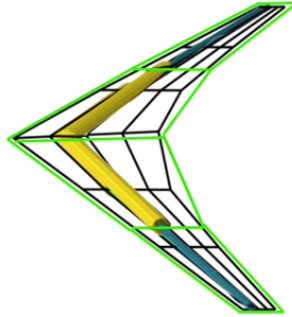# Multi-fidelity Methods for Improved Efficiency in Multi-disciplinary Optimization Problems

Margarida Cosme Gomes Pereira

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor: Prof. André Calado Marta
Dr. Thierry Lefebvre

Examination Committee

Chairperson: Prof. Filipe Szolnoky Ramos Pinto Cunha
Supervisor: Prof. André Calado Marta
Member of the Committee: Prof. Aurélio Lima Araújo

January 2021

To my parents.

Acknowledgments

Resumo

Uma das mais recentes abordagens para projetar sistemas complexos com precisão e tempo computacional praticável é o o algoritmo *Super Efficient Global Optimization* (SEGO) aplicado a multi-fidelidade. Além disso, muitos sistemas envolvem múltiplas disciplinas que devem ser consideradas durante o seu projeto. Como estas disciplinas influenciam-se diretamente informação entre as mesmas é trocada durante a otimização através de esquemas de transferência. Uma vez que os esquemas de transferência são uma parte significativa do tempo gasto numa otimização, novas configurações dos mesmos podem ser úteis para reduzir o tempo computacional do processo. Assim, este trabalho considera uma extensão de otimização multi-disciplinar para multi-fidelidade usando SEGO.

Neste trabalho, optimizamos uma asa de uma aeronave com o objetivo de consumir o mínimo combustível usando uma ferramenta de baixa fidelidade que considera modelos aerodinâmicos e estruturais, *OpenAeroStruct* (OAS). Verificamos que os esquemas de transferência do OAS não satisfazem o requisito de conservação. Desenvolvemos uma extensão dos esquemas de transferência para permitir que as disciplinas usem diferentes discretizações. Realizamos um estudo usando o algoritmo SEGO variando os níveis de fidelidade de um a três e usando duas formas de *Design of Experiments* onde verificamos que a melhor abordagem é a que usa dois níveis de fidelidade e amostras *nested*. Depois, a extensão dos esquemas de transferência é aplicada a um problema SEGO usando a melhor abordagem. Verificamos que resultados semelhantes são obtidos, mas com uma redução no tempo computacional.

Palavras-chave: Modelos *surrogate*, Otimização Bayesiana, *Efficient Global Optimization*, Projeto Aero-estrutural

Abstract

In the pursuit of designing complex systems accurately and with affordable computational cost, multi-fidelity Super Efficient Global Optimization (SEGO) is one of the most recent approaches. Furthermore, various systems involve multiple disciplines interaction that must be considered during the optimization. These disciplines directly influence each other, hence information between them needs to be exchanged during the optimization through transfer schemes. Since the transfer schemes are a significant part of the computational cost in the optimization process, new configurations can be used to try to reduce this time. Therefore, this work considers the extension of a multi-disciplinary optimization problem to multi-fidelity using SEGO.

An aircraft wing is desired to be optimized with the aim of consuming the minimum fuel during the flight mission. The problem is defined using a low fidelity multi-disciplinary tool, OpenAeroStruct (OAS), that considers models for the aerodynamics and structures. In this work, we verify that the transfer schemes implemented in OAS do not fulfill the conservation requirement. Then, an extension of the transfer schemes is developed to enable different disciplines discretization. Multi-fidelity SEGO with fidelity levels varied from one to three and two types of design of experiments is performed. We verify that the best approach is the one that uses two fidelity levels with only nested samples in the initial dataset. Then, the transfer schemes extension is employed in the best approach of multi-fidelity SEGO for the same problem. We verify that very similar results are obtained with a computational cost reduction.

x

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Contents

List of Tables

List of Figures

Nomenclature

Greek symbols

α Angle of attack.

β Basis function coefficients.

θ Hyperparameters, stRctural rotations.

δ Discrepancy function.

ϵ Residuals.

Γ Circulation.

μ Kriging model mean.

Ω Feasible domain.

Φ Cumulative distribution function.

ϕ Probability density function.

ρ Scaling factor, density.

σ Kriging model standard deviation.

σ_y Material yield strength.

σ_z Process standard deviation.

Roman symbols

A Virtual work.

c Costs, constraints.

C_D Coefficient of drag.

C_L Coefficient of lift.

C_T Specific fuel consumption.

d Dimensions.

$\mathbf{d}$	Structural displacements.
E	Young's Modulus.
f	Basis function, models of quantity of interest.
$\mathbf{F}$	Structural node force.
$\mathbf{f}$	Mesh loads.
G	Shear Modulus.
g	Gravitational constant.
h	Principal components, altitude.
L	Likelihood function.
M	Mach number.
m	Regression term.
$\mathbf{M}$	Structural node moment.
n	Samples.
$\mathbf{n}$	Normal Component.
p	Pressure.
R	Process kernel, residuals, range.
$\mathbf{r}$	Length.
S	Wing surface area.
$\mathbf{T}$	Aerodynamic load per unit area.
$\mathbf{u}$	Mesh displacements.
V	Flight speed.
W	Weight.
X	Sample data.
x, y, z	Cartesian components.
Y	Sample data responses.
y	Function of interest.
$\hat{y}$	Approximation of function of interest.
Z	Functional departure.

Subscripts

$1, 2$ Nodes indexes.

∞ Free-stream condition.

a Aerodynamic mesh.

cp Center of pressure.

i, j, k, b, k, n Computational indexes.

s StRctural mesh.

Acronyms

AD Aerodynamic Discretization.

AM Aerodynamic Mesh.

CRM Common Research Model.

DOE Design Of Experiments.

EGO Efficient Global Optimization.

EI Expected Improvement.

HF High Fidelity.

ISC Infill Sampling Criterion.

KPLS Kriging with Partial Least Squares.

LF Low Fidelity.

MDA Multi-Disciplinary Analysis.

MDAO Multi-disciplinary Design Analysis and Optimization.

MDF Multi-Disciplinary Feasible.

MDO Multi-disciplinary Design Optimization.

MF Medium Fidelity.

MFSEGO Multi-Fidelity Super Efficient Global Optimization.

NLBGS Non-Linear Block Gauss-Seidel.

OAS OpenAeroStruct.

PLS Partial Least Squares.

SD Structural Discretization.

SEGO Super Efficient Global Optimization.

SLSQP Sequential Least-Square Quadratic Programming.

SM Structural Mesh.

SQP Sequential Quadratic Programming.

WB2 Watson and Barnes.

Chapter 1

Introduction

1.1 Motivation

When designing systems, the objective of designers is to choose the project parameters that make the system satisfy its purpose in the best way possible. Taking an aircraft wing as example, the project parameters can be the angle of attack and the wing chord. In this case, we may choose this values in order to maximize the lift over drag ratio of the wing. This is known as the objective function to be optimized. To achieve the best design, one may consider physically modeling the system and testing its performance with different project parameters. This rapidly becomes unfeasible, therefore it is preferable to create a mathematical model of the system, feed it to a computer and simulate different project parameters until getting the optimal parameters. This process is known as optimization.

The next logic question is which model of the system to use? An initial naive answer would say to use the most accurate model. However, for many real world problems, a single simulation can take hours or even days to complete. As a result, design optimization becomes impossible with such models. To solve this issue, less accurate models, also known as low fidelity models are available. Recalling the wing example, a complex and accurate flow model for the wing aerodynamics would be the Direct Numerical Simulation and a low fidelity model would be the Reynolds-Average Navier-Stokes equations. Low fidelity models can have a very high degree of discrepancy from the true quantity of interest. A way of alleviating the prohibitive cost of high fidelity models without loosing so much accuracy is by constructing approximation models, also called surrogate models or metamodels, that mimic the behavior of the original model as closely as possible while being computationally cheaper to evaluate, as depicted in figure 1.1. The model is constructed based on a dataset of a limited number of intelligently chosen sample points. This sample points, that constitute the dataset, can be obtained using the most accurate model or can combine different models with different fidelity levels, enabling a cooperation between them. This way, multi-fidelity optimization emerged, combining different information sources to construct the surrogate model. Multi-fidelity optimization has proven to be more efficient than single fidelity optimization [2], since the combination of multiple information sources allow the model to better know the design space with less computational cost.
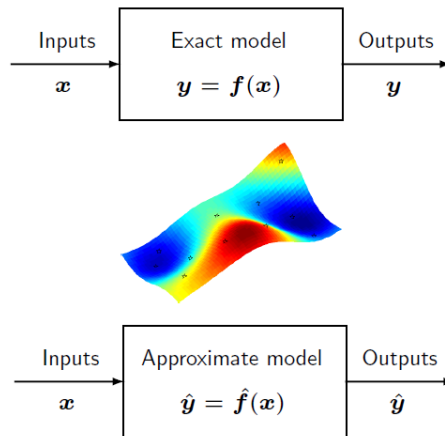
Figure 1.1: Surrogate: to model the model [1].

Another important topic when designing systems efficiently is the correct consideration of the multiple disciplines that interact with each other. Thereby, Multi-disciplinary Design Optimization (MDO) emerged. This field focuses on use of numerical optimization to perform the design of systems that involve a number of disciplines or subsystems [3]. Aerospace engineering due to its complexity and multi-disciplinary nature, was one of the first applications of MDO. As depicted in figure 1.2, aerodynamics, structures, propulsion and controls are some of the fields that interact when performing aircraft design. The aircraft designs obtained from the single discipline optimization for each discipline of figure 1.2 are very distinct and have a poor performance for the other disciplines that are not taken into consideration. The MDO allows obtain the best aircraft design that has a trade-off relationship between the multiple participating disciplines.
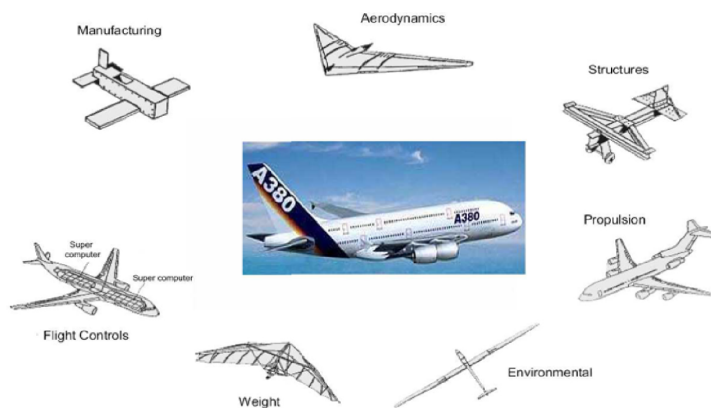
Figure 1.2: Multi-disciplinary domains in an aircraft design process [4].

1.2 Topic Overview

In this section, we overview surrogate models and Bayesian optimization. Moreover, aircraft design optimization and fluid-structure interaction are also approached.

1.2.1 Surrogate Models

A surrogate is an approximation model of a quantity of interest, cheap to evaluate, constructed on a finite set of evaluation samples called Design Of Experiments (DOE) [5]. Figure 1.3 illustrates a surrogate model constructed with three design points of the objective function. As referred in the motivation,
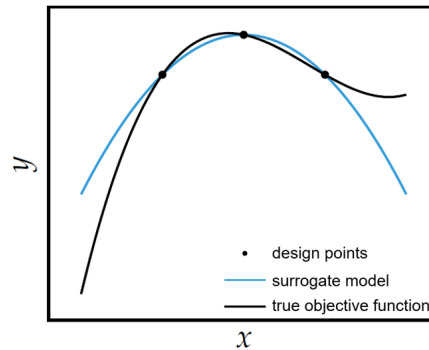
Figure 1.3: Surrogate model approximating the true objective function [6].

the surrogate can be constructed using samples from one or more information sources of the objective function, allowing the fusion of information between different fidelity levels. One of the popular choices to construct a surrogate is a Gaussian process [7], also referred as kriging [8]. Kriging models have been widely used in many optimization approaches, from multi-fidelity [9] to multi-disciplinary [10]. In section 2.1, we discuss in depth the making of DOE and single and multi-fidelity kriging models.

1.2.2 Bayesian Optimization

Bayesian optimization is an approach that uses an approximation of the quantity of interest, constructed based on a finite set of data, and determines what is the best sample to query per iteration to minimize (or maximize) a quantity of interest. This way, the surrogate model can be updated with the new information and becomes progressively better as the optimization takes place in the region where the minimum is supposed to be.

One particular Bayesian optimization algorithm is the Efficient Global Optimization (EGO), developed by Schonlau, Welch and Jones [11]. EGO uses a kriging model as approximation of the objective function, which provides access to the model uncertainty. This is an extremely convenient measure, useful to determine where the next sample should be queried and, in case of multi-fidelity optimization, which level of fidelity to use for that new sample. EGO has been extended to deal with constraint problems (SEGO) and multi-fidelity problems (MFSEGO). This methods will be discussed in detail in section 2.2.

1.2.3 Aircraft Design

As previously referred, the aircraft design involves the consideration of multiple disciplines. Traditionally, each of these disciplines are sequentially analyzed by an independent expertise team. However, these areas are tightly coupled and a change in the aircraft's design affects more than one of these

teams. This sequential approach optimization produces non-optimal results, which can be overpassed by the integrated optimization approach or MDO.
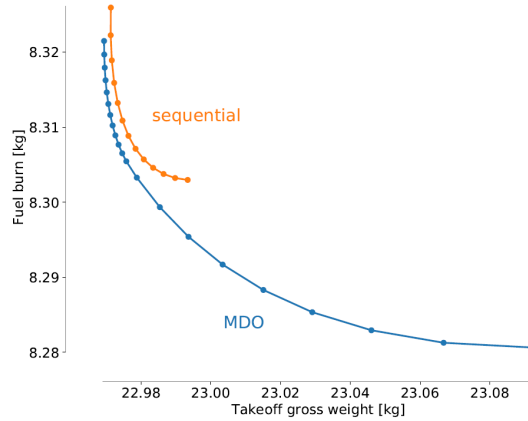
Figure 1.4: Comparison between Pareto fronts for sequential and multi-disciplinary optimization [12].

Figure 1.4 shows a comparison between sequential optimization and MDO using a Pareto front for fuel burn and takeoff gross weight. The objective is to minimize both quantities. A Pareto front is a set of optimal solutions, where a metric can not be improved without sacrificing at least other metric. Figure 1.4 demonstrates the superiority of the MDO compared to sequential approach, as all the design points computed by the MDO have a lower fuel burn for the same takeoff gross weight. Thus, all the solutions achieved using the sequential approach are dominated by the MDO. Another example comparing the performance of these two optimization approaches can be found in [13].

One of the most common applications of MDAO is coupled aerodynamic and structural optimization due to the strong iterations between these two disciplines using the Breguet range equation [14, 15] as metric. This quantity is given as

$$W_{\text{fuel}} = (W_s + W_0) \times \exp\Big(\frac{R.C_T}{V}\Big(\frac{CL}{CD}\Big)^{-1} - 1\Big), \tag{1.1}$$

where W_s is the wing structural weight and W_0 is the rest of the empty weight of the aircraft. In the exponent, R is the range, C_T is the specific fuel consumption, C_L is the lift coefficient, V is the flight speed and C_D is the drag coefficient. From equation (1.1), we see the aerodynamic and structural contribution to the fuel consumed during the flight mission. In this thesis, we perform aero-structural optimization using this metric.

1.2.4 Fluid-Structure Interaction

When performing aero-structural optimization, the two disciplines are tightly coupled. A change in the aerodynamic forces produces displacements in the structure and structural displacements deforms the aerodynamic surface producing different aerodynamic forces. Thereby, information between the two disciplines must be continuously exchanged during the optimization, through a fluid-structure interaction scheme. The transfer schemes are dependent on the models and the discretization of these models used to describe both disciplines.
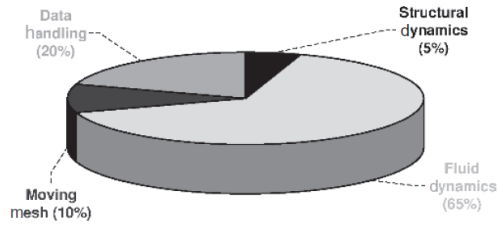
Figure 1.5: Percentage of computational cost per task for a hydrodynamics structural problem [16].

Figure 1.5 shows the computational cost percentage of the most demanding tasks for a hydrodynamics structural problem [16]. We observe that the computational cost is mostly driven by the fluid dynamics computations, but data handling is also demanding. This data handling refers to the data transfers between the two disciplines. The transfer schemes are therefore an important task in the process and the way they are implemented greatly affects the optimization, in terms of results and computational time. The ability to use different discretizations in both disciplines can be an interesting way to reduce the optimization time with slight or non performance damage.

1.3 Objectives and Deliverables

This thesis aims to explore the potential of extension of a multi-disciplinary optimization to multi-fidelity, as depicted in figure 1.6. Several researches have been developed in both topics. For instance, Meliani et al. [17] studied aerodynamic multi-fidelity EGO to optimize an airfoil shape using the high fidelity code ADflow and the low fidelity code Xfoil. The multi-disciplinary optimization topic has been widely studied. Aerodynamics and structures, two tightly coupled disciplines, were the first ones to be considered on a aircraft wing design [18, 19]. Then, MDO has been extended to complete aircraft optimization [20, 21]. In order to develop an optimization strategy that incorporates these two topics, some researches have been performed. Here, we highlight the work of Priem et al. [22] that implemented Bayesian optimization to an industrial MDO framework for aircraft design.
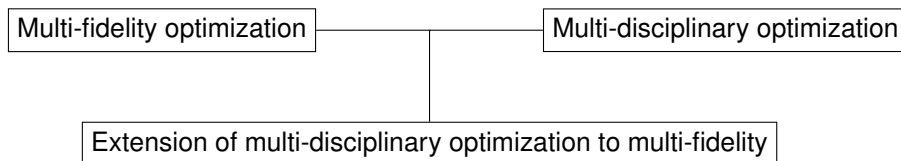
Figure 1.6: Simplified schematic of the two main thesis topics.

This is a recent research area, so it is of extreme importance to understand how this approach works using simple problems. This way, we will study the performance of multi-fidelity optimization using SEGO in an aircraft wing considering the aerodynamic and structural disciplines. To integrate the disciplines in the optimization, a low fidelity tool, OpenAeroStruct (OAS) [12], is used. This tool is responsible for both discipline analysis and it discretizes the aerodynamic discipline as a set of panels and the structural discipline as a set of finite element beams. The two discipline meshes have the same

spanwise discretization in the original OAS tool, as we will explore in section 4.3. Firstly, the OAS tool will be extended to work with non-identical spanwise discipline discretizations, in section 4.4. Then, a deep discussion on the performance of single and multi-fidelity optimization will be carried out. The number of fidelity levels used in the optimization is varied from one to three. The objective is to conclude which approach provides better results. After the results discussion, using the best approach previously identified and the same constraint problem, we will perform optimization using non-identical spanwise meshes to discretize aerodynamics and structures and conclude if any performance improvement is attained.

To summarize, this work aims to:

- Enhance the OAS tool to handle non-identical spanwise aerodynamic and structural mesh discretizations;

- Explore and compare SEGO results performed with single and multi-fidelity levels;

- Use the new transfer schemes to run SEGO with non-identical mesh disciplines discretization and compare with the previous results.

The main contributions of this thesis are the developed code and a detail study of the SEGO results with single and multi-fidelity levels and identical and non-identical spanwise disciple discretization.

1.4   Thesis Outline

This thesis is organized as follows. Firstly, in chapter 2, we describe in detail kriging models and EGO applied to single and multi-fidelity information sources.

In chapter 3, we gloss over some background of Multi-disciplinary Design Analysis and Optimization (MDAO) namely architecture, Multi-Disciplinary Analysis (MDA) approaches, optimization algorithms, aerodynamic and structural models and fluid-structure interaction.

After some theoretical background of the topics covered in the thesis scope, the wing mesh and the finite element structure is defined and the load and displacement transfer schemes implemented in OAS are introduced, in chapter 4. Then, the developed new transfer schemes for non-identical spanwise disciplines discretization are described.

In chapter 5, we start by defining the optimization problem along with the material properties, the flight conditions and other parameters. After the complete problem set up, we start by discussing the one fidelity level optimization results. The problem's complexity is then progressively incremented for two and three fidelity levels optimization. Then, the transfer schemes extension is employed in multi-fidelity SEGO using the same problem. We end this chapter by summing the overall conclusions.

Finally, in chapter 6, the work's achievements are summed up and suggestions for further developments are made.

Chapter 2

Surrogate Modeling and Bayesian Optimization

In this chapter, the mathematical fundamentals to perform multi-fidelity Bayesian optimization are discussed in detail. In section 2.1, the single and multi-fidelity model fitting procedures are explained. Then, in section 2.2, we consider the optimization techniques used to solve single and multi-fidelity surrogate based problems. Finally, in section 2.3, we present the Python toolboxes used to implement the previously described techniques.

2.1 Multiple Information Source Surrogate Modeling

A surrogate model, or metamodel, is a technique that mimics the behavior of the outcome of interest, constructed on a finite set of sample data while being computationally cheaper. This technique is convenient when the outcome of interest is too expensive to evaluate on an optimization process or sensitivity analysis, since multiple measures are required [23].
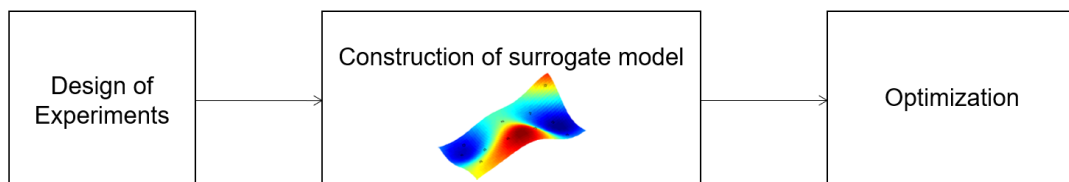
Figure 2.1: Key stages of the surrogate modeling optimization.

Figure 2.1 illustrates the key stages of the surrogate modeling approach. The first step is the construction of the DOE that can be obtained from different information sources, from computational simulations to experimental data. We will present the technique employed to build the DOE in section 2.1.1. Then, the surrogate construction is treated as a black box problem, where solely the input-output working is meaningful. The surrogate model construction is covered in sections 2.1.2, 2.1.3 and 2.1.4.

7

2.1.1 Design of Experiments

The DOE is the first step to construct a surrogate and it is a crucial stage, because it has a significant effect on the model's accuracy. In general, a good DOE should be 'space-filling', that is, the sample points should be spread over the entire design space as evenly as possible to capture the design behavior.

In particular, the Latin hypercube sampling is a random sampling technique, proposed by McKay et al. [24], widely used for engineering purposes due to its simplicity. For the sake of clarity, we present the most simple form of this sampling plan. Let n denote the required number of samples and d the number of random variables. An $n \times d$ matrix P, in which each of the d columns is a random permutation of 1,..., n, and an $n \times d$ matrix R of independent random numbers from the uniform (0, 1) distribution are established. The sampling plan is given as

$$S = \frac{1}{n}(P - R).$$ (2.1)

Each element of S, s_{ij}, is mapped according to its target marginal distribution as

$$x_{ij} = \Phi_{x_j}^{-1}(s_{ij}),$$ (2.2)

where $\Phi_{x_j}^{-1}$ is the inverse of the target cumulative distribution function for variable j. Now, vector $\hat{x}_i = [\hat{x}_{i1} \quad \hat{x}_{i2} \quad ... \quad \hat{x}_{id}]$, representing one sample, can be obtained.

An example of a Latin hypercube sampling for two dimensions and five samples is shown in figure 2.2. Note that for each column and for each row that partitions the design space there is one sample, a feature of this sampling plan [25].

$$P = \begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 3 \\ 4 & 1 \\ 5 & 5 \end{bmatrix} \quad R = \begin{bmatrix} 0.60 & 0.83 \\ 0.42 & 0.11 \\ 0.69 & 0.51 \\ 0.32 & 0.58 \\ 0.83 & 0.32 \end{bmatrix} \implies \quad S = \begin{bmatrix} 0.08 & 0.23 \\ 0.32 & 0.78 \\ 0.46 & 0.50 \\ 0.74 & 0.08 \\ 0.83 & 0.94 \end{bmatrix}$$

(a)   (b)
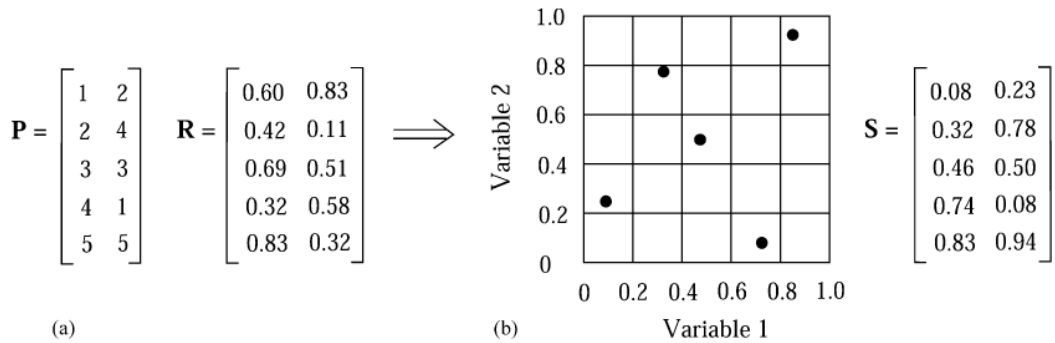
Figure 2.2: Latin hypercube sampling for two dimensions and five samples. The P and R matrices (a) determines the plan illustrated in (b) [25].

Although the previous example explores efficiently the design space, there is the risk of some undesired correlation of the sampling plan. In the last decades, researchers have tried to improve the space-filling properties of Latin hypercybe sampling, while managing the computational cost.

In this thesis, we use the Enhanced Stochastic Evolutionary algorithm designed by Jin et al. in 2005 [26] for building the Latin hypercube sampling. This algorithm improves the sampling process quality by maximizing the minimum distance between samples, referred to as the maximin distance optimality

criterion.

2.1.2   Surrogate Modeling via Kriging

Next, we present kriging metamodels following Sasena [23]. Most of the surrogate models rely on the assumption that the outcome of interest $y(\mathbf{x})$ can be expressed as $\hat{y}(\mathbf{x}) + \epsilon$, where $\hat{y}(\mathbf{x})$ is the model and ϵ are the residuals considered to be independent identically distributed normal random variables. The kriging model distinction is that the deviations, ϵ, are not independent, but rather a systematic function of the samples location.
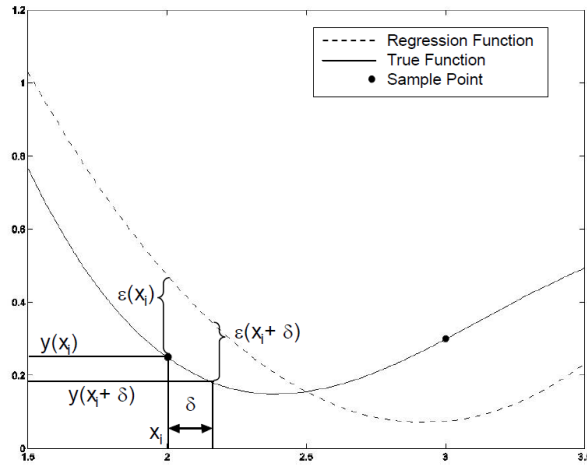
Figure 2.3: Correlation between errors $\epsilon(x)$ [23].

To illustrate the previous idea, figure 2.3 depicts a narrow region between two sample points of a quadratic function fitted by the least squares approach. At the sample point x_i, it is known the true function value $y(x_i)$ and the model value $\hat{y}(x_i)$ and therefore the error $\epsilon(x_i) = y(x_i) - \hat{y}(x_i)$. At $x_i + \delta$, for some small δ, it is only known the regression function value $\hat{y}(x_i + \delta)$. The kriging approach assumes that the smaller δ is, the more correlated the error $\epsilon(x_i + \delta)$ is with $\epsilon(x_i)$.

The kriging model assumes that the approximation of the outcome of interest $y(\mathbf{x})$ takes the form $\hat{y}(\mathbf{x}) = m(\mathbf{x}) + Z(\mathbf{x})$, constituted by two parts: a regression term (polynomial) $m(\mathbf{x})$ and a functional departure from that regression $Z(\mathbf{x})$. The regression term for the universal kriging can be written as

$$m(\mathbf{x}) = \sum_{j=1}^{r} \beta_j f_j(\mathbf{x}), \tag{2.3}$$

where $f_j(\mathbf{x})$ are the basis functions and β_j are the corresponding coefficients. In ordinary kriging, $m(\mathbf{x})$ takes the form of a constant, where $r = 1$ and $f_1 = 1$, leaving most of the prediction to $Z(\mathbf{x})$.

The functional departure $Z(\mathbf{x})$ is a Gaussian process with null mean and variance σ_z^2 ($\sim \mathcal{N}(0, \sigma_z^2)$) representing uncertainty in the mean of $\hat{y}(\mathbf{x})$. The corresponding covariance function can be expressed as

$$\text{cov}(Z(\mathbf{x}), Z(\mathbf{w})) = \sigma_z^2 R(\mathbf{x}, \mathbf{w}). \tag{2.4}$$

Here, σ_z^2 is a scalar factor known as the process variance, $\mathbf{x}$ and $\mathbf{w}$ are two design points in $\mathbb{R}^d$ and $R(\mathbf{x}, \mathbf{w})$ is the spatial correlation function, also known as process kernel. The choice of the kernel determines how the metamodel fits the data, that is, how quickly and how smoothly the function changes from point $\mathbf{x}$ to point $\mathbf{w}$.

One of the most commonly kernels used in kriging is the squared exponential correlation kernel,

$$R(\mathbf{x}, \mathbf{w}) = \exp(-\sum_{b=1}^{d} \theta_b(\mathbf{x}_b - \mathbf{w}_b)^2), \tag{2.5}$$

where $\boldsymbol{\theta} \in \mathbb{R}^d$ and $\boldsymbol{\theta} > 0$ is a vector of hyperparameters of the kriging model. Observing equation (2.5), it can be concluded that the function goes to zero as $(\mathbf{x}_b - \mathbf{w}_b)$ increases, showing that the influence of the sample data on the point to be predicted decreases as the distance between design points increases.

According to Forrester, Sóbester and Keane [27], the vector of hyperparameters $\boldsymbol{\theta}$ is a width parameter that affects how far that influence extends. For instance, for high values of $\boldsymbol{\theta}$, the correlation function is low and, consequently, there is a quick deterioration of the influence of the sampled data on prediction, that is, only data points near each other are well correlated. The vector of hyperparameters is fitted using a set of sample data $X = \{x_1, ..., x_n\}$ with $x_i \in \mathbb{R}^d$, with observed responses $Y = \{y_1, ..., y_n\}$ with $y_i \in \mathbb{R}$.

Once the kernel is defined, it can be proven [23] that the mean and variance of the kriging process can be expressed respectively as

$$\mu(\mathbf{x}) = f(\mathbf{x})'\boldsymbol{\beta} + r(\mathbf{x})'R^{-1}(Y - F\boldsymbol{\beta}) \tag{2.6}$$

and

$$\sigma^2(\mathbf{x}) = \sigma_z^2[1 - r(\mathbf{x})'R^{-1}r(\mathbf{x}) + (f(\mathbf{x})' - r(\mathbf{x})'R^{-1}F)(F'R^{-1}F)(f(\mathbf{x})' - r(\mathbf{x})'R^{-1}F)], \tag{2.7}$$

where $\mathbf{x} \in \mathbb{R}^d$ is the prediction point, R is the matrix of correlations among training points, $r(\mathbf{x})$ is the vector of correlations between the prediction point and the sample data, $\boldsymbol{\beta}$ is the vector of coefficients β_j presented in equation (2.3), F is the matrix of values of the regression basis function at the positions of the training points and $f(\mathbf{x})$ is the vector of values of these functions at the prediction point.

Parameters Estimation

With equations (2.6) and (2.7), one can compute an estimation of the mean and variance at a certain prediction point $\mathbf{x}$. However, the values of the parameters σ_z^2, $\boldsymbol{\theta}$ and β have to be estimated first. We follow Schonlau's approach [28] to show these parameters estimation. The estimation is based on maximizing the likelihood function, a function that measures the goodness of fit of a statistical model to Y given by

$$L(\mathbf{y}) = \prod_{i=1}^{n} \phi(y_i), \tag{2.8}$$

where ϕ is the probability density function of the process. By hypothesis, a Gaussian process distribution can be written as

$$\phi(\mathbf{y}; \boldsymbol{\theta}, \sigma_z^2, \boldsymbol{\beta}) = \frac{1}{\sqrt{2\pi\sigma_z^2}^2 \sqrt{\det(R)}} exp\Big(-\frac{1}{2\sigma^2}(\mathbf{y} - F\boldsymbol{\beta})' R^{-1}(\mathbf{y} - F\boldsymbol{\beta})\Big). \tag{2.9}$$

The opposite log-likelihood function is a preferable form to solve the problem, since the product of the density functions can quickly tend to zero, making the optimization process harder. This function is written as,

$$l(\mathbf{y}; \boldsymbol{\theta}, \sigma_z^2, \boldsymbol{\beta}) = -\frac{n}{2}(\ln(2\pi) + \ln(\sigma_z^2)) + \frac{1}{2}\det(R) + \frac{1}{2\sigma^2}(\mathbf{y} - F\boldsymbol{\beta})' R^{-1}(\mathbf{y} - F\boldsymbol{\beta}). \tag{2.10}$$

The vector and parameter $\boldsymbol{\beta}$ and σ_z^2 are determined by differentiating equation (2.10) w.r.t. $\boldsymbol{\beta}$ and σ_z^2 respectively yielding

$$\boldsymbol{\beta} = (F' R^{-1} F)^{-1} F' R^{-1} \mathbf{y} \tag{2.11}$$

and

$$\sigma_z^2 = \frac{(\mathbf{y} - F\boldsymbol{\beta})' R^{-1}(\mathbf{y} - F\boldsymbol{\beta})}{n}. \tag{2.12}$$

If we substitute $\boldsymbol{\beta}$ and σ_z^2 back into equation (2.10) and ignoring constant factors we obtain

$$l(\mathbf{y}; \boldsymbol{\theta}) = -\frac{1}{2}(n\ln(\sigma_z^2) + \det(R)). \tag{2.13}$$

The estimation of the vector of hyperparameters $\boldsymbol{\theta}$ is one of the major challenges when building a kriging model, in particular when dealing with problems with many dimensions or with a large number of sampling points. Inserting (2.11) and (2.12) into equation (2.13) we get

$$\begin{aligned} l(\mathbf{y}; \theta) = \\ -\frac{1}{2}\Big[n\ln\Big(\frac{1}{n}(\mathbf{y} - F(F' R^{-1} F)^{-1} F' R^{-1} \mathbf{y})' \times R^{-1}(\mathbf{y} - F(F' R^{-1} F)^{-1} F' R^{-1} \mathbf{y})\Big) + \ln(\det R)\Big]. \end{aligned} \tag{2.14}$$

The vector of hyperparameters are set by maximization of equation (2.14) w.r.t. $\boldsymbol{\theta}$ using the COBYLA optimizer [29], which stands for constrained optimization by linear approximation. This algorithm is gradient-free and it constructs successive linear approximations of the objective function and constraints via $d + 1$ points (in d dimensions) and optimizes these approximations in a trust region at each step.

2.1.3 Multi-fidelity Kriging

In this section, we are focus on kriging models constructed via multiple information sources, associated with a fidelity hierarchy, designated co-kriging. Let $f_1, f_2, ..., f_l$ be predictions of the quantity of interest y, where f_l has the best accuracy. We are interested in building a surrogate model for the y quantity using the knowledge of l information sources to enhance the model.

Relationship Enforcing

When studying physical problems, if a relationship between the information sources is known, it is enforced in the model, allowing learning the model with less data. Kennedy and O'Hagan [30] proposed a formulation that links the High Fidelity (HF) and Low Fidelity (LF) through a scaling factor ρ and a discrepancy function $\delta(\mathbf{x})$ as

$$f_{HF}(\mathbf{x}) = \rho f_{LF}(\mathbf{x}) + \delta(\mathbf{x}), \tag{2.15}$$

with $f_{LF}(\mathbf{x})$ and $\delta(\mathbf{x})$ independent. The ρ factor captures the correlation and scaling differences between the different fidelities into account. The discrepancy factor, $\delta(\mathbf{x})$, expresses other differences beyond scaling that cannot be captured by ρ.

Le Gratiet [31] gives a formal definition of the ρ factor,

$$\rho = \frac{\text{cov}(f_{HF}, f_{LF})}{\text{var}(f_{LF})} = \frac{\text{std}(f_{HF})}{\text{std}(f_{HF})} \frac{\text{cov}(f_{HF}, f_{LF})}{\text{std}(f_{LF})\text{std}(f_{LF})} =$$
$$\frac{\text{std}(f_{HF})}{\text{std}(f_{LF})} \frac{\text{cov}(f_{HF}, f_{LF})}{\text{std}(f_{LF})\text{std}(f_{HF})} = \frac{\text{std}(f_{HF})}{\text{std}(f_{LF})} \text{corr}(f_{HF}, f_{LF}), \tag{2.16}$$

where std(.) is the standard deviation and corr(.) is the correlation.

Implementation

Le Gratiet [31] proposed an implementation of the Kennedy-O'Hagan's multi-fidelity formulation that modifies the regression term from the kriging model presented in equation (2.3) to take the lower fidelity model as a basis function

$$m(\mathbf{x}) = \sum_{j=1}^{r} \beta_j f_j(\mathbf{x}) + \beta_\rho f_{LF}(\mathbf{x}), \tag{2.17}$$

where β_ρ is an estimation of ρ obtained via the maximization of the likelihood function as presented in section 2.1.2.

Based on equation (2.15), the mean and variance of a high fidelity model (only two levels) can be expressed respectively as

$$\mu_{HF} = \rho \mu_{LF} + \mu_\delta \tag{2.18}$$

and

$$\sigma_{HF}^2 = \rho^2 \sigma_{LF}^2 + \sigma_\delta^2. \tag{2.19}$$

Le Gratiet's learning is performed by firstly learn the lowest fidelity μ_{LF} and σ_{LF}^2, then learn the relationship between the two consecutive fidelity levels, that is, the scaling factor ρ and the discrepancy function $\delta(.)$. This approach can be applied to l levels of fidelity, therefore equations (2.18) and (2.19) can be rewritten as

$$\mu_k = \rho_{k-1} \mu_{k-1} + \mu_{\delta_k} \tag{2.20}$$

and

$$\sigma_k^2 = \rho_{k-1}^2 \sigma_{k-1}^2 + \sigma_{\delta_k}^2. \tag{2.21}$$

Le Gratiet's formulation requires the DOE to be closed-form. The nested DOE states that $X_l \subseteq X_{l-1}... \subseteq X_1$, where X_k represents the sample data to train the model of fidelity level 1 to l, being l the highest level. This formulation offers an expression of the uncertainty contribution of the fidelity level k at design point $\mathbf{x}$ to the model (corrected from page 163, [31]) as

$$\sigma_{\text{cont}}^2(k, \mathbf{x}) = \sigma_{\delta_k}^2(x) \prod_{j=k}^{l-1} \rho_j^2. \tag{2.22}$$

Equation (2.22) shows that the uncertainty of the fidelity level k is computed by the variance contribution of the same level scaled using ρ_j until we get to the highest fidelity level.

2.1.4 Kriging Model combined with Partial Least Squares

The kriging model has become extremely popular because of its ability to accurately mimic computationally expensive functions and capability to estimate the error of the prediction. However, the standard kriging model can be computationally expensive when the number of sample points n is large, since the matrix of correlation R of size $(n \times n)$ must be inverted in equation (2.14). Moreover, for highly dimensional problems, more hyperparameters $\boldsymbol{\theta}$ must be estimated, consequently the matrix R must be inverted multiple times.

To lighten this load, we use Partial Least Squares (PLS) to construct a new kernel that depends on only a few parameters rather than d dimensions. The resulting method is called Kriging with Partial Least Squares (KPLS). According to [32], PLS is a statistical method that projects input variables onto a new space, called principal components, to find a linear relationship between input variables and output variable. The principal components are a linear combination of the input variables. Usually, the number of principal components h is between one and four, according to what is deemed enough to explain a 'majority' of data variations. Figure 2.4 illustrates a simple three-dimensional problem ($d = 3$), where the PLS approach is employed to reduce the problem dimension to two principal components ($h = 2$).
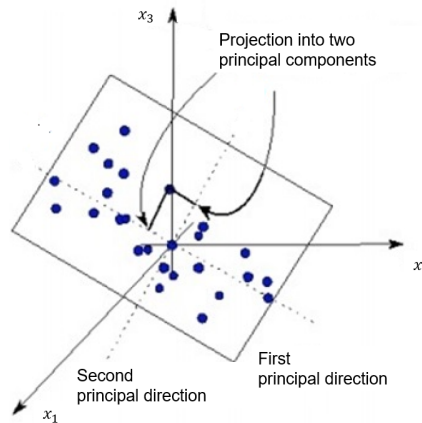
Figure 2.4: Partial least squares construction of two principal components [32].

The principal components are computed by searching the direction that maximizes the squared co-variance between that direction and the output variable. By searching the principal components, one obtains the matrix of the weights that transcribes the influence of the initial variables to the output variable, trimming the dimension from d to h. Thereby, a new kernel of dimension h is defined, reducing the problem dimension and making the surrogate model construction more efficient and faster.

The same concept is applied to co-kriging, only performing the PLS analysis step on high fidelity to preserve the robustness to poor correlations.

2.2 Bayesian Optimization

Let us now introduce the optimization techniques used to solve single and multi-fidelity surrogate based problems.

In 1975, Močkus [33] defined Bayesian optimization as an optimization technique based upon the minimization of the expected deviation from the extremum of the studied function. The objective function is treated as a black box function. This optimization approach places a prior (probability distribution expressing our beliefs) over the objective through a surrogate model which is constructed with an initial set of samples. The prior captures an estimated behavior of the function. This estimation is used to construct an acquisition function, often also referred as Infill Sampling Criterion (ISC), that determines what is the most appropriate query point to add.

Bayesian optimization is particularly advantageous for a multi-modal objective or if the objective function is expensive to evaluate. Let us take an example of gold mining inspired by Krige [34]. Consider that it is desired to mine for gold in an unknown area. Naturally, our goal is to find the location in the land with the maximum gold to know where we should mine. We can learn the gold distribution by drilling at different locations. However, drilling is costly, consequently we must find this location in order to execute the minimum number of drills. Bayesian optimization can be used to propose new location candidates with high promise of gold content.

2.2.1 Efficient Global Optimization (EGO)

In 1998, Jones [11] used Gaussian processes with the expected improvement function as ISC to successfully developed the EGO algorithm. Since EGO's publication, several researches have been developed to extend the algorithm. In particular, adding constraint optimization [23, 35], increasing the optimization efficiency [32] and using mixture of experts [36].

Next, we describe EGO following Jones [11]. Let y be the objective function. We start by getting a set of samples $X = \{x_1, x_2, ..., x_n\}$ yielding the responses $Y = \{y_1, y_2, ..., y_n\}$. Using this set of samples, a kriging model is built with a mean and variance function μ and σ^2 respectively, as described in section 2.1.2. Figure 2.5 shows a surrogate model, the error estimation depicted in (b) is computed through the normal function estimation in the domain, depicted in (a) at $x = 0.5$.

The information provided by the kriging model is used to establish the Expected Improvement func-

(a) Normal function estimation at $x = 0.5$ (b) SM with error estimation
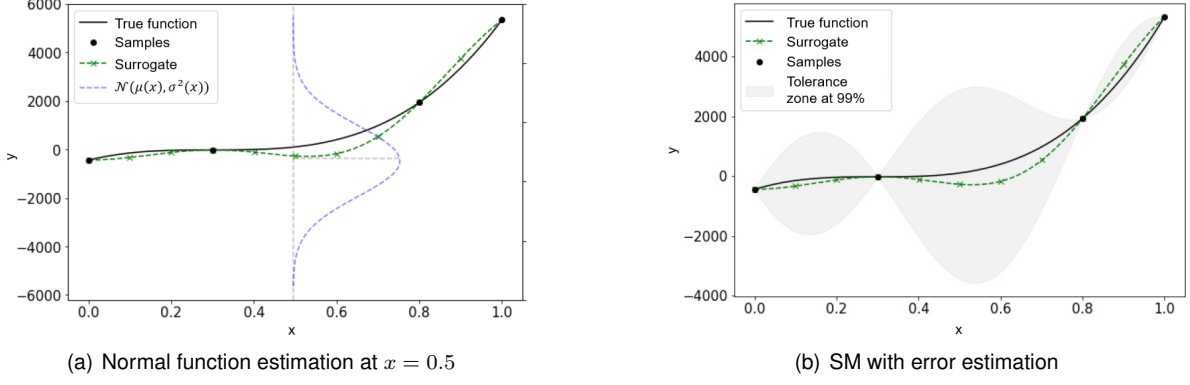
Figure 2.5: Kriging model [37].

tion (EI), the employed ISC in EGO. This function will determine the next point to query. First, let

$$y_{\mathsf{min}} = \mathsf{min}\{y_1, y_2, ..., y_n\} \tag{2.23}$$

be the current best function value. The EI is simply given by

$$E[I(\mathbf{x})] = E[\mathsf{max}(y_{\mathsf{min}} - Y(\mathbf{x}), 0)], \tag{2.24}$$

where $Y(\mathbf{x})$ is a random variable following the distribution $\mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$.

By expressing the right-hand side of equation (2.24) as an integral, and applying some tedious integration by parts, one can express the EI in closed form,

$$E[I(\mathbf{x})] = \underbrace{(y_{\mathsf{min}} - \mu(\mathbf{x}))\Phi\left(\frac{y_{\mathsf{min}} - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right)}_{\text{Exploitation}} + \underbrace{\sigma(\mathbf{x})\phi\left(\frac{y_{\mathsf{min}} - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right)}_{\text{Exploration}}, \tag{2.25}$$

where $\Phi(.)$ and $\phi(.)$ are respectively the cumulative distribution function and the probability density function of $\mathcal{N}(0, 1)$. Then, we determine the next point to be sampled as

$$\mathbf{x}_{n+1} = \underset{\mathbf{x}}{\mathsf{argmax}}\,(E[I(\mathbf{x})]), \tag{2.26}$$

and the respective response y_{n+1} is computed. This process is repeated for a certain number of times. When performing EGO, it is required to provide a budget, that is, a number that will influence the number of samples added to the model and, consequently the computational time of the optimization process. Generally, the budget is related with the computational cost to query a function's sample. Adding a new sample to the surrogate model is associated with a numerical cost. As the optimization proceeds, new samples are added to the model and the budget is consumed. The optimization will run until the budget is reached.

After presenting the steps of EGO, let us explore its properties to better understand how it works and how can it be extended to a multi-fidelity problem. The first property is the enrichment acquired by the model as the optimization process takes place, since the more samples we have, the better the

model approximates the objective. The second property is the exploration/exploitation trade-off when it comes to choose the next sampling point. Equation (2.25) is a balance between seeking promising areas of the design space (exploitation) and choosing something from where we can better learn the design space (exploration). The term relative to Φ is large when $\mu(\mathbf{x})$ is small with respect to y_{min}, which promotes exploitation. On the other hand, the term relative to ϕ is large when $\sigma(\mathbf{x})$ is large, promoting exploration. Thus, in areas where the kriging model is worse than the current best y_{min}, the EGO may perform exploration. In areas where the kriging model is better than the current best, the EGO may exploit. Note that $E[I] = 0$ when $\sigma(\mathbf{x}) = 0$, consequently there is no possibility of sample a point that was already chosen.

To demonstrate EGO's performance, let us observe figure 2.6 that shows a simple one dimensional multi-modal example. The dashed line represents the true objective function, while the solid line is the kriging approximation. The circles show the sample points in each stage of the optimization. The plot at the bottom represents the EI. Initially the kriging is constructed using four sample points, resulting in a model poorly fitted when compared with the true function. The first samples to be plotted are on the right side, where the model uncertainty is highest. Between iteration two and four, the model queries another few points in that region where there is a high probability that a better point can be found. After four iterations, the right region is explored, but the uncertainty of the model on the left region forces some samples to be plotted. Finally, by the sixth iteration the local minima is achieved in the right region and the kriging model mimics the true function closely.

a) Initialization   b) Iteration 2

c) Iteration 4   d) Iteration 6
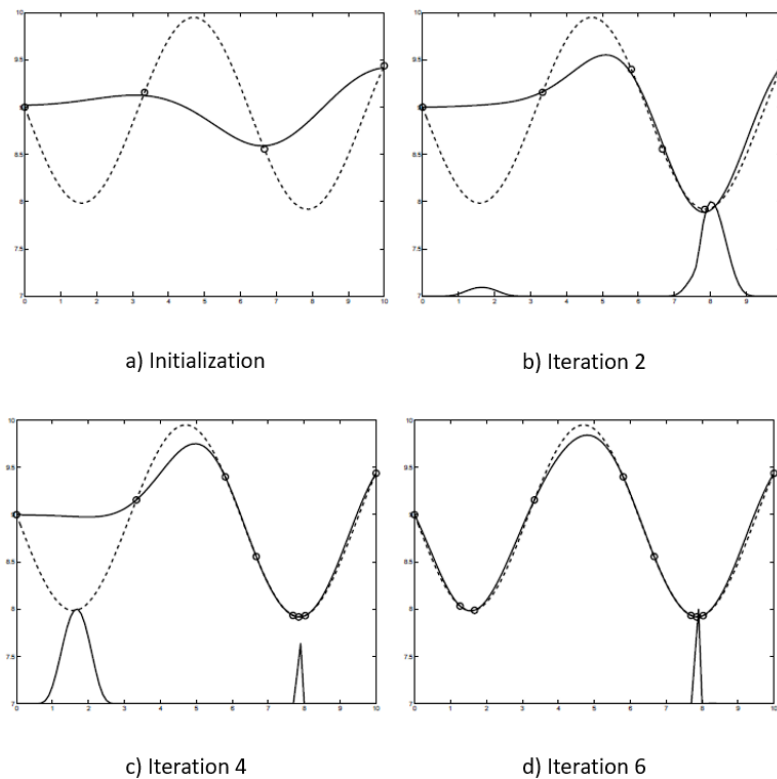
Figure 2.6: EGO example. Dashed line is the true function, solid line is the kriging model, circles are the sample points and the bottom plot is the EI [23].

More sophisticated ISC functions have been developed over the EI criterion, such as Watson and

Barnes (WB2) criterion [38] formulated as

$$WB2(\mathbf{x}) = -\mu(\mathbf{x}) + E[I(\mathbf{x})] \tag{2.27}$$

As for the EI criterion, the next sampling point is obtained by the maximization of (2.27) employing the SLSQP optimizer, which will be presented in section 3.4.

Figure 2.7 shows the EI and WB2 criterion for the same situation. The magnitude of EI is expected to decrease during the iterative process, as the surrogate model becomes more accurate in the promising areas of the design space. So, for high dimensional problems, where some areas have a high uncertainty, the EI can suffer from an excess of exploration on those areas. To address this issue, the WB2 criterion adds the term $-\mu(\mathbf{x})$, avoiding an excess of exploration of uncertainty areas [36]. In addition, Sasena [23] advised to use this criterion after comparing several infill sampling criteria with various test cases. Thereby, in this dissertation we use the WB2 criterion as ISC.

(a) EI criterion   (b) WB2 criterion
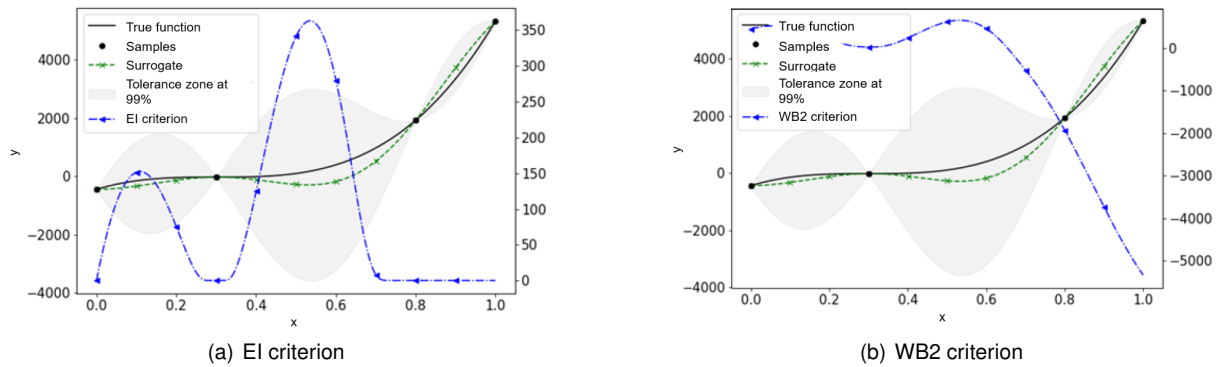
Figure 2.7: Infill sampling criteria [37].

2.2.2 Super-Efficient Global Optimization (SEGO)

One of the major drawbacks with EGO is that it only handles unconstrained problems. To address this issue, Michael Sasena [23] proposed an extension of EGO called SEGO.

As in the EGO approach, the objective function y is approximated by a kriging model. Additionally, we construct a surrogate for the m nonlinear constraints. The WB2 function is used to determine the next sampling point

$$\max_{\mathbf{x} \in \Omega_f} \quad WB2(\mathbf{x}), \tag{2.28}$$

where the feasible domain Ω_f is defined by the kriging model of the nonlinear constraints

$$\Omega_f = \{\mathbf{x} \in \mathbb{R}^d : \hat{c}_1(\mathbf{x}) \leq 0, ..., \hat{c}_m(\mathbf{x}) \leq 0\}. \tag{2.29}$$

The choice of the next sample is only driven by the $WB2$ of the objective function. Therefore, the kriging models of the constraints should be accurate in order to not compromised the solution's accuracy. At each iteration, the new point to be sampled may not be a feasible point, because only the mean value

of the kriging model $\hat{c}_i$ is used to approximate the constraints. Even if the new point is not feasible, evaluating the true functions adds information to the model. To consider the associated error estimation of the constraints in a Bayesian optimization, various approaches are possible, as expected violation [39] or predictive entropy search with constraints [40]. This approaches will not be handle in this dissertation.

We summarize the SEGO steps in figure 2.8.
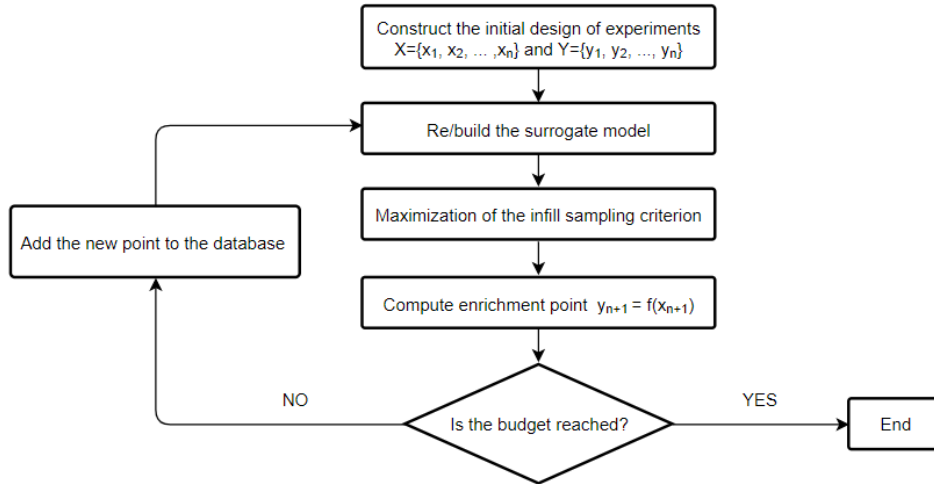
Figure 2.8: SEGO diagram.

The first step is to construct the initial DOE. Then, the surrogate model is built based on this DOE and the maximization of the ISC is computed to obtain the next sample to add to the data set. If the budget has not yet been reached, the point is added to the data set and the kriging model is rebuilt. The process is repeated until the budget is exhausted.

In the EGO, the surrogate building and the maximization of the ISC are optimization problems themselves that can be computationally expensive. Other optimization methods such as gradient-based methods or genetic algorithms require very little computational effort where to evaluate the function next. However, the number of functions evaluations to converge to a solution is large compared to EGO. This approach converges more rapidly because it uses as much information as possible when determining the next point to query. Thus, EGO is ideal for expensive functions, when the designer can not afford to perform a large number of functions evaluations, as the mining gold example referred at the beginning of this section [23].

2.2.3 Multi-fidelity Super-Efficient Global Optimization (MFSEGO)

In this section, we extend the SEGO algorithm to multi-fidelity problems. The new algorithm is named Multi-Fidelity Super Efficient Global Optimization (MFSEGO).

Following the approach presented by Meliani [17], the MFSEGO algorithm handles with a two-stage decision process: (1) the search of the most promising sample and (2) the choice of level of enrichment of this sample. The first problem is tackled by the ISC the same way as presented in section 2.2.1. The second problem $\hat{c}_i$ analysis the uncertainty of the chosen point and decides the fidelity level where

this point should be queried. The choice of the fidelity level of enrichment favors the use of low fidelity samples for exploration and high fidelity samples for exploitation.

Let $f_0, ..., f_l$ be the lowest to highest fidelity model of the objective function, with the respective querying costs $c_0, ..., c_l$. Recalling the recursive formulation of Le Gratiet introduced in section 2.1.3, we known that fidelity levels relate as

$$f_k = \rho_{k-1} f_{k-1} + \delta_k \quad \text{for} \quad k \in \{1, ..., l\} \tag{2.30}$$

$$\rho_{k-1} = \text{corr}(f_k, f_{k-1}) \frac{\text{std}(f_k)}{\text{std}(f_{k-1})} \tag{2.31}$$

$$\sigma_k^2 = \rho_{k-1}^2 \sigma_{k-1}^2 + \sigma_{\delta k}^2. \tag{2.32}$$

Recalling equation (2.22) from section 2.1.3, the variance contribution of the fidelity level k at design point $\mathbf{x}^*$ is given by

$$\sigma_{\text{cont}}^2(k, \mathbf{x}^*) = \sigma_{\delta k}^2(k, \mathbf{x}^*) \prod_{j=k}^{l-1} \rho_j^2. \tag{2.33}$$

Due to the necessity of nested DOE, the fidelity levels lower than k of the sampling point $\mathbf{x}^*$ are enriched as well so the uncertainty reduction becomes

$$\sigma_{\text{red}}^2(k, \mathbf{x}^*) = \sum_{i=0}^{k} \sigma_{\delta i}^2(k, \mathbf{x}^*) \prod_{j=k}^{l-1} \rho_j^2. \tag{2.34}$$

The corresponding cost of the enrichment through fidelity level k yields

$$\text{cost}_{\text{total}}(k) = \sum_{i=0}^{k} c_i. \tag{2.35}$$

Mostafa [17] proposed the following enrichment level criterion

$$t = \underset{k \in (0, ..., l)}{\text{argmax}} \quad \frac{\sigma_{\text{red}}^2(k, \mathbf{x}^*)}{\text{cost}_{\text{total}}(k)^2 + \text{offset}}, \tag{2.36}$$

where t is the highest fidelity level to be added and offset is the scalar that takes into account the cost of building the surrogate model and maximization of the ISC.

After presenting the MFSEGO, we now explore its properties. The MFSEGO algorithm only considers the high fidelity information to update the value of y_{min}, presented in equation (2.23). Other fidelity levels are only consider to help reducing some amount of the ISC, that is, the model's uncertainty (exploration) rather than effectively minimize the objective (exploitation). Thus, the algorithm ensures that the problem converges to a global minimum of the high fidelity function. Note that the enrichment criterion, presented in equation (2.36), integrates the correlation between fidelity levels. When the correlation is poor, $\rho_k^2 \to 0$, then $\frac{\sigma_{\text{red}}^2(k, \mathbf{x}^*)}{\text{cost}_{\text{total}}(k)^2 + \text{offset}} \to 0$, pushing the algorithm to higher fidelities. The algorithm will not resample high fidelity samples as ISC at these is zero. In addition, it will not resample low fidelity samples as the enrichment criterion is zero and the algorithm is prompt to highest levels.

Similarly to the EGO's problem presented in the previous section, let us introduce a one dimensional

toy problem with two fidelity levels to demonstrate MFSEGO's performance [41]. The high and low fidelity functions are given as

$$\begin{cases} f_{HF}(x) = (6x - 2)^2 \times sin(2(6x - 2)) \\ f_{LF}(x) = 0.5 f_{HF}(x) + 10(x - 0.5) - 5. \end{cases} \tag{2.37}$$

The cost ratio between the fidelity levels is assumed to be $1/1000$.
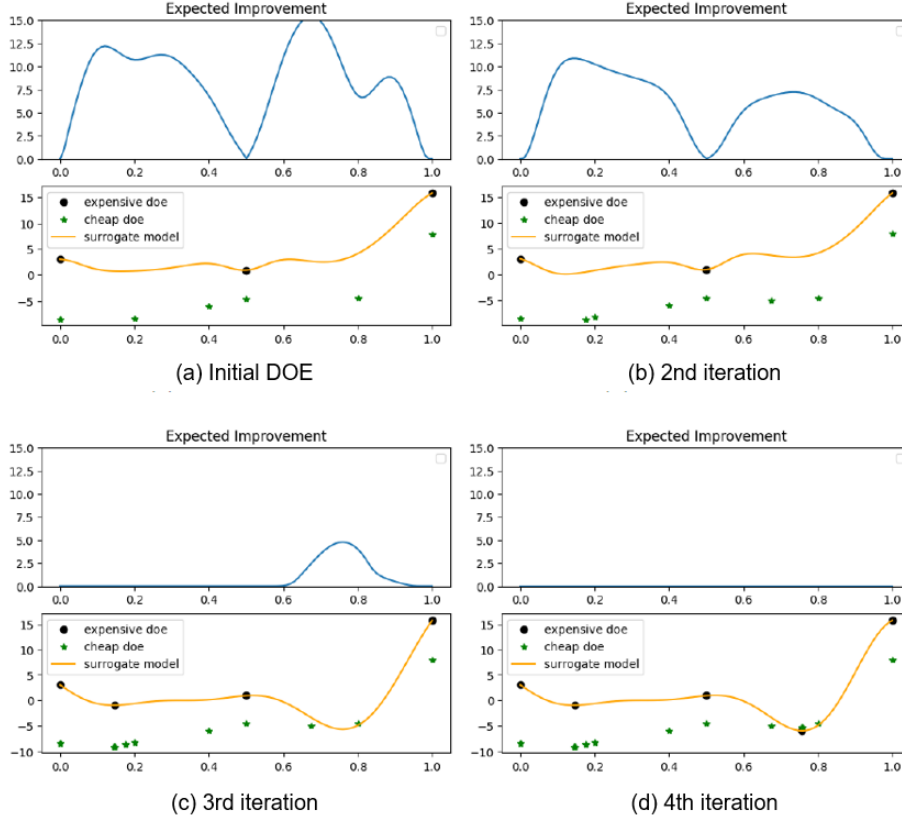
Figure 2.9: Evolution of EI criterion and kriging model throughout MFSEGO iterations [41].

Figure 2.9 shows the evolution of the EI criterion and the surrogate model throughout the optimization process. We start the optimization with three high fidelity samples and six low fidelity samples, as illustrated in figure 2.9 (a). After adding two low fidelity samples to explore the design space and reduce the uncertainty of the EI, at the third iteration, shown in figure 2.9 (c), one high fidelity sample is queried for local exploitation. Then, the next high fidelity sample added in the forth iteration finds the global optimum of the function.

To summarize, figure 2.10 schematizes the key stages of the MFSEGO algorithm. The general procedure of MFSEGO, depicted in figure 2.10 (a), is similar to the SEGO procedure. The offset computation is performed through the elapsed time between surrogate construction and maximization of the $WB2$ criterion. As MFSEGO evolves, the number of sample points increases, consequently the time to performed these two steps also increases. The enrichment level, where the algorithm computes the response y_{n+1} of the next sample to be added to the data set, is more complex and it is schematized below in figure 2.10 (b). The response y_{n+1} is computed for the lowest fidelity level function. The enrichment criterion of two consecutive levels is compared and if the criterion of the consecutive highest

(a) MFSEGO general procedure
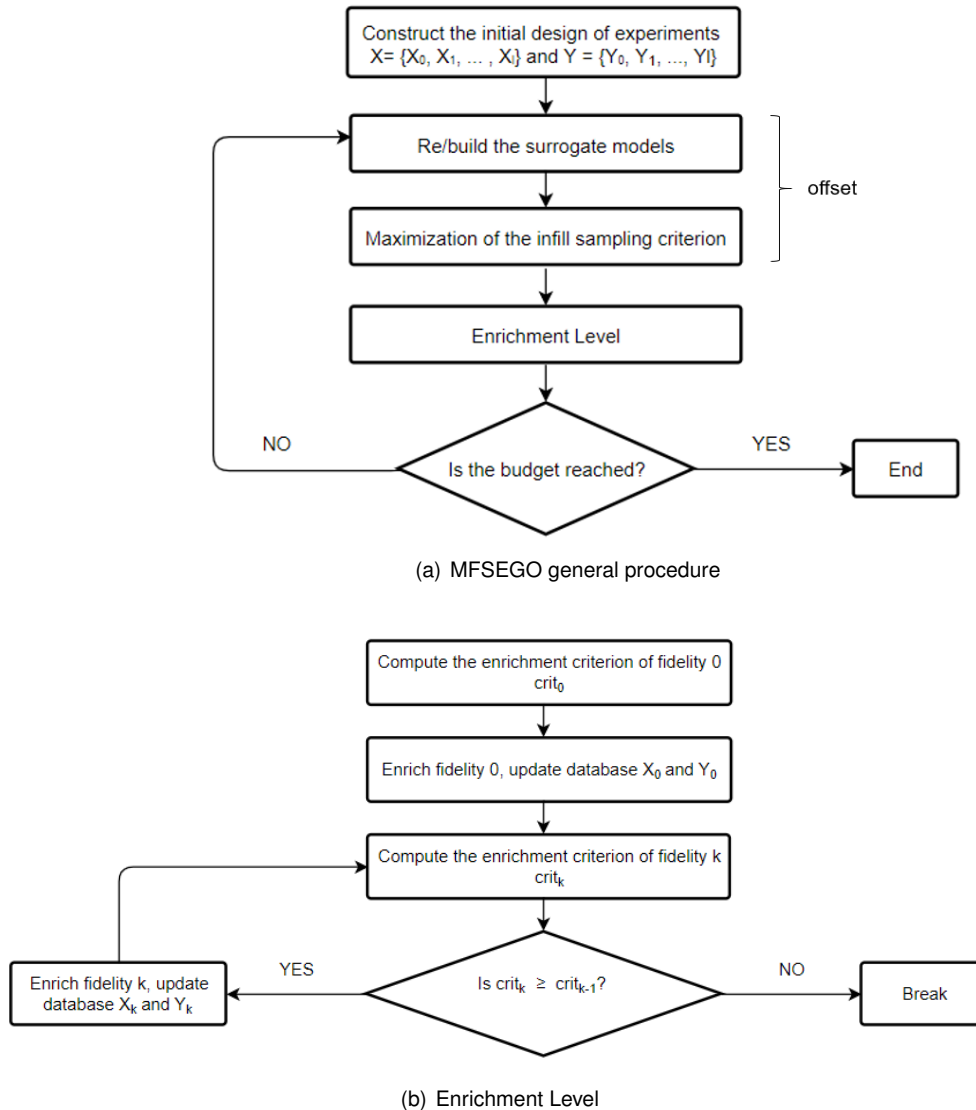
(b) Enrichment Level

Figure 2.10: MFSEGO algorithm.

fidelity level k is larger than the fidelity level $k-1$, the response is also computed for fidelity level k. This process is repeated until the enrichment criterion of fidelity level k is smaller than the one of fidelity $k-1$. Then, the surrogate model is rebuilt using the extra information added to the data base.

2.3 Surrogate Modeling Toolboxes

The Surrogate Modeling Toolbox (SMT) is a Python package developed by ISAE-Supaero, ONERA, NASA and the University of Michigan that facilitates the use of surrogate models and the implementation of additional methods. It provides multiple surrogate models, for instance radial basis functions, inverse-distance weighting and kriging. Sampling techniques, such as random sampling and the Latin hypercube sampling using the Enhanced Stochastic Evolutionary algorithm, are available, as well as multiple benchmarking problems, such as the Branin and Rosenbrock function. It includes the kriging

model using partial-least squares approach for single and multi-fidelity (KPLS, MKPLS) [42]. An example of how to use the SMT package to construct a kriging model is shown in the Python script below. The kriging model is plotted in figure 2.11. The DOE to train the model is defined in xt and yt vectors and the theta is set to 1×10^{-2}.

```python
import numpy as np
from smt.surrogate_models import KRG


xt = np.array([0.0, 1.0, 2.0, 3.0, 4.0])
yt = np.array([0.0, 1.0, 1.5, 0.5, 1.0])


sm = KRG(theta0=[1e-2])
sm.set_training_values(xt, yt)
sm.train()
```

Listing 2.1: Python script to construct a kriging model using SMT [42]
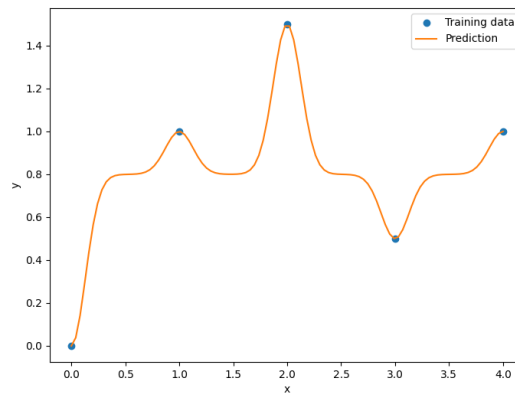
Figure 2.11: Kriging model constructed in SMT [42].

An implementation of Le Gratiet's formulation to build kriging models was implemented by Rémi Vauclin [43], a former intern at ISAE-Supaero and it is available as part of the OpenMDAO package [44]. Mostafa Meliani [2] adapted his implementation to fit the format of the SMT library.

Although the SMT library includes the EGO implementation, the implementation of the updated EGO for constraints (SEGO) for single and multi-fidelity, is a confidential Python package developed by ON-ERA.

Chapter 3

Multi-disciplinary Design Analysis and Optimization

In the current chapter, some Multi-disciplinary Design Analysis Optimization (MDAO) background within the scope of this thesis is glossed over. Let us start by introducing some basic concepts, followed by a survey on the employed MDAO architecture and optimization algorithms. Then, the aerodynamic and structural models are discussed and, finally, the interaction between disciplines is explained.

3.1   Terminology and Mathematical Notation

Before an optimization problem can be posed, it is essential to have a clear definition of some basic concepts:

- Objective function: the function that represents the goal of the optimization problem, that is, the quantity of interest that is being minimized (or maximized). Examples for our case could be the amount of fuel consumed;

- Design variables: also known as design parameters, these variables are controlled by the optimizer. These are henceforth the variables that are tuned to obtain the optimal value of the objective function. These variables can be local or shared by multiple disciplines. In our situation, examples of design variables could be angle of attack or wing span;

- Constraints: problem restrictions to obtain acceptable results. They can be simple bound constraints of some design variable, such as $-10° < \alpha < 10°$, or more complex imposed relations, such as $L = D$ where L is lift and D is drag;

Now, let us define some terminology directly related with MDAO:

- Discipline analysis: consists in solving a system of equations that describes the functioning of one discipline considered in the MDAO problem;

- State variables: the outputs of solving a discipline analysis, which constitute the current state of the system, at that iteration. They may or may not be controlled by the optimizer. Examples could be the flow pressure or the displacement distribution;

- Coupling variables: state variables of one discipline that are used by other disciplines, such as the aerodynamic loads on the surface that are passed to the structural analysis;

- Target variables: a copy of coupling variables made to allow discipline analyses to run independently;

In the following sections, many of these notations will be referred to. This work's terminology is consistent with the work of Lambe and Martins [3], presented in table 3.1.

Table 3.1: Notation in MDO problem formulation [3].

Symbol	Definition
$\mathbf{x}$	Vector of design variables
$\mathbf{y^t}$	Vector of coupling variable targets (inputs to a discipline analysis)
$\mathbf{y}$	Vector of coupling variable responses (outputs from a discipline analysis)
f	Objective function
$\mathbf{c}$	Vector of design constraints
$()_0$	Functions or variables that are shared by more than one discipline
$()_i$	Functions or variables that apply only to discipline i
$()^*$	Functions or variables at their optimal value

3.2 Architectures

One of the most important considerations when implementing MDAO is which architecture to adopt as it describes how a method handles the coupling of several disciplines and how the overall optimization problem is solved. The architectures can be either monolithic or distributed. In monolithic approaches, a single optimization problem is solved, whereas in a distributed approach the problem is partitioned into multiple sub-problems containing small subsets of the variables and constraints [3]. We will only explore monolithic architecture, as it is the only related with the thesis scope.

In a monolithic architecture framework, the MDAO problem is treated as a standard constrained nonlinear programming problem, thus a single optimizer is employed to solve the problem. This optimizer seeks out the values of the design variables that minimize the objective function, while satisfying the constraints. There are many different architectures in the monolithic approach, but we will solely present the Multi-Disciplinary Feasible (MDF) architecture, that we will be using during this dissertation. This architecture is employed because it always returns a consistent set of feasible coupling variables at every optimization iteration, as we will see. MDF solves the following optimization problem:

$$
\begin{aligned}
&\text{minimize} && f(x, y(x, y)) \\
&\text{with respect to} && x \\
&\text{subject to} && c_0(x, y(x, y)) \geq 0 \\
&&& c_i(x_0, x_i, y_i(x_0, x_i, y_{j \neq i})) \geq 0 \quad \text{for } i = 1, ..., N.
\end{aligned}
\tag{3.1}
$$
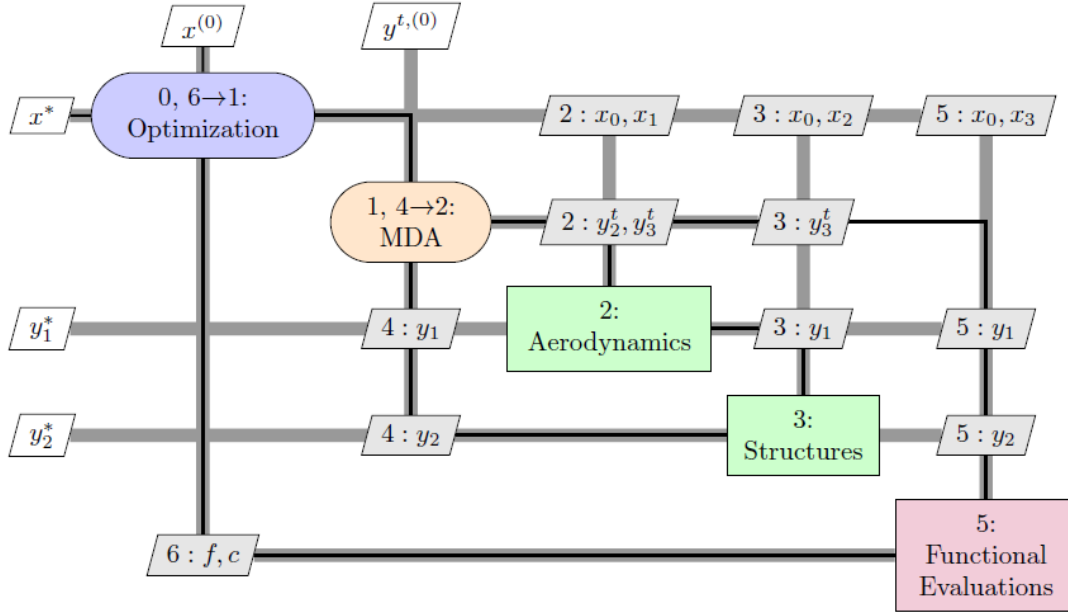
Figure 3.1: Extended design structure matrix for the multi-disciplinary feasible architecture with a Gauss-Seidel multi-disciplinary analysis solver [12].

Figure 3.1 depicts the extended design structure matrix diagram [45] borrowed from Lambe and Martins [12]. This diagram is based on the design structure matrix [46, 47] and it is used to visualize the dependency and process flow between components of the architecture on a single diagram. In the main diagonal of the matrix are placed the components, such as the disciplines analyses, the driver and the optimizer. In the off-diagonal locations placed in the same column to a component are the corresponding inputs and in the same row to a component are the corresponding outputs. As an example, the aerodynamics discipline has as inputs x_0, x_1 and y_2^t, y_3^t and as outputs y_1. External inputs are placed on the top outer edges of the diagram, $x^{(0)}$ and $y^{t,(0)}$, respectively initial design variables and initial guess of coupling variables. The final outputs are placed on the left outer side of the diagram, x^*, y_1^* and y_2^*.

Additionally, the process is broken down into stages by using a numbering system: the process begins in (0) with the initial conditions and proceeds in numerical order. Loops are symbolized by $\rightarrow$. In figure 3.1, there are two loops, the MDA loop within the optimizer loop. The thin black line connects the consecutive components to help visualize the process steps order. Further details of the diagram syntax and interpretation are presented by Lambe and Martins [45].

Analyzing in more detail the MDF architecture and following the sequential numerical flow of figure 3.1, we observe that each discipline analysis computes its outputs that are passed to other discipline analysis, or to the driver, when the MDA is being converged. After the MDA convergence, those outputs

are used to evaluate the objective function and constraints. Thus, one of the benefits of this architecture is that it always returns a fully consistent system design, even if the optimization process ended prematurely. However, the constraints are only evaluated after the MDA has converged, consequently the same cannot be said regarding the constraint satisfaction. Although ensuring that each optimization iteration is multi-disciplinary feasible is an advantage, developing the MDA procedure at every iteration can be time consuming. In next section, we explore the approaches to perform MDA, namely the Gauss-Seidel and Newton procedures.

3.3   Multi-Disciplinary Analysis (MDA)

3.3.1   Gauss-Seidel and Newton MDA

Different approaches can be used to solve the governing equations of the discipline analyses, the inner loop of figure 3.1. These approaches are divide in two groups: fixed-point and Newton [48]. The fixed-point approaches, keep the data of other analyses frozen (fixed), when solving a particular analysis. As depicted in figure 3.2 (a), each analysis is surrounded by dashed line representing each of them being solved individually. When doing so, the approach can use the previous iteration output, Jacobi approach, or use the most recent output from the other analysis, Gauss-Seidel approach, depicted in figure 3.1. Generally, the Gauss-Seidel converges faster than the Jacobi approach.

The Newton approach solves all the analyses simultaneously, as depicted in figure 3.2 (b) by the dashed line surrounding the complete diagram.

(a) Fixed-point approach.   (b) Newton approach.
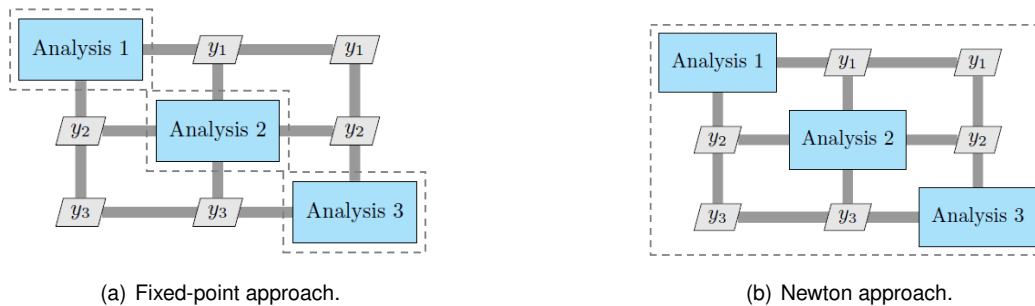
Figure 3.2: Fixed-point and Newton MDA approaches [1].

The decision of each approach to use in the MDA convergence is made based on a study presented in [12]. Figure 3.3 compares the computational costs of different MDA approaches for the solutions of the coupled aero-structural system for level flight (1 g) and pull-up maneuver (2.5 g) using OAS. The MDA approaches under investigation are the Newton, the Non-Linear Block Gauss-Seidel (NLBGS) and the NLBGS with Aitken relaxation (NLBGS w/ Aitken). The Aitken relaxation is derived from the Gauss-Seidel by introducing an extrapolation parameter. For the optimal extrapolation parameter, the NLBGS with Aitken relaxation may converge faster than simple NLBGS by an order of magnitude [48].

By decreasing the spar thickness, the wing flexibility increases and consequently the coupling strength. We observe that the NLBGS generally requires less time to solve the coupled system, but it can not con-
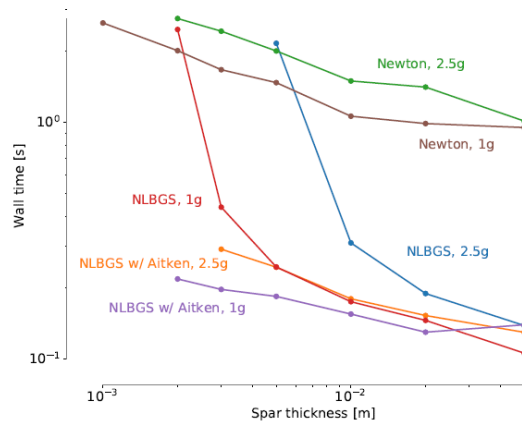
Figure 3.3: Comparison of different solvers for the solutions of the coupled aero-structural system for level flight and pull-up maneuver [12].

verge the systems with strongest coupling strength as well as Newton. In addition, as the spar thickness decreases, the NLBGS without Aitken relaxation cannot converge the coupled system as fast as the NLBGS with Aitken relaxation. Thereby, the approach used to converge the MDA in this dissertation is the NLBGS with Aitken relaxation.

3.3.2 Aerodynamics and Structures MDA

To conclude this section, let us summarize the MDA step of the optimization problem studied in this thesis. As referred, the two disciplines considered are the aerodynamics and structures. The aerodynamic discipline is the first one to be computed within the MDA iterative process and the structural discipline is the second, as illustrated in figure 3.1 from section 3.2.

Figure 3.4 synthesizes the data flow through both disciplines, represented in green boxes. The inputs of both disciplines are vertically aligned with the boxes and the outputs are horizontally aligned with them. The steps (1) to (4) represent the MDA analysis, while step (5) represents the discipline outputs used as inputs on the performance analysis.
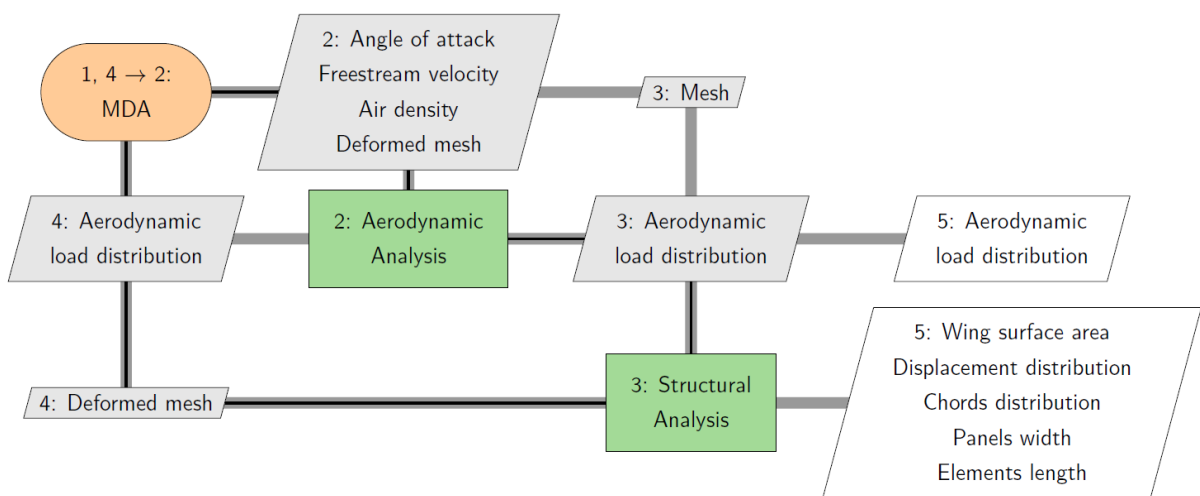
Figure 3.4: Aerodynamics and structures MDA data flow.

As depicted in figure 3.4, the aerodynamic discipline outputs the same data for the MDA convergence and for the performance analysis. The outputs from the structural discipline are vastly different for MDA and performance evaluation. For the MDA, we solely need the deformed mesh, whilst to evaluate the structural constraints and other aerodynamic quantities extra information is needed.

3.4   Optimization Methods and Sensitivity Analysis

The optimizer component, depicted in figure 3.1, can use several methods to solve our problem. The optimization methods are divided into gradient-free, also called zero-order or pattern search, and gradient-based. The topics covered in chapter 2 describe the methodology to implement Bayesian optimization, a gradient-free method. In this section, we will explore a gradient-based method, the Sequential Least-Square Quadratic Programming (SLSQP).

3.4.1   Gradient-free Methods

Gradient-free methods do not rely on derivative information to guide them toward a local minimum or identify when they have reached a local minimum [6]. Instead, they use solely the objective function value to perform those tasks. These methods are more adequate than gradient-based methods when dealing with discontinuous functions, discrete search space or noisy functions [1].

There are often problems, such as discrete design space or several local minima, where deterministic methods are not adequate. In such situations, one should use heuristic (stochastic) methods that use a degree of randomness to help escape local optima and thus increase the chance of finding global optimum. Some heuristic methods try to mimic some behavior found in nature, such as genetic algorithms or ant colony optimization [1].

The Bayesian optimization, described in section 2.2, is a gradient-free, stochastic method employed to optimize expensive to evaluate functions. This algorithm will be use to conduct an optimization study with a multi-disciplinary problem.

3.4.2   Gradient-based Methods

Gradient-based methods define the search direction to progress from one iteration to the next, based on both evaluations of the objective function and its derivative with respect to the design parameters. These methods require a smooth objective function within the design space, but they only guarantee convergence to a local optimum, being dependent of the initial guess. However, their main advantage is that they are faster since they converge to the optimum with a significantly smaller number of function evaluations.

The steepest descent is the simplest gradient-based method, where the search direction is obtained via the gradient vector. Newton methods require second-order information (Hessian matrix) and exhibit a much higher rate of convergence. To avoid the computation of the second-order derivatives, a new

class of methods, referred to as Quasi-Newton, uses solely first-order information and relies on function values and derivatives from previous iterations to approximate second-order information [14].

The gradient-based optimization algorithm employed in this work is the Sequential Least-Square Quadratic Programming (SLSQP) [49]. This algorithm allows to minimize a function of several variables with any combination of bounds, equality and inequality constraints. The idea behind Sequential Quadratic Programming (SQP) is to simplify the original problem, a non-linear programming problem, into a sequence of simpler quadratic sub-problems at the design point x_i and to use the solution of this sub-problem to find the new point x_{i+1}. We use this algorithm to minimize the $WB2$ criterion when performing Bayesian optimization and to minimize the same MDO problem employed in Bayesian optimization to obtain a reference solution.

To understand how the sub-problem is built in SQP, consider the equality constraint optimization problem as

$$
\begin{aligned}
&\text{minimize} && f(x) \\
&\text{with respect to} && x \in \mathbb{R}^n \\
&\text{subject to} && c_j(x) = 0 \quad \text{for } j = 1, ..., l.
\end{aligned}
\tag{3.2}
$$

We can define the Lagrangian function of this problem as

$$
\mathcal{L}(x, \lambda) = f(x) + \lambda^T c(x),
\tag{3.3}
$$

where λ is the vector of Lagrange multipliers, a set of scalars. Deriving equation (3.3) in order to x and λ we can obtain two important conditions:

$$
\begin{cases}
\frac{\delta \mathcal{L}}{\delta x_i} = \frac{\delta f(x)}{\delta x_i} + \sum_{n=1}^{N} \lambda_j \frac{\delta c_j(x)}{\delta x_i} = 0, \quad \text{for } i = 1, ..., n \\
\frac{\delta \mathcal{L}}{\delta \lambda_j} = c_j(x) = 0, \quad \text{for } j = 1, ..., l.
\end{cases}
\tag{3.4}
$$

These first order conditions, known as the Karush-Kuhn-Tucker (KKT) conditions, are necessary for the optimum of a constrained problem. This way, the constrained optimization problem with n variables and l constraints has been transformed into an unconstrained problem of $n + l$ variables.

Going back to the SQP formulation, let us define a quadratic sub-problem at the design point x_i as

$$
\begin{aligned}
&\text{minimize} && \frac{1}{2} p^T \nabla^2 (f(x))_i p + \nabla (f(x))_i p \\
&\text{subject to} && \nabla (c_j(x))_i p + (c_j(x))_i = 0 \quad \text{for } i = 1, ..., n,
\end{aligned}
\tag{3.5}
$$

where p is the only unknown and a factor that will determine the next point. We can define the first order KKT conditions presented in equation (3.4) based on this subproblem yielding

$$
\begin{cases}
\nabla^2 (f(x))_i p + \nabla (f(x))_i - \nabla (c_j(x))_i^T \lambda = 0 \\
\nabla (c_j(x))_i p + (c_j(x))_i = 0, \text{for } i = 1, ..., n.
\end{cases}
\tag{3.6}
$$

The system can be expressed in matrix form,

$$
\begin{bmatrix} \nabla^2(f(x))_i & -\nabla(c_j(x))_i^T \\ \nabla(c_j(x))_i & 0 \end{bmatrix} \begin{bmatrix} p_i \\ \lambda_{i+1} \end{bmatrix} = \begin{bmatrix} -\nabla(f(x))_i \\ -(c_n(x))_i. \end{bmatrix} \tag{3.7}
$$

The previous system has the form of the Newton's method, $H \times h = -g$, where H and g represent the Hessian matrix and Jacobian vector of the Lagrangian, respectively. The h is the finite difference interval or perturbation step-size.

The first set of variables, p_i, is used to update the design point as

$$
x_{i+1} = x_i + p_i. \tag{3.8}
$$

The second set of variables, λ_{i+1}, is the updated value for the next design point [1].

Most applications of SQP are quasi-Newton methods, that is, use Hessian approximations for the second-order terms rather than compute them analytically, as previously referred. The SLQP algorithm performs such approximation. The SQP method can be extended to handle inequality constraints, a useful feature since the problem we will be looking to minimize has equality and inequality constraints. The inequality constraints are linearized as the equality constraints in equation (3.5). The most common strategy to solve this problem is the active-set approach, where we only consider the active constraints at a given iteration and treat them as equality constraints [50].

3.4.3   Sensitivity analysis

An important aspect to define how the problem is approach is the way the derivatives of the objective function and the constraints with respect to the problem variables are computed. This is called sensitivity analysis. Sensitivity analysis is the study of how the outputs of a model change in response to changes in its inputs [51].

Several methods to perform sensitivity analysis exist, the most popular being finite differences, which can use forward step formula for the first order ($O(h)$) approximation to the first derivative given by

$$
\frac{\partial f}{\partial x} = \frac{f(x+h) - f(x)}{h} + \mathcal{O}(h), \tag{3.9}
$$

where h is the step-size and $\mathcal{O}(h)$ is the truncation error proportional to h. The order of approximation can be made higher at the expense of more complex formulae that require more function evaluation points. Although finite differences method lacks robustness, they are very popular due to its simplicity. This method is used to compute the derivatives in the maximization of the infill sampling criterion $WB2$ using the SLSQP algorithm.

To efficiently compute the derivatives of the MDO problem when using a gradient-based optimizer, the SLSQP optimizer in our situation, more complex methods have to be implemented [51, 52]. The computation of complete derivatives is required, that is, we want to compute derivatives across the several components that constitute the system.

The MDAO framework used in this work employs the modular analysis and unified derivatives sensitivity analysis architecture. Let us briefly introduce this architecture, that is completely derived in Martins and Hwang [52]. The first step is to concatenate the m input variables, the n state variables and the p output functions into a single vector $\mathbf{u}$ as

$$\mathbf{u} = [z_1, \ldots, z_m, y_1, \ldots, y_n, g_1, \ldots, g_p]. \tag{3.10}$$

The design variables are a subset of the input variables which are independent variables. Output functions are the quantities we are interested in, such as the objective function and constraints, represented in figure 3.1 in the fifth step.

Now, we define the residual functions associated to $\mathbf{u}$ as

$$R_i(\mathbf{u}) = z_i - z_i^*, \quad \text{for } i = 1, \ldots, m \tag{3.11}$$

for all the input variables z_i,

$$\begin{cases} R_{m+1}(\mathbf{u}) = y_i - y_i^*(z_1, \ldots, z_m, y_2, \ldots, y_n) & \text{if } y_i \text{ is explicitly defined} \\ R_{m+1}(\mathbf{u}) = -\mathcal{R}_i(z_1, \ldots, z_m, y_1, \ldots, y_n) & \text{if } y_i \text{ is implicitly defined} \end{cases} \text{, for } i = 1, \ldots, n \tag{3.12}$$

for the state variables y_i, and

$$R_{m+n+i} = g_i - g_i^*(z_1, \ldots, z_m, y_1, \ldots, y_n), \quad \text{for } i = 1, \ldots, p \tag{3.13}$$

for all the output functions g_i. For equations (3.11-3.13), the input variable z_i^* represents the value z_i at the point where we are evaluating the expressions. The y_i^* and g_i^* are the functions values at z_i^*.

We may now concatenate all the residuals equations into a single system of equations as

$$\left. \begin{array}{c} R_1(u_1, \ldots, u_q) = 0 \\ \vdots \\ R_q(u_1, \ldots, u_q) = 0 \end{array} \right\} \Leftrightarrow R(\mathbf{u}) = 0. \tag{3.14}$$

We introduce the vector $\mathbf{r}$ as the value of the residual vector. We desire to solve the system of equation (3.14) at $\mathbf{r} = 0$, the system's solution. Assuming that $\partial R / \partial \mathbf{u}$ is invertible at the solution of the system, we may define

$$\frac{d\mathbf{u}}{d\mathbf{r}} = \frac{\partial (R^{-1})}{\partial \mathbf{r}}, \tag{3.15}$$

evaluated at $\mathbf{r} = 0$. Applying the inverse function theorem at equation (3.15), one obtains

$$\frac{\partial (R^{-1})}{\partial \mathbf{r}} = \left[\frac{\partial R}{\partial \mathbf{u}} \right]^{-1}, \tag{3.16}$$

31

which we can now combine with equation (3.15) to obtain

$$\frac{\partial R}{\partial \mathbf{u}}\frac{d\mathbf{u}}{d\mathbf{r}} = \mathcal{I} = \frac{\partial R^T}{\partial \mathbf{u}}\frac{d\mathbf{u}^T}{d\mathbf{r}}. \tag{3.17}$$

Equation (3.17) is called the unifying derivative equation. Thereby, we shown that the derivatives of the outputs with respect to the inputs of the model can be computed by solving the linear system (3.17).

3.5 Discipline Models

Recalling figures 3.1 and 3.4, we need mathematical formulations to model the aerodynamic and structural disciplines analysis, presented inside the inner MDA loop. In next sections, we start by presenting the vortex lattice method used to model the aerodynamic discipline and then the finite element method used to model the structural discipline.

3.5.1 Aerodynamics

As a starting point to this section, let us have a look on some aerodynamic concepts and numerical methods used on several wing model schemes. In the next sections we follow Anderson [53].

Vortex Flow Element

Let us start by introducing the vortex flow element, the most essential concept when modeling lifting surfaces. It is defined as a flow where the streamlines are concentric circles of constant velocity about a given point, as illustrated in figure 3.5. Moreover, the streamline velocities vary from one to another inversely with distance from the origin.
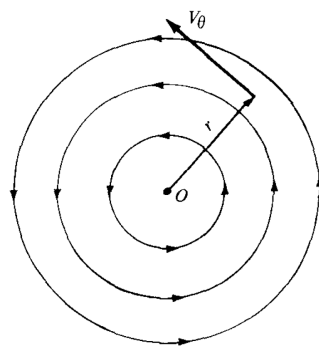
Figure 3.5: Vortex flow element [53].

The velocity components in the radial and tangential directions, V_r and V_θ, respectively, are expressed by

$$\begin{cases} V_\theta = \frac{constant}{r} = \frac{C}{r} \\ V_r = 0. \end{cases} \tag{3.18}$$

In order to evaluate the constant C, the flow's circulation is evaluated, defined as

$$\Gamma = -\oint_C \mathbf{V} ds = -V_\theta (2\pi r) \Rightarrow V_\theta = -\frac{\Gamma}{2\pi r}. \tag{3.19}$$

In equation (3.19), we see that by solely knowing Γ, the vortex strength, we know the tangential velocity of a vortex flow at any point.

Next, we need to extend the concept of vortex element, to something more related to the numerical methods of model lifting surfaces. Thus, it is discussed the concept of vortex filament, depicted in figure 3.6.
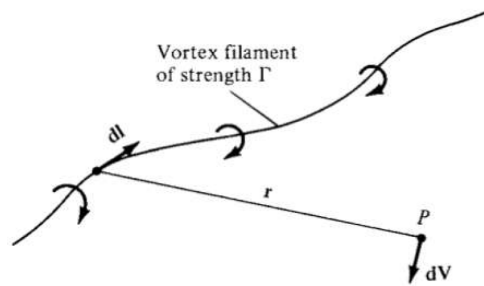
Figure 3.6: An arbitrarily shaped vortex filament of strength Γ [53].

A vortex filament is just an extension of the point vortex flow, previously described, applied to an arbitrarily shaped line. The flow induced in any plane perpendicular to this line by the filament itself is identical to that induced by a point vortex of strength Γ. To express the induced velocity by the filament segment $d\mathbf{l}$ at a point P, we resort the Biot-Savart law,

$$d\mathbf{V} = \frac{\Gamma}{4\pi} \frac{d\mathbf{l} \times \mathbf{r}}{||\mathbf{r}^3||}. \tag{3.20}$$

By integrating equation (3.20) from negative infinity to infinity, it can be obtained the velocity at point P induced by the entire vortex filament.

The physician Hermann von Helmholtz established basic principles of the vortex behavior, known as the Helmholtz's vortex theorem:

1. The strength of a vortex filament is constant along its length;

2. A vortex filament can not end in a fluid; it must extend to the boundaries of the fluid (be they finite or infinite) or form a closed path.

Lifting Line Theory

Based on the concepts presented in last subsection, we can now move on to a finite wing model built on a single vortex filament, the Prandtl's classical lifting line theory, illustrated in figure 3.7. This theory establishes that a wing is replaced by a vortex of strength Γ, bound to a fixed location in the flow, the lifting line. Recalling the Helmholtz's theorem that states that the vortex filament cannot end in a fluid,
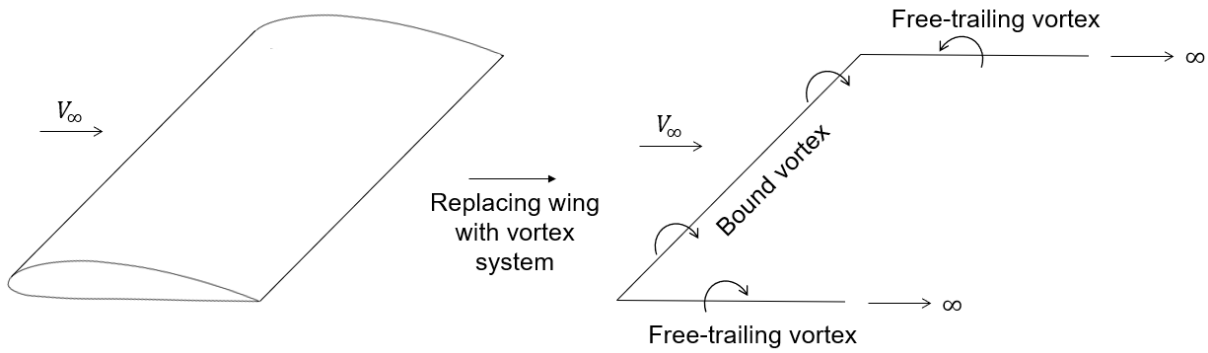
Figure 3.7: Diagram of a single horseshoe vortex wing modeling scheme.

this bound vortex continues as two free vortexes trailing downstream from the wing tips to infinity. Due to the shape formed by the vortex, it is called a horseshoe vortex.

We may now enhance the approach by superimposing multiple horseshoe vortexes, coincident along a single line, the lifting line. Figure 3.8 illustrates this concept, where only two horseshoe vortexes are
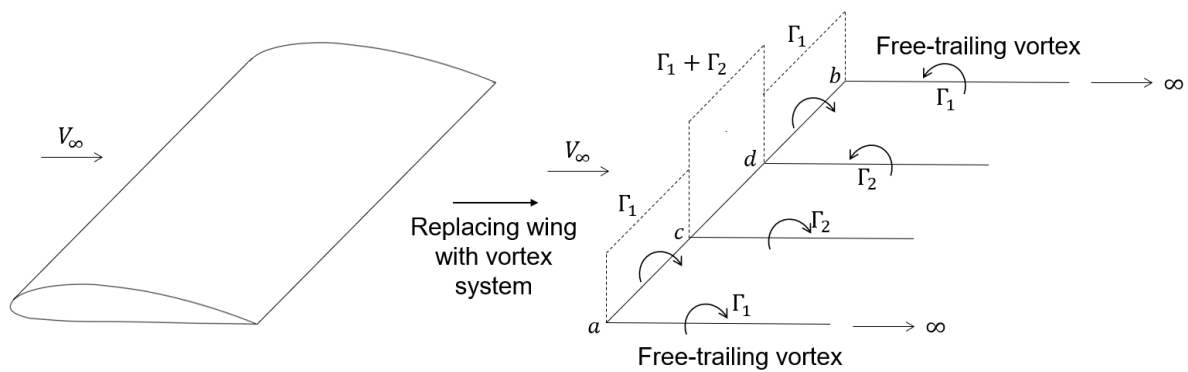
Figure 3.8: Diagram of a superimposition of two horseshoe vortexes on the lifting line.

represented for the sake of clarity. Analyzing figure 3.8, we observe that only a single bound vortex of strength Γ_1 acts on the entirety of the wing's span from point a to point b. It is also the strength of the vortexes that then trails off at the ends of the wing. Superimposed on this is a second horseshoe vortex on $\overline{cd}$. As a result, the strength of the vortex on this segment is $\Gamma_1 + \Gamma_2$. Additionally, two vortexes are trailing off to infinity on point c and d with Γ_2 strength.

Summarizing, the lifting line theory models a wing through several horseshoe vortexes on a single lifting line. This model enables only one-dimensional spanwise discretization, which translates in the number of horseshoe vortexes implemented.

Vortex Lattice Method

Let us extend upon the previous model by superimposing several lifting lines along the chordwise direction in order to enable a two-dimensional discretization, so that chordwise variation is also handled. More specifically, let us model the wing as a set of distributed spanwise and chordwise lifting panels. Each panel contains a single horseshoe of an unknown strength. Figure 3.9 shows one of these panels

34

in dashed line, while the horseshoe vortex associated is drawn with solid line. The bound vortex is located at one quarter of the chord from the front of the panel with two trailing vortex lines shed from each end. Additionally, we define a control point at the center line of the panel and three quarters from the front of the panel.
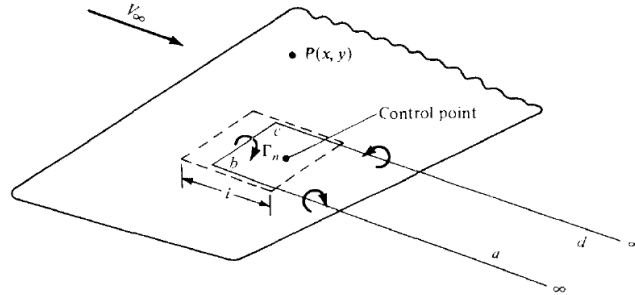
Figure 3.9: Lifting panel with a single horseshoe vortex [53].

The required strength of the vortex on each panel is obtained by applying the flow tangency condition at the respective control point, that specifies that the velocity normal to the panel must be zero. The normal velocity is made up of a freestream component and an induced flow component defined as

$$\mathbf{V}_{\infty}.\mathbf{n} + \mathbf{u}_n = 0 \tag{3.21}$$

where $\mathbf{V}_{\infty}$ is th freestream velocity, $\mathbf{n}$ is the normal to the panel and $\mathbf{u}_n$ is the induced flow component.

The induced flow component $\mathbf{u}_n$ is a linear combination of the effects of the strengths of all panels and can be expressed as

$$\mathbf{u}_n = \sum_{j=1}^{N} A_{i,j} \Gamma_j, \tag{3.22}$$

where N represents the total number of vortexes defined and $A_{i,j}$ represents a row of the aerodynamic influence coefficient matrix. The influence coefficient $A_{i,j}$ represents the induced flow on panel i due to the vortex on panel j. The aerodynamic influence coefficient matrix is computed through the Biot-Savart law, introduced in equation (3.20). Therefore, the zero flow normal to the surface can be described by equation (3.23)

$$\sum_{j=1}^{N} A_{i,j} \Gamma_j = -\mathbf{V}_{\infty}.\mathbf{n} \tag{3.23}$$

Solving the linear system (3.23), we obtain the circulation distribution for the entire surface. Subsequently, we can compute the aerodynamic forces acting on each individual panel by the Kutta-Joukowski theorem,

$$\mathbf{F}_i = \rho \left(\mathbf{V}_{\infty} + \mathbf{v}_i \right) \Gamma_i \times \mathbf{b}_i, \tag{3.24}$$

where $\mathbf{v}_i$ and $\mathbf{b}_i$ are, respectively, the velocity induced by the vortexes at the control point and the vector describing the bound vortex for each panel.

The resultant force distribution can be decomposed in lift and drag distributions, which can then be

integrated to obtain the total aerodynamic forces of the wing.

3.5.2 Structures

In the previous section, we established how we can determine the aerodynamic effect of the lifting surface. We must now define a structural model that allows us to obtain the wing stiffness, which is then used to compute the displacements of the structure.

The finite element method approach is engaged. This method divides the lifting surface into smaller parts with the aim to simplify the model computations. The relevant equations are solved for each part, in a local frame of reference, and then the results are transformed to a global reference frame and combined, so that, the complete system behavior is captured [54].

The most common wing structure component is the wing-box, with a thin-walled spar and hollow interior, as illustrated in figure 3.10 (a). However, for the sake of simplicity, a single one-dimensional thin-walled beam of circular cross-section is employed, as shown in figure 3.10 (b).

(a) Wing-box cross-section model.

(b) Spar cross-section model.

Figure 3.10: Wing-box and spar cross-section models.

Base Elements

When applying the finite element method, one important decision is which element to use. The beam element chosen to model the wing structure is a combination of three base elements, namely two plane beam elements, one truss element and one torsional bar element.

Let us first establish the truss element illustrated in figure 3.11 under pure axial loads. This element is used to capture the extension/compression suffered by the model, since its degrees of freedom are displacements on the x axis.
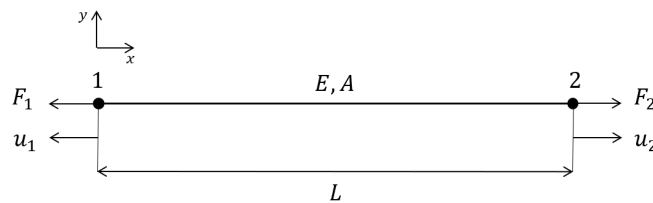
Figure 3.11: Finite element for axial analysis.

The stiffness matrix of this element can be written as

$$[k]_a = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{3.25}$$

36

and its displacement vector is

$$\mathbf{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}. \tag{3.26}$$

Here, E is the Young Modulus, A is the element's cross sectional area and L is the element's length.

Next, we define the element for torsion in the x axis, shown in figure 3.12.

Figure 3.12: Finite element for torsional analysis.

The stiffness matrix yields

$$[k]_t = \frac{GJ}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{3.27}$$

where G is the shear modulus and J is the polar moment of inertia. The corresponding displacement vector is given by

$$\mathbf{u} = \begin{bmatrix} \alpha_{x,1} & \alpha_{x,2} \end{bmatrix}. \tag{3.28}$$

Lastly, the element under pure bending condition is defined, depicted in figure 3.13 considering bending about the z axis.
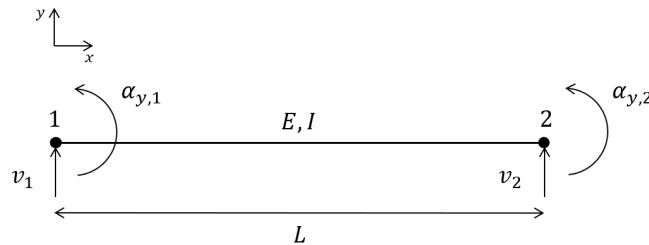
Figure 3.13: Finite element for bending analysis.

The stiffness matrix of such element is written as

$$[k]_b = \frac{EI_z}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \tag{3.29}$$

with the displacement vector

$$\mathbf{u} = \begin{bmatrix} v_1 & \alpha_{y,1} & v_2 & \alpha_{y,2} \end{bmatrix}, \tag{3.30}$$

where I_z is the second moment of inertia about its respective axes. This element is also employed for bending about the y axis, with I_z being replaced by I_y.

Employed Element

Finally, we can present the employed element, which superimposes the three elements described in the previous section. The complete element is illustrated in figure 3.14.
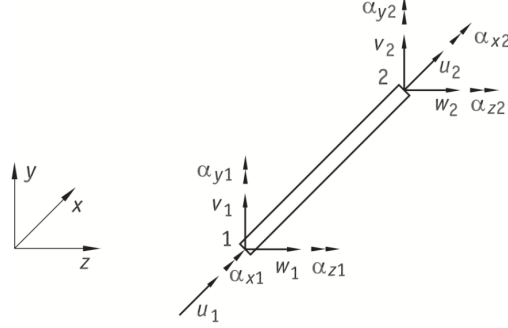
Figure 3.14: 12 degrees of freedom beam element [12].

Assuming the displacement/rotation vector is

$$\mathbf{u} = [u_1 \quad v_1 \quad w_1 \quad \alpha_{x,1} \quad \alpha_{y,1} \quad \alpha_{z,1} \quad u_2 \quad v_2 \quad w_2 \quad \alpha_{x,2} \quad \alpha_{y,2} \quad \alpha_{z,2}],$$ (3.31)

the local spatial beam element stiffness matrix is given by the combination of the stiffness matrices of the simpler elements as

$$[k]_e = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-EA}{L} & 0 & 0 & 0 & 0 & 0 \\ & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} & 0 & \frac{-12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \\ & & \frac{12EI_y}{L^3} & 0 & \frac{-6EI_y}{L^2} & 0 & 0 & 0 & \frac{-12EI_y}{L^3} & 0 & \frac{-6EI_y}{L^2} & 0 \\ & & & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-GJ}{L} & 0 & 0 \\ & & & & \frac{4EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\ & & & & & \frac{4EI_z}{L} & 0 & \frac{-6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \\ & & & & & & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ & \text{Symmetric} & & & & & & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{-6EI_z}{L^2} \\ & & & & & & & & \frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 \\ & & & & & & & & & \frac{GJ}{L} & 0 & 0 \\ & & & & & & & & & & \frac{4EI_y}{L} & 0 \\ & & & & & & & & & & & \frac{4EI_z}{L} \end{bmatrix}.$$ (3.32)

The elementary matrix $[k]_e$ is formulated for each element in the local reference frame and then transformed to the global frame. This transformation involves a rotation matrix, which will not be detailed. After that, the global stiffness matrix $[K]$ of the entire structure is obtained by assembling each element $[k]_e$ matrix transposed to the global frame. As discussed in the previous section, the aerodynamic analysis provides a set of loads applied on each of the aerodynamic panels. These loads are used to compute the equivalent loads and moments applied on the beam nodes on a global load vector $\mathbf{F}$. Once

both stiffness and load vectors are assembled, we can solve the Hook's law formulated as

$$[K]\mathbf{u} = \mathbf{F}, \tag{3.33}$$

in order to get the displacement and rotation vector enacting on the structure. Then, we can iterate upon the architecture using the updated mesh on the aerodynamic analysis.

3.6   Fluid-structure Interaction

To conclude this chapter, we introduce the fluid-structure interaction. In the previous sections, we described the aerodynamic and structural models used to obtain the pressure and displacement distributions along the wing, respectively. During the analysis, information needs to be exchange between the two domains, such aerodynamic pressures and structural displacements. Thus, we must now established this fluid-structure interaction.

Assuming that the aerodynamic and structural solution is known at iteration k, the steps to obtain the coupled system solution at iteration $k + 1$ are summarized below [1]:

1. Solve the aerodynamic system for the current iteration $k + 1$;

2. Integrate fluid pressure onto structural domain;

3. Solve the structural analysis with the aerodynamic loads as input;

4. Transfer structural displacements onto the aerodynamic mesh, that is, update the surface mesh;

5. Repeat steps 1 to 4 until convergence;

The update of the surface mesh, specified on step 4, will promote a new aerodynamic pressure field. The goal is to repeat the process until one obtains a pressure field that produces an updated mesh that when aerodynamically analyzed produces the same pressure field.
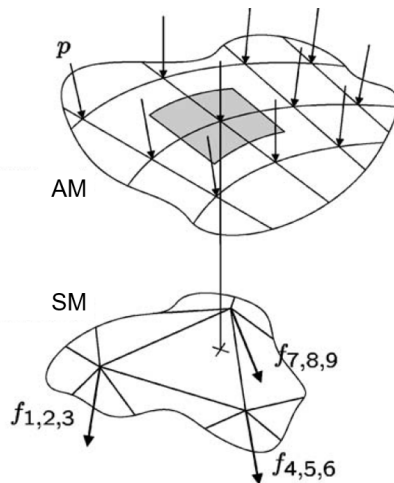
Figure 3.15: Transfer of the pressure of an aerodynamic node to the nodal forces on a given finite element (adapted from [14]).

An approach for data exchange between the two disciplines is necessary, as schematically represented in figure 3.15. The pressure field p of an aerodynamic node in the Aerodynamic Mesh (AM) is transformed into forces that are applied to the respective structural nodes in the Structural Mesh (SM).

The transfer scheme must satisfy the requirements of being consistent and conservative [14]. Consistency states that resultant forces and moments due to the pressure field, must be transferred into an equivalent set of nodal forces and moments, thereby satisfying the load conservation. However, there is an infinite number of sets of nodal forces that satisfy this requirement. The second requirement, conservation, states that the virtual work performed by the load vector, $\mathbf{f}_j$, over displacements, $\delta\mathbf{u}_{s,j}$, on the structural mesh, must be equal to the work performed by the pressure field, p, undergoing equivalent displacements, $\delta\mathbf{u}_{a,i}$, on the aerodynamic mesh [12]. Equations (3.34) and (3.35) translate the virtual work on each of the domains, structural and aerodynamic, where $\mathbf{n}_i$ represents the unit vector normal to the aerodynamic element and S is the panel's area.

$$\delta A_s = \mathbf{f}_j \, \delta\mathbf{u}_{s,j} \tag{3.34}$$

$$\delta A_a = \int_S p \, \mathbf{n}_i \, \delta\mathbf{u}_{a,i} \, dS \tag{3.35}$$

In section 4.3 and 4.4, the transfer schemes for identical and non-identical disciplines discretization, respectively, are presented.

Chapter 4

Aero-Structural Design Analysis and Optimization Framework

In this chapter, we will present the low fidelity tool employed to perform aero-structural analysis and optimization, OAS. We start by introducing the OAS tool, in section 4.1. In section 4.2, the aerodynamic wing mesh and structural spar mesh generation is presented. Then, in section 4.3, we describe the load and displacement transfer schemes of OAS. After the discussion of the original transfer schemes, the developed transfer schemes for non-identical spanwise discipline discretization are explained in section 4.4.

4.1   Aero-structural Design and Optimization Tool

To implement MDAO, the OAS tool is employed [12]. OAS is an open-source low-fidelity aero-structural analysis and optimization tool coupling the vortex-lattice model with 12 degree of freedom beam element, both described in sections 3.5.1 and 3.5.2. It is written entirely in Python within the OpenMDAO framework [55], an open-source high-performance computing platform for efficient optimization. How the data flow is carried out during the analysis, referred to as architecture, is an important aspect of MDO problems that was discussed in section 3.2. Fully analytic derivatives are provided for each component using the modular analysis and unified derivatives, as presented in section 3.4.3. Section A.1 explains how OAS work with more detail.

4.2   Aerodynamic Mesh and Finite Element Structure

OAS problems are initialized by the definition of the lifting surfaces, that is, the aerodynamic wing mesh and the structural beam finite elements.

To define the refinement of the aerodynamic mesh, two variables, $'num\_y'$ and $'num\_x'$, are specified. The $'num\_y'$ and $'num\_x'$ variables represent the spanwise and chordwise discretizations, respectively, and determine the number of panels that discretize the wing. The base shape to model the lifting

41

surface is the NASA Common Research Model (CRM) developed by Vassberg et al. [56]. This model was originally developed as a benchmark solution for aerodynamic analysis and it has been extensively used to compare simulations methods. Thus, CRM became the recurrent model used for investigating new analysis and optimization methods.

As an example of a mesh generation, figure 4.1 displays an aerodynamic mesh constructed based on the CRM shape. As it can be observed, structural beam elements are overlapped with the aerodynamic mesh. Moreover, these structural elements have the same spanwise discretization, since the elements radius changes are coincident with the panels limits. OAS imposes the same spanwise discretization in both disciplines for the sake of simplicity [12].
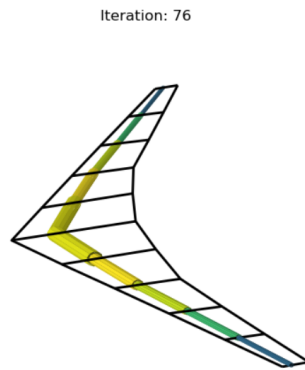
Figure 4.1: Aerodynamic CRM grid and beam finite elements.

Recalling figure 3.1 from section 3.2, the mesh generation is located in step (1), outside the MDA loop. Consequently, this process is repeated before the initialization of every disciplines analysis loop. It is on this mesh that aerodynamic and structural analyses are converged.

4.3   Load and Displacement Transfer Schemes

When working with more than one discipline within MDAO, information between the disciplines needs to be constantly exchanged as referred in section 3.6 and schematized by figure 4.2. One crucial aspect is how the information transfer between them is performed. As presented in section 3.6, the transfer schemes must be consistent and conservative. The OAS simplifies the transfer schemes by imposing
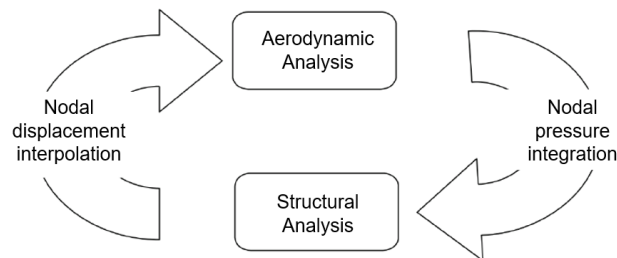
Figure 4.2: Aerodynamic and structural cycle flowchart (adapted from [57]).

the same spanwise discretization for aerodynamic and structural disciplines, as previously observed in figure 4.1.

Jasa, Hwang and Martins [12] described the transfer scheme of OAS satisfying the requirements of being consistent and conservative. However, the transfer schemes found in the OAS implementation were different from the ones described in [12]. For the sake of completeness, in this section both schemes will be examined.

4.3.1 Consistency and Conservation Requirements

Let us start by introducing the transfer scheme that satisfies the requirements of consistency and conservation, as presented in [12].

For the load transfer, let us first examine how the aerodynamic load per unit area $\mathbf{T}$ of one VLM panel is transferred to a finite element structure. Figure 4.3 illustrates a wing section, where the upper limit is the leading edge (LE) and the lower limit is the training edge (TE), and a spar structure. Additionally, figure 4.3 illustrates an aerodynamic panel $ABCD$, the two spanwise (y) aligned structural nodes 1 and 2, the position vectors from the structural nodes to the panel's center of pressure cp ($\mathbf{r_{cp,1}}$, $\mathbf{r_{cp,2}}$) and the forces and moments that have been transferred onto them ($\mathbf{F_{s,1}}$, $\mathbf{F_{s,2}}$, $\mathbf{M_{s,1}}$, $\mathbf{M_{s,2}}$). The subscript s indicates that the quantity is in the structural mesh and S is the panel's area. As observed in figure 4.3, the load distribution $\mathbf{T}$ is applied on the cp which is located at the middle of the panel $ABCD$ in the spanwise direction (y), so half of $\mathbf{T}$ is applied to each of the structural nodes 1 and 2.
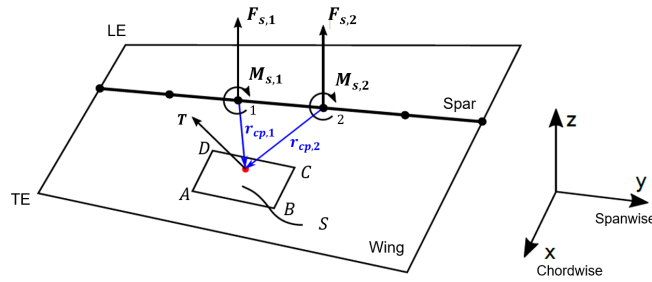
Figure 4.3: Scheme for transferring the load distribution $\mathbf{T}$ to adjacent structural nodes [12].

The nodal forces and moments on left and right structural nodes are given by

$$\mathbf{F_{s,i}} = \int_{\text{panel}} \frac{1}{2}\mathbf{T}dS = \frac{1}{2}\mathbf{T}S \tag{4.1}$$

and

$$\mathbf{M_{s,i}} = \int_{\text{panel}} \mathbf{r_i} \times \frac{1}{2}\mathbf{T}dS = \frac{1}{2}\mathbf{r_{cp,i}} \times \mathbf{T}S, \tag{4.2}$$

where $i = 1, 2$ indicates nodes 1 and 2, and $\mathbf{r_i}$ is the vector pointing from the structural node to a point on the panel.

Equations (4.1) and (4.2) guarantee a consistent scheme, because the forces and moments are equated to be equivalent results from the aerodynamic load distribution.

We must complete the transfer scheme by describing the displacement transfer. The objective is to transfer the computed displacements of the structural elements to deformations of the aerodynamic

mesh $\mathbf{u_a}$. The transfer scheme is defined as

$$\mathbf{u_a} = \frac{1}{2}\sum_{i=1}^{2}\left(\mathbf{d_{s,i}} + \boldsymbol{\theta}_{s,i} \times \mathbf{r_i}\right),\tag{4.3}$$

where $\mathbf{d_{s,i}}$ and $\boldsymbol{\theta}_{s,i}$ are the translational and rotational components of the structural displacements, and $\mathbf{r_i}$ points from structural node i to the aerodynamic center of pressure corresponding to $\mathbf{u_a}$. A constant factor of one-half is used, since we average the contributions from the left and right structural nodes, because we are evaluating the aerodynamic mesh at the midpoint in the spanwise direction.

After defining both transfer schemes, we verify if the conservativeness requirement is satisfied. The virtual work effected on the structural mesh, δA_s, by the nodal forces and moments is given by

$$\delta A_s = \sum_{i=1}^{2}\left(\mathbf{F_{s,i}} \cdot \delta\mathbf{d_{s,i}} + \mathbf{M_{s,i}} \cdot \delta\boldsymbol{\theta}_{s,i}\right).\tag{4.4}$$

Inserting equations (4.1) and (4.2) in (4.4), we obtain

$$\delta A_s = \frac{1}{2}\sum_{i=1}^{2}\left(\mathbf{T} \cdot \delta\mathbf{d_{s,i}} + (\mathbf{r_{cp,i}} \times \mathbf{T}) \cdot \delta\boldsymbol{\theta}_{s,i}\right)S.\tag{4.5}$$

The aerodynamic virtual work δA_a performed by $\mathbf{T}$ over $\delta\mathbf{u_a}$ is

$$\delta A_a = \int_{\text{panel}} \mathbf{T}\delta\mathbf{u_a}dS,\tag{4.6}$$

where we can insert $\mathbf{u_a}$ from equation (4.3), yielding

$$\delta A_a = \frac{1}{2}\sum_{i=1}^{2}\int\left(\mathbf{T} \cdot \delta\mathbf{d_{s,i}} + \mathbf{T} \cdot (\delta\boldsymbol{\theta}_{s,i} \times \mathbf{r_i})\right)dS.\tag{4.7}$$

Since $\mathbf{T}$, $\mathbf{d_{s,i}}$ and $\boldsymbol{\theta}_{s,i}$ are constant over the panel, equation (4.7) can be rewritten as

$$\delta A_a = \frac{1}{2}\sum_{i=1}^{2}\left(\mathbf{T} \cdot \delta\mathbf{d_{s,i}} + \mathbf{T} \cdot (\delta\boldsymbol{\theta}_{s,i} \times \mathbf{r_{cp,i}})\right)S.\tag{4.8}$$

By vector algebra, we have

$$\mathbf{T} \cdot (\delta\boldsymbol{\theta}_{s,i} \times \mathbf{r_{cp,i}}) = (\delta\boldsymbol{\theta}_{s,i} \times \mathbf{r_{cp,i}}) \cdot \mathbf{T} = \delta\boldsymbol{\theta}_{s,i} \cdot (\mathbf{r_{cp,i}} \times \mathbf{T}) = (\mathbf{r_{cp,i}} \times \mathbf{T}) \cdot \delta\boldsymbol{\theta}_{s,i}.\tag{4.9}$$

Thereby, equation (4.8) is equivalent to

$$\delta A_a = \frac{1}{2}\sum_{i=1}^{2}\left(\mathbf{T} \cdot \delta\mathbf{d_{s,i}} + (\mathbf{r_{cp,i}} \times \mathbf{T}) \cdot \delta\boldsymbol{\theta}_{s,i}\right)S.\tag{4.10}$$

Then, we can conclude that equations (4.5) and (4.10) are equivalent ($\delta A_s = \delta A_a$), which proves that the load and displacement transfer schemes are conservative.

4.3.2 Implemented Scheme

Let us now present the transfer schemes encountered in the OAS implementation (version 2.2.0). The load transfer scheme that was found in the OAS tool only differentiates from the one presented in subsection 4.3.1 by the position vectors $\mathbf{r}_{cp,i}$. The position vectors are defined from the middle of the structural beam element to the panel's center of pressure cp, so instead of each panel has two position vectors, as illustrated in figure 4.3, it has only one, as illustrated in figure 4.9. Then, recalling the transfer of load $\mathbf{T}$ of one panel to the adjacent structural nodes illustrated by figure 4.3, not only the forces transferred to the nodes are equal, but also the moments, once the same moment arm is used.

The displacement transfer implemented in OAS was given by

$$\mathbf{u}_a = \mathbf{d}_{s,i} + \boldsymbol{\theta}_{s,i} \times \mathbf{r}_i \tag{4.11}$$

where $\mathbf{r}_i$ is a vector pointing from the structural node to the aerodynamic mesh node, both aligned. The transfer of the displacement of a structural node to a mesh node, $\mathbf{u}_a$, presented in equation (4.11), is computed solely with the information from the respective aligned structural node. In contrast, equation (4.3) transfers the displacements from two structural nodes to the center of pressure of the panel. The displacements at the panel's center of pressure are applied to the four aerodynamic mesh nodes that compose the panel.

The implemented transfer schemes do not fulfill the conservation requirement. In fact, they prove only to be conservative for rectangular wings (taper ratio of one). For tapered wings, it is not conservative, although the difference between the virtual works is very small.

The discrepancy between the transfer schemes was communicated to the Michigan MDO laboratory research group [58]. The load scheme is easily corrected, since only the moment arm needs to be rectified. However, to fix the displacement scheme, it is necessary major structural code changes. A new version (2.2.2) will be released in the future, with both transfer schemes based on the article strategy [12].

New transfers schemes for non-identical discipline discretization are developed and presented in the next section. Figure 4.4 shows a timeline with the two OAS versions released through time. Since the rectification of the transfer schemes is a longstanding process, the new ones are developed based on version 2.2.0.
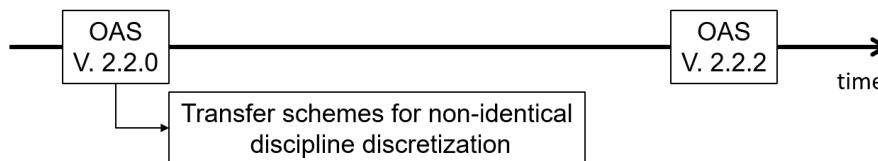
Figure 4.4: OpenAeroStruct versions chronologically ordered.

4.4 Transfer Schemes for Non-identical Discipline Discretization

When performing MDAO with more than one discipline, generally, the disciplines are modeled in two different ways, as in our case, owing to different requirements for the solution of the structural and flow equations, respectively. The aerodynamic analysis often requires an accurate description of the boundary (VLM panels), while for the structural analysis, the body can be simplified (beam-like elements). Even if the structural and flow discretizations are modeling the same boundary surface, often the structural computational nodes do not coincide with the ones used for the flow as illustrated in figure 4.5. Again, the flow computational requires higher resolution than the structural analysis due to the need for solving smaller scales [1, 16].
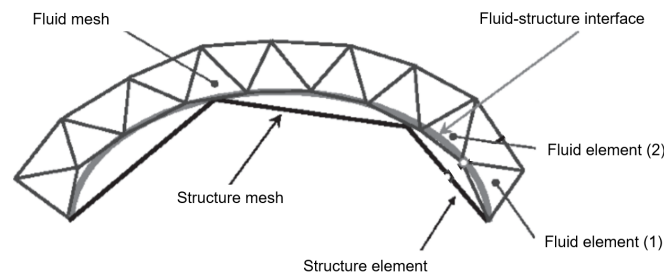
Figure 4.5: Different fluid and structure mesh modeling the same boundary (adapted from [16]).

The transfer schemes are very dependent on the approaches used to model the disciplines and the discretization of these models. The way they are constructed greatly influences the optimization performance in terms of results and computational time spent (see figure 1.5). Thereby, extend the transfer schemes of OAS might be a useful update to save some computational time with just slight or non damage in the optimization results. In this section, we extend the OAS transfer schemes for non-identical discipline spanwise discretization, based on OAS version 2.2.0. We start by presenting two methods to perform load transfer, and then we present one method to perform displacement transfer. Since these new schemes are based on transfer schemes that do not fulfill conservation, they do not satisfy this requirement either.

4.4.1 Load Transfer

Method 1

Method 1 performs the load transfer considering that the force and moment of a structural node is computed through the area bounded by half length of the left beam element to half length of the right beam element of that node. The content of that area can vary, from multiple sections of aerodynamic panels to just one panel section according to the applied discretization.

Figure 4.6 illustrates the area corresponding to node 1 for two different discipline discretizations. For figure 4.6 (a), the Aerodynamic Discretization (AD) is more refined than Structural Discretization (SD) and for figure 4.6 (b), SD is more refined than AD. The aerodynamic panels are represented by the continuous line and the area associated to node 1 is bounded by the dashed line. In figure 4.6 (a) node

1 has associated 3 panel slices and in figure 4.6 (b) node 1 has associated only 2 panel slices. Each of these parts has an associated area S_i and a load per unit area $\mathbf{T}_i$.

(a) AD more refined than SD

(b) SD more refined than AD

Figure 4.6: Panels sections associated to node 1 to perform load transfer.

The equations to compute the force and moment of a structural node yields

$$\begin{cases} \mathbf{F}_s = \sum_{i=1}^{n} \mathbf{T}_i S_i \\ \mathbf{M}_s = \sum_{i=1}^{n} \mathbf{r}_{cp,i} \times \mathbf{T}_i S_i, \end{cases} \tag{4.12}$$

where $\mathbf{T}_i$ and S_i are, respectively, the load and area associated to panel slice i. The vector $\mathbf{r}_{cp,i}$ points from the point on the beam element aligned with the center of pressure to the aerodynamic center of pressure of the panel's slice.

To illustrate this method, let us analyze how the load is transferred to the structural nodes using the same discipline discretization, depicted in figure 4.7.
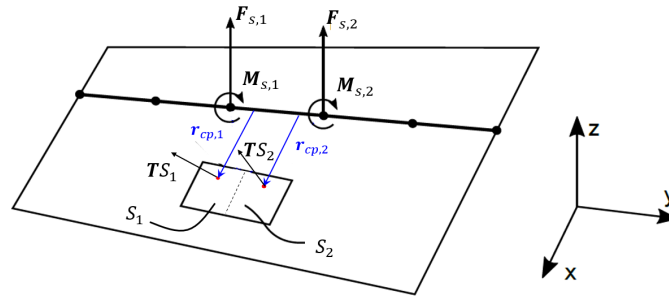
Figure 4.7: Scheme for transferring the load $\mathbf{T}$ to adjacent structural nodes for method 1 with same discipline discretization (adapted from [12]).

The forces and moments at the nodes are defined as

$$\begin{cases} \mathbf{F}_{s,1} = \mathbf{T}S_1 \\ \mathbf{F}_{s,2} = \mathbf{T}S_2 \end{cases} \tag{4.13}$$

and

$$\begin{cases} \mathbf{M}_{s,1} = \mathbf{r}_{cp,1} \times \mathbf{T}S_1 \\ \mathbf{M}_{s,2} = \mathbf{r}_{cp,2} \times \mathbf{T}S_2. \end{cases} \tag{4.14}$$

Method 2

Let us now establish method 2. This method considers the panel's area covered by each finite element, and applies a linear shape function to transferred the load **T** to both nodes.

Figure 4.8 depicts the area corresponding to nodes 1 and 2 for two cases: (a) AD more refined than SD and (b) SD more refined than AD. Similarly to figure 4.6, the aerodynamic panels are represented by the continuous lines and the area of the nodes is bounded by the dashed lines.

(a) AD more refined than SD   (b) SD more refined than AD

Figure 4.8: Panels sections associated to node 1 and 2 to perform load transfer.

The forces and moments of two consecutive nodes are given by

$$\begin{cases} \mathbf{F}_{s,i} = \sum_{j=1}^{n} \mathbf{T}_j S_j \cdot \frac{l_{FE} - x_j}{l_{FE}} \\ \mathbf{F}_{s,i+1} = \sum_{j=1}^{n} \mathbf{T}_j S_j \cdot \frac{x_j}{l_{FE}} \end{cases} \tag{4.15}$$

and

$$\begin{cases} \mathbf{M}_{s,i} = \sum_{j=1}^{n} \mathbf{r}_{ij} \times \mathbf{T}_j S_j \cdot \frac{l_{FE} - x_j}{l_{FE}} \\ \mathbf{M}_{s,i+1} = \sum_{j=1}^{n} \mathbf{r}_{ij} \times \mathbf{T}_j S_j \cdot \frac{x_j}{l_{FE}}, \end{cases} \tag{4.16}$$

where $\mathbf{T}_j$ and S_j are the load associated to the panel slice j, x_j is the y distance between node i and the center of pressure of panel slice j, l_{FE} is the y coordinate of the length of the beam element and $\mathbf{r}_{ij}$ points from the point on the beam aligned with the center of pressure to the aerodynamic center of pressure of panel slice j.

Once again, for the sake of clarity, let us introduce the load transfer of method 2 using the same discipline discretization outlined in figure 4.9.

The forces and moments of nodes 1 and 2 are defined as

$$\begin{cases} \mathbf{F}_{s,1} = \mathbf{T}S \cdot \frac{l_{FE} - x_j}{l_{FE}} = \mathbf{T}S \cdot \frac{1}{2} \\ \mathbf{F}_{s,2} = \mathbf{T}S \cdot \frac{x_j}{l_{FE}} = \mathbf{T}S \cdot \frac{1}{2} \end{cases} \tag{4.17}$$

and

$$\begin{cases} \mathbf{M}_{s,1} = \mathbf{r}_{cp} \times \mathbf{T}S \cdot \frac{l_{FE} - x_j}{l_{FE}} = \mathbf{r}_{cp} \times \mathbf{T}S \cdot \frac{1}{2} \\ \mathbf{M}_{s,2} = \mathbf{r}_{cp} \times \mathbf{T}_j S_j \cdot \frac{x_j}{l_{FE}} = \mathbf{r}_{cp} \times \mathbf{T}S \cdot \frac{1}{2}. \end{cases} \tag{4.18}$$

It is worth of mention that the edges of the panel are aligned with the structural mesh nodes, that is,
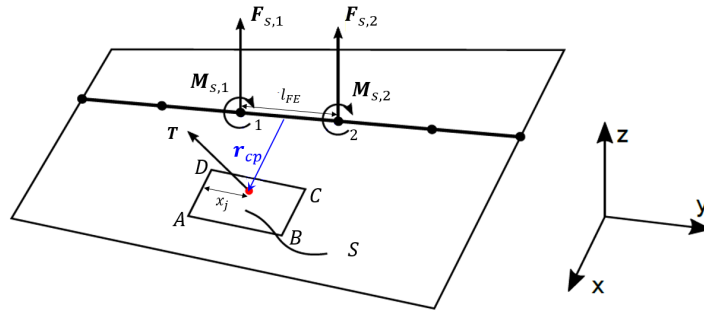
Figure 4.9: Scheme for transferring the load $\mathbf{T}$ to adjacent structural nodes for method 2 with same discipline discretization (adapted from [12]).

A, D have the same y coordinate as node 1 as well as C, D have the same y coordinate as node 2.

In the particular case of the same discipline discretization, x_j is half of l_{FE}, consequently the transfer scheme is equal to the one presented in section 4.3.

Comparison between Load Transfer Methods

After presenting in detail both developed load transfer schemes, we now compare them. As previously referred, both schemes are developed based on the implemented non-conservative version of OAS (version 2.2.0), presented in section 4.3.2. Since both schemes are non-conservative, we do not performed a conservation study to compare them. As future work, when the new OAS version is released (version 2.2.2), the transfer schemes can be updated and the conservation study to compare both methods must be performed. For now, we will just briefly comment both methods. The first method is more accurate when the wing chord tends to zero as the spanwise edge is closer. To examine this special case, let us assume the same discipline discretization is taken to distribute the load. If we use method 2, the two nodes closest to the wing tip have the same force distribution, while with method 1, the force is higher on the node associated with a higher chord, which is more accurate. Due to this reason, we employ method 1 to perform MFSEGO in section 5.8. Method 2 is more related with the implemented approach and uses shape functions to distribute the load. Shape functions are a common used technique in FEM to discretize an unknown field, usually the displacements. They are also employed to develop complex FSI schemes as presented in [16].

Additionally, for the new load transfer schemes based on the new OAS version (version 2.2.2), the moment arms will be defined from the structural nodes to the panels slice center of pressure, as it is depicted in figure 4.3. We can anticipate that for method 2 when the AD is much more refined than the SD, there is the possibility that the moment arms get to big for some nodes and consequently the moments get inaccurately too high. Figure 4.10 shows the moment arms associated to node 1 and node 2 in blue and orange, respectively. We see that the centers of pressure closer to node 1 and 2 produce big moment arms in node 2 and 1, respectively.

49
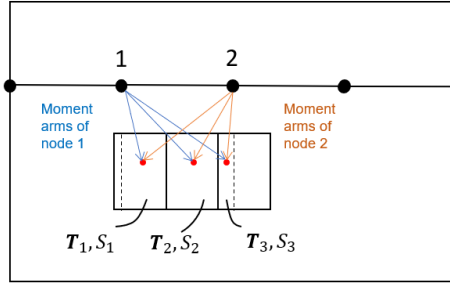
Figure 4.10: Moment arms associated to node 1 and 2 for method 2 when AD is more refined than SD.

4.4.2 Displacement Transfer

After establishing two load transfer schemes, let us now introduce the displacement transfer for non-identical discipline discretization.

As mentioned in section 4.4.1, the edges (at constant y) of the panels are aligned with the structural nodes for the same discipline discretization. The displacement transfer algorithm compares the y location of the structural nodes with the aerodynamic mesh nodes. In case this y coordinate of the aerodynamic mesh nodes does not match any of the structural nodes, a linear regression of the displacement with the information of the two surrounding structural nodes and the y coordinates is performed. Thus, we obtain displacements on a structural mesh with the same discretization as the aerodynamic mesh. Then, the displacement scheme concludes the computations by following the steps presented on section 4.3.2.

To clarify the scheme, let us introduce an example illustrated in figure 4.11. Since there are no structural nodes aligned with edge A, D, a linear interpolation with nodes 1 and 2 is computed to obtain the displacement of node 4 yielding
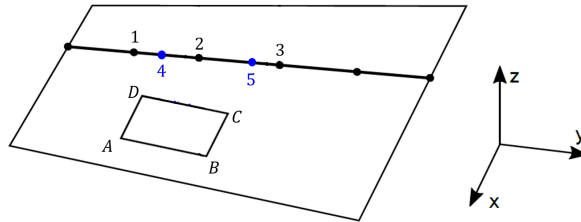
Figure 4.11: Scheme for transferring the displacement of the structural mesh to the aerodynamic mesh.

$$\mathbf{u}_{s,4} = \frac{\mathbf{u}_{s,2} - \mathbf{u}_{s,1}}{y_{s,2} - y_{s,1}} \cdot y_{s,4} + \mathbf{u}_{s,1} - y_{s,1} \left(\frac{\mathbf{u}_{s,2} - \mathbf{u}_{s,1}}{y_{s,2} - y_{s,1}} \right), \tag{4.19}$$

where $\mathbf{u}_{s,i}$ is the structural displacement associated with node i and $y_{s,i}$ is the y coordinate of node i. The same idea is applied in nodes 2 and 3 to produce node 5.

Chapter 5

Comparison of Optimization using Different Multi-fidelity Levels

In this chapter, it is presented an optimization study comparing the results of SEGO using one, two and three fidelity levels. Firstly, we define the optimization problem being minimized and the constants and optimizer parameters. Then, the multi-fidelity parameters are defined and the correlation between fidelity levels is explored. After the complete problem description, the results are thoroughly discussed.

5.1 Optimization Problem Definition

Let us start by summarizing our optimization problem, with 11 design variables and two constraint functions in table 5.1. The problem has taken from previous ONERA internal studies on the subject.

Table 5.1: Optimization problem definition.

	Function/variable	Note	Quantity	Bounds
minimize	fuel consumed	Breguet range equation	1	
w.r.t.	angle of attack		1	[8, 12]°
	wing twist	B-spline using 5 cp	5	[-6, 3]°
	spar thickness	B-spline using 5 cp	5	[0.0015, 0.05] m
	Total design variables		**11**	
subject to	lift $=$ weight	for cruise flight	1	
	$\sigma_{\text{von Mises}} \leq \frac{\sigma_y}{2.5}$	von Mises stresses using KS function	1	
	Total constraint functions		**2**	

The quantity we will be looking to minimize is the amount of fuel consumed during the flight. The fuel consumed is computed through the Breguet range equation given by

$$W_{\text{fuel}} = (W_s + W_0) \times \exp\Big(\frac{R.C_T}{V}\Big(\frac{C_L}{C_D}\Big)^{-1} - 1\Big), \tag{5.1}$$

where W_s is the wing structural weight, W_0 is the rest of the empty weight of the aircraft. In the exponent, R is the range, C_T is the specific fuel consumption, C_L is the lift coefficient, V is the flight speed and C_D is the drag coefficient. Equation (5.1) shows how the optimizer will minimize the amount of fuel spent: by means of reducing the wing weight and increasing the C_L/C_D ratio, keeping the other parameters constant.

The design variables wing twist and spar thickness are implemented by a spline spanwise distribution. We set the number of control points for these design variables and define a b-spline curve from these. Then, the values on the structural nodes are computed from the b-spline curve.

To ensure the non-failure of all finite elements, a von-Mises yield criterion evaluates the structural integrity. The material yield strength of the spar is represented by σ_y and a safety factor of 2.5 is used. We utilize a constraint aggregation method, based on the Kreisselmeier-Steinhauser function [59], thus, rather than define a constraint per each spar element, a single failure constraint needs to be applied.

5.2 Constants and Optimizer Parameters

Next, we specify the material properties and the flight conditions used in the optimization problem previously defined, presented in tables 5.2 and 5.3. These constants are for the commercial aircraft *B777-300* [60]. The material selected to model the beam elements is the aluminium 7075, a major constituent of the aircraft wings of the *B777* series [61]. The flight conditions and the thrust specific fuel consumption presented in table 5.3 are for cruise conditions.

Parameter	Value
Young's Modulus $E(Pa)$	70×10^9
Shear Modulus $G(Pa)$	30×10^9
Yield Strength $\sigma_y(Pa)$	500×10^6
Density $\rho_s(kg/m^3)$	3×10^3

Table 5.2: Spar material properties (Aluminium 7075).

Parameter	Value
Mach number M	0.84
Altitude $h(m)$	10.7×10^3
Air density $\rho(kg/m^3)$	0.38
Range $R(m)$	11×10^6
Empty weight $W_0(kg)$	1.2×10^5
Thrust specific fuel consumption C_T $(kg/W/s)$	1.54×10^{-5}

Table 5.3: Cruise flight conditions and thrust specific fuel consumption.

Lastly, we present the employed numerical tolerances to converge the MDA and the optimization problem, shown in table 5.4. The first two lines of the table 5.4 are the relative and absolute tolerances used to converge the Gauss-Seidel MDA solver, depicted in figure 3.1. The Gauss-Seidel approach solves a coupled system, a set of residual equations that represent the discipline analysis. The stopping criterion for each design point, that is, the MDA solution, is given by the two default tolerances of OAS.

The last line of table 5.4 is a relative tolerance used to converge the optimization problem when using the SLSQP optimizer.

Parameter	Value
Absolute error tolerance	10^{-7}
Relative error tolerance	10^{-30}
Optimizer tolerance	10^{-3}

Table 5.4: Gauss-Seidel MDA and SLSQP optimizer numerical tolerances.

5.3 Fidelity Levels and Associated Cost

The mesh discretization associated to each fidelity level (high, medium and low) is based on the mesh convergence study shown in figure 5.1.

The analyzed quantity to perform the mesh convergence is the structural failure of the beam elements. If structural failure is higher than 0, then failure has occurred in at least one beam element. The mesh convergence was performed analyzing different quantities, such as the fuel consumption and the lift coefficient (see appendix A.2). The structural failure proved to be the quantity that requires more mesh resolution to converge, thus it is the quantity used to select the different discretizations of each fidelity level.

The variables num_y and num_x in figure 5.1 are the spanwise and chordwise wing discretization, respectively introduced in section 4.2. As the mesh is progressively refined, the time to run the MDA increases and the model converges to -0.84. Obviously, a more refined mesh is a higher fidelity model. In industrial problems, the lowest fidelity level takes significant lower computational time to run the analysis than the high fidelity level model. Consequently, the results obtained from low fidelity level are usually less accurate than the high fidelity ones. Thus, we choose the less refined mesh of figure 5.1 to be the low fidelity mesh. As medium fidelity mesh, we choose the second less refined mesh. Although the structural failure of this mesh has not converged, it presents a great improvement upon the low fidelity mesh. The second most refined mesh of figure 5.1 is set as high fidelity mesh, since it presents excellent convergence results.
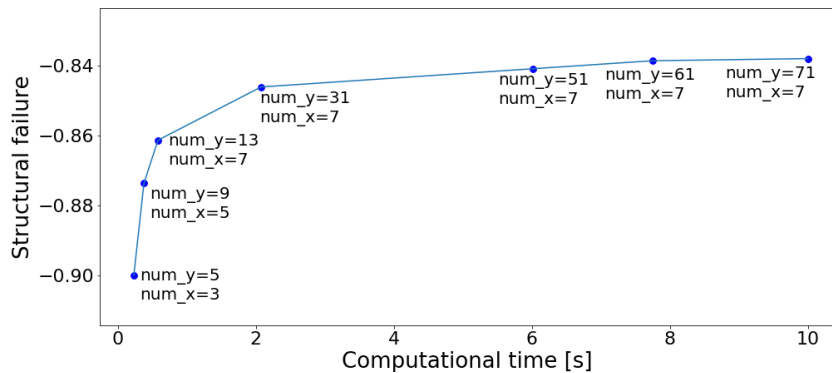
Figure 5.1: Structural failure mesh convergence study.

The discretization associated to each fidelity level is summarized in table 5.5 and it is illustrated in figure 5.2 (a), (b) and (c). Figure 5.2 shows the aerodynamic mesh and the structural finite elements.

Table 5.5: Discretization associated to high, medium and low fidelity levels.

Fidelity levels	High	Medium	Low
num_y	61	9	5
num_x	7	5	3

(a) High fidelity mesh.   (b) Medium fidelity mesh.   (c) Low fidelity mesh.
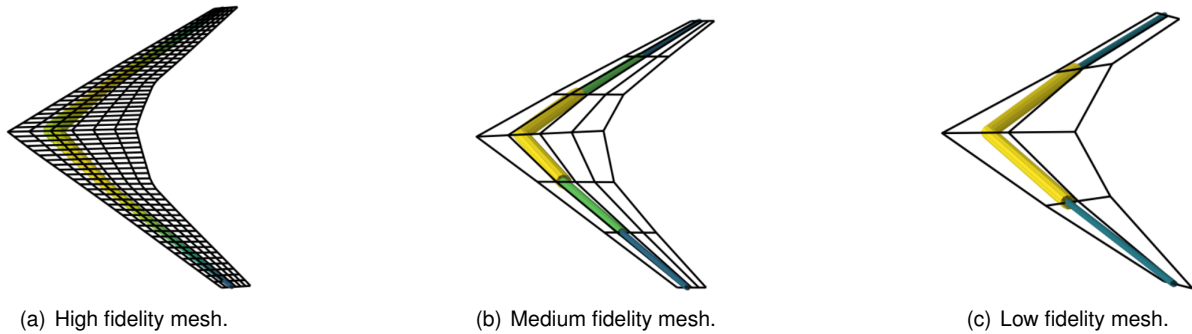
Figure 5.2: Wing discretization for varying fidelity meshes.

In industrial problems, the cost ratios imposed to a sample of each fidelity level are linked to the computational time needed to run the analysis. Here, as we are using the same low fidelity computational tool (OAS) for all fidelity levels, the time difference to run the analysis between the fidelity levels is not very significant. Consequently, the costs of each level obtained from the computational time to run the analysis are very different from real costs values. Since we are testing an optimization algorithm, we impose the costs of each fidelity level to be more related with real problems. The cost of the high fidelity samples is always set to 1 and thus, the other fidelity levels have smaller associated costs. The cost selected to the low fidelity samples is based on previous internal studies, where a multi-fidelity aerodynamic optimization problem with the costs related to the computational time analysis were studied. To select the cost of the medium fidelity samples, a linear interpolation between the selected costs for high and low samples and the structural failure values was performed from figure 5.1. We could have tested multiple costs ratios to study the algorithm behavior.

The cost ratios imposed to a sample of each fidelity level are summed up in table 5.6. As referred in section 2.1.3, Le Gratiet's formulation for multi-fidelity surrogate modeling requires nested DOE. Consequently, the real cost associated when performing enrichment through fidelity level k is the sum of the costs of fidelity level k to the lowest level. These costs are used to select the enrichment level of the next sampling $\mathbf{x}^*$ as shown in equation (2.36) in section 2.2.3.

Table 5.6: Cost ratios associated to each fidelity level.

Fidelity cost	High	Medium	Low
Normalized cost	1	50/125	1/125

5.4   Design of Experiments Sampling Size

The sampling size of the employed DOEs are summarized in table 5.7. The DOEs are partitioned in two different groups - modified DOE and complete DOE - each of them compound by two DOEs of two and three fidelity levels optimization (2 F and 3 F). Additionally, there is the DOE corresponding to one fidelity level optimization (1 F).

Table 5.7: Design of experiments sampling size.

| | | Number of fidelities | | | |
| | | Modified DOE | | Complete DOE | |
Fidelity level	1 F	2 F	3 F	2 F	3 F
Low	-	40	40	80	160
Medium	-	-	40	-	80
High	40	40	40	40	40

The complete DOE group considers that the number of DOE samples of the fidelity level k doubles the number of DOE samples of the most accurate consecutive fidelity level $k + 1$. Thus, half of the DOE samples of each fidelity level are equal to the samples of the most accurate consecutive fidelity level (nested DOE), and the other half of the samples is extra information to the model construction. The DOE from the complete group for 3 F is the only one that is computed using Latin hypercube sampling, explained in section 2.1.1, while the remaining four DOEs are obtained by filtering from it. To construct the modified DOE, the filtering process preserves the sample values that are nested, that is, equal. Consequently, each fidelity level has the same number of samples. The complete DOE of 2 F is achieved in the same way as the modified DOE of 2 F, but the first 40 non-nested samples are also considered to inform the model. Usually, the number of high fidelity samples must be three to four times the number of design variables of the problem. Thus, it was decided to use 40 high fidelity samples on each DOE.

The cost associated to obtain each of the DOEs can be computed through tables 5.6 and 5.7 and are presented in table 5.8. These are the initial costs before the optimization starts.

Table 5.8: Cost associated to each design of experiments.

| 1F | Modified DOE | | Complete DOE | |
	2 F	3 F	2 F	3 F
40	40.32	56.32	40.64	73.28

5.5   Correlation between Fidelity Levels

After we have chosen the mesh refinement for each fidelity level, we can study the correlation between them. To perform this study, we use the Pearson correlation coefficient given as

$$\rho = \frac{\sum_{i=0}^{n}(x_i - \overline{x})(w_i - \overline{w})}{\sqrt{\sum_{i=0}^{n}(x_i - \overline{x})^2}\cdot\sqrt{\sum_{i=0}^{n}(w_i - \overline{w})^2}} = \frac{\mathrm{cov}(X,W)}{\sqrt{\mathrm{var}(X).\mathrm{var}(W)}}, \tag{5.2}$$

where n is the sample size, x_i and w_i are individual sample points, and $\overline{x}$ and $\overline{w}$ are the mean of the variables. The cov() is the covariance between X and W and var() is the standard deviation of X and W. The Pearson correlation coefficient measures the strength of the linear correlation between two variables. The coefficient has a value between 1 and -1, where the value of 1 corresponds to a perfectly positive linear relationship, 0 represents no linear correlation, and 1 represents a perfectly negative linear correlation.

To evaluate the Pearson correlation between fidelity levels, we use 100 run tests of a DOE with 40 samples of each fidelity level, as the 3 F modified DOE, presented in table 5.7. The Pearson coefficient is measured between high fidelity and medium fidelity and high fidelity and low fidelity samples for the objective function and the two constraints functions of the optimization problem, defined in table 5.1. We depict the results in a box plot format (figures 5.3 and 5.4), where the maximum, minimum, mean and median values are shown to help the analysis.

(a) Fuel consumption.   (b) Trimmed flight.   (c) Structural failure.
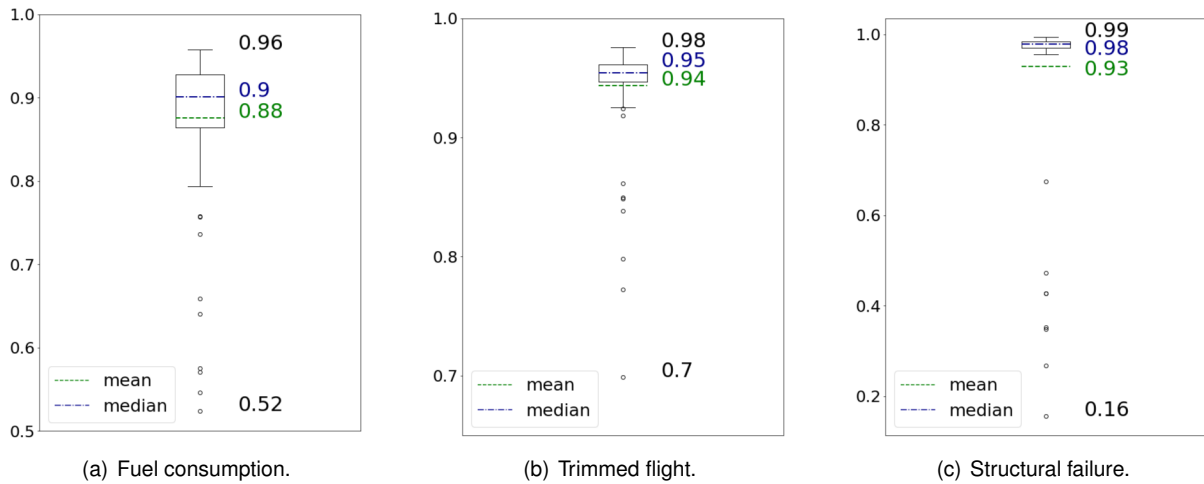
Figure 5.3: Pearson correlation between high fidelity and medium fidelity DOE samples for different functions of interest.

Examining figure 5.3, we observe that the majority of the high fidelity and medium fidelity samples have a Pearson coefficient close to one, that is, these samples have a strong positive linear relationship. This does not mean that high and medium fidelity samples provide the same values for the three quantities, but rather that, for the majority of the DOE, a linear relationship is verified between the 40 samples of high and medium fidelity. Thus, one can conclude that the medium fidelity samples add valuable information to the model. Nevertheless, there are some outliers DOE, where this almost perfect linear relationship is not verified.

Figure 5.4 shows that the Pearson correlation between high fidelity and low fidelity samples is very different from figure 5.3. The lower and upper quartile values, the maximum and minimum values of the box plots have decrease significantly. Thereby, we conclude that the quality of the information provided by the low fidelity samples to the model is very dependent on the DOE. Some have good or reasonable

(a) Fuel consumption. (b) Trimmed flight. (c) Structural failure.
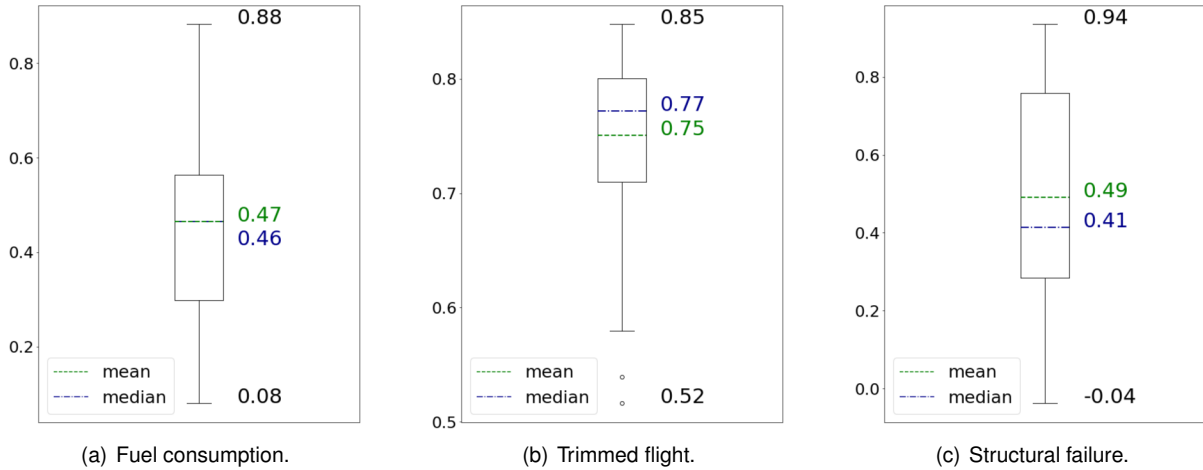
Figure 5.4: Pearson correlation between high fidelity and low fidelity DOE samples for different functions of interest.

correlations, but others have such low correlations that, instead of enrich, may erratically inform the surrogate model and, therefore, negatively influence the optimization. We will explore this idea in the next sections. Figure A.4, in appendix, represents the low, medium and high fidelity values for the objective function and the two problem constraints of the first (out of 100) 3 F modified DOE. This figure exemplifies the previous conclusions, since we observe stronger linear correlation between high and medium fidelity samples than between high and low fidelity samples.

In statistical analysis, the p-value is a number between zero and one representing the probability of obtaining test results at least as extreme as the results actually observed, under the assumption that the null hypothesis is correct [62]. The null hypothesis states that there is no relation between the data. A very small p-value means that such an extreme observed outcome would be very unlikely to have no relation. A threshold value for the p-value, called the significance level, is chosen. This threshold is usually between one and five percent. If the p-value is smaller than the threshold, then the null hypothesis can be rejected. The p-values of all the analysis above are mainly smaller than three percent, except for a few outliers.

5.6   Multi-fidelity Parameters Definition

Built on the five types of DOE previously presented, the kriging surrogate model is constructed using the PLS extension (KPLS) with four principal components. In general, its regression term is considered to be constant or linear. A parametric study using ten optimizations was made varying this parameter and slightly better results were found using the linear term. The parameters are summarized in table 5.9.

The optimization budget is a parameter to control the desired computational time to solve the problem and it is managed by summing progressively the associated cost of each point to be sampled during the optimization and stopping the optimization process when the budget has been reached. The chosen optimization budget is a trade-off between having acceptable computational time for several optimization

57

runs and acceptable objective function results. The objective function results are considered to be acceptable when they are close or smaller than those of the reference solution presented in section 5.7.1. After some experiments, the budget of 110 was defined. It is important to note that this budget is only to the optimization process and it does not include the cost associated to each DOE presented in table 5.8. As referred in section 2.2, the infill sampling criterion employed is the WB2. The optimizer used to maximize the WB2 criterion at every iteration step is the SLSQP. The optimization parameters are summed up in table 5.10.

Surrogate model	Number of components	Polynomial type
KPLS	4	linear

Table 5.9: Surrogate model parameters.

Optimization budget	ISC	Optimizer
110	WB2	SLSQP

Table 5.10: Optimization parameters.

The Root Square Constraint Violation (RSCV) is used to evaluate the results in terms of constraint violation and it is given by

$$RSCV = \sqrt{C_1^2 + C_2^2},$$ (5.3)

where C_1 is the equality constraint, lift=weight, and C_2 is the inequality constraint, von Mises stress for wing failure. If the inequality constraint is positive, then the wing has failed and the constraint is violated. Thus, we solely consider C_2 in the RSCV calculation when it is positive. The common logarithmic (log) RSCV is employed during the results analysis for simplicity.

5.7 Comparison of Different Multi-fidelity Levels in Optimization

In the next sections, let us compare the optimization results of 100 run tests for the five types of DOE previously described. We start by the simplest optimization with one fidelity level progressing up to the more complex ones employing samples of three fidelity levels.

5.7.1 Reference Solution

For each of the DOE types, we filter the 100 run tests based on a reference solution. This solution, shown in table 5.11 along with the corresponding log (RSCV) value, is obtained using the SLSQP optimizer, the numerical tolerances presented in table 5.4 and the high fidelity mesh discretization. The reference solution is presented with more detail in appendix A.7.1. Although one can use the number of analyses performed to obtain the reference solution as a reference cost, it is not a good term of comparison with the gradient-free solutions because the SLSQP optimizer uses derivatives that should be accounted to make a fair comparison. Our objective is, based on this reference solution, to define the minimum requirements that the solutions of the 100 run tests must satisfy. Thereby, we are looking for a tolerance to apply to the objective function solution and a maximum log (RSCV) value. Together, they

are going to define the maximum threshold of our filtering process. These limits are defined based on a trade-off relationship. It is desired that a significant amount of run tests fulfills the requirements to allow significant data statistical studies, where we can observe tendencies and take conclusions. However, it is also desired that the filtered results are close to the reference solution and have an acceptable degree of constraint violation. Naturally, if the chosen tolerance and the log (RSCV) value are too high, some of the filter results have poor quality. In contrast, if the chosen parameters are too low, we loss optimization results that are important in the conclusion making. After some tests, a tolerance of 1% is applied to the reference solution and the maximum log (RSCV) value of -2.5 is defined. The two last lines of table 5.11 show the defined threshold for the filtering process.

Table 5.11: Reference solution and maximum acceptable values to filter the results.

Parameter	Value
Reference objective function solution	1.06801×10^5 Kg
log (RSCV)	-5.23
Maximum acceptable objective function solution	1.07869×10^5 Kg
Maximum acceptable log (RSCV)	-2.5

During the filtering process of the 100 run tests, we are seeking for the best solution of each test. This solution is the one that has the lowest objective function (fuel consumption) value and fulfills the established requirements. Table 5.12 shows the number of optimizations for each DOE type that is conserved after the filtering process. The number of solutions that fulfilled the requirements has the same order of magnitude, except for the 3 F optimizations. We will explore the reasons for this phenomenon in section 5.7.4.

Table 5.12: Number of optimizations conserved after the filtering process.

1 F	Modified DOE		Complete DOE	
	2 F	3 F	2 F	3 F
74	72	32	78	43

5.7.2 One Fidelity Level Optimization

The filtered best solutions of one fidelity level optimization are shown below in the format of three box plots and one scatter plot. Figure 5.5 represents the box plots for three quantities, the log (RSCV) (a), the solution cost (b) and the fuel consumption (c). Each of the box plots have some statistical data (maximum, minimum, mean and median) on the right side to help the analysis, similarly to the Pearson box plots in section 5.5. Note that the cost box plot includes the cost associated to the building of the initial DOE (see table 5.8). Analyzing figure 5.5 (a), the log (RSCV) values are slightly concentrated in the higher values. The costs, represented in figure 5.5 (b), vary in a wide range, but are also clustered at the higher values. The fuel burn, in figure 5.5 (c), shows that there is a tendency for the acceptable solutions to be concentrated towards the lower values of the range.

(a) Log (RSCV) box plot.   (b) Cost box plot.   (c) Fuel consumption box plot.

Figure 5.5: Box plots of log (RSCV), solution cost and fuel consumption for the best solutions of the 1 fidelity level optimization.

Figure 5.6 shows the scatter plot where each circle represents a best filtered solution associated with a cost and a fuel consumption value in the vertical and horizontal axis, respectively. Moreover, the color of the circle provides a qualitative representation of the log (RSCV) value of that solution. The more negative the log (RSCV) value is, the less the solution violates the constraints. The vertical green line represents the reference solution presented in table 5.11. The scatter plot relates the information of the three box plots and enables to get data relations. Figure 5.6 depicts a cluster of circles in the lower values of fuel consumption, a result aligned with the fuel consumption box plot conclusions. Additionally, we see that this data cluster is mainly associated with the highest costs of box plot 5.5 (b).
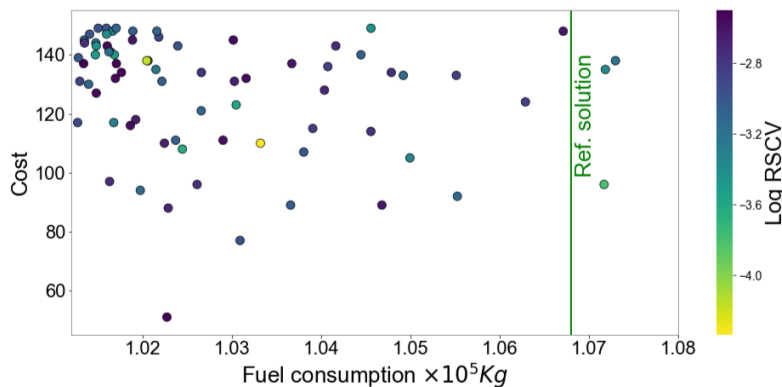
Figure 5.6: Scatter plot for the best solutions of 1 fidelity level optimization.

The results from all the optimizations will be presented in these format and will be compared with the one fidelity level optimization that will be used as a benchmark solution.

5.7.3   Two Fidelity Levels Optimization

Let us start by comparing the results of one fidelity level optimization with two fidelity levels optimization using the complete DOE (hereinafter referred to as 1 F and 2 F complete DOE, respectively). Figure 5.7 shows the box plots for three quantities, log (RSCV), cost and fuel consumption for the 2 F complete DOE.

(a) Log (RSCV) boxplot.   (b) Cost box plot.   (c) Fuel consumption box plot.
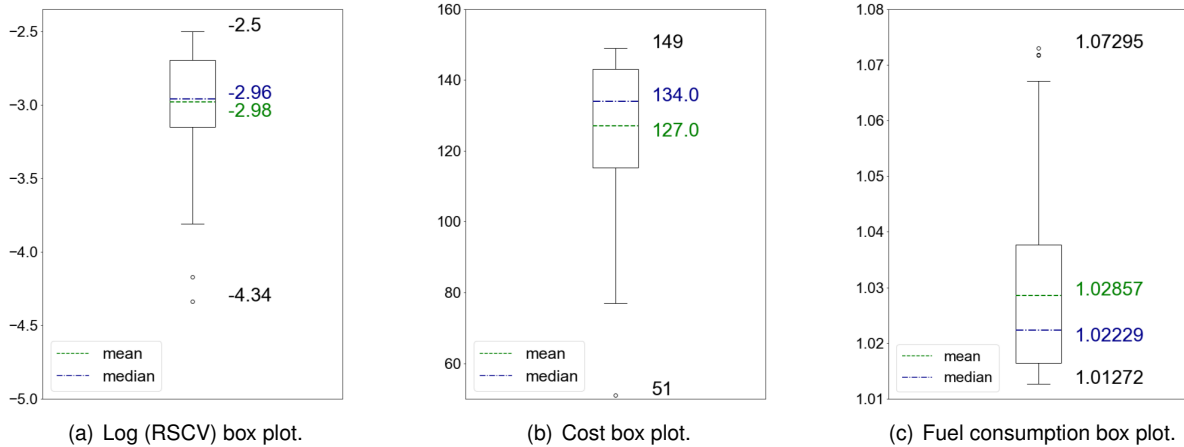
Figure 5.7: Box plots of log (RSCV), solution cost and fuel consumption for the best solutions of the 2 fidelity levels optimization using complete DOE.

Observing the RSCV box plots, figures 5.5 (a) and 5.7 (a), we see that the median of the 2 F complete DOE results is lower than that of the 1 F results. The interquartile range for the 2 F complete DOE is larger, the upper quartile is greater, and the lower quartile is lower when compared with 1 F. The same verifies for the lower adjacent value of the 2 F complete DOE that is lower than that of the 1 F. The batches have very close means, and the 2 F complete DOE is more upper skewed. This way, although half of the RSCV results of the 2 F complete DOE are in a smaller range than the ones of 1 F, the extension of the interquartile and the lower adjacent limit to lower values prevent from concluding which of the approaches has better constraint violation.

Comparing now the cost box plots, figures 5.5 (b) and 5.7 (b), we observe that for the 2 F complete DOE the median and mean costs are greater and the data set distribution is less disperse and more upper skewed. Thus, we can conclude that the costs of the 2 F complete DOE best solutions are more likely to reach the maximum budget. This means that this approach is progressively getting better results through the optimization process, whilst the 1 F tends to find the best solution earlier and the rest of the budget is spent without getting a better result than the one already obtained.

The cost analysis solely is not sufficient to conclude which of the approaches is better, since it depends on the quality of the objective function values associated to these costs. Thereby, we now discuss the last box plots relative to the fuel consumption, presented in figures 5.5 (c) and 5.7 (c), that suggest that the 2 F complete DOE best solutions tend to have lower fuel consumption values, because the median and mean are lower and the data set is more concentrated in the low values, that is, it is more down skewed.

Finally, we analyze the scatter plots 5.6 and 5.8, which connect the cost, the fuel consumption and the log (RSCV) quantities of the solutions and from which we can conclude which is the best approach. Starting by the log (RSCV) values, no clear conclusion can be taken between the log (RSCV) and the other two quantities. The first impression when observing both plots is the data clustering in the upper left corner of the 2 F complete DOE solutions in contrast to the 1 F scattered solutions. We can then conclude that, for the 2 F complete DOE, the fuel consumption concentration previously observed in box
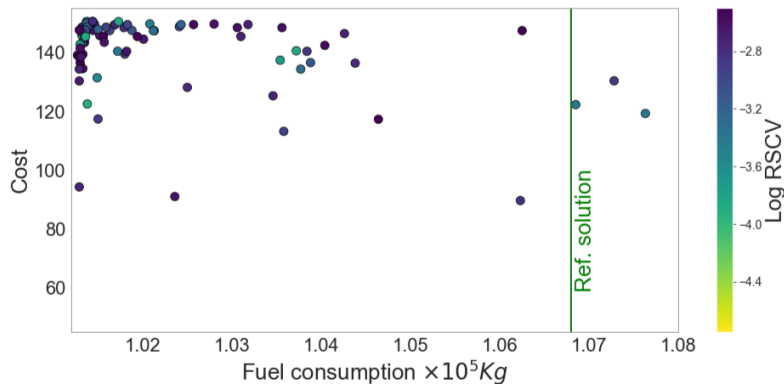
Figure 5.8: Scatter plot of the best solutions of 2 fidelity levels optimization using complete DOE.

plot 5.7 (c) is in general associated with the highest costs. Although, from the 1 F scatter plot, we can observe a slight trend of the best solutions values being associated with the higher costs, the relation is not so clear as for the 2 F complete DOE.

The 2 F approach not only starts with a more informed model, but also usually starts with the exploration of the design space by querying low fidelity (LF) samples and then the exploitation phase, where mainly high fidelity (HF) samples are queried, as previously explained in section 2.2.1. In the first stage, the optimizer reduces the uncertainty of the acquisition function, allowing a more efficient decision when querying HF samples. The data clustering in the scatter plot shows that this approach manages most of the times to find low fuel consumption values with different initial DOE, providing robustness to the method. Some best solutions have the highest budget, but not low fuel consumption values. These are solutions that, due to the quality of the DOE, would probably need higher budgets to achieve better solutions. As seen in section 5.5, the quality of the DOE is related with the Pearson coefficient and some DOEs have worse correlation between the samples than others. In contrast, the 1 F optimization, that only queries HF samples, is less complex than the 2 F approach, but it lacks of robustness. Due to the fact that there is no introductory space exploration neither LF samples in the initial DOE, the optimizer is deciding more randomly which samples to query. This way, the best solution is less often the lower one and the associated cost to this solution is very variable. A good solution is much more dependent on quality of the DOE used to initialize the problem. The quality of the DOE referred here is not related with the Pearson correlation, since the DOE has only HF samples, but rather the samples position that also influence the optimization result.

Thereby, we can conclude that the 2 F complete DOE approach is the one that provides the best solutions more often and so, we proceed the performance comparison between 2 F optimizations using complete and modified DOE (hereinafter referred to as 2 F complete DOE and 2 F modified DOE, respectively). Figure 5.9 shows the box plots for log (RSCV), cost and fuel consumption of 2 F modified DOE.

Starting again by the log (RSCV) box plot distributions, represented in figures 5.7 (a) and 5.9 (a), we see that the mean and median values of the 2 F modified DOE solutions are greater than that of the 2 F complete DOE solutions. Moreover, the interquartile range is smaller, and the lower adjacent limit is greater for the 2 F modified DOE solutions. We can then conclude that these data set distribution is

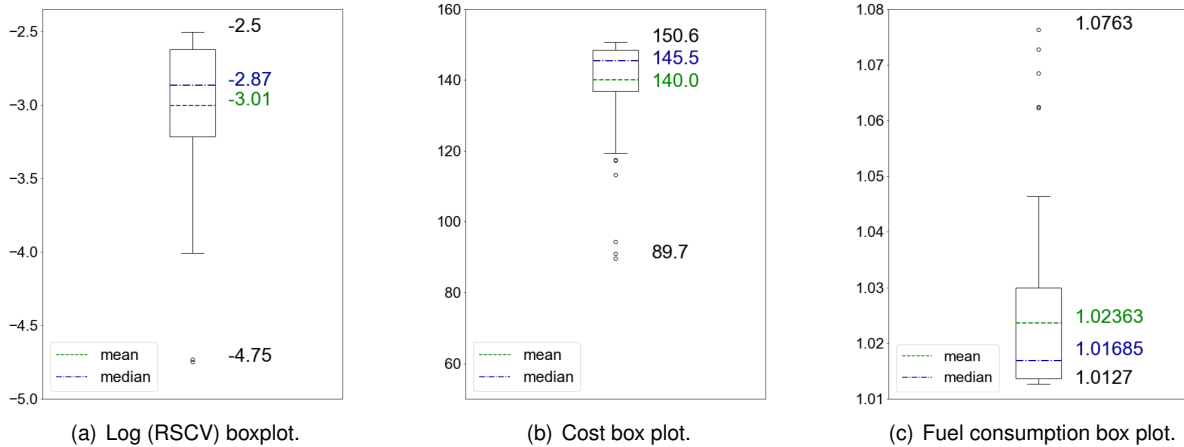(a) Log (RSCV) boxplot.　　　　　(b) Cost box plot.　　　　　(c) Fuel consumption box plot.
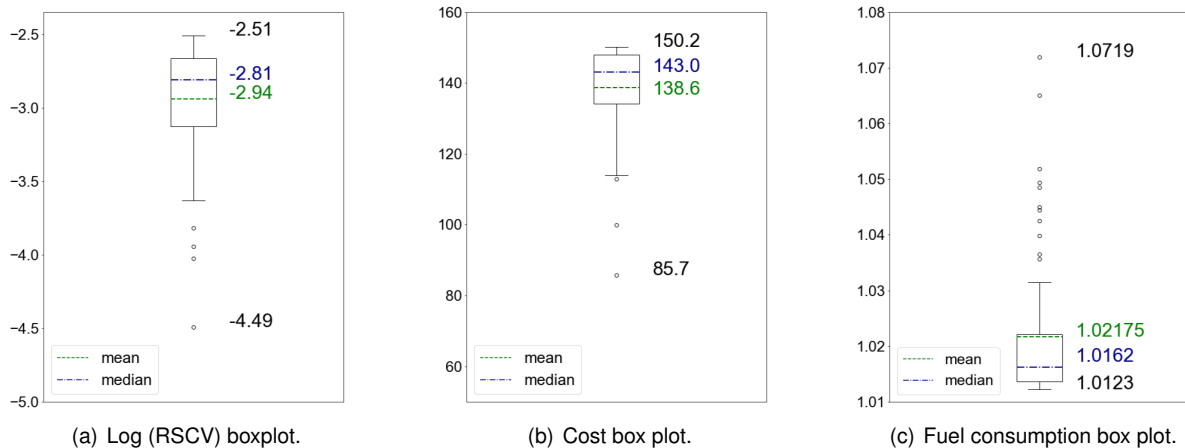
Figure 5.9: Box plots of log (RSCV), solution cost and fuel consumption for the best solution values of the 2 fidelity levels optimization using modified DOE.

less disperse. Although both batches are upper skewed, the 2 F modified DOE solutions have a more marked lack of symmetry. Therefore, one can conclude that the 2 F modified DOE produces solutions with higher constraint violation. The modified DOE supplies less information (less 40 LF samples) to construct the initial surrogate models of the objective function and constraints. We do not know the quality of these 40 LF samples for each DOE (how well they inform the model), but we can try to extract some conclusions from the Pearson coefficients, in figure 5.4. Observing this figure, we see that the coefficients of the trimmed flight constraint are much better than the failure constraint, since for the first constraint half of the DOEs samples have a correlation equal or higher than 0.77 and for the second constraint this value is 0.41. Thus, the information loss when we remove the 40 LF samples of the DOEs may be significant for the constraints surrogate models. Naturally, less informed constraint surrogates will tend to produce solution with higher constraint violations.

Next, let us discuss the cost distribution box plots, shown in figures 5.7 (b) and 5.9 (b). The mean and the median values are lower for the 2 F modified DOE solutions. The overall range has expanded due to the lower values of the lower quartile and lower adjacent limit. The data distribution is less upper skewed. Then, we conclude that the costs of the 2 F modified DOE solutions are more likely to be lower than that of 2 F complete DOE solutions.

The fuel consumption box plots (shown in figures 5.7 (c) and 5.9 (c)) show that the mean and the median values decrease also for the modified DOE solutions. The data dispersion is lower due to the lower values of the upper quartile and upper adjacent limit. We observe that half of the lower solutions of the complete and modified DOE approaches are approximately in the same range. But, for the upper half, the range extension is significantly higher for the complete DOE approach. Therefore, the fuel consumption values of the 2 F modified DOE solutions have a tendency to be lower than that of 2 F complete DOE.

Similarly to 1 F and 2 F complete DOE, figure 5.10 shows the same type of scatter plot for the 2 F modified DOE results. Observing the scatter plots 5.8 and 5.10, we start by noting that, again, it is difficult to detected any relation between the log (RSCV) values and the other two quantities. We
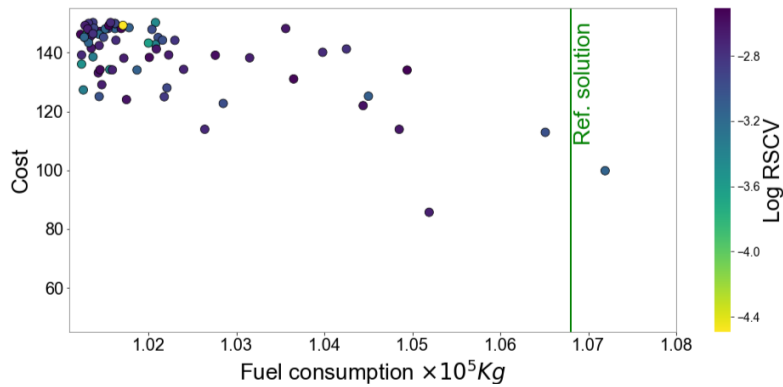
Figure 5.10: Scatter plot of the best solutions for 2 fidelity levels optimization using modified DOE.

also see that both have a data clustering in the upper left corner of the distributions, that is, the lower fuel consumption values are associated with the maximum costs. This means that there is progressive improvements in the course of both optimizations. We also observe that for the 2 F complete DOE there are more scattered solutions across the domain, a pattern which is in line with the fuel consumption box plot. Those scattered solutions are also more often associated with the maximum costs.

Thereby, the modified DOE approach is more likely to provide better solutions outcomes, which appears to be an odd tendency, since the problem is initiated with less information. Again, let us observe the Pearson coefficients in figure 5.4 to extract some conclusions. We see that for the HF and LF samples of the fuel consumption half of the DOEs have Pearson coefficients between 0.08 and 0.47. This means that a significant part of the DOEs have low Pearson coefficients and consequently low quality. Thus, for some DOEs, these 40 non-nested LF samples included in the complete DOE data set are erratically informing the kriging and inducing the optimizer into less promising areas of the design space where the budget is spent. These 40 non-nested LF samples are responsible for the larger data scattering observed in the complete DOE optimization.

Let us investigate the optimization process by having a deeper look into the queried samples. For each of the run tests that fulfilled our requirements for the 2 F approaches, we collect the number of non-nested HF and LF samples queried through the entire optimization process. Then, we plot figure 5.11 (a), (b), (c) and (d) that associate the fuel consumption and the log (RSCV) value of the best solution with the number of samples queried per fidelity. Figure 5.11 (a) and (c) correspond to the log (RSCV) and fuel consumption, respectively, for the complete DOE approach, while figure 5.11 (b) and (d) correspond to the log (RSCV) and fuel consumption, respectively, for the modified DOE. The different size of the dots of figure 5.11 is associated with the number of LF samples used in the optimizations. A larger dot means that more LF samples were queried.

Tables 5.13 and 5.14 complement figure 5.11 offering a more quantitative insight. The intervals of LF samples represent the minimum and the maximum number of LF samples that were queried for the corresponding number of HF samples. The associated percentages are in relation to the number of acceptable run tests of each approach, presented in table 5.12. The number of filtered solutions for the complete and modified DOE is 78 and 72, respectively.

One may ask how the optimizer can choose 110 HF samples and still LF samples without wasting

(a) Complete DOE log (RSCV)	(b) Modified DOE log (RSCV)
(c) Complete DOE Fuel consumption	(d) Modified DOE Fuel consumption
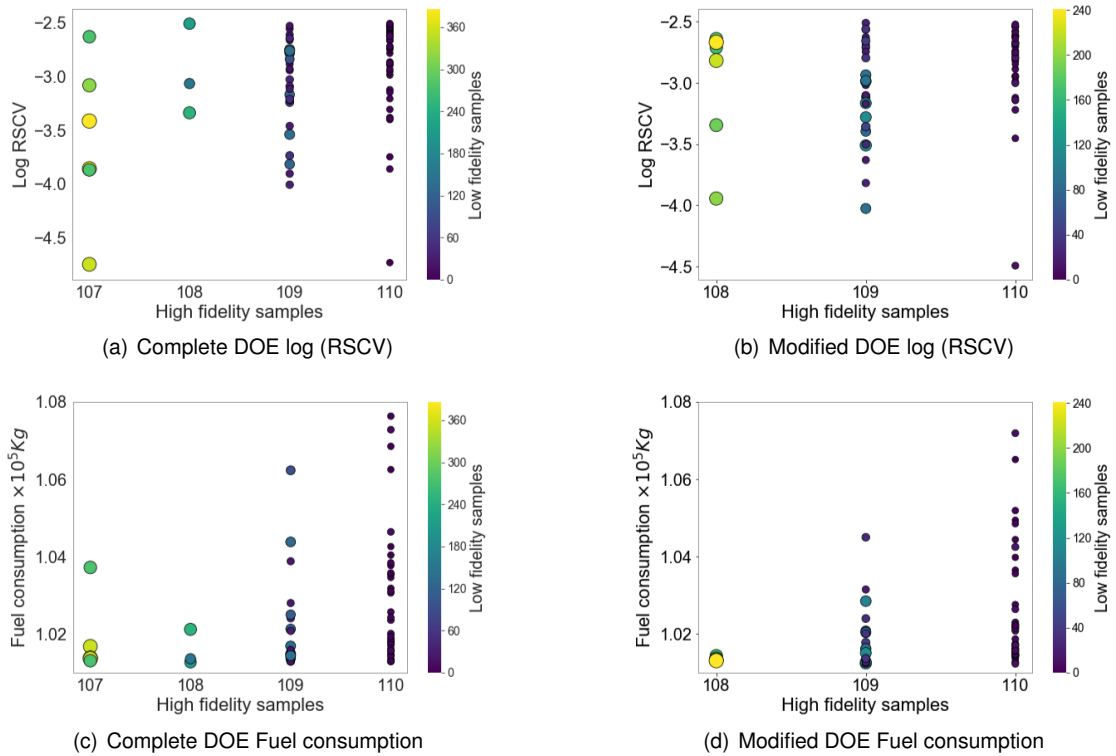
Figure 5.11: Scatter plot of the amount of samples queried per fidelity level in function of the log (RSCV) and fuel consumption quantities for the 2 fidelity levels optimization using complete and modified DOE.

Table 5.13: Percentage of filtered solutions by the number of queried samples per fidelity level for the 2 fidelity levels optimization using complete DOE.

High fidelity samples	107	108	109	110
Low fidelity samples	[275, 386]	[152, 248]	[19, 138]	[0, 16]
Percentage of optimizations	7.7%	3.8%	34.6%	53.8%

Table 5.14: Percentage of filtered solutions by the number of queried samples per fidelity level for the 2 fidelity levels optimization using modified DOE.

High fidelity samples	108	109	110
Low fidelity samples	[167, 241]	[18, 128]	[0, 15]
Percentage of optimizations	8.3%	38.9%	52.8%

the budget. In fact, the optimizer only stops picking samples when the established budget is reached or exceeded. Since the LF samples have a very low cost, sometimes even having queried LF samples, the budget has not been reached yet and it is still possible to query 110 HF samples. However, in such situations, the budget is always exceed.

At the first glance, we see that the two optimization approaches use a significant amount of HF samples. Moreover, from tables 5.13 and 5.14, more than half of the filtered solutions uses the maximum number of HF samples and few optimizations picked a larger number of LF samples. We also observe that the complete approach chooses four different quantities of HF samples, unlike modified approach that only chooses three. The extra quantity corresponds to 107 HF samples and it is chosen by a small

portion of the complete DOE optimizations that naturally, uses more LF samples. Nevertheless, having a longer exploration phase does not mean worst results. So, for which situations does the complete DOE provide worst results than the modified DOE?

Examining figure 5.11, we observe that the maximum values of the fuel consumption for the complete DOE approach are greater than the modified DOE, for the same number of queried HF samples (110, 109 and 108). Thus, the 40 non-nested DOE that sometimes erratically inform the model and make the algorithm explore less promising areas lead the optimizer to start the exploitation phase focusing on those areas.

Analyzing how the number of DOE per fidelity level relates with the log (RSCV) values, we see that there is no clear relation. We also see that the best fuel consumption values are associated with the optimizations where more non-nested LF samples were picked, while when the exploration phase (solely with LF samples) is not executed or is very poor, there is a higher probability of not getting such good results. Observing the enrichment level criterion in equation (2.36), in section 2.2.3, we see that by increasing the difference between the cost levels, we can increase the number of non-nested LF samples queried. Thus, one should select tools to enable cost reduction, such as using a less refined discretization for the low fidelity samples. It is expected that increasing the costs difference between fidelity samples, not only the best solutions of the acceptable run tests improve, but also the number of acceptable run tests increase. Please, see in appendix A.4 the optimization history of the 59th acceptable run test for two fidelity levels optimization using modified DOE, where further conclusions are taken.

5.7.4   Three Fidelity Levels Optimization

In this section, we discuss the three fidelity levels optimization results using complete and modified DOE (hereinafter referred to as 3 F complete DOE and 3 F modified DOE, respectively).

We start by comparing the log (RSCV) box plots of the 3 F solutions with the 1 F shown in figures 5.12 (a), 5.14 (a) and 5.5 (a). The batches of the 3 F have a smaller range and are more upper skewed, so the provided solutions have higher constraint violations. This result suggests that the 3 F approaches use much less HF samples than the 1 F and the 2 F approaches, compromising the log (RSCV) solutions.

Next, making the same analysis between the cost box plots of 3 F and 1 F optimizations, depicted in figures 5.12 (b), 5.14 (b) and 5.5 (b), we notice a behavior similar to that observed for the log (RSCV) quantity. As the DOE of the 3 F is significantly more expensive than the 1 F DOE (see table 5.8), this tendency can be anticipated. We are interested in discovering if this general cost increase generated better results than the 1 F results.

Thus, we compare the fuel consumption box plots of the 3 F and the 1 F, presented in figures 5.12 (c), 5.14 (c) and 5.5 (c). We observe that the overall batches of the 3 F approaches have greater median, mean, minimum and maximum values. The interquartile range of both approaches is wider and its limits are higher. Moreover, the data distribution is less down skewed. We then conclude that the 3 F optimizations have a tendency to provide fuel consumption results much worse than the 1 F approach. The skewess loss suggests that the data is more scattered through the domain as we will confirm by the

scatter plots.

(a) log (RSCV) box plot.

(b) Cost box plot.

(c) Fuel consumption box plot.

Figure 5.12: Box plots of log (RSCV), solution cost and fuel consumption for the best solution values of the 3 fidelity levels optimization using complete DOE.

Figure 5.13: Scatter plot of the best solutions for 3 fidelity level optimization using complete DOE.

Finally, we analyze the scatter plots of 3 F and 1 F optimization, presented in figures 5.6, 5.13 and 5.15, and observe that the results of the 3 F approaches have higher costs and are less clustered than the best fuel consumption values, as expected.

The 3 F optimizations not only tend to provide worst results with higher costs, but also have more solutions that failed to fulfill the requirements, as noted in table 5.12. We can then conclude that the 3 F optimizations are less efficient than the 1 F and, consequently, less efficient than the 2 F approaches. The reason for this tendency may be related with less HF samples chosen during the optimization. The extra fidelity level and the necessity of nested DOE increases the cost of those samples, making the optimizer choosing less of them. We will investigate the queried samples of the 3 F approaches, as we did for the 2 F approaches, in order to better understand why the 3 F provides such bad results.

Before we have a deeper look into the optimization process, let us first compare the two 3 F optimizations results. Starting by the log (RSCV) box plots presented in figures 5.12 (a) and 5.14 (a), we see that in one hand, for the 3 F modified results, the lower quartile limit and lower adjacent value are higher, but the median is lower and the mean is similar. Thus, no clear conclusion about the results constraint violation can be take. Next, analyzing the cost box plots of the 3 F approaches, depicted in figures 5.12

(b) and 5.14 (b), we observe that the 3 F modified DOE approach is shifted down, that is, it is associated with lower cost values. The cost to build the 3 F modified DOE is lower than to build the complete DOE, as presented in table 5.8, so this is an expected behavior. Analyzing now the fuel consumption box plots, in figures 5.12 (c) and 5.14 (c), we see that the upper quartile limit of the modified DOE is higher, making the interquartile range of this approach higher. In addition, the median and mean values are also higher and the data distribution is more symmetric, that is, less down skewed. Then, we conclude that the 3 F modified DOE tends to produce more solutions with higher objective function values than the complete DOE.

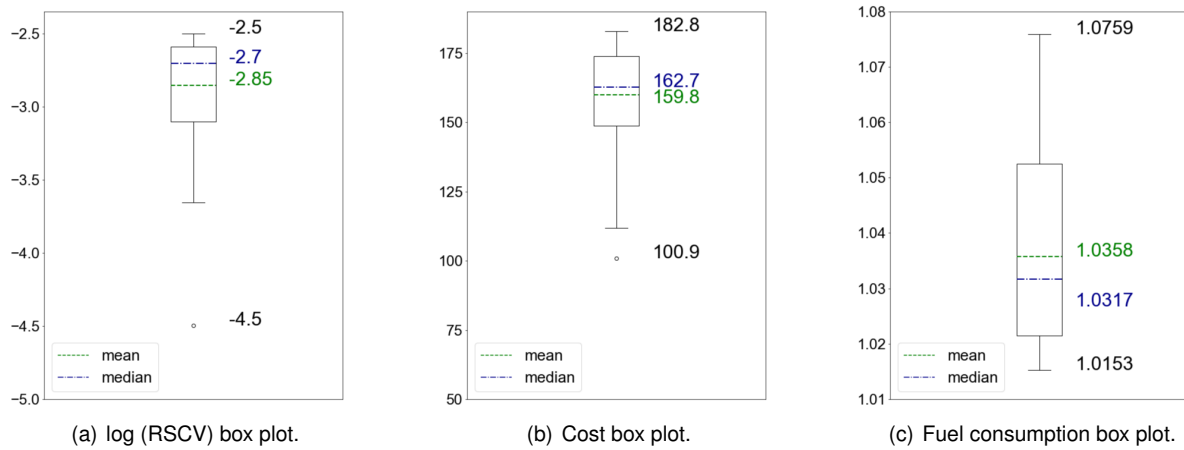| (a) log (RSCV) box plot. | (b) Cost box plot. | (c) Fuel consumption box plot. |
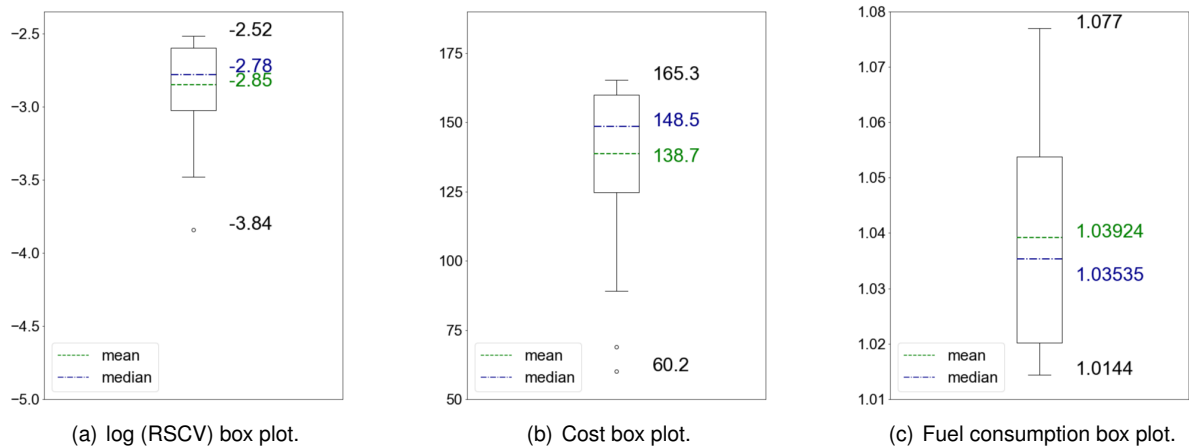
Figure 5.14: Box plots of log (RSCV), solution cost and fuel consumption for the best solution values of the 3 fidelity levels optimization using modified DOE.
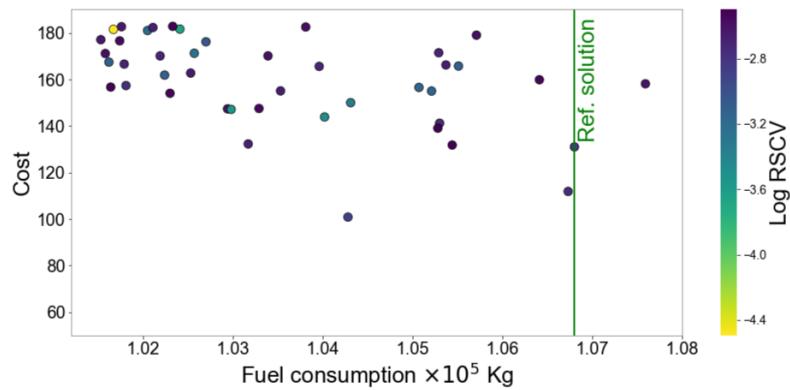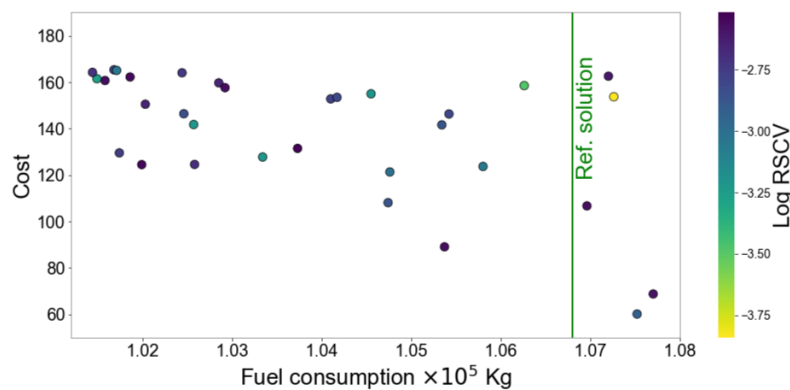

Figure 5.15: Scatter plot of the best solutions for 3 fidelity levels optimization using modified DOE.

Indeed, observing the scatter plots 5.13 and 5.15, we verify that, although the modified DOE approach produces less costly results, they are more scattered so the performance of this approach is worse than the performance of the 3 F complete DOE optimization. Additionally, the number of run tests that fulfilled the requirements established in section 5.7.1 is smaller for the 3 F modified DOE, as it can be seen in table 5.12. The reason for this is related with the difference between the complete and modified DOE of 3 F. The DOE of the modified DOE has less 120 LF and 40 Medium Fidelity (MF) samples than the complete DOE. As commented previously, the LF samples can have very low Pearson coefficients, so the information provided by the LF for the initial model can be incorrect. However, observing

figure 5.3, the Pearson coefficient between the HF and the MF samples is very high for the most of the DOE. Thus, the information provided by the MF to the initial model is very reliable. Thereby, we conclude that, in contrast to the 2 F optimizations, the information loss from the complete to the modified DOE in the 3 F optimizations is significant to the model and negatively influence the performance.

Let us now analyze the type of samples queried during the 3 F optimization process, as we did for the 2 F approaches in the previous section. Once again, the number of non-nested HF, MF and LF samples are collected for each acceptable run test. We constructed figure 5.16 that depicts the number of queried samples by fidelity in function of the fuel consumption. The number of HF samples is always represented in the horizontal axis, while MF and LF samples quantity is represented by the graph's hue. The plots aligned vertically are for the same approach (complete or modified DOE), the hue of the upper plots represents the MF samples and the hue of the down plots represents the LF samples. Since

(a) Complete DOE - Medium fidelity samples

(b) Modified DOE - Medium fidelity samples

(c) Complete DOE - Low fidelity samples

(d) Modified DOE - Low fidelity samples
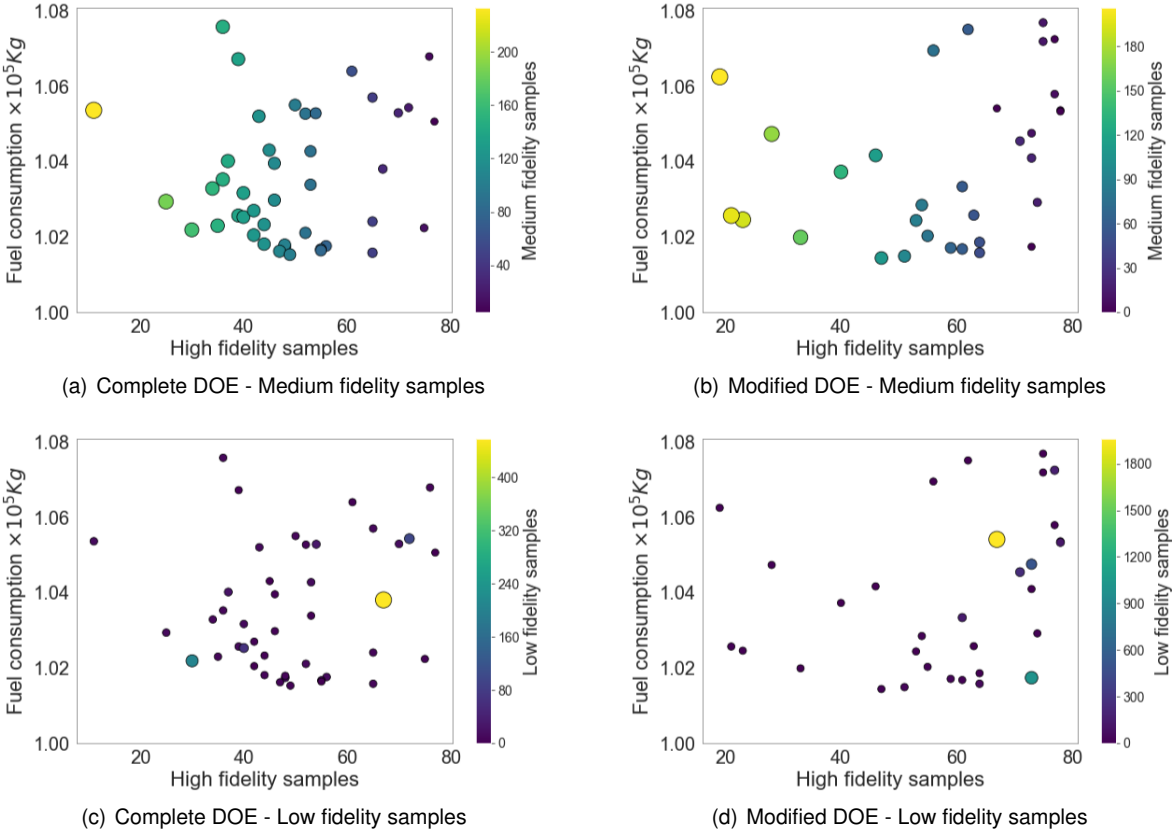
Figure 5.16: Scatter plot of the amount of samples queried per fidelity level in function of the fuel consumption quantities for the 3 fidelity levels optimization using complete and modified DOE.

the same approach (complete or modified DOE) are aligned vertically in figure 5.16, each acceptable solution of each approach is represented twice and aligned vertically. For each solution, we can easily see the number of samples per fidelity level and the respective fuel consumption. The log (RSCV) is not presented here because, once again, no clear relation is observed between this quantity and the samples queried.

Tables 5.15 and 5.16 complement figure 5.16 by providing the percentage of acceptable run tests per number of chosen samples. We divided the solutions in three groups of queried HF samples and we

present the maximum and the minimum number of MF and LF samples for each group.

Table 5.15: Percentage of filtered solutions by the number of queried samples per fidelity level for the 3 fidelity levels optimization using complete DOE.

High fidelity samples	[11, 39]	[40, 59]	[60, 77]
Medium fidelity samples	[135, 233]	[78, 133]	[5, 61]
Low fidelity samples	[2, 206]	[0, 66]	[0, 458]
Percentage of optimizations	23.3%	53.5%	23.3%

Table 5.16: Percentage of filtered solutions by the number of queried samples per fidelity level for the 3 fidelity levels optimization using modified DOE.

High fidelity samples	[19, 39]	[40, 59]	[60, 78]
Medium fidelity samples	[157, 206]	[66, 133]	[0, 60]
Low fidelity samples	[3, 13]	[1, 32]	[0, 1961]
Percentage of optimizations	15.6%	28.1%	56.3%

We immediately see that, unlike the 2 F approaches, the number of HF samples queried during the optimization widely varies. However, as expected, the quantity of chosen HF samples is much lower for the 3 F approaches when compared with the 2 F approaches. From tables 5.15 and 5.16, the number of MF samples queried inversely varies with the number of HF samples queried, for both approaches, but the same can not be said about the LF samples.

The optimizations that fulfilled the requirements of the complete DOE approach tend to choose the number of HF samples between [40, 59], while the ones of the modified DOE tend to choose the number of HF samples between [60, 78]. As previously referred, the modified DOE has less 120 LF and 40 MF samples informing the initial model. When performing the enrichment criterion level (see equation (2.36)) during the optimization of the 3 F modified DOE, the uncertainty reduction with HF samples is more advantageous than querying lower fidelity samples. For a more informed initial model (complete DOE), this reduction of uncertainty with HF samples is less often beneficial. For both approaches, the best fuel consumption values are obtained for the number of HF samples between 45 to 70. However, even between this range of HF samples, the performance of the method is very uncertain and the best solutions never reach the minimum fuel consumption values obtained using 2 F approaches. This leads us to conclude that, even though the Pearson coefficient between the HF and MF samples is close to one for the majority of the DOEs, it is essential to have more HF samples queried during the optimization to obtain better fuel consumption results and more run tests that fulfill the requirements. To query more HF samples using 3 F approach, one can try to reduce the discretization and consequently the cost of the MF samples. This action will probably generate a reduction in the Pearson coefficients between HF and MF samples. However, having more HF samples queried will compensate this reduction.

5.8 Optimization using Two Multi-fidelity Levels for Non-identical Discipline Discretization

In this section, we study the potential of multi-fidelity optimization using the transfer schemes developed in section 4.4 for non-identical discipline discretization. Again, 100 run tests are performed and the same problem, optimization constants, parameters and reference solution are used. Since it was previously concluded that the best approach is the 2 F modified DOE, this approach will be adopted here. Nevertheless, new discretizations and costs associated to the fidelity levels must be defined.

Therefore, in appendix A.5, we perform a mesh convergence study using different discipline discretizations, to observe the behavior of three quantities: fuel consumption, trimmed flight and structural behavior. The main conclusion is that the three quantities can still produce quality results when the AM is refined and the SM is coarser than the spanwise AM. In this case, the analysis computational time naturally decreases, which can lead to faster optimizations without a significant result penalty. This way, we opted to change the HF discretization and keep the LF discretization, as presented in table 5.17

Fidelity level	High	Low
num_y	61	5
num_x	7	3
num_s	31	5

Table 5.17: Discretization of high and low fidelity levels for non-identical discipline discretization.

High fidelity	Low fidelity
1	1/200

Table 5.18: Cost of high and low fidelity levels for non-identical discipline discretization.

The cost associated to each fidelity level is presented in table 5.18. As previously concluded in section 5.7.3, the cost of the LF samples should be reduced to yield better results. Thus, our LF samples adopted the same cost as the LF samples of an aerodynamics multi-fidelity problem presented by Meliani [17]. The Pearson correlation between the samples of the 100 DOE is presented in section A.6.1 in appendix. Next, in figure 5.17 we present the fuel consumption results that fulfilled the requirements. The other two qualities (log (RSCV) and cost) are very similar to the ones of the 2 F modified DOE approach, so we do not present them.
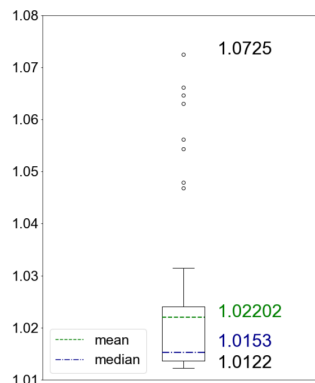

Figure 5.17: Fuel consumption for multi-fidelity optimization using non-identical discipline discretization.

We observe similar results as the ones of two fidelity levels optimization using modified DOE, which

proves that reducing the structural discretization simplifies the analysis model without worsening the optimization results. The number of acceptable run tests was 86, which is a significant increase in relation to the number of acceptable run tests of the 2 F approach in table 5.12. This increase is related with the lower cost associated to the LF samples. In appendix A.6.2, we observe the scatter plot of this results.

5.9   Summary and Overview

To conclude this chapter, we present the mean time to perform ten optimizations of all the studied approaches in table 5.19. The 100 run tests optimizations performed previously were not possible to use in this study, since they were ran in different machines. The computational time is an important aspect to take into account when comparing the fidelity approaches.

Table 5.19: Mean computational time to run 10 optimizations of each fidelity level approach.

Time	1F	Modified DOE		Complete DOE		Non-identical discipline discretization
		2 F	3 F	2 F	3 F	
Hours	0.63	1.70	3.15	1.67	3.27	1.60

Comparing the run time between optimizations with a different number of fidelity levels used, we observe that the higher the number of fidelity levels, the higher the computational time. This is an expected behavior, since adding a fidelity level to the optimization substantially increases the overall complexity of the algorithm, namely the DOE, the surrogate construction, and the enrichment level criterion. Thus, when choosing the number of fidelity levels, we must be aware that adding a fidelity level to the optimization implies a time penalty, so the decision of use an extra fidelity level must provide a significantly improvement of the results, as observed between 1 F and 2 F. Comparing the time of the 2 F approaches with the non-identical discipline discretization approach, we observe a slight reduction, as expected.

Finally, we conclude this chapter by ordering the approaches by performance improvement from the left to the right.
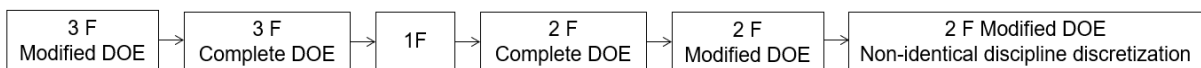
Figure 5.18: Fidelity level performance improvement from the left to the right.

As it can be seen in figure 5.18, the 2 F approaches have a better performance than the 1 F, proving the advantage of performing multi-fidelity optimization over single-fidelity optimization. Although the 3 F approaches have lower performance than the 1 F, we can change the parameters of 3 F in order to obtain a better performance. The approach that uses non-identical discretization is the best performance approach due to the computational time reduction in relation to the 2 F modified DOE. In appendix A.7 the best solutions of 2 F modified DOE and reference solution are shown and discussed.

Chapter 6

Conclusions

6.1 Summary and Achievements

In this work, we extended the transfer schemes of OAS to non-identical discipline discretization. To developed these new schemes, we had to deeply understand the implemented schemes in OAS. During this process, we discovered that the original implemented schemes in OAS were not in compliance with the conservation requirement. We informed the MDO lab that developed OAS and they started to work on a new version (version 2.2.2) that has not yet been released. Thus, we developed our new schemes to account with different discipline discretization based on the implemented transfer schemes that do not fulfilled the conservation requirements. Two different load transfer schemes and one discipline transfer schemes were developed.

We performed a study to explore and compare the results of SEGO using a different number of fidelity levels, varying from one to three, and different DOE. In total, we tested 5 different approaches by running 100 run tests of each, each run test starting with a different dataset to construct the kriging model. We concluded that the best approach is the multi-fidelity optimization using 2 fidelity levels with only nested samples in the initial dataset, which proves the superiority of multi-fidelity optimization over single-fidelity optimization using SEGO.

Then, the optimization approach that provided the best results was used to test the new transfer schemes. We selected new mesh discretizations and associated costs and performed again 100 run tests. We concluded that the results were very similar to the ones of 2 fidelity levels with a reduction of the computational time needed to run the optimization. Thereby, we can deduce that using a different spanwise discretization in the aerodynamic and structural disciplines is a viable solution to reduce the computational time of an optimization process while keeping the quality of the result.

6.2 Future Work

As future work, it would be interesting to update the three developed transfer schemes for non-identical spanwise discretization of aerodynamic and structural disciplines according to the new version

of OAS (version 2.2.2), which is still being developed. After this update, it should be performed a conservation study of the aerodynamic and structural virtual work using load transfer method one and two and the displacement transfer. From this study, it can be concluded which of the proposal methods has a better performance in terms of virtual work conservation. Then, more detailed mesh convergence studies with non-identical disciplines meshes can be done and multi-fidelity SEGO using the new updated OAS ability can be performed.

Another interesting development would be to implement the derivatives on the new transfer schemes and all the functions that were created in OAS in order to enable different discipline discretization. This way, the transfer schemes can be updated in the OAS Github repository and problems using gradient-based optimizer with different discipline discretization can be executed.

Bibliography

[1] A. Marta. Aircraft optimal design - courses notes. IST, 2018, Portugal.

[2] M. Meliani. Smt. `https://www.github.com/m-meliani/smt`. Accessed: 2020-11-02.

[3] J. Martins and A. Lambe. Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51:2049–2075, 09 2013. doi: 10.2514/1.J051895.

[4] A. Dumas, M. Madonia, I. Giuliani, and M. Trancossi. MAAT cruiser/feeder project: criticalities and solution guidelines. Technical report, SAE Technical Paper, 2011.

[5] R. Lam and P. Alain. *Surrogate modeling based on statistical techniques for multi-fidelity optimization*. PhD thesis, Massachusetts Institute of Technology, 2014.

[6] M. Kochenderfer and T. Wheeler. *Algorithms for optimization*. MIT Press, 2019.

[7] C. Williams and C. Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[8] G. Matheron. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963.

[9] A. Forrester and A. Keane. Recent advances in surrogate-based optimization. *Progress in aerospace sciences*, 45(1-3):50–79, 2009.

[10] T. W. Simpson, T. M. Mauery, J. Korte, and F. Mistree. Kriging models for global approximation in simulation-based multidisciplinary design optimization. *AIAA journal*, 39(12):2233–2241, 2001.

[11] D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[12] J. P. Jasa, J. T. Hwang, and J. R. Martins. Open-source coupled aerostructural optimization using python. *Structural and Multidisciplinary Optimization*, 57(4):1815–1827, 2018.

[13] I. Chittick and J. Martins. An asymmetric suboptimization approach to aerostructural optimization. *Optimization and Engineering*, 10(1):133, 2009.

[14] J. Martins. *A coupled-adjoint method for high-fidelity aero-structural optimization*. PhD thesis, Stanford University, October 2002, USA.

[15] G. Kenway and J. Martins. Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration. *Journal of Aircraft*, 51(1):144–160, 2014.

[16] J.-F. Sigrist. *Fluid-structure interaction: an introduction to finite element coupling*. John Wiley & Sons, 2015.

[17] M. Meliani, N. Bartoli, T. Lefebvre, M.-A. Bouhlel, J. Martins, and J. Morlier. Multi-fidelity efficient global optimization: Methodology and application to airfoil shape design. In *AIAA Aviation 2019 Forum*, page 3236, 2019, Toulouse, France.

[18] B. Grossman, Z. Gurdal, J. Strauch, W. Eppard, and R. Haftka. Integrated aerodynamic/structural design of a sailplane wing. *Journal of Aircraft*, 25(9):855–860, 1988.

[19] B. Grossman, R. Haftka, P.-J. Kao, D. M. Polen, M. Rais-Rohani, and J. Sobieszczanski-Sobieski. Integrated aerodynamic-structural design of a transport wing. *Journal of Aircraft*, 27(12):1050–1056, 1990.

[20] I. Kroo, S. Altus, R. Braun, P. Gage, and I. Sobieski. Multidisciplinary optimization methods for aircraft preliminary design. In *5th symposium on multidisciplinary analysis and optimization*, page 4325. Standford University, 1994, Standford, USA.

[21] V. Manning. *Large-scale design of supersonic aircraft via collaborative optimization*. Ph.D. thesis, Standford University, 1999, USA.

[22] R. Priem, H. Gagnon, I. Chittick, S. Dufresne, Y. Diouane, and N. Bartoli. An efficient application of bayesian optimization to an industrial mdo framework for aircraft design. In *AIAA AVIATION 2020 FORUM*, page 3152, 2020.

[23] M. Sasena. *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*. PhD thesis, University of Michigan, 2002, USA.

[24] M. McKay, R. Beckman, and W. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

[25] A. Olsson, G. Sandberg, and O. Dahlblom. On latin hypercube sampling for structural reliability analysis. *Structural safety*, 25(1):47–68, 2003.

[26] R. Jin, W. Chen, and A. Sudjianto. An efficient algorithm for constructing optimal design of computer experiments. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 37009, pages 545–554, 2003, USA.

[27] A. Forrester, A. Sóbester, and A. Keane. *Engineering Design via Surrogate Modelling*, chapter 2, pages 51–54. John Wiley Sons, Ltd, 2008. doi: 10.1002/9780470770801.ch2.

[28] M. Schonlau. *Computer Experiments and Global Optimization*. PhD thesis, University of Waterloo, 1997, USA.

[29] M. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, volume 275, pages 51–67. Springer, 1994.

[30] M. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464.

[31] L. L. Gratiet. *Multi-fidelity Gaussian process regression for computer experiments*. PhD thesis, University Paris-Diderot, 2013, France.

[32] M. A. Bouhlel, N. Bartoli, A. Otsmane, and J. Morlier. Improving kriging surrogates of high-dimensional design models by partial least squares dimension reduction. *Structural and Multidisciplinary Optimization*, 53(5):935–952, 2016.

[33] J. Močkus. *On bayesian methods for seeking the extremum*, pages 400–404. Springer, Berlin, Heidelberg, 1975. doi: 10.1007/978-3-662-38527-2_55.

[34] D. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6):119–139, 1951.

[35] M. Schonlau, W. Welch, and D. Jones. Global versus local search in constrained optimization of computer models. *Lecture Notes-Monograph Series*, pages 11–25, 1998.

[36] N. Bartoli, T. Lefebvre, S. Dubreuil, R. Olivanti, N. Bons, J. Martins, M.-A. Bouhlel, and J. Morlier. An adaptive optimization strategy based on mixture of experts for wing aerodynamic design optimization. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 4433, 2017, Toulouse, France.

[37] N. Bartoli. Kriging - EGO. ONERA, DTIS/M2CI, 2020, Toulouse, France.

[38] A. Watson and R. Barnes. Infill sampling criteria to locate extremes. *Mathematical Geology*, 27(5): 589–608, 1995.

[39] C. Audet, J. Denni, D. Moore, A. Booker, and P. Frank. A surrogate-model-based method for constrained optimization. In *8th symposium on multidisciplinary analysis and optimization*, page 4891, 2000, Seatle, USA.

[40] J. Hernández-Lobato, M. Gelbart, R. Adams, M. Hoffman, and Z. Ghahramani. A general framework for constrained bayesian optimization using information-based search. *The Journal of Machine Learning Research*, 17(1):5549–5601, 2016.

[41] M. Meliani. High dimensional efficient global optimization via multi-fidelity surrogate modeling. Master's thesis, ISAE-SUPAERO, September 2018, Toulouse, France.

[42] M. Bouhlel, J. Hwang, N. Bartoli, R. Lafage, J. Morlier, and J. Martins. A python surrogate modeling framework with derivatives. *Advances in Engineering Software*, page 102662, 2019.

[43] R. Vauclin. Multi-fidelity surrogate models for structural optimization. Master's thesis, ISAE-Supaero, September 2014, Toulouse, France.

[44] R. Vauclin. Multifi cokriging in openmdao. `http://openmdao.org/twodocs/versions/2.0.0/_srcdocs/packages/surrogate_models/multifi_cokriging.html`. Accessed: 2020-11-01.

[45] A. Lambe and J. Martins. Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Structural and Multidisciplinary Optimization*, 46(2):273–284, 2012.

[46] D. Steward. The design structure matrix: A method for managing the design of complex systems. *IEEE Transactions on Engineering Management*, 28(1981):pp, 1981.

[47] T. Browning. Applying the design structure matrix to system decomposition and integration problems: a review and new directions. *IEEE Transactions on Engineering management*, 48(3):292–306, 2001.

[48] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the solution of linear systems: building blocks for iterative methods*. SIAM, 1994.

[49] D. Kraft. A software package for sequential quadratic programming. 1988. DFVLR Obersfaffeuhofen, Germany.

[50] D. Luenberger, Y. Ye, et al. *Linear and nonlinear programming*, volume 2. Springer, 1984.

[51] J. Martins and J. Hwang. Review and unification of methods for computing derivatives of multidisciplinary computational models. *AIAA journal*, 51(11):2582–2599, 2013.

[52] J. Hwang and J. Martins. A computational architecture for coupling heterogeneous numerical models and computing coupled derivatives. *ACM Transactions on Mathematical Software (TOMS)*, 44 (4):1–39, 2018.

[53] J. Anderson. *Fundamentals of aerodynamics*. McGraw-Hill, 2010.

[54] S. Rao. *The finite element method in engineering*. Butterworth-heinemann, 2017.

[55] C. Heath and J. Gray. OpenMDAO: framework for flexible multidisciplinary design, analysis and optimization methods. In *53rd AIAA*, page 1673, 2012, Honolulu, Hawaii. doi: 10.2514/6.2012-1673.

[56] J. Vassberg, M. DeHaan, M. Rivers, and R. Wahls. Retrospective on the common research model for computational fluid dynamics validation studies. *Journal of Aircraft*, 55(4):1325–1337, 2018.

[57] I. Berezin, P. Sarkar, and J. Malecki. Fluid–structure interaction simulation. In *Recent Progress in Flow Control for Practical Flows*, pages 263–281. Springer, 2017.

[58] Mdo lab, University of Michigan, USA. `http://mdolab.engin.umich.edu/`. Accessed: 2020-11-02.

[59] J. Martins and N. Poon. On structural optimization using constraint aggregation. In *VI World Congress on Structural and Multidisciplinary Optimization WCSMO6, Rio de Janeiro, Brasil*. Citeseer, 2005.

[60] P. Doornbos. Boeing 777 specs, what makes this giant twin work? Modern Airlines. Accessed: 2020-09-01.

[61] T. Salih. Nature of the materials for modern airplane parts. Babylon University, Iraq, 12 2010.

[62] R. Wasserstein and N. Lazar. The ASA statement on p-values: context, process, and purpose. *The American Statistician*, 70(2):129–133, 2016. doi: 10.1080/00031305.2016.1154108.

[63] J. Jasa, J. Hwang, and J. Gray. OpenAeroStruct documentation. `http://mdolab.engin.umich.edu/OpenAeroStruct/`. Accessed: 2020-12-01.

Appendix A

Appendix A

A.1   Aero-Structural Design and Optimization Tool: OpenAeroStruct

OAS is a low fidelity aero-structural analysis and optimization tool. In general, to run an optimization problem in OAS we start by defining the lifting surface, then we specify the flow and problem conditions followed by the definition of the design variables, constraints and objective function. The definition of the lifting surface is addressed in section 4.2. As we will see, in the definition of the lifting surface we decide the number of panels and structural elements that discretize the wing. In addition, we also define some parameters, such as the structural properties. The flow and problem conditions are treated as independent variables. As an example of these parameters we have the flight speed, the Mach number and the Reynolds number. The last step before the optimization is the definition of the design variables, constraints and objective function. There is a certain set of parameters that can be used as design variables, such as angle of attack, twist distribution, taper and span. Common constraints include structural failure and fixed lift coefficient. Finally, the objective function has also different alternatives including drag, weight and fuel burn.

There are two types of variables that can be assigned as wing design variables. While sweep, taper and dihedral are scalars associated to the entire surface, twist and chord distribution are arrays that contain values for specific spanwise locations along the surface. The design variables defined as arrays along the span are controlled by control points that influence a b-spline interpolation instead of directly controlling the nodal mesh points.

Figure A.1 shows the semi-span being controlled by the b-spline knots, the green points. The blue curve is interpolated from the green points and it modified the aerodynamic mesh. The number of control points is defined according to the desired interpolation order, independently of the fidelity of the aerodynamic mesh.

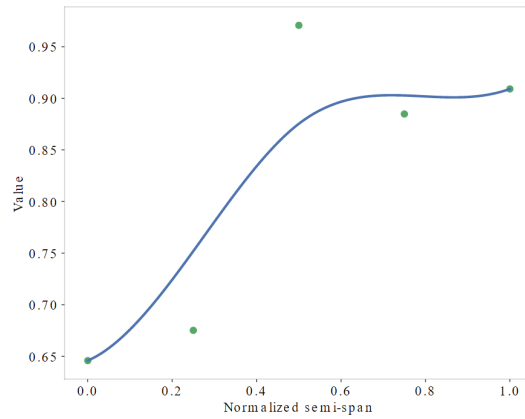More documentation about how OAS work can be found in [12, 63].

Figure A.1: Normalized Semi-span controlled by b-spline knots [63].

A.2 Mesh Convergence for Identical Spanwise Discipline Discretization

In this section, we include the other two quantities used to perform mesh convergence for identical spanwise discipline discretization, fuel burn and trimmed flight, presented in Figures A.2 and A.3. As commented in section 5.3, the structural failure is slower to converge, thereby it is the quantity used to decide the mesh discretization associated to each fidelity level.
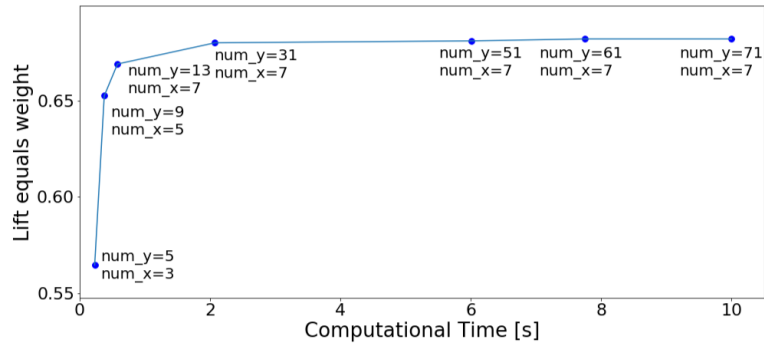

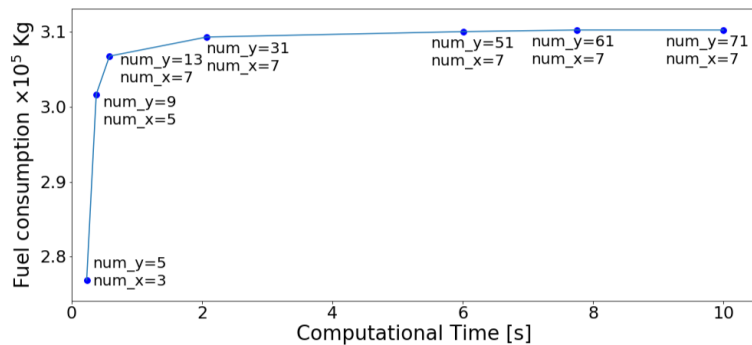Figure A.2: Mesh convergence for trimmed flight.

Figure A.3: Mesh convergence for fuel consumption.

81

A.3   Correlation between Three Fidelity levels Modified DOE

Figure A.4 shows the value of the fuel consumption and the two problem constraints for the 40 nested DOE samples of the first 3 F modified DOE. The LF samples show a high discrepancy from the HF samples with a poor linear relationship. The MF samples show a good linear relationship between the HF samples.
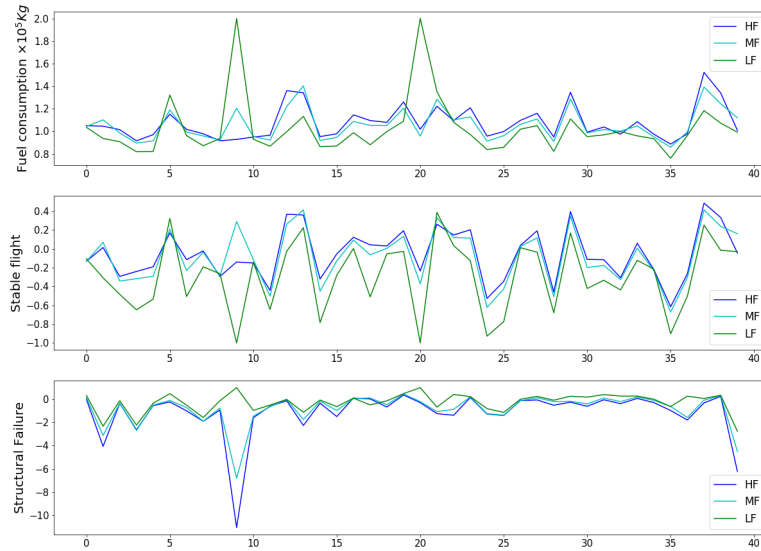
Figure A.4: Nested DOE of the first 3 fidelity level modified DOE.

A.4   Optimization History of the 59th Acceptable Run Test of the Optimization for Two Fidelity Levels using Modified DOE
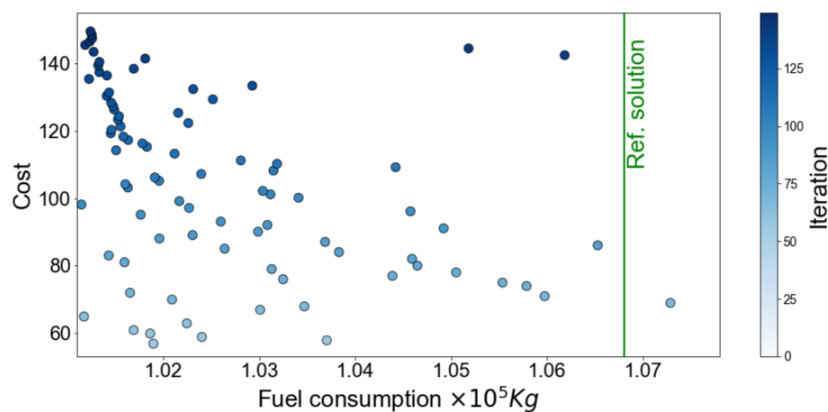
Figure A.5: Scatter plot of the optimization history of the 59th acceptable run test for 2 fidelity levels optimization using modified DOE.

In figure A.5 we show the 59th acceptable run test of the 2 F modified DOE optimization. Each circle of figure A.5 represents an HF sample that fulfills the requirements and, additionally, the circle's color represents the iteration stage. We observe that, at the beginning of the optimization, the objective

function value of the queried samples is very variable, which corresponds to the exploration phase, and as the optimization takes place the objective function of the queried samples is getting more localized, which corresponds to the exploitation phase. As previously commented, for a significant amount of acceptable optimizations (this one included), the number of queried non-nested LF samples is very low, due to the cost associated to the LF level. This means that the exploration phase, where only by LF samples should be queried, also queries HF samples. If the fidelity levels had a more pronounced cost difference, an exploration phase with solely LF samples would be performed. This exploration would also query much more samples than the exploration depicted in figure A.5, due to the low cost of LF samples. As approximately from the cost 120 the optimizer reaches a fuel consumption value very close to the best solution, we conclude that the optimization budget can be reduced without significantly affecting the best solution.

Many run tests that fulfilled the requirements using two fidelity level have an optimization history similar to the one shown in figure A.5. Then, one can reduce the budget of these approaches and still get valuable solutions. However, not all the acceptable run tests have an history evolution similar to figure A.5 and thus one must be aware that reducing the budget will leads to fewer acceptable run tests.

A.5 Mesh Convergence for Non-identical Spanwise Discipline Discretization

In this section, we present a mesh convergence study for identical and non-identical spanwise discipline discretization. The objective is to perform MDA and study the behavior of the objective function and the two constraints for multiple mesh discretizations. Additionally, we also present the computational time of each analysis. Two aerodynamic mesh discretizations are used, the high and medium fidelity meshes of chapter 5. For both meshes, the spanwise structural discretization starts to be the same as the aerodynamic and then it is progressively coarsened. The notation to designate the aerodynamic and structural discretization (AD and SD) used in this document, num_y, num_x and num_s, is replaced here by n_y, n_x and n_s respectively, to make the graphs of this section easier to interpret.

A.5.1 Fuel Consumption

The quantity of interest to minimize is the fuel consumption using the Breguet range equation, as explained in section 5.1. Let us recall the Breguet range equation as

$$W_{\text{fuel}} = (W_s + W_0) \times \exp\Big(\frac{R.C_T}{V}\Big(\frac{CL}{CD}\Big)^{-1} - 1\Big). \tag{A.1}$$

Here, the only variables are the structural weight W_s and the lift-to-drag coefficient ratio C_L/C_D.

We are going to observe the fuel consumption and other quantities for different mesh discretizations using an histogram. Each column of the histogram represents a different combination of AD and SD. The histograms are organized as follows: in the first four columns we use the more refined AD corresponding

to the high fidelity mesh of chapter 5 and SD is progressively coarsened, starting with the spanwise AD equals to the SD. Then, we use the medium fidelity AD of chapter 5 and SD is again progressively coarsened, starting with the spanwise AD equals to the SD. The most accurate values are naturally associated with the most refined meshes for both disciplines, located in the leftmost column of the histograms.

To study the behavior of the fuel consumption for multiple mesh discretizations, let us start by analyzing the structural mass and the drag-to-lift coefficient ratio, presented in figure A.6 (a) and (b), respectively.

(a) Structural mass.   (b) Lift-to-drag coefficient ratio.
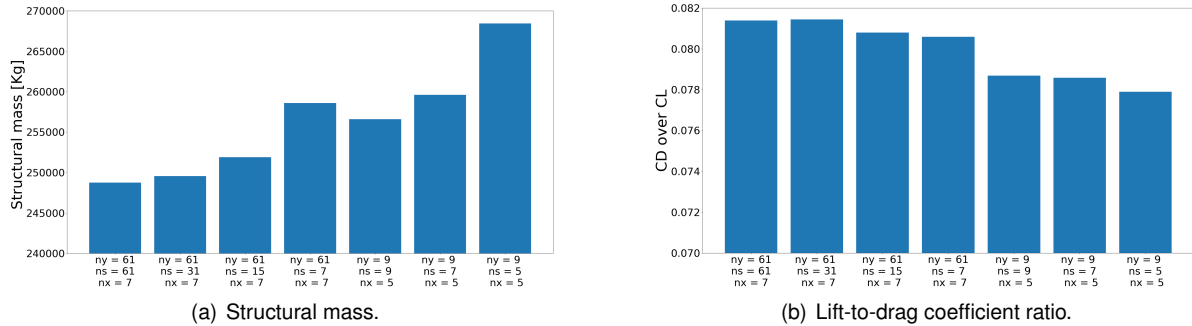
Figure A.6: Structural mass and lift-to-drag coefficient ratio for multiple mesh discretizations.

As it can be observed, when the AD is kept constant and the SD is progressively coarsened, the structural mass tends to increase and deviates from the most accurate model, the leftmost column. The structural mass, depicted in figure A.6 (a), is influenced by the structural element radius and the structural element thickness. To explain this behavior, let us explore this quantities performance. The element radius are computed using the structural mesh (SM) and the thickness over chord ratio, a quantity that is related with the wing profile. The SM is similar to the aerodynamic mesh (AM), but it has the same spanwise discretization has the structural discipline. Naturally, when the same spanwise discipline discretization is used, the SM is coincident with the AM, however for different disciplines discretizations, the SM is obtained from linear interpolations of the AM. Figure A.7 illustrates the AM and SM, in black and green color, respectively, when the spanwise AD is 9 and the spanwise SD is 5. It also illustrates the structural elements in different colors. Elements with the same radius and thickness are represented in the same color.
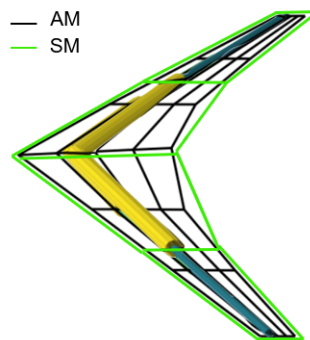

Figure A.7: Aerodynamic and structural meshes for non-identical spanwise discretization.

The element radius is the thickness of the mean chord of the two chords that surround the structural element. These two chords are obtained from the SM. When the SD is coarse, as the example illustrated in figure A.7, less elements discretize the wing beam. Consequently, the length of the chords that surround these elements are very different. For one structural element, the chord associated with the node closer to the wing tip is much smaller than the chord associated with the node closer to the wing interior. Thereby, the mean chord will generate a radius for the element that is too thick for the element tip closer to the wing tip and too thin for the element tip closer to the wing interior. This corresponds to an approximation of the structural mass by excess and default, respectively. The structural mass dramatically increases when the SD is coarse, as it can be observed in figure A.6 (a) in the three rightmost columns. This indicates that the mass 'increase' is higher than the mass 'loss'. For situations when the SD is less coarse, this phenomenon is less accentuated, as depicted in figure A.6 (a) for the most refined AD.

Figure A.6 (a) shows us an interesting behavior when the SD is constant and two AD are used, as depicted by columns four and six. If the AM is more refined than the SM (column four), the AM is more informed than the structural discipline, so the linear interpolations obtain a SM similar to the situation where the same discipline discretization is used. The same structural mass is obtained. Thereby, for the structural mass quantity, it is counter-productive to use smaller SD than the AD. However, for other quantities this is not the case, as we will see.

Let us now focus on the thickness parameter behavior. Each structural element has an associated thickness. This quantity is computed using the b-spline curve defined from the control points. We defined the control points so that the thickness increases as we get closer to the wing root. For a refined SD, the element thickness is more accurate, it is progressively getting thicker as we approach the wing root. As the SD is getting coarse for the two AD, this progressive thickening is not so accentuated. Again, the thickness will be too thick for the wing tip and too thin for the wing root. So, when the SD is coarsen, there is a mass excess in the wing tip and a mass default in the wing root. From figure A.6 (a), we may say again that the mass 'loss' is less significant than the mass 'increase'. Once again, for the same SD and different AD, the thickness of the elements is similar.

The drag-to-lift coefficient ratio, depicted in figure A.6 (b), shows that this quantity has approximately a constant value for the same AD. However, we see some fluctuations, specially for the more refined AD. To understand these variations, we must understand the lift and drag coefficients. The lift coefficient is computed from the lift that is computed from the panels forces. The panels forces mainly depend on the wing deformed mesh. The drag coefficient has two components, the induced drag and the viscous drag. Both drags are also mainly influenced by the wing deformed mesh geometry. So, the deformed mesh geometry is the quantity that dominates the behavior of figure A.6 (b). The deformed mesh is computed in the displacement transfer function. When the AD is not coincident with the SD, the displacement transfer function does a linear interpolation to transfer the displacements and rotations of the nodes and thus obtain the deformed mesh. This process was described in detail in section 4.4.2. This linear interpolation may sometimes provide worse deformed meshes, that influence the lift and drag coefficients. Naturally, this effects is more accentuated when the AD is much more refined than the SD.

This quantity show us that using an AD higher than the SD is worth it, in contrast to the structural mass.
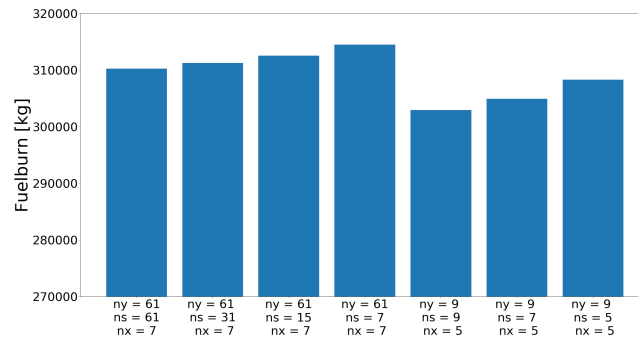

Figure A.8: Fuel consumption for multiple mesh discretizations.

Finally, we observe the fuel consumption histogram, depicted in figure A.8. We conclude that this quantity has a behavior similar to the structural mass, presented in figure A.6 (a). Decreasing the SD relatively to the spanwise AD may be a good way to reduce the computational time without significantly worsening the results or even improving the results, as depicted in the last two columns of figure A.8.

A.5.2 Trimmed Flight

The trimmed flight is one of the constraints of our optimization problem that translates in lift equals weight during the flight. The formula used to verify this requirement is given as

$$\frac{L}{W} = 1 - 0.5\rho V^2 S \frac{C_L}{(W_s + W_{\text{fuel}} + W_0) \times g},$$ (A.2)

where the variables are S, the wing surface, C_L, the lift coefficient and W_s and W_{fuel}, the structural and fuel weight. The structural and fuel weight for multiple mesh discretizations are represented in the last subsection A.5.1 in figures A.6 (a) and A.8. Thus, we present in figure A.9 similar histograms for the wing area and lift coefficient.
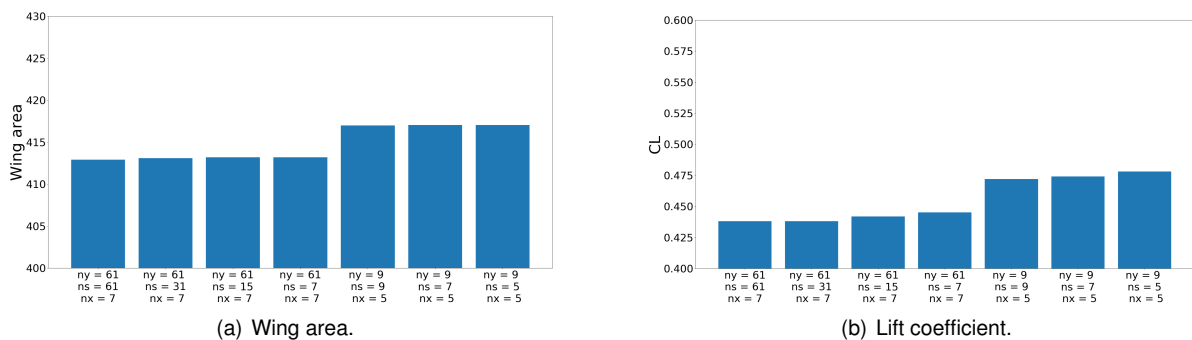

(a) Wing area.

(b) Lift coefficient.

Figure A.9: Wing area and lift coefficient for multiple mesh discretizations.

The wing area is constant for the same AD, regardless of the SD. The coarser the AD, the larger the wing area. This is an intuitive result if we observe figure 5.2 that shows three different AM discretizations. This figure shows that due to the reduced number of panels, a coarser mesh models the wing more

roughly, specially at the leading edge, with an excess of wing area. When more panels are used, the wing shape is captured more effectively, and therefore the wing area is smaller.

Investigating the lift coefficient histogram, in figure A.9 (b), we observe that for the same AD the C_L is constant with some fluctuations that we explored in the previous section A.5.1.
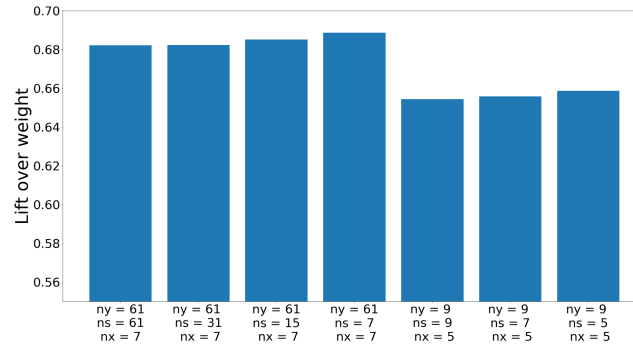

Figure A.10: Lift over weight for multiple mesh discretizations.

We are now ready to analyze the lift over weight histogram, depicted in figure A.10. We see that this quantity behaves similarly to the fuel consumption and the structural mass, which is expected, since they are variables of equation (A.2). That way, the same conclusions of the fuel consumption analysis are applied to the lift over weight. Decreasing the SD relatively to the spanwise AD can be used to reduce the computational time without significantly worsening the results or even improving them, as also explained in the previous section. The second and third columns from the left of figure A.10 show discretizations where the decreasing of SD deteriorates the results. The sixth column shows a discretization where decreasing the SD improves the results.

A.5.3   Structural Failure

The structural failure is the second constraint of our problem. As referred in section 5.1, to simplify the structural failure we use the Kreisselmeier-Steinhauser aggregation method. Consequently, a single scalar function defines the complete structural integrity, that if positive, implies that at least one element has failed. This constraint is related with the structural displacement, the element radius and the element thickness. Figure A.11 show the tip displacement in the vertical (z) direction (see figure 4.3) and the failure for the same discipline discretizations. We decided to depicted the tip displacement because the structural element of the wing tip is the one that experiences the largest displacement.

Starting by analyzing figure A.11 (a), we observe that when the AD is constant and the SD decreases, the tip displacement increases and deviates from the most accurate value (leftmost column). As the SD is getting coarser, there are less structural nodes to distribute the panels loads, consequently the nodes are overloaded and the tip displacement in the z direction is accentuated. Let us also analyze the case when the SD is constant and two AD are used (forth and sixth column). We observe that for this case the tip displacement is higher and less accurate when the AD is more coarse. For a coarser AD, the panels forces are less accurate, which influences the load transfer and, consequently, influences the displacements.

(a) Tip displacement.
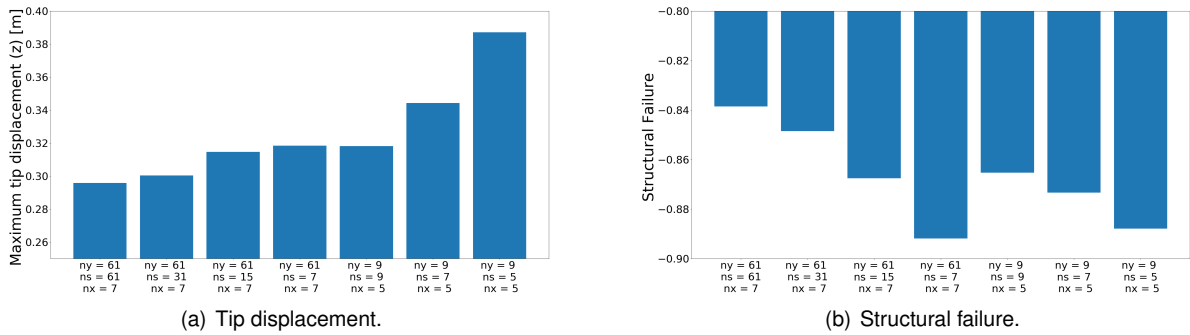

(b) Structural failure.

Figure A.11: Tip displacement and structural failure for multiple mesh discretizations.

We can now analyze the structural failure depicted in figure A.11 (b). We observe that as the SD is coarsened, the structure is less close to failure. From figure A.11 (a), we just concluded that, for the same condition, the tip displacement in the z direction is more accentuated. Thus, it would be expected that the failure quantity would be higher as the SD is coarsened. Then, the radius and the thickness of the elements are the parameters that influence the failure behavior. As explained in section A.5.1, the element radius is computed from the mean chord of the element of the SM and the thickness-over-chord ratio. The mean chord associated to the structural elements are larger than when SD is more refined. Consequently, the element radius are also larger than for a more refined the SD, influencing the failure.

Analyzing the behavior of the structural failure when the SD is kept constant and the AD is varied, we observed that, as the AD is refined, the failure decreases. This is solely related with the displacements behavior, since the element thickness and radius are constant. From figure A.11 (a), we just concluded that, for the same condition, the tip displacement in the z direction decreases. Assuming that this is the most probable element to fail, naturally the failure is smaller.

A.5.4 Computational Time

Finally, we observe the time needed to compute each of the analysis for the same discretizations studied in the last sections. Figure A.12 shows that the computational time is directly related with the disciplines discretizations, as expected. As the discretization increases, whether aerodynamic or structural, the analysis time increases.
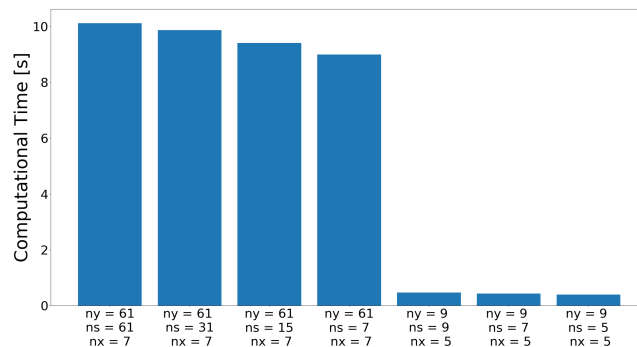

Figure A.12: Computational time for multiple mesh discretizations.

88

A.5.5 Conclusion

From the mesh convergence study and the corresponding time needed to run the analyses, we can conclude that for the refined aerodynamic discretization, decreasing the structural discretization for half of the spanwise aerodynamic discretization still provides good results for the three quantities. The computational time needed to run this analysis is slightly reduced but, in a complete optimization, it might lead to a time reduction of several minutes. For less refined aerodynamic meshes, the quality of the results of the three quantities decreases more rapidly as the structural discretization is decreased, which demonstrates the importance of a high aerodynamic discretization. Naturally, the time needed to complete these analyses also reduces as the structural discretization reduces.

A.6 Optimization using Different Multi-fidelity Levels for Non-identical Discipline Discretization

A.6.1 Correlation between Fidelity Levels

In this section, it is presented the Pearson correlation plots for the DOE samples of the non-identical discretize problem. The results are presented in box plots similar to the ones depicted throughout this dissertation.

Figure A.13 shows the Pearson correlation box plots for the SD more refined than AD problem. Comparing with the Pearson correlation between the HF and LF samples presented in figure 5.4 in section 5.5, we observe that the batches of figure A.13 have a similar behavior, with the coefficients of the fuel consumption and structural failure slightly smaller and the coefficients for trimmed flight slightly higher.

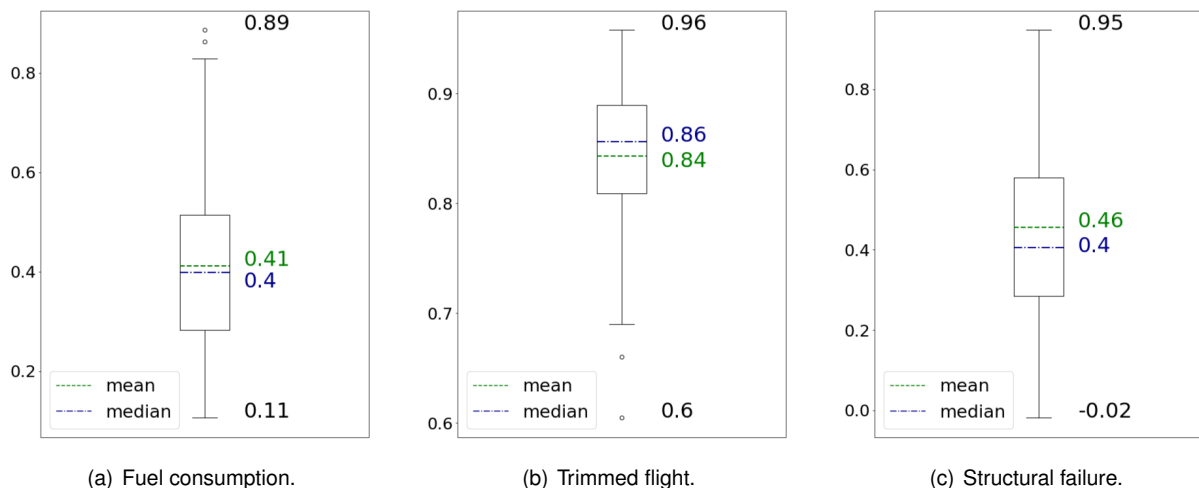(a) Fuel consumption.   (b) Trimmed flight.   (c) Structural failure.

Figure A.13: Pearson correlation between high fidelity and low fidelity DOE samples of the non-identical discretize problem for three different quantities.

Once again, the p-values of all the analyses above are mainly smaller than three percent, except for a few outliers, so the null hypothesis can be rejected.

A.6.2 Scatter Plot

Figure A.14 shows the scatter plot for the acceptable run tests of the two fidelity levels optimization using modified DOE for non-identical discipline discretization.
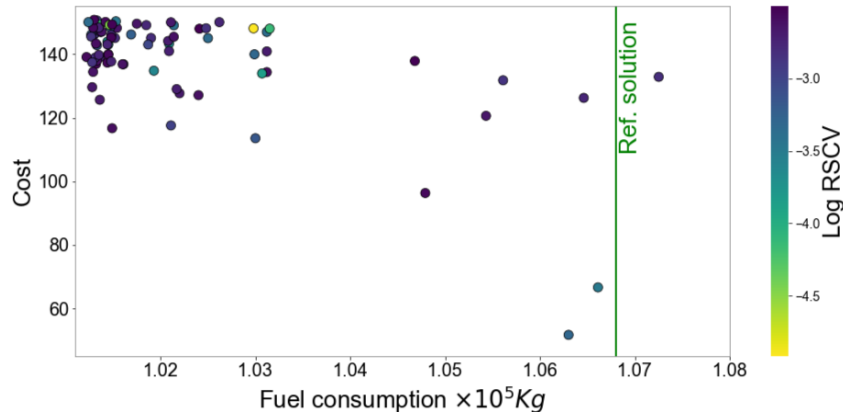

Figure A.14: Scatter plot for the best solutions of 2 fidelity levels optimization using modified DOE for non-identical discipline discretization.

As expected, this scatter plot is very similar to the ones of two fidelity levels presented in section 5.7.3 in figures 5.8 and 5.10. Most of the acceptable solutions have low fuel consumption values and are associated with a high cost, that is, the optimizer is progressively getting better results throughout the optimization process.

A.7 Solutions of the gradient-based and gradient-free approaches

A.7.1 Reference Solution

In this section, we present the design variables, constraints and objective function values of the reference solution presented in section 5.7.1.

Table A.1: Reference solution.

Parameter	Final value
Angle of attack	$8.5°$
Wing twist	[3.0112, 2.2642, 1.0056, -1.5633, 2.5506] $°$
Spar thickness	[0.025, 0.026, 0.0323, 0.0316, 0.0352, 0.025] m
Stable flight	1.08×10^{-6}
Structural Failure	5.77×10^{-6}
Fuel consumption	1.06801×10^5 Kg

Figure A.15 shows the lifting surface geometry of the reference solution along with the twist, lift, thickness and von mises failure spanwise distribution. The elliptical lift distribution is shown in the lift plot in green dashed line. This lift distribution results in the least induced drag for a given wing span and it would be obtained if only the aerodynamic discipline was considered. We observe in figure A.15 that the lift distribution of the reference solution is not the elliptical one. When considering aerodynamic and structural disciplines, the elliptical distribution of circulation is not obtained, because loading more the wing near the fuselage and alleviating on the tip promotes the reduction of the bending moments, allowing a more lighter wing structure and a reduction on the load per span. From the von mises plot in figure A.15, we observe that the wing failure is closer to be reached near the fuselage.
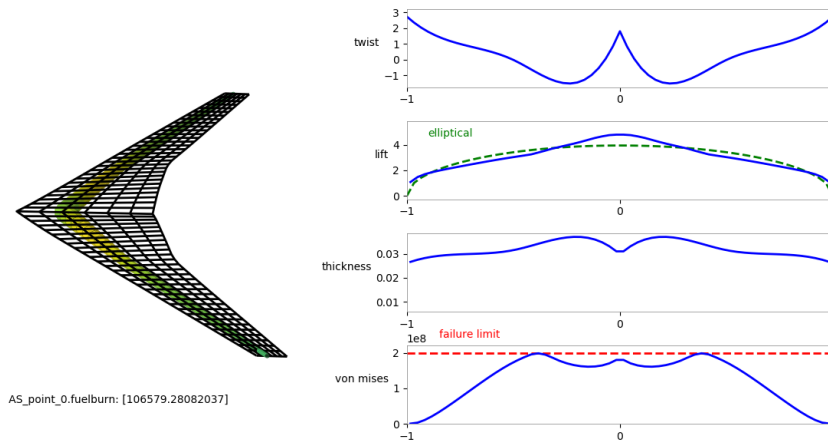

Figure A.15: Reference Solution lifting surface and twist, lift, thickness and von mises failure spanwise distribution.

A.7.2 Solution for the Optimization with Two Fidelity Levels using Modified DOE

Next, we present the best solution for the optimization with two fidelity levels using the modified DOE in section 5.7.3. Among the acceptable solutions of this approach, we consider that the best solution is the one in the Pareto front of figure 5.10 that has the lowest fuel consumption and consequently the highest cost. The best solution has an associated cost of 146.24 and a log (RSCV) value of -3.0172. The results are presented next as

Table A.2: Best solution of the acceptable solutions of two fidelity levels optimization using modified DOE.

Parameter	Final value
Angle of attack	$10.1°$
Wing twist	[-3.1678, -1.0029, -1.1013, -3.0864, -2.8588] $°$
Spar thickness	[0.0015, 0.0015, 0.0323, 0.0316, 0.0252, 0.05] m
Stable flight	$-9,6121 \times 10^{-4}$
Structural Failure	-8.9059×10^{-5}
Fuel consumption	1.0123×10^{5} Kg

A.7.3 Solution for the Optimization with Two Fidelity Level using Modified DOE and Non-identical Discipline Discretization

Again, we present the same type of solution for the optimization using non-identical discipline discretization, presented in section 5.8. The best solution has an associated cost of 139.0941 and the log (RSCV) of -2.5681.

Table A.3: Best solution of the acceptable solutions of two fidelity levels optimization using modified DOE and non-identical discipline discretization.

Parameter	Final value
Angle of attack	$11.7°$
Wing twist	[-3.0025, -2.8831, -1.3464, -3.7778, -3.9574] $°$
Spar thickness	[0.0015, 0.0015 , 0.0365, 0.0272, 0.0254, 0.05] m
Stable flight	-1.9154×10^{-4}
Structural Failure	2.7139×10^{-3}
Fuel consumption	1.0124×10^5 Kg

Comparing this solution with the one of table A.2, we observe that they are very similar. This proves that employing structural discretization lower than the spanwise aerodynamic discretization is a good alternative to reduce the computational weight.

Comparing the solutions of tables A.2 and A.3, two gradient-free approaches, with the one of table A.1, the reference solution that uses a gradient-based optimizer, we see that the reference solution has a higher fuel consumption value and lower constraints violation. The result of the reference solution is influenced by the applied tolerances (see table 5.4) and the optimization starting point. If the tolerances were relived, the constraints would be more violated and that would affect the final design variables, for instance the spar thickness would be more thin in the tip. This would promote a lower fuel consumption result. The multi-fidelity optimizations are not restricted by any starting point, although the initial dataset greatly influences the optimizer performance. The optimizations are mainly impacted by the budget. From the results obtained, we can conclude that in order to obtain results with less violation of constraints the budget would have to be increased.