scarsity of data in some applications, but in those, where a quantity of data, moderately abundant considering the parameter space dimension and the complexity of the objective function, is available, partitioned or local studies are beneficial. In this chapter we propose a new method called Local Active Subspaces (LAS), which explores the synergies of active subspaces with supervised clustering techniques in order to perform a more efficient dimension reduction in the parameter space for the design of accurate response surfaces. We also developed a procedure to exploit the local active subspace information for classification tasks. Using this technique as a preprocessing step onto the parameter space, or output space in case of vectorial outputs, brings remarkable results for the purpose of surrogate modelling. All the results presented in this chapter appeared in [223]

3.1 LITERATURE REVIEW

Parameter space reduction is a rapidly growing field of interest which plays a key role in fighting the curse of dimensionality. The need of reducing the number of design inputs is particularly important in engineering for advanced CFD simulations to model complex phenomena, especially in the broader context of model order reduction [226] and industrial numerical pipelines [227, 230, 268].

Active subspaces [54] is one of the most used techniques for linear reduction in input spaces. It has been proved useful in many numerical tasks such as regression, using a multi-fidelity data fusion approach with a surrogate model built on top of the AS as low-fidelity model [224]; shape optimization [168] and a coupling with the genetic algorithm to enhance its performance [69, 73]; or inverse problems [292]. It has also been used to enhance classical model order reduction techniques such as POD-Galerkin [266], and Proper Orthogonal Decomposition with Interpolation (PODI) [72]. Other attempts towards nonlinear parameter space reduction have been proposed recently: kernel-based active subspaces [221], nonlinear level-set learning [294], and active manifold [35] are the most promising.

In this work we propose a new local approach for parameter space dimensionality reduction for both regression and classification tasks, called Local Active Subspaces. Other methods have been developed in the last years exploiting the localization idea. We mention localized slice inverse regression (LSIR) [289] which uses local information of the slices for supervised regression and semi-supervised classification. LSIR improves local discriminant information [109] and local Fischer discriminant analysis [264] with more efficient computations for classification problems. The main difference between Slice Inverse Regression (SIR) [161] and AS is in the construction of the projection matrix. While SIR needs the elliptic assumption, AS exploits the gradients of the function of interest with respect to the input parameters.

From a wider point of view, there is an analogy between local parameter space reduction and local model order reduction. With the latter, we mean both a spatial domain decomposition approach for model order reduction of parametric PDEs in a spatial domain $\Omega \subset \mathbb{R}^d$ and a local reduction approach in the parameter space. As representative method for the first paradigm we report the reduced basis element method [167], which combines the reduced basis method in each subdomains with a mortar type method at the interfaces. For the second approach we cite the interpolation method in the Grassmann manifold of the reduced subspaces [6]; in particular in [63] the K-medoids clustering algorithm with Grassmann metric is applied to the discrete Grassmann manifold of the training snapshots as a step to perform local model order reduction.

Similarly, in our work we do not simply apply a clustering technique to preprocess the input data, we propose a supervised metric induced by the presence of a global active subspace. The directions individuated by local active subspaces are locally linear, and they better capture the latent manifold of the target function. Moreover the effectiveness of the chosen metric is linked to the fact that normally we do not employ the optimal profile (which is a conditioned random variable and in this sense nonlinear) for the construction of response surfaces on the active subspace, and with the use of clustering with AS-based metric we try to reduce this approximation error. Essentially, we are splitting the response surface design transversally with respect to the active subspace, which, having in mind that the optimal profile is obtained integrating out the inactive variable, means splitting the integral in a sum of terms associated to the number of clusters.

This chapter is organized as follows: in section 3.2 we introduce the clustering algorithms used and the supervised distance metric based on the presence of a global active subspace, focusing on the construction of response surfaces and providing theoretical results. In section 3.3 we present the algorithms to exploit LAS for classification. We provide extensive numerical results in section 3.4 from simple illustrative bidimensional dataset to high-dimensional scalar and vector-valued functions. Finally in section 3.5 we draw some conclusions and future perspectives.

3.2 LOCALIZED PARAMETER SPACE REDUCTION

Sometimes we do not have *a priori* knowledge about the target function's behaviour in a particular parameter space region. So it is difficult to properly set the parameters range. In these cases a preprocessing of the data using a clustering technique could be highly beneficial. With a clustering of the input parameters we can treat each subregion separately, and thus being able to better capture the target function's variability. This is always true for any function of interest, but for functions with global lower intrinsic dimensionality we can exploit such structure to enhance the clustering. To this end we propose a new distance metric for K-medoids and hierarchical top-down clustering methods which exploits the global active subspace of the target function.

By applying AS on each cluster we find the optimal rotation of the corresponding subregion of the input domain, which aligns the data along the active subspace of a given dimension.

In this section we review two of the main partitioning methods [108] such as K-means, K-medoids, and the hierarchical top-down clustering technique [137, 188]. We are going to use K-means as the baseline since the input parameter space is assumed to be an hyperrectangle — as is done in every practical case — and we do not expect samples which are highly concentrated in a specific subregion.

3.2.1 *K-means clustering*

Let $\{x_i\}_{i=1}^N$ be a set of N samples in $\mathbb{R}^{N_F}$, where N_F denotes the number of features. The K-means algorithm divides this set into K disjoint clusters $S = \{S_j\}_{j=1}^K$, with $S_l \cap S_m = \emptyset$ for $1 \leq l, m \leq K$ and $l \neq m$. The partitioning quality is assessed by a function which aims for high intracluster similarity and low intercluster similarity. For K-means this is done by minimizing the total within-cluster sum-of-squares criterion W_T, which reads as

$$W_T(S) := \sum_{j=1}^{K} W(S_j) = \sum_{j=1}^{K} \sum_{x_i \in S_j} \|x_i - c_j\|_{L^2}^2, \tag{3.1}$$

where c_j is the centroid describing the cluster S_j. A centroid of a cluster is defined as the mean of all the points included in that cluster. This means that the centroids are, in general, different from the samples x_i.

K-means is sensitive to outliers, since they can distort the mean value of a cluster and thus affecting the assignment of the rest of the data. Algorithm 7 shows the pseudo-code of the K-means algorithm, while in figure 3.1 there is a comparison with K-medoids on a toy dataset — a complete description is provided after the introduction of the K-medoids algorithm below.

Algorithm 7 K-means algorithm.

```
Input:
  set of samples $\{x_i\}_{i=1}^N \in \mathbb{R}^{N_F}$
  number of clusters $K$
Output:
  set of clusters $S = \{S_j\}_{j=1}^K$
select initial cluster centroids
repeat
    assign each sample to its closest centroid using the Euclidean distance
    update the centroid of each cluster
until criterion $W_T$ of equation (3.1) is minimized
```

3.2.2 *K-medoids clustering with active subspaces-based metric*

In order to overcome some limitations of the K-means algorithm, such as sensitivity to outliers, we can use K-medoids clustering technique [137, 173, 198, 239]. It uses an actual sample as cluster representative (i.e. medoid) instead of the mean of the samples within the cluster.

Following the notation introduced in the previous section, let m_j be the medoid describing the cluster S_j. The partitioning method is performed by minimizing the sum of the dissimilarities between the samples within a cluster and the corresponding medoid. To this end an absolute-error criterion E is used, which reads as

$$E(S) := \sum_{j=1}^{K} E(S_j) = \sum_{j=1}^{K} \sum_{x_i \in S_j} \|x_i - m_j\|. \tag{3.2}$$

By looking at the formula above it is clear that the use of a data point to represent each cluster's center allows the use of any distance metric for clustering. We remark that the choice of the Euclidean distance does not produce the same results as K-means because of the different references representing the clusters. In figure 3.1 we show the differences between the two methods. The position of the centroids, emphasized in red, in regions with a small density of samples — especially for cluster 1 — is due to the presence of outliers which force the centroid towards them. This results also in the different classification of the samples in purple, which fall in the cluster 1 using K-medoids and in cluster 2 using K-means.

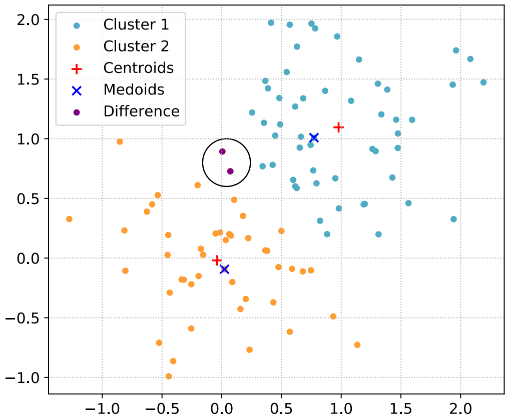


Figure 3.1: Comparison between K-means and K-medoids partitioning. The medoids (in blue) correspond to actual samples, while the centroids (in red) are, in general, not an element of the dataset. In purple two samples which are classified in different clusters depending on the method used.

We propose a new supervised distance metric inspired by the global active subspace of the function f we want to approximate. We define a scaled L^2 norm using the eigenpairs of the second moment matrix of ∇f, which is the matrix from which we calculate the global active subspace:

$$\|x_i - x_j\|_\Lambda = \sqrt{(x_i - x_j)^T \mathbf{W} \Lambda^2 \mathbf{W}^T (x_i - x_j)}, \tag{3.3}$$

where Λ stands for the diagonal matrix with entries the eigenvalues of equation (2.8), and $\mathbf{W}$ is the eigenvectors matrix from from its eigendecomposition. As we are going to show in section 3.4 this new metric allows a better partitioning both for regression and classification tasks because of it exploits both global and local informations. For stronger insights about the heuristic behind, see remark 17.

To actually find the medoids, the partitioning around medoids (PAM) algorithm [137] is used. It uses a greedy approach after the initial selection of the medoids, also called representative objects. They are changed with a non-representative object, i.e. one of the remaining samples, if it would improve the clustering quality. This iterative process of replacing the medoids by other objects continues until the quality of the resulting clustering cannot be improved by any replacement. Algorithm 8 illustrates this approach with pseudo-code.

3.2.3 *Hierarchical top-down clustering*

In this section we present Hierarchical top-down clustering with Active Subspaces (HAS), a variant of the previous methodologies with hierarchical top-down clustering [137, 188], that exploits, as previously, the additional information from the active subspace.

Algorithm 8 K-medoids algorithm with AS metric.

```
  Input:
    set of samples $\{x_i\}_{i=1}^{N} \in \mathbb{R}^{N_F}$
    number of clusters $K$
    distance metric $d$ defined in equation (3.3)
  Output:
    set of clusters $S = \{S_j\}_{j=1}^{K}$
select initial cluster medoids
repeat
    assign each sample to its closest medoid using the distance metric $d$
    randomly select $K$ non-representative objects
    swap the medoids with the new selected objects by minimizing equation (3.2)
until clustering quality converges
```

In top-down hierarchical clustering, at each iteration the considered clusters, starting from the whole dataset, are split further and further based on some refinement criterion, until convergence. A nice feature of hierarchical clustering algorithms, with respect to K-means and K-medoids, is that the number of clusters can be omitted. Moreover, stopping at the first refinement and forcing the total number of clusters, HAS can be seen as a generalization of the previous methods: for this reason we wanted to make the implementation consistent with K-means and K-medoids with AS induced metric as close as possible, as shown in the numerical results in section 3.4.

Pushing further the potential of clustering algorithms applied to local dimension reduction in the parameter space, HAS is a versatile clustering method that takes into account the variability of the AS dimension along the parameter space. The price paid for this is the overhead represented by the tuning of some hyper-parameters introduced later in section 3.2.3.2.

3.2.3.1 *Ridge approximation with hierarchical clustering and active subspaces*

Before presenting the algorithm and the numerical results, some straightforward theoretical considerations linked to the theory of ridge approximation with active subspaces [54, 291] are due. We will consider scalar outputs f but the following statements can be extended to vector-valued outputs.

Definition 1 (Local ridge approximation with active subspaces). *Given a partition of the domain $\mathcal{P} := \{S_i\}_{i\in\{1,\dots,d\}}$ and a map $r : \mathcal{P} \to \{1, \dots, n_r\}$, $n_r \ll n$ representing the local reduced dimension, the local ridge approximation with active subspaces of $(f, \boldsymbol{\mu})$ is the function $R_{AS}(r, f, \boldsymbol{\mu}) : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$ that is defined locally for every $S_i \in \mathcal{P}$ as*

$$g|_{S_i} = \mathbb{E}_{\boldsymbol{\mu}_i}[f|P_{r(S_i),i}] \circ P_{r(S_i),i} \tag{3.4}$$

where $\boldsymbol{\mu}_i := (1/\boldsymbol{\mu}(S_i)) \cdot \boldsymbol{\mu}|_{S_i} \mathbb{R}^n$, and $P_{r,i} : S_i \subset \mathbb{R}^n \to$ is the orthogonal projector with rank r that satisfies the minimization problem

$$P_{r,i} = \underset{\substack{P^2=P, P=P^T,\\ rank(P)=r}}{\operatorname{argmin}} \mathbb{E}_{\boldsymbol{\mu}_i} \|(Id - P)\nabla f\|^2 = \underset{\substack{P^2=P, P=P^T,\\ rank(P)=r}}{\operatorname{argmin}} tr((Id - P)\mathbb{E}_{\boldsymbol{\mu}_i}\left[\nabla f \otimes \nabla f\right](Id - P)).$$

The above minimization problem is an adaptation from Proposition 2.6 of [291]. With this definition we can state the problem of local ridge approximation with active subspaces.

Problem 2 ($(\mathcal{P}, r)$ minimizers of ridge approximation error). *Find the partition $\mathcal{P}$ of the domain $\mathcal{X} \subset \mathbb{R}^n$ and the local reduced dimension map $r : \mathcal{P} \to \{1, \dots, n_r\}$, $n_r \ll n$, such that the L^2-error between the objective function f and its local ridge approximation with active subspaces is minimized. Assuming that the subspace Poincaré inequality [197] is valid also for $(f, \boldsymbol{\mu})$ restricted to the elements of the partition $\mathcal{P}$, a straightforward bound is obtained applying the Poincaré inequality for every element of the partition*

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\mu}}\left[\|f - R_{AS}(r, f)\|^2\right] &= \sum_{S_i \in \mathcal{P}} \mathbb{E}_{\boldsymbol{\mu}}\left[\| f|_{S_i} - \mathbb{E}_{\boldsymbol{\mu}_i}[f|P_{r(S_i),S_i}] \circ P_{r(S_i),S_i}\|^2\right] \\ &\lesssim \sum_{S_i \in \mathcal{P}} \mathbb{E}_{\boldsymbol{\mu}}\left[\|(Id - P_{r(S_i),S_i})^T \nabla f\|^2\right].\end{aligned}$$

To state problem 2 we made an assumption about the Poincaré subspace inequality that in general is not satisfied by any probability measure $\boldsymbol{\mu}$ chosen.

Remark 11 (Subspace Poincaré inequality). *The probabilistic Poincaré inequality for conditional probability densities or subspace Poincaré inequality [197] is valid at least for the following classes of absolutely continuous probability densities $\boldsymbol{\mu}$ with p.d.f. ρ.*

Assumption 3. *The p.d.f $\rho : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$ satisfies one of the following:*

1. *$\mathcal{X}$ is bounded connected open with Lipschitz boundary, ρ is the uniform density distribution.*
2. *$\mathcal{X}$ is convex and bounded, $\exists \delta, D > 0 : 0 < \delta \leq \|\rho(\mathbf{x})\|_{L^\infty} \leq D < \infty, \forall \mathbf{x} \in \mathcal{X}$,*
3. *$\mathcal{X} = \mathbb{R}^n$, $\rho(\mathbf{x}) \sim \exp(-V(\mathbf{x}))$ where $V : \mathbb{R}^n \to (-\infty, \infty], V \in \mathcal{C}^2$ is α-uniformly convex,*

   $$\mathbf{u}^T Hess(V(\mathbf{x}))\mathbf{u} \geq \alpha \|\mathbf{u}\|_2^2, \quad \forall \mathbf{x}, \mathbf{u} \in \mathbb{R}^n, \tag{3.5}$$

   where $Hess(V(\mathbf{x}))$ is the Hessian of $V(\mathbf{x})$.
4. *$\mathcal{X} = \mathbb{R}^n$, $\rho(\mathbf{x}) \sim \exp(-V(\mathbf{x}))$ where V is a convex function. In this case we require also f Lipschitz continuous.*

The last class of p.d.f. provides a weaker bound (Lemma 4.3, [197]) on the ridge approximation error. For the previous classes $i \in \{1, 2, 3, 4\}$ of p.d.f. an upper bound of the Poincaré constant $C_{P,i}$ is also provided:

$$C_{P,1} = C_{P,1}(\Omega), \qquad C_{P,2} = \frac{D diam(\mathcal{X})}{\pi \delta}, \qquad C_{P,3} = \frac{1}{\alpha}, \tag{3.6}$$

while the upper bound for $C_{P,4}$ requires the definition of other quantities and is proved in Lemma 4.4 [197]. These results are useful to asses what properties the elements of the partition $\mathcal{P} = \{S_i\}_{i \in \{1,\dots,d\}}$ should satisfy in order for the subspace Poincaré inequality to be valid and thus for the bound in remark 12 to be applied recursively with respect to the refinement levels. In the numerical experiments we are going to present, the partition $\mathcal{P} = \{S_i\}_{i \in \{1,\dots,d\}}$ is defined by the decision boundaries of the clustering algorithm chosen.

For the moment we will consider the local reduced dimension map r constant and in general the codomain of r is a subset of $\{1, \dots, n_r\}$, $n_r \ll n$.

The previous bound suggests that a good indicator for refinement could be represented by the residual eigenvalues of the local correlation matrix. We also have the following immediate result that hints to indefinitely many successive refinements to lower the L^2-error.

Remark 12 (Relationships between the upper bounds of consecutive refinements)**.** *Considering the sum over the number of refined clusters $cl \in \{1, \dots, d\}$ we have that*

$$\begin{aligned}\int_{\mathcal{X}} \|(Id - P_r^T)\nabla f\|^2 \, d\boldsymbol{\mu} &= \sum_{cl=1}^{d} \int_{S_{cl}\subset\mathcal{X}} \|(Id - P_r^T)\nabla f\|^2 \, d\boldsymbol{\mu} \\ &\geq \sum_{cl=1}^{d} \int_{S_{cl}\subset\mathcal{X}} \|(Id - P_{r,cl}^T)\nabla f\|^2 \, d\boldsymbol{\mu},\end{aligned} \tag{3.7}$$

since the projectors $\{P_{r,cl}\}_{cl\in\{1,\dots,d\}}$ are the minimizers of

$$P_{r,cl} = \underset{\substack{P^2=P, P=P^T,\\ rank(P)=r}}{\operatorname{argmin}} \int_{S_{cl}\subset\mathcal{X}} \|(Id - P^T)\nabla f\|^2 \, d\boldsymbol{\mu}. \tag{3.8}$$

The RHS of equation (3.7) can be used as indicator for refinement. We remark that since the refinements increase the weight decay of the eigenvalues in the RHS of equation (3.7), the choice of the dimension of the active subspace may be shifted towards lower values, as we are going to show in the numerical experiments, in section 3.4.

Unfortunately, the minimizers of the ridge approximation error and of the upper bound are not generally the same. There is a counterexample in [291]. We start from this counterexample to show that in general the L^2-error of the local ridge approximation does not decrease between consequent refinements, even if the indicator from the RHS of equation (3.7) does, as stated in the previous remark.

Corollary 1 (Counterexample for indefinite refinement as optimal clustering criterion)**.** *Let $\mathcal{P} = \{A, B, C\}$ be a partition of $\mathcal{X} = [-1,1]^2$ such that $A = [-1, \epsilon] \times [-1, 1]$, $B = [-\epsilon, \epsilon] \times [-1, 1]$, and $C = [\epsilon, 1] \times [-1, 1]$. Let $\boldsymbol{\mu}$ be the uniform probability distribution on $\mathcal{X}$. The objective function we want to approximate is*

$$f : \mathcal{X} \subset \mathbb{R}^2 \to \mathbb{R}, \quad f = \begin{cases} x_1 + \epsilon, & \mathbf{x} \in A, \\ x_1(x_1+\epsilon)(x_1-\epsilon)\cos(\omega x_2), & \mathbf{x} \in B, \\ x_1 - \epsilon, & \mathbf{x} \in C, \end{cases} \tag{3.9}$$

with local reduced dimension map $r(A) = r(B) = r(C) = 1$. There exist $\epsilon > 0, \omega > 0$, such that

$$\mathbb{E}_{\boldsymbol{\mu}}\left[\|f - R_{AS}(r, f, \boldsymbol{\mu})\|^2\right] \geq \mathbb{E}_{\boldsymbol{\mu}}\left[\|f - \mathbb{E}_{\boldsymbol{\mu}}\left[f | P_{1,\mathcal{X}}\right] \circ P_{1,\mathcal{X}}\|^2\right],$$

where $P_{1,\mathcal{X}}$ is the optimal projector on the whole domain $\mathcal{X}$ with one-dimensional active subspace.

Proof. Let us use the notation $h_1(x_1) := x_1(x_1+\epsilon)(x_1-\epsilon)$, and $h_2(x_2) := \cos(\omega x_2)$, it can be shown that

$$\begin{aligned}\mathbb{E}_{\mu}\left[\nabla f \otimes \nabla f\right] &= \int_B \begin{pmatrix} (h_1')^2(h_2)^2 & h_1 h_1' h_2 h_2' \\ h_1 h_1' h_2 h_2' & (h_1)^2(h_2')^2 \end{pmatrix} d\mu(\mathbf{x}) + \mu(A\cup C)\cdot \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} \frac{2}{5}\epsilon^5\left(1+\frac{\sin(2\omega)}{2\omega}\right) & 0 \\ 0 & \frac{4}{105}\omega^2\epsilon^7\left(1-\frac{\cos(2\omega)}{2\omega}\right) \end{pmatrix} + \\ &\quad + \mu(A\cup C)\cdot \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},\end{aligned}$$

thus, since we are considering a one dimensional active subspace, the active eigenvector belongs to the set $\{(1,0),(0,1)\}$. Similarly we evaluate

$$\begin{aligned}\mathbb{E}_{\mu_B}\left[\nabla\, f|_B \otimes \nabla\, f|_B\right] &= \begin{pmatrix} \frac{8}{5}\epsilon^4\left(1+\frac{\sin(2\omega)}{2\omega}\right) & 0 \\ 0 & \frac{16}{105}\omega^2\epsilon^6\left(1-\frac{\cos(2\omega)}{2\omega}\right) \end{pmatrix}, \\ \mathbb{E}_{\mu_A}\left[\nabla\, f|_A \otimes \nabla\, f|_A\right] = \mathbb{E}_{\mu_C}\left[\nabla\, f|_C \otimes \nabla\, f|_C\right] &= \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},\end{aligned}$$

and conclude that there exist $\epsilon > 0, \omega > 0$ such that:

$$\frac{2}{5}\epsilon^5\left(1+\frac{\sin(2\omega)}{2\omega}\right) + 4(1-\epsilon) \geq \frac{4}{105}\omega^2\epsilon^7\left(1-\frac{\cos(2\omega)}{2\omega}\right), \tag{3.10}$$

$$\frac{8}{5}\epsilon^4\left(1+\frac{\sin(2\omega)}{2\omega}\right) \leq \frac{16}{105}\omega^2\epsilon^6\left(1-\frac{\cos(2\omega)}{2\omega}\right), \tag{3.11}$$

for example $\epsilon \sim 10^{-2}$, $\omega \sim 10^4$ (approximately $10\epsilon^{-2} \leq \omega^2 \leq 10\epsilon^{-7}$). In this way, using the notations of definition 1, we have

$$P_{1,\mathcal{X}} = e_1 \otimes e_1, \quad P_{1,A} = P_{1,C} = e_1 \otimes e_1, \quad P_{1,B} = e_2 \otimes e_2,$$

and it follows that

$$\mathbb{E}_{\mu}\left[\|f - R_{AS}(r,f)\|^2\right] = \mathbb{E}_{\mu}\left[f^2|_B\right] = (1/\mu(\mathcal{X}))\|h_1\|^2_{L^2(\mathcal{X},\lambda)}\|h_2\|^2_{L^2(\mathcal{X},\lambda)},$$

$$\begin{aligned}\mathbb{E}_{\mu}\left[\|f - \mathbb{E}_{\mu}\left[f|P_r\right]\circ P_r\|^2\right] &= \frac{1}{\mu(\mathcal{X})}\|h_1\|^2_{L^2(\mathcal{X},\lambda)}\left\|h_2 - \frac{1}{\mu(\mathcal{X})}\int h_2 dx_2\right\|^2_{L^2(\mathcal{X},\lambda)} \\ &= \frac{1}{\mu(\mathcal{X})}\|h_1\|^2_{L^2(\mathcal{X},\lambda)}\left(\|h_2\|^2_{L^2(\mathcal{X},\lambda)} - \frac{7}{16}\left(\int h_2 dx_2\right)^2\right),\end{aligned}$$

where λ is the Lebesgue measure. □

The heuristic behind the previous proof rests on the fact that ridge approximation with active subspaces performs poorly when the objective function has a high variation. The counterexample is valid whenever the global projector $P_{1,\mathcal{X}}$ is the minimizer of a local L^2 ridge approximation error for which the minimizer of the gradient-based indicator in equation (3.7) does not coincide. This leaves us with an indicator in equation (3.7) that does not guarantee a non increasing L^2-error decay for subsequent refinements.

We conclude the section with some remarks about the response surface design through the ridge approximation with active subspaces.

Remark 13 (Approximation of the optimal profile). *In practice we do not consider the optimal profile* $h(y) = \mathbb{E}_\mu [f|\sigma(P_r)](y)$ *but we employ the approximation* $h(y) = f(P_r\mathbf{x})$. *The reason lies on the fact that to approximate the optimal profile at the value* $\{y_i\}_i$, *additional samples from the conditional distribution* $p(z|y_i = P_r\mathbf{x})$ *must be obtained; even if the accuracy of the ridge approximation could benefit from it, this is not always possible in practice because of the difficulty to sample from the conditional distribution or because of computational budget constraints.*

If the data is split in training, validation, and test set, the local R^2 score on the validation set can be used as indicator for refinement.

Remark 14 (Estimator based on local R^2 scores). *The* R^2 *score of a single cluster can be written with respect to the* R^2 *scores* $\{R_l^2\}_{l\in\{1,\dots,d\}}$ *relative to the clusters of the subsequent refinement. Let the sum be over the refinement clusters* $l \in \{1,\dots,d\}$, *we have*

$$\begin{aligned} R^2 &= 1 - \frac{\mathbb{E}[\|f - \mathbb{E}[f|P_r] \circ P_r\|^2]}{Var(f)} \\ &= 1 - \sum_{l=1}^{d} \frac{\mathbb{E}[\| f|_{S_l} - \mathbb{E}[f|P_{r,l}] \circ P_{r,l}\|^2]}{Var(f)} \\ &= 1 - \sum_{l=1}^{d} \frac{Var(f|_{S_l})}{Var(f)} \cdot \frac{\mathbb{E}[\| f|_{S_l} - \mathbb{E}[f|P_{r,l}] \circ P_{r,l}\|^2]}{Var(f|_{S_l})} \\ &= 1 - \sum_{l=1}^{d} \frac{Var(f|_{S_l})}{Var(f)} \cdot (1 - R_l^2), \end{aligned} \tag{3.12}$$

which, substituting with the empirical variance, becomes

$$R_{emp}^2 = 1 - \sum_{l=1}^{d} \frac{Var_{emp}(f|_{S_l})}{Var_{emp}(f)} \cdot (1 - R_{emp;l}^2) \cdot \frac{N_l - 1}{N - 1}, \tag{3.13}$$

where $R_{emp;l}^2$ *is the empirical local* R^2 *score relative to cluster number* l. *The definition can be extended for component-wise vector-valued objective functions* f. *The numerical results shown in section 3.4 consider the mean* R^2 *score along the components when the output is vectorial.*

In practice every mean is approximated with simple Monte Carlo, and without the number of training samples increasing, the confidence on the approximation is lower and lower, the more the domain is refined. This is taken into consideration while clustering: the procedure balances the error from the poor Monte Carlo approximation with the generally (see corollary 1) higher accuracy of the subsequent refinement.

The following remark clarifies the link between the number of Monte Carlo samples, the numerical method chosen for the discretization of the integral $\mathbb{E}_\mu [\nabla f \otimes \nabla f]$, and the approximation of the active subspace. For example for deterministic models, one could employ the more efficient Sobol sequences or a Latin hypercube sampling; if f is more regular and the parameter space dimension is not too high one could employ tensor product Gauss quadrature rule. See for example [265].

Remark 15 (Generalization of the upper bound on the approximation of the active subspace)**.** *We want to make some brief considerations about the accuracy of the active subspace as eigen subspace of the correlation matrix approximated with Monte Carlo. If we use the notation* $W_1 \in \mathbb{R}^{n\times r}, W_2 \in \mathbb{R}^{n\times(n-r)}$ *for the active and inactive subspaces (i.e.* $P_r = W_1W_1^T$, $Id - P_r = W_2W_2^T$*) and* $\hat{W}_1 \in \mathbb{R}^{n\times r}, \hat{W}_2 \in \mathbb{R}^{n\times(n-r)}$ *for the approximated active and inactive subspaces), we can bound the approximation error as done by Constantine in [54]: assuming f Lipschitz continuous, with high probability the following inequality is valid,*

$$dist(Im(W_1), Im(\hat{W}_1)) \lesssim \frac{4L\sqrt{n}(\log(n))^{\frac{1}{2}}}{N^{\frac{1}{2}}\lambda_1(\lambda_r - \lambda_{r+1})}, \tag{3.14}$$

where L is the Lipschitz constant of f, $\{\lambda_1, \dots \lambda_n\}$ are the non-negative eigenvalues of $\mathbb{E}_{\mu_i}[\nabla f \otimes \nabla f]$ *ordered decreasingly, and N is the number of Monte Carlo samples.*

The bound in equation (3.14) is obtained from Corollary 3.8 and Corollary 3.10 in [54]. It is founded on a matrix Bernstein inequality for a sequence of random uniformly bounded matrices (Theorem 6.1, [276]) and on the Corollary 8.1.11 from [104] that holds a bound on the sensitivity of perturbation of an invariant subspace. We report this last result for completeness, stated as Lemma 3.9 from [54]:

Theorem 2 (Theorem 8.1.11 from [104], Lemma 3.9 from [54])**.** *Let C and $\hat{C} = C + E$ be symmetric $n \times n$ matrices with respective eigenvalues $\lambda_1, \dots, \lambda_n$ and $\hat{\lambda}_1, \dots, \hat{\lambda}_n$ and eigenvector matrices*

$$W = [W_1\, W_2], \qquad \hat{W} = [\hat{W}_1\, \hat{W}_2], \tag{3.15}$$

as defined previously. If $\lambda_r > \lambda_r + 1$ and

$$\|E\|_F \leq \frac{\lambda_r - \lambda_{r+1}}{5}, \tag{3.16}$$

then

$$dist(Im(W_1), Im(\hat{W}_1)) \leq \frac{4\|W_2^T E W_1\|_F}{\lambda_r - \lambda_{r+1}} \leq \frac{4\|E\|_F}{\lambda_r - \lambda_{r+1}}. \tag{3.17}$$

From this result, a bound on the approximation error of the active subspace W_1 can be obtained expliciting $\|W_2^T E W_1\|_F$ with respect to the chosen numerical method for the discretization $\hat{C}$ of the integral $C = \mathbb{E}_{\mu_i}[\nabla f \otimes \nabla f]$: in [54] this has been done for the Monte Carlo method. In practice we could use quasi Monte Carlo sampling methods with Halton or Sobol sequences [265], since

$$\begin{aligned}
\|W_2^T E W_1\|_F &\leq \sqrt{r(n-r)}\|W_2^T E W_1\|_{max} \\
&\lesssim \sqrt{r(n-r)}D^*(\{x_i\}_i) \cdot max_{i,j\in\{1,\dots,n\}}(V^{HK}(\nabla f_i \nabla f_j)) \\
&\lesssim 2\sqrt{r(n-r)}D^*(\{x_i\}_i) \cdot max(|f|) \cdot max_{i\in\{1,\dots,n\}}(V^{HK}(\nabla f_i)) \\
&\lesssim 2\sqrt{r(n-r)} \cdot max_{i\in\{1,\dots,n\}}(V^{HK}(\nabla f_i))\frac{\log(N)^n}{N},
\end{aligned}$$

where V^{HK} is the Hardy–Krause variation and $D^*(\{x_i\}_i)$ is the star discrepancy of the quasi random sequence $\{x_i\}_i$. For the above result we have imposed $\mathcal{X} = [0,1]^n$ but it can be extended to different domains [16]. Thus we obtain the bound

$$\begin{aligned} dist(Im(W_1), Im(\hat{W}_1)) &\lesssim \frac{4\|W_2^T E W_1\|_F}{\lambda_r - \lambda_{r+1}} \\ &\lesssim \frac{8L\sqrt{r(n-r)} \cdot max_{i\in\{1,\dots,n\}}(V^{HK}(\nabla f_i))}{\lambda_r - \lambda_{r+1}} \cdot \frac{\log(N)^n}{N}. \end{aligned} \tag{3.18}$$

Other numerical integration rules can be chosen so that different regularity conditions on the objective function may appear on the upper bound of the error, as the Lipschitz constant on equation (3.14) or the Hardy–Krause variation on equation (3.18). For example, for tensor product quadrature formulae [265], if the regularity of f is C^s we have generally,

$$dist(Im(W_1), Im(\hat{W}_1)) \lesssim \frac{4\|W_2^T E W_1\|_F}{\lambda_r - \lambda_{r+1}} \lesssim \frac{4\sqrt{r(n-r)}}{N^{s/n}(\lambda_r - \lambda_{r+1})}, \tag{3.19}$$

or for Smolyak's sparse quadrature rule [265],

$$dist(Im(W_1), Im(\hat{W}_1)) \lesssim \frac{4\|W_2^T E W_1\|_F}{\lambda_r - \lambda_{r+1}} \lesssim \frac{4\sqrt{r(n-r)}\,(\log N)^{(n-1)(r+1)}}{N^s(\lambda_r - \lambda_{r+1})}. \tag{3.20}$$

For high-dimensional datasets and f less regular, the estimate 3.14 is the sharpest.

3.2.3.2 *Implementation of hierarchical clustering with active subspaces*

A schematic representation of the procedure of top-down clustering is reported in algorithm 9. The design is straightforward and it employs a tree data structure that assigns at each node a possible clustering of the whole dataset: consequent refinements are represented by children nodes down until the leaves of the tree, that represent the final clusters.

Remark 16 (Normalization of the clusters at each refinement iteration)**.** *Each cluster, at every refinement step, is normalized uniformly along dimensions onto the hyper-cube domain $[-1,1]^n$. Another possible choice for normalization is standardization, centering the samples with their mean and dividing them by their standard deviation.*

The procedure depends on many parameters that have to be tuned for the specific case or depend *a priori* on the application considered: the maximum number of clusters, the minimum and maximum number of children nodes, the tolerance for the score on the whole domain, the minimun and maximum dimension of the active subspace, and the minimum number of elements (el) of each cluster (usually $el > r$, where r is the local AS dimension).

More importantly the method is versatile for the choice of clustering criterion, indicator for refinement, and regression method. In the following sections we will consider K-means and K-medoids with the active subspaces distance as clustering criterion (see section 3.2.2), but other clustering algorithms could in principle be applied at each refinement.

Algorithm 9 Hierarchical top-down algorithm.

```
Input:
  set of samples $\mathbf{S} = \{x_i\}_{i=1}^{N} \in \mathbb{R}^{N_F}$
  number of clusters per tree refinement level $\mathbf{K}$
  range of number of children $\{\mathbf{n}_{\mathbf{min}}^{\mathbf{child}}, \mathbf{n}_{\mathbf{max}}^{\mathbf{child}}\}$
  minimum number of elements in a cluster $\mathbf{n_{el}}$
  indicator for refinement $\mathbf{I}$
  distance metric $\mathbf{d}$
  minimum and maximum dimension of the active subspace $\mathbf{r_{min}}, \mathbf{r_{max}}$
  score tolerance $\boldsymbol{\epsilon}$
Output:
  refinement tree $T$
add the initial cluster $S$ to FIFO queue $q = \{S\}$
while $q \neq \varnothing$ do
    take $\mathbf{S}_j$, the first element from queue $q$
    apply the refinement function in algorithm 10 to $\mathbf{S}_j$ to get the outputs $\{S_i\}_i$
    add $\{\mathbf{S}_i\}_i$ to the queue $q$
    if the score tolerance $\epsilon$ is reached or other constraints are violated then
        break
    end if
end while
```

Remark 17 (Heuristic behind the choice of the active subspaces metric for K-medoids)**.** *Having in mind that the optimal profile $h(y) = \mathbb{E}_{\mu_i}[f|P_{r(S_i),i}]$ from definition 1 is approximated as $h(y) = f(y) = f(P_r\mathbf{x})$ as reported in remark 13, we can argue that clustering with the AS metric from equation (3.3) is effective since, for this choice of the metric, the clusters tend to form transversally with respect to the active subspace directions. This is because the metric weights more the components with higher eigenvalues. So clustering with this metric reduces heuristically also the approximation error induced by the choice of the non-optimal profile.*

Other clustering criterions employed must satisfy the subspace Poincaré inequality for each cluster. Regarding the regression method we employ Gaussian process regression with RBF-ARD kernel [287]. The procedure for response surface design with Gaussian processes and ridge approximation with active subspaces can be found in [54, 221]. As for the indicator for refinement, the local R^2 score in remark 14 is employed to measure the accuracy of the ridge approximation against a validation dataset and the estimator from the RHS of equation (3.7) is used to determine the dimension of the active subspace of each cluster.

Regarding the complexity of the algorithm, for each refinement, considering an intermediate cluster of K elements, the most expensive tasks are the active subspace evaluation $O(Knp^2 + Kn^2p + n^3)$ (the first two costs refer to matrix multiplications, while the third for eigendecomposition), the clustering algorithm, for example K-medoids with AS distance $O(K(K-m)^2)$, and the Gaussian process regression $O(K^3p^3)$, where p is the dimension of the outputs and m is the minimum number of children clusters. With M we denote the maximum number of children clusters. At

Algorithm 10 Refinement function.

```
Input:
  cluster $\mathbf{S} = \{x_i\}_{i=1}^{N} \in \mathbb{R}^{N_F}$
  number of clusters per tree refinement level $\mathbf{K}$
  range of number of children $\{\mathbf{n}^{\mathbf{child}}_{\mathbf{min}}, \mathbf{n}^{\mathbf{child}}_{\mathbf{max}}\}$
  minmum number of elements in a cluster $\mathbf{n}_{\mathbf{el}}$
  indicator for refinement $\mathbf{I}$
  distance metric $\mathbf{d}$
  minimum and maximum dimension of the active subspace $\mathbf{r}_{\mathbf{min}}, \mathbf{r}_{\mathbf{max}}$
Output:
  $\{S_j\}_{j=1}^{\mathbf{n}^{\mathbf{child}}}$, the children of cluster $\mathbf{S}$
set best score to $b = 0$
for each $\mathbf{n}^{\mathbf{child}}$ from $\mathbf{n}^{\mathbf{child}}_{\mathbf{min}}$ to $\mathbf{n}^{\mathbf{child}}_{\mathbf{max}}$ do
    apply the chosen clustering algorithm (e.g. K-medoids) with $\mathbf{n}^{\mathbf{child}}$ clusters and metric $\mathbf{d}$ to obtain the clusters $\{S_j\}_j^{\mathbf{n}^{\mathbf{child}}}$
    evaluate the estimator of the error $\mathbf{I}$ for the refinement $\{S_j\}_j$, taking also into account the minimum and maximum reduced dimensions $\mathbf{r}_{\mathbf{min}}, \mathbf{r}_{\mathbf{max}}$
    if $\mathbf{I} > b$ and the minimum number of elements $\mathbf{n}_{\mathbf{el}}$ is not reached and the maximum number of clusters $\mathbf{K}$ is not reached then
        save the best refinement $\{S_j\}_j$ and update the best score $b$
    end if
end for
```

each refinement level, the most costly operation is the Gaussian process regression. In the worst case the height of the refinement tree is $l = \log_m N/el$ where el is the minimum number of elements per cluster. From table 3.1, assuming $p = 1$, it can be noticed that at the beginning the most expensive operation is the clustering algorithm (especially when $k \ll N$, in general $N > n$), while near the final refinement the most expensive operations are the GPR and the evaluation of the AS when m is moderately greater than the minimum number of elements el (in general it can happen to have $m > el > r$). In fact, at the i-th refinement level the AS and GPR are computed $(M - m)m^i$ times, while the clustering is performed m^i times.

3.3 CLASSIFICATION WITH LOCAL ACTIVE SUBSPACE DIMENSION

A poor design of the parameter space could add an avoidable complexity to the surrogate modeling algorithms. Often, in practical applications, each parameter range is chosen independently with respect to the others. Then, it is the responsibility of the surrogate modelling procedure to disentangle the correlations among the parameters. However, in this way, looking at the response surface from parameters to outputs, regions that present different degrees of correlation are treated indistinctly. In this matter, a good practice is to study as a preprocessing step some sensitivity measures, like the total Sobol indexes [265] among groups of parameters, and

Table 3.1: Computational complexity of hierarchical top-down clustering.

Step	Cost	Description
Root	$O(Nnp^2 + Nn^2p + n^3)$	AS
	$O(N^3p^3)$	GPR
First refinement:	$O(N(N-k)^2)$	K-medoids
k from m to M	$O((N/k)np^2 + (N/k)n^2p + n^3)$	AS
	$O((N/k)^3p^3)$	GPR
Intermediate refinements	-	-
Last refinement:	$O((N/k^{l-1})((N/k^{l-1}) - k)^2)$	K-medoids
k from m to M	$O((N/k^l)np^2 + (N/k^l)n^2p + n^3)$	AS
for each one of the m^{l-1} clusters	$O((N/k^l)^3p^3)$	GPR

split the parameter space accordingly in order to avoid the use of more expensive surrogate modeling techniques later.

Of course Sobol indices or the global active subspace sensitivity scores give summary statistics on the whole domain. So in general, one could study the parameter space more in detail, classifying nonlinearly regions with respect to the complexity of the response surface, if there are enough samples to perform such studies.

We introduce an effective approach to tackle the problem of classification of the parameter space with respect to a local active subspace information. With the latter we mean two possible alternatives.

Definition 2 (Local active subspace dimension). *Given a threshold $\epsilon > 0$, the pairs of inputs and gradients $\{(\mathbf{X}_i, \mathbf{dY}_i)\}_i$ associated to an objective function of interest $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$, the size of the neighbour of sample points to consider $N \geq n$, and a subsampling parameter $p \in \mathbb{N},\ p \leq N$, the local active subspace dimension r_i associated to a sample point $\mathbf{X}_i \in \mathcal{X}$ is the positive integer*

$$r_i = \underset{1\leq r\leq p}{\operatorname{argmin}} \left\{ tr\left((Id - P_r) \left(\frac{1}{p} \sum_{i\in J} \mathbf{dY}_i \otimes \mathbf{dY}_i \right) (Id - P_r) \right) \leq \epsilon \;\middle|\; J \in C(N,p) \right\},$$

where $C(N,p)$ is the set of combinations without repetition of the N elements of the Euclidean neighbour of $\mathbf{X}_i$ with class p.

Definition 3 (Local active subspace). *Given the pairs of inputs and gradients $\{(\mathbf{X}_i, \mathbf{dY}_i)\}_i$ associated to an objective function of interest $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$, the size of the neighbour of sample points to consider $N \geq n$, and a fixed dimension $p \in \mathbb{N},\ 1 \leq p \leq N$, the local active subspace W_i associated to a sample point $\mathbf{X}_i \in \mathcal{X}$ is the matrix of the first p eigenvectors of the spectral decomposition of*

$$(Id - P_r) \left(\frac{1}{p} \sum_{i\in U} \mathbf{dY}_i \otimes \mathbf{dY}_i \right) (Id - P_r), \tag{3.21}$$

where U is the neighbour of sample points of $\mathbf{X}_i$ with respect to the Euclidean distance. In practice we choose p close to the global active subspace dimension. The pairs $\{(\mathbf{X}_i, W_i)\}_i$ can be thought as a discrete vector bundle of rank p and $\{W_i\}_i$ can be thought as a subset of points of the Grassmannian $Gr(N, p)$.

Starting from the pairs of inputs-gradients $\{(\mathbf{X}_i, \mathbf{dY}_i)\}_i$, the procedure follows the following steps:

1. Each parameter sample is enriched with the additional feature corresponding to the local active subspace dimension from definition 2 or the local active subspace from definition 3, represented by the variable $\mathbf{Z}$.

2. Each sample $\mathbf{X}_i$ is labelled with an integer l_i that will be used as classification label in the next step. To label the pairs $\{(\mathbf{X}_i, \mathbf{Z}_i)\}_i$ we selected K-medoids with Grassmanian metric

$$d((X_i, Z_i), (X_j, Z_j)) = \|Z_i - Z_j\|_F, \tag{3.22}$$

   where $\|\cdot\|_F$ is the Frobenious distance, in case $\mathbf{Z}_i$ represents the local active subspace or spectral clustering in case $\mathbf{Z}_i$ is the local active subspace dimension. In the last case, the labels correspond to the connected components of the graph built on the nodes $\{(\mathbf{X}_i, \mathbf{Z}_i)\}_i$ with adjacency list corresponding to the nearest nodes with respect to the distance

$$d((X_i, Z_i), (X_j, Z_j)) = \begin{cases} \infty, & Z_i \neq Z_j \\ \|X_i - X_j\|, & Z_i = Z_j \end{cases}, \tag{3.23}$$

   where $\|\cdot\|$ is the Euclidean metric in $\mathbb{R}^n$. The connected components are obtained from the eigenvectors associated to the eigenvalue 0 of the discrete Laplacian of the graph [188].

3. A classification method is applied to the inputs-labels pairs $\{(\mathbf{X}_i, l_i)\}_i$. Generally, for our relatively simple applications we apply a multilayer perceptron with 1000 hidden nodes and 2 layers.

Remark 18 (Grassmann distance)**.** *In general regarding the definition 3, the dimension p could be varying among samples $\mathbf{X}_i$ and one could use a more general distance with respect to the one from equation (3.22) that can have as arguments two vectorial subspaces of possibly different and arbitrary large dimensions.*

Remark 19 (Gradient-free active subspace)**.** *In general both the response surface design and the classification procedure above can be carried out from the pairs $\{(\mathbf{X}_i, \mathbf{Y}_i)\}_i$ of inputs, outputs instead of the sets $\{(\mathbf{X}_i, \mathbf{dY}_i)\}_i$ of inputs, gradients. In fact, the gradients $\{\mathbf{dY}_i\}$ can be approximated in many different ways [54] from $\{(\mathbf{X}_i, \mathbf{Y}_i)\}_i$. In the numerical results in section 3.4 when the gradients are not available they are approximated with the gradients of the local one-dimensional polynomial regression built on top of the neighbouring samples.*

Algorithm 11 Classification with local features from the AS information.

```
Input:
  inputs-gradients pairs ${(\mathbf{X}_i, \mathbf{dY}_i)}_{i\in I}$ as training dataset with index set $I$
  local features based on AS information ${\mathbf{Z}_i}_{i\in I}$
  labelling method based on the distance $d$ from equation (3.22) or from equation (3.23)
  classification method taking as input the inputs-labels pairs ${(\mathbf{X}_i, l_i)}_{i\in I}$
Output:
  predictor for new test inputs and classes of the training dataset.
for each $i \in I$ do
    evaluate the feature $\mathbf{Z}_i$ from $(\mathbf{X}_i, \mathbf{dY}_i)$ and the neighbouring points of $\mathbf{X}_i$
end for
initialize the$|I| \times |I|$ distance matrix $M$ associated to the pairs ${(\mathbf{X}_i, \mathbf{Z}_i)}_{i\in I}$
for each $i \in I$ do
    for each $i \leq j \in I$ do
        $M(i, j) = d((\mathbf{X}_i, \mathbf{Z}_i), (\mathbf{X}_j, \mathbf{Z}_j))$
    end for
end for
use the labelling method with input $M$, to assign a label $l_i$ for each pair $(\mathbf{X}_i, \mathbf{Z}_i)$
train the classification method with the inputs-labels training pairs ${(\mathbf{X}_i, l_i)}_{i\in I}$
```

3.4 NUMERICAL RESULTS

In this section we are going to apply the proposed LAS method to some datasets of increasing complexity. We compare the clustering techniques we presented in section 3.2, and we show how the active subspaces-based distance metric outperforms the Euclidean one for those functions which present a global lower intrinsic dimensionality. We remark that for hierarchical top-down clustering we can use both metrics, and we always show the best case for the specific dataset.

We start from a bidimensional example for which we can actually plot the clusters and the regressions, and compare the different techniques. Even if it is not a case for which one should use parameter space dimensionality reduction we think it could be very useful for the reader to understand also visually all the proposed techniques. For the higher dimensional examples we compare the accuracy of the methods in terms of R^2 score and classification performance.

All the computations regarding AS are done with the open source Python package[1] called ATHENA [222], for the classification algorithms we use the scikit-learn package [42] and for the Gaussian process regression GPy [91].

3.4.1 *Some illustrative bidimensional examples*

We start by presenting two bidimensional test cases in order to be able to present every aspect of the methodology together with illustrative plots. First we analyse a case where a global active subspace, even if present, does not provide a regression

[1] Available at https://github.com/mathLab/ATHENA/.

accurate enough along the active direction in section 3.4.1.1. Then we consider a radial symmetric function for which, by construction, an AS does not exist in section 3.4.1.2 and the use of K-means is instead preferable since we cannot exploit a privileged direction in the input domain.

3.4.1.1 *Quartic function*

Let us consider the following bidimensional quartic function:

$$f(\mathbf{x}) = x_1^4 - x_2^4, \qquad \mathbf{x} = (x_1, x_2) \in [0,1]^2. \tag{3.24}$$

In figure 3.2 we can see the contour plot of the function, the active subspace direction — translated for illustrative reasons — and the corresponding sufficient summary plot of the global active subspace, computed using 400 uniformly distributed samples. With sufficient summary plot we intend $f(\mathbf{x})$ plotted against the input parameters projected onto the active subspace, that is $W_1^T\mathbf{x}$. It is clear how, in this case, a univariate regression does not produce any useful prediction capability.

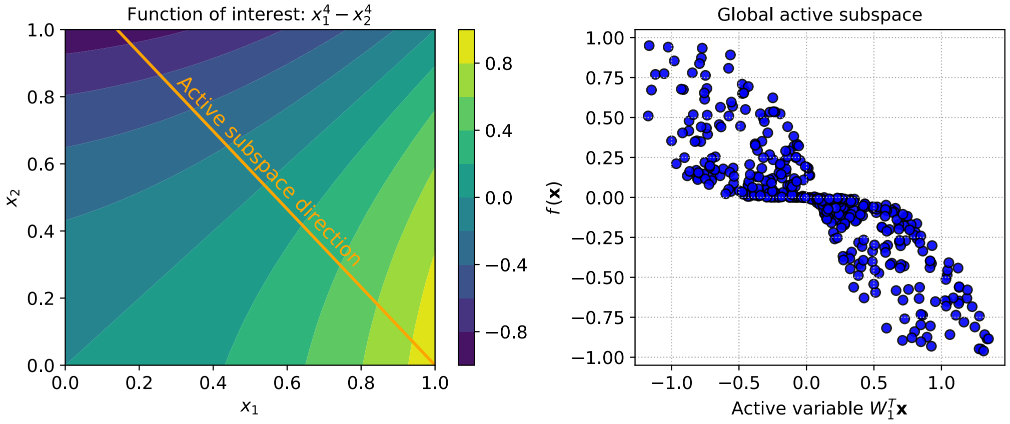


Figure 3.2: On the left panel the contour plot of the quartic function considered in equation (3.24) and in orange the global active subspace direction. On the right panel the sufficient summary plot resulting projecting the data onto the global AS.

Let us apply the clustering techniques introduced in the previous sections fixing the number of clusters to 4. In figure 3.3 we can clearly see how the supervised distance metric in equation (3.3) acts in dividing the input parameters. On the left panel we apply K-means which clusters the data into 4 uniform quadrants, while in the middle and right panels we have K-medoids and hierarchical top-down, respectively, with a subdivision aligned with the global AS. We notice that for this simple case the new metric induces an identical clustering of the data. In figure 3.4 we plotted the sufficient summary plots for each of the clusters individuated by K-medoids or hierarchical top-down in figure 3.3. It is clear how by using a single univariate regression for each cluster the R^2 score improves a lot with respect to a global approach (see right panel of figure 3.2).

We can also compare the R^2 scores for all the methods, using a test datasets of 600 samples. In figure 3.5 we report the scores for K-means, K-medoids and for hierarchical top-down with AS-based distance metric. The score for the global AS,

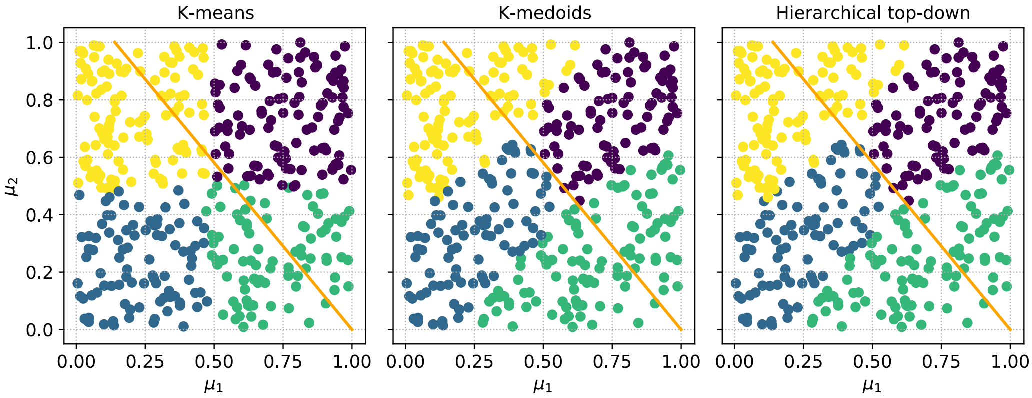


Figure 3.3: Comparison between the different clusters obtained by K-means (on the left), K-medoids (middle panel), and hierarchical top-down (on the right) with AS induced distance metric defined in equation (3.3) for the quartic test function. In orange the global active subspace direction. Every cluster is depicted in a different color.

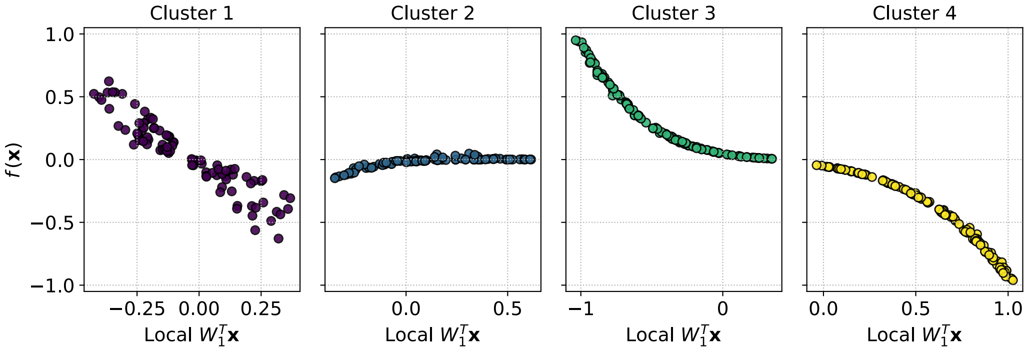


Figure 3.4: Local sufficient summary plots for the 4 clusters individuated by K-medoids or hierarchical top-down in figure 3.3 (colors correspond).

which is 0.78, is not reported in figure 3.5 for illustrative reasons. The results are very similar due to the relatively simple test case, but we can see that even with 2 clusters the gain in accuracy is around 23% using the metric in equation (3.3).

The hierarchical top-down clustering method was ran with the following hyper-parameters: the total number of clusters is increasing from 2 to 10, the minimum number of children equal to the maximum number of children equal to 3, uniform normalization of the clusters, the minimum size of each cluster is 10 elements, the clustering method is K-medoids with AS distance, the maximum active subspace dimension is 1.

Then we want to increase the accuracy of the regression with 3 clusters, loosing in some regions the reduction in the parameter space. Starting from the clustering with hierarchical top-down and 3 clusters of dimension 1, the AS dimension of each of the 3 clusters is increased if the threshold of 0.95 on the local R^2 score is not met. In general, the local R^2 score is evaluated on a validation set, for which predictions from the local response surfaces are obtained, after each valdidation sample is classified into one of the 3 clusters.

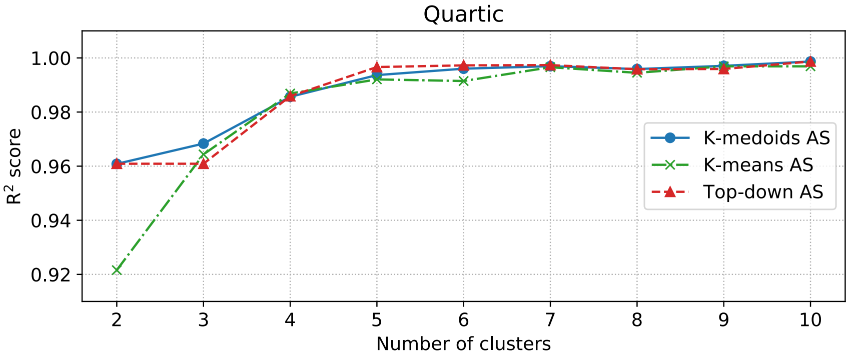


Figure 3.5: R^2 scores comparison between local versions varying the number of clusters for the quartic function in equation (3.24). Global AS has a score equal to 0.78.

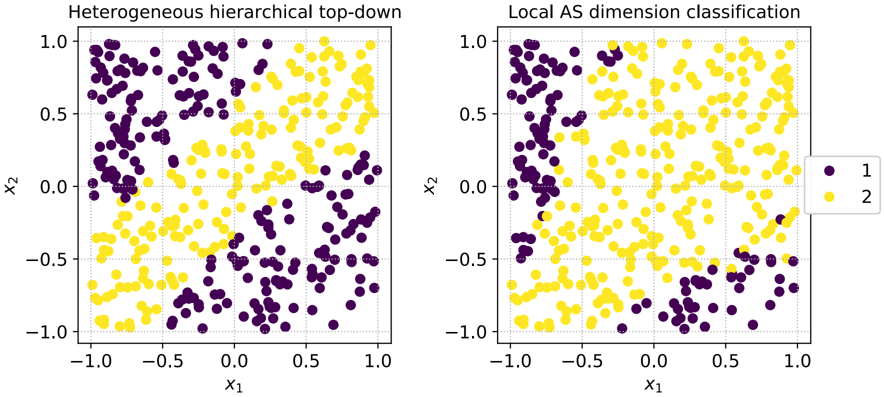


Figure 3.6: On the left panel the hierarchical top-down clustering with heterogeneous AS dimension and R^2 score equal to 1. On the right panel the labels of the local AS dimension from definition 2.

The 3 clusters are reported in figure 3.6 on the left. The R^2 score on the test set is 1, instead of around 0.97 from figure 3.5. To obtain this result, the central cluster AS dimension is increased from 1 to 2. We compare the clustering with respect to the classification of the local AS dimension with algorithm 11 using as features the local AS dimension as defined in definition 2, on the right of figure 3.6. Actually algorithm 11 is stopped after the plotted labels are obtained as the connected components of the underlying graph to which spectral clustering is applied: no classification method is employed, yet. It can be seen that hierarchical top-down clustering with heterogeneous AS dimension is more efficient with respect to the classes of algorithm 11, regarding the number of samples associated to a response surface of dimension 2.

3.4.1.2 *Radial symmetric cosine*

This example addresses the case for which an active subspace is not present. This is due to the fact that there are no preferred directions in the input domain since the function f has a radial symmetry. For this case the exploitation of the supervised distance metric does not provide any significant gain and K-means clustering works better on average, since it does not use the global AS structure. The model function we consider is the following

$$f(\mathbf{x}) = \cos(\|\mathbf{x}\|^2)\,, \qquad \mathbf{x} \in [-3,3]^2. \tag{3.25}$$

In figure 3.7 we compare the R^2 scores for K-means, K-medoids with AS-based metric, and hierarchical top-down with Euclidean metric. We used 500 training samples and 500 test samples. We see K-medoids has not a clear behaviour with respect to the number of clusters, while the other methods present a monotonic trend and better results on average, especially K-means. On the other hand local models improve the accuracy considerably, even for a small number of clusters, with respect to a global model.

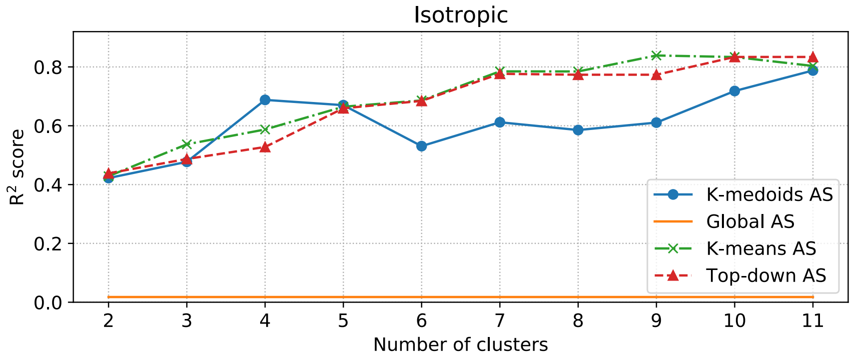


Figure 3.7: R^2 scores comparison between global AS and local versions varying the number of clusters for the isotropic model function in equation (3.25). Global AS corresponds to no clustering.

In this case the specifics of hierarchical top-down clustering are: the minimum number of children is equal to the maximum, the minimum number of elements per cluster is 10, the clustering method chosen is K-means, the normalization employed it the uniform one, and the total number of clusters is increasing from 2 to 11.

3.4.2 *High-dimensional datasets*

In this section we consider some more interesting benchmarks, for which dimension reduction in the parameter space is more useful since the starting dimension of the parameter space is higher. We test the classification procedure in algorithm 11 with an objective function with 6 parameters and defined piecewise as a paraboloid with different AS dimensions. We also test the procedure of response surface design with LAS, with a classical 8-dimensional epidemic benchmark model.

3.4.2.1 *Multi-dimensional hyper-paraboloid*

The objective function $f : [-4,4]^6 \to \mathbb{R}$ we consider is defined piecewise as follows

$$f(x) = \begin{cases} x_1^2 & \text{if } x_1 > 0 \text{ and } x_2 > 0, \\ x_1^2 + x_2^2 & \text{if } x_1 < 0 \text{ and } x_2 > 0, \\ x_1^2 + x_2^2 + x_3^2 & \text{if } x_1 > 0 \text{ and } x_2 < 0, \\ x_1^2 + x_2^2 + x_3^2 + x_4^2 & \text{if } x_1 < 0 \text{ and } x_2 < 0. \end{cases} \tag{3.26}$$

In the 4 domains in which f is defined differently, we expect an AS dimension ranging from 1 to 4, respectively. We employed algorithm 11 using the local AS dimensions as additional features, from definition 2: the values of the hyper-parameters are the following: $\epsilon = 0.999$, $N = 6$, $p = 4$. In figure 3.8 we plot the accuracy of the classification of the labels, associated to the connected components of the graph built as described in algorithm 11, and also the accuracy of the classification of the local active subspace dimension, that takes the values from 1 to 4. The test dataset for both the classification errors has size 1000. The score chosen to asses the quality of the classification is the mean accuracy, that is the number of correctly predicted labels over the total number of labels. For both the classification tasks 100 train samples are enough to achieve a mean accuracy above 80%.

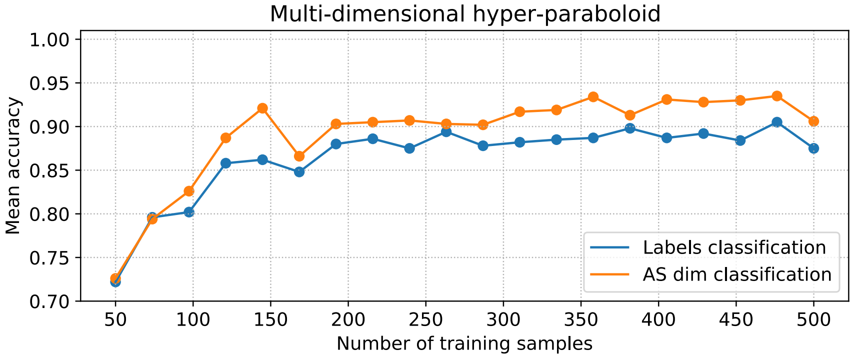


Figure 3.8: Mean accuracy study for a training dataset increasing in size from 50 to 500 samples. The test set is made of 1000 independent samples. The classification accuracy for the procedures of connected component classification (in blue) and local AS dimension classification (in orange) are both shown.

We remark that every step is applied to a dataset of samples in a parameter space of dimension 6, even if, to get a qualitative idea of the performances of the method, in figure 3.9 we show only the first two components of the decision boundaries of the 4 classes for both the previously described classification problems.

3.4.2.2 *Ebola epidemic model*

In this section we examine the performance of the proposed methods over the dataset created with the SEIR model for the spread of Ebola[2]. The output of interest

[2] The dataset was taken from `https://github.com/paulcon/as-data-sets`.

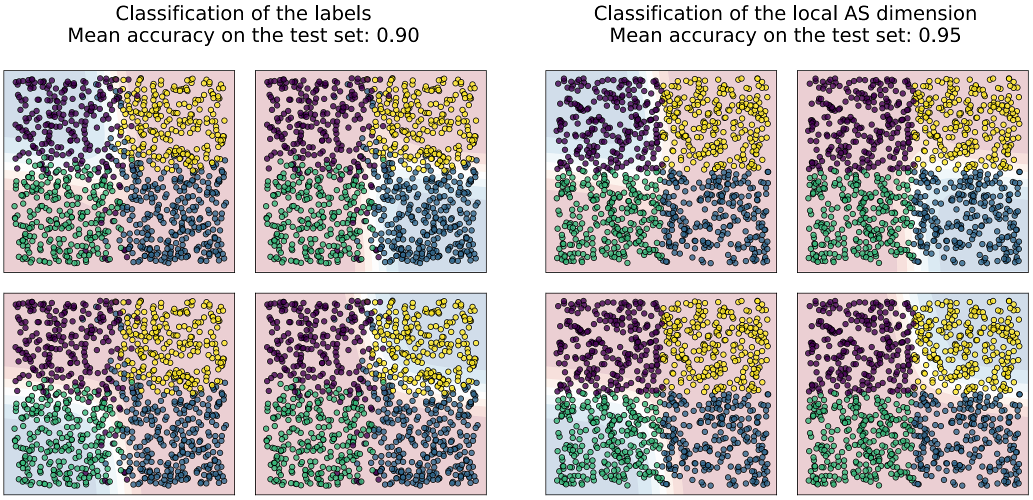


Figure 3.9: On the left panel the decision boundaries of the 4 classes associated to the connected components of the graph built as described in algorithm 11. On the right panel the decision boundaries of the 4 classes associated to the local AS dimension from 1 to 4. The datasets has dimension 6, only the first two components of the decision boundaries and of the test samples are plotted.

in this case is the basic reproduction number R_0 of the SEIR model, described in [74], which is computed using 8 parameters as follows

$$R_0 = \frac{\beta_1 + \frac{\beta_2 \rho_1 \gamma_1}{\omega} + \frac{\beta_3}{\gamma_2}\psi}{\gamma_1 + \psi}. \tag{3.27}$$

As shown in previous works, this function has a lower intrinsic dimensionality, and thus a meaningful active subspace, in particular of dimension 1. To evaluate the performance of the local AS we compute the R^2 score, as in equation (3.12), varying the number of clusters from 2 to 10 for all the methods presented. The test and training datasets are composed by 500 and 300, respectively, uniformly distributed and independent samples. The results are reported in figure 3.10, where as baseline we reported the R^2 for the GPR over the global AS. We can see how the use of the AS-based distance metric contributes the most with respect to the actual clustering method (compare K-medoids and hierarchical top-down in the plot). K-means, instead, does not guarantee an improved accuracy (for 4 and 9 clusters), and in general the gain is limited with respect to the other methods, especially for a small number of clusters which is the most common case in practice, since usually we work in a data scarcity regime. The results for K-medoids and top-down are remarkable even for a small amount of clusters with an R^2 above 0.9 and an improvement over 10% with respect to the global AS, which means that no clustering have been used.

The hyper-parameters for the hierarchical top-down algorithm are the following: the maximum local active subspace dimension is 1, the maximum number of children is equal to the number of total clusters, the minimum number of children is 2 at each refinement level, the minimum number of elements per cluster is 10, and the clustering method for each refinement is K-medoids with AS distance.

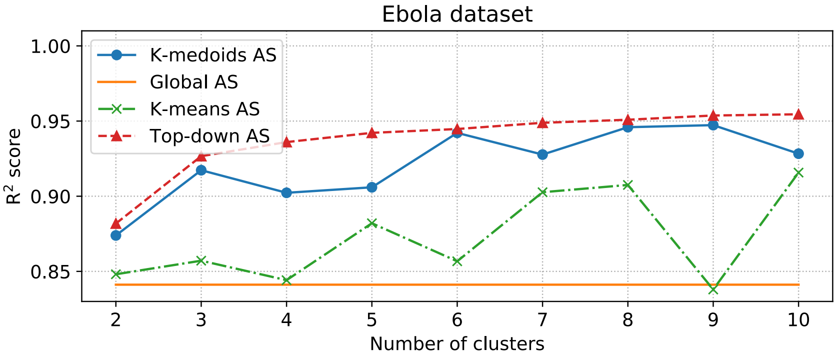


Figure 3.10: R^2 scores comparison between global AS and local versions varying the number of clusters for the Ebola spread model defined in equation (3.27). Global AS corresponds to no clustering.

3.4.3 *Datasets with vectorial outputs*

In this section we want to show how hierarchical top-down clustering and the classification procedure of algorithm 11 can be combined to improve the overall reduction in the parameter space, for a fixed lower threshold in the R^2 score. For the response surface design with active subspaces for vectorial outputs we refer to [221, 291].

3.4.3.1 *Poisson equation with random diffusivity*

Let us consider the stochastic Poisson problem on the square $\mathbf{x} = (x, y) \in \Omega := [0,1]^2$, defined as:

$$\begin{cases} -\nabla \cdot (\kappa \; \nabla u) = 1, & \mathbf{x} \in \Omega, \\ u = 0, & \mathbf{x} \in \partial\Omega_{\text{top}} \cup \partial\Omega_{\text{bottom}}, \\ u = 10y(1-y), & \mathbf{x} \in \partial\Omega_{\text{left}}, \\ \mathbf{n} \cdot \nabla u = 0, & \mathbf{x} \in \partial\Omega_{\text{right}}, \end{cases} \tag{3.28}$$

with homogeneous Neumann boundary condition on $\partial\Omega_{\text{right}}$, and Dirichlet boundary conditions on the remaining part of $\partial\Omega$. The diffusion coefficient $\kappa : (\Omega, \mathcal{A}, P) \times \Omega \to \mathbb{R}$, with $\mathcal{A}$ denoting a σ-algebra, is such that $\log(\kappa)$ is a Gaussian random field, with covariance function $G(\mathbf{x}, \mathbf{y})$ defined by

$$G(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\beta^2}\right), \quad \forall\, \mathbf{x}, \mathbf{y} \in \Omega, \tag{3.29}$$

where the correlation length is $\beta = 0.03$. We approximate this random field with the truncated Karhunen–Loève decomposition as

$$\kappa(s, \mathbf{x}) \approx \exp\left(\sum_{i=0}^{m} X_i(s)\gamma_i \boldsymbol{\psi}_i(\mathbf{x})\right), \qquad \forall (s, \mathbf{x}) \in \Omega \times \Omega, \tag{3.30}$$

where $(X_i)_{i\in 1,\dots,m}$ are independent standard normal distributed random variables, and the eigenpairs of the Karhunen–Loève decomposition of the zero-mean random field κ are denoted with $(\gamma_i, \boldsymbol{\psi}_i)_{i\in 1,\dots,d}$. The parameters $(X_i)_{i\in 1,\dots,m=10}$ sampled from a standard normal distribution are the coefficients of the Karhunen-Loève expansion, truncated at the first 10 modes, so the parameter space has dimension $m = 10$.

The domain Ω is discretized with a triangular unstructured mesh $\mathcal{T}$ with 3194 triangles. The simulations are carried out with the finite element method with polynomial order 1. The solution u is evaluated at 1668 degrees of freedom, thus the output is vectorial with dimension $d = 1688$. As done in [221, 291], the output is enriched with the metric induced by the Sobolev space $H^1(\Omega)$ on to the finite element space of polynomial order 1: the metric is thus represented by a $d \times d$ matrix M obtained as the sum of the mass and stiffness matrices of the numerical scheme and it is involved in the AS procedure when computing the correlation matrix $\mathbb{E}\left[Df\, M\, Df^T\right]$, where Df is the $m \times d$ Jacobian matrix of the objective function $f : \mathbb{R}^{10} \to \mathbb{R}^d$.

Since the output is high-dimensional we classified with algorithm 11 the output space in 6 clusters, using the Grassmann distance from equation (3.22), as shown in figure 3.11.

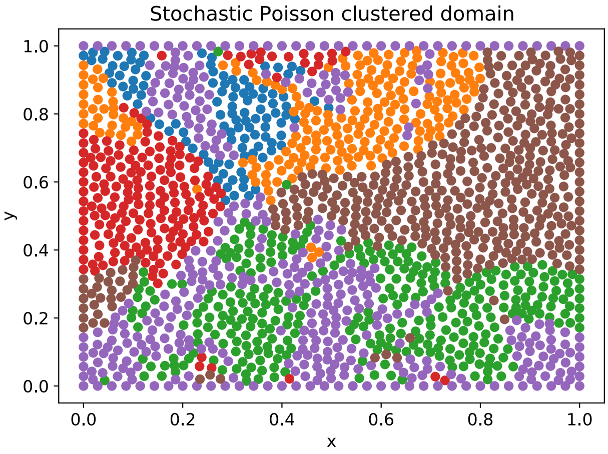


Figure 3.11: Subdivision of the spatial domain Ω in 6 clusters based on the Grassmann distance from definition 3, i.e. the clusters correspond to the connected components of the graph built on top of the degrees of freedom with adjacency list determined using as distance definition 3.

Afterwards we applied hierarchical top-down clustering to every one of the 6 triples of inputs-outputs-gradients, obtained restricting the outputs and the gradients to each one of the 6 clusters. The specifics of hierarchical top-down clustering we employed are the following: the minimum and maximum number of children for each refinement are equal to the total number of clusters, which is 4, the minimum number of elements in each cluster is 10, and the clustering algorithm chosen is K-medoids with the AS distance. The size of the training and test datasets is respectively of 500 and 150. The gradients are evaluated with the adjoint method. Since the output is vectorial we employed the mean R^2 score, where the average is made among the components of the vectorial output considered.

Then for every lower threshold on the R^2 score we increase one by one the dimension of the 6×4 local clusters, until all the R^2 scores of each of the 6 triples are above the fixed threshold. The same procedure is applied to the whole dataset of inputs-outputs-gradients but executing hierarchical top-down clustering just once, for all the output's components altogether.

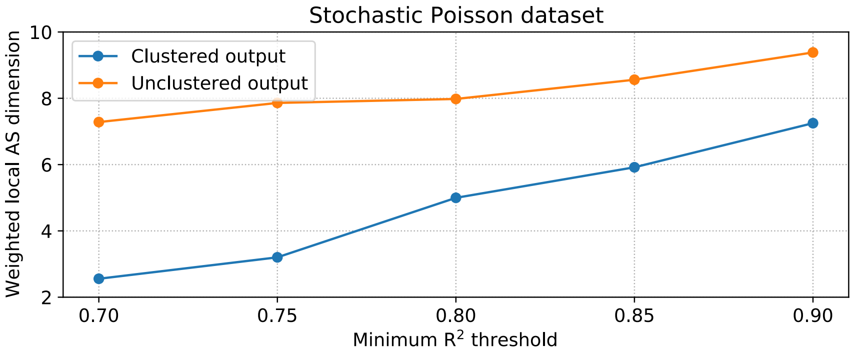


Figure 3.12: In orange the local AS dimensions weighted on the number of elements of each of the 4 clusters in the parameter space, obtained with hierarchical top-down clustering. In blue the local AS dimensions weighted on the number of elements of each of the 4 clusters in the parameter space, obtained with hierarchical top-down clustering, times 6 clustered outputs (see figure 3.11) for a total of 24 terms in the weighted average.

The results are reported in figure 3.12. In the case of the clustered outputs, the local dimension of each one of the 6 clustered outputs times 4 local clusters in the parameter space, for a total of 24 local clusters, are weighted with the number of elements of each cluster. In the same way the 4 clusters of the case with unclustered outputs is weighted with the number of the elements of each one of the 4 clusters. It can be seen that for every fixed threshold, there is an evident gain, with respect to the dimension reduction in the parameter space, in clustering the outputs and then performing hierarchical top-down clustering in the parameter space.

3.4.3.2 *Shape design of an airfoil*

For this vectorial test case we consider the temporal evolution of the lift coefficient of a parametrized NACA airfoil. Here we briefly present the problem we solve to create the dataset, we refer to [271] and to chapter 10 for a deeper description.

Let us consider the unsteady incompressible Navier-Stokes equations described in an Eulerian framework on a parametrized space-time domain $S(\boldsymbol{\mu}) = \Omega(\boldsymbol{\mu}) \times$

$[0, T] \subset \mathbb{R}^2 \times \mathbb{R}^+$. The vectorial velocity field $\mathbf{u} : S(\boldsymbol{\mu}) \to \mathbb{R}^2$, and the scalar pressure field $p : S(\boldsymbol{\mu}) \to \mathbb{R}$ solve the following parametric PDE:

$$\begin{cases} \mathbf{u_t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot 2\nu\nabla^s\mathbf{u} = -\nabla p & \text{in } S(\boldsymbol{\mu}), \\ \nabla \cdot \mathbf{u} = \mathbf{0} & \text{in } S(\boldsymbol{\mu}), \\ \mathbf{u}(t, x) = \mathbf{f}(\mathbf{x}) & \text{on } \Gamma_{\text{in}} \times [0, T], \\ \mathbf{u}(t, x) = \mathbf{0} & \text{on } \Gamma_0(\boldsymbol{\mu}) \times [0, T], \\ (\nu\nabla\mathbf{u} - p\mathbf{I})\mathbf{n} = \mathbf{0} & \text{on } \Gamma_{\text{out}} \times [0, T], \\ \mathbf{u}(0, \mathbf{x}) = \mathbf{k}(\mathbf{x}) & \text{in } S(\boldsymbol{\mu})_0 \end{cases} . \tag{3.31}$$

Here, $\Gamma = \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_0$ denotes the boundary of $\Omega(\boldsymbol{\mu})$ composed by inlet boundary, outlet boundary, and physical walls, respectively. With $\mathbf{f}(\mathbf{x})$ we indicate the stationary non-homogeneous boundary condition, and with $\mathbf{k}(\mathbf{x})$ the initial condition for the velocity at $t = 0$. The geometrical deformation are applied to the boundary $\Gamma_0(\boldsymbol{\mu})$. The undeformed configuration corresponds to the NACA 4412 wing profile [1, 126]. To alter such geometry, we adopt the shape parametrization and morphing technique proposed in [114], where 5 shape functions are added to the airfoil profiles. Let y_u and y_l be the upper and lower ordinates of the profile, respectively. The deformation of such coordinates is described as follows

$$y_u = \overline{y_u} + \sum_{i=1}^{5} c_i r_i, \qquad y_l = \overline{y_l} - \sum_{i=1}^{5} d_i r_i, \tag{3.32}$$

where the bar denotes the reference undeformed state. The parameters $\boldsymbol{\mu} \in \mathbb{D} \subset \mathbb{R}^{10}$ are the weights coefficients, c_i and d_i, associated with the shape functions r_i. In particular we set $\mathbb{D} := [0, 0.03]^{10}$. The explicit formulation of the shape functions can be found in [114]. For this datasets, the Reynolds number is $Re = 50000$. The time step is $dt = 1e - 3s$. For other specifics regarding the solver employed and the numerical method adopted see [271].

As outputs we considered the values of the lift coefficient, every 15 time steps from $100ms$ to $30000ms$, for a total of 1994 components. Even in this case the output is classified with algorithm 11 with distance defined in definition 2. The values of the lift coefficient physically interesting are collected at last, after an initialization phase. Nonetheless for the purpose of having a vectorial output we considered its value from the time instant $100ms$. The procedure finds two classes and splits the ordered output components in two parts: from the component 0 to 996, the local AS dimension is 1, for the remaining time steps it is higher. So we can expect an improvement on the efficiency of the reduction in the parameter space when considereing separately these two sets of outputs components as figure 3.13 shows. The weighted local AS dimension is in fact lower when using clustering, for every minimum R^2 threshold.

3.5 CONCLUSIONS AND PERSPECTIVES

In this chapter we present a new local approach for parameter space reduction which exploits supevised clustering techniques, such as K-means, K-medoids, and

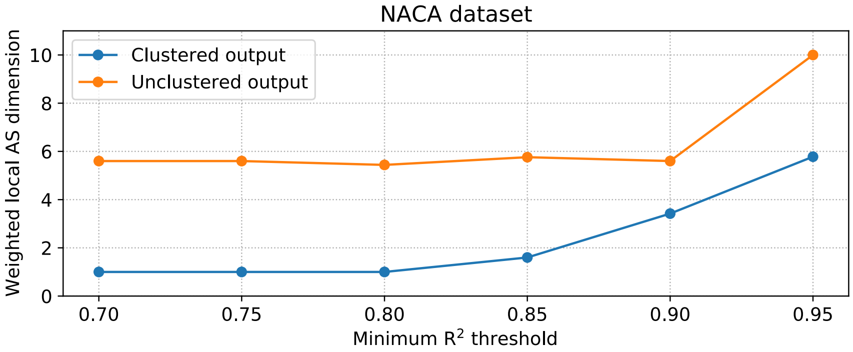


Figure 3.13: In orange the local AS dimensions weighted on the number of elements of each of the 2 clusters in the parameter space, obtained with hierarchical top-down clustering. In blue the local AS dimensions weighted on the number of elements of each of the 2 clusters in the parameter space, obtained with hierarchical top-down clustering, times 2 clustered outputs for a total of 4 terms in the weighted average.

hierarchical top-down, with a distance metric based on active subspaces. We called this method Local Active Subspaces.

The theoretical formulation provides error estimates for the contruction of response surfaces over the local active subspaces. We also present a classification approach to capture the optimal AS dimension for each cluster and can be used as a preprocessing step, both for the inputs and the vectorial outputs, for the construction of more accurate regressions and surrogate modeling. The proposed approach is very versatile, especially the hierarchical top-down clustering which can incorporate quite different criteria. The methodology has been validated over a vast range of datasets, both scalar and vector-valued, showing all the strengths and a possible weakness, in case of radial symmetric functions. In all the test cases LAS achieved superior performance with respect to the classical global approach.

Possible future lines of research can focus on the study of the extension of this methods to nonlinear parameter space reduction techniques, or on the use of more advanced clustering criteria.

4

MULTI-FIDELITY DATA FUSION THROUGH ACTIVE SUBSPACES

In this chapter we present a multi-fidelity approach based on a nonlinear autoregressive scheme involving Active Subspaces, which we called Nonlinear Autoregressive Multi-fidelity Gaussian Process Regression with Active Subspaces (NARGPAS).

Gaussian processes are employed for non-parametric regression in a Bayesian setting. They generalize linear regression embedding the inputs in a latent manifold inside an infinite-dimensional reproducing kernel Hilbert space. We can augment the inputs with the observations of low-fidelity models in order to learn a more expressive latent manifold and thus increment the model's accuracy. This can be realized recursively with a chain of Gaussian processes with incrementally higher fidelity. We would like to extend these multi-fidelity model realizations to case studies affected by an high-dimensional input space but with a low intrinsic dimensionality. In this cases physical supported or purely numerical low-order models are still affected by the curse of dimensionality when queried for responses. When the model's gradients information is provided, the presence of an active subspace can be exploited to design low-fidelity response surfaces and thus enable Gaussian process multi-fidelity regression, without the need to perform new simulations. This is particularly useful in the case of data scarcity. After presenting the multi-fidelity scheme we test it on two different high-dimensional benchmarks. All the results presented in this chapter appeared in [224].

4.1 LITERATURE REVIEW

Every day more and more complex simulations are made possible thanks to the high performance computing facilities spread and to the advancements in computational methods. Still the study and the approximation of high-dimensional functions of interest represent a problem due to the curse of dimensionality and the data scarcity because of limited computation budgets.

Gaussian Processes (GP) [287] have been proved as a versatile and powerful technique for regression GPR, classification, inverse problem resolution, and optimization, among others. In the last years several studies and extensions have been proposed in the context of non-parametric and interpretable Bayesian models. For a review on Gaussian processes and kernel methods we suggest [135], while for approximation methods in the framework of GP see [213]. Progress has also been made to deal with big data and address some memory limitations of GPR, as for example by sparsifying the spectral representation of the GP [154] or introducing stochastic variational inference for GP models [164].

Multi-fidelity modelling has been proven effective in a heterogeneous set of applications [29, 30, 87, 138, 143, 218], where expensive but accurate high-fidelity measurements are coupled with cheaper to compute and less accurate low-fidelity data. Recent advancements have been made for Nonlinear Autoregressive Multi-fidelity Gaussian Process Regression (NARGP) as proposed in [203], and with physics informed neural networks (PINNs) [217] in the context of multi-fidelity approximation in [176].

The increased expressiveness of these models is achieved thanks to some kind of nonlinear approach that extend GP models to non-Gaussian processes with the disadvantage of an additional computational cost. In this direction are focused the following works which aim to obtain computationally efficient heteroscedastic GP models with a variational inference approach [155] or employing a nonlinear transformation [253]. This approach is extended to multi-fidelity models departing from the linear formulation of Kennedy and O'Hagan [138] towards deep Gaussian processes [62] and NARGP.

When the models depend on a high-dimensional input space even the low-fidelity approximations supported by a physical interpretation or a purely numerical model order reduction suffer from the curse of dimensionality especially for the design of high-dimensional GP models. Active Subspaces [54, 291] can be used to build a surrogate low-fidelity model with reduced input space taking advantage of the correlations of the model's gradients when available. Reduction in parameter space through AS has been proven successful in a diverse range of applications such as: shape optimization [73, 168], hydrologic models [129], naval and nautical engineering [67, 184, 267, 270, 272], coupled with intrusive reduced order methods in cardiovascular studies [266], in CFD problems in a data-driven setting [72, 271]. A kernel-based extension of AS for both scalar and vectorial functions can be found in [221].

The aim of the present contribution is to propose a multi-fidelity regression model which exploits the intrinsic dimensionality of high-dimensional functions of interest and the presence of an active subspace to reduce the approximation error of high-fidelity response surfaces. Our approach employs the design of a NARGP using an AS response surface as low-fidelity model. In the literature the multi-fidelity approximation paradigm has been adopted in a different way to search for an active subspace from given high- and low-fidelity models [152].

The outline of this chapter is the following: in section 4.2 we present the general framework of Gaussian process regression and in particular the NARGP multi-fidelity approach; to recall Active Subspaces, which we exploit as low-fidelity model, the reader can refer to section 2.2; section 4.3 is devoted to present how data fusion with AS is performed with the aid of algorithms; in section 4.4 we apply the proposed approach to the piston and Ebola benchmark models showing the better performance achieved by the multi-fidelity regression; finally in section 4.5 we present our conclusions and we draw some future perspectives.

4.2 MULTI-FIDELITY GAUSSIAN PROCESS REGRESSION

In the next subsections we are going to present the GPR [287] technique which is the building block of multi-fidelity GPR, and the NARGP [203] we are going to use in this work. The numerical methods are presented for the general case of several levels of fidelity.

4.2.1 *Gaussian process regression*

Gaussian process regression [287] is a supervised technique to approximate unknown functions given a finite set of input/output pairs $\mathcal{S} = \{x_i, y_i\}_{i=1}^N$. Let $f : \mathcal{X} \subset \mathbb{R}^D \to \mathbb{R}$ be the scalar function of interest. The set $\mathcal{S}$ is generated through f with the following relation: $y_i = f(x_i)$, which are the noise-free observations. To f is assigned a prior with mean $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}'; \theta)$, that is $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'; \theta))$. The prior expresses our beliefs about the function before looking at the observed values. From now on we consider zero mean $\mathcal{GP}$, $m(\mathbf{x}) = \mathbf{0}$, and we define the covariance matrix $\mathbf{K}_{i,j} = k(x_i, x_j; \theta)$, with $\mathbf{K} \in \mathbb{R}^{N \times N}$. To use the Gaussian process to make prediction we still need to find the optimal values of the elements of the hyper-parameters vector θ. We achieve this by maximizing the log marginal likelihood:

$$\log p(\mathbf{y}|\mathbf{x}, \theta) = -\frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}| - \frac{N}{2}\log 2\pi. \tag{4.1}$$

Let $\mathbf{x}_*$ be the test samples, and $\mathbf{K}_{N*} = k(\mathbf{x}, \mathbf{x}_*; \theta)$ be the matrix of the covariances evaluated at all pairs of training and test samples, and in a similar fashion $\mathbf{K}_{*N} = k(\mathbf{x}_*, \mathbf{x}; \theta)$, and $\mathbf{K}_{**} = k(\mathbf{x}_*, \mathbf{x}_*; \theta)$. By conditioning the joint Gaussian distribution on the observed values we obtain the predictions f_* by sampling the posterior

$$f_*|\mathbf{x}_*, \mathbf{x}, \mathbf{y} \sim \mathcal{N}(\mathbf{K}_{*N}\mathbf{K}^{-1}\mathbf{y}, \mathbf{K}_{**} - \mathbf{K}_{*N}\mathbf{K}^{-1}\mathbf{K}_{N*}). \tag{4.2}$$

4.2.1.1 *A Gaussian process regression example*

As an explanatory example we are going to approximate the function $f : \Omega \subset \mathbb{R} \to \mathbb{R}$ defined as

$$f(x) := \sin((x-2.5)^2) + 0.6\cos(x^2+0.5) \qquad x \in \Omega := [0,5], \tag{4.3}$$

using Gaussian process regression.

A Gaussian process provides a prior over some infinite-dimensional function, defined by a mean function m and a covariance function k:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x^*)). \tag{4.4}$$

When we sample from the kernel k, to create a matrix over some sample space, we are creating a matrix of values that describe the covariance between sample points. Since it is not possible to sample every single point in an infinite dimensional function, we have to sample a finite subset of the input domain. Let $\mathbf{X}$ denote some inputs, and $\mathbf{K}$ the covariance matrix associated, that is $K_{ij} = k(\mathbf{X}_i, \mathbf{X}_j)$, then we can describe the prior over $f(\mathbf{X})$ as a (finite-dimensional) normal distribution with covariance $\mathbf{K}$. As such, we can easily create samples of f which are representative of the true function.

We can also sample from the kernel prior by creating a covariance matrix over a sample space and sampling from a zero-mean multivariate normal distribution with covariance $\mathbf{K}$. After the Gaussian process regression, the fitted posterior can also be sampled in the same manner, to get samples of the fitted function. In figure 4.1 we show some sampling from the prior (top panel) and from the posterior (bottom panel) of a Gaussian process resulting from an RBF kernel, that is a stationary infinitely differentiable kernel parametrized by a legthscale and a variance. While the kernel function controls the smoothness and the amplitude of the samples from the GP, the prior mean represents a possible offset. In practice is set to a constant value and then inferred from data, but in general it is a way to incorporate expert knowledge using a problem dependent, possible non-linear, prior mean. We can see how after 10 function evaluations the posterior mean is closer to the exact solution and the confidence intervals are smaller.

4.2.2 *Nonlinear multi-fidelity Gaussian process regression*

We adopt the NARGP scheme proposed in [203]. It extends the concepts present in [138, 158] to nonlinear correlations between the different fidelities available.

We introduce p levels of increasing fidelities and the corresponding sets of input/output pairs $\mathcal{S}_q = \{x_i^q, y_i^q\}_{i=1}^{N_q} \subset \mathcal{X} \times \mathbb{R} \subset \mathbb{R}^n \times \mathbb{R}$ for $q \in \{1, \dots, p\}$, where $y_i^q = f_q(x_i^q)$. With p we indicate the highest fidelity. We also assume the design sets to have a nested structure: $\mathcal{S}_p \subset \mathcal{S}_{p-1} \subset \dots \subset \mathcal{S}_1$.

The NARGP formulation considers the following autoregressive multi-fidelity scheme:

$$f_q(\mathbf{x}) = g_q(\mathbf{x}, f_{*q-1}(\mathbf{x})), \tag{4.5}$$

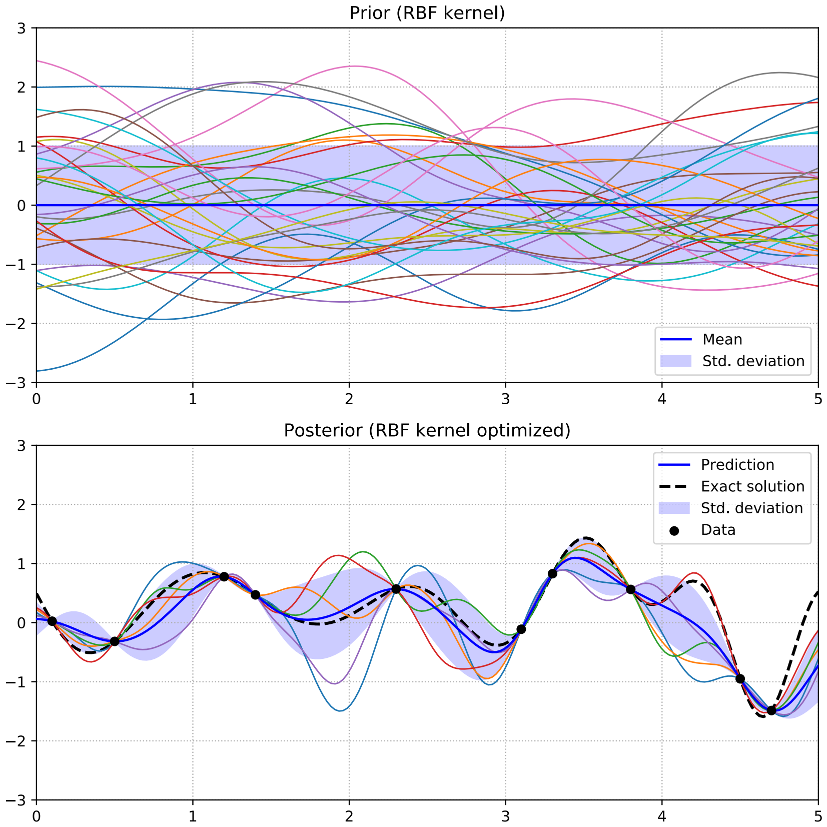


Figure 4.1: Different samples from the prior and from the posterior of a Gaussian process resulting from an RBF kernel. In the top panel 30 samples from the prior. In the bottom panel 5 samples from the posterior after 10 data points evaluations. With the black dashed line we show the exact solution.

where $f_{*q-1}(\mathbf{x})$ is the GP posterior from the previous inference level $q-1$, and to g_q is assigned the following prior:

$$g_q \sim \mathcal{GP}(f_q|\mathbf{0}, k_q; \theta_q) \tag{4.6}$$

$$k_q = k_q^{\rho}(\mathbf{x}, \mathbf{x}'; \theta_q^{\rho}) \cdot k_q^{f}(f_{*q-1}(\mathbf{x}), f_{*q-1}(\mathbf{x}'); \theta_q^{f}) + k_q^{\delta}(\mathbf{x}, \mathbf{x}'; \theta_q^{\delta}), \tag{4.7}$$

with k_q^{ρ}, k_q^{f}, and k_q^{δ} squared exponential kernel functions. With this scheme, through g_q we can infer the high-fidelity response by projecting the lower fidelity posterior to a latent manifold of dimension $D+1$. This structure allows for nonlinear and more general cross-correlations between subsequent fidelities.

A part from the first level of fidelity $q=1$ the posterior probability distribution given the previous fidelity models is no longer Gaussian since the inputs are couples $((\mathbf{x}, \mathbf{x}_*), (y_{q-1}(\mathbf{x}), f_{*q-1}(\mathbf{x}_*))$ where f_{q-1} is a Gaussian process $g_q \sim \mathcal{GP}(f_q|\mathbf{0}, k_q; \theta_q)$, the training set is $(\mathbf{x}, f_{q-1}(\mathbf{x}))$ and $\mathbf{x}_*$ is the new input. So in order to evaluate the predictive mean and variance for a new input $\mathbf{x}_*$ we have to integrate the usual Gaussian posterior $p(f_{*q}(\mathbf{x}_*, f_{*q-1}(\mathbf{x}_\star))|f_{*q-1}, \mathbf{x}_*, \mathbf{x}_q, \mathbf{y}_q)$ explicited as

$$f_{*q}(\mathbf{x}_*, f_{*q-1}(\mathbf{x}_\star))|f_{*q-1}, \mathbf{x}_*, \mathbf{x}_q, \mathbf{y}_q \sim \mathcal{N}(\mathbf{K}^q_{*N}(\mathbf{K}^q)^{-1}y_q, \mathbf{K}^q_{**} - \mathbf{K}^q_{*N}(\mathbf{K}^q)^{-1}\mathbf{K}^q_{N*}), \tag{4.8}$$

$$K^q_{*N} = k_q((\mathbf{x}_*, f_{*q-1}(\mathbf{x}_*), (\mathbf{x}_{q-1}, y_{q-1}); \theta), \tag{4.9}$$

$$K^q_{N*} = k_q((\mathbf{x}_{q-1}, y_{q-1}), (\mathbf{x}_*, f_{*q-1}(\mathbf{x}_*); \theta), \tag{4.10}$$

$$K^q = k_q((\mathbf{x}_{q-1}, y_{q-1}), (\mathbf{x}_{q-1}, y_{q-1}); \theta), \tag{4.11}$$

over the Gaussian distribution of the prediction at the previous level $f_{*q-1}(\mathbf{x}_*) \sim \mathcal{N}(\mu_{x_*}, \sigma_{\mathbf{x}_*})$. In practice the following integral is approximated with recursive Monte Carlo in each fidelity level

$$p(f_{*q}(\mathbf{x}_*, f_{*q-1}(\mathbf{x}_*))) = \int_{\mathcal{X}} p(f_{*q}(\mathbf{x}_*, f_{*q-1}(\mathbf{x}_*))|f_{*q-1}, \mathbf{x}_*, \mathbf{x}_q, \mathbf{y}_q) p(f_{*q-1}(\mathbf{x}_*)) d\mathbf{x}_*. \tag{4.12}$$

4.3 MULTI-FIDELITY DATA FUSION WITH ACTIVE SUBSPACES

Our study is based on the design of a NARGP [203] whose low-fidelity level is learnt from a response surface built through the active subspaces methodology. In fact we suppose that the model in consideration has indeed a high dimensional input space but its intrinsic dimensionality is sufficiently lower. This is often the case as shown by the numerous industrial applications [129, 168, 266, 272].

The whole procedure requires the knowledge of an input/output high-fidelity training set $\{(\mathbf{x}^H_i, y^H_i)\}^{N_H}_{i=1} \subset \mathbb{R}^m \times \mathbb{R}$, completed by the gradients $\{dy^H_i\}^{N_H}_{i=1} \subset \mathbb{R}^m$ needed for the active subspace's presence inquiry and a low-fidelity input set $\{\mathbf{x}^L_i\}^{N_L}_{i=1} \subset \mathbb{R}^m$. We represent with N_H, N_L the number of high-fidelity and low-fidelity training set samples, respectively. Differently from the usual procedure the low-fidelity outputs $\{y^L_i\}^{N_L}_{i=1}$ are predicted with the response surface built thanks to the knowledge of the active subspace through the dataset $\{(\hat{W}_1\mathbf{x}^H_i, y^H_i)\}^{N_H}_{i=1}$. At the same time the response surface is also queried for the predictions $\{y^{H,\text{train}}_i\}^{N_H}_{i=1}$ at the high-fidelity inputs $\{\mathbf{x}^H_i\}^{N_H}_{i=1}$ that will be used for the training of the multi-fidelity model. Now all the ingredients for the same procedure described in [203] are ready: the multi-fidelity model is trained at the low-fidelity level with $\{(x^L_i, y^L_i)\}^{N_L}_{i=1}$ and at the high-fidelity level with $\{((x^H_i, y^{H,\text{train}}_i), y^H_i)\}^{N_H}_{i=1}$.

We remark that in this case the same high-fidelity outputs $\{y^H_i\}^{N_H}_{i=1} \subset \mathbb{R}$ are used for the response surface training and the high-fidelity training of the multi-fidelity model. In fact the outputs $\{y^{H,\text{train}}_i\}^{N_H}_{i=1}$ predicted with the response surface are equal to $\{y^H_i\}^{N_H}_{i=1} \subset \mathbb{R}$ since the response surface is a Gaussian process with no noise trained on the dataset $\{(\hat{W}_1\mathbf{x}^H_i, y^H_i)\}^{N_H}_{i=1}$ and queried for the same inputs $\{\hat{W}_1\mathbf{x}^H_i\}^{N_H}_{i=1}$ for the predictions, that is $\{y^{H,\text{train}}_i\}^{N_H}_{i=1}$. This results in the training of the high-fidelity level of the multi-fidelity model with the dataset $\{((x^H_i, y^{H,\text{train}}_i), y^H_i)\}^{N_H}_{i=1} = \{((x^H_i, y^H_i), y^H_i)\}^{N_H}_{i=1}$.

A second procedure is developed where part of the high-fidelity inputs is used only to train the response surface such that in general $\{y^H_i\}^{N_H}_{i=1} \neq \{y^{H,\text{train}}_i\}^{N_H}_{i=1}$. The

Algorithm 12 NARGP with the same high-fidelity samples for AS response surface design.

```
Input:
  high-fidelity inputs, outputs, gradients triplets:
    $\{(\mathbf{x}_i^H, y_i^H, dy_i^H)\}_{i=1}^{N_H} \subset \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^m$
  low-fidelity inputs $\{\mathbf{x}_i^L\}_{i=1}^{N_L} \subset \mathbb{R}^m$
  training dataset $\{(\mathbf{x}_i^{\text{test}}, y_i^{\text{test}})\}_{i=1}^{N_{\text{test}}}$
Output:
  multi-fidelity model:
    $g_M = ((f_H|x_i^H, y_i^{H,\text{train}}),\ (f_L|x_i^L)) \sim (\mathcal{GP}(f_H|m_H, \sigma_H), \mathcal{GP}(f_L|m_L, \sigma_L))$
Compute the active subspace $\hat{W}_1$ with the high-fidelity gradients $\{dy_i^H\}_{i=1}^{N_H}$
Build the one-dimensional response surface $\mathcal{R}(\hat{W}_1\mathbf{X})$ with a GP regression from
  $\{(\hat{W}_1\mathbf{x}_i^H, y_i^H)\}_{i=1}^{N_H}$
Predict the low-fidelity outputs $\{y_i^L\}_{i=1}^{N_L}$ at $\{\mathbf{x}_i^L\}_{i=1}^{N_L}$ and the training high-fidelity
  inputs $\{y_i^{H,\text{train}}\}_{i=1}^{N_H}$ at $\{\mathbf{x}_i^H\}_{i=1}^{N_H}$ with the response surface
Train the multi-fidelity model at the low-fidelity level $g_L$ with the training
  dataset $\{(x_i^L, y_i^L)\}_{i=1}^{N_L}$
Train the multi-fidelity model at the high-fidelity level $g_H$ with the training
  dataset $\{((x_i^H, y_i^{H,\text{train}}), y_i^H)\}_{i=1}^{N_H}$
```

main difference is that the surrogate low-fidelity model is built independently from the high-fidelity level of the multi-fidelity model.

We expect that with the multi-fidelity approach, thanks to the nonlinear fidelity fusion realized by the method, not only the lower accuracy of the low-fidelity model will be safeguarded against, but also a hint towards the presence of an active subspace will be transferred from the low-fidelity to the high-fidelity level. In fact the low-fidelity GPR model is built from the predictions obtained with the r-dimensional response surface which expressiveness is guaranteed by the additional assumption that the model under investigation has a r-dimensional active subspace. So a part from the lower computational budget and the reduced accuracy, our low-fidelity model should transfer to the high-fidelity level the knowledge of the presence of an active subspace when learning correlations among the inputs $\{x_i^H\}_{i=1}^{N_H}$, the response surfaces predictions $\{y_i^{H,\text{train}}\}_{i=1}^{N_H}$ and the high-fidelity targets $\{y_i^H\}_{i=1}^{N_H}$. The overhead with respect to the original procedure [203] is the evaluation of the active subspace from the high-fidelity inputs.

The procedure is synthetically reviewed through algorithm 12 for the use of the same high-fidelity samples in the training of the response surface and of the second fidelity level of the multi-fidelity model, and algorithm 13 for the use of independent samples. The main difference in the two procedures is the set of samples with which the active subspace is computed.

Some finals remarks are due. As in [203] we assume that the observations $\{y_i^q\}$ are noiseless for each level of fidelity q. We employ Radial Basis Functions kernels with Automatic Relevance Determination (ARD). The hyperparameters tuning is achieved maximizing the log-likelihood with the gradient descent optimizer L-BFGD.

Algorithm 13 NARGP with additional independent high-fidelity samples for AS response surface design.

```
Input:
  high-fidelity inputs, outputs, gradients triplets:
    $\{(\mathbf{x}_i^H, y_i^H, dy_i^H)\}_{i=1}^{N_H} \subset \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^m$
  low-fidelity inputs $\{\mathbf{x}_i^L\}_{i=1}^{N_L} \subset \mathbb{R}^m$
  training dataset $\{(\mathbf{x}_i^{\text{test}}, y_i^{\text{test}})\}_{i=1}^{N_{\text{test}}}$
  AS procedure's inputs, outputs, gradients triplets:
    $\{(\mathbf{x}_i^{\text{AS}}, y_i^{\text{AS}}, dy_i^{\text{AS}})\}_{i=1}^{N_{\text{AS}}} \subset \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^m$
Output:
  multi-fidelity model:
    $g_M = ((f_H | x_i^H, y_i^{H,\text{train}}),\ (f_L | x_i^L)) \sim (\mathcal{GP}(f_H | m_H, \sigma_H), \mathcal{GP}(f_L | m_L, \sigma_L))$
Compute the active subspace $\hat{W}_1$ with the active subspaces dataset's gradients $\{dy_i^{\text{AS}}\}_{i=1}^{N_H}$
Build the one-dimensional response surface $\mathcal{R}(\hat{W}_1 \mathbf{X})$ with a GP regression from $\{(\hat{W}_1 \mathbf{x}_i^{\text{AS}}, y_i^{\text{AS}})\}_{i=1}^{N_{\text{AS}}}$
Predict the low-fidelity outputs $\{y_i^L\}_{i=1}^{N_L}$ at $\{\mathbf{x}_i^L\}_{i=1}^{N_L}$ and the training high-fidelity inputs $\{y_i^{H,\text{train}}\}_{i=1}^{N_H}$ at $\{\mathbf{x}_i^H\}_{i=1}^{N_H}$ with the response surface
Train the multi-fidelity model at the low-fidelity level $g_L$ with the training dataset $\{(x_i^L, y_i^L)\}_{i=1}^{N_L}$
Train the multi-fidelity model at the high-fidelity level $g_H$ with the training dataset $\{((x_i^H, y_i^{H,\text{train}}), y_i^H)\}_{i=1}^{N_H}$
```

4.4 NUMERICAL RESULTS

We consider two different benchmark test problems for which a multi-fidelity model will be built. The first is a 8-dimensional model for the spread of Ebola[1] and the second is a 7-dimensional model to compute the time it takes a cylindrical piston to complete a cycle[2]. The library employed to implement the NARGP model is Emukit [195] while for the active subspace's response surface design we have chosen ATHENA[3] [222] for the active subspaces presence study and GPy [91] for the GPR.

These tests have already been analyzed for the presence of an active subspace and they indeed present a low intrinsic dimensionality. For each model we show the sufficient summary plot along the one-dimensional active subspace found, and the correlation among the low-fidelity level and the high-fidelity level of the multi-fidelity model. We also present a comparison of the prediction error with respect to a low-fidelity model (LF) represented by a GPR on the low-fidelity input/output dataset, an High-fidelity (HF) model represented by a GPR on the high-fidelity input/output dataset, and the proposed Multi-fidelity (MF) model. In each test case the number of low-fidelity samples is 200 and an error study over the number of high-fidelity samples used is undergone. We apply both the algorithms presented

[1] The Ebola dataset was taken from https://github.com/paulcon/as-data-sets.
[2] The piston dataset was taken from https://github.com/paulcon/active_subspaces.
[3] Available at https://github.com/mathLab/ATHENA.

in the previous section. In particular for algorithm 12, the number of samples used to find the active subspace is one third of the total number of high-fidelity samples rounded down.

THE PISTON MODEL The algebraic cylindrical piston model appeared as a test for statistical screening in [20], while applications of AS to this model can be found in [55]. The scalar target function of interest is the time it takes the piston to complete a cycle, and its computation involves a chain of nonlinear functions. This quantity depends on 7 input parameters uniformly distributed. The corresponding variation ranges are taken from [55]. The 10000 test points are samples with Latin-Hypercube sampling.

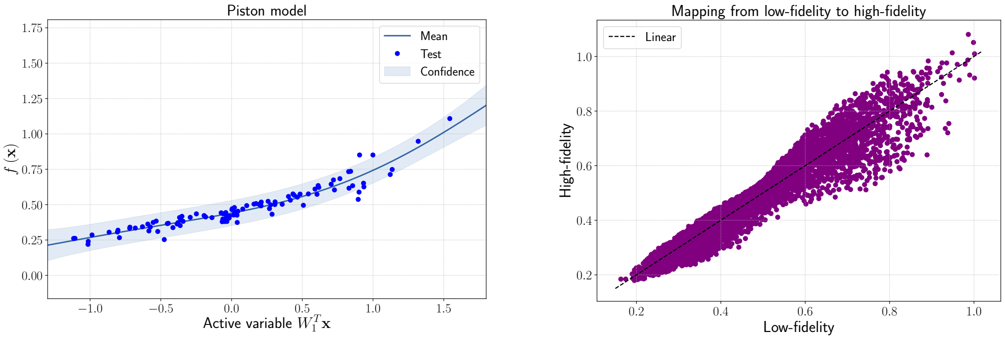


Figure 4.2: Left: sufficient summary plot of the surrogate model built with active subspaces. Right: correlation among the low-fidelity level and the high-fidelity level of the multi-fidelity model.

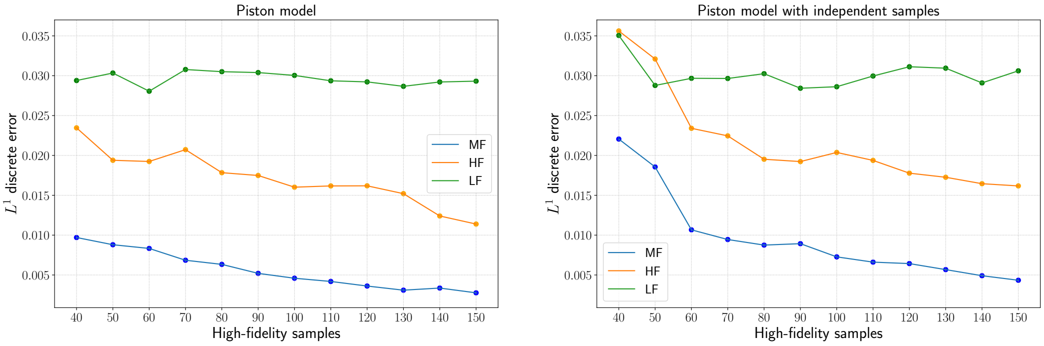


Figure 4.3: L^1 discrete error of the posterior of the multi-fidelity (MF), high-fidelity (HF) and low-fidelity (LF) models against the number of high-fidelity samples used to find the active subspace and build the Gaussian process regressions of the MF, HF, LF models. The 10000 test samples are distributed with Latin-Hypercube sampling. Left: identical high-fidelity samples for AS and NARGP. Right: independent high-fidelity samples for AS and NARGP.

It is qualitatively evident from the sufficient summary plot in the left panel of figure 4.2 that a one-dimensional active subspace is enough to explain with a fairy good accuracy the dependence of the output from the 7-dimensional inputs. This statement could be supported looking at the ordered eigenvalues of the correlation matrix of the gradients, which would show a spectral gap between the first and the second eigenvalues. We can also see in the right panel of figure 4.2 the correlations scatter plot among the two different fidelity levels of the NARGP model. The low-fidelity GPR built on the dataset $\{(x_i^L, y_i^L)\}_{i=1}^{N_L}$ performs already a good regression without many outliers in the predictions evaluated at the test samples.

Figure 4.3 shows the errors of the MF models built with different procedures, as described in section 4.3. It can be seen that using independent samples for the active subspace evaluation does not improve the predictions obtained. Since in the right panel one third of the high-fidelity samples are used to identify the active subspace, to evaluate the differences between the two algorithms we have to compare the 150 samples on the right with the 100 samples on the left, for example. We can clearly notice that the two approches perform almost the same.

SEIR MODEL FOR EBOLA The SEIR model for the spread of Ebola depends on 8 parameters and the output of interest is the basic reproduction number R_0. A complete AS analysis was made in [74], while a kernel-based active subspaces comparison can be found in [221]. The formulation is the following:

$$R_0 = \frac{\beta_1 + \frac{\beta_2 \rho_1 \gamma_1}{\omega} + \frac{\beta_3}{\gamma_2}\psi}{\gamma_1 + \psi}, \tag{4.13}$$

where the parameters range are taken from [74].

Differently from the previous test case, the one-dimensional response function in the left panel of figure 4.4 does not explain well the model: in this case kernel-based active subspaces could be employed to reach a better expressiveness of the surrogate model [221]. Even the scatter plot in the right panel of figure 4.4, which shows the correlations between the low-fidelity and high-fidelity levels of the NARGP model, exhibits a worse accuracy in the low-fidelity level with respect to the previous test case. These results are quantified in figure 4.5 with the L^1 discrete error for the different fidelities models which are one order of magnitude higher than the piston model, see figure 4.3.

From a comparison between the HF and MF models in figure 4.5 and the respective models in figure 4.3 it can be seen that the nonlinear autoregressive fidelity fusion approach learns relatively worse correlations of the low-/high-fidelity levels of the NARGP Ebola model with respect to the piston model.

For both the test cases the multi-fidelity regression approach with active subspaces results in better performance with a consistent reduction of the L^1 discrete error over a test dataset of 10000 points.

4.5 CONCLUSIONS

In this chapter we proposed a nonlinear multi-fidelity approach for the approximation of scalar function with a low intrinsic dimensionality. Such dimension is identified by searching for the existence of an active subspace for the function of interest. With a regression along the active variable we build a low-fidelity model over the full parameter space which is fast to evaluate and does not need any new simulations. We just extract new informations from the high-fidelity data we

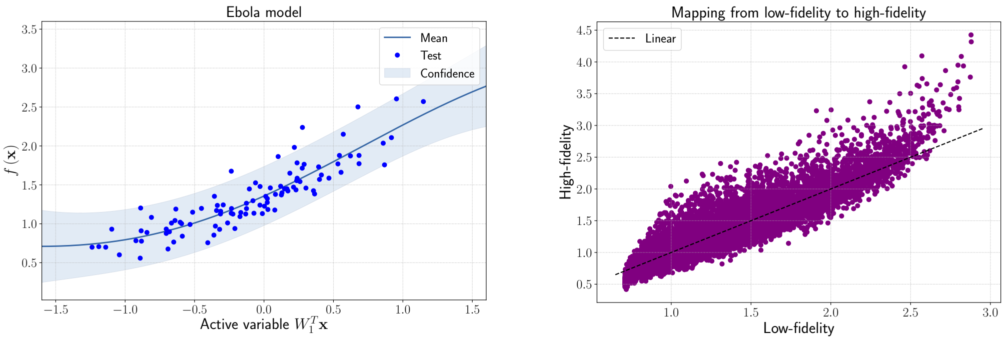


Figure 4.4: Left: sufficient summary plot of the Ebola model, 150 samples were used to build the AS surrogate model shown. Right: correlation among the low-fidelity level and the high-fidelity level of the multi-fidelity model.

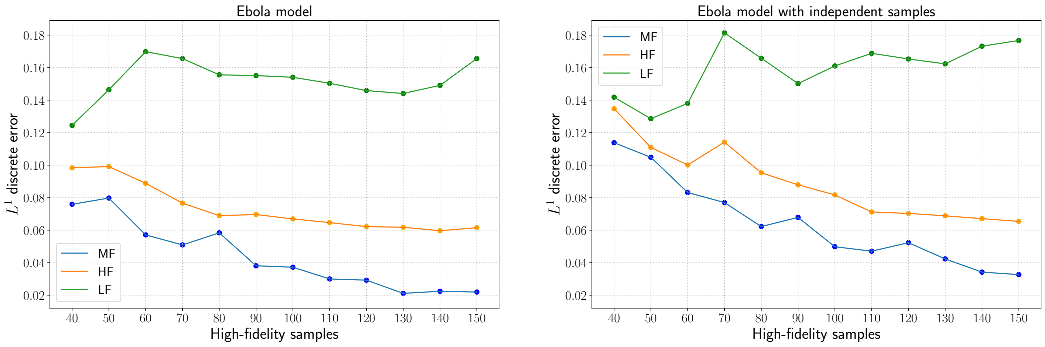


Figure 4.5: L^1 discrete error of the posterior of the multi-fidelity (MF), high-fidelity (HF) and low-fidelity (LF) models against the number of high-fidelity samples used to find the active subspace and build the Gaussian process regressions of the MF, HF, LF models. The 10000 test samples are distributed with Latin-Hypercube sampling. Left: identical high-fidelity samples for AS and NARGP. Right: independent high-fidelity samples for AS and NARGP.

already have. This multi-fidelity approach results in a decreased regression error and improved approximation capabilities over all the parameter space.

We apply the multi-fidelity with AS method to two different benchmark problems involving high dimensional scalar functions with an active subspace. We achieve promising results with a reduction of the L^1 discrete error around 60–70% with respect to the high-fidelity regression in one case (piston) and around 20–66% in the other one (Ebola), depending on the number of high-fidelity samples used.

As we are going to see in chapter 7, the methodology presented in this chapter has the potential to greatly improve data-driven non-intrusive reduced order methods [226, 227, 230, 268] through modal coefficients reconstruction and prediction for parametric problems. We also mention the possible application to shape optimization problems for the evaluation of both the target function and the constraints.

Further investigation will involve the use of more active subspaces based fidelities, such as KAS [221], or AS with different dimensions.

5

NON-INTRUSIVE REDUCED ORDER METHODS

In this chapter we review two of the most spread techniques in the context of data-driven non-intrusive model order reduction for parametric problems. We first present a general abstract framework for POD-based ROMs which goes under the name Proper Orthogonal Decomposition with Interpolation (PODI). In the second part we focus on time-depenent problems where the considered parameter is the time, showing how Dynamic Mode Decomposition (DMD) works. For a survey of projection-based ROMs for parametrized dynamical systems, instead, we suggest [21].

Applications of these two methods can be found in part iii for naval engineering optimization tasks, and in part iv where we try to extend DMD in a parametric setting for CFD predictions and we compare DMD and PODI for hydroacoustic analysis.

5.1 A GENERAL FRAMEWORK FOR NON-INTRUSIVE PARAMETRIC ROM

In this section we are going to present a general framework for non-intrusive data-driven reduced order models. We assume we do not know the equations behind the computations of the solutions snapshots, or, even if we know the model been used, we do not have access to the solver, so we cannot perform intrusive model order reduction.

From an abstract point of view we can identify three main steps to construct such kind of ROMs: the creation of the solutions database, the computation of a basis of reduced dimension, and the choice of a regression method to reconstruct the solution manifold. As we can already see there are multiple choices for every one of these steps. We can use, for example, different sampling methods to collect the snapshots, employ linear or nonlinear techniques for modes selection, and finally exploit from polynomial interpolators to artificial neural networks to reconstruct the parameters to modal coefficients map.

The framework described in this section with its modular structure can be found in the open source Python package[1] called EZyRB [70].

5.1.1 *Database creation*

Let us denote the parameters with μ and the state snapshots with x. The solutions database is thus the set of parameter–snapshot pairs defined by $\{\mu_i, x_i\}_{i=1}^m$, with $\mu_i \in P \subset \mathbb{R}^p$ and $x_i \in \mathbb{R}^n$. Here, n represents the number of degrees of freedom of our system, while p is the number of input parameters. We remark that the snapshots can represent data coming from experiments, from simulations, or even sensors and acquired in real-time. We arrange the snapshots by column in X as

$$X = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \dots & x_m \\ | & | & & | \end{bmatrix}. \tag{5.1}$$

There are several possible initial sampling strategies to select the parameters location, depending on the specific task for which we construct the reduced order model. For example to cover as much as possible the parameter space the most used approaches are latin hypercube sampling (LHS) [262], Sobol sequences [255], which are quasi-random low-discrepancy sequences, or other space filling strategies such as dynamic propagation sampling [141]. If the parameters represent data coming from sensors or other components with an associated probability distribution a weighted approach is suggested [273]. For optimization tasks any a priori knowledge about the possible region of interest and/or the simulated process should be incorporated within the sampling method. We remark that some of the aforementioned sampling schemes could not work in case the parameters can assume only a finite set of possible values.

[1] Available at https://github.com/mathLab/EZyRB/.

5.1.2 *Linear dimensionality reduction*

Linear dimensionality reduction is the most widespread technique to unveil structures within a large dataset. The main assumption is that the data can be expressed by a linear combination of few global basis functions, often called modes. Following the notation introduced in the previous section, we seek to approximate the snapshots x_i as

$$x_i = \sum_{k=1}^{m} \varphi_k c_i^k \approx \sum_{k=1}^{r} \varphi_k c_i^k, \qquad \forall i \in [1, \dots, m], \;\; r \ll m, \tag{5.2}$$

for some modes $\varphi_k \in \mathbb{R}^n$ and for some modal coefficients $c_i^k \in \mathbb{R}$. We can rewrite equation (5.2) in matrix form arranging all the data by column as done in equation (5.1) obtaining:

$$\begin{bmatrix} | & | & & | \\ x_1 & x_2 & \dots & x_m \\ | & | & & | \end{bmatrix} \approx \begin{bmatrix} | & | & & | \\ \varphi_1 & \varphi_2 & \dots & \varphi_r \\ | & | & & | \end{bmatrix} \begin{bmatrix} | & | & & | \\ c_1 & c_2 & \dots & c_m \\ | & | & & | \end{bmatrix}, \tag{5.3}$$

or equivalently $X \approx \Phi C$, with $X \in \mathbb{R}^{n \times m}$, $\Phi \in \mathbb{R}^{n \times r}$, and $C \in \mathbb{R}^{r \times m}$.

As we can see there are many possible choices for the modes computation: from the classical principal component analysis (PCA), also called proper orthogonal decomposition (POD), to other nonlinear approaches such as kernel POD [237] and autoencoders [159], to mention a few. In this chapter we are going to review only POD which has been proven successful in a variety of different industrial applications [227, 231, 268] and is both explainable and can be linked to energetic considerations. Moreover many high-dimensional data can be approximated by low-rank matrices [278]. In case of missing data or sensors providing partial informations, a gappy POD [41, 286] approach can be used. We emphasize that the basis identified by the POD can be also exploited in a Galerkin projection framework [100, 112, 263], or in an hybrid framework combining data-driven methods with projection [98, 116].

If the pair (Φ, C) minimizes the sum of the squares of the residuals $X - \Phi C$, that is in a least squares sense, we have that Φ is the matrix composed by the POD modes. Using the l_1 norm, instead of the least squares, leads to the so-called robust PCA [44], which is very useful in case outliers are present. To actually compute the modes Φ in the POD case, we can perform the SVD of the snapshots matrix X:

$$X = U \Sigma V^*, \tag{5.4}$$

where * denotes the conjugate transpose, and retain only the first r components. The columns of U are the so-called POD modes. They are orthogonal and they span the optimal low-dimensional subspace in the least square sense. They can also be computed as the eigenvectors (up to a normalization factor) of the Gram matrix $X^T X$. The diagonal matrix $\Sigma = \text{diag}(\lambda_1, \dots, \lambda_r)$ is composed by the singular values, arranged in descending order, which indicate the energetic contribution of the corresponding modes. The error introduced by the truncation r can be measured as [211]:

$$\| X - U \Sigma V^* \|_2^2 = \lambda_{r+1}^2, \tag{5.5}$$

$$\|X - U\Sigma V^*\|_F = \sqrt{\sum_{i=r+1}^{m} \lambda_i^2}, \tag{5.6}$$

where the subscripts 2 and F refer to the Euclidean norm and to the Frobenius norm, respectively. The singular values λ_i are arranged in descending order. In figure 5.1 we can see an example of POD modes for a flow past a cylinder modeled with a parametrized Navier–Stokes problem.

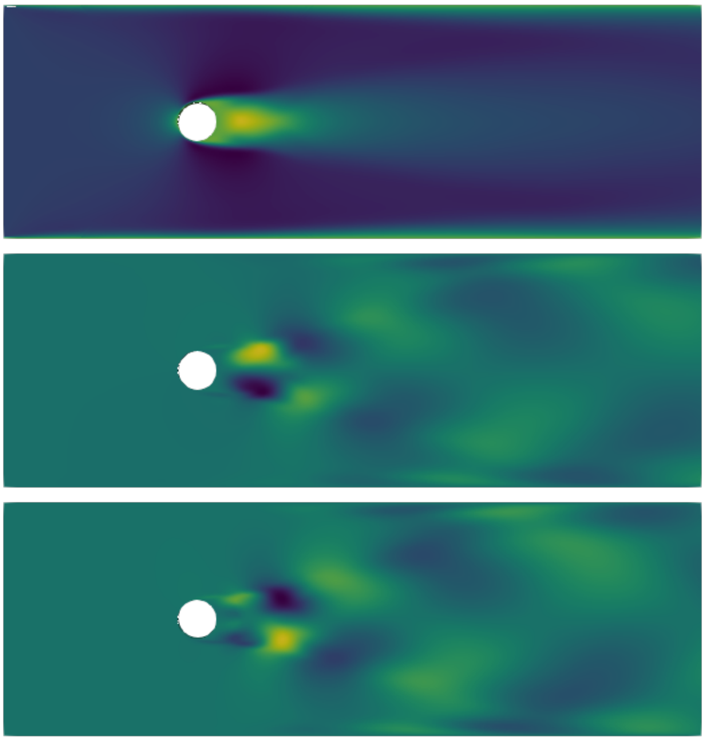


Figure 5.1: First three POD modes obtained from the resolution of a parametrized Navier–Stokes problem, describing the flow past a cylinder.

We remark that given the first r modes, we only need r coefficients c_i to reconstruct the whole state x_i. To compute the modal coefficients we project the data onto the POD subspace:

$$C = U^T X. \tag{5.7}$$

Given such coefficients we can only reconstruct the original snapshots which comprise the initial database. To have a fast and accurate prediction of the state x^*, corresponding to an unseen parameter μ^*, we need to reconstruct the parameters to modal coefficients map. Then we need only to perform a matrix multiplication with the POD modes to obtain the whole field of interest. In the next section we show some possible choice to reconstruct such map.

5.1.3 *Solution manifold approximation*

In this section we are going to present some possible choices to reconstruct the solution manifold described by the modal coefficients corresponding to the initial database. In particular we want to construct a regression $s : P \to \mathbb{R}^r$ which approximates the map $f : \mu \in P \to c \in \mathbb{R}^r$, given a set of m input-output pairs $\{\mu_i, c_i\}_{i=1}^m$, where $c_i = U^T x_i(\mu_i)$. The regression model s can then be used to predict the state x^* for a new input parameter μ^* by computing

$$x^* = Us(\mu^*). \tag{5.8}$$

To build the interpolant s we can use several techniques, such as linear interpolation [95, 231], nonlinear Gaussian Process Regression [72, 106, 193, 287], Radial Basis Functions interpolation [39, 268], inverse distance weighting (IDW) [14, 288], and artificial neural networks [205, 282] to cite a few. Here we will not review all the possibilities, some of them are presented in chapter 8 in the context of shape morphing. For what concerns GPR we refer to chapter 4 where we also presented a simple explanatory example. In chapter 8 can also be found an application of Proper Orthogonal Decomposition with Gaussian Process Regression (POD-GPR) used as surrogate model for shape optimization, while in chapter 9 we used a POD with RBF approach. In chapter 7, instead, we show a comparison between POD-GPR and POD-NARGPAS for structural problems in naval engineering.

From a computational point of view the the differences between the Full Order Model (FOM) and the reduced order model are remarkable: whereas the FOM requires the solution of a system of dimension n, the ROM only requires the query of the interpolation function and a matrix multiplication with the modes. So the computational cost of a ROM is mainly in the construction phase and not in the prediction. Since we rely on the SVD the method shows an algorithmic complexity of $\mathcal{O}(\min(n, m)nm)$.

5.2 Dynamic mode decomposition for time dependent problems

DMD is a powerful method to identify and approximate dynamical systems using only few spatiotemporal coherent structures [37, 149, 235]. The method is equation-free since it works using only high-dimensional snapshots data, and make almost no assumptions regarding the underlying system to approximate. This technique has been applied to a diverse range of fields, with a particular success in computational fluid dynamics. We can highlight three main tasks which can be accomplished by DMD: system identification, future-state prediction, and control.

Despite its versatility and performance DMD presents some limitations, mainly inherited from the SVD at its core. DMD is thus incapable of efficiently handling translations or rotations, and transient phenomena are not well characterized. Moreover the presence of an external input acting on the state results in a wrong identification of the system. To overcome these issues many variants have been developed and proposed in the past years, among them we mention DMD with control [209], multi-resolution DMD [150], compressed DMD [83], higher order DMD [157], and randomized DMD [25, 84].

For an efficient implementation of all the algorithms presented in this section we suggest to use the open source Python package[2] called PyDMD [71].

5.2.1 *Classical Dynamic Mode Decomposition algorithm*

We start by collecting the snapshots $\{x_i\}_{i=1}^m$ representing the state of the system at each time instant, with $x_i \in \mathbb{R}^n$. We arrange them by column in two matrices, X and X', where X' is the time-shifted version of X, that is

$$X = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \dots & x_{m-1} \\ | & | & & | \end{bmatrix}, \qquad X' = \begin{bmatrix} | & | & & | \\ x_2 & x_3 & \dots & x_m \\ | & | & & | \end{bmatrix}. \tag{5.9}$$

We seek a regression of the data onto locally linear dynamics such as $x_{k+1} = Ax_k$ for $k = 1, \dots, m-1$. In matrix form this can be written as

$$X' \approx AX. \tag{5.10}$$

A can be found as the best-fit matrix using the Moore–Penrose pseudoinverse of X, that is $X^\dagger$, as follows:

$$A = X'X^\dagger. \tag{5.11}$$

This solution minimizes the error in Frobenius norm $\|X' - AX\|_F$. Since the usual CFD problem is characterized by high-dimensional state snapshots with $n \gg m$, we cannot solve this problem directly with a regression method. We, instead, rely on a lower-dimensional operator $\tilde{A}$ which evolves over the POD modes of the snapshots matrix in order to approximate the eigenpairs of the full operator A. The eigenvectors of A are called DMD modes.

The DMD algorithm starts with a SVD of the snapshots matrix X with a truncation rank r:

$$X \approx U\Sigma V^*, \tag{5.12}$$

where $U \in \mathbb{C}^{n\times r}$, $\Sigma \in \mathbb{C}^{r\times r}$, and $V \in \mathbb{C}^{m-1\times r}$. With $*$ we denote the conjugate transpose. We emphasize that U and V are orthonormal, and the columns of U are the so-called POD modes. Substituting $X^\dagger$ in equation (5.11) with its SVD we obtain

$$A = X'V\Sigma^{-1}U^*. \tag{5.13}$$

From a computational point of view it is more efficient to project the operator A onto the POD modes obtaining its low-rank representation:

$$\tilde{A} = U^*AU = U^*X'V\Sigma^{-1}. \tag{5.14}$$

With this operator we can express the evolution of the low-dimensional projection of the snapshots as $\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k$, with $\tilde{x}_k = U^*x_k$. The final step it to compute the eigenpairs of $\tilde{A}$ with its eigendecomposition

$$\tilde{A}W = W\Lambda, \tag{5.15}$$

[2] Available at https://github.com/mathLab/PyDMD/.

where Λ stands for the diagonal matrix of the eigenvalues, and the columns of W are the eigenvectors. We remark that the eigenvalues of A are given by Λ, while the eigenvectors Φ, also called exact DMD modes [277], are given by

$$\Phi = X' V \Sigma^{-1} W. \tag{5.16}$$

In the literature we also be found the so-called projected DMD modes [235] expressed by $\Phi = UW$, which tend to converge to the exact ones if the matrixes in equation (5.9) share the column space.

The i-th eigenvalue λ_i provides information about the frequency ω_i of the corresponding mode, defined as

$$\omega_i = \frac{\operatorname{Im}(\log \lambda_i)}{2\pi\Delta t}, \tag{5.17}$$

and also about its stability. In figure 5.2 we can see the DMD modes corresponding to different eigenvalues: if the eigenvalue is on the unit circle we have a stable mode, if it is inside the unit circle is convergent, otherwise it diverges.

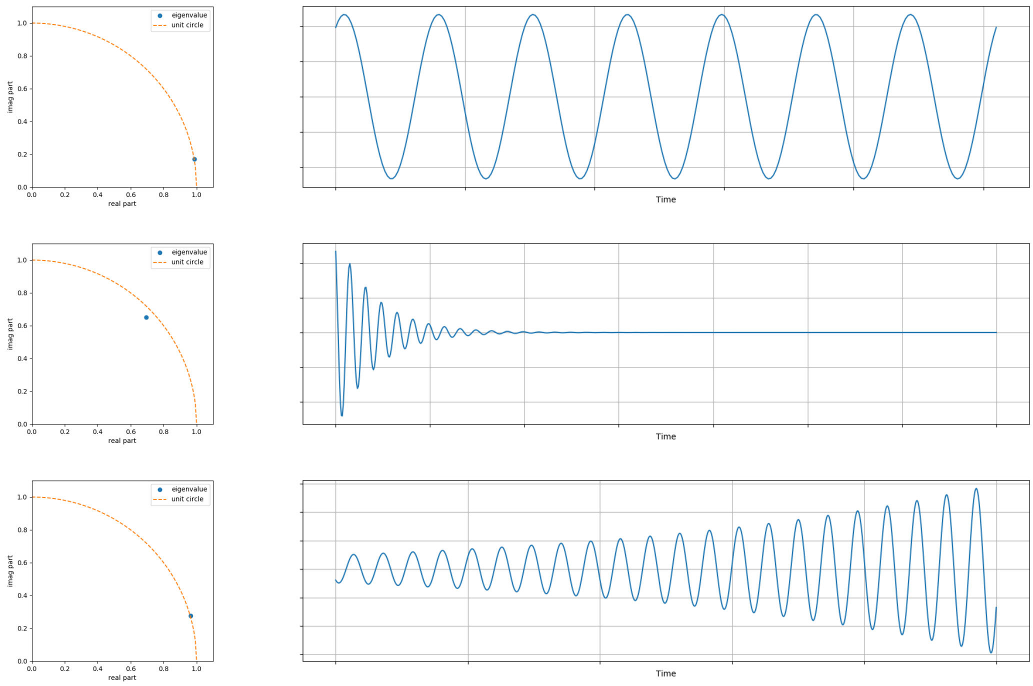


Figure 5.2: On the left column the eigenvalue position with respect to the unit circle, on the right column the corresponding DMD mode. From top to bottom we have a stable, a convergent, and a divergent mode, corresponding to an eigenvalue on the unit circle, inside it, and outside it, respectively.

Finally, to compute the future state prediction for every time, we use

$$x(t) \approx \sum_{k=1}^{r} \phi_k e^{\psi_k t} b_k = \Phi e^{\Psi t} b, \tag{5.18}$$

where $\Psi = \text{diag}(\psi_k) = \text{diag}(\ln(\lambda_k)/\Delta t)$, and b_k denotes the initial amplitude of the corresponding k-th DMD modes ϕ_k. To compute such amplitudes we use the best-fit in the least-squares sense:

$$b = \Phi^{\dagger} x_1, \tag{5.19}$$

where we use the first snapshot x_1. We remark that this can be done with respect to every snapshot, and of course it will improve the accuracy in the neighborhood of that specific time instant. The amplitudes can also be computed by minimizing the error over all the snapshots [134].

Part II

OPTIMIZATION METHODS

6

GENETIC ALGORITHM ENHANCED BY ACTIVE SUBSPACES

In this chapter we present an extension of the Genetic Algorithm (GA) which exploits the active subspace of the function to minimize to evolve the individuals on a lower dimensional space. In many cases, GA requires in fact more function evaluations than other optimization methods to converge to the global optimum. Thus, complex and high-dimensional functions can result extremely demanding (from the computational point of view) to be optimized with the standard algorithm. To address this issue, we propose to linearly map the input parameter space of the original function onto its AS before the evolution, performing the *mutation* and *mate* processes in a lower dimensional space. Here, we describe the novel method called Active Subspaces Genetic Algorithm (ASGA), presenting differences and similarities with the standard GA method. We test the proposed method over n-dimensional benchmark functions — *Rosenbrock*, *Ackley*, *Bohachevsky*, *Rastrigin*, *Schaffer N. 7*, and *Zakharov* — and finally we apply it to an aeronautical shape optimization problem. All the results presented in this chapter appeared in [73]. An application of such method in a naval engineering context can be found in chapter 8.

6.1 LITERATURE REVIEW

GA is a well-known and widespread methodology, mainly adopted in optimization problems [120, 144]. It emulates the evolutive process of natural selection by following an iterative process where the individuals are selected by a given objective function and subsequently they mutate and reproduce [5, 46, 82, 151]. This gradient-free technique is particular effective when the objective function contains many local minima: thanks to the stochastic component, GA explores the domain without being blocked into local minima. The main disadvantages of such algorithm is the (relative) high number of required evaluations of the objective function during the evolution to explore the input space [190], that makes in several industrial and engineering contexts this method unfeasible for the global computational cost.

In this work, we propose a novel extension of standard GA, exploiting the emerging Active Subspaces property [54, 57] for the dimensionality reduction. AS is a supervised learning technique which allows the approximation of a scalar function with a lower dimensional one, whose parameters are a linear combination of the original inputs. AS has been successfully employed in naval engineering applications [267, 268, 270, 272], coupled with reduced order methods such as POD-Galerkin in biomedical applications [226, 266], POD with interpolation [72] in structural and CFD analysis, and Dynamic Mode Decomposition [271] in CFD contexts. Other applications include aerodynamic shape optimization [168], artificial neural networks to reduce the number of neurons [60], non-linear structural analysis [106], and AS for multivariate vector-valued model functions [291]. Several non-linear AS extension have been proposed recently. We mention Active Manifold [35], Kernel-based Active Subspaces [221] which exploits the random Fourier features to map the inputs in a higher dimensional space. We also mention the application of artificial neural networks for non-linear reduction in parameter spaces by learning isosurfaces [294]. Despite these new non-linear extensions of AS, in this work we exploit the classical linear version because of the possibility to map points in the reduced space onto the original parameter space.

The main idea of the proposed algorithm is to force the individuals of the population to evolve along the AS, which has a lower dimension, avoiding evolution along the meaningless directions. Further, the high number of function evaluations that characterize the GA is exploited within this new approach for the construction (and refinement) of the AS, making these techniques — GA produces a large dataset of input-output pairs, whereas AS needs large datasets for an accurate subspace identification — particularly suited together. This new method has the potential to improve existing optimization pipeline involving both input and model order reduction.

A similar approach has been proposed in [48], where an active subspace is constructed in order to obtain an efficient and adaptive sampling strategy in a evolution strategy framework. This approach shares with the one we are proposing the idea of efficiently exploring the input space by constructing a subspace based on the collected data. In contrast with our approach in [48] the subspace construction is done with a singular value decomposition based method, and the optimization technique is completely different, even if evolution strategy methods and genetic algorithm present some analogies. To the best of the authors knowledge, the current contribution presents a novel approach, not yet explored in the literature. For similar approach, we cite also random subspace embeddings for unconstrained global optimization of functions with low effective dimensionality that can be found in [45, 283], while for evolutionary methods and derivative-free optimization we mention [210, 233], respectively. For a survey on linear dimensionality reduction in the context of optimization programs over matrix manifolds we mention [61].

The outline of this chapter is the following: the proposed method is described in section 6.3, while section 6.2 is devoted to recall the general family of genetic algorithms. Section 6.4 presents the numerical results obtained applying the proposed extension to some popular benchmark functions for optimization problems, then to a typical engineering problem where the shape of a NACA airfoil is morphed to maximize the lift-to-drag coefficient. Finally section 6.5 summarizes the benefits of the method and proposes some extensions for future developments.

6.2 GENETIC ALGORITHMS

In this work we propose an extension of the standard GA. We start recalling the general method in order to easily let the reader understand the differences. We define GA as the family of computational methods that are inspired by Darwin's theory of evolution. The basic idea is to generate a population of individuals with random genes, and make them evolve through mutations and crossovers, mimicking the evolution of living beings. Iterating this process by selecting at each step the best-fit individuals results in the optimization — according to a specific objective function — of the original population. As such this method can be easily adopted as a global optimization algorithm.

Initially proposed by Holland in [119], GA has had several modifications during the years (see for example [78, 80, 81, 144, 251]), but it keeps its fundamental steps: *selection*, *mutation* and *mate*.

Let us define formally the individuals: a population composed by N individuals $\mathbf{x}_i \in \mathbb{R}^P$ with P genes is defined as $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$. We express the fitnesses of such individuals with the scalar function $f : \mathbb{R}^P \to \mathbb{R}$. The first generation $\mathbf{X}^1$ is randomly created — with possible constraints — and the fitness is evaluated for all the individuals: $\mathbf{y}_i = f(\mathbf{x}_i)$ for $i = 1, \dots, N$. Then the following iterative process starts:

SELECTION: The best individuals of the previous generation $\mathbf{X}^i$ are chosen accordingly to their fitnesses to breed the new generation. For the selection, several strategies can be adopted depending on the problem and on the cardinality of the population N.

MATE: The selected individuals are grouped into pairs and, according to a mate probability, they combine their genes to create new individuals. The process, also called *crossover*, emulates the species reproduction. These individuals form the new generation $\mathbf{X}^{i+1}$. An example of a crossover method is sketched in figure 6.1.

MUTATION: The individuals evolve by changing some of their genes. The mutation of an individual is usually controlled by a mutation probability. In figure 6.2 we show an illustrative example where two genes have randomly mutated.

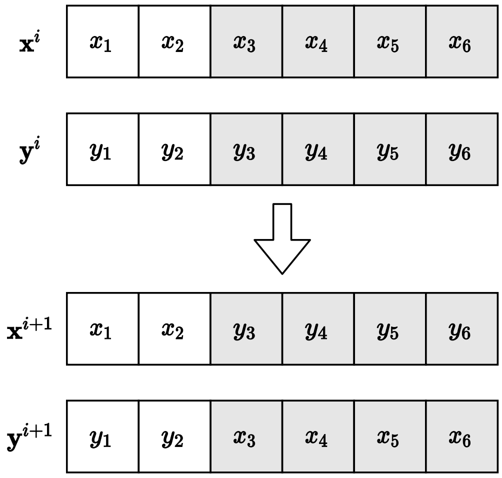

$\mathbf{x}^i$ x_1 x_2 x_3 x_4 x_5 x_6
$\mathbf{y}^i$ y_1 y_2 y_3 y_4 y_5 y_6
$\mathbf{x}^{i+1}$ x_1 x_2 y_3 y_4 y_5 y_6
$\mathbf{y}^{i+1}$ y_1 y_2 x_3 x_4 x_5 x_6

Figure 6.1: Graphical example of mate by crossover. $\mathbf{x}^i$ and $\mathbf{y}^i$ indicate two generic individuals of the i-th generation.

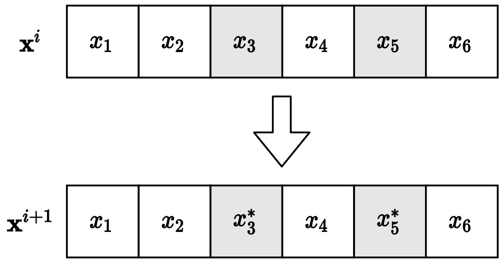

$\mathbf{x}^i$ x_1 x_2 x_3 x_4 x_5 x_6
$\mathbf{x}^{i+1}$ x_1 x_2 x_3^* x_4 x_5^* x_6

Figure 6.2: Graphical example of mutation. $\mathbf{x}^i$ indicates a generic individuals of the i-th generation.

After the mutation step, the fitness of the new individuals is computed and the algorithm restarts with the selection of the best-fit individuals. In this way, the population evolves, generation after generation, towards the optimal individual, avoiding getting blocked in a local minima thanks to the stochastic component introduced by mutation and crossover. Thus, this method is very effective for global optimization where the objective function is potentially non-linear, while standard gradient-based methods can converge to local minima. However GA usually requires an high number of evaluations to perform the optimization, making this procedure very expensive in the case of computational costly objective functions.

6.3 THE PROPOSED ASGA OPTIMIZATION ALGORITHM

In this section we are going to describe the proposed active subspaces extension of the standard GA, named ASGA. Before starting, we emphasize that in what follows, we will maintain the selection, mutation and mate procedures — presented in section 6.2 — as general as possible, without going into technical details, given the large variety of different options for these steps. In fact the proposed extension is independent on the chosen evolution strategies, and we only perform them in a lower dimension exploiting AS. In algorithm 14 we summarize the standard approach, while in algorithm 15 we highlight the differences introduced by ASGA. We also present an illustration for both the methods in figure 6.3, where the yellow boxes indicate the main steps peculiar to ASGA. In both cases, the first step is the generation of the random individuals composing the initial population, and the sequential evaluation of all of them. For ASGA these individuals and their fitness are stored into two additional sets, $\mathbf{X}^{\mathrm{AS}}$ for the individuals, and $\mathbf{y}^{\mathrm{AS}}$ for the fitness. We will exploit them as input-output pair for the construction of the AS. After the selection of the best-fit individuals, the active subspace of dimension M is built and the selected offspring is projected onto it. Let $\mathbf{W}_1$ be the eigenvectors defining the active subspace of the current population, say $\mathbf{X}^i$. We project its best individuals onto the current active subspace with

$$\mathbf{x}_k^{i\,\mathrm{AS}} = \mathbf{W}_1^T \mathbf{x}_k^i, \qquad \forall k \in [1, \dots, N], \tag{6.1}$$

where $\mathbf{x}_k^{i\,\mathrm{AS}}$ is the reduced individual of the i-th population.

The low-dimensional individuals mate and mutate in the active subspace. Thanks to the reduced dimension and to the fact that we retain only the most important dimensions, these operations are much more efficient. Thus, even if the AS of dimension M does not provide an accurate approximation of the original full-dimensional space, the active dimensions will provide preferential directions for the evolution, making the iterative process smarter and faster.

After the evolution, the low-dimensional offspring is mapped back to the original space. In section 2.2.1 we described how for any point in the active subspace we can find several points in the original space which are mapped onto it. So we select, for any individual in the offspring, B full-dimensional points which correspond to the individual in the active subspace by sampling the inactive variable $\boldsymbol{\eta}$, such that

$$\mathbf{x}_k^{i+1} = \mathbf{W}_1 \mathbf{x}_k^{i\,\mathrm{AS}} + \mathbf{W}_2 \boldsymbol{\eta}, \qquad \text{with } -\mathbf{1} \leq \mathbf{x}_k^{i+1} \leq \mathbf{1}, \qquad \forall k \in [1, \dots, N], \tag{6.2}$$

where $\mathbf{1}$ denotes a vector with all components equal to 1 — the original parameters are usually rescaled in $[-1, 1]^P$ before applying AS —. We emphasize that to preserve the same dimensionality of the offspring between the original GA and the AS extension, in the proposed algorithm we select the $\frac{N}{B}$ best individuals, instead of selecting N. In this way, after the back-mapping, the offspring has dimension N in both versions. The number of back-mapped points B, and the active subspace dimension M — that can be a fixed parameter or dynamically selected from the spectral gap of the covariance matrix $\mathbf{C}$ — represent the new (hyper-)parameters of the proposed method.

Finally the fitness of the new individuals, now in the full-dimensional space, are evaluated. To make the AS more precise during the iterations, the evaluated individuals and their fitness are added to $\mathbf{X}^{\mathrm{AS}}$ and $\mathbf{y}^{\mathrm{AS}}$. The process restarts from the selection of the offspring from the new generation, continuing as described above until the stopping criteria are met.

We stress that the structure of the algorithm is similar to the original GA approach, with the difference that the gradients at the sample points are approximated in

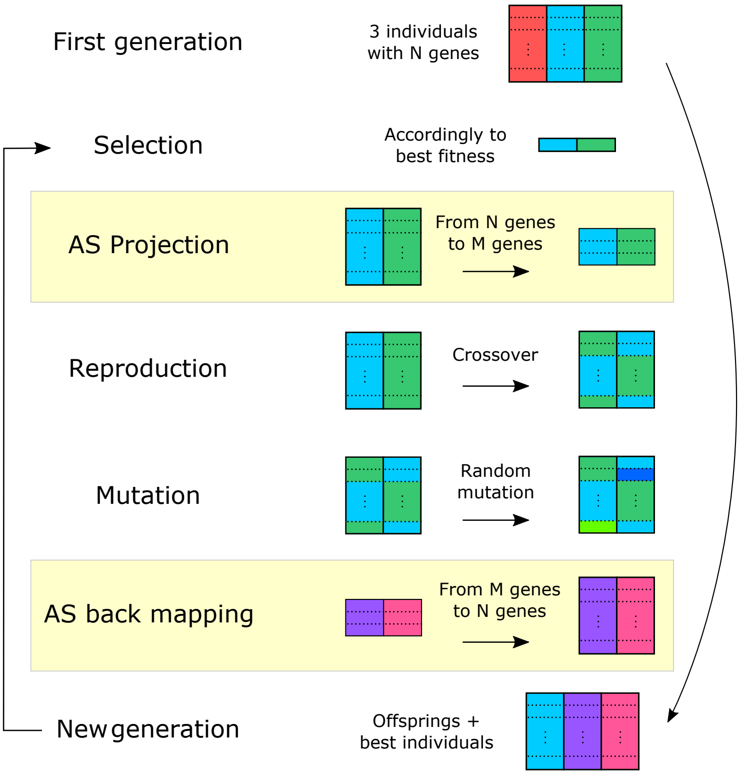


Figure 6.3: Active subspaces-based genetic algorithm scheme. The main step of the classical GA are depicted from top to bottom. The yellow boxes represent projections onto and from lower dimension active subspace, which are specific to ASGA.

order to identify the dimensions with highest variance. Even if such information about the function gradient is used, the ASGA method is different from gradient-based methods: numerically computing the gradient with a good accuracy at a specific point — that is the fundamental step of gradient-based methods to move on the solution manifold — is a very expensive procedure, especially in a high-dimensional space. In ASGA we avoid such computation, exploiting instead the already collected function evaluations. Further, gradient-based techniques converges (relatively) fast to optimum, but they get blocked into local minima, contrarily to the ASGA approach. It is important to remark that, for each generation, the AS is rebuilt from scratch, losing efficiency but gaining more precision due to the growing number of elements in the two sets $\mathbf{X}^{\text{AS}}$ and $\mathbf{y}^{\text{AS}}$. We also remark the samples are generated with a uniform distribution only at the first generation. After that, due to the ASGA steps the distribution changes in a way which can not be known a priori. For the computation of the expectation operator in equation (2.8) in this work we assume a uniform distribution. Even if this may introduce an unknown error, the numerical results achieved by ASGA seem to support such choice. Of course the numerical estimates present in the literature for the uniform distribution do not apply in such case. This method can be viewed as an active learning procedure in a Bayesian integration context, where the maximized acquisition function is heuristic and given by the application of AS and GA steps. Another interpretation is that we are enriching the local informations near the current minimum to feed the AS algorithm, so it can be viewed as a weighted AS.

Algorithm 14 Standard GA.

```
Input:
    initial population size $N_0$
    population size $N$
    selection routine SELECT
    mutation routine MUTATE
    mate routine MATE
    objective function FOBJ
    stop criteria
Output:
    final population $\mathbf{X}^{\text{end}}$

procedure GENETICALGORITHM
    $g \leftarrow 0$
    $\mathbf{X}^g \leftarrow$ random pop of size $N_0$
    $\mathbf{y}^g \leftarrow$ FOBJ$(\mathbf{X}^g)$
    repeat
        $g \leftarrow g + 1$
        $\mathbf{X}^* \leftarrow$ SELECT$(\mathbf{X}^{g-1}, \mathbf{y}^{g-1}, N)$
        $\mathbf{X}^* \leftarrow$ MATE$(\mathbf{X}^*)$
        $\mathbf{X}^g \leftarrow$ MUTATE$(\mathbf{X}^*)$
        $\mathbf{y}^g \leftarrow$ FOBJ$(\mathbf{X}^g)$
    until stop criteria reached
    $\mathbf{X}^{\text{end}} \leftarrow \mathbf{X}^g$
    return $\mathbf{X}^{\text{end}}$
end procedure
```

Algorithm 15 Proposed ASGA.

```
Input:
    initial population size $N_0$
    population size $N$
    active dimension $M$
    number backward $B$
    selection routine SELECT
    mutation routine MUTATE
    mate routine MATE
    objective function FOBJ
    stop criteria
Output:
    final population $\mathbf{X}^{\text{end}}$

procedure ASGA
    $g \leftarrow 0$
    $\mathbf{X}^g \leftarrow$ random pop of size $N_0$
    $\mathbf{y}^g \leftarrow$ FOBJ$(\mathbf{X}^g)$
    $\mathbf{X}^{\text{AS}} \leftarrow \mathbf{X}^g$
    $\mathbf{y}^{\text{AS}} \leftarrow \mathbf{y}^g$
    repeat
        $g \leftarrow g + 1$
        $\mathbf{X}^* \leftarrow$ SELECT$(\mathbf{X}^{g-1}, \mathbf{y}^{g-1}, \frac{N}{B})$
        build AS$(\mathbf{X}^{\text{AS}}, \mathbf{y}^{\text{AS}}, M)$
        $\mathbf{X}^*_M \leftarrow$ FORWARD$(\mathbf{X}^*)$
        $\mathbf{X}^*_M \leftarrow$ MATE$(\mathbf{X}^*_M)$
        $\mathbf{X}^*_M \leftarrow$ MUTATE$(\mathbf{X}^*_M)$
        for $\mathbf{x}$ in $\mathbf{X}^*_M$ do
            for $i \leftarrow 1$ to $B$ do
                $\mathbf{X}^g \leftarrow$ BACKWARD$(\mathbf{x})$
            end for
        end for
        $\mathbf{y}^g \leftarrow$ FOBJ$(\mathbf{X}^g)$
        $\mathbf{X}^{\text{AS}} \leftarrow \mathbf{X}^{\text{AS}} \cup \mathbf{X}^g$
        $\mathbf{y}^{\text{AS}} \leftarrow \mathbf{y}^{\text{AS}} \cup \mathbf{y}^g$
    until stop criteria reached
    $\mathbf{X}^{\text{end}} \leftarrow \mathbf{X}^g$
    return $\mathbf{X}^{\text{end}}$
end procedure
```

6.4 NUMERICAL RESULTS

In this section we are going to present the results obtained by applying the proposed algorithm, firstly to some test functions that are usually used as benchmarks for optimization problems. Since this method is particularly suited for high-dimensional functions, we analyze the optimization convergence for three different input dimension (2, 15, and 40), i.e. the number of genes of each individual. The second test case we propose is instead a typical engineering problem, where we optimize the lift-to-drag coefficient of a NACA airfoil which is deformed using a map $\mathcal{M} : \mathbb{R}^{10} \to \mathbb{R}$ defined in section 6.4.2. In this example we opted for the use of a surrogate model only to evaluate the individuals' fitness for computational considerations, since we just want to compare ASGA with GA. We do not rely on the surrogate for the gradients approximation. In [69], instead, we apply ASGA on a naval engineering hydrodynamics problem, where we do not rely on a surrogate model of the target function, but instead we exploit data-driven model order reduction methods to reconstruct the fields of interest and then compute the function to optimize. All the details for this naval engineering application can be also found in chapter 8.

In both the test cases, in order to collect a fair comparison, we adopt the same routines for the selection, the mutation and the crossover steps. In particular:

- for the *mate* we use the blend BLX-alpha crossover [85] with $\alpha = 1.0$, with a mate probability of 50%. With this method, the offspring results:

  $$\begin{cases} \mathbf{x}_a^i = (1-\gamma)\mathbf{x}_a^{i-1} + \gamma \mathbf{x}_b^{i-1} \\ \mathbf{x}_b^i = \gamma \mathbf{x}_a^{i-1} + (1-\gamma)\mathbf{x}_b^{i-1} \end{cases} \qquad \text{for } a, b = 1, \dots, N, \tag{6.3}$$

  where $\mathbf{x}_a^{i-1}$ and $\mathbf{x}_b^{i-1}$ refer to the parent individuals (at the $i-1$th generation), $\mathbf{x}_a^i$ and $\mathbf{x}_b^i$ are the mated individuals, N is the cardinality of population, and γ is a random variable chosen in the interval $[-\alpha, 1+\alpha)$. We mention that equation (6.3) can recover the graphical description of mating in figure 6.1 if γ is taken to be discrete, either 0 or 1, and applied component-wise.

- for the *mutation*, a Gaussian operator [117] has been used with a mutation probability of 50%. This strategy changes genes by adding a normal noise. Since we do not have any knowledge about the low-dimensional space, tuning the variance of such mutation may result in a not trivial procedure. This quantity has in fact to be set in order to explore the input space but, as the same time, producing minimal differences between parents and offspring. A fixed variance for both the spaces may cause a too big distance — in l_2 sense — between parents and offspring, inhibiting the convergence. To overcome this potential problem, we correlate the Gaussian variance with the genes theirselves, ensuring a reasonable mutation in both spaces. The adopted mutation method is:

  $$\mathbf{x}_a^i = \mathbf{x}_a^{i-1} + \varepsilon\, \mathbf{x}_a^{i-1} \qquad \text{for } a = 1, \dots, N, \tag{6.4}$$

  where ε is a random variable with probability distribution $\mathcal{N}(\mu, \sigma^2)$, that is $\varepsilon \sim \mathcal{N}(\mu, \sigma^2)$, with $\mu = 0$ and $\sigma^2 = 0.1$.

Regarding the *selection*, since the limited number of individuals per population, we adopt one of the simplest criteria, by selecting the N best individuals in terms of fitness.

We also keep fixed the additional parameters for the AS extension: the number of active dimensions M is set to 1, while the number of back-mapped points is 2. All the gradients computations are done using local linear models. For the actual computation regarding AS we used the ATHENA[1] Python package [222]. The only varying parameters are the size of the initial population N_0, the size of population during the evolution N, and the number of generations in the evolutive loop, which are chosen empirically based on the objective function. We emphasize that, due to the stochastic nature of these methods, we repeated the tests 15 times, with different initial configurations, presenting the mean value, the minimum and the maximum over the 15 runs.

[1] Freely available at `https://github.com/mathLab/ATHENA`.

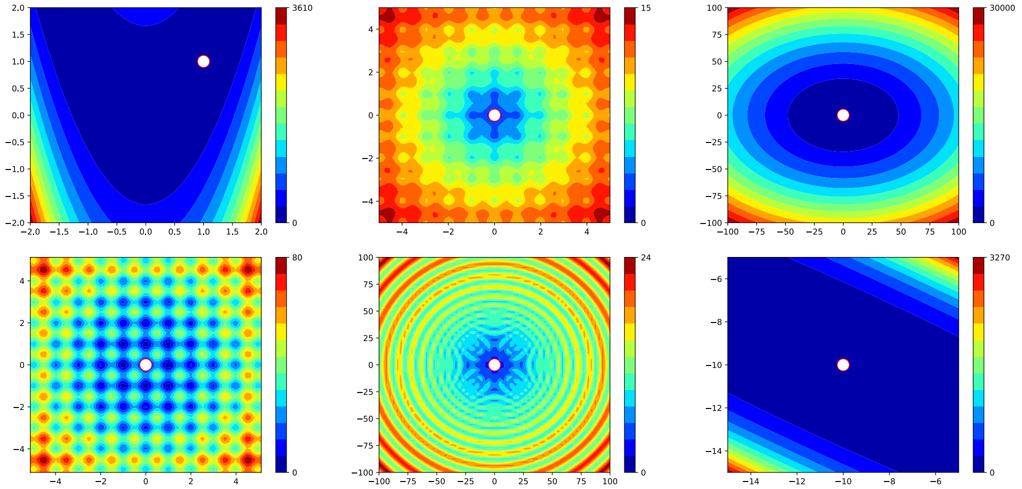


Figure 6.4: Benchmark test functions representation in 2D. White dots indicate the global minima. In the top row from left to right we have the following functions: Rosenbrock, Ackley, and Bohachevsky. In the bottom row we have the Rastrigin, Schaffer N. 7, and Zakharov functions.

6.4.1 *Benchmark test functions*

We applied the optimization algorithm to 6 different n-dimensional test functions, which have been chosen to cover a large variety of possible shapes. For all the functions, the results of the proposed method are compared to the results obtained using the standard genetic approach. In detail, the functions we tested are the so called: *Rosenbrock*, *Ackley*, *Bohachevsky*, *Rastrigin*, *Schaffer N. 7* and *Zakharov*. In figure 6.4 we depict the test functions, in their two-dimensional form. In the following paragraphs we briefly introduce them before presenting the obtained results. For a complete literature survey on benchmark functions for global optimization problems we suggest [128].

6.4.1.1 *(a) Rosenbrock function*

The Rosenbrock function is a widespread test function in the context of global optimization [18, 76, 204]. We choose it as representative of the valley-shaped test functions. The general d-dimensional formulation is the following:

$$f(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]. \tag{6.5}$$

Its global minimum is $f(x^*) = 0$, at $x^* = (1, 1, \dots, 1)$. As we can see from figure 6.4 (first in the top row) the minimum lies on a easy to find parabolic valley, but the convergence to the actual minimum is notoriously difficult. We evaluated the function in the hypercube $[-5, 10]^d$.

6.4.1.2 *(b) Ackley function*

The Ackley function is characterized by many local minima making it difficult to find the global minimum, especially for hillclimbing algorithms [2, 9]. The general d-dimensional formulation is the following:

$$f(x) = -a \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\sqrt{\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)}\right) + a + \exp(1), \quad (6.6)$$

where a, b, and c are set to 20, 0.2, and 2π, respectively. Its global minimum is $f(x^*) = 0$, at $x^* = (0, 0, \dots, 0)$. As we can see from figure 6.4 (second in the top row) the function is nearly flat in the outer region, with many local minima, and the global minimum lies on a hole around the origin. The function has been evaluated in the domain $[-15, 30]^d$.

6.4.1.3 *(c) Bohachevsky function*

The Bohachevsky function is a representative of the bowl-shaped functions. There are many variants and we chose the general d-dimensional formulation as the following:

$$f(x) = \sum_{i=1}^{d-1}(x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7). \quad (6.7)$$

Its global minimum is $f(x^*) = 0$, at $x^* = (0, 0, \dots, 0)$. As we can see from figure 6.4 (third in the top row) the function has a clear bowl shape. This function has been evaluated in the domain $[-100, 100]^d$.

6.4.1.4 *(d) Rastrigin function*

The Rastrigin function is another difficult function to deal with for global optimization with genetic algorithm due to the large search space and its many local minima [186]. The general d-dimensional formulation is the following:

$$f(x) = 10d + \sum_{i=1}^{d}[x_i^2 - 10\cos(2\pi x_i)]. \quad (6.8)$$

Its global minimum is $f(x^*) = 0$, at $x^* = (0, 0, \dots, 0)$. As we can see from figure 6.4 (first in the bottom row) the function is highly multimodal with local minima regularly distributed. We evaluated this function in the input domain $[-5.12, 5.12]^d$.

6.4.1.5 *(e) Schaffer N. 7 function*

The Schaffer N. 7 function [234] is a stretched V sine wave. The general d-dimensional formulation is the following:

$$f(x) = \sum_{i=1}^{d-1}(x_i^2 + x_{i+1}^2)^{0.25}\left[\sin^2(50(x_i^2 + x_{i+1}^2)^{0.10}) + 1\right]. \quad (6.9)$$

Its global minimum is $f(x^*) = 0$, at $x^* = (0, 0, \dots, 0)$. As we can see from figure 6.4 (second in the bottom row) the function presents many local minima. The optimization has been performed in the hypercube $[-100, 100]^d$.

6.4.1.6 *(f) Zakharov function*

The Zakharov function is a representative of the plate-shaped functions. It has one global minimum and no additional local minima. The general d-dimensional formulation is the following (after a shift):

$$f(x) = \sum_{i=1}^{d}(x_i + 10)^2 + \left(\sum_{i=1}^{d}\frac{i}{2}(x_i + 10)\right)^2 + \left(\sum_{i=1}^{d}\frac{i}{2}(x_i + 10)\right)^4. \tag{6.10}$$

We emphasize that we used a shifted version with global minimum $f(x^*) = 0$, at $x^* = (-10, -10, \dots, -10)$. This choice is made to prove that the proposed method is not biased towards minima around the origin. We can see from figure 6.4 (third in the bottom row) the function for $d = 2$. We evaluated the Zakharov function in the domain $[-15, 0]^d$.

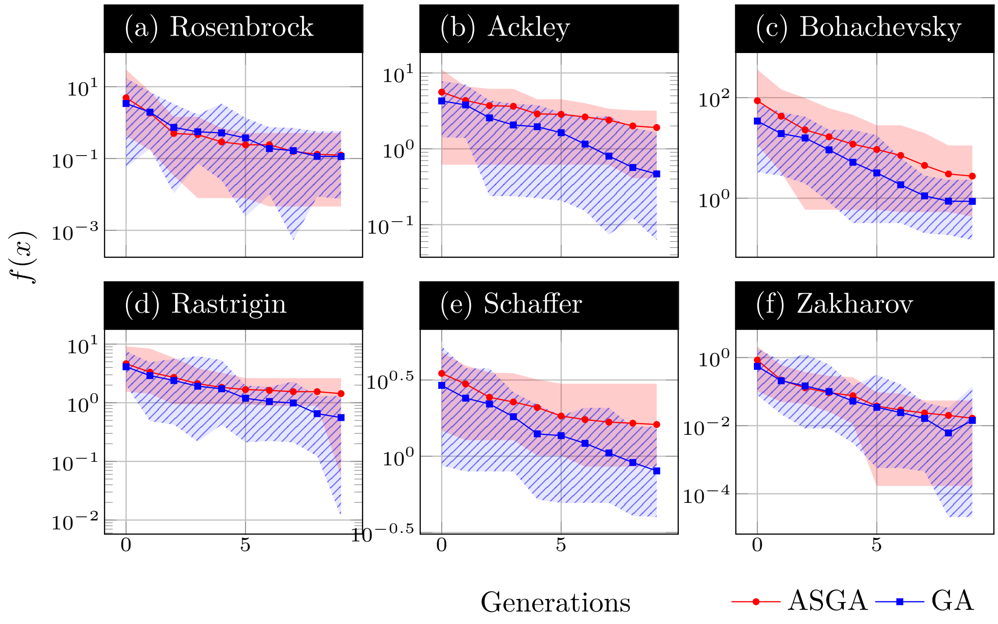


Figure 6.5: Results of the optimization of the benchmark functions in a space of dimension $d = 2$. We compare the standard GA (in blue square dots) with the proposed algorithm ASGA (in red circle dots) using an initial population of size 30, while the dimension for each generation is fixed to 10. The solid lines represent the mean, over 15 runs, of the objective function corresponding to the best individual at each generation. The shaded areas show the interval between minimum and maximum (blue with lines for GA, red for ASGA).

6.4.1.7 *Comments*

All the test cases presented share the same hyper-parameters described at the beginning of this section, except for the population size. For the 2-dimensional benchmark functions, the two algorithms are tested creating $N_0 = 200$ random individuals for the initial population, then keeping an offspring of dimension $N = 100$. Figure 6.5 shows the behaviour for all the test functions. For this space dimension, the two trends are very similar: the usage of the proposed algorithm does not make the optimization faster, and adds the computational overhead for the AS construction. Despite that, the results after 10 generations are very similar, and we can consider this as a worst case scenario, where a clear reduction in the parameter space is not possible.

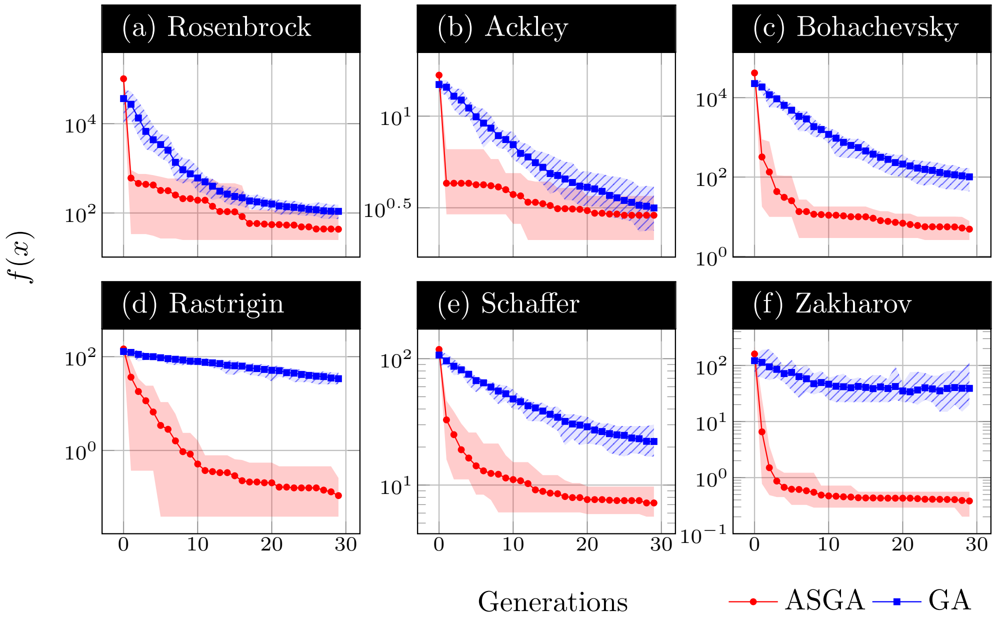


Figure 6.6: Results of the optimization of the benchmark functions in a space of dimension $d = 15$. We compare the standard GA (in blue square dots) with the proposed algorithm ASGA (in red circle dots) using an initial population of size 2000, while the dimension for each generation is fixed to 200. The solid lines represent the mean, over 15 runs, of the objective function corresponding to the best individual at each generation. The shaded areas show the interval between minimum and maximum (blue with lines for GA, red for ASGA).

The ASGA performance gain changes drastically increasing the number of dimension to $d = 15$, as demonstrated in figure 6.6. For such dimension, the two parameters N_0 and N are set to 2000 and 200, respectively. Starting from this dimension, it is possible to note a remarkable difference between the standard method and the proposed one. The greater the input dimension, the greater the gain produced by ASGA, due to the exploitation of the AS reduction. All the benchmarks show a faster decay, but we can isolate two different patterns in the evolution: *Rosenbrock* and *Ackley* show a very steep trend in the first generation gain, while for the next generations the population is not able to decrease its fitness as much as before, showing a quasi-constant behaviour. The difference with the standard genetic algorithm is maximized in the first generation, but even if the evolution using ASGA is not so effective after the first generation in these two cases, the proposed method is able to achieve anyway a better result (on average) after 30 generations. The other benchmarks instead show a much smoother decay, gradually converging to the optimum. Despite the lack of the initial step, for these benchmarks the gain with respect to the standard approach becomes bigger, even if after several generations the convergence rate decreases.

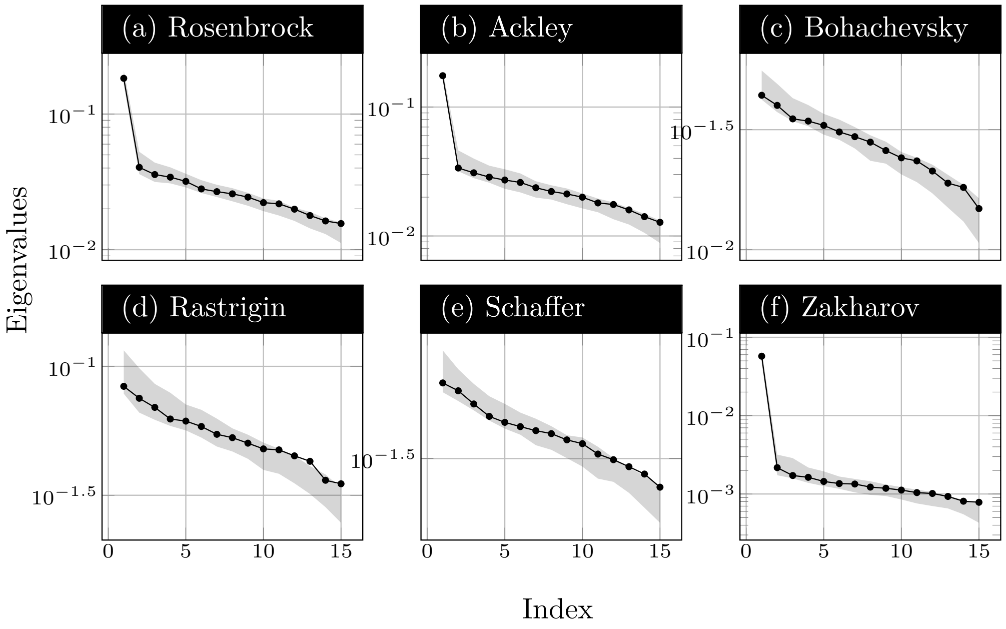


Figure 6.7: Eigenvalues estimates of the matrix **C** in equation (2.8) for all the benchmarks, at the first generation, for $d = 15$. The black dots in the plot indicate the eigenvalues, while the grey area is defined by the bootstrap intervals.

In order to better understand these differences, we investigate the spectra of the AS covariance matrices for all the benchmarks, which are reported in figure 6.7. The patterns individuated in the optimizations are partially reflected in the eigenvalues: *Rosenbrock*, *Ackley* and *Zakharov* have an evident gap between the first and the second eigenvalues, which results in a better approximation (of the original function) in the 1-dimensional subspace. However, the order of magnitude of the first eigenvalue is different between the three functions: for *Rosenbrock* and *Ackley* the magnitude is greater than 0.1 whereas for *Zakharov* is around 0.05.

Since for all the tests, the ASGA approach performs better than its classical counterpart despite the absence, in some cases, of an evident spectral gap in the AS covariance matrix, we perform further investigations. In particular, we use the same tests as before (15-dimensional functions) but with a different number of active dimensions, i. e. $M = \{2, 5\}$, instead of $M = 1$. In figure 6.8 we show the comparison between the classical GA and the ASGA outcomes. It is possible to note that by increasing the active dimension, the differences between the performances of the two methods become smaller. Only for the Rosenbrock and for the Ackley functions we can see that ASGA with $M = 5$ is not able to reach the same order of magnitude reached by GA (we remark the original space has dimension $d = 15$).

Increasing the input dimension to $d = 40$ shows a much clearer benefit in using the proposed method, as we can see in figure 6.9. Here we set $N_0 = 5000$ and $N = 1000$. We specify that we set the active dimension $M = 1$. Also with this dimensionality, we are able to isolate two main behaviours in the convergence of the six benchmarks: a very steep trend in the first generation, and a more smooth

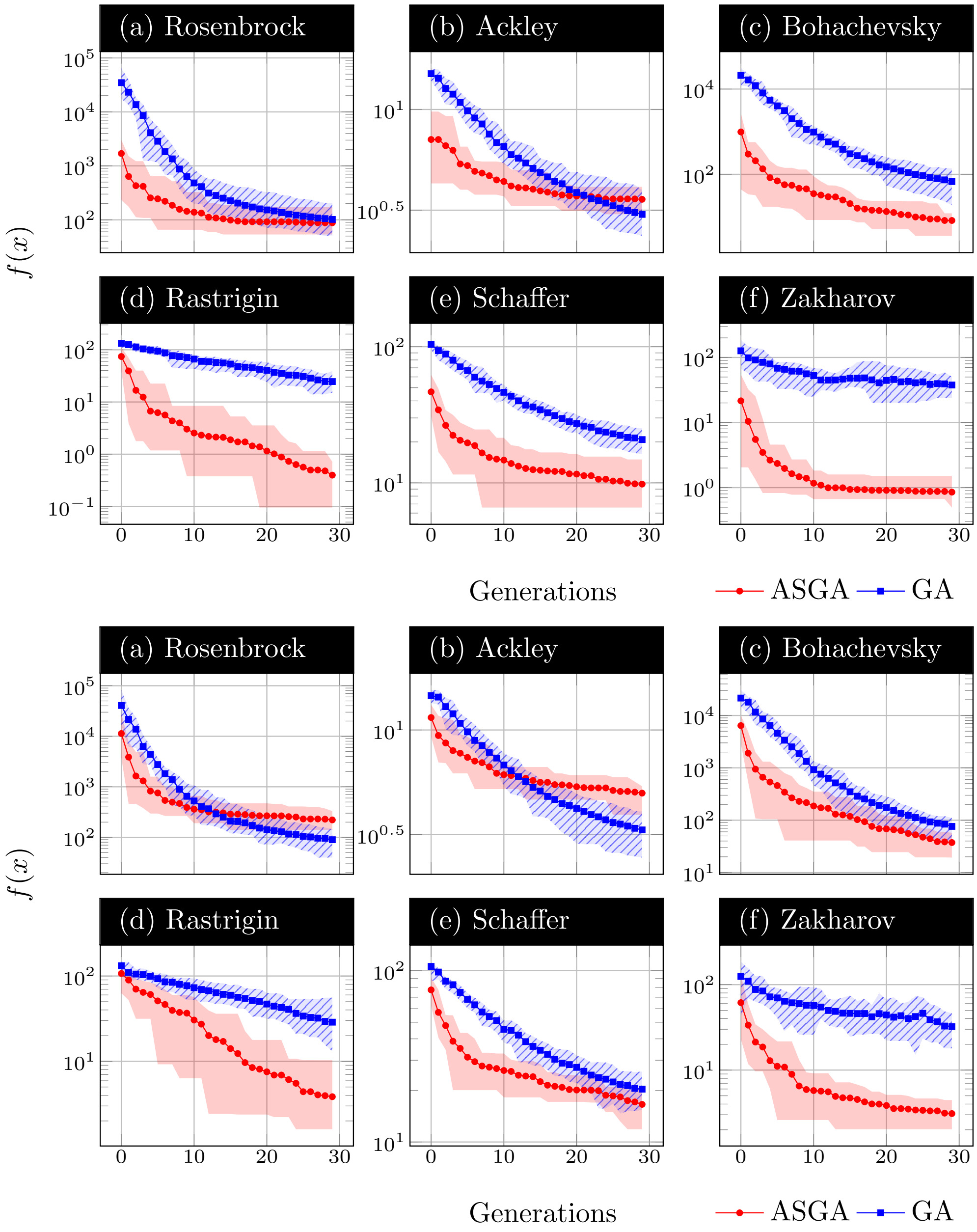


Figure 6.8: Results of the optimization of the benchmark functions in a space of dimension $d = 15$, with active dimension $M = 2$ (top) and $M = 5$ (bottom). We compare the standard GA (in blue square dots) with the proposed algorithm ASGA (in red circle dots) using an initial population of size 2000, while the dimension for each generation is fixed to 200. The solid lines represent the mean, over 15 runs, of the objective function corresponding to the best individual at each generation. The shaded areas show the interval between minimum and maximum (blue with lines for GA, red for ASGA).

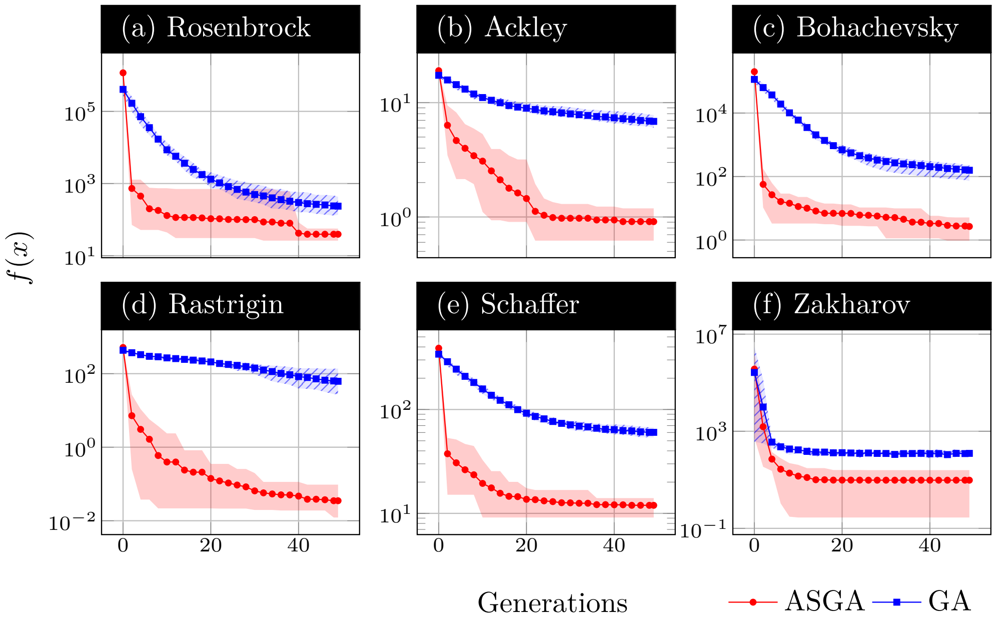


Figure 6.9: Results of the optimization of the benchmark functions in a space of dimension $d = 40$. We compare the standard GA (in blue square dots) with the proposed algorithm ASGA (in red circle dots) using an initial population of size 5000, while the dimension for each generation is fixed to 1000. The solid lines represent the mean, over 15 runs, of the objective function corresponding to the best individual at each generation. The shaded areas show the interval between minimum and maximum (blue with lines for GA, red for ASGA).

one, but still equally effective. The interesting thing is that some benchmarks do not reflect the behaviour collected with $d = 15$. While *Rosenbrock*, *Rastrigin* and *Zakharov* show a similar convergence rate for ASGA, the other benchmarks present a change in the slope. The different behaviours observed for the same benchmarks evaluated at different input dimensional spaces is due to the fact that the method is sensitive to the approximation accuracy of the gradients of the model function with respect to the input data. This is an issue inherited by the application of AS. Moreover, since we are keeping just one active variable, we are discarding several information, thus the representation of the function along the active subspace could present some noise. So the genetic procedure enhanced by AS is able to converge fast to the optimum, but this optimum may be — for the space simplification — distant to the true optimum. From the tests with higher active dimension, we note that the improvement in the first iterations is not as rapid as by using a 1-dimensional AS. Also, keeping more active dimensions, the performance of ASGA becomes similar to the standard GA. We can conclude that with one (or few) active dimension, ASGA reaches the global minimum with less function evaluations, but we get stuck with the projection error introduced by AS, whereas by increasing the active dimension we reduce the projection error but we lose the effectiveness of the evolution steps in a reduced space. A possible solution for this problem can be a smarter (and dynamical) strategy to select the number of active dimensions.

Over all the three test cases, where we vary the input space dimension, the performance of ASGA are better or equal than the standard GA. In table 6.1 we summarize the relative gain on average achieved after the entire evolution and after only one generation, divided by test function, both with the GA and ASGA methods. The relative gain is computed as the mean over 15 runs of the ratio between the objective function evaluated at the best-fit individual at the beginning of the evolution ($f(x^0_{\text{opt}})$) and the objective function evaluated at the best-fit individual after k generations, with $k = 1$ and $k = N_{\text{gen}}$, where N_{gen} is the maximum number of generations depending on the input dimension. This relative gain $G(k)$ reads,

$$G(k) = \frac{1}{15} \sum_{i=1}^{15} f(x^0_{\text{opt}_i}) / f(x^k_{\text{opt}_i}), \tag{6.11}$$

where $x^k_{\text{opt}_i}$ is the best-fit individual of the population at the i-th run and at k-th generation. Highest values correspond to a more effective optimization, and for dimensions 15 and 40 we can see from table 6.1 that ASGA performs better than standard GA for all the benchmarks. Even the gain after just one evolutive iteration is bigger in all the collected tests, reaching in some cases some order of magnitude of difference with respect to GA. These results suggest that despite the computational overhead for the construction of AS and the back-mapping, an application of the ASGA over the standard GA produces usually better or at least comparable results for a fixed generation.

Table 6.1: Summary comparison between GA and ASGA with respect to the gain $G(k)$ defined in equation (6.11). We compare the gain for the first and for the last generation.

Function	**Method**	**dim = 2**		**dim = 15**		**dim = 40**	
		$G(9)$	$G(1)$	$G(29)$	$G(1)$	$G(49)$	$G(1)$
Ackley	GA	9.17	1.13	4.71	1.03	2.53	1.10
	ASGA	2.93	1.29	5.81	3.89	20.91	3.00
Bohachevsky	GA	39.58	1.78	223.86	1.22	729.05	1.81
	ASGA	31.66	2.04	8608.41	130.72	75104.33	3548.70
Rastrigin	GA	7.34	1.41	3.80	1.05	6.97	1.17
	ASGA	3.24	1.39	1343.41	4.00	14738.40	71.77
Rosenbrock	GA	30.04	1.74	335.33	1.34	1723.89	2.42
	ASGA	39.68	2.66	2343.57	167.48	29747.56	1600.24
Schaffer	GA	3.64	1.21	4.83	1.11	5.66	1.18
	ASGA	2.16	1.17	16.41	3.61	32.57	10.38
Zakharov	GA	38.59	2.65	3.11	1.07	2148.39	26.50
	ASGA	51.14	3.88	417.86	24.46	37739.61	237.48

6.4.1.8 *On the convergence of ASGA*

The aim of this section is to provide further insights about the convergence of the ASGA method. We perform a single run on all the benchmark functions presented above, in a space of dimension $d = 2$. We kept unaltered all the ASGA numerical settings described above. We emphasize that we used the same hyper-parameters of the 2-dimensional optimization test, except for the number of generations which we increased to 100.

We summarize in figure 6.10 and figure 6.11 the spatial coordinates of the best individual after each generation using the standard GA and ASGA. The proposed method reaches the global minimum for all the testcases, performing better than the standard counterpart for the *Rosenbrock* and *Rastrigin* functions.

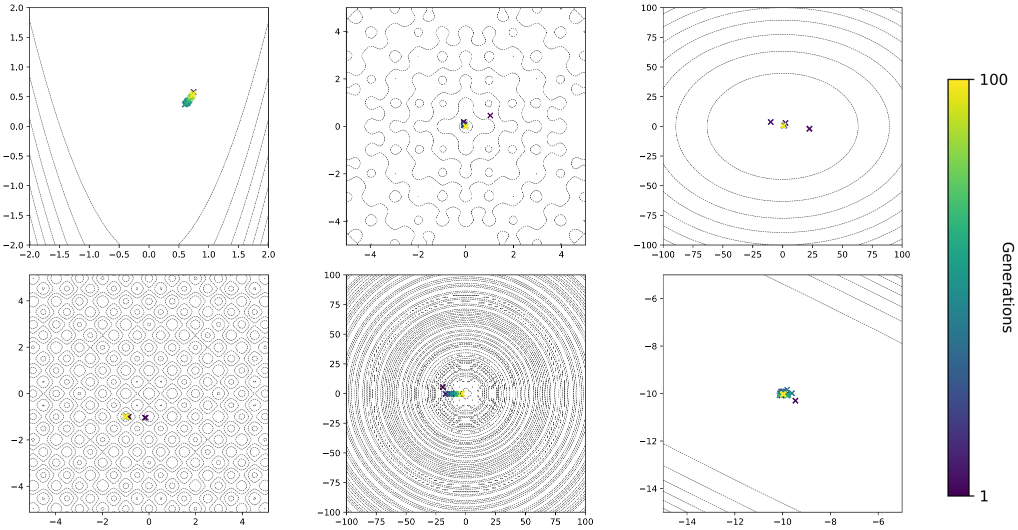


Figure 6.10: Results of the single optimization run using GA for the 2-dimensional benchmark functions. The colored crosses indicate the spatial coordinates of the best individual at each generation. Black lines indicate the isolines of the functions.

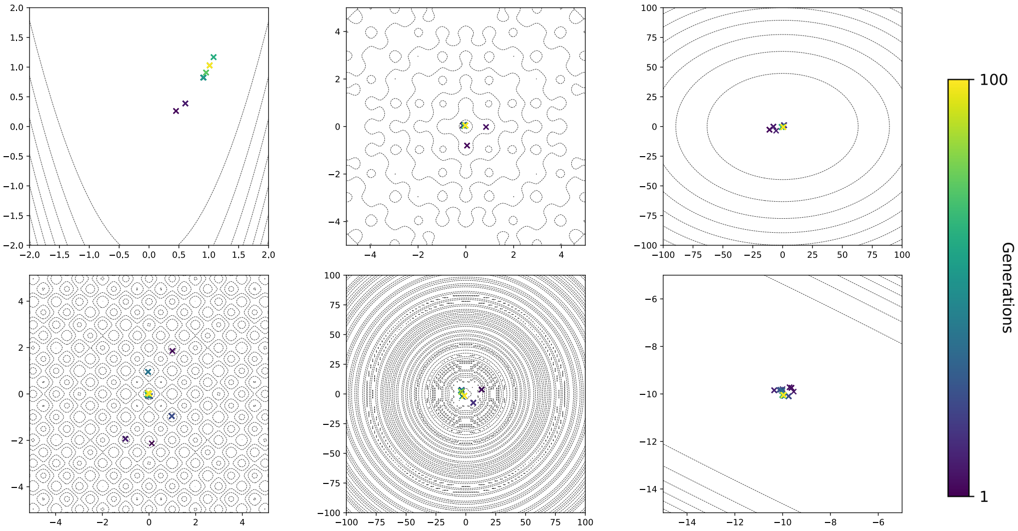


Figure 6.11: Results of the single optimization run using ASGA for the 2-dimensional benchmark functions. The colored crosses indicate the spatial coordinates of the best individual at each generation. Black lines indicate the isolines of the functions.

We also measure the convergence as the Euclidean distance between the best individual fitness and the global optimum, and the *spatial* convergence as the Euclidean distance between the coordinates of the best individual and the coordinates of the optimal point. We kept the same numerical settings, only raising the number of generation to 1000. Figure 6.12 presents the plots where we compare the trend using GA and ASGA: the proposed method shows a better performance, not only thanks to the faster convergence but also because in all the cases ASGA is able to get closer than the GA to the global optimum.

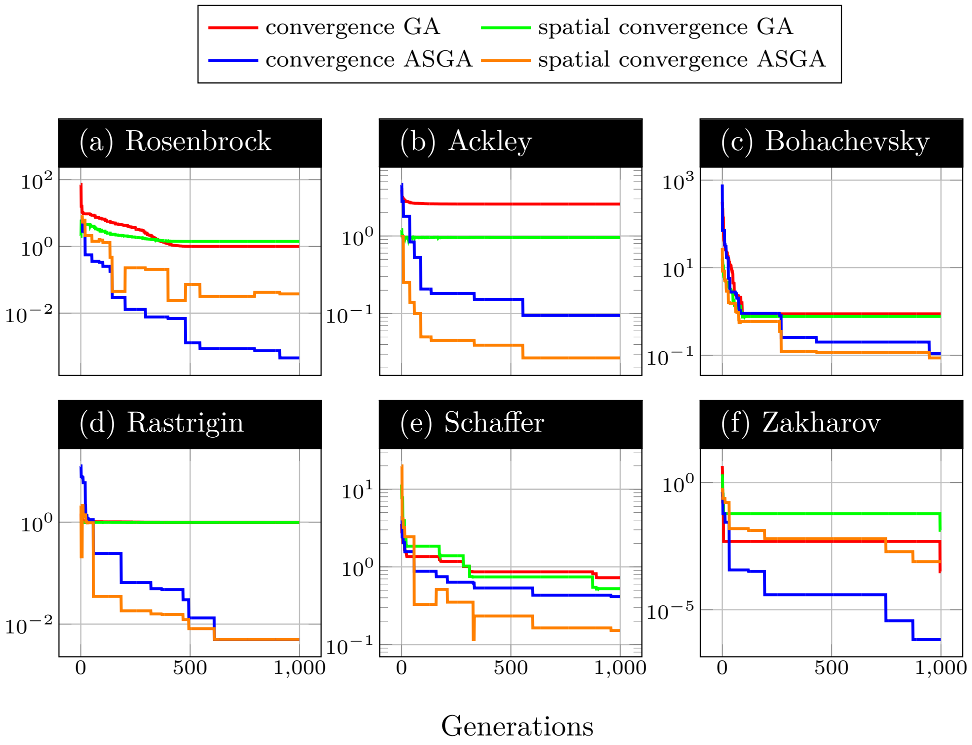


Figure 6.12: Convergence of GA and ASGA for the 2-dimensional benchmark functions.

6.4.2 *Shape design optimization of a NACA airfoil*

Here we present the shape design optimization of a NACA 4412 airfoil [1]. Since the purpose of this work is limited to the extension of the GA, we briefly present the details of the complete model, with a quick overview of the application. To reproduce the full order simulations please refer to [271] and to chapter 10.

Let be given the unsteady incompressible Navier-Stokes equations described in an Eulerian framework on a parametrized space-time domain $Q(\boldsymbol{\mu}) = \Omega \times [0, T] \subset$

$\mathbb{R}^d \times \mathbb{R}^+$, $d = 2, 3$ with the velocity field denoted by $\mathbf{u} : Q(\boldsymbol{\mu}) \to \mathbb{R}^d$, and the pressure field by $p : Q(\boldsymbol{\mu}) \to \mathbb{R}$, such that:

$$\begin{cases} \mathbf{u_t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot 2\nu\nabla^{\mathbf{s}}\mathbf{u} = -\nabla p & \text{in } Q(\boldsymbol{\mu}), \\ \nabla \cdot \mathbf{u} = \mathbf{0} & \text{in } Q(\boldsymbol{\mu}), \\ \mathbf{u}(t, x) = \mathbf{f}(\mathbf{x}) & \text{on } \Gamma_{\text{in}} \times [0, T], \\ \mathbf{u}(t, x) = \mathbf{0} & \text{on } \Gamma_0(\boldsymbol{\mu}) \times [0, T], \\ (\nu\nabla\mathbf{u} - p\mathbf{I})\mathbf{n} = \mathbf{0} & \text{on } \Gamma_{\text{out}} \times [0, T], \\ \mathbf{u}(0, \mathbf{x}) = \mathbf{k}(\mathbf{x}) & \text{in } Q(\boldsymbol{\mu})_0, \end{cases} \tag{6.12}$$

holds. Here, we have that $\Gamma = \Gamma_{\text{in}} \cup \Gamma_0 \cup \Gamma_{\text{out}}$ is the boundary of Ω and it is composed by the inlet boundary Γ_{in}, the outlet boundary Γ_{out} and the physical walls $\Gamma_0(\boldsymbol{\mu})$. The term $\mathbf{f}(\mathbf{x})$ stands for the stationary non-homogeneous boundary condition, whereas $\mathbf{k}(\mathbf{x})$ indicates the initial condition for the velocity at $t = 0$. Shape changes are applied to the boundary $\Gamma_0(\boldsymbol{\mu})$ corresponding to the airfoil wall, which in the undeformed configuration corresponds to the 4-digits, NACA 4412 wing profile. Such shape modifications are associated to numerical parameters contained in the vector $\boldsymbol{\mu} \in \mathbb{R}^k$ with $k = 10$.

As geometrical deformation map $\mathcal{M}$ we adopt the shape morphing proposed in [114], where 5 shape functions r_i are added to the upper and lower part of the airfoil profile denoted by y^+ and y^-, respectively. Each shape function is multiplied by a possible different coefficient as in the following

$$y^+ = \overline{y^+} + \sum_{i=1}^{5} a_i r_i, \qquad y^- = \overline{y^-} - \sum_{i=1}^{5} b_i r_i, \tag{6.13}$$

where the bar denotes the reference undeformed profile. These 10 coefficients (a_i and b_i) represent the input parameters $\boldsymbol{\mu} \in \mathbb{D} := [0, 0.03]^{10}$. In figure 6.13 we depict the NACA 4412 together with the 5 rescaled shape functions r_i. The output function we want to maximize is the lift-to-drag coefficient, one of the typical quantities of interest in aeronautical problems. To recast the problem in a minimization setting, we just minimize the opposite of the coefficient. To compute it, we model a turbulent flow pasting around the 2D airfoil using the incompressible Reynolds Averaged Navier–Stokes equations. Regarding the main numerical settings, we adopt a finite volume approach with the Spalart–Allmaras model, with a computational grid of 46500 degrees of freedom. The flow velocity, at the inlet boundary, is set to 1 m/s, while the Reynolds number is fixed to 50000. For the detailed problem formulation we refer to the experiments conducted in [271].

Instead of running the high-fidelity solver for any new untested parameter, we optimize a RBF response surface built using the initial dataset. Due to the stochastic nature of the method, also in this test case we test the methods for several initial settings — 25 different runs — making the total computational load very high. Thus, we decided to build a response surface using a dataset of 333 samples, computed with the numerical scheme described above, mimicking at the same time a typical industrial workflow.

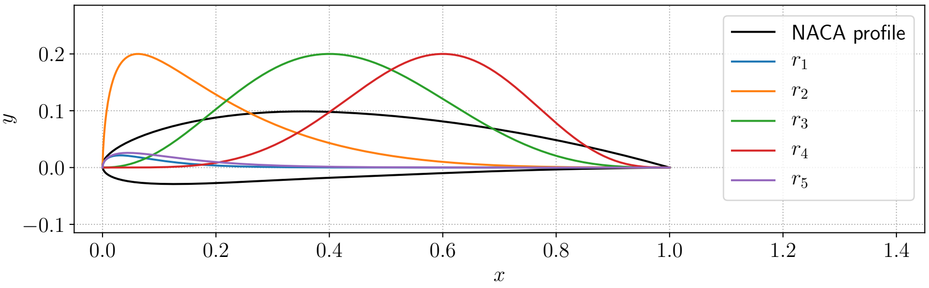


Figure 6.13: NACA 4412 profile with the 5 shape functions r_i rescaled by a factor equal to 0.2.

The objective function $f_{obj}(\boldsymbol{\mu}) : \mathbb{D} \subset \mathbb{R}^{10} \to \mathbb{R}$ we are going to minimize is the following:

$$f_{obj}(\boldsymbol{\mu}) = \begin{cases} g(\boldsymbol{\mu}) & \text{if } \boldsymbol{\mu} \in \mathbb{D}, \\ \alpha & \text{if } \boldsymbol{\mu} \notin \mathbb{D}, \end{cases} \tag{6.14}$$

where $g(\boldsymbol{\mu})$ is the response surface built using the RBF interpolation technique [39] over the samples, while $\alpha \in \mathbb{R}$ is a penalty constant. To prevent the evolution from creating new individuals that do not belong to $\mathbb{D}$, we impose a penalty factor $\alpha = 10$. We recall that we minimize the opposite of the lift-to-drag coefficient.

Figure 6.14 reports the evolution of the best-fit individual over 10 generations. Also in this case, we apply the proposed algorithm and the standard GA to 25 different initial settings, using an initial population size $N_0 = 20$ and selecting at each generations the $N = 10$ best-fit individuals for the offspring. The plot depicts the mean best-fit individual with solid lines, whereas the shaded areas show the interval between the minimum and maximum (of the 25 runs) for each generation. Even if the dimension of the parameter space is not very high (10), we can see that on average the proposed algorithm is able to converge faster. The difference between the two methods is not as remarkable as in a higher dimensional test case, but we can see that the best run using standard GA is slightly worse than the mean optimum achieved by ASGA. This again demonstrates the value in the proposed method. Moreover, we emphasize that also in this case the decay of the objective function in the first generations with ASGA is faster.

6.5 CONCLUSIONS

In this chapter, we have presented a novel approach for optimization problems coupling the supervised learning technique called Active Subspaces with the standard genetic algorithm. We have demonstrated the benefits of such method by applying it to some benchmark functions and to a realistic engineering problem. The proposed method achieves faster convergence to the optimum, since the individuals evolve only along few principal directions (discovered exploiting the AS property). Further, from the results it emerges that the gain induced from the ASGA method is greater

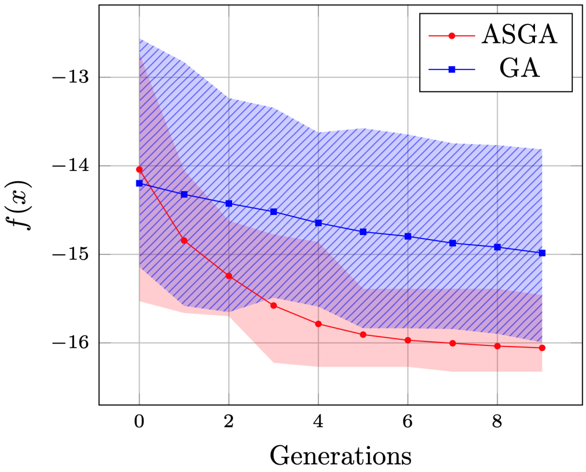


Figure 6.14: Results of the optimization of the NACA airfoil design in a 10-dimensional space. We compare the standard GA (in blue square dots) with the proposed algorithm ASGA (in red circle dots) using an initial population of size 20, while the dimension for each generation is fixed to 10. The solid lines represent the mean, over 25 runs, of the objective function corresponding to the best individual at each generation. The shaded areas show the interval between minimum and maximum (blue with lines for GA, red for ASGA).

for high-dimensionality functions, making it particularly suited for models with many input parameters.

This new method can also be integrated in numerical pipelines involving model order reduction and reduction in parameter space. Reducing the number of input parameters is a key ingredient to improve the computational performance and to allow the study of very complex systems.

Since the number of active dimensions is important for the accuracy of the AS, future developments will focus on an efficient criterion to select dynamically the number of AS dimensions, which in the presented results are kept fixed. Future studies will also address the problem of incorporating non-linear extensions of active subspaces into the ASGA, focusing on the construction of a proper back-mapping from the reduced space to the original full parameter space.

Part III

APPLICATIONS IN NAVAL ENGINEERING

7

STRUCTURAL OPTIMIZATION OF PASSENGER SHIP HULLS

Nowadays the shipbuilding industry is facing a radical change towards solutions with a smaller environmental impact. This can be achieved with low emissions engines and by reducing the metal raw materials used during the manufacturing. This chapter focus of this second aspect by presenting a complete structural optimization pipeline for modern passenger ship hulls which exploits advanced model order reduction techniques to reduce the dimensionality of both input parameters and outputs of interest. We test the whole framework over the same hull but with two different parameterizations, showing how we can deal with high-dimensional optimization.

This chapter has been possible thanks to the collaboration with Fincantieri S.p.A., in particular with Matteo Sidari and Mauro Sicchiero.

7.1 LITERATURE REVIEW

When considering optimization of complex systems in an industrial context we must rely on surrogate models in order to alleviate the computational cost of this kind of many query problems. Many studies have been conducted to access the structural behaviour of passenger ship hulls [111, 199, 220, 254]. In [208] they compare different surrogate models to improve the design process of complex thin-walled ship structures, without using any POD-based model order reduction. For structural behaviour and optimisation of passenger ships we cite [216], and [215] where they used efficient finite element modelling, evolutionary optimisation algorithm and indirect constraint relaxation. We used a similar idea for the stability constraints, where local stress peaks are allowed to exceed the rule-based strength limits.

Scientific machine learning [11] is widely used in applied mathematics and in engineering applications [38] such as inverse problems, optimization, and prediction of the behaviour of parametrized systems, to cite a few. Structural analysis of complex systems through reduced order modeling is not limited to naval engineering. Recently a component-based data-driven approach has been proposed to access the structural integrity of aircraft components [136] in the context of modern digital twins incorporating not only data but also physics models, also referred to as hybrid twins [47]. For multidisciplinary analysis and optimization involving reduction in both input and output spaces we cite [27, 28], while for a specific naval engineering application we suggest [142].

In this work we propose an optimization framework, to be used in the preliminary design phase, involving many reduced order models to access the structural behaviour of modern passenger ship hulls under different parametric configurations and loading conditions. We incorporate parameter space reduction by constructing a multi-fidelity surrogate model, without the need of running simplified simulations, exploiting the presence of an active subspace [54, 224]. This represents a new data-driven non-intrusive ROM, more accurate with respect to a more classical interpolation method such as GPR. The optimization pipeline, thanks to its modularity, allows for different target function to minimize, from the total mass of the hull, to the manufacturing cost of the structure.

This chapter is organized as follows: in section 7.2 we present the entire pipeline; in section 7.3 and section 7.4 we describe the full order model and the reduced order one, respectively. We also compare the new proposed data fusion approach to the more classical POD with Gaussian Process Regression. In section 7.5 we briefly summarize the Bayesian optimization scheme we used for the numerical results in section 7.6, where we tested the framework on two configurations of the same hull. Finally in section 7.7 we draw the conclusions and show some future perspectives on multi-objective optimization.

7.2 STRUCTURAL OPTIMIZATION PIPELINE

In this chapter we are going to use the Nested Analysis and Design (NAND) [8, 13] approach. We consider the problem:

$$\min_{\boldsymbol{\mu}\in\mathcal{P}} \quad f(\mathbf{w},\boldsymbol{\mu}), \qquad \text{s.t.} \quad R(\mathbf{w},\boldsymbol{\mu}) = 0, \tag{7.1}$$

where $\mathbf{w}$ represents the state vector, and $R(\mathbf{w},\boldsymbol{\mu})$ a general high-dimensional discretized parametric PDE, which we are going to characterized in section 7.3. Following the NAND approach, where $\mathbf{w}$ is considered an implicit function of $\boldsymbol{\mu}$, we can rewrite the optimization problem as

$$\min_{\boldsymbol{\mu}\in\mathcal{P}} \quad f(\mathbf{w}(\boldsymbol{\mu}),\boldsymbol{\mu}). \tag{7.2}$$

So for every queried parameter point $\boldsymbol{\mu}$, we solve the PDE and evaluate the function to minimize. For a fast and accurate solution of the PDE we use reduced order models described in section 7.4.

The complete structural mono-objective optimization workflow is depicted in figure 7.1, in which for every building block we emphasize the software used.

We start from the construction of the parametrized structural model with MSC Patran, and we construct a database of full order solutions with MSC Nastran corresponding to a given set of parameters for every loading condition. With this database we construct different reduced order models depending on the quantity of interest we want to approximate. We use PODI [40, 170] for the stress tensor and buckling usage factor tensor field reconstruction, and GPR [287] for the approximation of scalar functions. Moreover we exploit Active Subspaces [54] for the reduction of the parameter space dimension to build low-fidelity models and improve the PODI prediction capabilities in a multi-fidelity setting. These parameter and model reduction methods are combined for a computational efficient and reliable evaluation of the constraints regarding the stability of the whole hull: we check how many elements are yielded, and how many elements are subjected to buckling phenomena. We remark that we allow local stress peaks to exceed the classification society rule limits, since we automatically incorporate within the function to optimize the necessary interventions at the shipyard to stabilise such elements. This is particularly important since the proposed pipeline is going to be used in the preliminary design phase. The optimization is done with a Bayesian approach. The approximated optimum is then validated with the full order model and the snapshots database enriched accordingly.

We are going to present all the numerical methods employed and finally the application of the whole pipeline to an actual passenger ship hull depicted in figure 7.2 and built by Fincantieri S.p.A..

7.3 FULL ORDER MODEL

In this section we describe the PDE we need to solve and the high-fidelity solver used to create the solutions database.

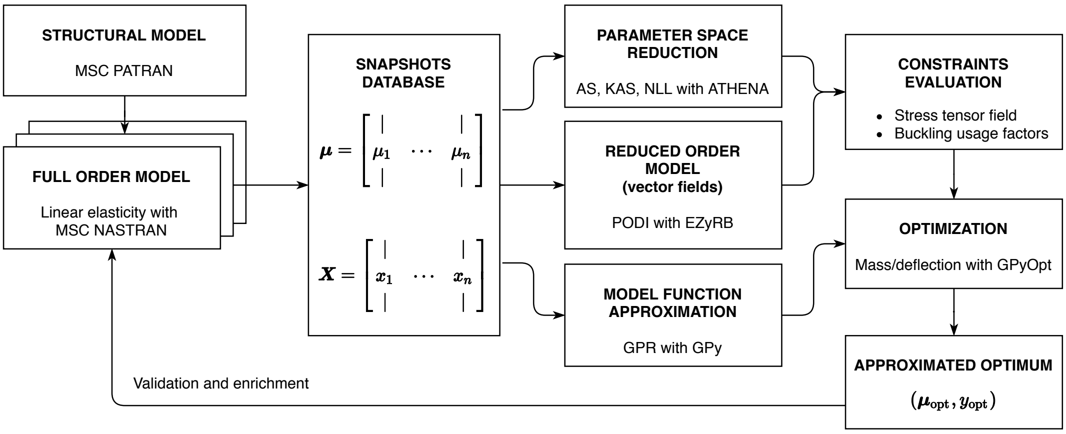

STRUCTURAL MODEL
MSC PATRAN
FULL ORDER MODEL
Linear elasticity with MSC NASTRAN
SNAPSHOTS DATABASE
$\boldsymbol{\mu} = [\mu_1 \cdots \mu_n]$
$\boldsymbol{X} = [x_1 \cdots x_n]$
PARAMETER SPACE REDUCTION
AS, KAS, NLL with ATHENA
REDUCED ORDER MODEL (vector fields)
PODI with EZyRB
MODEL FUNCTION APPROXIMATION
GPR with GPy
CONSTRAINTS EVALUATION
Stress tensor field
Buckling usage factors
OPTIMIZATION
Mass/deflection with GPyOpt
APPROXIMATED OPTIMUM
$(\boldsymbol{\mu}_{\text{opt}}, y_{\text{opt}})$
Validation and enrichment

Figure 7.1: Structural optimization workflow, from the base structural model creation, to the approximated optimum and validation. Each block of the pipeline reports the software used.

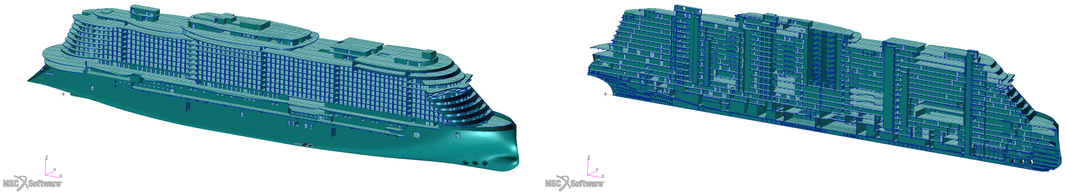


Figure 7.2: A complete view on the hull on the left, and a longitudinal section on the right.

The equations governing the linear elastic isotropic problem are the equilibrium equation, the linearised small-displacement strain-displacement relationship, and the Hooke's law, respectively:

$$\begin{cases} -\nabla \cdot \sigma = f, \\ \epsilon = \frac{1}{2}[\nabla u + \nabla u^T], \\ \sigma = C(E, \nu) : \epsilon, \end{cases} \tag{7.3}$$

where σ is the Cauchy stress tensor, f is the body force, ϵ is the infinitesimal strain tensor, u is the displacement vector, and C is the fourth-order stiffness tensor depending on E, the Young modulus, and on ν, the Poisson's ratio.

The finite element method employed uses 2-dimensional elements, commonly referred to as plate and shell elements. They are used to represent areas in the model where one of the dimensions is small in comparison to the other two. As shown in figure 7.4, the height or thickness of the element is substantially less than the width and the length. We are going to use MSC Nastran CQUAD4 and CTRIA3 elements, which are general-purpose plate elements capable of carrying inplane force, bending forces, and transverse shear force. The membrane stiffness of the 2-dimensional elements is calculated using the plane stress theory. Most thin structures constructed from common engineering material, such as aluminum and steel, can be modeled effectively using plane stress. In figure 7.3 an example of solution for a given loading condition.

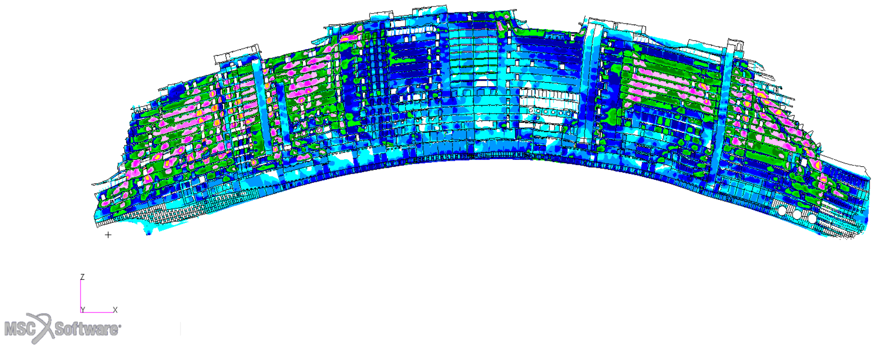


Figure 7.3: Possible deformation of the hull under the hogging loading condition. Displacements are magnified. Colors refer to the von Mises criterion.

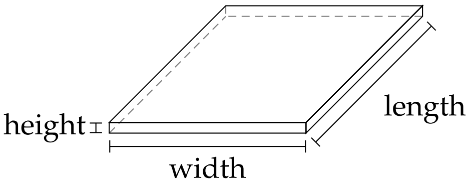


Figure 7.4: Sketch of a plate element.

7.4 REDUCED ORDER MODEL

In this section we describe and compare the new proposed non-intrusive data-driven ROM exploiting the low-intrinsic dimensionality of the parameter to reduced snapshots map.

In the entire chapter the parameters vector $\boldsymbol{\mu} \in \mathcal{P}$ represent the thickness of some selected macro areas of steel plates. In order to speed up the optimization procedure we construct a reduced order model for every field of interest, and from them we compute the derived quantities which describe the constraints and the functions to optimize.

As reduced order model we use the approach presented in section 5.1 which uses NARGPAS to approximate all the POD modal coefficients we retain. We can compare the L^2 relative error using this approach, called POD-NARGPAS, with POD-GPR, which uses only a single fidelity. In the first case each modal coefficient is approximated, using algorithm 12, with the following multi-fidelity model:

$$g_M = ((f_H|x_i^H, y_i^H),\ (f_L|x_i^L)) \sim (\mathcal{GP}(f_H|m_H, \sigma_H), \mathcal{GP}(f_L|m_L, \sigma_L)), \tag{7.4}$$

where the H and L denote the high and low-fidelity, respectively. The low-fidelity is built by extending on the whole parameter space a one-dimensional response surface constructed over the AS of each POD coefficient. For the active subspace's response surface design we used ATHENA[1] [222], while for the actual construction of the reduced order models we used EZyRB[2] [70].

We used 100 samples selected with a discrete space filling criterion for the train and another 100 uniformly distributed samples for the test. We compare the

[1] Available at https://github.com/mathLab/ATHENA.
[2] Available at https://github.com/mathLab/EZyRB.

prediction error for two quantities: the stress tensor field and the buckling usage factors tensor field. For the first case, in figure 7.5 we can see how the autoregressive scheme outperforms the POD-GPR with a gain around 20% without the necessity of performing new simulations. We also notice how the single fidelity approach becomes a little unstable with respect to the other, especially for higher truncation ranks. For lower truncation ranks the performance are almost the same and we did not include them in the plots to avoid too different scales in the y-axis.

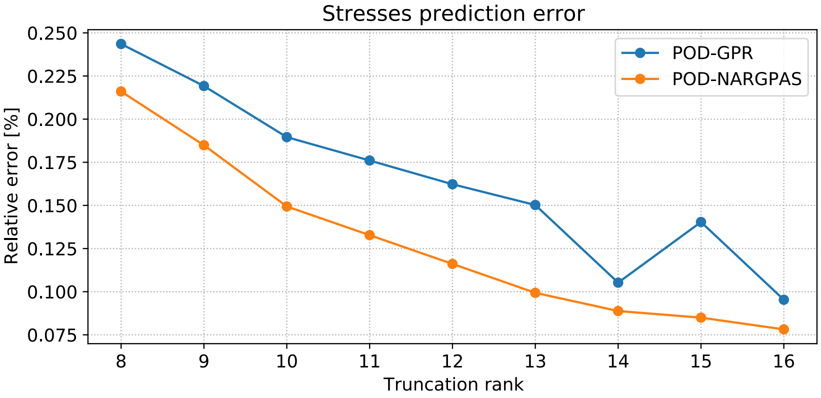


Figure 7.5: Comparison between the stress tensor field prediction error using POD-GPR and POD-NARGPAS varying the truncation rank.

The same behaviour is present for the buckling usage factors for which, in figure 7.6, we see a plateau for both the methods. The gain is again around 20%. The higher relative error is due to the fact that to compute this tensor field we used classification society rules, which, in general, are nonlinear and present discontinuities due to the presence of different safety factors for different cases. For the hull showed in figure 7.2 the prediction accuracy is better when considering the elastic buckling stress for plate panels in uni-axial compression with respect to the bi-axial compression, and to the bi-axial compression and in-plane shear stresses.

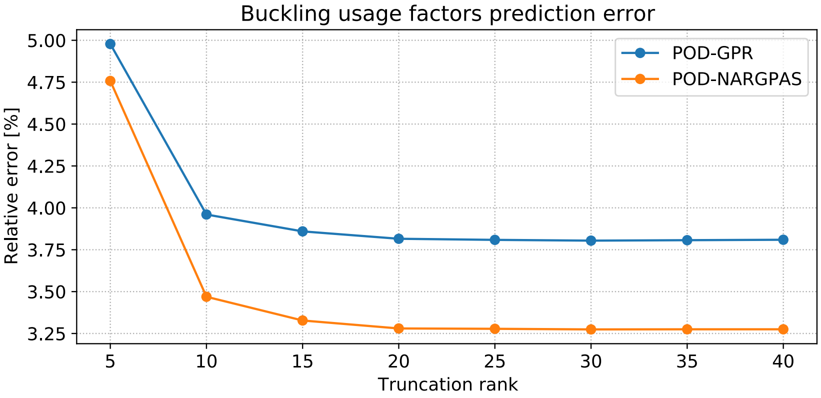


Figure 7.6: Comparison between the buckling usage factors tensor field prediction error using POD-GPR and POD-NARGPAS varying the truncation rank.

The new proposed data-driven approach outperforms the more classical single-fidelity, without the need of any additional simulation. We remark that POD-GPR, for naval engineering problems, has proven better than RBF and linear interpolation [65].

7.5 BAYESIAN OPTIMIZATION

To minimize the model functions of interest we use Bayesian optimization [90, 202, 247], which we are going to briefly present in this section. It is a class of machine-learning-based derivative-free global optimization methods, which is suited for input parameters domain with less than 20 dimensions. One of the main assumptions is that we do not have any information about the structure of the function to optimize, so it is intended as a black-box. Bayesian optimization was first introduced in [148, 181, 296], and successively made popular in [132] in the context of efficient global optimization.

Mathematically, we are considering the problem of finding a global minimizer of an unknown function $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$, that is

$$x_{\text{opt}} = \underset{x \in \Omega}{\operatorname{argmin}}\, f(x), \tag{7.5}$$

where $\Omega \subset \mathbb{R}^n$ is the design space of interest, which in our case is the parameter space $\mathcal{P}$. More general settings can include design spaces with less regularity due to the presence of possible nonlinear constraints. Here we consider a sequential search algorithm which selects the next location where to query f. The Bayesian posterior represent the best current knowledge of the function to optimize. The locations are selected by evaluating an acquisition function $\alpha : \Omega \to \mathbb{R}$ which leverages the uncertainty of the posterior to guide the exploration of the design space. In algorithm 16 we sketch a pseudo-code to highlight the main steps of the whole process.

As acquisition function we use the expected improvement (EI), which is one of the most commonly used. Let

$$f_n^*(x) := \min_{l \leq n} f(x_l), \tag{7.6}$$

be the value of the smallest observed values. We want to perform a new evaluation, say $y = f(x)$, which has the highest expected improvement defined as:

$$\alpha(x) := \mathrm{EI}_n(x) := \mathbb{E}_n\left[(f_n^*(x) - f(x))_+\right]. \tag{7.7}$$

With $\mathbb{E}_n$ we denote the expectation taken under the posterior distribution given the first n evaluations, that is $\mathbb{E}_n[\cdot] = \mathbb{E}_n[\cdot | x_1, \dots, x_n, y_1, \dots, y_n]$. This acquisition function can be computed in closed form [132]. The actual next point x_{n+1} to evaluate is then given by

$$x_{n+1} = \underset{x}{\operatorname{argmax}}\, \mathrm{EI}_n(x). \tag{7.8}$$

To find it we exploit the simpler structure of α with respect to the target function f, which allows for inexpensive evaluations and also easy computation of first and second derivatives.

Algorithm 16 Bayesian optimization pseudo-code.

```
Input:
    model function $f$ to minimize
    acquisition function $\alpha$
    initial number of evaluation points $n_0$
    maximum number of iterations $N$
Output:
    smallest value for $f(x)$
Compute the Gaussian process prior on $f$.
Evaluate $f$ at the initial $n_0$ points.
Set $n = n_0$.
while $n \leq N$ do
    Update the posterior probability distribution on $f$ with all the available evaluations.
    Using the current posterior distribution find $\mathbf{x}_n = \text{argmax}_x \; \alpha(x)$.
    Evaluate $y_n = f(x_n)$.
    Increment $n$: $n = n + 1$.
end while
return either the point with the smallest $f(x)$, or the one with the smallest posterior mean.
```

In figure 7.7 we show 2 iterations of a Bayesian optimization procedure for a test function, highlighting both the prediction and the value of the acquisition function. We notice that the acquisition is high where the model predicts a low objective (so-called exploitation) and where the prediction uncertainty is high (so-called exploration). Note that the area on the far left remains unsampled, as while it has high uncertainty, it is predicted to offer a smaller improvement over the best observation.

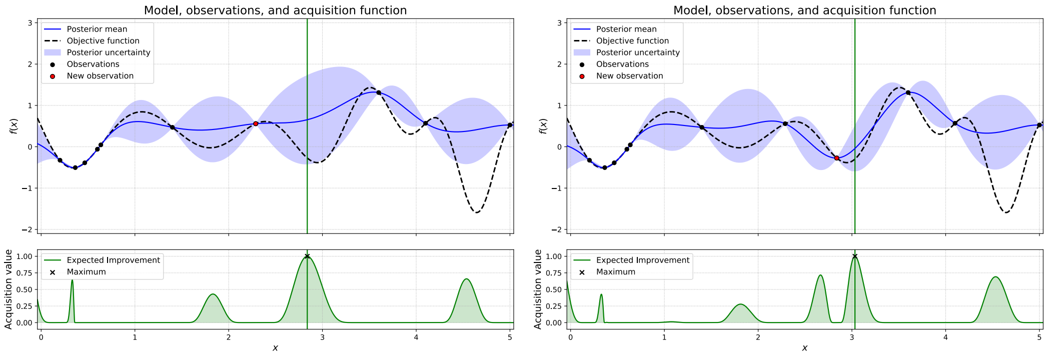


Figure 7.7: Illustration fo the Bayesian optimization procedure over 2 intermediate iterations. The top part of the plots shows the estimated mean and confidence intervals of the unknown objective function (in dashed line). In the bottom part of the plots we show the acquisition function in green and its maximum.

7.6 NUMERICAL RESULTS

In this section we are going to apply the optimization pipeline described in section 7.2 to the parametric hull depicted in figure 7.2. We analyse two different parameterization of the same reference hull, starting from a 6 parameters test case, and then showing the results for the 16 parameters case. We perform a discrete mono-objective optimization of the mass of the parametric part of the hull, with stability contraints. Even if we are going to minimize a single scalar function of interest we provide some insights regarding multi-objective optimization considering also the deflection at a prescribed node.

7.6.1 *Objective function definition*

The target function we are going to minimize is the total mass m of the parametrized regions of the hull plus the mass of the buckling stiffeners needed to stabilise the buckled elements of the entire hull. Other possibility are available within the optimization framework such as the deflection for a given loading condition, and the total cost related to the parametrized decks which considers both the acquisition cost of the metal raw materials and the manufacturing cost for the installation of the steel plates and buckling stiffeners, specific for each shipyard. For industrial reasons in this chapter we are not going to present the results for the cost optimization, but we want to emphasize that the framework if very versatile and allows the use of different target functions.

As for the stability constraints we set some thresholds for the Cauchy stress tensor components in order to count how many plate elements are yielded for a prescribed set of loading conditions, which are hogging and sagging. See figure 7.3 for an example of hogging condition. Given the symmetric Cauchy stress tensor in the global reference frame, whose components for a single element are

$$\sigma = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_y & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_z, \end{bmatrix}, \tag{7.9}$$

we define an element yielded if at least one of the following conditions is not satisfied:

$$-245 < \sigma_i < 245, \qquad \text{for } i \in \{x, y, z\}, \tag{7.10}$$

$$-153 < \tau_i < 153, \qquad \text{for } i \in \{xy, xz, yz\}, \tag{7.11}$$

$$\sigma_{\text{VM}} := \sqrt{\sigma_x^2 + \sigma_y^2 - \sigma_x\sigma_y + 3\tau_{xy}^2} < 307, \tag{7.12}$$

where σ_{VM} stands for the von Mises yield criterion. The actual constraint for the optimization is the maximum number N_{max}^y of elements that can yield. Exactly the same is done for the buckling usage factors associated to each element. An element is considered buckled, based on the DNV GL classification rules[3], if at least one of the 11 components of the buckling usage factors tensor is greater than 1, for at

[3] Det Norske Veritas (DNV) Rules for Ships, part 3, chapter 1, section 13: Buckling control.

least one loading condition. The maximum number of allowed buckled elements is $N^b_{\max}$. To incorporate these stability constraints we penalize the objective function f_{obj} with a parabolic function depending on the the violated constraint, that is

$$f_{\text{obj}}(\boldsymbol{\mu}) := m(\boldsymbol{\mu}) + m_{\text{bs}} N^b(\boldsymbol{\mu}) + c_y (N^y(\boldsymbol{\mu}) - N^y_{\max})^2_+ + c_b (N^b(\boldsymbol{\mu}) - N^b_{\max})^2_+, \quad (7.13)$$

where $m(\boldsymbol{\mu})$ is the mass of the parametric decks, $N^y(\cdot)$ and $N^b(\cdot)$ denote the number of yielded and buckled elements, respectively, m_{bs} is the mass of a single buckling stiffener, $c_y = 1$, $c_b = 0.001$, and $(\cdot)_+ = \max(\cdot, 0)$ stands for the positive part. The coefficients c_y and c_b can be changed in case the user wants a particular constraint to weight more with respect to the other.

7.6.2 *First test case: 6 parameters*

For the first test case we assign a parametric thickness to 6 different regions of the hull. In particular we choose the ones highlighted with solid colors in the left panel of figure 7.8. Each parameter correspond to a different deck. Table 7.1 summarizes the parameters, the corresponding deck, the default thickness associated to the reference hull, and the ranges of variation.

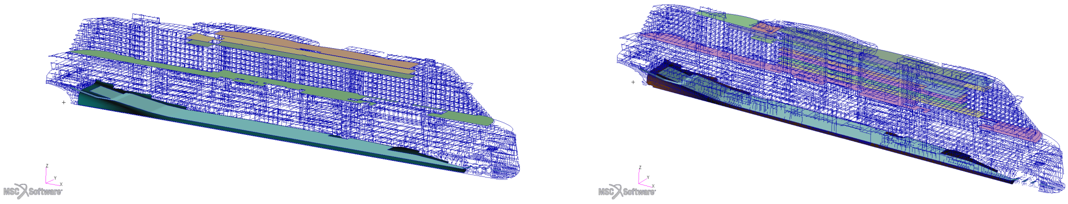


Figure 7.8: View of the hull parametric decks. In solid colors the regions of the hull affected by the parameters described in table 7.1 (left panel), and in table 7.2 (right panel).

Table 7.1: Parameters description of the hull for the first test case. All data are in mm.

Parameter	Region	Default thickness	Lower bound	Upper bound
μ_1	Deck 17	9.0	5.0	12.0
μ_2	Deck 16	8.0	5.0	12.0
μ_3	Deck 15	7.5	5.0	12.0
μ_4	Deck 09	8.0	5.0	15.0
μ_5	Deck 01	16.0	12.0	25.0
μ_6	Deck 00	20.0	12.0	25.0

Since the steel plates can have only a finite set of possible thickness, the input parameter space is discrete. In order to cover the domain in the most uniform way, we generate many random uniform sampling $S^i := \{\boldsymbol{\mu}^i_j\}_{j=1}^M$ composed by M samples each. We compute the minimum pairwise distance d^i associated to each S^i and we retain the samples with maximum d. Let us call the chosen samples $S := \{\boldsymbol{\mu}_j\}_{j=1}^M$.

To have a better idea on how to set the stability constraints for a given type of hull, we can plot the distribution of the parametric hulls with respect to the number of yielded elements (left panel of figure 7.9). We see that if we set $N^y_{\max} = 200$, the valid samples will be roughly 82% of all the manufacturable hulls. In the right panel of figure 7.9 we plot the distribution of the hulls with respect to the number of buckled elements and we set $N^b_{\max}$ accordingly.

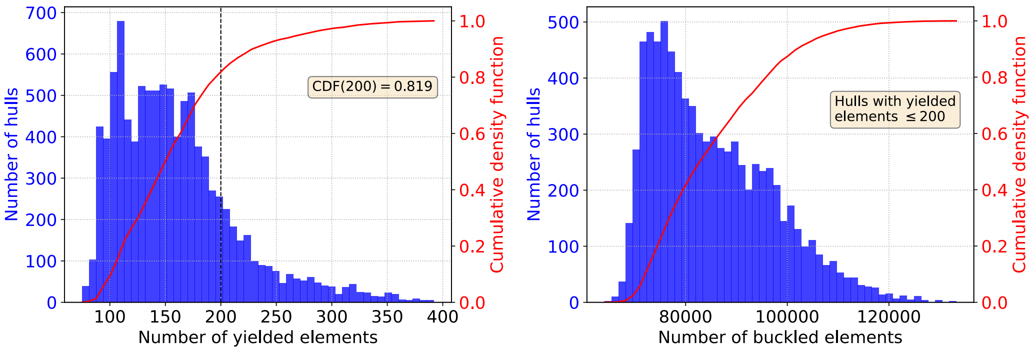


Figure 7.9: Distribution of the hulls with respect to the stability constraints. In the left panel we consider the number of yielded elements, while in the right panel we have the distribution of the parametric hulls satisfying the first constraint with respect to the number of buckled elements.

After constructing a reduced order model for every component of the stress tensor and the buckling usage factors tensor field we can perform the Bayesian optimization described above. The truncation rank for the stress components is 10, while for the buckling usage factors is 20. We emphasize that every parametric term in equation (7.13), except $m(\boldsymbol{\mu})$, is predicted using the precomputed reduced order models. In this way a single parameter evaluation takes approximately 1 second. The term $m(\boldsymbol{\mu})$ is computed exactly. We can access the prediction error for the stress tensor components and for the von Mises criterion using 5-fold cross validation. In figure 7.10 we see the mean L^2 relative error when using 80% of the snapshots as train dataset and the rest as test dataset. We see that the error is always below 0.7%.

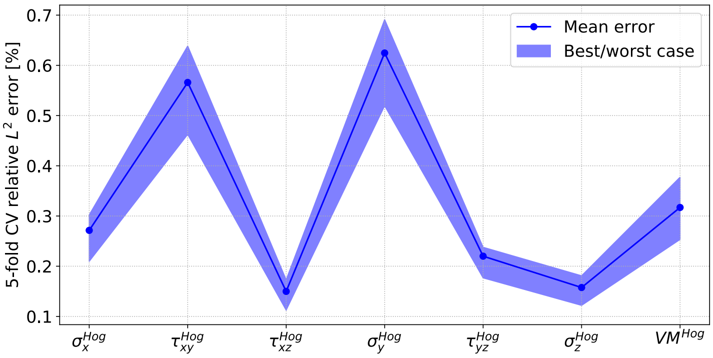


Figure 7.10: L^2 relative error for the prediction of the stress tensor components for the hogging loading condition. The solid line identifies the mean error over a 5-fold cross validation.

We set the computational budget for the Bayesian optimization to 300 iterations. We remark that at every iteration the GPR for the target function has to be recomputed. This means that the GPR construction time increases at every iteration. In figure 7.11 the results for all the successive optimization runs. After an optimization cycle is completed, the high-fidelity solution corresponding to the approximated optimum is added to the snapshots database and the reduced order models are recreated. Thus the accuracy in the neighborhood of the current optimum is increased and the successive runs will exploit such information by focusing in that specific region, or will explore different areas if the error committed at the current optimum is too high. In this case one run is sufficient to find the optimum. In fact, when we add that solution to the database the optimizer is not able to find a better hull, and the small discrepancy we see in the plot is due to the prediction error in counting the buckling stiffeners needed to stabilise the buckled elements. Even if the mass reduction seems not as remarkable we have to remember that the parametrized regions are only a portion of the entire hull, thus a small amount of tons corresponds to a relative high percentage. Absolute values are not present for industrial reasons.

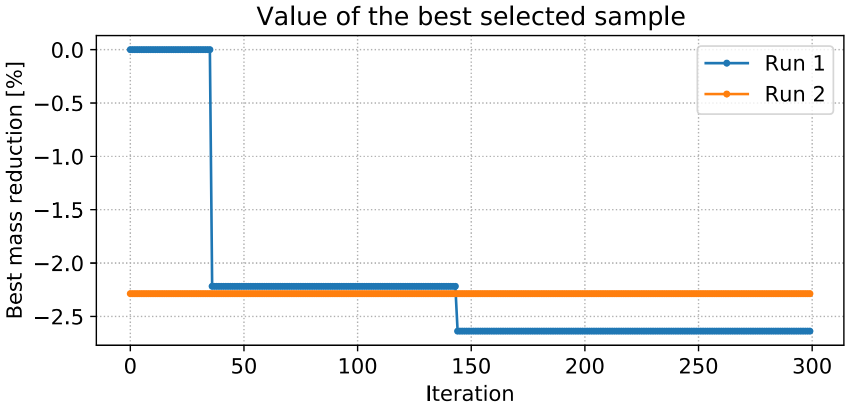


Figure 7.11: Different optimizations runs for the first test case. The relative reduction is with respect to the best sample among the initial solutions database.

7.6.3 *Second test case: 16 parameters*

For the second test case we increased the number of input design parameters to 16. The parametric regions are depicted with solid colors in the right panel of figure 7.8, while a deeper description of the range of variations of each parameter μ_i, $i = 1, \dots, 16$, can be found in table 7.2. We used the same sampling strategy described above to generate 300 samples.

Given the higher dimensional input space the solution manifold is much difficult to approximate, so we expect higher prediction error, especially for the buckling usage factors. This will result in a greater number of optimization runs before arriving at convergence, as we can see from figure 7.12, where the first runs produce a greater reduction polluted by the prediction error. The truncation rank for the stress components is 30, while for the buckling usage factors is 60. We increased the computational budget for the Baysian optimization to 600 evaluations. The fourth run does not produce any new improvement with respect to the snapshots database,

Table 7.2: Parameters description of the hull for the second test case. All data are in mm.

Parameter	Region	Default thickness	Lower bound	Upper bound
μ_1	Deck 15	7.5	5.0	15.0
μ_2	Deck 16	8.0	5.0	20.0
μ_3	Deck 17	9.0	5.0	20.0
μ_4	Deck 14	7.5	5.0	15.0
μ_5	Deck 13	7.0	5.0	15.0
μ_6	Deck 12	6.5	5.0	15.0
μ_7	Deck 11	6.0	5.0	15.0
μ_8	Deck 10	5.5	5.0	15.0
μ_9	Deck 17	6.0	5.0	20.0
μ_{10}	Deck 17	15.0	5.0	20.0
μ_{11}	Deck 17	6.0	5.0	20.0
μ_{12}	Deck 16	6.0	5.0	20.0
μ_{13}	Deck 16	6.0	5.0	20.0
μ_{14}	Deck 09	8.0	5.0	15.0
μ_{15}	Deck 01	16.0	12.0	25.0
μ_{16}	Deck 00	20.0	12.0	25.0

so we can confirm we converged to the global optimum in the approximated solution manifold. In particular from the figure we conclude that the third run produces the actual optimum. The slight difference is due to the approximation error in the computation of the number of buckled elements, which is not present in the successive run since that snapshot has been added to the database.

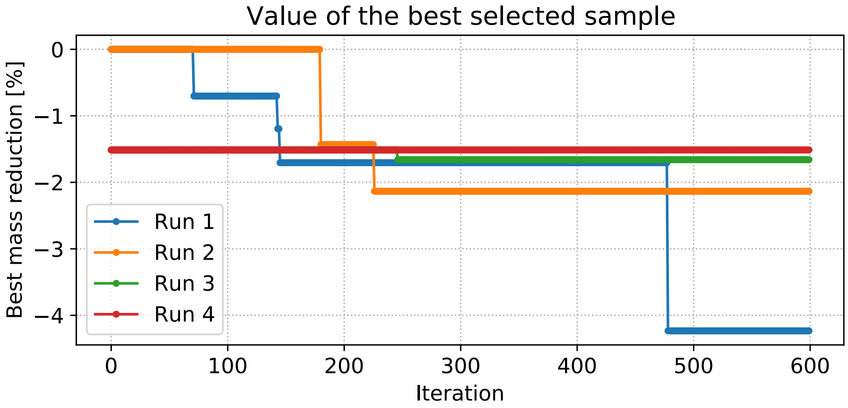


Figure 7.12: Different optimizations runs for the second test case. The relative reduction is with respect to the best sample among the initial solutions database.

7.7 Conclusions and future perspectives

In this chapter we proposed a modular data-driven structural optimization framework for modern passenger ships. We exploit several reduced order models coupled with parameter space reduction in a multi-fidelity setting. This new approach is called POD-NARGPAS. We demonstrated its performance against the classical POD-GPR.

Our efficient numerical pipeline allows for different discrete mono-objective optimization given a precomputed set of high-fidelity simulations. We parametrized the thickness of various regions of the reference hull and we perform a mass minimization considering stability constraints such as the total number of yielded and buckled elements.

Future works will focus on improving the accuracy of constraints evaluations, for example with a multi-fidelity approximation of the scalar output and not only for the reconstruction of the entire field [224]. Another possibility is the exploitation of local information with Local Active Subspaces [223], presented in chapter 3, to further improve the regression performance.

Regarding the optimization procedure a natural evolution is the implementation of a multi-objective optimization which considers at the same time both mass and deflection at a given point, for example. This can be done in a Bayesian setting, but if we want to increase the number of input parameters other approaches should be considered, such as genetic algorithms [73]. In figure 7.13 we show the distribution of the hulls in terms of mass and deflection at a given point, emphasizing the valid samples from the one violating the stability constraints. This is just the first step in order to provide the user useful insight towards the multi-objective optimization which we will study in the near future.

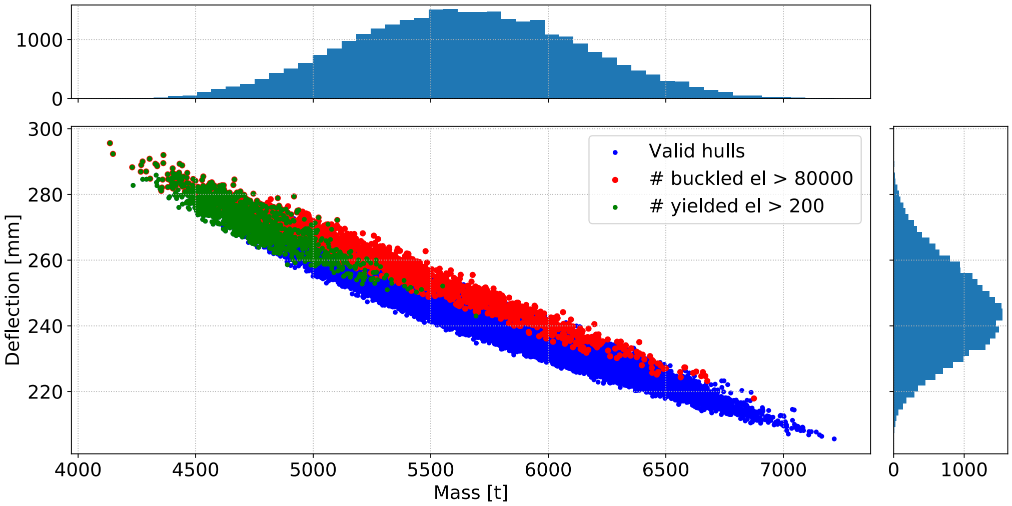


Figure 7.13: Parametric hulls distribution in terms of mass and deflection at a given node. In red the hulls violating the buckling constraint, in green the ones violating the yielding constraint, and in blue the admissible samples.

8

HULL SHAPE DESIGN OPTIMIZATION WITH SELF-LEARNING MESH MORPHING

In the field of parametric partial differential equations, shape optimization represents a challenging problem due to the required computational resources. In this chapter, a data-driven framework involving multiple reduction techniques is proposed to reduce such computational burden. POD and ASGA (introduced in chapter 6) are applied for a dimensional reduction of the original (high-fidelity) model and for an efficient genetic optimization based on active subspace property. The parameterization of the shape is applied directly to the computational mesh, propagating the generic deformation map applied to the surface (of the object to optimize) to the mesh nodes using RBF interpolation. Thus, topology and quality of the original mesh are preserved, enabling application of POD-based reduced order modeling techniques, and avoiding the necessity of additional meshing steps. Model order reduction is performed coupling POD and GPR in a data-driven fashion. The framework is validated on a benchmark ship. All the results presented in this chapter appeared in [69].

8.1 LITERATURE REVIEW

In the framework of parameterized PDE problems for engineering, ROMs and optimization algorithms are two instruments that particularly benefit a synergic use. In several cases of engineering interest in which PDEs solution require considerable computational effort, ROMs enable in fact a remarkable reduction in the resources required for each calculation. There are of course several ways to reduce the dimensionality of discretized PDEs. The most naive approaches, such as coarsening the computational grids clearly have negative effects on the quality of the solutions. This is particularly true for problems characterized by complex physics and geometrical features, which in most cases require a very high number of degrees of freedom, ultimately resulting in expensive computations. In the context of an optimization algorithm execution, where many discretized PDE solutions must be computed, the overall computational load often becomes unaffordable. With only modest negative effects on the PDE solution accuracy, ROMs can be conveniently exploited to reduce the high dimensionality of the original discrete problem — to which we will herein refer to as FOM or *high fidelity* model. ROM algorithms can be employed in several industrial design processes, and in particular to shape optimization, in which the objective of the computations is to find the best shape of a particular product or artifact. Such problems are in fact typically modeled through parametric PDEs, in which input parameters control the geometric features of the object at hand. ROMs efficiently approximate the numerical solution of the full order PDE with a suitable reduced surrogate, enabling drastic reduction in the computational burden of the overall optimization procedure.

There are of course several different algorithms which allow for an efficient reduction of the dimensionality of parametric problem. In the present contribution, we make use of a data-driven approach based on POD [227, 230]. The equation-free nature of such method is often an essential feature in the industrial sector, where modularity and solvers *encapsulation* play a fundamental role. Indeed, the data-driven POD based ROM employed in the present optimization framework can be coupled with any PDE solver, as the data integration is enforced through the output of interest of the full order problem. Similar reduced methods have been proposed in [65, 66] for the shape optimization of a benchmark hull, while additional improvements have been made coupling the ROM with active subspace analysis and different shape parameterization algorithms in [67, 68, 270, 272]. We refer the readers interested in parametric hull shape variations using ROMs to [279], while we mention [75, 244] for design-space dimensionality reduction in shape optimization with POD. Moving from hulls to propellers, data-driven POD has also been successfully incorporated in the study of marine propellers efficiency [94, 184] as well as hydroacoustics performance [92].

A further aspect of novelty of the optimization framework proposed is related to the parameterization of the geometry. In typical shape optimization cycles, the surface of the object under study is deformed before the domain discretization takes place. Thus, the meshing phase is repeated for any deformed entity. Such approach has the clear advantage of allowing for good control of the quality of the computational grid produced for each geometry tested. Yet, it suffers of two

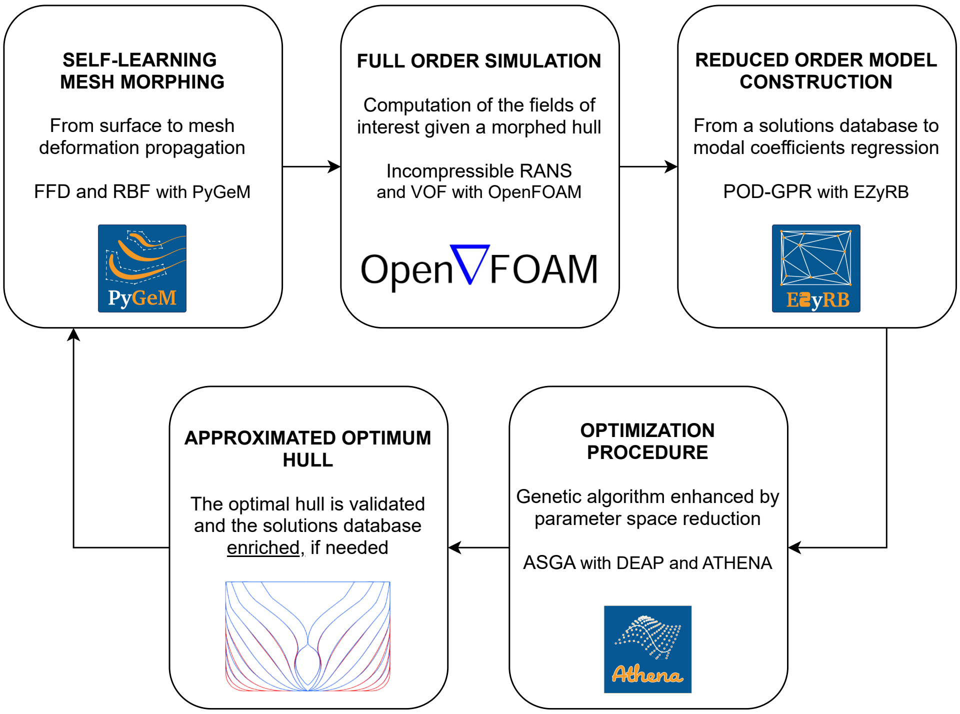

SELF-LEARNING MESH MORPHING
From surface to mesh deformation propagation
FFD and RBF with PyGeM
PyGeM
FULL ORDER SIMULATION
Computation of the fields of interest given a morphed hull
Incompressible RANS and VOF with OpenFOAM
Open∇FOAM
REDUCED ORDER MODEL CONSTRUCTION
From a solutions database to modal coefficients regression
POD-GPR with EZyRB
EZyRB
APPROXIMATED OPTIMUM HULL
The optimal hull is validated and the solutions database enriched, if needed
OPTIMIZATION PROCEDURE
Genetic algorithm enhanced by parameter space reduction
ASGA with DEAP and ATHENA
Athena

Figure 8.1: Illustration of the key steps of the proposed optimization pipeline with the methods and the softwares used.

main problems: *i)* the meshing step may be expensive, both because its CPU time might be comparable to the resolution of the problem itself, and because mesh generation is specially intensive in terms of human operator hours required; *ii)* a different mesh for each geometry does not allow for the application of POD or several other ROM approaches, which require that the mesh topology, as well as the number of degrees of freedom of the discretized problem, are conserved across all the shapes tested. Thus, assuming a generic deformation map is available, which morphs the initial object surface — not the grid —, we exploit such deformation to train a RBF interpolation that will extend the surface deformation to the nodes of the PDE volumetric mesh. In this sense, the method is capable to learn and propagate any deformation to a given mesh. Properly selecting the RBF kernel, we can then obtain a smooth deformation in all the discretized domain, not only ensuring that the overall parameterization map preserves the initial mesh quality but also its topology. We remark that in this work, Free Form Deformation (FFD) is used to deform the surface of the object under study. Yet, we stress that the RBF extension methodology is completely independent from the parameterization method chosen for the object geometry. A similar approach has been recently investigated in [130].

The optimization algorithm used in this chapter is the recently developed active subspaces extension of the classical genetic algorithm called ASGA [73], which performs the mutation and cross-over steps on a reduced dimensional space for a faster convergence.

All the algorithms used in this chapter are implemented in open source software libraries [70, 91, 222, 269], which we will briefly introduce in the discussions of the corresponding numerical methods. In figure 8.1 we depicted an outline of the whole numerical pipeline we are going to present, emphasizing the methods and the softwares used. One of the main goals of this contribution it that of testing the full pipeline composed by data-driven POD ROM, combined FFD-RBF shape parameterization algorithm and ASGA optimizer on a problem that can be both meaningful to the ship hydrodynamics community and easily reproducible. For such reason, the test case considered is that of the DTC hull [182], for which online tutorials are available to run fairly accurate flow simulations in fixed sink and trim conditions. Since in such set up, the hull optimizing resistance is a trivial, zero volume hull, the DTC benchmark hull is here optimized based on the total resistance coefficient C_t. The chapter is organized as follows: section 8.2 presents a deeper discussion about the parameterization of the object and of the computational grid; section 8.3 describes the full order model and the reduced order one. The final sections, section 8.4 and section 8.5, show the numerical results obtained and present the conclusive summary, respectively.

8.2 SHAPE AND GRID PARAMETERIZATION

Whenever industrial design processes as the ones discussed in this work are aimed at improving, among other aspects, the geometric features of a particular artifact, a shape parameterization algorithm is a cornerstone of the whole optimization pipeline. Optimization tools, as well as the non-intrusive model reduction techniques employed in the present investigation, are in fact based on the parameterized PDEs paradigm introduced in the previous section. In such framework, a set of geometric input parameters affects the output of a parametric PDE through the deformation of its domain geometry. Thus, the shape parameterization algorithm role is that of mapping the variation of a set of numerical parameters, to the corresponding deformation of the PDE domain geometry. In other words, since optimization tools are mathematical algorithms which must be fed with numbers, the shape parameterization algorithms translate shape deformations into variations of the numeric quantities they need.

8.2.1 *How to combine different shape parametrization strategies*

In this work, we make combined use of two general purpose shape parameterization algorithms to deform the three dimensional geometry of a ship hull, and accordingly update the volumetric grid used for ship hydrodynamics simulations in a fully automated fashion. More specifically, FFD is first used to generate a family of deformations of the surface of a base hull. In a second step, RBF interpolation is used to propagate the hull surface deformation to the internal nodes of the fluid dynamic simulation computational grid. For visual reference, figure 8.2 depicts the side view (on the left) and front view (on the right) of a container ship hull bow region. In the picture, several sections perpendicular to the hull longitudinal axis are indicated by red lines.

Despite an extensive discussion of FFD and RBF theoretical foundations is clearly beyond the scope of the present contribution, this section will introduce the key

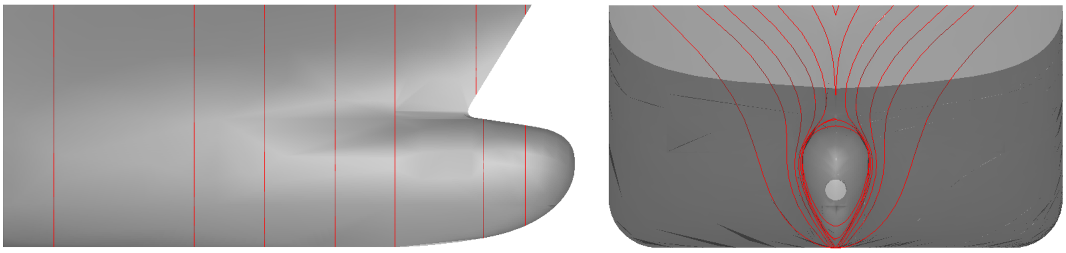


Figure 8.2: Side view (left) and front view (right) of a typical container ship hull bow region.

concept upon which both algorithms are based and describe their combined deployment in the framework of our optimization pipeline.

The first shape parameterization algorithm applied in this work is the free form deformation [153, 240, 249]. As mentioned, it is a general purpose algorithm, designed to be applied to arbitrarily shaped geometries. FFD is fundamentally made up of three different geometrical transformations, as illustrated in figure 8.3. The first transformation ψ maps the physical domain Ω into a reference domain $\widehat{\Omega}$. In such domain, a lattice of points is generated, and are used as the control points of a set of smooth shape functions such as the Bernstein polynomials used in this work. Thus, once a displacement is prescribed to one or more of the control points in the lattice, the shape functions are used to propagate such displacement to all the points in the reference domain Ω. The smooth displacement field obtained, is the second and most important transformation $\widehat{T}$ in the FFD process. In the third, final step, the deformed reference domain is mapped back into the physical one by means of ψ^{-1} to obtain the resulting morphed geometry.

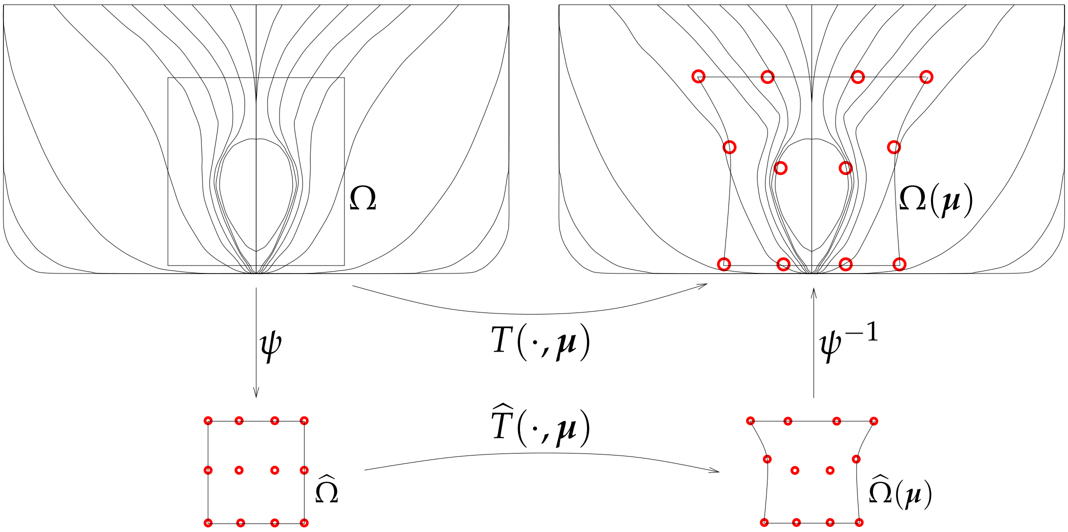

Ω
$\Omega(\boldsymbol{\mu})$
ψ
$T(\cdot,\boldsymbol{\mu})$
ψ^{-1}
$\widehat{T}(\cdot,\boldsymbol{\mu})$
$\widehat{\Omega}$
$\widehat{\Omega}(\boldsymbol{\mu})$

Figure 8.3: A two dimensional sketch of the FFD procedure applied to the surface of a container ship hull, including the three transformations ψ, $\widehat{T}(\cdot,\boldsymbol{\mu})$ and ψ^{-1} composing the process.

The current description suggests that the parameters $\boldsymbol{\mu}$ of the final FFD map $T(\cdot,\boldsymbol{\mu})$ are the displacements prescribed to one or more of the lattice control points. The procedure can account for both a variable number of lattice points and of displaced control points. For such reason, FFD deformations can be built with an arbitrary number of parameters.

We point out that the FFD algorithm results in a displacement law for each 3D space point within the control points lattice. Thus, it can be readily deployed to deform shapes specified through surface triangulations (such as STL geometries) and surface grids in general. In addition, it can be also used to directly deform volumetric grids used for fluid dynamic simulations. Yet, mainly for practical reasons, in this work we only make use of FFD to deform the STL surface triangulation describing the hull geometry. In fact, we must point out that if FFD has to be used to modify the volumetric mesh used for CFD simulations, the control points lattice dimensions must be much bigger than those needed when only deforming the hull surface, leading to infeasible optimization procedures. This is due to the fact that when deforming volumetric meshes, it is often convenient to distribute the deformations over a high number of cells, rather than concentrating all the displacements in a very confined region in which cells can get distorted or even inverted. But because FFD only affects points located within the control points lattice, this means that the latter must extend for a bigger volume. In addition, to maximize the volumetric mesh quality, the user must include more control points in the lattice to make sure that different deformation magnitudes are imposed in regions close to the hull and far from it. Such *manual* control over the local mesh deformation can often become quite cumbersome.

For such reasons, after the hull surface mesh has been modified by means of FFD, we resort to RBF to propagate the hull boundary displacements to the internal nodes of the volumetric mesh for CFD simulations. In a broader sense, RBF is an interpolation algorithm, in which linear combinations of radial bases are used to approximate a function with values prescribed only in a finite number of points, in every point of a domain. In the case of interest, the displacement field function prescribed on the points of the hull surface must be interpolated in the positions corresponding to every node of the volumetric mesh. Thus, the displacement obtained from the m surface nodes original position $\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_m\}$ and the corresponding displaced position $\{\boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_m\}$ must be interpolated at the positions $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ of the n volumetric mesh nodes. Such interpolation reads

$$\boldsymbol{d}(\boldsymbol{x}) = \sum_{j=1}^{m} \boldsymbol{w}_j \varphi_j(\boldsymbol{x}), \tag{8.1}$$

where the radial bases $\varphi_j(\boldsymbol{x}) = \varphi_j(||\boldsymbol{x} - \boldsymbol{x}_j||)$ are functions that only depend on the distance between evaluation point $\boldsymbol{x}$ and control point $\boldsymbol{x}_j$. The weights $\boldsymbol{w}_j$ are computed by imposing the interpolation constraints $\boldsymbol{d}(\boldsymbol{s}_i) = \boldsymbol{s}'_i - \boldsymbol{s}_i$, after a radial basis has been centered at every constrained point $(\boldsymbol{x}_j = \boldsymbol{s}_j)$. This results in the linear system

$$\boldsymbol{A}\boldsymbol{X} = \boldsymbol{B}, \tag{8.2}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \varphi_1(\boldsymbol{s}_1) & \ldots & \varphi_1(\boldsymbol{s}_m) \\ \vdots & \ddots & \vdots \\ \varphi_m(\boldsymbol{s}_1) & \ldots & \varphi_m(\boldsymbol{s}_m) \end{bmatrix}, \quad \boldsymbol{X} = \begin{Bmatrix} \boldsymbol{w}_1 \\ \vdots \\ \boldsymbol{w}_m \end{Bmatrix}, \quad \boldsymbol{B} = \begin{Bmatrix} \boldsymbol{s}'_1 - \boldsymbol{s}_1 \\ \vdots \\ \boldsymbol{s}'_m - \boldsymbol{s}_m \end{Bmatrix}. \tag{8.3}$$

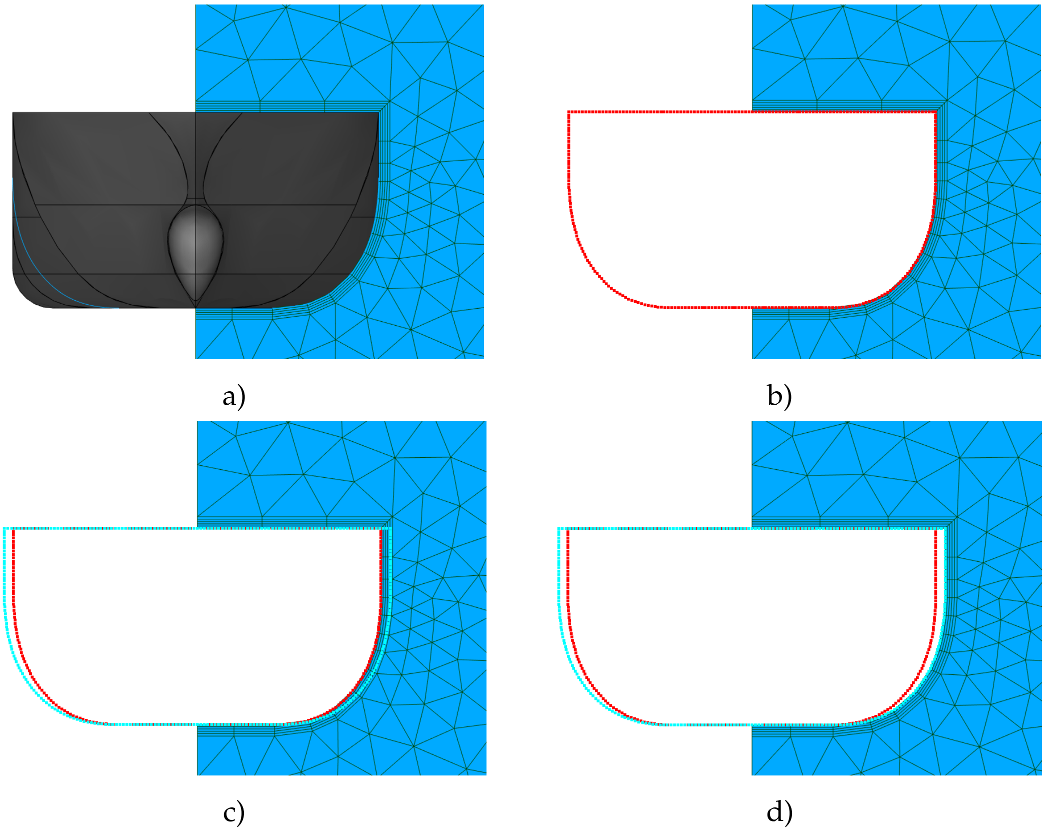


Figure 8.4: A section view example illustrating the RBF morphing steps carried out to propagate the hull surface deformations to a volumetric mesh for ship hydrodynamics simulations.

Linear system equation (8.2) is solved in a pre-processing phase, and the weights computed are then used to compute the displacement of every node of the volumetric mesh by means of equation (8.1). The latter operation can be conveniently carried out in a parallel fashion, and is highly efficient. On the other hand, $\boldsymbol{A}$ is a full $m \times m$ matrix which can make the solution of system equation (8.2) quite time and memory demanding when a large number of RBF control points are considered. That is why, in some cases only a portion of the surface mesh nodes are used as RBF control points, which limits the computational cost more than linearly, and in most cases has only modest effect on the morphing accuracy.

Both the FFD and RBF algorithms briefly described in this section have been implemented in the Python library for geometrical morphing PyGeM [269], which has been used to produce all the deformed geometries and computational grids used in this work. An example of the RBF application to volumetric mesh morphing described in this paragraph is presented in figure 8.4. The figure illustrates all the steps involved in the procedure, which starts with a) a first volumetric mesh around the hull, and b) a surface mesh on the hull surface. In step c) the latter mesh is then deformed and d) the surface mesh displacement field is finally used to feed the RBF algorithm and propagate the boundary motion to the internal volumetric mesh nodes. As it can be appreciated in the illustration, to avoid distortion of the volumetric mesh symmetry plane, the surface mesh must include both sides of the hull. In the present work, the deformation of the surface mesh has been carried out by means of FFD. Yet, we remark that any deformation law which results in a one to one correspondence between original and deformed surface grids can be propagated to the nodes of the volumetric mesh with RBF interpolation.

8.3 THE MATHEMATICAL MODEL FOR INCOMPRESSIBLE FLUIDS

The computational gain of the proposed pipeline is obtained by using a model order reduction based on POD to approximate the solution of the parametric PDEs describing the studied phenomenon. This technique assumes an initial solutions database produced by solving the FOM, for some values of the parameters. We refer to such solutions as high-fidelity solutions, or *snapshots*. Depending on the intrusiveness of the reduced order method, also the discrete operators of the numerical problem can be required. In this contribution, we propose a non-intrusive approach, constructing a ROM within a data driven setting using the FOM snapshots and the corresponding parameter values (described in section 8.2). This allows a modular structure where any numerical solver, also commercial, can be adopted, since the ROM relies only on input and output couples.

The following paragraphs present the full order model used in this work and the ROM constructed with it. We briefly describe the incompressible Reynolds Averaged Navier–Stokes (RANS) equations and its numerical solution in a finite volume framework, then we proceed with an algorithmic analysis of the POD-GPR.

8.3.1 *The full order model: incompressible RANS*

The FOM used in this work is the RANS model complemented by a Volume of Fluid (VOF) front capturing method to deal with the multi phase nature of the fluid surrounding the hull. The resulting govern equations are discretized by means of a Finite Volumes (FV) strategy implemented in the open source library openFOAM [284]. Such mathematical and numerical setup is nowadays adopted in many industrial naval contexts thanks to its robustness and accuracy. The test case considered is one of the tutorials of the library, which is designed to reproduce the DTC experiments reported in reference [182]. We here provide a minimal overall description of the model. We refer to the original documentation of the library for all the numerical and technical details.

The RANS equations model the turbulent incompressible flow, while the VOF technique [118] is applied to handle the biphase nature of the fluid (water and air). The equations governing our system are the following

$$\begin{cases} \frac{\partial \bar{u}}{\partial t} + (\bar{u} \cdot \nabla)\bar{u} - \nabla \cdot (\tilde{u} \otimes \tilde{u}) = -\frac{1}{\rho}\nabla \bar{p} + \nabla \cdot \nu \nabla \bar{u} + g, \\ \nabla \cdot \bar{u} = 0, \\ \frac{\partial \alpha}{\partial t} + \nabla \cdot (\bar{u}\alpha) = 0, \end{cases} \tag{8.4}$$

where $\bar{u}$ and $\tilde{u}$ refer to the mean and fluctuating velocity after the RANS decomposition, respectively, $\bar{p}$ denotes the mean pressure, ρ is the density, ν the kinematic viscosity, and α is the discontinuous variable belonging to interval $[0, 1]$ representing the fraction of the second flow in the infinitesimal volume. Finally, vector g represents the body accelerations associated with gravity.

The first two equations are the continuity and momentum conservation, where the new term, the Reynolds stresses tensor $\tilde{u} \otimes \tilde{u}$, have to be modeled with additional equations in order to close the system. Among all the turbulence models available in literature, we use the SST$k - \omega$ turbulence model [177]. The third equation represents the transport of the VOF variable α. Such variable controls also the density ρ and the kinematic viscosity ν, since they are defined using an algebraic formula expressing them as a convex combination of the corresponding properties of the two flows such that

$$\rho = \alpha\rho_{\text{air}} + (1 - \alpha)\rho_{\text{water}}, \qquad \nu = \alpha\nu_{\text{air}} + (1 - \alpha)\nu_{\text{water}}. \tag{8.5}$$

To compute the steady solution in a discrete environment, we apply the finite volume (FV) approach. We set a pseudo–transient simulation, applying a first order implicit local scheme for the temporal discretization, while for the spatial scheme we apply the linear upwind one. Regarding the software, as mentioned the simulation is carried out using the C++ library OpenFOAM [284].

8.3.2 *The reduced order model: POD-GPR*

In this section we briefly present the POD-GPR approach, and we refer to section 5.1 for a more general presentation of POD-based data-driven model order reduction.

POD is a linear dimensional reduction technique capable to construct a reduced order model from a set of high-fidelity snapshots. Such space is spanned by (typically few) basis functions, that are computed by minimizing the error between the original snapshots and their orthogonal projection [281]. In a parametric context, it enables — provided a proper set of parameter samples — the possibility to approximate the solution manifold in a very efficient way. Formally, we define the set of parameters $\{\boldsymbol{\mu}_i\}_{i=1}^M$ such that $\boldsymbol{\mu}_i \in \mathbf{P} \subset \mathbb{R}^p$ for $i = 1, \dots, M$. For each parameter, the solution is computed using the FOM. Let $\mathcal{N}$ be number of degrees of freedom of the full simulation, we obtain the solutions $\mathbf{x}_i \in \mathbb{X}_i^{\mathcal{N}}$ for $i = 1, \dots, M$. Since the finite volume space is created only once and then it is deformed, all the geometric configurations have the same dimensionality even if they belong to different spaces. The vectorial solutions are arranged as columns of the snapshots matrix, such that

$$\mathbf{X} = \begin{bmatrix} | & \dots & | \\ \mathbf{x}_1 & \dots & \mathbf{x}_M \\ | & \dots & | \end{bmatrix} \in \mathbb{R}^{\mathcal{N} \times M}. \tag{8.6}$$

The basis of the POD space, composed by the so called POD modes, is computed using the SVD of the snapshots matrix $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$. The unitary matrix $\mathbf{U} \in \mathbb{R}^{\mathcal{N} \times M}$ contains the left-singular vectors of $\mathbf{X}$, which are the POD modes. Moreover the diagonal matrix $\mathbf{\Sigma} = \text{diag}(\lambda_1, \dots, \lambda_M)$, where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_M$, contains the singular values, which indicate the energetic contribution of the corresponding modes. By looking at the spectral decay we can retain the first N most energetic modes, which span the optimal space of dimension N.

Such basis can be exploited to project onto the reduced space the initial snapshots. Thus we can approximate the snapshots $\mathbf{x}_j$ as a linear combination of the modes as

$$\mathbf{x}_j = \sum_{i=1}^{M} \mathbf{c}_j^i \boldsymbol{\psi}_i \approx \sum_{i=1}^{N} \mathbf{c}_j^i \boldsymbol{\psi}_i \quad \text{for } j = 1, \dots, M, \tag{8.7}$$

where $\boldsymbol{\psi}_i$ refers to the i-th POD mode. The coefficients $\mathbf{c}_j^i$ of the linear combination represent the low-dimensional solution and are usually called *modal coefficients*. Using the matrix notation, to compute such coefficients it is sufficient a matrix multiplication $\mathbf{C} = \mathbf{U}_N^T \mathbf{X}$, where the columns of $\mathbf{C}$ are the vectors $\mathbf{c}^j \in \mathbb{R}^N$ for $j = 1, \dots, N$, the matrix $\mathbf{U}_N \in \mathbb{R}^{\mathcal{N} \times N}$ contains the first N POD basis and the superscript T indicates the matrix transpose.

The new pairs $(\boldsymbol{\mu}_i, \mathbf{c}_i)$, for $i = 1, \dots, M$, can be exploited in order to find a function $f : \mathbf{P} \to \mathbb{R}^N$ capable to predict the modal coefficients for untested parameters. As anticipated, in this work we apply a GPR [287], fitting the distribution of the modal coefficients with a multivariate Gaussian distribution, such that

$$f(\boldsymbol{\mu}) \sim \mathrm{GP}(m(\boldsymbol{\mu}), K(\boldsymbol{\mu}, \boldsymbol{\mu})), \tag{8.8}$$

where $m(\cdot)$ and $K(\cdot, \cdot)$ indicate the mean and the covariance of the distribution, respectively. Given a covariance function, an optimization step is required to set the corresponding hyperparameters. In this contribution we use the squared exponential covariance defined as $K(x_i, x_j) = \sigma^2 \exp\left(-\frac{\|x_i - x_j\|^2}{2l}\right)$. Once the hyperparameters (σ and l) of the covariance kernel have been fit to the input dataset, we can query such distribution to predict the new modal coefficients . Finally the modal coefficients are projected back to the high-dimensional vector space $\mathbb{R}^{\mathcal{N}}$ using equation (8.7).

On the technical side, we construct and use the POD-GPR model using EZyRB [70], an open source Python package which deals with several data-driven model order reduction techniques, exploiting the library GPy [91] for the GPR implementation.

8.4 NUMERICAL RESULTS

In this section, we describe the application of the proposed optimization pipeline to the DTC hull surface. The reader can refer to chapter 6 for all the details regarding the ASGA optimization algorithm. Table 8.1 shows the main particulars in the design loading condition at model scale (which is set to $1 : 59.407$). This will provide a test case which closely simulates a typical workflow for industrial hull design problems. Figure 8.5 shows the original CAD geometry of the hull used in this work, where we marked 21 longitudinal sections which divide the ship into 20 equispaced chunks. Such 21 slices will be referred to as *sections* during the results discussion, and are numbered from 1 to 21 going from the ship stern to its bow.

The structure of this section mirrors that of the whole article, reporting the intermediate results of all the methods employed throughout the optimization pipeline.

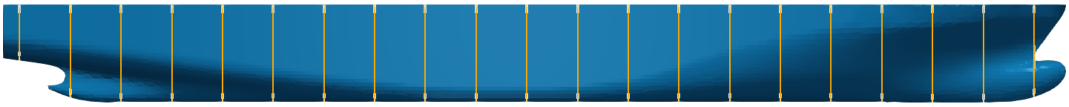


Figure 8.5: The surface of the DTC hull. The highlighted sections divide the ship into 20 equispaced chunks at the free-surface level.

8.4.1 *Self-learning mesh morphing parameters*

To set up the FFD hull surface deformation, we position the control points lattice in order to control the immersed part of the ship prow region. The equispaced control points are positioned as follows:

- ***x* axis:** 7 points layers located on sections 10, 12, 14, 16, 18, 20, 22;
- ***y* axis:** 11 points layers that cover the whole hull beam, with the second and the second-to-last positioned on the lateral walls of the ship;
- ***z* axis:** 7 points layers that cover the whole hull draft, aligning the 2^{nd} and the 5^{th} of them to the hull bottom and to the waterline, respectively.

As can be appreciated by the values reported, to distribute the FFD control points, we have made use of an additional 22nd *virtual* section located ahead of the bow. The motion of the $7 \times 11 \times 7 = 539$ points is governed by only 10 parameters, which are described in table 8.2. We point out that the displacement of all the boundary points in the x and z direction is set to zero so as to enforce surface continuity. In addition, the displacement of the points on the internal x and z layers closest to the boundary ones is also set to zero so as to enforce continuity of all surface derivatives. Finally, the hull symmetry along y direction is ensured by selecting symmetric values for parameters associated to x and z displacements, as well as antisymmetric values for parameters associated to y displacements (the latter points are also indicated in the table by the corresponding footnote).

Once defined the geometric parameters $\boldsymbol{\mu} = [\mu_0, \dots, \mu_9]$, we set the parametric space to $\mathbf{P} = [-0.2, 0.2]^{10}$. The parameter space boundary values are selected so as to obtain feasible deformations from an engineering point of view and, at same time, to explore a large variety of possible shapes. Figure 8.6 shows the two "extreme" hull deformations, obtained setting all the parameters equal to the lower and upper bound of the space, respectively.

Table 8.1: Main quantities of the DTC at scale model.

Quantity	**Value**
Length between perpendiculars L_{pp} [m]	5.976
Waterline breadth B_{wl} [m]	0.859
Draught midships T_m [m]	0.244
Volume displacement V [m^3]	0.827
Block coefficient C_B	0.661

Table 8.2: FFD control points displacement. The indices refer to the relative position of the points within the lattice.The layers order, which starts from 0, is maintained consistent with the reference system. The intervals indicated by the – symbol are inclusive.

Lattice Points			Parameter	Displacement direction
index x	index y	index z		
2	0	2–4	μ_0	x
2	10	2–4	μ_0	x
3	0	2–4	μ_1	x
3	10	2–4	μ_1	x
4	0	2–4	μ_2	x
4	10	2–4	μ_2	x
4	2–4	2	μ_3	y
4	6–8	2	$-\mu_3$ [1]	y
4	2–4	3	μ_4	y
4	6–8	3	$-\mu_4$ [1]	y
4	2–4	4	μ_5	y
4	6–8	4	$-\mu_5$ [1]	y
3	2–4	2	μ_6	y
3	6–8	2	$-\mu_6$ [1]	y
5	2–4	3	μ_7	y
5	6–8	3	$-\mu_7$ [1]	y
4	0–1	2	μ_8	z
4	9–10	2	μ_8	z
5	0	3	μ_9	z
5	10	3	μ_9	z

The FFD deformation of the hull points has been extended to the nodes of the volumetric grid for the CFD simulations making use of the Beckert-Wendland radial basis function kernel [17], defined as follows

$$\varphi_j(||\boldsymbol{x} - \boldsymbol{x}_j||) = \left(1 - \frac{||\boldsymbol{x} - \boldsymbol{x}_j||}{R}\right)_+^4 \left(1 + 4\frac{||\boldsymbol{x} - \boldsymbol{x}_j||}{R}\right), \tag{8.9}$$

where $R > 0$ is a prescribed finite radius and the $(\cdot)_+$ symbol indicates the positive part.

The output of the OpenFOAM library checkMesh utility has been used to assess the quality of the grids obtained with the combined FFD-RBF methodology. Figure 8.7 presents some of the main quality indicators of the 200 meshes generated for the present campaign, as computed by checkMesh. In particular, the indicators considered are minimum face area (top left plot), minimum cell volume (top right plot), maximum mesh non-orthogonality (bottom left plot) and average mesh non-orthogonality (bottom right plot). In all the diagrams, the vertical axis refers to the mesh quality indicator considered, while the variable associated with the horizontal

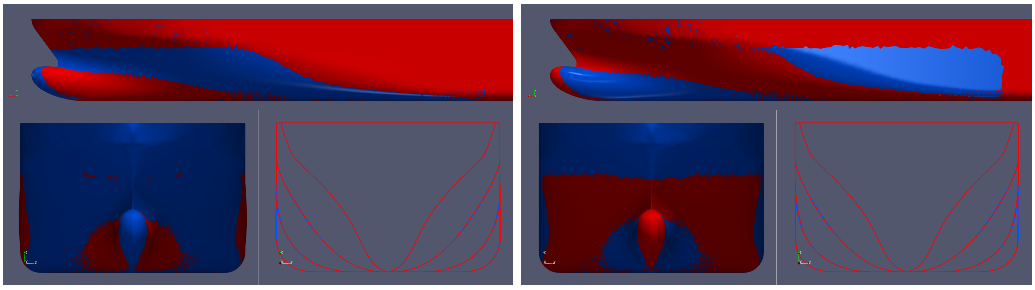


Figure 8.6: Visual examples of hull deformation with $\boldsymbol{\mu} = [-0.2]^{10}$ (on left) and $\boldsymbol{\mu} = [0.2]^{10}$ (on right). The red surface refers to the deformed ships, while the blue one is the original hull.

axis is the index corresponding to each of the 200 volumetric meshes produced for the simulation campaign.

The minimum face area and minimum cell volume results indicate that the morphing procedure does not produce negative cells or faces which would impair the simulations. In fact, the average of both indicators across the 200 grids produced is extremely close to the corresponding value of the original grid. The lowest value of minimum face area observed in the 200 grids generated is less than 0.1% off the original value, while the lowest value of minimum cell volume observed is merely 0.01% off the original mesh minimum cell volume. Such trend is confirmed by the maximum non-orthogonality values reported in the bottom left diagram. In the plot, is possible to appreciate that the average over the 200 grids produced falls exactly on value of the original mesh, and the highest difference with respect to the original mesh non-orthogonality is merely 0.05%. These values ensured that all the simulations in the present campaign could be completed in fully automated fashion without crashes were reported or significant issues were observed. The results reported in the bottom right plot indicate that the effect of the mesh morphing algorithm proposed is that of increasing the grid average non-orthogonality values. This is somewhat expected, as the original volumetric grid in this work was generated making use of the snappyHexMesh tool of the OpenFOAM library. In such framework, most of the cells in the internal regions of the domain are substantially the result of an octree refinement of an original block mesh aligned with the coordinate axes. It is clear that the RBF procedure described in section 8.2 does quite clearly alter in a non negligible way the orthogonal angles of a portion of the hexahedral cells produced by snappyHexMesh. Yet, the average increase in the average mesh non-orthogonality index is 2%, while the maximum increase observed is 7.2%, which are values that should not significantly affect the results of the simulations.

8.4.2 *Reduced order model construction*

We set the full order model in scale 1 : 59.407, keeping it unaltered from the original work mainly for validation purpose. The computational domain, that is a parallelepiped of dimension $[-26, 16] \times [-19, 0] \times [-16, 4]$ along x, y and z

[1] Imposed for y symmetry conservation.

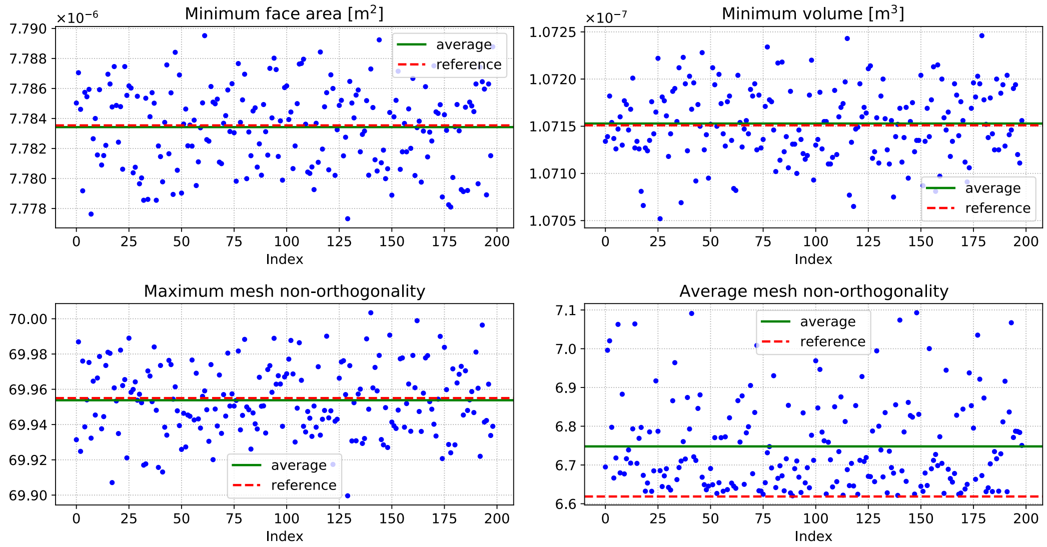


Figure 8.7: Values of the main mesh quality indicators as reported by checkMesh utility of OpenFOAM library, as a function of the index corresponding to each of the 200 volumetric meshes produced for the simulation campaign.

directions is discretized in 8.5×10^5 cells, with anisotropic vertical refinements located particular in the free-surface region, in order to avoid a too diffusive treatment of the VOF variable. Boundaries of such domain are imposed as follows:

- at the *inlet* we set constant velocity, fixed flux condition for the pressure and a fixed profile for the VOF variable;
- at the *outlet* we set constant average velocity, zero-gradient condition for the pressure and variable height flow rate condition for VOF variable;
- at the bottom and lateral planes, we impose symmetric conditions for all the quantities;
- at the top plane, we set a pressure inlet outlet velocity condition for the velocity and nil pressure; VOF variable is fixed to 1 (air);
- at the hull surface, we impose no-slip condition for velocity, fixed flux condition for the pressure and zero-gradient condition for VOF variable.

The adopted solver is *interFoam*, which is able to solve the Navier–Stokes equations for two incompressible, isothermal immiscible fluids. Time discretization uses a first order implicit scheme with local-step, since we are interested to the steady solution. For the spatial discretization, we apply a Gaussian integration using second order upwind scheme for divergence operators and linear interpolation for gradient and laplacian operator. By imposing a inlet velocity of 1.668 m/s, the Froude number is around 0.22. The time required to converge to the steady solution within such setting on a parallel machine (32 processors) is approximately 2 hours.

For the construction of the reduced order model, we randomly sample the parametric space with uniform distribution. We performed 203 simulations with the full

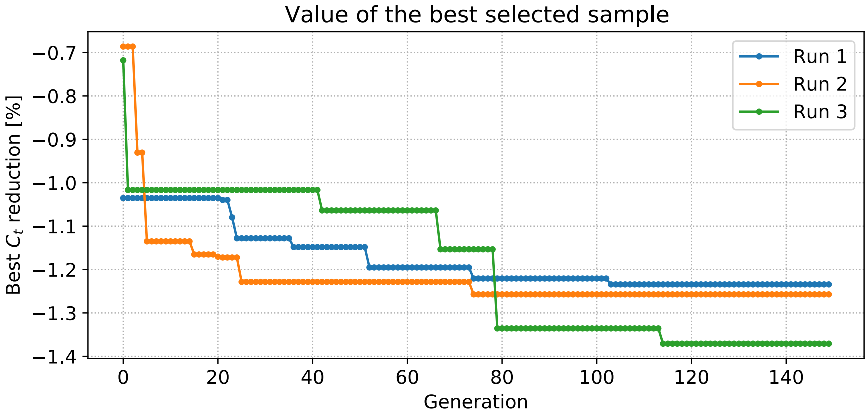


Figure 8.8: ASGA runs. The reduction of the C_t is to be intended with respect to the undeformed reference hull.

order model, collecting the corresponding pressure and shear stress distributions (the latter implicitly containing the distribution of the VOF variable) over the hull surface. Thus, only the surface fields are considered at the reduced level. We then flatten the shear stress vector field in order to construct two snapshots matrices, one for the pressure and one for the stress. Both are then decomposed using POD technique. The number of modes considered is fixed to 20. Approximating the manifold with the GPR method, we obtain two different POD-GPR model that approximate the pressure field and the shear stress field. Such quantities are used for the computation of the objective function during the optimization procedure.

Even if the difference of hardware used for full order model simulations and for reduced order approximation limits the possible speedup obtained — a HPC facilities versus an ordinary personal computer —, we achieve satisfactory computational gain. In fact, whereas the FOM lasts approximately two hours, the ROM approximation only consisting in two distribution queries and two matrix multiplications, takes less than 1 second in a single-processor environment. Such results are very effective in the framework of an iterative process, as the optimization pipeline here proposed. The overall time is in fact mainly constituted by the initial FOM simulations needed for the offline database, while the ROM approximation can be considered negligible from the computational point of view. Moreover, it can be performed on significantly less powerful machines.

Adopting data-driven methodologies rather than projection-based ones has different advantages which we have already discussed, but shows also some drawback in the error bounding. For an a posteriori quantification of the ROM accuracy we need then to validate the approximated optimal result by carrying out a FOM simulation. We remark that we consider the output of such simulation as truth solution. This requires an additional computational cost, but allow also for an effective refinement of the ROM. Once a geometrical configuration is validated in such fashion, depending on the error observed we can add this last snapshot to the database and re-build the ROMs.

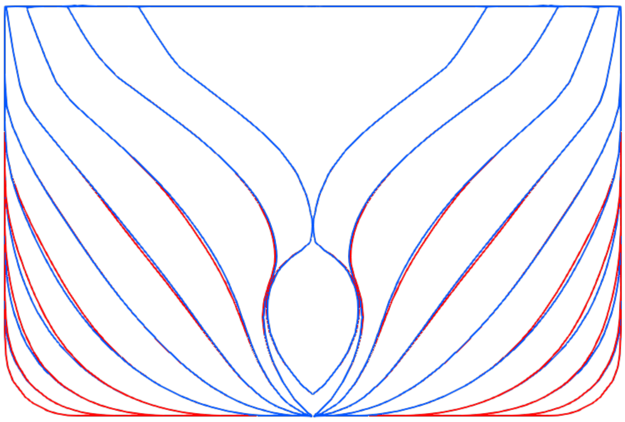


Figure 8.9: The sections (from 10 to 20) of the original ship in blue and of the optimized one in red.

8.4.3 *Optimization procedure*

We first define the objective function we applied to the optimization procedure. The quantity to minimize is the total resistance coefficient C_t, which is defined as

$$\min_{\mu} C_t \equiv \min_{\mu} \int_{\Omega(\mu)} \frac{\tau_x \rho - p n_x}{\frac{1}{2} \rho V^2 S}, \tag{8.10}$$

where τ_x is the x-component of the shear stress, ρ is the fluid density, p indicates the pressure, n_x the x-component of the surface normal, V and $S = \Delta^{2/3}$ the reference fluid velocity and the reference surface, respectively. As reported, the CFD simulations have been carried out in fixed sink and trim conditions. Thus, the specific reference surface used to obtain C_t has been selected to penalize hulls obtaining resistance gains through immersed volume reduction. All the geometrical quantities, as well as the normals and the reference surface depend by the imposed deformation. Thus, to evaluate the C_t for any design, we deform the hull surface using the FFD map, then project the ROM approximated fields — pressure and shear stress — on it to numerically compute the integral defined in equation (8.10).

Regarding the ASGA hyperparameters, we set the probability of crossover and mutation as $P_C = P_M = 0.5$. For each solutions database we perform an optimization run with ASGA composed by 150 generations, with an initial random population of 100 individuals and an offspring of 20 individuals. The number of points returned by the AS back mapping is $B = 2$, while the dimension of the AS is set to 1 for every population. The covariance matrix for the active subspace computation is approximated using local linear models [54].

For each optimum found by ASGA we run a new high-fidelity simulation for validating the approximated C_t, adding the high-fidelity snapshots to the database in order to refine the POD-GPR model. In figure 8.8 we show the comparison of all the runs. The third and last optimization reached a reduction of $\sim 1.4\%$ of the C_t coefficient compared to the original shape.

Figure 8.9 presents the frontal sections of the optimal shape compared to the undeformed one, showing a volumetric increment in the frontal part which balances the reduction near the central zone. The a posteriori validation confirmed the positive trend: the C_t coefficient of the optimal shape is 1.2% less, with a relative error of the ROM model of 0.18%. As is appreciable in figure 8.9, the optimal hull has a wider section in the region immediately downstream with respect to the bulbous bow, while it appears slightly narrower in the middle ship sections. The immersed volume of the optimized hull is only 0.08% different from that of the original hull, which suggests that the C_t reduction obtained is the result of a total resistance reduction. A possible interpretation of such a resistance decrease is that having a more streamlined hull along the longitudinal direction, is likely able to reduce the extent and dimension of the separation bubble located on the side of the bulbous bow, and corresponding to the dark blue strip visible in the wall shear stress contours presented in figure 8.10 and figure 8.11. As a consequence, the optimal hull presents slightly lower pressures with respect to the original hull, in the region located downstream of the bulbous bow. Such a minimal reduction is hardly noticeable in the pressure contour plots presented in figure 8.12 and figure 8.13. More appreciable differences are visible instead in the free surface elevation plot presented in figure 8.14. Reducing the extent of the aforementioned detachment bubble, the shape modification leading to the optimal hull has the effect of moving forward the trough which follows the bow. This indicates that the pressures in the bow region are reduced, which results in a net decrease of the resistance pressure component. In fact, this leads to a 4.92% reduction in the pressure component of the resistance, against a more modest 0.55% reduction of viscous resistance. Yet, considering that the latter component accounts for approximately 83% of the total resistance, this translates into the 1.2% reduction reported. Finally, to exclude the possibility that the differences observed in the total resistance coefficient values are a result of possible discretization error due to the mesh morphing procedure, we report that the average and maximum values of wall y^+ of the optimized hull do not significantly differ from those obtained with the original one. The average and maximum wall y^+ values for the original hull simulation are 6.18426 and 99.5631, respectively, while the corresponding average and maximum values for the optimized hull are 6.19071 and 99.6255, respectively. We point out that the y^+ maxima here reported for the DTC tutorial appear outside of the range prescribed for the turbulence model here used. Yet, the accuracy of the DTC tutorial results suggests that maxima y^+ is likely located outside the water. In fact, considering the small density of air with respect to water, the impact of the resulting inaccurate estimation of surface derivatives is minimal.

We remark that the POD-GPR model approximates the distribution of the output of interest, not the objective function — which is computed using the predicted fields. For this reason, we can also compare the pressure and shear stresses over the optimal hull with respect to the undeformed one. Figure 8.10 and figure 8.12 present the graphical investigations about the ROM approximation error distribution over the undeformed hull, both for pressure and stresses distributions. For a more realistic comparison, we specify that the FOM snapshots referring to the undeformed geometry has been removed from the database, emulating the approximation any untested parameter. We proceed in the same way also for the optimal shape (figure 8.11 and figure 8.13), not only to measure the accuracy of the POD-GPR model, but also for investigating the reasons of the C_t reduction from a physical perspective.

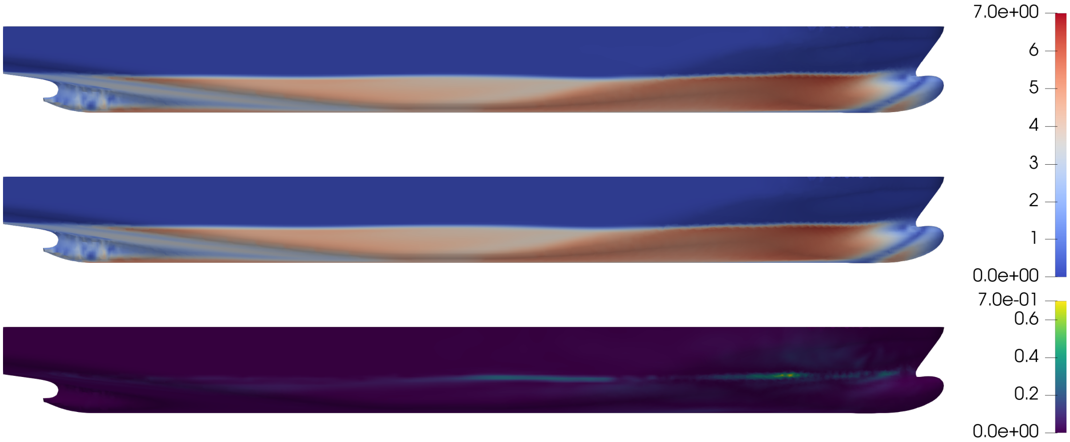


Figure 8.10: Distribution of the shear stresses measured in Pascal over the undeformed hull: the FOM validation (top) is compared to the ROM approximation (middle) and the absolute error is shown (bottom).

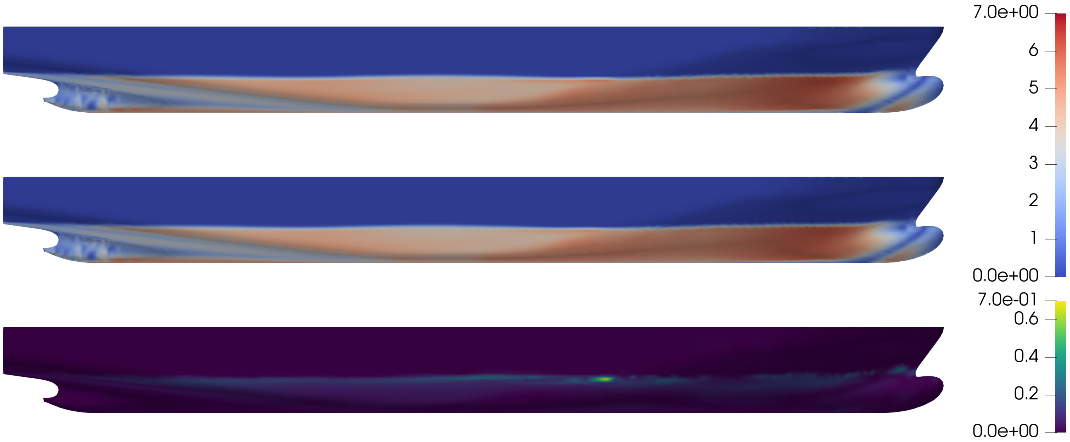


Figure 8.11: Distribution of the shear stresses measured in Pascal over the optimal hull: the FOM validation (top) is compared to the ROM approximation (middle) and the absolute error is shown (bottom).

The absolute error is quite small, but it is possible to note that for both the fields it is mainly concentrated along the free-surface.

Comparing the original hull with the optimal one we emphasize that the optimal shape seems to be able to slightly reduce the height of the wave created by its body, inducing a reduction of the wet surface. The friction resistance computed as the integral of the x component of shear stresses over the two hulls shows in fact this marginal gain: the 12.76 N of the original ship becomes 12.69 N in the optimal configuration. However, the main contribution of the resistance reduction comes from the pressure resistance. While in the original shape we measure 2.64 N, in the optimized such quantity decreases to 2.51 N.

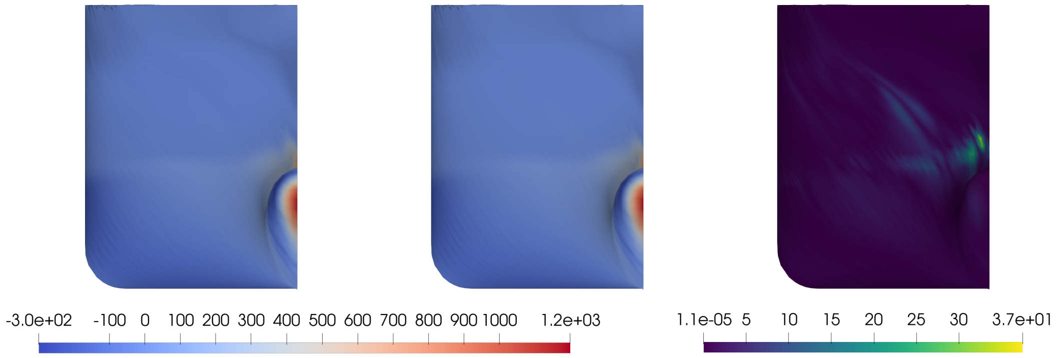


Figure 8.12: Distribution of pressure measured in Pascal over the undeformed hull: the FOM validation (left) is compared to the ROM approximation (center) and the absolute error is shown (right).

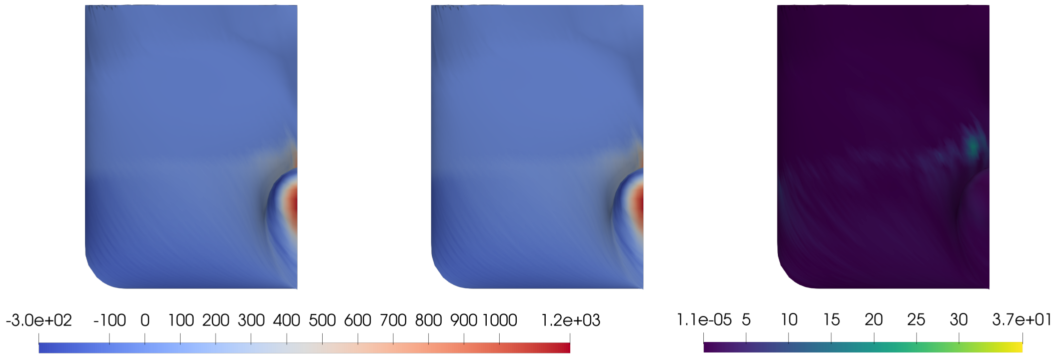


Figure 8.13: Distribution of the pressure measured in Pascal over the optimal hull: the FOM validation (left) is compared to the ROM approximation (center) and the absolute error is shown (right).

8.5 CONCLUSIONS

In this chapter we presented a complete numerical pipeline for the hull shape design optimization of the DTC benchmark hull. We proposed a self-learning geometrical deformation technique, where different morphing methods are coupled together to propagate surface deformations to volumetric meshes. Though in this work we used a FFD approach for the CAD modifications, we emphasize that our methodology can exploit any surface deformation. The optimization procedure is based on the coupling between active subspaces and genetic algorithm, called ASGA. This also present an actual application of the new optimization algorithm in an industrial context. For the evaluation of the total resistance coefficient for new untested parameters we exploits the non-intrusive data driven reduced order method called POD-GPR. This results in a great computational saving for the computation of the pressure and viscous forces fields, while preserving a good accuracy. We performed 3 optimization runs, with high-fidelity validation of the approximated optimum and enrichment of the solutions database to increase the accuracy of the ROM in its neighborhood. We obtained a reduction of the total resistance coefficient equal to 1.2% with respect to the original reference hull.

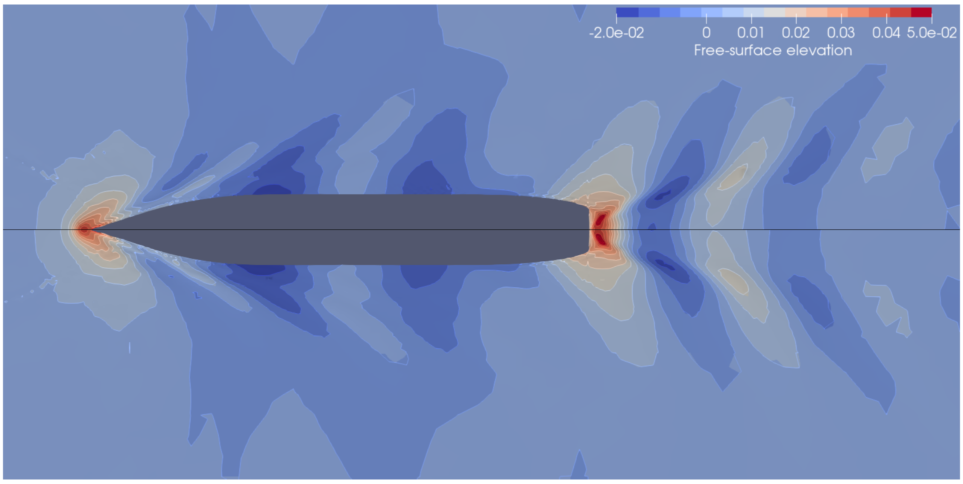


Figure 8.14: Contours of free surface elevation field around the original hull (top half) and optimal (bottom half).

In the future, further investigations will be carried out to study a dynamic selection of the active subspace dimension, and a varying number of points returned by the back mapping procedure. Further improvements in the shape parameterization algorithms could be obtained improving the efficiency of the RBF weights computation. This could be obtained with a smarter selection of the RBF control points or, in a more invasive fashion, by resorting to fast algorithms — such as Fast Multipole Method [64] — for the computation of the control points mutual distances.

Part IV

ACADEMIC CFD APPLICATIONS

9

ON THE COMPARISON OF LES DATA-DRIVEN REDUCED ORDER APPROACHES FOR HYDROACOUSTIC ANALYSIS

In this work, DMD and POD methodologies are applied to hydroacoustic dataset computed using Large Eddy Simulation (LES) coupled with Ffowcs Williams and Hawkings analogy. First, a low-dimensional description of the flow fields is presented with modal decomposition analysis. Sensitivity towards the DMD and POD bases truncation rank is discussed, and extensive dataset is provided to demonstrate the ability of both algorithms to reconstruct the flow fields with all the spatial and temporal frequencies necessary to support accurate noise evaluation. Results show that while DMD is capable to capture finer coherent structures in the wake region for the same amount of employed modes, reconstructed flow fields using POD exhibit smaller magnitudes of global spatiotemporal errors compared with DMD counterparts. Second, a separate set of DMD and POD modes generated using half the snapshots is employed into two data-driven reduced models respectively, based on DMD mid cast and PODI. In that regard, results confirm that the predictive character of both reduced approaches on the flow fields is sufficiently accurate, with a relative superiority of PODI results over DMD ones. This infers that, discrepancies induced due to interpolation errors in PODI is relatively low compared with errors induced by integration and linear regression operations in DMD, for the present setup. Finally, a post processing analysis on the evaluation of Ffowcs Williams and Hawkings acoustic signals utilizing reduced fluid dynamic fields as input demonstrates that both DMD and PODI data-driven reduced models are efficient and sufficiently accurate in predicting acoustic noises. All the results presented in this chapter appeared in [92].

9.1 LITERATURE REVIEW

In several engineering fields, there has been recently a growing need to include fluid dynamic performance evaluation criteria associated with acoustic emissions. This is, for example, the case in aircraft engine design, car manufacturing, and ship design optimization. A particular motivation behind the present research article is to investigate and propose a methodology that can be successively used for the noise level prediction of naval propellers since the early design process [52, 123].

The need for a reduction of the acoustic emissions through ship design optimization usually involves virtual prototyping and parametric high fidelity simulations. Thanks to the increase of the available computational resources, a deeper insight towards the complex physics associated with hydroacoustic phenomena has become nowadays affordable with unprecedented spatial and temporal scales (see, for example, wall-resolving LES [145, 207]). However, the enormous data sizes resulting from such simulations pose several challenges on the input/output operations, post-processing, or the long-term data storage. On the other hand, hybrid techniques such as, among others, Detached-Eddy Simulation or wall-layer model LES (WLES), have allowed obtaining eddy resolving field data with a reasonable use of computational resources. These techniques are, however, still expensive for an early stage design process, when a number of different geometric configurations has to be rapidly analysed to restrict the range of variation of the principal design parameters. Therefore, seeking a suitable data compression strategy that allows extracting the most relevant and revealing information in a reduced order manner, hence providing quick access as well as efficient data storage, becomes a crucial asset.

Through multidisciplinary scale, efforts have been made to realize optimal shape design for underwater noise sources, including ship hulls [43, 267, 272] and propellers [184], using efficient geometrical parameterization techniques [93, 269]. On the acoustic side, the development of new generation noise prediction tools was considered a major focus in this work. Particularly, based on the Ffowcs Williams and Hawkings (FWH) analogy [86], several improvements have been developed. For example, in [50] they compared several implementations of the non-linear quadrupole term, highlighting its significant contribution to the overall hydrodynamic noise emissions in wide range of frequencies. In a companion paper [49], they showed the effect of shape deformation on the radiated noise for elementary geometries. On a more engineering level, the generated hydrodynamic noise from a benchmark marine propeller was evaluated in open sea conditions [51, 52] using FWH coupled with LES.

The aforementioned hydroacoustic models are typically described by a system of non-linear PDEs, the resolution of which results in the fluid dynamic fields necessary to reproduce the noise source for an acoustic predictions. In fact, a reliable reconstruction of the noise source is crucially dependent on the flow structures and the resolved spatial and temporal scales of the fluid dynamic fields (for a discussion, see [34]).

Besides LES works [12, 49–51, 145, 192, 245], several fluid dynamic models have been employed in marine hydroacoustics. To name a few, we mention the potential flow theory [139], boundary element method [243], RANS [10, 102, 123], DES [174], and Direct Numerical Simulation (DNS) [196, 232, 242]. Being regarded as an optimal advance between RANS and DNS, recent literature has reported LES to be the most suitable model which reproduces the noise source with high level of realism [12, 122, 192]. The resolution of the described system of PDEs using standard discretization methods (finite elements, finite volumes, finite differences), which we will refer hereafter as the FOM, allows for high fidelity acoustic evaluation.

Although hybrid fluid dynamic techniques constitute a good compromise between accuracy and computational cost for engineering purposes, they are still expensive in the case of parametric analysis and shape optimization. To overcome this issue, the development of a ROM [112, 226, 227, 230] — which alleviates both the computational complexity and data capacity — becomes essential. Moreover, several ROM developments have been realized to account for various effects including subgrid scales [116], heat transfer [98] and stabilization of the Galerkin projection [259].

One of the necessary assumptions to construct an efficient ROM is that, the solution manifold of the underlying problem lies in a low dimensional space and, therefore, can be expressed in terms of a linear combination of a few number of global basis functions (reduced basis functions). Among various techniques to generate such reduced basis set, the POD and the DMD have been widely exploited due to their versatile properties [261]. POD provides a set of orthogonal and optimal basis functions [147], whereas DMD computes a set of modes with an intrinsic temporal behavior, hence it is particularly suited for time advancing problems [235]. One of the main goals of this work is to understand whether these two established techniques are able to generate global basis functions which include all the wide range of frequencies associated with hydroacoustic phenomena, and therefore allow for accurate noise predictions. In addition, the interest is in assessing the efficiency of the modal decompositions algorithms, evaluating the amount of modal shapes required to accurately reproduce all high frequencies relevant to acoustic analysis. This investigation is of course a first, fundamental step towards developing efficient reduced order models for hydroacoustic applications. As will be thoroughly discussed, in the present work we only considered data-driven ROMs, but the POD and DMD efficiency assessment reported provides valuable information also for the development of projection based ROMs.

In literature, POD has been widely used for the past few decades to identify the coherent structures of turbulent flows (see, among others, [22, 250]), and has been applied towards various flow conditions [178, 225]. Correspondingly, DMD has been also exploited [235, 241]. An intuitive question may arise considering the comparative performance. In that regard, several works have carried out both DMD and POD on various flow configurations. For instance, Liu et al. [295] conducted comprehensive analysis, concluding that DMD has the ability to clearly separate the flow coherent structure in both spatial and spectral senses, whereas POD was contaminated by other uncorrelated structures. Consistent to the previous, in [24] it was noted that DMD is useful when the main interest is to capture the dominant frequency of the phenomenon, while the optimality of the POD modes prevails for coherent structure identification that are energetically ranked. In high-speed train DES, Muld et al. [187] examined the convergence and reported that the most dominant DMD mode requires a longer sample time to converge when compared to the POD counterpart.

As discussed earlier, parametric studies such as shape optimization can result in extremely high computational costs. Possible ways to circumvent such issue could be degrading the high fidelity model, or restricting the design parameter space sampling. Since the only analysed parameter in the unsteady fluid dynamic problem is the time, in this work we decided to rely on data-driven ROMs. A significant source of error in data-driven ROMs is in fact associated with the interpolation of modal coefficients based on the problem parameters. Thus, such methods are an extremely valid alternative in presence of a one dimensional parameter space, in which interpolation errors are rather modest. In particular, PODI can be an adequate solution in such scenarios [170]. The basic idea is to exploit POD on selected ensemble of high fidelity solutions in the design space to identify the set of optimal basis functions and associated projection coefficients representing the solution dynamics. Such finite set of scalar coefficients are then utilized to train a response surface that allows predictions at parameter values that are not in the original high-fidelity ensemble. It was demonstrated that PODI can be efficiently utilized in various events: 1) enhancing the temporal resolution of experimental measurements [33], 2) optimal control [170], 3) reconstruction of incomplete data [79], 4) multi-dimensional parametric analysis [88, 290], 5) inverse design [40], 6) variable fidelity models [179], and more.

Both DMD and PODI are regarded as data-driven reduced models [37] since they operate on the snapshots produced from the FOM, and predict the system dynamics in a non-intrusive manner. In shape optimization context, DMD and PODI have been successively applied as in [66–68, 89, 267, 270] in naval engineering, [77, 105, 231] in automotive engineering, or [125, 219] in aeronautics. For acoustic analysis, only very few studies have been reported in literature. This can possibly include the DMD application as in [36, 133] or POD as in [101, 171, 248].

As previously demonstrated, there are very limited examples of ROMs specifically tailored for acoustic analysis. To the best of authors' knowledge, the literature is presently devoid of documents which characterize, in a thorough and systematic fashion, the performance of DMD and PODI techniques on the hydroacoustic flow fields reconstruction or prediction.

The overarching goal of this study is to investigate the use of DMD and PODI on a hydroacoustic dataset corresponding to turbulent incompressible flow past sphere at Re $= 5000$, and computed using wall-resolving LES for the fluid dynamic fields, and FWH analogy with direct integration of the nonlinear quadrupole terms for the acoustic fields. In particular, the objectives of this work are 1) to understand the effect of DMD/POD modal truncation on the local and global reconstruction accuracy of the fluid dynamic fields, 2) to compare the efficiency of DMD and PODI on recovering the flow and spectral information when half the dataset are utilized, 3) to evaluate the performance of DMD and PODI in terms of data compression and dipole/quadrupole acoustic prediction.

The chapter is organized as follows: first, a brief overview on the full order models (LES and FWH) and respective specific works are presented in section 9.2. In

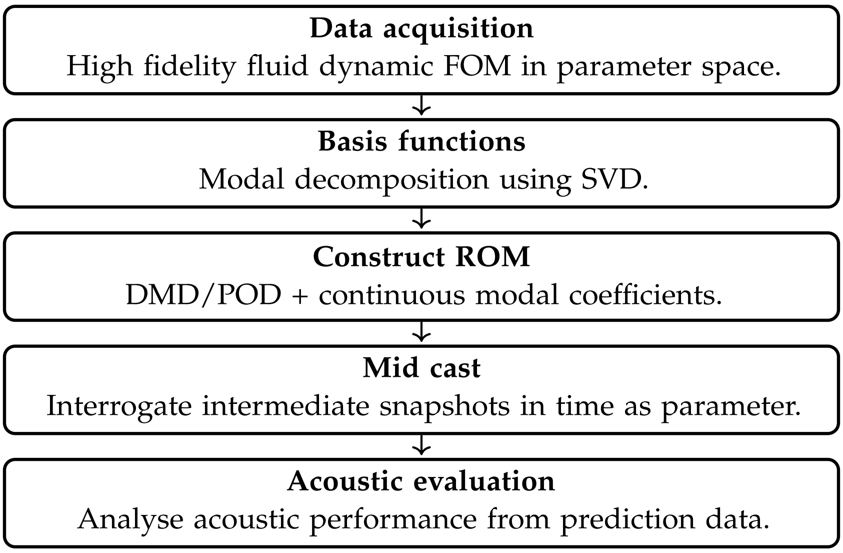

Data acquisition
High fidelity fluid dynamic FOM in parameter space.
Basis functions
Modal decomposition using SVD.
Construct ROM
DMD/POD + continuous modal coefficients.
Mid cast
Interrogate intermediate snapshots in time as parameter.
Acoustic evaluation
Analyse acoustic performance from prediction data.

Figure 9.1: Flow chart of the FOM/ROM procedure

section 9.3.1, the FOM results are presented and then followed by modal analysis with DMD and POD in section 9.3.2. For a review of the reduced order methods we refer to chapter 5. The reconstructed and predicted fluid dynamic data obtained from both ROMs are discussed in section 9.3.3 and section 9.3.4, while their spectral and acoustic performances are addressed in section 9.3.5. Conclusions are drawn in section 9.5.

9.2 METHODOLOGY

First, the high fidelity data are generated and uniformly sampled in time using high fidelity simulations with LES turbulence modelling. The resulting matrix of snapshots is then factorized using SVD which is then used to construct both the DMD and POD spaces. The scalar coefficients resulting from the projection of the FOM data onto the POD space are used to train a continuous representation of the system temporal dynamics, i.e. PODI approach. In this context, both DMD and PODI are considered as linear approximation of the dynamical system of the snapshot matrix and, therefore, are used to predict the data at intermediate timesteps. The predicted snapshots are finally exploited to run a post-processing acoustic analogy and validate the accuracy of the noise generation compared to the FOM data. A summary of the procedure is presented in the flow chart in figure 9.1. This section is organized as follows: in section 9.2.1 the FOM is introduced recalling also the utilized LES turbulence model, while in section 9.2.2 the acoustic model used to perform the hydro-acoustic analysis is introduced. The non-intrusive pipeline used to perform model order reduction using DMD and PODI can be found in chapter 5.

9.2.1 *Full order model*

The full order model, adopted to provide the snapshots dataset as input for the reduced order model, is a Large Eddy Simulation. In LES, the large anisotropic and energy-carrying scales of motion are directly resolved through an unsteady three dimensional (3D) simulation, whereas the more isotropic and dissipative small scales of motion are confined in the sub-grid space. Scale separation is carried out through a filtering operation of the flow variables. In literature, the contribution of the Sub-Grid Scales (SGS) of motion on noise generation and propagation has been found negligible [206, 245]. This means that the LES model can be considered accurate enough to provide a noise-source flow field, when compared to DNS. At the same time, the unsteady vortex and coherent structures can be of extreme importance when computing the noise signature. Indeed, it has been shown (see among others [123]) that, in case of complex configurations (e.g. marine propellers), the RANS methodology may be unsatisfactory in reproducing the flow fields, adopted as input for the acoustic analysis.

The detailed numerical simulation of the flow around sphere was described in a previous work [49], where three different bluff bodies (a sphere, a cube and a prolate spheroid) were investigated concerning their noise signature. In the mentioned work, validations of the LES solver against experimental and DNS data were presented. In particular, the flow fields resulting from the employed framework reproduce the thin boundary layer developing along the wall-normal direction, the leading edge, and in the wake. Moreover, they were able to reproduce distributions of the friction coefficients, as well as capturing the underlying flow dynamics with a wide coverage of the spectral content.

The filtered Navier-Stokes equations in the incompressible regime are considered. Within this setting, the SGS stress tensor $\tau_{ij}^{\text{sgs}} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j$, which represents the effect of the unresolved fluctuations on the resolved motion, is modeled considering the Smagorinsky eddy-viscosity closure:

$$\tau_{ij}^{\text{sgs}} - \frac{1}{3}\tau_{kk}^{\text{sgs}}\delta_{ij} = -2\nu_t \bar{S}_{ij}, \ \text{ with } \ \bar{S}_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_j}{\partial x_i} + \frac{\partial \bar{u}_i}{\partial x_j}\right), \tag{9.1}$$

and δ_{ij} is the isotropic second order tensor, while the overbar denotes the filtering operator. The SGS eddy viscosity ν_t is expressed as:

$$\nu_t = (C_s \Delta)^2 \, |\bar{S}_{ij}|, \tag{9.2}$$

where $\Delta = \sqrt[3]{V_c}$ is the filter width which is defined as the cubic root of the cell volume V_c, and the Smagorinsky constant C_s is computed dynamically using the Lagrangian procedure of [175], averaging over the fluid-particle Lagrangian trajectories.

9.2.2 *Acoustic model*

The acoustic model herein considered is the one proposed by Ffowcs Williams and Hawkings [86], which is an extension of the Lighthill theory.

The basic idea behind the acoustic analogies is that pressure perturbation originates in the flow field and propagates in the far-field where the medium is assumed quiescent and uniform. The integral solution of the acoustic wave equation presented by Ffowcs Williams and Hawkings consists of surface and volume integrals, meaning that the sources of fluid-dynamic noise can be found as pressure-velocity fluctuations developing in the fluid region or as reflected pressure on an immersed solid surface.

We consider the original formulation presented in [86], and modified according to the works of Najafi et al. [189] and Cianferra et al. [49]. The modification of the original FWH equation takes into account the advection of acoustic waves. To account for the surrounding fluid moving at a constant speed (along the x axis), the advective form of the Green's function must be considered. A derivation of the advective FWH equation is reported in [189], where the authors developed an integral solving formulation for the linear (surface) terms. The advective formulation of the volume term for the particular case of the wind tunnel flow is reported in [49].

In the present work, as done for the fluid dynamic part, we describe the formulation without dwelling into details, for which we refer to the previous works [49, 50].

The acoustic pressure $\hat{p}$, at any point $\mathbf{x}$ and time t, is represented by the sum of surface ($\hat{p}_{2\mathrm{D}}$) and volume ($\hat{p}_{3\mathrm{D}}$) integrals, respectively:

$$4\pi\hat{p}_{2\mathrm{D}}(\mathbf{x},t) = \frac{1}{c_0}\frac{\partial}{\partial t}\int_S \left[\frac{\tilde{p}\hat{n}_i\hat{r}_i}{r^*}\right]_\tau dS + \int_S \left[\frac{\tilde{p}\hat{n}_i\hat{r}^*_i}{r^{*2}}\right]_\tau dS, \tag{9.3}$$

$$\begin{aligned} 4\pi\hat{p}_{3\mathrm{D}}(\mathbf{x},t) = & \frac{1}{c_0^2}\frac{\partial^2}{\partial t^2}\int_{f>0}\left\{T_{ij}\left[\frac{\hat{r}_i\hat{r}_j}{r^*}\right]\right\}_\tau dV \\ & + \frac{1}{c_0}\frac{\partial}{\partial t}\int_{f>0}\left\{T_{ij}\left[\frac{2\hat{r}_i\hat{r}^*_j}{r^{*2}} + \frac{\hat{r}^*_i\hat{r}^*_j - R^*_{ij}}{\beta^2 r^{*2}}\right]\right\}_\tau dV \\ & + \int_{f>0}\left\{T_{ij}\left[\frac{3\hat{r}^*_i\hat{r}^*_j - R^*_{ij}}{r^{*3}}\right]\right\}_\tau dV. \end{aligned} \tag{9.4}$$

The pressure perturbation with respect to the reference value p_0 is denoted with $\tilde{p} = p - p_0$, $\hat{n}$ is the (outward) unit normal vector to the surface element dS, and c_0 is the sound speed. $\hat{r}$ and $\hat{r}^*$ are unit radiation vectors, r and r^* are the module of the radiation vectors $\mathbf{r}$ and $\mathbf{r}^*$ respectively. Their description is given in detail in [49].

Equation (9.4) contains two second–order tensors: R^*_{ij} and the Lighthill stress tensor T_{ij}, the latter characterizing the FWH quadrupole term. Under the assumption of negligible viscous effects and iso-entropic transformations for the fluid in the acoustic field, the Lighthill tensor reads as:

$$T_{ij} = \rho_0 u_i u_j + \left(\tilde{p} - c_0^2\tilde{\rho}\right)\delta_{ij}, \tag{9.5}$$

where $\tilde{\rho}$ is the density perturbation of the flow which, in our case, is equal to zero. The surface integrals in equation (9.3) are referred to as linear terms of the FWH equation and represent the loading noise term. The volume integrals in equation (9.4) are slightly different from the standard FWH (non-advective) equation. For their derivation we consider a uniform flow with velocity U_0 along the streamwise direction.

Obviously, the direct integration of the volume terms gives accurate results, however, this method can be used if the calculation of the time delays can be

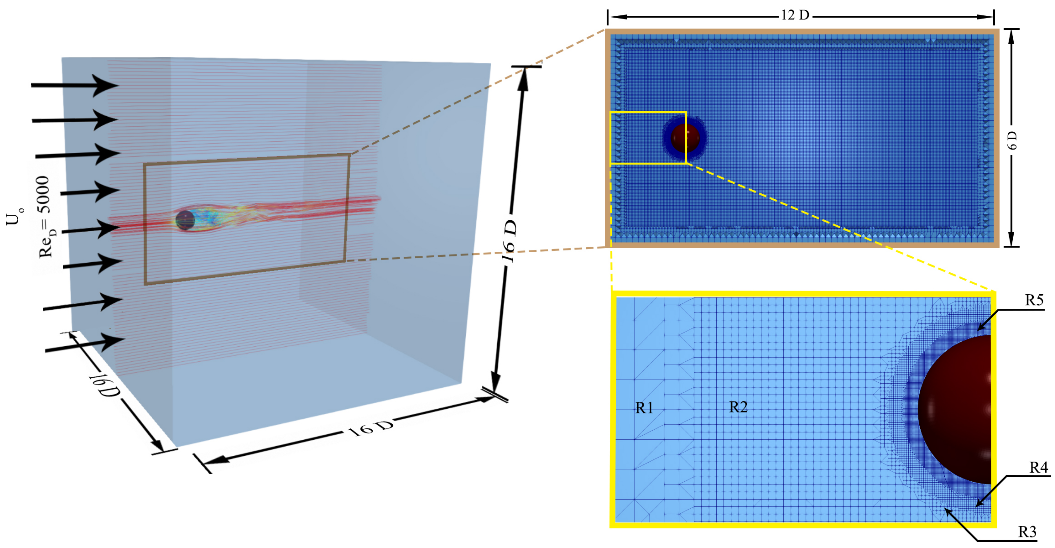


Figure 9.2: Sketch of the computational domain used for the FOM. The sphere diameter is $D = 0.01$ m. Successive mesh refinement layers ($R2$, $R3$, $R4$) are performed through cell splitting approach until reaching the finest grid spacing $0.001D$ in the region $R5$.

omitted, otherwise the computational burden makes it unfeasible. In fact, the calculation of the time delays requires storing at each time step the pressure and velocity data related to the entire (noise-source) volume, in order to perform an interpolation over all the data.

However, for the case herein investigated, the evaluation of the non-dimensional Maximum Frequency Parameter [50] (which is greater than unity for every microphone considered) allows to adopt the assumption of compact noise source. This means that, in the investigated case, the time delay is very small and the composition of the signals is not expected to contribute to the radiated noise. Since the evaluation of the time delays may be reasonably omitted, a remarkable saving of the CPU time is achieved, and the direct computation of the quadrupole volume terms becomes feasible.

In the two previous works [49, 50], as to perform a validation test for the acoustic model, the solution of the advective FWH equation was compared with the pressure signal provided by LES, considered as reference data. This comparison is useful to verify the ability of the acoustic post processing to accurately reconstruct the pressure field. Also, it points out the frequency range which is important to consider.

9.3 NUMERICAL RESULTS

9.3.1 *Full order CFD*

The fluid dynamic fields are solved in the framework of the OpenFOAM library [284] which is based on the Finite Volume Method (FVM). The filtered Navier-Stokes equations are solved using the PISO pressure-velocity coupling algorithm implemented in the pisoFoam solver. The spatial derivatives are discretized through second-order central differences. Implicit time advancement runs according to the Euler scheme. The numerical algorithm, including the SGS closure, has been customized at the laboratory of Industrial and Environmental Fluid Mechanics (IE-Fluids) of the University of Trieste, and more details can be found in [53].

The fluid dynamic full order model simulates a sphere of diameter $D = 0.01$ m, immersed in a water stream with constant streamwise velocity $U_0 = 1$ m/s. The kinematic viscosity is $\nu = 2.0 \times 10^{-6}$ m^2/s, so that the Reynolds number based on the sphere diameter is $\mathrm{Re}_D = 5000$.

The computational domain, depicted in figure 9.2, is a box with dimensions $16D \times 16D \times 16D$ along the x, y and z axes respectively. The sphere is located such that a distance of $12D$ is attained downstream, along the x-axis, while it is centered with respect to the other axes. A zero-gradient condition is set for the pressure at the domain boundaries, except for the outlet where pressure is set to zero. The velocity is set to U_0 at the inlet, stress-free condition is set at the lateral boundaries, and zero-gradient condition is set for the velocity components at the outlet.

The grid, unstructured, and body-fitted, consists of about 5 millions of cells. It is created using the OpenFOAM *snappyHexMesh* utility. The grid spacing normal to the wall for the densest layer of cells (indicated as $R5$) is such to have first cell center within a wall unit y^+ ($y^+ = u_\tau y/\nu$ with $u_\tau = \sqrt{\tau_w/\rho_0}$ and τ_w the mean shear stress). An *A posteriori* analysis showed that about 5 grid points are placed within 10 wall units off the wall. The grid spacing is obtained through successive transition refinements (indicated as $R3$ and $R4$ in figure 9.2). A refinement box around the body (named $R2$ in figure 9.2) is considered so as to obtain, in the wake region, a grid size of less than $0.1D$ at a distance of $8D$. Out of the region of interest, a coarser grid (indicated as $R1$) allows for possible extension of the domain dimensions, and reducing possible disturbance effects coming from the boundaries.

For the time integration, a constant time step is set to $\Delta t = 10^{-5}$ sec in order to keep the Courant number under the threshold of 0.5. The flow around the sphere is completely developed after about 80 characteristic times D/U_0.

9.3.2 *Modal decomposition*

In this section, we report the numerical results concerning the modal decomposition of the full order snapshots. This phase is particularly useful in order to have an insight onto the dominant structures and on the frequencies hidden in the full order dynamical system. We report both an analysis on the eigenvalue decay which is associated with the Kolmogorov width, and the modal representation which permits to visualize the turbulent structures associated with each mode. Moreover, we analyse the time evolution of the temporal coefficients in order to identify the time frequencies associated with each mode. All the computations have been carried out using the PyDMD Python package [71] and the ITHACA-FV library [258, 260].

9.3.2.1 *Singular values decay*

Figure 9.3 depicts the first step of the modal analysis applied to the Navier–Stokes fluid dynamic problem considered. The plot shows the magnitude of the normalized Singular Values (SV) obtained from the SVD factorization of the snapshot matrix, obtained both for the streamwise velocity field component and for the pressure field.

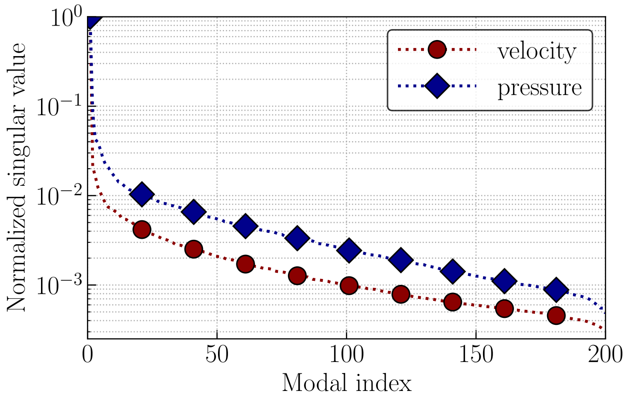


Figure 9.3: Normalized singular values (SV) for streamwise velocity and pressure snapshots. SV are arranged in descending order. The absence of any gaps in the plots suggests no specific truncation rank for the SV-based reduced models.

In the diagram — and in all the following discussion — the modes are arranged in descending order according to the corresponding SV magnitude. Typically, a presence of SV magnitude gaps in such plot provides an indication of a convenient truncation rank for the modal analysis. In the present case, a steady and continuous decay is observed after a steep slope corresponding to the first 15 to 20 modes. Thus, the absence of SV magnitude gaps in the higher frequencies region suggests that, for both fluid dynamic fields considered, there is no specific truncation rank for the modal analysis.

9.3.2.2 *Modal representation*

As shown, the SV magnitude observation is not resulting in an obvious indication of the modal truncation rank. To start understanding the effect of modal truncation rank on the fluid dynamic solution accuracy, we then resort to considerations based on the spatial and time frequencies which need to be reproduced in the hydro-acoustic simulations.

Figure 9.4 depicts a set of three dimensional modal shapes resulting from the longitudinal velocity field DMD and POD modal decomposition, respectively. The purple diagrams represent isosurfaces passing through the field data points of the DMD modes at value -15×10^{-5} m/s, while the olive color plots refer to isosurfaces of POD modes at value 6.5 m/s. For each modal decomposition methodology, the images in the Figure are arranged in tabular fashion and refer to modes 1, 2, 4 on the first row, and 8, 36, 128 on the second one. The plots suggest that the DMD modal shapes present a more pronounced tendency to be organized according to spatial frequencies with respect to the POD modes. In fact wider turbulent structures are only found in the very first DMD modes, while in the case of POD modes they can be identified also in higher rank modes, among higher frequency patterns. Along with this tendency, the turbulent structures associated with the low rank DMD modes also appear organized in longitudinal streaks, and gradually become more isotropic for higher rank modes. It should be pointed out though that despite the fact that POD modes are in general not designed to separate the contributes of single harmonic

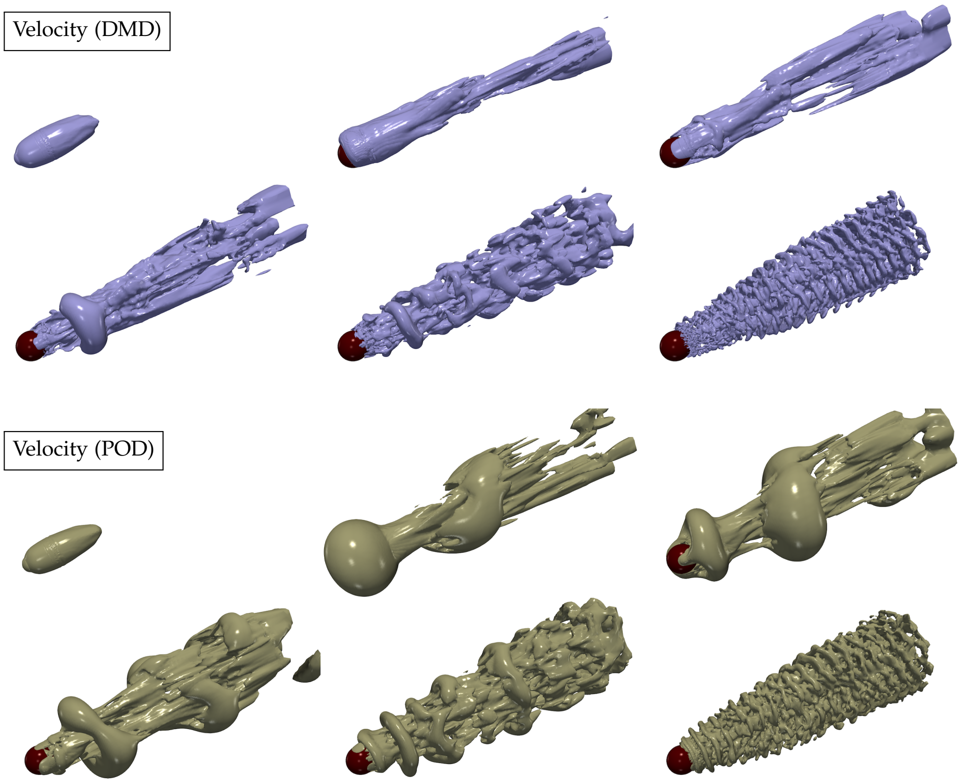


Figure 9.4: Isosurfaces of the DMD modes (purple color) at value of -15×10^{-5} m/s versus the isosurfaces of POD modes (olive color) at value of 6.5 m/s for the streamwise velocity field. Modes: 1, 2, 4, 8, 36, and 128.

components, the modal shapes obtained for the longitudinal velocity do appear to be at least qualitatively correlated to spatial frequencies. In fact, by a qualitative standpoint, the spatial frequencies appearing in the plots corresponding to higher modes are in general higher with respect to those associated to the first modes.

Similar considerations can been drawn from the observation of similar plots corresponding to the modal decomposition of the pressure fields, presented in figure 9.5. Also in this case, the purple plots refer to isosurfaces passing through all data points of value -5×10^{-4} $\mathrm{m}^2/\mathrm{s}^2$ of DMD modal shape functions, while the olive diagrams refer to isosurfaces of POD modes at the value 300 $\mathrm{m}^2/\mathrm{s}^2$. The plots are again arranged, for each decomposition methodology considered, in tabular fashion portraying modes 1, 2, 4 on the first row, and 8, 36, 128 on the second one. The pressure modes associated with both methods seem again qualitatively arranged according to spatial frequency content. As previously observed for the longitudinal velocity modes, the DMD modes appear more closely correlated to spatial frequencies, and present a less isotropic appearance with respect to their POD counterparts.

9.3.2.3 *Associated coefficients*

After having characterized the spatial frequency content of different POD and DMD modes, we now want to analyse the time frequencies associated to each mode. To do

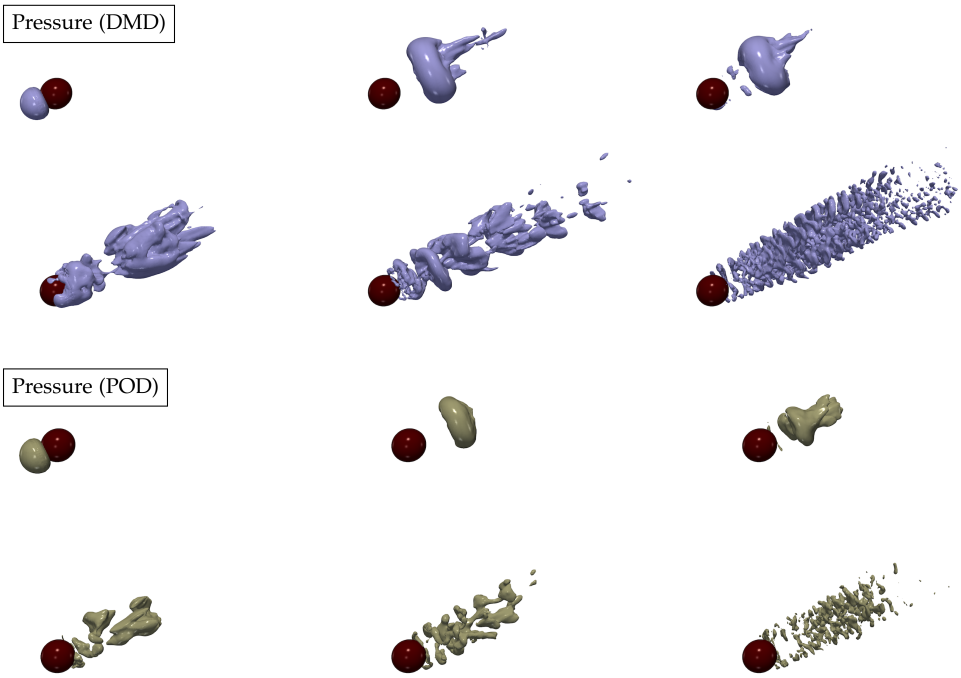


Figure 9.5: Isosurfaces of the DMD modes (purple color) at value of -5×10^{-4} Pa versus the isosurfaces of POD modes (olive color) at value of 300 Pa for the pressure field. Modes: 1, 2, 4, 8, 36, and 128.

this, we decompose each snapshot into its modal components and observe the time evolution of the modal coefficients. In fact, as the POD and DMD modes generated from the snapshots are constant in time, the corresponding modal coefficients must depend on time to allow for the reconstructed solution to reproduce the correct time variation.

The four plots presented in figure 9.6 show, the temporal evolution of the modal coefficients associated, respectively, to the DMD of the longitudinal velocity field (top left), to the POD of the longitudinal velocity field (top right), to the DMD of the pressure field (bottom left) and to the POD of the pressure field (bottom right). Each diagram reports four lines referring to the coefficients of modes 2, 4, 8, 36.

By a qualitative perspective, the diagrams in figure 9.6 suggest that DMD and POD modal coefficients are strongly correlated with the time frequencies. In fact, higher frequency harmonics appear in the time evolution of higher order modal coefficients, which are not observed in lower ones. As expected, also in this case the frequency-mode association is definitely stronger for DMD modes, in which a single dominant harmonic can be identified in correspondence with each modal coefficient. As for POD, the respective frequency content seems to cover wider set of harmonics, associated with higher frequencies as higher modes are utilized.

9.3.3 *Flow fields reconstruction*

A first aim of this work is to assess whether the proposed DMD and POD algorithms, are able to accurately reproduce the full order model solutions. A first step in such

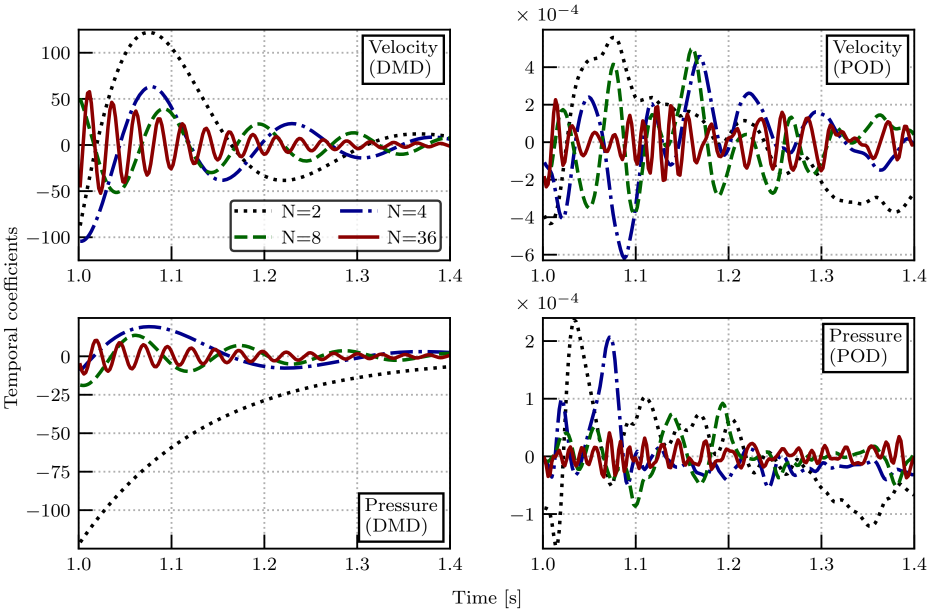


Figure 9.6: Temporal coefficients associated to the DMD and POD (left to right) modes number 2, 4, 8, and 36 of the streamwise velocity and pressure data (top to bottom).

assessment will be that of checking the effectiveness of the SVD based modal decomposition strategies of the model reduction algorithm considered. In particular, the reconstructed solution convergence to the snapshots considered will be discussed both through the visualization of single snapshots flow fields and by presenting convergence plots of error averaged among snapshots. Finally, we will present similar plots for the solution predicted by means of both DMD and PODI.

Figure 9.7 presents a first evaluation of the effectiveness of the DMD and POD modal decomposition algorithm in reducing the number of degrees of freedom of the fluid dynamic problem. The results refer to a reconstruction exercise in which the LES solutions at all the time steps have been used for the modal decomposition. The curves in the plots indicate the relative reconstruction error at each time step for both velocity (top plots) and pressure (bottom plots) when a growing number of modes are considered. Such relative reconstruction error is computed as the Frobenius norm of the difference between the LES solution vector and the reconstructed one, divided by the Frobenius norm of the LES solution. We here remark that to make the velocity and pressure fields error values comparable, the gauge atmospheric pressure value in the simulations has been set to one. A null value would in fact result in lower LES solution norm, leading in turn to large pressure relative errors compared to the velocity ones, even in presence of comparable absolute errors. In figure 9.7, the two plots on the left refer to DMD reconstruction results, while the ones on the right present the POD reconstruction error. As can be appreciated, for both modal decomposition methods the errors presented follow the expected behavior, and reduce as a growing number of modes is used in the reconstruction, until machine precision error is obtained when all the 200 modes available are used to reconstruct the 200 snapshots. More interestingly, the data indicate that both for DMD and POD, a number of modes between 80 and 120

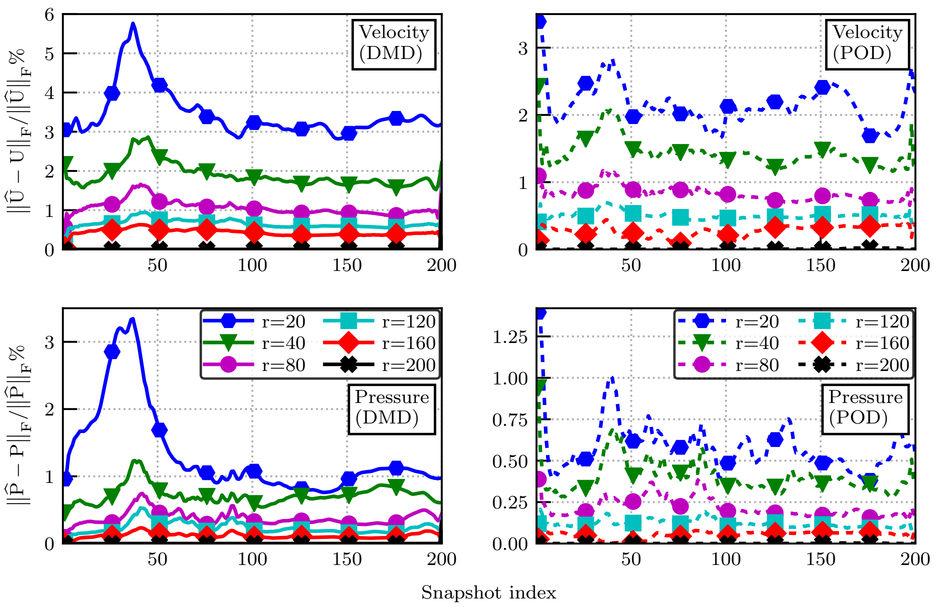


Figure 9.7: Relative error percentage in the Frobenius norm for the velocity and pressure (top and bottom, resp.) reconstructed fields using the DMD and POD (left and right, resp.) modes. As expected, reconstruction accuracy converges with higher SVD truncation rank (r).

leads to velocity and pressure reconstruction errors which fall under 1% across all the time interval considered. It is worth pointing out that, compared to most typical low Reynolds and RANS flows applications of DMD and POD methodologies, such convergence rate is rather slow, as higher number of modes are needed to obtain comparable accuracy. This should not surprise, as LES resolves more turbulent structures than the aforementioned models, resulting in higher spatial frequencies which in turn require a higher number of modal shapes to be accurately reproduced. The peak we see for smaller rank truncations around the 40-th snapshot for the DMD reconstruction is justified by the fact that those snapshots present a wider range of frequencies thus we need more DMD modes to properly reconstruct them. Finally, the plots suggest that the reconstruction with POD modes leads to errors that are slightly lower to the corresponding DMD errors. In fact, the results consistently show that for both the pressure and the velocity fields, the POD reconstruction error is approximately half of the DMD error obtained with the same amount of modes.

9.3.3.1 *Global error*

Figure 9.8 presents further verification of the modal reconstruction accuracy. The curves in the plot represent the percentage modal reconstruction error — computed in Frobenius norm and averaged among all snapshots — as a function of the number of modes considered in the reconstruction. The black and red curves confirm that the reconstructed velocity and pressure fields, respectively, converge to the corresponding LES fields as the number of modes is gradually increased. Moreover, these results indicate that an efficient reconstruction, characterized for instance by a 1% relative error, would require more than 80 modes for the velocity

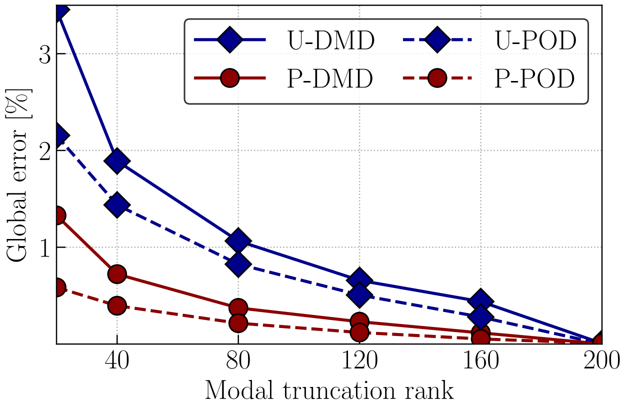


Figure 9.8: Global error of the reconstructed velocity and pressure fields versus SVD truncation rank. The error denotes spatio-temporal averaging of the flow field data, first by evaluating relative error in the Frobenius norm, then by averaging over all the snapshots.

fields, and more than 30 for the pressure fields. These values are higher than those typically observed for RANS and low Reynolds simulations, probably due to the higher spatial frequencies typically found in eddy-resolving solution fields.

9.3.3.2 *Fields visualization*

The error indicators considered in the previous sections are extremely useful in confirming that the reconstructed solution is globally converging to the LES one as the number of modes is increased. Yet, they offer little information on the error distribution in the flow field, and the impact of the reconstruction on local flow characteristics of possible interest. In particular, for the test case considered in the present work, it is quite important to assess whether the reconstruction error does not alter the flow in proximity and in the wake of the sphere, as such regions are crucial both to the evaluation of the fluid dynamic forces on the sphere and to the acoustic analysis. To this end, in the present section we present a series of visualization of the reconstructed flow fields, which are compared to their LES counterparts.

Making use of the Q-criterion, defined as $Q = 0.5(\|\mathbf{\Omega}\|^2 - \|\mathbf{S}\|^2)$ with $\mathbf{\Omega}$ and $\mathbf{S}$ denoting the vorticity and strain rate tensors respectively, figure 9.9 depicts the turbulent structures characterizing the flow in the wake region past the sphere. By definition, a positive value of Q implies relative dominance of the vorticity magnitude over the strain rate [107]. Here, a positive value of 10^4 $1/\mathrm{s}^2$ is chosen to generate isosurfaces which pass through all data points holding this value. In the figure, the top plot refers to the original LES solution obtained at the last snapshot of the dataset, while the centered and bottom plots refer to the corresponding DMD and POD reconstructions, respectively, utilizing 160 modes. The images show that both POD and DMD reconstruction algorithms lead to fairly accurate representation of the turbulent structures shape past the sphere. In fact, the configuration of the wider vortical structures detaching from the sphere appears to be correctly reproduced in the reconstructed solution. As for finer details associated with the smaller turbulent scales, the DMD reconstruction is observed to be in closer

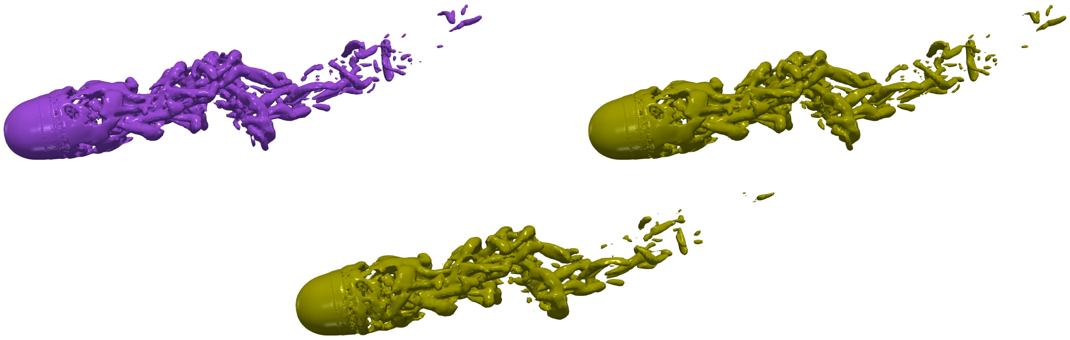


Figure 9.9: Coherent structures represented by the iso-contours of the Q-criterion at non-dimensional value $QD^2/U_0^2 = 4$ for the last snapshot (top), compared with corresponding ones obtained from modal reconstruction with DMD (centered) and POD (bottom) using 160 modes.

agreement with the original LES solution than the POD reconstructed field. Such observation is consistent with findings from [24, 295] in which they demonstrated the superiority of DMD to accurately determine spectral and convective information of the vortical structures in wake regions.

For a more significant and quantitative assessment, figure 9.10 includes a series of of contour plots representing the instantaneous flow field at time instance corresponding to a maximized relative error, cf. figure 9.7. The three plots in the first row represent contours of the LES streamwise velocity component field, and of its DMD and POD reconstructed counterparts, respectively, utilizing 160 modes. At a first glance, the reconstructed fields seem to reproduce the main features of the LES flow. In particular, both the stagnation region ahead of the sphere and the flow detachment past it appear to be correctly reproduced by both modal reconstruction strategies. In addition, the detached vortex, located downstream with respect to the sphere in this particular time instant, is also correctly reproduced. For a better assessment, the two images of the second row represent contours of the local error for the DMD and POD streamwise velocity reconstruction, respectively. Both for the DMD and POD reconstruction, higher local error values are found in the wake region downstream with respect to the sphere. In particular, it observed that the local error peaks in the DMD reconstruction is larger than that for POD. Additionally, the high frequency error pattern and the elevated local error values located in the wake region seem to indicate that, as expected, the modes disregarded in the reconstruction are associated with high spatial frequencies. A similar comparison is presented for the pressure field in the following rows of the figure. The three plots in the third row represent the instantaneous pressure field obtained with LES, and its reconstructions computed with DMD and POD, respectively, for the same snapshot. Again, the full order field appears well reproduced by both DMD and POD reconstruction algorithms, as features like the peak pressure in the stagnation region and the pressure minimum within the vortex detaching past the sphere are correctly reproduced. The plots of pressure reconstruction error for DMD and POD, respectively, presented in the last row confirm that both methods are capable to adequately represent the LES solution.

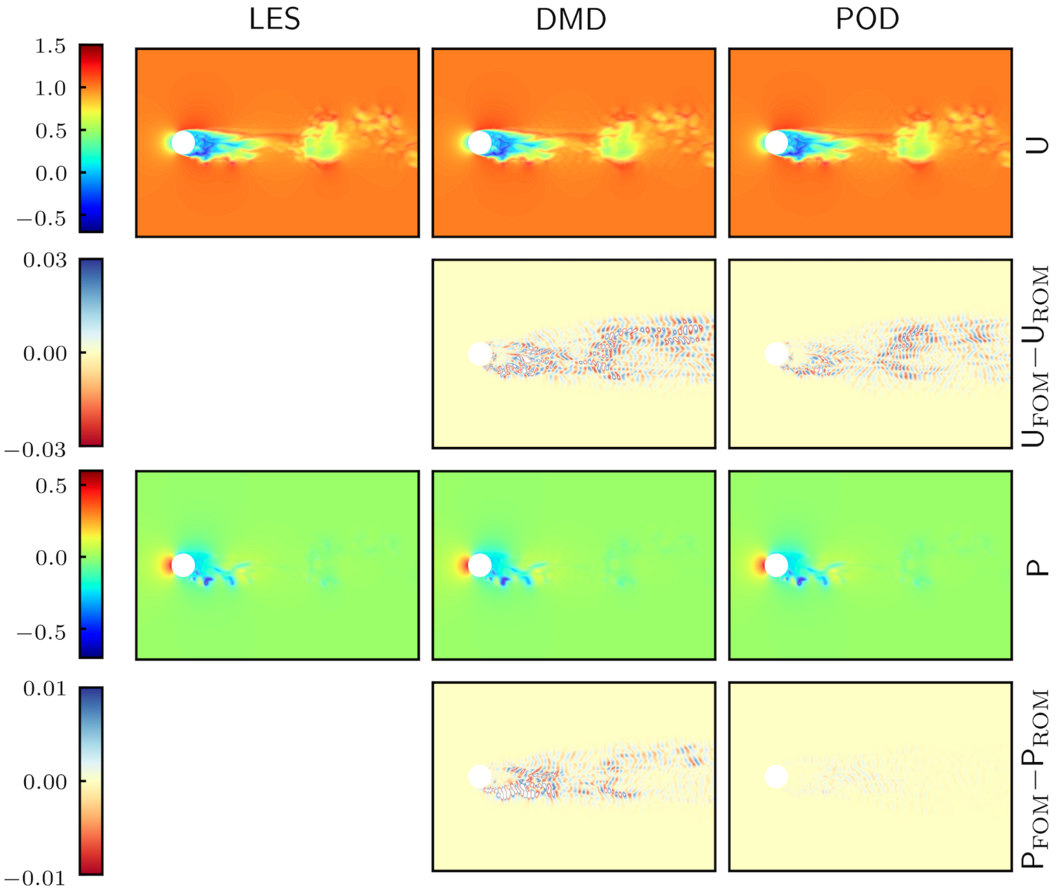


Figure 9.10: Left to right: LES, DMD and POD reconstructions with $N = 160$ modes. Top to bottom: streamwise velocity, corresponding error fields, pressure, corresponding error fields. Instantaneous snapshot is selected based on errors peak.

Also here, the disregarded high spatial frequency modes are likely responsible for the high frequency error pattern observed.

9.3.3.3 *Error statistics in the wake region*

As figure 9.10 shows, the highest reconstruction errors in the velocity and pressure fields reconstruction are mostly located in the wake region. Therefore, a follow up analysis is required to quantify the spatial error distribution within such region and to assess the local convergence behavior of both the DMD and POD modal decomposition methodologies. Figure 9.11 presents a normalized density function plot based on the reconstruction error of both the velocity and pressure fields corresponding to the snapshot illustrated in figure 9.10. To generate the plot, the following steps are performed. First, the computational cells in the wake region are identified according to the condition $\|\boldsymbol{\omega}_x\| = \|\nabla_x \times \mathbf{U}\| > 1.0$ with $\boldsymbol{\omega}_x$ denoting the streamwise vorticity component. Second, the arrays corresponding to the velocity and pressure error fields in the wake region are computed, i.e. $e_w^U = (\hat{U}_w - U_w)$ and $e_w^P = (\hat{P}_w - P_w)$ wherein the subscript denotes the wake cells. Third, to account for variations in the grid resolution, the mentioned error fields within the wake region are weighted by the corresponding normalized cell volumes V_c^{norm}, i.e.

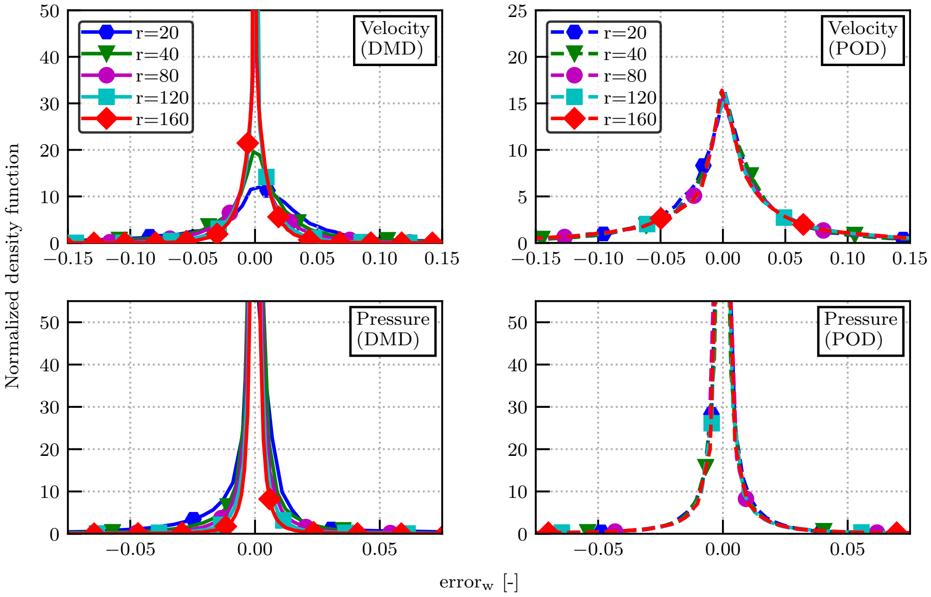


Figure 9.11: Error statistics of velocity and pressure field data of selected snapshot corresponding to maximized relative error. Sampled data points are conditioned by $\|\boldsymbol{\omega}_x\| = \|\nabla_x \times \mathbf{U}\| > 1.0$ to identify the wake region. Resulting density function is weighted by normalized cell volumes. Presented plots correspond to various modal truncation ranks (r).

$\bar{e}_w = (\sum_{i=1}^{n} V_{c,i}^{\text{norm}} e_{w,i} / \sum_{i=1}^{n} V_{c,i}^{\text{norm}})$ wherein n is the cell count in the wake. Finally, the error interval for both the velocity and pressure data are uniformly divided into equal-width bins and the density function (normalized histogram) is plotted for each bin.

The scattered plots highlight differences in the behavior of the two modal reconstruction strategies considered. The diagrams on the left, which refer to the DMD results for velocity (top) and pressure (bottom), show in fact a clear error reduction as the modal truncation order is increased. The curves corresponding to growing truncation orders tend to get closer to the vertical axis as the error, displayed on the horizontal axis, is progressively reduced. The same convergence rate cannot be observed in the POD plots on the right, as both the velocity (top) and pressure (bottom) reconstruction error statistical distribution curves appear less affected by an increase of the modal truncation order. Again, this observation can be explained in the light of the DMD theory and the ability of its modes to be organized according to the field spatial frequencies, unlike POD ones which can instead become contaminated by uncorrelated structures, according to the claim reported in [295].

It is worth pointing out that for image definition purposes, the left and right tails of figure 9.11 have not been reported. Yet, a cross comparison with figure 9.10 readily suggests that POD results showed lower error margin — hence narrower tails — in this regard. Therefore, we could infer that higher DMD modes are capable to capture more frequencies in the wake, resulting in lower mean error but higher peak errors compared to POD modes.

Table 9.1: ROM performance at various modal truncation rank. Data compression level is defined as $(\mathrm{GB_{FOM}}/\mathrm{GB_{ROM}})$ with $\mathrm{GB_{FOM}}$=74. Speedup is defined as $(\mathrm{CPU_{FOM}}/\mathrm{CPU_{ROM}})$ with $\mathrm{CPU_{FOM}} = 2.4 \times 10^5$sec. Compression level is averaged between DMD and PODI.

Rank	POD cumulative energy		Speedup		Compress.
(r)	U	P	DMD	PODI	level
10	0.999992	0.994093	29.29	35.67	19.995
20	0.999996	0.997076	28.46	32.46	9.9973
40	0.999998	0.998735	26.41	27.93	4.9986
60	0.999999	0.999295	25.41	24.68	3.3324
80	0.999999	0.999571	23.58	22.48	2.4993
100	1.000000	1.000000	23.18	20.46	1.9995

9.3.4 *Fields mid cast using DMD and PODI*

The previous sections are focused on assessing the accuracy of the POD and DMD modal decomposition strategies. The reconstruction results confirmed that both methods can be considered effective tools for the reduction of the degrees of freedom of the fluid dynamic problem. We now want to analyse the ability of the data-driven DMD and PODI reduced order models considered in this work, in predicting the LES solution at time steps that are not included in the original snapshot set. In particular, throughout the remaining analysis of this work, the original 200 LES snapshots are decomposed into two sets, one set comprises the 100 odd snapshots from which they are used to train the ROMs and are called the *train* dataset hereafter, while the remaining even snapshots within the temporal interval of the train dataset are contained in a *test* dataset, and are used to compare the full order solution with the DMD and PODI model prediction results (i.e. ROM *prediction* dataset).

Before progressing with analysis, a summary on the POD energetic content, as well as the data compression level and computational speedup for the considered DMD and PODI data-driven models at various modal truncation ranks is listed in table 9.1. Here, the kinetic energy content within a ROM is described by cumulative sum of the POD eigenvalues. The compression level, taken as the arithmetic mean between DMD and PODI in the table, considers the size ratio of the FOM data (train and test datasets, totalling 74 GB) to the ROM data (spatial modes of the train dataset, and continuous representation of the temporal dynamics computed via time integration or cubic spline interpolation for DMD or PODI, respectively). Speedup is defined as $(\mathrm{CPU_{FOM}}/\mathrm{CPU_{ROM}})$ with $\mathrm{CPU_{FOM}} = 2.4 \times 10^5$sec corresponding to the time required to solve for the 200 snapshots and to write out the train dataset, while $\mathrm{CPU_{ROM}}$ is the time needed to 1) perform modal decomposition on the train dataset, 2) extract the associated dynamics with a continuous representation, and 3) write out the prediction dataset in OpenFOAM format.

It is worth noting that, since time is the considered parameter in the present ROM procedure, an offline phase still requires solving for the same temporal window as in the FOM solution, in order to obtain the train dataset. Nevertheless, a computational gain in the generated ROM can be still attained, since the less amount of stored data (i.e. few modes and associated dynamics) compared with FOM are utilized for a swift construction of the fluid dynamic fields at arbitrary sample points. Such procedure is considered a lot faster and more economic than recomputing the simulation to write out the fluid dynamic field data at those sample points. Indeed, an extension of the present procedure to consider additional parameters while performing, for instance, multiparameter interpolation using PODI, would result in a further speedup since the offline phase would solve a full order solution only for a subset of the parameter combinations in the parameter space, hence saving up complete FOM computations from being performed. Such multiparameter investigation is considered a follow up study of this work. To this end, now we progress the analysis by considering the temporal evolution of the error due to ROMs prediction.

9.3.4.1 *Prediction error analysis*

The first test presented is designed to assess the predictive accuracy of the DMD and PODI methodologies described in this work. figure 9.12 depicts the results obtained with the ROM methodologies applied in such alternate snapshots arrangements as previously described. The top plots report the percentage Frobenius norm error between the DMD and PODI predictions with respect to the LES solution for the longitudinal component of the velocity, while the bottom diagrams present similar error norms corresponding to the pressure field. The left plots refer to the DMD results, while the right ones depict the PODI errors. We point out that, the — significantly lower — error values corresponding to the train dataset have been omitted for clarity. Finally, the different colors in the plots indicate growing number of modes considered in the DMD and PODI prediction, up to a maximum of 100.

The results show a substantial convergence of the data-driven reduced model solutions to the full order one. In particular, selecting a number of modes between 80 and 90 results in errors lower than 2% on the velocity field for most time steps in both DMD and PODI methods. As for the pressure field, the bottom left diagram in figure 9.12 shows that 1% error goal can be obtained with an even slightly lower amount of DMD modes. It must be pointed out though, that the DMD solution in the very first time steps is not converging to the LES one, and the error grows as the number of modes is increased. This behavior, which could be related to the high frequencies introduced by the higher DMD modes added to the solution, is currently under further investigation. Aside from the first few time steps, it is generally observed that PODI errors are consistently lower by a factor two with respect to the DMD ones, especially in the cases utilizing fewer modes. It is worth pointing out that a direct comparison between the PODI plots in figure 9.8 and those in figure 9.12 can indirectly result in a possible estimate of the interpolation error associated with the PODI strategy. The plots suggest that the effect of interpolation on the global spatial error at each time step is rather low, as the errors in figure 9.12 present the same behavior and are not significantly higher than the ones obtained with pure reconstruction.

Now, to summarize the performance of each ROM with respect to the modal truncation level, a global spatio-temporal error is measured against growing number of modes, as depicted in figure 9.13. In particular, similar to the previous analysis

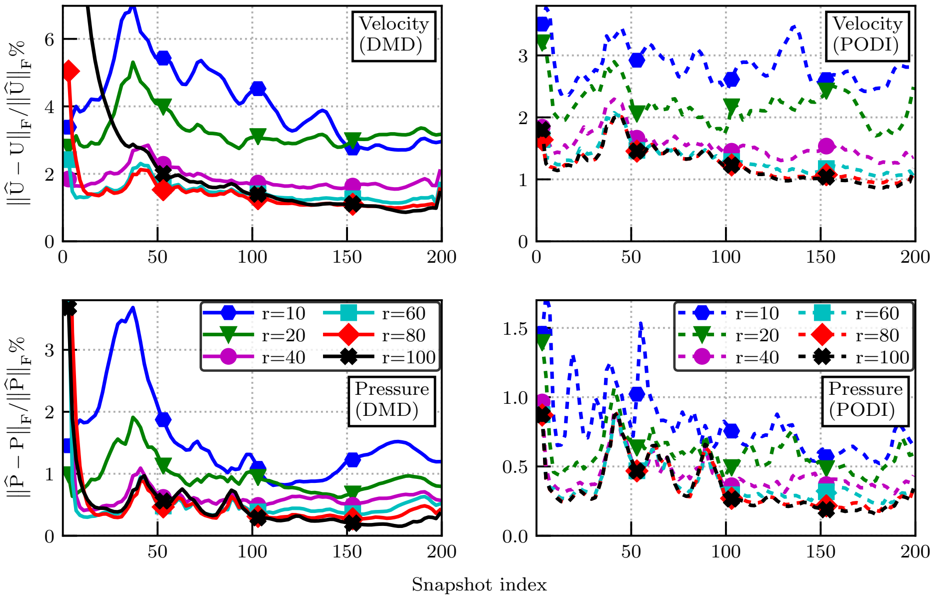


Figure 9.12: Relative error percentage in the Frobenius norm for velocity and pressure (top and bottom, resp.) fields, where half sample rate snapshots are used to train the reduced model, DMD (left) or PODI (right), for particular truncation (r) while predicting the intermediate snapshots. Plots show only the snapshot-wise prediction error, while disregarding errors in the training set, which is almost null.

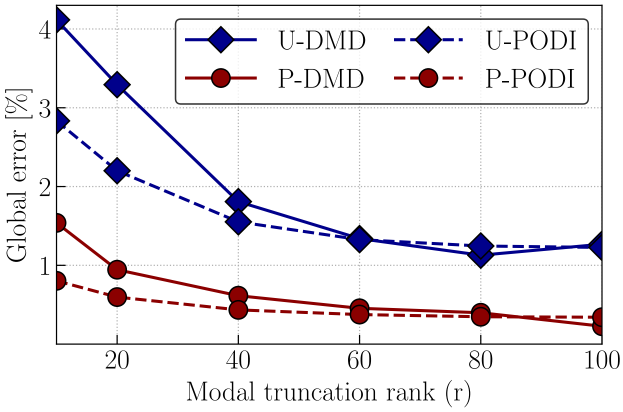


Figure 9.13: Global error of the prediction dataset versus SVD truncation rank. The error denotes spatio-temporal averaging of the flow field data, first by evaluating relative error in the Frobenius norm, then by averaging over all the snapshots.

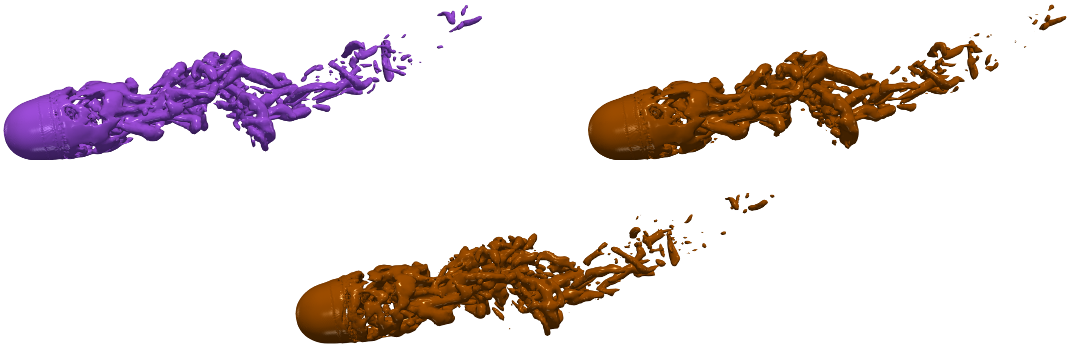


Figure 9.14: Coherent structures of the last snapshot in the prediction dataset, represented by the iso-contours of the Q-criterion at non-dimensional value of $QD^2/U_0^2 = 4$, compared with corresponding ones obtained from DMD-100 (centered), and PODI-100 (bottom).

in figure 9.8, the mean absolute error is evaluated for the percentage Frobenius norm spatial error for all the prediction dataset, cf. figure 9.12. Here, it is observed that, for both the velocity and pressure predictions, a significant reduction in the global error is achieved in the PODI models with respect to DMD up to a utilization of 60 modes, after which the global error becomes comparable between both ROMs. Additionally, a finite global error is noted in the figure even when a full modal rank is utilized. Indeed, such observation should not be surprising since half the dataset is only employed to train both ROMs while interrogating the remaining sample points. This is not the case in pure field reconstructions, cf. figure 9.8, where the global error vanishes with a full modal rank since dynamic and spectral information become entirely recovered.

9.3.4.2 *Coherent structures*

The prediction error indicators considered in the previous section indicate whether the reduced models solution is globally converging to the LES one as the number of modes is increased. We now resort to flow visualizations to obtain better information on the error distribution in the flow field and on the reduced models performance in reproducing local flow characteristics of possible interest.

Again, making use of the Q-criterion isosurfaces at value $QD^2/U_0^2 = 4$, figure 9.14 portrays the turbulent structures characterizing the wake flow past the sphere. The top plot refers to the original LES solution obtained at the last snapshot of the prediction dataset, while the centered and bottom plots refer to the respective DMD and PODI predicted solution utilizing 100 modes. Also in this case, the images show that both DMD and PODI reduced order models allow for rather accurate reproduction of the turbulent structures past the sphere. For both ROM solutions, the main vortical structures detaching from the sphere appear in fact very similar to those of the original LES flow field. Finer details associated with smaller turbulent scales are also in good agreement, hence suggesting that errors in the PODI time interpolation and DMD time integration are not significantly higher with respect to the reconstruction error analysed earlier.

9.3.4.3 *Drag and lift coefficients*

Besides coherent structures, it is important to assess the performance of data-driven ROMs in capturing hydrodynamic phenomena that could find particular interest in the engineering community. One of these phenomena is the drag and lift forces on the sphere surface. First, pressure-induced drag and lift forces are evaluated by integrating the pressure in the streamwise and the flow-normal directions, respectively, over the sphere surface. Then, the pressure forces are normalized by the dynamic pressure acting on the sphere, with a sphere projected cross-sectional area being considered for the reference area, to compute coefficients of drag (C_D) and lift (C_L). The temporal evolution of both coefficients on the sphere surface is depicted in figure 9.15 for various ROM predictions compared with the FOM. The presented plots show that both C_D and C_L are accurately predicted while only half the dataset is employed. It is noted that the accuracy of C_D predictions is higher in comparison to C_L predictions, and that employing as low as 40 modes are able to predict the forces coefficients with a sufficient accuracy. More importantly, the predictive character of PODI models are observed to be superior over DMD models for a particular modal truncation level. The observed discrepancies in the early temporal window of C_L predictions via DMD at $r = 80$ are related to the poor DMD predictions of the flow fields at early snapshots, as previously noted in the relative error plots, cf. figure 9.12. On a general note, the presented data-driven ROMs are able to predict the pressure dynamics on the surface of the immersed sphere while employing considerably lower levels for modal truncation. We would like to highlight that the framework that we are proposing is solely based on input and output quantities. Therefore, instead of computing engineering quantities from the reconstructed flow fields, one could also construct a reduced order model to directly approximate them. In such a case we would expect that the so generated ROM would produce even more accurate results.

9.3.5 *Spectral analysis*

9.3.5.1 *Data probes inside and outside the wake region*

A first necessary step in order to evaluate how the frequency content of the FOM solution is reproduced at the ROM level, consists in observing the time dependency of the solution at fixed points in the computational domain. To this end, figure 9.16 presents the reduced models results for the time evolution of the streamwise velocity component in correspondence with two different locations of the flow field. In the 2×2 tabular arrangement of the figure, the top plots correspond to a point located outside of the sphere turbulent wake, while the bottom plots refer to a point inside the turbulent wake. In addition, the two diagrams on the left refer to the DMD results, and the ones on the right report the PODI result. In the plots, the red filled circles refer to the LES result at prediction sample points, while the dash-dotted lines represent the ROM results, which have been drawn for a growing number of modes. Finally, the black continuous line represents the pure DMD and PODI reconstruction result utilizing 160 modes.

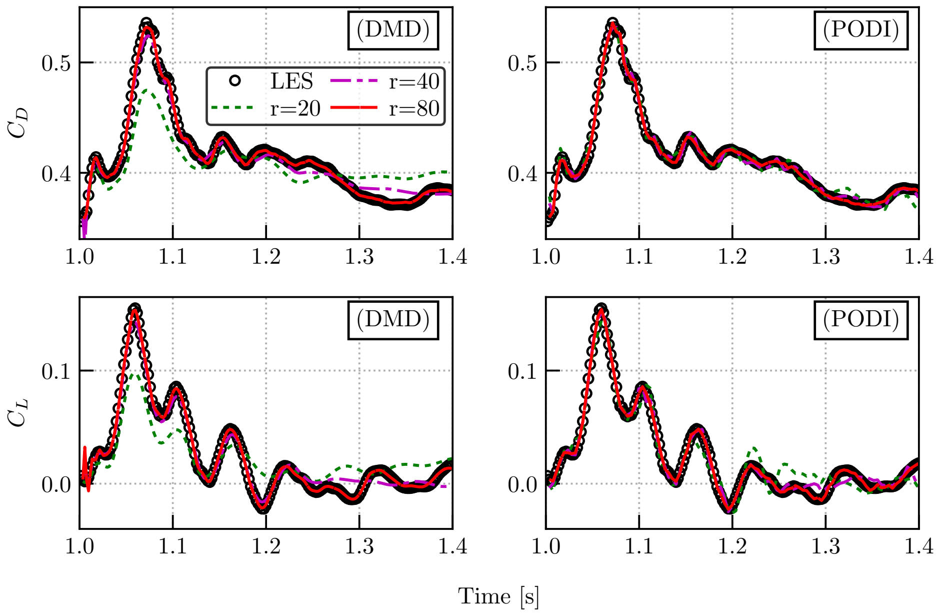


Figure 9.15: Temporal evolution of the drag and lift coefficients corresponding to pressure forces on the immersed sphere surface. The force coefficients are computed for various DMD and PODI models in comparison to FOM data. Presented plots show the predictive performance of both data-driven ROMs on estimating the sphere pressure forces, with noted superiority of PODI predictions over DMD.

As expected, the plots show that the velocity variation over time becomes more chaotic in the wake region, due to the high frequency fluctuations associated with the turbulent structures typically observed in such part of the flow domain. All the plots show that, as the number of DMD and PODI modes used is increased, the reduced solution approaches the full order model one. At a first glance, the PODI convergence at the point located outside of the wake appears faster than that obtained with DMD. The reduced PODI solution obtained with 20 modes is, in fact, already rather close to the original time signal, while the corresponding DMD solution is still far from the LES one. The behavior of both DMD and PODI reduced models is clearly more accurate when a higher number of modes is selected. The 80 modes curve obtained with both ROMs is practically indistinguishable from the LES one. Given the small estimate of the time interpolation error, as discussed in section 9.3.4.1, the faster PODI convergence observed should depend on the fact that single PODI modes are richer in spatial frequency content than the DMD ones. The low order PODI modes might then already include higher frequencies not contained in the corresponding DMD modes. Such spatial frequency content should finally reverberate in the time evolution of the solution including some higher frequencies also when the only lower order PODI modes are used. Despite these favorable characteristics, when the full order model solution presents even higher frequencies, also the PODI solution requires a higher number of modes to obtain satisfactory accuracy. This is clear by the plots of the time history of the velocity at the point within the wake region. Here, both PODI and DMD reduced result curves become sufficiently close to the original solution one only when 80 modes are considered.

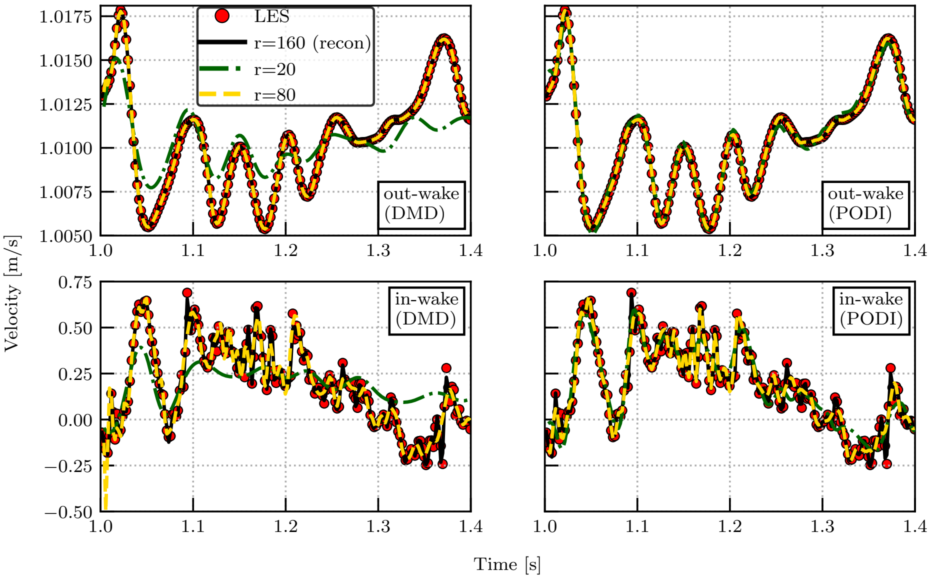


Figure 9.16: Time history of streamwise velocity signals at fixed points located outside the wake at $\mathbf{x}_A = \{2D, 0, 2D\}$ (top) and inside the wake at $\mathbf{x}_B = \{2D, 0, 0\}$ (bottom) for DMD (left) and PODI (right) models. Red symbols represent LES results, while dashed lines represent the ROM results for different SVD truncation ranks (r). Black continuous line represents the reconstructed field utilizing 160 modes.

This should not come as a surprise, as the in-wake signal presents higher frequency fluctuations due to the wake turbulent structures, which can be fully represented only by including high order modes.

The reduced pressure local time evolution prediction results presented in figure 9.17 exhibit the same behavior observed for the reduced velocity field. Also this Figure, in which the plots and the curve colors are arranged as described in figure 9.16, suggests in fact that both inside and outside the wake region, employing 80 modes allows for both ROMs to obtain pressure predictions that are sufficiently close to the corresponding LES time signals. Again, the plots also suggest that PODI in the region outside the wake is able to obtain viable pressure predictions with fewer modes. Thus, these results indicate that in presence of fully attached flows, in which wake effects are less dominant, the choice of POD as modal decomposition methodology could result in more economic reduced models.

9.3.5.2 *Fast Fourier transform*

The plots in figure 9.16 and figure 9.17 seem to indicate that, on a qualitative level, the PODI and DMD solutions are able to recover a growing portion of the full order model frequency content as the number of modes is gradually increased. This aspect should be, of course, investigated in a more accurate way, as the presence of high frequency components in the ROM solution results in their ability to be effective surrogates in acoustic analysis. Thus, we make use of the Fast Fourier Transform (FFT) spectra with the aim of obtaining a quantitative assessment of the impact of the POD and DMD modes considered on the frequency content of the

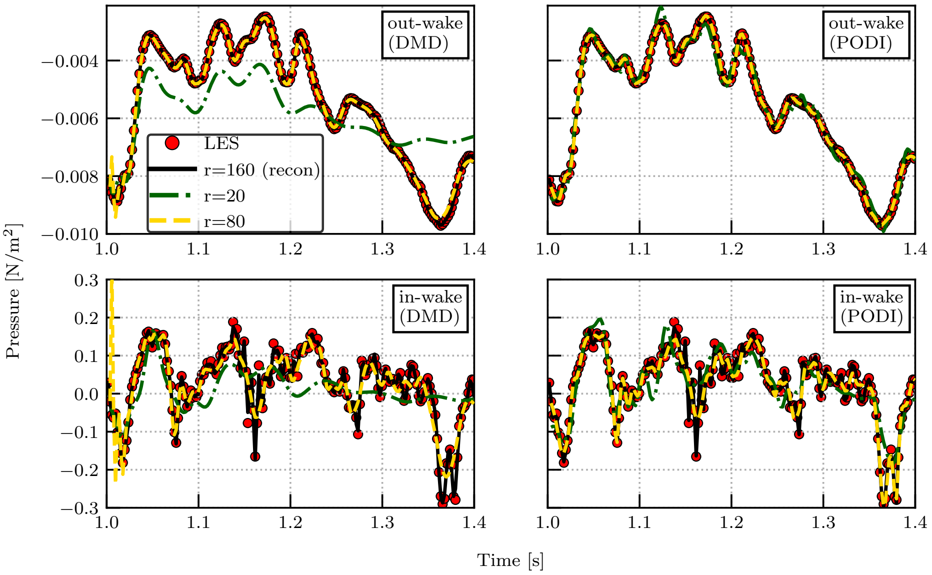


Figure 9.17: Time history of pressure signals at fixed points located outside the wake at $\mathbf{x}_A = \{2D, 0, 2D\}$ (top) and inside the wake at $\mathbf{x}_B = \{2D, 0, 0\}$ (bottom) for DMD (left) and PODI (right) models. Red symbols represent LES results, while dashed lines represent the ROM results for different SVD truncation ranks (r). Black continuous line represents the reconstructed field utilizing 160 modes.

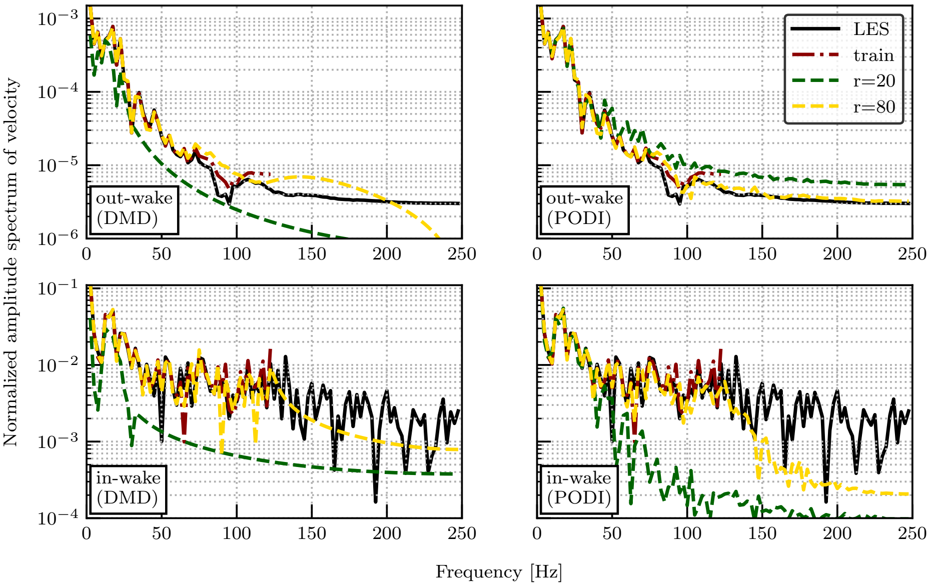


Figure 9.18: Normalized amplitude spectrum of the streamwise velocity component, computed as $(|\text{fft}(U - \overline{U})|/N_{\text{freq}})$ with $\overline{U}$ denoting mean velocity, in a fixed point located outside the wake (top) and inside the wake (bottom) for DMD (left) and PODI (right) at different truncation ranks (r). Black continuous and red dash-dotted lines refer to LES results of the full dataset and train dataset, respectively. Green and yellow dashed lines mark ROM surrogates employing 20 and 80 modes, respectively.

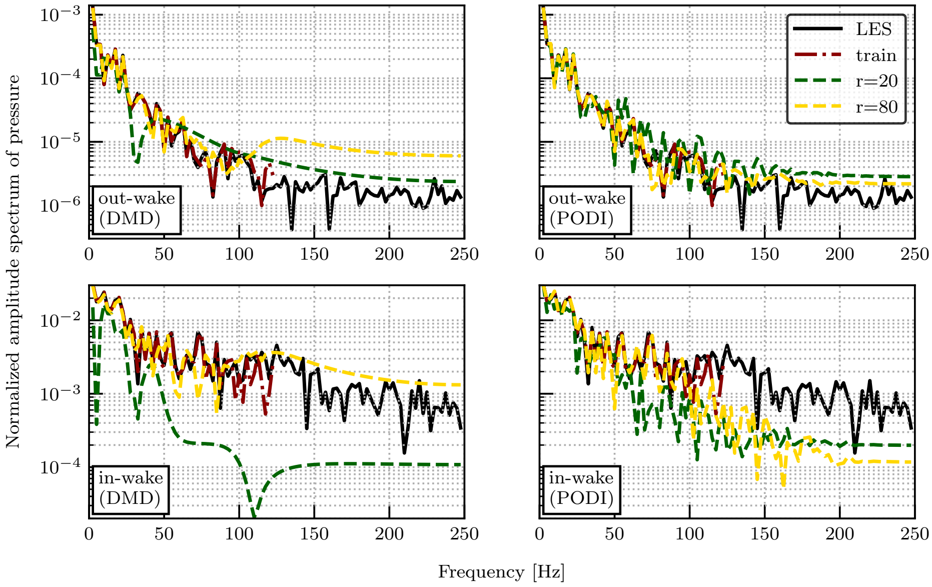


Figure 9.19: Normalized amplitude spectrum of the pressure signal, computed as $(|\mathrm{fft}(p - \overline{p})|/N_{\mathrm{freq}})$ with $\overline{p}$ denoting the mean pressure, in a fixed point located outside the wake (top) and inside the wake (bottom) for DMD (left) and PODI (right) at different truncation ranks (r). Color arrangements are same as in figure 9.18.

solution. Figure 9.18 presents the magnitude of the FFT of the local streamwise velocity signals previously presented. Also in this case, the top plot refers to the point located outside of the wake, while the bottom plot refers to the in-wake point spectrum. The black, continuous lines refer to the FFT of the local velocity signal obtained with the LES full order model. The red dash-dotted lines represent the spectrum of the signal composed only by the train dataset which, of course, is characterized by half the sampling frequency with respect to the full LES signal. Finally, the green and yellow dashed lines respectively denote the corresponding plots obtained by means of truncated DMD (left panel) and PODI (right panel) models employing 20 and 80 modes each. figure 9.19 — which also employs the line color arrangement just described — displays analogous spectral results obtained when the pressure field is considered. We point out that, based on experimental results presented in [229], a Re = 5000 flow past a sphere results in the Strouhal number $S_t = 0.2$ associated with vortex shedding. This corresponds to $f = S_t U/D = 20\,Hz$ being $U = 1.0\,m/s$ and $D = 0.01\,m$. In figure 9.18 and figure 9.19, it is clearly visible that a 20 Hz peak appears in the LES signal spectra when the velocity and pressure probe is located within the cylinder wake (lower plots). Such peak is accurately reproduced by both PODI and DMD results.

The top plots in both figure 9.18 and figure 9.19 show that all the reduced models considered allow for a sufficiently good reconstruction of the solution spectra for a point outside of the wake. In fact, all the yellow dashed curves appear very close to the black continuous one representing the LES solution spectrum. Thus, this confirms that considering 80 modes or more leads to a spectrum that is indistinguishable from the original, especially for the PODI results in which a relative superiority is again noted in comparison with the spectra from DMD models. Yet, the higher frequency turbulent structures occur in the wake. In such region, as suggested by the bottom plots in figure 9.18 and figure 9.19, the frequency content of the streamwise velocity and the pressure solutions is definitely richer, and the accuracy of the reduced PODI and DMD solutions is clearly lower when higher frequencies are considered. Here, for velocity signals, the plot suggests that considering 80 modes both PODI and DMD algorithms lead to good quantitative spectral reconstructions of the velocity signal for frequencies up to approximately 110 Hz. Interestingly, being based on the train signal yellow dashed line, in this case the PODI and DMD algorithms seem able to improve the behavior of the signal interpolated at half the sampling frequency, and somewhat extend its accuracy at frequencies very close to the Nyquist one.

It is worth commenting on the ROM sensitivity against grid resolution. When a coarser grid is utilized, larger length scales are modeled. As it is expected to note deviations in the corresponding FOM performance with respect to DNS data, it is also doubtful to expect recovering these missing details via a ROM. Furthermore, ROM predictions closely correlate with the energetic content and the induced frequencies from the employed dataset and respective modes. This implies that, while shifting from LES towards RANS grid resolutions, the underlying field dynamics are expected to be sufficiently recovered with less snapshots and less modes for the ROM.

9.3.6 *Acoustic analysis*

The acoustic analysis carried out in this work is aimed at comparing FWH signal obtained using the full order model data and the corresponding signal obtained with both the PODI and the DMD reduced order models. By a practical standpoint, the FWH post processing described in section 9.2.2 is applied to the pressure and velocity fields obtained from the LES full order simulation, and from both the PODI and DMD reduced models. The FWH integrals are here computed considering two microphones located at $\mathbf{x}_{\text{mic A}} = \{0, 2D, 0\}$ (microphone A) and $\mathbf{x}_{\text{mic B}} = \{2D, 2D, 0\}$ (microphone B). As mentioned, the sphere is centered at the origin $\mathbf{O} = \{0, 0, 0\}$ and has diameter D. The acoustic pressure time-history is converted to sound Spectrum Level SpL $= 20 \log(p/p_{\text{ref}})$, considering $p_{\text{ref}} = 1$ ¯Pa.

We will first focus on the results of the PODI model. The corresponding SpL from both microphones is reported in figure 9.20 and figure 9.21, right panels. In the figures, the different FWH signal curves denote the different number of modes employed in the PODI reconstruction. Also, as seen in previous works where the FWH formulation is used, it is convenient to separate the linear contribution to the acoustic pressure, obtained from the surface integrals in equation (9.3), from the nonlinear part obtained from the volume integrals in equation (9.4). Such contributions are referred to as dipole and quadrupole terms, respectively. This distinction is particularly relevant, as the former contribution only requires the knowledge of the pressure field on the body surface, while the latter component takes into consideration the evolution of the pressure and velocity field in the wake region. More specifically, we emphasize that to obtain an adequate reconstruction of the non-linear acoustic signal, an accurate reconstruction of the vorticity field is necessary.

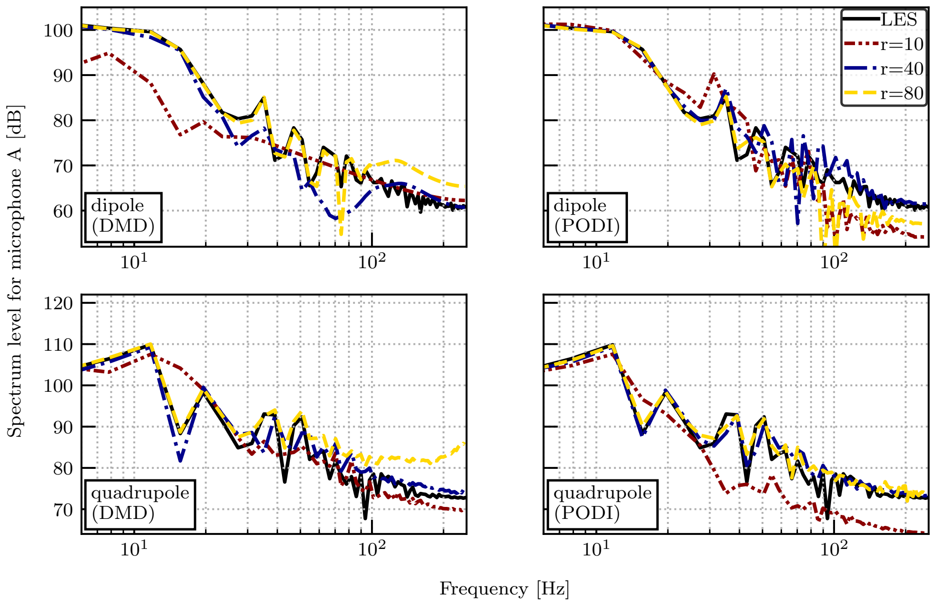


Figure 9.20: Linear dipole (top) and nonlinear quadrupole (bottom) terms of FWH equation evaluated from LES data and compared to corresponding DMD (left) and PODI (right) reduced models at different truncation ranks (r). Microphone A.

In general, the PODI results seem satisfactory especially for what concerns utilizing 80 modes. Figure 9.20 and figure 9.21 (right top plots) show in fact good agreement between the linear acoustic pressure contribution based on the LES pressure field and the corresponding linear contribution computed with PODI employing 80 modes (yellow lines in the plots). As the plots suggest, this trend is appreciable both at microphones A and B. As for the nonlinear contribution, figure 9.20 and figure 9.21 (right bottom panels) also show a satisfactory agreement between the acoustic pressure signals obtained from the LES flow fields and those computed based on PODI. In this case, in the plots referring to both microphones, the blue and yellow lines (40 and 80 modes respectively) on the right bottom panels are sufficiently close to their reference FOM counterpart, represented by the continuous black line.

At a closer look, close agreement between PODI and FOM acoustic pressures, both for the linear and non-linear contributions, can be established at least up to frequencies in the range 80–100 Hz, which may be considered the meaningful frequency interval related to the considered test case. In fact, we point out that, since the reference pressure is set to 1 ¯Pa, the ambient noise may be considered in the range of 60–100 dB. At higher frequencies though, we note that some spurious oscillations appear in the PODI results, especially in the nonlinear noise terms corresponding to microphone B (figure 9.21), at which the value of the integrals in the FWH formulation are strongly conditioned by the presence of the wake. The observed spurious oscillations could be due to an imperfect reconstruction of the vorticity field, although it must be emphasized that the sampling frequency of the original dataset is not very high, $f_s = 500$ Hz, thereby for the train dataset it becomes $f_s = 250$ Hz for such an alternative arrangement. Therefore, it is reasonable to obtain less accurate results at frequencies higher than 125 Hz. Further work will

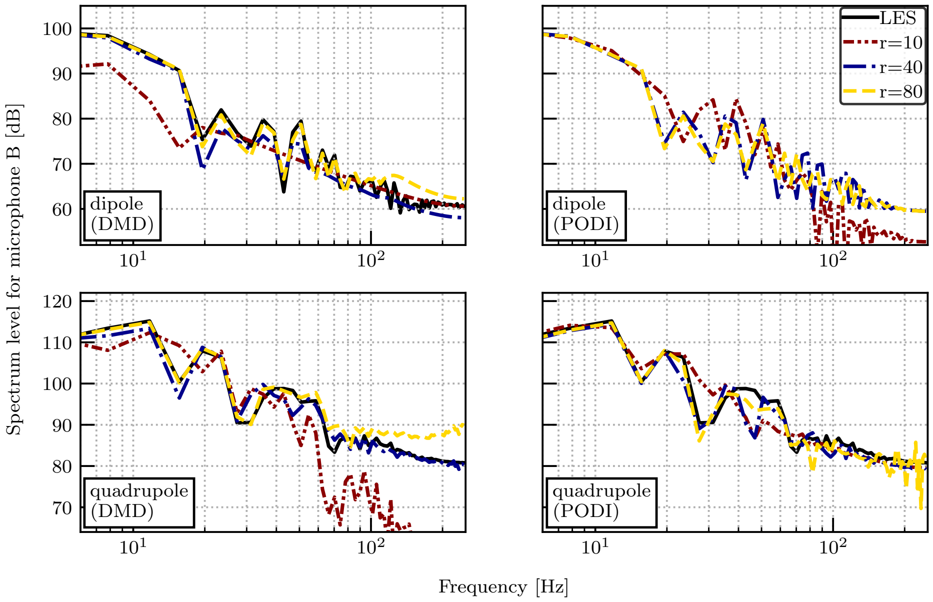


Figure 9.21: Linear dipole (top) and nonlinear quadrupole (bottom) terms of FWH equation evaluated from LES data and compared to corresponding DMD (left) and PODI (right) reduced models at different truncation ranks (r). Microphone B.

be devoted to investigating whether longer temporal records or higher sampling frequencies will eliminate or mitigate the problem.

However, considering the 80 modes PODI reconstruction, the maximum error is of the order of 5 dB, observed at microphone B, for the non linear terms (figure 9.21 bottom right panel). In addition, the main peaks observed at very low frequencies, up to 20 Hz, are well captured by the PODI signal. In such case, the agreement is also verified for the 40 modes signals. As these low frequency peaks describe the main and most energetic features of the acoustic signal, this indicates that a moderate amount of modes might still be a good compromise when the acoustic analysis is only aiming at a general characterization of the principal noise features. Finally, we must point out that the errors observed in the non-linear noise contributions are not in general higher than those introduced in the linear contributions, as might be expected given the interpolation procedure involved in PODI. In some occurrence (figure 9.20 top right plot), the linear contribution errors are even quite surprisingly higher than their nonlinear counterparts. This might be related to the fact that the linear noise contributions are only based on pressure evaluations on the body surface, while the non-linear contributions are based on volume integrals. The POD procedure used selects the modal shapes so as to minimize the error in a norm based on the whole volumetric solution, rather than only on a surface restriction. For such reason the modal shape selected might result in non optimal results in the computation of surface integrals. Possible gains might then be obtained, if needed, adding a separate PODI only built on the body surface degrees of freedom, which would result in a modest increase of the computational cost.

As regards the acoustic signals provided by the DMD method, a good agreement is observed with respect to the FOM spectrum, up to about 80 Hz, for both micro-

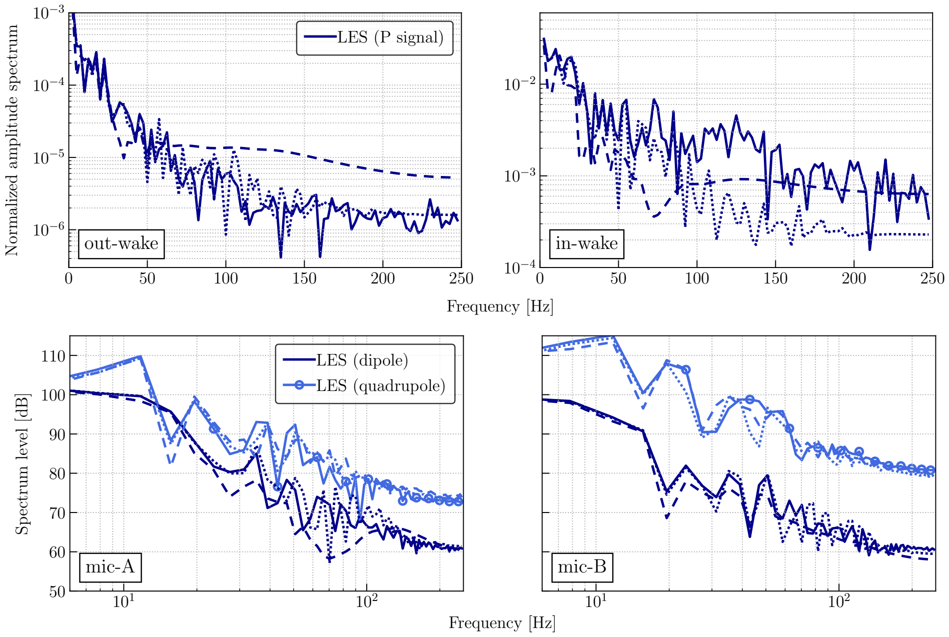


Figure 9.22: Top: FFT spectrum of the pressure signal at points located out of wake (left) and inside the wake (right). Bottom: dipole and quadrupole acoustic noise spectrum measured at mic-A (left) and mic-B (right) locations. PODI spectra (·····) demonstrate better agreement with LES data (——) compared with DMD (– –) ones. Both PODI and DMD results in the graphs have been obtained employing $r = 40$ modes.

phones, and both linear and non-linear terms, see figure 9.20 and figure 9.21 (left panels). The most accurate signals are obtained using 40 and 80 modes, while for the signal related to 10 modes we observe, as expected, a considerable discrepancy. In general, the reconstruction of the vorticity field, obtained by both POD and DMD, is found to provide an adequate input field for the acoustic model. Particularly, the volume integral of the FWH equation involves both the velocity and the pressure field in the wake region. Having observed a good match of the ROM signals with respect to the FOM reference signals, we may conclude that the entire spectrum associated with the vortex wake has been adequately reconstructed.

9.4 ROMS CROSS-COMPARISON

For additional clarity, the spectral and acoustic analyses discussed in the previous sections, cf. figure 9.18, figure 9.19, figure 9.20, and figure 9.21, are here reported in single graphs directly comparing both DMD and PODI results against LES data. In particular, we are considering both ROMs truncated at $r = 40$. The aim is to provide an easier comparison of the ROMs performance. Figure 9.22 depicts FFT spectrum of the pressure signal (inside and outside the wake, top plots) and the dipole and quadrupole acoustic noises (microphones A and B, bottom plots). In all plots DMD (– –) and PODI (·····) results are reported alongside LES (——) results. The plots clearly indicate that, given equal modal truncation level, PODI models appear to accurately recover a wider range of frequencies compared to DMD models.

9.5 CONCLUSIONS AND PERSPECTIVES

This chapter discussed the application of data-driven dimensionality reduction algorithms on a hydroacoustic dataset which was numerically measured using Large Eddy Simulation fluid dynamic turbulence model and the Ffowcs Williams and Hawkings acoustic analogy. An extensive set of data was presented to fully characterize the ability of both DMD and POD to reconstruct the flow fields, based on their spectral and energetic contents, with all spatial and temporal frequencies needed to support accurate noise predictions.

First, SVD analysis did not indicate the presence of significant constraints on the modal truncation rank for such flows. In fact, no significant singular value energy gaps were observed. SVD was then used to extract DMD and POD modes with associated coefficients, and to employ them for flow fields reconstruction. In general, both DMD and POD algorithms showed efficient reconstruction accuracy. Spatial and temporal error analyses indicated relatively lower error magnitudes in POD-based reconstructed fields. On the other hand, statistical analysis and vortical structures identification methods demonstrated the ability of DMD to capture finest wake scales by employing higher modes, compared with POD.

Second, two data-driven reduced models based on DMD mid cast and on PODI were created utilizing half the LES original dataset. Both DMD and POD based reduced models showed good efficiency and accurate flow reconstruction. In addition, the spectral analysis of the reduced flow solutions at selected points inside and outside vortical wake regions indicated that both models were able to recover most of the model flow frequencies in the ranges of interest for the acoustic analysis. In particular, PODI showed notable capability to capture additional frequencies which were present in the original dataset, but not in the subset employed to train the model. As a consequence, both data-driven reduced models developed proved efficient and sufficiently accurate in predicting acoustic noises.

Given the results obtained, introducing bluff body geometrical parameterization as additional dimension for the PODI analysis is an interesting possibility for future works. A further future perspective which is currently being explored, is represented by the development of a reduced model employing the POD modal decomposition for the Galerkin projection of the continuity and momentum equations for the fluid dynamic variables. Other studies could be devoted to better understand how the choice of the interpolator for PODI affects the results.

10

A STEP TOWARDS PARAMETRIC DMD FOR CFD PREDICTIONS

In this chapter we present an advanced computational pipeline for the approximation and prediction of the lift coefficient of a parametrized airfoil profile. The non-intrusive reduced order method is based on DMD and it is coupled with Dynamic Active Subspaces (DyAS) to enhance the future state prediction of the target function and reduce the parameter space dimensionality. The pipeline is based on high-fidelity simulations carried out by the application of finite volume method for turbulent flows, and automatic mesh morphing through radial basis functions interpolation technique. The proposed pipeline is able to save 1/3 of the overall computational resources thanks to the application of DMD. Moreover exploiting DyAS and performing the regression on a lower dimensional space results in the reduction of the relative error in the approximation of the time-varying lift coefficient by a factor 2 with respect to using only the DMD. All the results presented in this chapter appeared in [271].

10.1 LITERATURE REVIEW

Model Order Reduction is nowadays a quite popular and consolidated technique, applied to several fields of engineering and computational science thanks to the remarkable computational gain granted for the solution of the governing equations. The Model Order Reduction (MOR) goal is in fact that of reducing the dimension of the studied system without altering some important properties of the original problem. This typically results in more efficient, time saving computations. Among other fields, ROMs are frequently and successfully applied to problems governed by parametric PDEs, for which many solutions of the same PDE in correspondence with different parameters are required. This paradigm is for example encountered in the context of parametric optimal control problems, uncertainty quantification, and shape optimization.

Model reduction for PDEs has been historically obtained in different ways. In some cases, very successful reduced models have been obtained at the level of the governing equations, based on physical considerations. This is for instance the case of the potential flow theory in the fluid dynamics field. In other cases, the reduction can be introduced at the discretization level, as is the case, for instance, for the Boundary Element Method used in structural analysis, fluid mechanics, electro-magnetism and acoustics studies. In the case in which parametric PDEs are considered, a possible approach to obtain efficient reduced order models is to sample the solution manifold by creating a solutions database corresponding to different parameters, using a high-dimensional discretization, then combine the latter to identify the intrinsic lower dimension of the problem. For parametric reduced order models see [112, 212, 226], while for a more applications oriented overview we suggest [227, 230, 268].

For parametric time-dependent problems, a proper orthogonal decomposition approach can be applied to reduce the dimensionality of the system, as in [97, 116]. In this chapter we propose a novel data-driven approach for parametric dynamical systems, combining DMD with AS property. These two relatively new methodologies provide a simplification of the dynamical system, and an analysis of the input parameter space of a given target function, respectively. Exploiting AS property we are able to obtain an estimation of the importance of the parameters of such function, as well as a reduction in the number of parameters. Moreover the methods are equation-free, being based only on input/output couples and do not make assumptions on the underlying governing equations.

We define a generic scalar output $v(\boldsymbol{\mu}, t) \in \mathbb{R}$ that depends both on time t and on the parameters of the model $\boldsymbol{\mu} \in \mathbb{D} \subset \mathbb{R}^k$, with k denoting the dimension of the parameter space. We denote the state of the parametric system at time t with $v_t(\boldsymbol{\mu}) \in \mathbb{R}$. The solution manifold in time is approximated using the DMD in order to obtain an approximation of the linear map A defined as:

$$v_{t+1}(\boldsymbol{\mu}) = A(v_t(\boldsymbol{\mu})). \tag{10.1}$$

It is easy to note that using equation (10.1) we have the possibility to forecast a generic future state of the parametric system.

To numerically compute the linear operator A, we need to sample the parameter space $\mathbb{D}$, and for each time store the quantity of interest for each parametric configuration. Formally, considering a set of parameter samples with dimension Ns, the discrete vector referring to the system state at time t results:

$$\mathbf{v}_t = \begin{bmatrix} v_t(\boldsymbol{\mu}_1) & \dots & v_t(\boldsymbol{\mu}_{Ns}) \end{bmatrix}^T \in \mathbb{R}^{Ns}. \tag{10.2}$$

Collecting several time states $\mathbf{v}_i(\boldsymbol{\mu})$ for $i = 1, \dots, m$, we compute the operator $\mathbf{A}$ with a best-fit approach such that $\mathbf{v}_{t+1} \approx \mathbf{A}\mathbf{v}_t$. Once computed the future prevision, we are able to exploit the relation between the input parameters $\boldsymbol{\mu}_i$ and the related outputs $v_{\text{future}}(\boldsymbol{\mu}_i)$ to approximate the output for any new parameter. In this work we use a GPR [106, 287], but any regression or interpolation method can be used. We underline that the chosen regression model has to be fitted for any forecasted time we want to analyse.

The high dimensionality in the parameter space may incur on the inability to solve many-query problems with sufficiently high fidelity, thus causing a decrease in the accuracy of the solution approximation. For this reason we couple the regression with the AS property in order to perform a sensitivity analysis of function $v_t(\boldsymbol{\mu})$. AS indeed is able to provide an approximation g of a scalar function f, where the input parameters of g are a linear combination of the original parameters of f. The coefficients of such combination give information about the importance of the original parameters. In this work, we use this information to reduce the dimension of the parameter space — in which we build the regression — by not considering the parameters whose AS coefficients are smaller than a certain threshold, that is they are almost zero.

The developed methodology is tested on an aeronautics application given by the flow past an airfoil profile. As output of interest we considered the lift coefficient and the parameters vector $\boldsymbol{\mu}$ describes geometrical transformations according to the morphing technique proposed in [114]. The fluid dynamics problem is described using the incompressible Navier–Stokes equations with turbulence modeling. These are discretized using a finite volume approximation. The deformed meshes corresponding to different input parameters are automatically obtained exploiting a RBF mesh morphing technique.

This chapter is structured as follows: in section 10.2 we present the general parametric problem over which we apply the proposed numerical pipeline, providing some information about the geometrical deformation. In section 10.3 we briefly present the AS extension to time-dependent scalar functions, while to recall the DMD method we refer to section 5.2. Finally in section 10.4 we show the numerical setting of the problem and the results obtained, while in section 10.5 we propose some final remarks and highlight possible future developments.

10.2 THE PARAMETRIC PROBLEM

Let be given the unsteady incompressible Navier-Stokes equations described in an Eulerian framework on a parametrized space-time domain $Q(\boldsymbol{\mu}) = \Omega(\boldsymbol{\mu}) \times [0, T] \subset$

$\mathbb{R}^d \times \mathbb{R}^+$, $d = 2, 3$ with the vectorial velocity field $\mathbf{u} : Q(\boldsymbol{\mu}) \to \mathbb{R}^d$, and the scalar pressure field $p : Q(\boldsymbol{\mu}) \to \mathbb{R}$ such that:

$$\begin{cases} \mathbf{u_t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot 2\nu \nabla^s \mathbf{u} = -\nabla p & \text{in } Q(\boldsymbol{\mu}), \\ \nabla \cdot \mathbf{u} = \mathbf{0} & \text{in } Q(\boldsymbol{\mu}), \\ \mathbf{u}(t, x) = \mathbf{f}(\mathbf{x}) & \text{on } \Gamma_{\text{in}} \times [0, T], \\ \mathbf{u}(t, x) = \mathbf{0} & \text{on } \Gamma_0(\boldsymbol{\mu}) \times [0, T], \\ (\nu \nabla \mathbf{u} - p\mathbf{I})\mathbf{n} = \mathbf{0} & \text{on } \Gamma_{\text{out}} \times [0, T], \\ \mathbf{u}(0, \mathbf{x}) = \mathbf{k}(\mathbf{x}) & \text{in } Q(\boldsymbol{\mu})_0, \end{cases} \tag{10.3}$$

holds. Here, $\Gamma = \Gamma_{\text{in}} \cup \Gamma_0 \cup \Gamma_{\text{out}}$ is the boundary of $\Omega(\boldsymbol{\mu})$ and it is composed by three different parts Γ_{in}, Γ_{out} and $\Gamma_0(\boldsymbol{\mu})$ that indicate, respectively, inlet boundary, outlet boundary, and physical walls. The term $\mathbf{f}(\mathbf{x})$ depicts the stationary non-homogeneous boundary condition, whereas $\mathbf{k}(\mathbf{x})$ denotes the initial condition for the velocity at $t = 0$. Shape changes are applied to the domain Ω, and in particular to its boundary $\Gamma_0(\boldsymbol{\mu})$ corresponding to the airfoil wall. Such shape modifications are associated to numerical parameters contained in the vector $\boldsymbol{\mu} \in \mathbb{R}^k$ which, in the numerical examples shown in this work has dimension $k = 10$. As said, the only portion of the domain boundary subject to shape parametrization is the physical wall of the airfoil $\Gamma_0(\boldsymbol{\mu})$, which in the undeformed configuration corresponds to the 4-digits, NACA 4412 wing profile [1, 126]. To alter such geometry, we adopt the shape parametrization and morphing technique proposed in [114], where k shape functions are added to the airfoil profiles. Let y_u, and y_l be the upper and lower ordinates of a NACA profile, respectively. We express the deformation of such coordinates as

$$y_u = \overline{y_u} + \sum_{i=1}^{5} c_i r_i, \tag{10.4}$$

$$y_l = \overline{y_l} - \sum_{i=1}^{5} d_i r_i, \tag{10.5}$$

where the bar denotes the reference undeformed state, which is the NACA 4412 profile.

The parameters $\boldsymbol{\mu} \in \mathbb{D} \subset \mathbb{R}^{10}$ are the weights coefficients, c_i and d_i, associated with the shape functions r_i. The range of each parameter will be specified in section 10.4. The explicit formulation of the shape functions can be found in [114], we report them in figure 10.1.

After the reference profile is deformed, we also apply the same morphing to the mesh coordinates by using a RBF interpolation method [39, 172, 185]. With this approach the movement $\mathbf{s}$ of all the points which do not belong to the moving boundaries is approximated by an interpolatory radial basis function:

$$\mathbf{s}(\mathbf{x}) = \sum_{i=1}^{N_b} \beta_i \xi(||\mathbf{x} - \mathbf{x}_{b_i}||) + q(\mathbf{x}), \tag{10.6}$$

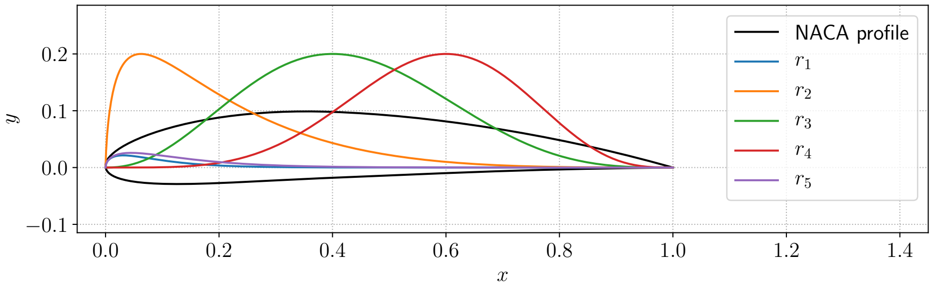


Figure 10.1: Airfoil shape functions with respect to the profile abscissa. The leading edge corresponds to $x = 0$. The functions r_i are rescaled by a factor equal to 0.2 for illustrative reasons.

where $\mathbf{x}_{b_i}$ are the coordinates of points for which we know the boundary displacements, for this particular case the points located on the wing surface. N_b is the number of control points on the boundary, ξ is a given basis function, $q(\mathbf{x})$ is a polynomial. The coefficients β_i and the polynomial $q(\mathbf{x})$ are obtained by the imposition of interpolation conditions

$$\mathbf{s}(\mathbf{x}_{b_i}) = \mathbf{d}_{b_i}, \tag{10.7}$$

where $\mathbf{d}_{b_i}$ is the displacement value at the boundary points and by the additional requirement:

$$\sum_{i=1}^{N_b} \beta_i q(\mathbf{x}_{b_i}) = 0. \tag{10.8}$$

In the present case, we select basis functions for which it is possible to use linear polynomials $q(\mathbf{x})$. For more informations concerning the selection of the order of polynomials see [17]. Finally the values of the coefficients β_i and the coefficients δ_i of the linear polynomials q can be obtained by solving the linear problem:

$$\begin{bmatrix} \mathbf{d}_b \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{b,b} & P_b \\ P_b^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\delta} \end{bmatrix}, \tag{10.9}$$

where $\mathbf{M}_{b,b} \in \mathbb{R}^{N_b \times N_b}$ is a matrix containing the evaluation of the basis functions $\xi_{b_i b_j} = \xi(\|\mathbf{x}_{b_i} - \mathbf{x}_{b_j}\|)$, and $\mathbf{P}_b \in \mathbb{R}^{N_b \times (d+1)}$ is a matrix where d is the spatial dimension. Each row of this matrix, that contains the coordinates of the boundary points, is given by $\text{row}_i(\mathbf{P}_b) = \begin{bmatrix} 1 & \mathbf{x}_{b_i} \end{bmatrix}$. Once the system of equation (10.9) is solved one can obtain the displacement of all the internal points using the RBF interpolation:

$$\mathbf{d}_{\text{in}_i} = \mathbf{s}(\mathbf{x}_{\text{in}_i}), \tag{10.10}$$

where $\mathbf{x}_{\text{in}_i}$ are the coordinates of the internal grid points. The computation of the displacement of the grid points entails the resolution of a dense system of equations that has dimension $N_b + d + 1$. Usually, the number of boundary points N_b is much smaller than the number of grid points N_h.

10.3 GLOBAL SENSITIVITY ANALYSIS THROUGH ACTIVE SUBSPACES

In this chapter we want to study the behaviour of a target function $f(\boldsymbol{\mu}, t) : \mathbb{R}^k \times \mathbb{R}^+ \to \mathbb{R}$ that depends on the parameters $\boldsymbol{\mu}$ and on time t as well. This results in extending the active subspaces property to dynamical systems, that means having to deal with time-dependent uncentered covariance matrix $\mathbf{C}(t)$, and corresponding eigenvectors $w_i(t)$. Efforts in this direction has been done in [56] for a lithium ion battery model, in [165] for long term model of HIV infection dynamics, and more recently an application of dynamic mode decomposition and sparse identification to approximate one-dimensional active subspaces in [3]. In these works they refer to DyAS as the time evolution of the active subspaces of a time-dependent quantity of interest.

DyAS are useful to assess the importance of each input parameter at given times and to study how the weights associated to the inputs evolve. In the following we are going to compute the AS for a set of equispaced times t_i. If some of the parameters are almost zero in the entire time window we can safely ignore them in the construction of the Gaussian process regression.

10.4 COMPUTATIONAL PIPELINE

In the present section we will discuss the numerical experiments carried out to test the DyAS analysis and present the results obtained. As reported in section 10.2, each high fidelity simulation is based on a parametric fluid dynamic model governed by the RANS equations. Thus, a number of flow simulations have been carried out selecting different samples in the parametric space to test the performance — in terms of lift coefficient — of different airfoil shapes. The simulations made use of both the RANS solver provided in the OpenFOAM [284] finite volumes library, and of the DMD acceleration methodology described in section 5.2. Once the lift coefficients output were available for all the samples tested in the input parameters space, the DyAS analysis was applied to assess possible parameter redundancy. The elimination of the redundant parameters detected in the DyAS analysis allowed for the generation of a surface response model based on a lower dimensional space, which has been finally tested against the original RANS model accelerated through DMD, and against the surface response model based on the original input parameter space. Figure 10.2 graphically summarizes the proposed pipeline, clarifying how the methods (and the software) are integrated together, while the following sections will further detail each part of the computational pipeline just outlined.

10.4.1 *Parametric shape deformation*

The fluid dynamics problem is resolved using the finite volume method. The wing is immersed in a rectangular domain according to figure 10.3. The reference mesh counts 46500 hexahedral cells and is constructed using the *blockMesh* utility of the OpenFOAM library. Figure 10.3 depicts a detail of the grid in proximity of the wing. The meshes in the deformed configuration have been obtained starting from the reference configuration using a radial basis function smoothing algorithm similar to

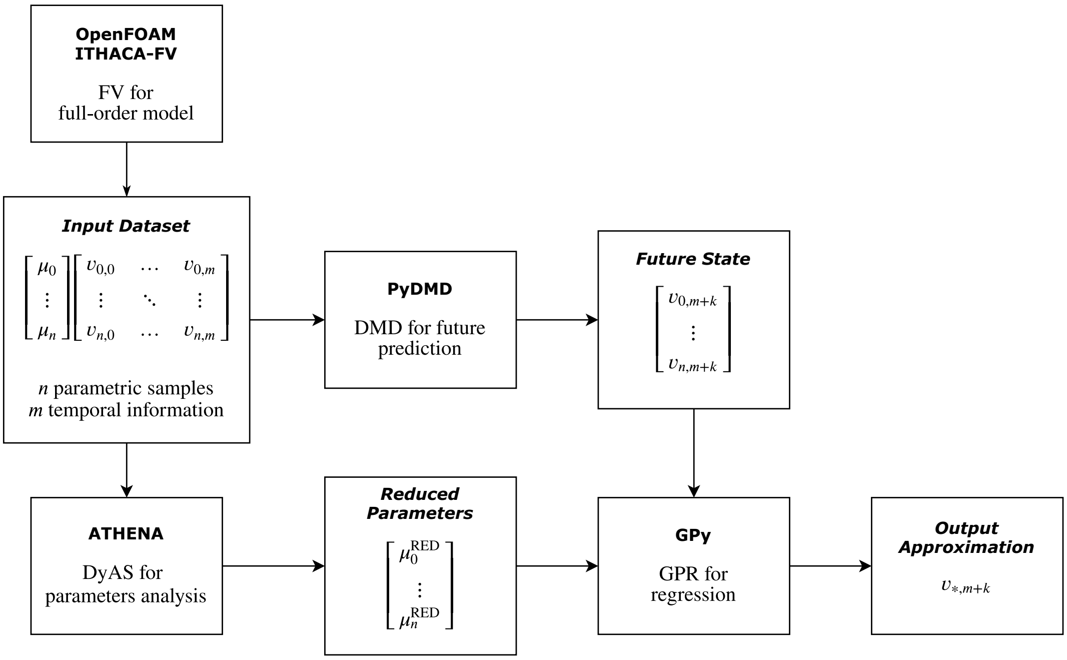

OpenFOAM
ITHACA-FV
FV for full-order model
Input Dataset
$\begin{bmatrix}\mu_0\\ \vdots\\ \mu_n\end{bmatrix}\begin{bmatrix}v_{0,0} & \cdots & v_{0,m}\\ \vdots & \ddots & \vdots\\ v_{n,0} & \cdots & v_{n,m}\end{bmatrix}$
n parametric samples
m temporal information
PyDMD
DMD for future prediction
Future State
$\begin{bmatrix}v_{0,m+k}\\ \vdots\\ v_{n,m+k}\end{bmatrix}$
ATHENA
DyAS for parameters analysis
Reduced Parameters
$\begin{bmatrix}\mu_0^{\text{RED}}\\ \vdots\\ \mu_n^{\text{RED}}\end{bmatrix}$
GPy
GPR for regression
Output Approximation
$v_{*,m+k}$

Figure 10.2: Flowchart representing the proposed computational pipeline.

the one implemented in [31]. A single deformation corresponds to a sample $\boldsymbol{\mu}$ in the parameter space $\mathbb{D} := [0, 0.03]^{10} \subset \mathbb{R}^{10}$. Therefore all the deformed meshes share the same number of cells and the same mesh topology. In particular Wendland [285] second order kernel functions with radius $r_{\text{RBF}} = 0.1$ m have been used. The control points of the RBF procedure have been placed on each mesh boundary point located onto the wing surface. Since the outer boundary points are fixed we decided to neglect them from the RBF computation using a smoothing function defined in such a way that the RBF contribution reduces to zero after a certain distance from a focal point [127]. Particularly, the focal point has been placed in the geometric center of the airfoil chord segment and the distance from the focal point after which the RBF contribution is neglected is set to $r_{\text{out}} = 7$ m. In figure 10.4 we depict the envelope of all the tested configurations, and the flow velocity streamlines for a particular sample in the parameter space. A uniform and constant velocity equal to $\mathbf{u}_{\text{in}} = 1$ m/s is set at the inlet boundary, while the constant value of the kinematic viscosity is set to $\nu = 2\text{e}{-5}$ m^2/s. This configuration, considering a chord length $D = 1$ m, corresponds to Reynolds number $\text{Re} = 50000$. As well known, a flow characterized by Reynolds number of such magnitude requires turbulence modeling to be numerically simulated with reasonable computational effort. In the present work, turbulence has been modeled using a RANS approach with a Spalart-Allmaras turbulence model [256]. The pressure velocity coupling is resolved in a segregated manner making use of the PIMPLE algorithm which merges the PISO [124] and the SIMPLE [200] algorithm. The time step used to advance the simulation in time is set constant and equal to $\Delta t = 1\text{e} - 3$ s. The convective terms have been discretized using a second-order upwinding scheme, while the diffusion terms are discretized using a linear approximation scheme with non-orthogonal correction. The time discretization is resolved using a second order backward differentiation formula.

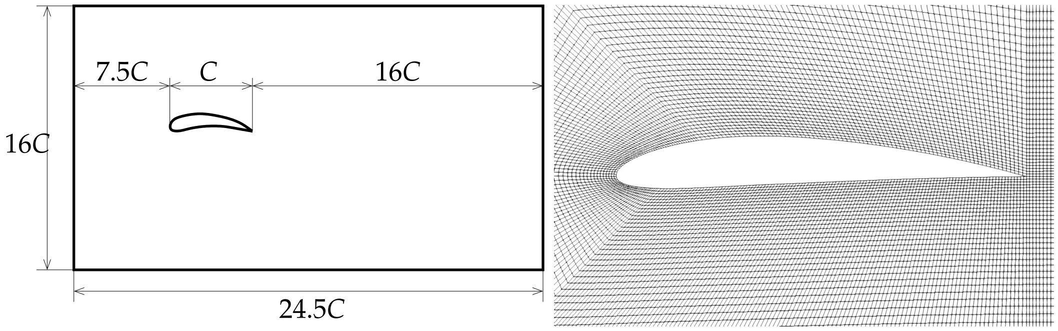


Figure 10.3: Sketch of the computational domain used to solve the fluid dynamics problem in its reference configuration. The left picture reports a schematic view on the domain with the main geometrical dimensions. The right plot reports a zoom on the mesh in the proximity of the wing.

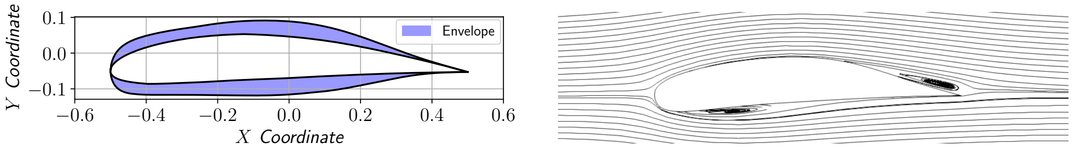


Figure 10.4: The left picture reports in light blue the envelope of all the tested configurations used during the training stage. The right picture depicts the flow velocity streamlines for one particular sample inside the training set $\boldsymbol{\mu} = [0.0071, 0.0229, 0.0015, 0.0015, 0.0087, 0.0107, 0.0033, 0.0130, 0.0247, 0.0280]$.

The simulation is advanced in time until the flow has reached stationary behavior. For the present problem, setting a total simulation time $T_s = 30$ s is sufficient to reach a solution which is reasonably close to the steady state one. In order to check the consistency of the numerical results, the stationary lift coefficient computed for the reference configuration, which corresponds to a standard NACA 4412 profile with a $0°$ angle of attack, has been compared with data from literature [4]. The computed lift coefficient for such setting is equal to $C_L = 0.355$ and the available reference value varies between $C_L = 0.1804$ and $C_L = 0.3708$ depending on the value of N_{crit} (which is used to model the turbulence of the fluid or roughness of the airfoil). Therefore, our numerical results are in line with available data in existing literature[1].

10.4.2 *Parameter space reduction*

The present section will discuss the application of DyAS to the problem of the two dimensional turbulent flow simulation past airfoil sections with parameterized shape. Such a fluid dynamic problem is relevant in several engineering fields, as it is encountered in a number of industrial applications, ranging from aircraft and automotive design, to turbo machinery and propeller modeling. We must

[1] Such comparison is not exhaustive to completely verify the accuracy and the reliability of the full order model numerical simulations. It is however beyond the scope of this work to perfectly match experimental activities or previous numerical results with the full order simulations. More accurate FOM results would of course result in more accurate ROM results but would not affect the presented methodology.

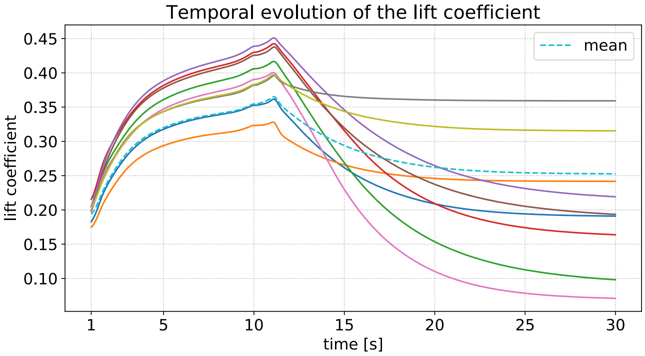


Figure 10.5: The temporal evolution of the lift coefficient from 1 s to 30 s for 9 different parameters, together with the mean (dashed). The angle of attack is fixed for all the airfoil profiles and it is equal to 0°.

here point out that in this work, the DMD method is used for faster evaluation of the parameterized airfoils lift towards a steady state regime solution. We remark that, since DMD is designed for time evolutionary problems, the same procedure can be used in the same fashion, to speed up convergence to periodic regime solutions [156]. Indeed, recent work on hydroacoustic computations based on LES suggested that DMD modal decomposition can successfully be employed in the reconstruction of complex and turbulent flow fields [92] provided that the snapshots used are enough to characterize all the relevant time and space frequencies in the flow. In addition, we observed that complex full order flows characterized by richer spectra require a higher amount of modes to obtain accurate flow fields reconstruction. Thus, our experience suggests that the ROMs used in this work are indeed effective when employed with more complex physics. For such reason, given our experience, we infer that the design pipeline here presented can also be used to study the unsteady dynamics of bubbles and vortices past the airfoil. Obviously one requirement of such type of problems would be a suitable FOM able to capture transition phenomena occurring in the stall region. For example, we believe that the underlying high fidelity URANS solver would not be appropriate and that a transition to a LES approach would be required. For projection-based ROMs in a turbulent setting see [115, 116, 259].

A few plots describing the DyAS results for the lift coefficient output are presented in figure 10.6, figure 10.7, figure 10.8, and figure 10.9. The plots in the figures are aimed at representing the evolution of the active subspace effectiveness and composition over the time dependent flow simulations. More specifically, the left diagram in each figure plots the lift coefficient at each sample point tested, as a function of the first active variable obtained through a linear combination of the sample point coordinates in the parameter space, that is $f(\boldsymbol{\mu}, t)$ against $\mathbf{W}_1^T \boldsymbol{\mu}$. Presenting the components of the first eigenvector of the uncentered covariance matrix, the right plot in each figure indicates the weights used in such linear combination to obtain the first active variable. In summary, the right diagram in each Figure suggests the impact of each of the original parameters on the first active variable, while the left diagram is an indicator of how well a one dimensional active subspace is able to represent the input to output relationship. Following

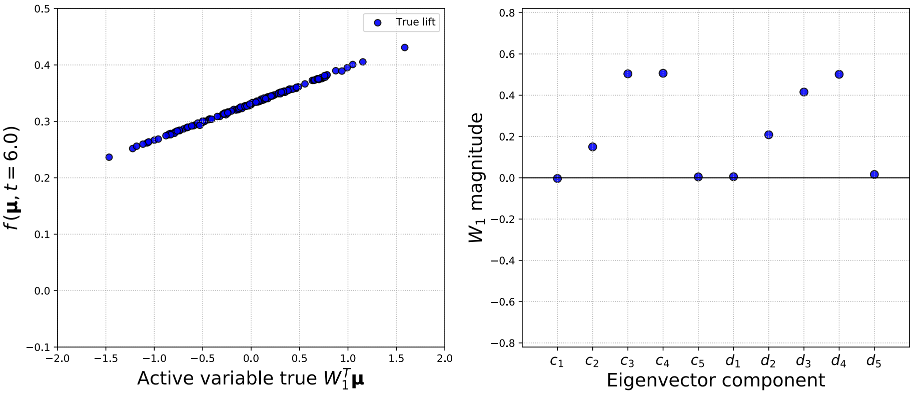


Figure 10.6: On the left the sufficiency summary plot for the lift coefficient at time $t = 6.0$ seconds. On the right the first eigenvector components at the corresponding parameters.

the evolution of these two indicators it is possible, at each time instant, to assess how effective the one dimensional parameter dimension reduction is, and what is the sensitivity of the reduced lift coefficient output to variations of the original parameters. The plots in figure 10.6, figure 10.7, figure 10.8, and figure 10.9 show the results of the DyAS at the fixed time instants $t = 6$ s, 10 s, 14 s, 18 s, respectively. We here remark that, given the aforementioned considerations about the solution build up in the first 12 seconds of the simulations, the solutions at $t = 6$ s and $t = 10$ s are not entirely relevant by a physical perspective. Yet, presenting such cases is still helpful in illustrating how the DyAS evolves over time and can be used to evaluate the system behavior and the output sensitivities with respect to the input parameters. For completeness in figure 10.5 we depicted the temporal evolution of 9 different morphed airfoils, and the mean among all the airfoils. A first look at the right plots for each time steps, suggests that the contribution of the parameters corresponding to the bump shape functions r_1, and r_5, for both the top and the bottom part of the airfoil profile are almost negligible. This means the lift coefficient is almost insensitive to variations of these 4 parameters. Alternatively, it can be said that the output function is on average almost flat along directions corresponding to the axes corresponding to parameters c_1, c_5, d_1, and d_5.

Figure 10.6 and figure 10.7 present the characterization of the one dimensional active subspace at time $t = 6$ s and $t = 10$ s, respectively. We can clearly see that the lift coefficient is perfectly approximated along the identified direction, and such direction (the eigenvector elements) is almost the same at $t = 6$ s and $t = 10$ s. This should not completely surprise as both time instants are included in an initial acceleration phase during which the air coming from the inflow boundary is reaching the airfoil. Given the domain arrangement described in figure 10.3, the flow velocity around the impulsively started airfoil leading edge is expected to reach the inflow value at time $t = 10$ s. For such reason, we will focus the description on the plots for $t = 10$ s, although the considerations can be immediately reproduced for previous time steps. The left plot in figure 10.7 suggests that at this meaningful instant, the first active subspace represents the input to output relationship with remarkably good accuracy. In fact, only a single output value corresponds to each

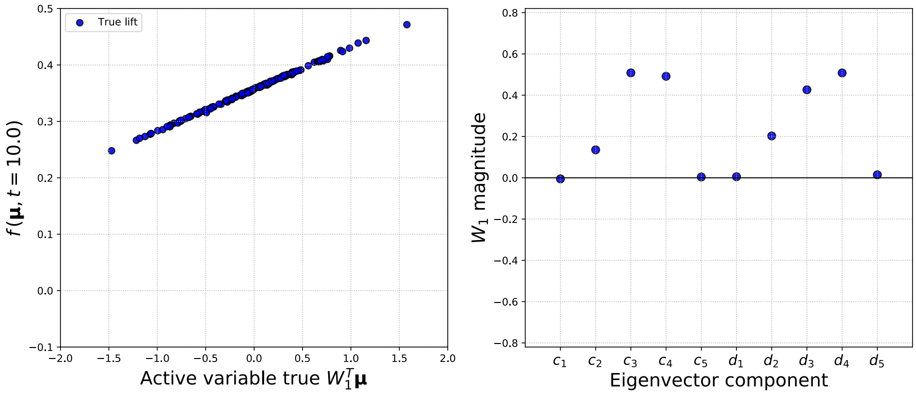


Figure 10.7: On the left the sufficiency summary plot for the lift coefficient at time $t = 10.0$ seconds. On the right the first eigenvector components at the corresponding parameters.

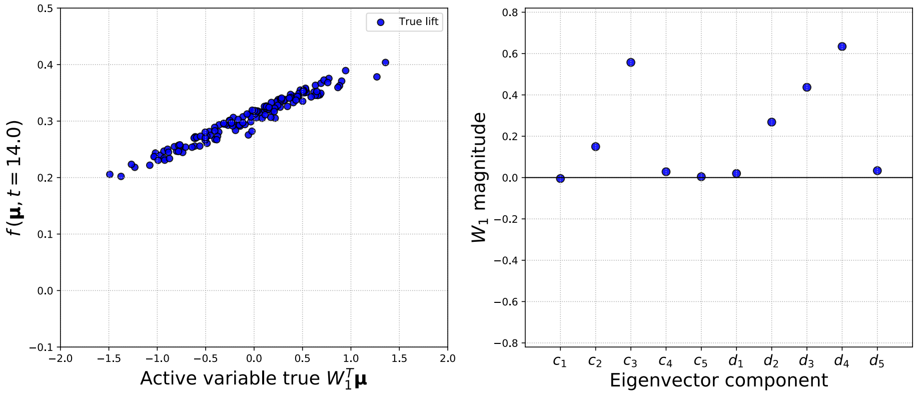


Figure 10.8: On the left the sufficiency summary plot for the lift coefficient at time $t = 14.0$ seconds. On the right the first eigenvector components at the corresponding parameters.

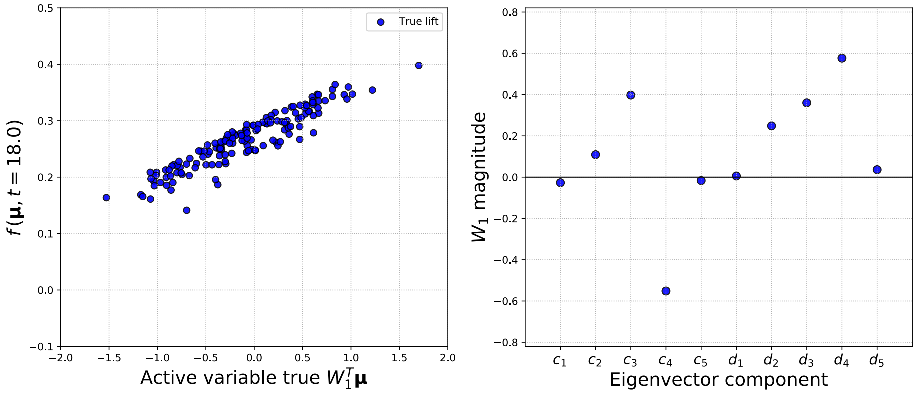


Figure 10.9: On the left the sufficiency summary plot for the lift coefficient at time $t = 18.0$ seconds. On the right the first eigenvector components at the corresponding parameters.

active variable value. In other words, when plotted against the first variable, the output appears like a curve — a line in the present case. A look at the right diagram suggests that the shape parameters having the most impact on the lift generated by the airfoil are c_3, c_4, d_3 and d_4, which are the ones associated to shape functions with peaks located around the middle of the airfoil chord. The positive values of the eigenvector components associated to c_3, c_4, d_3 and d_4, along with the positive slope of the curve in the left plot in figure 10.7 suggest that, at this particular time instant, higher values of lift can be obtained by increasing the airfoil thickness in the mid-chord region.

Similar considerations can be drawn from figure 10.8, which refers the the DyAS analysis carried out at $t = 14$ s. Here, the points in the left diagram do not completely cluster on top of a single valued curve as was the case for the previous time step considered. Compared to what has been observed at $t = 10$ s, the data clearly indicate that at $t = 14$ s an input to output relationship obtained using only a one dimensional active subspace will lead to less accurate lift coefficient predictions. Yet, the points in the plot are still all located within a rather narrow band surrounding a regression line having positive slope. Thus, all the considerations on the lift coefficient sensitivity with respect to variations of the shape parameters that can be inferred from the right plot, will still hold at least by a qualitative standpoint. Here, the eigenvector components suggest that the most influential parameters on the lift coefficient are c_3, d_3 and d_4, while c_2 and d_2 affect the output in lesser but not negligible fashion. Compared to the previous case the importance of coefficient c_4 on the output is significantly reduced. We recall that c_4 is associated with increased y coordinates of the airfoil suction side past the mid-chord region. Thus, we might infer that in the acceleration phase higher lift values are obtained not only increasing the front thickness, but also lowering the camber line in the region past mid-chord.

Figure 10.9 shows the results of the DyAS analysis at $t = 18$ s, when the flow approaches the final regime solution. Following the trend observed for $t = 14$ s, the left plot in the figure indicates that a one dimensional active subspace is not completely able to represent the input to output relationship in a satisfactory fashion. With respect to the previous plots, the output values are here located in an even wider band around a regression line with positive slope. Again, on one hand this increasingly blurred picture suggests that higher dimensional active subspaces are required to reproduce the steady state solution with sufficient accuracy; on the other hand, the diagram still suggests a quite definite trend in the output, which can be exploited for qualitative considerations. Quite interestingly, at the present time step the eigenvector component corresponding to the c_4 coefficient has negative sign. Given the positive slope of the input to output relationship in the left plot of figure 10.9, this implies that increases in the airfoil ordinates on the top side in the region past the mid-chord result in lift loss. Thus, this seems to suggest that an airfoil with a higher camber line curvature, combined with a thicker leading edge region might result in increased lift. This should not surprise, as a similar kind of airfoil would result in a higher downwash due to the increased camber line curvature, yet being able to avoid stall by means of a thicker and rounder leading edge. Thus, the DyAS analysis at different time steps

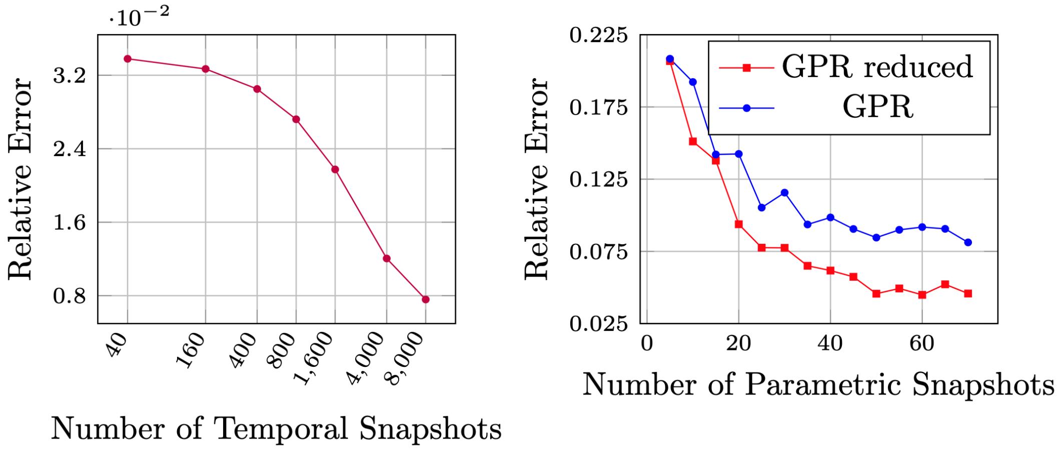


Figure 10.10: Sensitivity analysis of the dimension of the training set for the DMD (left) and for the response surface using GPR (right). For the DMD, we use 70 samples (of the parametric space) evolving in time in $[12, 20]$ s and we measure the mean relative error at time 30 s varying the sampling frequency; for the GPR, we build the response surface using up to 70 sampling lift coefficients at time 20 s and computing the mean relative error over the test dataset composed by 100 test deformations.

shows that as the impulsively started airfoil moves from an acceleration phase to a steady state regime solution, the shape modifications leading to increased lift transit from a purely symmetric increase of the thickness in the mid-chord region, to a non-symmetric modification of the camber line united with a symmetric leading edge thickness increase, respectively. Such behavior is indicated by the sign of c_4 coefficient in the eigenvector characterizing the one dimensional active subspace, which is likely detecting that at steady state, regime solution, airfoils with higher camber line curvature and thicker leading edges produced higher downwash.

We underline that the eigenvector components of all the time instants presented corresponding to the coefficients c_1, c_5, d_1, and d_5 are almost zero. This means that on average the lift coefficient is almost flat along these directions. We are going to exploit this fact by freezing these parameters and constructing a GPR on a reduced parameter space.

10.4.3 *GPR approximation and prediction of the lift coefficient*

The previous analysis pointed out the presence of several input parameters with minimal average influence on the target function. Making use of such consideration we construct a response surface which only depends on the remaining parameters. Both for the full parameter space and the reduced one, we use a Gaussian process regression with a RBF kernel implemented in the open source Python package GPy [91]. We then compare the performance of the two regression strategies by computing the relative error over a test data set composed by 100 samples. The error is computed as the Euclidean norm of the difference between the exact and the approximated solution over the norm of the exact solution. The training set is composed by the same 70 samples, in 10 dimensions for the GPR over the original

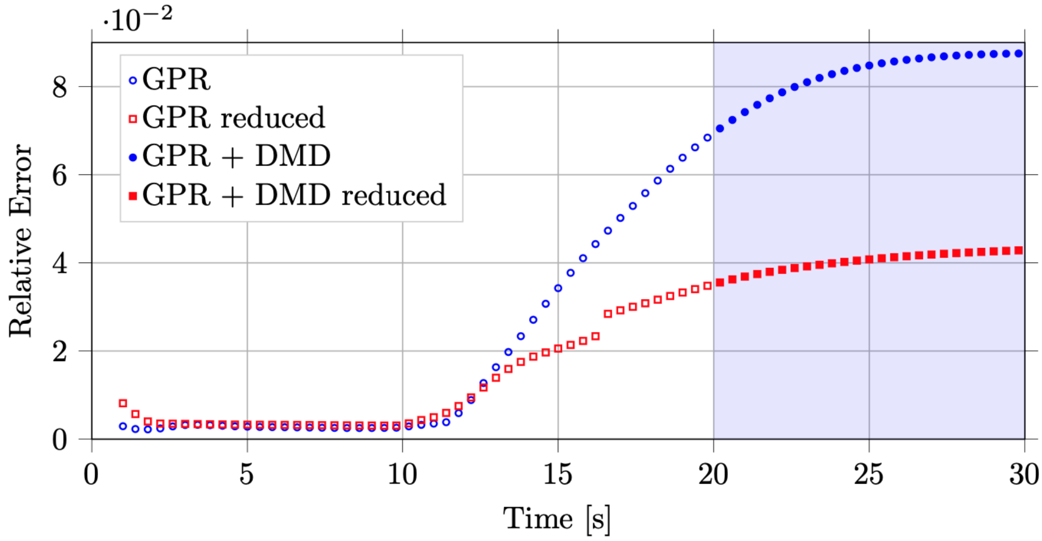


Figure 10.11: The relative error of the approximated outputs at different times. The relative error is computed on 100 test samples, using the high-fidelity lift coefficient to train the regression for $t \leq 20$ s, while for $t > 20$ s the DMD forecasted states are used for the training.

parameter spaces, and in 6 dimensions for the reduced one. Up to $t = 20$ s the training is done using the high-fidelity simulations.

To speed up the convergence to the regime state ($t = 30$ s) we applied the DMD to get the future-state prediction of the lift. In particular, due to the initial propagation of the boundary conditions, for all the 70 training deformations we use the trend of lift coefficients within the temporal interval $[12, 20]$ s to fit the DMD model, that means 8000 temporal information ($\Delta t = 0.001$ s). Since we used 10 POD modes — selected using the energetic criterion — for the projection of the DMD operator, our low-rank operator results of dimension 10. Despite in this case the dimensional reduction is not huge, this approach allows to predict the future state in a very fast fashion. In the high-fidelity model, we need in fact 1508 CPU seconds (on average) to simulate 1 second of the physical model, instead using DMD we can approximate a future state in less than 0.1 CPU seconds. In practices, this means that, to reach the regime state with the standard approach, the simulation lasts $1508 \text{ s} \times 30 \approx 45000$ s, while with the DMD we have $1508 \text{ s} \times 20 + 0.1 \text{ s} \approx 30000$ s, guaranteeing to save $\frac{1}{3}$ of the overall computational load. All the simulations, both at the FOM and at the ROM level have been run serially on an Intel Xeon E5-2640, 2.50GHz CPU. We highlight that this is only a part of the computational saving of the pipeline that we are proposing and is related to the training stage. The DMD allows in fact for $1/3$ reduction of the simulation time required to the FOM as the remaining time is simulated by an approximated model. On the other side, once the reduced order model has been constructed, exploiting the combination of the Gaussian Process approximation and the DMD, it is possible to test new geometries in real time, with a negligible computational cost. Regarding the accuracy, we present in figure 10.10 a sensitivity analysis on the number of training snapshots, varying the temporal sampling period Δt_{DMD} from $1\text{e}-3$ s to 0.2 s and measuring the error on the predicted state at $t = 30$ s. Similarly, we propose an analysis on the GPR accuracy: using a varying number of lift coefficients at $t = 20$ s, we build the response surface and measure the error for untried parameters, both in the full dimensional space and in the reduced one. In figure 10.11 we compare the two GPR performance at each of the time steps analyzed in the simulations. Until 12 s, the regressions behave in a very similar fashion, while from 15 s the accuracy gain obtained by distributing the 70 samples in a lower dimensional space becomes significant. The error gap between the 6 and 10 dimensional response surface in fact, consistently increases from 0.016 at 15 s to 0.045 at steady state. This corresponds to a decrement of the error by a factor 2.

The proposed method achieves better results because it exploits the DyAS to discard the directions of the input parameter space along which the target function does not vary.

10.5 CONCLUSIONS AND PERSPECTIVES

In this chapter we presented a computational pipeline to improve the approximation of the time-varying lift coefficient of a parametrized NACA airfoil. The pipeline comprises automatic mesh deformation through RBF interpolation, high-fidelity simulation with finite volume method of turbulent flow past the airfoil, global sensitivity analysis exploiting AS, and future state prediction via DMD reduced order method. This resulted in more accurate Gaussian process regression of the lift coefficient even if in a reduced parameter space. Despite the turbulent nature of the flow, the selected testcase does not show highly nonlinear phenomena — e.g. stall, reattachment — that usually occur in several fluid dynamics problems. The proposed framework can be extended to address also more complex applications, provided that a suitable number of snapshots is given to characterize the parameter space and frequencies required by the DMD training. Of course such more demanding training requirements would likely result in reduced ROMs speed up and would require case-specific treatments.

After the creation of the high-fidelity solutions database the application of AS highlighted a possible reduction of the parameter space due to negligible contributions of 4 different parameters. We exploit this reduction to construct a GPR over a smaller parameter space, thus improving its performance. Since the training of the regression model is done over 6 dimension instead of 10, given the same high-fidelity database dimension, the GPR is able to better approximate the solution manifold. This results in better lift coefficient predictions for new untried parameters. We also applied DMD to have future-state prediction of the target function up to 30 seconds and proved that the effective gain of the new GPR is preserved also for any time after the 20 seconds simulated with **FV!** (FV!). In particular from 13 seconds the actual gain is significant, at 15 seconds we have an increased performance by a factor 2 in the relative error, which means that performing the regression in the reduced parameter space produces a relative error equal to 0.02, instead of 0.036. Evolving in the future the error drop increases up to 0.045 at regime (0.042 instead of 0.087, keeping the factor 2).

This computational pipeline can be seen as a parametric dynamic mode decomposition for some extent. Moreover, the sensitivity analysis has a negligible computational cost with respect to the creation of the offline high-fidelity database.

Future developments can be the study of adaptive sampling strategies exploiting a generic n-dimensional active subspace, and the coupling of different model order reduction methods. Another possible extension of the presented method regards the possibility to apply the framework to a flow field — e.g. pressure, velocity — rather than to a scalar output. It would be interesting to use this non-intrusive setting as a preprocessing tool to reduce the number of simulations required to build a reduced basis space which is later used in an intrusive manner [258]. We think this new computational pipeline can be of much interest in the context of shape optimization and dynamical systems.

11

CONCLUSIONS AND FINAL REMARKS

In this thesis we developed and analysed several data-driven non-intrusive reduced order models for both input and output spaces. Inspired by the industrial challenges we faced, we proposed more accurate methods for parameter space reduction, such as Kernel-based Active Subspaces and Local Active Subspaces. We also investigate nonlinear data fusion exploiting Active Subspaces to improve the regression performance for scalar functions with low-intrinsic dimensionality. This has been coupled with a general non-intrusive POD-based ROM to improve the solution manifold approximation. Finally we proposed a new algorithm for high-dimensional optimization called Active Subspaces Genetic Algorithm and test it in different contexts.

We applied these new methods to several engineering applications with a particular focus on naval engineering. We presented structural optimization of cruise ships, shape optimization of a combatant hull and a NACA airfoil profile, and the prediction of hydroacoustic noises.

Let us recap some conclusions:

- We studied a nonlinear extension of AS, where we map the input parameters onto a reproducing kernel Hilbert space and then we apply the classical linear AS in this higher dimensional space. We showed how this allows to deal with radial symmetric functions, which do not have an AS by definition, and we applied it to both scalar and vector-valued output functions.

- Using a different approach, we also proposed a local version of parameter space reduction, in which we exploit the existence of a global AS to divide the inputs into clusters aligned with the active direction, and then use Local Active Subspaces. We extensively studied this method for both regression and classification tasks.

- In the multi-fidelity context we investigated the integration of low-fidelity models built through AS without running any additional simulation, within a nonlinear autoregressive scheme. We showed the improved accuracy of NARGPAS with respect to the use of a single high-fidelity in the approximation of functions with a low-intrinsic dimensionality. This represents a cornerstone for the coupling of model and parameter space reduction since we can use this scheme to better approximate each modal coefficient.

- We reviewed classical data-driven ROMs such a Proper Orthogonal Decomposition with Interpolation and Dynamic Mode Decomposition, for parametric and time-dependent problems, respectively. We present the PODI method within a general non-intrusive framework where every abstract component can be changed and adapted for the specific task of interest.

- We extended the classical Genetic Algorithm by exploiting the AS of every generation, and perform the mating and mutation steps within a reduced input space. This greatly accelerate the convergence of the algorithm, especially for high dimensional target functions. We called this method ASGA.
- The main and most challenging application has been the structural optimization of modern cruise ships in collaboration with Fincantieri S.p.A., both for the high dimensionality of the problem and for the integration within the existing software used in the preliminary design phase. We successfully incorporated a data-driven pipeline with reduction in parameter space and PODI. We presented the results obtained in the optimization of a single hull with two different parameterizations.
- Another application in naval engineering has been the shape optimization of a benchmark combatant hull, where we used advanced geometrical morphing techniques to propagate surface modifications of the hull to the actual volumetric mesh used to perform the simulations campaign. This allowed to use non-intrusive ROMs without remeshing the deformed hull. We also used ASGA in an engineering context.
- Regarding more academic applications, we compared PODI and DMD for the approximation and prediction of the flow past a sphere using Large Eddy Simulation simulations. We also used both ROMs to predict the hydroacoustic noise.
- Finally, we made a first step towards a parametric extension of DMD for the approximation of the lift coefficient of a NACA airfoil profile. We also coupled this methodology with a time-dependent AS analysis.

11.1 FUTURE PERSPECTIVES

During the investigation of the different methodologies presented in this thesis, we have identified several research lines that should be further studied and developed.

Parameter space reduction is an active area of research both from the methodological and the applications point of view. The localization method LAS presented in chapter 3 has a great potential also for nonlinear techniques. Moreover the hierarchical top-down clustering developed is very versatile and allows for more sophisticated criteria. Active subspaces could also be more integrated in various optimization algorithms to further accelerate convergence and in general to reduce the time needed to properly explore the design space. The ASGA algorithm proposed in this thesis goes in that direction but it could be extended in many ways. A dynamic selection of the optimal AS dimension at every generation is a possible line of study. On the other side, its mono-objective nature could be a limitation in an industrial context. More work has to be done to extend it for multi-objective optimization tasks, possibly exploiting a shared AS among all the functions to minimize.

For what concerns non-intrusive ROMs, we could investigate the potentialities of NARGP when more fidelities are available, maybe combining linear and nonlinear parameter space reduction. Furthermore NARGPAS can be exploited to design better sampling criteria for ROMs construction. When dealing with parametric time-dependent problem the application of DMD is not straightforward. Further studies to extend DMD to a full parametric setting are necessary, perhaps using POD and Grassmannian geodesics to interpolate the DMD operator between different parameters.

Regarding the applications side, even if the ones presented in this thesis are specific to a particular field such as naval engineering, thanks to the modularity of all the numerical pipelines introduced, they can be applied to different optimization tasks. The data-driven nature of the formulation allows to handle a vast and general class of problems in engineering. So, future lines of research can involve the integration of real-time data coming from sensors installed on the artifact to optimize. This has the potential to better tune the ROMs built through simulations, and adapt the digital twin to different operational scenarios.

Finally, more effort is needed to better couple parameter space and model order reduction in order to fight the curse of dimensionality for modern higher dimensional optimization problems.

BIBLIOGRAPHY

[1] I. H. Abbott and A. E. Von Doenhoff. *Theory of wing sections: including a summary of airfoil data*. Courier Corporation, 2012.

[2] E. P. Adorio and U Diliman. "MVF-multivariate test functions library in C for unconstrained global optimization." In: *Quezon City, Metro Manila, Philippines* (2005), pp. 100–104.

[3] I. P. Aguiar. "Dynamic Active Subspaces: a Data-Driven Approach to Computing Time-Dependent Active Subspaces in Dynamical Systems." MA thesis. University of Colorado Boulder, 2018.

[4] *Airfoil Tools. Airfoil database search*. http://www.airfoiltools.com/. 2020. (Visited on 01/2020).

[5] M. M. Ali, C. Khompatraporn, and Z. B. Zabinsky. "A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems." In: *Journal of global optimization* 31.4 (2005), pp. 635–672. DOI: 10.1007/s10898-004-9972-2.

[6] D. Amsallem and C. Farhat. "Interpolation method for adapting reduced-order models and application to aeroelasticity." In: *AIAA journal* 46.7 (2008), pp. 1803–1813. DOI: 10.2514/1.35374.

[7] D. N. Arnold. "An interior penalty finite element method with discontinuous elements." In: *SIAM journal on numerical analysis* 19.4 (1982), pp. 742–760. DOI: 10.1137/0719052.

[8] J. Arora and Q Wang. "Review of formulations for structural and mechanical system optimization." In: *Structural and Multidisciplinary Optimization* 30.4 (2005), pp. 251–272. DOI: 10.1007/s00158-004-0509-6.

[9] T. Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, 1996.

[10] D.-G. Baek, H.-S. Yoon, J.-H. Jung, K.-S. Kim, and B.-G. Paik. "Effects of the advance ratio on the evolution of a propeller wake." In: *Computers & Fluids* 118 (Sept. 2015), pp. 32–43. DOI: 10.1016/j.compfluid.2015.06.010.

[11] N. Baker, F. Alexander, T. Bremer, A. Hagberg, Y. Kevrekidis, H. Najm, M. Parashar, A. Patra, J. Sethian, S. Wild, et al. *Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence*. Tech. rep. USDOE Office of Science (SC), Washington, DC (United States), 2019. DOI: 10.2172/1478744.

[12] E. Balaras, S. Schroeder, and A. Posa. "Large-Eddy Simulations of Submarine Propellers." In: *Journal of Ship Research* 59.4 (Dec. 2015), pp. 227–237. DOI: 10.5957/josr.59.4.150047.

[13] M. Balesdent, N. Bérend, P. Dépincé, and A. Chriette. "A survey of multidisciplinary design optimization methods in launch vehicle design." In: *Structural and Multidisciplinary Optimization* 45.5 (2012), pp. 619–642. DOI: 10.1007/s00158-011-0701-4.

[14] F. Ballarin, A. D'Amario, S. Perotto, and G. Rozza. "A POD-Selective Inverse Distance Weighting method for fast parametrized shape morphing." In: *International Journal for Numerical Methods in Engineering* 117.8 (2018), pp. 860–884. DOI: 10.1002/nme.5982.

[15] E. Barshan, A. Ghodsi, Z. Azimifar, and M. Z. Jahromi. "Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds." In: *Pattern Recognition* 44.7 (2011), pp. 1357–1371. DOI: 10.1016/j.patcog.2010.12.015.

[16] K. Basu and A. B. Owen. "Transformations and Hardy–Krause Variation." In: *SIAM Journal on Numerical Analysis* 54.3 (2016), pp. 1946–1966. DOI: 10.1137/15M1052184.

[17] A. Beckert and H. Wendland. "Multivariate interpolation for fluid-structure-interaction problems using radial basis functions." In: *Aerospace Science and Technology* 5.2 (2001), pp. 125–134. DOI: 10.1016/S1270-9638(00)01087-7.

[18] J. Bect, D. Ginsbourger, L. Li, V. Picheny, and E. Vazquez. "Sequential design of computer experiments for the estimation of a probability of failure." In: *Statistics and Computing* 22.3 (2012), pp. 773–793. DOI: 10.1007/s11222-011-9241-4.

[19] C. J. Bélisle, H. E. Romeijn, and R. L. Smith. "Hit-and-run algorithms for generating multivariate distributions." In: *Mathematics of Operations Research* 18.2 (1993), pp. 255–266. DOI: 10.1287/moor.18.2.255.

[20] E. N. Ben-Ari and D. M. Steinberg. "Modeling data from computer experiments: an empirical comparison of kriging with MARS and projection pursuit regression." In: *Quality Engineering* 19.4 (2007), pp. 327–338. DOI: 10.1080/08982110701580930.

[21] P. Benner, S. Gugercin, and K. Willcox. "A survey of projection-based model reduction methods for parametric dynamical systems." In: *SIAM review* 57.4 (2015), pp. 483–531. DOI: 10.1137/130932715.

[22] G Berkooz, P Holmes, and J. L. Lumley. "The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows." In: *Annual Review of Fluid Mechanics* 25.1 (Jan. 1993), pp. 539–575. DOI: 10.1146/annurev.fl.25.010193.002543.

[23] A. Berlinet and C. Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[24] D. A. Bistrian and I. M. Navon. "An improved algorithm for the shallow water equations model reduction: Dynamic Mode Decomposition vs POD." In: *International Journal for Numerical Methods in Fluids* 78.9 (Apr. 2015), pp. 552–580. DOI: 10.1002/fld.4029.

[25] D. A. Bistrian and I. M. Navon. “Randomized dynamic mode decomposition for nonintrusive reduced order modelling.” In: *International Journal for Numerical Methods in Engineering* 112.1 (2017), pp. 3–25. DOI: 10.1002/nme.5499.

[26] A. Bobrowski. *Functional analysis for probability and stochastic processes: an introduction*. Cambridge University Press, 2005.

[27] G. Boncoraglio and C. Farhat. “Active manifold and model reduction for multidisciplinary analysis and optimization.” In: *AIAA Scitech 2021 Forum*. 2021, p. 1694. DOI: 10.2514/6.2021-1694.

[28] G. Boncoraglio, C. Farhat, and C. Bou-Mosleh. “Model Reduction Framework with a New Take on Active Subspaces for Optimization Problems with Linearized Fluid-Structure Interaction Constraints.” In: *International Journal for Numerical Methods in Engineering* (2020). DOI: 10.1002/nme.6376.

[29] L Bonfiglio, P Perdikaris, S Brizzolara, and G. Karniadakis. “Multi-fidelity optimization of super-cavitating hydrofoils.” In: *Computer Methods in Applied Mechanics and Engineering* 332 (2018), pp. 63–85. DOI: 10.1016/j.cma.2017.12.009.

[30] L. Bonfiglio, P. Perdikaris, G. Vernengo, J. S. de Medeiros, and G. Karniadakis. “Improving SWATH Seakeeping Performance using Multi-Fidelity Gaussian Process and Bayesian Optimization.” In: *Journal of Ship Research* 62.4 (2018), pp. 223–240. DOI: 10.5957/JOSR.11170069.

[31] F. M. Bos, B. W. van Oudheusden, and H. Bijl. “Radial basis function based mesh deformation applied to simulation of flow around flapping wings.” In: *Computers & Fluids* 79 (June 2013), pp. 167–177. DOI: 10.1016/j.compfluid.2013.02.004.

[32] N. Botkin and V. Turova-Botkina. “An algorithm for finding the Chebyshev center of a convex polyhedron.” In: *Applied Mathematics and Optimization* 29.2 (1994), pp. 211–222. DOI: 10.1007/BF01204183.

[33] E. Bouhoubeiny and P. Druault. “Note on the POD-based time interpolation from successive PIV images.” In: *Comptes Rendus Mécanique* 337.11-12 (Nov. 2009), pp. 776–780. DOI: 10.1016/j.crme.2009.10.003.

[34] K. S. Brentner and F. Farassat. “Modeling aerodynamically generated sound of helicopter rotors.” In: *Progress in Aerospace Sciences* 39.2-3 (Feb. 2003), pp. 83–120. DOI: 10.1016/s0376-0421(02)00068-4.

[35] R. A. Bridges, A. D. Gruber, C. R. Felder, M. Verma, and C. Hoff. “Active Manifolds: A non-linear analogue to Active Subspaces.” In: *Proceddings of the 36th International Conference on Machine Learning, ICML 2019*. Long Beach, California, USA, 2019, pp. 764–772.

[36] A. Broatch, J. García-Tíscar, F. Roig, and S. Sharma. “Dynamic mode decomposition of the acoustic field in radial compressors.” In: *Aerospace Science and Technology* 90 (2019), pp. 388–400. DOI: 10.1016/j.ast.2019.05.015.

[37] S. L. Brunton and J. N. Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

[38] S. L. Brunton, J. N. Kutz, K. Manohar, A. Y. Aravkin, K. Morgansen, J. Klemisch, N. Goebel, J. Buttrick, J. Poskin, A. W. Blom-Schieber, et al. "Data-driven aerospace engineering: reframing the industry with machine learning." In: *AIAA Journal* (2021), pp. 1–26. DOI: 10.2514/1.J060131.

[39] M. D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Vol. 12. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2003. DOI: 10.1017/CBO9780511543241.

[40] T Bui-Thanh, M. Damodaran, and K. Willcox. "Proper Orthogonal Decomposition Extensions for Parametric Applications in Compressible Aerodynamics." In: *21st AIAA Applied Aerodynamics Conference*. 2003, p. 4213. DOI: 10.2514/6.2003-4213.

[41] T. Bui-Thanh, M. Damodaran, and K. Willcox. "Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition." In: *AIAA journal* 42.8 (2004), pp. 1505–1516. DOI: 10.2514/1.2159.

[42] L. Buitinck et al. "API design for machine learning software: experiences from the scikit-learn project." In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 2013, pp. 108–122.

[43] E. F. Campana, D. Peri, Y. Tahara, and F. Stern. "Shape optimization in ship hydrodynamics using computational fluid dynamics." In: *Computer Methods in Applied Mechanics and Engineering* 196.1-3 (Dec. 2006), pp. 634–651. DOI: 10.1016/j.cma.2006.06.003.

[44] E. J. Candès, X. Li, Y. Ma, and J. Wright. "Robust principal component analysis?" In: *Journal of the ACM (JACM)* 58.3 (2011), pp. 1–37. DOI: 10.1145/1970392.1970395.

[45] C. Cartis and A. Otemissov. "A dimensionality reduction technique for unconstrained global optimization of functions with low effective dimensionality." In: *Information and Inference: A Journal of the IMA,* (2021). DOI: 10.1093/imaiai/iaab011.

[46] R. Chelouah and P. Siarry. "A continuous genetic algorithm designed for the global optimization of multimodal functions." In: *Journal of Heuristics* 6.2 (2000), pp. 191–213. DOI: 10.1023/A:1009626110229.

[47] F. Chinesta, E. Cueto, E. Abisset-Chavanne, J. L. Duval, and F. El Khaldi. "Virtual, digital and hybrid twins: a new paradigm in data-based engineering and engineered data." In: *Archives of computational methods in engineering* 27.1 (2020), pp. 105–134. DOI: 10.1007/s11831-018-9301-4.

[48] K. M. Choromanski, A. Pacchiano, J. Parker-Holder, Y. Tang, and V. Sindhwani. "From Complexity to Simplicity: Adaptive ES-Active Subspaces for Blackbox Optimization." In: *Advances in Neural Information Processing Systems*. 2019, pp. 10299–10309.

[49] M. Cianferra, V. Armenio, and S. Ianniello. "Hydroacoustic noise from different geometries." In: *International Journal Heat Fluid Flow* 70 (2018), pp. 348–362. DOI: 10.1016/j.ijheatfluidflow.2017.12.005.

[50] M. Cianferra, S. Ianniello, and V. Armenio. "Assessment of methodologies for the solution of the Ffowcs Williams and Hawkings equation using LES of incompressible single-phase flow around a finite-size square cylinder." In: *Journal of Sound and Vibration* 453 (2019), pp. 1–24. DOI: 10.1016/j.jsv.2019.04.001.

[51] M. Cianferra, A. Petronio, and V. Armenio. "Non-linear noise from a ship propeller in open sea condition." In: *Ocean Engineering* 191 (Nov. 2019), p. 106474. DOI: 10.1016/j.oceaneng.2019.106474.

[52] M. Cianferra, A. Petronio, and V. Armenio. "Hydrodynamic Noise from a Propeller in Open Sea Condition." In: *Technology and Science for the Ships of the Future: Proceedings of NAV 2018: 19th International Conference on Ship & Maritime Research*. IOS Press, 2018, pp. 149–156. DOI: 10.3233/978-1-61499-870-9-149.

[53] C. Cintolesi, A. Petronio, and V. Armenio. "Large eddy simulation of turbulent buoyant flow in a confined cavity with conjugate heat transfer." In: *Phys fluids* 27 (2015). DOI: 10.1063/1.4930815.

[54] P. G. Constantine. *Active subspaces: Emerging ideas for dimension reduction in parameter studies*. Vol. 2. SIAM Spotlights. SIAM, 2015. DOI: 10.1137/1.9781611973860.

[55] P. G. Constantine and P. Diaz. "Global sensitivity metrics from active subspaces." In: *Reliability Engineering & System Safety* 162 (2017), pp. 1–13. DOI: 10.1016/j.ress.2017.01.013.

[56] P. G. Constantine and A. Doostan. "Time-dependent global sensitivity analysis with active subspaces for a lithium ion battery model." In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 10.5 (2017), pp. 243–262. DOI: 10.1002/sam.11347.

[57] P. G. Constantine, E. Dow, and Q. Wang. "Active subspace methods in theory and practice: applications to kriging surfaces." In: *SIAM Journal on Scientific Computing* 36.4 (2014), A1500–A1524. DOI: 10.1137/130916138.

[58] P. G. Constantine, M. Emory, J. Larsson, and G. Iaccarino. "Exploiting active subspaces to quantify uncertainty in the numerical simulation of the HyShot II scramjet." In: *Journal of Computational Physics* 302 (2015), pp. 1–20. DOI: 10.1016/j.jcp.2015.09.001.

[59] R. D. Cook and L. Ni. "Sufficient dimension reduction via inverse regression: A minimum discrepancy approach." In: *Journal of the American Statistical Association* 100.470 (2005), pp. 410–428. DOI: 10.1198/016214504000001501.

[60] C. Cui, K. Zhang, T. Daulbaev, J. Gusak, I. Oseledets, and Z. Zhang. "Active Subspace of Neural Networks: Structural Analysis and Universal Attacks." In: *SIAM Journal on Mathematics of Data Science* 2.4 (2020), pp. 1096–1122. DOI: 10.1137/19M1296070.

[61] J. P. Cunningham and Z. Ghahramani. "Linear dimensionality reduction: Survey, insights, and generalizations." In: *The Journal of Machine Learning Research* 16.1 (2015), pp. 2859–2900.

[62] A. Damianou and N. Lawrence. "Deep Gaussian processes." In: *Artificial Intelligence and Statistics*. 2013, pp. 207–215.

[63] T. Daniel, F. Casenave, N. Akkari, and D. Ryckelynck. "Model order reduction assisted by deep neural networks (ROM-net)." In: *Advanced Modeling and Simulation in Engineering Sciences* 7.1 (2020), pp. 1–27. DOI: 10.1186/s40323-020-00153-6.

[64] E. Darve. "The Fast Multipole Method: Numerical Implementation." In: *Journal of Computational Physics* 160.1 (2000), pp. 195–240. DOI: 10.1006/jcph.2000.6451.

[65] N. Demo, G. Ortali, G. Gustin, G. Rozza, and G. Lavini. "An efficient computational framework for naval shape design and optimization problems by means of data-driven reduced order modeling techniques." In: *Bollettino dell'Unione Matematica Italiana* (2020). ISSN: 2198-2759. DOI: 10.1007/s40574-020-00263-4.

[66] N. Demo, M. Tezzele, G. Gustin, G. Lavini, and G. Rozza. "Shape optimization by means of proper orthogonal decomposition and dynamic mode decomposition." In: *Technology and Science for the Ships of the Future: Proceedings of NAV 2018: 19th International Conference on Ship & Maritime Research*. IOS Press, 2018, pp. 212–219. DOI: 10.3233/978-1-61499-870-9-212.

[67] N. Demo, M. Tezzele, A. Mola, and G. Rozza. "An efficient shape parametrisation by free-form deformation enhanced by active subspace for hull hydrodynamic ship design problems in open source environment." In: *Proceedings of ISOPE 2018: The 28th International Ocean and Polar Engineering Conference*. Vol. 3. 2018, pp. 565–572.

[68] N. Demo, M. Tezzele, A. Mola, and G. Rozza. "A complete data-driven framework for the efficient solution of parametric shape design and optimisation in naval engineering problems." In: *Proceedings of MARINE 2019: VIII International Conference on Computational Methods in Marine Engineering*. Ed. by R. Bensow and J. Ringsberg. 2019, pp. 111–121.

[69] N. Demo, M. Tezzele, A. Mola, and G. Rozza. "Hull Shape Design Optimization with Parameter Space and Model Reductions, and Self-Learning Mesh Morphing." In: *Journal of Marine Science and Engineering* 9.2 (2021), p. 185. DOI: 10.3390/jmse9020185.

[70] N. Demo, M. Tezzele, and G. Rozza. "EZyRB: Easy Reduced Basis method." In: *The Journal of Open Source Software* 3.24 (2018), p. 661. DOI: 10.21105/joss.00661.

[71] N. Demo, M. Tezzele, and G. Rozza. "PyDMD: Python Dynamic Mode Decomposition." In: *The Journal of Open Source Software* 3.22 (2018), p. 530. DOI: 10.21105/joss.00530.

[72] N. Demo, M. Tezzele, and G. Rozza. "A non-intrusive approach for reconstruction of POD modal coefficients through active subspaces." In: *Comptes Rendus Mécanique de l'Académie des Sciences, DataBEST 2019 Special Issue* 347.11 (2019), pp. 873–881. DOI: 10.1016/j.crme.2019.11.012.

[73] N. Demo, M. Tezzele, and G. Rozza. "A Supervised Learning Approach Involving Active Subspaces for an Efficient Genetic Algorithm in High-Dimensional Optimization Problems." In: *SIAM Journal on Scientific Computing* 43.3 (2021), B831–B853. DOI: 10.1137/20M1345219.

[74] P. Diaz, P. Constantine, K. Kalmbach, E. Jones, and S. Pankavich. "A modified SEIR model for the spread of Ebola in Western Africa and metrics for resource allocation." In: *Applied Mathematics and Computation* 324 (2018), pp. 141–155. DOI: 10.1016/j.amc.2017.11.039.

[75] M. Diez, E. F. Campana, and F. Stern. "Design-space dimensionality reduction in shape optimization by Karhunen–Loève expansion." In: *Computer Methods in Applied Mechanics and Engineering* 283 (2015), pp. 1525–1544. DOI: 10.1016/j.cma.2014.10.042.

[76] L. C. W. Dixon and G. P. Szego. "The Global Optimization Problem: An Introduction." In: *Towards Global Optimisation*. North-Holland Pub. Co, Amsterdam, 1978, pp. 1–15.

[77] V. Dolci and R. Arina. "Proper Orthogonal Decomposition as Surrogate Model for Aerodynamic Optimization." In: *International Journal of Aerospace Engineering* (2016). DOI: 10.1155/2016/8092824.

[78] Z. Drezner. "A New Genetic Algorithm for the Quadratic Assignment Problem." In: *INFORMS Journal on Computing* 15.3 (2003), pp. 320–330. DOI: 10.1287/ijoc.15.3.320.16076.

[79] P. Druault and C. Chaillou. "Use of Proper Orthogonal Decomposition for reconstructing the 3D in-cylinder mean-flow field from PIV data." In: *Comptes Rendus Mécanique* 335.1 (Jan. 2007), pp. 42–47. DOI: 10.1016/j.crme.2006.11.004.

[80] T. A. El-Mihoub, A. A. Hopgood, L. Nolle, and A. Battersby. "Hybrid Genetic Algorithms: A Review." In: *Engineering Letters* 13.2 (2006), pp. 124–137.

[81] S. M. Elsayed, R. A. Sarker, and D. L. Essam. "A new genetic algorithm for solving optimization problems." In: *Engineering Applications of Artificial Intelligence* 27 (2014), pp. 57–69. DOI: 10.1016/j.engappai.2013.09.013.

[82] C. Emmeche. *The garden in the machine: the emerging science of artificial life*. Vol. 17. Princeton University Press, 1996.

[83] N. B. Erichson, S. L. Brunton, and J. N. Kutz. "Compressed dynamic mode decomposition for background modeling." In: *Journal of Real-Time Image Processing* 16.5 (2019), pp. 1479–1492. DOI: 10.1007/s11554-016-0655-2.

[84] N. B. Erichson, L. Mathelin, J. N. Kutz, and S. L. Brunton. "Randomized dynamic mode decomposition." In: *SIAM Journal on Applied Dynamical Systems* 18.4 (2019), pp. 1867–1891. DOI: 10.1137/18M1215013.

[85] L. J. Eshelman and J. D. Schaffer. "Real-coded genetic algorithms and interval-schemata." In: *Foundations of genetic algorithms*. Vol. 2. Elsevier, 1993, pp. 187–202. DOI: 10.1016/B978-0-08-094832-4.50018-0.

[86] J. E. Ffowcs Williams and D. L. Hawkings. "Sound generation by turbulence and surfaces in arbitrary motion." In: *Philosophical Transaction of Royal Society* 264 (1969), pp. 321–342. DOI: 10.1098/rsta.1969.0031.

[87] A. I. Forrester, A. Sóbester, and A. J. Keane. "Multi-fidelity optimization via surrogate modelling." In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 463.2088 (2007), pp. 3251–3269. DOI: 10.1098/rspa.2007.1900.

[88] M. Fossati and W. G. Habashi. "Multiparameter Analysis of Aero-Icing Problems Using Proper Orthogonal Decomposition and Multidimensional Interpolation." In: *AIAA Journal* 51.4 (Apr. 2013), pp. 946–960. DOI: 10.2514/1.j051877.

[89] T. Franz, R. Zimmermann, S. Görtz, and N. Karcher. "Interpolation-based reduced-order modelling for steady transonic flows via manifold learning." In: *International Journal of Computational Fluid Dynamics* 28.3-4 (Mar. 2014), pp. 106–121. DOI: 10.1080/10618562.2014.918695.

[90] P. I. Frazier. "A tutorial on Bayesian optimization." In: *arXiv preprint arXiv:1807.02811* (2018).

[91] GPy. *GPy: A Gaussian process framework in Python*. http://github.com/SheffieldML/GPy. since 2012.

[92] M. Gadalla, M. Cianferra, M. Tezzele, G. Stabile, A. Mola, and G. Rozza. "On the comparison of LES data-driven reduced order approaches for hydroacoustic analysis." In: *Computers & Fluids* 216 (2021), p. 104819. ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2020.104819.

[93] M. Gadalla, M. Tezzele, A. Mola, and G. Rozza. "BladeX: Python Blade Morphing." In: *The Journal of Open Source Software* 4.34 (2019), p. 1203. DOI: 10.21105/joss.01203.

[94] S. Gaggero, G. Vernengo, D. Villa, and L. Bonfiglio. "A reduced order approach for optimal design of efficient marine propellers." In: *Ships and Offshore Structures* 15.2 (2020), pp. 200–214. DOI: 10.1080/17445302.2019.1606877.

[95] F. Garotta, N. Demo, M. Tezzele, M. Carraturo, A. Reali, and G. Rozza. "Reduced Order Isogeometric Analysis Approach for PDEs in Parametrized Domains." In: *Quantification of Uncertainty: Improving Efficiency and Technology: QUIET selected contributions*. Ed. by M. D'Elia, M. Gunzburger, and G. Rozza. Vol. 137. Lecture Notes in Computational Science and Engineering. Cham: Springer International Publishing, 2020, pp. 153–170. ISBN: 978-3-030-48721-8. DOI: 10.1007/978-3-030-48721-8_7.

[96] J. Gazdag. "Time-differencing schemes and transform methods." In: *Journal of Computational Physics* 20.2 (1976), pp. 196–207. DOI: 10.1016/0021-9991(76)90064-4.

[97] S. Georgaka, G. Stabile, G. Rozza, and M. J. Bluck. "Parametric POD-Galerkin Model Order Reduction for Unsteady-State Heat Transfer Problems." In: *Communications in Computational Physics* 27.1 (June 2020), pp. 1–32. DOI: 10.4208/cicp.oa-2018-0207.

[98] S. Georgaka, G. Stabile, K. Star, G. Rozza, and M. J. Bluck. "A hybrid reduced order method for modelling turbulent heat transfer problems." In: *Computers & Fluids* 208 (2020), p. 104615. ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2020.104615.

[99] S. F. Ghoreishi, S. Friedman, and D. L. Allaire. "Adaptive Dimensionality Reduction for Fast Sequential Optimization With Gaussian Processes." In: *Journal of Mechanical Design* 141.7 (2019), p. 071404. DOI: 10.1115/1.4043202.

[100] M. Girfoglio, A. Quaini, and G. Rozza. "A POD-Galerkin reduced order model for a LES filtering approach." In: *Journal of Computational Physics* 436 (2021), p. 110260. DOI: 10.1016/j.jcp.2021.110260.

[101] S. A. Glegg and W. J. Devenport. "Proper orthogonal decomposition of turbulent flows for aeroacoustic and hydroacoustic applications." In: *Journal of Sound and Vibration* 239.4 (2001), pp. 767–784. DOI: 10.1006/jsvi.2000.3128.

[102] X. Gloerfelt, F. Pérot, C. Bailly, and D. Juvé. "Flow-induced cylinder noise formulated as a diffraction problem for low Mach numbers." In: *Journal of Sound and Vibration* 287.1-2 (Oct. 2005), pp. 129–151. DOI: 10.1016/j.jsv.2004.10.047.

[103] *Gmesh, a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities.* http://gmsh.info/.

[104] G. H. Golub and C. F. Van Loan. *Matrix computations*. Vol. 3. Johns Hopkins University Press, 2013. ISBN: 9781421407944.

[105] Y. L. Guennec, J.-P. Brunet, F.-Z. Daim, M. Chau, and Y. Tourbier. "A parametric and non-intrusive reduced order model of car crash simulation." In: *Computer Methods in Applied Mechanics and Engineering* 338 (Aug. 2018), pp. 186–207. DOI: 10.1016/j.cma.2018.03.005.

[106] M. Guo and J. S. Hesthaven. "Reduced order modeling for nonlinear structural analysis using Gaussian process regression." In: *Computer Methods in Applied Mechanics and Engineering* 341 (2018), pp. 807–826. DOI: 10.1016/j.cma.2018.07.017.

[107] G. Haller. "An objective definition of a vortex." In: *Journal of Fluid Mechanics* 525 (Feb. 2005), pp. 1–26. DOI: 10.1017/s0022112004002526. URL: https://doi.org/10.1017/s0022112004002526.

[108] J. Han, M. Kamber, and J. Pei. "Data Mining: Concepts and Techniques." In: *The Morgan Kaufmann Series in Data Management Systems* 5.4 (2012), pp. 83–124. DOI: 10.1016/C2009-0-61819-5.

[109] T. Hastie and R. Tibshirani. "Discriminant adaptive nearest neighbor classification." In: *IEEE transactions on pattern analysis and machine intelligence* 18.6 (1996), pp. 607–616. DOI: 10.1109/34.506411.

[110] P. Héas, C. Herzet, and B. Combes. "Generalized Kernel-Based Dynamic Mode Decomposition." In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 3877–3881. DOI: 10.1109/ICASSP40776.2020.9054594.

[111] M. Heder and A. Ulfvarson. “Hull beam behaviour of passenger ships.” In: *Marine structures* 4.1 (1991), pp. 17–34. DOI: 10.1016/0951-8339(91)90022-4.

[112] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. 1st ed. Springer Briefs in Mathematics. Switzerland: Springer, 2015, p. 135. ISBN: 978-3-319-22469-5. DOI: 10.1007/978-3-319-22470-1.

[113] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.

[114] R. M. Hicks and P. A. Henne. “Wing Design by Numerical Optimization.” In: *Journal of Aircraft* 15.7 (1978), pp. 407–412. DOI: 10.2514/3.58379.

[115] S. Hijazi, S. Ali, G. Stabile, F. Ballarin, and G. Rozza. “The Effort of Increasing Reynolds Number in Projection-Based Reduced Order Methods: from Laminar to Turbulent Flows.” In: *Lecture Notes in Computational Science and Engineering*. Cham: Springer International Publishing, 2020, pp. 245–264. ISBN: 978-3-030-30705-9. DOI: 10.1007/978-3-030-30705-9_22.

[116] S. Hijazi, G. Stabile, A. Mola, and G. Rozza. “Data-Driven POD–Galerkin reduced order model for turbulent flows.” In: *Journal of Computational Physics* 416 (2020), p. 109513. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2020.109513.

[117] R. Hinterding. “Gaussian mutation and self-adaption for numeric genetic algorithms.” In: *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*. Vol. 1. IEEE. 1995, p. 384. DOI: 10.1109/ICEC.1995.489178.

[118] C. Hirt and B. Nichols. “Volume of fluid (VOF) method for the dynamics of free boundaries.” In: *J. Comput. Phys.* 39 (Jan. 1981), pp. 201–225. DOI: 10.1016/0021-9991(81)90145-5.

[119] J. H. Holland. “Genetic Algorithms and the Optimal Allocation of Trials.” In: *SIAM Journal on Computing* 2.2 (1973), pp. 88–105. DOI: 10.1137/0202009.

[120] J. H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[121] *HopeFOAM extension of OpenFOAM*. https://github.com/HopeFOAM/HopeFOAM. Last release 15th September 2017.

[122] S. Ianniello, R. Muscari, and A. D. Mascio. “Ship underwater noise assessment by the acoustic analogy. Part I: nonlinear analysis of a marine propeller in a uniform flow.” In: *Journal of Marine Science and Technology* 18.4 (July 2013), pp. 547–570. DOI: 10.1007/s00773-013-0227-0.

[123] S. Ianniello, R. Muscari, and A. D. Mascio. “Ship underwater noise assessment by the acoustic analogy. Part II: hydroacoustic analysis of a ship scaled model.” In: *Journal of Marine Science and Technology* 18(4) (2013), pp. 547–570. DOI: 10.1007/s00773-013-0236-z.

[124] R. Issa. “Solution of the implicitly discretised fluid flow equations by operator-splitting.” In: *Journal of Computational Physics* 62.1 (Jan. 1986), pp. 40–65. DOI: 10.1016/0021-9991(86)90099-9.

[125] E. Iuliano and D. Quagliarella. "Proper Orthogonal Decomposition, surrogate modelling and evolutionary optimization in aerodynamic design." In: *Computers & Fluids* 84 (Sept. 2013), pp. 327–350. DOI: 10.1016/j.compfluid.2013.06.007.

[126] E. N. Jacobs, K. E. Ward, and R. M. Pinkerton. *The Characteristics of 78 Related Airfoil Sections from Tests in the Variable-Density Wind Tunnel*. Tech. rep. 430. N.A.C.A., 1933.

[127] S. Jakobsson and O. Amoignon. "Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization." In: *Computers & Fluids* 36.6 (July 2007), pp. 1119–1136. DOI: 10.1016/j.compfluid.2006.11.002.

[128] M. Jamil and X.-S. Yang. "A literature survey of benchmark functions for global optimisation problems." In: *International Journal of Mathematical Modelling and Numerical Optimisation* 4.2 (2013), pp. 150–194. DOI: 10.1504/IJMMNO.2013.055204.

[129] J. L. Jefferson, J. M. Gilbert, P. G. Constantine, and R. M. Maxwell. "Active subspaces for sensitivity analysis and dimension reduction of an integrated hydrologic model." In: *Computers & Geosciences* 83 (2015), pp. 127–138. DOI: 10.1016/j.cageo.2015.07.001.

[130] K.-L. Jeong and S.-M. Jeong. "A Mesh Deformation Method for CFD-Based Hull form Optimization." In: *Journal of Marine Science and Engineering* 8.6 (2020), p. 473. DOI: 10.3390/jmse8060473.

[131] W. Ji, J. Wang, O. Zahm, Y. M. Marzouk, B. Yang, Z. Ren, and C. K. Law. "Shared low-dimensional subspaces for propagating kinetic uncertainty to multiple outputs." In: *Combustion and Flame* 190 (2018), pp. 146–157. DOI: 10.1016/j.combustflame.2017.11.021.

[132] D. R. Jones, M. Schonlau, and W. J. Welch. "Efficient Global Optimization of Expensive Black-Box Functions." In: *Journal of Global optimization* 13.4 (1998), pp. 455–492. DOI: 10.1023/A:1008306431147.

[133] G. Jourdain, L.-E. Eriksson, S. H. Kim, and C. H. Sohn. "Application of dynamic mode decomposition to acoustic-modes identification and damping in a 3-dimensional chamber with baffled injectors." In: *Journal of Sound and Vibration* 332.18 (Sept. 2013), pp. 4308–4323. DOI: 10.1016/j.jsv.2013.02.041.

[134] M. R. Jovanović, P. J. Schmid, and J. W. Nichols. "Sparsity-promoting dynamic mode decomposition." In: *Physics of Fluids* 26.2 (2014), p. 024103. DOI: 10.1063/1.4863670.

[135] M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur. "Gaussian processes and kernel methods: A review on connections and equivalences." In: *arXiv preprint arXiv:1807.02582* (2018).

[136] M. G. Kapteyn, D. J. Knezevic, D. Huynh, M. Tran, and K. E. Willcox. "Data-driven physics-based digital twins via a library of component-based reduced-order models." In: *International Journal for Numerical Methods in Engineering* (2020). DOI: 10.1002/nme.6423.

[137] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Vol. 344. Wiley Series in Probability and Statistics. John Wiley & Sons, 2005. ISBN: 978-0-471-73578-6. DOI: 10.1002/9780470316801.

[138] M. C. Kennedy and A. O'Hagan. "Predicting the output from a complex computer code when fast approximations are available." In: *Biometrika* 87.1 (2000), pp. 1–13.

[139] J. E. Kerwin. "Marine Propellers." In: *Annual Review of Fluid Mechanics* 18.1 (Jan. 1986), pp. 367–403. DOI: 10.1146/annurev.fl.18.010186.002055.

[140] I. Kevrekidis, C. Rowley, and M. Williams. "A kernel-based method for data-driven Koopman spectral analysis." In: *Journal of Computational Dynamics* 2.2 (2015), pp. 247–265. DOI: 10.3934/jcd.2015005.

[141] S. Khan and E. Gunpinar. "Sampling CAD models via an extended teaching–learning-based optimization technique." In: *Computer-Aided Design* 100 (2018), pp. 52–67. DOI: 10.1016/j.cad.2018.03.003.

[142] S. Khan, A. Serani, M. Diez, and P. Kaklis. "Physics-informed feature-to-feature learning for design-space dimensionality reduction in shape optimisation." In: *AIAA Scitech 2021 Forum*. 2021, p. 1235. DOI: 10.2514/6.2021-1235.

[143] B. Kramer, A. N. Marques, B. Peherstorfer, U. Villa, and K. Willcox. "Multifidelity probability estimation via fusion of estimators." In: *Journal of Computational Physics* 392 (2019), pp. 385–402. DOI: 10.1016/j.jcp.2019.04.071.

[144] M. Kumar, M. Husain, N. Upreti, and D. Gupta. "Genetic algorithm: Review and application." In: *Available at SSRN 3529843* (2010).

[145] P. Kumar and K. Mahesh. "Large eddy simulation of propeller wake instabilities." In: *Journal of Fluid Mechanics* 814 (Feb. 2017), pp. 361–396. DOI: 10.1017/jfm.2017.20.

[146] P. K. Kundu, I. M. Cohen, and D. R. Dowling, eds. *Fluid Mechanics (Fifth Edition)*. Boston: Academic Press, 2012. ISBN: 978-0-12-382100-3. DOI: 10.1016/C2009-0-63410-3.

[147] K. Kunisch and S. Volkwein. "Proper orthogonal decomposition for optimality systems." In: *ESAIM: Mathematical Modelling and Numerical Analysis* 42.1 (Jan. 2008), pp. 1–23. DOI: 10.1051/m2an:2007054.

[148] H. J. Kushner. "A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise." In: *Journal of Basic Engineering* 86.1 (1964), pp. 97–106. DOI: 10.1115/1.3653121.

[149] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM, 2016. DOI: 10.1137/1.9781611974508.

[150] J. N. Kutz, X. Fu, and S. L. Brunton. "Multiresolution dynamic mode decomposition." In: *SIAM Journal on Applied Dynamical Systems* 15.2 (2016), pp. 713–735. DOI: 10.1137/15M1023543.

[151] M. Laguna and R. Martí. “Experimental testing of advanced scatter search designs for global optimization of multimodal functions.” In: *Journal of Global Optimization* 33.2 (2005), pp. 235–255. DOI: 10.1007/s10898-004-1936-z.

[152] R. R. Lam, O. Zahm, Y. M. Marzouk, and K. E. Willcox. “Multifidelity Dimension Reduction via Active Subspaces.” In: *SIAM Journal on Scientific Computing* 42.2 (2020), A929–A956. DOI: 10.1137/18M1214123.

[153] T Lassila and G Rozza. “Parametric free-form shape design with PDE models and reduced basis method.” In: *Computer Methods in Applied Mechanics and Engineering* 199.23–24 (2010), pp. 1583–1592. DOI: 10.1016/j.cma.2010.01.007.

[154] M. Lázaro-Gredilla, J. Quiñonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal. “Sparse spectrum Gaussian process regression.” In: *The Journal of Machine Learning Research* 11 (2010), pp. 1865–1881.

[155] M. Lázaro-Gredilla and M. K. Titsias. “Variational Heteroscedastic Gaussian Process regression.” In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*. ICML’11. Bellevue, Washington, USA: Omnipress, 2011, pp. 841–848. ISBN: 9781450306195.

[156] S. Le Clainche, D. Rodríguez, V. Theofilis, and J. Soria. “Flow around a hemisphere-cylinder at high angle of attack and low Reynolds number. Part II: POD and DMD applied to reduced domains.” In: *Aerospace Science and Technology* 44 (2015), pp. 88–100. DOI: 10.1016/j.ast.2014.10.009.

[157] S. Le Clainche and J. M. Vega. “Higher order dynamic mode decomposition.” In: *SIAM Journal on Applied Dynamical Systems* 16.2 (2017), pp. 882–925. DOI: 10.1137/15M1054924.

[158] L. Le Gratiet and J. Garnier. “Recursive co-kriging model for design of computer experiments with multiple levels of fidelity.” In: *International Journal for Uncertainty Quantification* 4.5 (2014), pp. 365–386. DOI: 10.1615/Int.J.UncertaintyQuantification.2014006914.

[159] K. Lee and K. T. Carlberg. “Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders.” In: *Journal of Computational Physics* 404 (2020), p. 108973. DOI: 10.1016/j.jcp.2019.108973.

[160] M. R. Lee. “Modified Active Subspaces Using the Average of Gradients.” In: *SIAM/ASA Journal on Uncertainty Quantification* 7.1 (2019), pp. 53–66. DOI: 10.1137/17M1140662.

[161] K.-C. Li. “Sliced inverse regression for dimension reduction.” In: *Journal of the American Statistical Association* 86.414 (1991), pp. 316–327. DOI: 10.2307/2290563.

[162] L. Li. “Sparse sufficient dimension reduction.” In: *Biometrika* 94.3 (2007), pp. 603–613. DOI: 10.1093/biomet/asm044.

[163] Z Li, J.-F. Ton, D Oglic, and D Sejdinovic. "Towards a unified analysis of random Fourier features." In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research 2019. ICML, 2019, pp. 3905–3914.

[164] H. Liu, Y.-S. Ong, X. Shen, and J. Cai. "When Gaussian process meets big data: A review of scalable GPs." In: *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[165] T. Loudon and S. Pankavich. "Mathematical analysis and dynamic active subspaces for a long term model of HIV." In: *Mathematical Biosciences & Engineering* 14.3 (2017), p. 709. DOI: 10.3934/mbe.2017040.

[166] L. Lovász and S. Vempala. "Hit-and-run from a corner." In: *SIAM Journal on Computing* 35.4 (2006), p. 985. DOI: 10.1137/S009753970544727X.

[167] A. E. Løvgren, Y. Maday, and E. M. Rønquist. "A reduced basis element method for the steady Stokes problem." In: *ESAIM: Mathematical Modelling and Numerical Analysis* 40.3 (2006), pp. 529–552. DOI: 10.1051/m2an:2006021.

[168] T. W. Lukaczyk, P. Constantine, F. Palacios, and J. J. Alonso. "Active subspaces for shape optimization." In: *10th AIAA multidisciplinary design optimization conference*. 2014, p. 1171. DOI: 10.2514/6.2014-1171.

[169] T. W. Lukaczyk. "Surrogate Modeling and Active Subspaces for Efficient Optimization of Supersonic Aircraft." PhD thesis. Stanford University, 2015.

[170] H. V. Ly and H. T. Tran. "Modeling and control of physical processes using proper orthogonal decomposition." In: *Mathematical and Computer Modelling* 33.1-3 (Jan. 2001), pp. 223–236. DOI: 10.1016/s0895-7177(00)00240-5.

[171] M. Mancinelli, T. Pagliaroli, R. Camussi, and T. Castelain. "On the hydrodynamic and acoustic nature of pressure proper orthogonal decomposition modes in the near field of a compressible jet." In: *Journal of Fluid Mechanics* 836 (Dec. 2017), pp. 998–1008. DOI: 10.1017/jfm.2017.839.

[172] A. Manzoni, A. Quarteroni, and G. Rozza. "Model reduction techniques for fast blood flow simulation in parametrized geometries." In: *International journal for numerical methods in biomedical engineering* 28.6-7 (2012), pp. 604–625. DOI: 10.1002/cnm.1465.

[173] F. E. Maranzana. "On the Location of Supply Points to Minimize Transport Costs." In: *Journal of the Operational Research Society* 15.3 (1964), pp. 261–270. DOI: 10.1057/jors.1964.47.

[174] A. D. Mascio, R. Muscari, and G. Dubbioso. "On the wake dynamics of a propeller operating in drift." In: *Journal of Fluid Mechanics* 754 (July 2014), pp. 263–307. DOI: 10.1017/jfm.2014.390.

[175] C. Meneveau, T. Lund, and W. Cabot. "A Lagrangian dynamic subgrid-scale model of turbulence." In: *Journal of Fluid Mechanics* 319 (1996), pp. 353–385. DOI: 10.1017/S0022112096007379.

[176] X. Meng and G. E. Karniadakis. “A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems.” In: *Journal of Computational Physics* 401 (2020), p. 109020. DOI: 10.1016/j.jcp.2019.109020.

[177] F. Menter. “Zonal Two Equation k-w Turbulence Models For Aerodynamic Flows.” In: *23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference*. 1993, p. 2906. DOI: 10.2514/6.1993-2906.

[178] K. E. Meyer, J. M. Pedersen, and O. Özcan. “A turbulent jet in crossflow analysed with proper orthogonal decomposition.” In: *Journal of Fluid Mechanics* 583 (2007), pp. 199–227. DOI: 10.1017/s0022112007006143.

[179] M. J. Mifsud, D. G. MacManus, and S. Shaw. “A variable-fidelity aerodynamic model using proper orthogonal decomposition.” In: *International Journal for Numerical Methods in Fluids* 82.10 (Apr. 2016), pp. 646–663. DOI: 10.1002/fld.4234.

[180] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. “Kernel PCA and de-noising in feature spaces.” In: *Advances in neural information processing systems*. 1999, pp. 536–542.

[181] J. Močkus. “On Bayesian methods for seeking the extremum.” In: *Optimization techniques IFIP technical conference*. Springer. 1975, pp. 400–404.

[182] O. e. Moctar, V. Shigunov, and T. Zorn. “Duisburg Test Case: Post-Panamax Container Ship for Benchmarking.” In: *Ship Technology Research* 59.3 (2012), pp. 50–64. DOI: 10.1179/str.2012.59.3.004.

[183] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[184] A. Mola, M. Tezzele, M. Gadalla, F. Valdenazzi, D. Grassi, R. Padovan, and G. Rozza. “Efficient reduction in shape parameter space dimension for ship propeller blade design.” In: *Proceedings of MARINE 2019: VIII International Conference on Computational Methods in Marine Engineering*. Ed. by R. Bensow and J. Ringsberg. 2019, pp. 201–212.

[185] A. Morris, C. Allen, and T. Rendall. “CFD-based optimization of aerofoils using radial basis functions for domain element parameterization and mesh deformation.” In: *International Journal for Numerical Methods in Fluids* 58.8 (2008), pp. 827–860. DOI: 10.1002/fld.1769.

[186] H. Mühlenbein, M Schomisch, and J. Born. “The parallel genetic algorithm as function optimizer.” In: *Parallel computing* 17.6-7 (1991), pp. 619–632. DOI: 10.1016/S0167-8191(05)80052-3.

[187] T. W. Muld, G. Efraimsson, and D. S. Henningson. “Flow structures around a high-speed train extracted using Proper Orthogonal Decomposition and Dynamic Mode Decomposition.” In: *Computers & Fluids* 57 (2012), pp. 87–97. DOI: 10.1016/j.compfluid.2011.12.012.

[188] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012. ISBN: 9780262018029.

[189] A. Najafi-Yazdi, G.A.Bres, and L. Mongeau. “An acoustic analogy formulation for moving sources in uniformly moving media.” In: *Proceedings of the Royal Society A* 467.2125 (2011), pp. 144–165. DOI: 10.1098/rspa.2010.0172.

[190] Y. Nesterov and V. Spokoiny. “Random Gradient-Free Minimization of Convex Functions.” In: *Foundations of Computational Mathematics* 17.2 (2017), pp. 527–566. DOI: 10.1007/s10208-015-9296-2.

[191] N. C. Nguyen, J. Peraire, and B. Cockburn. “An implicit high-order hybridizable discontinuous Galerkin method for linear convection–diffusion equations.” In: *Journal of Computational Physics* 228.9 (2009), pp. 3232–3254. DOI: 10.1016/j.jcp.2009.01.030.

[192] Z. Nitzkorski and K. Mahesh. “A dynamic end cap technique for sound computation using the Ffowcs Williams and Hawkings equations.” In: *Physics of Fluids* 26.11 (Nov. 2014), p. 115101. DOI: 10.1063/1.4900876.

[193] G. Ortali, N. Demo, and G. Rozza. “Gaussian process approach within a data-driven POD framework for fluid dynamics engineering problems.” In: *arXiv preprint arXiv:2012.01989* (Submitted, 2020).

[194] M. Palaci-Olgun. “Gaussian Process Modeling and Supervised Dimensionality Reduction Algorithms via Stiefel Manifold Learning.” MA thesis. University of Toronto Institute of Aerospace Studies, 2018.

[195] A. Paleyes, M. Pullin, M. Mahsereci, N. Lawrence, and J. González. “Emulation of physical processes with Emukit.” In: *Second Workshop on Machine Learning and the Physical Sciences, NeurIPS*. 2019.

[196] M. F. de Pando, P. J. Schmid, and D. Sipp. “A global analysis of tonal noise in flows around aerofoils.” In: *Journal of Fluid Mechanics* 754 (July 2014), pp. 5–38. DOI: 10.1017/jfm.2014.356.

[197] M. T. Parente, J. Wallin, B. Wohlmuth, et al. “Generalized bounds for active subspaces.” In: *Electronic Journal of Statistics* 14.1 (2020), pp. 917–943. DOI: 10.1214/20-EJS1684.

[198] H.-S. Park and C.-H. Jun. “A simple and fast algorithm for K-medoids clustering.” In: *Expert Systems with Applications* 36.2 (2009), pp. 3336–3341. DOI: 10.1016/j.eswa.2008.01.039.

[199] O. Parmasto et al. “Mechanics of the passenger ship structure with non-longitudinal-load-carrying accomodation decks.” MA thesis. Aalto University, 2012.

[200] S. Patankar and D. Spalding. “A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows.” In: *International Journal of Heat and Mass Transfer* 15.10 (1972), pp. 1787–1806. ISSN: 0017-9310. DOI: 10.1016/0017-9310(72)90054-3.

[201] W. Pazner and P.-O. Persson. “Stage-parallel fully implicit Runge–Kutta solvers for discontinuous Galerkin fluid simulations.” In: *Journal of Computational Physics* 335 (2017), pp. 700–717. DOI: 10.1016/j.jcp.2017.01.050.

[202] M. Pelikan, D. E. Goldberg, E. Cantú-Paz, et al. "BOA: The Bayesian optimization algorithm." In: *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation - Volume 1*. Vol. 1. GECCO'99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 525–532. ISBN: 1558606114.

[203] P. Perdikaris, M. Raissi, A. Damianou, N. D. Lawrence, and G. E. Karniadakis. "Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling." In: *Proceedings of the Royal Society A* 473.2198 (2017), p. 20160751. DOI: 10.1098/rspa.2016.0751.

[204] V. Picheny, T. Wagner, and D. Ginsbourger. "A benchmark of kriging-based infill criteria for noisy optimization." In: *Structural and Multidisciplinary Optimization* 48.3 (2013), pp. 607–626.

[205] F. Pichi, F. Ballarin, G. Rozza, and J. S. Hesthaven. "Artificial neural network for bifurcating phenomena modelled by nonlinear parametrized PDEs." In: *PAMM* 20.S1 (2021), e202000350.

[206] U. Piomelli, L. S. Craig, and S. Sarkar. "On the computation of sound by large-eddy simulations." In: *Journal of Engineering and Mathematics* 32 (1997), pp. 217–236. DOI: 10.1023/A:1004236206327.

[207] A. Posa, R. Broglia, M. Felli, M. Falchi, and E. Balaras. "Characterization of the wake of a submarine propeller via Large-Eddy simulation." In: *Computer and Fluids* 184 (2019), pp. 138–152. DOI: 10.1016/j.compfluid.2019.03.011.

[208] P. Prebeg, V. Zanic, and B. Vazic. "Application of a surrogate modeling to the ship structural design." In: *Ocean engineering* 84 (2014), pp. 259–272. DOI: 10.1016/j.oceaneng.2014.03.032.

[209] J. L. Proctor, S. L. Brunton, and J. N. Kutz. "Dynamic Mode Decomposition with Control." In: *SIAM Journal on Applied Dynamical Systems* 15.1 (2016), pp. 142–161. DOI: 10.1137/15M1013857.

[210] H. Qian, Y.-Q. Hu, and Y. Yu. "Derivative-Free Optimization of High-Dimensional Non-Convex Functions by Sequential Random Embeddings." In: *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. 2016, pp. 1946–1952.

[211] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*. Vol. 92. Springer, 2015. DOI: 10.1007/978-3-319-15431-2.

[212] A. Quarteroni, G. Rozza, et al. *Reduced order methods for modeling and computational reduction*. Vol. 9. Springer, 2014.

[213] J. Quinonero-Candela, C. E. Rasmussen, and C. K. Williams. "Approximation methods for Gaussian process regression." In: *Large-scale kernel machines*. MIT Press, 2007, pp. 203–223.

[214] A. Rahimi and B. Recht. "Random features for large-scale kernel machines." In: *Proceedings of the 20th International Conference on Neural Information Processing Systems*. NIPS'07. Vancouver, British Columbia, Canada: Curran Associates Inc., 2008, pp. 1177–1184.

[215] J. Raikunen, E. Avi, H. Remes, J. Romanoff, I. Lillemäe-Avi, and A. Niemelä. "Optimization of Passenger Ship Global Model Utilizing Equivalent Single Layer Element." In: *International Conference on Ships and Offshore Structures*. 2018.

[216] J. Raikunen, E. Avi, H. Remes, J. Romanoff, I. Lillemäe-Avi, and A. Niemelä. "Optimisation of passenger ship structures in concept design stage." In: *Ships and Offshore Structures* 14.sup1 (2019), pp. 320–334. DOI: 10.1080/17445302.2019.1590947.

[217] M. Raissi, P. Perdikaris, and G. E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations." In: *Journal of Computational Physics* 378 (2019), pp. 686–707. DOI: 10.1016/j.jcp.2018.10.045.

[218] M. Raissi, P. Perdikaris, and G. E. Karniadakis. "Inferring solutions of differential equations using noisy multi-fidelity data." In: *Journal of Computational Physics* 335 (2017), pp. 736–746. DOI: 10.1016/j.jcp.2017.01.060.

[219] M Ripepi, M. Verveld, N. Karcher, T Franz, M Abu-Zurayk, S Görtz, and T. Kier. "Reduced-order models for aerodynamic applications, loads and MDO." In: *CEAS Aeronautical Journal* 9.1 (2018), pp. 171–193. DOI: 10.1007/s13272-018-0283-6.

[220] J. Romanoff, H. Remes, P. Varsta, J. Jelovica, A. Klanac, A. Niemelä, S. Bralic, and H. Naar. "Hull-superstructure interaction in optimised passenger ships." In: *Ships and Offshore Structures* 8.6 (2013), pp. 612–620. DOI: 10.1080/17445302.2012.675196.

[221] F. Romor, M. Tezzele, A. Lario, and G. Rozza. "Kernel-based Active Subspaces with application to CFD parametric problems using Discontinuous Galerkin method." In: *arXiv preprint arXiv:2008.12083* (2020).

[222] F. Romor, M. Tezzele, and G. Rozza. "ATHENA: Advanced Techniques for High dimensional parameter spaces to Enhance Numerical Analysis." In: *arXiv preprint arXiv:2105.06713* (2020).

[223] F. Romor, M. Tezzele, and G. Rozza. "A local approach to parameter space reduction for regression and classification tasks." In: *arXiv preprint arXiv:2107.10867* (2021).

[224] F. Romor, M. Tezzele, and G. Rozza. "Multi-fidelity data fusion for the approximation of scalar functions with low intrinsic dimensionality through active subspaces." In: *Proceedings in Applied Mathematics & Mechanics*. Vol. 20. S1. Wiley Online Library. 2021. DOI: 10.1002/pamm.202000349.

[225] C. W. Rowley, T. Colonius, and R. M. Murray. "Model reduction for compressible flows using POD and Galerkin projection." In: *Physica D: Nonlinear Phenomena* 189.1-2 (Feb. 2004), pp. 115–129. DOI: 10.1016/j.physd.2003.03.001.

[226] G. Rozza, M. Hess, G. Stabile, M. Tezzele, and F. Ballarin. "Basic Ideas and Tools for Projection-Based Model Reduction of Parametric Partial Differential Equations." In: *Model Order Reduction*. Ed. by P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. H. A. Schilders, and L. M. Silveira. Vol. 2. Berlin, Boston: De Gruyter, 2020. Chap. 1, pp. 1–47. ISBN: 9783110671490. DOI: 10.1515/9783110671490-001.

[227] G. Rozza, M. H. Malik, N. Demo, M. Tezzele, M. Girfoglio, G. Stabile, and A. Mola. "Advances in Reduced Order Methods for Parametric Industrial Problems in Computational Fluid Dynamics." In: *ECCOMAS ECFD 7 - Proceedings of 6th European Conference on Computational Mechanics (ECCM 6) and 7th European Conference on Computational Fluid Dynamics (ECFD 7)*. Ed. by R. Owen, R. de Borst, J. Reese, and P. Chris. Glasgow, UK, 2018, pp. 59–76.

[228] T. M. Russi. "Uncertainty quantification with experimental data and complex system models." PhD thesis. UC Berkeley, 2010.

[229] H. Sakamoto and H. Haniu. "A Study on Vortex Shedding From Spheres in a Uniform Flow." In: *Journal of Fluids Engineering* 112.4 (Dec. 1990), pp. 386–392. ISSN: 0098-2202. DOI: 10.1115/1.2909415.

[230] F. Salmoiraghi, F. Ballarin, G. Corsi, A. Mola, M. Tezzele, and G. Rozza. "Advances in geometrical parametrization and reduced order models and methods for computational fluid dynamics problems in applied sciences and engineering: Overview and perspectives." In: *ECCOMAS Congress 2016 - Proceedings of the 7th European Congress on Computational Methods in Applied Sciences and Engineering* 1 (2016), pp. 1013–1031. DOI: 10.7712/100016.1867.8680.

[231] F. Salmoiraghi, A. Scardigli, H. Telib, and G. Rozza. "Free-form deformation, mesh morphing and reduced-order methods: enablers for efficient aerodynamic shape optimisation." In: *International Journal of Computational Fluid Dynamics* 32.4-5 (2018), pp. 233–247. DOI: 10.1080/10618562.2018.1514115.

[232] R. D. Sandberg and N. D. Sandham. "Direct numerical simulation of turbulent flow past a trailing edge and the associated noise generation." In: *Journal of Fluid Mechanics* 596 (Jan. 2008), pp. 353–385. DOI: 10.1017/s0022112007009561.

[233] M. L. Sanyang and A. Kabán. "REMEDA: Random Embedding EDA for optimising functions with intrinsic dimension." In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2016, pp. 859–868. DOI: 10.1007/978-3-319-45823-6_80.

[234] J. D. Schaffer, R. Caruana, L. J. Eshelman, and R. Das. "A study of control parameters affecting online performance of genetic algorithms for function optimization." In: *Proceedings of the 3rd international conference on genetic algorithms*. 1989, pp. 51–60.

[235] P. J. Schmid. "Dynamic mode decomposition of numerical and experimental data." In: *Journal of fluid mechanics* 656 (2010), pp. 5–28. DOI: 10.1017/S0022112010001217.

[236] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT press, 2002.

[237] B. Schölkopf, A. Smola, and K.-R. Müller. "Kernel principal component analysis." In: *International conference on artificial neural networks*. Springer. 1997, pp. 583–588. DOI: 10.1007/BFb0020217.

[238] B. Schölkopf, A. Smola, and K.-R. Müller. "Nonlinear component analysis as a kernel eigenvalue problem." In: *Neural computation* 10.5 (1998), pp. 1299–1319. DOI: 10.1162/089976698300017467.

[239] E. Schubert and P. J. Rousseeuw. "Faster k-medoids clustering: improving the PAM, CLARA, and CLARANS algorithms." In: *International conference on similarity search and applications*. Springer. 2019, pp. 171–187. DOI: 10.1007/978-3-030-32047-8_16.

[240] T. Sederberg and S. Parry. "Free-Form Deformation of solid geometric models." In: *Proceedings of SIGGRAPH - Special Interest Group on GRAPHics and Interactive Techniques*. SIGGRAPH. 1986, pp. 151–159. DOI: 10.1145/15886.15903.

[241] A. Seena and H. J. Sung. "Dynamic mode decomposition of turbulent cavity flows for self-sustained oscillations." In: *International Journal of Heat and Fluid Flow* 32.6 (Dec. 2011), pp. 1098–1110. DOI: 10.1016/j.ijheatfluidflow.2011.09.008.

[242] J. H. Seo, Y. J. Moon, and B. R. Shin. "Prediction of cavitating flow noise by direct numerical simulation." In: *Journal of Computational Physics* 227.13 (June 2008), pp. 6511–6531. DOI: 10.1016/j.jcp.2008.03.016.

[243] H. Seol, B. Jung, J.-C. Suh, and S. Lee. "Prediction of non-cavitating underwater propeller noise." In: *Journal of Sound and Vibration* 257.1 (Oct. 2002), pp. 131–156. DOI: 10.1006/jsvi.2002.5035.

[244] A. Serani, E. F. Campana, M. Diez, and F. Stern. "Towards augmented design-space exploration via combined geometry and physics based Karhunen-Loève expansion." In: *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2017, p. 3665. DOI: 10.2514/6.2017-3665.

[245] C. Seror, P. Sagaut, C. Bailly, and D. Juvé. "On the radiated noise computed by large-eddy simulation." In: *Physics of Fluids* 13.2 (Feb. 2001), pp. 476–487. DOI: 10.1063/1.1336150.

[246] K. Shahbazi. "An explicit expression for the penalty parameter of the interior penalty method." In: *Journal of Computational Physics* 205.2 (2005), pp. 401–407. DOI: 10.1016/j.jcp.2004.11.017.

[247] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. "Taking the human out of the loop: A review of Bayesian optimization." In: *Proceedings of the IEEE* 104.1 (2015), pp. 148–175. DOI: 10.1109/JPROC.2015.2494218.

[248] W. Shen, T. K. Patel, and S. A. Miller. "Extraction of Large-Scale Coherent Structures from Large Eddy Simulation of Supersonic Jets for Shock-Associated Noise Prediction." In: *AIAA Scitech 2020 Forum*. American Institute of Aeronautics and Astronautics, Jan. 2020. DOI: 10.2514/6.2020-0742.

[249] D. Sieger, S. Menzel, and M. Botsch. "On shape deformation techniques for simulation-based design optimization." In: *New Challenges in Grid Generation and Adaptivity for Scientific Computing*. Springer, 2015, pp. 281–303. DOI: 10.1007/978-3-319-06053-8_14.

[250] L. Sirovich. "Turbulence and the dynamics of coherent structures. II. Symmetries and transformations." In: *Quarterly of Applied Mathematics* 45.3 (Oct. 1987), pp. 573–582. DOI: 10.1090/qam/910463.

[251] R Sivaraj and T Ravichandran. "A review of selection methods in genetic algorithm." In: *International journal of engineering science and technology* 3.5 (2011), pp. 3792–3797.

[252] R. L. Smith. "The hit-and-run sampler: a globally reaching Markov chain sampler for generating arbitrary multivariate distributions." In: *Proceedings Winter Simulation Conference*. IEEE. 1996, pp. 260–264. DOI: 10.1109/WSC.1996.873287.

[253] E. Snelson, Z. Ghahramani, and C. E. Rasmussen. "Warped Gaussian processes." In: *Advances in Neural Information Processing Systems*. Ed. by S. Thrun, L. Saul, and B. Schölkopf. Vol. 16. MIT Press, 2004, pp. 337–344.

[254] C. G. Soares and Y Garbatov. *Progress in the Analysis and Design of Marine Structures: Proceedings of the 6th International Conference on Marine Structures (MARSTRUCT 2017), May 8-10, 2017, Lisbon, Portugal*. CRC Press, 2017.

[255] I. M. Sobol'. "On the distribution of points in a cube and the approximate evaluation of integrals." In: *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki* 7.4 (1967), pp. 784–802.

[256] P. Spalart and S. Allmaras. "A one-equation turbulence model for aerodynamic flows." In: *30th Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, Jan. 1992. DOI: 10.2514/6.1992-439.

[257] B. Sriperumbudur and N. Sterge. "Approximate kernel PCA using random features: computational vs. statistical trade-off." In: *arXiv preprint arXiv:1706.06296* (2017).

[258] G. Stabile and G. Rozza. "Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations." In: *Computers & Fluids* 173 (2018), pp. 273–284. DOI: 10.1016/j.compfluid.2018.01.035.

[259] G. Stabile, F. Ballarin, G. Zuccarino, and G. Rozza. "A reduced order variational multiscale approach for turbulent flows." In: *Advances in Computational Mathematics* 45 (2019), pp. 2349–2368. DOI: 10.1007/s10444-019-09712-x.

[260] G. Stabile, S. Hijazi, A. Mola, S. Lorenzi, and G. Rozza. "POD-Galerkin reduced order methods for CFD using Finite Volume Discretisation: vortex shedding around a circular cylinder." In: *Communications in Applied and Industrial Mathematics* 8.1 (2017), pp. 210–236. DOI: 10.1515/caim-2017-0011.

[261] P. Stegeman, A. Ooi, and J. Soria. "Proper Orthogonal Decomposition and Dynamic Mode Decomposition of Under-Expanded Free-Jets with Varying Nozzle Pressure Ratios." In: *Instability and Control of Massively Separated Flows*. Springer, 2015, pp. 85–90. DOI: 10.1007/978-3-319-06260-0_12.

[262] M. Stein. "Large sample properties of simulations using Latin hypercube sampling." In: *Technometrics* 29.2 (1987), pp. 143–151. DOI: 10.1080/00401706.1987.10488205.

[263] M. Strazzullo, F. Ballarin, and G. Rozza. "POD–Galerkin Model Order Reduction for Parametrized Time Dependent Linear Quadratic Optimal Control Problems in Saddle Point Formulation." In: *Journal of Scientific Computing* 83.3 (2020), p. 55. DOI: 10.1007/s10915-020-01232-x.

[264] M. Sugiyama. "Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis." In: *Journal of machine learning research* 8.5 (2007).

[265] T. J. Sullivan. *Introduction to Uncertainty Quantification*. Vol. 63. Springer, 2015. DOI: 10.1007/978-3-319-23395-6.

[266] M. Tezzele, F. Ballarin, and G. Rozza. "Combined parameter and model reduction of cardiovascular problems by means of active subspaces and POD-Galerkin methods." In: *Mathematical and Numerical Modeling of the Cardiovascular System and Applications*. Ed. by D. Boffi, L. F. Pavarino, G. Rozza, S. Scacchi, and C. Vergara. Vol. 16. SEMA-SIMAI Series. Springer International Publishing, 2018, pp. 185–207. DOI: 10.1007/978-3-319-96649-6_8.

[267] M. Tezzele, N. Demo, M. Gadalla, A. Mola, and G. Rozza. "Model order reduction by means of active subspaces and dynamic mode decomposition for parametric hull shape design hydrodynamics." In: *Technology and Science for the Ships of the Future: Proceedings of NAV 2018: 19th International Conference on Ship & Maritime Research*. IOS Press, 2018, pp. 569–576. DOI: 10.3233/978-1-61499-870-9-569.

[268] M. Tezzele, N. Demo, A. Mola, and G. Rozza. "An integrated data-driven computational pipeline with model order reduction for industrial and applied mathematics." In: *Special Volume ECMI, Springer, In Press* (2021).

[269] M. Tezzele, N. Demo, A. Mola, and G. Rozza. "PyGeM: Python Geometrical Morphing." In: *Software Impacts* 7 (2021), p. 100047. ISSN: 2665-9638. DOI: 10.1016/j.simpa.2020.100047.

[270] M. Tezzele, N. Demo, and G. Rozza. "Shape optimization through proper orthogonal decomposition with interpolation and dynamic mode decomposition enhanced by active subspaces." In: *Proceedings of MARINE 2019: VIII International Conference on Computational Methods in Marine Engineering*. Ed. by R. Bensow and J. Ringsberg. 2019, pp. 122–133.

[271] M. Tezzele, N. Demo, G. Stabile, A. Mola, and G. Rozza. "Enhancing CFD predictions in shape design problems by model and parameter space reduction." In: *Advanced Modeling and Simulation in Engineering Sciences* 7.40 (2020). DOI: 10.1186/s40323-020-00177-y.

[272] M. Tezzele, F. Salmoiraghi, A. Mola, and G. Rozza. “Dimension reduction in heterogeneous parametric spaces with application to naval engineering shape design problems.” In: *Advanced Modeling and Simulation in Engineering Sciences* 5.1 (2018), p. 25. ISSN: 2213-7467. DOI: 10.1186/s40323-018-0118-3.

[273] D. Torlo, F. Ballarin, and G. Rozza. “Stabilized weighted reduced basis methods for parametrized advection dominated problems with random inputs.” In: *SIAM/ASA Journal on Uncertainty Quantification* 6.4 (2018), pp. 1475–1502. DOI: 10.1137/17M1163517.

[274] R. Tripathy and I. Bilionis. “Deep Active Subspaces: A Scalable Method for High-Dimensional Uncertainty Propagation.” In: *ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection. 2019. DOI: 10.1115/DETC2019-98099.

[275] R. Tripathy, I. Bilionis, and M. Gonzalez. “Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation.” In: *Journal of Computational Physics* 321 (2016), pp. 191–223. DOI: 10.1016/j.jcp.2016.05.039.

[276] J. A. Tropp. “User-Friendly Tail Bounds for Sums of Random Matrices.” In: *Foundations of computational mathematics* 12.4 (2012), pp. 389–434. DOI: 10.1007/s10208-011-9099-z.

[277] J. Tu, C. Rowley, D. Luchtenburg, S. Brunton, and N. Kutz. “On dynamic mode decomposition: Theory and applications.” In: *Journal of Computational Dynamics* 1.2 (2014), pp. 391–421. DOI: 10.3934/jcd.2014.1.391.

[278] M. Udell and A. Townsend. “Why are big data matrices approximately low rank?” In: *SIAM Journal on Mathematics of Data Science* 1.1 (2019), pp. 144–160. DOI: 10.1137/18M1183480.

[279] D. Villa, S. Gaggero, A. Coppede, and G. Vernengo. “Parametric hull shape variations by Reduced Order Model based geometric transformation.” In: *Ocean Engineering* 216 (2020), p. 107826. DOI: 10.1016/j.oceaneng.2020.107826.

[280] P. Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.” In: *Nature Methods* (2020). DOI: 10.1038/s41592-019-0686-2.

[281] S. Volkwein. “Proper Orthogonal Decomposition: Theory and Reduced-Order Modelling.” In: *Lecture Notes, University of Konstanz* (Jan. 2012).

[282] Q. Wang, J. S. Hesthaven, and D. Ray. “Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem.” In: *Journal of computational physics* 384 (2019), pp. 289–307. DOI: 10.1016/j.jcp.2019.01.031.

[283] Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. de Feitas. “Bayesian optimization in a billion dimensions via random embeddings.” In: *Journal of Artificial Intelligence Research* 55.1 (2016), pp. 361–387.

[284] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. "A tensorial approach to computational continuum mechanics using object-oriented techniques." In: *Computers in physics* 12.6 (1998), pp. 620–631. DOI: 10.1063/1.168744.

[285] H. Wendland. "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree." In: *Advances in Computational Mathematics* 4.1 (Dec. 1995), pp. 389–396. DOI: 10.1007/bf02123482.

[286] K. Willcox. "Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition." In: *Computers & fluids* 35.2 (2006), pp. 208–226. DOI: 10.1016/j.compfluid.2004.11.006.

[287] C. K. Williams and C. E. Rasmussen. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning series. MIT press Cambridge, MA, 2006.

[288] J. Witteveen and H Bijl. "Explicit mesh deformation using Inverse Distance Weighting interpolation." In: *19th AIAA Computational Fluid Dynamics*. AIAA. 2009. DOI: 10.2514/6.2009-3996.

[289] Q. Wu, F. Liang, and S. Mukherjee. "Localized Sliced Inverse Regression." In: *Journal of Computational and Graphical Statistics* 19.4 (2010), pp. 843–860. DOI: 10.1198/jcgs.2010.08080.

[290] D. Xiao, F. Fang, C. Pain, and G. Hu. "Non-intrusive reduced-order modelling of the Navier-Stokes equations based on RBF interpolation." In: *International Journal for Numerical Methods in Fluids* 79.11 (July 2015), pp. 580–595. DOI: 10.1002/fld.4066.

[291] O. Zahm, P. G. Constantine, C. Prieur, and Y. M. Marzouk. "Gradient-based dimension reduction of multivariate vector-valued functions." In: *SIAM Journal on Scientific Computing* 42.1 (2020), A534–A558. DOI: 10.1137/18M1221837.

[292] O. Zahm, T. Cui, K. Law, A. Spantini, and Y. Marzouk. "Certified dimension reduction in nonlinear Bayesian inverse problems." In: *arXiv preprint arXiv:1807.03712* (2018).

[293] M. J. Zahr and P.-O. Persson. "An adjoint method for a high-order discretization of deforming domain conservation laws for optimization of flow problems." In: *Journal of Computational Physics* 326 (2016), pp. 516–543. DOI: 10.1016/j.jcp.2016.09.012.

[294] G. Zhang, J. Zhang, and J. Hinkle. "Learning nonlinear level sets for dimensionality reduction in function approximation." In: *Advances in Neural Information Processing Systems*. 2019, pp. 13199–13208.

[295] Q. Zhang, Y. Liu, and S. Wang. "The identification of coherent structures using proper orthogonal decomposition and dynamic mode decomposition." In: *Journal of Fluids and Structures* 49 (2014), pp. 53–72. DOI: 10.1016/j.jfluidstructs.2014.04.002.

[296] A. Zhilinskas. "Single-step Bayesian search method for an extremum of functions of a single variable." In: *Cybernetics* 11.1 (1975), pp. 160–166. DOI: 10.1007/BF01069961.