Disciplinary surrogates for the analysis and optimization of partitioned multidisciplinary systems : application to aeroelastic coupling in wing design

Inês da Costa Cardoso

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

ISAE SUPAERO Institut Supérieur de l'Aéronautique et de l'Espace
Université de Toulouse

Doctorat de l'Université de Toulouse

Délivré par l'ISAE-SUPAERO

Disciplinary surrogates for the analysis and optimization of partitioned multidisciplinary systems : application to aeroelastic coupling in wing design

Thèse présentée et soutenue, le 4 décembre 2024 par

Inês DA COSTA CARDOSO

École doctorale
AA - Aéronautique et Astronautique

Spécialité
Mathématiques et Applications

Unité de recherche
ISAE-ONERA MOIS Modélisation et Ingénierie des Systèmes

Thèse dirigée par
Christian GOGU et Sylvain DUBREUIL

Composition du jury
M. Pierre-Alain BOUCARD, président, Examinateur, ENS Paris-Saclay
M. Frederico AFONSO, Examinateur, Universidade de Lisboa
Mme Mathilde CHEVREUIL, Examinatrice, Université de Nantes
M. Sylvain DUBREUIL, Co-directeur de thèse, ONERA
M. Christian GOGU, Directeur de thèse, ISAE-SUPAERO
M. Rodolphe LE RICHE, Rapporteur, CNRS/LIMOS
M. Joaquim MARTINS, Examinateur, University of Michigan
M. David NÉRON, Rapporteur, ENS Paris-Saclay

Membres invités
M. Michel SALAÜN, co-encadrant, ISAE-SUPAERO
Mme Nathalie BARTOLI, co-encadrante, ONERA

Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgments

First and foremost I thank my PhD supervisors, without whom this work would not have been possible. I especially thank Sylvain, who had to put up with my incessant questions, but that always responded with enthusiasm and new ideas. I equally thank Nathalie, who to me was not only a supervisor but a good friend and a role model. And I thank Christian and Michel for both the work discussions as well as their continuous support in my pursuit of a career in the research and academic fields.

My second thanks goes to all the members of my jury. I was especially fortunate to have had a large committee willing to read and evaluate my work. I therefore thank both Rodolphe Le Riche and David Néron for their careful reading of my manuscript and for the kind words they wrote about it on their reports. I am also thankful to Pierre-Alain Boucard and Mathilde Chevreuil, both for their interest in my work as well as for the pertinent questions they posed during my defense. I equally thank Joaquim Martins, who I had the chance to meet at SCITECH and who immediately showed interest in this work. And finally, I thank Frederico Afonso, who I particularly appreciated to have in my jury, as he had been my professor during my Master studies.

My third thanks goes to all the PhD students and interns that I met during these three years. Although I cannot cite all, I do feel some must be thanked in particular. Indeed, I thank Gaspard who not only pioneered this work but that, like Sylvain, withstood my continuous pestering during my first few months at ONERA. I would equally like to thank Charles and Jason that, due to the unfortunate placement of their office, were often annoyed by their loud neighbors. I also thank Romain and Didier for the hours they spent either listening to my complaints or entertaining me when I was bored. I would also like to thank Maëlys, whom I befriended later on but who has become a particularly good friend. Finally, they say to save the best for last. So I would like to thank Julien, with whom I shared my office during these three years. Despite our disagreement in almost every matter, we undoubtedly made these years the most fun that they could be. To him, I wish the best of luck as he continues his life in research.

My fourth thanks goes to all the members of the M2CI team, which I integrated during my time at ONERA. I was lucky to be able to both discuss work with but also befriend most of the permanent researchers. I particularly thank Sébastien, who was not only the kindest team leader, but who also trusted me to teach the Aircraft Design course alongside him. I equally thank Rémi, who is an honorary member of M2CI, but whose help to this work was indispensable.

And last, but certainly not least, I would like to thank my family and friends for their continuous support in my "incomprehensible" decision to pursue a career in research. I especially thank Mariana and Nelson who always welcome me back in Portugal as though I have never left. As for those I met in France, I especially thank my EVA team for their full support during the last and hardest year. In particular, I thank Ninii who has become a close and valuable friend.

Introduction

Context

This thesis was carried out within the doctoral school ED-AA (École Doctorale Aéronautique et Astronautique) and directly follows the work of a previous doctoral student [Berthelin, 2022]. Financing for this thesis was provided by ONERA (Office National d'Études et de Recherche Aérospatiale) and ISAE-SUPAERO (Institut Supérieur de l'Aéronautique et de l'Espace) in the context of the Féderation de recherche ONERA-ISAE-ENAC. At ONERA the work was developed as part of the research activities of the M2CI (Méthodes Multidisciplinaires, Concepts Intégrés) team. The supervision of the thesis was assured by Christian Gogu, professor at ISAE-SUPAERO, and Sylvain Dubreuil, research scientist at ONERA. Nathalie Bartoli, senior researcher at ONERA, and Michel Salaün, professor at ISAE-SUPAERO, equally participated in the supervision of this work.

Problem formulation

Overall aircraft design is the conceptual or preliminary stage of the design process of an aircraft. As the global effort to limit climate changes increases, there is a need to significantly reduce carbon emissions in the aviation industry. To achieve this reduction, different studies point to the transition to sustainable energy sources or to increased aircraft efficiency. These solutions translate into new aircraft configurations, presenting changes in propulsion technologies (such as hybrid-electric [Sgueglia et al., 2018] or hydrogen powered aircraft [Onorato et al., 2022, Adler and Martins, 2024]) or more efficient lifting devices (such as blended wing bodies [Brown and Vos, 2018] or high aspect ratio wings [Brooks et al., 2019]). Overall aircraft design provides a framework to explore these new configurations for which little to no experimental data is available. To do so, the inherent multidisciplinary behavior of the aircraft needs to be appropriately modelled. Indeed, different interacting disciplines are used to model the different physical phenomena that ultimately determine aircraft performance. For instance, one discipline may be used to describe the aerodynamics of the wing, while another may be used to describe its structural deformation. To allow the disciplines to exchange information with one another, a nonlinear system of equations is defined. This system, often called Multidisciplinary Design Analysis (MDA), is illustrated in Figure 1.

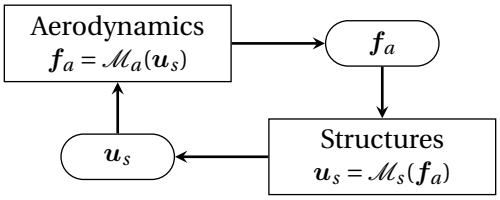


Figure 1: Illustration of an MDA problem involving the aerodynamics and structural disciplines which interact with one another via the coupling variables f_a and u_s, respectively the aerodynamic loads and the structural displacement.

In MDA, the quantities exchanged between the participating disciplines are called coupling variables. Moreover, the coupling variable values for which the feedback loop is verified define the solution of the MDA. Most commonly, this solution is found using iterative approaches, where the values of the coupling variables are updated iteratively until the relative error between two successive updates is within some small tolerance. In overall aircraft design, when coupling the structural and aerodynamics disciplines, the coupling variables are typically the aerodynamic loads $\boldsymbol{f}_a$ and the structural displacements $\boldsymbol{u}_s$. A precise evaluation of these quantities requires the resolution of partial differential equations using expensive numerical solvers. However, due to the multiple iterations, the use of such numerical solvers directly in the MDA framework can become computationally prohibitive.

This computational cost is further increased during the preliminary design studies, where a large number of different aircraft configurations needs to be assessed. Indeed, if either aerodynamic or structural conditions are altered (for example, through changes in flight conditions or wing geometry), the whole resolution of the MDA must be repeated. Moreover, the choice of one aircraft configuration over another is based on certain quantities of interest, such as the fuel burnt or the maximum take-off weight. Because these performance functions depend on the solution of the MDA, overall aircraft design is typically formulated as a Multidisciplinary Design Optimization (MDO) problem. In MDO design variables which control the disciplinary solver conditions are varied in order to find the best possible value of the chosen performance function. A possible way to do this, is by solving the MDA at each optimizer iteration, and to compute the objective function value at the MDA solution, as illustrated in Figure 2. However, other MDO formulations which do not solve the MDA at each iteration can also be used.

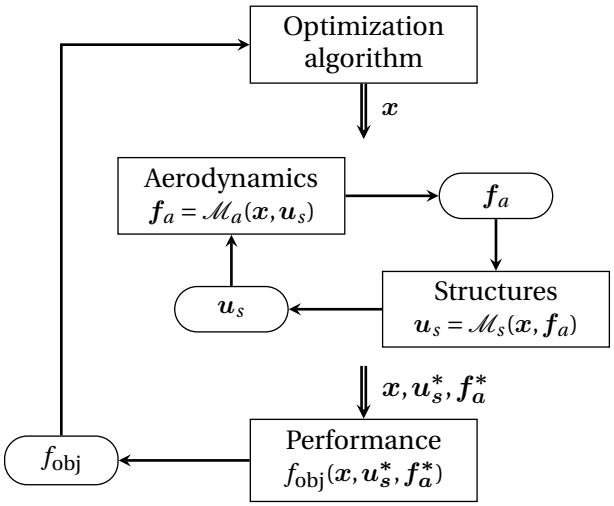


Figure 2: Illustration of an MDO problem involving the aerodynamics and structural disciplines. The set of design variables $\boldsymbol{x}$ determine the disciplinary solver conditions. The performance function to optimize is f_{obj}. To evaluate f_{obj} the solution of the MDA $(\boldsymbol{u}_s^*, \boldsymbol{f}_a^*)$ is required.

Due to the high computational cost associated with the MDO, most preliminary design studies are carried out using empirical formulas or low-fidelity disciplinary solvers. Although these solvers allow for a rapid exploration of the design space, they lack the accuracy offered by their high-fidelity counterparts. When attempting to design new aircraft configurations, for which experimental data is scarce, optimization algorithms take advantage of the lack of accuracy of these low-fidelity solvers to produce unrealistic optimal designs [Knill et al., 1996]. As a result, using empirical formulas fine-tuned for existing aircraft configurations is no longer a feasible option and one must rely on high-fidelity solvers.

The initial efforts to use high-fidelity disciplinary solvers in the context of MDO date back to the early 2000s [Giunta, 2000, Maute et al., 2001]. At the time, the use of high-fidelity disciplinary solvers was motivated by the lack of accuracy obtained when designing supersonic transport aircraft using simpler, low-fidelity models. Indeed, at such high speeds, non-linear phenomena such as wave drag become of relative importance and dedicated high-fidelity disciplinary solvers must be used to properly account for them. As the need for more accurate disciplinary solvers was identified, research to allow for the use of expensive high-fidelity solvers directly in the MDO framework was increased. An important step in this direction was the coupled-adjoint method, initially developed for the calculation of aero-structural sensitivities [Martins et al., 2005]. The coupled-adjoint method allows to efficiently perform gradient-based optimization of tightly coupled multidisciplinary systems and continues to be extensively used in this context [Jasa et al., 2018, Gray et al., 2019, Bons and Martins, 2020]. But even though the coupled-adjoint method is compatible with the use of high-fidelity disciplinary solvers, it requires that the disciplinary solver functions be differentiable. Besides that, it needs to compute gradient information at a disciplinary solver level, which is not cost free. Most high-fidelity solvers, however, are commercial codes for which only inputs and outputs are known. Moreover, gradient-based optimizers are designed to converge to local optima. When designing new aircraft configurations, little is known about the performance function and the existence of multiple local minima cannot, *a priori*, be discarded. For example, in [Bons et al., 2019] multiple local optima are found when design variables are allowed to take values outside their conventional bounds. To address this challenge, gradient-based optimizers must thus resort to multi-start strategies, greatly increasing the computational cost.

Bayesian optimization algorithms [Močkus, 1975] provide an alternative to gradient-based optimization which may be more fit for the preliminary design studies of new aircraft configurations. These algorithms typically replace expensive to evaluate functions by approximated models which they use to directly search for the global optimum. Among others, the Efficient Global Optimization (EGO) algorithm [Jones et al., 1998] stands out as one of the first Bayesian optimization algorithms. In EGO, the objective function is approximated by a Gaussian Process (GP) [Krige, 1951, Rasmussen and Williams, 2005] built from an initial Design of Experiments (DoE) [Sacks et al., 1989]. The global optimum of the function is then found by adaptively sampling the design space through a combination of exploration of the domain and exploitation of the regions where the global optimum is likely to be. And although the original EGO algorithm was developed for single discipline optimization, variants of this algorithm have been used in recent literature to perform MDO [Priem et al., 2020b, Needels and Alonso, 2023].

It is not astonishing that the reduced number of objective function calls makes Bayesian optimization algorithms appealing for problems involving expensive disciplinary solvers. However, in an MDO context, both the initial DoE as well as each newly sampled point require a new resolution of the MDA. This means that algorithms like EGO do not exploit the fact that the solution of the MDA is itself the result of a partitioned system. A more interesting idea is thus to replace the costly disciplinary solvers themselves by surrogate models. This approach is applied in an MDO context in [Zhang et al., 2017, Wang et al., 2017], where disciplinary surrogates are used to perform global optimization using an evolutionary algorithm. More recently, in [Scholten and Hartl, 2021], disciplinary surrogates were also used to perform the static aeroelastic analysis of an aircraft wing. Despite the observed reduced computational cost, the proposed approaches present the shortcoming of requiring sufficiently accurate surrogates throughout the entire design space. For the purpose of overall aircraft design, where interest lies primarily in promising regions of the design space, the construction of globally accurate surrogates may represent a waste of computational effort.

One should thus look for an optimization framework that combines the reduced computa-

tional cost of disciplinary surrogates with the adaptive sampling of Bayesian algorithms. Efforts to develop such an optimization framework have recently been started in [Berthelin, 2022], where an MDA strategy called Disciplinary Proper Orthogonal Decomposition and Interpolation (DPOD+I) was developed. The DPOD+I strategy proposes to approximate the disciplinary solver outputs using a combination of Proper Orthogonal Decomposition (POD) with GP interpolation. By adaptively enriching the resulting disciplinary surrogates during the resolution of the MDA, the DPOD+I strategy significantly reduces the number of disciplinary solver calls required. Application of the DPOD+I strategy in an MDO context further confirmed the interest of this strategy for applications requiring the use of expensive numerical solvers. Particularly, the reduction of computational cost was shown to be most significant when combined with the Efficient Global Multidisciplinary Design Optimization (EGMDO) algorithm proposed in [Dubreuil et al., 2020]. EGMDO is a global optimization algorithm which relies on the use of disciplinary GP approximations to obtain a surrogate model of the performance function. Through a dedicated uncertainty reduction strategy, the EGMDO algorithm is capable of enriching the disciplinary surrogates only in promising regions of the design space.

One of the limitations of the DPOD+I strategy, however, is that the obtained reduced coupling variable space must remain of relatively low dimension. This is because classical GP approximations do not perform well in high dimensions (typically over 20). However, due to the spatial discretization performed by the numerical solvers, the coupling variables involved in the MDA are often high-dimensional vectors, representing for example pressure or displacement fields. The POD step of the DPOD+I strategy is thus specifically thought to reduce the dimension of the coupling variable space and allow for the constructions of the disciplinary surrogates. Nevertheless, the chosen Model Order Reduction (MOR) strategy was shown to lead, in some cases, to a large dimension of the coupling variable space, notably when changes in the design variables contributed with significantly new information to the disciplinary POD bases. As a consequence, both accuracy and performance of the DPOD+I strategy were impacted.

A possible lead to address this challenge may rely on the choice of a different MOR strategy. Indeed, the linear approximation obtained by a global POD basis may not be the most fit for the combined use with GP interpolation, as the number of coefficients required to obtain a small projection error is often large. In the context of MOR, some authors propose to replace the use of a global POD basis by the interpolation of a database of local reduced order bases [Amsallem and Farhat, 2008, Amsallem et al., 2009], while other works propose to abandon POD altogether in favor of quadratic approximations [Barnett and Farhat, 2022, Geelen et al., 2023]. Since, for the same level of accuracy, these strategies result in a lower dimension than that obtained by a global POD basis, they may be more fit for the combined use with GP interpolation.

The EGMDO algorithm also presents some limitations that must be addressed before it can be used for preliminary design studies. Indeed, despite its computational efficiency, the EGMDO algorithm lacks a proper constraint handling strategy. Constraints are used in overall aircraft design to ensure the feasibility of the optimal solution found. Without them, the designed aircraft may, for example, not respect air travel regulations or, in more serious cases, incur structural failure. Development of a constraint handling strategy that does not deteriorate the performance of the algorithm is particularly desirable. Fortunately, several works have been dedicated to extending the Bayesian optimization framework to constrained problems [Sasena et al., 2002, Parr et al., 2012, Picheny, 2014] and could serve as a foundation for possible developments.

Finally, another lead to reduce the computational cost of the MDO problem is to combine the DPOD+I strategy for the resolution of the MDA with a gradient-based optimization algorithm. We have previously discarded the use of gradient-based algorithms due to the black-box nature of most commercial codes. However, when using the DPOD+I strategy, the derivatives of the disciplinary GPs can be obtained analytically. This allows for the use of gradient-based

optimizers while still retaining a non-intrusive approach. Additionally, when using adaptively enriched disciplinary surrogates, the multi-start techniques required to find the global optimum do not necessarily translate into a significant increase of the computational cost.

Objectives

The objective of this thesis is to address the shortcomings of existing analysis and optimization frameworks that use disciplinary surrogates to reduce the computational cost of preliminary design studies of multidisciplinary systems. The previously described DPOD+I strategy and EGMDO algorithm serve as main foundation for the work developed. The main shortcomings of each of these frameworks are described in the following:

1. The DPOD+I strategy loses both performance and accuracy when the dimension of the reduced coupling variable space is large.
2. The EGMDO algorithm lacks a dedicated constraint handling strategy and thus is unfit for most preliminary design studies.
3. Gradient-based optimization is too costly when the derivatives of the disciplinary solvers are not available and must be approximated.

To address the first shortcoming presented above, it is necessary to adapt the dimension reduction strategy of the DPOD+I strategy. As a result, one of the main objectives of this thesis is to explore other dimension reduction strategies and to adapt their formulation to the context of multidisciplinary analysis. Moreover, we propose to apply the developed framework to a test case involving a non-linear aerodynamics solver, for which the original DPOD+I strategy performed poorly. It is equally necessary to extend the EGMDO algorithm to constrained problems, so that it can be used in preliminary design studies. Another main objective of this thesis is thus to develop a dedicated constraint handling strategy for the EGMDO algorithm that retains the cost saving approach of the original formulation. Finally, when the derivatives of the disciplinary solvers are not available and must be approximated, gradient-based optimization is too costly. However, the use of disciplinary surrogates allows for an analytical approximation of the disciplinary solver derivatives. Thus, the third main objective of this work is to develop a gradient-based optimization framework that uses the derivatives of the disciplinary surrogates to drive the search. The outline for this thesis is provided in the following.

Thesis outline

This thesis is divided into five main chapters. The first two chapters describe the state of the art, while the remaining three chapters contain the contributions of this work. Each chapter is briefly described in the following.

In Chapter 1 the concepts and numerical tools for the optimization of multidisciplinary systems are introduced. To do so, the chapter is divided into two main sections. The first section introduces numerical optimization, with special focus given to gradient-based optimization and surrogate-based optimization algorithms. Constrained optimization algorithms are equally introduced in this section. The second section is dedicated to multidisciplinary design analysis and optimization. The main algorithms used to find the solution of the MDA are described and a review of the existing MDO formulations is provided. Then, the EGMDO algorithm is presented.

In Chapter 2 the concepts or tools to obtain computationally cheaper approximations of the disciplinary solvers are presented. To do so, an introduction to the numerical resolution of the

parameterized Partial Differential Equations (PDEs) that describe the physical behavior of the considered disciplines is given. Next, we introduce projection-based Model Order Reduction as a method for reducing the computational burden associated with the resolution of the parameterized PDEs in a multi-query context. Finally, the DPOD+I strategy is described.

In Chapter 3 we address the challenges encountered by the DPOD+I strategy, by proposing to replace the use of a global POD basis with the interpolation of a database of pointwise local POD bases. The use of pointwise local POD bases has most often been used in the literature in the context of single-discipline analysis where one of the parameters was time. In this chapter, we propose to use this strategy in an MDA context. We thus adapt the training and enrichment strategies of the DPOD+I approach to accommodate the use of local POD bases. Numerical test cases are used to confirm the interest of the proposed approach.

In Chapter 4 we propose to use the disciplinary surrogates obtained via the adapted DPOD+I strategy to solve MDO problems using a gradient-based optimizer. To that end, we propose to use the derivatives of the built disciplinary surrogates, which can be obtained analytically, to drive the search for a local optimum. Application to both an analytical and engineering test cases confirms the interest of the proposed approach.

In Chapter 5 we develop a dedicated constraint handling strategy for the EGMDO algorithm. In it is shown through numerical tests that the reformulated algorithm, named Constrained EGMDO (C-EGMDO), is capable of handling both equality and inequality constraints. Moreover, thanks to the use of disciplinary surrogates, the developed algorithm results in fewer disciplinary solver calls than other existing approaches.

Other activities

One of the main goals during this thesis was that any developed algorithm could be easily used for other research activities at ONERA and namely for those carrying out preliminary aircraft design studies. As a result, an effort has been made to produce computer code that was compatible for integration within the in-house web application WhatsOpt [Lafage et al., 2019]. Additionally, this thesis gave place to an internship where the goal was to implement some of the developed approximation methods within the python package Surrogate Modeling Toolbox [Bouhlel et al., 2019, Saves et al., 2024]. This package is developed and used both at ONERA and ISAE-SUPAERO. Finally, during these last three years I have actively participated in teaching activities at ISAE-SUPAERO. Among others, I have supervised practical courses in the fields of optimization, aircraft design and solid mechanics.

Contributions

Journal papers:

- I. Cardoso, S. Dubreuil, N. Bartoli, C. Gogu, M. Salaün, *Constrained efficient global multidisciplinary design optimization using adaptive disciplinary surrogate enrichment*, Structural and Multidisciplinary Optimization [Cardoso et al., 2024a].

Conference papers:

- I. Cardoso, S. Dubreuil, N. Bartoli, C. Gogu, M. Salaün, R. Lafage, *Disciplinary surrogates for gradient based optimization of multidisciplinary systems*, in proceedings of II ECCOMAS Thematic Conference on Multidisciplinary Design Optimization of Aerospace Systems [Cardoso et al., 2023].

- I. Cardoso, S. Dubreuil, N. Bartoli, C. Gogu, M. Salaün, *Model order reduction for parameterized multidisciplinary analysis using disciplinary surrogates: application to non-linear solvers*, AIAA SCITECH 2024 Forum [Cardoso et al., 2024b].

Oral communications:

- I. Cardoso, G. Berthelin, S. Dubreuil, M. Salaün, N. Bartoli, C. Gogu, *Model Order Reduction and Bayesian Optimization for MDO problems*, 25$^{\text{ème}}$ Congrès Français de Mécanique, Nantes, France (2022).
- I. Cardoso, S. Dubreuil, N. Bartoli, C. Gogu, M. Salaün, *Bayesian optimization formulation for constrained multidisciplinary problems*, 15$^{\text{th}}$ World Congress on Structural and Multidisciplinary Optimization, Cork, Ireland (2023).
- I. Cardoso, S. Dubreuil, N. Bartoli, C. Gogu, M. Salaün, R. Lafage, *Disciplinary surrogates for gradient based optimization of multidisciplinary systems*, AeroBest, Lisbon, Portugal (2023).
- I. Cardoso, S. Dubreuil, N. Bartoli, C. Gogu, M. Salaün, R. Lafage, *Dealing with a non-Gaussian objective function in aeroelastic Bayesian optimization with disciplinary Gaussian Processes*, Journée du GST "Mécanique et Incertain", Toulouse, France (2023).
- I. Cardoso, S. Dubreuil, N. Bartoli, C. Gogu, M. Salaün, *Model order reduction for parameterized multidisciplinary analysis using disciplinary surrogates: application to non-linear solvers*, AIAA SciTech Forum, Orlando, Florida, USA (2024).
- I. Cardoso, S. Dubreuil, N. Bartoli, C. Gogu, *Solving PDE constrained MDO problems using non-intrusive disciplinary surrogates*, 9th European Congress on Computational Methods in Applied Sciences and Engineering, Lisbon, Portugal (2024).

I

State of the Art

CHAPTER 1

Optimization of multidisciplinary systems

Contents

In this chapter we will:

- *Introduce gradient-based and global optimization algorithms.*
- *Introduce constrained optimization.*
- *Introduce multidisciplinary design analysis and optimization.*
- *Introduce the EGMDO algorithm.*
- *Introduce existing tools for numerical optimization.*

In this chapter we introduce the concepts and numerical tools for the optimization of multidisciplinary systems. To do so, the chapter is divided into three main sections. The first section introduces numerical optimization, with special focus given to gradient-based optimization and surrogate-based optimization algorithms. Constrained optimization algorithms are equally introduced in this section. The second section is dedicated to multidisciplinary design analysis and optimization. The main algorithms used to find the solution of the MDA are described and a review of the existing MDO formulations is provided. Then, the EGMDO algorithm is presented in detail. Finally, a third section introduces some of the existing tools to perform numerical optimization.

1.1 Numerical optimization

In this first part, we will forget the multidisciplinary nature of the system and address the numerical tools that exist in the literature for the optimization of a given performance function. Constraint functions will equally be put momentarily aside and will be addressed later on in this section. The unconstrained optimization problem to treat is thus defined as:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathscr{X}} f_{\text{obj}}(\boldsymbol{x}) \tag{1.1}$$

where $f_{\text{obj}} : \mathscr{X} \subset \mathbb{R}^n \to \mathbb{R}$ is the objective or performance function, $\mathscr{X}$ is the design space and $\boldsymbol{x} \in \mathscr{X}$ is the vector of design variables which influence the value of f_{obj}. Note that, for simplification, in Eq. (1.1) $\boldsymbol{x}$ is presumed to be made of n continuous design variables. Discrete or categorical design variables often require dedicated algorithms, not considered in this work. We also note that, although $\boldsymbol{x}$ could take any value, it is more computationally efficient to limit the design space based on problem specifications or on the physical limitations of the system. As a result, $\mathscr{X}$ is typically not equal to $\mathbb{R}^n$ but is defined as a subset of it. Some assumptions are also made for the objective function. Namely, we assume that:

1. Evaluating f_{obj} is computationally expensive; calls to the objective function should be kept to a minimum.
2. The function f_{obj} is treated as a black-box; only its inputs and outputs are known.
3. The gradient of f_{obj} is presumably not available; when required, it must be numerically approximated.
4. The function f_{obj} may have multiple local minima; it is our goal to find the global optimum.

The role of optimization algorithms is to solve the problem defined in Eq. (1.1) within reasonable computational times. However, not all existing optimization algorithms are well suited un-

der the above described assumptions. Indeed, stochastic algorithms such as Simulated Annealing [Kirkpatrick et al., 1983], Genetic Algorithms [Katoch et al., 2021] or Particle Swarm Optimization [Kennedy and Eberhart, 1995] require, by default, a large number of objective function evaluations and thus go against assumption 1. Of the remaining classes of algorithms, we will distinguish between two categories: algorithms that perform local optimization and algorithms that use surrogate approximations to directly look for the global optimum.

Optimality conditions. Before presenting the different algorithms used to solve Eq. (1.1), we introduce the optimality conditions which ensure that $\boldsymbol{x}^*$ is a local minimum of f_{obj} [Nocedal and Wright, 1999]. By definition, $\boldsymbol{x}^*$ is a local minimum of f_{obj} if, for all points lying in the neighborhood of $\boldsymbol{x}^*$:

$$f_{\text{obj}}(\boldsymbol{x}^*) \leq f_{\text{obj}}(\boldsymbol{x}) \tag{1.2}$$

Assuming f_{obj} is smooth and twice continuously differentiable, Eq. (1.2) can be verified by looking at the gradient ∇f_{obj} and the Hessian $\mathbf{H}_{f_{\text{obj}}}$ at $\boldsymbol{x}^*$. Indeed, Eq. (1.2) is true if:

$$\nabla f_{\text{obj}}(\boldsymbol{x}^*) = 0 \tag{1.3}$$

Equation (1.3) is called the first-order optimality condition. And while it is a necessary condition, it is not sufficient to ensure optimality. A second-order optimality condition is thus required, which states that $\mathbf{H}_{f_{\text{obj}}}(\boldsymbol{x}^*)$ must be at least positive semidefinite. If both conditions are verified, then $\boldsymbol{x}^*$ is a local minimum of f_{obj}. Local optimization algorithms, introduced in the following, attempt to find $\boldsymbol{x}^*$ such that first and second order optimality conditions are verified.

1.1.1 Local optimization algorithms

Local optimization algorithms rely on a search direction which advances the optimization process towards a local optimum. These algorithms typically require a starting point and move through the design space by iteratively searching the direction which is most likely to reduce the objective function value. When a new descent direction can no longer be found, the search is ended. Different strategies can be used to find the search direction, but the most popular way is to use gradient information.

1.1.1.1 Gradient-based algorithms

As the name implies, gradient-based algorithms use gradient information to drive the search for the optimal solution. Namely, they build a local approximation of the objective function by means of a Taylor expansion, which is then used to predict the objective function value around an initial point $\boldsymbol{x}_0 \in \mathscr{X} \subset \mathbb{R}^n$. For any given objective function $f_{\text{obj}}(\boldsymbol{x})$, the second-order Taylor expansion around $\boldsymbol{x}_0$ is given by:

$$\hat{f}_{\text{obj}}(\boldsymbol{x}) \approx f_{\text{obj}}(\boldsymbol{x}_0) + \nabla f_{\text{obj}}(\boldsymbol{x}_0)^\intercal (\boldsymbol{x} - \boldsymbol{x}_0) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_0)^\intercal \mathbf{H}_{f_{\text{obj}}}(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0) \tag{1.4}$$

where $\nabla f_{\text{obj}}(\boldsymbol{x}_0)$ and $\mathbf{H}_{f_{\text{obj}}}(\boldsymbol{x}_0)$ are, respectively, the gradient and Hessian matrix of the objective function at $\boldsymbol{x}_0$. When either gradient or Hessian are not available, they may be approximated via finite differences. The forward finite-difference approximation of the first derivative is:

$$\frac{\partial f_{\text{obj}}(\boldsymbol{x}_0)}{\partial x_i} \approx \frac{f_{\text{obj}}(\boldsymbol{x}_0 + h\boldsymbol{e}_i) - f_{\text{obj}}(\boldsymbol{x}_0)}{h}, \quad i = 1, \ldots, n \tag{1.5}$$

where h is the step size ($h \ll 1$) and $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$ are the vectors of the canonical basis of $\mathbb{R}^n$. While finite-difference approximations are easy to implement, they are very sensitive to the choice

of step-size h. A more robust approach is the complex step approximation [Lyness and Moler, 1967, Squire and Trapp, 1998], written as:

$$\frac{\partial f_{\text{obj}}(\boldsymbol{x}_0)}{\partial x_j} \approx \frac{\text{Im}[f_{\text{obj}}(\boldsymbol{x}_0 + ih\boldsymbol{e}_j)]}{h}, \quad j = 1, ..., n \tag{1.6}$$

where $i = \sqrt{-1}$. It should be noted, however, that while the finite-difference approximation can be easily used in a black-box approach, the complex-step approximation requires the objective function code to be able to handle complex numbers.

Gradient-based algorithms are considered to be rather efficient as, once the gradient and Hessian matrix are available, these algorithms are able to start from an initial point $\boldsymbol{x}_0 \in \mathcal{X}$ and generate a series of points $\boldsymbol{x}_1, ..., \boldsymbol{x}_k \in \mathcal{X}$ that converge to a local optimum. The different gradient-based algorithms thus differ in the manner in which they generate this converging series of points. Among the most common methods are the gradient descent method, which uses only gradient information and therefore is called a first-order method and Newton's method which uses Hessian information and therefore is considered a second-order method. Because the latter usually performs better, only Newton's method is described in the following.

Newton's method. Second order methods, such as Newton's method, perform better than first order methods because they use curvature information to get a better estimate of the search direction. Indeed, when using the curvature information it is possible to predict a descent direction and step length which will bring the function derivative close to zero (recall that this a necessary optimality condition). Denoting $d_k \in \mathbb{R}^n$ the descent direction, the Taylor series expansion around any point $\boldsymbol{x}_k$ can be written as:

$$\hat{f}_{\text{obj}}(\boldsymbol{x}_k + d_k) \approx f_{\text{obj}}(\boldsymbol{x}_k) + d_k \nabla f_{\text{obj}}(\boldsymbol{x}_k) + \frac{1}{2} d_k{}^{\intercal} \mathbf{H}_{f_{\text{obj}}}(\boldsymbol{x}_k) d_k \tag{1.7}$$

Differentiating Eq. (1.7) with respect to d_k and setting the derivative to zero yields the following linear system of equations:

$$\mathbf{H}_{f_{\text{obj}}}(\boldsymbol{x}_k) d_k = -\nabla f_{\text{obj}}(\boldsymbol{x}_k) \tag{1.8}$$

which can be used to determine d_k at iteration k. The step length $s_k \in \mathbb{R}$ is then determined by performing a 1-D line search in the descent direction [Grippo et al., 1989]. Note that, in Eq. (1.8), if the Hessian matrix is not positive definite, d_k may not be a valid descent direction. To help mitigate this issue, Quasi-Newton methods, such as the Broyden-Fletcher-Goldfarb-Shanno (BFGS) [Broyden, 1970, Fletcher, 1970, Goldfarb, 1970, Shanno, 1970] algorithm, replace the real Hessian by an approximation of it which is positive definite by construction. Quasi-Newton approaches are also more fit for combined use with commercial codes, for which Hessian information is usually not available. Algorithm 1 summarizes the main steps of the Quasi-Newton method.

Newton methods, like other gradient based algorithms, are stopped when a new search direction can no longer be found. In practice, this is decided based on the relative progress of the algorithm at the most recent iteration. Typical stopping criteria include a minimum change in the gradient norm, step size or objective function value. A limit on the number of iterations is also typically set to prevent the algorithm from running indefinitely.

1.1.1.2 Trust-region algorithms

Trust region methods [Powell, 1970, Conn et al., 2000] were first developed to address the convergence issues that occur for gradient-based algorithms whenever the point $\boldsymbol{x}_{k+1}$ is far from the current point $\boldsymbol{x}_k$. These issues arise because the built approximation for the objective function

Algorithm 1 Quasi-Newton method

Input: $\boldsymbol{x}_0, k_{\max}$ ▷ Starting point and max. number of iterations
  $k = 0$ ▷ Initialize iteration count
  conv = False
  while conv == False **and** $k < k_{\max}$ **do**
    Compute or approximate $\nabla f_{\text{obj}}(\boldsymbol{x}_k)$
    Approximate $\tilde{\mathbf{H}}_{f_{\text{obj}}}(\boldsymbol{x}_k)$ ▷ Quasi-Newton approximation
    Obtain descent direction d_k ▷ Newton step (Eq. (1.8))
    Compute step length s_k ▷ 1D line search
    $\boldsymbol{x}_{k+1} = x_k + s_k d_k$ ▷ Update design variable values
    if stopping criteria **then**
      conv = True ▷ Stop if convergence criteria are met
    end if
    $k = k + 1$ ▷ Increase iteration count
  end while
Output: $\boldsymbol{x}_k$

(Eq. (1.7)) is only locally accurate. To address this challenge, trust-region approaches propose to iteratively minimize the objective function within a restricted region where the local model is expected to be sufficiently good. This restricted region, called *trust-region*, is typically defined as a sphere S_k of radius ρ_k centered around the current point $\boldsymbol{x}_k$. Saying that the next point $\boldsymbol{x}_{k+1}$ should belong to S_k is thus equivalent to solving the following optimization problem:

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x} \in S_k} f_{\text{obj}}(\boldsymbol{x}) \tag{1.9}$$

According to Eq. (1.9), two consecutive points $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k+1}$ cannot be farther away from one another than the radius ρ_k. As a result, if too small a radius is chosen, the algorithm can take a long time to converge. Contrarily, if the radius is too large, the local model is not sufficiently accurate within the entire trust-region and convergence issues may arise. For this reason, most trust-regions approaches propose to update the size of the trust-region at each iteration. One metric that can be used to decide if the trust-region size should be updated is the ratio of actual function decrease over the expected function decrease:

$$r_k = \frac{f_{\text{obj}}(\boldsymbol{x}_k) - f_{\text{obj}}(\boldsymbol{x}_{k+1})}{\hat{f}_{\text{obj}}(\boldsymbol{x}_k) - \hat{f}_{\text{obj}}(\boldsymbol{x}_{k+1})} \tag{1.10}$$

In practice, if r_k is too small, the model does not accurately represent the actual function, the new point should be rejected and the trust-region should be shrunk. Contrarily, if r_k is close to unity it means the actual function is well represented, the new point should be accepted and the trust-region can be expanded. If r_k is neither too small nor close to unity, the new point can be accepted but the size of the trust-region is kept. Algorithm 2 summarizes the main steps of the trust-region method.

1.1.1.3 Gradient-free algorithms

Local, gradient-free algorithms are also present in the literature. For instance the Constrained Optimization by Linear Approximation (COBYLA) algorithm [Powell, 1994] is a type of trust-region algorithm which builds a linear approximation of the objective function in each iteration by using only objective function evaluations. Other popular gradient-free methods include the

Algorithm 2 Trust-region method

Input: $\boldsymbol{x}_0, \rho_0, k_{\max}$ ▷ Starting point, initial radius and max. iterations
  $k = 0$ ▷ Initialize iteration count
  conv = False
  while conv == False **and** $k < k_{\max}$ **do**
    Compute or approximate $\nabla f_{\text{obj}}(\boldsymbol{x}_k)$
    Compute or approximate $\tilde{\mathbf{H}}_{f_{\text{obj}}}(\boldsymbol{x}_k)$
    Find $\boldsymbol{x}_{k+1}$ within the trust-region ▷ New point found using Eq. (1.9)
    Compute metric r_k ▷ Performance evaluation (Eq. (1.10))
    if $r_k < \eta_1$ **then**
      $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$ ▷ Reject new point
      Compute ρ_{k+1} such that $\rho_{k+1} < \rho_k$ ▷ Shrink trust-region
    else if $r_k > \eta_2$ **then**
      Compute ρ_{k+1} such that $\rho_{k+1} > \rho_k$ ▷ Expand trust-region
    end if
    if stopping criteria **then**
      conv = True ▷ Stop if convergence criteria are met
    end if
    $k = k + 1$ ▷ Increase iteration count
  end while
Output: $\boldsymbol{x}_k$

simplex-based Nelder-Mead algorithm [Nelder and Mead, 1965] and generalized pattern search approaches [Torczon, 1997, Audet and Dennis, 2006]. Compared to gradient-based approaches, gradient-free algorithms are more suitable for the optimization of noisy or discontinuous functions. However, because they do not use gradient information, these methods typically require more iterations than their gradient-based counterparts.

1.1.1.4 Disadvantages of local optimization algorithms

Local optimization algorithms which use gradient information are particularly attractive because they efficiently find a search direction which is likely to improve the objective function value. Thus, even if the algorithm is stopped due to budget limitations, the obtained solution will still be better than the initial starting point. Like all other methods, however, local optimization algorithms also present some downsides. Namely, they are not appropriate for the optimization of multi-modal functions, as, depending on the starting point, they may get stuck on local minima. In this case, a multi-start technique [Betro and Schoen, 1992], where the optimization is run for different initial points, is required. The global optimum is then taken as the best within all optima found. If gradient information is not available and needs to be approximated using finite differences, multi-start techniques might be computationally unfeasible. Indeed, for $\boldsymbol{x} \in \mathbb{R}^n$, each gradient evaluation requires at least $n+1$ objective function evaluations (see Eq. (1.5)). If the objective function is costly to evaluate or n is large, the computational cost may even become prohibitive. In the next section, surrogate-based optimization, which attempts to preserve computational cost by directly searching for the global optimum, is presented.

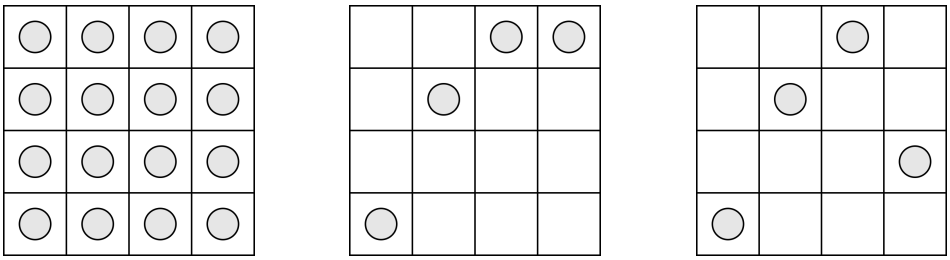
1.1.2 Surrogate-based optimization

Global optimization algorithms, as the name implies, attempt to explore the regions of the design space where the global optimum is likely to be. Of the existing types of optimization al-

gorithms, surrogate-based optimization algorithms are of particular interest due to their cost-saving approach. These algorithms typically build a model of the objective function which they exploit in order to solve the optimization problem. By adaptively enriching the surrogate in regions of the design space where the global optimum is expected to lie, surrogate-based algorithms are able to keep objective function calls to a minimum. In the following, an overview of surrogate-based optimization algorithms is provided.

1.1.2.1 The Design of Experiments (DoE)

Surrogate-based algorithms typically begin by generating a DoE [Sacks et al., 1989] composed of a set of d selected points $\boldsymbol{x}_{\text{DoE}} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_d\}$ and their corresponding outputs $\boldsymbol{f}_{\text{DoE}} = \{f^1_{\text{obj}}, ..., f^d_{\text{obj}}\}$, with $f^i_{\text{obj}} = f_{\text{obj}}(\boldsymbol{x}_i)$. Typically, space-filling sampling strategies which attempt to cover the entire design space are preferred for the generation of the DoE. Several space-filling strategies exist in the literature but not all are appropriate for use with expensive numerical solvers. For instance, an intuitive space-filling method is full factorial sampling. For an n-dimensional input space, full factorial sampling builds, in each dimension, a grid of p evenly spaced points. However, if a fine grid is used or if the input space is high-dimensional, this results in many objective function calls (p^n points). Simple random sampling or Monte Carlo methods could also be used. These methods draw random samples from a probability distribution over the design space. By fixing the number of drawn samples to a maximum number of objective function calls, the computational cost can be reduced compared to full factorial sampling. Regardless, even if a uniform probability distribution is used, Monte Carlo methods will not necessarily result in a DoE which is well spread throughout the design space. Stratified sampling strategies, such as the Latin Hypercube Sampling (LHS) strategy [Mckay et al., 1979], typically result in DoEs which are more space-filling than the ones obtained using Monte Carlo methods. To do so, the LHS strategy partitions the range of each input variable into d equally probable intervals. Then, by drawing random permutations, it places a point per interval. Figure 1.1 illustrates the generated DoEs obtained using full factorial sampling (Figure 1.1(a)), Monte Carlo sampling (Figure 1.1(b)) and LHS (Figure 1.1(c)) for a two-dimensional input space. It is worth noting that Figure 1.1 is merely representative and that, in practice, the training points are not necessarily placed at cell centers.



(a) Full factorial (b) Monte Carlo (c) LHS

Figure 1.1: Generation of the DoE for a two-dimensional input space. Illustration of the DoEs obtained via (a) full factorial sampling with $n = 2$ and $p = 4$, (b) Monte Carlo sampling using a uniform probability distribution and (c) LHS with $d = 4$.

1.1.2.2 Gaussian Process (GP) approximation

Once the initial DoE is generated, it can be used as training data to build a surrogate model of the objective function. Different types of surrogates exist in the literature which are used to approximate scalar functions. Among the most popular methods are linear least squares regres-

sion [Hastie et al., 2001], Radial Basis Function (RBF) [Broomhead and Lowe, 1988], GP interpolation [Krige, 1951, Rasmussen and Williams, 2005] (also called *kriging*), Support Vector Machines [Vapnik, 1999] and Artificial Neural Networks [McCulloch and Pitts, 1943]. Of the existing types of surrogate methods, GP interpolation is particularly appealing since it allows not only to predict the value of the surrogate function, but also provides an idea for the corresponding uncertainty. The next paragraphs detail the construction of a surrogate model via GP interpolation.

A GP $\mathcal{G}$ is an infinite set of random variables, of which any finite subset follows a joint Gaussian distribution. It is characterized by its mean function:

$$\mu(\mathcal{G}(\boldsymbol{x})) = \mathbb{E}[\mathcal{G}(\boldsymbol{x})] \tag{1.11}$$

and by its covariance function:

$$k(\boldsymbol{x},\boldsymbol{x}') = \text{CoVar}(\mathcal{G}(\boldsymbol{x}),\mathcal{G}(\boldsymbol{x}')) \tag{1.12}$$

where $\mu : \mathcal{X} \mapsto \mathbb{R}$, $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ and $\text{CoVar}(\cdot,\cdot)$ denotes the covariance between two random variables. Interpolation by GP assumes that the function we want to approximate is the realization of a GP. The goal is thus to parameterize the functions $\mu(\mathcal{G}(\boldsymbol{x}))$ and $k(\boldsymbol{x},\boldsymbol{x}')$ so that the obtained GP, when conditioned to the sampling data, best describes the objective function behavior. One way to do this is by using a Universal Kriging Model [Matheron, 1969, Rasmussen and Williams, 2005, Forrester et al., 2008, Picheny et al., 2010], where the GP is written as:

$$\mathcal{G} = \mu(\mathcal{G}(\boldsymbol{x})) + Z(\boldsymbol{x}) \quad \text{with} \quad Z(\boldsymbol{x}) \sim \mathcal{N}(0, k(\boldsymbol{x},\boldsymbol{x}')) \tag{1.13}$$

Assuming linear mean and stationary covariance function ($k(x,x') = k(\|\boldsymbol{x}-\boldsymbol{x}'\|)$), then $\mathcal{G}$ may be rewritten as:

$$\mathcal{G} = \boldsymbol{e}(\boldsymbol{x})^\intercal \beta + \sigma_Z^2 r_\theta(\boldsymbol{x},\boldsymbol{x}') \tag{1.14}$$

where $\boldsymbol{e}(\boldsymbol{x})$ are linearly independent basis functions, β are the associated coefficients, σ_Z^2 is the variance of Z and r_θ is a correlation function (also called kernel) which depends on the set of hyperparameters $\boldsymbol{\theta}$. To build the GP $\mathcal{G}$ it is thus necessary to make an assumption for the basis functions as well as the correlation function. This assumption typically considers the basis functions to be polynomial functions, whereas the correlation function is typically chosen as either an exponential kernel, a Gaussian kernel or a Matérn kernel. They are defined as follows:

– Exponential kernel:
$$r_\theta(\boldsymbol{x},\boldsymbol{x}') = \exp\left[-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_2}{\boldsymbol{\theta}}\right] \tag{1.15}$$

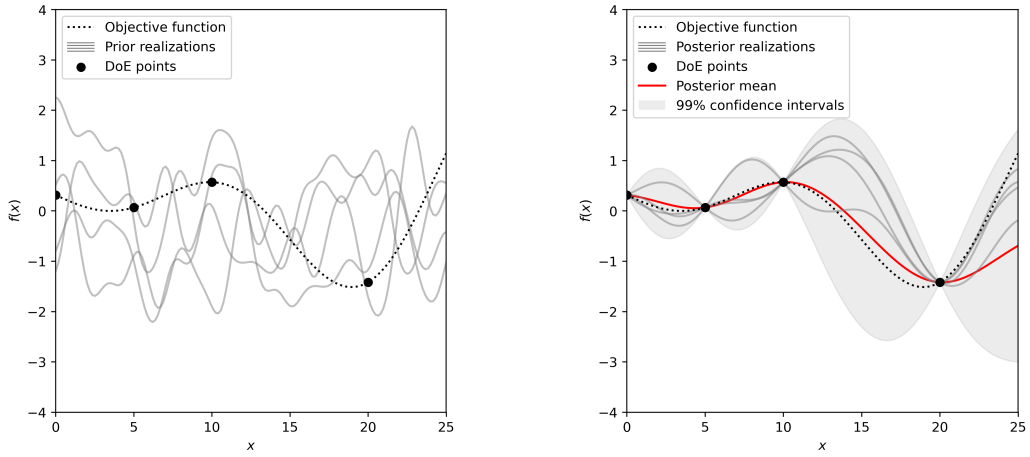
– Gaussian kernel:
$$r_\theta(\boldsymbol{x},\boldsymbol{x}') = \exp\left[-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_2^2}{2\boldsymbol{\theta}^2}\right] \tag{1.16}$$

– Matérn 3/2 kernel:
$$r_\theta(\boldsymbol{x},\boldsymbol{x}') = \left(1+\frac{\sqrt{3}\|\boldsymbol{x}-\boldsymbol{x}'\|_2}{\boldsymbol{\theta}}\right)\exp\left[-\frac{\sqrt{3}\|\boldsymbol{x}-\boldsymbol{x}'\|_2}{\boldsymbol{\theta}}\right] \tag{1.17}$$

– Matérn 5/2 kernel:
$$r_\theta(\boldsymbol{x},\boldsymbol{x}') = \left(1+\frac{\sqrt{5}\|\boldsymbol{x}-\boldsymbol{x}'\|_2}{\boldsymbol{\theta}}+\frac{5\|\boldsymbol{x}-\boldsymbol{x}'\|_2^2}{3\boldsymbol{\theta}^2}\right)\exp\left[-\frac{\sqrt{5}\|\boldsymbol{x}-\boldsymbol{x}'\|_2}{\boldsymbol{\theta}}\right] \tag{1.18}$$

Based on these assumptions for the mean and covariance functions, the initial GP $\mathcal{G}$ (also called prior) can be built. An illustration of the obtained prior GP $\mathcal{G}$ for a constant zero mean and Gaussian correlation function is given in Figure 1.2(a). As is shown, realizations drawn from the prior do not necessarily pass through the d DoE points. We can however exclude all realizations of $\mathcal{G}$ that do not interpolate the training points by conditioning $\mathcal{G}$ to the DoE:

$$\hat{\mathcal{G}}_c = \mathcal{G}|\{\boldsymbol{x}_1,...,\boldsymbol{x}_d\},\{f_{\text{obj}}^1,...,f_{\text{obj}}^d\} \tag{1.19}$$



(a) Prior GP $\mathcal{G}$ (b) Posterior GP $\hat{\mathcal{G}}_c^*$

Figure 1.2: Prior and posterior GP realizations. For the prior, a constant zero mean and Gaussian correlation function were assumed. The dotted black line depicts the true objective function while the four black dots are the DoE points. An illustration of the 99% confidence intervals (obtained at three times the standard deviation) for the posterior GP are equally drawn.

The result is a conditioned GP $\hat{\mathcal{G}}_c$ (also called posterior) defined by its mean and variance functions, respectively $\hat{\mu}_c(\boldsymbol{x})$ and $\hat{\sigma}_c^2(\boldsymbol{x})$. The mean and variance functions are defined at any point $\boldsymbol{x} \in \mathcal{X}$ thanks to the correlation matrix $\mathbf{R}$ of $\mathcal{G}$ at the DoE points:

$$\mathbf{R} = \begin{pmatrix} r_\theta(\boldsymbol{x}_1, \boldsymbol{x}_1) & \dots & r_\theta(\boldsymbol{x}_1, \boldsymbol{x}_d) \\ \vdots & \ddots & \vdots \\ r_\theta(\boldsymbol{x}_d, \boldsymbol{x}_1) & \dots & r_\theta(\boldsymbol{x}_d, \boldsymbol{x}_d) \end{pmatrix} \tag{1.20}$$

and the correlation vector $\boldsymbol{r}(\boldsymbol{x})$ between $\boldsymbol{x}$ and the DoE points:

$$\boldsymbol{r}(\boldsymbol{x}) = \{r_\theta(\boldsymbol{x}, \boldsymbol{x}_1), ..., r_\theta(\boldsymbol{x}, \boldsymbol{x}_d)\} \tag{1.21}$$

The mean and variance of the posterior GP $\hat{\mathcal{G}}_c$ for any point $\boldsymbol{x} \in \mathcal{X}$ are thus given by:

$$\hat{\mu}_c(\boldsymbol{x}) = \boldsymbol{e}(\boldsymbol{x})^\top \beta + \boldsymbol{r}(\boldsymbol{x})^\top \mathbf{R}^{-1}(\boldsymbol{f}_{\text{DoE}} - \mathbf{E}\beta) \tag{1.22}$$

$$\hat{\sigma}_c^2(\boldsymbol{x}) = \sigma_Z^2[1 - \boldsymbol{r}(\boldsymbol{x})^\top \mathbf{R}^{-1}\boldsymbol{r}(\boldsymbol{x}) + (\boldsymbol{e}(\boldsymbol{x}) - \boldsymbol{r}(\boldsymbol{x})^\top \mathbf{R}^{-1}\mathbf{E})(\mathbf{E}^\top \mathbf{R}^{-1}\mathbf{E})^{-1}(\boldsymbol{e}(\boldsymbol{x}) - \boldsymbol{r}(\boldsymbol{x})^\top \mathbf{R}^{-1}\mathbf{E})^\top] \tag{1.23}$$

where $\mathbf{E} = \{\boldsymbol{e}(\boldsymbol{x}_1), ..., \boldsymbol{e}(\boldsymbol{x}_d)\}$. In addition to defining the mean and variance functions of $\hat{\mathcal{G}}_c$ it is equally necessary to estimate the coefficients β and σ_Z^2, as well as the hyperparameters $\boldsymbol{\theta}$ that ensure that $\hat{\mathcal{G}}_c$ is likely to represent the objective function. This is done by maximizing the likelihood function $\mathcal{L}(\beta, \sigma_Z^2)$, written as:

$$\mathcal{L}(\beta, \sigma_Z^2) = \frac{1}{(2\pi\sigma_Z^2)^{d/2}|\mathbf{R}|^{1/2}} \exp\left[-\frac{(\boldsymbol{f}_{\text{DoE}} - \mathbf{E}\beta)^\top \mathbf{R}^{-1}(\boldsymbol{f}_{\text{DoE}} - \mathbf{E}\beta)}{2\sigma_Z^2}\right] \tag{1.24}$$

The coefficients β^* and $\{\sigma_Z^2\}^*$ that maximize this likelihood can be obtained analytically [Jones, 2001, Rasmussen and Williams, 2005] as follows:

$$\begin{aligned} \beta^* &= (\mathbf{E}^\top \mathbf{R}^{-1}\mathbf{E})^{-1}(\mathbf{E}^\top \mathbf{R}^{-1}\boldsymbol{f}_{\text{DoE}}) \\ \{\sigma_Z^2\}^* &= \frac{1}{d}(\boldsymbol{f}_{\text{DoE}} - \mathbf{E}\beta^*)^\top \mathbf{R}^{-1}(\boldsymbol{f}_{\text{DoE}} - \mathbf{E}\beta^*) \end{aligned} \tag{1.25}$$

The optimal hyperparameters $\boldsymbol{\theta}^*$, however, cannot be found analytically so we typically rely on local optimization methods. Alternatively, other authors propose to find the optimal hyperparameters using the leave-one-out mean-square error criterion [Bachoc, 2013]. Figure 1.2(b) illustrates the posterior GP $\hat{\mathcal{G}}_c^*$ obtained after optimization of the hyperparameters. As shown, realizations of $\hat{\mathcal{G}}_c^*$ interpolate the objective function at the DoE points. Moreover, thanks to the information provided by the variance $\hat{\sigma}_c^2(\boldsymbol{x})$, it is possible to define a confidence interval of where the true objective function is likely to belong. In the figure, the confidence intervals obtained at three times the standard deviation ($\hat{\mu}_c(\boldsymbol{x}) \pm 3\hat{\sigma}_c(\boldsymbol{x})$) are denoted 99% confidence intervals. However, in practice, the actual probability is unknown as the value of 99% relies on the hypothesis that f_{obj} is the realization of a GP whose mean and covariance function are known. In the following we describe how surrogate-based algorithms exploit the obtained model of the objective function to solve the optimization problem.

1.1.2.3 Exploitation of the surrogate model

Different approaches can be used to exploit the obtained objective function model. A first, seemingly logical option is to replace the real objective function by its surrogate directly in the optimization problem:

$$\hat{\boldsymbol{x}}^* = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \hat{\mu}_c(\boldsymbol{x}) \tag{1.26}$$

Because calls to the surrogate are inexpensive, Eq. (1.26) can be solved using local optimization methods combined with multi-start strategies. However, to find the true objective function optimum, $\hat{\mu}_c(\boldsymbol{x})$ must be sufficiently accurate throughout the entire design space. Thus, even when space-filling strategies are used for the generation of the initial DoE, many points may be required to obtain a globally accurate surrogate. When the objective function is costly to evaluate or the input space is high-dimensional, this may not be a feasible option. As a result, $\hat{\mu}(\boldsymbol{x})$ is typically built using only a few points, hoping that the solution obtained based on the surrogate approximation will lie not too far from the global optimum. An illustration of this approach is given in Figure 1.3.

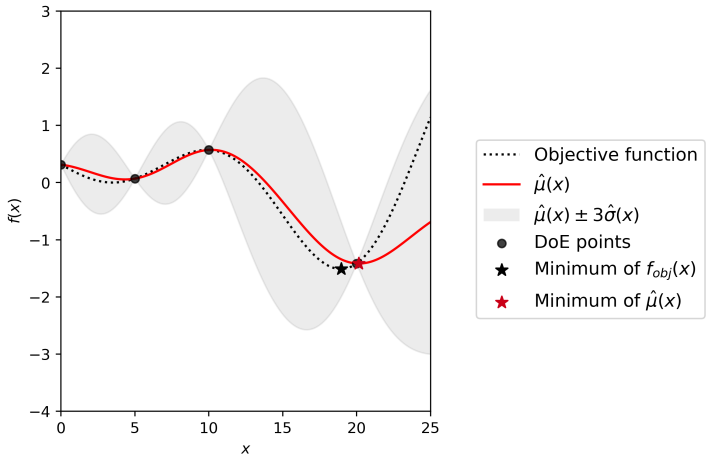


Figure 1.3: Illustration of the direct optimization of the objective function surrogate $\hat{\mu}_c(\boldsymbol{x})$, obtained from a DoE of four points. Comparison of the optimum found (red star) with the global optimum of the true objective function (black star).

Figure 1.3 shows that, when replacing the objective function by its surrogate, it is possible to find an optimal point which lies close to the true global optimum, but typically does not coin-

cide with it. To address this issue, other approaches propose to enrich the objective function surrogate by adaptively adding selected points to the DoE.

Surrogate Based Optimization (SBO). Surrogate based optimization proposes to add the current optimum, i.e. the solution to Eq. (1.26), to the DoE. By successively adding new points, this approach can be expected to converge to the true global optimum. Although SBO performs better than the direct optimization of the surrogate, it still requires an initial approximation that accurately captures the overall behavior of the objective function. Otherwise, SBO can get stuck placing new points around local optima, rather than exploring the regions around the global optimum. Figure 1.4 illustrates this scenario, which occurs due to a poorly distributed initial DoE.

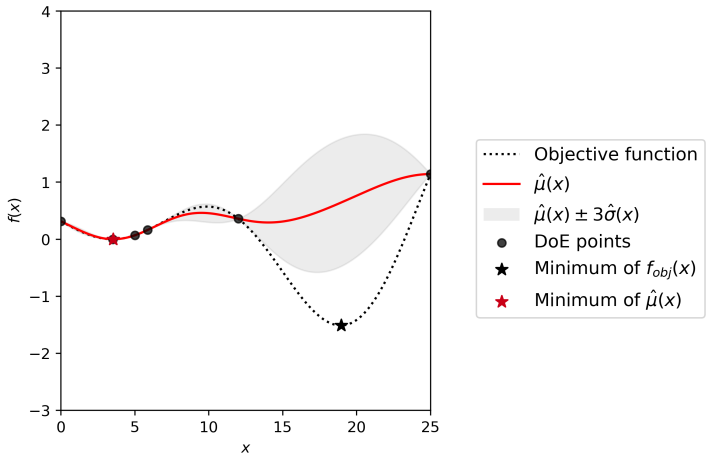


Figure 1.4: Illustration of the SBO approach using a poor initial DoE of four points. Comparison of the optimum found after two surrogate enrichments (red star) with the global optimum of the true objective function (black star).

Efficient Global Optimization (EGO). To attenuate the influence of the initial surrogate on the solution found, the EGO approach [Jones et al., 1998] proposes to account for the uncertainty of the surrogate approximation when selecting new points to add to the DoE. To do so, EGO replaces the optimization problem defined in Eq. (1.26) by the optimization of the Expected Improvement (EI) function. The EI is an acquisition function which attempts to maximize the improvement of the current best objective function value by making a compromise between the exploitation of the current minimum of the surrogate and the exploration of the regions of the design space where the uncertainty is high. The improvement is defined as:

$$\mathscr{I}(\boldsymbol{x}) = \max(f_{\min} - \hat{\mathscr{G}}_c^*(\boldsymbol{x}), 0) \tag{1.27}$$

where $f_{\min}$ denotes the current objective function minimum, chosen from within the DoE points. Note that $\mathscr{I}(\boldsymbol{x})$ is a random variable because $\hat{\mathscr{G}}_c^*(\boldsymbol{x})$ is a Gaussian random variable characterized by its mean and variance: $\hat{\mathscr{G}}_c^*(\boldsymbol{x}) \sim \mathcal{N}(\hat{\mu}_c(\boldsymbol{x}), \hat{\sigma}_c^2(\boldsymbol{x}))$. The likelihood of achieving a positive improvement is given by the normal probability density function, accordingly:

$$\mathscr{L}(\mathscr{I}(\boldsymbol{x}) > 0) = \frac{1}{\sqrt{2\pi}\hat{\sigma}_c(\boldsymbol{x})} \exp\left[-\frac{(f_{\min} - \mathscr{I}(\boldsymbol{x}) - \hat{\mu}_c(\boldsymbol{x}))^2}{2\hat{\sigma}_c^2(\boldsymbol{x})}\right] \tag{1.28}$$

Integrating over this density allows to obtain the EI as the the expected value of $\mathscr{I}(\boldsymbol{x})$, at any point $\boldsymbol{x} \in \mathscr{X}$. The EI is written as:

$$\mathbb{E}[\mathscr{I}(\boldsymbol{x})] = \begin{cases} (f_{\min} - \hat{\mu}_c(\boldsymbol{x}))\Phi\left(\frac{f_{\min}-\hat{\mu}_c(\boldsymbol{x})}{\hat{\sigma}_c(\boldsymbol{x})}\right) + \hat{\sigma}_c(\boldsymbol{x})\phi\left(\frac{f_{\min}-\hat{\mu}_c(\boldsymbol{x})}{\hat{\sigma}_c(\boldsymbol{x})}\right) & \text{if } \hat{\sigma}_c(\boldsymbol{x}) > 0 \\ 0 & \text{if } \hat{\sigma}_c(\boldsymbol{x}) = 0 \end{cases} \tag{1.29}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are, respectively, the Cumulative Distribution Function (CDF) and the Probability Density Function (PDF) of the standard normal distribution $\mathscr{N}(0,1)$.

Because the analytical derivative of the EI function is available, Eq. (1.29) can be maximized using both gradient-based or gradient-free approaches, combined with multi-start strategies to find the new point to add to the DoE. By iteratively adding new points, the EGO algorithm can be expected to converge to the global optimum even if the initial DoE is poorly distributed. An illustration of the EGO algorithm is provided in Figure 1.5 where the initial DoE is the one used in Figure 1.4. The main steps of the EGO approach are summarized in Algorithm 3.

Algorithm 3 Efficient Global Optimization (EGO)

Input: $\boldsymbol{x}_{\text{DoE}}, f_{\text{DoE}}, k_{\max}$ ▷ Initial DoE and max. number of iterations
  $k = 0$ ▷ Initialize iteration count
  while $k < k_{\max}$ **do**
    Build $\hat{f}_{\text{obj}}(\boldsymbol{x})$ ▷ Objective function surrogate
    Find $\boldsymbol{x}_{\text{new}}$ ▷ Optimize acquisition function
    Add $\boldsymbol{x}_{\text{new}}$ and $f_{\text{obj}}(\boldsymbol{x}_{\text{new}})$ to the DoE
    $f_{\min} = \min f_{\text{DoE}}$ ▷ Update current best point
    $k = k + 1$ ▷ Increase iteration count
  end while
  $\boldsymbol{x}_{\min} = \arg f_{\min}$
Output: $\boldsymbol{x}_{\min}, f_{\min}$

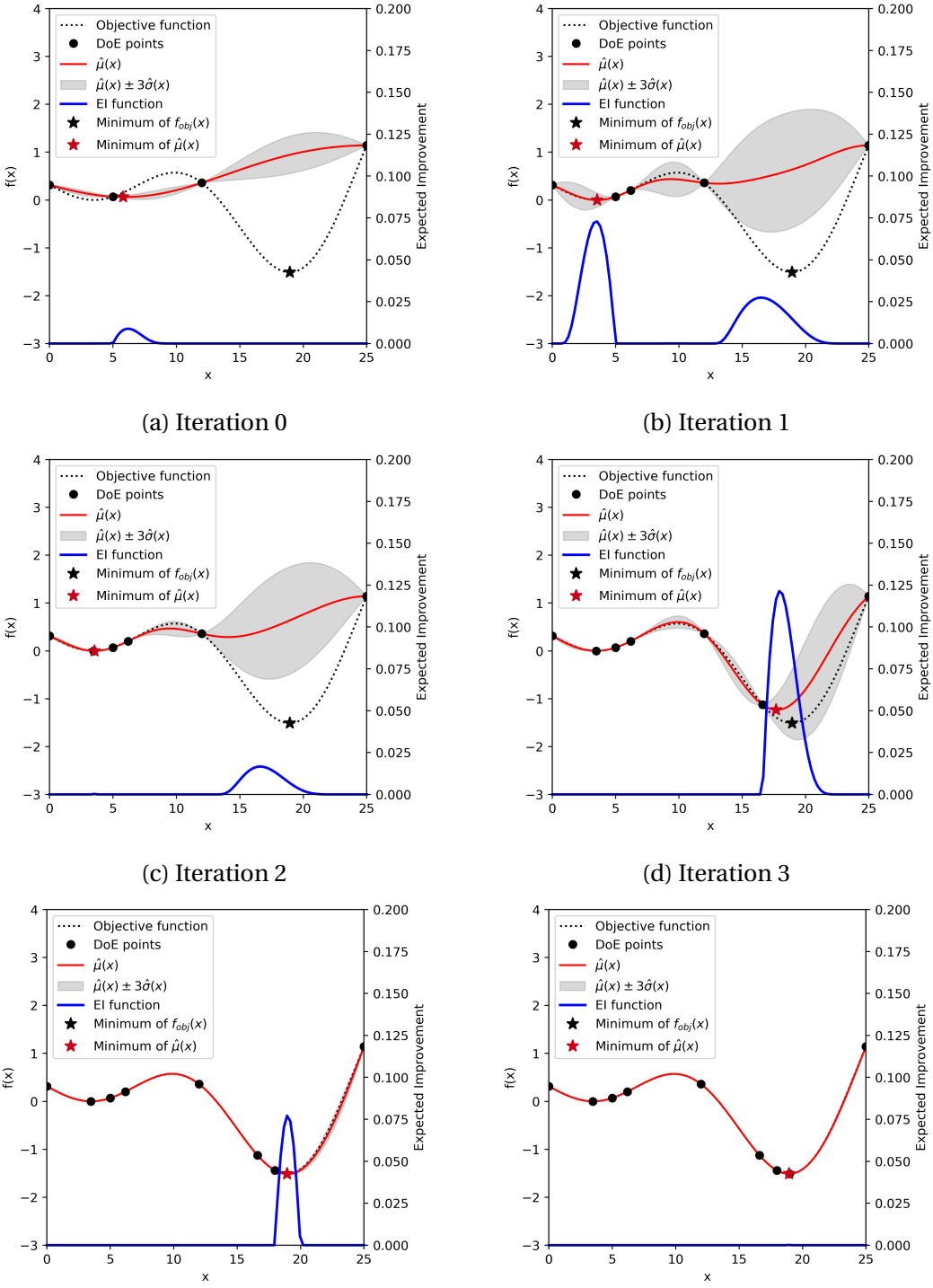
1.1.2.4 The acquisition function

Although the EI function performs rather well in the provided example, having successfully converged to the global optimum after five iterations, some remarks should still be made. Namely, it should be noted that the maximization of the EI function often presents a challenge. Indeed, from Eq. (1.29) we can see that the EI is zero at the DoE points and positive elsewhere, resulting in several local maxima. Local optimizers struggle to optimize functions with this type of behavior and, even when combined with multi-start strategies, they may fail to find the global optimum. To address this issue other acquisition functions have been proposed in the literature.

The Upper Confidence Bound (UCB) acquisition function. Like the EI function, the UCB acquisition function [Auer, 2002, Auer et al., 2002] attempts to find a compromise between the exploitation of the current surrogate and the exploration of the design space. It directly balances the surrogate mean and standard deviation accordingly:

$$\text{UCB}(\boldsymbol{x}) = -\hat{\mu}_c(\boldsymbol{x}) + \lambda\hat{\sigma}_c(\boldsymbol{x}) \tag{1.30}$$

where $\lambda \in \mathbb{R}^+$ is chosen depending on the level of exploration allowed.



(a) Iteration 0

(b) Iteration 1

(c) Iteration 2

(d) Iteration 3

(e) Iteration 4

(f) Iteration 5

Figure 1.5: Illustration of the EGO approach, starting from a poor initial DoE of four points. In each iteration, the DoE is enriched at the point that maximizes the EI function, shown in blue. The best value found (red star) is compared with the true objective function optimum (black star) at each iteration.

The Watson and Barnes 2 (WB2) acquisition function. The WB2 acquisition function [Watson and Barnes, 1995] was specifically developed to get rid of the near-zero flat zones of the EI function and thus ease the optimization problem. It is written as:

$$\text{WB2}(\boldsymbol{x}) = \mathbb{E}[\mathscr{I}(\boldsymbol{x})] - \hat{\mu}_c(\boldsymbol{x}) \tag{1.31}$$

By subtracting the posterior mean $\hat{\mu}_c(\boldsymbol{x})$ to the EI, the WB2 function essentially follows the mean surrogate behavior where the EI would otherwise be a near-zero plateau. The result is a less multimodal criterion, easier to optimize. Regardless, the WB2 function does not come without its disadvantages. Indeed, the WB2 criterion can significantly favor the exploitation of the surrogate if the order of magnitude of $\hat{\mu}_c(\boldsymbol{x})$ is much greater than that of the EI. Note that this is bound to occur since the EI tends to zero as the iterations progress (see Figure 1.5(f)). An illustration of the WB2 criterion at the third iteration of the EGO algorithm is provided in Figure 1.6(a).

The Watson and Barnes 2 scaled (WB2S) acquisition function. The WB2S acquisition function [Bartoli et al., 2019] avoids the excessive exploitation that can occur for the WB2 function by applying a non-negative scaling factor to the EI, as follows:

$$\text{WB2S}(\boldsymbol{x}) = s\mathbb{E}[\mathscr{I}(\boldsymbol{x})] - \hat{\mu}_c(\boldsymbol{x}) \tag{1.32}$$

where s is the scaling factor, defined as:

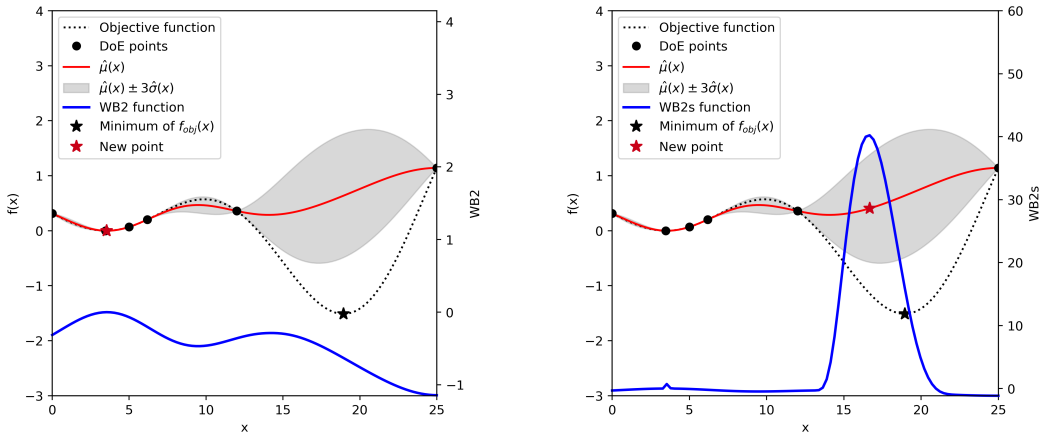
$$s = \begin{cases} \beta \frac{|\hat{\mu}_c(\boldsymbol{x}_{\text{maxEI}})|}{\mathbb{E}[I(\boldsymbol{x}_{\text{maxEI}})]} & \text{if } \mathbb{E}[I(\boldsymbol{x}_{\text{maxEI}})] > 0 \\ 1 & \text{if } \mathbb{E}[I(\boldsymbol{x}_{\text{maxEI}})] = 0 \end{cases} \tag{1.33}$$

where $\beta > 1$ is chosen as a sufficiently large coefficient (typically $\beta = 100$) and $\boldsymbol{x}_{\text{maxEI}}$ is the value that maximizes the EI within the design space. In practice, $\boldsymbol{x}_{\text{maxEI}}$ is approximated by sampling in the design space (for example via LHS) and choosing $\boldsymbol{x}_{\text{maxEI}}$ as the sampled point with the highest EI value. The sampled points are then used as starting points for the multi-start strategy employed during the optimization of the WB2S criterion. An illustration of the WB2S criterion at the third iteration of the EGO algorithm is provided in Figure 1.6(b).

1.1.2.5 Disadvantages of surrogate-based optimization

Global optimization algorithms are more fit for the optimization of multimodal functions as they attempt to search directly for the global optimum. Moreover, the presented global optimization algorithms rely on a surrogate of the objective function to perform the search. This reduces the cost of the optimization problem by limiting expensive function calls to interesting regions of the design space. Nevertheless, global optimization algorithms present some limitations. Namely, the choice of an appropriate stopping criterion for global optimization algorithms is not straightforward. One idea might be to stop the algorithm when the maximum of the EI function is below a given threshold. However, due to the numerous near-zero flat zones, this criterion is typically not very reliable. As a result, it is usually preferred to fix a computational budget and to stop the algorithm once all the budget has been spent. Another idea is to combine the global search with local steps, which can be used to check for convergence [Regis, 2016, Diouane et al., 2023].

Surrogate modeling also suffers from the curse of dimensionality. Indeed, the computational cost required to invert the covariance matrix increases significantly with the increase in the number of sampled points. Moreover, the number of hyperparameters to optimize scales with the dimension of the input space. Some works address this challenge by looking for a lower dimensional representation of the input space [Garnett et al., 2013, Bouhlel et al., 2016, Zhang et al., 2019, Gaudrie et al., 2020]. By building the covariance matrix in this lower dimensional space, the computational cost can be reduced.



(a) WB2 criterion. (b) WB2S criterion.

Figure 1.6: Illustration of (a) the WB2 and (b) the WB2S acquisition functions for the third iteration of the EGO algorithm. Note the excessive exploitation given by the WB2 criterion, which does not occur for the WB2S criterion.

1.1.3 Constrained optimization

The previous sections described the existing methods to perform unconstrained minimization. However, optimization problems are often subject to constraints. This section briefly addresses some of the existing methods to handle constraints in optimization problems. To do so, we define a general constrained optimization problem as:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathscr{X}} \; & f_{\text{obj}}(\boldsymbol{x}) \\ \text{s.t.} \; & h_i(\boldsymbol{x}) = 0, \; i = 1,...,n_h \\ & g_j(\boldsymbol{x}) \leq 0, \; j = 1,...,n_g \end{aligned} \tag{1.34}$$

where $h_i : \mathbb{R}^n \mapsto \mathbb{R}$ denotes the i^{th} equality constraint and $g_j : \mathbb{R}^n \mapsto \mathbb{R}$ denotes the j^{th} inequality constraint. If the problem is well posed, there exists a feasible region $\mathscr{X}_f \subset \mathscr{X}$ where, for all $\boldsymbol{x} \in \mathscr{X}_f$, both equality and inequality constraints are verified. It is the job of constrained optimization algorithms to find the minimum attainable value of the objective function within the feasible region. A brief review of these algorithms is presented in this section.

Optimality conditions. As for the unconstrained case, we first present the optimality conditions which ensure that $\boldsymbol{x}^*$ is a constrained local optimum. To do so, we define the Lagrangian function $\mathscr{L} : \mathbb{R}^n \mapsto \mathbb{R}$, as:

$$\mathscr{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\sigma}) = f_{\text{obj}}(\boldsymbol{x}) + \sum_{i=1}^{n_h} \lambda_i h_i(\boldsymbol{x}) + \sum_{j=1}^{n_g} \sigma_j g_j(\boldsymbol{x}) \tag{1.35}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{n_h}$ is the vector of Lagrange multipliers associated with the equality constraints and $\boldsymbol{\sigma} \in (\mathbb{R}_0^+)^{n_g}$ is the vector of Lagrange multipliers associated with the inequality constraints. If the objective and constraint functions are continuously differentiable, $\boldsymbol{x}^*$ is a constrained local optimum, if:

$$\nabla \mathscr{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\sigma}^*) = 0 \tag{1.36}$$

In addition, $\boldsymbol{x}^*$ should belong to $\mathscr{X}_f$, which means:

$$h_i(\boldsymbol{x}^*) = 0,\ i = 1,...,n_h \tag{1.37}$$

$$g_j(\boldsymbol{x}^*) \leq 0,\ j = 1,...,n_g \tag{1.38}$$

To ensure that there is no feasible descent direction, we impose:

$$\sigma_j^* \geq 0,\ j = 1,...,n_g \tag{1.39}$$

Finally, if a given inequality constraint is active at $\boldsymbol{x}^*$, then $g_j(\boldsymbol{x}^*) = 0$, otherwise the constraint is not active and $\sigma_j^* = 0$. This is written as:

$$\sigma_j^* g_j(\boldsymbol{x}^*) = 0,\ j = 1,...,n_g \tag{1.40}$$

Equations (1.36) through (1.40) are known as the Karush-Kuhn-Tucker (KKT) optimality conditions [Boyd and Vandenberghe, 2004]. Local constrained optimization algorithms attempt to find the constrained local optimum such that the KKT conditions are verified. Some of these algorithms are presented in the following.

1.1.3.1 Penalty methods

A first class of algorithms uses penalty methods. These methods approximate the constrained problem of Eq. (1.34) as an unconstrained problem by adding a penalty to the objective function if any of the constraints are violated. The resulting unconstrained minimization problem is of the form:

$$\hat{\boldsymbol{x}}^* = \arg\min_{\boldsymbol{x}\in\mathscr{X}} f_{\text{obj}}(\boldsymbol{x}) + \lambda_h \sum_{i=1}^{n_h} h_i(\boldsymbol{x})^2 + \lambda_g \sum_{j=1}^{n_g} \max(0, g_j(\boldsymbol{x}))^2 \tag{1.41}$$

where $\lambda_h \in \mathbb{R}^+$ and $\lambda_g \in \mathbb{R}^+$ are scalar parameters defining the severity of the penalization, n_h is the number of equality constraints and n_g is the number of inequality constraints. In theory, as λ_h and λ_g tend to infinity, the solution to Eq. (1.41) tends to the true solution of the constrained problem. However, using large values of λ_h and λ_g can lead to significant curvature changes close to the borders of the feasible region, making it harder to find the true optimum. Due to this issue, other methods are typically preferred for the optimization of continuous functions. Regardless, penalty methods are easy to implement and can be solved using any of the previously described approaches.

1.1.3.2 Sequential Quadratic Programming

Sequential Quadratic Programming (SQP) [Boggs and Tolle, 2000, Gill and Wong, 2012] methods are an extension of the Newton method, described in Section 1.1.1.1, to constrained optimization. They build a local quadratic approximation of the objective function and a linear approximation of the constraints. Then, starting from an initial guess $\boldsymbol{x}_0, \boldsymbol{\lambda}_0, \boldsymbol{\sigma}_0$ and by attempting to bring $\nabla\mathscr{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\sigma})$ close to zero, SQP methods are able to find a search direction and step size used to generate a series of points $\boldsymbol{x}_1, ..., \boldsymbol{x}_k$ that converge to a constrained local optimum. Examples of SQP methods include the Sequential Least Squares Programming (SLSQP) algorithm [Kraft, 1988] as well as the Sparse Nonlinear Optimizer (SNOPT) algorithm [Gill et al., 2002].

Limitations of SQP methods. SQP methods present the same disadvantages of gradient-based optimizers. Namely, if multiple constrained minima exist, it is necessary to combine these methods with multi-start strategies to find the constrained global optimum. Moreover, if in addition to the objective function, $h_i(\boldsymbol{x})$ and $g_j(\boldsymbol{x})$ are also computed using expensive numerical solvers, for which derivative information is unavailable, the computational burden required to approximate $\nabla\mathscr{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\sigma})$ using finite differences can quickly become intractable.

1.1.3.3 Constrained Bayesian optimization

Constrained Bayesian optimization methods are extensions of the surrogate-based optimization algorithms presented in Section 1.1.2 to constrained problems. These methods build GP approximations of the constraint functions which are accounted for when selecting new points to add to the DoE. Typically, this means adapting the acquisition function, so that computational effort is mostly spent on feasible points or close to the borders of the feasible region. Some works propose to combine the EI criterion with feasibility indicators [Schonlau et al., 1998, Sasena et al., 2002, Parr et al., 2012], while others propose the use of new acquisition functions [Picheny, 2014, Hernández-Lobato et al., 2015].

Limitations of constrained Bayesian optimization. Constrained Bayesian optimization algorithms present the same disadvantages listed for global optimization algorithms. Namely, it remains difficult to establish an appropriate stopping criterion and a maximum number of iterations is typically used. Because they rely on surrogate models, constrained Bayesian optimization algorithms also suffer from the curse of dimensionality. Extending the constrained Bayesian framework to high-dimensional input spaces has been the focus of several recent works [Bouhlel et al., 2018, Eriksson and Poloczek, 2021]. Another disadvantage of constrained Bayesian optimization algorithms is that they can perform poorly if the initial constraint surrogates are too inaccurate. Indeed, this can lead to an initial constraint surrogate where there is no feasible set. Some works overcome this issue by considering the uncertainty of the constraint functions when sampling new points [Feliot et al., 2017, Priem et al., 2020a, Eriksson and Poloczek, 2021]. Finally, constraint handling strategies have also been developed in the context of robust Bayesian optimization (i.e. where some input parameters defining the objective or constraint functions are uncertain) [El Amri et al., 2023, Pelamatti et al., 2024].

1.1.4 Summary of numerical optimization

In this section we introduced the main tools to perform the optimization of a given performance function. Local optimization algorithms which use gradient information to drive the search for the local optimum were introduced first. Despite their efficiency, however, these algorithms are not always appropriate for the optimization of multimodal functions as they tend to get stuck on local minima. Moreover, these algorithms can lead to a significant number of function calls if gradient information is not available. This is typically undesirable if the evaluation of the performance function makes use of expensive numerical solvers. Global optimization algorithms, which make use of surrogate models to search directly for the global optimum were introduced next. These algorithms attempt to limit function calls to relevant regions of the design space, making them more appropriate for handling costly, multimodal performance functions. Finally, extension of both local and global optimization algorithms to constrained optimization problems was briefly discussed. In the next section we will focus on the multidisciplinary nature of the system and address the existing strategies to find the solution of the MDA and solve MDO problems.

1.2 Multidisciplinary design analysis and optimization

In the previous section we introduced the existing methods for the single discipline optimization of a given objective function, subject or not to constraints. Multidisciplinary Design Optimization (MDO) differs from single discipline optimization, because it accounts for the interactions between the different disciplines of the system. As a result, objective and constraint functions

depend not only on the design variables, but also on the quantities exchanged between the disciplinary solvers, called coupling variables. The MDO problem can then be written as:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathscr{X}} \{f_{\text{obj}}(\boldsymbol{x}, \boldsymbol{y}_c^*(\boldsymbol{x})) \text{ s.t. } \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}_c^*(\boldsymbol{x})) \leq 0 \text{ and } \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}_c^*(\boldsymbol{x})) = 0\} \tag{1.42}$$

where $\boldsymbol{y}_c^*(\boldsymbol{x})$ is the vector of converged coupling variables, found by solving the following nonlinear system of equations (also called MDA):

$$\boldsymbol{y}_i = f_i(\boldsymbol{x}, \boldsymbol{y}_{c_i}),\ i = 1, ..., n_d \quad \forall \boldsymbol{x} \in \mathscr{X} \tag{1.43}$$

where $\boldsymbol{y}_i \in \mathscr{C}^i \subset \mathbb{R}^{d^i}$ is the output of a given numerical solver f_i modeling discipline i, $\boldsymbol{y}_{c_i}$ is the vector of coupling variables that influence the value of $\boldsymbol{y}_i$ and c_i is a subset of $\{1, ..., n_d\} \setminus \{i\}$. It is assumed in this work that Eq. (1.43) has a unique solution for all $\boldsymbol{x}$ in $\mathscr{X}$. Depending on the choice of MDO formulation, direct resolution of the MDA may be required. In this section we first introduce the existing approaches to find the solution of the MDA. Then, an overview of the different formulations to solve the MDO problem is provided. Finally, the EGMDO algorithm for the global optimization of multidisciplinary systems is described.

The eXtended Design Structure Matrix (XDSM). In this section, XDSM [Lambe and Martins, 2012] diagrams are used to illustrate some of the presented approaches. The XDSM is a visualization tool which describes data exchanges between the different system components and provides an overview of the underlying process. To do so, disciplines and drivers are placed on the main diagonal, while inputs and outputs are placed off-diagonal, such that inputs of a given component are placed in the same column, while outputs are placed in the same row. Moreover, user inputs, initial guesses and process outputs are placed on the outer edges of the diagram. Finally, gray lines are used to identify connections between the different components. An example of an XDSM for the optimization of an aircraft wing is provided in Figure 1.7.

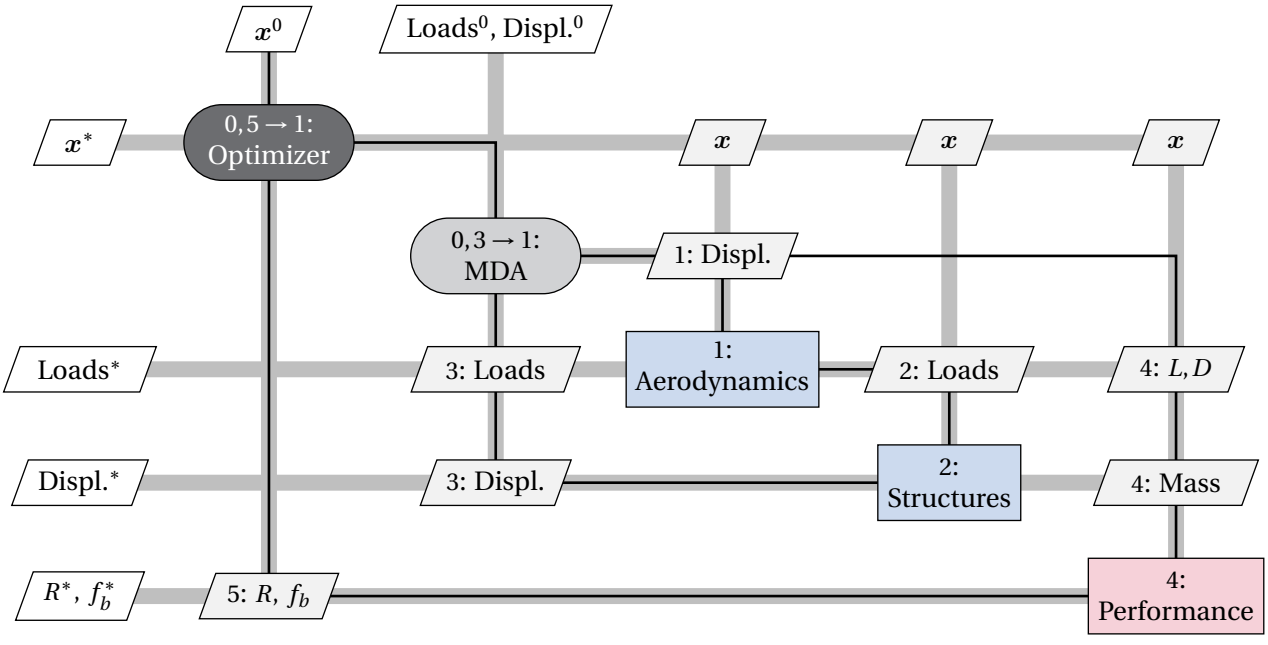


Figure 1.7: Example of an XDSM diagram for the optimization of an aircraft wing.

1.2.1 Illustrative 1-D analytical MDO problem

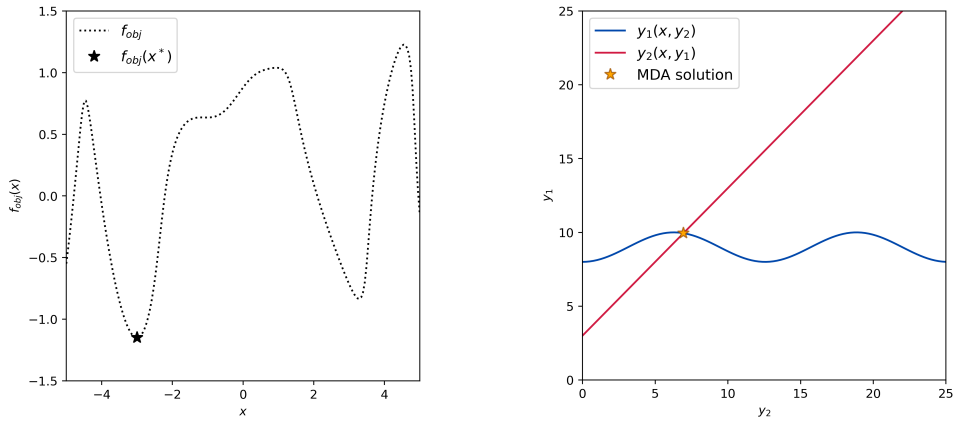
To aid in the visualization of the different approaches introduced in this section, an unconstrained analytical MDO problem will be used. It considers one scalar design variable $x \in \mathcal{X} = [-5,5]$ and two scalar coupling variables $y_1, y_2 \in \mathbb{R}$. The objective function is defined as:

$$f_{\text{obj}}(x, y_1, y_2) = \cos\left(\frac{y_1 + \exp(-y_2)}{\pi}\right) + \frac{x}{20} \tag{1.44}$$

where y_1 and y_2 are the solution of the following non-linear system of equations:

$$\begin{cases} y_1(x, y_2) = x^2 - \cos(\frac{y_2}{2}) \\ y_2(x, y_1) = x + y_1 \end{cases} \tag{1.45}$$

Figure 1.8(a) plots the objective function and identifies the unconstrained global optimum, while Figure 1.8(b) shows the MDA solution at the identified global optimum, obtained for $x^* \approx -3$.



(a) Objective function.

(b) MDA solution at $x^* \approx -3$.

Figure 1.8: 1-D analytical MDO problem. (a) Objective function and corresponding global optimum (black star). (b) Disciplinary solvers and MDA solution (yellow star) at $x^* = -3$.

1.2.2 Monolithic approaches

One option to solve the MDA is to use monolithic approaches [Hübner et al., 2004, Bathe and Zhang, 2004]. In this type of approach, a single system of equations describing all disciplinary behaviors is solved by the same numerical solver, capable of modeling the interactions between the different physical phenomena. The main advantage of using such monolithic solvers is that the interface between the different physical models is directly included in the analysis. Nevertheless, these approaches show little flexibility, making it difficult to account for additional disciplines or to replace the physical model of a given discipline (for instance to change between different fidelity levels). As a result, modular or partitioned approaches are more often used in MDA. They are described in the following.

1.2.3 Partitioned approaches

Partitioned approaches [Quarteroni and Valli, 1999, Felippa et al., 2001, Degroote, 2013, Gervasio and Quarteroni, 2018] rely on the fact that specialized disciplinary solvers which efficiently solve the corresponding physical equations already exist. They thus partition the system into

subdomains, each handled by a dedicated disciplinary solver. The different disciplinary solvers are then allowed to interact by exchanging information at a common interface. To illustrate this concept, we introduce the domain $\Omega \in \mathbb{R}^d$, with $d \in \{2,3\}$, the two non-overlapping subdomains Ω_1, Ω_2, such that $\Omega = \Omega_1 \cup \Omega_2$, and their common interface $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$. The resulting coupled system is of the form:

$$\begin{aligned}
&\boldsymbol{y}_1 = \mathscr{M}_1(\boldsymbol{x}, \boldsymbol{y}_2) && \text{in } \Omega_1 && (1.46)\\
&\boldsymbol{y}_2 = \mathscr{M}_2(\boldsymbol{x}, \boldsymbol{y}_1) && \text{in } \Omega_2 && (1.47)\\
&\Phi_1(\boldsymbol{y}_1) = \Phi_2(\boldsymbol{y}_2) && \text{on } \Gamma && (1.48)\\
&\Psi_1(\boldsymbol{y}_1) + \Psi_2(\boldsymbol{y}_2) = 0 && \text{on } \Gamma && (1.49)\\
&\text{Boundary conditions} && \text{on } \partial\Omega && (1.50)
\end{aligned}$$

where $\mathscr{M}_1, \mathscr{M}_2$ denote the disciplinary models, Φ_1, Φ_2 are functions used to model Dirichlet-like conditions and Ψ_1, Ψ_2 are functions used to model Neumann-like conditions. For instance, in Fluid-Structure Interaction (FSI) problems it is common to partition the system into a fluid domain and a structural domain. In that case, $\mathscr{M}_1$ and $\mathscr{M}_2$ denote the fluid and structural governing equations, Φ_1, Φ_2 define the no-slip condition across the fluid-structure interface and Ψ_1, Ψ_2 denote the continuity of normal fluxes across the fluid-structure interface.

The main advantage of partitioned approaches is that they can be easily used in the context of black-box disciplinary solvers. However, because they do not directly model the interface between the defined subdomains, a coupling algorithm is required to iteratively propagate information between the disciplinary solvers until convergence is achieved. Moreover, if the different disciplines use different spatial or temporal discretizations of the corresponding subdomains, an additional operation to interpolate across Γ is needed. For instance, in FSI problems, the aerodynamics discipline will typically use a finer grid at the fluid-structure interface than the structural discipline. As a result, an additional operation, which allows the transfer of loads and displacements between the non-coincidental grids is required. Different strategies exist in the literature to perform interpolation across the common interface, including nearest-neighbor interpolation, projection methods [Cebral and Lohner, 1997, Farhat et al., 1998] and Radial Basis Function (RBF) interpolation [Rendall and Allen, 2008].

1.2.3.1 Fixed-point methods for MDA

In the previous section we defined the coupled system arising from the decomposition of the system domain into several subdomains. In this section we assume that appropriate disciplinary solvers have been chosen to model the physical phenomena associated with each subdomain and instead focus on the available means to find the solution of the MDA. Typically, fixed-point iterative methods are used, the simplest of which is the nonlinear Jacobi method.

Convergence of fixed-point algorithms. Fixed-point methods obtain new solutions based on previously computed ones. The value of the coupling variable $\boldsymbol{y}_i^{k+1}$ at iteration $k+1$ is written as:

$$\boldsymbol{y}_i^{k+1} = \mathrm{G}(\boldsymbol{y}_i^k), \;\; i = 1, ..., n_d \tag{1.51}$$

If G defines a contraction on $\mathscr{D} \subset \prod_{i=1}^{n_d} \mathscr{C}^i$, i.e. if:

$$\|\mathrm{G}(\boldsymbol{y}_i^{k+1}) - \mathrm{G}(\boldsymbol{y}_i^k)\| \leq \alpha \|\boldsymbol{y}_i^{k+1} - \boldsymbol{y}_i^k\|, \;\; \alpha \in [0,1[\tag{1.52}$$

then there exists a unique fixed-point $\boldsymbol{y}_i^* \in \mathscr{D}$, such that $\boldsymbol{y}_i^* = \mathrm{G}(\boldsymbol{y}_i^*)$, $i = 1, ..., n_d$ and any series of points $\boldsymbol{y}_i^0, \boldsymbol{y}_i^1, ...$ converges to this point, for any initial guess $\boldsymbol{y}_i^0 \in \mathscr{D}$. In this work, we suppose that the fixed-point algorithms used to solve the MDA always verify Eq. (1.52).

The nonlinear Jacobi method. The nonlinear Jacobi method, illustrated in Figure 1.9, takes as input an initial guess for the values of each coupling variable. Then, it computes the solution of each discipline using the provided initial guesses and updates the coupling variables to the newly obtained values. By iteratively repeating this process until the coupling variable values do not significantly change throughout the iterations, the solution of the MDA can be found. The main advantage of the Jacobi method is that computations are made using values from the previous iteration, thus allowing to run the different disciplinary solvers in parallel. Moreover, it is a non-intrusive approach as it uses only inputs and outputs of the disciplinary solvers. The main steps of the nonlinear Jacobi method are presented in Algorithm 4, where ϵ_{MDA} denotes a chosen tolerance for the relative change in the coupling variable values between two successive iterations.

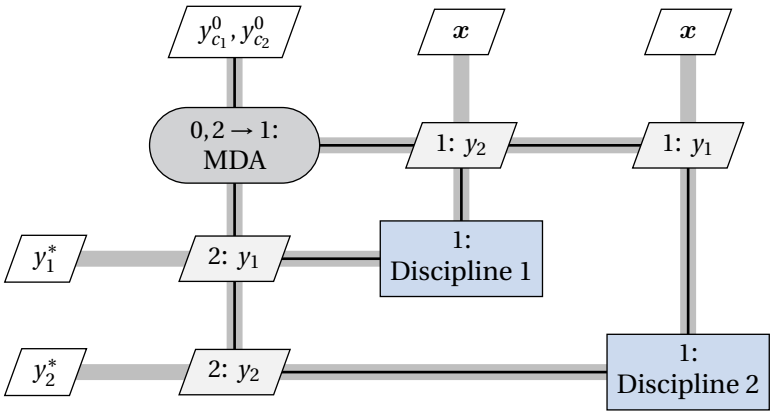


Figure 1.9: XDSM diagram for the resolution of a two-discipline MDA using a Jacobi solver.

Algorithm 4 Nonlinear Jacobi method

Input: $\boldsymbol{x}, \boldsymbol{y}_i^0,\ i = 1,...,n_d, k_{\max}$ ▷ Design point, initial guess for y_i and max. iterations
  $k = 0$ ▷ Initialize iteration count
  error $= 1$ ▷ Initialize relative error
  while error $> \epsilon_{\text{MDA}}$ **and** $k < k_{\max}$ **do**
    for $i = 1,...,n_d$ **do**
      $\boldsymbol{y}_i^{k+1} = f_i(\boldsymbol{x}, \boldsymbol{y}_{c_i}^k)$ ▷ Evaluate all disciplinary solvers
    end for
    error $= \max_{\{i=1,...,n_d\}} \frac{\|\boldsymbol{y}_i^{k+1} - \boldsymbol{y}_i^k\|_2}{\|\boldsymbol{y}_i^{k+1}\|_2}$ ▷ Update relative error
    $k = k + 1$ ▷ Increase iteration count
  end while
Output: $\boldsymbol{y}_i^k,\ i = 1,...,n_d$

The nonlinear Gauss-Seidel method. The nonlinear Gauss-Seidel method, illustrated in Figure 1.10, is similar to the Jacobi method, except that it uses the last computed value of the coupling variables, which may have been obtained in the current iteration. Thanks to these coupling variable updates, Gauss-Seidel solvers typically converge faster than Jacobi solvers. This is illustrated in Figure 1.11 where three iterations of the Jacobi and Gauss-Seidel algorithms are performed to solve the MDA problem defined in Eq. (1.45). In both figures, the same initial guess for the coupling variable values is used. After three iterations, the Jacobi method (Figure 1.11(a))

is still somewhat far from the MDA solution while the Gauss-Seidel solver (Figure 1.11(b)) has nearly converged.

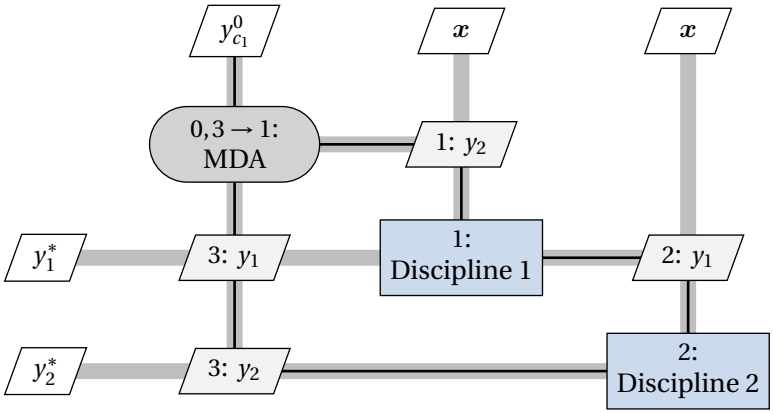
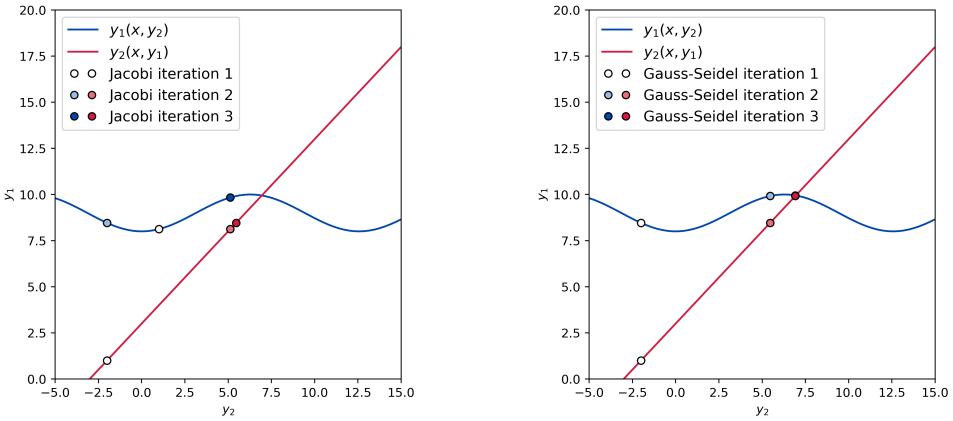


Figure 1.10: XDSM diagram for the resolution of a two-discipline MDA using a Gauss-Seidel solver.



(a) Jacobi method.

(b) Gauss-Seidel method.

Figure 1.11: Illustration of the Jacobi and Gauss-Seidel fixed-point algorithms. (a) Three iterations of the Jacobi algorithm. (b) Three iterations of the Gauss-Seidel algorithm.

Despite its faster convergence rates, the Gauss-Seidel solver is not necessarily computationally faster than the Jacobi method. Indeed, for the Gauss-Seidel method it is no longer possible to run the disciplinary solvers in parallel. Thus, if all disciplinary solvers require long computational times, the Jacobi method may be preferred. The main steps of the Gauss-Seidel method are summarized in Algorithm 5. Note that, in practice, this method only requires an initial guess for the coupling variables which influence the value of the first disciplinary solver output.

Gauss-Seidel solvers are sometimes used in combination with relaxation methods to dampen oscillations and avoid divergence. In these methods, a relaxation factor θ^k is introduced when updating the value of the coupling variables, as follows:

$$\boldsymbol{y}^{k+1} = \boldsymbol{y}^k + \theta^k \Delta \boldsymbol{y}^k \tag{1.53}$$

where $\Delta \boldsymbol{y}^k$ is the change in the coupling variable values obtained when no relaxation is used. In Eq. (1.53), the relaxation factor can be fixed throughout the iterations or adapted as the algo-

rithm progresses. The Aitken method [Irons and Tuck, 1969] is a popular a relaxation method where the relaxation factor is updated in each iteration.

Algorithm 5 Nonlinear Gauss-Seidel method

Input: $\boldsymbol{x}, \boldsymbol{y}_{c_1}^0, k_{\max}$ ▷ Design point, initial guess for y_i and max. iterations
  $k = 0$ ▷ Initialize iteration count
  error $= 1$ ▷ Initialize relative error
  while error $> \epsilon_{\text{MDA}}$ **and** $k < k_{\max}$ **do**
    for $i = 1, ..., n_d$ **do**
      $\boldsymbol{y}_i^{k+1} = f_i(\boldsymbol{x}, \boldsymbol{y}_{c_i}^{\text{GS}})$ with $\boldsymbol{y}_{c_i}^{\text{GS}} = \begin{cases} \boldsymbol{y}_j^{k+1} & \text{if } j < i \\ \boldsymbol{y}_j^k & \text{otherwise} \end{cases}$ ▷ Call disciplinary solvers in order
    end for
    error $= \max_{\{i=1,...,n_d\}} \frac{\|\boldsymbol{y}_i^{k+1} - \boldsymbol{y}_i^k\|_2}{\|\boldsymbol{y}_i^{k+1}\|_2}$ ▷ Update relative error
    $k = k + 1$ ▷ Increase iteration count
  end while
Output: $\boldsymbol{y}_i^k,\ i = 1, ..., n_d$

Newton's method. Another way to look at the fixed-point problem is by considering the following residual:

$$r_i = \boldsymbol{y}_i - f_i(\boldsymbol{y}_{c_i}), i = 1, ..., n_d \tag{1.54}$$

whose norm should be zero at the solution of the fixed-point problem ($\|r_i^*\|_2 = 0,\ i = 1, ..., n_d$). The fixed-point problem may then be reformulated as a residual minimization problem, which can be solved using any of the optimization algorithms introduced in Section 1.1. In fact, Newton's method has often been used to solve the MDA. The Newton step at each iteration of the MDA is written as:

$$\mathbf{J}_{\text{F}}(\boldsymbol{x}, \boldsymbol{y}^k)(\boldsymbol{y}^{k+1} - \boldsymbol{y}^k) = -\text{F}(\boldsymbol{x}, \boldsymbol{y}^k) \tag{1.55}$$

where $\text{F} = \{f_1(\boldsymbol{x}, \boldsymbol{y}_{c_1}), ..., f_{n_d}(\boldsymbol{x}, \boldsymbol{y}_{c_{n_d}})\}$ and $\mathbf{J}_{\text{F}}$ is the Jacobian of F. If derivative information is not available, Broyden's method [Broyden, 1965] can be used to approximate the Jacobian matrix or its inverse. The use of derivative information allows the Newton method to converge faster than the Jacobi or Newton methods. However, Newton's method may fail to converge if a poor initial guess is provided. As a result, in this work, Gauss-Seidel methods will be favored when solving the MDA.

1.2.3.2 Efficient global multidisciplinary design analysis

In the previous section we introduced different methods to solve the MDA (Eq. (1.43)). The presented algorithms solve the non-linear coupled system of equations by iteratively calling the disciplinary solvers until a given convergence criterion is achieved. When the disciplinary solvers are expensive numerical solvers, however, the resolution of the MDA can result in a significant computational burden. This computational burden becomes prohibitive when the MDA must be solved several times, for different design space points. To ease this computational burden, it is proposed in [Dubreuil et al., 2020, Berthelin et al., 2022] to replace the disciplinary solvers by GP approximations, adaptively enriched throughout the resolution of the MDA. In this work we will refer to this strategy as Efficient Global Multidisciplinary Design Analysis (EGMDA). The main steps of EGMDA are described in the following.

Disciplinary surrogates. The EGMDA strategy, illustrated in Figure 1.12, proposes to solve an approximated version of the MDA, obtained by replacing each disciplinary solver by a GP surrogate. To build the disciplinary surrogates, a DoE for each discipline is first obtained, by sampling over the space $\mathscr{X} \times \prod_{j \in c_i} \mathscr{C}^j$. Then, the disciplinary surrogates are trained using the corresponding disciplinary DoEs. The obtained disciplinary GP approximations are written as:

$$\hat{f}_i(\boldsymbol{x}, \boldsymbol{y}_{c_i}) = \mu_i(\boldsymbol{x}, \boldsymbol{y}_{c_i}) + \epsilon_i(\boldsymbol{x}, \boldsymbol{y}_{c_i}) \tag{1.56}$$

where $\mu_i(\boldsymbol{x}, \boldsymbol{y}_{c_i})$ is the mean function of the GP and $\epsilon_i(\boldsymbol{x}, \boldsymbol{y}_{c_i})$ is a GP of zero mean conditioned on the respective disciplinary DoE. We note that, for the construction of the disciplinary DoEs, often no information is available concerning $\mathscr{C}^i$ and an initial guess of its bounds must be provided. While this is a drawback associated with the use of disciplinary surrogates, in practice we may rely on expert judgment to estimate the bounds of $\mathscr{C}^i$. Moreover, even if a poor initial guess is provided, by enriching the disciplinary surrogates outside the initial bounds, the definition of the coupling variable spaces is updated.

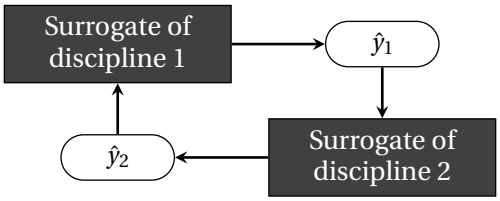


Figure 1.12: Illustration of the EGMDA strategy for the resolution of the MDA. An approximated version of the MDA is obtained by replacing each disciplinary solver by a GP surrogate.

By replacing the disciplinary solvers by their corresponding surrogates in Eq. (1.43), an approximated version of the MDA is obtained:

$$\hat{\boldsymbol{y}}_i(\boldsymbol{x}, \hat{\boldsymbol{y}}_{c_i}) = \mu_i(\boldsymbol{x}, \hat{\boldsymbol{y}}_{c_i}) + \epsilon_i(\boldsymbol{x}, \hat{\boldsymbol{y}}_{c_i}),\ i = 1, ..., n_d \quad \forall \boldsymbol{x} \in \mathscr{X} \tag{1.57}$$

An illustration of this approximation on the 1-D example is provided in Figure 1.13, where the dotted lines represent the exact disciplinary solvers, the continuous lines display the mean values of the corresponding disciplinary surrogates and the shaded regions are the confidence intervals. To obtain the GP approximations, the following initial guess for the coupling variable space was provided: $\mathscr{C}^1 = [0, 25]$ and $\mathscr{C}^2 = [0, 25]$.

As is shown, for the current GP approximations, the intersection of the surrogate mean solution does not correspond to the real MDA solution. To find the true MDA solution it is thus necessary to enrich the disciplinary GPs. In the following we describe how the uncertainty of the disciplinary surrogates is evaluated and how it may be reduced via an adaptive sampling strategy.

Solving the approximated MDA. A consequence of replacing the disciplinary solvers by GP surrogates is that the solution of the approximated MDA (Eq. (1.57)) is no longer deterministic. The first step in the resolution of the approximated MDA is thus to obtain random solutions to model the uncertainty of the disciplinary GPs. It is proposed in [Dubreuil et al., 2020] to model this uncertainty by considering perfectly dependent GPs, i.e., by considering conditioned GPs whose correlation function is constant and equal to one. The approximated MDA becomes:

$$\hat{\boldsymbol{y}}'_i(\boldsymbol{x}, \boldsymbol{y}'_{c_i}) = \mu_i(\boldsymbol{x}, \boldsymbol{y}'_{c_i}) + \sigma_i(\boldsymbol{x}, \boldsymbol{y}'_{c_i})\xi_i,\ i = 1, ..., n_d \quad \forall \boldsymbol{x} \in \mathscr{X} \tag{1.58}$$

where $\sigma_i(\boldsymbol{x}, \boldsymbol{y}_{c_i})$ is the standard deviation and ξ_i is a standard Gaussian random variable. It is worth noting that, at any given design space point $\boldsymbol{x} \in \mathscr{X}$, the solution to Eq. (1.58) depends on

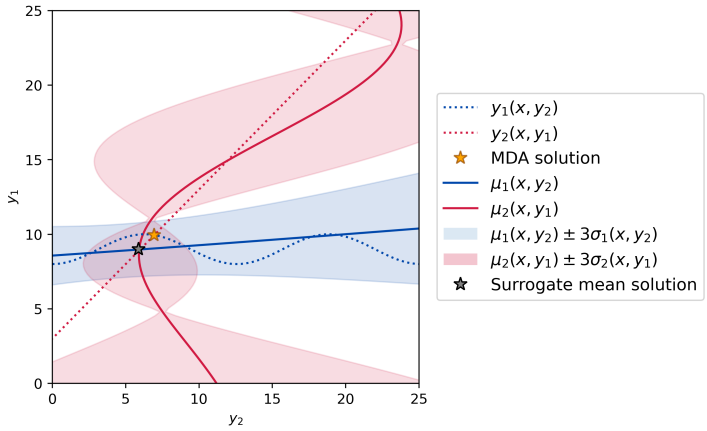


Figure 1.13: Illustration of the approximated MDA, obtained when replacing the disciplinary solvers by their GP surrogates. Comparison of true MDA solution (yellow star) with the surrogate mean solution at $x^* \approx -3$ (gray star).

the sample $\Xi = \{\xi_i,\ i = 1, ..., n_d\}$ drawn. Thus, for different drawn samples, different MDA solutions are obtained. Figure 1.14 illustrates different realizations of the disciplinary GPs (dashed lines) obtained from different samples of Ξ. For each sample $\Xi^{(k)} = \{\xi_1^{(k)}, \xi_2^{(k)}\}$ drawn, the corresponding MDA solution is found by solving Eq. (1.58) using the fixed-point methods described in the Section 1.2.3.1. The random MDA solutions obtained from the drawn samples are identified in Figure 1.14 with black crosses.

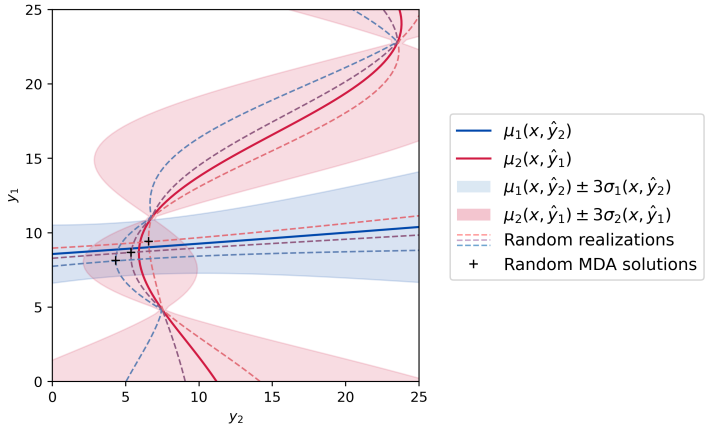


Figure 1.14: Illustration of three random realizations of the disciplinary GPs (dashed lines). Identification of the corresponding random MDA solutions (black crosses).

To characterize the variability of the solution of the approximated MDA, a sufficiently large number of samples should be drawn. Nonetheless, because the disciplinary surrogates are cheap to evaluate, a direct Monte Carlo method can be used. Figure 1.15 illustrates the random MDA solutions obtained when 200 samples of Ξ are drawn.

To decide if the surrogate approximations are sufficiently accurate, a portion τ of the random MDA solutions should be sufficiently close to the mean solution (in other words, the dispersion of the random MDA solutions should be small). The mean value of each coupling variable can

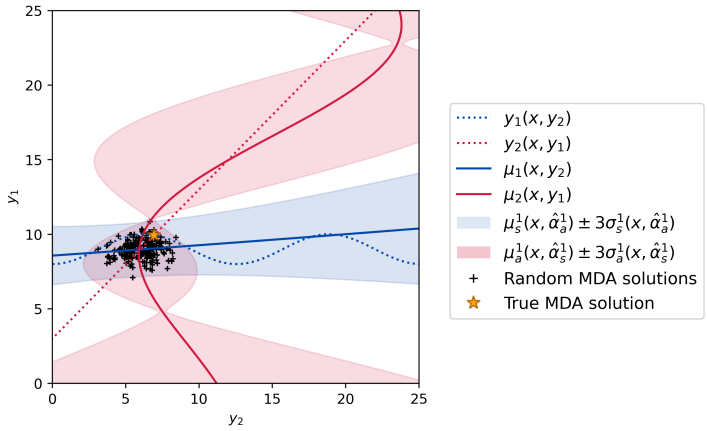


Figure 1.15: Illustration of the random MDA solutions (black crosses) obtained when 200 samples of Ξ are drawn. Comparison with the true MDA solution (yellow star).

be estimated from the random MDA solutions as:

$$\bar{\boldsymbol{y}}_i^* = \frac{1}{n_{\text{MC}}} \sum_{j=1}^{n_{\text{MC}}} \hat{\boldsymbol{y}}_i^*(\xi_i^{(k)}),\ \forall i = 1, ..., n_d \tag{1.59}$$

where n_{MC} is the number of samples of Ξ drawn. The relative distance between the coupling variable values at a given random MDA solution and the mean solution is:

$$\widehat{\text{QoI}}_i(\Xi^{(k)}) = \frac{\|\hat{y}_i^*(\Xi^{(k)}) - \bar{\boldsymbol{y}}_i^*\|_2}{\|\bar{\boldsymbol{y}}_i^*\|_2},\ \forall k = 1, ..., n_{\text{MC}},\ \forall i = 1, ..., n_d \tag{1.60}$$

When the surrogate approximations are sufficiently accurate, the τ-quantile of $\widehat{\text{QoI}}_i(\Xi)$, $\forall i = 1, ..., n_d$ is less than a small threshold. If this condition is not met, the disciplinary surrogates are not sufficiently accurate and should be locally enriched. To avoid waste of computational effort, however, it is proposed in [Berthelin, 2022, Berthelin et al., 2022] to enrich only the least accurate disciplinary surrogate. This least accurate surrogate is chosen based on the Sobol sensitivity indices [Sobol', 2001] which can be estimated from the random MDA solutions, as follows:

$$\text{S}_i = \frac{\mathbb{V}[\mathbb{E}[\widehat{\text{QoI}} \mid \boldsymbol{\xi}_i]]}{\mathbb{V}[\widehat{\text{QoI}}]},\ i = 1, ..., n_d \tag{1.61}$$

where $\widehat{\text{QoI}} = \widehat{\text{QoI}}_1(\Xi) + ... + \widehat{\text{QoI}}_{n_d}(\Xi)$. The Sobol sensitivity indices presented above are estimated thanks to a Polynomial Chaos Expansion (PCE) approximation, as proposed in [Sudret, 2008]. The discipline with the highest first order Sobol index is then enriched by adding $f_i(\boldsymbol{x}, \bar{\boldsymbol{y}}_{c_i}^*)$ to the corresponding disciplinary DoE. Figure 1.16 illustrates the obtained random MDA solutions after two iterations of disciplinary surrogate enrichments. As shown, after only two surrogate enrichments, the dispersion of the random MDA solutions has been significantly reduced and the surrogate mean solution nearly coincides with the true MDA solution. The main steps of the EGMDA algorithm are summarized in Algorithm 6.

Comparison between EGMDA and other strategies. Although EGMDA is a recent strategy, it should be noted that the idea of iteratively building approximations of the disciplinary functions is not new. Indeed, EGMDA can be compared with Newton's method, which iteratively

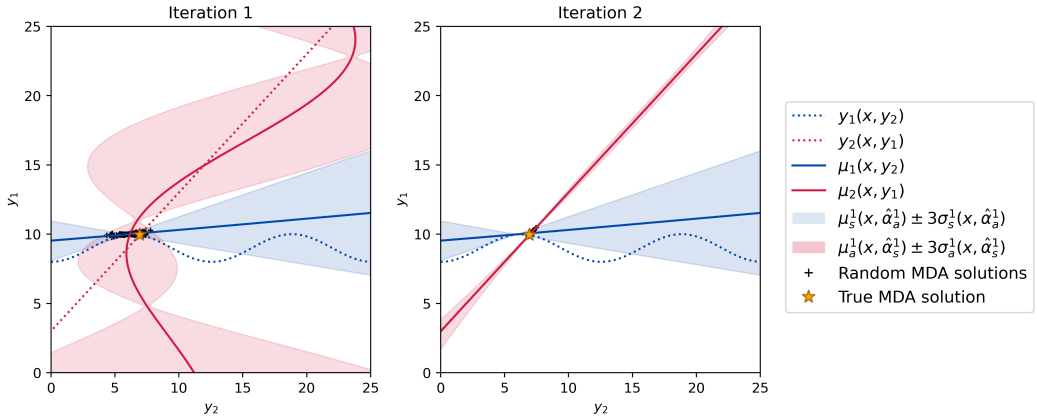


Figure 1.16: Illustration of the random MDA solutions (black crosses) obtained for two iterations of disciplinary surrogate enrichments.

builds linear approximations of the disciplinary functions. The intersection between the linear approximations is then found via the Newton step, introduced in Eq. (1.55). Figure 1.17 presents two iterations of Newton's method for the 1-D example. In the figure, the linear approximation for the first disciplinary solver is identified with a dashed blue line. Because the second disciplinary solver arises from a linear equation, the true function and its linear approximation coincide. As a result, the approximation $\hat{y}_2(x, y_1)$ is omitted for the sake of clarity.

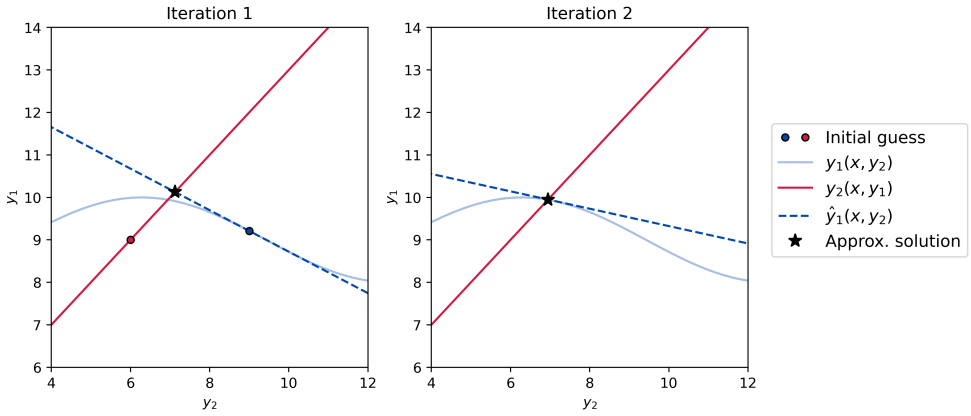


Figure 1.17: Two iterations of the Newton method for the 1-D illustrative example at $x^* = -3$. For the sake of clarity, the approximation $\hat{y}_2(x, y_1)$ is omitted.

The treatment of uncertain coupling variables is also not new. Indeed, Uncertainty Multidisciplinary Design Optimization (UMDO) methods account for the uncertainties introduced in the coupling variable values, for example due to the use of low fidelity models [Brevault et al., 2016]. As in the EGMDA approach, early UMDO methods solved the disciplinary coupling for each realization of the uncertain coupling variables [Oakley et al., 1998, Koch et al., 2002]. While more recent UMDO strategies have abandoned this approach due to the inherent computational cost, in EGMDA using direct Monte Carlo sampling remains computationally feasible, since the disciplinary solvers have been replaced by their corresponding surrogates.

Algorithm 6 Efficient Global Multidisciplinary Design Analysis (EGMDA)

Input: x, $j_{\max}$ ▷ Design space point and max. iterations
  $j = 0$ ▷ Initialize iteration count
  conv = False
  while conv == False and $j < j_{\max}$ **do**
    Compute $\hat{y}_i^*(\Xi^{(k)})$, $\forall k = 1,...,n_{\text{MC}}$ ▷ Solve Eq. (1.58) for n_{MC} samples of Ξ
    Compute $\widehat{\text{QoI}}_i(\Xi)$, $\forall i = 1,...,n_d$ ▷ Quantity of interest (Eq. (1.60))
    if τ-quantile of $\widehat{\text{QoI}}_i < \epsilon_{\text{MDA}}$ $\forall i = 1,...,n_d$ **then**
      conv = True
    else
      Compute S_i, $\forall i = 1,...,n_d$ ▷ Estimate Sobol indices by PCE
      Enrich disciplinary GP associated with highest S_i
    end if
    $j = j + 1$ ▷ Increase iteration count
  end while
Output: $\hat{y}_i^*$, $\forall i = 1,...,n_d$

1.2.4 MDO formulations

Most of the methods introduced for the resolution of the MDA rely on iterative approaches, which require many calls to the disciplinary solvers. When the disciplinary solvers are expensive numerical solvers, the cost of solving one MDA becomes important. To optimize the multidisciplinary system, not one, but several MDAs may need to be solved at different design space points. Thus, we might try to reduce the computational cost by avoiding the resolution of the MDA altogether. Indeed, we could try to iteratively optimize each discipline of the system individually. This approach is referred to as sequential design optimization and it was shown in [Grossman et al., 1988, Chittick and Martins, 2009] that it does not necessarily lead to the optimal solution of the coupled system. Several other MDO formulations have been proposed in the literature which attempt to efficiently solve the MDO problem. In this section we provide only a brief overview of some of these formulations, but a more thorough review can be found in [Martins and Lambe, 2013, Martins and Ning, 2021].

1.2.4.1 The Multidisciplinary Feasible approach

In the Multidisciplinary Feasible (MDF) approach [Cramer et al., 1994] the MDA is treated as if it were a single discipline. As a result, to compute objective or constraint functions, the MDA must be solved at each iteration of the optimizer. Although this can be computationally costly, the MDF approach will always return a consistent system design, even if the optimization process is terminated early. This can be advantageous if the goal is simply to improve the initial design, rather than finding the true optimum. We note, however, that if constraint functions exist, stopping the optimization prematurely may result in a design which violates some or all constraint functions. The XDSM diagram for the resolution of the MDO using an MDF approach is depicted in Figure 1.18. The provided example is for an unconstrained optimization problem of a two-discipline system.

1.2.4.2 The Individual Discipline Feasible approach

In the Individual Disciplinary Feasible (IDF) approach [Cramer et al., 1994], the coupling variables are added to the set of design variables and constraints are defined to assure the consistency of the optimal solution. The optimization problem is thus no longer given by Eq. (1.42)

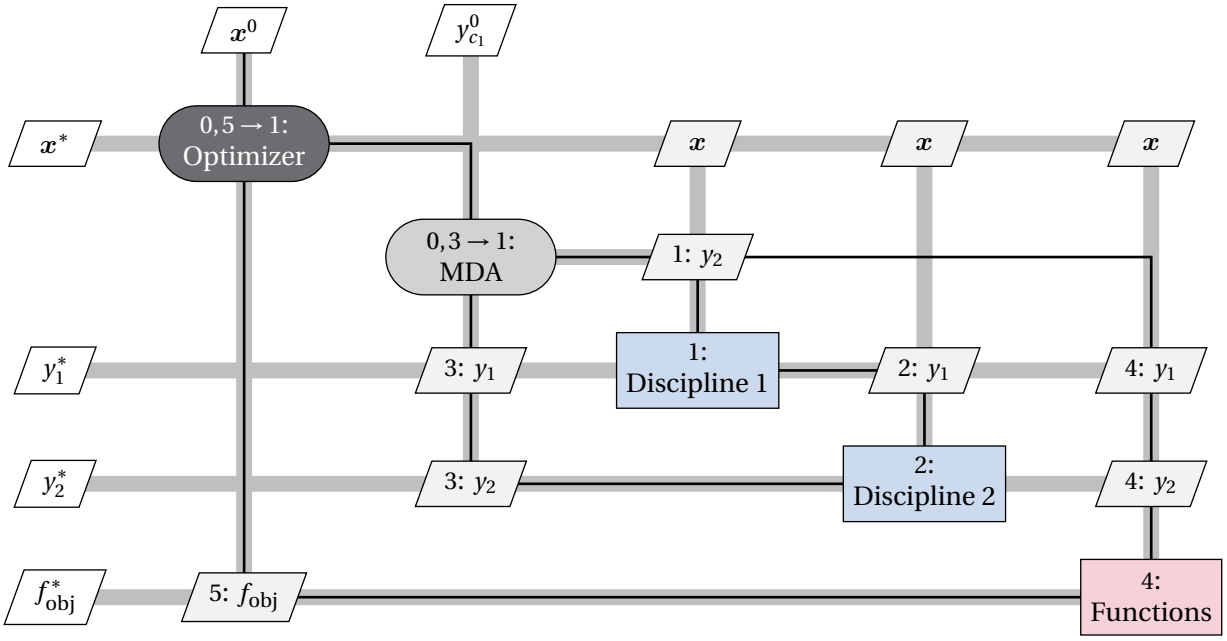


Figure 1.18: XDSM diagram for the resolution of the MDO using an MDF approach. Example for an unconstrained optimization of a two-discipline system. Note the numbering system which implies that the MDA is solved before evaluating the objective function.

but is instead written as:

$$\begin{aligned} \boldsymbol{x}^* = \arg\min_{\boldsymbol{x},\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{n_d}} \quad & f_{\text{obj}}(\boldsymbol{x},\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{n_d}) \\ \text{s.t.} \quad & r_i = \boldsymbol{y}_i - f_i(\boldsymbol{x},\boldsymbol{y}_{c_i}) = 0, \ \forall i = 1,\ldots,n_d \\ & g(\boldsymbol{x},\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{n_d}) \leq 0 \\ & h(\boldsymbol{x},\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{n_d}) = 0 \end{aligned} \tag{1.62}$$

where r_i, $i = 1,\ldots,n_d$ are the consistency constraints. This formulation allows to search for the optimal set of design parameters without having to solve the MDA at each optimizer iteration. This typically results in less disciplinary solver calls, making the IDF approach less costly than the MDF approach. Moreover, because only coupling variable values computed in the previous iteration are used, the disciplinary solvers can be run in parallel. The IDF approach also presents some disadvantages. Namely, should the optimization fail or be terminated early, an inconsistent solution may be returned. In addition, in this formulation, the number of design variables and constraints scales with the number of coupling variables. As a result, if the number of coupling variables is too large, the optimization problem may become too hard to solve efficiently. An XDSM of the IDF approach is provided in Figure 1.19. The provided example is for an unconstrained problem, where the disciplinary solvers are run in parallel.

Simultaneous Analysis and Design. Variations of the IDF approach also exist in the literature, one of which is the Simultaneous Analysis and Design (SAND) formulation [Haftka, 1985]. In the SAND approach, the optimizer handles not only the multidisciplinary consistency but also the consistency of the disciplinary residual equations. Although this approach can be computationally faster than the IDF approach, it is an intrusive approach which requires access to the underlying numerical implementation of the disciplinary solvers.

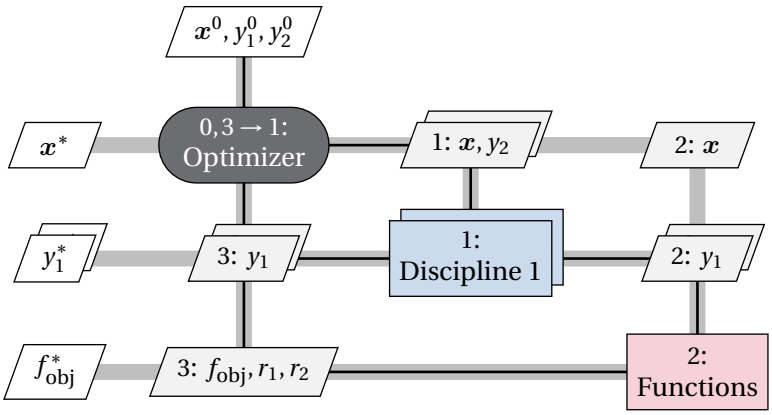


Figure 1.19: XDSM diagram for the resolution of the MDO using an IDF approach.

1.2.4.3 Distributed MDO formulations

Both MDF and IDF approaches solve a single optimization problem. However, other approaches have been proposed in the literature where disciplinary optimization subproblems are solved in addition to the main system-level optimization problem. These are called distributed MDO formulations. Distributed MDO formulations attempt to mimic the structure of large engineering design projects, where dedicated disciplinary design teams work separately and do not communicate frequently. Although they represent the industrial setting, distributed MDO formulations typically converge slower than their single-level equivalents and thus have received less attention in the academic setting. For this reason we do not present these formulations in detail but provide only a summary of the existing formulations.

Some distributed MDO formulations enforce multidisciplinary consistency by directly solving the MDA and thus are called distributed MDF approaches, while others enforce the consistency of multidisciplinary system through constraint or penalty functions and thus are called distributed IDF approaches. Examples of distributed MDF approaches include the Concurrent Subspace Optimization (CSSO) [Sobieszczanski-Sobieski, 1988, Bloebaum et al., 1992] and the Bilevel Integrated System Synthesis (BLISS) [Sobieszczanski-Sobieski et al., 2003] formulations whereas examples of distributed IDF approaches include the Collaborative Optimization (CO) [Braun, 1996] and the Analytical Target Cascading (ATC) [Kim et al., 2003, Tosserams et al., 2006] formulations.

1.2.5 Optimization algorithms for MDO

1.2.5.1 Gradient-based MDO

In practice, the MDO formulations presented above can be used with any of the optimization algorithms described in Section 1.1. However, some remarks should be made concerning gradient-based MDO. Indeed, even if gradient information is available at a disciplinary solver level, we still need to compute objective and constraint function derivatives. To obtain these derivatives, the chain rule can be applied. However, the computation is not straightforward, as the solution of the MDA is required. To address this issue, the coupled-adjoint method, proposed in [Martins, 2002, Martins et al., 2005], provides a strategy to compute analytically the derivatives of the coupled system via the chain rule. A brief overview of this approach is given in the following.

Analytical derivatives of coupled systems. To perform gradient-based optimization, the total derivatives of the quantities of interest (objective or constraint functions) with respect to the

design variables are needed. These total derivatives can be obtained using the chain rule. For the objective function, they are written as:

$$\frac{\mathrm{d}f_{\text{obj}}}{\mathrm{d}x_k} = \frac{\partial f_{\text{obj}}}{\partial x_k} + \frac{\partial f_{\text{obj}}}{\partial \boldsymbol{y}} \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x_k} \tag{1.63}$$

with $k = 1, ..., n$ and $\boldsymbol{y} = \{\boldsymbol{y}_1, ..., \boldsymbol{y}_{n_d}\}$. We note the importance of distinguishing between total and partial derivatives in this equation. Indeed, while partial derivatives can be evaluated directly, the total derivatives require the solution of the MDA. Thus, in Eq. (1.63) all terms are known, except for $\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x_k}$. In [Martins, 2002, Martins et al., 2005] it is proposed to use the residual of the governing equations of the disciplinary solvers to find this term. Because the disciplinary solvers typically work by driving the residuals of the respective governing equations to zero, we can write:

$$\mathscr{R}(\boldsymbol{x}, \boldsymbol{y}_1(\boldsymbol{x}), ..., \boldsymbol{y}_{n_d}(\boldsymbol{x})) = 0 \tag{1.64}$$

where $\mathscr{R}$ denotes the residuals of all governing equations of the different disciplines. Since the governing equations must always be satisfied, the total derivatives of the residuals with respect to any design variable must also be equal to zero:

$$\frac{\mathrm{d}\mathscr{R}}{\mathrm{d}x_k} = \frac{\partial \mathscr{R}}{\partial x_k} + \frac{\partial \mathscr{R}}{\partial \boldsymbol{y}} \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x_k} = 0 \tag{1.65}$$

Rewriting the above equation allows to obtain the total derivatives of the coupling variables with respect to the design variables as the solution of the following linear system of equations:

$$\frac{\partial \mathscr{R}}{\partial \boldsymbol{y}} \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x_k} = -\frac{\partial \mathscr{R}}{\partial x_k} \tag{1.66}$$

Solving Eq. (1.66) for $\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x_k}$ and replacing the result into Eq. (1.63) is the direct method of obtaining the total derivatives of the objective function with respect to the design variables. Because Eq. (1.66) does not depend on the quantity of interest, the direct mode is useful when the number of objectives or constraints is large. However, when the number of design variables is significantly greater than the amount of quantities of interest, the adjoint method can be more efficient. When using the adjoint method, we substitute the term $\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x_k}$ into the total derivative equation (Eq. (1.63)), as follows:

$$\frac{\mathrm{d}f_{\text{obj}}}{\mathrm{d}x_k} = \frac{\partial f_{\text{obj}}}{\partial x_k} - \underbrace{\frac{\partial f_{\text{obj}}}{\partial \boldsymbol{y}} \left[\frac{\partial \mathscr{R}}{\partial \boldsymbol{y}}\right]^{-1}}_{-\Psi} \frac{\partial \mathscr{R}}{\partial x_k} \tag{1.67}$$

The auxiliary vector Ψ, also called adjoint vector, can be obtained by solving the adjoint equations, given by:

$$\frac{\partial \mathscr{R}}{\partial \boldsymbol{y}} \Psi = -\frac{\partial f_{\text{obj}}}{\partial \boldsymbol{y}} \tag{1.68}$$

The result is then substituted into Eq. (1.67) to obtain the total derivatives. As opposed to the direct method, Eq. (1.68) does not depend on x_k, but depends on the quantity of interest. As a result, the adjoint method is preferred when dealing with a large number of design variables, but a small number of quantities of interest.

Limitations of gradient-based MDO. If the derivatives at a disciplinary solver level are not available, it is not possible to use the direct or adjoint methods and the objective and constraint function derivatives must be approximated using finite-differences or complex step. However, this remains a computationally expensive task, more so if we need to solve the MDA for each step of the approximations. Additionally, if multiple minima exist, a multi-start strategy is still required to find the global optimum of the MDO problem.

1.2.5.2 Surrogate-based MDO

Surrogate-based approaches, like the EGO algorithm presented in Section 1.1.2 can also be used to solve MDO problems. In an MDO context, however, both the initial DoE as well as each newly sampled point require a new resolution of the MDA. Thus, the use of the EGO algorithm combined with an MDF approach can remain computationally costly. This computational cost can be decreased by building disciplinary surrogates and using the EGMDA strategy (see Section 1.2.3.2) to solve the MDA at each queried design space point. Despite the reduced computational cost, however, the combined EGMDA and EGO strategy will still seek to enrich the disciplinary surrogates at points which are unlikely to be the global optimum. To address this issue, another strategy, called Efficient Global Multidisciplinary Design Optimization (EGMDO) [Dubreuil et al., 2020], has been specifically developed for the global optimization of multidisciplinary systems. EGMDO is introduced in the following.

1.2.5.3 Efficient Global Multidisciplinary Design Optimization

In EGMDO, like in the EGMDA strategy, we replace each disciplinary solver by a GP surrogate. Rather than solving the MDA at each queried point, however, in EGMDO the uncertainty of the disciplinary GPs is propagated onto an objective function approximation. Based on this approximation, the disciplinary surrogates are enriched only at design space points which are likely to be the global optimum. Like in the EGO approach, new points can also be sampled into the objective function DoE with the aid of a dedicated acquisition function. Thanks to the adaptive enrichment of the disciplinary and objective function surrogates, the EGMDO is able to reduce the computational cost required for the resolution of the MDO problem. The construction of the objective function approximation from the random MDA solutions is detailed in the following.

Objective function approximation. As a consequence of having random MDA solutions, the objective function, which depends on the set of random coupling variables $\boldsymbol{\mathcal{Y}}_c^*(\boldsymbol{x},\boldsymbol{\Xi})$, becomes itself a random variable:

$$\hat{f}_{\text{obj}}(\boldsymbol{x},\boldsymbol{\Xi}) = f_{\text{obj}}(\boldsymbol{x},\boldsymbol{\mathcal{Y}}_c^*(\boldsymbol{x},\boldsymbol{\Xi})),\ \forall \boldsymbol{x} \in \mathscr{X} \tag{1.69}$$

where $\boldsymbol{\Xi} = \{\Xi^{(k)},\ k = 1,...,n_{\text{MC}}\}$. One of the key features of the EGMDO algorithm lies in the construction of a continuous objective function approximation which takes into account the uncertainty induced by the disciplinary GPs in $\hat{f}_{\text{obj}}(\boldsymbol{x},\boldsymbol{\Xi})$. The proposed approximation is divided into two steps. First, a pointwise PCE approximation of the objective function is obtained for an initial $\text{DoE}_{\text{UQ}} = \{\boldsymbol{x}_1,...,\boldsymbol{x}_{n_{\text{UQ}}}\}$. Then, to extend the existing PCE approximation to the remainder of the design space $\mathscr{X}$, a combination of a Karhunen-Loève (KL) decomposition with GP interpolation is proposed.

The PCE approximation step allows us to write the random objective function as a series expansion, instead of a random output. Assuming that $\hat{f}_{\text{obj}}(\boldsymbol{x},\boldsymbol{\Xi})$ is square integrable, i.e. that $\mathbb{E}[(\hat{f}_{\text{obj}}(\boldsymbol{x},\boldsymbol{\Xi}))^2] < \infty$, then $\hat{f}_{\text{obj}}(\boldsymbol{x},\boldsymbol{\Xi})$ can be approximated as [Ghanem and Spanos, 2003]:

$$\hat{f}_{\text{obj}}(\boldsymbol{x},\boldsymbol{\Xi}) \approx \hat{f}_{\text{obj}}^{(N)}(\boldsymbol{x},\boldsymbol{\Xi}) = \sum_{j=1}^{N} a_j(\boldsymbol{x})\text{H}_j(\boldsymbol{\Xi}),\ \forall \boldsymbol{x} \in \text{DoE}_{\text{UQ}} \tag{1.70}$$

where $a_j(\boldsymbol{x})$ are the PCE coefficients and $\mathrm{H}_j(\boldsymbol{\Xi})$ are the n_d-variate Hermite polynomials. The PCE approximation of Eq. (1.70) converges in the mean-square sense when $N \to +\infty$, that is:

$$\lim_{N \to +\infty} \mathbb{E}[(\hat{f}_{\text{obj}}^{(N)}(\boldsymbol{x}, \boldsymbol{\Xi}) - \hat{f}_{\text{obj}}(\boldsymbol{x}, \boldsymbol{\Xi}))^2] = 0 \tag{1.71}$$

In practice, however, Eq. (1.70) is truncated to the first P terms, with P chosen so that only the polynomials with a degree less or equal to d are kept:

$$P = \frac{(n_d + d)!}{n_d! d!} \tag{1.72}$$

where n_d is the stochastic dimension of the problem, here assumed to be the number of coupling variables (see Eq. (1.58)). The PCE approximation becomes:

$$\hat{f}_{\text{obj}}^{\text{PCE}}(\boldsymbol{x}, \boldsymbol{\Xi}) = \sum_{j=1}^{P} a_j(\boldsymbol{x}) \mathrm{H}_j(\boldsymbol{\Xi}), \ \forall \boldsymbol{x} \in \text{DoE}_{\text{UQ}} \tag{1.73}$$

Figure 1.20 compares the PDF of the obtained PCE approximation (left) with the PDF of the corresponding Monte Carlo approximation (right) for the 1-D illustrative problem. Box plots representing the random solutions used to obtain the drawn PDFs are equally shown.

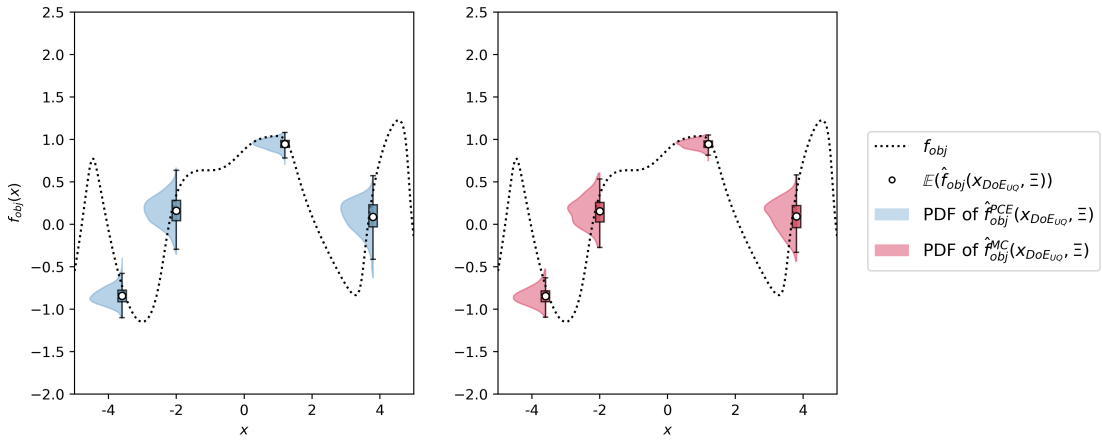


Figure 1.20: Comparison of the PDFs of the obtained PCE approximation (left) with the corresponding Monte Carlo approximation (right) at four different points $\boldsymbol{x} \in \text{DoE}_{\text{UQ}}$. The coefficients for the PCE approximation are obtained by regression over 200 random MDA solutions.

As illustrated, the PCE provides a reasonable approximation of the random variable $\hat{f}_{\text{obj}}(\boldsymbol{x}, \boldsymbol{\Xi})$ for all points $\boldsymbol{x} \in \text{DoE}_{\text{UQ}} = \{-3.6, -2, 1.2, 3.8\}$. It is then possible to extend this approximation to the remainder of the design space via a combined KL-GP interpolation, as proposed in [Dubreuil et al., 2018]. The KL decomposition allows us to reduce the number of quantities to interpolate in order to extend the PCE approximation to the whole domain. Considering the random vector $\hat{\boldsymbol{f}}_{\text{obj}}^{\text{PCE}}(\boldsymbol{\Xi}) = \{\hat{f}_{\text{obj}}^{\text{PCE}}(\boldsymbol{x}_1, \boldsymbol{\Xi}), ..., \hat{f}_{\text{obj}}^{\text{PCE}}(\boldsymbol{x}_{n_{\text{UQ}}}, \boldsymbol{\Xi})\}$, its KL decomposition may be written as:

$$\hat{\boldsymbol{f}}_{\text{obj}}^{\text{PCE}}(\boldsymbol{\Xi}) = \boldsymbol{\mu}_{\hat{f}_{\text{obj}}^{\text{PCE}}} + \sum_{k=1}^{n_{\text{UQ}}} \gamma_k(\boldsymbol{\Xi}) \sqrt{\lambda_k} \boldsymbol{\varphi}_k \tag{1.74}$$

where $\boldsymbol{\mu}_{\hat{f}_{\text{obj}}^{\text{PCE}}}$ denotes the mean of the vector $\hat{\boldsymbol{f}}_{\text{obj}}^{\text{PCE}}(\boldsymbol{\Xi})$, $\gamma_k(\boldsymbol{\Xi})$ are zero-mean, uncorrelated, random variables, such that:

$$\gamma_k = \frac{1}{\sqrt{\lambda_k}} \left(\hat{\boldsymbol{f}}_{\text{obj}}^{\text{PCE}}(\boldsymbol{\Xi}) - \boldsymbol{\mu}_{\hat{f}_{\text{obj}}^{\text{PCE}}} \right)^{\mathsf{T}} \boldsymbol{\varphi}_k \tag{1.75}$$

and λ_k and φ_k are, respectively, the eigenvalues and eigenvectors of the covariance matrix $\mathbf{K}_{\text{PCE}}$ of $\hat{\boldsymbol{f}}_{\text{obj}}^{\text{PCE}}(\boldsymbol{\Xi})$. It is shown in [Arnst et al., 2012] that, for the particular case where the random vector is obtained by PCE, the KL decomposition may be obtained from the PCE coefficients, according to:

$$\hat{\boldsymbol{f}}_{\text{obj}}^{\text{PCE}}(\boldsymbol{\Xi}) = \boldsymbol{\mu}_{\hat{f}_{\text{obj}}^{\text{PCE}}} + \sum_{k=1}^{n_{\text{UQ}}} \left(\sum_{j=2}^{P} \boldsymbol{a}_j^{\intercal} \varphi_k \mathrm{H}_j(\boldsymbol{\Xi}) \right) \varphi_k \tag{1.76}$$

where $\boldsymbol{\mu}_{\hat{f}_{\text{obj}}^{\text{PCE}}} = \boldsymbol{a}_1$, $\boldsymbol{a}_j = \{a_j(\boldsymbol{x}_1), ..., a_j(\boldsymbol{x}_{n_{\text{UQ}}})\}$, $j = 2, ..., P$ are the coefficients of the PCE at each point $\boldsymbol{x} \in \text{DoE}_{\text{UQ}}$ and φ_k are the eigenvectors of the covariance matrix, defined as $\mathbf{K}_{\text{PCE}} = \sum_{l=2}^{P} \boldsymbol{a}_j \boldsymbol{a}_j^{\intercal}$. Equation (1.76) may be further simplified by truncating it so that only the M eigenvectors corresponding to the M non-negligible eigenvalues are retained. The KL decomposition becomes:

$$\hat{\boldsymbol{f}}_{\text{obj}}^{\text{PCE}}(\boldsymbol{\Xi}) \approx \hat{\boldsymbol{f}}_{\text{obj}}^{\text{KL}}(\boldsymbol{\Xi}) = \boldsymbol{\mu}_{\hat{f}_{\text{obj}}^{\text{PCE}}} + \sum_{k=1}^{M} \left(\sum_{l=2}^{P} \boldsymbol{a}_l^{\intercal} \varphi_k \mathrm{H}_l(\boldsymbol{\Xi}) \right) \varphi_k \tag{1.77}$$

Interpolation by GP of the mean and eigenvectors can then be used to extend the approximation to the whole domain. The obtained random field reads:

$$\hat{f}_{\text{obj}}^{\text{PCE}}(\boldsymbol{x}, \boldsymbol{\Xi}) \approx \tilde{f}_{\text{obj}}(\boldsymbol{x}, \boldsymbol{\Xi}, \boldsymbol{\eta}) = \tilde{\boldsymbol{\mu}}_{\hat{f}_{\text{obj}}}(\boldsymbol{x}, \boldsymbol{\eta}) + \sum_{k=1}^{M} \left(\sum_{l=2}^{P} \boldsymbol{a}_l^{\intercal} \varphi_k \mathrm{H}_l(\boldsymbol{\Xi}) \right) \tilde{\varphi}_k(\boldsymbol{x}, \boldsymbol{\eta}) \tag{1.78}$$

where $\tilde{\boldsymbol{\mu}}_{\hat{f}_{\text{obj}}}(\boldsymbol{x}, \boldsymbol{\eta})$ and $\tilde{\varphi}_k(\boldsymbol{x}, \boldsymbol{\eta})$ denote, respectively, the GP interpolation of the mean and eigenvectors and $\boldsymbol{\eta}$ is a vector of independent normal random variables. Figure 1.21 plots the obtained GP approximation for the term $\tilde{\boldsymbol{\mu}}_{\hat{f}_{\text{obj}}}(\boldsymbol{x}, \boldsymbol{\eta})$.

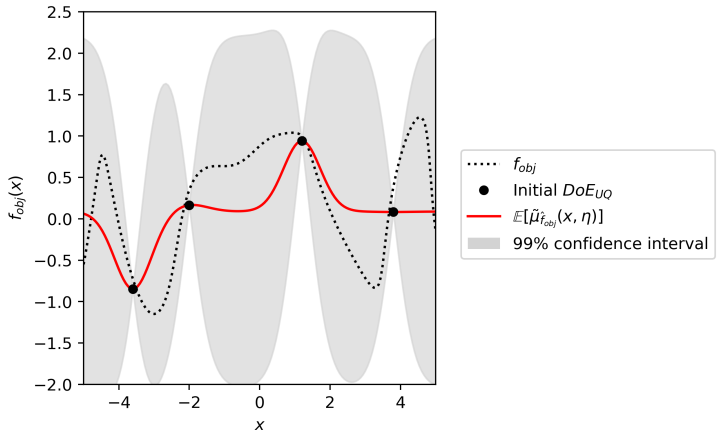


Figure 1.21: Illustration of the KL-GP approximation of the mean value $\tilde{\boldsymbol{\mu}}_{\hat{f}_{\text{obj}}}(\boldsymbol{x}, \boldsymbol{\eta})$, obtained from four DoE points.

Note that the GP approximation $\tilde{\boldsymbol{\mu}}_{\hat{f}_{\text{obj}}}(\boldsymbol{x}, \boldsymbol{\eta})$, illustrated in Figure 1.21, corresponds to the conditional expectation of $\tilde{f}_{\text{obj}}(\boldsymbol{x}, \boldsymbol{\Xi}, \boldsymbol{\eta})$ with respect to the random variable $\boldsymbol{\Xi}$. As a consequence, the presented confidence intervals only depict the uncertainty due to the GP interpolation of $\tilde{\boldsymbol{\mu}}_{\hat{f}_{\text{obj}}}(\boldsymbol{x}, \boldsymbol{\eta})$ and do not take into account uncertainty due to the use of disciplinary GPs. This is evidenced by the fact that, although the confidence intervals state that no uncertainty exists at the DoE_{UQ} points, the obtained approximation for $\tilde{\boldsymbol{\mu}}_{\hat{f}_{\text{obj}}}(\boldsymbol{x}, \boldsymbol{\eta})$ does not interpolate the deterministic objective function at these points. This uncertainty is introduced in the model by the term

$\sum_{k=1}^{M}\left(\sum_{l=2}^{P} \boldsymbol{a}_l^{\mathrm{T}} \boldsymbol{\varphi}_k \mathrm{H}_l(\boldsymbol{\Xi})\right) \tilde{\varphi}_k(\boldsymbol{x},\boldsymbol{\eta})$ of Eq. (1.78), not represented in Figure 1.21. Like in the EGO approach, the idea now is to sample new points into the disciplinary and objective function DoEs to improve the objective function approximation and find the global optimum. In EGMDO this is achieved thanks to a two-step uncertainty reduction strategy, detailed in the following.

Two-step uncertainty reduction strategy. The first step of the uncertainty reduction strategy consists on sampling new points to add to $\mathrm{DoE}_{\mathrm{UQ}}$, with the aid of an acquisition function. In EGMDO, the chosen acquisition function is a modified EI criterion [Dubreuil et al., 2018] inspired by the one proposed in the EGO algorithm. Similarly to the original EI criterion, the modified EI takes into account the uncertainty of the objective function model when choosing the next point to add to the $\mathrm{DoE}_{\mathrm{UQ}}$, thus promoting a good trade-off between the exploration of the design space and the exploitation of the surrogate function. It is defined as follows:

$$\mathrm{EI}(\boldsymbol{x}) = \mathbb{E}\left[\left(\hat{f}_{\min}(\boldsymbol{\Xi}) - \tilde{f}_{\mathrm{obj}}(\boldsymbol{x},\boldsymbol{\Xi},\boldsymbol{\eta})\right) \times \mathbb{1}_{\tilde{f}_{\mathrm{obj}}(\boldsymbol{x},\boldsymbol{\Xi},\boldsymbol{\eta}) \leq \hat{f}_{\min}(\boldsymbol{\Xi})}\right] \tag{1.79}$$

where $\hat{f}_{\min}(\boldsymbol{\Xi})$ is the current minimum, given by:

$$\hat{f}_{\min}(\boldsymbol{\Xi}) = \min_{\boldsymbol{x}\in \mathrm{DoE}_{\mathrm{UQ}}} \hat{f}_{\mathrm{obj}^{\mathrm{PCE}}}(\boldsymbol{\Xi}) \tag{1.80}$$

and:

$$\mathbb{1}_{\tilde{f}_{\mathrm{obj}}(\boldsymbol{x},\boldsymbol{\Xi},\boldsymbol{\eta}) \leq \hat{f}_{\min}(\boldsymbol{\Xi})} = \begin{cases} 0 & \text{if } \tilde{f}_{\mathrm{obj}}(\boldsymbol{x},\boldsymbol{\Xi},\boldsymbol{\eta}) > \hat{f}_{\min}(\boldsymbol{\Xi}) \\ 1 & \text{if } \tilde{f}_{\mathrm{obj}}(\boldsymbol{x},\boldsymbol{\Xi},\boldsymbol{\eta}) \leq \hat{f}_{\min}(\boldsymbol{\Xi}) \end{cases} \tag{1.81}$$

We remark that the EI criterion presented in Eq. (1.79) takes into account both sources of uncertainty: the one due to the use of disciplinary GPs and the one due to the combined KL-GP interpolation. Moreover, we note that, contrarily to the EI criterion proposed in the EGO algorithm, this modified EI cannot be computed analytically because the approximation given in Eq. (1.78) is not a GP. Instead, Eq. (1.79) is estimated by Monte Carlo sampling. The new point to add to $\mathrm{DoE}_{\mathrm{UQ}}$ is then the solution of the optimization problem:

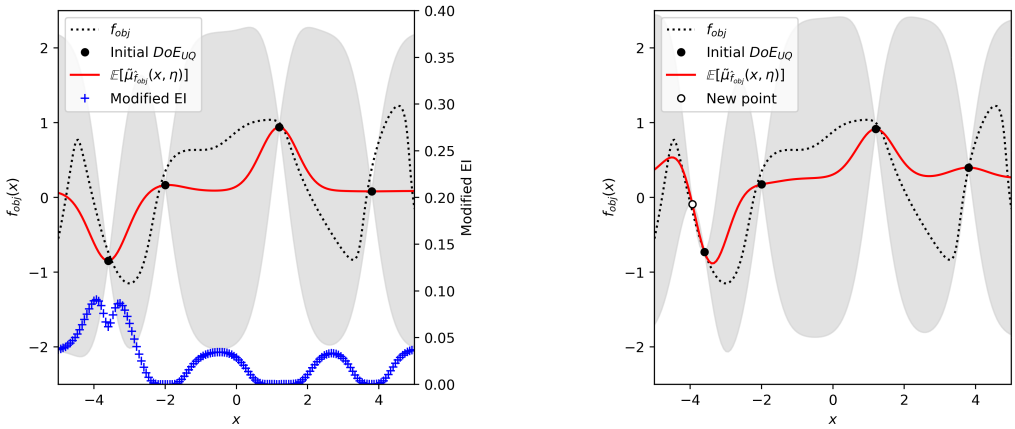
$$\boldsymbol{x}_{\mathrm{UQ}}^{\mathrm{new}} = \arg\max_{\boldsymbol{x}\in\mathscr{X}} \mathrm{EI}(\boldsymbol{x}) \tag{1.82}$$

At this new point, uncertainty quantification by PCE is performed and a new approximation of the random field $\tilde{f}_{\mathrm{obj}}(\boldsymbol{x},\boldsymbol{\Xi},\boldsymbol{\eta})$ is obtained. Figure 1.22 presents the modified EI criterion (Figure 1.22(a)) as well as the resulting approximation for $\tilde{\boldsymbol{\mu}}_{\hat{f}_{\mathrm{obj}}}(\boldsymbol{x},\boldsymbol{\eta})$ after adding $\boldsymbol{x}_{\mathrm{UQ}}^{\mathrm{new}}$ to $\mathrm{DoE}_{\mathrm{UQ}}$ (Figure 1.22(b)). As is shown, by adding a new point to $\mathrm{DoE}_{\mathrm{UQ}}$, the uncertainty due to the KL-GP approximation of $\tilde{\boldsymbol{\mu}}_{\hat{f}_{\mathrm{obj}}}(\boldsymbol{x},\boldsymbol{\eta})$ is reduced in the vicinity of this point. However, the uncertainty induced by the use of disciplinary GPs remains unchanged, as no points have been added to the disciplinary DoEs. The second step of the uncertainty reduction strategy addresses this issue by proposing a criterion for the improvement of the disciplinary GPs.

Enrichment of the disciplinary DoEs. According to the proposed criterion, two conditions must be met so that a given point $\boldsymbol{x}_i \in \mathrm{DoE}_{\mathrm{UQ}}$ is selected for improvement. First, the point must have some likelihood of being solution to Eq. (1.44). This likelihood is given by the probability $\mathrm{P}_{\min}(\boldsymbol{x}_i)$ that $\boldsymbol{x}_i$ is the minimum, written as:

$$\mathrm{P}_{\min}(\boldsymbol{x}_i) = \mathbb{P}\left(\min \hat{f}_{\mathrm{obj}}^{\mathrm{PCE}}(\boldsymbol{\Xi}) = \hat{f}_{\mathrm{obj}}^{\mathrm{PCE}}(\boldsymbol{x}_i,\boldsymbol{\Xi})\right) \;\forall\, \boldsymbol{x}_i \in \mathrm{DoE}_{\mathrm{UQ}} \tag{1.83}$$

Note that, in practice, $\mathrm{P}_{\min}(\boldsymbol{x}_i)$ is estimated by Monte Carlo method, using the $\boldsymbol{\Xi}$ samples drawn from the PCE approximations. $\mathrm{P}_{\min}(\boldsymbol{x}_i)$ is then required to be greater than $\frac{1}{n_{\mathrm{UQ}}}$ for $\boldsymbol{x}_i$ to be chosen



(a) Iteration 0.

(b) Iteration 1.

Figure 1.22: Illustration of (a) the modified EI criterion and (b) the resulting approximation for $\tilde{\boldsymbol{\mu}}_{\hat{f}_{\text{obj}}}(\boldsymbol{x},\boldsymbol{\eta})$ after adding $\boldsymbol{x}_{\text{UQ}}^{\text{new}}$ to DoE_{UQ}.

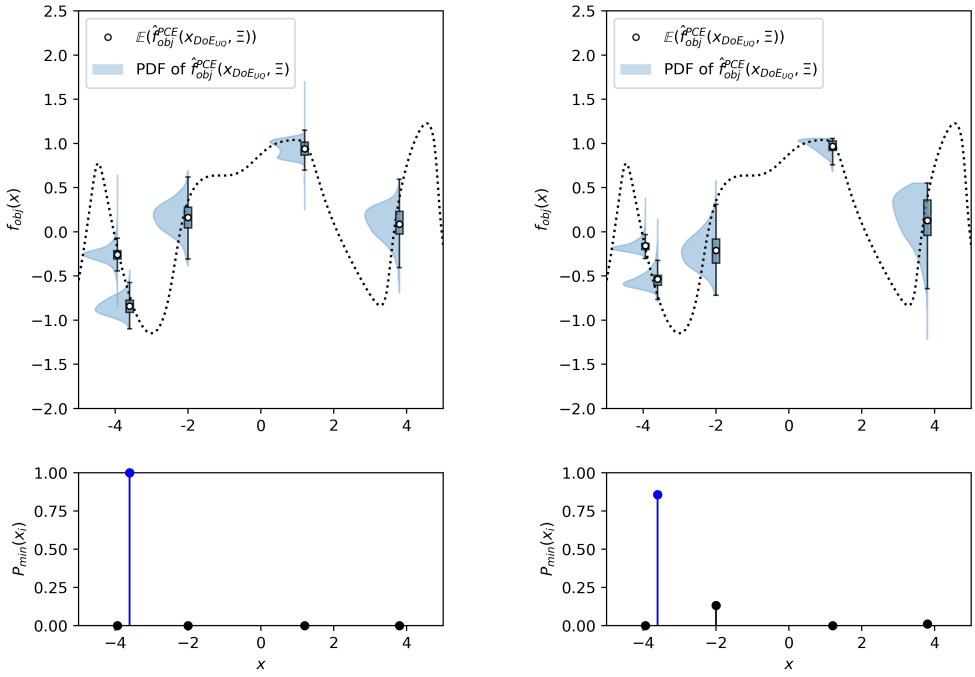
for enrichment. A second condition that must be met is that the coefficient of variation of the random objective function at $\boldsymbol{x}_i$ should be greater than a given threshold. This coefficient of variation is estimated as:

$$\text{CV}(\hat{f}_{\text{obj}}^{\text{PCE}}(\boldsymbol{x}_i,\boldsymbol{\Xi})) = \frac{\sqrt{\mathbb{V}(\hat{f}_{\text{obj}}^{\text{PCE}}(\boldsymbol{x}_i,\boldsymbol{\Xi}))}}{\mathbb{E}(\hat{f}_{\text{obj}}^{\text{PCE}}(\boldsymbol{x}_i,\boldsymbol{\Xi}))} \tag{1.84}$$

If both conditions are met, the disciplinary GPs are enriched until the coefficient of variation is less than the defined threshold for all $\boldsymbol{x}_i$ that verify $\text{P}_{\min}(\boldsymbol{x}_i) \geq \frac{1}{n_{\text{UQ}}}$. Figure 1.23 illustrates the impact of this second enrichment criterion on the PCE approximation of the objective function at the DoE_{UQ} points. As shown, by successively enriching the disciplinary solvers, we reduce the uncertainty due to the random variable $\boldsymbol{\Xi}$ at points which have high probability of being the minimum. Consequently, the KL-GP approximation is also improved. Additionally, we note that, although in the original EGMDO framework is it proposed to enrich all disciplinary surrogates, the Sobol sensitivity indices could also be used to determine the least accurate surrogate, as done in the EGMDA strategy.

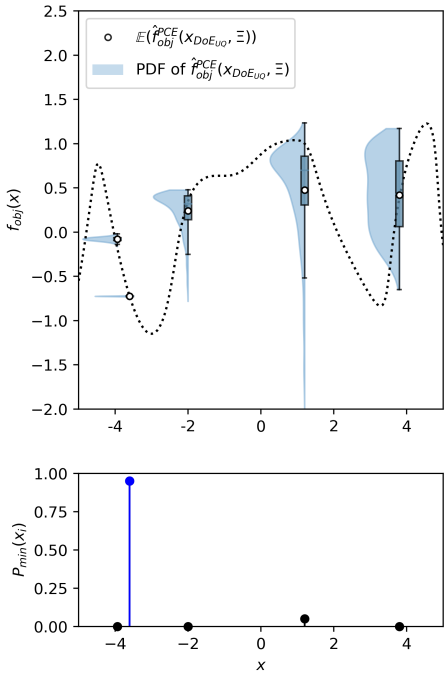
Stopping criterion. The previous section showed how the accuracy of the random field of Eq. (1.78) can be improved both by adding points to DoE_{UQ} and by reducing the uncertainty due to the use of disciplinary GPs. By iteratively performing these operations, the precision of the approximation increases around the relevant zones of the design space and the EGMDO algorithm is capable of identifying the global optimum. Nevertheless, like in the global optimization algorithms presented in Section 1.1.2, it remains difficult to define a stopping criterion for this type of approach. As a result, in EGMDO it is assumed that a fixed budget exists and that the algorithm is stopped when the entire budget has been spent. Figure 1.24 shows the obtained objective function approximation after respectively 5 and 10 iterations of the EGMDO algorithm. Realizations of $\tilde{f}_{\text{obj}}(\boldsymbol{x},\boldsymbol{\Xi},\boldsymbol{\eta})$ are equally drawn to illustrate the corresponding uncertainty.

As illustrated, as the algorithm progresses, the uncertainty of the objective function surrogate around the global optimum ($x^* \approx -3$) is reduced. Moreover, the presented objective function realizations show that in other areas of the design space, where the global minimum is unlikely to be found, the uncertainty has not been reduced. This implies that little computational effort has been spent trying to improve the disciplinary surrogate approximations in these regions.

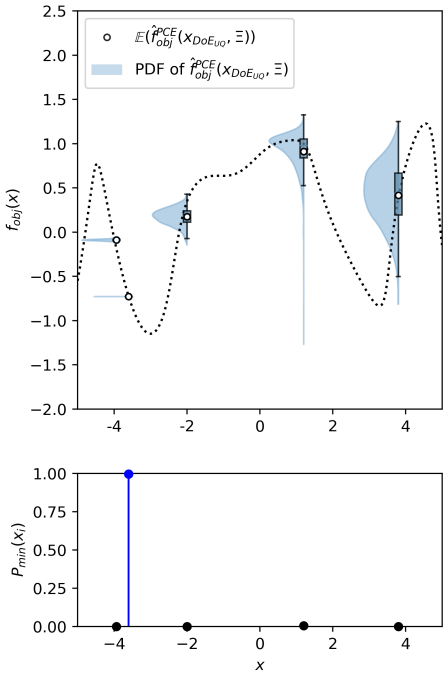


(a) Iteration 0

(b) Iteration 1

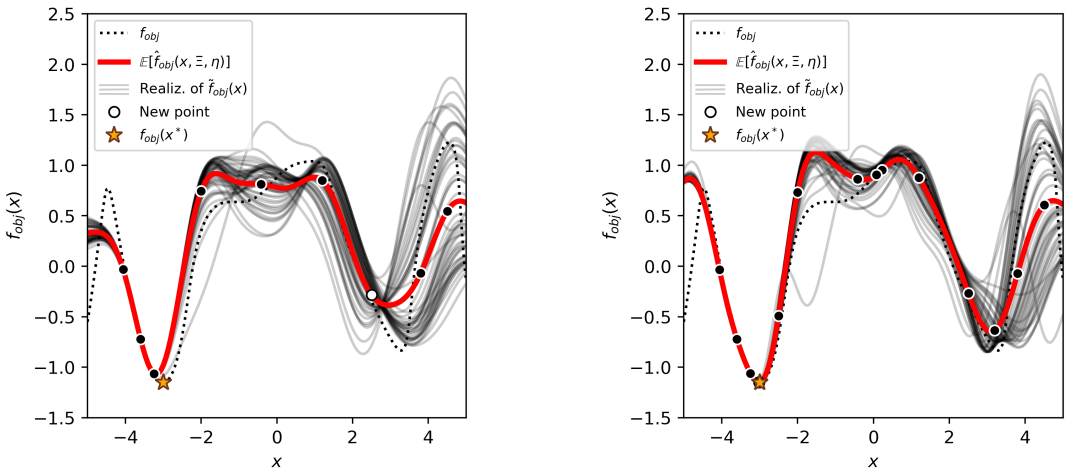


(c) Iteration 2



(d) Iteration 3

Figure 1.23: Enrichment of the disciplinary GPs in the EGMDO algorithm. The disciplinary surrogates are enriched at the point $\boldsymbol{x}_i \in \mathrm{DoE}_{\mathrm{UQ}}$ that maximizes $\mathrm{P}_{\min}(\boldsymbol{x}_i)$ in each iteration (in blue). Note that, even if the same design space point was chosen for enrichment multiple times, the coupling variable values added to the disciplinary DoEs were not the same.



(a) Iteration 5.

(b) Iteration 10.

Figure 1.24: Illustration of the obtained objective function approximation of after (a) 5 iterations and (b) 10 iterations of the EGMDO algorithm. The design of experiments DoE_{UQ} is represented in black dots and the last point added is identified with a white dot. Realizations of $\tilde{f}_{\text{obj}}(\boldsymbol{x},\boldsymbol{\Xi},\boldsymbol{\eta})$ are drawn to illustrate the corresponding uncertainty.

Indeed, during the 10 iterations performed, calls to the disciplinary solvers were only made at three points (three times at point $x \approx -3.6$, two times at point $x \approx -3.2$ and three times at point $x \approx -3$). At other points, where the probability of being minimum is low, the disciplinary solvers were never enriched. As a consequence, $\tilde{\boldsymbol{\mu}}_{\hat{f}_{\text{obj}}}(\boldsymbol{x},\boldsymbol{\eta})$ at these points is very far from the real objective function value (in the figure, this is particularly evident at points $x \approx -2.5$ and $x \approx 4.5$). The main steps of the EGMDO algorithm are summarized in Algorithm 7.

Algorithm 7 Efficient Global Multidisciplinary Design Optimization (EGMDO)

Input: $\text{DoE}_{\text{UQ}}, \hat{f}_i,\ i = 1,...,n_d$ ▷ Initial disciplinary GPs
  Compute PCE approximation at DoE_{UQ} points ▷ From Eq. (1.70)
  $j = 0$ ▷ Initialize iteration count
  while $j < j_{\max}$ **do**
    Compute KL-GP approximation ▷ From Eq. (1.74)
    Find $\boldsymbol{x}_{\text{new}}$ ▷ Maximize modified EI (Eq. (1.79))
    Add $\boldsymbol{x}_{\text{new}}$ to DoE_{UQ}
    Enrich disciplinary GPs if needed
    $j = j + 1$ ▷ Increase iteration count
  end while
  $f_{\min} = \min_{\boldsymbol{x} \in \text{DoE}_{\text{UQ}}} \boldsymbol{\mu}_{\hat{f}^{\text{PCE}}_{\text{obj}}}(\boldsymbol{x})$
  $\boldsymbol{x}_{\min} = \arg f_{\min}$
Output: $\boldsymbol{x}_{\min}, f_{\min}$

Comparison with other MDO formulations. Although EGMDO solves the approximated MDA to obtain random MDA solutions, EGMDO cannot be called an MDF approach. Indeed, because EGMDO does not seek to accurately find the set of converged coupling variables in regions of

the design space where the global optimum is unlikely to be, the EGMDO formulation is actually closer to the IDF approach. Moreover, in EGMDO the disciplinary surrogates are built over the design and coupling variable spaces and uncertainty of the coupling variables is accounted for in the optimization of the acquisition function. This is similar to the IDF approach where the coupling variables are added to the set of design variables and the residual of the coupling variables is accounted for when computing the step at each optimizer iteration. Working in both design and coupling variable spaces ultimately allows both the IDF and EGMDO formulations to save computational cost.

Limitations of the EGMDO strategy. Like other approaches, the EGMDO strategy also presents some limitations. Indeed, in the original framework, no constraint handling strategy is proposed. However, if the constraint functions depend on any set of the coupling variables, the use of disciplinary surrogates will result in uncertain constraint functions. It can then be expected that EGMDO will encounter issues if the initial constraint approximations are not sufficiently accurate or if they lead to an empty feasible set. Another limitation is that the uncertainty reduction strategy relies on the fact that the objective function depends on a subset of the coupling variables. If only the constraint functions depend on the coupling variable values, the presented uncertainty reduction strategy is not applicable. Finally, because it is based on surrogate approximations, EGMDO also suffers from the curse of dimensionality.

1.2.6 Summary of MDAO

In this section we introduced the main tools to solve the MDA and MDO problems. Monolithic and partitioned approaches for the resolution of the MDA were first introduced. In monolithic approaches, a single system of equations is solved by the same numerical solver. Contrarily, in partitioned approaches, the system is split into subdomains, each handled by a dedicated disciplinary solver. Due to their offered flexibility in adding or replacing disciplinary solvers, partitioned approaches are often preferred over monolithic approaches. To solve the partitioned coupled system of equations, fixed-point methods, like the Jacobi, Gauss-Seidel or Newton's methods can be used. These methods, however, typically require many calls to the disciplinary solvers, which may be expensive numerical solvers. To reduce the computational burden, the EGMDA strategy replaces the disciplinary solvers involved in the MDA by GP approximations. By adaptively enriching the disciplinary surrogates, the EGMDA strategy is capable of finding the MDA solution at a reduced computational cost.

The existing formulations to solve the MDO problem were introduced next. The MDF approach solves the MDA at each iteration of the optimizer and thus always returns a consistent system design, even if the optimization is terminated early. Contrarily, the IDF approach adds the coupling variables to the set of design variables and uses consistency constraints to assure the multidisciplinary consistency at the optimal solution. Although this results in less disciplinary solver calls, the resulting optimization problem is harder to solve, due to the increased number of design variables and constraint functions. Gradient-based and surrogate-based MDO were equally discussed. When disciplinary gradients are available, the coupled-adjoint method can be used to obtain the objective and constraint function derivatives. However, when gradient information is not available, approximating the derivatives of the coupled system using approximation methods can become computationally challenging. Surrogate-based optimization, which does not require derivative information, becomes an appealing alternative. However, most surrogate-based approaches do not take advantage of the partitioned nature of the multidisciplinary system. An exception is the EGMDO algorithm which uses adaptively enriched disciplinary surrogates to reduce the computational cost of the MDO problem.

1.3 Tools for numerical optimization

Most of the approaches presented in this chapter are already implemented in different programming languages. Although it is out of the scope of this thesis to provide a detailed overview of all existing implementations, we do find it relevant to present a few tools due to their extensive use during our numerical tests.

The Surrogate Modeling Toolbox (SMT). SMT is an open-source Python package that contains a collection of surrogate models, sampling techniques, and benchmarking functions [Bouhlel et al., 2019]. SMT stands apart from other surrogate modeling libraries due to its focus on derivatives and, more recently, its capability to handle hierarchical and mixed inputs as well as multi-fidelity data [Saves et al., 2024]. A version of the EGO algorithm is equally available in SMT. In this work, SMT is used in sampling tasks and for the construction of GP approximations.

The OpenMDAO framework. OpenMDAO is an open-source MDO framework [Gray et al., 2019] which facilitates derivative computation of coupled systems. One of the main features of OpenMDAO is that it enables a modular implementation of computational models through units of code called components. Data transfer between the different components is then performed by OpenMDAO to solve the MDA. Implementations of the Jacobi, Gauss-Seidel (with or without acceleration) and Newton solvers are available. Additionally, provided that derivatives at a disciplinary solver level are given, OpenMDAO efficiently computes total derivatives of the multidisciplinary system to perform gradient-based optimization. In this work, OpenMDAO is used to set up our numerical tests involving MDA or MDO problems.

The WhatsOpt collaborative environment. WhatsOpt is a web application developed to support overall vehicle design activities at ONERA. It allows users to define MDA problems in terms of disciplines and data exchanges [Lafage et al., 2019]. Once the problem has been defined, WhatsOpt is capable of generating the OpenMDAO skeleton code where implementations of the disciplinary solvers can be plugged into. In this work, WhatsOpt is used to generate the skeleton code for our numerical tests.

The OpenTURNS software. OpenTURNS is an open-source software platform dedicated to uncertainty propagation by probabilistic methods [Baudin et al., 2015]. It includes a set of efficient mathematical methods for uncertainty quantification and is easily used with external black box solvers. In this work, OpenTURNS is used to perform tasks such as uncertainty quantification, uncertainty propagation and sensitivity analysis.

Chapter summary:

In this chapter we discussed the optimization of multidisciplinary systems. To do so, we first introduced the following concepts or tools for the numerical optimization of a given performance function (with or without constraints):

- *Gradient-based optimizers, which use derivative information to find a descent direction, but may get stuck on local optima.*
- *Surrogate-based optimizers, which use an approximation of the objective function to search directly for the global optimum.*

To solve the coupling between the different disciplines of the system, typically defined as a fixed-point problem, the following algorithms were then presented:

- *The Jacobi and Gauss-Seidel algorithms, which call the disciplinary solvers iteratively until the solution of the MDA is found.*
- *The EGMDA algorithm which uses adaptively enriched disciplinary GPs to reduce the computational cost of solving the MDA.*

Finally, to handle the MDO problem, different formulations were introduced:

- *The MDF approach, which solves the MDA at each optimizer iteration and therefore always returns a consistent system design.*
- *The IDF approach, which adds the coupling variables to set of design variables and uses consistency constraints to ensure multidisciplinary consistency at the optimal solution.*
- *The EGMDO algorithm, a global optimization algorithm, specifically developed for the optimization of partitioned multidisciplinary systems.*

CHAPTER 2

Model order reduction

Contents

In this chapter we will:

- *Introduce parameterized PDEs.*
- *Introduce projection-based Model Order Reduction (pMOR).*
- *Introduce the interpolation of local reduced-order bases.*
- *Introduce MOR for multidisciplinary systems.*
- *Introduce the DPOD+I algorithm.*

In this chapter we introduce the concepts or tools to obtain computationally cheaper approximations of the disciplinary solvers whose governing equations are PDEs. To do so, we first introduce the numerical resolution of the parameterized PDEs that describe the physical behavior of the aerodynamics and structural disciplines. Next, we introduce projection-based Model Order Reduction (pMOR) as a method for reducing the computational burden associated with the resolution of the parameterized PDEs in a multi-query context. Finally, the DPOD+I strategy that combines MOR with GP interpolation is presented. This strategy aims at reducing the computational cost of multi-query problems requiring several resolutions of an MDA involving disciplinary solvers whose governing equations are PDEs.

2.1 Parameterized partial differential equations

Physical phenomena are described by relevant quantities, called state variables, that depend on their position in the spatial or temporal domains. For example, in fluid dynamics, the state variables can be the speed and pressure of the moving fluid. Similarly, in structural mechanics, we often wish to find the displacement field of an object subject to a load. These physical phenomena can be modeled using PDEs that are solved for the state variables, for given a set of initial and boundary conditions. Certain changes in design parameters, however, lead to changes in the boundary conditions, in the shape of the physical domain or in the material properties. For multi-query applications, such as optimization or uncertainty quantification, the PDEs must be solved for every different set of design parameters, leading to an important computational burden. Model order reduction (MOR) methods have become a popular tool to reduce the computational cost of solving PDEs [Lieu et al., 2006, Lassila et al., 2014]. These methods work by building low-dimensional models of the governing equations, which are cheaper to evaluate than their high-dimensional counterparts. In this section, we address the numerical resolution of the PDEs and introduce the governing equations used to model the aerodynamics and structural disciplines. In practice, these PDEs represent the high-dimensional or full-order models whose cheaper, lower-dimensional approximation we will seek later on in this chapter.

2.1.1 Numerical resolution of PDEs

The solution of the PDEs used to model different physical phenomena usually cannot be obtained analytically and we often resort to numerical methods. These methods first discretize the differential equations over the entire domain, and then solve for the state variables. Different numerical methods exist including finite-difference, finite-volume or finite-element methods. In finite-difference methods, the derivatives of the PDEs on each nodal point are replaced by an approximation. Rewriting the original equations using the approximated derivatives yields a discrete system of equations which can be used to obtain the state variables at the nodal points.

In finite-volume methods, the domain is discretized into control volumes, called cells, within which the conservation laws apply. Solving the integral form of the governing equations for each cell allows to obtain the state variables at the cell centroids or at the cell vertices. Lastly, in finite-element methods the domain is divided into elements, within which the solution of the governing equations is approximated by simple functions (also called shape functions). Thanks to the shape functions, a continuous representation of the state variables can be obtained.

Finite-difference, finite-volume and finite-element methods have historically been used to solve problems arising from different disciplines. Indeed, finite-difference and finite-volume methods have more often been used to solve fluid dynamics problems, whereas finite-element methods are usually employed to solve structural analysis or heat transfer problems. In this work both fluid dynamics and structural analysis applications are considered, thus both finite-element and finite-volume methods are used. In the following, we present the partial differential equations modeling the aerodynamics and structural disciplines.

2.1.2 Disciplinary models

2.1.2.1 Compressible Euler equations

To obtain the aerodynamic loads, the compressible Euler equations can be used. These equations enforce the conservation of mass (Eq. (2.1a)), momentum (Eq. (2.1b)) and energy (Eq. (2.1c)), but neglect viscous effects and thermal conductivity. They are written as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho v_i) = 0, \quad i = \{1,2,3\} \tag{2.1a}$$

$$\frac{\partial}{\partial t}(\rho v_i) + \frac{\partial}{\partial x_i}(\rho v_i v_j + p) = 0, \quad i,j = \{1,2,3\} \tag{2.1b}$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_i}(v_j(\rho E + p)) = 0, \quad i,j = \{1,2,3\} \tag{2.1c}$$

where ρ is the fluid density, $v \in \mathbb{R}^3$ is the fluid velocity, p is the pressure and E is the total energy per unit mass. To solve the Euler equations we use a finite-volume method, which discretizes the spatial domain into N control volumes. Denoting, $\mathrm{U} = (\rho, \rho \boldsymbol{v}, \rho E)$ the state vector and $\mathrm{F}(\mathrm{U}) = (\rho \boldsymbol{v}, \rho \boldsymbol{v} \otimes \boldsymbol{v} + p\mathbf{I}, \rho E \boldsymbol{v} + p\boldsymbol{v})$ the flux vector, where $\mathbf{I}$ is the identity matrix, we can write the volume integral over Ω_i as:

$$\int_{\Omega_i} \frac{\partial \mathrm{U}}{\partial t} d\Omega_i + \int_{\Gamma_i} \mathrm{F}(\mathrm{U}) \cdot \boldsymbol{n}_i d\Gamma_i = 0, \quad i = 1, ..., N \tag{2.2}$$

where Ω_i is i^{th} the control volume, $\Gamma_i = \partial\Omega_i$ is the surface of the control volume and $\boldsymbol{n}_i$ is the normal vector to the surface. Solving the Euler equations for the state variables allows the computation of certain quantities of interest, such as aerodynamic lift or drag. However, due to their nonlinear nature, the numerical resolution of the Euler equations can often represent an important computational burden, especially in a multi-query context. As a result, simpler aerodynamics models are often preferred, despite their reduced accuracy. One such model, is the Vortex Lattice method, introduced in the following.

2.1.2.2 Vortex Lattice method

The Vortex Lattice Method (VLM) [Katz and Plotkin, 2001, Anderson, 2016] is derived from potential flow theory, which assumes the flow to be incompressible, inviscid and irrotational. In the VLM, the lifting surface is modeled as a thin plate, discretized into a structured mesh made of N elements or panels. At a quarter of the length of each panel, a horseshoe vortex is placed,

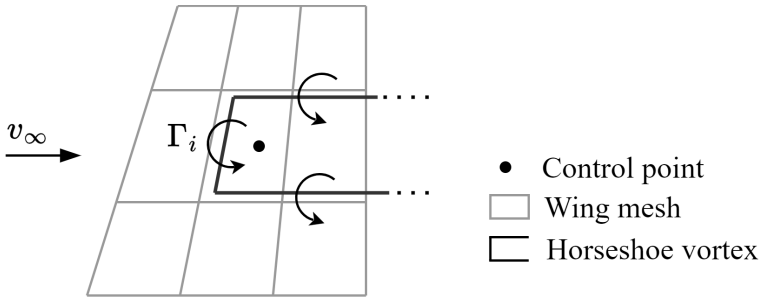


Figure 2.1: Illustration of the VLM. A horseshoe vortex is placed at a quarter length of each element of the surface mesh.

as illustrated in Figure 2.1. Each horseshoe vortex has a strength (or circulation) Γ_i, which can be obtained thanks to the conservation of flux. It allows to write the total velocity as:

$$v = v_\infty + \nabla\phi \tag{2.3}$$

where v_∞ is the freestream velocity and ϕ is the velocity potential. This velocity potential is obtained at any control point i by summing the contribution of all horseshoe vortices, accordingly:

$$\nabla\phi_i = \sum_{j=1}^{N} a_{ij}\Gamma_j \tag{2.4}$$

where a_{ij} are the aerodynamic influence coefficients, representing the induced velocity perturbation on the control point i due to the horseshoe vortex on panel j. To determine the strength of each vortex, the VLM imposes that the velocity component normal to the panel surface must be zero at the control points. Combining this condition with Eq. (2.3) and Eq. (2.4) results in following linear system of equations:

$$v_i \cdot \boldsymbol{n}_i = \left(v_\infty + \sum_{j=1}^{N} a_{ij}\Gamma_j \right) \cdot \boldsymbol{n}_i = 0, \;\; i = 1, ..., N \tag{2.5}$$

where $\boldsymbol{n}_i$ is the vector normal to the panel at point i. Once the strength of each vortex is known, it is possible to obtain the aerodynamic forces acting upon the wing surface. Because the VLM solves a linear system of equations, it typically does not require as much computational effort as solving the Euler equations. Regardless, due to the simplifications made, the range of application of the VLM is limited to the analysis of low speed flows around thin surfaces.

2.1.2.3 Linear elastic model

To model the structural displacement, a linear elastic model can be used under the assumption of small deformations. In this case, the governing equations are:

$$\text{div}\,\boldsymbol{\sigma} + \boldsymbol{f} = 0 \tag{2.6}$$

where $\boldsymbol{\sigma}$ is the stress tensor, $\boldsymbol{f}$ is the vector of forces per unit volume and div denotes the divergence operator:

$$\text{div}\,\boldsymbol{\sigma} = \frac{\partial \sigma_{ij}}{\partial x_i}, \;\; i, j = \{1, 2, 3\} \tag{2.7}$$

In the case of linear elasticity, the stress tensor can be related to the structural displacement $u \in \mathbb{R}^3$ through Hooke's law:

$$\boldsymbol{\sigma} = \boldsymbol{C}\varepsilon(u) \tag{2.8}$$

where C is the elasticity tensor and ε is the strain tensor, defined as:

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \quad i,j = \{1,2,3\} \tag{2.9}$$

If the material is homogeneous and isotropic, the elasticity tensor depends on only two coefficients: the Young's modulus E and the Poisson coefficient ν. Using these coefficients, Hooke's law can be rewritten as:

$$\sigma_{ij} = \frac{E}{1+\nu}\left(\varepsilon_{ij}(u) + \frac{\nu}{1-2\nu}\left(\varepsilon_{11}(u) + \varepsilon_{22}(u) + \varepsilon_{33}(u)\right)\delta_{ij}\right), \quad i,j = \{1,2,3\} \tag{2.10}$$

where δ_{ij} is the Kronecker delta. Equation (2.6) is typically solved via a finite-element method. To do so, we first write the variational form, as follows:

$$\int_\Omega \boldsymbol{\sigma} v d\Omega - \int_\Omega \boldsymbol{f} v d\Omega - \int_\Gamma \boldsymbol{\sigma}\cdot\boldsymbol{n} v d\Gamma = 0 \tag{2.11}$$

where $\Omega \subset \mathbb{R}^3$ is the considered spatial domain, $\Gamma = \partial\Omega$ denotes the bounds of Ω, $\boldsymbol{n}$ is the vector normal to Γ and $v \in V$ is any test function which satisfies the boundary conditions. Discretizing Eq. (2.11) into N finite elements and integrating over each element leads to following linear system of equations:

$$\mathbf{K}\boldsymbol{u} = \mathbf{F} \tag{2.12}$$

where $\mathbf{K} \in \mathbb{R}^{N\times N}$ is the rigidity matrix and $\mathbf{F} \in \mathbb{R}^N$ is the vector of external loads. As for the aerodynamics discipline, other structural models also exist in the literature which can be used to model the structural displacement of the wing. However, in this work, only the linear elastic model is considered.

2.1.3 Numerical tools

Several numerical tools exist to solve the PDEs presented above. While it is out of the scope of this work to provide a comprehensive overview for all existing tools, we do find it relevant to present a select few, due to their extensive use during our numerical tests.

The SU2 numerical solver. SU2 [Palacios et al., 2013, Economon et al., 2016] is an open-source tool for solving PDEs. SU2 has been mostly developed to address fluid dynamics problems, including those dealing with compressible, turbulent flows. SU2 is also capable of providing gradient information, through the use of an adjoint method, making this tool useful for a wide range of applications, including optimization and uncertainty quantification. In this work, SU2 is used to solve the compressible Euler equations modeling the external flow around a wing.

The VLM solver. Implementations of the VLM method have also been made available through open-source software, including the Athena Vortex Lattice (AVL) [Drela and Youngren, 2004] and NASA's VSPAERO [McDonald and Gloudemans, 2022]. For simplification, in this work, an in-house implementation is used during our numerical tests.

The Code Aster numerical solver. Code Aster [EDF, 2017] is an open-source code which uses the finite-element method to solve different types of structural mechanics problems, including those dealing with static, dynamic or thermal analysis. In this work, Code Aster is used to model the linear elastic behavior of the wing subject to the aerodynamic loads.

2.2 Projection-based Model Order Reduction

Projection-based model order reduction has proven itself as method to reduce the computational cost of solving PDEs while retaining a relatively high level of accuracy. It relies on the idea that although the solution of the discretized PDEs is in general a high-dimensional vector, it can often be represented in a low-dimensional subspace. Let $\boldsymbol{w}(\boldsymbol{\mu}) \in \mathscr{W} \subset \mathbb{R}^N$ denote the state vector, solution of a parameterized partial differential equation, with N the number of degrees of freedom resulting from the discretization of the PDEs and $\boldsymbol{\mu}$ the vector of parameters $\boldsymbol{\mu} \in \mathscr{P} \subset \mathbb{R}^p$. Projection-based reduced order models approximate the state vector in a subspace of dimension $n \ll N$, as follows:

$$\tilde{\boldsymbol{w}}(\boldsymbol{\mu}) \approx \boldsymbol{w}_{\text{ref}} + \Phi\boldsymbol{\alpha}(\boldsymbol{\mu}) \tag{2.13}$$

where $\boldsymbol{w}_{\text{ref}}$ is an element of $\mathscr{W}$, $\Phi = [\phi_1, ..., \phi_n] \in \mathbb{R}^{N\times n}$ is a given reduced-order basis and $\boldsymbol{\alpha}(\boldsymbol{\mu}) = \{\alpha_1(\boldsymbol{\mu}), ..., \alpha_n(\boldsymbol{\mu})\} \in \mathbb{R}^n$ is the vector of reduced (or generalized) coordinates. Different methods exist for constructing the reduced-order basis and for obtaining the generalized coordinates. Some of these methods are presented in the following.

2.2.1 The generalized coordinates

Projection-based MOR is typically a data-driven (or simulation-driven) approach. This means that we first build a reduced-order model based on a collection of state vectors, corresponding to the input parameters $\{\boldsymbol{\mu}_i\}_{i=1}^d \in \mathscr{P}_t \subset \mathscr{P}$. Then, we use the built reduced-order model to predict the state vector at a new point $\boldsymbol{\mu}^* \notin \mathscr{P}_t$. In this section we presume that the reduced order basis is readily available and present two methods for predicting the state vector at the new point $\boldsymbol{\mu}^*$. The first method relies on the representation of the governing equations in the low-dimensional space and therefore is considered an intrusive approach. The second method relies on the interpolation of the generalized coordinates and therefore is considered a non-intrusive approach.

2.2.1.1 Galerkin or Petrov-Galerkin projection

Consider the following linear parameterized system of equations, representing either the linear full-order model of a given discipline or a linearization step in the resolution of a given non-linear model:

$$\boldsymbol{A}(\boldsymbol{\mu})\boldsymbol{w}(\boldsymbol{\mu}) = \boldsymbol{b}(\boldsymbol{\mu}) \tag{2.14}$$

where $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, $\boldsymbol{b} \in \mathbb{R}^{N\times 1}$ and where time-dependency is not considered for simplicity. An approximation of the full-order model can be obtained in the low-dimensional space by replacing Eq. (2.13) into Eq. (2.14) and projecting the resulting overdetermined system of equations onto the subspace defined by the left reduced-order basis $\Psi \in \mathbb{R}^{N\times n}$, accordingly:

$$\underbrace{\left(\Psi^\intercal \boldsymbol{A}(\boldsymbol{\mu})\Phi\right)}_{\boldsymbol{A}_r(\boldsymbol{\mu})}\boldsymbol{\alpha}(\boldsymbol{\mu}) = \underbrace{\Psi^\intercal\left(\boldsymbol{b}(\boldsymbol{\mu}) - \boldsymbol{A}(\boldsymbol{\mu})\boldsymbol{w}_{\text{ref}}\right)}_{\boldsymbol{b}_r(\boldsymbol{\mu})} \tag{2.15}$$

In the general case, the left and right reduced-order bases may differ ($\Psi \neq \Phi$) and the obtained projection is denoted a Petrov-Galerkin projection. When the left reduced-order basis is chosen to be the same as the right reduced-order basis ($\Psi = \Phi$), the obtained projection is denoted a Galerkin projection. The choice between a Petrov-Galerkin or a Galerkin projection is typically related to the minimization of the residual term that arises due to the fact that $\tilde{\boldsymbol{w}}(\boldsymbol{\mu})$ is not a solution to the full-order model [Grimberg et al., 2020]. Regardless of the choice, however, both Galerkin and Petrov-Galerkin projections are intrusive approaches that require the construction of the high-dimensional matrices $\boldsymbol{A}(\boldsymbol{\mu}^*)$ and $\boldsymbol{b}(\boldsymbol{\mu}^*)$ for every new queried point $\boldsymbol{\mu}^*$. Moreover, although solving the projected equations is computationally less intensive than solving the

full-order model, assembling these matrices in a multi-query context can represent significant computational burden. To ease this computational burden, hyperreduction methods have been proposed. They are only briefly introduced in the following, but a more thorough review can be found in [Farhat et al., 2020].

Hyperreduction. Hyperreduction methods work by either building an approximation of the full-order matrices $\boldsymbol{A}(\boldsymbol{\mu})$ and $\boldsymbol{b}(\boldsymbol{\mu})$ or building an approximation of their low-dimensional equivalents $\boldsymbol{A}_r(\boldsymbol{\mu})$ and $\boldsymbol{b}_r(\boldsymbol{\mu})$ and thus avoiding the construction of the full-order matrices altogether. When the approximation is built for the full-order matrices, the hyperreduction methods are called *approximate-then-project* methods. Contrarily, when the approximation is built for the reduced matrices, the hyperreduction methods are called *project-then-approximate* methods. Examples of *approximate-then-project* methods include the gappy POD method [Everson and Sirovich, 1995], the Empirical Interpolation Method (EIM) [Barrault et al., 2004, Grepl et al., 2007], as well as its discrete version (DEIM) [Chaturantabut and Sorensen, 2010], and the Gauss-Newton with Approximated Tensors (GNAT) [Carlberg et al., 2011, Carlberg et al., 2013] method. *Project-then-approximate* methods were developed more recently. They include the Empirical Curbature Method (ECM) [Hernandez et al., 2017] and the Energy-Conserving Sampling and Weighting (ECSW) [Farhat et al., 2014, Grimberg et al., 2021] methods.

2.2.1.2 Interpolation of the generalized coordinates

Instead of seeking to reconstruct the full-order matrices for every new parameter $\boldsymbol{\mu}^*$, other authors propose to interpolate the generalized coordinates $\alpha_i(\boldsymbol{\mu})$, $i = 1,...,n$ [Coelho et al., 2009, Hesthaven and Ubbiali, 2018]. To do so, a set of state vectors is first projected onto the corresponding reduced-order basis, as follows:

$$\alpha_i(\boldsymbol{\mu}_j) = \langle \boldsymbol{w}(\boldsymbol{\mu}_j) - \boldsymbol{w}_{\text{ref}}, \phi_i \rangle, \quad i = 1,...,n, \quad \boldsymbol{\mu}_j \in \mathscr{P}_t \tag{2.16}$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product between two vectors. Then, a surrogate model is built for each of the generalized coordinates. Using the surrogate approximations, the full-order state vector can be obtained at any new point $\boldsymbol{\mu}^*$, accordingly:

$$\tilde{\boldsymbol{w}}(\boldsymbol{\mu}^*) \approx \boldsymbol{w}_{\text{ref}} + \sum_{i=1}^{n} \phi_i \hat{\alpha}_i(\boldsymbol{\mu}^*) \tag{2.17}$$

where $\hat{\alpha}_i$ denotes the surrogate model associated with the generalized coordinate α_i. Because this approach does not require access to the full-order matrices, it is considered a non-intrusive approach. Moreover, there is no restriction on the choice of interpolation method; any method capable of approximating scalar functions can be used (for instance the GP approximations, introduced Section 1.1.2.2, can be used). In this work, interpolation of the generalized coordinates is preferred over Galerkin or Petrov-Galerkin projections. This choice is motivated by the assumption of the black-box nature of the disciplinary solvers.

2.2.2 The reduced-order basis

Now that it has become clear how to obtain the generalized coordinates at new queried points, we address the construction of the reduced-order basis. Although many approaches exist in the literature, in this section only a select few are presented. They are the Proper Orthogonal Decomposition (POD) method of snapshots, the greedy procedure and the Proper Generalized Decomposition (PGD).

2.2.2.1 Proper Orthogonal Decomposition

One of the most popular methods for building the reduced-order basis is through the POD method of snapshots [Sirovich, 1987, Chatterjee, 2000]. In the POD method, the reduced order basis is trained from a set of precomputed state vectors $\boldsymbol{w}(\boldsymbol{\mu}_i)$, $i = 1,...,d$, called snapshots. The idea is then to look for the orthonormal basis $\Phi = [\phi_1,...,\phi_n]$ that minimizes the projection error in the least-squares sense, as follows:

$$\phi_1,...,\phi_n = \arg\min_{\phi_1,...,\phi_n \in \mathbb{R}^N} \sum_{i=1}^{d} \left\| \boldsymbol{w}(\boldsymbol{\mu}_i) - \Pi_\Phi(\boldsymbol{w}(\boldsymbol{\mu}_i)) \right\|_2^2 \tag{2.18}$$

where:

$$\Pi_\Phi(\boldsymbol{w}(\boldsymbol{\mu}_i)) = \boldsymbol{w}_{\text{ref}} + \sum_{j=0}^{n} \langle (\boldsymbol{w}(\boldsymbol{\mu}_i) - \boldsymbol{w}_{\text{ref}}), \phi_j \rangle \phi_j \;\text{ such that }\; \langle \phi_i, \phi_j \rangle = \delta_{ij} \tag{2.19}$$

It turns out that solving Eq. (2.18) is equivalent to finding the first n left singular vectors of the snapshot matrix $\boldsymbol{S} \in \mathbb{R}^{N \times d}$, defined as:

$$\boldsymbol{S} = [\boldsymbol{w}(\boldsymbol{\mu}_1) - \boldsymbol{w}_{\text{ref}}, ..., \boldsymbol{w}(\boldsymbol{\mu}_d) - \boldsymbol{w}_{\text{ref}}] \tag{2.20}$$

where $\boldsymbol{w}_{\text{ref}}$ is typically chosen as the mean of the snapshots, i.e. $\boldsymbol{w}_{\text{ref}} = \frac{1}{d}\sum_{i=1}^{d} \boldsymbol{w}(\boldsymbol{\mu}_i)$. The Singular Value Decomposition (SVD) of $\boldsymbol{S}$ can then be written as:

$$\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V} \tag{2.21}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1,...,\boldsymbol{u}_N] \in \mathbb{R}^{N \times N}$ and $\boldsymbol{V} = [\boldsymbol{v}_1,...,\boldsymbol{v}_d] \in \mathbb{R}^{d \times d}$ are, respectively, the matrices of the left and right singular vectors of $\boldsymbol{S}$ and $\boldsymbol{\Sigma}$ is a matrix containing the singular values of $\boldsymbol{S}$ in its main diagonal, i.e. $\boldsymbol{\Sigma}_{ij} = \sigma_j$, if $i = j$, and $\sigma_1 \geq ... \geq \sigma_d \geq 0$. The number of columns of $\boldsymbol{U}$ to retain can be determined by finding the minimum integer value of $n \leq d$ for which:

$$\frac{\sum_{j=1}^{n} \sigma_j^2}{\sum_{j=1}^{d} \sigma_j^2} > \eta, \;\; 0 < \eta < 1 \tag{2.22}$$

In practice, Eq. (2.22) defines a threshold for the projection error in the least-squares sense (see Eq. (2.18)). The reduced-order basis can then be written as:

$$\Phi = [\boldsymbol{u}_1,...,\boldsymbol{u}_n] \tag{2.23}$$

where $\boldsymbol{u}_j$ is the left singular vector associated with the singular value σ_j. Algorithm 8 summarizes the main steps of the POD method.

Algorithm 8 Proper Orthogonal Decomposition (POD) method of snapshots

Input: $\boldsymbol{w}(\boldsymbol{\mu}_i)$, $i = 1,...,d, \eta$ ▷ Precomputed snapshots and error threshold
Build $\boldsymbol{S} = [\boldsymbol{w}(\boldsymbol{\mu}_1) - \boldsymbol{w}_{\text{ref}}, ..., \boldsymbol{w}(\boldsymbol{\mu}_d) - \boldsymbol{w}_{\text{ref}}]$ ▷ Assemble snapshot matrix
Compute $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}$ with $\boldsymbol{U} = [\boldsymbol{u}_1,...,\boldsymbol{u}_N]$ ▷ Singular Value Decomposition
Choose n that satisfies Eq. (2.22) ▷ Error criterion
Output: $\Phi = [\boldsymbol{u}_1,...,\boldsymbol{u}_n]$

Equivalence between POD and other methods. POD is also known under other names, notably as Principal Component Analysis (PCA) [Hotelling, 1933] in the field of statistical analysis and as Karhunen-Loève (KL) decomposition [Karhunen, 1947, Loève, 1955] in the theory of stochastic processes. Equivalence between these three methods is discussed in detail in [Liang et al., 2002, Wu et al., 2003].

***A priori* versus adaptive sampling.** The POD method provides a reduced-order basis that optimally represents the sampled snapshots in the least-squares sense. However, the error criterion in Eq. (2.22) does not provide any information concerning the regions of the parameter space where no snapshots exist. As a result, the precomputed snapshots should be sampled evenly throughout the design space to ensure the quality of the obtained reduced-order basis. However, if the dimension of the parameter space is important, this *a priori* sampling can represent a heavy computational burden. An adaptive sampling approach, where the parameter space is sampled iteratively has been proposed to address this issue. This approach, introduced in the following, is typically referred to as greedy procedure.

2.2.2.2 Greedy procedure

Rather than attempting to cover the entire parameter space, the greedy procedure [Veroy et al., 2003, Bui-Thanh et al., 2008] builds an initial reduced-order basis using only one or a small number of snapshots. Then, by maximizing the projection error, it iteratively adds new snapshots to the training database. The projection error at any point $\boldsymbol{\mu} \in \mathscr{P}$ is given as:

$$e(\boldsymbol{\mu}) = \|\boldsymbol{w}(\boldsymbol{\mu}) - \tilde{\boldsymbol{w}}(\boldsymbol{\mu})\| \tag{2.24}$$

where $\tilde{\boldsymbol{w}}(\boldsymbol{\mu})$ is the state vector approximation obtained using the current reduced-order basis. Due to the term $\boldsymbol{w}(\boldsymbol{\mu})$, maximizing the projection error is, in practice, too expensive a problem. As a result, error estimators, which allow to bound the value of $e(\boldsymbol{\mu})$ within a given range, are typically used [Prud'homme et al., 2001, Grepl, M. and Patera, A., 2005]. Regardless, computing these estimators requires assembling the high-dimensional matrices for every point $\boldsymbol{\mu} \in \mathscr{P}$. To avoid the computational burden associated with this task, rather than searching the parameter space using an optimization algorithm, most applications of the greedy procedure obtain the error estimation at randomly sampled points. Then, they select the candidate point with the highest error estimation to add to the training database. By iteratively repeating this process, the greedy procedure is able to produce a reduced-order basis that is sufficiently accurate throughout the whole parameter space.

2.2.2.3 Proper Generalized Decomposition

Both the POD method and the greedy procedure rely on an *offline-online* strategy. In other words, they begin with an *offline* phase where the reduced-order basis is built and then switch to an *online* phase where the state vector can be computed for new parameter values. The main issue with this strategy is the computational cost required during the *offline* phase to ensure a reduced-order basis that is sufficiently accurate throughout the parametric space. To address this issue, another method called PGD [Ladevèze, 1985, Néron and Ladevèze, 2010, Ammar, 2010, Chevreuil and Nouy, 2012] has been developed. PGD directly tries to compute the reduced-order basis that minimizes an error indicator over the whole parametric space. It relies on the variable separation hypothesis, which allows to approximate the state vector $\boldsymbol{w}(\mu^1, \mu^2)$ as:

$$\tilde{\boldsymbol{w}}(\mu^1, ..., \mu^j) = \sum_{i=1}^{n} \prod_{j=1}^{m} \gamma_i^j(\mu^j) \boldsymbol{w}_i \tag{2.25}$$

where $\mu^j \in \mathscr{P}^j,\ j = 1, ..., m$ are parameters assumed to be independent of each other and $\gamma_i^j(\mu^j)$, $i = 1, ...n,\ j = 1, ..., m$ are unknown shape functions and $\boldsymbol{w}_i$ is an element of $\mathscr{W}$. The idea is then to start with an initial guess for $\boldsymbol{w}_1$, $\gamma_1^j,\ j = 1, ..., m$ (typically originating from initial or boundary conditions) and then to iteratively find $\boldsymbol{w}_i$, $\gamma_i^j,\ i = 1, ..., n,\ j = 1, ..., m$, that minimize the residual

$r(\tilde{\boldsymbol{w}})$ of the governing equations over the whole parametric domain:

$$\boldsymbol{w}_i^*, \{\gamma_i^*\}_{j=1}^m = \arg \min_{\boldsymbol{w}_i, \gamma_i^1, ..., \gamma_i^m} \int_{\mathscr{P}^1} ... \int_{\mathscr{P}^m} r(\tilde{\boldsymbol{w}})\, d\mu^1 ... d\mu^m \tag{2.26}$$

where $r(\tilde{\boldsymbol{w}}) = 0$ if the approximation is exact in $\mathscr{P}^1 \times ... \times \mathscr{P}^m$. In practice, Eq. (2.26) is solved alternatively for $\boldsymbol{w}_i$ and γ_i^j, $j = 1, ..., m$ using a fixed-point algorithm. Moreover, the number of modes n can be set *a priori* to a maximum number of modes, or the search for new modes can be stopped once the residual $r(\tilde{\boldsymbol{w}})$ is below a defined threshold. Once all modes have been obtained, the approximated state vector $\tilde{\boldsymbol{w}}(\mu^1, ..., \mu^j)$ can be obtained for any point using Eq. (2.25).

Although the PGD method avoids the *offline* computational effort needed to obtain the snapshot matrix, it remains a highly intrusive approach. Some recent efforts have attempted to render this approach non-intrusive [Courard et al., 2016, Zou et al., 2018, Tsiolakis et al., 2020], but they often still require significant knowledge of the underlying model.

2.2.2.4 Summary of reduced-order basis

In this section, three methods for obtaining the reduced order basis were discussed. The POD method uses precomputed solution snapshots to train an orthonormal basis which minimizes the projection error of the sampled snapshots in the least-squares sense. The greedy procedure also uses snapshots to train the reduced order basis, but iteratively places new snapshots in regions of the parametric space where the projection error is high, allowing to obtain a globally accurate reduced-order basis. The greedy procedure is, however, an intrusive method, which requires assembling the full-order matrices to compute the error estimation. Finally, the PGD method avoids the *offline* computation of the snapshot matrix, by directly attempting to minimize the projection error over the entire parametric space. This is possible thanks to a variable separation hypothesis. Nevertheless, the PGD is an intrusive approach which requires integration over the parametric domain. Because POD is the only method among all three that enables a fully non-intrusive approach, it is the preferred method in this work for obtaining a reduced-order basis.

2.2.3 The problem of slow decaying Kolmogorov n-width

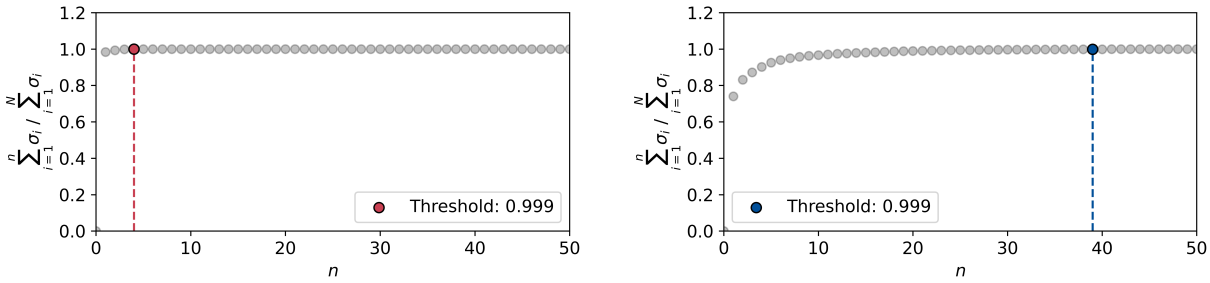
The main assumption that drove the development of the reduced-order models presented in the previous section was that any element $\boldsymbol{w}$ in $\mathscr{W}$ can be accurately approximated by an element $\boldsymbol{w}_n$ belonging to a linear subspace of dimension n, where n is sufficiently small. The measure that determines how well a reduced-order basis of dimension n can approximate the elements of $\mathscr{W}$ is the Kolmogorov n-width of $\mathscr{W}$ [Pinkus, 1985], defined as:

$$d_n(\mathscr{W}) = \inf_{\substack{\mathscr{Y}_n \subset \mathscr{Y} \\ \dim(\mathscr{Y}_n) \leq n}} \left[\sup_{\boldsymbol{w} \in \mathscr{W}} \left(\inf_{\boldsymbol{w}_n \in \mathscr{Y}_n} \|\boldsymbol{w} - \boldsymbol{w}_n\|_{\mathscr{Y}} \right) \right] \tag{2.27}$$

where $\mathscr{Y}$ is some normed linear space. In practice, the faster $d_n(\mathscr{W})$ decays as n increases, the more likely we are to obtain a low-dimensional approximation that accurately represents the full-order solution. Contrarily, if $d_n(\mathscr{W})$ decays slowly as n increases, the dimension of the reduced-order basis required to obtain an accurate approximation remains important.

In general, there is no analytic way to compute $d_n(\mathscr{W})$, but an heuristic way to assess its decay with n is to check the cumulative sum of the eigenvalues of the SVD of the snapshot matrix. Figure 2.2 illustrates the normalized cumulative sum for two different rates of decay of the Kolmogorov n-width. As is shown, when there is a rapid decay of $d_n(\mathscr{W})$ with n, the normalized

cumulative sum of the eigenvalues quickly approaches one. Contrarily, when there is a slow decay of $d_n(\mathcal{W})$ with n, a large number of modes is required for the normalized cumulative sum to be close to one.



(a) Rapid decay of $d_n(\mathcal{W})$ with n.

(b) Slow decay of $d_n(\mathcal{W})$ with n.

Figure 2.2: Illustration of different rates of decay of the Kolmogorov n-width. (a) Rapid decay of the Kolmogorov n-width. (b) Slow decay of the Kolmogorov n-width. For the two cases, the value of n at which the normalized cumulative sum of the eigenvalues reaches the threshold of 99.9% is indicated.

Slow decays of the Kolmogorov n-width often lead to reduced-order models which lack robustness with respect to certain parameter changes. For example, one can imagine the flow around an aircraft flying in transonic regime, which is highly sensitive to variations in the angle of attack and freestream Mach number. The resulting reduced-order model will then require a large number of basis vectors to be able to accurately represent the different solutions across the parametric space. To address the lack of robustness of the global reduced-order models, some authors propose the use of local POD bases [Amsallem and Farhat, 2008, Amsallem et al., 2009, Boncoraglio and Farhat, 2022, Geelen and Willcox, 2022]. Notably, earlier works propose to construct several POD bases, each obtained for a given set of parameter values. More recent works propose to partition the parameter space into subdomains and to obtain a database of piecewise-global POD bases, each trained within the respective subdomain. The common idea behind these approaches is that the obtained local POD bases are of lower dimension than their global counterparts. Some other authors propose to abandon POD altogether in favor of non-linear approximations [Lee and Carlberg, 2020, Touzé et al., 2021, Barnett and Farhat, 2022, Geelen et al., 2023]. The idea is that, for the same level of accuracy, the non-linear approximation requires a lower number of reduced variables compared to the linear approximation. This property is observed recently in [Barnett and Farhat, 2022, Geelen et al., 2023] where a quadratic approximation manifold is used for projection-based MOR. In this work, only the use of a database of pointwise local POD bases will be considered as an alternative to the use of a global POD basis. Note that, in this section, and throughout the rest of this work, the term "global basis" is used to indicate a reduced-order basis whose vectors remain constant in the parametric space.

2.2.3.1 Using a database of pointwise local POD bases

Up until now we addressed reduced-order models which approximate the state vector $\boldsymbol{w}(\boldsymbol{\mu})$ according to Eq. (2.13). When using pointwise local POD bases, the dependence of the reduced-order basis Φ on the parameter $\boldsymbol{\mu}$ must be considered and the approximation of the state vector becomes:

$$\tilde{\boldsymbol{w}}(\boldsymbol{\mu}) \approx \boldsymbol{w}_{\text{ref}} + \Phi(\boldsymbol{\mu})\boldsymbol{\alpha}(\boldsymbol{\mu}) \tag{2.28}$$

The remaining discussions, however, still stand. That is, from Section 2.2.1 we know how to obtain the generalized coordinates $\boldsymbol{\alpha}(\boldsymbol{\mu}^*)$, at any queried point $\boldsymbol{\mu}^*$, if $\Phi(\boldsymbol{\mu}^*)$ is available. Moreover, from Section 2.2.2.1, we know how to train a POD basis $\Phi(\boldsymbol{\mu}^{\text{DoE}})$ if a collection of full-order snapshots $\boldsymbol{w}(\boldsymbol{\mu}^{\text{DoE}}, t_1)$,...,$\boldsymbol{w}(\boldsymbol{\mu}^{\text{DoE}}, t_d)$ is available, where t is some parameter on which the local POD bases do not depend (t is often chosen as the time parameter). The only issue left to address is then how to obtain a new reduced order basis $\hat{\Phi}(\boldsymbol{\mu}^*)$ by interpolating a database of pointwise local POD bases $\Phi(\boldsymbol{\mu}_i) \in \mathbb{R}^{N\times n}$, each associated with a parameter point $\boldsymbol{\mu}_i$, with $i = 1,...,n_{\text{DoE}}$.

Illustration of the approach. The following section describes how a new reduced-order basis $\hat{\Phi}(\boldsymbol{\mu}^*)$ associated with the parameter point $\boldsymbol{\mu}^*$ can be constructed by interpolating a set of precomputed POD bases. To aid in the comprehension of the described methodology, some illustrations are provided. We remark, however, that said illustrations are merely used as a visual aid and do not arise from any physical model.

2.2.3.2 Interpolation in a tangent space to the Grassmann manifold

We briefly introduce the Grassmann manifold $\mathcal{G}(p,n)$ [Wong, 1967, Edelman et al., 1998, Boothby, 2003, Absil et al., 2004], of dimension $p(n-p)$, which is the set of all p-dimensional subspaces of $\mathbb{R}^n$. In practice, one point $\mathcal{X}$ on $\mathcal{G}(p,n)$ is a linear subspace, represented by the orthogonal basis $\mathbf{X} \in \mathbb{R}^{n\times p}$, whose columns span $\mathcal{X}$. Accordingly, the subspaces $\{\mathcal{Y}_i\}_{i=1}^{n_{\text{DoE}}}$ defined, respectively, by the columns of $\{\Phi(\boldsymbol{\mu}_i)\}_{i=1}^{n_{\text{DoE}}}$ are points on the Grassmann manifold $\mathcal{G}(n,N)$. Figure 2.3 illustrates the Grassmann manifold $\mathcal{G}(1,3)$, the points $\{\mathcal{Y}_i\}_{i=1}^{n_{\text{DoE}}}$ as colored dots lying on this 2-D manifold and the mapping to be interpolated (Eq. (2.29)) as a blue line.

$$\begin{aligned} \mathcal{P} &\mapsto \mathcal{G}(n,N) \\ \boldsymbol{\mu} &\mapsto \mathcal{Y}(\boldsymbol{\mu}) \end{aligned} \tag{2.29}$$

At each point $\mathcal{X} \in \mathcal{G}(p,n)$ there exists a tangent space $\mathcal{T}_{\mathcal{X}}$ of the same dimension as $\mathcal{G}(p,n)$. This tangent space is an Euclidean vector space with origin at point $\mathcal{X}$ and where classical interpolation can be performed. It is thus proposed in [Amsallem and Farhat, 2008, Amsallem et al., 2009] to first transport the set of precomputed reduced-order bases to the tangent space $\mathcal{T}_{\mathbf{P}}$ to $\mathcal{G}(n,N)$ at a chosen point of tangency $\mathbf{P}$, according to:

$$\mathbf{Z}(\boldsymbol{\mu}_i) = \log_{\mathbf{P}}(\Phi(\boldsymbol{\mu}_i)),\ i = 1,...,n_{\text{DoE}} \tag{2.30}$$

where $\mathbf{Z}(\boldsymbol{\mu}_i) \in \mathbb{R}^{N\times n}$ is the mapping of $\Phi(\boldsymbol{\mu}_i)$ onto the tangent space to $\mathcal{G}(n,N)$ at $\mathbf{P}$ and $\log_{\mathbf{P}}(\Phi(\boldsymbol{\mu}_i))$

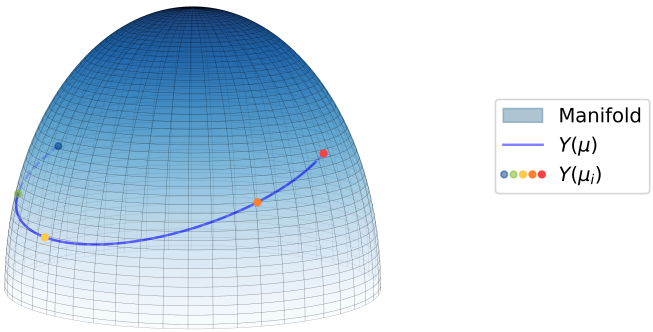


Figure 2.3: Illustration of the initial DoE of subspaces $\{\mathcal{Y}_i\}_{i=1}^{n_{\text{DoE}}}$ on the Grassmann manifold $\mathcal{G}(1,3)$. The mapping to be interpolated is represented by a blue line.

denotes the logarithmic map given by:

$$(\mathbf{I}-\mathbf{P}\mathbf{P}^{\intercal})\Phi_i(\boldsymbol{\mu}_i)(\mathbf{P}^{\intercal}\Phi(\boldsymbol{\mu}_i))^{-1}=\mathbf{U}_i\Sigma_i\mathbf{V}_i^{\intercal}\ \ (SVD) \tag{2.31}$$

$$\mathbf{Z}(\boldsymbol{\mu}_i)=\mathbf{U}_i\tan^{-1}(\Sigma_i)\mathbf{V}_i^{\intercal} \tag{2.32}$$

The subspaces $\{\mathcal{Z}_i\}_{i=1}^{n_{\text{DoE}}}$ spanned by the columns of $\mathbf{Z}(\boldsymbol{\mu}_i)$, respectively, can themselves be seen as points in the tangent space $\mathcal{T}_{\mathbf{P}}$. Figure 2.4 illustrates the points $\{\mathcal{Z}_i\}_{i=1}^{n_{\text{DoE}}}$, obtained from the mapping of $\{\mathcal{Y}_i\}_{i=1}^{n_{\text{DoE}}}$ onto the tangent space to the simplified 2-D manifold at point $\mathbf{P}$. In the figure, the point identified with a white star is chosen as point of tangency $\mathbf{P}$.

Remark 2.1 *In Figure 2.4, $\mathbf{P}$ is chosen as the Fréchet mean [Fréchet, 1948] of the set of points $\{\mathcal{Y}_i\}_{i=1}^{n_{DoE}}$, obtained by solving the following minimization problem:*

$$\mathbf{P}=\arg\min_{P\in\mathcal{G}(n,N)}\sum_{i=1}^{n_{DoE}}d^2(\mathbf{P},\mathcal{Y}_i) \tag{2.33}$$

where d denotes the geodesic distance in $\mathcal{G}(n,N)$. However, in practice, $\mathbf{P}$ can be taken as any other point in $\mathcal{G}(n,N)$. Namely, one of the points $\{\mathcal{Y}_i\}_{i=1}^{n_{DoE}}$ could be used as point of tangency. It is shown in [Amsallem and Farhat, 2008] that, as long as $\mathbf{P}$ lies in the neighborhood of $\{\mathcal{Y}_i\}_{i=1}^{n_{DoE}}$, the proposed matrix interpolation strategy remains robust with respect to the choice of $\mathbf{P}$.

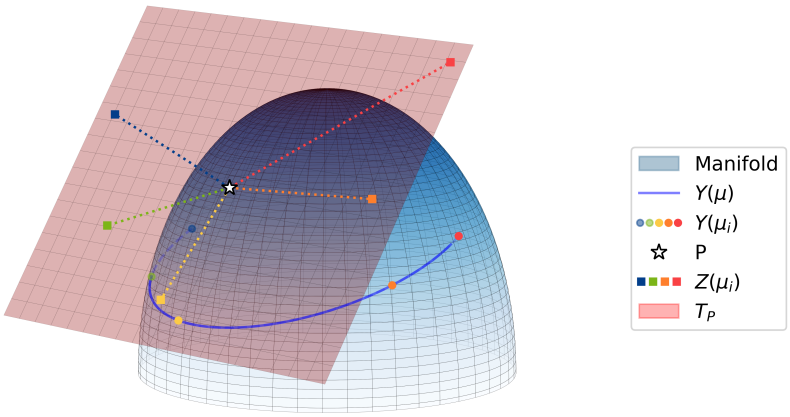


Figure 2.4: Illustration of the mapping $\{\mathcal{Z}_i\}_{i=1}^{n_{\text{DoE}}}$ of each point $\{\mathcal{Y}_i\}_{i=1}^{n_{\text{DoE}}}$ onto the tangent space to the simplified 2-D manifold at point $\mathbf{P}$.

Since $\{\mathcal{Z}_i\}_{i=1}^{n_{\text{DoE}}}$ lie in a linear space, the matrices $\mathbf{Z}(\boldsymbol{\mu}_i)$, $i=1,\ldots,n_{\text{DoE}}$ can be interpolated to obtain $\hat{\mathbf{Z}}(\boldsymbol{\mu}^*)$ at any new queried point $\boldsymbol{\mu}^*$. In [Amsallem and Farhat, 2008, Amsallem et al., 2009], interpolation using Lagrange polynomials is used, as follows:

$$\hat{\mathbf{Z}}(\boldsymbol{\mu}^*)=\sum_{i=1}^{n_{\text{DoE}}}\prod_{i\neq j}\frac{\boldsymbol{\mu}^*-\boldsymbol{\mu}_j}{\boldsymbol{\mu}_i-\boldsymbol{\mu}_j}\mathbf{Z}(\boldsymbol{\mu}_i) \tag{2.34}$$

Once the interpolated matrix $\hat{\mathbf{Z}}(\boldsymbol{\mu}^*)$ has been computed, it suffices to transport it from the tangent space back to $\mathcal{G}(n,N)$, to obtain $\hat{\Phi}(\boldsymbol{\mu}^*)$. This is done by performing the inverse operation of Eq. (2.30):

$$\hat{\Phi}(\boldsymbol{\mu}^*)=\text{Exp}_{\mathbf{P}}(\hat{\mathbf{Z}}(\boldsymbol{\mu}^*)) \tag{2.35}$$

where $\text{Exp}_{\mathbf{P}}(\hat{\mathbf{Z}}(\boldsymbol{\mu}^*))$ denotes the exponential mapping [Wong, 1967] given by:

$$\hat{\mathbf{Z}}(\boldsymbol{\mu}^*) = \mathbf{U}_* \Sigma_* \mathbf{V}_* \text{ (SVD)} \tag{2.36}$$

$$\hat{\Phi}(\boldsymbol{\mu}^*) = \mathbf{P}\mathbf{V}_* \cos(\Sigma_*) + \mathbf{U}_* \sin(\Sigma_*) \tag{2.37}$$

Figure 2.5 illustrates the points $\mathscr{Z}_*$ and $\mathscr{Y}_*$, obtained after the interpolation of the matrices $\mathbf{Z}(\boldsymbol{\mu}_i)$, $i = 1, ..., n_{\text{DoE}}$. As is shown, when the chosen interpolation strategy is sufficiently accurate, the point $\mathscr{Y}_*$ falls back on the blue line. Algorithm 9 summarizes the main steps required for the interpolation of a database of local POD bases.

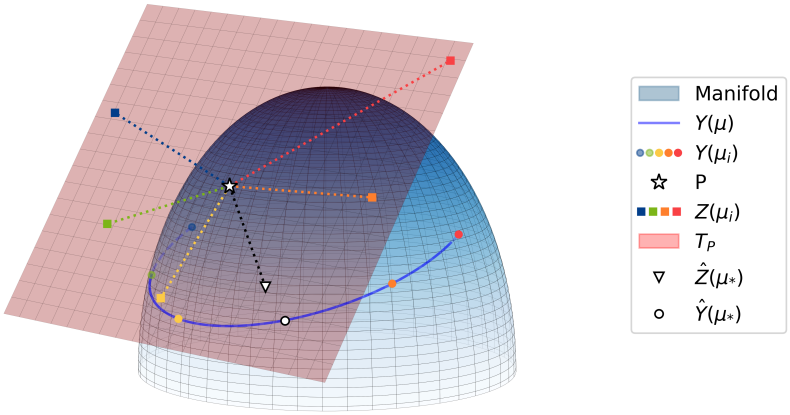


Figure 2.5: Illustration of the points $\mathscr{Z}_*$ and $\mathscr{Y}_*$, obtained after interpolation in the tangent space. We note that, when the interpolation is sufficiently accurate, the point $\mathscr{Y}_*$ lies on the blue line.

Algorithm 9 Interpolation of a database of local POD bases

Input: $\Phi(\boldsymbol{\mu}_i)$, $i = 1, ..., n_{\text{DoE}}$, $\boldsymbol{\mu}^*$ ▷ Precomputed local POD bases
Choose $\mathbf{P}$ in $\mathscr{G}(n, N)$ ▷ Point of tangency
Compute $\mathbf{Z}(\boldsymbol{\mu}_i)$, $i = 1, ..., n_{\text{DoE}}$ ▷ Logarithmic mapping (Eq. (2.30))
Compute $\hat{\mathbf{Z}}(\boldsymbol{\mu}^*)$ from $\mathbf{Z}(\boldsymbol{\mu}_i)$, $i = 1, ..., n_{\text{DoE}}$ ▷ Interpolation in the tangent space
Compute $\hat{\Phi}(\boldsymbol{\mu}^*)$ ▷ Exponential mapping (Eq. (2.35))
Output: $\hat{\Phi}(\boldsymbol{\mu}^*)$

Interpolation strategy. In practice, interpolation in the tangent space can be carried out using other strategies, including RBF [Choi et al., 2020] or GP regression [Mallasto and Feragen, 2018]. Another strategy, which performs interpolation via Inverse Distance Weighting (IDW) has also been developed in [Mosquera et al., 2019]. The main advantage of this strategy is that it does not require mapping onto the tangent space, and therefore provides results which are independent on the choice of $\mathbf{P}$. Finally, recent developments propose to interpolate the local POD bases directly, rather than the corresponding linear subspaces [Goutaudier et al., 2023]. The main idea behind this approach is to conserve the order of the singular values in the interpolated matrix to achieve greater accuracy. In this work, only interpolation of the linear subspaces using GP regression will be considered, albeit using a different strategy from the one introduced in [Mallasto and Feragen, 2018].

Stability of POD basis interpolation. Although not discussed in this section, necessary stability conditions for POD basis interpolation on Grassmann manifolds have been established

in [Friderikos et al., 2022]. Indeed, for the logarithmic mapping (Eq. (2.30)) to be well defined, the matrix $\mathbf{P}^\intercal\Phi(\boldsymbol{\mu}_i)$ must be non-singular for all $i = 1,...,n_{\text{DoE}}$. Moreover, the exponential mapping (Eq. (2.35)) is only injective if $\sigma_1^* < \pi/2$, with σ_1^* the maximum singular value $\hat{\mathbf{Z}}(\boldsymbol{\mu}^*)$. Finally, to ensure that the accuracy of the interpolation increases with the number of modes in the local POD bases, then the subspace $\mathscr{Y}^{(n)}$, spanned by the columns of $\Phi(\boldsymbol{\mu}^*) = [\phi_1(\boldsymbol{\mu}^*),...,\phi_n(\boldsymbol{\mu}^*)]$, must verify $\mathscr{Y}^{(n)} \subset \mathscr{Y}^{(n')}$, with $n' > n$. In [Friderikos et al., 2022], this condition is checked by considering the geometric distance $\delta\left(\mathscr{Y}^{(n)}, \mathscr{Y}^{(n')}\right)$ between $\mathscr{Y}^{(n)}$ and $\mathscr{Y}^{(n')}$, defined as:

$$\delta(\mathscr{Y}^{(n)}, \mathscr{Y}^{(n')}) = \left(\sum_{i=1}^{n} \arccos^2(\sigma_i)\right)^{1/2} \tag{2.38}$$

where $\sigma_i,\ i = 1,...,n$ are the singular values of $\Phi(\boldsymbol{\mu}^*)^\intercal\Phi'(\boldsymbol{\mu}^*)$, also known as the principal angles between $\mathscr{Y}^{(n)}$ and $\mathscr{Y}^{(n')}$ [Ye and Lim, 2016].

2.2.4 Summary of projection-based MOR

Projection-based MOR provides a means to reduce the computational cost of solving PDEs, while retaining a relatively high level of accuracy. Through pMOR, a low-dimensional representation of the full-order model is obtained by projecting the discretized governing equations onto a reduced-order basis. This reduced-order basis is usually trained using solution snapshots sampled over the entire parametric space. For some disciplines, however, certain parameter changes can contribute with significantly new information to the global approximation, leading to a large number of basis vectors. To address this issue, the interpolation of a database of local reduced-order bases has been developed. The main idea behind this approach is that, for the same order of accuracy, the local reduced-order approximations require fewer basis vectors than the global counterpart. In the following, the use of pMOR for multi-query applications involving multidisciplinary systems is discussed.

2.3 Model order reduction for MDA

Designing multidisciplinary systems often requires coupling several disciplines, each involving the resolution of PDEs. To reduce the computational cost of solving the MDA, we may thus want to build a reduced-order model for each discipline and solve an approximated version of the MDA. The reduced-space MDA is written as:

$$\boldsymbol{y}_i^r = f_i^r(\boldsymbol{x}, \boldsymbol{y}_{c_i}^r),\ i = 1,...,n_d \quad \forall \boldsymbol{x} \in \mathscr{X} \tag{2.39}$$

where $\boldsymbol{y}_i^r$ is the approximated coupling variable, given by:

$$\boldsymbol{y}_i^r = \phi_i^0 + \sum_{j=0}^{n_i} \alpha_i^j(\boldsymbol{x}, \boldsymbol{y}_{c_i}^r)\phi_i^j \tag{2.40}$$

where $\Phi_i = \{\phi_1,...,\phi_{n_i}\}$ denotes the POD basis associated with discipline i and $\boldsymbol{\alpha} = \{\alpha_i^1,...,\alpha_i^{n_i}\}$ are the corresponding generalized coordinates. Building the reduced-order bases in the case of a multidisciplinary system, however, entails some additional difficulties. Indeed, to build the disciplinary POD basis Φ_i, it is necessary to obtain snapshots of $\boldsymbol{y}_i$ by sampling in $\mathscr{X} \times \mathscr{C}^{c_i}$, where $\mathscr{C}^{c_i} = \prod_{j\neq i}^{n_d} \mathscr{C}^j$. And while sampling in $\mathscr{X}$ is trivial, sampling in $\mathscr{C}^{c_i}$ can be more difficult, as $\mathscr{C}^{c_i}$ is a high-dimensional space, whose bounds are unknown. To address this issue it is proposed in [Coelho et al., 2009] to obtain snapshots of the coupling variables by solving the MDA for

a collection of sampled design space points $\boldsymbol{x}_1,...,\boldsymbol{x}_d$. Then, using the obtained snapshots, the disciplinary POD bases can be built and interpolation of the generalized coordinates can be used to compute the solution of the MDA at new queried design space points.

Although the above described strategy alleviates the computational burden of a multi-query problem requiring many MDA resolutions, it is not computationally efficient, in the sense that it requires globally accurate disciplinary POD bases as well as globally accurate interpolations of the generalized coordinates. Indeed, an adaptive sampling strategy, which allows the enrichment of the disciplinary surrogates during the resolution of the MDA should be envisioned to minimize the training costs. Such a strategy has been recently developed in [Berthelin, 2022, Berthelin et al., 2022] and is introduced in the following.

2.3.1 Disciplinary Proper Orthogonal Decomposition and Interpolation

The Disciplinary Proper Orthogonal Decomposition and Interpolation (DPOD+I) strategy solves the MDA by combining disciplinary POD bases with adaptively enriched GP approximations of the generalized coordinates. An evaluation of the projection error when calls to the disciplinary solvers are made, further allows the DPOD+I strategy to enrich the disciplinary POD bases during the resolution of the MDA. Because it uses the uncertainty of the generalized coordinates interpolations to solve the MDA, the DPOD+I strategy can be seen as the extension of the EGMDA strategy to high-dimensional coupling variables.

2.3.1.1 Disciplinary GPs for high-dimensional coupling variables

The main idea behind the DPOD+I strategy is to obtain an approximated version of the MDA, written in terms of the GP approximations of the generalized coordinates. Let Ψ_i denote the projection application allowing to obtain the generalized coordinates associated with discipline i:

$$\begin{aligned} \Psi^i : \mathscr{C}^i \subset \mathbb{R}^{d^i} &\rightarrow \tilde{\mathscr{C}}^i \subset \mathbb{R}^{n^i} \\ \boldsymbol{y}_i &\mapsto \{\alpha_i^1,...,\alpha_i^{n^i}\} = \{\phi_i^1(\boldsymbol{y}_i - \phi_i^0),...,\phi_i^{n^i}(\boldsymbol{y}_i - \phi_i^0)\} \end{aligned} \tag{2.41}$$

and $\{\Psi_i\}^{-1}$ denote its inverse operation, allowing to reconstruct the high-dimensional vector $\boldsymbol{y}_i$:

$$\begin{aligned} \{\Psi_i\}^{-1} : \tilde{\mathscr{C}}^i &\rightarrow \mathscr{C}^i \\ \alpha_i^1,...,\alpha_i^{n^i} &\mapsto \{\Psi_i\}^{-1}(\alpha_i^1,...,\alpha_i^{n^i}) = \phi_i^0 + \textstyle\sum_{j=1}^{n^i} \alpha_i^j \phi_i^j \end{aligned} \tag{2.42}$$

Then, it is possible to rewrite the MDA in the reduced space, as follows:

$$\alpha_i^1(\boldsymbol{x},\boldsymbol{\alpha}_{c_i}),...,\alpha_i^{n_i}(\boldsymbol{x},\boldsymbol{\alpha}_{c_i}) = \Psi^i\left(\boldsymbol{y}_i\left(\boldsymbol{x},\{\Psi^i\}^{-1}(\boldsymbol{\alpha}_{c_i})\right)\right),\ i = 1,...,n_d \tag{2.43}$$

To obtain the disciplinary surrogates, it is then proposed in the DPOD+I strategy to replace each coefficient α_i^j by a GP approximation $\hat{\alpha}_i^j$, according to:

$$\hat{\alpha}_i^j = \mu_i^j(\boldsymbol{x},\boldsymbol{\alpha}_{c_i}) + \epsilon_i^j(\boldsymbol{x},\boldsymbol{\alpha}_{c_i})\ \ i = 1,...,n^d,\ j = 1,...,n^i \tag{2.44}$$

where $\mu_i^j(\boldsymbol{x},\boldsymbol{\alpha}_{c_i})$ are the means of the disciplinary GPs and $\epsilon_i^j(\boldsymbol{x},\boldsymbol{\alpha}_{c_i})$ are GPs of zero mean conditioned on the respective disciplinary DoEs. The GP approximations can then be plugged back onto Eq. (2.43), to obtain an approximated reduced MDA. This approximated MDA can, in turn, be solved by adaptively enriching the disciplinary GPs as proposed in the EGMDA strategy, presented in Section 1.2.3.2. The only challenge left to address is thus the construction of the initial disciplinary DoEs, allowing to obtain the disciplinary reduced-order basis, as well as the initial surrogate approximations. The sampling strategy proposed in [Berthelin, 2022, Berthelin et al., 2022] is presented in the following.

2.3.1.2 Training the disciplinary POD+I surrogates

In the DPOD+I approach, a sampling strategy which exploits the structure of the MDA, but does not solve it, is used. The idea is that, rather than keeping the design variables fixed through the iterates, they are randomly changed at each MDA iteration. This allows to obtain disciplinary DoEs which are evenly spread throughout the design and coupling variable spaces. To simplify, we describe the proposed training strategy for a two-discipline system, but the proposed approach can be easily extended to include more disciplines.

We begin by generating a 1-D manifold made of r initial guesses of the first coupling variable:

$$\mathrm{Y}_1^0 = \{\lambda_i \boldsymbol{y}_1^0, \lambda_i \in [\lambda^-, \lambda^+] \subset \mathbb{R}, i = 1, ..., r\} \tag{2.45}$$

where Y_1^0 is the generated manifold, $\boldsymbol{y}_1^0$ is the initial guess of the MDA solver for the first coupling variable, and $[\lambda^-, \lambda^+]$ are the bounds that define the range of variation of the r initial guesses.

Remark 2.2 *In practice,* $[\lambda^-, \lambda^+]$ *may be obtained by expert judgment so that every element of* Y_1^0 *is likely to belong to* $\mathscr{C}^1$*. Moreover, even if the initial guess for* $[\lambda^-, \lambda^+]$ *is poor, its influence on the sampled snapshots will decrease as the training algorithm progresses.*

Once the initial guesses $(\boldsymbol{y}_1)_i^0 = \lambda_i \boldsymbol{y}_1^0,\ i = 1, ..., r$ are obtained, they can be used to generate a set of r initial guesses for the second coupling variable, as follows:

$$(\boldsymbol{y}_2)_i^0 = f_2(\boldsymbol{x}_i, (\boldsymbol{y}_1)_i^0),\ \ i = 1, ..., r \tag{2.46}$$

where the design variable points $\boldsymbol{x}_i \in \mathscr{X}$ are sampled using a space filling strategy. Note that, although the samples $(\boldsymbol{y}_2)_i^0,\ i = 1, ..., r$ are obtained through calls to the real disciplinary solver, they do not constitute valid MDA snapshots because the inputs $(\boldsymbol{y}_1)_i^0,\ i = 1, ..., r$ were artificially generated. As a result these samples are used to initialize the training procedure, but are excluded from the corresponding training sets.

The objective is then to successively generate batches of disciplinary solver solutions, using as inputs the r solutions of the previous batch. For each newly generated batch, the mean relative projection error $\bar{e}_j^b$ is computed, as follows:

$$(e_j)_i^b = \frac{\|(\boldsymbol{y}_j)_i^b - \{\Psi_j\}^{-1}(\Psi_j(\boldsymbol{y}_j)_i^b))\|}{\|(\boldsymbol{y}_j)_i^b\|_2},\ i = 1, ..., r,\ j = 1, 2 \tag{2.47}$$

$$\bar{e}_j^b = \frac{1}{r}\sum_{i=1}^{r}(e_j)_i^b,\ j = 1, 2 \tag{2.48}$$

where b is the batch number. If the projection error is too large, the batch is added to the training set of the respective disciplinary POD basis and a new batch is used to calculate the projection error. This iterative procedure continues until the mean relative projection error is less than a given tolerance ϵ_{POD} for both disciplines. Algorithm 10 summarizes the main steps of the DPOD+I training strategy for a two-discipline system. Note that the proposed training strategy not only provides the disciplinary POD bases, but also two sets of DoEs which can be used to train the disciplinary GPs. Moreover, the proposed approach can be compared to the greedy procedure introduced in Section 2.2.2.2, in the sense that it monitors the projection error to determine when the computed disciplinary POD bases are sufficiently accurate.

Alternative training strategies. The DPOD+I strategy is not the only one proposing a training of the reduced-order models which does not rely on MDA resolutions. Indeed, in [Discacciati and Hesthaven, 2023, Discacciati and Hesthaven, 2024] an artificial parameterization of the interface of the non-overlapping disciplinary domains is proposed. Using this artificial parameterization, it is possible to generate physically-relevant boundary conditions to obtain snapshots and train the disciplinary reduced-order models.

Algorithm 10 DPOD+I training strategy for a two-discipline system

Input: $(\boldsymbol{y}_2)_i^0,\ i=1,...,r,\ \epsilon_{\text{POD}}$ ▷ Initial samples and error tolerance
  $\Phi_1 = [\boldsymbol{y}_1^0]$ ▷ Initialize POD basis 1
  $\Phi_2 = [\boldsymbol{y}_2^0]$ ▷ Initialize POD basis 2
  $\bar{e}_1^b,\ \bar{e}_2^b = 1$
  $b = 1$ ▷ Initialize batch count
  while $\bar{e}_1^b,\ \bar{e}_2^b > \epsilon_{\text{POD}}$ **do**
    Sample $(\boldsymbol{x})_i^{b,1},\ i=1,...,r$ in $\mathscr{X}$
    for $i=1,...,r$ **do**
      Compute $(\boldsymbol{y}_1)_i^b = f_1(\boldsymbol{x}_i, (\boldsymbol{y}_2)_i^{b-1})$ ▷ Obtain new snapshots
    end for
    Sample $(\boldsymbol{x})_i^{b,2},\ i=1,...,r$ in $\mathscr{X}$
    for $i=1,...,r$ **do**
      Compute $(\boldsymbol{y}_2)_i^b = f_2(\boldsymbol{x}_i, (\boldsymbol{y}_1)_i^b)$ ▷ Obtain new snapshots
    end for
    Compute $\bar{e}_1^b,\ \bar{e}_2^b$ ▷ Projection error (Eq. (2.48))
    Compute Φ_1 using $(\boldsymbol{y}_1)_i^j,\ i=1,...,r,\ j=1,...,b$ ▷ Update POD basis 1
    Compute Φ_2 using $(\boldsymbol{y}_2)_i^j,\ i=1,...,r,\ j=1,...,b$ ▷ Update POD basis 2
    $b = b+1$ ▷ Increase batch count
  end while
Output: $\Phi_1,\ \Phi_2,\ (\boldsymbol{x}_i)^{j,1},\ (\boldsymbol{y}_1)_i^j,\ (\boldsymbol{x}_i)^{j,2},\ (\boldsymbol{y}_2)_i^j,\ i=1,...,r,\ j=1,...b$

2.3.1.3 *Online* enrichment of the disciplinary POD bases

Once the disciplinary POD bases have been obtained and the disciplinary surrogates for each of the generalized coordinates have been built, it is possible to solve the MDA in any point of the design space using the EGMDA strategy. As discussed in Section 1.2.3.2, this strategy adaptively enriches the disciplinary GPs based on the dispersion of the random MDA solutions. This not only allows to solve the MDA at a reduced computational cost, but also allows monitoring the projection error each time a disciplinary surrogate is selected for enrichment. Indeed, because enriching the disciplinary GPs requires one call to the corresponding disciplinary solver, the obtained snapshot can be used to compute the projection error on the current disciplinary POD basis. If the projection error is above the previously defined tolerance ϵ_{POD}, the reduced-order basis can be updated as follows:

$$\Phi_i = \left[\phi_i^1, ..., \phi_i^{n^i}, \frac{\boldsymbol{y}_i(\boldsymbol{x}, \hat{\boldsymbol{y}}_{c_i}) - \{\Psi_i\}^{-1}(\Psi_i(\boldsymbol{y}_i(\boldsymbol{x}, \hat{\boldsymbol{y}}_{c_i}))}{\|\boldsymbol{y}_i(\boldsymbol{x}, \hat{\boldsymbol{y}}_{c_i}) - \{\Psi_i\}^{-1}(\Psi_i(\boldsymbol{y}_i(\boldsymbol{x}, \hat{\boldsymbol{y}}_{c_i}))\|_2}\right],\ \forall i = 1, ..., n_d \tag{2.49}$$

This allows to maintain a desired level of accuracy, even in regions of the design space that were not well represented by the initial POD bases. Furthermore, for applications which explore only certain regions of the design space (such as MDO problems), the error tolerance ϵ_{POD} can be relaxed during the training stage to reduce the computational cost and made stricter during the *online* stage to improve accuracy. This enrichment strategy is similar to the one employed in [Gogu and Passieux, 2013, Gogu, 2015] where the new snapshot is orthogonalized, normalized, and added to the basis.

2.3.2 Shortcomings of the DPOD+I strategy

The DPOD+I strategy allows to obtain disciplinary surrogates by combining disciplinary POD bases with GP approximations of the generalized coordinates. The built disciplinary surrogates

can then be used with the EGMDA strategy to reduce the computational cost of solving the MDA. This approach was shown in [Berthelin, 2022] to work well when the number of POD basis vectors remained small (typically less than 10). Regardless, some of the considered test cases presented POD basis with a significant number of basis vectors for at least one of the disciplinary solvers. This resulted in large input spaces for the remaining disciplines, which were unfit for the construction of the corresponding GP approximations of the generalized coordinates. Additionally, there was a large number of random variables in the approximated MDA, which made it difficult to attain the convergence criterion of the EGMDA approach.

In light of the obtained results, it became clear that the use of global POD bases as a dimension reduction strategy can sometimes be detrimental to the performance of the DPOD+I approach. To address this issue, a dimension reduction strategy which relies on the interpolation of a database of pointwise local POD bases was introduced in Section 2.2.3. The main idea behind this approach is that the number of basis vectors in each local POD basis is smaller than the one obtained for the equivalent global POD basis. By building the disciplinary surrogates in these lower dimensional spaces, the issues encountered in [Berthelin, 2022] may be resolved. Adaptation of the DPOD+I training and enrichment strategies to allow for the construction of the pointwise local POD bases should thus be envisioned.

Chapter summary:

In this chapter we were interested in reduced-order models to obtain low dimensional approximations of disciplinary solvers whose governing equations are partial differential equations (PDEs). We thus first introduced the following disciplinary models:

- *The compressible Euler equations and the Vortex Lattice method, used to model the flow around the wing.*
- *The linear elastic model, used to model the structural displacement of the wing.*

Projection-based model order reduction (MOR) was introduced next as a popular method to obtain the low-dimensional approximations of the disciplinary models. To obtain the generalized coordinates, the following methods were introduced:

- *The Petrov and Petrov-Galerkin projections, which obtain the state vector at new queried points by solving the governing equations in the low dimensional space, and therefore constitute an intrusive approach.*
- *Interpolation of the generalized coordinates, which uses surrogate models to obtain the state vector at new queried points, and therefore constitutes a non-intrusive approach.*

To obtain the reduced-order basis, the following approaches were introduced:

- *The Proper Orthogonal Decomposition (POD), which uses a collection of pre-computed snapshots to train the reduced order basis, in an non-intrusive manner.*
- *The greedy procedure, which iteratively samples new snapshots in regions on the parametric space where the error estimator is large, to obtain a globally accurate reduced-order model.*

- *The Proper Generalized Decomposition, which relies on the variable separation hypothesis to directly attempt to minimize the projection error over the entire parametric space.*

The issue of slow decaying Kolmogorov n-width was discussed next and interpolation of a database of pointwise local POD bases was introduced as an alternative to the use of global POD bases.

Finally, model order reduction for MDA was discussed. The DPOD+I strategy which combines disciplinary POD bases with GP approximations of the generalized coordinates was introduced as a model order reduction strategy which efficiently reduces the computational cost of multi-query problems requiring the resolution of several MDAs.

II

Contributions to an MDAO framework using disciplinary surrogates

CHAPTER 3

Local POD bases for disciplinary surrogates in MDA

Contents

The developments presented in this chapter led to the following conference paper, which was presented at the AIAA SCITECH Forum, in January 2024:

> Cardoso, I., Dubreuil, S., Bartoli, N., Gogu, C., and Salaün, M. (2024). Model order reduction for parameterized multidisciplinary analysis using disciplinary surrogates: application to non-linear solvers. In *AIAA SCITECH 2024 Forum*. DOI: 10.2514/6.2024-1407

In this chapter we will:

- *Introduce the interpolation of local POD bases using GPs.*
- *Introduce the use of pointwise local POD bases for single-discipline analysis.*
- *Introduce the use of pointwise local POD bases in MDA.*
- *Present numerical results for the static aeroelastic analysis of an aircraft wing.*

3.1 Chapter contributions

In Chapter 2 we introduced the DPOD+I strategy for the reduction of the computational cost required to solve MDA problems in a multi-query context. One of the main shortcomings of this strategy, however, is the decreased accuracy that occurs when there is a large number of basis vectors in the disciplinary POD bases. In this chapter we attempt to address this challenge by replacing the use of a global POD basis with the interpolation of a database of pointwise local POD bases. The use of pointwise local POD bases has most often been used in the literature in the context of single-discipline analysis where one of the parameters was time. In this chapter, we propose to use this strategy in an MDA context. We thus adapt the training and enrichment strategies of the DPOD+I approach to accommodate the use of local POD bases. Numerical test cases are used to confirm the interest of the proposed approach.

3.2 Interpolation of local POD bases using GPs

The use of pointwise local POD bases allows to obtain reduced-order models which, for the same level of accuracy, present fewer modes than the corresponding global POD basis. To obtain the reduced-order basis at a new queried point, interpolation in the tangent space to the Grassmann manifold can be used, as described in Section 2.2.3.2. The original interpolation strategy, however, relied on interpolation using Lagrange polynomials, which does not provide any indication of the committed interpolation error. In this work, we instead propose to use an interpolation strategy which uses GPs. We note that, although we rely on the definition of a basis of the tangent plane to the Grassmann manifold to perform interpolation, the proposed approach is an adaptation of the strategy introduced in [Mallasto and Feragen, 2018].

3.2.1 Obtaining a basis of the tangent plane to the Grassmann manifold

We recall that, in order to interpolate a database of pointwise local POD bases, the points $\{\mathscr{Y}_i\}_{i=1}^{n_{\text{DoE}}}$ are first mapped onto the tangent space $\mathscr{T}_{\mathbf{P}}$ to the Grassmann manifold $\mathscr{G}(n,N)$ at a chosen point of tangency $\mathbf{P}$, thanks to the logarithmic mapping of Eq. (2.30). The result of this mapping are the subspaces $\{\mathscr{Z}_i\}_{i=1}^{n_{\text{DoE}}}$, spanned by the columns of $\mathbf{Z}(\boldsymbol{\mu}_i)$, respectively. We also recall that the dimension of the Grassmann manifold is $n(N-n)$, which means that $\mathbf{Z}(\boldsymbol{\mu}_i)$ are high-dimensional matrices ($n \ll N$). However, because the subspaces $\{\mathscr{Z}_i\}_{i=1}^{n_{\text{DoE}}}$ are points in the tangent space, they can be used to define a basis of the tangent plane $\mathbf{T}_P$, subspace of $\mathscr{T}_P$. In this work, the vectors $(\vec{e}_j),\ j = 1,...,n_{\mathbf{T}_P}$, defining a basis of $\mathbf{T}_P$, are given by the left singular vectors of the following SVD:

$$\mathbf{Z} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V} \tag{3.1}$$

where $\mathbf{Z}$ is the matrix whose columns are $\mathbf{Z}(\boldsymbol{\mu}_i),\ i = 1,...,n_{\text{DoE}}$ flattened to a vector form. Figure 3.1 illustrates the vectors $(\vec{e}_1,\vec{e}_2)$, used to define a basis of the tangent plane to $\mathscr{G}(n,N)$ at $\mathbf{P}$.

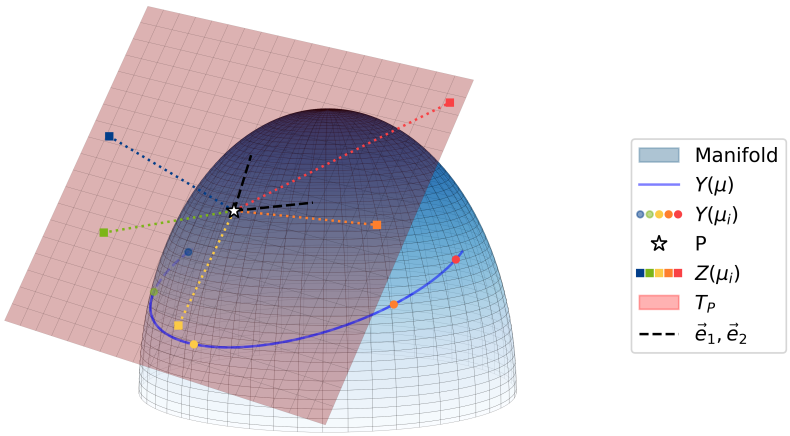


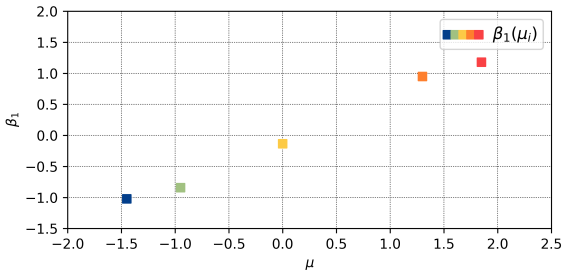
Figure 3.1: Illustration of the vectors $(\vec{e}_1, \vec{e}_2)$, used to define a basis of the tangent plane $\mathbf{T}_P$ to $\mathcal{G}(n,N)$ at $\mathbf{P}$.

3.2.2 Interpolation of the coordinates of the tangent space basis

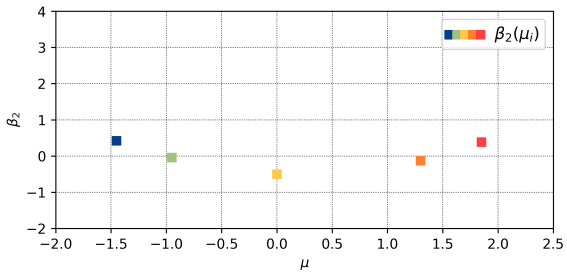
Having established the basis $(\vec{e}_j),\ j = 1,...,n_{\mathbf{T}_P}$, each point $\{\mathcal{Z}_i\}_{i=1}^{n_{\text{DoE}}}$ can then be described by a set of coordinates $\beta_j,\ j = 1,...,n_{\mathbf{T}_P}$, each associated with a vector in the tangent plane basis:

$$\{\mathcal{Z}_i\}_{i=1}^{n_{\text{DoE}}} = \sum_{j=1}^{n_{\mathbf{T}_P}} \beta_j(\boldsymbol{\mu}_i)\vec{e}_j,\ \ i = 1,...,n_{\text{DoE}} \tag{3.2}$$

Moreover, since the points $\{\mathcal{Z}_i\}_{i=1}^{n_{\text{DoE}}}$ are each associated with a parameter value $\boldsymbol{\mu}_i$, their coordinates on the tangent plane basis are themselves each associated with a parameter value $\boldsymbol{\mu}_i$. Figure 3.2 illustrates the coordinates of each point $\{\mathcal{Z}_i\}_{i=1}^{n_{\text{DoE}}}$ in the corresponding basis $(\vec{e}_1, \vec{e}_2)$.



(a) Coordinate associated with $\vec{e}_1$.



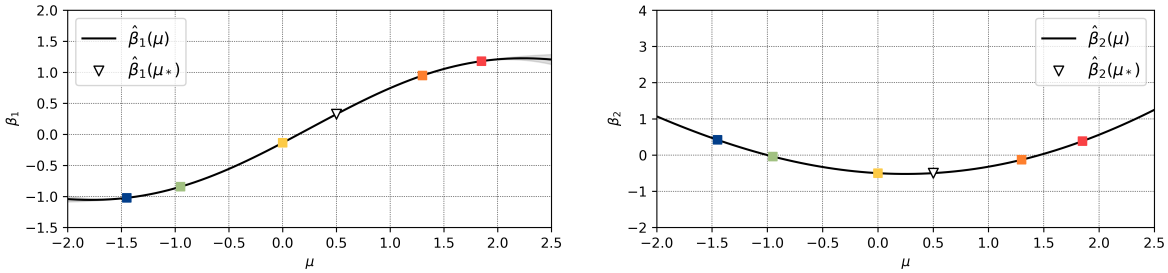
(b) Coordinate associated with $\vec{e}_2$.

Figure 3.2: Coordinates of each point $\{\mathcal{Z}_i\}_{i=1}^{n_{\text{DoE}}}$ on the basis $(\vec{e}_1, \vec{e}_2)$ as a function of the associated parameter value $\boldsymbol{\mu}_i$.

The obtained coordinates can be interpolated independently to obtain $\hat{\mathbf{Z}}(\boldsymbol{\mu}_*)$ at a new queried point $\boldsymbol{\mu}_*$. The result can then be mapped back onto the Grassmann manifold to construct a new local POD basis $\hat{\Phi}(\boldsymbol{\mu}_*)$. In this work, a GP for each tangent space coordinate is obtained, by taking as training data the parameter values $\boldsymbol{\mu}_i,\ i = 1,...,n_{\text{DoE}}$ and the associated tangent space coordinates $\beta_j(\boldsymbol{\mu}_i),\ i = 1,...,n_{\text{DoE}}$. The resulting approximation reads:

$$\hat{\beta}_j(\boldsymbol{\mu}) = m_j(\boldsymbol{\mu}) + \epsilon_j(\boldsymbol{\mu})\ \ j = 1,...,n^{\mathbf{T}_P} \tag{3.3}$$

where m_j denotes the mean and ϵ_j is a GP of zero mean conditioned on $\beta_j(\boldsymbol{\mu}_i),\ i = 1,...,n_{\text{DoE}}$. Figure 3.3 illustrates the obtained GP approximations for $\hat{\beta}_1(\boldsymbol{\mu})$ and $\hat{\beta}_2(\boldsymbol{\mu})$.



(a) Coordinate associated with $\vec{e}_1$.

(b) Coordinate associated with $\vec{e}_2$.

Figure 3.3: GP approximations of the tangent plane coordinates. The coordinate values at a new queried point $\boldsymbol{\mu}_*$ are identified with a white triangle.

Using the interpolated coordinates $\hat{\beta}_1(\boldsymbol{\mu}_*)$ and $\hat{\beta}_2(\boldsymbol{\mu}_*)$, it is possible to obtain the interpolated point $\mathcal{Z}_*$ through Eq. (3.2). Then it suffices to apply the exponential mapping of Eq. (2.35) to obtain $\mathcal{Y}_*$. We here recall that our goal is to obtain the best possible approximation of the mapping:

$$\begin{aligned} \mathcal{P} &\mapsto \mathcal{G}(n,N) \\ \boldsymbol{\mu} &\mapsto \mathcal{Y}(\boldsymbol{\mu}) \end{aligned} \tag{3.4}$$

represented by a blue line in Figure 3.1. Thus, if the interpolation of the coordinates of the tangent plane basis is sufficiently accurate, the point $\mathcal{Y}_*$ will lie on the blue line. However, if the initial database does not include enough points, or if they are poorly distributed in the parametric space, the interpolation error may become significant and the point $\mathcal{Y}_*$ might lie far from the blue line. An illustration of this situation is provided in Figure 3.4, where a poorly distributed DoE leads to an interpolated point $\mathcal{Y}_*$ that does not lie on the blue line.

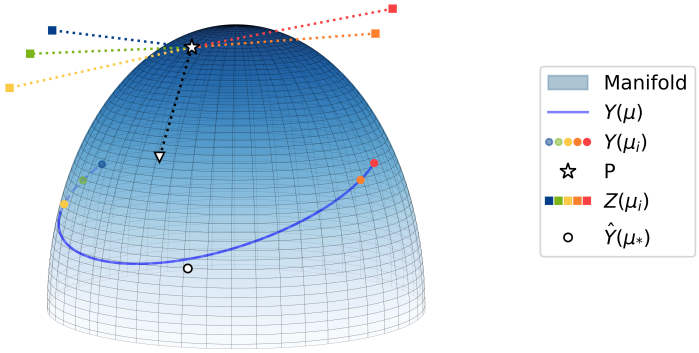


Figure 3.4: Interpolation of the local POD bases using a poorly distributed initial DoE, which results in an interpolated point $\mathcal{Y}_*$ that does not lie on the blue line. The interpolated point $\mathcal{Z}_*$ is identified with a white triangle, while its mapping to the Grassmann manifold is identified with a white dot.

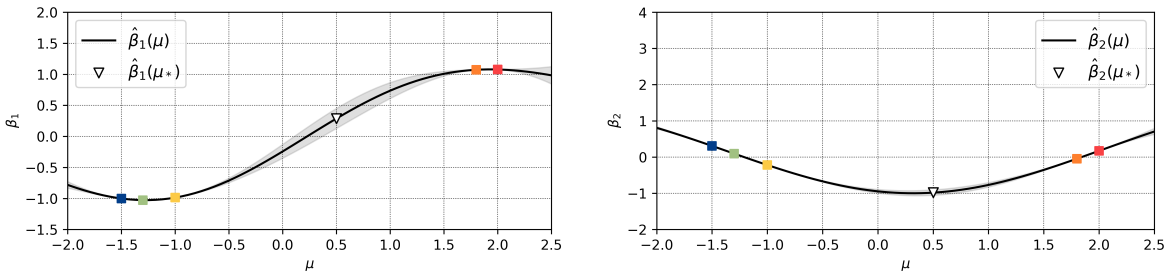
Poorly distributed DoEs, like the one presented in Figure 3.4 can be avoided by using a space-filling strategy to select the parameters $\boldsymbol{\mu}_i$, $i = 1,...,n_d$ at which the local POD bases are built. Nevertheless, the situation depicted in Figure 3.4 raises the important question of how to estimate the quality of the interpolated matrix $\hat{\Phi}(\boldsymbol{\mu}_*)$. We recall that, in this work, we wish to use pointwise local POD bases to reduce the computational cost of solving the MDA problem in a multi-query context. This means that, in an initial *offline* phase we build a database of pointwise local POD bases for one, or several of the disciplinary solvers. Then, when the MDA is queried at

a new point, a new matrix $\hat{\Phi}(\boldsymbol{\mu}_*)$ is obtained by interpolating *online* the existing database of local POD bases. Being able to estimate the quality of the interpolation can help us decide where new local bases should be trained, provided that there is still computational cost available for the sampling of the initial DoE.

3.2.3 Estimating the quality of the interpolation

The most direct way to compute the error committed due to the interpolation of the reduced-order basis $\hat{\Phi}(\boldsymbol{\mu}_*)$ is to obtain new snapshots at point $\boldsymbol{\mu}_*$, build a "true" reduced-order basis $\Phi(\boldsymbol{\mu}_*)$ from the obtained snapshots and compute the geometric distance (Eq. (2.38)) between the subspaces $\hat{\mathscr{Y}}_*$ and $\mathscr{Y}_*$ spanned by the columns of $\hat{\Phi}(\boldsymbol{\mu}_*)$ and $\Phi(\boldsymbol{\mu}_*)$, respectively. If the calculated distance is too large, at any point $\boldsymbol{\mu}_* \in \mathscr{P}$, then the basis $\Phi(\boldsymbol{\mu}_*)$ should be added to the initial database of reduced-order bases. In practice, however, the computational burden associated with this method is too significant for it to be used as an error indicator. Indeed, many snapshots may be required to build the reduced-order basis $\Phi(\boldsymbol{\mu}_*)$. Moreover, to ensure that the interpolation error is small at any queried point $\boldsymbol{\mu}_* \in \mathscr{P}$, the process would have to be repeated throughout the parametric space.

Another idea is thus to use realizations of the GP approximations of the coordinates of the tangent plane basis to estimate the quality of the interpolation. Indeed, by drawing different realizations of $\hat{\beta}_1^{(k)}(\boldsymbol{\mu}_*)$ and $\hat{\beta}_2^{(k)}(\boldsymbol{\mu}_*)$, different bases $\hat{\Phi}^{(k)}(\boldsymbol{\mu}_*)$ can be obtained, with $k = 1, \ldots, n_{\text{MC}}$. When there is a lot of uncertainty in the interpolation, the corresponding points $\mathscr{Y}_*^{(k)}$ in the Grassmann manifold are significantly dispersed, and the basis $\Phi(\boldsymbol{\mu}_*)$ should be added to the database of local POD bases. Figure 3.5, illustrates the confidence intervals of the GP interpolations of $\hat{\beta}_1(\boldsymbol{\mu})$ and $\hat{\beta}_2(\boldsymbol{\mu})$ for the poorly distributed DoE of Figure 3.4, while Figure 3.6 shows the points $\mathscr{Y}_*^{(k)}$, $k = 1, \ldots, n_{\text{MC}}$ on the 2-D simplified manifold. Note that obtaining $\mathscr{Y}_*^{(k)}$, $k = 1, \ldots, n_{\text{MC}}$ by drawing different realizations of $\hat{\beta}_1(\boldsymbol{\mu}_*)$ and $\hat{\beta}_2(\boldsymbol{\mu}_*)$ is essentially cost-free, as no new snapshots need to be computed.



(a) Coordinate associated with $\vec{e}_1$.

(b) Coordinate associated with $\vec{e}_2$.

Figure 3.5: Uncertainty of the GP approximations of the tangent plane coordinates. The 99% confidence intervals of the approximation are shaded in gray and the mean coordinate values at the queried point $\boldsymbol{\mu}_*$ are identified with a white triangle.

A similar approach to the above described is proposed in [Porrello et al., 2024], where the uncertainty of the interpolation of the local reduced-order bases is accounted for in the objective function model of a Bayesian optimization algorithm. In the next section, we discuss how an approximation of the state vector $\tilde{\boldsymbol{w}}(\boldsymbol{\mu}_*)$ is obtained when using local reduced-order bases and study how the uncertainty of the interpolation of local POD bases affects the obtained approximation for the state vector.

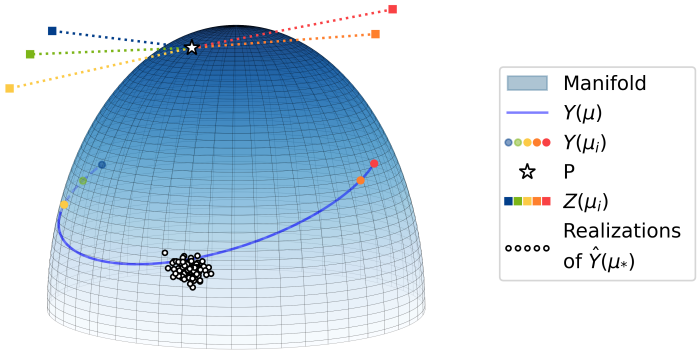


Figure 3.6: Dispersion of the interpolated points $\mathscr{Y}_*^{(k)}$, $k = 1,...,n_{\text{MC}}$. Note that, although there seems to be little uncertainty in the GP approximations of $\hat{\beta}_1(\boldsymbol{\mu})$ and, especially, $\hat{\beta}_2(\boldsymbol{\mu})$, the obtained points $\mathscr{Y}_*^{(k)}$, $k = 1,...,n_{\text{MC}}$ are rather dispersed.

3.3 Local POD bases for single-discipline analysis

Having established a procedure to obtain the matrix $\hat{\Phi}(\boldsymbol{\mu}_*)$, we can now write the state vector at any queried point $\boldsymbol{\mu}_*$ as:

$$\tilde{\boldsymbol{w}}(\boldsymbol{\mu}_*, t) = \boldsymbol{w}_{\text{ref}} + \hat{\boldsymbol{\alpha}}(\boldsymbol{\mu}_*, t)\hat{\Phi}(\boldsymbol{\mu}_*) \tag{3.5}$$

where $\hat{\boldsymbol{\alpha}}(\boldsymbol{\mu}_*, t) = \{\hat{\alpha}_1(\boldsymbol{\mu}_*, t),...,\hat{\alpha}_n(\boldsymbol{\mu}_*, t)\}$ are GP interpolations of the generalized coordinates in the local POD basis $\hat{\Phi}(\boldsymbol{\mu}_*)$ and t denotes a parameter on which the local POD bases do not depend (usually t is the time parameter). To obtain these interpolations, all previously computed snapshots (namely those used to train the set of local POD bases $\Phi(\boldsymbol{\mu}_i)$, $i = 1,...,n_{\text{DoE}}$) are first projected on the interpolated basis $\hat{\Phi}(\boldsymbol{\mu}_*)$. Then a GP approximation is built for each of the generalized coordinates independently. Application of this approach to the case of a wing in transonic flight is presented in the following.

3.3.1 The case of a wing in transonic flight

In this section we consider the steady flow around an aircraft wing, whose governing equations are the compressible Euler equations, introduced in Section 2.1.2.1. Due the compressible effects, when flying at transonic speeds, shock waves may appear on the upper surface of the wing. Increasing or decreasing the angle of attack or the freestream Mach number tends to move the shock wave back and forth. As a result, global POD bases struggle to efficiently represent the transonic flow around the wing when both angle of attack and freestream Mach number are allowed to vary. Figure 3.7 presents the obtained coefficient of pressure distribution on the wing upper surface, for two different values of angle of attack (AoA) and freestream Mach number.

State vector approximation. To compare the use of a global POD basis with the interpolation of a database of local POD bases, we compute the following approximations for the vector $\boldsymbol{f}_a \in \mathbb{R}^{N_a}$ of aerodynamic loads acting upon the wing surface:

$$\boldsymbol{f}_a(\text{AoA}, M_\infty) \approx \sum_{i=1}^{n_a} \hat{\alpha}_a^i(\text{AoA}, M_\infty)\hat{\phi}_a^i(M_\infty) \tag{3.6}$$

$$\boldsymbol{f}_a(\text{AoA}, M_\infty) \approx \sum_{i=1}^{n_a} \hat{\alpha}_a^i(\text{AoA}, M_\infty)\phi_a^i \tag{3.7}$$

where Eq. (3.6) is the approximation obtained via the local POD bases and Eq. (3.7) is the approximation obtained via the global basis. Note that here we have chosen to build the local

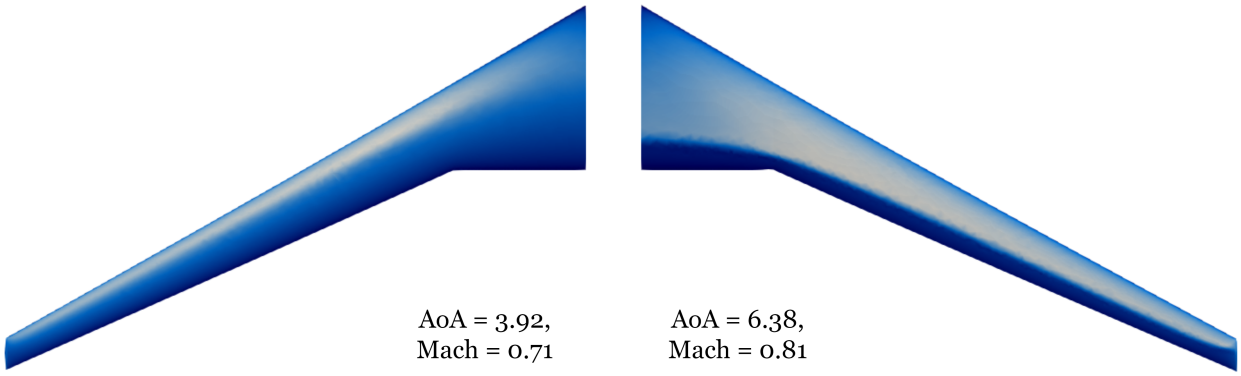


Figure 3.7: Obtained coefficient of pressure distributions for two different values of angle of attack (AoA) and freestream Mach number. On the left, no shock wave is formed. On the right, a shock wave appears close to the trailing edge of the wing.

POD bases at fixed values of freestream Mach number ($\mu = M_\infty$). However, this was an arbitrary choice and we could have equally chosen to train the local POD bases for fixed values of the AoA.

Range of variation of the input parameters. The vector of aerodynamic loads depends on two parameters: the wing angle of attack (AoA) and the freestream Mach number (M_∞). The considered range of variation for each parameter is AoA $\in [1, 12.5]$ and $M_\infty \in [0.65, 0.85]$. However, to account for physical constraints, the parametric space is limited for points that present simultaneously high values of AoA and M_∞. To do so, points are first sampled in the space $[0,1]\times[0,1] \in \mathbb{R}^2$ and then mapped onto the defined input space using the following parametric transformation:

$$M_\infty^* = 0.2 M_\infty + 0.65 \tag{3.8a}$$

$$\text{AoA}^* = (-22.2 M_\infty^* + 25.9)\text{AoA} + 1 \tag{3.8b}$$

where M_∞ and AoA denote the values sampled in the normalized input space and M_∞^* and AoA* denote, respectively, the freestream Mach number and angle of attack at which the governing equations are actually solved. Figure 3.8 illustrates five sampled points in the initial sampling space and their corresponding values in the transformed input space.

3.3.2 Approximation of the state vector via an interpolated local POD basis

To train the local POD bases it is necessary to generate snapshots at fixed values of M_∞ (see Eq. (3.6)). As a result, we choose to sample 5 values of AoA, at 6 different values of M_∞, for a total training cost of 30 calls to the aerodynamics solver. Both samplings are performed using LHS. Figure 3.9(a) illustrates the location of the training snapshots in the transformed input space.

Using the obtained snapshots, a local POD basis is then trained at each sampled value $M_\infty^{(1)}, \ldots, M_\infty^{(6)}$, with the criterion η for the projection error set to 99.99% (see Eq. (2.22)). This leads to a different number of basis vectors n_a in each of the local POD bases. To have the same number of basis vectors in all bases and thus enable interpolation, we set $n_a^* = \max\{n_a^{(1)}, \ldots, n_a^{(6)}\}$. For the present test case, $n_a \in \{3,4\}$, which leads to a choice of 4 vectors in each of the local POD bases.

Projection error. To assess the quality of the interpolated local POD basis $\hat{\Phi}(M_\infty^*)$, we construct a grid of 100 design space points at which we compute the corresponding solution vector. For each point, the relative projection error on the corresponding interpolated local POD basis is computed. The obtained results are presented in Figure 3.9(b). As shown, the relative projection

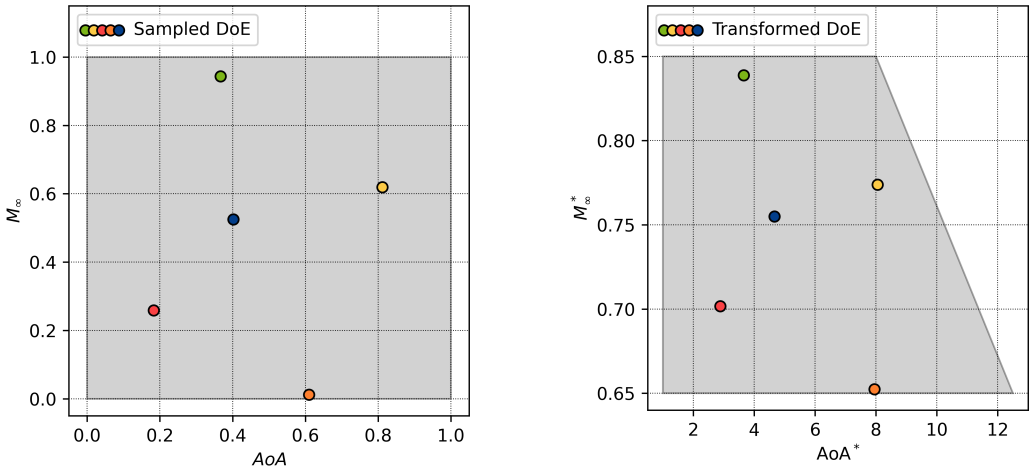
error remains small for the majority of the tested points (91 out of 100 points show less than 5% of projection error), which validates the proposed interpolation strategy. Moreover, we see that the maximum projection error is obtained for $M_\infty^* \approx 0.828$, where the relative projection error is around 7.39%. This means that, if we still had computational cost to spare in the *offline* stage, we might want to train a new local POD basis at this value of M_∞^*. A similar conclusion can be drawn by computing the geometric distance between the subspaces defined by $\Phi(M_\infty^*)$ and $\hat{\Phi}(M_\infty^*)$, respectively, the "true" and approximate local POD bases at M_∞^*. Figure 3.10 plots the computed distances as a function of M_∞^*. As expected, the maximum computed distance occurs for $M_\infty^* \approx 0.828$, respectively the value of M_∞^* that presented the largest projection error. Moreover, we see that, the farther the values of M_∞^* are from the training points, the larger the computed distances.

Finally, although the relative projection error presented above allows us to decide where to train new local POD bases, its computation is rather expensive, as we resort to 100 additional solution vectors. In the following, we attempt to obtain a cost-free error indicator.

Interpolation error and adaptive sampling criterion. Using the GP approximations of the generalized coordinates, we first predict new load vectors for the grid of 100 test points. At each point, we compute the relative interpolation error by comparing the predicted and true load vectors. The obtained results are shown on Figure 3.11(a).

The interpolation error presented above includes both the error due to the interpolation of the basis $\hat{\Phi}(M_\infty^*)$ and the error due to interpolation of the generalized coordinates $\hat{\alpha}_a^1, ..., \hat{\alpha}_a^4$. Nevertheless, the obtained relative interpolation error is only slightly higher than the relative projection error presented in Figure 3.9(b), indicating that most of the committed error is due to the interpolation of $\hat{\Phi}(M_\infty^*)$. Moreover, the overall trend of the interpolation and projection errors is similar throughout the parametric space, with a maximum interpolation error also found at $M_\infty^* \approx 0.828$. This suggests that we can use predicted solution vectors to decide where a new local POD basis should be trained.

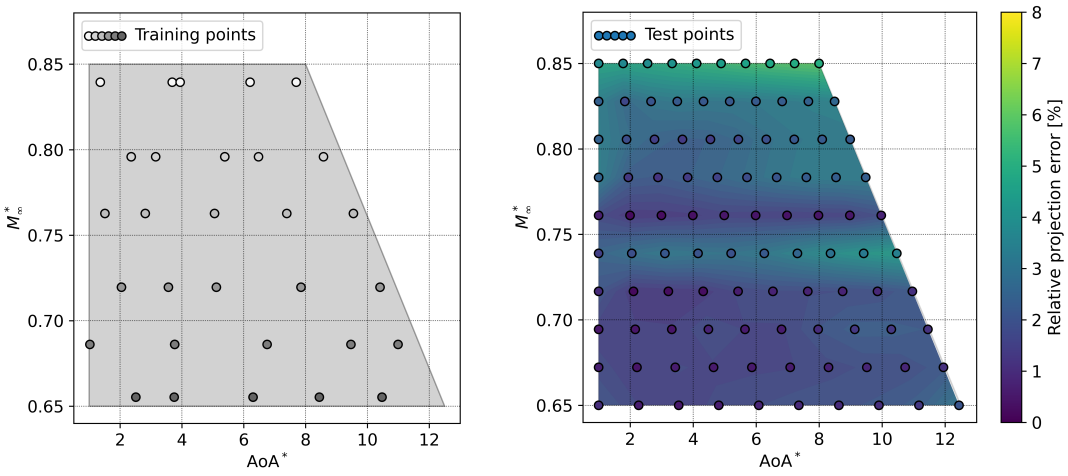
To test this hypothesis, we draw 100 random realizations of the interpolated basis $\hat{\Phi}(M_\infty^*)$ and, for each realization, we predict a solution vector at the mean value of the generalized coordinates. We then compute the standard deviation of the norm of the load vector from the



(a) Sampling space.

(b) Transformed input space.

Figure 3.8: Illustration of (a) five sampled points in the initial sampling space and (b) their corresponding values in the transformed input space.



(a) Training points. (b) Relative projection error.

Figure 3.9: Training and testing of a database of local POD bases. (a) Location of the training snapshots. (b) Grid of tested points and corresponding relative projection error on the interpolated local POD basis. The maximum relative projection error is obtained for $M^*_\infty \approx 0.828$.

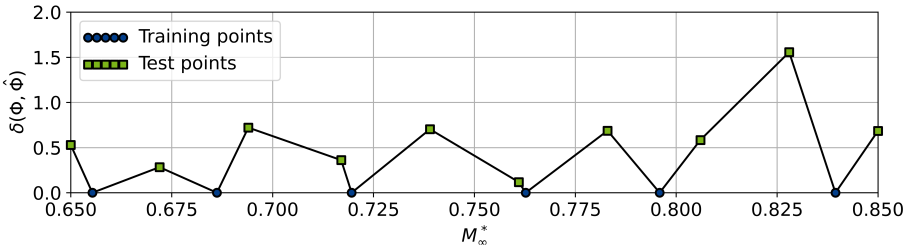
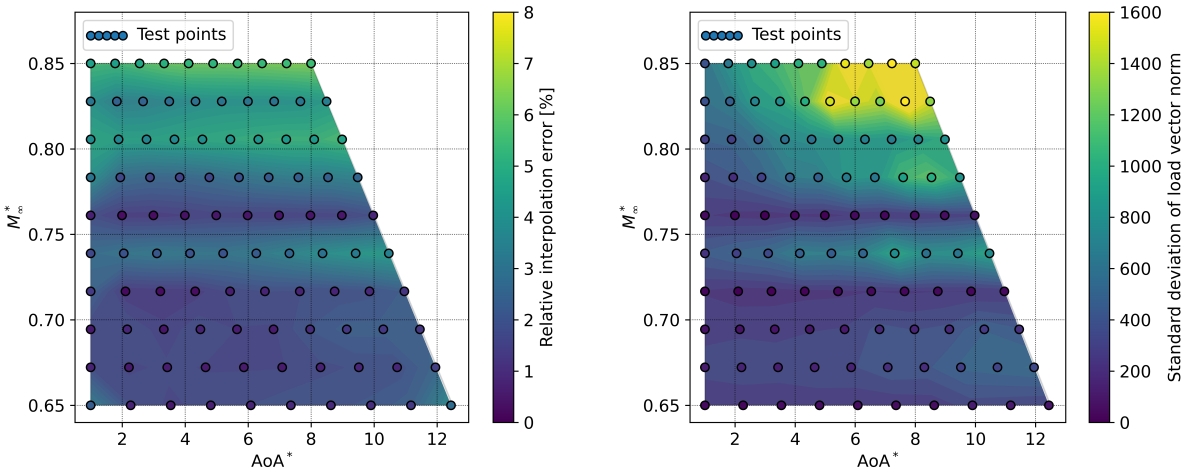


Figure 3.10: Distance between the subspaces defined by the "true" and approximate local reduced-order bases as a function of M^*_∞. The training points are given as reference and the maximum distance is obtained for $M^*_\infty \approx 0.828$, respectively the value of M^*_∞ that presented the largest projection error.

100 random realizations. The obtained results are shown on Figure 3.11(b). We can observe that the computed standard deviation correlates well with both projection and interpolation errors, and therefore constitutes an appropriate criterion for the adaptive sampling of local POD bases. Additionally, this criterion can be evaluated without any additional calls to the aerodynamics solver. Finally, the main difference observed between the standard deviation and error plots occurs close to the border of the parametric space where the freestream Mach number is at its maximum value. This is due to the position of the training snapshots (see Figure 3.9(a)), which leads to a predicted matrix $\hat{\Phi}(M^*_\infty)$ at $M^*_\infty = 0.85$ that is obtained via extrapolation. In the following, we compare the obtained results with those given by a global POD basis.

3.3.3 Approximation of the state vector via a global POD basis

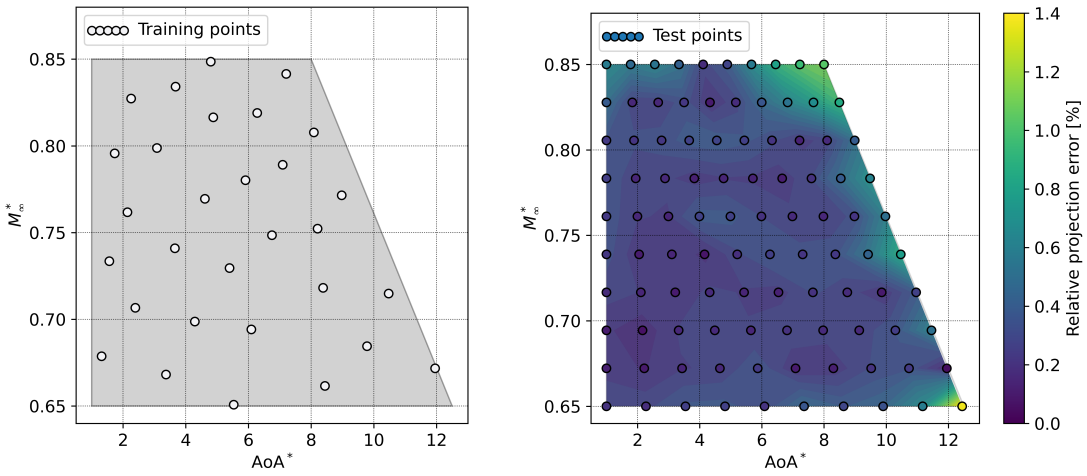
Training of the global POD basis is made by sampling over the entire parametric space. To ensure a space-filling DoE, an LHS strategy is used. Moreover, to ensure a fair comparison, we allow the same computational cost spent during the training of the local POD bases. As a result, 30 points are sampled. Figure 3.12(a) shows the obtained DoE in the transformed input space.



(a) Relative interpolation error.

(b) Standard deviation.

Figure 3.11: Adaptive sampling criterion for the training of new local POD bases. (a) Relative interpolation error between the true state vector and its prediction obtained via the GP approximations of the generalized coordinates in the interpolated local POD basis, for the grid of 100 test points. (b) Standard deviation of the load vector norm obtained from 100 realizations of the interpolated basis $\hat{\Phi}(M_\infty^*)$.



(a) Training points.

(b) Relative projection error.

Figure 3.12: Training and testing of a global POD basis. (a) Location of the training snapshots. (b) Relative projection error on the obtained global POD basis for the grid of 100 test points.

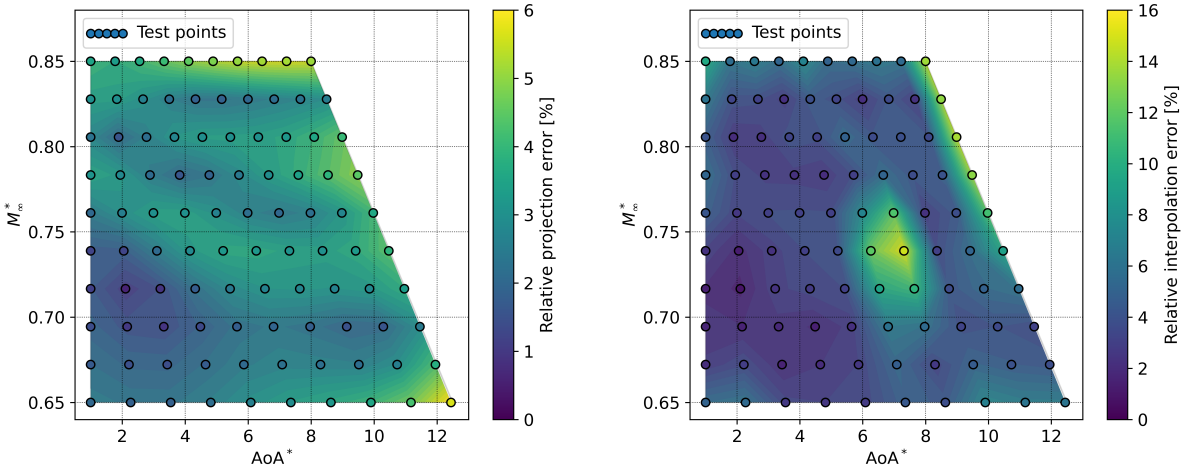
Using the obtained snapshots, a global POD basis is trained, with the criterion η for the projection error set to 99.99%. This leads to $n_a = 18$ basis vectors, significantly more than the 4 vectors used in each of the local POD bases.

Projection error. As was done for the local POD bases, we calculate the projection error on the same grid of 100 test points. The obtained results are given in Figure 3.12(b), where it is shown that the considered global POD basis presents a projection error of less than 1% for the ma-

jority of the tested points, thus outperforming the use of local POD bases. While this result may seem discouraging, it is not unexpected. Indeed, it has been previously discussed in [Goutaudier et al., 2023] that interpolation in the tangent space to Grassmann manifold will always result in a projection error that is greater (or at best the same) as the one given by the a global POD basis trained from the concatenation of the local POD bases snapshots.

We recall, however, that we do not simply want to build a reduced-order basis that accurately represents the solution space. Indeed, our goal is to use approximations of the generalized coordinates to reduce the computational cost of solving the MDA problem. As discussed in [Berthelin, 2022], the DPOD+I strategy struggles when there is a large number of basis vectors because the training inputs of one disciplinary surrogate are the generalized coordinates of the other discipline. In this context, using an aerodynamics POD basis made of 18 vectors is not a feasible option. In the following, we instead consider a global POD basis truncated to the same number of modes as the one used in the local POD bases (i.e. $n_a = 4$). The four modes associated with the four largest eigenvalues are kept.

Truncation to the four most relevant modes. We use the same 30 snapshots of Figure 3.12(a) to train a global POD basis, but this time truncate the approximation to the four modes associated with the four largest eigenvalues. We then compute the projection error on the grid of 100 test points. The obtained results are shown on Figure 3.13(a). As expected, when using only four basis vectors, the projection error increases compared to when 18 vectors were used. Nevertheless, the projection error remains small, with the majority of the test points presenting a relative projection error of less than 5%. Compared with the local POD bases, the maximum relative projection error is smaller when using the truncated global POD basis. However, comparing the mean projection error over the 100 tested points, the interpolated local POD basis performs better than the truncated global POD basis. This is shown on Table 3.1, where the maximum and mean relative projection errors for each of the considered reduced-basis methods are compared.



(a) Relative projection error.

(b) Relative interpolation error.

Figure 3.13: Testing of a global POD basis truncated to the four most relevant modes. (a) Relative projection error and (b) relative interpolation error obtained for the grid of 100 test points.

Method	Local ($n_a^* = 4$)	Global ($n_a = 18$)	Global ($n_a = 4$)
$\max(\epsilon_p^r)$	7.39%	1.36%	5.63%
$\mathbb{E}(\epsilon_p^r)$	2.09%	0.28%	2.65%

Table 3.1: Maximum and mean relative projection errors for the different reduced-order basis methods considered. Values obtained based on the grid of 100 test points.

Next we compute the interpolation error, by predicting new load vectors via interpolation of the generalized coordinates of the truncated global POD basis. The obtained results are presented on Figure 3.13(b) and compared against the other reduced-basis methods in Table 3.2. Comparing with the local POD basis (see Figure 3.11(a)) the interpolation error is more important overall when using the truncated global POD basis. Indeed, both mean and maximum interpolation errors are nearly twice as large for the truncated global POD basis than for the interpolated local POD basis. The global POD basis using 18 vectors performs slightly better than the truncated global POD basis, having obtained a mean relative interpolation error of 3.65% for the 100 tested points. Regardless, the local POD basis method performs the best among all three methods, with a mean relative interpolation error of 2.43%.

Method	Local ($n_a^* = 4$)	Global ($n_a = 18$)	Global ($n_a = 4$)
$\max(\epsilon_i^r)$	7.71%	14.93%	14.80%
$\mathbb{E}(\epsilon_i^r)$	2.43%	3.65%	4.60%

Table 3.2: Maximum and mean relative interpolation errors for the different reduced-order basis methods considered. Values obtained based on the grid of 100 tested points.

3.3.4 Summary of local POD bases for single-discipline analysis

In this section we proposed to use the interpolation of a database of local POD bases to obtain a surrogate approximation for single-discipline analysis. To assess the performance of the proposed approach, we compared the projection and interpolation errors obtained when using local POD bases with those obtained when using a global POD basis truncated to the same number of vectors and with those obtained using a global POD basis with a larger number of vectors. The obtained results showed that, although the global POD basis with a larger number vectors outperforms the remaining approaches in terms of relative projection error, the use of local POD bases leads to a smaller error in the obtained state vector approximation, for the chosen variable separation strategy. This suggests that when using a global POD basis, the value of the generalized coordinates changes significantly when the design variables change, leading to significant interpolation errors. Contrarily, when using local POD bases, the value of the generalized coordinates does not change as much, making the interpolation easier, and reducing the error committed on the obtained approximation of the state vector.

3.3.5 Implementation of the proposed approach in SMT

The implementation of the interpolation of the generalized coordinates using GP approximations, in the context of global POD bases, has been made available in SMT (version 2.6.0) in a class called POD+I. The interpolation of a database of local POD bases using approximations of the coordinates of the tangent plane basis has equally been implemented in SMT (version 2.6.3). All integrations with SMT were carried out by Hugo Reimeringer (Master student at ISAE-SUPAERO) during his five month internship at ONERA. During his time at ONERA, Hugo worked, in part, under my supervision.

3.4 Static aeroelastic analysis of an aircraft wing

In the previous section we saw that, for a single-discipline analysis, the use of local POD bases leads to a less important error on the approximation of the state vector, compared to when a global POD basis is used. Moreover, the obtained local reduced-order basis typically consists of only a few basis vectors (less than 10), which makes this approach appropriate for the construction of disciplinary GPs. In this section we thus explore the use of pointwise local POD bases for disciplinary surrogates in MDA.

3.4.1 Problem formulation

A fluid-structure interaction problem describing the static aeroelastic analysis of an aircraft wing is used to test the proposed approach. This problem couples the aerodynamics and structural disciplines through the following coupled system of equations:

$$\boldsymbol{u}_s = \mathcal{M}_s(\boldsymbol{x}, \boldsymbol{f}_s) \text{ in } \Omega_s \tag{3.9a}$$
$$\boldsymbol{f}_a = \mathcal{M}_a(\boldsymbol{x}, \boldsymbol{u}_a) \text{ in } \Omega_a \tag{3.9b}$$
$$\mathbf{H}\boldsymbol{u}_{s,\Gamma} = \boldsymbol{u}_{a,\Gamma} \text{ on } \Gamma \tag{3.9c}$$

where $\boldsymbol{u}_s \in \mathbb{R}^{d^s}$ and $\boldsymbol{u_a} \in \mathbb{R}^{d^a}$ denote, respectively, the structural and aerodynamic mesh motions, $\boldsymbol{f_s} \in \mathbb{R}^{d^s}$ and $\boldsymbol{f}_a \in \mathbb{R}^{d^a}$ denote the aerodynamic forces acting upon the wing structure in the structural and aerodynamic nodal coordinates; Ω_s and Ω_a denote the structural and aerodynamic domains; $\mathcal{M}_s$ and $\mathcal{M}_a$ are the chosen structural and aerodynamic models; $\mathbf{H}$ is a transfer matrix allowing to express the structural mesh motion on the fluid mesh at the fluid-structure interface Γ; and $\boldsymbol{x} \in \mathcal{X} \in \mathbb{R}^n$ is a set of design variables which influences the aerodynamic behavior, the structural behavior, or both. In the following we assume that $\boldsymbol{u}_s$ and $\boldsymbol{f}_a$ are high-dimensional vectors of dimensions d^s and d^a, arising from the discretization of the corresponding PDEs. Interpolation across the fluid-structure interface is described next.

Interpolation across the fluid-structure interface. In this work we choose to use RBF to perform interpolation across the fluid-structure interface, however, as discussed in Section 1.2.3, other approaches could have been used. Interpolation across the fluid-structure interface seeks to ensure the conservation of energy through the use of the principal of virtual work:

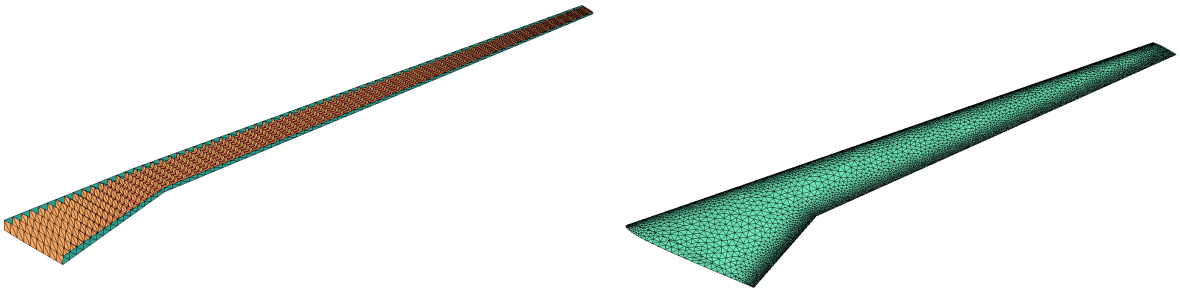
$$\delta W = (\delta \boldsymbol{u}_{s,\Gamma})^\intercal \boldsymbol{f}_{s,\Gamma} = (\delta \boldsymbol{u}_{a,\Gamma})^\intercal \boldsymbol{f}_{a,\Gamma} \tag{3.10}$$

where δW is the virtual work, $\delta \boldsymbol{u}$ are the virtual displacements and $\boldsymbol{f}$ are the load vectors. Using the nodal coordinates, an interpolation matrix $\mathbf{H} \in \mathbb{R}^{d^{a,\Gamma} \times d^{s,\Gamma}}$ may be built using RBF interpolation, as proposed in [Rendall and Allen, 2008]. Note that $d^{a,\Gamma}$ and $d^{s,\Gamma}$ denote the number of nodes of the aerodynamics load vector and structural displacement vector on the fluid-structure interface, respectively. Using the defined interpolation matrix, the structural displacement can then be written at the aerodynamic nodes as $\boldsymbol{u}_{a,\Gamma} = \mathbf{H}\boldsymbol{u}_{s,\Gamma}$ and the aerodynamic loads can be written at the structural nodal points via $\boldsymbol{f}_{s,\Gamma} = \mathbf{H}^\intercal \boldsymbol{f}_{a,\Gamma}$.

3.4.1.1 Disciplinary models

To obtain the structural displacement, the finite element solver Code Aster [EDF, 2017] is used, assuming linear elastic behavior. The built structural finite element model is displayed in Figure 3.14(a). As shown, the model is made of shell elements used to represent the wing ribs, skins and spars. The dimension of structural displacement vector is $d^s = 25524$, respectively the number of translation degrees of freedom. To obtain the aerodynamic loads, the compressible Euler

equations are solved using the finite volume solver SU2 [Palacios et al., 2013, Economon et al., 2016]. The aerodynamic mesh is a three-dimensional mesh containing over 1 million elements, used to model the air flow around the wing. However, only the wing surface mesh, illustrated on Figure 3.14(b), is used in the disciplinary coupling. The total number of nodes on the wing surface mesh is 15697, leading to $d^{a,\Gamma} = 47091$.



(a) Structural mesh. (b) Aerodynamic surface mesh.

Figure 3.14: Example of (a) structural mesh and (b) aerodynamic surface mesh from which the displacement and load vectors are obtained. For the structural mesh, the upper skin was removed to allow for the visualization of the wing internal structure.

3.4.1.2 Choice of design variables

For the design variables, the angle of attack and freestream Mach number are chosen, with the same range of variation defined in Section 3.3.1. As was done before, the design space is limited for points that present simultaneously high values of AoA and M_∞ to account for physical constraints. As a result, the transformed input space of Figure 3.8 is used.

3.4.2 Global DPOD+I strategy

To reduce the computational cost of solving the MDA, disciplinary surrogates combining global reduced-order bases and interpolation of the generalized coordinates can be used to approximate the vectors of structural displacement and aerodynamic loads at the fluid-structural interface, as proposed in [Berthelin, 2022]. The obtained disciplinary surrogates are written as:

$$\mathbb{\Lambda}_s(\hat{\boldsymbol{\alpha}}_a) = \sum_{i=1}^{n_s} \hat{\alpha}_s^i(\hat{\boldsymbol{\alpha}}_a)\boldsymbol{\phi}_s^i \tag{3.11}$$

$$\mathbb{\Lambda}_{a,\Gamma}(\text{AoA}, M_\infty, \hat{\boldsymbol{\alpha}}_s) = \sum_{i=1}^{n_a} \hat{\alpha}_a^i(\text{AoA}, M_\infty, \hat{\boldsymbol{\alpha}}_s)\boldsymbol{\phi}_a^i \tag{3.12}$$

where $\hat{\boldsymbol{\alpha}}_a = \{\hat{\alpha}_1, ..., \hat{\alpha}_{n_a}\}$ and $\hat{\boldsymbol{\alpha}}_s = \{\hat{\alpha}_1, ..., \hat{\alpha}_{n_s}\}$ are GP interpolations of the generalized coordinates of the aerodynamics and structural POD bases, respectively, and $\boldsymbol{\phi}_a^1, ..., \boldsymbol{\phi}_a^{n_a}$ and $\boldsymbol{\phi}_s^1, ..., \boldsymbol{\phi}_s^{n_s}$ are the corresponding reduced-basis vectors. It should be noted that the design parameters are only used directly by the aerodynamics solver. As a result, the structural surrogate will be trained over an input space of dimension n_a, while the aerodynamics surrogate will be trained over an input space of dimension $n_s + 2$ (respectively, $n_s + \text{AoA} + M_\infty$).

To obtain the disciplinary global POD bases we use the training strategy defined in Algorithm 10, with the following parameters. The vector of structural displacements is chosen as the first coupling variable, with initial bounds λ^- and λ^+ set to lie between 0 and 3.6 meters. The number r of initial samples is set to 10 and the tolerance for the mean relative projection error is set

to 1%. Since the training of the POD bases uses randomly generated DoEs, three runs are performed using three different initial DoEs. Table 3.3 shows the number of vectors obtained for the structural and aerodynamic POD bases, respectively, n_s and n_a, as well as the mean relative projection error, ϵ_s^p and ϵ_a^p, obtained for each of the bases. The number of disciplinary solver calls made to each discipline during the training is equally noted.

	n_s	ϵ_s^p[%]	n_a	ϵ_a^p[%]	n_s^{eval}	n_a^{eval}
Run #1	2	0.51	30	0.39	56	66
Run #2	3	0.09	28	0.79	46	56
Run #3	3	0.28	31	0.59	56	66
Avg.	2.7	0.29	29.7	0.59	52.7	62.7

Table 3.3: Number of vectors, maximum relative projection error and number of disciplinary solver calls obtained during the training of the disciplinary global POD bases.

Table 3.3 shows that, while the number of vectors obtained in the structural POD basis is small, the number of vectors obtained for the aerodynamics discipline is rather large, with an average of $n_a = 29.7$ modes obtained in the aerodynamics global POD basis. As a result, despite the achieved reduction in the dimension of the coupling variable space, the dimension of the input space for the structural discipline still presents a challenge for the accurate construction of the respective GP approximations of the generalized coordinates. Moreover, due to the difference in the number of modes obtained in each disciplinary POD basis, the sensitivity analysis is rendered useless, as only the aerodynamics discipline is selected for enrichment. For all these reasons, we can conclude that the use of global POD bases, as proposed in the original DPOD+I strategy, is not compatible with the use of disciplinary GP approximations for the treated MDA problem. In the following, we propose to adapt the DPOD+I strategy to accommodate the use of local reduced-order bases.

3.4.3 Local DPOD+I strategy

The previous section showed that the DPOD+I training strategy, while efficient in reducing the dimension of the coupling variable space, leads to a significant number of vectors in the aerodynamics global POD basis. To address this challenge, we propose to instead use pointwise local POD bases in the approximation of the aerodynamics load vector. To approximate the structural displacement vector, a global POD basis is kept. The new surrogate for the aerodynamics load vector reads:

$$\hat{\boldsymbol{f}}_{a,\Gamma}(\text{AoA}, M_\infty, \hat{\boldsymbol{\alpha}}_s) = \sum_{i=1}^{n_a^*} \hat{\alpha}_a^i(\text{AoA}, M_\infty, \hat{\boldsymbol{\alpha}}_s)\hat{\phi}_a^i(\text{AoA}, M_\infty) \tag{3.13}$$

where $\hat{\Phi}_a(\text{AoA}, M_\infty) = \hat{\phi}_a^1(\text{AoA}, M_\infty), ..., \hat{\phi}_a^{n_a^*}(\text{AoA}, M_\infty)$ is the interpolated local POD basis, obtained as described in Section 3.2. Note that, in this case, we chose to train the local POD bases at fixed design variable values. However, we could have equally chosen to keep only of of the design variables fixed, as we did in Section 3.3.2. Training of the initial database of local POD bases in the context of MDA is discussed in the following. A strategy for the enrichment of the interpolated local POD basis $\hat{\Phi}_a(\text{AoA}, M_\infty)$ during the *online* stage is equally proposed.

3.4.3.1 *Offline* training of local POD bases for MDA

To train a database of local POD bases for the aerodynamics discipline, it is necessary to generate snapshots of the load vector $\boldsymbol{f}_{a,\Gamma}$ for fixed values of $\boldsymbol{x}_a = \{\text{AoA}, M_\infty\}$. Here, we obtain these

snapshots by solving the exact MDA at a set of sampled points $\boldsymbol{x}_a^1, ..., \boldsymbol{x}_a^{n_{\text{DoE}}}$. We recall that, to solve the MDA, a Gauss-Seidel solver is used which calls the disciplinary solvers iteratively. This means that, at each iteration of the algorithm, a snapshot of $\boldsymbol{f}_{a,\Gamma}$ is computed. The obtained snapshots can then be used to train a local POD basis $\Phi(\boldsymbol{x}_a^i)$, $i = 1, ..., n_{\text{DoE}}$. Note that, in practice, the Gauss-Seidel solver does not need to be run until convergence is achieved, but rather it can be stopped once a sufficient number of snapshots has been computed. Here, 10 iterations of the Gauss-Seidel solver are performed at five different design space points, sampled using an LHS strategy. The number of basis vectors kept in each of the local POD bases, as well as the number of disciplinary solver calls made during the training are presented in Table 3.4. Once more, since the number of basis vectors depends on the tested DoE, three different runs are performed.

	n_a^*	n_s^{eval}	n_a^{eval}
Run #1	6	50	50
Run #2	6	50	50
Run #3	6	50	50
Avg.	6	50	50

Table 3.4: Number of aerodynamics basis vectors and number of disciplinary solver calls made during the training of the local POD bases for the aerodynamics discipline.

To train the structural global POD basis, snapshots of $\boldsymbol{u}_s$ obtained during the MDA resolutions of Table 3.4 could be used. Unfortunately, this strategy leads to a structural DoE which is not optimal for the construction of the corresponding GPs approximations, as the corresponding points in the reduced space lie close to one another. This issue could be attenuated by randomly changing the structural design variables in each iteration of the Gauss-Seidel solver, as proposed in the DPOD+I training strategy. However, since, for the present test case, no structural design variables exist (the only variables are AoA and M_∞), we instead propose to train the structural POD basis separately by running the DPOD+I training strategy until the structural projection error is below 1% (see Algorithm 10). Table 3.5 notes the number of basis vectors obtained in the structural global POD basis, as well as the number of disciplinary solver calls made during the training, for three different runs.

	n_s	n_s^{eval}	n_a^{eval}
Run #1	3	36	36
Run #2	3	26	26
Run #3	3	26	26
Avg.	3	29.3	29.3

Table 3.5: Number of structural basis vectors and number of disciplinary solver calls made during the training of the global POD basis for the structural discipline.

Comparing the results of Tables 3.4 and 3.5 with those of Table 3.3, we can see that the number of vectors n_s in the structural reduced-order basis does not change. This is expected as, in both cases, a global POD basis is used. Contrarily, the number of vectors n_a in the aerodynamics reduced-order basis has been drastically reduced from nearly 30 vectors, obtained when using a global POD basis, to only 6 vectors, obtained when using local POD bases.

3.4.3.2 *Online* enrichment of the interpolated local POD basis

Once both disciplinary POD bases have been trained, the approximated MDA can be solved at any point $\boldsymbol{x}_a^*$ using the EGMDA strategy (see Section 1.2.3.2). A remark should nonetheless be made concerning the cases when the interpolated local POD basis $\hat{\Phi}_a(\boldsymbol{x}_a^*)$ does not accurately represent the load vector at a given design space point. Indeed, sometimes, while enriching the aerodynamics surrogate, it can happen that the relative projection error of a given snapshot $\boldsymbol{f}_{a,\Gamma}$ on the corresponding interpolated local POD basis is greater than an allowed threshold. When this is the case, $\hat{\Phi}_a(\boldsymbol{x}_a^*)$ is also enriched, as follows:

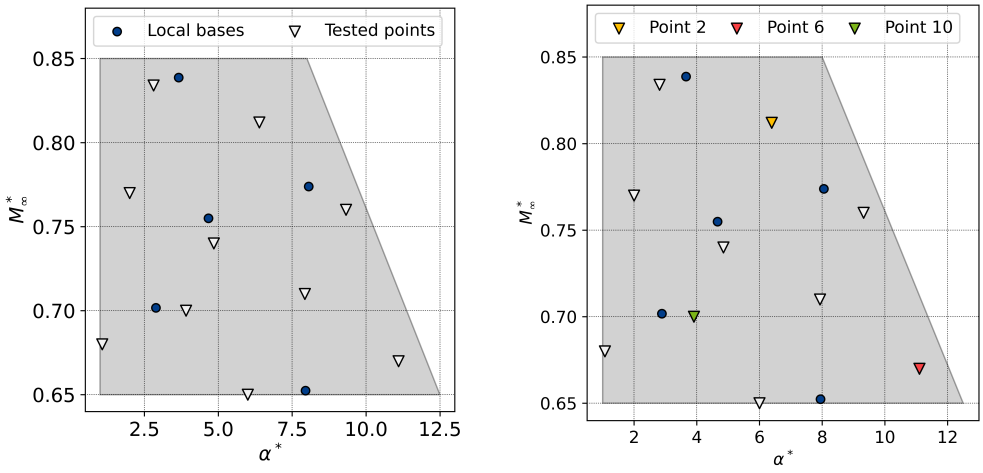
$$\hat{\Phi}_a(\boldsymbol{x}_a^*) = \left[\hat{\phi}_a^1, ..., \hat{\phi}_a^{n_a}, \frac{\boldsymbol{f}_{a,\Gamma}(\boldsymbol{x}_a^*, \hat{\boldsymbol{\alpha}}_s) - \{\hat{\Psi}_a\}^{-1}(\hat{\Psi}_a(\boldsymbol{f}_{a,\Gamma}\boldsymbol{x}_a^*, \hat{\boldsymbol{\alpha}}_s))}{\|\boldsymbol{f}_{a,\Gamma}\boldsymbol{x}_a^*, \hat{\boldsymbol{\alpha}}_s) - \{\hat{\Psi}_a\}^{-1}(\hat{\Psi}_a(\boldsymbol{f}_{a,\Gamma}\boldsymbol{x}_a^*, \hat{\boldsymbol{\alpha}}_s))\|_2}\right] \tag{3.14}$$

where $\hat{\Psi}_a$ and $\{\hat{\Psi}_a\}^{-1}$ denote, respectively, the operations of projection and reconstruction associated with the interpolated local POD basis $\hat{\Phi}_a(\boldsymbol{x}_a^*)$ before enrichment. Note that this criterion is similar to the enrichment criterion already used for the global POD bases (see Eq. (2.49)), with the sole difference being that now the new vector only affects the current MDA resolution. During our numerical tests, we saw that the number of vectors added to $\hat{\Phi}_a(\boldsymbol{x}_a^*)$ was never such that it negatively affected the construction of the disciplinary GPs. In other words, even when adding new vectors to $\hat{\Phi}_a(\boldsymbol{x}_a^*)$, n_a^* usually remains small (typically below 10).

3.4.4 Numerical results

3.4.4.1 Resolution of the true MDA

To study the performance of the proposed approach, the true MDA is first solved at 10 different design space points, sampled using an LHS strategy. Figure 3.15(a) shows the location of the tested points as well as the location of the initial database of local POD bases on the transformed input space.



(a) LHS of tested points.

(b) Points 2, 6 and 10.

Figure 3.15: Tested points for the static aeroelastic analysis of an aircraft wing using local POD bases. (a) LHS of tested points. (b) Points #2, #6 and #10.

The problem is then set up within the OpenMDAO framework and a non-linear Gauss-Seidel solver is chosen to solve the MDA, with stopping criterion set to 1% of relative change on the

coupling variable values between two consecutive iterations. The number of disciplinary solver calls made at each of the tested points is noted in Table 3.6. As shown, when solving the true MDA, the number of disciplinary solver calls needed varies between 3 and 13 calls to each solver, with an average of 7.3 calls obtained from all 10 runs. For the used aerodynamics solver, one solver call can take as long as 15 minutes of computational time, while for the structural solver, one solver call takes around 2 minutes. This means that the average MDA takes almost 2 hours to run while the longest MDA requires over 3 hours of computational time. While these times can be reduced by restarting the aerodynamic computations from previously converged solutions or by employing relaxation strategies (e.g. Aitken acceleration), the resolution of the true MDA in a multi-query context quickly becomes prohibitively expensive.

Point	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Avg.
n_s^{eval}	13	9	7	7	5	3	4	11	7	7	7.3
n_a^{eval}	13	9	7	7	5	3	4	11	7	7	7.3

Table 3.6: Number of disciplinary solver calls made during the resolution of the true MDA at 10 tested design space points.

3.4.4.2 Resolution of the approximated MDA

The approximated MDA is now solved at the same 10 points of Table 3.6, using the following settings. All GP approximations are built using the SMT package, with constant mean trend and squared-exponential correlation function. For the random MDA, the number of Monte Carlo samples drawn for each random variable is set to 1500 and the convergence criterion is chosen so that the 0.9-quantile of the random MDA solutions is less than 1% away from the surrogate mean solution. The maximum projection error allowed is equally set to 1%. If this threshold is surpassed, the respective disciplinary POD basis is enriched. Table 3.7 displays the number of disciplinary solver calls, as well as the number of vectors added to the disciplinary POD bases at each of the tested points. We recall that the vectors added to $\hat{\Phi}_a(x_a^*)$ are only used locally and are not kept when solving the approximated MDA at other points of the design space.

Point	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Avg.
n_s^{eval}	3	4	5	4	2	2	4	3	1	2	3
n_a^{eval}	3	2	2	3	2	2	2	2	1	2	2.1
n_s^{new}	0	0	0	0	0	0	0	0	0	0	0
n_a^{new}	1	1	1	1	1	1	1	1	1	1	1

Table 3.7: Number of disciplinary solver calls made during the resolution of the approximated MDA and number of vectors added to each disciplinary POD basis for the same 10 design space points where the true MDA was solved.

Table 3.7 shows that, on average, the number of disciplinary solver calls required to solve one approximated MDA is half of that needed to solve one true MDA. Notably, the average number of calls made to the structural solver was 3 while the average number of aerodynamics solver calls was 2.1. In terms of computational time, these results translate into less than one hour for the resolution of one MDA. And although the computational time spent during the training of the disciplinary POD bases is non-negligible, it is quickly compensated for applications where the MDA must be solved several times, such as design optimization or reliability analysis. Table 3.8 compares the total computational cost spent when using the real disciplinary solvers with the total computational cost spent when using the disciplinary surrogates.

	True MDA			Approximated MDA		
	Training	Resolution	Total	Training	Resolution	Total
n_s^{eval}	0	73	73	79.3	30	109.3
n_a^{eval}	0	73	73	79.3	21	100.3

Table 3.8: Comparison of the total number of disciplinary solver calls made when solving the true MDA and when solving the approximated MDA at the 10 tested design space points.

Table 3.8 shows that, for the true MDA, solver calls are exclusively made during the *online* stage, and thus the total computational cost is the sum of all solver calls for all queried points. Contrarily, for the approximated MDA, an important number of solver calls are made during the *offline* training of the disciplinary POD bases. These calls must be accounted for in the total computational cost of the approach. Based on the obtained results, it can then be said that solving 10 true MDAs is computationally less expensive than solving 10 approximated MDAs. However, it can be expected that the approximated MDA will become computationally cheaper once the number of required MDA resolutions (n_{MDA}) verifies $3n_{\text{MDA}}+79.3 < 7.3n_{\text{MDA}}$. Taking into account that the number resolutions must be an integer, this condition is true when $n_{\text{MDA}} \geq 19$.

Concerning the enrichment of the disciplinary POD bases, Table 3.7 shows that the obtained global POD basis for the structural discipline was never enriched ($n_s^{\text{new}} = 0$, for all tested points). Contrarily, for the aerodynamics discipline it was always necessary to add one vector to the interpolated POD basis $\hat{\Phi}_a(x_a^*)$ ($n_a^{\text{new}} = 1$, for all tested points). This led to a maximum number of aerodynamics coefficients to interpolate of $n_a^* + 1 = 7$. We do remark however that, although we use the term "enrichment", adding a new vector to the disciplinary POD bases is essentially cost-free, as the used snapshot had already been previously computed to enrich the corresponding GP approximations of the generalized coordinates.

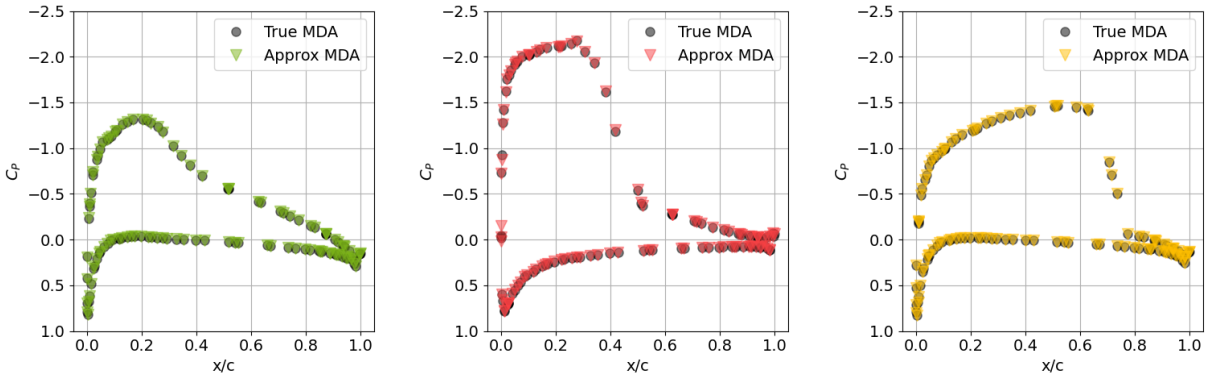
Finally, to assess the quality of the results given by the proposed approach, we compare the relative error between the coupling variable values at the solution of the true MDA and those given by the surrogate mean solution of the approximated MDA. For each tested point, the obtained relative error is shown in Table 3.9.

Point	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Avg.
ϵ_s^{rel}[%]	1.86	0.65	1.00	0.37	0.73	0.79	0.62	0.33	1.21	0.62	0.82
ϵ_a^{rel}[%]	0.87	0.41	0.11	0.16	0.28	0.28	0.18	0.11	0.88	0.16	0.34

Table 3.9: Relative error ϵ_{rel} of the obtained coupling variable values obtained at the surrogate mean solution of the approximated MDA with those given by the true MDA for the 10 tested design space points.

As is shown, the proposed approach is capable of providing accurate results, with average values of the relative error remaining under 1% for both disciplines. The maximum relative error obtained occurred for the displacement vector of the first tested point, where the obtained relative error was of 1.86%. The accuracy of the obtained solutions is also reflected in the obtained coefficient of pressure distributions, presented in Figure 3.16 for three design space points. The position of the chosen points in the transformed input space is shown in Figure 3.15(b). The corresponding coefficient of pressure distributions are given using the same color code.

Figure 3.16 shows that the coefficient of pressure distributions obtained via the approximated MDA are very close to those obtained via the true MDA. This is in agreement with the small relative error obtained in Table 3.9. We further remark the absence of shock wave for tested



(a) AoA = 3.92°. $M_\infty = 0.70$. (b) AoA = 11.1°. $M_\infty = 0.67$. (c) AoA = 6.38°. $M_\infty = 0.81$.

Figure 3.16: Comparison of the coefficient of pressure distribution at the wing half span, for three of the tested points. (a) Point #10. (b) Point #6. (c) Point #2.

point #10 (in green), and the two different shock wave positions for tested points #6 and #2 (in red and yellow, respectively). The existence of such different solutions justifies the large number of basis vectors obtained when using a global POD basis in the approximation of the aerodynamics load vector. Finally, we note that, although the coefficient of pressure distributions presented in Figure 3.16 only compare the obtained results at the wing half span, the good agreement between the true and approximated MDAs is present throughout the rest of the wing. This is evidenced in Figure 3.17, where the true and approximated pressure distributions over the wing upper surface are compared for Point #2 (identified in yellow in Figure 3.15(b)).

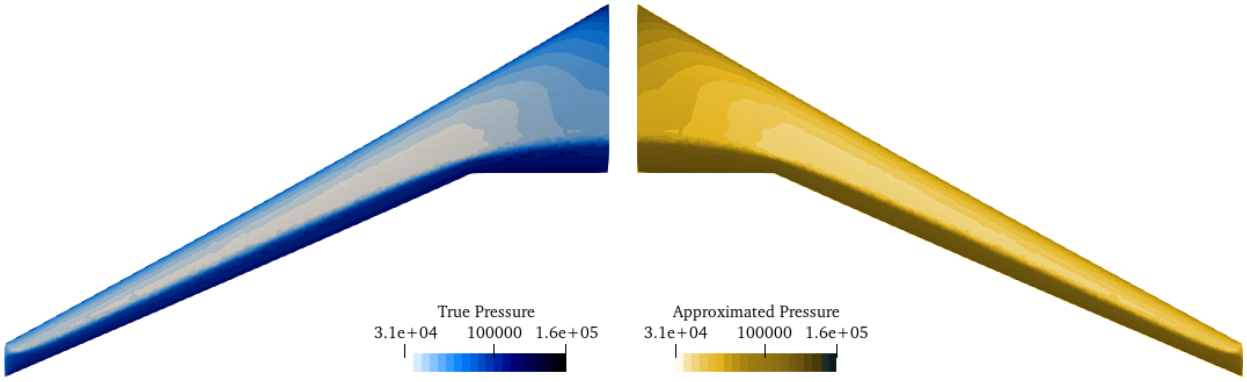


Figure 3.17: Comparison of the obtained pressure distribution over the wing upper surface when solving the true MDA with that obtained when using the disciplinary surrogates. The provided example is for AoA = 6.38° and $M_\infty = 0.81$ (Point #2).

3.4.5 Summary of local POD bases for MDA

In this chapter we proposed the use of local POD bases for building disciplinary surrogates in MDA. In this context, the use of global POD basis has been shown to require a significant number of basis vectors for cases where the disciplinary outputs are very sensitive to certain parameter changes. This, in turn, leads to a large interpolation error for the built GP approximations of the generalized coordinates. The use of pointwise local POD bases, on the other hand, allows to keep a small number of basis vectors and retain a small interpolation error in the approximation

of the generalized coordinates. Application to a static aeroelastic analysis of a wing in transonic flight confirmed the interest of the proposed approach for applications requiring several MDA resolutions. Not only was the computational cost linked to the resolution of one MDA reduced, but also very little compromise is made in terms of accuracy of the obtained solution.

In terms of future perspectives, a more in-depth study of the variable separation strategy used in the local POD basis approximation of Eq. (3.13) should be considered. Indeed, it may have been more interesting to train the local POD basis in terms of either angle of attack or freestream Mach number, rather than both. Application to systems where both aerodynamics and structural disciplines require the use of pointwise local POD bases could equally be envisioned as future work, in order to properly assess the robustness of the proposed approach.

Chapter summary:

In this chapter we proposed the use of local POD bases for building disciplinary surrogates in MDA. To that end, we first introduced a strategy for the interpolation of local POD bases using GPs. The main steps were:

- *Obtain a basis of the tangent plane to the Grassmann manifold at a point of tangency $\boldsymbol{P}$;*
- *Interpolate the coordinates of the tangent plane basis using GPs.*

A strategy for estimating the quality of the POD basis interpolation was equally proposed, by generating random realizations of the interpolated matrix and looking at the dispersion of the corresponding points in the Grassmann manifold.

The use of local POD bases for single-discipline analysis was introduced next. Using as illustrative example the case of a wing in transonic flight, we showed that:

- *The number of basis vectors in the interpolated local POD basis is less than the one obtained for a global POD basis, trained over the entire parametric space;*
- *The projection error is greater for the interpolated local POD basis than for the global POD basis. However, the projection error is smaller for the interpolated local POD basis, than for a global POD basis truncated to the same number of vectors as the interpolated local POD basis;*
- *The interpolation error is smaller when using local POD bases, compared to when a global POD basis is used, for all tested truncation choices.*

Testing the proposed approach in the static aerolastic analysis of an aircraft wing, subject to different flight conditions in terms of angle of attack and freestream Mach number further confirmed the interest of the proposed approach for applications requiring many MDA resolutions. Indeed, the use of local POD bases allowed for:

- *The reduction of the average computational cost of one MDA resolution;*
- *Almost no compromise in terms of accuracy of the obtained solution, with average relative errors remaining below* 1% *compared to when the real solvers are used.*

CHAPTER 4

Disciplinary surrogates for gradient-based MDO

Contents

The developments presented in this chapter led to the following conference paper, which was presented at the AeroBest conference in July 2023:

> I. Cardoso, S. Dubreuil, N. Bartoli, C. Gogu, M. Salaün, R. Lafage, *Disciplinary surrogates for gradient based optimization of multidisciplinary systems*, in proceedings of II ECCOMAS Thematic Conference on Multidisciplinary Design Optimization of Aerospace Systems. ISBN: 978-989-53599-4-3

Another communication was made on the ECCOMAS congress in June 2024, which included more recent developments, namely those involving the use of pointwise local POD bases.

In this chapter we will:

- *Introduce the analytical derivatives of the disciplinary GPs.*
- *Introduce the derivatives of the DPOD+I surrogates.*
- *Present numerical results for an analytical test case.*
- *Present numerical results for a wing structural sizing problem using local POD bases to reduce the dimension of the aerodynamics load vector.*

4.1 Chapter contributions

In Chapter 3 we introduced disciplinary surrogates which combine pointwise local POD bases with GP interpolation of the generalized coordinates to reduce the computational cost of solving the MDA in a multi-query context. Adaptive enrichment of the disciplinary surrogates at queried points allowed to keep the training cost to a minimum while still achieving a desired level of accuracy in the obtained MDA solution. In this chapter we propose to use these same disciplinary surrogates to solve MDO problems. Moreover, we propose to use the derivatives of the built disciplinary surrogates, which can be obtained analytically, to drive the search for a local optimum. Application to both an analytical and engineering test cases confirms the interest of the proposed approach in terms of computational cost.

4.2 Analytical derivatives of the disciplinary surrogates

An advantage of using disciplinary GPs is that, although the derivatives of the true disciplinary solvers are assumed to be unavailable, these can be approximated analytically from the built disciplinary surrogates. Through the chain rule, it is then possible to obtain the total derivatives of any objective or constraint function which depend on the solution of the MDA, using either the direct or adjoint methods, as described in Section 1.2.5.1. In the following, we detail the computation of the derivatives of the disciplinary GPs with respect to the design variables in the case of scalar coupling variables. We then extend this concept to disciplinary surrogates combining global or local POD bases with GP interpolation.

4.2.1 Derivative computation for scalar coupling variables

When the coupling variables are scalars, the partial derivative of the GP approximation $\hat{\boldsymbol{y}}_i(\boldsymbol{x}, \boldsymbol{y}_{c_i})$ with respect to the design variable x_k is given by:

$$\frac{\partial \hat{\boldsymbol{y}}_i(\boldsymbol{x}, \boldsymbol{y}_{c_i})}{\partial x_k} = \frac{\partial \mu_i(\boldsymbol{x}, \boldsymbol{y}_{c_i})}{\partial x_k} + \frac{\partial \sigma_i(\boldsymbol{x}, \boldsymbol{y}_{c_i})}{\partial x_k} \xi_i, \ \forall \boldsymbol{x} \notin \boldsymbol{x}_{\mathrm{DoE}_i}, \ i=1,\dots,n_d, \tag{4.1}$$

where μ_i denotes the mean of the disciplinary GP, σ_i its standard deviation and ξ_i is a standard Gaussian random variable. The obtained derivative is thus a random variable whose mean value is given by the first term of Eq. (4.1) and whose uncertainty is described by the second term of the same equation. Because the uncertainty on the value of the derivative stems from the uncertainty of the corresponding disciplinary GP, we can reduce it at any queried point, by first enriching the disciplinary surrogates via the EGMDA strategy. This allows to neglect the second

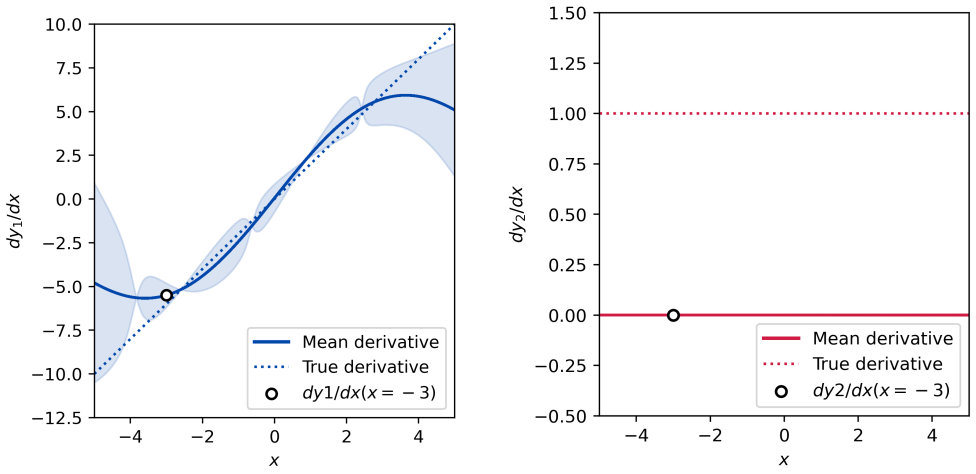
term of Eq. (4.1) and rewrite the derivative of $\hat{\boldsymbol{y}}_i(\boldsymbol{x}, \boldsymbol{y}_{c_i})$ with respect to the design variable x_k as:

$$\frac{\partial \hat{\boldsymbol{y}}_i(\boldsymbol{x}, \boldsymbol{y}_{c_i})}{\partial x_k} \approx \frac{\partial \mu_i(\boldsymbol{x}, \boldsymbol{y}_{c_i})}{\partial x_k}, \quad i = 1, ..., n_d \tag{4.2}$$

Illustration of the proposed derivative approximation. To illustrate the proposed derivative approximation, we consider the 1-D illustrative example of Chapter 1, whose multidisciplinary analysis is given by the following non-linear system of equations:

$$\begin{cases} y_1(x, y_2) = x^2 - \cos(\frac{y_2}{2}) \\ y_2(x, y_1) = x + y_1 \end{cases} \tag{4.3}$$

Figure 4.1 compares the true derivatives of the disciplinary solvers of Eq. (4.3) with the derivatives of the corresponding disciplinary surrogates, before any surrogate enrichment. In both plots, the input coupling variable values are fixed to their respective value at the MDA solution for $x^* = -3$.

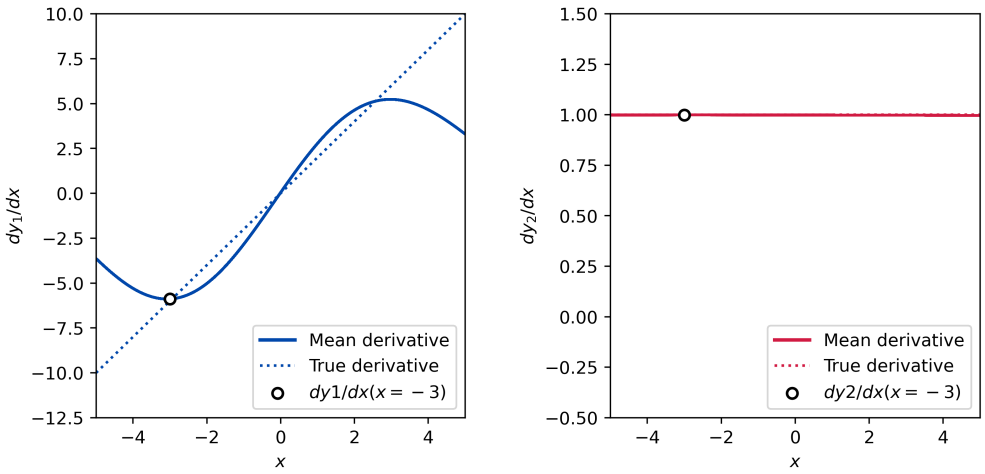


(a) Discipline 1. (b) Discipline 2.

Figure 4.1: Derivative of the disciplinary surrogates of the 1-D illustrative example before enrichment. Confidence intervals are given to illustrate the associated uncertainty and the true derivative is given as reference.

Figure 4.1 shows that, before enrichment of the disciplinary surrogates, the value of the disciplinary surrogate derivatives on the queried point $x^* = -3$ does not match that of the true derivatives. For the first disciplinary surrogate, the derivative uncertainty hints at the fact that the computed derivative value may be false. However, for the second disciplinary surrogate, there seems to be almost no uncertainty, even though the value of the approximate derivative does not match the true value. This illustrates the importance of enriching the disciplinary surrogates before computing the corresponding derivatives. Figure 4.2 illustrates the same derivatives after the disciplinary surrogates have been enriched at point $x^* = -3$.

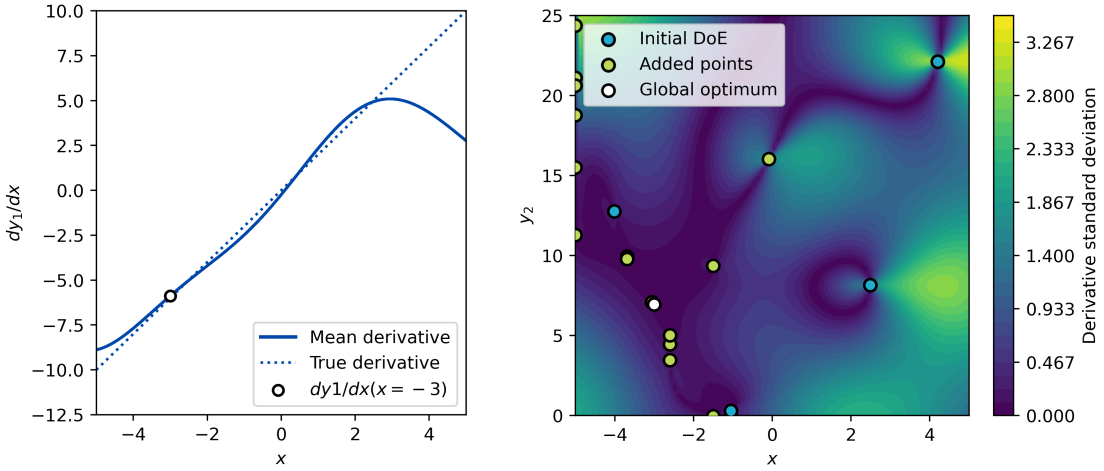
As shown, after enrichment, the mean value of the derivatives at $x^* = -3$ is close to their true value for both disciplinary surrogates, which justifies the proposed simplification (Eq. (4.2)). We can remark, however, that around the queried point, a large error is committed on the value of the derivative for the first disciplinary surrogate. In the following we will show that, as gradient-based optimizers tend to query the MDA at points around the local optimum, the quality of the disciplinary derivative approximations will also improve around the local optimum.



(a) Discipline 1. (b) Discipline 2.

Figure 4.2: Derivative of the disciplinary surrogates of the 1-D illustrative example after enrichment. For both disciplinary surrogates, the mean derivative is close to the true derivative at the enrichment point.

Improvement of the derivative approximation during the resolution of the MDO problem. When solving the optimization problem, gradient-based optimizers tend to query the MDA at points around the local optimum. In our approach, for each queried point, we enrich the disciplinary surrogates via the EGMDA strategy. As a consequence, the quality of the disciplinary derivative approximations is also improved around the local optimum. This is illustrated in Figure 4.3, where the derivative of the first disciplinary surrogate is obtained, after having solved the MDO problem using the proposed approach.



(a) Predicted derivative for discipline 1. (b) Derivative standard deviation.

Figure 4.3: Improvement of the predicted derivative thanks to the enrichments made during the resolution of the MDO. (a) Predicted derivative for discipline 1 at a fixed value of y_2. (b) Standard deviation of the predicted derivative for discipline 1. The initial DoE for discipline 1 as well as the added points are identified in blue and green dots, respectively.

Figure 4.3(a) shows that, thanks to the enrichments made during the resolution of the MDO problem, the prediction of the derivative of the first disciplinary surrogate around the global optimum has been significantly improved. Moreover, Figure 4.3(b) further confirms that the enrichments made during the resolution of the MDA lead to a reduction of the uncertainty on the value of the derivative around the global optimum, at $x^* = -3$. Indeed, the derivative standard deviation is close to zero in this region. Finally, we remark that for the presented example, the problem was run from starting point $x = -1.5$. Using a different starting point, the region around the local optimum might have been improved instead.

4.2.2 Derivative computation for vector-valued coupling variables

When the coupling variables are high-dimensional vectors, we first reduce the dimension of the coupling variable space, either by means of a global POD basis, or via the interpolation of a database of pointwise local POD bases, as proposed in Chapter 3. Whichever the case, the derivative of the disciplinary surrogates is no longer given by Eq. (4.2). In the following, we discuss how the derivatives of the disciplinary surrogates are obtained depending on the chosen dimension reduction strategy.

Using global POD bases. When using a disciplinary global POD basis as dimension reduction strategy, only the generalized coordinates depend on the set of design variables. As a result, the derivative of $\hat{\boldsymbol{y}}_i(\boldsymbol{x}, \boldsymbol{\alpha}_{c_i})$ with respect to the design variable x_k can be obtained analytically from the GP approximations of the generalized coordinates, as follows:

$$\frac{\partial \hat{\boldsymbol{y}}_i(\boldsymbol{x}, \hat{\boldsymbol{\alpha}}_{c_i})}{\partial \boldsymbol{x}_k} = \sum_{j=0}^{n_i} \left(\frac{\partial \mu_i^j(\boldsymbol{x}, \hat{\boldsymbol{\alpha}}_{c_i})}{\partial x_k} + \frac{\partial \sigma_i^j(\boldsymbol{x}, \hat{\boldsymbol{\alpha}}_{c_i})}{\partial x_k} \xi_i \right) \phi_i^j, \quad i = 1, ..., n_d \tag{4.4}$$

As for the scalar coupling variables, we assume that uncertainty has been previously reduced thanks to the enrichment of the disciplinary surrogates, and thus Eq. (4.4) can be approximated as:

$$\frac{\partial \hat{\boldsymbol{y}}_i(\boldsymbol{x}, \hat{\boldsymbol{\alpha}}_{c_i})}{\partial \boldsymbol{x}_k} \approx \sum_{j=0}^{n_i} \frac{\partial \mu_i^j(\boldsymbol{x}, \hat{\boldsymbol{\alpha}}_{c_i})}{\partial x_k} \phi_i^j, \quad i = 1, ..., n_d \tag{4.5}$$

Using pointwise local POD bases. When using the interpolation of a database of pointwise local POD bases as dimension reduction strategy, not only the generalized coordinates, but also the basis vectors are a function of the set of design variables (or a subset of it, depending on the variable separation strategy employed). In this case, the derivative of $\hat{\boldsymbol{y}}_i(\boldsymbol{x}, \boldsymbol{\alpha}_{c_i})$ with respect to the design variable x_k is given by:

$$\frac{\partial \hat{\boldsymbol{y}}_i(\boldsymbol{x}, \hat{\boldsymbol{\alpha}}_{c_i})}{\partial \boldsymbol{x}_k} = \sum_{j=0}^{n_i} \frac{\partial}{\partial x_k} \left(\hat{\boldsymbol{\alpha}}_i(\boldsymbol{x}, \hat{\boldsymbol{\alpha}}_{c_i}) \hat{\phi}_i^j(\boldsymbol{x}) \right), \quad i = 1, ..., n_d \tag{4.6}$$

To obtain the derivative of the state vector with respect to the design variables it is thus necessary to differentiate the interpolation of the coordinates in the tangent plane basis. Unfortunately, differentiation across the exponential map is not trivial due to the singular value decomposition (see Eq. (2.35)). As a result, in this work we simplify the derivative calculation when using pointwise local POD bases, by assuming that the basis vectors can be considered constant in a small neighborhood of the queried point. The approximated derivative reads:

$$\frac{\partial \hat{\boldsymbol{y}}_i(\boldsymbol{x}, \hat{\boldsymbol{\alpha}}_{c_i})}{\partial \boldsymbol{x}_k} \approx \sum_{j=0}^{n_i} \left(\frac{\partial \mu_i^j(\boldsymbol{x}, \hat{\boldsymbol{\alpha}}_{c_i})}{\partial x_k} + \frac{\partial \sigma_i^j(\boldsymbol{x}, \hat{\boldsymbol{\alpha}}_{c_i})}{\partial x_k} \xi_i \right) \hat{\phi}_i^j(\boldsymbol{x}), \quad i = 1, ..., n_d \tag{4.7}$$

Finally, assuming that uncertainty has been previously reduced thanks to the enrichment of the disciplinary surrogates, Eq. (4.7) can be further simplified to:

$$\frac{\partial \hat{\boldsymbol{y}}_i(\boldsymbol{x}, \hat{\boldsymbol{\alpha}}_{c_i})}{\partial \boldsymbol{x}_k} \approx \sum_{j=0}^{n_i} \frac{\partial \mu_i^j(\boldsymbol{x}, \hat{\boldsymbol{\alpha}}_{c_i})}{\partial x_k} \hat{\phi}_i^j(\boldsymbol{x}), \;\; i = 1, ..., n_d \tag{4.8}$$

In the following, numerical tests are used to assess the performance of the proposed approach.

4.3 Numerical tests

To test the use of disciplinary GPs for gradient-based MDO, we first solve a benchmark analytical problem using scalar coupling variables. Then we apply the proposed approach to a wing structural sizing problem, where the coupling variables are the structural displacement vector and the aerodynamics load vector. The obtained results are presented in the following.

4.3.1 Application to an analytical test case

The Sellar benchmark test case [Sellar et al., 1996] is a constrained MDO problem using analytical disciplinary solvers. It is defined as:

$$\begin{aligned} \underset{z \in \mathcal{Z}}{\arg\min} \quad & z_3^3 + z_2 + y_1^* + \exp(-y_2^*) \\ \text{s.t.} \quad & 3.16 - y_1^* \leq 0 \\ & y_2^* - 24 \leq 0 \end{aligned} \tag{4.9}$$

where $\boldsymbol{z} = \{z_1, z_2, z_3\}$ is the set of design variables, $\mathcal{Z}$ is the design space defined as $\mathcal{Z} = [-10, 10] \times [0, 10] \times [0, 10]$ and $\boldsymbol{y}^* = \{y_1^*, y_2^*\}$ is the solution of the following non-linear system of equations:

$$\begin{cases} y_1 = z_1^2 + z_2 + z_3 - 0.2 y_2 \\ y_2 = \sqrt{y_1} + z_1 + z_2 \end{cases} \tag{4.10}$$

The Sellar problem presents both a global and local optima. The design variable, objective and constraint function values at each optima are summarized in Table 4.1.

Optimum	z_1	z_2	z_3	f_{obj}^*	$3.16 - y_1$	$y_2 - 24$
Global	1.9776	0.0	0.0	3.1834	0	-20.2472
Local	-1.7171	0.1384	0.1128	4.1307	0	-23.8145

Table 4.1: Global and local optima for the Sellar test case. Objective and constraint function values at each optima are given. Note that the second constraint function is not active for either global nor local optima.

In the following, we solve the Sellar problem using a local optimization algorithm, ran from different starting points. Depending on the chosen starting point, the optimizer will be expected to converge to either global or local optima. A run will thus be considered successful if either of the two points is found.

4.3.1.1 Reference results

We first solve the MDO problem of Eq. (4.9) using the real disciplinary solvers. To that end, the problem is set up in OpenMDAO, using the following settings. As MDO formulation we use an MDF approach, where we solve the MDA at each iteration using a Gauss-Seidel solver. The convergence criterion for the Gauss-Seidel solver is chosen as 0.1% of relative change in the coupling variable values between two successive iterations. For the optimizer, the gradient-based SLSQP is chosen with convergence criterion set to 10^{-4}. Under the assumption of black-box disciplinary solvers, we obtain the derivatives of the objective and constraint functions via finite-difference approximation. The problem is then run from five different starting points, sampled using an LHS strategy. The optima found as well as the number of disciplinary solver calls made are given in Table 4.2.

z^0	z_1^*	z_2^*	z_3^*	f_{obj}^*	n_1^{eval}	n_2^{eval}
#1	1.9776	0.0	0.0	3.1834	127	127
#2	1.9776	0.0	0.0	3.1834	125	125
#3	-1.7171	0.1384	0.1128	4.1307	188	188
#4	-1.7171	0.1384	0.1128	4.1307	142	142
#5	1.9776	0.0	0.0	3.1834	110	110
				Total number of solver calls	692	692

Table 4.2: Optima found and number of disciplinary solver calls obtained when solving the Sellar MDO problem using the real disciplinary solvers in an MDF approach. Optimization results obtained from five different starting points.

The obtained results show that the optimizer successfully converged to the local optimum for runs #3 and #4 and to the global optimum for runs #1, #2 and #5. However, due to the finite-difference approximation of the derivatives, a significant number of solver calls is required. To ease this computational burden, an IDF approach may be used. We recall that, in the IDF approach, the MDA is not solved at each queried point, but instead consistency constraints are added to ensure the feasibility of the optimal solution. In this manner, the IDF spares computational cost not only in each iteration of the optimizer, but also during the finite-difference computation of the derivatives. Table 4.3 shows the optima found as well as the number of disciplinary solver calls made when using an IDF approach to solve the Sellar MDO problem. To allow for a fair comparison, we choose as starting points the same points used for the MDF approach.

Comparing the results of Table 4.3 with those obtained using the MDF approach we see that, by employing an IDF formulation, the computational cost is significantly reduced (less than half of the solver calls made using the MDF approach). Nevertheless, for starting point #3 the IDF approach returned a point that is neither global nor local optimum.

4.3.1.2 Proposed approach

We now solve the Sellar problem using the disciplinary surrogates. To that end, we build initial disciplinary DoEs of four points each, using as guess for the coupling variable space bounds $y_1 \in [0,25]$ and $y_2 \in [0,25]$. For the optimizer, the gradient-based SLSQP is kept, with the derivatives calculated at the mean of the disciplinary GPs, as proposed in Eq. (4.2). To solve the MDA, the EGMDA strategy is used with convergence criterion set to $\epsilon_{\text{CV}} = 0.001$. The optima found as well as the number of disciplinary solver calls made when using the disciplinary surrogates are shown

z^0	z_1^*	z_2^*	z_3^*	f_{obj}^*	n_1^{eval}	n_2^{eval}
#1	1.9776	0.0	0.0	3.1834	56	56
#2	1.9776	0.0	0.0	3.1834	50	50
#3	-1.5655	0.7879	0.1210	4.3304	61	61
#4	1.9776	0.0	0.0	3.1834	67	67
#5	1.9776	0.0	0.0	3.1834	62	62
				Total number of solver calls	296	296

Table 4.3: Optima found and number of disciplinary solver calls obtained when solving the Sellar MDO problem using the real disciplinary solvers in an IDF approach. Optimization results obtained from the same starting points used for the MDF approach.

in Table 4.4, for the same five starting points used in the MDF and IDF approaches. The relative errors committed on the solution points found and corresponding objective function value are equally presented. These were calculated with respect to the closest optimum.

z^0	$\hat{z}_1^*$	$\hat{z}_2^*$	$\hat{z}_3^*$	$\epsilon_{\text{rel}}(\hat{z}^*)[\%]$	$\hat{f}_{\text{obj}}^*$	$\epsilon_{\text{rel}}(\hat{f}_{\text{obj}}^*)[\%]$	n_1^{eval}	n_2^{eval}
#1	1.9776	0.0	0.0	$\leq 10^{-2}$	3.1834	$\leq 10^{-2}$	15	14
#2	1.9783	0.0	0.0	0.04	3.1828	0.02	10	10
#3	-1.7089	0.1287	0.1492	2.23	4.1223	0.2	13	11
#4	-1.7157	0.1407	0.1153	0.21	4.1306	$\leq 10^{-2}$	13	9
#5	1.9776	0.0	0.0	$\leq 10^{-2}$	3.1834	$\leq 10^{-2}$	12	10
						Number of disciplinary DoE points	4	4
						Total number of solver calls	67	58

Table 4.4: Optima found and number of disciplinary solver calls obtained when solving the Sellar MDO problem using the disciplinary surrogates, with stopping criterion chosen as $\epsilon_{\text{CV}} = 0.001$. Optimization results obtained from the same five starting points used for the MDF and IDF approaches.

The results of Table 4.4 show that, when using the disciplinary surrogates, the optima found depend not only on the starting point, but also on the quality of the disciplinary surrogate approximations. Indeed, for starting points number #1 and #5 the global optimum was found with a relative error of less than 0.01%. However, for starting point #2 the quality of the disciplinary GP approximations led the optimizer to a slightly more optimistic solution. Similarly, points #3 and #4 did not converge to the local optimum, but to a point in a neighborhood of it. Despite the small differences in the points found, the obtained results still present a good accuracy, with a maximum relative error in the design variable values of 2.23%, committed for starting point number #3. This led to an objective function value that is 0.2% more optimistic. In terms of computational cost, we remark that using the disciplinary surrogates allows to significantly reduce the number of disciplinary solver calls made compared to when the real disciplinary solvers are used. Comparing the obtained results with those of Tables 4.2 and 4.3, we see that the disciplinary surrogates required 10 times less solver calls than the MDF approach and 4 times less solver calls than the IDF approach.

Influence of EGMDA stopping criterion. Because the quality of the computed derivative approximations depends on the quality of the obtained disciplinary surrogates, we can expect that

the chosen value for ϵ_{CV} will play an important role on the obtained results. To test this hypothesis, we solve the MDO problem using the proposed approach, but this time we relax the EGMDA stopping criterion to $\epsilon_{CV} = 0.1$. The obtained results are presented on Table 4.5.

z^0	$\hat{z}_1^*$	$\hat{z}_2^*$	$\hat{z}_3^*$	$\epsilon_{rel}(\hat{z}^*)$[%]	$\hat{f}_{obj}^*$	$\epsilon_{rel}(\hat{f}_{obj}^*)$[%]	n_1^{eval}	n_2^{eval}
#1	1.9489	0.0037	0.1051	5.51	3.1986	0.47	6	5
#2	2.4623	0.0	0.0	24.51	3.1600	0.73	4	0
#3	1.9784	0.0	0.0	0.4	3.1830	0.01	8	4
#4	-1.7176	0.1113	0.1375	2.12	4.1291	0.04	9	4
#5	1.9773	0.0	0.0	0.02	3.1833	$\leq 10^{-2}$	5	4
					Number of disciplinary DoE points		4	4
					Total number of solver calls		36	21

Table 4.5: Optima found and number of disciplinary solver calls obtained when using a relaxed stopping criterion (ϵ_{CV}) in the EGMDA strategy.

Table 4.5 shows that, by using a more relaxed stopping criterion, we allow some uncertainty to exist in the random MDA solutions. As a consequence, the obtained MDO solutions lie farther away from either global or local optima, with only runs #3 and #5 presenting a relative error of less than 1% in design variable values obtained. For run #2, the quality of the disciplinary surrogates led the optimizer to converge to neither local or global optima, which had not occurred when a stricter stopping criterion was used. Despite the committed error on the design variable values, the obtained objective function values remain close to the their true optimal value, suggesting that the use of a relaxed criterion can nonetheless be used to quickly explore the design space and obtain an idea of the best objective function value attainable. Indeed, because the imposed stopping criterion does not require the disciplinary surrogate approximations to be as accurate, the number of calls to the disciplinary solvers has been drastically reduced from 67 to 36 calls for the first disciplinary solver and from 58 to 21 calls for the second disciplinary solver.

Quality of the computed derivatives. To assess the quality of the computed derivatives, we compare the approximate derivatives with the true derivatives of the Sellar problem, for both strict ($\epsilon_{CV} = 0.001$) and relaxed ($\epsilon_{CV} = 0.1$) stopping criteria. The relative error committed on the norm of the total derivatives of the objective and constraint functions at the global optimum is presented in Table 4.6.

	$f_{obj}^{\epsilon_{CV}=0.001}$	$g_1^{\epsilon_{CV}=0.001}$	$g_2^{\epsilon_{CV}=0.001}$	$f_{obj}^{\epsilon_{CV}=0.1}$	$g_1^{\epsilon_{CV}=0.1}$	$g_2^{\epsilon_{CV}=0.1}$
$\epsilon_{rel}\left(\frac{d}{dx}\right)$	0.35%	0.50%	1.61%	3.54%	3.32%	43.3%

Table 4.6: Relative error committed on the total derivatives of the objective and constraint functions. Values computed for both strict ($\epsilon_{CV} = 0.001$) and relaxed ($\epsilon_{CV} = 0.1$) stopping criteria.

The results of Table 4.6 show that the chosen stopping criterion for the EGMDA strategy greatly impacts the quality of the computed the derivatives. Indeed, when using a stricter stopping criterion, the relative error committed on the total derivatives remains small for all quantities of interest. Contrarily, when using a relaxed criterion, the relative error becomes more important, having obtained as much as 43.3% of relative error on the total derivative for the second constraint function. We do note, however, that this constraint is not active at the global opti-

mum, which explains why the proposed approach is still capable of finding the global optimum, despite the committed error.

4.3.2 Application to a wing structural sizing problem

We now apply the proposed approach to solve a wing structural sizing problem, whose MDA is the one defined in Section 3.4. The disciplinary solvers used to obtain the aerodynamic loads and structural displacements are also kept unchanged, and the same disciplinary models are considered. The MDO problem is then defined as a mass minimization problem under failure constraints, accordingly:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathscr{X}}{\arg\min} \quad & m_{\mathrm{w}}(\boldsymbol{x}) \\ \text{s.t.} \quad & \frac{\mathrm{vMIS}(\boldsymbol{x},\boldsymbol{u}_s,\boldsymbol{f}_{a,\Gamma})}{\sigma_y/2} - 1 \leq 0 \\ & \frac{L(\boldsymbol{x},\boldsymbol{u}_s,\boldsymbol{f}_{a,\Gamma})}{2.5W} - 1 = 0 \end{aligned} \tag{4.11}$$

where $\boldsymbol{x}$ is the set design variables, $\mathscr{X}$ the corresponding design space, m_{w} is the wing mass, vMIS denotes the maximum value of the von Mises stress and σ_y is the material yield stress. In practice, to avoid discontinuities in the constraint function, the actual maximum of the von Mises stress is not calculated at each iteration, but is evaluated at a same structural mesh node, where the maximum stress occurs for the chosen design variable bounds. Moreover, the stress constraint is evaluated for a 2.5g pull-up maneuver condition at an altitude of 0ft and a freestream Mach number of 0.64. To that end, an additional constraint is added to the problem which imposes the generated lift L to be 2.5 times the aircraft weight W. An illustration of the wing internal structure is given in Figure 4.4, where the main structural components (ribs, skins and spars) are identified.

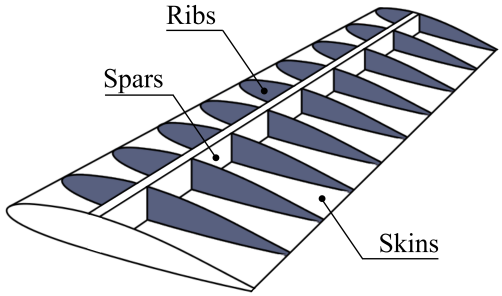


Figure 4.4: Wing internal structure. Identification of the structural components considered in the mass minimization problem.

Choice of design variables. To size the wing internal structure, the thicknesses of the wing ribs, skins and spars are chosen as design variables. The range of variation of each variable is scaled to take values in $[0,1]$, however, the actual thickness bounds are chosen so that a sufficient variation of the MDA outputs is obtained. Moreover, to be able to satisfy the lift constraint, the angle of attack (AoA) must be added to the set of design variables, leading to $\boldsymbol{x} = \{\mathrm{AoA}, t_{\mathrm{r}}, t_{\mathrm{sk}}, t_{\mathrm{sp}}\}$. The angle of attack is also scaled to take values in $[0,1]$, but a sufficient range of variation is allowed to ensure that the $L = 2.5W$ condition can be met. Figure 4.5 illustrates the deformation of the structural mesh, obtained for the $L = 2.5W$ condition, when all scaled thickness are set to either zero or one. As is shown, for the same flight condition, the vertical wing tip displacement is

significantly larger when all thickness parameters are set to zero (in this condition, the vertical wing tip displacement was of 2.32 meters).

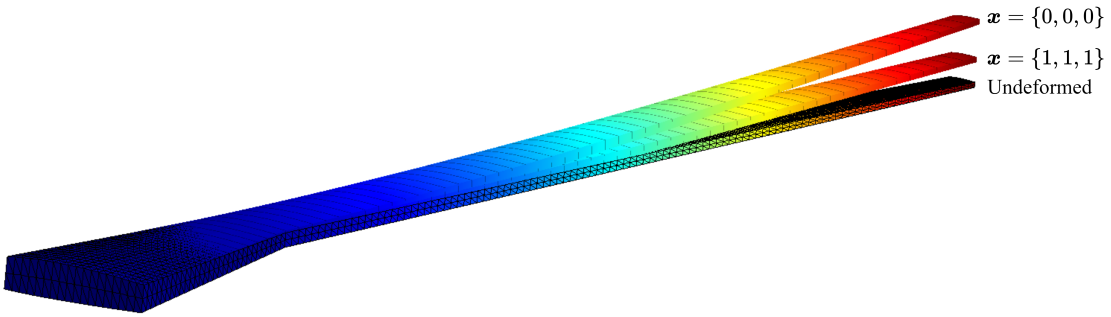


Figure 4.5: Deformation of the wing structural mesh when the scaled thicknesses are all set to either zero or one. The undeformed shape of the wing is given as reference.

4.3.2.1 Training the disciplinary surrogates

As dimension reduction strategy we choose to use a global POD basis for the structural discipline and the interpolation of a database of local POD bases for the aerodynamics discipline. The obtained disciplinary surrogates for the structural displacement and aerodynamic loads at the fluid-structure interface are written as:

$$\mathbf{u}_s(t_\mathrm{r}, t_\mathrm{sk}, t_\mathrm{sp}, \hat{\boldsymbol{\alpha}}_a) = \sum_{i=1}^{n_s} \hat{\alpha}_s^i(t_\mathrm{r}, t_\mathrm{sk}, t_\mathrm{sp}, \hat{\boldsymbol{\alpha}}_a)\phi_s^i \tag{4.12}$$

$$\mathbf{f}_{a,\Gamma}(\mathrm{AoA}, \hat{\boldsymbol{\alpha}}_s) = \sum_{i=1}^{n_a} \hat{\alpha}_a^i(\mathrm{AoA}, \hat{\boldsymbol{\alpha}}_s)\phi_a^i(\mathrm{AoA}) \tag{4.13}$$

where $t_\mathrm{r}, t_\mathrm{sk}, t_\mathrm{sp}$ denote the thicknesses of the wing ribs, skins and spars, respectively. To train the aerodynamics local POD bases ($\phi_a^i(\mathrm{AoA})$), it is necessary to generate snapshots at fixed values of angle of attack. On the other hand, to obtain a global POD basis (ϕ_s^i) for the structural discipline that accurately represents the whole solution space, some variation in the training snapshots is desired. We thus propose to use a training strategy similar to the one employed in the DPOD+I strategy, where we solve the MDA for fixed values of angle of attack, but change the thickness values at each iteration. In total, five MDAs are run, for five different values of angle of attack. Each MDA is allowed to run for 8 iterations, leading to a total computational cost for the training of the disciplinary surrogates of 40 solver calls to each discipline. In terms of the dimension of the reduced coupling variable space, the training of the disciplinary POD bases led to 4 vectors in the structural global POD basis and to 4 vectors in each of the aerodynamics local POD bases.

4.3.2.2 Computation of the approximate derivatives

To test the accuracy of the built disciplinary surrogates, we first solve the MDA and compute the derivatives at three different design space points. For the real disciplinary solvers, for which the derivatives are not available, finite-difference approximation is used. To solve the true MDA a Gauss-Seidel solver is used with convergence criterion set to 1% of relative change in the coupling variable values between two successive iterations. To solve the approximated MDA, the EGMDA strategy is employed, with convergence criterion set to $\epsilon_\mathrm{CV} = 0.01$. Enrichment of the

disciplinary POD bases is performed, when necessary, as described in Section 3.4.3.2. Table 4.7 presents the obtained values for the objective and inequality constraint functions at the solution of the true and approximated MDAs. The number of disciplinary solver calls required for the resolution of the MDA and computation of the total derivatives for each of the approaches is equally given.

	Real solvers				Disciplinary GPs				
Point	m_w^* [kg]	g_{vMIS}^*	n_s^{eval}	n_a^{eval}	m_w^* [kg]	$\hat{g}_{\text{vMIS}}^*$	ϵ_{vMIS} [%]	n_s^{eval}	n_a^{eval}
#1	16396.6	-0.3516	32	32	16396.6	-0.3458	1.65	2	1
#2	8651.1	-0.2324	32	32	8651.1	-0.2268	2.41	2	1
#3	22402.3	-0.5144	24	24	22402.3	-0.5113	0.60	1	0
Training	−−	−−	0	0	−−	−−	−−	40	40
Total	−−	−−	88	88	−−	−−	−−	45	42

Table 4.7: Obtained MDA solutions using the real disciplinary solvers and the disciplinary surrogates for three different design space points. The number of disciplinary solver calls required to solve the MDA and compute the derivatives is equally noted and the error ϵ_{vMIS} committed on the constraint function value when using the disciplinary surrogates is given.

Table 4.7 shows that solving the true MDA and computing the total derivatives for the real disciplinary solvers requires an important number of solver calls. Contrarily, when using the disciplinary surrogates, the computational cost can be significantly reduced. Even when accounting for the training of the disciplinary surrogates, the use of the disciplinary GPs represents about half the computation effort than that spent using the real disciplinary solvers. This ratio can be expected to further decrease in favor of the disciplinary surrogate approach, as the number of queried points increases. Moreover, we see that almost no error is committed on the approximated constraint function value, with maximum relative error obtained for test point #2, where the relative error was 2.41%. Note that the objective function does not depend on the disciplinary outputs and therefore it has the same value, regardless of the approach used.

Quality of the computed derivatives. To assess the quality of the computed derivatives, we compare the total derivatives of the approximated MDA (computed analytically from the disciplinary surrogates), with the derivatives of the true MDA (approximated via finite-differences) at the three design space points of Table 4.7. The obtained results are presented in Table 4.8.

	Real solvers			Disciplinary GPs			
$\boldsymbol{x}^*$	$\frac{\mathrm{d}g_{\text{vMIS}}^0}{\mathrm{d}t_r}$	$\frac{\mathrm{d}g_{\text{vMIS}}^0}{\mathrm{d}t_{sk}}$	$\frac{\mathrm{d}g_{\text{vMIS}}^0}{\mathrm{d}t_{sp}}$	$\frac{\mathrm{d}\hat{g}_{\text{vMIS}}^0}{\mathrm{d}t_r}$	$\frac{\mathrm{d}\hat{g}_{\text{vMIS}}^0}{\mathrm{d}t_{sk}}$	$\frac{\mathrm{d}\hat{g}_{\text{vMIS}}^0}{\mathrm{d}t_{sp}}$	$\epsilon_{\text{rel}}\left(\frac{\mathrm{d}\hat{g}_{\text{vMIS}}^0}{\mathrm{d}\boldsymbol{x}}\right)$
#1	-0.0567	-0.4084	-0.0736	-0.0697	-0.4558	-0.0985	13.2%
#2	-0.1563	-0.5807	-0.1000	-0.0882	-0.5490	-0.1696	16.8%
#3	-0.0461	-0.2338	-0.0521	-0.0303	-0.1585	-0.0876	34.7%

Table 4.8: Total derivatives of the inequality constraint function with respect to the design variables at three different design space points. Comparison between the true and approximated MDAs.

The results of Table 4.8 show that, despite the small error committed on the constraint function value at the different MDA solutions computed, the approximated derivatives do not quite match the true derivative. Indeed, for points number #1 and #2, the error on the constraint function derivative is of around 15%, while for point #3 the same error was twice as large. We note

that, for the same point, only one disciplinary solver enrichment was made, which might explain the obtained error. Despite the committed error, the approximated derivatives seem to correctly explain the overall behavior of $\hat{g}_{\text{vMIS}}$ with respect to variations in the design variables (same sign and same order of magnitude are obtained). In the following a mass minimization problem at fixed angle of attack is presented.

4.3.2.3 Mass minimization at fixed angle of attack

By fixing the angle of attack we are left with only structural design variables. This means that we can compute the derivatives at the mean of the disciplinary surrogates analytically, without any simplifications (we recall that a global POD basis is used for the structural discipline). The chosen value for the angle of attack is such that, at the optimal solution, the generated lift is close but not equal to 2.5 times the aircraft weight. The obtained optima, as well as the number of disciplinary solver calls made are given in Table 4.9 for three different starting points. We note that the three runs are performed sequentially, and thus runs #2 and #3 benefit from previous enrichments of the disciplinary surrogates.

$\boldsymbol{x}^0$	m_{w}^0 [kg]	t_{r}^*	t_{sk}^*	t_{sp}^*	$\epsilon_{\text{rel}}(\boldsymbol{x})$	m_w^* [kg]	$\epsilon_{\text{rel}}(m_w^*)$	$\hat{g}_{\text{vMIS}}^*$	n_s^{eval}	n_a^{eval}
#1	16396.6	0	0.1435	0.1096	6.8%	4201.7	1.65%	$\approx 10^{-4}$	12	6
#2	8651.1	0	0.1302	0.1505	17.6%	4225.7	2.24%	$\approx 10^{-4}$	6	1
#3	22402.3	0	0.1329	0.1468	15.6%	4183.0	1.20%	$\approx 10^{-5}$	6	1
								Training of the disciplinary surrogates	40	40
								Total number of solver calls	64	48

Table 4.9: Optima found and number of disciplinary solver calls made for the mass minimization problem at fixed angle of attack. The error committed on the design variable and objective function values with respect to the reference solution is equally given. Problem ran from three different starting points, with chosen optimizer the gradient based SLSQP and convergence criterion set to 10^{-3}.

Looking at the results of Table 4.9 we remark that, for all three runs, the optimizer was able to converge to a solution which respects the approximate failure constraint (denoted $\hat{g}_{\text{vMIS}}^*$). Moreover, regardless of the starting point, the solution found lied in the same region of the design space, suggesting that there is only one local optimum and that the small differences obtained are likely due to the quality of the disciplinary GP approximations. To find the real local optimum, we then solved the MDO problem using as starting point the design space point found for run #1. A reference solution was then established at $\boldsymbol{x}^* = \{0, 0.1358, 0.1192\}$ for which the wing mass was 4131.3kg. Comparing this result with the optima obtained via the disciplinary surrogates, we note that all solutions found lie close to the reference solution, with maximum relative error in the design variable values of 17.6%, obtained for starting point #2. Despite the error committed on the design variable values, however, this result translated into only 2.24% of error in the obtained objective function value.

In terms of disciplinary solver calls, Table 4.9 shows the advantage of using disciplinary surrogates in a multi-start context. Indeed, because the angle of attack was kept fixed, the aerodynamics load vector did not significantly change with respect to the design variables. As a result, runs #2 and #3 required significantly less calls to the aerodynamics solver than run #1, as they benefited from the several enrichments of the aerodynamics surrogate made during the first run. Moreover, we see that the overall computational cost of solving the MDO problem using the disciplinary surrogates (accounting for all three runs) required a total of 64 calls to the structural solver and 48 calls to the aerodynamics solver. This represents less solver calls than that spent

when querying the true MDA at only three different points (see Table 4.7). Finally, to illustrate the progression of the optimization process, we plot in Figure 4.6 the objective and constraint function values as well as the number of disciplinary solver calls throughout the iterations, for the three performed runs. As shown, for all runs, most disciplinary solver calls happen during the first few iterations. Indeed, in the last few iterations, the optimizer is mostly querying points around the local optimum for which the disciplinary surrogates are already sufficiently accurate (according to the chosen MDA stopping criterion). Comparing all three runs, it is also noticeable that run #2 struggled more to converge than the remaining runs, despite not having significantly changed the values of either objective or constraint function after iteration 5. This is likely due to some existing uncertainty in the constraint function approximation, which makes it difficult for the algorithm to properly establish the edge of the feasible region.

Quality of the computed derivatives. To assess the quality of the computed derivatives at the optimal solution, we compare the total derivatives of the approximated MDA (computed analytically from the GP approximations of the generalized coordinates), with the derivatives of the true MDA (approximated via finite-differences) at the three solution points of Table 4.9. The obtained results are presented in Table 4.10.

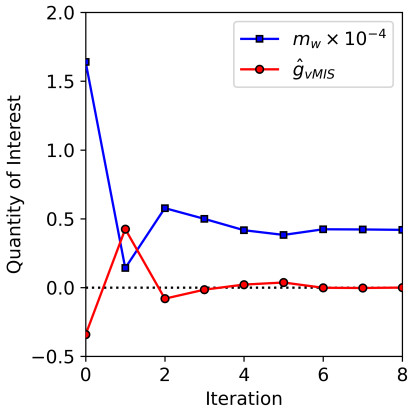
	Real solvers			Disciplinary GPs			
$\boldsymbol{x}^*$	$\frac{\mathrm{d}g^*_{\mathrm{vMIS}}}{\mathrm{d}t_{\mathrm{r}}}$	$\frac{\mathrm{d}g^*_{\mathrm{vMIS}}}{\mathrm{d}t_{\mathrm{sk}}}$	$\frac{\mathrm{d}g^*_{\mathrm{vMIS}}}{\mathrm{d}t_{\mathrm{sp}}}$	$\frac{\mathrm{d}\hat{g}^*_{\mathrm{vMIS}}}{\mathrm{d}t_{\mathrm{r}}}$	$\frac{\mathrm{d}\hat{g}^*_{\mathrm{vMIS}}}{\mathrm{d}t_{\mathrm{sk}}}$	$\frac{\mathrm{d}\hat{g}^*_{\mathrm{vMIS}}}{\mathrm{d}t_{\mathrm{sp}}}$	$\epsilon_{\mathrm{rel}}\left(\frac{\mathrm{d}\hat{g}^*_{\mathrm{vMIS}}}{\mathrm{d}\boldsymbol{x}}\right)$
#1	-0.4859	-1.0407	-0.2719	-0.1815	-1.0588	-0.3000	25.9%
#2	-0.4341	-1.0748	-0.2304	-0.1562	-1.3263	-0.3727	33.9%
#3	-0.4762	-1.0711	-0.2348	-0.2139	-1.0983	-0.2707	22.3%

Table 4.10: Total derivatives of the inequality constraint function with respect to the design variables at the three solution points found for the mass minimization problem at fixed angle of attack. Comparison between the true and approximated MDAs.

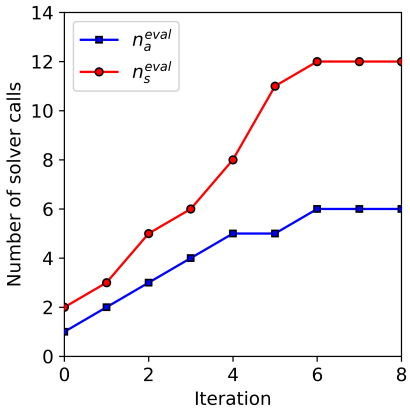
Comparing the derivatives of the true and approximated MDAs, we notice that we obtain very different results for the derivative of the inequality constraint function with respect to the first design variable ($\mathrm{d}g^*_{\mathrm{vMIS}}/\mathrm{d}t_{\mathrm{r}}$) for all tested points. This can be explained by the fact that all optimal points found lied on the edge of the design space for this variable (see Table 4.9). Because the disciplinary surrogates cannot be enriched outside the provided design variable bounds, an important error is committed when computing the derivative of the disciplinary surrogates at the edges of the design space. We also note that for starting point #2 the obtained disciplinary approximations led to an error on the total derivative that is more important than the one obtained for the remaining points. Once more, this suggests that there may still be some uncertainty on the constraint function approximation at this design space point. This eventually led the optimizer towards a solution point that is less optimal than the ones obtained for runs #1 and #3. In the following we solve the MDO problem considering the pull-up maneuver condition.

4.3.2.4 Structural sizing at pull-up maneuver conditions

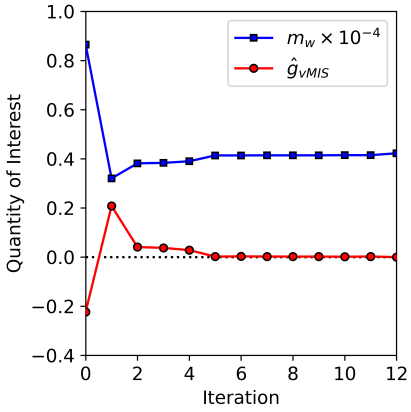
In this section we solve the structural sizing problem at pull-up maneuver conditions, as defined in Eq. (4.11). Because the angle of attack is now allowed to vary, the computed derivatives for the aerodynamics surrogate with respect to the design variables are no longer exact, but are instead an approximation obtained by assuming that the basis vectors are constant in a small neighborhood of the queried point (see Eq. (4.7)). Table 4.11 presents the optima found, the



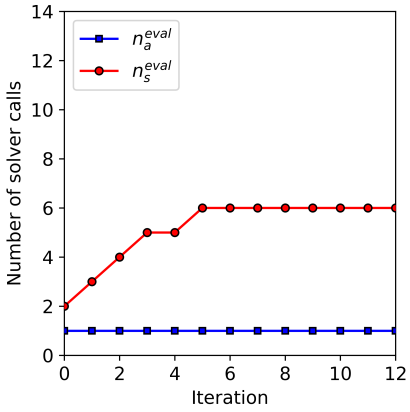
(a) Run #1. Quantities of interest.



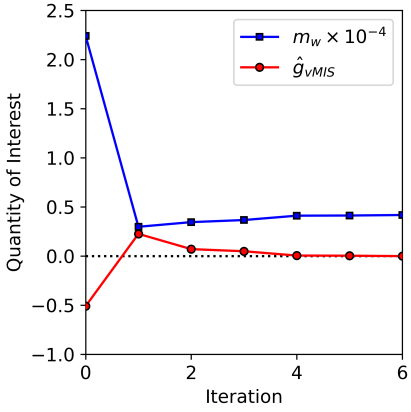
(b) Run #1. Disciplinary solver calls.



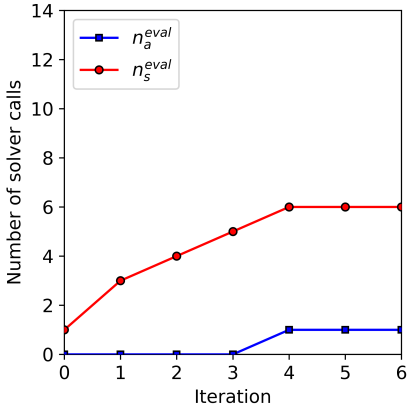
(c) Run #2. Quantities of interest.



(d) Run #2. Disciplinary solver calls.



(e) Run #3. Quantities of interest.



(f) Run #3. Disciplinary solver calls.

Figure 4.6: Illustration of the progression of the gradient-based optimization using the derivatives of the disciplinary surrogates, for the three runs performed. The objective (blue) and constraint (red) function values are shown on the left while the number of aerodynamics (blue) and structural (red) disciplinary solver calls are shown on the right. We remark the number of iterations for run #2, which was significantly greater than that made in the remaining runs.

corresponding objective and constraint function values, as well as the number of disciplinary solver calls made when solving the structural sizing problem for three different starting points.

x^0	m_w^0 [kg]	AoA*	t_r^*	t_{sk}^*	t_{sp}^*	m_w^* [kg]	$\hat{g}_{vMIS}^*$	$\hat{h}_{L=2.5W}^*$	n_s^{eval}	n_a^{eval}
#1	16396.6	0.2930	0.0	0.1816	0.2356	5117.6	$\approx 10^{-5}$	$\approx 10^{-5}$	12	7
#2	8651.1	0.2928	0.0	0.1882	0.2043	5103.2	$\approx 10^{-3}$	$\approx 10^{-4}$	13	6
#3	22402.4	0.2928	0.0	0.1979	0.1706	5122.4	$\approx 10^{-3}$	$\approx 10^{-4}$	10	6
								Training of the disciplinary surrogates	40	40
								Total solver calls	75	59

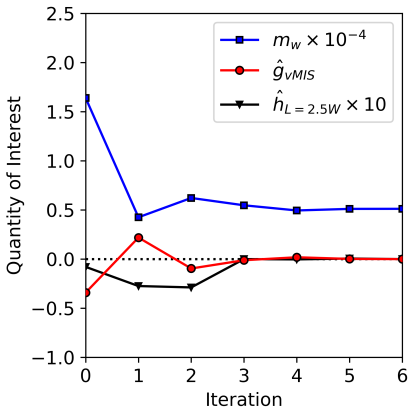
Table 4.11: Structural sizing problem at pull-up maneuver condition. Optima found and number of disciplinary solver calls obtained when solving the MDO using the disciplinary surrogates. Problem ran from three different starting points, with chosen optimizer the gradient based SLSQP and convergence criterion set to 10^{-3}.

The results of Table 4.11 show that, despite the simplification made on the derivative of the aerodynamics disciplinary surrogate, the optimizer successfully converges to a feasible solution for all tested starting points. Moreover, the optima found are similar, once more suggesting that there is only one local optimum and that the small differences occur due to the quality of the surrogate approximations. Finally, we see that solving the MDO problem using three different starting points required a total of 75 calls to the structural solver and 59 calls to the aerodynamics solver. On our machine, equipped with an Intel Xeon CPU E5-2650 v4 @ 2.20 GHz core and 128 GB of memory, this translated to around 16 hours of computations.

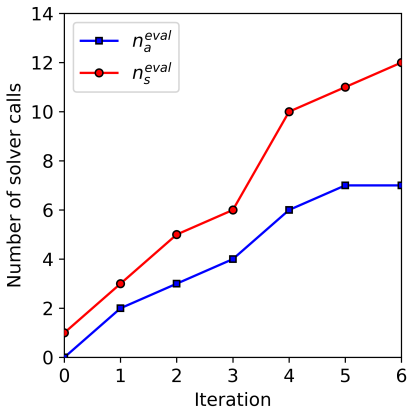
Progression of the optimizer. To illustrate the optimization process, we show in Figure 4.7 the objective and constraint function values, as well as the total number of disciplinary solver calls, at each iteration of the three runs performed. As for the mass minimization problem at fixed angle of attack, we see that fewer disciplinary solver calls are made during the last few iterations, as the disciplinary surrogates have become sufficiently accurate around the local optima. We also note that, although nearly twice as many enrichments are made for the structural discipline than for the aerodynamics discipline, this was not detrimental for the considered problem, as structural solver calls are significantly less expensive than aerodynamic solver calls.

Cost comparison. To estimate how much computational cost is saved by using the disciplinary surrogates, we let the optimization run using the real disciplinary solvers for 120 hours (five days), leading to over 600 solver calls to each discipline, 8 times more than the total number of solver calls made when using the disciplinary surrogates. Unfortunately, the optimization process was terminated before convergence, but the best point found lied in the neighborhood of the optima of Table 4.11 and presented a wing mass of around 5110kg. This suggests that the results obtained using the disciplinary surrogates are close to the true optimum.

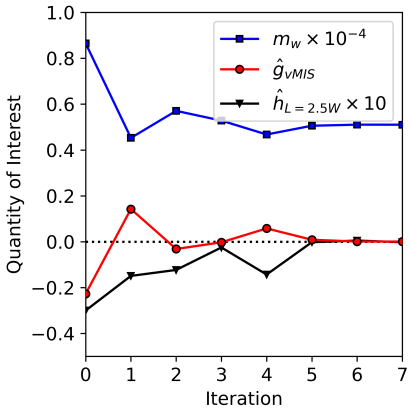
Quality of the computed derivatives. To assess the impact of the proposed simplification on the derivative of the aerodynamics surrogate with respect to variations in the angle of attack, we first compare the derivatives of the constraint functions obtained via the proposed simplification and via finite-difference approximation at the three solution points of Table 4.11. The obtained results are presented on Table 4.12. When using finite-difference approximation, the computed derivatives take into account the variation of the local POD basis with variations in the angle of attack. Contrarily, the proposed simplification assumes that the local POD basis



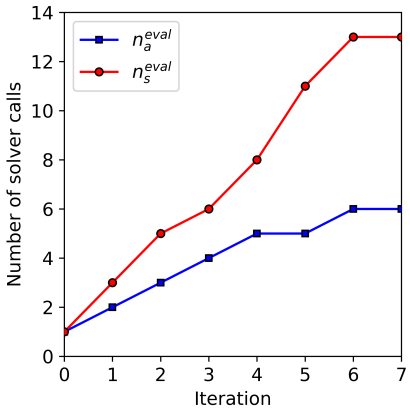
(a) Run #1. Quantities of interest.



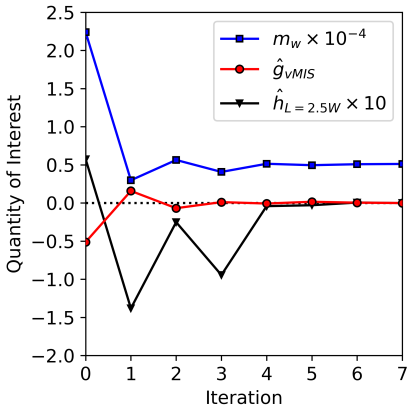
(b) Run #1. Disciplinary solver calls.



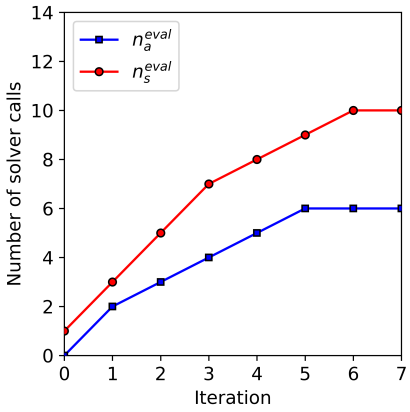
(c) Run #2. Quantities of interest.



(d) Run #2. Disciplinary solver calls.



(e) Run #3. Quantities of interest.



(f) Run #3. Disciplinary solver calls.

Figure 4.7: Optimization results for the structural sizing problem, for the three runs performed. The objective and constraint function values are shown on the left while the number of disciplinary solver calls are shown on the right.

	Finite-difference		Proposed simplification			
$\boldsymbol{x}^*$	$\frac{\mathrm{d}g^*_{\mathrm{vMIS}}}{\mathrm{dAoA}}$	$\frac{\mathrm{d}h^*_{L=2.5W}}{\mathrm{dAoA}}$	$\frac{\mathrm{d}\hat{g}^*_{\mathrm{vMIS}}}{\mathrm{dAoA}}$	$\frac{\mathrm{d}\hat{h}^*_{L=2.5W}}{\mathrm{dAoA}}$	$\epsilon_{\mathrm{rel}}\left(\frac{\mathrm{d}\hat{g}^*_{\mathrm{vMIS}}}{\mathrm{dAoA}}\right)$	$\epsilon_{\mathrm{rel}}\left(\frac{\mathrm{d}\hat{h}^*_{L=2.5W}}{\mathrm{dAoA}}\right)$
#1	1.6202	1.4449	1.6675	1.4441	2.92%	0.06%
#2	1.5814	1.4444	1.5924	1.4463	0.70%	0.13%
#3	1.5714	1.4460	1.5806	1.4487	0.59%	0.19%

Table 4.12: Total derivatives of the aerodynamics surrogate with respect to variations in the angle of attack at three different design space points. Comparison between the finite-difference approximation and proposed simplification.

remains constant in a small neighborhood of the queried point. As a result, a small error is committed on the total derivative of the constraint functions with respect to the angle of attack. Still, the computed relative error remains small, having obtained under 5% of relative error for the inequality constraint function and under of relative error 1% for the equality constraint function, for all considered points. These results confirm that the proposed simplification is a reasonable one. In Table 4.13 we now compare the derivatives of the inequality constraint function for the real disciplinary solvers (computed via finite-difference approximation) with the approximated derivatives at the same three solution points of Table 4.11.

	Real solvers				Disciplinary GPs				
$\boldsymbol{x}^*$	$\frac{\mathrm{d}g^*_{\mathrm{vMIS}}}{\mathrm{dAoA}}$	$\frac{\mathrm{d}g^*_{\mathrm{vMIS}}}{\mathrm{d}t_{\mathrm{r}}}$	$\frac{\mathrm{d}g^*_{\mathrm{vMIS}}}{\mathrm{d}t_{\mathrm{sk}}}$	$\frac{\mathrm{d}g^*_{\mathrm{vMIS}}}{\mathrm{d}t_{\mathrm{sp}}}$	$\frac{\mathrm{d}\hat{g}^*_{\mathrm{vMIS}}}{\mathrm{dAoA}}$	$\frac{\mathrm{d}\hat{g}^*_{\mathrm{vMIS}}}{\mathrm{d}t_{\mathrm{r}}}$	$\frac{\mathrm{d}\hat{g}^*_{\mathrm{vMIS}}}{\mathrm{d}t_{\mathrm{sk}}}$	$\frac{\mathrm{d}\hat{g}^*_{\mathrm{vMIS}}}{\mathrm{d}t_{\mathrm{sp}}}$	$\epsilon_{\mathrm{rel}}\left(\frac{\mathrm{d}\hat{g}^*_{\mathrm{vMIS}}}{\mathrm{d}\boldsymbol{x}}\right)$
#1	1.6896	-0.4392	-0.9762	-0.2222	1.6675	-0.2449	-1.0028	-0.2455	9.87%
#2	1.7052	-0.4407	-0.9489	-0.2215	1.5924	-0.1936	-1.0047	-0.2670	14.0%
#3	1.4688	-0.4540	-1.0925	-0.2488	1.5806	-0.1881	-0.9792	-0.2830	16.4%

Table 4.13: Total derivatives of the inequality constraint function with respect to the design variables at the three solution points found for the mass minimization problem at fixed angle of attack. Comparison between the true and approximated MDAs.

The obtained results show that the order of magnitude of the committed relative error on the inequality constraint function derivatives is similar to that obtained when the angle of attack was fixed, despite the simplifications made. Moreover, we see that the greatest error is committed for the total derivative with respect to the thickness of the ribs (t_{r}). Once more, this can be justified by the fact that the optima found lied on the edge of the design space for this design variable.

4.4 Using the proposed approach via WhatsOpt

An important objective envisioned for the developments presented in this chapter were their easy implementation and compatibility with any disciplinary solvers. To that end, the proposed approach has been integrated within ONERA's WhatsOpt collaborative environment, in collaboration with Sylvain Dubreuil and Rémi Lafage. The details on how to use disciplinary surrogates for gradient-based optimization via WhatsOpt are given on the Appendix.

4.5 Summary of disciplinary GPs for gradient-based MDO

In this chapter we proposed to use the derivatives of the disciplinary surrogates to perform gradient-based MDO, under the assumption that, as the disciplinary surrogates are enriched,

the uncertainty of the corresponding derivatives with respect to the design variables is reduced. We then used the disciplinary surrogate derivatives to solve both analytical and engineering MDO problems. The obtained results confirmed the interest of the proposed approach in terms of computational cost. In terms of accuracy of the obtained solution we observed that the optima found using the disciplinary surrogates often lie close to the true solution, but do not exactly match it. This suggests that, despite having attained the converge criterion on the coefficient of variation of the random MDA solution, some uncertainty may still be present in the computed derivatives. To improve this result, future work could focus on the development of a surrogate enrichment strategy which aims at learning the derivative of the disciplinary solvers with respect the design variables, rather than learning solution of the MDA itself.

Finally, we note that although in this work we assume that the disciplinary solvers behave as black-boxes, the proposed approach still has value in other contexts. Indeed, even if the derivatives of the real disciplinary solvers are available, the use of disciplinary surrogates derivatives remains an interesting option to quickly explore the design space, via a multi-start strategy. To refine the obtained solution, an optimization using the real solvers may then be run using as starting point the best point found by the disciplinary surrogates.

Chapter summary:

In this chapter we proposed to use the derivatives of the disciplinary surrogates to perform gradient-based MDO. To that end we discussed how to obtain the surrogate derivatives when:

- *The coupling variables are scalars;*
- *The coupling variables are vectors whose dimension is reduced via a global POD basis or via the interpolation of a database of pointwise local POD bases.*

We then applied the proposed strategy to solve both an analytical and engineering MDO problems. The obtained results showed that:

- *Using the disciplinary surrogates allows to save a significant amount of computational cost when the derivatives of the real solver are not available;*
- *The approximate MDO problem often converges to a point that lies close to the true optimum but does not exactly match it. This solution could be refined by solving the real MDO problem with starting point the best solution found using the disciplinary surrogates.*

Finally, we presented the implementation of the proposed approach in the WhatsOpt collaborative environment (see Appendix). This implementation allows users to employ the proposed approach with minimal implementation effort.

CHAPTER 5

Disciplinary GPs for constrained global optimization

Contents

The developments presented in this chapter led to the following article, published in the Structural and Multidisciplinary Optimization journal in February 2024:

> I. Cardoso, S. Dubreuil, N. Bartoli, C. Gogu, M. Salaün, *Constrained efficient global multidisciplinary design optimization using adaptive disciplinary surrogate enrichment*, Structural and Multidisciplinary Optimization. 10.1007/s00158-023-03736-0

The work was equally presented at the WCSMO conference in June 2023.

In this chapter we will:

- *Extend the EGMDO algorithm to constrained problems.*
- *Compare the proposed constrained optimization algorithm against existing approaches.*
- *Apply the constrained EGMDO algorithm to both analytical and engineering test cases.*

5.1 Chapter contributions

In Chapter 4 we proposed to use the derivatives of the disciplinary GPs to perform gradient-based MDO, which resulted in important cost reductions when dealing with black-box disciplinary solvers. However, to find the global optimum, gradient-based optimizers need to resort to multi-start strategies, where the optimization is run several times from different starting points. For complex functions, with multiple local minima, these strategies can represent a heavy computational cost. An alternative strategy is thus to combine the built disciplinary surrogates with a Bayesian optimization algorithm. This is exactly the strategy proposed in the EGMDO algorithm, introduced in Section 1.2.5.3. The interest of the EGMDO has been previously confirmed for unconstrained problems in [Dubreuil et al., 2020] and was shown to perform well in combination with the DPOD+I approach in [Berthelin, 2022], once more in the context of unconstrained optimization. Many industrial problems, however, are subject to constraints, as their inclusion in the optimization problem allows to ensure that a physically relevant solution is attained. In this chapter we extend the existing EGMDO framework to be able to handle both equality and inequality constraints.

5.2 Illustrative 1-D constrained MDO problem

We recall that a generic constrained optimization problem can be defined as:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathscr{X}} \; & f_{\text{obj}}(\boldsymbol{x}) \\ \text{s.t.} \; & h_i(\boldsymbol{x}) = 0, \; i = 1,...,n_h \\ & g_j(\boldsymbol{x}) \leq 0, \; j = 1,...,n_g \end{aligned} \tag{5.1}$$

where $\boldsymbol{h}_i : \mathbb{R}^n \mapsto \mathbb{R}$ denotes the i^{th} equality constraint and $\boldsymbol{g}_j : \mathbb{R}^n \mapsto \mathbb{R}$ denotes the j^{th} inequality constraint. In the following, we propose a constraint handling strategy for the EGMDO algorithm, which allows to solve constrained MDO problems, such as the one defined in Eq. (5.1). To illustrate the proposed developments, we consider a 1-D constrained MDO problem defined as:

$$\begin{aligned} \arg\min_{x\in\mathscr{X}} \quad & \cos\left(\frac{y_1^*(x) + \exp(-y_2^*(x))}{\pi}\right) + \frac{x}{20} \\ \text{s.t.} \quad & g_1(x) \geq 0 \end{aligned} \tag{5.2}$$

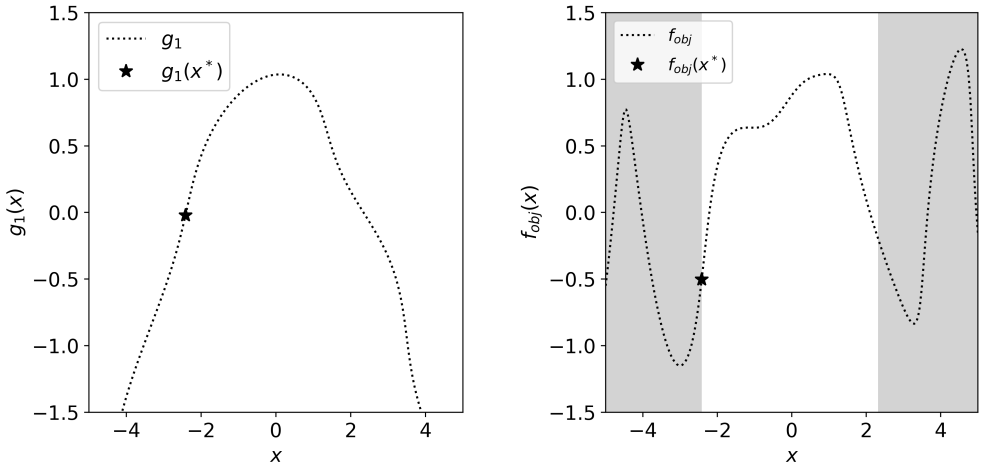
where $\boldsymbol{y}^*(x) = \{y_1^*(x), y_2^*(x)\}$ is the solution of the following non-linear system of equations:

$$\begin{cases} y_1(x, y_2) = x^2 - \cos(\frac{y_2}{2}) \\ y_2(x, y_1) = x + y_1 \end{cases} \tag{5.3}$$

and:

$$g_1(x) = -0.15 y_1^*(x) + 0.9 \tag{5.4}$$

is an inequality constraint which depends on the solution of the MDA. Figure 5.1(a) plots the constraint g_1 as a function of x, while Figure 5.1(b) displays the obtained infeasible region, after imposing $g_1 \geq 0$. As is shown, due to the imposed constraint, the global optimum is found for $x^* \approx -2.42$.



(a) Inequality constraint $g_1(x)$.

(b) Infeasible region.

Figure 5.1: Inequality constraint and resulting infeasible region for the 1-D illustrative MDO problem. (a) Inequality constraint $g_1(x)$. (b) Objective function $f_{\text{obj}}(x)$ with the shaded infeasible region. The global minimum of the constrained problem is identified with a black star.

5.3 Random constraint function

When constraints are present, Bayesian optimization approaches often build approximate models of the constraint functions which are then used to limit the maximization of the acquisition function. In other words, these models ensure that any point proposed by the acquisition function is both feasible (according to the current constraint approximation) and provides an improvement with respect to the current best point. The purpose of this section is to describe how continuous models of the constraint functions may be obtained when the disciplinary solvers are replaced by GPs. To that end we recall that, when replacing the disciplinary solvers by their corresponding GP approximations, all functions which depend on any subset of the converged coupling variables become themselves random variables, of non-Gaussian distribution. For these variables, uncertainty quantification by PCE may be performed (see Eq. (1.73)), resulting in the following set of random variables, obtained for all $\boldsymbol{x} \in \text{DoE}_{\text{UQ}}$:

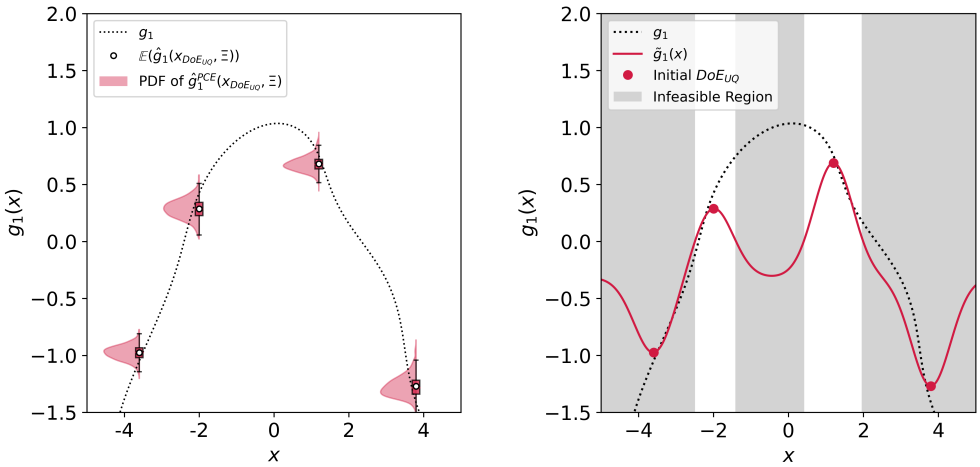
$$\widehat{\text{QoI}}(\boldsymbol{x}, \boldsymbol{\Xi}) = \{\hat{f}_{\text{obj}}^{\text{PCE}}(\boldsymbol{x}, \boldsymbol{\Xi}),\ \hat{g}_m^{\text{PCE}}(\boldsymbol{x}, \boldsymbol{\Xi}),\ \hat{h}_r^{\text{PCE}}(\boldsymbol{x}, \boldsymbol{\Xi})\},\ m = 1, ..., n_g,\ r = 1, ...n_h \tag{5.5}$$

For the objective function, the PCE approximation is extended to the remainder of the design space according to the KL-GP interpolation of Eq. (1.76). This approach results in an objective function model which allows to take into account the uncertainty due to the use of disciplinary GPs during the maximization of the acquisition function. For the constraint functions, however,

a simplification is made by directly interpolating the mean value of the respective PCE approximations. This approach is similar to the one used in the Super Efficient Global Optimization (SEGO) algorithm [Sasena et al., 2002], where only the mean value $\mu_{\tilde{g}}$ and $\mu_{\tilde{h}}$ of the GP interpolation of the constraint functions is used in the maximization of the acquisition function.

Other, less conservative approaches could also be used. For instance, in [Priem et al., 2020a] it is shown that using only the mean values of the GPs can lead to an increased infeasible region and, consequently, prevent the exploration of interesting regions of the design space. As an alternative, the authors of [Priem et al., 2020a] propose the use of upper trust bounds, where the uncertainty of the constraint approximations is used to obtain a more relaxed feasible domain. The idea is that this relaxation is more important during the first iterations of the algorithm, when the uncertainty is high. Then, as the knowledge of the constraint functions is improved, the relaxation is reduced and the algorithm converges to the constrained global optimum.

For the remainder of this section we will consider that a GP interpolation of the mean value of the PCE approximation is used to model the constraint functions. However, during the numerical tests of Section 5.5, the use of a more relaxed criterion will be studied. Figure 5.2 illustrates the obtained PCE and GP approximations of $g_1(x)$.



(a) $\hat{g}_1^{\text{PCE}}(x, \Xi)$ at DoE_{UQ} points.

(b) GP approximation $\tilde{g}_1(x)$.

Figure 5.2: Constraint function approximations. (a) Obtained PCE approximation for the constraint function $g_1(x)$ at four different points. (b) GP approximation of the constraint function $g_1(x)$ and resulting infeasible region.

Figure 5.2 shows that, in light of the current GP approximation, the obtained infeasible region does not correspond to the real one (see Figure 5.1(b)). It will be shown in the following that, as we reduce the uncertainty of both objective and constraint functions, the approximated infeasible region begins to resemble the real one.

5.4 Adapted uncertainty reduction strategy

In EGMDO, the proposed uncertainty reduction strategy allows us to spend most of the computational effort in relevant zones of the design space with respect to the minimization of the objective function. This is possible thanks to the use of enrichment criteria. In this section we describe how these enrichment criteria can be adapted to handle constrained problems.

5.4.1 Constrained EI

The first step of the uncertainty reduction strategy involves the maximization of an acquisition function, which is used in the adaptive sampling of the design space by determining the new points to add to DoE_{UQ}. In the original EGMDO framework the chosen acquisition function is a modified EI criterion [Dubreuil et al., 2018], inspired by the one proposed in the EGO algorithm. Here, we extend this criterion to constrained problems by defining a Constrained Expected Improvement (CEI) criterion as follows:

$$\text{CEI}(\boldsymbol{x}) = \mathbb{E}\left[\left(\hat{f}'_{\min}(\boldsymbol{\Xi}) - \tilde{f}_{\text{obj}}(\boldsymbol{x},\boldsymbol{\Xi},\boldsymbol{\eta})\right) \times \mathbb{1}_{\tilde{f}_{\text{obj}}(\boldsymbol{x},\boldsymbol{\Xi},\boldsymbol{\eta}) \leq \hat{f}'_{\min}(\boldsymbol{\Xi})}\right] \tag{5.6}$$

where $\hat{f}'_{\min}(\boldsymbol{\Xi})$ is the current minimum, given by:

$$\hat{f}'_{\min}(\boldsymbol{\Xi}) = \min_{\boldsymbol{x}\in \text{DoE}_{\text{UQ}}} \hat{f}^{\text{PCE}}_{\text{obj}}(\boldsymbol{\Xi}) \mid_{\hat{g}^{\text{PCE}}_m(\boldsymbol{\Xi})\geq 0,\, \hat{h}^{\text{PCE}}_r(\boldsymbol{\Xi})=0},\ m = 1,...,n_g,\ r = 1,...,n_h \tag{5.7}$$

$$\mathbb{1}_{\tilde{f}_{\text{obj}}(,\boldsymbol{\Xi},\boldsymbol{\eta}) \leq \hat{f}_{\min}(\boldsymbol{\Xi})} = \begin{cases} 0 & \text{if } \tilde{f}_{\text{obj}}(\boldsymbol{x},\boldsymbol{\Xi},\boldsymbol{\eta}) > \hat{f}_{\min}(\boldsymbol{\Xi}) \\ 1 & \text{if } \tilde{f}_{\text{obj}}(\boldsymbol{x},\boldsymbol{\Xi},\boldsymbol{\eta}) \leq \hat{f}_{\min}(\boldsymbol{\Xi}) \end{cases} \tag{5.8}$$

We note that the expressions above imply that, for each $\boldsymbol{\Xi}^{(k)} \in \boldsymbol{\Xi}$, the minimum will be chosen among feasible points. However, a scenario might occur where, for all $\boldsymbol{\Xi}^{(k)} \in \boldsymbol{\Xi}$, no point respects the constraints. This is often linked to the presence of equality constraints. When this is the case, the chosen $\hat{f}'_{\min}(\boldsymbol{\Xi})$ is that which violates the least the constraints for each $\boldsymbol{\Xi}^{(k)} \in \boldsymbol{\Xi}$. The constraint violation $\theta(\boldsymbol{\Xi}^{(k)})$ is calculated for all $\boldsymbol{x} \in \text{DoE}_{\text{UQ}}$ as follows:

$$\theta(\boldsymbol{\Xi}^{(k)}) = \sum_{m=1}^{n_g} \mid \min(\hat{g}^{\text{PCE}}_m(\boldsymbol{\Xi}^{(k)}), 0) \mid + \sum_{r=1}^{n_h} \mid \hat{h}^{\text{PCE}}_r(\boldsymbol{\Xi}^{(k)}) \mid \tag{5.9}$$

Once $\hat{f}'_{\min}(\boldsymbol{\Xi})$ has been determined, the new point to add to DoE_{UQ} is found by solving the following optimization problem:

$$\begin{aligned} \arg\max_{\boldsymbol{x}\in\mathcal{X}} \quad & \text{CEI}(\boldsymbol{x}) \\ \text{s.t.} \quad & \mu_{\tilde{g}_m}(\boldsymbol{x}) \geq 0,\ \ m = 1,...,n_g \\ & \mu_{\tilde{h}_r}(\boldsymbol{x}) = 0,\ \ r = 1,...,n_h \end{aligned} \tag{5.10}$$

where $\mu_{\tilde{g}_m}(\boldsymbol{x})$ and $\mu_{\tilde{h}_r}(\boldsymbol{x})$ are the mean values of the constraint GP approximations. At this new point, uncertainty quantification by PCE is performed and new approximations for the objective and constraint functions are obtained.

Remark 5.1 *The optimization of the acquisition function is performed using a gradient based solver, combined with a multi-start strategy. During our tests, a point that already exists in* DoE_{UQ} *is sometimes proposed, leading to numerical issues in the KL-GP interpolation step. To prevent these numerical issues, when a point that already exists in* DoE_{UQ} *is proposed, we add the next best point, found via the multi-start strategy. In practice, to determine if the point already exists in* DoE_{UQ}*, the euclidean distance between the new point and the remaining points is calculated. If this distance is lower than a defined threshold, the point is considered to already exist in the DoE.*

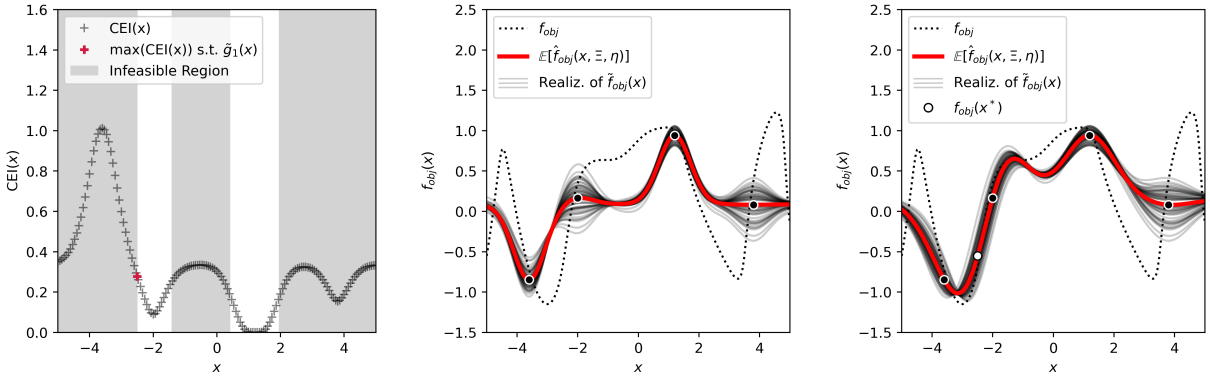
Figure 5.3 illustrates the obtained CEI criterion as a function of $\boldsymbol{x}$ as well as the resulting KL-GP approximation of $\tilde{f}_{\text{obj}}(\boldsymbol{x},\boldsymbol{\Xi},\boldsymbol{\eta})$ after the new point is added to DoE_{UQ}. By solving Eq. (5.10), a point that lies on the edge of the approximated feasible region is proposed. This is a relevant

point for the constrained problem, as it promotes learning the constraint function where it is likely to be active. Moreover, we see that by adding this point to DoE_{UQ}, the objective function value in the same region becomes closer to its true value. However, the uncertainty induced by the use of disciplinary GPs remains unchanged, as no points have been added to the disciplinary DoEs. This uncertainty will be reduced with the aid of the second step of the uncertainty reduction strategy, presented in the next section.

Finally, we recall that, while the EI is the only acquisition function used in the original EGMDO algorithm, other acquisition functions exist in the literature. For the purpose of this work we will consider one other, namely, the WB2 [Watson and Barnes, 1995] acquisition function. It is obtained by subtracting the mean value of the objective function model from the CEI, as follows:

$$\text{WB2}(\boldsymbol{x}) = \text{CEI}(\boldsymbol{x}) - \tilde{\boldsymbol{\mu}}_{\tilde{f}_{\text{obj}}}(\boldsymbol{x}, \boldsymbol{\Xi}, \eta) \tag{5.11}$$

As shown in Section 1.1.2.4, the WB2 acquisition function is less multimodal than the EI function, making it easier to maximize. In the numerical tests of Section 5.5, the use of the WB2 criterion in the constrained EGMDO algorithm will be considered and its performance will be compared against that of the CEI criterion.



(a) CEI criterion. (b) Initial approximation. (c) New approximation.

Figure 5.3: Enrichment of DoE_{UQ} at the points proposed by the acquisition function. (a) CEI acquisition function and corresponding constrained maximum, found for $x \approx -2.5$. The shaded gray area represents the approximated infeasible region. (b) Initial objective function approximation. (c) Objective function approximation after adding the new point to DoE_{UQ}. Realizations of $\tilde{f}_{\text{obj}}(\boldsymbol{x}, \boldsymbol{\Xi}, \eta)$ are drawn to illustrate the corresponding uncertainty.

5.4.2 Constrained enrichment of the disciplinary GPs

The second criterion used in the uncertainty reduction strategy involves adding new points to the disciplinary DoEs. In EGMDO, the point selected for disciplinary solver enrichment is the point among all DoE_{UQ} points which is the most likely to be a solution to the unconstrained problem. When extending this criterion to a constrained problem, this is equivalent to saying that we will enrich the point which is most likely of being the minimum, among feasible points. This likelihood is given by the probability P_{sol} defined as follows:

$$\text{P}_{\text{sol}}(\boldsymbol{x}_i) = \mathbb{P}\left(f'_{\min}(\boldsymbol{\Xi}) = \hat{f}^{\text{PCE}}_{\text{obj}}(\boldsymbol{x}_i, \boldsymbol{\Xi})\right) \; \forall \, \boldsymbol{x}_i \in \text{DoE}_{\text{UQ}} \,, \tag{5.12}$$

where $f'_{\min}(\boldsymbol{\Xi})$ is the discrete random variable defined in Eq. (5.7). Note that, for different drawn samples $\boldsymbol{\Xi}^{(k)}$ of $\boldsymbol{\Xi}$, the same point $\boldsymbol{x}_i$ may be feasible or unfeasible. As a consequence, P_{sol} takes

into account not only the uncertainty of the objective function approximation, but also the uncertainty of the constraint approximations, due the use of disciplinary GPs.

In addition to presenting a high likelihood of being a solution to the constrained problem, a second condition must be met for the disciplinary surrogates to be enriched at a given design space point. Namely, the Coefficient of Variation (CV) of at least one quantity of interest $\widehat{\text{QoI}}$ (see Eq. (5.5)) must be greater than a given threshold ϵ_{CV}. For any quantity $\widehat{\text{QoI}}_j$ in $\widehat{\text{QoI}}$, this coefficient of variation is given by:

$$\text{CV}(\widehat{\text{QoI}}_j(\boldsymbol{x}_i, \boldsymbol{\Xi})) = \frac{\sqrt{\mathbb{V}(\widehat{\text{QoI}}_j(\boldsymbol{x}_i, \boldsymbol{\Xi}))}}{\mathbb{E}(\widehat{\text{QoI}}_j(\boldsymbol{x}_i, \boldsymbol{\Xi}))}, \tag{5.13}$$

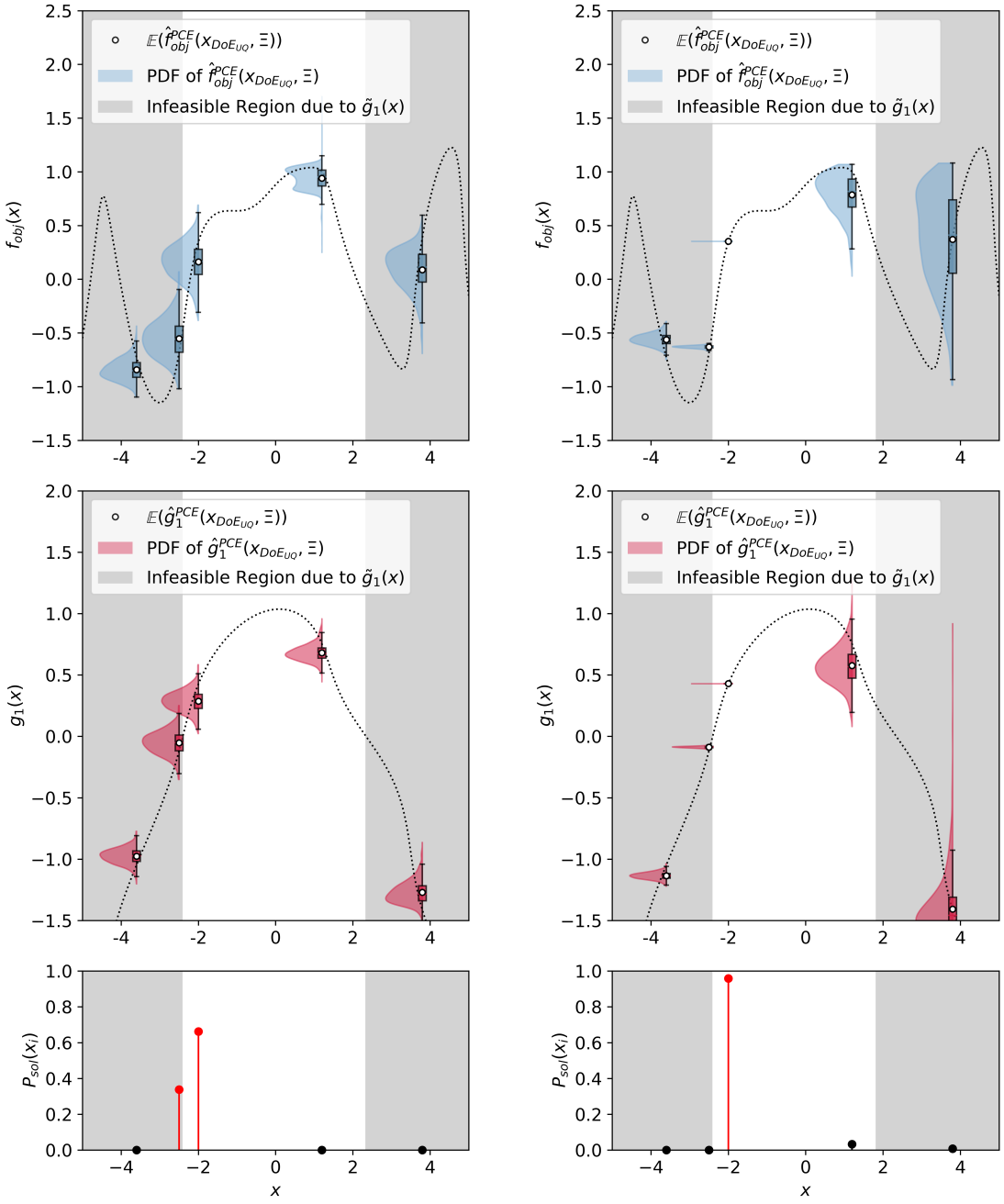
where $\mathbb{E}$ is the expected value and $\mathbb{V}$ is the variance. If both conditions are met, the disciplinary solvers are enriched until the CV of both objective and constraint functions are sufficiently small for all points $\boldsymbol{x} \in \text{DoE}_{\text{UQ}}$ that verify $\text{P}_{\text{sol}}(\boldsymbol{x}_i) \geq \epsilon_{\text{P}}$, where ϵ_{P} is some defined threshold.

Some remarks should, nonetheless, be made concerning equality constraints for this uncertainty reduction step. When equality constraints are present, it is likely that all DoE_{UQ} points will present values of $\text{P}_{\text{sol}}(\boldsymbol{x}_i)$ that are very close to zero, especially during the first few iterations of the algorithm. When this is the case, we calculate the constraint violation of Eq. (5.9) for all points $\boldsymbol{x}_i \in \text{DoE}_{\text{UQ}}$ and we enrich the disciplinary solvers for the point that is most likely to minimize $\theta(\boldsymbol{\Xi})$ until the CV of all $\widehat{\text{QoI}}$ is sufficiently small. This strategy prioritizes increasing our knowledge of the equality constraint functions, before seeking to minimize the objective function. Since equality constraints are always active constraints, we considered that this option is not detrimental to the overall computational cost of the algorithm.

Figure 5.4 illustrates the PCE approximations of the objective and constraint functions of the 1-D constrained problem, before and after enriching the disciplinary GPs. The obtained $\text{P}_{\text{sol}}(\boldsymbol{x}_i)$ criterion for the initial and final iterations is equally plotted. As is shown, in the initial iteration (see Figure 5.4(a)), both points $x = -2.5$ and $x = -2$ have a high value of P_{sol}. This is because point $x = -2.5$ is more likely to minimize the objective function, but point $x = -2$ is more likely to respect the constraint. Contrarily, once the uncertainties have been reduced (see Figure 5.4(b)), we become sure that point $x = -2.5$ does not respect the constraint and, as a consequence, only point $x = -2$ presents a high value of P_{sol}. Algorithm 11 summarizes the proposed approach for the reduction of the uncertainty associated with the random variable $\boldsymbol{\Xi}$, for the constrained problem.

Note that the proposed strategy to handle constraint functions is easily integrated within the original EGMDO formulation defined in Algorithm 7. Indeed, Algorithm 11 corresponds to the step "Enrich disciplinary GPs if needed" of Algorithm 7, while the maximization of the CEI criterion subject to the current approximation of the constraint functions, corresponds to the step "Find $\boldsymbol{x}_{\text{new}}$" of the same algorithm. Finally, we remark that while Algorithm 11 concludes the proposed developments in order to solve constrained problems when the disciplinary solvers are replaced by GPs, it does not explicitly state how to handle constraint functions which do not depend on the coupling variable values. The following paragraph briefly addresses this question.

Constraint functions which do not depend on the coupling variable values. Some MDO problems deal with constraint functions which do not depend on the coupling variable values. To account for this type of constraint functions in the proposed uncertainty reduction strategy, we distinguish between two situations, depending on how costly the constraint functions are to evaluate. If we can consider that they are free to evaluate, the real constraint function can be used directly in the maximization of the CEI criterion. Otherwise, if the constraint is costly to evaluate, a surrogate may be required. This last can be built over the initial DoE_{UQ}, and en-



(a) Iteration 0.

(b) Iteration 4.

Figure 5.4: Enrichment of the disciplinary GPs at the points that maximize the P_{sol} criterion. (a) Initial iteration, before any enrichment is performed. (b) Last iteration, after all points which verify $P_{sol} > \epsilon_P$ have been enriched until a sufficiently small CV is obtained for both objective and constraint functions. The threshold ϵ_P was set to 0.1, while the threshold for the coefficient of variation was $\epsilon_{CV} = 0.001$. The blue PDFs refer to pointwise PCE approximations of the random objective function while the red PDFs refer to the pointwise PCE approximations of the random constraint function. The shaded gray areas represent the approximated infeasible region.

Algorithm 11 Constrained-EGMDO: reduction of the uncertainty due to the random variable $\boldsymbol{\Xi}$

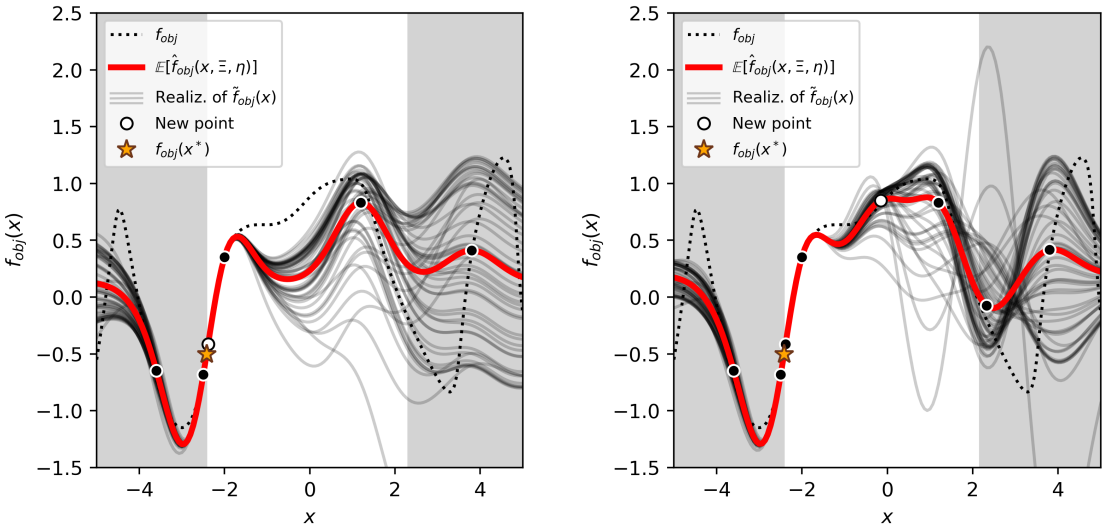
Input: $\epsilon_{\text{CV}}, \epsilon_{\text{P}}$ ▷ Threshold for CV and P_{sol}
  Compute $\widehat{\text{QoI}}(\boldsymbol{x}_i, \boldsymbol{\Xi})$, $\forall \boldsymbol{x}_i \in \text{DoE}_{\text{UQ}}$ ▷ PCE approximations (Eq. (5.5))
  Compute $\text{P}_{\text{sol}}(\boldsymbol{x}_i)$, $\forall \boldsymbol{x}_i \in \text{DoE}_{\text{UQ}}$ ▷ Probability of being solution (Eq. (5.12))
  if $\text{P}_{\text{sol}}(\boldsymbol{x}_i) < \epsilon_{\text{P}}$, $\forall \boldsymbol{x} \in \text{DoE}_{\text{UQ}}$ **then**
    $\text{P}_{\text{sol}}(\boldsymbol{x}_i) = \mathbb{P}\left(\min \boldsymbol{\theta}(\boldsymbol{\Xi}) = \boldsymbol{\theta}(\boldsymbol{x}_i, \boldsymbol{\Xi})\right)$, $\forall \boldsymbol{x}_i \in \text{DoE}_{\text{UQ}}$ ▷ Minimum constraint violation
  end if
  Compute $\text{CV}(\widehat{\text{QoI}}(\boldsymbol{x}_i, \boldsymbol{\Xi}))$, $\forall \boldsymbol{x}_i \in \text{DoE}_{\text{UQ}}$ ▷ Coefficient of variation (Eq. (5.13))
  while $\exists \boldsymbol{x}_i \in \text{DoE}_{\text{UQ}}$ such that $\text{P}_{\text{sol}}(\boldsymbol{x}_i) \geq \epsilon_{\text{P}}$ **and** $\text{CV}(\boldsymbol{x}_i) \geq \epsilon_{\text{CV}}$ **do**
    $\mathbf{X}_{\text{enrich}} = \{\boldsymbol{x}_i \mid \text{P}_{\text{sol}}(\boldsymbol{x}_i) \geq \epsilon_{\text{P}}\}$
    Sort $\mathbf{X}_{\text{enrich}}$ in decreasing order with respect to $\text{P}_{\text{sol}}(\boldsymbol{x}_i)$
    Set $k = 0$ and pts_added = False
    while $k \leq \#(\mathbf{X}_{\text{enrich}})$ **and** pts_added == False **do**
      $\boldsymbol{x} = \mathbf{X}_{\text{enrich}}[k]$ ▷ Loop through candidate points
      if $\text{CV}(\boldsymbol{x}) > \epsilon_{\text{CV}}$ **then**
        Enrich disciplinary surrogates
        Recompute $\widehat{\text{QoI}}(\boldsymbol{x}_i, \boldsymbol{\Xi})$, $\text{P}_{\text{sol}}(\boldsymbol{x}_i)$, $\text{CV}(\widehat{\text{QoI}}(\boldsymbol{x}_i, \boldsymbol{\Xi}))$
        pts_added == True
      else
        $k = k + 1$
      end if
    end while
  end while

riched at points proposed by the CEI criterion. Whichever the case, these functions will never depend on the random variable $\boldsymbol{\Xi}$. Therefore, during the disciplinary surrogate enrichments, the probability of violating these constraints, at any $\boldsymbol{x}_i \in \text{DoE}_{\text{UQ}}$ is either zero or one, for all $\boldsymbol{\Xi}^{(k)} \in \boldsymbol{\Xi}$. Similarly, their contribution to the constraint violation $\boldsymbol{\theta}(\boldsymbol{\Xi})$ is the same for all $\boldsymbol{\Xi}^{(k)} \in \boldsymbol{\Xi}$. With this in mind, it is considered that the implementations proposed in this section are sufficient to handle all types of constraint functions, including those that do not depend on the coupling variable values.

5.4.3 Stopping criterion

The previous section showed how the accuracy of both objective and constraint function approximations can be improved both by reducing the uncertainty due to the disciplinary GPs and by adding new points to DoE_{UQ}. By iteratively performing these operations, the precision of the approximations increases around the relevant zones of the design space and the Constrained-EGMDO (C-EGMDO) algorithm is capable of identifying the constrained global optimum. Nevertheless, it remains difficult to define a stopping criterion for this type of approach, and thus we consider that a fixed budget exists and that the algorithm is stopped when the entire budget has been spent. Figure 5.5 illustrates the progression of the C-EGMDO algorithm when the maximum number of iterations is set to 5 (note that one iteration consists on adding a new point to DoE_{UQ} and enriching the disciplinary surrogates, if needed).

As is shown, for the illustrative 1-D constrained MDO problem, the algorithm quickly finds the constrained global optimum to be at $x \approx -2.42$. In fact, the first three points proposed by the CEI criterion were in this region (see Figure 5.5(a)), leading to a decrease of the uncertainty around the constrained global optimum. In the iterations that came after, the points proposed by the CEI were driven by the exploration of the design space where the uncertainty was high



(a) Iteration 3.

(b) Iteration 5.

Figure 5.5: Progression of the C-EGMDO algorithm. Illustration of the objective function approximation after (a) 3 iterations and (b) 5 iterations. The last point added is identified with white dot and the constrained global optimum is marked with a yellow star. The approximate infeasible region is shaded in gray.

(see Figure 5.5(b)). Regardless, the presented objective function realizations show that in areas of the design space where the global minimum is unlikely to be found the uncertainty has not been reduced. This implies that little computational effort has been spent trying to improve disciplinary surrogate approximations in these regions. Indeed, during the 5 iterations performed, calls to the disciplinary solvers were only made at points $x = -2$, $x = -2.5$ and $x = -2.4$. At other points in DoE_{UQ}, where the probability of being solution to the constrained problem is low, the disciplinary solvers were never enriched. As a consequence, the objective function approximation at these points does not correspond to its real value.

5.5 Numerical tests

To test the performance of the proposed constraint handling strategy, we use the C-EGMDO algorithm to solve an analytical benchmark MDO problem as well as an engineering test case. The obtained results are presented in the following.

5.5.1 Analytical Sellar benchmark test case

The analytical Sellar benchmark test case, previously introduced in Section 4.3.1, is chosen to test the proposed constraint handling strategy. We recall that this problem presents a single objective function (f_{obj}), two scalar coupling variables (y_1, y_2), two inequality constraints ($g_1 \geq 0, g_2 \geq 0$) and three design variables (z_1, z_2, z_3).

5.5.1.1 Robustness study

To test the robustness of the proposed approach, 100 initial DoEs ($\text{DoE}_{f_1}, \text{DoE}_{f_2}, \text{DoE}_{\text{UQ}}$) are randomly generated. The number of samples in each disciplinary DoE is 5, generated via LHS, and

using an initial guess for the coupling variable space of $\mathscr{C}_1 = \mathscr{C}_2 = [0,25]$. The size of DoE_{UQ} was set to 20. The random MDA is solved for 200 points and a third-degree PCE is retained. For all GP approximations a squared-exponential correlation function is used. The threshold for the coefficient of variation ϵ_{CV} is set to 1%, while the probability threshold ϵ_{P} is set to 10^{-3}. The maximum number of iterations for the C-EGMDO algorithm is set to 10. Out of the 100 performed runs, 63 were able to find the constrained global optimum with a relative error of less than 1% with respect to the reference objective function value. Table 5.1 summarizes the obtained results for the 63 successful runs and compares them with the reference values. The number of disciplinary solver calls is also presented.

	z_1^*	z_2^*	z_3^*	f_{obj}^*	n_1^{eval}	n_2^{eval}
Ref.	1.9776	0	0	3.183	$--$	$--$
$\mathbb{E}$	1.965	$\leq 10^{-12}$	5.60×10^{-2}	3.197	$5+7.1$	$5+4.5$
CV	7.56×10^{-3}	6.07	0.90	4.23×10^{-3}	0.24	0.31
ϵ_{rel}	0.61%	$--$	$--$	0.27%	$--$	$--$

Table 5.1: Mean values, coefficients of variation and relative errors between the mean and reference values of z^* and f_{obj}^* obtained during the 63 successful runs when using the CEI criterion. The number of disciplinary solver calls is equally given.

Out of the remaining 37 runs, 5 presented a relative error of less than 5%, suggesting that they would benefit from a greater number of iterations. For the other 32 runs, the optimization of the acquisition function failed to propose points in the vicinity of the global optimum. This problem arises due to the near-flat zones of the CEI criterion (see Eq. (5.10)) which make its maximization difficult, despite the use of a multi-start strategy. The use of an alternative acquisition function can ease this problem and lead to a more robust framework. Indeed, when replacing the CEI criterion by the WB2 criterion, the C-EGMDO algorithm converged to the constrained global optimum with less 1% of relative error for 96 out of 100 runs. Table 5.2 summarizes the results obtained for the 96 successful runs, when the WB2 criterion is used as acquisition function.

	z_1^*	z_2^*	z_3^*	f_{obj}^*	n_1^{eval}	n_2^{eval}
Ref.	1.9776	0	0	3.183	$--$	$--$
$\mathbb{E}$	1.955	$\leq 10^{-8}$	8.72×10^{-2}	3.198	$5+8.0$	$5+4.8$
CV	7.95×10^{-3}	3.45	0.65	4.45×10^{-3}	0.17	0.33
ϵ_{rel}	1.10%	$--$	$--$	0.30%	$--$	$--$

Table 5.2: Mean values, coefficients of variation and relative errors between the mean and reference values of z^* and f_{obj}^* obtained using the WB2 infill criterion. The number of disciplinary solver calls is equally given.

Table 5.2 shows that the use of the WB2 criterion as acquisition function not only increases the number of successful runs, but leads to a similar level of accuracy in the obtained results, and a similar computational cost compared to when the CEI criterion is used. For this reason, the WB2 infill criterion is used as acquisition function for the remaining tests conducted.

5.5.1.2 Constraint relaxation

The relaxation of inequality constraints is now studied. We recall that a continuous model of the constraint functions is obtained by interpolating the mean value of the respective PCE ap-

proximations. Thus, one can relax the inequality constraints by rather interpolating at a given quantile, leading to an increase of the approximated feasible region, when the uncertainty is high. The choice of quantile depends on the nature of the constraints, i.e. it depends on whether they are negative or positive constraints. For the chosen problem, where we deal with positive constraints (see Eq. (4.9)), we test the 95% and 99% quantiles. Table 5.3 presents the number of successful runs ($n_{(\epsilon\leq 1\%)}$) as well as the mean number of disciplinary solver calls obtained after 100 runs for each tested quantile.

τ-quantile	$\tau = 0.95$	$\tau = 0.99$
$n_{(\epsilon\leq 1\%)}$	88	74
$\mathbb{E}(n_1^{\text{eval}})$	13.5	13.5
$\mathbb{E}(n_2^{\text{eval}})$	10.0	10.0

Table 5.3: Number of successful runs and mean number of disciplinary solver calls made when interpolating the PCE approximations of the constraint functions at different quantiles.

The obtained results show that, on this problem, relaxing the constraints leads to a less robust framework, compared to the case where interpolation is made at the mean value of the PCE approximation, where the number of successful runs was 96 out of 100 performed runs. Moreover, from within the successful runs, the computational cost for the expected number of disciplinary solver calls was approximately the same for all tested quantiles. Since it leads to a more robust framework, in the remaining conducted tests, interpolation is made at the mean value of the PCE approximation.

5.5.1.3 Comparison with other approaches

Comparisons with other MDO frameworks are now provided. For that, we choose a gradient based optimizer (SLSQP), a gradient free optimizer (COBYLA) and the Bayesian Super Efficient Global Optimization with Mixture of Experts (SEGOMOE) optimizer [Bartoli et al., 2019]. For all three frameworks, both MDF and IDF approaches are tested and compared against the C-EGMDO framework. A total of 100 runs are performed for each framework using different starting points for the SLSQP and COBYLA algorithms and different initial DoEs for the SEGOMOE framework. For the SLSQP and COBYLA frameworks the problem is implemented in OpenMDAO. For the gradient-based optimizer, we use finite-difference approximation of the derivatives, under the assumption of black-box disciplinary solvers. For the SEGOMOE framework, different sizes of the initial DoEs were considered for the MDF (4 points) and IDF (5 points) implementations, to account for the increase in the number of design variables in the IDF approach. Both initial DoEs are generated using LHS. Table 5.4 presents the number of successful runs as well as the total number of disciplinary solver calls obtained for all tested frameworks during the 100 performed runs. We recall that a run is considered successful if the global optimum is found with less than 1% or relative error.

The obtained results show that the gradient-based SLSQP optimizer finds the global optimum for about half of the runs (43/100 runs for the MDF approach and 59/100 runs for the IDF approach). The gradient-free COBYLA optimizer performs a little better, having found the global optimum for 90/100 runs when employing the IDF approach. Nevertheless, the number of disciplinary solver calls is significantly higher for the COBYLA optimizer compared with the remaining frameworks. The Bayesian SEGOMOE algorithm presents the best results among the three alternative frameworks, having found the global optimum for almost all runs, for both MDF and IDF approaches. Moreover the IDF-SEGOMOE framework required only an average of 56.6 disciplinary solver calls. This number is halved by the proposed C-EGMDO algorithm which required

	MDF-SLSQP	MDF-COBYLA	MDF-SEGOMOE	C-EGMDO
$n_{(\epsilon\leq 1\%)}$	43	55	94	96
$\mathbb{E}(n_1^{\text{eval}} + n_2^{\text{eval}})$	136.2	334.2	259.5	22.8
	IDF-SLSQP	IDF-COBYLA	IDF-SEGOMOE	
$n_{(\epsilon\leq 1\%)}$	59	90	95	
$\mathbb{E}(n_1^{\text{eval}} + n_2^{\text{eval}})$	103.1	210.9	56.6	

Table 5.4: Number of successful runs ($n_{(\epsilon\leq 1\%)}$) and expected value for the total number of disciplinary solver calls ($\mathbb{E}(n_1^{\text{eval}} + n_2^{\text{eval}})$) obtained after 100 runs of each tested framework.

only an average of 22.8 disciplinary solver calls.

5.5.2 Engineering test case

The C-EGMDO framework is now tested on an engineering test case, where the disciplinary solvers are low-fidelity solvers, used to perform the static aeroelastic analysis of an aircraft wing. Two disciplines are considered: a structural solver determines the wing displacement based on the applied aerodynamic loads while an aerodynamic solver calculates the aerodynamic forces based on the wing deformed shape. The following non-linear system of equations is then defined:

$$\begin{cases} u_s = \mathcal{M}_s(\boldsymbol{x}, f_a) \\ f_a = \mathcal{M}_a(\boldsymbol{x}, u_s) \end{cases} \tag{5.14}$$

where u_s is the structural displacement of the wing spar, f_a are the aerodynamic loads and $\boldsymbol{x}$ is the set of design variables. Low-fidelity solvers present the advantage of allowing for a rapid exploration of the design space and have been often used to assess new wing designs [Jansen et al., 2010, Jasa et al., 2018]. A brief description of the chosen disciplinary solvers is provided in the following.

5.5.2.1 Disciplinary models

In this work, the wing structure is modeled as a single tubular spar, assumed to be placed at 35% of the wing chord. The deformation of the spar due to the aerodynamic loads is obtained using a finite-element model made of beam elements. The model is implemented in the finite element solver Code Aster and a linear elastic behavior is assumed. The aerodynamics model is described by potential flow theory and is solved using an in-house implementation of the Vortex Lattice Method, described in Section 2.1.2.2.

5.5.2.2 Load and displacement transfer

To allow for the construction of the disciplinary surrogates, coupling variables should be of low-dimension. Although model-order reduction could have been employed, in this section, the transfer of loads and displacements is simplified by parameterizing the distribution of aerodynamic loads and structural displacements along the span. To that end, the resultant of lift and torsion moment along the chord are obtained at the spar position. Figure 5.6 illustrates the panel forces $f_{(j,i)}$, obtained from the VLM solver, where j is the panel spanwise index and i is the panel chordwise index. The chordwise resultant lift (F_j) and torsion moment (M_j) are also represented.

The calculated resultants give the distribution of aerodynamic loads along the span, for which we can obtain a parameterized approximation. To simplify the parameterization, an elliptical lift

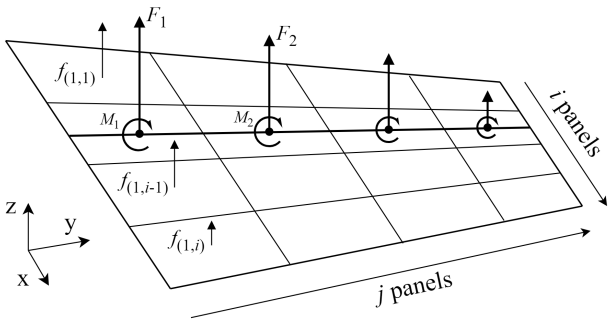


Figure 5.6: Illustration of the load transfer scheme. The chordwise resultant of the vertical force F_j and torsion moment M_j along the chord is obtained at the spar position.

distribution was considered:

$$L(y) = L_{\max}\sqrt{1 - y^2/b^2}\,, \tag{5.15}$$

where $L_{\max}$ is the maximum lift along the span b. For the torsion moment a third-degree polynomial is fitted to the obtained spanwise distribution:

$$M(y) = m_3 y^3 + m_2 y^2 + m_1 y + m_0. \tag{5.16}$$

The set of aerodynamic coupling variables is then written as $f_a = \{L_{\max}, m_0, m_1, m_2, m_3\}$.

For the structural coupling variables, a similar parameterization is used. The wing vertical displacement δz is assumed to follow a quadratic spanwise distribution, written as:

$$\delta z(y) = \delta z_{\text{tip}} \frac{y^2}{b^2}\,, \tag{5.17}$$

where δz_{tip} is the vertical wing tip displacement. The wing is also assumed to deform due to the torsion of the spar and a third-degree polynomial is used to describe the spanwise rotation of the wing around its elastic axis:

$$\theta(y) = r_3 y^3 + r_2 y^2 + r_1 y + r_0. \tag{5.18}$$

The set of structural coupling variables is defined as $u_s = \{\delta z_{\text{tip}}, r_0, r_1, r_2, r_3\}$.

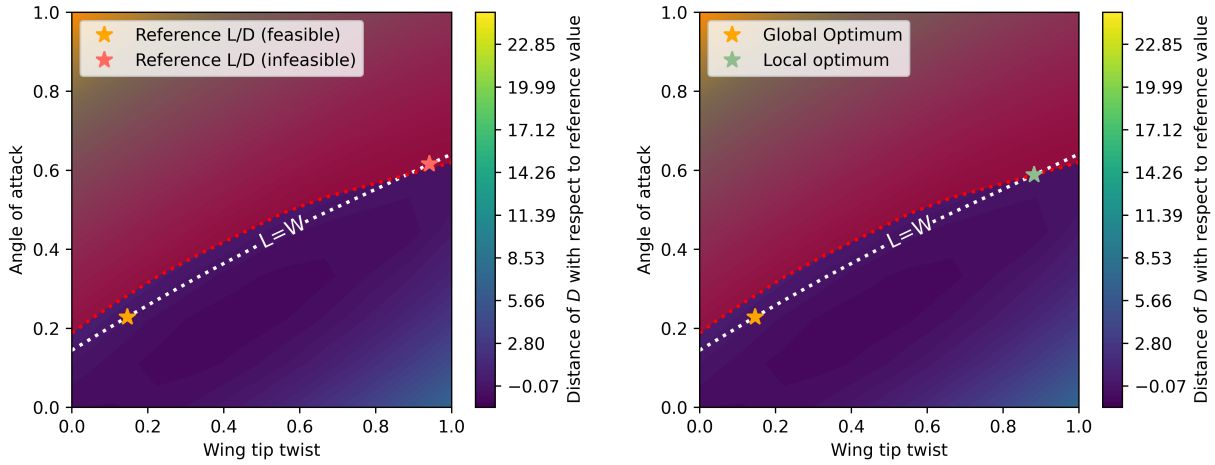
5.5.2.3 Optimization problem

An inverse design problem is then defined, where the optimizer must find a reference wing configuration, characterized by a given lift-to-drag ratio. To find the reference configuration, the optimizer is allowed to vary the wing angle of attack (α) and twist at tip (θ_t). Constraints are also applied. They are used to assure that, at the optimal solution, the wing meets the lift-equals-weight condition ($L = W$) and that the vertical wing tip displacement is less than a pre-defined limit ($\delta z_{\text{tip}} \leq \delta z_{\max}$). Figure 5.7(a) illustrates the considered design space, as well as the imposed constraints. As is shown, in the considered design space, there exist two wing configurations that simultaneously present the reference lift-to-drag ratio and respect the lift-equals-weight condition. One of these configurations presents a low angle of attack and an upwards wing tip twist, while the other presents a high angle of attack and a downwards wing tip twist. Despite their similar lift-to-drag ratio, however, the two configurations present different wing deformations due to the differently distributed loads. Because of this, only one of the configurations is feasible

with respect to the maximum displacement constraint. The optimization problem is defined as follows:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathcal{X}}{\arg\min} \quad & D_{\text{diff}}(\boldsymbol{x}) = \frac{\|D(\boldsymbol{x}) - D_{\text{ref}}\|_2}{D_{\text{ref}}} \\ \text{s.t.} \quad & L(\boldsymbol{x}) = W \\ & \delta z_{\text{tip}}(\boldsymbol{x}) \leq \delta z_{\max} \end{aligned} \tag{5.19}$$

where $\boldsymbol{x}$ is the set of coupling variables ($\boldsymbol{x} = \{\alpha, \theta_t\}$), D_{diff} refers to the normalized relative distance of the evaluated drag with respect to the reference value, D is the evaluated drag for any given configuration and D_{ref} is the reference drag value. Table 5.5 provides the bounds for the design variables and constraint functions, where applicable. It should be noted that the design variables are scaled to take values within $[0,1]$. The reference solution of the problem, established using an MDF-SLSQP framework ran from different starting points, is also listed in Table 5.5. An illustration of the identified global and local optima in the considered design space is given in Figure 5.7(b).



(a) Constraint functions.

(b) Global and local optima.

Figure 5.7: Illustration of the defined engineering test case. (a) Identification of the two design space points that provide the reference lift-to-drag ratio. (b) Identification of global and local optima. The defined constraint functions are equally outlined and the infeasible region with respect to the inequality constraint is shaded in red.

		Bounds		Optima	
		Lower	Upper	Global	Local
minimize	D_{diff}	$--$	$--$	$\approx 1\times10^{-6}$	$\approx 8\times10^{-2}$
w.r.t.	α	0	1	0.2287	0.5885
	θ_t	0	1	0.1462	0.8821
subject to	$L = W$	0	0	0	0
	$\delta z_{\text{tip}} - \delta z_{\max}$	$--$	0	−0.1589	0

Table 5.5: Description of the defined engineering test case. Bounds for design variables and constraints are provided, where applicable. The global and local optima are also identified.

5.5.2.4 Running C-EGMDO

In order to set up the C-EGMDO framework, an initial guess for the coupling variable space is provided, based on some initial runs made for the aerodynamics and structural disciplines. We recall that even if the initial guess is poor, the coupling variable space may be updated when the disciplinary solvers are enriched. Tables 5.6 and 5.7 show, respectively, the initial guess for the aerodynamic and structural coupling variable spaces.

	$L_{\max}$	m_0	m_1	m_2	m_3
Lower	-75000	-10×10^5	-5×10^3	-5×10^2	-5
Upper	75000	10×10^5	5×10^3	5×10^2	5

Table 5.6: Aerodynamic coupling variable space initial guess.

	$\delta z_{\max}$	r_0	r_1	r_2	r_3
Lower	-5	-2.5×10^{-3}	-5×10^{-3}	-2.5×10^{-4}	-1×10^{-5}
Upper	5	5×10^{-3}	5×10^{-3}	2.5×10^{-4}	1×10^{-5}

Table 5.7: Structural coupling variable space initial guess.

In terms of sample size, the disciplinary DoEs are made of 10 points each, while the DoE_{UQ} consists of 15 points (all DoEs are obtained via LHS). The number of random MDAs is set to 500 due to the dimension of the coupling variable space. Uncertainty propagation by PCE is carried out over the 15 points, and a PCE of degree 3 is retained. The threshold for the coefficient of variation ϵ_{CV} is set to 0.01, while the probability threshold ϵ_{P} is set to 10^{-3} (see Algorithm 11). The maximum number of iterations is set to 15. The C-EGMDO algorithm is then run for 10 different initial DoEs. Out the 10 performed runs, 8 converged to the global optimum with less than 5% of relative error with respect to the design variable values at the reference solution. The results obtained for the 8 successful runs are presented in Table 5.8.

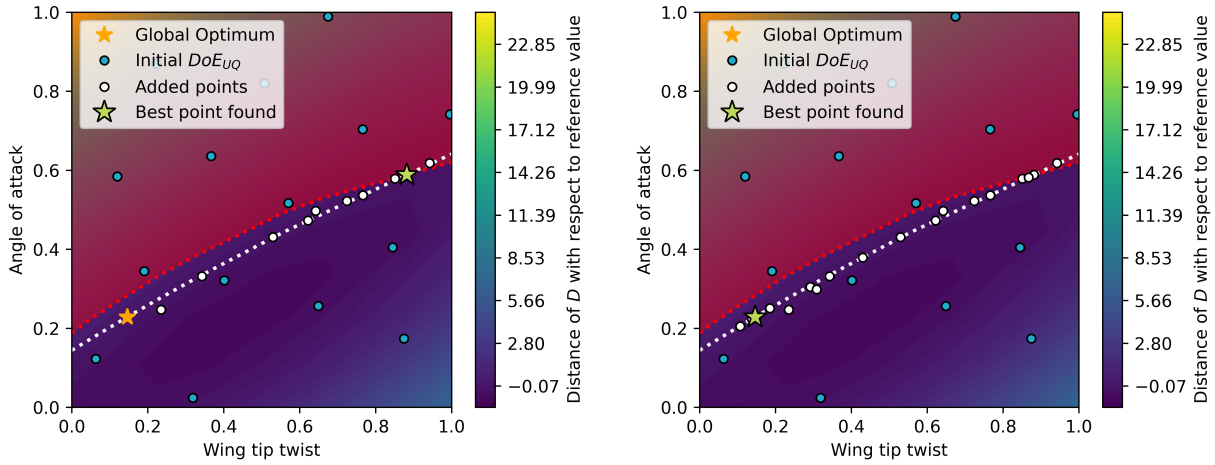
	α^*	θ_t^*	D^*_{diff}
Ref.	0.2287	0.1462	1×10^{-6}
$\mathbb{E}$	0.2296	0.1474	5.04×10^{-3}
CV	0.0111	0.0299	0.8172
ϵ_{rel}	0.40%	0.83%	$--$

Table 5.8: Engineering test case results. Expected value and coefficient of variation of the design variables and objective function, obtained from the 8 successful runs.

The results of Table 5.8 show that, for the successful runs, the proposed C-EGMDO framework finds the constrained global optimum with a relatively small error. In the following we address the failed runs, i.e., the 2 runs where the global optimum was not found.

Failed runs. To analyze the cases where the global optimum was not found, the final design of experiments DoE_{UQ} obtained for one of the two unsuccessful runs is presented in Figure 5.8(a). As is shown, for the plotted run, several points that do not respect the equality constraint ($L = W$) were added to the DoE. This suggests that the initial constraint function approximation was of poor quality and that the algorithm could have benefited from a greater number of iterations. To test this hypothesis, 5 additional iterations are made. Figure 5.8(b) presents the obtained

DoE, when the maximum number of iterations is set to 20. As is shown, by adding 5 additional iterations, the algorithm is capable of finding the global optimum. We further note that, after the initial 15 iterations the algorithm had already found the local optimum. Nonetheless, using the 5 additional iterations, the algorithm managed not only to explore the region around the global optimum, but also to correctly identify the true optimum.



(a) After 15 iterations.

(b) After 20 iterations.

Figure 5.8: Engineering test case generated DoE_{UQ} for an unsuccessful run. (a) DoE obtained after (a) 15 and (b) 20 iterations. In both figures, the initial DoE is represented in blue dots while the points proposed by the acquisition function are represented in white dots. The global optimum is marked with a yellow star and the best point found is identified with a green star. The imposed constraints are outlined in dotted lines.

The case depicted in Figure 5.8 reflects the importance of choosing an appropriate stopping criterion for the optimization. While in the proposed C-EGMDO framework we stick to the classical approach of terminating the algorithm after a fixed number of iterations, some works have studied this issue more in depth and proposed an alternative stopping criteria. Among these criteria, some works suggest looking to the maximum value of the EI function [Nguyen et al., 2017, Lorenz et al., 2015]. The idea is that the EI decreases as the iterations progress. Therefore, one can decide to stop the search when the maximum value of the EI is below a defined threshold. Although in this work we have not studied the performance of this stopping criterion, we remark that it can be easily implemented with Algorithm 11.

Computational cost. Lastly, the computational cost of the C-EGMDO framework is studied. To do so, ten additional runs are performed. The number of successful runs as well as the number of disciplinary solver calls is obtained after 5, 10, 15, 20 and 25 iterations. The sample sizes as well as the thresholds ϵ_{CV} and ϵ_{P} remain unchanged. The obtained results are presented in Table 5.9. As is shown, as the iterations progress, the number of successful runs increases. Moreover, at 25 iterations, all runs have found the global optimum with less that 5% of relative error. As expected, however, the increase in the number of iterations is also accompanied by an increase in the number of disciplinary solver calls. This increase is less significant after the 15 iteration threshold, suggesting that most disciplinary solver calls are made during the initial iterations, when the disciplinary GPs are still very uncertain.

$n_{\max}$	5	10	15	20	25
$n_{(\epsilon\leq 5\%)}$	2	6	7	9	10
$\mathbb{E}(n_a^{\text{eval}})$	17.5	22.83	25.71	26.33	26.4
$\mathbb{E}(n_s^{\text{eval}})$	13.0	14.17	14.29	14.78	15.0

Table 5.9: Number of successful runs ($n_{(\epsilon\leq 5\%)}$) as a function of the maximum number of iterations ($n_{\max}$) of the C-EGMDO algorithm. The mean number of aerodynamic ($\mathbb{E}(n_a^{\text{eval}})$) and structural ($\mathbb{E}(n_s^{\text{eval}})$) disciplinary solver calls is equally presented.

5.5.3 Summary of disciplinary GPs for constrained global optimization

In this chapter we extended the EGMDO framework to handle equality and inequality constraints. In the extended framework, it is proposed to build approximate models of the constraint functions, which are then used to limit the adaptive sampling of the design space. For this purpose, a constrained Expected Improvement criterion was defined and a procedure for the enrichment of disciplinary GPs was proposed, for which the uncertainty of the constraint functions is used to determine the most relevant regions of the design space. Application of the proposed C-EGMDO framework to an analytical test case showed that the use of a less multimodal acquisition function (namely, the WB2 function) results in a more robust framework. Constraint relaxation was also studied, however the obtained results showed that interpolation of the constraint function at the mean value of the PCE approximation leads to a greater success rate. Finally comparison with other frameworks showed that the proposed C-EGMDO framework outperforms the remaining approaches in terms of disciplinary solvers calls.

Application to an engineering test case further confirmed that the proposed C-EGMDO algorithm can handle different types of constraint functions and is capable of saving computational cost by preferentially enriching the disciplinary surrogates only in regions of the design space where the constrained global optimum is likely to be. Finally, in the present chapter we chose to test the proposed C-EGMDO algorithm without resorting to model-order reduction strategies to reduce the dimension of the coupling variable space, to simplify the conducted performance studies. However, future work should be expected to combine the proposed C-EGMDO algorithm with the dimension reduction strategy proposed in Chapter 3, to properly assess both accuracy and cost reductions of the proposed approach.

Chapter summary:

In this chapter we proposed to extend the EGMDO framework to handle constrained MDO problems. To that end the following adaptations were made:

- *As acquisition function, we proposed the use of a constrained Expected Improvement criterion, which limits the adaptive sampling of the design space to regions that are likely to be feasible;*
- *For the enrichment of the disciplinary surrogates, we proposed a criterion which accounts for constraint function uncertainty.*

We then used the resulting C-EGMDO algorithm to solve both an analytical and an engineering MDO problems. The obtained results showed that:

- *The proposed algorithm is capable of handling both equality and inequality constraints;*
- *The proposed approach requires significantly less disciplinary solver calls compared to other existing frameworks.*

Finally, we studied the performance of the proposed approach as a function of the maximum number of iterations and showed that, for the considered engineering test case, when the algorithm is allowed to run for a sufficiently large number of iterations, all runs are able to find the global optimum. Moreover, the computational cost did not significantly increase with the increase of maximum number of iterations.

Conclusion and perspectives

Conclusion

In this thesis we contribute to development of an optimization framework that use disciplinary surrogates to reduce the computational cost of preliminary design studies of multidisciplinary systems. To that end, we first introduced in Chapter 1 the existing numerical tools to perform numerical optimization of a given performance function, with or without constraints. We equally discussed the analysis of multidisciplinary systems and introduced the EGMDA strategy, which uses adaptively enriched disciplinary GPs to reduce the computational cost of solving the MDA. Finally, we discussed MDO problems and introduced the EGMDO algorithm as a global optimization algorithm, specifically developed for the optimization of partitioned multidisciplinary systems.

In Chapter 2 we discussed existing reduced-order models that can be used to obtain low dimensional approximations of disciplinary solvers whose governing equations are PDEs. Such solvers allow to accurately compute relevant quantities of interest. However, due to their computational cost, they have seldom been used for preliminary design studies. The methods introduced in Chapter 2 allow to obtain computationally cheaper approximations of the disciplinary outputs and thus are appropriate for multi-query contexts, such as the one encountered in MDO. Finally, the DPOD+I strategy which combines disciplinary POD bases with GP approximations of the generalized coordinates was introduced as a model order reduction strategy, specifically developed to reduce the computational cost associated with the analysis of multidisciplinary systems, whose disciplines are modelled by PDEs.

Having introduced the state of the art, we then proposed three main contributions to efficiently solve MDO problems. First, a dimension reduction strategy which combines local POD bases with GP interpolation of the generalized coordinates was proposed. Then, we proposed to use the derivatives of the built disciplinary surrogates to perform gradient-based MDO. Finally, we devised a constraint handling strategy for the EGMDO algorithm, a Bayesian optimization algorithm specifically developed for MDO problems. The main conclusions drawn from each contribution are discussed in the following.

In Chapter 3 we proposed the use of local POD bases for building disciplinary surrogates in MDA. To that end, we first introduced a strategy for the interpolation of local POD bases using GPs. A strategy for estimating the quality of the local POD basis interpolation was equally proposed, by generating random realizations of the interpolated matrix and looking at the dispersion of the corresponding points in the Grassmann manifold. The use of local POD bases for single-discipline analysis was studied next. Application to a wing in transonic flight showed that, when using local POD bases, the number of basis vectors is less than the one obtained for a global POD basis, trained over the entire parametric space. Moreover, the interpolation error is smaller when using local POD bases, compared to when a global POD basis is used, for all tested truncation choices. Finally, the proposed approach was used in the static aeroelastic analysis of an aircraft wing, subject to different flight conditions in terms of angle of attack and freestream Mach number. This test case confirmed the interest of the proposed approach for applications requiring many MDA resolutions. Indeed, the use of local POD bases allowed for the reduction of the average computational cost of one MDA resolution, having shown almost no compromise

in terms of accuracy of the obtained solution.

In Chapter 4 we proposed to use disciplinary surrogates for gradient-based MDO. To that end, we introduced the analytical derivatives of the disciplinary GPs and showed that by successively enriching the disciplinary surrogates at a queried design space point, the quality of the corresponding derivatives improves. We equally discussed how the derivatives of vector valued coupling variables can be obtained, both when using global and local POD bases as dimension reduction strategy. We then applied the proposed approach to solve both an analytical and engineering MDO problems. The obtained results showed that using the disciplinary surrogates allows to save a significant amount of computational cost when the derivatives of the real solvers are not available. Moreover, we saw that, despite the errors committed on the computed derivatives, the obtained MDO solutions lied close to the true optimum. Finally, we presented the use of the proposed approach thanks to the WhatsOpt collaborative environment. This allows users to employ the proposed approach with minimal implementation effort.

In Chapter 5 we extended the EGMDO framework to handle constraint MDO problems. To that end, we proposed the following adaptation to the two-step uncertainty reduction strategy. As acquisition function, we proposed the use of a constrained Expected Improvement criterion as acquisition function. This criterion limits the adaptive sampling of the design space to region that are likely to be feasible. For the enrichment of the disciplinary surrogates, we proposed a criterion which accounts for both objective and constraint function uncertainties, to restrict computational cost to areas of the design space where the constrained global optimum is likely to be. We then applied the proposed Constrained-EGMDO (C-EGMDO) algorithm to solve both an analytical and an engineering MDO problems. The obtained results showed that C-EGMDO is capable of handling both equality and inequality constraints. Moreover, we saw that it required significantly less disciplinary solver calls, compared to other existing frameworks. Finally, we studied the performance of the C-EGMDO algorithm as a function of the maximum number of iterations. It was shown that, for the considered engineering test case, when the C-EGMDO algorithm is allowed to run for a sufficiently large number of iterations, all runs were able to find the global optimum. Additionally, the number of disciplinary solver calls did not significantly increase with the increase of the maximum number of iterations. In the following, some perspectives for future work are given.

Future perspectives

The developments made in this thesis also gave rise to some short-term perspectives for future work. Indeed, in this work the choice of variable separation strategy to train the database of pointwise local POD bases was made arbitrarily. However, a sensitivity analysis procedure could be envisioned to allow for an informed decision on the variable separation strategy. Similarly, in this work, the considered multidisciplinary systems only required the use of local POD bases for the aerodynamics disciplines. In the future, systems requiring the use of local POD bases for more than one discipline should be studied. For the gradient-based optimization using disciplinary surrogate derivatives, we observed that, despite having attained the convergence criterion of the coefficient of variation of the random MDA solution, some error is committed on the derivatives of the objective and constraint functions with respect to the design variables. To improve this result, future work could focus on the development of a surrogate enrichment strategy which aims at learning the derivative of the disciplinary solvers with respect to the design variables, rather than learning the solution of the MDA itself. Finally, concerning the proposed C-EGMDO algorithm, we chose to test the proposed approach without resorting to model-order reduction strategies to reduce the dimension of the coupling variable space, to simplify the conducted performance studies. However, future work should be expected to combine the proposed

C-EGMDO algorithm with the use of local POD bases to reduce the dimension of the coupling variable space.

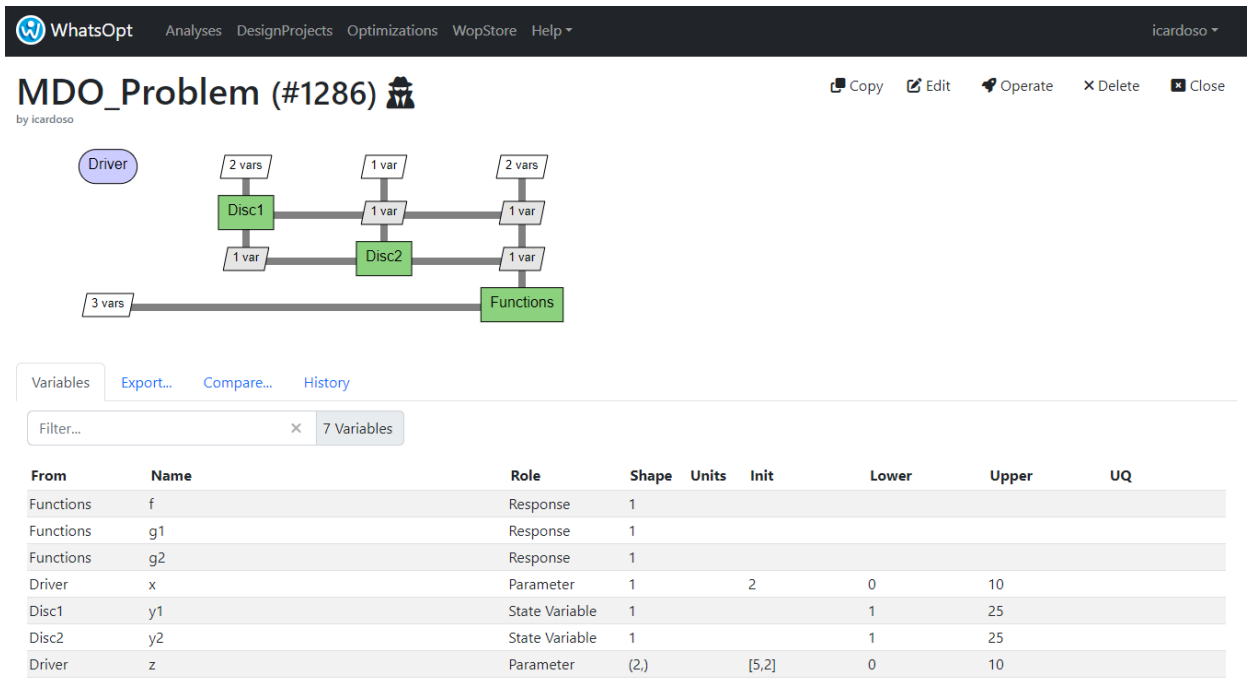
Long-term perspectives could also be envisioned. Indeed, only the use of local-reduced order bases was considered in this work to address the issues encountered by the original DPOD+I strategy. However, the use of quadratic approximations, such as those proposed in [Barnett and Farhat, 2022, Geelen et al., 2023], is a valid alternative. Integrating these approximations with the DPOD+I strategy and comparing their performance with that obtained for the proposed approach should be envisioned as future work. Similarly, in this work it was assumed that disciplinary output data used to train the disciplinary GPs come from a single source. However, several works have proposed using multiple levels of fidelity to reduce the overall computational cost of the optimization problem [Lam et al., 2015, Meliani et al., 2019]. Integrating the use of multi-fidelity GPs [Kennedy and O'Hagan, 2000, Qian and Wu, 2008] within the proposed approach could equally be envisioned as future work. Lastly, we assumed that we do not have access to the disciplinary solver derivatives. For some disciplinary solvers, although we do not have access to the underlying numerical implementations, gradient information is still provided in addition to the output at any queried point. In such cases, gradient-enhanced *kriging* [Liu, 2003, Liem et al., 2015, Bouhlel and Martins, 2019] could be used to build the disciplinary surrogates.

In terms of applications, we note that, although most applications considered in this work involved only two disciplines, future work should include the application of the proposed approach to systems involving three or more disciplines. Indeed, in most preliminary design studies, not only aerodynamics and structural disciplines are considered, but also propulsion, control and mission analysis, among others. Problems involving a greater number of design variables should also be considered in future work. Such problems will increase the dimension of the input space for the disciplinary surrogates and can therefore be expected to pose a challenge to the proposed approach. To address this issue, we could consider looking for a lower dimensional representation of the input space, as proposed in [Garnett et al., 2013, Bouhlel et al., 2016, Zhang et al., 2019, Gaudrie et al., 2020]. Finally, some preliminary design studies also deal with categorical or discrete design variables. For instance, in aircraft design, integer variables are used to choose the number of engines while categorical variables are used to choose between different materials. To handle these type of design variables, some works have proposed the use of mixed-categorical GPs [Halstrup, 2016, Pelamatti et al., 2019, Garrido-Merchán and Hernández-Lobato, 2020, Saves, 2024]. Integration of these mixed-categorical GPs with the proposed approach should be envisioned as future work to increase the corresponding range of application.

Appendix

Basic WhatsOpt usage

WhatsOpt is a web application which allows users to define multidisciplinary analysis problems in terms of disciplines and data exchanges. Once the problem has been defined, WhatsOpt generates the OpenMDAO skeleton code where implementations of the disciplinary solvers can be plugged into. Figure A1 displays the user graphical interface when a generic two-discipline MDO problem is defined using WhatsOpt. As shown, when implementing an MDO problem in WhatsOpt, an interactive XDSM diagram of the problem is generated automatically, allowing users to easily control the flow of information between the different disciplines. Additionally, an interactive table allows users to define all disciplinary inputs and outputs as design or state variables, as well as objective or constraint functions. The bounds for the design variables can equally be defined in this step.



From	Name	Role	Shape	Units	Init	Lower	Upper	UQ
Functions	f	Response	1					
Functions	g1	Response	1					
Functions	g2	Response	1					
Driver	x	Parameter	1		2	0	10	
Disc1	y1	State Variable	1			1	25	
Disc2	y2	State Variable	1			1	25	
Driver	z	Parameter	(2,)		[5,2]	0	10	

Figure A1: WhatsOpt user graphical interface. Implementation of a generic two-discipline MDO problem.

Once users are done with the problem definition, they can request WhatsOpt to generate the necessary scripts to run the problem in OpenMDAO. For all disciplines, objective and constraint functions, WhatsOpt will generate a python script where the respective output computations should be implemented by the user. WhatsOpt will equally generate the required files to solve the MDA and MDO problems. The latter will use by default OpenMDAO solver options. The list

of files generated for the generic two-discipline problem of Figure A1 is given below:

- `Disc1_base.py, Disc2_base.py, Functions_base.py`: these files contain connection information (inputs and outputs) for all defined components;
- `Disc1.py, Disc2.py, Functions.py`: these are python class files where output computations need to be manually implemented by the user;
- `MDO_Problem_base.py, MDO_Problem.py`: these files contain the definition of the problem;
- `mda_init.py, run_mda.py, run_mdo.py`: these files allow to solve the MDA and MDO problems using the real disciplinary solvers.

Some additional files are equally generated by WhatsOpt, but they do not relate to the developments proposed in this work. As a result, we abstain from providing further details on them.

Disciplinary surrogate option

To replace the true MDA by the approximated one, users must state that they want the disciplinary solvers to be replaced by surrogates. This option is available under the OpenMDAO tab of the WhatsOpt graphical interface. Figure A2 displays the OpenMDAO menu, where the option 'EGMDO surrogate' has been ticked for the two disciplinary solvers of the generic two-discipline problem defined in Figure A1.

It should be noted that when ticking the 'EGMDO surrogate' option, users are required to provide an initial guess for the coupling variable space. This can be done in the interactive table shown in Figure A1 or directly in the generated files. Then, all users have left to do is to ask WhatsOpt to generate the code. Since the option 'EGMDO surrogate' has been selected, WhatsOpt will automatically generate the necessary scripts to solve the MDO problem using the derivatives of the disciplinary surrogates (should we choose to use a gradient-based optimizer). The following additional scripts are generated:

- `algorithms.py`: this file contains the implementation of the EGMDA strategy;
- `doe_factory.py`: this file is responsible for the generation of the disciplinary DoEs;
- `gp_factory.py`: this file creates the disciplinary surrogates and updates them whenever enrichments are performed;
- `random_analysis.py`: this file computes random solution of the approximated MDA;
- `MDO_Problem_egmda.py`: this file contains the problem definition, when the disciplinary solvers are replaced by their respective surrogates;
- `run_egmda.py, run_egmdo.py`: these files allow to solve the MDA and MDO problems using the disciplinary surrogates.

Assuming that the true disciplinary solvers, objective and constraint functions have been previously implemented in the respective python files, users don't need to make any additional implementations in order to solve the MDO problem using the derivatives of the disciplinary surrogates. Some remarks should nonetheless be made concerning the generated scripts. Indeed, by default, sampling of the design and coupling variable spaces is made using LHS strategy. Moreover, in WhatsOpt the disciplinary GPs are built using the `KRG` class of SMT using a constant mean and squared exponential correlation functions. These options can be changed manually in the `gp_factory.py` script. Finally, at present time, WhatsOpt generates the files assuming scalar coupling variables. To solve problems involving vector-valued coupling variables, users need to manually replace the computation of the disciplinary surrogate derivatives so that it matches the expression given in Eq. (4.4).

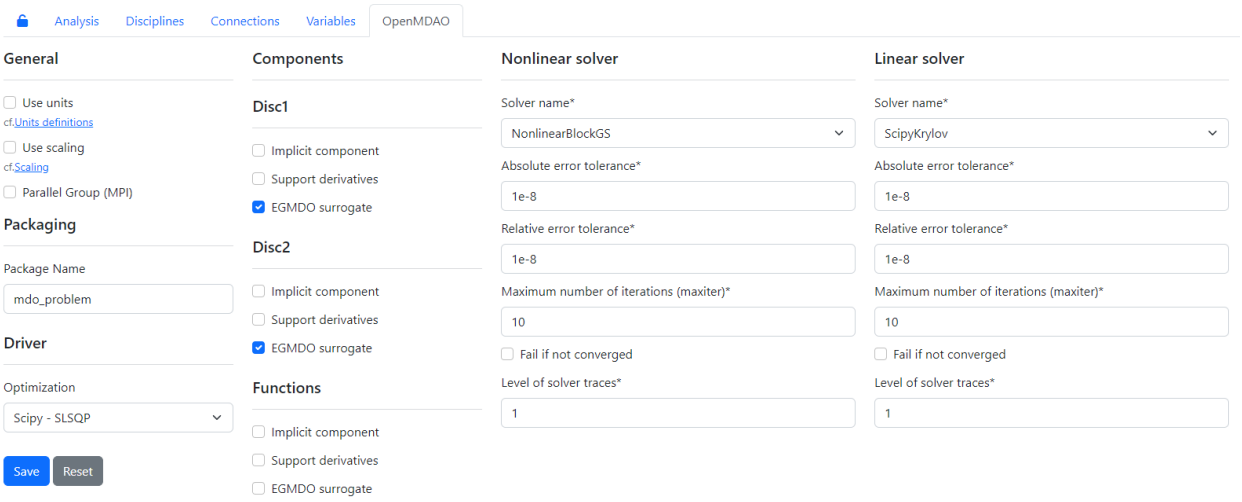


Figure A2: WhatsOpt OpenMDAO menu. Option 'EGMDO surrogate' selected for both disciplinary solvers.

Bibliography

[Absil et al., 2004] Absil, P.-A., Mahony, R. E., and Sepulchre, R. (2004). Riemannian geometry of grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematica*, 80:199–220. 76

[Adler and Martins, 2024] Adler, E. and Martins, J. (2024). Blended wing body configuration for hydrogen-powered aviation. *Journal of Aircraft*. 13

[Ammar, 2010] Ammar, A. (2010). The proper generalized decomposition: a powerful tool for model reduction. *International Journal of Material Forming*, 3(2):89–102. 73

[Amsallem et al., 2009] Amsallem, D., Cortial, J., Carlberg, K., and Farhat, C. (2009). A method for interpolating on manifolds structural dynamics reduced-order models. *International Journal for Numerical Methods in Engineering*, 80(9):1241–1258. 16, 75, 76, 77

[Amsallem and Farhat, 2008] Amsallem, D. and Farhat, C. (2008). Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA Journal*, 46(7):1803–1813. 16, 75, 76, 77

[Anderson, 2016] Anderson, J. (2016). *Fundamentals of Aerodynamics, 6th Edition*. McGraw hill. 67

[Arnst et al., 2012] Arnst, M., Ghanem, R., Phipps, E., and Red-Horse, J. (2012). Dimension reduction in stochastic modeling of coupled problems. *International Journal for Numerical Methods in Engineering*, 92(11):940–968. 56

[Audet and Dennis, 2006] Audet, C. and Dennis, J. E. (2006). Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1):188–217. 28

[Auer, 2002] Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422. 34

[Auer et al., 2002] Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256. 34

[Bachoc, 2013] Bachoc, F. (2013). Cross validation and maximum likelihood estimations of hyper-parameters of gaussian processes with model misspecification. *Computational Statistics & Data Analysis*, 66:55–69. 32

[Barnett and Farhat, 2022] Barnett, J. and Farhat, C. (2022). Quadratic approximation manifold for mitigating the kolmogorov barrier in nonlinear projection-based model order reduction. *Journal of Computational Physics*, 464:111348. 16, 75, 151

[Barrault et al., 2004] Barrault, M., Maday, Y., Nguyen, N. C., and Patera, A. T. (2004). An "empirical interpolation" method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672. 71

[Bartoli et al., 2019] Bartoli, N., Lefebvre, T., Dubreuil, S., Olivanti, R., Priem, R., Bons, N., Martins, J., and Morlier, J. (2019). Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design. *Aerospace Science and Technology*, 90:85–102. 36, 140

[Bathe and Zhang, 2004] Bathe, K.-J. and Zhang, H. (2004). Finite element developments for general fluid flows with structural interactions. *International Journal for Numerical Methods in Engineering*, 60:213 – 232. 41

[Baudin et al., 2015] Baudin, M., Dutfoy, A., Iooss, B., and Popelin, A.-L. (2015). Open turns: An industrial software for uncertainty quantification in simulation. *arXiv preprint arXiv:1501.05242*. 62

[Berthelin, 2022] Berthelin, G. (2022). *Optimisation multidisciplinaire et réduction d'ordre de modèle*. PhD thesis, Université de Toulouse. 13, 16, 48, 80, 83, 97, 100, 130

[Berthelin et al., 2022] Berthelin, G., Dubreuil, S., Salaün, M., Bartoli, N., and Gogu, C. (2022). Disciplinary proper orthogonal decomposition and interpolation for the resolution of parameterized multidisciplinary analysis. *International Journal for Numerical Methods in Engineering*, 123(15):3594–3626. 45, 48, 80

[Betro and Schoen, 1992] Betro, B. and Schoen, F. (1992). Optimal and sub-optimal stopping rules for the multistart algorithm in global optimization. *Mathematical Programming*, 57(1):445–458. 28

[Bloebaum et al., 1992] Bloebaum, C. L., Hajela, P., and Sobieszczanski-Sobieski, J. (1992). Non-hierarchic system decomposition in structural optimization. *Engineering Optimization*, 19(3):171–186. 52

[Boggs and Tolle, 2000] Boggs, P. T. and Tolle, J. W. (2000). Sequential quadratic programming for large-scale nonlinear optimization. *Journal of Computational and Applied Mathematics*, 124(1):123–137. 38

[Boncoraglio and Farhat, 2022] Boncoraglio, G. and Farhat, C. (2022). Piecewise-global nonlinear model order reduction for pde-constrained optimization in high-dimensional parameter spaces. *SIAM Journal on Scientific Computing*, 44(4):A2176–A2203. 75

[Bons et al., 2019] Bons, N. P., He, X., Mader, C. A., and Martins, J. R. R. A. (2019). Multimodality in aerodynamic wing design optimization. *AIAA Journal*, 57(3):1004–1018. 15

[Bons and Martins, 2020] Bons, N. P. and Martins, J. R. R. A. (2020). Aerostructural design exploration of a wing in transonic flow. *Aerospace*, 7(8):118. 15

[Boothby, 2003] Boothby, W. M. (2003). *An introduction to differentiable manifolds and Riemannian geometry*. Academic Press, New York. 76

[Bouhlel et al., 2016] Bouhlel, M., Bartoli, N., Morlier, J., and Otsmane, A. (2016). Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction. *Structural and Multidisciplinary Optimization*, 53:935–952. 36, 151

[Bouhlel et al., 2018] Bouhlel, M. A., Bartoli, N., Regis, R. G., Otsmane, A., and Morlier, J. (2018). Efficient global optimization for high-dimensional constrained problems by using the kriging models combined with the partial least squares method. *Engineering Optimization*, 50(12):2038–2053. 39

[Bouhlel et al., 2019] Bouhlel, M. A., Hwang, J. T., Bartoli, N., Lafage, R., Morlier, J., and Martins, J. R. R. A. (2019). A python surrogate modeling framework with derivatives. *Advances in Engineering Software*, page 102662. 18, 62

[Bouhlel and Martins, 2019] Bouhlel, M. A. and Martins, J. R. (2019). Gradient-enhanced kriging for high-dimensional problems. *Engineering with Computers*, 35(1):157–173. 151

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge University press. 38

[Braun, 1996] Braun, R. D. (1996). *Collaborative optimization: an architecture for large-scale distributed design*. PhD thesis, Stanford University. 52

[Brevault et al., 2016] Brevault, L., Balesdent, M., Bérend, N., and Le Riche, R. (2016). Decoupled multidisciplinary design optimization formulation for interdisciplinary coupling satisfaction under uncertainty. *AIAA Journal*, 54(1):186–205. 49

[Brooks et al., 2019] Brooks, T. R., Martins, J. R., and Kennedy, G. J. (2019). High-fidelity aerostructural optimization of tow-steered composite wings. *Journal of Fluids and Structures*, 88:122–147. 13

[Broomhead and Lowe, 1988] Broomhead, D. and Lowe, D. (1988). Radial basis functions, multi-variable functional interpolation and adaptive networks. *Royal Signals and Radar Establishment Malvern (United Kingdom)*. 30

[Brown and Vos, 2018] Brown, M. and Vos, R. (2018). Conceptual design and evaluation of blended-wing body aircraft. In *2018 AIAA Aerospace Sciences Meeting*. 13

[Broyden, 1965] Broyden, C. G. (1965). A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19:577–593. 45

[Broyden, 1970] Broyden, C. G. (1970). The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90. 26

[Bui-Thanh et al., 2008] Bui-Thanh, T., Willcox, K., and Ghattas, O. (2008). Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing*, 30(6):3270–3288. 73

[Cardoso et al., 2024a] Cardoso, I., Dubreuil, S., Bartoli, N., Gogu, C., and Salaün, M. (2024a). Constrained efficient global multidisciplinary design optimization using adaptive disciplinary surrogate enrichment. *Structural and Multidisciplinary Optimization*, 67. 18

[Cardoso et al., 2024b] Cardoso, I., Dubreuil, S., Bartoli, N., Gogu, C., and Salaün, M. (2024b). Model order reduction for parameterized multidisciplinary analysis using disciplinary surrogates: application to non-linear solvers. In *AIAA SCITECH 2024 Forum*. 19

[Cardoso et al., 2023] Cardoso, I., Dubreuil, S., Bartoli, N., Gogu, C., Salaün, M., and Lafage, R. (2023). Disciplinary surrogates for gradient-based optimization of multidisciplinary systems. In *II ECCOMAS Thematic Conference on Multidisciplinary Design Optimization of Aerospace Systems*. 18

[Carlberg et al., 2011] Carlberg, K., Bou-Mosleh, C., and Farhat, C. (2011). Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compressive tensor approximations. *International Journal for numerical methods in engineering*, 86(2):155–181. 71

[Carlberg et al., 2013] Carlberg, K., Farhat, C., Cortial, J., and Amsallem, D. (2013). The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242:623–647. 71

[Cebral and Lohner, 1997] Cebral, J. R. and Lohner, R. (1997). Conservative load projection and tracking for fluid-structure problems. *AIAA Journal*, 35(4):687–692. 42

[Chatterjee, 2000] Chatterjee, A. (2000). An introduction to the proper orthogonal decomposition. *Current Science*, 78. 72

[Chaturantabut and Sorensen, 2010] Chaturantabut, S. and Sorensen, D. C. (2010). Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764. 71

[Chevreuil and Nouy, 2012] Chevreuil, M. and Nouy, A. (2012). Model order reduction based on proper generalized decomposition for the propagation of uncertainties in structural dynamics. *International Journal for Numerical Methods in Engineering*, 89(2):241–268. 73

[Chittick and Martins, 2009] Chittick, I. R. and Martins, J. R. (2009). An asymmetric suboptimization approach to aerostructural optimization. *Optimization and Engineering*, 10:133–152. 50

[Choi et al., 2020] Choi, Y., Boncoraglio, G., Anderson, S., Amsallem, D., and Farhat, C. (2020). Gradient-based constrained optimization using a database of linear reduced-order models. *Journal of Computational Physics*, 423:109787. 78

[Coelho et al., 2009] Coelho, R. F., Breitkopf, P., Knopf-Lenoir, C., and Villon, P. (2009). Bilevel model reduction for coupled problems. *Structural and Multidisciplinary Optimization*, 39(4):401–418. 71, 79

[Conn et al., 2000] Conn, A., Gould, N., and Toint, P. (2000). *Trust Region Methods*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics. 26

[Courard et al., 2016] Courard, A., Néron, D., Ladevèze, P., and Ballère, L. (2016). Integration of PGD-virtual charts into an engineering design process. *Computational Mechanics*, 57(4):637–651. 74

[Cramer et al., 1994] Cramer, E. J., Dennis, Jr, J. E., Frank, P. D., Lewis, R. M., and Shubin, G. R. (1994). Problem formulation for multidisciplinary optimization. *SIAM Journal on Optimization*, 4(4):754–776. 50

[Degroote, 2013] Degroote, J. (2013). Partitioned simulation of fluid-structure interaction. *Archives of Computational Methods in Engineering*, 20. 41

[Diouane et al., 2023] Diouane, Y., Picheny, V., Riche, R. L., and Perrotolo, A. S. D. (2023). Trego: a trust-region framework for efficient global optimization. *Journal of Global Optimization*, 86(1):1–23. 36

[Discacciati and Hesthaven, 2023] Discacciati, N. and Hesthaven, J. S. (2023). Localized model order reduction and domain decomposition methods for coupled heterogeneous systems. *International Journal for Numerical Methods in Engineering*, 124(18):3964–3996. 81

[Discacciati and Hesthaven, 2024] Discacciati, N. and Hesthaven, J. S. (2024). Model reduction of coupled systems based on non-intrusive approximations of the boundary response maps. *Computer Methods in Applied Mechanics and Engineering*, 420:116770. 81

[Drela and Youngren, 2004] Drela, M. and Youngren, H. (2004). Athena vortex lattice. Software Package, Ver. 3. 69

[Dubreuil et al., 2020] Dubreuil, S., Bartoli, N., Gogu, C., and Lefebvre, T. (2020). Towards an efficient global multidisciplinary design optimization algorithm. *Structural and Multidisciplinary Optimization*, 62:1739–1765. 16, 45, 46, 54, 130

[Dubreuil et al., 2018] Dubreuil, S., Bartoli, N., Gogu, C., Lefebvre, T., and Mas Colomer, J. (2018). Extreme value oriented random field discretization based on an hybrid polynomial chaos expansion - Kriging approach. *Computer Methods in Applied Mechanics and Engineering*, 332:540–571. 55, 57, 133

[Economon et al., 2016] Economon, T. D., Palacios, F., Copeland, S. R., Lukaczyk, T. W., and Alonso, J. J. (2016). Su2: An open-source suite for multiphysics simulation and design. *AIAA Journal*, 54(3):828–846. 69, 100

[Edelman et al., 1998] Edelman, A., Arias, T. A., and Smith, S. T. (1998). The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353. 76

[EDF, 2017] EDF (1989–2017). Finite element *code_aster*, analysis of structures and thermomechanics for studies and research. Open source on www.code-aster.org. 69, 99

[El Amri et al., 2023] El Amri, R., Riche, R. L., Helbert, C., Blanchet-Scalliet, C., and Veiga, S. D. (2023). A Sampling Criterion for Constrained Bayesian Optimization with Uncertainties. *The SMAI Journal of computational mathematics*, 9:285–309. 39

[Eriksson and Poloczek, 2021] Eriksson, D. and Poloczek, M. (2021). Scalable constrained bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 730–738. 39

[Everson and Sirovich, 1995] Everson, R. and Sirovich, L. (1995). Karhunen–loève procedure for gappy data. *Journal of the Optical Society of America A*, 12(8):1657–1664. 71

[Farhat et al., 2014] Farhat, C., Avery, P., Chapman, T., and Cortial, J. (2014). Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *International Journal for Numerical Methods in Engineering*, 98(9):625–662. 71

[Farhat et al., 2020] Farhat, C., Grimberg, S., Manzoni, A., and Quarteroni, A. (2020). *Computational bottlenecks for PROMs: precomputation and hyperreduction*, pages 181–244. 71

[Farhat et al., 1998] Farhat, C., Lesoinne, M., and Le Tallec, P. (1998). Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157(1):95–114. 42

[Feliot et al., 2017] Feliot, P., Bect, J., and Vazquez, E. (2017). A bayesian approach to constrained single-and multi-objective optimization. *Journal of Global Optimization*, 67(1):97–133. 39

[Felippa et al., 2001] Felippa, C. A., Park, K., and Farhat, C. (2001). Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190(24):3247–3270. Advances in Computational Methods for Fluid-Structure Interaction. 41

[Fletcher, 1970] Fletcher, R. (1970). A new approach to variable metric algorithms. *The Computer Journal*, 13(3):317–322. 26

[Forrester et al., 2008] Forrester, A., Sobester, A., and Keane, A. (2008). *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons. 30

[Friderikos et al., 2022] Friderikos, O., Baranger, E., Olive, M., and Néron, D. (2022). On the stability of POD basis interpolation on grassmann manifolds for parametric model order reduction. *Computational Mechanics*, 70(1):181–204. 79

[Fréchet, 1948] Fréchet, M. (1948). Les éléments aléatoires de nature quelconque dans un espace distancié. *Annales de l'institut Henri Poincaré*, 10(4):215–310. 77

[Garnett et al., 2013] Garnett, R., Osborne, M. A., and Hennig, P. (2013). Active learning of linear embeddings for gaussian processes. *ArXiv*, abs/1310.6740. 36, 151

[Garrido-Merchán and Hernández-Lobato, 2020] Garrido-Merchán, E. C. and Hernández-Lobato, D. (2020). Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes. *Neurocomputing*, 380:20–35. 151

[Gaudrie et al., 2020] Gaudrie, D., Le Riche, R., Picheny, V., Enaux, B., and Herbert, V. (2020). Modeling and optimization with gaussian processes in reduced eigenbases. *Structural and Multidisciplinary Optimization*, 61(6):2343–2361. 36, 151

[Geelen and Willcox, 2022] Geelen, R. and Willcox, K. (2022). Localized non-intrusive reduced-order modelling in the operator inference framework. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 380(2229):20210206. 75

[Geelen et al., 2023] Geelen, R., Wright, S., and Willcox, K. (2023). Operator inference for non-intrusive model reduction with quadratic manifolds. *Computer Methods in Applied Mechanics and Engineering*, 403:115717. 16, 75, 151

[Gervasio and Quarteroni, 2018] Gervasio, P. and Quarteroni, A. (2018). INTERNODES for heterogeneous couplings. In *Domain Decomposition Methods in Science and Engineering XXIV 24*, pages 59–71. Springer. 41

[Ghanem and Spanos, 2003] Ghanem, R. and Spanos, P. (2003). *Stochastic Finite Elements: A Spectral Approach*. Civil, Mechanical and Other Engineering Series. Dover Publications. 54

[Gill and Wong, 2012] Gill, P. and Wong, E. (2012). *Sequential Quadratic Programming Methods*, volume 154, pages 147–224. 38

[Gill et al., 2002] Gill, P. E., Murray, W., and Saunders, M. A. (2002). Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12(4):979–1006. 38

[Giunta, 2000] Giunta, A. (2000). A novel sensitivity analysis method for high fidelity multidisciplinary optimization of aero-structural systems. In *AIAA Paper*. 15

[Gogu, 2015] Gogu, C. (2015). Improving the efficiency of large scale topology optimization through on-the-fly reduced order model construction. *International Journal for Numerical Methods in Engineering*, 101(4):281–304. 82

[Gogu and Passieux, 2013] Gogu, C. and Passieux, J.-C. (2013). Efficient surrogate construction by combining response surface methodology and reduced order modeling. *Structural and Multidisciplinary Optimization*, 47:821–837. 82

[Goldfarb, 1970] Goldfarb, D. (1970). A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24(109):23–26. 26

[Goutaudier et al., 2023] Goutaudier, D., Nobile, F., and Schiffmann, J. (2023). A new method to interpolate POD reduced bases–application to the parametric model order reduction of a gas bearings supported rotor. *International Journal for Numerical Methods in Engineering*, 124(18):4141–4170. 78, 97

[Gray et al., 2019] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A. (2019). OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, 59(4):1075–1104. 15, 62

[Grepl et al., 2007] Grepl, M. A., Maday, Y., Nguyen, N. C., and Patera, A. T. (2007). Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 41(3):575–605. 71

[Grepl, M. and Patera, A., 2005] Grepl, M. and Patera, A. (2005). A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: M2AN*, 39(1):157–181. 73

[Grimberg et al., 2021] Grimberg, S., Farhat, C., Tezaur, R., and Bou-Mosleh, C. (2021). Mesh sampling and weighting for the hyperreduction of nonlinear petrov–galerkin reduced-order models with local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 122(7):1846–1874. 71

[Grimberg et al., 2020] Grimberg, S., Farhat, C., and Youkilis, N. (2020). On the stability of projection-based model order reduction for convection-dominated laminar and turbulent flows. *Journal of Computational Physics*, 419:109681. 70

[Grippo et al., 1989] Grippo, L., Lampariello, F., and Lucidi, S. (1989). A truncated newton method with nonmonotone line search for unconstrained optimization. *Journal of Optimization Theory and Applications*, 60:401–419. 26

[Grossman et al., 1988] Grossman, B., Gurdal, Z., Strauch, G. J., Eppard, W. M., and Haftka, R. T. (1988). Integrated aerodynamic/structural design of a sailplane wing. *Journal of Aircraft*, 25(9):855–860. 50

[Haftka, 1985] Haftka, R. T. (1985). Simultaneous analysis and design. *AIAA Journal*, 23(7):1099–1103. 51

[Halstrup, 2016] Halstrup, M. (2016). *Black-box optimization of mixed discrete-continuous optimization problems*. PhD thesis, Dissertation, Dortmund, Technische Universität, 2016. 151

[Hastie et al., 2001] Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer. 30

[Hernandez et al., 2017] Hernandez, J. A., Caicedo, M. A., and Ferrer, A. (2017). Dimensional hyper-reduction of nonlinear finite element models via empirical cubature. *Computer methods in applied mechanics and engineering*, 313:687–722. 71

[Hernández-Lobato et al., 2015] Hernández-Lobato, J. M., Gelbart, M., Hoffman, M., Adams, R., and Ghahramani, Z. (2015). Predictive entropy search for bayesian optimization with unknown constraints. In *International conference on machine learning*, pages 1699–1707. 39

[Hesthaven and Ubbiali, 2018] Hesthaven, J. S. and Ubbiali, S. (2018). Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78. 71

[Hotelling, 1933] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417. 72

[Hübner et al., 2004] Hübner, B., Walhorn, E., and Dinkler, D. (2004). A monolithic approach to fluid–structure interaction using space–time finite elements. *Computer Methods in Applied Mechanics and Engineering*, 193(23):2087–2104. 41

[Irons and Tuck, 1969] Irons, B. M. and Tuck, R. C. (1969). A version of the Aitken accelerator for computer iteration. *International Journal for Numerical Methods in Engineering*, 1(3):275–277. 45

[Jansen et al., 2010] Jansen, P., Perez, R. E., and Martins, J. R. R. A. (2010). Aerostructural optimization of nonplanar lifting surfaces. *Journal of Aircraft*, 47(5):1491–1503. 141

[Jasa et al., 2018] Jasa, J. P., Hwang, J. T., and Martins, J. R. R. A. (2018). Open-source coupled aerostructural optimization using Python. *Structural and Multidisciplinary Optimization*, 57(4):1815–1827. 15, 141

[Jones et al., 1998] Jones, D., Schonlau, M., and Welch, W. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492. 15, 33

[Jones, 2001] Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21:345–383. 31

[Karhunen, 1947] Karhunen, K. (1947). *Ueber lineare Methoden in der Wahrscheinlichkeitsrechnung*, volume 37 of *Annales Academiae scientiarum Fennicae. Series A. 1, Mathematica-physica*. 72

[Katoch et al., 2021] Katoch, S., Chauhan, S. S., and Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80(5):8091 – 8126. 25

[Katz and Plotkin, 2001] Katz, J. and Plotkin, A. (2001). *Low-Speed Aerodynamics*. Cambridge Aerospace Series. Cambridge University Press, 2 edition. 67

[Kennedy and Eberhart, 1995] Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4. 25

[Kennedy and O'Hagan, 2000] Kennedy, M. C. and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13. 151

[Kim et al., 2003] Kim, H. M., Rideout, D. G., Papalambros, P. Y., and Stein, J. L. (2003). Analytical Target Cascading in Automotive Vehicle Design . *Journal of Mechanical Design*, 125(3):481–489. 52

[Kirkpatrick et al., 1983] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680. 25

[Knill et al., 1996] Knill, D., Balabanov, V., Grossman, B., Mason, W., and Haftka, R. (1996). Certification of a cfd code for high-speed civil transport design optimization. In *34th Aerospace Sciences Meeting and Exhibit*. 14

[Koch et al., 2002] Koch, P. N., Wujek, B., Golovidov, O., and Simpson, T. W. (2002). Facilitating probabilistic multidisciplinary design optimization using kriging approximation models. In *9th AIAA/ISSMO symposium on multidisciplinary Analysis and Optimization*, pages 1–11. 49

[Kraft, 1988] Kraft, D. (1988). *A Software Package for Sequential Quadratic Programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR. 38

[Krige, 1951] Krige, D. G. (1951). A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of The South African Institute of Mining and Metallurgy*, 52:201–203. 15, 30

[Ladevèze, 1985] Ladevèze, P. (1985). Sur une famille d'algorithmes en mécanique des structures. In *Comptes rendus des séances de l'Académie des sciences. Série 2, Mécanique, physique, chimie, sciences de l'univers, sciences de la terre*, volume 300, pages 41–44. 73

[Lafage et al., 2019] Lafage, R., Defoort, S., and Lefebvre, T. (2019). Whatsopt: a web application for multidisciplinary design analysis and optimization. In *AIAA Aviation 2019 Forum*, page 2990. 18, 62

[Lam et al., 2015] Lam, R., Allaire, D. L., and Willcox, K. E. (2015). Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0143. 151

[Lambe and Martins, 2012] Lambe, A. B. and Martins, J. R. R. A. (2012). Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Structural and Multidisciplinary Optimization*, 46(2):273–284. 40

[Lassila et al., 2014] Lassila, T., Manzoni, A., Quarteroni, A., and Rozza, G. (2014). Model order reduction in fluid dynamics: challenges and perspectives. *Reduced Order Methods for modeling and computational reduction*, pages 235–273. 66

[Lee and Carlberg, 2020] Lee, K. and Carlberg, K. T. (2020). Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973. 75

[Liang et al., 2002] Liang, Y., Lee, H., Lim, S., Lin, W., Lee, K., and Wu, C. (2002). Proper orthogonal decomposition and its applications — Part I: Theory. *Journal of Sound and vibration*, 252(3):527–544. 72

[Liem et al., 2015] Liem, R. P., Mader, C. A., and Martins, J. R. (2015). Surrogate models and mixtures of experts in aerodynamic performance prediction for aircraft mission analysis. *Aerospace Science and Technology*, 43:126–151. 151

[Lieu et al., 2006] Lieu, T., Farhat, C., and Lesoinne, M. (2006). Reduced-order fluid/structure modeling of a complete aircraft configuration. *Computer methods in applied mechanics and engineering*, 195(41-43):5730–5742. 66

[Liu, 2003] Liu, W. (2003). *Development of gradient-enhanced kriging approximations for multidisciplinary design optimization*. PhD thesis, University of Notre Dame. 151

[Loève, 1955] Loève, M. (1955). *Probability theory : foundations, random sequences*. Van Nostrand, Princeton, NY. 72

[Lorenz et al., 2015] Lorenz, R., Monti, R. P., Violante, I. R., Faisal, A. A., Anagnostopoulos, C., Leech, R., and Montana, G. (2015). Stopping criteria for boosting automatic experimental design using real-time fMRI with bayesian optimization. *arXiv: Neurons and Cognition*. 145

[Lyness and Moler, 1967] Lyness, J. N. and Moler, C. B. (1967). Numerical differentiation of analytic functions. *SIAM Journal on Numerical Analysis*, 4(2):202–210. 26

[Mallasto and Feragen, 2018] Mallasto, A. and Feragen, A. (2018). Wrapped gaussian process regression on riemannian manifolds. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5580–5588. 78, 88

[Martins, 2002] Martins, J. R. R. A. (2002). *A coupled-adjoint method for high-fidelity aero-structural optimization*. PhD thesis, Standford University. 52, 53

[Martins et al., 2005] Martins, J. R. R. A., Alonso, J. J., and Reuther, J. J. (2005). A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design. *Optimization and Engineering*, 6(1):33–62. 15, 52, 53

[Martins and Lambe, 2013] Martins, J. R. R. A. and Lambe, A. B. (2013). Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51(9):2049–2075. 50

[Martins and Ning, 2021] Martins, J. R. R. A. and Ning, A. (2021). *Engineering Design Optimization*. Cambridge University Press. 50

[Matheron, 1969] Matheron, G. (1969). *Le krigeage universel*, volume 1 of *Les Cahiers du Centre de Morphologie Mathématique de Fontainebleau*. École Nationale Supérieure des Mines de Paris. 30

[Maute et al., 2001] Maute, K., Farhat, C., and Nikbay, M. (2001). Coupled analytical sensitivity analysis and optimization of three-dimensional nonlinear aeroelastic systems. *Aiaa Journal*, 39:2051–2061. 15

[McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133. 30

[McDonald and Gloudemans, 2022] McDonald, R. A. and Gloudemans, J. R. (2022). *Open Vehicle Sketch Pad: An Open Source Parametric Geometry and Analysis Tool for Conceptual Aircraft Design*. 69

[Mckay et al., 1979] Mckay, M., Beckman, R., and Conover, W. (1979). A comparison of three methods for selecting vales of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245. 29

[Meliani et al., 2019] Meliani, M., Bartoli, N., Lefebvre, T., Bouhlel, M.-A., Martins, J. R., and Morlier, J. (2019). Multi-fidelity efficient global optimization: Methodology and application to airfoil shape design. In *AIAA aviation 2019 forum*, page 3236. 151

[Močkus, 1975] Močkus, J. (1975). On bayesian methods for seeking the extremum. In Marchuk, G. I., editor, *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, pages 400–404, Berlin, Heidelberg. Springer Berlin Heidelberg. 15

[Mosquera et al., 2019] Mosquera, R., Hamdouni, A., Hamidi, A. E., and Allery, C. (2019). POD basis interpolation via Inverse Distance Weighting on Grassmann manifolds. *Discrete and Continuous Dynamical Systems - S*, 12(6):1743–1759. 78

[Needels and Alonso, 2023] Needels, J. T. and Alonso, J. J. (2023). Efficient global optimization for multidisciplinary conceptual design of hypersonic vehicles. In *AIAA AVIATION 2023 Forum*. 15

[Nelder and Mead, 1965] Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *The computer journal*, 7(4):308–313. 28

[Néron and Ladevèze, 2010] Néron, D. and Ladevèze, P. (2010). Proper generalized decomposition for multiscale and multiphysics problems. *Archives of Computational Methods in Engineering*, 17:351–372. 73

[Nguyen et al., 2017] Nguyen, V., Gupta, S., Rana, S., Li, C., and Venkatesh, S. (2017). Regret for expected improvement over the best-observed value and stopping condition. In Zhang, M.-L. and Noh, Y.-K., editors, *Proceedings of the Ninth Asian Conference on Machine Learning*, volume 77 of *Proceedings of Machine Learning Research*, pages 279–294, Yonsei University, Seoul, Republic of Korea. PMLR. 145

[Nocedal and Wright, 1999] Nocedal, J. and Wright, S. J. (1999). *Numerical optimization*. Springer. 25

[Oakley et al., 1998] Oakley, D. R., Sues, R. H., and Rhodes, G. S. (1998). Performance optimization of multidisciplinary mechanical systems subject to uncertainties. *Probabilistic Engineering Mechanics*, 13(1):15–26. 49

[Onorato et al., 2022] Onorato, G., Proesmans, P., and Hoogreef, M. (2022). Assessment of hydrogen transport aircraft: Effects of fuel tank integration. *CEAS Aeronautical Journal*, 13(4):813–845. 13

[Palacios et al., 2013] Palacios, F., Colonno, M., Aranake, A., Campos, A., Copeland, S., Economon, T., Lonkar, A., Lukaczyk, T., Taylor, T., and Alonso, J. (2013). Stanford university unstructured (su2): An open-source integrated computational environment for multi-physics simulation and design. 69, 100

[Parr et al., 2012] Parr, J. M., Keane, A. J., Forrester, A. I., and Holden, C. M. (2012). Infill sampling criteria for surrogate-based optimization with constraint handling. *Engineering Optimization*, 44(10):1147–1166. 16, 39

[Pelamatti et al., 2019] Pelamatti, J., Brevault, L., Balesdent, M., Talbi, E.-G., and Guerin, Y. (2019). Efficient global optimization of constrained mixed variable problems. *Journal of Global Optimization*, 73:583–613. 151

[Pelamatti et al., 2024] Pelamatti, J., Le Riche, R., Helbert, C., and Blanchet-Scalliet, C. (2024). Coupling and selecting constraints in bayesian optimization under uncertainties. *Optimization and Engineering*, 25(1):373–412. 39

[Picheny, 2014] Picheny, V. (2014). A stepwise uncertainty reduction approach to constrained global optimization. In *Artificial intelligence and statistics*, pages 787–795. 16, 39

[Picheny et al., 2010] Picheny, V., Ginsbourger, D., Roustant, O., Haftka, R. T., and Kim, N. (2010). Adaptive Designs of Experiments for Accurate Approximation of a Target Region. *Journal of Mechanical Design*, 132(7). 30

[Pinkus, 1985] Pinkus, A. (1985). *N-Widths in Approximation Theory*. Ergebnisse der Mathematik und ihrer Grenzgebiete : a series of modern surveys in mathematics. Folge 3. Springer Berlin Heidelberg. 74

[Porrello et al., 2024] Porrello, C., Dubreuil, S., and Farhat, C. (2024). Bayesian framework with projection-based model order reduction for efficient global optimization. In *AIAA AVIATION FORUM AND ASCEND 2024*. 91

[Powell, 1970] Powell, M. J. D. (1970). A new algorithm for unconstrained optimization. In *Nonlinear programming*, pages 31–65. 26

[Powell, 1994] Powell, M. J. D. (1994). A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in Optimization and Numerical Analysis*, pages 51–67. Springer Netherlands. 27

[Priem et al., 2020a] Priem, R., Bartoli, N., Diouane, Y., and Sgueglia, A. (2020a). Upper trust bound feasibility criterion for mixed constrained bayesian optimization with application to aircraft design. *Aerospace Science and Technology*, page 105980. 39, 132

[Priem et al., 2020b] Priem, R., Gagnon, H., Chittick, I., Dufresne, S., Diouane, Y., and Bartoli, N. (2020b). An efficient application of bayesian optimization to an industrial MDO framework for aircraft design. In *AIAA AVIATION 2020 FORUM*. 15

[Prud'homme et al., 2001] Prud'homme, C., Rovas, D. V., Veroy, K., Machiels, L., Maday, Y., Patera, A. T., and Turinici, G. (2001). Reliable Real-Time Solution of Parametrized Partial Differential Equations: Reduced-Basis Output Bound Methods . *Journal of Fluids Engineering*, 124(1):70–80. 73

[Qian and Wu, 2008] Qian, P. Z. and Wu, C. J. (2008). Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. *Technometrics*, 50(2):192–204. 151

[Quarteroni and Valli, 1999] Quarteroni, A. and Valli, A. (1999). *Domain Decomposition Methods for Partial Differential Equations*. Numerical mathematics and scientific computation. Clarendon Press. 41

[Rasmussen and Williams, 2005] Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. The MIT Press. 15, 30, 31

[Regis, 2016] Regis, R. G. (2016). Trust regions in kriging-based optimization with expected improvement. *Engineering optimization*, 48(6):1037–1059. 36

[Rendall and Allen, 2008] Rendall, T. and Allen, C. (2008). Unified fluid-structure interpolation and mesh motion using radial basis functions. *International Journal for Numerical Methods in Engineering*, 74(10):1519–1559. 42, 99

[Sacks et al., 1989] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and Analysis of Computer Experiments. *Statistical Science*, 4(4):409 – 423. 15, 29

[Sasena et al., 2002] Sasena, M. J., Papalambros, P., and Goovaerts, P. (2002). Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering Optimization*, 34(3):263–278. 16, 39, 132

[Saves, 2024] Saves, P. (2024). *High dimensional multidisciplinary design optimization for eco-design aircraft*. PhD thesis, Université de Toulouse. 151

[Saves et al., 2024] Saves, P., Lafage, R., Bartoli, N., Diouane, Y., Bussemaker, J., Lefebvre, T., Hwang, J. T., Morlier, J., and Martins, J. R. (2024). Smt 2.0: A surrogate modeling toolbox with a focus on hierarchical and mixed variables gaussian processes. *Advances in Engineering Software*, 188:103571. 18, 62

[Scholten and Hartl, 2021] Scholten, W. and Hartl, D. (2021). Uncoupled method for static aeroelastic analysis. *Journal of Fluids and Structures*, 101:103221. 15

[Schonlau et al., 1998] Schonlau, M., Welch, W. J., and Jones, D. R. (1998). Global versus local search in constrained optimization of computer models. *Lecture notes-monograph series*, pages 11–25. 39

[Sellar et al., 1996] Sellar, R., Batill, S., and Renaud, J. (1996). *Response surface based, concurrent subspace optimization for multidisciplinary system design*. 34th Aerospace Sciences Meeting and Exhibit. 114

[Sgueglia et al., 2018] Sgueglia, A., Schmollgruber, P., Bartoli, N., Atinault, O., Benard, E., and Morlier, J. (2018). Exploration and sizing of a large passenger aircraft with distributed ducted electric fans. In *2018 AIAA Aerospace Sciences Meeting*. 13

[Shanno, 1970] Shanno, D. F. (1970). Conditioning of quasi-newton methods for function minimization. *Mathematics of Computation*, 24(111):647–656. 26

[Sirovich, 1987] Sirovich, L. (1987). Turbulence and the dynamics of coherent structures. i. coherent structures. *Quarterly of applied mathematics*, 45(3):561–571. 72

[Sobieszczanski-Sobieski, 1988] Sobieszczanski-Sobieski, J. (1988). *Optimization by decomposition: A step from hierarchic to non-hierarchic systems*. NASA Langley Research Center. 52

[Sobieszczanski-Sobieski et al., 2003] Sobieszczanski-Sobieski, J., Altus, T. D., Phillips, M., and Sandusky, R. (2003). Bilevel integrated system synthesis for concurrent and distributed processing. *AIAA Journal*, 41(10):1996–2003. 52

[Sobol', 2001] Sobol', I. (2001). Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and Computers in Simulation*, 55(1):271–280. The Second IMACS Seminar on Monte Carlo Methods. 48

[Squire and Trapp, 1998] Squire, W. and Trapp, G. (1998). Using complex variables to estimate derivatives of real functions. *SIAM Review*, 40(1):110–112. 26

[Sudret, 2008] Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964–979. 48

[Torczon, 1997] Torczon, V. (1997). On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 7(1):1–25. 28

[Tosserams et al., 2006] Tosserams, S., Etman, L., Papalambros, P., and Rooda, J. (2006). An augmented lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers. *Structural and multidisciplinary optimization*, 31:176–189. 52

[Touzé et al., 2021] Touzé, C., Vizzaccaro, A., and Thomas, O. (2021). Model order reduction methods for geometrically nonlinear structures: a review of nonlinear techniques. *Nonlinear Dynamics*, 105(2):1141–1190. 75

[Tsiolakis et al., 2020] Tsiolakis, V., Giacomini, M., Sevilla, R., Othmer, C., and Huerta, A. (2020). Nonintrusive proper generalised decomposition for parametrised incompressible flow problems in openfoam. *Computer Physics Communications*, 249:107013. 74

[Vapnik, 1999] Vapnik, V. (1999). *The Nature of Statistical Learning Theory*. Information Science and Statistics. Springer New York. 30

[Veroy et al., 2003] Veroy, K., Prud'homme, C., Rovas, D., and Patera, A. (2003). A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *16th AIAA Computational Fluid Dynamics Conference*. 73

[Wang et al., 2017] Wang, X., Li, M., Liu, Y., Sun, W., and Song, X. (2017). Surrogate based multidisciplinary design optimization of lithium-ion battery thermal management system in electric vehicles. *Structural and Multidisciplinary Optimization*, 56:1555–1570. 15

[Watson and Barnes, 1995] Watson, A. G. D. and Barnes, R. J. (1995). Infill sampling criteria to locate extremes. *Mathematical Geology*, 27(5):589–608. 36, 134

[Wong, 1967] Wong, Y. (1967). Differential geometry of grassmann manifolds. *Proceedings of the National Academy of Sciences*, 57(3):589–594. 76, 78

[Wu et al., 2003] Wu, C., Liang, Y., Lin, W., Lee, H., and Lim, S. (2003). A note on equivalence of proper orthogonal decomposition methods. *Journal of Sound and Vibration*, 265(5):1103–1110. 72

[Ye and Lim, 2016] Ye, K. and Lim, L.-H. (2016). Schubert varieties and distances between subspaces of different dimensions. *SIAM Journal on Matrix Analysis and Applications*, 37(3):1176–1197. 79

[Zhang et al., 2017] Zhang, M., Gou, W., Li, L., Yang, F., and Yue, Z. (2017). Multidisciplinary design and multi-objective optimization on guide fins of twin-web disk using kriging surrogate model. *Structural and Multidisciplinary Optimization*, 55:361–373. 15

[Zhang et al., 2019] Zhang, M., Li, H., and Su, S. (2019). High dimensional bayesian optimization via supervised dimension reduction. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, page 4292–4298. AAAI Press. 36, 151

[Zou et al., 2018] Zou, X., Conti, M., Díez, P., and Auricchio, F. (2018). A nonintrusive proper generalized decomposition scheme with application in biomechanics. *International Journal for Numerical Methods in Engineering*, 113(2):230–251. 74

Acronyms

ATC Analytical Target Cascading. 52

BFGS Broyden-Fletcher-Goldfarb-Shanno. 26

BLISS Bilevel Integrated System Synthesis. 52

C-EGMDO Constrained-EGMDO. 137–141, 144–146, 150, 151

CDF Cumulative Distribution Function. 34

CEI Constrained Expected Improvement. 133–135, 137, 139

CO Collaborative Optimization. 52

COBYLA Constrained Optimization by Linear Approximation. 27, 140

CSSO Concurrent Subspace Optimization. 52

CV Coefficient of Variation. 135, 136

DoE Design of Experiments. 15, 23, 29–35, 39, 46, 48, 54, 56, 57, 59, 76, 80, 81, 90, 91, 95, 101, 102, 112, 115, 133, 134, 138, 140, 144, 145, 154

DPOD+I Disciplinary Proper Orthogonal Decomposition and Interpolation. 16–18, 66, 80–84, 88, 97, 101, 102, 110, 119, 130, 149, 151

ECM Empirical Curbature Method. 71

ECSW Energy-Conserving Sampling and Weighting. 71

EGMDA Efficient Global Multidisciplinary Design Analysis. 45, 46, 48, 49, 54, 58, 61, 63, 80, 82, 83, 103, 110, 112, 115, 117, 119, 149, 154

EGMDO Efficient Global Multidisciplinary Design Optimization. 16–18, 24, 40, 54, 57–61, 63, 130, 132–135, 146, 149, 150

EGO Efficient Global Optimization. 15, 33–37, 54, 57, 62, 133

EI Expected Improvement. 33–36, 39, 57, 58, 133, 134, 145

EIM Empirical Interpolation Method. 71

FSI Fluid-Structure Interaction. 42

GNAT Gauss-Newton with Approximated Tensors. 71

GP Gaussian Process. 5, 15, 16, 23, 29–32, 39, 45–47, 50, 54–59, 61–63, 66, 71, 78, 80–84, 88–92, 94, 96, 98–107, 110, 113–116, 120–122, 126, 130–137, 139, 145, 146, 149–151, 154

IDF Individual Disciplinary Feasible. 50–52, 61, 63, 115, 116, 140

IDW Inverse Distance Weighting. 78

KKT Karush-Kuhn-Tucker. 38

KL Karhunen-Loève. 54–58, 72

LHS Latin Hypercube Sampling. 29, 36, 93, 95, 102, 103, 115, 138, 140, 144, 154

MDA Multidisciplinary Design Analysis. 7, 13–18, 24, 39–55, 60–63, 66, 79–84, 88, 90, 97, 99–107, 110–113, 115, 117–120, 122, 126, 127, 131, 139, 144, 149, 150, 153, 154

MDF Multidisciplinary Feasible. 50–52, 54, 60, 61, 63, 115, 116, 140

MDO Multidisciplinary Design Optimization. 14–18, 24, 39–41, 50–52, 54, 60–63, 82, 110, 112–118, 121, 122, 124, 126, 127, 130, 131, 135, 137, 138, 140, 146, 149, 150, 153, 154

MOR Model Order Reduction. 16, 66, 70, 75, 79, 83

PCA Principal Component Analysis. 72

PCE Polynomial Chaos Expansion. 8, 48, 50, 54–58, 131–133, 135–137, 139, 140, 144, 146

PDEs Partial Differential Equations. 18, 66, 69, 70, 79, 83, 99, 149

PDF Probability Density Function. 34, 55, 136

PGD Proper Generalized Decomposition. 71, 73, 74

pMOR projection-based Model Order Reduction. 66, 79

POD Proper Orthogonal Decomposition. 7, 16, 18, 71–76, 78–84, 88–107, 109, 110, 113, 119–121, 124, 127, 149–151

RBF Radial Basis Function. 30, 42, 78, 99

SAND Simultaneous Analysis and Design. 51

SBO Surrogate Based Optimization. 33

SEGO Super Efficient Global Optimization. 132

SEGOMOE Super Efficient Global Optimization with Mixture of Experts. 140

SLSQP Sequential Least Squares Programming. 38, 115, 121, 124, 140

SMT Surrogate Modeling Toolbox. 62, 98, 104, 154

SNOPT Sparse Nonlinear Optimizer. 38

SQP Sequential Quadratic Programming. 38

SVD Singular Value Decomposition. 72, 74, 77, 88

UCB Upper Confidence Bound. 34

UMDO Uncertainty Multidisciplinary Design Optimization. 49

VLM Vortex Lattice Method. 67–69, 141

WB2 Watson and Barnes 2. 36, 37, 134, 139, 146

WB2S Watson and Barnes 2 scaled. 36, 37

XDSM eXtended Design Structure Matrix. 40, 43, 44, 50–52, 153

Résumé

L'analyse et l'optimisation multidisciplinaire (MDAO) est un outil couramment utilisé dans les études préliminaires de conception d'aéronefs. Cependant, la plupart de ces études sont menées à l'aide de modèles de basse-fidélité associés aux différentes disciplines du système. Ces modèles, basées sur des données expérimentales, ne sont plus valables lors de l'étude de nouvelles configurations d'aéronefs pour lesquelles peu d'informations sont disponibles. Il est donc nécessaire de recourir à des modèles de haute-fidélité. Néanmoins, l'utilisation de solveurs haute-fidélité dans un contexte MDAO reste rare, du fait de leur coût de calcul et du manque d'information concernant les dérivées disciplinaires. Pour résoudre ce problème, les approches d'optimisation bayésienne, où l'optimisation est effectuée à l'aide des modèles de substitution, sont apparues comme des approches moins coûteuses qui ne nécessitent pas de connaître les dérivées du système. L'algorithme EGMDO (Efficient Global Multidisciplinary Design Optimization) est l'une de ces approches. EGMDO réduit le coût de calcul du problème d'optimisation multidisciplinaire en remplaçant les solveurs disciplinaires par des modèles de substitution basés sur les processus gaussiens (GP) enrichis de manière adaptative. Bien qu'EGMDO réduise avec succès le coût de calcul, il ne possède pas de stratégie de traitement des contraintes et n'est donc pas directement utilisable pour la plupart des études préliminaires. De plus, l'utilisation de modèles de substitution dans le cadre de solveurs haute-fidélité peut entraîner des difficultés supplémentaires. En effet, la sortie de ces solveurs est généralement une quantité discrétisée sur un maillage, pour laquelle des modèles de substitution ne peuvent pas être directement construits. Les stratégies de réduction d'ordre de modèle constituent une piste possible pour surpasser cette difficulté, car elles permettent de représenter les sorties disciplinaires à haute dimension dans un espace de plus faible dimension. La stratégie Disciplinary Proper Orthogonal Decomposition and Interpolation (DPOD+I) a développé cette idée en combinant des bases POD globales avec l'interpolation par GP des coordonnées généralisées. Mais bien que la stratégie DPOD+I ait permis l'utilisation de méta modèles avec des solveurs de haute fidélité, sa performance est significativement diminuée pour les problèmes où le nombre de vecteurs de base requis pour obtenir une petite erreur de projection est important. Cette thèse propose d'aborder les défis rencontrés par les méthodes EGMDO et DPOD+I en développant une extension de l'algorithme EGMDO aux problèmes d'optimisation sous contraintes et en proposant une stratégie de réduction de modèle basée sur l'interpolation de bases POD locales. Nous explorons également l'optimisation basée sur les gradients en exploitant les dérivées analytiques des méta-modèles disciplinaires. L'application des développements proposés au couplage aéroélastique d'une aile d'avion montre que l'approche proposée est capable de réduire le coût de calcul des problèmes d'analyse et d'optimisation multidisciplinaires souvent rencontrés lors des études préliminaires de conception d'aéronefs.

Mots-clés: Analyse et optimisation multidisciplinaire, méta-modèle, processus gaussien, réduction d'ordre de modèle, conception avion avant-projet

Abstract

Multidisciplinary design analysis and optimization (MDAO) is commonly used in preliminary aircraft design studies. Nevertheless, most of these studies are conducted using low-fidelity simulations of the different participating disciplines. With growing interest in new aircraft configurations for which little to no experimental data is available, the need for high-fidelity simulations increases. However, due to their computational cost and lack of derivative information, the use of high-fidelity solvers in MDAO remains scarce. To address this issue, Bayesian optimization approaches, where optimization is carried out using surrogates of some or all of the system's functions have risen as cost-saving, derivative-free approaches. One such approach is the Efficient Global Multidisciplinary Design Optimization (EGMDO) algorithm. EGMDO reduces the cost of the MDO problem by replacing the disciplinary solvers by adaptively enriched Gaussian Process (GP) surrogates. Although EGMDO successfully reduces the computational burden, it does not possess a constraint handling strategy and therefore is not directly useful for most preliminary design studies. Moreover, using disciplinary surrogates when dealing with high-fidelity solvers can lead to additional challenges. Indeed, the output of these solvers is typically some quantity discretized over a mesh, for which surrogate models cannot be directly built. Model order reduction strategies provide a possible lead to address this issue, as they allow to represent the high-dimensional disciplinary outputs in a lower dimensional space. The Disciplinary Proper Orthogonal Decomposition and Interpolation (DPOD+I) strategy developed this idea by combining disciplinary global POD bases with GP interpolation. But although the DPOD+I strategy successfully enabled the use of disciplinary surrogates with high-fidelity solvers, its performance is significantly decreased for problems where the number of basis vectors required to obtain a small projection error is large. This thesis proposes to address the shortcomings of both EGMDO and DPOD+I frameworks by developing an extension of the EGMDO algorithm to constrained optimization problems and proposing a different model reduction strategy for the DPOD+I approach, based on the interpolation of local POD bases. We equally explore the possibility of gradient-based optimization through exploitation of the analytical derivatives of the disciplinary surrogates. Application of the proposed developments to the aeroelastic coupling of a wing shows that the proposed approach is capable of reducing the computational cost of the multidisciplinary analysis and optimization problems often found in preliminary aircraft design studies.

Keywords: Multidisciplinary design analysis and optimization, surrogate models, Gaussian Process, model order reduction, preliminary aircraft design