DEPARTAMENTO DE MOTOPROPULSIÓN Y TERMOFLUIDO DINÁMICA
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS AERONÁUTICOS

AN OPTIMIZATION METHOD FOR CONCEPTUAL DESIGN IN ENGINEERING SYSTEMS BASED ON HIGH ORDER SINGULAR VALUE DECOMPOSITION

DOCTORAL THESIS

By:

Sergio de Lucas Bodas
Aeronautical Engineer

Mentored by:

José Manuel Vega de Prada  Ángel Velázquez López
Professor of Applied Mathematics  Professor of Aerospace Engineering

2012
Acknowledgements

The work contained in this thesis has been partially supported by the Spanish Ministry of Education (under grants: DPI2009-07591 and TRA2007-65699). I would like to thank Ángel Velázquez López and José Manuel Vega de Prada for their help, support, and dedication in mentoring this doctoral thesis. I am also indebted to Diego Alonso Fernandez for his continual help, and for developing and lending me certain methods such as the local optimization methodology. Finally, I would like to give thanks to my friends and family for their constant encouragement. In particular, I am deeply indebted to my parents, and to my wife, Nathalie, for their constant support and for always believing in me. Without them, this thesis would not have been possible.
Abstract

An optimization method for conceptual design in Aeronautics is presented that is based on the use of surrogate models. The various ingredients in the target function are calculated for each individual using surrogates of the associated technical disciplines that are constructed via high order singular value decomposition and one dimensional interpolation. These surrogates result from a limited number of CFD calculated snapshots. The surrogates are combined with an optimization method, which can be either a global optimization method such as a genetic algorithm or a local optimization method, such as a gradient-like method. The resulting method is both flexible and much more computationally efficient than the conventional method based on direct calculation of the target function, especially if a large number of free design parameters and/or tunable modeling parameters are present. The method is illustrated considering a simplified version of the conceptual design of an aircraft empennage.
Resumen

Se presenta un nuevo método de diseño conceptual en Ingeniería Aeronáutica basado el uso de modelos reducidos, también llamados modelos sustitutos (‘surrogates’). Los ingredientes de la función objetivo se calculan para cada individuo mediante la utilización de modelos sustitutos asociados a las distintas disciplinas técnicas que se construyen mediante definiciones de descomposición en valores singulares de alto orden (HOSVD) e interpolaciones unidimensionales. Estos modelos sustitutos se obtienen a partir de un número limitado de cálculos CFD. Los modelos sustitutos pueden combinarse, bien con un método de optimización global de tipo algoritmo genético, o con un método local de tipo gradiente. El método resultante es flexible a la par que mucho más eficiente, computacionalmente hablando, que los modelos convencionales basados en el cálculo directo de la función objetivo, especialmente si aparecen un gran número de parámetros de diseño y/o de modelado. El método se ilustra considerando una versión simplificada del diseño conceptual de un avión.
# Contents

Acknowledgements iii

Abstract v

Resumen vii

Contents ix

1 Introduction 1

2 Mathematical Methods 7

2.1 The HOSVD+I method 9

2.2 High order singular value decomposition 9

2.3 Multi-dimensional interpolation 12

2.4 Global optimization: genetic algorithms 14

2.5 Local optimization: gradient-like methods 16

2.5.1 Unconstrained optimization 16

2.5.2 Trust regions 21

2.5.3 Optimization with constraints 23

2.5.3.1 Penalty methods 24

2.5.3.2 Lagrange multipliers 25

2.5.3.3 Optimization within the constraints border 26

2.5.3.4 The Karush-Kuhn-Tucker conditions 27

2.5.3.5 The combined algorithm for constraint optimization 29

2.6 Genetic algorithm plus gradient-like optimization 32

2.7 Concluding remarks 33

3 Reduced models and computational fluid dynamics in engineering design 35

3.1 The basic optimization tool using reduced models 37

3.2 Modular optimization tool using reduced models 38

3.3 Computational fluid dynamics methods 40

3.3.1 Vortex lattice method 40

3.3.2 The panel method 41

3.3.2.1 Panels calibration 41

3.3.3 Aerodynamic model: Panel method plus boundary layer formulation 42

3.3.4 Boundary layer formulation 43

3.3.4.1 Drag validation 45

3.4 Drag models description 46
## CONTENTS

3.4.1 Geometry parametrization ........................................... 49
3.4.2 Test cases ................................................................. 52
3.4.3 Test points ................................................................. 53
3.4.4 Errors definition ........................................................ 53
3.4.5 Direct models ............................................................. 55
  3.4.5.1 Wetted area model .................................................. 55
  3.4.5.2 Drag coefficient integration ....................................... 56
3.4.6 Corrected models ......................................................... 57
  3.4.6.1 Database computation .............................................. 58
  3.4.6.2 Corrected wetted area model ..................................... 59
  3.4.6.3 Corrected drag coefficient integration model ................. 60
3.4.7 HOSVD drag description ............................................... 61
  3.4.7.1 Database generation ............................................... 62
  3.4.7.2 Database reduction ............................................... 63
  3.4.7.3 Multidimensional interpolation .................................. 65
3.5 Results and discussion ................................................... 67
3.6 Concluding remarks ....................................................... 69

4 Case Study: conceptual design of a commercial aircraft empennage 71
  4.1 The objective function ................................................. 72
  4.2 Geometry ................................................................. 72
    4.2.1 Design parameters ............................................... 72
    4.2.2 Parametrization of the geometry ................................ 73
      4.2.2.1 HTP generation .............................................. 73
      4.2.2.2 VTP generation .............................................. 75
      4.2.2.3 Some possible geometries .................................. 75
  4.3 Objective function ingredients ....................................... 75
    4.3.1 Complexity ......................................................... 75
    4.3.2 Drag ................................................................. 77
    4.3.3 Weight .............................................................. 78
      4.3.3.1 Aerodynamic loads .......................................... 78
      4.3.3.2 Structure design ............................................. 79
      4.3.3.3 Weight computation ......................................... 80
  4.4 Aerodynamics restrictions ............................................ 80
    4.4.1 Stability and control derivatives ............................... 81
    4.4.2 Stall restrictions ................................................. 82
  4.5 Reference configuration ............................................... 82
5 Global Optimization 85
  5.1 Genetic algorithm parameters 86
  5.2 Results and discussion 87
    5.2.1 First optimization campaign 88
    5.2.2 Second optimization campaign 89
    5.2.3 Reduced optimization tool vs a conventional optimization tool 91
    5.2.4 Further improvements of the method 93
  5.3 Preliminary conclusions 95

6 Local optimization 99
  6.1 Case study application 99
    6.1.1 Objective function 100
    6.1.2 Geometry 100
    6.1.3 Restrictions 101
      6.1.3.1 Geometric restrictions 101
      6.1.3.2 Aerodynamic restrictions 101
    6.1.4 Constraints particularity 102
    6.1.5 Method stability tests 104
  6.2 Global and local optimization tools combination 107
    6.2.1 Local optimization improving global solutions HOM1 109
    6.2.2 Combined global/local optimization tool HOM2 109
  6.3 Results and discussion 110
  6.4 The influence of using a higher fidelity model for viscous drag 111
  6.5 Conclusions 116

7 Conclusions 117

Bibliography 121
Introduction

Nowadays, aircraft conceptual design in industry is evolving along two seemingly contradictory lines. On one hand, the number of design parameters and tunable modeling parameters keeps growing and, on the other, commercial pressure tends to emphasize the need to shorten the time span allocated to the design cycles. The problem is further complicated by the fact that the free/tunable parameters are of a highly multidisciplinary nature. This involves the need to use different sets of Physics/Mathematics modeling equations, and also the need to select the fidelity level of both the models themselves and the interactions between them. In this context, it can be said that conceptual design is becoming more and more "mathematized", not only in the description of the different technical modules but, perhaps more acutely, in the optimization method itself.

A rather comprehensive review on the subject of integrated systems engineering in aircraft design has been published in 2006 by Price et al [46], who describe at some length the basic aspects of aircraft integrated design, including those (such as cost and life cycle) that are not easily amenable to formalization. Concerning the state of the art, the authors conclude that it is currently feasible to integrate aerodynamics, structures, manufacturing, and CAD for reduced cost and reduced lead time. However, they argue that further integration of the life cycle with regard to performance, cost, environmental impact, and safety is still in its infant stage. It is also worth mentioning their opinion that the main challenge still remains to embody the core principles of integrated systems engineering into formalized models and tools.

Multidisciplinary design optimization (one of the practical realizations of integrated systems engineering) is a technical area that has been, and still is, the subject of a large R&D effort. This effort has been disseminated via hundreds of research articles that would be impossible to reference here. Therefore, we will only quote those articles that are somewhat closer to the problem that we deal with. Multi-physics optimization of aircraft design aiming to minimize the environmental impact has been
addressed by Antoine and Kroo [4, 5]. They considered an optimization platform that linked modules for the different technical disciplines of engine performance, noise, and engine emissions. In reference [4], the optimizer was based on a Nelder-Mead algorithm [44], while in reference [5] the authors used a genetic algorithm (GA). Other studies on the practical applications of multidisciplinary optimization have been reported, for example, by Piperni et al [45] in the field of business jets, Schumacher et al [53] in the structural design of a regional aircraft wing box, Bower and Kroo [9] in the minimization of cost and emissions over specific route networks, Rao et al [48] in the conceptual design of a two-seater propeller driven aircraft, Choi et al [16] in the design optimization of a supersonic jet, and Werner-Westphal et al [62] in connection with aircraft generated noise evaluation in the preliminary design.

Regarding the optimization environment rather than the application, one of the latest trends in aircraft conceptual design optimization methods is the use of surrogate models. The basic methodology (see Forrester and Keane [25]) can be roughly summarized in a series of five steps: a) selection of the variables to be optimized, b) analysis of some initial sample designs, c) generation of a model of the actual problem using a surrogate approach, d) search in the design space for new design points of interest, e) addition of new designs to those already available, and back (via iteration) to step (c). In the already referred review article by Forrester and Keane [25], the authors discussed various alternatives and explicitly stated that "since no method is truly universal, we give guidance as to each method’s strengths and weaknesses". In particular, they described and discussed cross-validation, polynomials, moving least-squares, radial basis functions, kriging, support vector regression, and enhanced modeling with additional design information. Another review on the same field was published a few years back by Queipo et al [47]. In this article, apart from describing and discussing various techniques, the authors presented a practical case study consisting of the multi-objective design of a liquid-rocket injector. A different approach is the so called Collaborative Optimization (CO) proposed initially by Kroo et al [35]. The idea consists of decomposing the design problem into two levels: the system level whose responsibility is the coordination of the optimization process, and the lower level made up of sub-spaces for the various technical disciplines. The method has evolved along the years and has led to several improved versions such as those reported by Roth and Kroo [51] and Legresley and Alonso [38]. In this last version, the authors considered a high-fidelity aero-elastic design problem, in which the iterative process leading to the optimized solution is accelerated by means of improving the process of information exchange between
the aerodynamics and structures computations. In particular, POD modal description of the variables was used at the interfaces of the disciplines instead of the actual values of variables. For the sake of clarity, it is to be said that the algorithm being reported in the present thesis does not belong to the family of Collaborative Optimization approaches. However, future coupling of CO and HOSVD may prove to be a suitable method to further increase the computational efficiency of the optimization process.

Finally, regarding the mathematical structure of the optimization algorithm, much effort has been devoted to the development and use of gradient-like methods in aircraft conceptual design related activities. Comprehensive reviews of these techniques can be found in, e.g., Chacksfield [14] and Jameson [32]. In parallel, it was soon recognized that other types of approaches were also well suited to the multidisciplinary nature of aerospace design. Back in 2002, Hajela [30] published a review on artificial neural networks, fuzzy logic based function approximations, and evolutionary search and genetic algorithms for aerospace design applications. A recent similar review presenting later developments and a wider spectrum of approaches has been published by Lian et al [39].

In this thesis, we present an optimization method for conceptual design in aeronautics applications that is based on the use of high order singular value decomposition (HOSVD). In particular, we have a higher system level for global optimization, based on either a genetic algorithm (GA) or a gradient-like method, and a lower level with sub-spaces for the technical disciplines based on HOSVD. Thus, two optimization tools, global and local, will be developed that may complement each other in actual engineering design applications. The main difference with other methods is that prior to the start of the optimization process, we fill in some multi-parametric databases with a limited number of computations (snapshots) from the technical disciplines. Then, we use these databases to generate surrogates of the technical disciplines that consist of HOSVD modal representation of the databases plus a quite efficient interpolation. Such surrogates allow for the fast calculation of the fitness of the individuals that come up during the GA evolution process. In other words, we perform the optimization not on the actual parametric space but on the HOSVD representation of this space.

Conceptual design has a large impact in the aeronautics industry. The reason is that even though, in theory, it should only aim to address the highest level questions about the proposed aircraft, in practice it tends to influence strongly decisions being made downstream during the design process. The whole structure of this design process has been described in detail by Price et al [46]. In particular, they consider four phases: (1) con-
ceptual design, (2) preliminary design, (3) detail design, and (4) general engineering design. The reason for this large impact of the conceptual design phase is that, usually, the whole process is tightly constrained by time and cost, meaning that the penalty associated with going backwards to re-bate an earlier decision might not be assumed by the project management. In principle, the outcome of a conceptual phase, quoting Price [46] literally, should be “a configuration with the basic size and arrangement of its main aspects such as the wing, empennage, engines, fuselages, control surfaces, etc”.

In this context, conceptual design should aim to explore a very broad scope of many different configurations within a limited time constrain. The need to explore simultaneously distant regions of the parameter space has been pointed out by Antoine and Kroo [5] in their development of a framework for aircraft conceptual design. Also, in the introduction chapter of their article, Berard and Isikveren [8] state that: “The conceptual design phase is typified by the sizing of a multitude of design candidates generated by the need to construct a voluminous array of technical feasibility or trade studies under the limits of compressed timelines. This means that simple prediction tools must demonstrate characteristics of conformity, robustness, consistency, and applicability for the design problem at hand, with emphasis being placed upon an appropriate functional sensitivity to the free design variables”. These statements also suggest the important fact that, in principle, fidelity of the tools used in this conceptual design phase should be low. For example, Price et al [46] recommend the use of a few initial equations and empirical data, and Berard and Isikveren [8] propose the use of semi-empirical constructs. However, the main problem in dealing with unconventional configurations is that these empirical data/semi-empirical formulae for aerodynamic loads may not exist. This fact has been highlighted by Werner-Westphal et al [62] when they write: "For unconventional aircraft configurations, empirical models are usually not available. The combination of higher fidelity models (more physics) and multidisciplinary design tools (more complexity) becomes necessary".

Summarizing, a tool for the conceptual design of unconventional configurations should:

1. Be able to explore a parameter space containing a very large number of potential candidates that, most likely, will differ widely from one another.

2. Be robust, consistent, and malleable. In other words, it should be such that different sub-modules can be integrated and inter-exchanged.

3. Be able to work with prediction tools that go one step beyond the
standard semi-empirical constructs, which might be unavailable. And this should be done without steeping into the preliminary design phase, where higher fidelity models should be used.

Anticipating the main results in the thesis, the exploration of the whole parameter space is well served using a global optimization tool that uses a genetic algorithm based search approach. The local optimization tool helps to improve solutions at a reasonable computational cost. Robustness and consistency are inherent in the HOSVD description of a multi-parameter database. And the combination of the the optimization method, either global or local, with the HOSVD description is computationally inexpensive enough as to allow for considering aerodynamic tools (such as aerodynamics vortex lattice methods) that, without being high fidelity, are a more complex and reliable tool than semi-empirical constructs.

Regarding the organization of the thesis, the required mathematical methods are summarized in chapter 2, where both the required tensor representation/interpolation methods needed to construct the surrogates and the global and local optimization methods are considered.

Chapter 3 is devoted, first, to the general method to construct surrogates of the technical disciplines. The surrogates are based on some snapshots of each technical discipline that are calculated offline using standard numerical tools. The second part of the chapter is devoted to the description/improvement of the appropriate methods to calculate the snapshots. Most part of the offline effort is devoted to aerodynamics calculations, which are performed using a computational fluid dynamics (CFD) solvers that must be carefully selected. CFD solvers are briefly described, as a second goal of the chapter, emphasizing the calculation of viscous drag, and preparing it to be combined with aerodynamics solvers that do not provide directly this property. Thus, some new methods are presented for the direct calculation of viscous drag, using only overall properties of the external aerodynamic field.

A particular test problem is described in chapter 4 that is associated with the conceptual design of a commercial aircraft empennage. In particular, a simple and efficient method to describe the geometry of the empennage is presented and a description of the objective function and the restrictions is provided. The objective function is based on a combination geometrical complexity, drag, and weight, whose definition/calculation-by-classical computational tools is provided. The reference configuration is also described.

Results for global optimization of the empennage are provided in chapter 5, where various optimization campaigns are performed to both calibrate some free parameters/optaining some suboptimal configurations,
CHAPTER 1. INTRODUCTION

and improve the suboptimal solutions. Results are compared with its conventional counterpart, attending to both precision and required computational time. The obtained results and their discussion suggest various further improvements that are also commented.

The local optimization tool is applied to the empennage optimal design in chapter 6. Application of the method requires appropriate preprocessing of both the objective function and the restrictions, which are considered first. The validation of the method requires, in particular, performing some stability stability tests on various runs at representative zones of the design parameters space. The method is then applied in two cases, namely initiated at a random initial position in the design parameters space, which provides suboptimal solutions, as expected, and initiated at those optimals provided by the global optimization tool, which provides a fine tuning of the latter optimals.

The overall results of the thesis are summarized and discussed in chapter 7, where comparison between the various optimization tools developed in the theses will be made. Some possible extensions/improvements of the method presented in the thesis are also considered.
This chapter is devoted to the description of the basic mathematical methods that will be used along the thesis. The main objective is to explain/justify the use of these methods in practical engineering applications, emphasizing the main underlying ideas and the main advantages and drawbacks of the methods.

Discretizing a multidimensional functional relationship among some dependent variables, such as the pressure and the velocity, and some independent variables, such as space, time, and physical parameters leads to a tensor representation that can be considered as a multidimensional database. Physical laws that are inherent to the underlying physical problem lead to redundancies among the data that allow for a much more compact representation of it. In other words, the actual physical information contained in the database is much smaller than the database size or, equivalently, the physically meaningful number of degrees of freedom is much smaller than the total number of database elements. And, even more, since redundancies are expected along all dimensions of the database, the ratio of the numbers of database elements and physical degrees of freedom is expected to grow (exponentially!) with the database dimension. Thus, identifying this physically relevant information is the first step to model the database to facilitate its handling. Standard singular value decomposition (SVD) is a good means for this task, but it is restricted to two-dimensional databases. SVD was invented 150 years ago by Beltrami and Jordan [57] and is today a classical matrix decomposition method that is used for various purposes. Its extension to higher dimensions is not unique, and furthermore some of the extensions bear unsolved mathematical problems (rank of a tensor problem). A good compromise between efficiency in identifying the underlying relevant information and computational cost is the so called high order singular value decomposition (HOSVD), which will be described in section 2.2, where its practical use in engineering environments will be emphasized. A quite important advantage of HOSVD will be considered in section 2.3. It is connected with interpolation in the multidimensional
database, which should be in principle a multidimensional interpolation, a subtle task. HOSVD allows for converting, in a quite natural way, the multidimensional interpolation to a series of one-dimensional interpolations, which are much more easy to handle and can be performed in a quite efficient and robust way. The resulting combination will be called the HOSVD+I method, and will provide most of the surrogate models that will be used in the remaining of the thesis.

The remaining of the chapter will be devoted to optimization methods, considering both global and local methods.

As a global optimization method, able to search (in principle) the whole design parameter space, genetic algorithms (GAs) will be briefly considered in section 2.4, emphasizing its practical use and calibration. An important drawback of GAs is the speed of convergence is quite low.

Gradient-like, local methods are much faster but require a much careful implementation, especially in the presence of restrictions/contraints, defined by either equalities or inequalities. Constrains are quite important in engineering applications since the sought optimal occurs frequently at limiting values of the design parameters defined by some of the constrains. Typical constraints can be given either by some exact relations among the design parameters or by inequalities, which will be referred to as exact constraints and one-sided constraints, respectively. Unconstrained optimization is first considered in subsection 2.5.1, where the selected gradient-like method will be a Broiden method, which does not require calculating the Hessian of the objective function, whose computational cost can be huge. The local character of the method can be extended using a continuation method based on trust regions. Constrained optimization will be considered in subsection 2.5.3, where the selected method will be based on Lagrange multipliers, whose application in the present scenario requires some care. In particular, the number of constrains that must be taken into account must be left free and the method must be combined with the HODVD+I method, which involves some subtleties. And it may happen that the objective function is not defined when some one-sided constraints are not satisfied. For instance, a typical constraint in the design of an aircraft wing results from imposing that the span be positive, and obviously negative spans do not make physical sense.

Appropriate combination of global and local methods may produce a new hybrid method that exhibits the advantage of both. This combination is standard nowadays when classical computational tools are used to calculate the objective function, but its implementation in the present context is more subtle. The extension will be considered in section 2.6.

The chapter ends with some concluding remarks, which will be made
in section 2.7.

2.1 The HOSVD+I method

The object of this section is to provide a general method for constructing surrogates of each technical discipline (e.g., Aerodynamics), whose outcome can be seen as a scalar function of some design parameters and/or physical variables. Namely, the scalar outcome is given by

\[ q = f(\lambda_1, \ldots, \lambda_n). \]  

(2.1)

For example, the state variable, \( q \), can be either the total lift of a wing depending on \( n \) design parameters, \( \lambda^1, \ldots, \lambda^n \), or the overall chord wise lift, depending on \( n-1 \) design parameters, \( \lambda^1, \ldots, \lambda^{n-1} \), and the position along the span, \( \lambda_n \). Discretizing the design parameters and/or physical variables in a structured manner, namely considering all combinations of \( \lambda^1, \ldots, \lambda^n \), such that \( \lambda^k = \lambda^1_k, \ldots, \lambda^n_k \), we can construct the following \( n \)-th order tensor from the function (2.1)

\[ A_{i_1 \ldots i_n} = f(\lambda^1_{i_1}, \ldots, \lambda^n_{i_n}). \]

This tensor builds an \( n \)-dimensional database, whose elements could be obtained using either wind tunnel tests or a computational fluid dynamics (CFD) solver. Anticipating the test problem described in chapter 4, even with the low fidelity method that will be used in the application to conceptual design (namely, a Vortex Lattice Method, which will be referred to hereafter as the VLM method) to calculate aerodynamics, the computational effort can require non-affordable computational resources because the number or realizations required by the global optimization tool can be huge. This difficulty will be highly alleviated using the HOSVD+I that is developed in this subsection. This method is of independent interest, since the size of many aerodynamic databases that are used in daily engineering tasks can be huge (especially if a large number of design parameters/physical variables is present), which can pose difficulties in manipulating the resulting data (to, e.g., calculating loads). Thus, a safe compression of such data is advisable. Multidimensional interpolation is also necessary to obtain the output \( q \) for intermediate values of the design parameters/physical variables.

2.2 High order singular value decomposition

In order to avoid a too involved notation, HOSVD is described for a generic third order tensor; the extension to higher order tensors is straight-
forward. The HOSVD of a \((N_1 \times N_2 \times N_3)\)-tensor is of the form

\[
A_{ijk} = \sum_{p, q, r} \sigma_{pqr} U^p_i V^q_j W^r_k \tag{2.2}
\]

where the new tensor \(\sigma_{pqr}\) is known as the reduced tensor and the three vector families \(\{U^p_i, ..., U^{N_1}_i\}, \{V^q_j, ..., V^{N_2}_j\}\) and \(\{W^r_k, ..., W^{N_3}_k\}\) are known as modes of the decomposition.

The HOSVD can be seen as an extension to tensors of classical SVD, which only applies to matrices, but such extension is not a straightforward one. Let us recall here that the SVD of a matrix \(A_{ij}\) is \([28]\) \(A_{ij} = \sum \sigma_p U^p_i V^p_j\). Its most direct extension to tensors would be such that the coefficients of the modes depend only on one index, namely (cf, eq.\((2.2)\)) \(A_{ijk} = \sum \sigma_p U^p_i V^q_j W^r_k\), but calculating such decomposition with a minimum number of terms is an ill-posed problem \([18]\), except for some quite particular types of third order tensors (e.g., tensors of order \(2 \times N \times N\)). And even defining such minimum number (known as the rank of the tensor) is an open problem nowadays \([20]\). The decomposition \((2.2)\) is called a Tucker decomposition \([34]\) and can be performed in a computationally efficient manner provided that the reduced tensor and the modes are appropriately defined. HOSVD was in fact invented by Tucker \([60]\) 45 years ago and has been revisited more recently by de Lathauwer et al. \([19]\). HOSVD is a Tucker decomposition in which modes are defined as the (orthonormal) eigenvectors associated with the positive eigenvalues of the positive definite, symmetric matrices \(B^1, B^2\) and \(B^3\), defined as

\[
B^1_{il} = \sum_{jk} A_{ijk} A_{ikl}, \quad B^2_{jl} = \sum_{ik} A_{ijk} A_{ilk}, \quad B^3_{kl} = \sum_{ij} A_{ijk} A_{ijl} \tag{2.3}
\]

Namely, the HOSVD modes are given by

\[
\sum_{l=1}^{N_1} B^1_{il} U^p_i = (\alpha_p)^2 U^p_i, \quad p = 1, ..., N_1 \\
\sum_{l=1}^{N_2} B^2_{jl} V^q_j = (\beta_q)^2 V^q_j, \quad q = 1, ..., N_2 \\
\sum_{l=1}^{N_3} B^3_{kl} W^r_k = (\gamma_r)^2 W^r_k, \quad r = 1, ..., N_3 \tag{2.4}
\]

where the positive scalars \(\alpha_p, \beta_q\) and \(\gamma_r\) will be referred to as the high order singular values (HOSVs) of the decomposition. Once the HOSVD modes have been calculated, the reduced tensor \(\sigma_{pqr}\) is readily obtained multiplying eq.\((2.2)\) by the \(i, j, \) and \(k\) components of the HOSVD modes,
2.2. HIGH ORDER SINGULAR VALUE DECOMPOSITION

adding in the indexes $i$, $j$, and $k$, and recalling that HOSVD modes are orthonormal. It follows that

$$\sigma_{pqr} = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sum_{k=1}^{N_3} A_{ijk} U^p_i V^q_j W^r_k.$$  \hfill (2.5)

Substitution of this into (2.2) provides an exact representation of the tensor $A$. But, if the elements of $A$ show redundancies along the three dimensions (which can be due to, e.g., physical laws), then appropriate truncation of the expansion (2.2) still provides a good approximation. In fact, if the three sets of HOSVD modes (namely, the eigenvalues of (2.4)) are sorted in a decreasing order and the decomposition (2.2) is truncated to $S_1 < N_1$, $S_2 < N_2$, and $S_3 < N_3$ terms, as

$$A_{ijk} \approx \sum_{p=1}^{S_1} \sum_{q=1}^{S_2} \sum_{r=1}^{S_3} \sigma_{pqr} U^p_i V^q_j W^r_k.$$ \hfill (2.6)

Then the relative root mean square (RMS) error of the truncated reconstruction of $A$, denoted hereafter as RMSE, is bounded by an a priori relative error bound (APREB), which can be calculated in terms the singular values of the decomposition. In other words, [41]

$$\text{RMSE} \leq \text{APREB} \equiv \left[ \frac{\sum_{p=S_1+1}^{N_1} (\alpha_p)^2 + \sum_{q=S_2+1}^{N_2} (\beta_q)^2 + \sum_{r=S_3+1}^{N_3} (\gamma_r)^2}{\sum_{p=1}^{N_1} (\alpha_p)^2 + \sum_{q=1}^{N_2} (\beta_q)^2 + \sum_{r=1}^{N_3} (\gamma_r)^2} \right]^{1/2} \hfill (2.7)$$

which allows for estimating the error before calculating the decomposition. Thus, truncated HOSVD allows for storing an approximation of the $N_1 \times N_2 \times N_3$ elements of the tensor $A$ by means of only the $S_1 \times S_2 \times S_3 + S_1 \times N_1 + S_2 \times N_2 + S_3 \times N_3$ numbers involved in the right hand side of eq.(2.6). This provides a strong memory saving (and a potential decrease of the CPU time required to manipulate the elements of the tensor) when $S_m$ is small compared to $N_m$. In other words, the compression factor scales with $S_1 \times S_2 \times S_3/N_1 \times N_2 \times N_3$, which means that compression increases exponentially as the dimension of the tensor increases [41]. This emphasizes the fact that because of redundancies, the relevant information in a database is much smaller than its size, and that the ratio between both can be quite small if all redundancies along all dimensions in the database are appropriately taken advantage of (as done when using HOSVD). Summarizing, the HOSVD modes and singular values are calculated from the eigenvalue problems (2.4), with the coefficient matrices $B^1$, $B^2$, and $B^3$ as given in eq.(2.3). The a priori error estimate (2.7) allows for truncating the
decomposition (2.6) to meet error requirements. Finally, the reduced tensor is calculated using \(eq. (2.5)\).

Some care must be taken to optimize the computational cost of the process:

1. Equation (2.3) may involve a huge number of operations, especially in higher dimensional tensors. In order to control this in the calculation of, e.g., \(B^1\), when \(N_2 \times N_3\) is large compared with \(N_1\), the second and third indexes of the tensor \(A\) are sorted together to obtain a matrix, with \(N_1\) rows and \(N_2 \times N_3\) columns, which is such that \(B^1 = \tilde{A}^\top \tilde{A}\). Application of a QR decomposition to such matrix yields \(\tilde{A}E = QR\), where \(E\) is a column permutation orthogonal matrix, \(Q\) is an orthogonal matrix, and \(R\) is an upper triangular matrix. Thus, \(B^1\) is calculated as \(B^1 = \tilde{A}^\top \tilde{A} = ER^\top QE^\top = ER^\top RE^\top\), which involves much fewer operations than those required in eq.(2.3).

2. Appropriate factoring of the right hand sides of eqs.(2.5) and (2.6) (to avoid repeating calculations) greatly improves computational efficiency.

Now, with the above HOSVD decomposition we can construct a compressed model of the database that results from discretizing the function (2.1). Considering, as above, the case \(n=3\), the HOSVD representation is of the form

\[ f(\lambda_1, \lambda_2, \lambda_3) = A_{ijk} \approx \sum_{p=1}^{S_1} \sum_{q=1}^{S_2} \sum_{r=1}^{S_3} \sigma_{pqr} U_p^i V_q^j W_r^k. \]  

(2.8)

### 2.3 Multi-dimensional interpolation

Equation (2.8) provides the values of the function \(f\) at the discretized values of the independent variables. Such representation can also be seen as

\[ f(\lambda^1, \lambda^2, \lambda^3) \approx \sum_{p=1}^{S_1} \sum_{q=1}^{S_2} \sum_{r=1}^{S_3} \sigma_{pqr} u^p(\lambda^1) v^q(\lambda^2) w^r(\lambda^3). \]

where the functions \(u^p, v^q\) and \(w^r\) are defined in terms of the HOSVD modes as

\[ u^p(\lambda^1_p) = U_p^i, \quad v^q(\lambda^2_q) = V_q^j, \quad w^r(\lambda^3_k) = W_k^r. \]

The values of these three functions at other, intermediate values of \(\lambda^1, \lambda^2,\) and \(\lambda^3\) are readily obtained by (e.g., spline) interpolation. Thus, HOSVD allows for reducing the three-dimensional interpolation that would be needed
in principle to three one-dimensional interpolations, which are easier and much more computationally efficient. This method to obtain surrogates of technical disciplines will be referred to as HOSVD+I below.

Let us now have a look at interpolation itself, not restricting to one-dimensional interpolation. Multidimensional interpolation in a vector variable \( \lambda = (\lambda_1, \ldots, \lambda_N) \) is needed whenever a function \( \phi = \phi(\lambda) \) (whose analytic expression is not at hand) is only known at a set of points \( \lambda_1, \ldots, \lambda_n \), called data set, and the value of \( \phi \) at a new point \( \lambda^{\text{int}} \) is required. A minimal convex (hyper-)polyhedron can be defined with the points where the function is known. If \( \lambda^{\text{int}} \) is inside the polyhedron, determination of \( \phi(\lambda) \) is called interpolation; otherwise is called extrapolation.

Interpolation is made below in two steps: (i) fit an interpolating function to the data set and (ii) evaluate that interpolating function at the required new point \( \lambda^{\text{int}} \). Generally speaking, the most accurate and time consuming methods are global interpolation methods, namely methods in which the interpolation function is calculated using the available information from all points in the data set. On the other hand, when extreme accuracy is not required and/or the computational time cost must be kept small local interpolation methods (which only used the information from a vicinity of the target point) can be advisable.

Cubic splines are a good compromise in one-dimensional local interpolation. These are piecewise cubic polynomials with continuous first and second derivatives at the data set. Depending on the two extra conditions imposed at the extreme points in the interpolating interval, various spline algorithms are obtained. Natural cubic splines [58] and Akima’s splines [1] are most frequently used. Typically, the former provide a good error properties when interpolating functions with continuous fourth derivatives, while the latter is to be preferred under weaker smoothness requirements.

Similarly, concerning two-dimensional interpolation:

- If high accuracy is not needed, Shepard’s interpolation [55] is a good choice. This local method does not require a Cartesian mesh and uses four neighboring points, weighted with the inverse of the distance to the target point.

- A more accurate method is Akima’s bivariate interpolation for scattered data [2]. This method uses a Delaunay triangulation to generate a mesh whose vertices are at the data set points. Using these, a fifth order polynomial is used, whose coefficients are calculating upon fitting each triangular cell.

- When the date set conform a structured (not necessarily Cartesian) mesh, the interpolated value can be obtained using SVD plus inter-
CHAPTER 2. MATHEMATICAL METHODS

Interpolation, which is the restriction of the HOSVD+I described above to two-dimensional tensors; in fact, this two-dimensional case was already considered by Bui Thanh [12]. SVD decompose the data set into modes, as

\[ \phi(\lambda_1^r, \lambda_2^s) = \sum_k \sigma_k U_k(\lambda_1^r) V_k(\lambda_2^s), \]

which reduces two-dimensional interpolation to two one-dimensional interpolations, one for each mode, \( U_k \) or \( V_k \). The latter is performed using splines.

2.4 Global optimization: genetic algorithms

Genetic Algorithms (GAs) are optimization methods that exhibit the advantages of providing the global minimum and being robust. The main drawback is associated with the computational cost, since convergence is usually quite low compared to, e.g., the gradient like algorithms that will be described in next section.

A basic GA is used to calculate the global minimum of a function \( \phi = \phi(\lambda_1, \ldots, \lambda_N) \). Beginning with an initial guess, the GA is an iterative method in which individuals evolve in iterative steps, called generations, until convergence occurs:

- \( N_1 \) individuals are considered in each generation. Each individual is defined by its \( N_m \) chromosomes, which are the unknowns \( \lambda_k \) that are to be selected to minimize the objective function. Each chromosome in turn contains \( g \) genes, which are the bits that codify each chromosome (namely, each optimization variable). The unknowns cannot take arbitrary values but are restricted to some intervals (one for each chromosome).

- The fitness of each individual (which is to be minimized) is defined according to the value of the objective function \( \phi \) at the genes of the individual.

- The individuals of each generation compete between themselves, mutate, and breed, as follows to produce new individuals for the next generation.

- In the first generation, the algorithm uses an initial number, \( \alpha_1 \times N_1 \), of equal individuals with the genes of the above mentioned initial guess; the remaining \( (1 - \alpha_1) \times N_1 \) individuals are selected with randomly chosen genes. These individuals form the initial population in the first generation.
2.4. GLOBAL OPTIMIZATION: GENETIC ALGORITHMS

- In the remaining generations, the initial population is obtained from the former population as follows. In each generation, the individuals in the initial population are ordered according to their fitness. The first $\alpha_E \times N_I$ individuals are called *elite individuals* and survive to the next generation no matter what the results of the following operations could be. The remaining individuals compete randomly among themselves, namely two randomly chosen individuals compare their fitness and it is only the more fit that survives. Then, the *survivors* cross their genes in randomly chosen pairs to produce $(1 - \alpha_E) \times N_I$ *new individuals*, whose genes are chosen randomly from the genes of the parents; such crossover operation have a probability of occurring equal to $\alpha_C$. Each gene has a probability $\alpha_M$ of suffering a further *random mutation*. The resulting new individuals plus the elite individuals form the initial population for the next generation.

- The whole process is repeated, either for $N_G$ generations or until the value of the fitness remains unchanged for $N_g$ generations. Then, that individual with the smallest fitness is considered the optimal individual and its genes codify the solution of the minimization problem.

Note that the method exhibits various intrinsic parameters, which need calibration. These *tunable parameters* are:

- $N_I$, the number of individuals per generation.
- $\ell_j$, the intervals in which the genes are allowed to vary for each chromosome.
- $\alpha_I$, the fraction of individuals whose initial values of the genes are equal to that of the initial guess.
- $\alpha_E$, the fraction of individuals considered as elite individuals in each generation.
- $\alpha_C$, the probability of crossover.
- $\alpha_M$, the probability of mutation.
- $N_G$, the maximum allowed number of generations.
- $N_g$, the number of generations in which fitness remains constant to accept convergence.

Note that imposing constraints (either exact or one-sided) is straightforward in genetic algorithms: only individuals satisfying the constraints are allowed along the process.
2.5 Local optimization: gradient-like methods

Optimization problems are of paramount industrial and scientific interests. Thus, they have received attention in the literature for a long time, beginning with early, classical local methods in the eighteenth Century. The common feature of these methods is that they iterate using local approximations of the left hand side of the objective function, which requires smoothness properties. In particular, they always use the gradient of the objective function. Thus, they are known as gradient-like methods. A large variety of local methods have been proposed that are summarized in [24], where a brief description of the local minima search evolution is also provided. A common feature is that these methods are usually based on ‘models’ that somewhat approximate the objective function in a neighborhood of the local minima that are being sought. Most successful/popular methods are those that somewhat use a quadratic approximation of the objective function.

The main advantage of gradient like methods is that convergence is usually superlinear, much faster than that of global methods such as genetic algorithms. But they also exhibit some drawbacks compared to genetic algorithms, namely:

- They are less robust and their implementation is more subtle, especially in the presence of constraints.
- They provide local minimizers, not global.
- A sufficiently good initial approximation must be provided.

The simplest gradient-like methods, for the unconstrained case, will be considered in subsection 2.5.1, where the standard gradient-like methods will be first considered, and the trust region method (which can be seen as a continuation method, which uses iteratively a gradient-like method in appropriate regions and does not need a good initial iterate) will be also described. Constrained optimization will be considered next, in subsection 2.5.3.

2.5.1 Unconstrained optimization

Gradient like methods are iterative, local optimization methods that can be seen as a generalization of standard, one-dimensional Newton/secant methods to higher dimensions. These methods reduce the problem of minimizing the objective function \( \phi = \phi(\lambda_1, \ldots, \lambda_N) \) to calculating stationary points of the gradient, namely to solving the vector equation

\[
\phi_\lambda = 0,
\]  

(2.9)
where $\lambda = (\lambda_1, \ldots, \lambda_N)$ and $\phi_\lambda$ denotes the gradient of $\phi$.

The Newton method solves this equation iteratively, the result of each iteration resulting from minimizing a local quadratic approximation of the objective function, which requires solving the linear problem. In this sense, the Newton method is a quadratic method. It exhibits some advantages [24], namely:

- Quadratic functions are smooth and simple to handle, and their minima are easy to determine.
- A general function expanded about a local minimum is well approximated by a quadratic function.
- Methods based on quadratic approximations can be made invariant under linear transformations of the variables.

The quadratic approximation of the objective function is obtained from a truncated Taylor series expansion of $\phi(\lambda)$ in the vicinity of the former iteration $\lambda^{(k)}$, namely for $\lambda = \lambda^{(k)} + \delta$, with $|\delta| \ll 1$,

$$
\Phi(\lambda^{(k)} + \delta) \approx \phi^{(k)} + g^{(k)} \delta + \frac{1}{2} \delta^T H^{(k)} \delta, \quad (2.10)
$$

where the scalar $\phi^{(k)}$, vector $g^{(k)}$, and the matrix $H^{(k)}$ are the objective function, its gradient, and its Hessian matrix, respectively, all calculated at $\lambda^{(k)}$. Minimizing this quadratic function provides the following linear system in $\delta^{(k)}$

$$
H^{(k)} \delta^{(k)} + g^{(k)} = 0, \quad (2.11)
$$

which uniquely determines $\delta^{(k)}$ if the Hessian matrix is nonsingular. The solution can be written as

$$
\delta^{(k)} = -M^{(k)} g^{(k)} \quad \text{with} \quad M^{(k)} = [H^{(k)}]^{-1}.
$$

In principle, the next iterate could be taken as

$$
\lambda^{(k+1)} = \lambda^{(k)} - M^{(k)} g^{(k)},
$$

which would be the most basic version of the Newton method. But it may well happen that the value of the objective function at the next iterate is larger than at the former iterate. In order to avoid this, a further one dimensional problem is considered along the straight line with direction $d^{(k)}$, namely along the straight line

$$
\lambda^{(k+1)} = \lambda^{(k)} + \alpha^{(k)} d^{(k)}, \quad \text{with} \quad d^{(k)} = M^{(k)} g^{(k)}/|M^{(k)} g^{(k)}|. \quad (2.12)
$$
In other words, the scalar $\alpha^{(k)}$ is calculated minimizing

$$\phi(\lambda^{(k)} + \alpha^{(k)} d^{(k)}).$$  \hfill (2.13)

Sometimes, it is the method with this additional one dimensional minimization step that is called the Newton method. In either case, once the iteration steps are defined, the iterative method proceeds until a point is reached where the gradient of $\phi$ is appropriately small, namely until that step where

$$|g^{(k)}| \ll \varepsilon_c,$$  \hfill (2.14)

for some small residual bound, $\varepsilon_c$.

If the Hessian matrix is well conditioned and positive definite, then the Newton method is generally a quadratic method, namely the error is squared in each iteration step, provided that the initial iterate is sufficiently close to a local minimum. Nevertheless, the method exhibits two main drawbacks:

- It requires calculating (and storing) the Hessian matrix, which can be quite computationally expensive.
- If the Hessian matrix is ill-conditioned, then solving the linear system (2.11) can be a subtle problem.

Thus, other methods have been developed that can be seen as the multidimensional extension of the secant method, which is not straightforward. Note that, in one dimension, the secant method can be seen as either (i) solving the equation along the secant at each iteration or (ii) applying the Newton method, but approximating the first derivative of the left hand side of the equation using the last two iterates. Similarly, in higher dimensions, several extensions exist that can be seen as either:

- Restricting to a line the approximation in each iterate, as we did in (2.13). These methods are known as descend methods. The main difficulty relies in guessing the direction $d^{(k)}$ in each iterate.

- Solving iteratively the system of linear equations involved in each iteration of the Newton method, and somewhat guessing the Hessian matrix. These methods are known as quasi-Newton methods. There is a great difference between the one dimensional and the multidimensional cases in connection with guessing the Hessian matrix. Namely, it only requires guessing one scalar in one dimension, which is readily done in the secant method using the last two iterates. But guessing the Hessian matrix from the last two iterates is undetermined in higher dimensions.
2.5. LOCAL OPTIMIZATION: GRADIENT-LIKE METHODS

Let us begin with descend methods. The iterates are sought in search lines of the form

\[ \lambda^{(k+1)} = \lambda^{(k)} + \alpha^{(k)} d^{(k)}, \]

where the calculation of the direction \( d^{(k)} \) is precisely what distinguishes the various descend methods. The scalar \( \alpha^{(k)} \) is calculated solving the one dimensional minimization problem

\[ \phi(\lambda^{(k)} + \alpha^{(k)} d^{(k)}), \quad (2.15) \]

which can be done using standard one dimensional methods; for instance, the one dimensional secant method is frequently a good choice. Note that the minimum of this function must be a stationary point, which means that the gradient at the new iteration \( \lambda^{(k)} \) must be such that

\[ [d^{(k)}]^{\top} g^{(k+1)} = 0. \quad (2.16) \]

The simplest descend method is the steepest descend (SD) method, which bears the quite intuitive idea of moving along that direction where the objective function shows the largest descend. This direction is precisely the direction of the gradient of the objective function, namely the vector \( d^{(k)} \) in (2.13) is

\[ d^{(k)} = g^{(k)}/||g^{(k)}||, \]

where \( g^{(k)} \) is the gradient of the objective function at the former iterate \( \lambda^{(k)} \). The fact that the steepest descend is given by the gradient is readily seen noticing that the gradient is orthogonal to the level lines of the objective function.

The main drawback of the SD method is that, intuitive as it is, it may happen that the steepest descend direction is not the most appropriate one. This is seen in the quadratic approximation (2.10) noting that (considering the two dimensional case for illustration) the level lines of the right hand side of (2.10) as iteration proceeds are ellipses, such as those sketched in Fig.2.1. Note that (to the quadratic approximation relevant here) the minimum of the objective function is the common center of these ellipses. Then each iteration in SD method (left plot) searches the minimum along the normals to the ellipses, which does not contain the center if the ellipses exhibit a non-unit eccentricity (namely, if they are not circles). Thus, the method needs various iteration steps just to reach a quadratic approximation. Furthermore, the smaller the eccentricity, the larger the required number of steps. In other words, the performance of the SD method is not expected to be acceptable when the Hessian matrix is ill-conditioned.
A possible solution to this difficulty is provided by the conjugate gradient method. In the two dimensional case considered above, the best direction would be that direction that is $H$-conjugate to the level sets of the objective function. If the restriction, (2.15) has been minimized in the previous step, then $d^{(k)}$ is tangent to the level sets, and the new search direction in each iteration step should be $H$-conjugate to the former search direction, namely such that

$$[d^{(k-1)}]^T H^{(k)} d^{(k)} = 0. \quad (2.17)$$

The argument above applies to quadratic nonlinearities (which if fact are relevant near a minimum of the objective function), but an additional argument can be used to justify (2.17). The argument relies in choosing that direction for the new iterate such that the gradient of $\phi$ along the new search line is as orthogonal as possible to the former search line. This is to impose that the new minimization in the new search line ‘does not spoil’ a part of the former minimization, in the former search line. In other words, it would be advisable that

$$[d^{(k-1)}]^T \nabla \phi(\lambda^{(k)} + \alpha d^{(k)}) \approx [d^{(k-1)}]^T [g^{(k)} + \alpha H^{(k)} d^{(k)}] \quad (2.18)$$

be as small as possible for small $|\alpha|$. But, using (2.16) (with $k = k - 1$), it turns out that the right hand side of (2.18) can be set to zero for all $\alpha$ precisely if (2.17) holds.

Now, the conjugancy condition (2.17) can be imposed in many (asymptotically equivalent) ways. A definition of the new search direction in each iteration step that satisfies (asymptotically) the conjugancy condition is the so called Fletcher-Reeves algorithm

$$d^{(k)} = -g^{(k)} + \beta_k d^{(k-1)},$$
where

$$\beta_k = \frac{|g^{(k)}|^2}{|g^{(k-1)}|^2}.$$ 

Let us now consider quasi-Newton methods, which as in the standard Newton method, provides the next iterate as in (2.12), minimizing (2.13), but estimating, instead of calculating the inverse of the Hessian matrix, $M^{(k)}$, for the next iteration. The various quasi-Newton methods differ among each other precisely on how this matrix is estimated in each iteration step. Many of those methods have been tested in the optimization test problem that will be addressed in this thesis. The conclusion was that the BFGS method, suggested by Broyden [11], Fletcher [23], Goldfarb [27] and Shanno [54], has provided the best results. The inverse of the Hessian matrix for the next iteration step is calculated as

$$M^{(k+1)} = M^{(k)} + \left( 1 + \frac{\gamma^T M^{(k)} \gamma}{\delta^T \gamma} \right) \delta \delta^T - \frac{\delta \gamma^T M^{(k)} + M^{(k)} \gamma \delta^T}{\delta^T \gamma},$$

(2.19)

where $\gamma = g^{(k)} - g^{(k-1)}$ and $\delta = \lambda^{(k)} - \lambda^{(k-1)}$ (the superscript $(k)$ has been skipped in these two vectors to simplify notation) can be calculated after performing the one-dimensional minimization of (2.13) in the present iteration. Similarly, the Hessian matrix itself, $H$, is approximated as

$$H^{(k+1)} = H^{(k)} + \frac{\gamma^T \gamma}{\delta^T \delta} - \frac{H^{(k)} \delta \delta^T H^{(k)}}{\delta^T H^{(k)} \delta}.$$ 

Equation (2.19) requires an initial guess, which can be taken as the identity matrix, namely $M^{(1)} = I$. Figure 2.1 compares convergence properties in the SD method and BFGS methods, considering a simple quadratic function. Note that while the BFGS method reaches the solution using only two steps, the SD method is not yet converged after six steps.

2.5.2 Trust regions

The methods described in previous section are local methods and require a sufficiently good initial estimate, where what ‘sufficiently good’ means in each particular case is not easily guessed. In fact, it may well happen that the solution that is being sought is not contained in the neighborhood of the initial estimate where the various approximations made above apply. In this case, the method does not converge. A possible strategy to overcome this difficulty is to apply a continuation-like method that proceeds as follows.

---

21
A local method is applied in a circular neighborhood of the initial estimate, of radius $R^{(1)}$. This neighborhood is known as the first trust region. In other words, the minimum of the one-dimensional problems (2.13) and (2.15) (for Newton/quasi-Newton and descend methods, respectively) is sought only in the trust region. Note that the minimum, $\lambda^{(1)}$, can either be an interior point of the trust region or a point in its boundary. In the latter case, the actual minimum is not calculated since doing that would require solving a constrained optimization problem. Instead, an approximation is calculated as follows. If an iterate of the gradient-like method leaves the trust region, then it is assumed that the minimum is in the boundary and it is approximated by the intersection of the last search line with the boundary. In either case, a new trust region of radius $R^{(2)}$ is considered with center at $\lambda^{(1)}$, which provides a new iterate $\lambda^{(2)}$. The procedure continues until condition (2.14) holds. The method is adaptive, in the sense the radii of the trust regions, $R^{(1)}, R^{(2)}, \ldots$ are selected by the method as follows. An instrumental variable $\rho$ is defined to adjust the trust region radius. This variable is restricted to the $[0, 1]$ interval and somewhat measures how well eq.(2.9) is approximated in each iteration. In other words, the closer $\rho^{(k)}$ is to the unity the better the approximation is. The trust region radius is increased in the next iteration if $\rho$ is close to one, and is decreased if $\rho$ is small. A reasonable strategy is to select the radius of the trust region in the next iteration as

- $R^{(k+1)} = 2R^{(k)}$ if $\rho^{(k)} > 0.75$,
- $R^{(k+1)} = R^{(k)}$ if $0.75 \geq \rho^{(k)} \geq 0.25$,
- $R^{(k+1)} = R^{(k)}/2$ if $0.25 > \rho^{(k)}$,

In descend methods, $\rho^{(k)} = |\alpha^{(k)}|/R^{(k)}$, where $\alpha^{(k)}$ is the minimizer of the one dimensional problem (2.15)). However, in quasi-Newton methods a different selection is advisable that takes into account the approximated Hessian. In particular, in the BFGS method $\rho^{(k)}$ is selected as (using the same notation as in the last subsection)

$$
\rho^{(k)} = \frac{|\phi^{(k+1)} - \phi^{(k)}|}{\alpha^{(k)}R^{(k)}[g^{(k)}]^{\top}d^{(k)} + 0.5|\alpha^{(k)}R^{(k)}|^{2}[d^{(k)}]^{\top}H^{(k)}d^{(k)}}.
$$

Using the above, the trust regions radii adapt itself to possible, sudden changes in the objective function local behavior, and generally converges to a (sub-optimal) minimizer of the objective function.

Trust regions are specially useful when one-sided constraints are considered. These kind of constraints, which are considered in the next subsection, are used to restrict the optimization problem to a subset of the
2.5. LOCAL OPTIMIZATION: GRADIENT-LIKE METHODS

Parameter space. In these case, if local methods are applied without using trust regions, it is possible that some iterate, \( \lambda^{(k+1)} \), does not satisfy the constraints. Trust regions avoid this, imposing in each one dimensional minimization (using either descend or quasi-Newton methods) that the minimizer satisfies the restrictions (in addition to the requirement that it belongs to the trust region). In order to illustrate this, the following simple two-dimensional example is considered

\[
\phi(\lambda_1, \lambda_2) = \lambda_1^2 + \lambda_2^2, \\
\text{with } C(\lambda_1, \lambda_2) \equiv \lambda_1^2 - 10\lambda_2 - 50 \geq 0. 
\tag{2.20}
\]

Taking as initial guess \( x^{(k)} = (20, 20) \), which satisfies the constraint in (2.20) \( C(20, 20) = 15 \), the first iterate in the quasi-Newton method is \( \mu^1 = (0, 0) \), which is the objective function minimum in the whole plane but does not satisfy the problem constriction because \( C(0, 0) = -5 < 0 \). However, if trust regions are used, their radii are updated after each iteration to reach the constriction border without overstepping it. Figure 2.2 illustrates both processes.

![Figure 2.2: Quasi-Newton method applied to the minimization problem (2.20), without (left) and with (right) trust regions, considering the constraint border (dashed line), the initial guess (circle), the attraction point (square), and the optimization path (solid line).](image)

Note, however, that the solution provided by the quasi-Newton/trust regions method is not an acceptable minimizer, whose calculation requires solving the minimization problem along the border of the allowable subset of the parameter space defined by the (one-sided) restriction. This restricted problem is considered in the next subsection.

2.5.3 Optimization with constraints

As explained at the beginning of this chapter, constraints are very important in engineering design and, in fact, the relevant minimizers frequently belong to the border of the allowable subset of the parameter space...
defined by one-sided constraints. Constraints have been largely ignored in the local methods quoted above, except for the remark in connection with trust regions at the end of the last subsection. The strictly local methods developed above, may either not converge in the allowable region or converge to an interior point that is only a suboptimal solution.

The constrained problem is formulated as

\[
\phi = \phi(\lambda)
\]

with \(C_E^i(\lambda) = 0 \) for \(i = 1, \ldots, N_E\) and \(C_I^j(\lambda) \geq 0 \) for \(j = 1, \ldots, N_I\) (2.21)

where the objective function \(\phi\) and the left hand sides of the constraints, \(C_E^i\) and \(C_I^j\), are assumed to be smooth and generally nonlinear. There are many methods in the literature to solve this problem. Most of them are based on the addition of penalty functions to the objective function, in such a way that if some constriction is not met, then the objective function value increases. The description of some of these penalty methods is the object of this subsection, along with their advantages and drawbacks.

### 2.5.3.1 Penalty methods

One of the most popular penalty methods is due to Courant [17], who replaces the objective function by a family of new functions (for \(m = 1, 2, \ldots\))

\[
\phi^{(m)}(\lambda, \sigma_E^m, \sigma_I^m) = \phi(\lambda) + \sigma_E^m \sum_i (C_E^i(\lambda))^2 + \sigma_I^m \sum_j \min((C_I^j(\lambda)), 0)^2,
\]

where the coefficients \(\sigma_E^m > 0\) and \(\sigma_I^m > 0\) increase as \(m\) increase, according to a tunable law. The optimization problems for these new objective functions are solved and the final solution is obtained as \(\sigma_E^m, \sigma_I^m \to \infty\). Note that the second derivatives of the new objective functions are not generally continuous due to that term accounting for one-sided constraints, which may pose some difficulties, but this can be doctored smoothing the contribution of these constraints. Also note that this penalty method requires that the objective function be defined in the whole parameter space, even at those points that do not satisfy the constraints. In principle, this can always be accomplished by appropriately extending the original objective function (which can generally be done in infinitely many ways), but such extension can be not obvious, and difficult to implement, see below.

Other methods have been invented for one-sided constraints that only require defining the objective function inside the allowable subset of the parameter space defined by the constraints. Two such methods are due to [13], who uses the following penalty objective function

\[
\phi(x, \sigma) = \phi(x) + r \sum_i (C_I^i(x))^{-1}
\]
2.5. LOCAL OPTIMIZATION: GRADIENT-LIKE METHODS

and to Frisch [26], who replaces the original objective function by

\[ \phi(\lambda, \sigma) = \phi(x) - \tau \sum_i \log(C_i(x)) \]

The final solution with Carrol and Frisch penalty functions is obtained as \( \tau \to 0 \). Both Carrol and and Frisch penalty functions avoid stepping out of the constraint border. Thus, they can be applied to the case study in chapter 4. In fact, these methods have been tested in this test problem concluding that:

- Third order derivatives are large near the border. Thus, the BFGS method is very slow and not appropriate to approximate the Hessian.

- Constraints cannot be exactly equal to zero because of the penalty function method nature.

- When using an initial guess provided by a genetic algorithm and this guess is in the border region, it may happen that the results provided by the method worsens the initial iterate.

As somewhat anticipated by the end of the last subsection, both exact and one-sided constraints can also be dealt with using a combination of Lagrange multipliers (which are appropriate to deal with exact constraints) and trust regions, which will be considered below.

2.5.3.2 Lagrange multipliers

Lagrange multipliers are useful to solve optimization problems with exact constraints. But since this method will be combined with trust regions, the one-sided constraints will be turned into exact constraints, which means seeking solutions in the border of the allowable region. In other words, the optimization problem (2.21) is substituted by

\[ \phi = \phi(\lambda) \]

\[ C^E_i(\lambda) = 0, \quad C^I_j(\lambda) = 0 \text{ for } i = 1, \ldots, N^E \text{ and } j = 1, \ldots, N^I. \]

The Lagrange multipliers method solves this problem noting that minimizing the restriction of the objective function to the (generally nonlinear) manifold defined by the restrictions, requires that the intrinsic gradient of the restricted function vanishes. But the intrinsic gradient is the orthogonal projection of the plain gradient onto the tangent plane of the manifold, which is defined as the span of those vectors that are orthogonal to the gradients of the restrictions. Thus, the above mentioned condition is equivalent to imposing that the plain gradient of the objective function be linearly
dependent to the gradients of the restrictions, namely to imposing that
\[ \nabla \phi = \sum_i \mu_i^E \nabla C_i^E + \sum_j \mu_j^I \nabla C_j^I, \]
which in turn is equivalent to
\[ \frac{\partial \phi}{\partial \lambda_j} = \sum_i \mu_i^E \frac{\partial C_i^E}{\partial \lambda_j} + \sum_i \mu_j^I \frac{\partial C_j^I}{\partial \lambda_j}, \] (2.22)
\[ C_i^E = 0 \text{ for } i = 1, \ldots, N^E, \] (2.23)
\[ C_j^I = 0 \text{ for } j = 1, \ldots, N^I. \] (2.24)

where we have also imposed that the restrictions be satisfied. This is a system of \( N + N^I + N^E \) equations with \( N + N^I + N^E \) unknowns (namely, the \( N^I + N^E \) coefficients \( \mu_i^E \) and \( \mu_j^I \) and the \( N \) design parameters \( \lambda_l \)). This is the system that is to be solved. Note that the system depends linearly on the coefficients \( \mu_i^E \) and \( \mu_j^I \), but the dependence on the design parameters is generally nonlinear. The solutions of this system are candidates of local restricted minima.

When the objective function is defined in a neighborhood of the restrictions, standard quasi-Newton methods can be used to solve the problem (2.22)-(2.24). But this is not always the case. Thus, a method must be devised to deal with the case in which the objective function is not defined at the wrong side of some of the one-sided restrictions, which is done in the next subsection

2.5.3.3 Optimization within the constraints border

In principle, the system of equations (2.22)-(2.24) can be solved using the various local methods developed in subsection 2.5.1, albeit it is the counterparts of these methods for solving general sets of equations (whose Jacobian matrix needs not be symmetric) that should be used. But if one-sided constraints are present, then it may happen that the objective function and the remaining constraints are not defined when the one-sided constraints are negative, which may happen along the local methods iteration because the equations (2.24) are not satisfied during the iteration and can take either positive or negative values. Thus, some care must be taken to ensure that all iterations satisfy (2.24), namely that they belong to the constraints border.

In order to ensure that conditions (2.24) are exactly satisfied in each iteration step (assuming that they are satisfied in the former step), we note that if
\[ |\nabla \phi - \sum_i \mu_i^E \nabla C_i^E - \sum_j \mu_j^I \nabla C_j^I| < \varepsilon_c, \] (2.25)
for some fixed $\varepsilon_c$, then the process can be terminated. Otherwise, the new steepest descend direction, $d_k$, is taken as the orthogonal projection of $\nabla \phi - \sum_i \mu_i^k \nabla C_i^k - \sum_j \mu_j^k \nabla C_j^k$ onto the tangent hyperplane of the constrains. This is the right search direction to proceed, but the one-dimensional minimization (2.15) is not made along the straight line in the search direction (which do not generally satisfy the one-sided constraints). Instead, it is made along the orthogonal projection of this line on the constraint border. Such projection requires solving a set of nonlinear equations, namely

$$C_j(\lambda_k + \alpha^{(k)} d^{(k)} + \sum \beta_j \nabla C_j) = 0,$$

where the index $j = 1, \ldots, \tilde{N}$ applies to those constraints (either exact of one-sided) that are being considered. This system of $\tilde{N}$ equations provides the $\tilde{N}$ unknowns $\beta_j$. They can be solved using a Newton method.

In order to illustrate the process, the following constraint problem is considered

$$\phi(\lambda_1, \lambda_2) = \lambda_1^2 + \lambda_2^2, \quad \text{with} \quad C(\lambda_1, \lambda_2) = \lambda_1^2 - 10\lambda_2 - 50 \geq 0. \quad (2.27)$$

Fig.2.3 shows a scheme of the process. The example with the one-dimensional constraint could be seen as a too simple example. In fact, it could be treated in a simpler way. But its two-dimensional character facilitates graphical illustration. The stop condition is (2.25), with $\varepsilon_c = 10^{-6}$. The initial guess is $\lambda^0 = (15, 17.5)$ (circle in Fig.2.3 top), which satisfies the exact constraint equation in (2.27), namely it belongs to the constraint border, which is the parabola plotted with solid line. At this point, $\nabla \phi = (2\lambda_1, 2\lambda_2) = (30, 35)$ defines the straight line that also contains square point in Fig.2.3, which is not tangent to the constraint border. Using the constrain gradient at this point, $\nabla C = (\lambda_1/5, -1) = (3, -1)$, the vector $\nabla \phi$ is projected onto the tangent of the constraint border, to obtain the search direction $d = (-13.5, -40.5)$. One-dimensional optimization of the objective function is performed along the line defined by equation (2.26) to obtain $\lambda^1 = (5.6, -1.87)$ (upper star in Fig.2.3-bottom left), which is the projection on the constraint border of the point $\lambda = (8.25, -2.75)$ (right triangled point in Fig.2.3-bottom left). Continuing the process, 5 additional iterations are needed that provide those points plotted with stars in Fig.2.3-bottom, namely $\lambda^2 = (3.9, -3.47), \ldots$, until the final solution that satisfies the stop condition is obtained, namely $\lambda^5 = (0, -5)$ (diamond in Fig.2.3).

### 2.5.3.4 The Karush-Kuhn-Tucker conditions

The above concerns the restricted objective function, namely the case in which the inequality restrictions are converted into equalities. The solutions of (2.22)-(2.24) are required to be stationary points of the restricted
objective function. The additional condition (known as dual feasibility condition)

$$\mu^I \geq 0$$

(2.28)

imposes that the gradient of the objective function points to the interior of the allowable subset of the parameter domain, meaning that the objective function increases towards the interior of the allowable region, as required by the inequality restrictions. Thus, this condition is necessary for the stationary point to be a possible minimum when the inequality restrictions in (2.21) are taken into account, but still only points in the border of the allowable region are considered. But in fact, points inside the allowable region are readily considered noticing that in that case, conditions (2.24) do not
apply and conditions (2.22) must be replaced by
\[
\frac{\partial \phi}{\partial \lambda_j} = \sum_i \mu^E_i \frac{\partial C^E_i}{\partial \lambda_j} + \sum_i \mu^I_i \frac{\partial C^I_i}{\partial \lambda_j},
\]
which coincide with (2.22) setting \( \mu^I_i = 0 \). Both cases, namely interior minima and minima at the border, can be considered together using the so-called complementary slackness condition
\[
\mu^I_j C^I_j = 0 \quad \text{for} \quad j = 1, \ldots, N^I.
\]  
(2.29)

Summarizing the above, the necessary conditions for a point to be a minimum of the constrained problem (2.21) are (2.22)-(2.23), (2.28), and (2.29). These set of four (necessary) conditions are known as the Karush-Kuhn-Tucker conditions.

The Karush-Kuhn-Tucker conditions can be seen as the extension of the Lagrange multipliers conditions when one-sided constraints are present. But even with these conditions, the extension of the local methods (either descend methods or quasi-Newton) developed so far alone do not yield a practical algorithm when the objective function is not defined outside the allowable set of the parameter space, as it happens in the test problem that is to be considered in this thesis. The appropriate algorithm will be obtained in the next subsection combining the above with the trust regions method.

2.5.3.5 The combined algorithm for constraint optimization

The object of this section is to combine the local methods developed in section 2.5.1, modified according to the Lagrange multipliers/Karush-Kuhn-Tucker conditions with trust regions. The idea is to use trust regions both inside the allowable subset of the design parameter space and at the border of this region. In fact, the local method that is going to be:

- A quasi-Newton BFGS method to minimize the objective function at interior points.
- A modified steepest descend at the border, as explained in subsection 2.5.3.3.

The whole process is summarized in Fig.2.4.

For the sake of simplicity, exact constraints are supposed to be included in the objective function using Lagrange multipliers. Thus, only one-sided constraints will be mentioned below. These will be referred to as \( C_j \). In fact, concerning one-sided constraints, two cases must be distinguished as iteration proceeds:
If all one-sided constraints are strictly larger than zero (namely, if the iteration is not within any one-sided constraint border), then a BFGS with trust region is applied. Note that line search must be modified to account for the possibility of encountering one-sided constraints as the iteration proceeds, which leads us to the next alternative case.

If any of the one-sided constraints is equal to zero (namely, if the iteration is within some one-sided constraint border), that one-sided constraint is considered as active and the optimization process within the constraint border starts. This means that Lagrange multipliers for the active constraints must be either added or skipped:

- If the Lagrange multipliers associated with one-sided constraints are all positive, then the Karush-Kuhn-Tucker conditions hold and the optimization process continues within all active constraints.
- If some of the Lagrange multipliers associated with one-sided
constraints are negative, the Karush-Kuhn-Tucker additional condition do not hold. In this case, the gradient of the objective function points to the outside of the allowable domain, meaning that the function decreases towards the inside of the allowable domain, and the associated constraint border must be abandoned and declared as non-active. Note that new Lagrange multipliers must be evaluated with the current active constraints in this case.

This process may yield (in an adaptive way) to scenarios in which (i) none of the one-sided constraints is active, (ii) some one-sided constraints are active, or (iii) all one-sided constraints apply.

\[ \phi(\lambda_1, \lambda_2) = \lambda_1^2 + \lambda_2^2 \]

is to be minimized, subject to the constraints

\[
\begin{align*}
C_1(\lambda_1, \lambda_2) &= \lambda_1^2 + 10\lambda_2 - 50 \geq 0, \\
C_2(\lambda_1, \lambda_2) &= 75 + \lambda_1^2 - 10\lambda_1 - 10\lambda_2 \geq 0,
\end{align*}
\]

in which the allowable region is that in between of the two parabolas plotted with dashed lines in Fig.2.5. The whole optimization process summarized in Fig.2.4. The initial guess is \( \lambda = (15, 12) \), which satisfies both constraints \( (C_1 = 295 > 0 \) and \( C_2 = 30 > 0) \), and the stop condition is \( (2.25) \), with \( \varepsilon_c = 10^{-6} \). The optimization path is summarized in Fig.2.5 and described hereafter. Tables 2.1 and 2.2 show the evolution of the design variables along the optimization process. The various points in Fig.2.5 and these tables are:
CHAPTER 2. MATHEMATICAL METHODS

### Table 2.1: Optimization example, critical points

<table>
<thead>
<tr>
<th>Point</th>
<th>( x )</th>
<th>( C_1 )</th>
<th>( C_2 )</th>
<th>( \mu_1 )</th>
<th>( \mu_2 )</th>
<th>( \phi(x, y) )</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>(15.0,12.0)</td>
<td>295.0</td>
<td>30.0</td>
<td>-</td>
<td>-</td>
<td>369.0</td>
<td>38.4</td>
</tr>
<tr>
<td>#2</td>
<td>(11.5,9.2)</td>
<td>172.7</td>
<td>0.0</td>
<td>-</td>
<td>0.42</td>
<td>215.0</td>
<td>29.3</td>
</tr>
<tr>
<td>#3</td>
<td>(8.7,6.3)</td>
<td>88.4</td>
<td>0.0</td>
<td>-</td>
<td>0.0</td>
<td>115.2</td>
<td>21.4</td>
</tr>
<tr>
<td>#4</td>
<td>(4.3,3.2)</td>
<td>0.0</td>
<td>19.0</td>
<td>0.79</td>
<td>-</td>
<td>28.4</td>
<td>10.7</td>
</tr>
<tr>
<td>#5</td>
<td>(0.0,5.0)</td>
<td>0.0</td>
<td>25.0</td>
<td>0.5</td>
<td>-</td>
<td>25.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

### Table 2.2: Optimization example, optimization path

<table>
<thead>
<tr>
<th>Transition</th>
<th>( C_1 )</th>
<th>( C_2 )</th>
<th>( \mu_1 )</th>
<th>( \mu_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1 - #2</td>
<td>&gt;0</td>
<td>&gt;0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>#2 - #3</td>
<td>&gt;0</td>
<td>0</td>
<td>-</td>
<td>&gt;0</td>
</tr>
<tr>
<td>#3 - #4</td>
<td>&gt;0</td>
<td>&gt;0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>#4 - #5</td>
<td>0</td>
<td>&gt;0</td>
<td>&gt;0</td>
<td>-</td>
</tr>
</tbody>
</table>

1. Point #1: the initial guess. Since this satisfies both constraints, the standard BFGS method is applied to begin with.

2. Point #2: first intersection of the optimization path with the second constraint border. From this point on, the restricted SD method along the border of the second constraint is used.

3. Point #3: first point where the Lagrange multiplier associated to second constraint changes sign from positive to negative. This indicates that the border of the second constraint must be left and the process comes back to the interior of the allowable region. From this point on, the standard BFGS method is used.

4. Point #4: first intersection of the optimization path with the first constraint border. From this point on, the restricted SD method along the border of the first constraint is used.

5. Point #5: first point where the stop condition applies. This is a local minimum of the restricted problem.

### 2.6 Genetic algorithm plus gradient-like optimization

Genetic algorithms (GA) and gradient-like (GL) methods have been described in sections 2.4 and 2.5. In principle, both methods are useful in industrial environments, since they can be used subsequently as follows.
2.7. CONCLUDING REMARKS

A GA can be first used imposing a mild convergence requirement, such that it provides only a rough approximation of the global minimum; this will save computational time significantly because GAs exhibit poor convergence properties. Now, if the approximated solution is sufficiently close to the exact solution, application of a local, GL method will provide the solution in a quite computationally efficient way.

But the methods derived above can be combined in a more efficient way, using the gradient-like plus trust regions (GL+TR) method (also developed in section 2.5). This method is somehow a global method, since it provides local minima (namely, suboptimal solutions) without the necessity of providing a good initial guess. In other words, this method can be seen as a dynamical systems in the design parameter space that, given an initial guess, evolves and converges to a suboptimal solution. In other words, the orbits of this dynamical system provides evolutions from initial guesses to suboptimal solutions, one of which is the global minimum of the constrained problem function. On the other hand, we may define the basin of attraction of each suboptimal solution as the set of those initial guesses such that the above mentioned dynamical system converges to that suboptimal solution. The basin of attractions of the suboptimal solutions (in particular, that of the global minimum) are expected to cover a significant part of the design parameter space. Thus, if the above-mentioned dynamical system is combined with a GA, then it would be only necessary that the GA provides a point in the basin of attraction of the global minimum to obtain a good initial iterate to the dynamical system that provides the global minimum.

With these ideas in mind, an algorithm is being prepared that combines the GA and GL+TR methods in such a way that a GL+TR method is used with initial guesses at the elite individuals of each GA generation. Some preliminary calculations have shown that the resulting method is extremely computationally efficient (specially when the number of design parameters is large), as expected from the ideas sketched above.

2.7 Concluding remarks

Two classes of methods have been considered to obtain surrogates of engineering technical disciplines and to optimize an objective function defined with these surrogates. The selected methods for each task have been made attending to both their efficiency in providing good approximations and their computational cost. Based on these, a HOSVD-tensor representation method has been selected to construct the surrogates that is quite appropriate to perform multidimensional interpolation.

Concerning optimization, a genetic algorithm has been selected for global
optimization. And various gradient-like methods have been presented for both unconstrained and constrained optimization. These methods have been combined with a trust region method (as a continuation method) and a flexible Lagrange multipliers method (to deal with constraints). The resulting method is robust and flexible enough to deal with a variety of engineering problems exhibiting a number of one-sided and exact constraints, and is semi-global in the sense that, for reasonable initial iterates converges to a suboptimal solution. As it will be further explained below, identifying suboptimal solutions is quite important in daily engineering applications, well beyond their academic interest. In addition, the semi-global method can be seen as a dynamical system, whose ‘steady’ attractors are the suboptimal solutions of the optimization problem. Thus, these methods can be quite convenient to be combined with genetic algorithms, since these need to provide only one point in the basin of attraction (for the above mentioned dynamical system) of the global minimum. This is enough as a good initial guess to obtain the global minimum itself using the semi-global method.

The mathematical methods described above are prepared for its use in the remaining of the thesis. In particular, the HOSVD+Interpolation developed above will be used in next chapter to develop surrogates of the technical disciplines, and the combination of these with the optimization tools described above will be used to develop two new optimization tools, one basic (which uses a surrogate of the objective function itself) and one modular, which relies on surrogates of the various ingredients of the objective function.
Reduced models and computational fluid dynamics in engineering design

Surrogate models are playing an increasing role in engineering design, especially in the conceptual design phase, in which the need for preliminary estimates and tests is increasing nowadays. Moreover, at present the companies competitiveness has induced the necessity of manufacturing new, efficient, cheaper, and differentiated products. On the other hand, the number of design variables, $\lambda_1, \ldots, \lambda_N$, grows increasingly. The new solutions envisaged by designers introduce new design variables that make the estimation of the optimal and suboptimal solutions increasingly complex. It must be emphasized that suboptimal solutions are very important. This is for two related reasons. On the one hand, they allow for considering some objective, quantitative ingredients of the design process that have not been taken into account in the definition of the objective function that is been minimized. For instance, a typical objective function at the early stages of conceptual design is based on, e.g., geometric complexity, weight, and drag, but does not take into account other important properties (which may bear important economic implications) such as the number of passengers. Similarly, qualitative criteria is also relevant that may be of a subjective or even aesthetic nature. These additional (both quantitative and qualitative) criteria can be evaluated a posteriori for suboptimal solutions.

The design variables are of a very different nature, which means that various technical disciplines are mixed in the design process. In order to find the best combination of the design variables for a specific combination of the technical disciplines, $A_1, \ldots, A_M$, designers often use some kind of multi-disciplinary optimization methodology.

There are various surrogate models that are currently used in industry, which can classified into two large families. Both require (as a first step) calculating (offline) some snapshots, from which some modes are identified. The first family projects the governing equations onto the linear manifold spanned by the modes, and solves the resulting set of (generally nonlinear)
equations. Thus, the online operation of these surrogates involves iterative processes that require a non-negligible (albeit much smaller than its CFD counterpart) computational time. The second family relies on some kind of interpolation, and is quite computationally inexpensive, and amenable to engineering tasks that require computing large a number of cases, such as the global optimization tasks that are required in conceptual design. Thus, the differences between the various surrogate models of this second type stand in the way in which modes are identified and interpolation is performed. The HOSVD+I method developed in last chapter belongs to this second family.

As is usual in daily engineering practice, some kind of compromise is necessary between fidelity and computational cost, which in turn must take into account both the required CPU time and memory. The most efficient surrogates must meet both constraints to an somewhat optimal degree. The surrogates based on HOSVD+I described in last chapter have been selected precisely to improve existing surrogates attending to the two above mentioned requirements. In addition, flexibility and robustness (two additional important issues in engineering practice) have been also accounted for in the selection. In particular, the surrogates can be combined with modeling equations of a quite different fidelity.

With these ideas in mind, this chapter is organized as follows. Sections 3.1 and 3.2 are devoted to the development of two optimization tools based on surrogates. A basic optimization tool is first developed (in section 3.1) that only uses one surrogate for calculating the objective function. This simpler tool can be not convenient when the number of design parameters is large. This is due to the so called curse of dimensionality, after Richard Bellman [7], a difficulty associated with the fact that the number of elements in a multidimensional database increases exponentially with the database dimension, which soon requires huge computational resources (whatever the numerical task is) as the database dimension increases. Thus, if each technical discipline depends on a fewer number of parameters than its total number, treating independently the technical disciplines provides a significant decrease of the computational effort. Based on this, a modular optimization tool is developed in section 3.2 that uses independent surrogates for the various technical disciplines. Each of these optimization tools, both basic and modular can in turn be either global or local, depending on the type of optimization tool, either a genetic algorithm or a gradient-like method, as explained in the last chapter.

The surrogate-based optimization tools require a smaller computational time in their online operation. The computational cost of the offline stage (associated with the snapshots calculation) depends on the computational
3.1. THE BASIC OPTIMIZATION TOOL USING REDUCED MODELS

tool that is used to calculate the snapshots. Thus, CFD methods are re-
viewed in section 3.3, keeping in mind the context. The outer aerodynamic
field could be calculated using CFD tools of various fidelities, depending
on computational resources and the required fidelity. At early stages of the
design process, very low fidelity solvers, such as vortex lattice methods
(VLMs) are most appropriate with standard PCs and small clusters. Unfor-
tunately, these solvers do not provide the outer velocity field, which would
be necessary for a sensible calculation of skin friction. Thus, the remaining
of the chapter is devoted to both standard CFD methods (subsection 3.3.4)
and new surrogate methods (section 3.4) to calculate the skin friction. The
new surrogate models are tested in section 3.5.

Some conclusions and comments/remarks on the results of the chapter
are provided in section 3.6.

3.1 The basic optimization tool using reduced models

Surrogate-based optimization tools try to replace the full computation
of both the objective function and the constraints by surrogate calcula-
tions, intending to diminish the computational time of the optimization
process. As already anticipated, the surrogated models that will be used
in this thesis are based on a combination of high order singular value de-
composition and interpolation (HOSVD+I), as described in last chapter.

For illustration, we consider the conceptual design stage of a wind tur-
bine. After a careful analysis of the possible design configurations, the
process is reduced to minimizing an objective function, which exhibits various
ingredients that depend on some overall properties of the device, such as
those related to the intended average power, the power stability, the manu-
facturing and operational costs, the environmental impact, etcetera. These
involve a large number of design variables, such as the wind turbine
blades number and shape (which in turn involves various free variables),
the tower length and shape, and the structure weight. Some aesthetic vari-
ables such as the color may also be considered, since its has an environ-
mental impact. The design variables can be considered either together or
somewhat grouped in technical disciplines associated with, e.g., the aero-
dynamics, the structures, the electrical generator, etcetera. Note that some
design variables may pertain to various technical disciplines.

In other words, an objective function is identified that either (i) may be
considered as a function of some design variables, $\lambda_1, \ldots, \lambda_N$, or (ii) may
be consider to depend on some technical disciplines, $A_1, \ldots, A_M$, which in
turn depend on subsets of the complete set of design variables.

A basic surrogate optimization tool results from treating the version (i) of
the objective function with the HOSVD+I method. The offline construction of such basic surrogate generally requires discretizing the design parameter space in a structured fashion and computing the objective function in the nodes of the mesh. If the \( N \) design variables are discretized using \( p_1, \ldots, p_N \) values, then the number of required calculations of the objective function is

\[
S = \prod_{j=1}^{N} p_j,
\]

which grows exponentially with \( N \) and can be huge if \( N \) is large. Similarly, the number of online operations roughly scales with the product of the retained number of modes along the directions associated with the various design parameters. Thus, the number of online operations also grows exponentially with \( N \), and can be huge when \( N \) is large. In other words, the computational resources (CPU time and storage capacity) can be huge when the number of design variables is large. This is due to the above mentioned curse of dimensionality, which is alleviated but not solved by the surrogate model in the basic optimization tool. This difficulty is overcome using a modular optimization tool, which considers the version (ii) of the objective function and is considered in next subsection.

### 3.2 Modular optimization tool using reduced models

The curse of dimensionality difficulty can be overcome noting that the various technical disciplines do not generally depend on the whole set of design variables. Instead, each technical discipline usually depends on a fewer number of design variables. Thus, the optimization tool can be organized in a modular fashion, constructing a surrogate model for each technical discipline. In this way, both the offline and online computational costs and the required storage capacity scales exponentially only with the number of design variables involved in the most computationally expensive technical discipline. This technical discipline needs not be that exhibiting the largest number of variables.

Using again for illustration the wind turbine example with a fixed number of blades, assume that the geometry of each blade and the tower are discretized with \( N_B = 8 \) and \( N_T = 2 \) design parameters, respectively. Thus, the total number of design parameters associated with the geometry is

\[
N_g = N_B + N_T + 1 = 6 + 2 = 8.
\]

Similarly, the structure parameters are assumed to be \( N_S = 8 \) and \( N_T = 4 \) for each blade and the tower, respectively, giving a total number of design
parameters for the structure
\[ N_s = N_s^B + N_s^T + 1 = 8 + 4 = 12. \]
The electric generator, on the other hand could require
\[ N_e = 6 \]
design parameters. Finally, the wind turbine operation depends on
\[ N_o = 5 \]
design parameters (namely, the incoming wind intensity and direction, which requires 3 parameters, the angle of attack of the blades, and the charge of the electric generator). On the other hand, the objective function usually has several ingredients (e.g., manufacturing and operational costs, delivered electric power, and some environmental issues such as the aerodynamic noise and the visual impact). The combination of these ingredients involves several free parameters that must be calibrated in the early stages of the conceptual design process. A number
\[ N_i = 3 \]
of ingredients is assumed for simplicity. Summarizing, the total number of design, operation, and free parameters involved in the objective function is
\[ N = N_g + N_s + N_e + N_o + N_1 = 8 + 12 + 6 + 5 + 3 = 34, \]
which, assuming that each parameter is discretized using \( p = 5 \) mesh points, requires a computational effort that scales with
\[ 5^{34} \sim 6 \times 10^{23}. \]
This is too large, by all means. If, instead, the surrogates are constructed for the individual technical disciplines (namely, aerodynamics, structure, and the electric generator), then it is the structures that depends on a largest number of design parameters. But the structures usually involve quite simple computations and could be even calculated directly, without a surrogate, in terms of the aerodynamics loads. The most expensive discipline is by far the aerodynamics (the bottleneck of the computational tasks), which involves the geometric parameters and those associated with the wind turbine operation. Thus, the aerodynamics depend on \( N_g + N_o = 8 + 5 = 13 \) parameters. The associated surrogate requires a computational effort that scales with
\[ 5^{13} = 10^9, \]
which is still quite large, but affordable using either a low fidelity CFD tool or a surrogate.

Still, details of the shapes of the blades and the tower are somewhat decoupled, and the blades aerodynamics could be calculated using a baseline tower, which decreases by two the number of required free parameters, giving a computational effort of

$$5^{11} = 4 \times 10^7,$$

which is fairly reasonable.

On the other hand, note that the HOSVD+I surrogate model is independent of the CFD solver that is used to calculate the aerodynamics, although the nature of the solver may highly affect the required computational resources.

### 3.3 Computational fluid dynamics methods

Various CFD methods are used in this thesis with a different fidelity level, which somehow amounts to a multi-fidelity approach. A vortex lattice method method (VLM) is used as a low fidelity level method (as usually done in usual in conceptual design). In addition, as a higher fidelity CFD method, a panel method (medium fidelity level) is used to test the HOSVD+I surrogate methodology developed in the last chapter, in sections 2.2 and 2.3. Both CFD methods are summarized in sections 3.3.1 and 3.3.2. The panel method is to be combined with a boundary layer analogy to calculate skin friction, which will be the basis of the derivation of various surrogates for the calculation of skin friction, in the next section of this chapter.

#### 3.3.1 Vortex lattice method

The simplest methods for considering three-dimensional effects in CFD are the vortex systems for thin planar wings. As it is pointed in [43], in these methods a finite span planar surface is considered through an undisturbed potential flow. Then, the aerodynamic forces are considered as sources of force on the fluid. In particular, in VLMs, that source of force is introduced considering a singularity distribution based on discrete horseshoe vortices in some quadrilateral panels. Some flow tangency boundary conditions, related with the surface geometry, are applied in some control points to obtain a system of linear equations for the vortices strengths. These vortices define the near flow and allow to obtain some of its characteristics. In particular, the vortex lattice software used in this thesis, AVL (Athena
3.3. COMPUTATIONAL FLUID DYNAMICS METHODS

Vortex Lattice), provides some extrados-intrados coefficient pressure increment distribution which will be used in chapter 4 to estimate the loads in some non-conventional stabilizers of a commercial aircraft. Some aerodynamic global characteristics as stability derivatives are also provided by AVL. This software is widely used in the literature as in [31] and [50] and its results are compared with other CFD methods in [42].

3.3.2 The panel method

The panel method tries to solve the potential flow problem with numerical techniques. This problem only can be solved analytically after doing some geometrical simplifications in the boundary conditions. As explained in [33], the application of numerical techniques avoid the necessity of those geometrical simplifications and thus provides results that are more realistic.

The main advantage of this method compared to those that solve the whole fluid volume is the reduction of the time required to compute an approximate solution. The panel method reduces the problem to calculate the strengths of some distributed singularities on the body’s surfaces while other methods, like finite-differences, solve the flow field in the whole volume. On the other hand, if the problem complexity increases (e.g. transonic and supersonic flows) the panel method accuracy decreases. In this case, highly fidelity methods are needed. However, it is possible to use panel methods in some known cases as supersonic flows making some required corrections to account for compressibility.

In [56] the panel methods are defined as those that solve a linear partial differential equation using an appropriate approximation of the surface by a set of panels where some singularities are defined. The singularity strengths are obtained by imposing some boundary conditions. This strengths described the entire flow near the surfaces.

3.3.2.1 Panels calibration

In order to calibrate the number of panels and validate the results, the flow around some variants of a NACA0012 profile have been computed at Reynolds $3 \cdot 10^6$. All cases listed in table 3.1 are run on straight 3D wings exhibiting a variable number of panels. Then, some 2D sections near the symmetry plane are selected to obtain the drag coefficient.

Considering case #2 as a reference case, various comparisons are performed to study the sensitivity of the method to the number of elements used to define the stabilizer. Pressure drag values (in Newton) are compared in table 3.2. Two different studies are done:
Case Panels per section Number of sections Total panels number Estimated time Wing aspect ratio
#1 50 21 1000 6 s 20:1
#2 100 21 2000 35 s 20:1
#3 200 21 4000 3.5 m 20:1
#4 100 51 5000 6.2 m 20:1

Table 3.1: Definition of test cases

<table>
<thead>
<tr>
<th>α(deg)</th>
<th>Number of sections</th>
<th>Panels per section</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Case #2</td>
<td>Case #4</td>
</tr>
<tr>
<td>2</td>
<td>1.44</td>
<td>1.43</td>
</tr>
<tr>
<td>4</td>
<td>3.44</td>
<td>3.43</td>
</tr>
<tr>
<td>6</td>
<td>6.73</td>
<td>6.70</td>
</tr>
<tr>
<td>8</td>
<td>11.26</td>
<td>11.22</td>
</tr>
</tbody>
</table>

Table 3.2: Pressure drag comparison

- Sensitivity to the number of sections.
- Sensitivity to the number of panels per section.

Regarding table 3.2, cases #2 and #4 are quite similar while cases #1 #2 and #3 are very different, so we can affirm that drag computations are more sensitive to the number of panels per section than to the number of sections. Also, it is clear that relative differences tend to decrease when the number of points per section grows. For example, at α = 6, the pressure drag variation is 27% between cases #1 and #2, while it is 11% between cases #2 and #3. It is expected that if we increase the number of points per section pressure drag tends to a constant while the maximum panel aspect ratio recommended was respected. Then, we assume we can reduce the number of sections (in the span-wise direction) in order to increase the number of panels per section.

Due to the time needed to run case #3 (3.5 minutes), case #2 (35 seconds) is taken as selected case. In section 3.3.4.1, drag coefficient for this case is compared with experimental data.

### 3.3.3 Aerodynamic model: Panel method plus boundary layer formulation

In chapter 4, it is described an optimization methodology applied to an empennage configuration. In this optimization, the drag contribution
to optimization function is estimated with a wetted area method. This method is very common as it allows to estimate the drag value with an acceptable error in many phases of the design process and without the necessity of dealing with computationally expensive CFD methods. However, along this chapter it is questioned the reliability of wetted area method when the study configurations are non-conventional. This is because it is a semi-empirical method, which is based in some known results that have been calibrated for conventional wings only. Thus, there is space to increase the fidelity level of this method. The AVL aerodynamic method used to estimate lift and lateral force in both HTP and VTP are not useful in viscous drag estimation. This is because it does not provide information enough to allow for viscous drag estimation (which could be done using, e.g., using a boundary layer analogy plus the Bernoulli equation if the pressure at both sides of the surface were available). But AVL only provides intrados-extrados pressure coefficients variation, which is useful to estimate the lift and pressure drag, but not viscous drag.

Summarizing, reliable of viscous drag requires some higher fidelity method. Then, friction coefficients in the stabilizers surfaces are to be integrated. Those coefficients can be obtained from different methods, such as those that integrate either the Navier-Stokes or Euler-plus-boundary layer equations in an appropriate volume near the object to be studied. However, the computational time needed to calculate friction coefficients with these methods is large enough to decline its use in any optimization process.

The most feasible solution is using methods that first (i) efficiently provides the outer flow, then (ii) integrate the boundary layer equations along the streamlines to obtain the friction coefficients, and finally (iii) integrate these coefficients on the stabilizers wetted surfaces. This process should improve the results obtained in chapter 4.

In this case, the chosen solution consists in a combination of a panel method to obtain mean velocities near the stabilizers surfaces and a boundary layer equations integration method (Thwaits’ method in laminar boundary layer and Head’s method in turbulent boundary layer).

### 3.3.4 Boundary layer formulation

As anticipated, a medium fidelity model is considered here, namely a combination of a panel method (to calculate the outer potential flow) and a boundary layer analogy [43]. A further simplification is made, since the boundary layer equations are not integrated along the streamlines, but along the free-streamwise sections of the stabilizer. The method proceeds as follows. At each section of the stabilizer, defined as the two-dimensional airfoil profile that was used to generate the stabilizer (see subsection 3.4.1),
a set of empirical boundary layer equations is integrated. These equations depend on whether the boundary layer is laminar (near the leading edge) or turbulent, after the transition point, which is assumed to occur at that position along the span at which

\[ Re_\theta = 1.174(1 + 22.4/Re_\delta)Re_\delta^{0.46} \]  

(3.1)

where the Reynolds number based on the local boundary layer thickness, \( \hat{\theta} \), and the arch-length distance to the leading edge along the two-dimensional profile at the considered section of the stabilizer, \( \hat{s} \), are defined as

\[ Re_\delta = \rho V_e \hat{\theta}/\mu \quad \text{and} \quad Re_\delta = \rho V_e \hat{s}/\mu. \]

Here, \( V_e \) is the local streamwise velocity of the outer flow (as provided by the panel method), and \( \rho \) and \( \mu \) are the air density and viscosity at the flight level (both assumed to be constant along the stabilizer). Note that consistency of the approximation requires that

\[ Re_\delta \sim 1. \]

Now, the local viscous drag is calculated in terms of the local friction coefficient \( c_f \), defined as

\[ c_f = 2\tau_f/(\rho V_e^2), \]

where \( \tau_f \) is the local tangential stress. Thus, the total viscous drag in the stabilizer is obtained integrating the tangential stress along the stabilizer, namely

\[ D_e = \rho \int_{\text{stabilizer}} V_e^2 c_f \, dA, \]

where the local friction coefficient varies along the chord direction as

\[ \frac{d\hat{\theta}}{ds} + \frac{(2 + H)\hat{\theta}}{V_e} \frac{dV_e}{ds} = \frac{c_f}{2}. \]  

(3.2)

Here, \( V_e \) is provided by the panel method, but the local boundary layer thickness, \( \hat{\theta} \), and the empirical scalar \( H \) are unknowns. Eq.(3.2) is an ordinary differential equation that allows for calculating \( \hat{\theta} \) provided that two additional equations are available for the additional unknowns \( c_f \) and \( H \). These equations depend on whether the boundary layer are laminar or turbulent, as follows:

- For laminar boundary layers, following the Thwaites’ method [43], \( c_f \) and \( H \) are calculated in terms of

\[ \lambda = \frac{\rho \hat{\theta}^2}{\mu} \frac{dV_e}{\hat{\theta}}, \]  

(3.3)
as
\[
\frac{\text{Re}_\lambda c_f}{2} = 0.22 + 1.57\lambda - 1.8\lambda^2, \quad H = 2.61 - 3.75\lambda + 5.24\lambda^2
\]
if \( \lambda \geq 0 \), and
\[
\frac{\text{Re}_\lambda c_f}{2} = 0.22 + 1.40\lambda + \frac{0.018\lambda}{\lambda + 0.107}, \quad H = 2.088 + \frac{0.0731}{\lambda + 0.14} \quad (3.4)
\]
if \( \lambda < 0 \). Note that the parameter \( \lambda \) can also be written as
\[
\lambda = \text{Re}_\lambda \left( \frac{\hat{\theta}/V_e}{(dV_e/d\hat{\theta})} \right),
\]
and thus \( |\lambda| \) must be small for the validity of the approximation. A good empirical criterion is \( |\lambda| \leq 0.1 \), which will hold in all calculations below. The following empirical initial condition is used to integrate \((3.2)\)
\[
\hat{\theta} = \sqrt{0.075\mu/(pdV_e/d\hat{s})} \quad \text{at} \quad \hat{s} = 0. \quad (3.5)
\]
• For turbulent boundary layer, \( c_f \) is calculated using the Head’s method [43], as
\[
c_f = 0.246 \cdot 10^{-0.678H} \text{Re}_\lambda^{-0.268}, \quad (3.6)
\]
and eq.\((3.2)\) is coupled with the following empirical equation
\[
\frac{1}{V_e} \frac{d(V_e\hat{\theta}H_1)}{d\hat{s}} = 0.0306(H_1 - 3)^{-0.6169},
\]
where \( H_1 \) is an empirical function of \( H \), defined as
\[
H_1 = 3.3 + 0.8234(H - 1.1)^{-1.287} \quad \text{if} \quad H \leq 1.6,
H_1 = 3.3 + 1.5501(H - 0.6778)^{-3.064} \quad \text{if} \quad H > 1.6. \quad (3.7)
\]
Now, the method described above is applied as follows. At each chordwise section of the stabilizer, the arclength variable \( \hat{s} \) is defined with origin at the leading edge along both the pressure and suction sides, and eq.\((3.2)\) is integrated (in conjunction with \((3.3)-(3.4)\)) with the initial condition \((3.5)\). The integration proceeds until the first value of \( \hat{s} \) such that the transition condition \((3.1)\) holds. From this stage on, eq.\((3.1)\) is integrated in conjunction with \((3.6)-(3.7)\); usually, transition occurs fairly close to the leading edge, namely at a distance of the order of 10% of the chord.

### 3.3.4.1 Drag validation

In order to validate the panel method plus boundary layer integration, two cases have been described in table 3.3. Both cases have been chosen
CHAPTER 3. REDUCED MODELS AND COMPUTATIONAL FLUID DYNAMICS IN ENGINEERING DESIGN

<table>
<thead>
<tr>
<th>Case</th>
<th>Panels per section</th>
<th>Number of sections</th>
<th>Total panels number</th>
<th>Wing aspect ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>21</td>
<td>2000</td>
<td>20:1</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>61</td>
<td>6000</td>
<td>60:1</td>
</tr>
</tbody>
</table>

Table 3.3: Definition of validation cases

<table>
<thead>
<tr>
<th>α (deg)</th>
<th>Drag coefficient</th>
<th>Case #1</th>
<th>Case #2</th>
<th>Experimental data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0065</td>
<td>0.0063</td>
<td>0.0064</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.0071</td>
<td>0.0063</td>
<td>0.0066</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.0088</td>
<td>0.0069</td>
<td>0.0073</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.0115</td>
<td>0.0079</td>
<td>0.0079</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.0154</td>
<td>0.0094</td>
<td>0.0106</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.4: Drag coefficient validation

to have identical elements size, but case #2 aspect ratio is larger to obtain results more similar to the two-dimensional profile values. In table 3.4, cases #1 and #2 drag coefficients are compared with experimental data.

Case #2 results are very similar to those obtained in experimental data while case #1 results are worse than those obtained in case #2 specially when the angle of attack increase. This is due to its larger aspect ratio which closely reproduces the two-dimensional flow conditions. Anyway, as the panels sizes are identical, and the time needed to compute case #2 is too large, case #1 discretization, which is the same that the selected in section 3.3.2.1), is also selected here as its accuracy is acceptable.

3.4 Drag models description

In chapters 4, 5, and 6, a case study to illustrate the methodology described in chapter 2 is described and analyzed. At the end of chapter 5, in section 5.2.4, some guidelines to improve the solution to the proposed problem have been given. One of these guidelines consists in increasing the fidelity of the viscous drag model. One of the lines that is being followed in the field of the development of new conceptual design methodology, is the generation of surrogate models for the viscous drag. The reason is that this technical discipline tends to rely on the use of computationally expensive CFD solver that, by their own nature, are not suited for conceptual design phases. Then, the question to be answered is: is it feasible to integrate a limited amount of information obtained from a higher fidelity model into
3.4. DRAG MODELS DESCRIPTION

A surrogate model that can be used for conceptual design? The key aspects of this question are twofold: a) if a method with a fidelity higher than plain correlations or wetted-area like approaches is to be used, the surrogate model should mark with a limited amount of information only because the time desired for obtain this information must be also limited, and b) the surrogate model accuracy must be reasonable and this is especially important when unconventional, or even radical, new configurations are being sought. Regarding methods for drag prediction, a comprehensive review can be found in the article by van Dam [61]. Also, a more aeronautics industry oriented review has been published by Kulfan [36]. In this last article, the author points out the significant differences that exist between friction drag predictions on the same test problem (the flat plate with fully turbulent flow) obtained when using different CFD solvers and turbulence models. A similar conclusion can be found in the work of Laflin et al [37] although the authors suggest that proper grid resolution may be the single most important factor in achieving accurate CFD drag predictions. Alternative methods to compute drag, based on the far field information of the flow, have been reported by Yamazaki et al [63], and Chao and Van Dam [15]. Finally, and because of its relevance to the present thesis, the work of Gur et al [29] is to be presented. In particular, Gur et al [29] describe a method for the estimation of drag (induced, friction, wave, and interference drag) in full aircraft configurations. This method, that is not based on a CFD analysis, divide any given configuration into two main entity classes: wing like surfaces and bodies. Then, for each of the entities, the different component of drag are computed using low fidelity methods. The reason, as the authors point out, is that drag may be estimated thousands of times during a multidisciplinary design optimization. Interesting’s enough, in their results section, the authors present a rather good agreement between experimental results and the results they provide, although it is to be said that accuracy tends to determinate at the higher Mach numbers and angle of attack.

In chapter 6, some tests to analyze drag fidelity in the optimization results are done. Some different methods of different fidelity levels will be described to illustrate the possibility of increasing the design fidelity level, without increasing the time needed as much as it could be thought.

Five different models are compared hereafter:

1. Wetted area model 3.4.5.1
2. Drag coefficient integration model 3.4.5.2
3. Wetted area corrected model 3.4.6.2
CHAPTER 3. REDUCED MODELS AND COMPUTATIONAL FLUID DYNAMICS IN ENGINEERING DESIGN

4. Drag coefficient corrected integration model 3.4.6.3

5. HOSVD surrogate model 3.4.7

Both wetted area and two-dimensional drag coefficients integration are the simplest models. Neither CFD computations nor experimental data is needed to estimate drag with them. They only depend on the Reynolds number and the wetted area.

Some corrections can be made to both models as drag depends on other parameters as swept or dihedral distributions. To correct both models database #1 (DB #1) is computed. By observing the dependence of the drag with these parameters, some additional ingredients can be added that correct the wetted area and drag coefficients formulations. Some tunable parameters must be determined in these corrections, which are set using a least square adjustment on the DB #1 snapshots. These snapshots could be obtained either from CFD computation, from experiments, or from interpolations on a smaller database.

The fifth method, HOSVD description, allows to reduce the errors committed in the drag estimation. Some databases are needed to define the model. The whole process is described in 3.4.7.

![Figure 3.1: Drag models process](image)

A sketch of the different level models is provided in figure 3.1. Note that the smallest errors are obtained with the HOSVD surrogate model while the needed time to its preprocess increase. Thus, it is the required accuracy and other peculiarities of each specific problem that determines the selected method.
3.4. DRAG MODELS DESCRIPTION

The results of the various methods will be compared in section 3.5 below.

3.4.1 Geometry parametrization

The stabilizer is made up of two pieces, an internal piece near the fuselage whose planform is flat, horizontal, and trapezoidal, and an external piece near the tip that may show some curvature along the span direction. Both pieces are smoothly assembled at their junction. The stabilizer is defined using five parameters, which are related to the lengths of the internal and external pieces, the swept angle, the curvature of the external piece, and the taper ratio. Thus, no torsion is allowed in this simplified description.

Figure 3.2: Sketch of the parametrization used to define the empennage

The stabilizer is defined in a Cartesian coordinate system such that the oxz plane contains the root section and the oxy plane contains the midlines of the internal piece. The stabilizer is generated by moving a two-dimensional airfoil profile in such a way that its aerodynamic center follows a generating line and the profile itself moves in such a way that it remains always perpendicular to the projection of the generating line on the oyz coordinate plane; see Fig. 3.2. The generating line is defined in terms of an instrumental parameter \( t \) that varies in the intervals \( 0 \leq t < t_1 \) and \( t_1 \leq t \leq t_2 \) for the internal and external pieces, respectively, as

\[
\begin{align*}
x &= \lambda_3 t, \\
y &= t & \text{for } 0 \leq t \leq t_2, \\
z &= 0 & \text{if } 0 \leq t < t_1, \\
&= \lambda_4 (t - t_1)^2 & \text{if } t_1 \leq t \leq t_2,
\end{align*}
\]

where \( t_1 \) and \( t_2 \) are to be chosen. But instead of using \( t_1 \) and \( t_2 \) as design parameters, we use the arch lengths of the internal and external pieces of
the generating line, $\lambda_1$ and $\lambda_2$. The latter are related to $t_1$ and $t_2$ as

$$(1 + \lambda_2^2)^{1/2}t_1 = \lambda_1, \quad \int_{t_1}^{t_2} [1 + \lambda_2^2 + 4\lambda_4^2(t - t_1)^2]^{1/2} dt = \lambda_2.$$ 

Note that an arch length parameter $s$ can be defined along the generating line, which is related to the parameter $t$ as

$$s = (1 + \lambda_2^2)^{1/2}t \quad \text{if } 0 \leq t < t_1,$$

$$s = \lambda_1 + \int_{t_1}^{t} [1 + \lambda_2^2 + 4\lambda_4^2(t - t_1)^2]^{1/2} dt \quad \text{if } t_1 \leq t \leq t_2.$$ 

Now, the two-dimensional airfoil profile is homothetic to a NACA0012. Its size varies in such a way that its chord $c$ depends linearly on the archlength along the generating line, as

$$c(s) = c_r [1 + (\lambda_5 - 1)s/(\lambda_1 + \lambda_2)], \quad \text{as } 0 < s < \lambda_1 + \lambda_2,$$

where $c_r$ is the chord at the root section.

The allowed range of variation of the five parameters is as follows

$$0 \text{ m} \leq \lambda_1, \lambda_2 \leq 15 \text{ m}, \quad -1 \leq \lambda_3 \leq 1,$$

$$-1 \text{ m}^{-1} \leq \lambda_4 \leq 1 \text{ m}^{-1}, \quad 0.1 \leq \lambda_5 \leq 1.$$ \hspace{1cm} (3.8)

As defined, these five parameters in these parameter ranges allow for a large variety of very different configurations. Some sample horizontal stabilizer configurations are shown in Fig. 3.3, which correspond to the following parameter values:

Figure 3.3: Sketches of the HTP configurations 1 (top left), 2 (top right), 3 (bottom left), and 4 (bottom right).
3.4. DRAG MODELS DESCRIPTION

- Config. #1: \( \lambda_1 = 15 \, \text{m}; \lambda_2 = 0 \, \text{m}; \lambda_3 = 0; \lambda_4 = 0 \, \text{m}^{-1}; \lambda_5 = 0.5; \)
- Config. #2: \( \lambda_1 = 9 \, \text{m}; \lambda_2 = 8 \, \text{m}; \lambda_3 = -0.8; \lambda_4 = 0.2 \, \text{m}^{-1}; \lambda_5 = 0.5; \)
- Config. #3: \( \lambda_1 = 10 \, \text{m}; \lambda_2 = 7 \, \text{m}; \lambda_3 = 0.6; \lambda_4 = 1 \, \text{m}^{-1}; \lambda_5 = 0.5; \)
- Config. #4: \( \lambda_1 = 10 \, \text{m}; \lambda_2 = 7 \, \text{m}; \lambda_3 = 0.5; \lambda_4 = -0.3 \, \text{m}^{-1}; \lambda_5 = 0.2. \)

For illustration, Fig.3.4 shows the HTP configuration #3 mounted in a fuselage.

![Sketches of the HTP configuration #3 mounted in a fuselage.](image)

Figure 3.4: Sketches of the HTP configuration #3 mounted in a fuselage.

Some assumptions are made to further reduce the computational time.

- No interference between HTPs, the VTP, and fuselage is considered.
- If an stabilizer is rotated around a streamwise axis, then its is not affected. This is so even in connection with the symmetry condition.
- VTP is considered as a particular HTP, with \( \lambda_2 = \lambda_5 = 0 \)

These assumptions are sketched in figures 3.5 and 3.6. Figure 3.5 shows the rotation shift that is needed to eliminate the linear dihedral parameter. Note that in an isolated stabilizer, the \( \theta \)-rotation shown in fig.3.5 does not affect the obtained value of drag. More complex is the influence of symmetry plane (see fig.3.6) because the boundary condition changes under rotation. But since the interactions between fuselage and VTP are being ignored, and the aim of this thesis is not to obtain the actual empennage configurations, but to illustrate a methodology, we can neglect the error that result from this assumption.
3.4.2 Test cases

The calculation of the viscous drag in a stabilizer must be performed in terms of the five design parameters, $\lambda_1, \ldots, \lambda_5$, which can take any value in the allowed range defined in eq. (3.8). The surrogates considered in the remaining of the chapter will provide this multidimensional functional relationship. Viscous drag also depends on flight conditions, which involve various additional parameters associated with, e.g., the angle of attack, the sideslip angle, the Mach number, and the altitude. But the allowed range of the latter parameters need not be covered in calculations. Instead, only a limited number of flight conditions, such as cruise conditions
3.4. DRAG MODELS DESCRIPTION

(high altitude, with zero angle of attack and zero sideslip angle) and take off/landing conditions need to be covered in conceptual design. Thus, the above mentioned surrogates will not be intended to cover all possible flight conditions; instead, in order to somewhat mimic these flight conditions, a surrogate will be constructed for three (somewhat representative) specific set of flight conditions. Thus, that flight conditions are defined as

Flight conditions # 1: \( M = 0.8, h = 10,000 \text{ m}, Re = 3.35 \times 10^7 \), \( (3.9) \)

Flight conditions # 2: \( M = 0.5, h = 1,000 \text{ m}, Re = 5.5 \times 10^7 \), \( (3.10) \)

Flight conditions # 3: \( M = 0.3, h = 1,000 \text{ m}, Re = 3.3 \times 10^7 \). \( (3.11) \)

Also, the number of panels used by the panel method must be selected. To begin with, the following number of panels has been chosen

\[
\# \text{ of panels} = 100 \times 20 = 2,000,
\]

where the two multiplying factors correspond to the vertices along the chord and span directions, respectively. The number of panels has been previously calibrated and tested comparing the results for 2,000 and 3,000 panels in the test points. Results are similar in both cases, so the method is considered converged with 2,000 panels.

3.4.3 Test points

The performance of the methods described above, which are to be developed in next section, will be tested in the 36 test points of the parameter space defined in table 3.5. According to the ranges (3.8), the test points TP1 – TP32 are located near the boundaries of the allowed parameter domain, and thus these points are the most demanding ones in connection with the performance of the method. The remaining four test points (TP33 – TP36) instead, are located well inside the allowed parameter domain, meaning that the method is expected to provide a better approximation at these points. In figure 3.7, the different values of each parameter to construct the test points are presented while in figure 3.8 a three-dimensional scheme of the test points situation within the parametric space is presented.

3.4.4 Errors definition

The performance of the methods developed in this chapter is tested for the test points listed in table 3.5, which are divided into two subsets. For each subset of test points, the difference between the CFD and the surrogate results will be made in terms of three definitions of errors, namely the mean
Table 3.5: Definition of the 36 test points used to study the performance of the surrogate modes

<table>
<thead>
<tr>
<th>TP</th>
<th>λ1</th>
<th>λ2</th>
<th>λ3</th>
<th>λ4</th>
<th>λ5</th>
<th>TP</th>
<th>λ1</th>
<th>λ2</th>
<th>λ3</th>
<th>λ4</th>
<th>λ5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>-0.75</td>
<td>-0.75</td>
<td>0.2125</td>
<td>19</td>
<td>2</td>
<td>13</td>
<td>-0.75</td>
<td>-0.75</td>
<td>0.8875</td>
</tr>
<tr>
<td>2</td>
<td>13</td>
<td>13</td>
<td>0.75</td>
<td>0.75</td>
<td>0.8875</td>
<td>20</td>
<td>13</td>
<td>13</td>
<td>-0.75</td>
<td>-0.75</td>
<td>0.2125</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>2</td>
<td>0.75</td>
<td>0.75</td>
<td>0.8875</td>
<td>21</td>
<td>2</td>
<td>13</td>
<td>-0.75</td>
<td>0.75</td>
<td>0.2125</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
<td>13</td>
<td>-0.75</td>
<td>0.75</td>
<td>0.8875</td>
<td>22</td>
<td>2</td>
<td>2</td>
<td>0.75</td>
<td>-0.75</td>
<td>0.8875</td>
</tr>
<tr>
<td>5</td>
<td>13</td>
<td>13</td>
<td>0.75</td>
<td>0.75</td>
<td>0.2125</td>
<td>23</td>
<td>13</td>
<td>2</td>
<td>0.75</td>
<td>-0.75</td>
<td>0.2125</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>13</td>
<td>0.75</td>
<td>0.75</td>
<td>0.8875</td>
<td>24</td>
<td>2</td>
<td>2</td>
<td>0.75</td>
<td>0.75</td>
<td>0.2125</td>
</tr>
<tr>
<td>7</td>
<td>13</td>
<td>13</td>
<td>0.75</td>
<td>-0.75</td>
<td>0.8875</td>
<td>25</td>
<td>13</td>
<td>2</td>
<td>-0.75</td>
<td>0.75</td>
<td>0.8875</td>
</tr>
<tr>
<td>8</td>
<td>13</td>
<td>2</td>
<td>-0.75</td>
<td>0.75</td>
<td>0.8875</td>
<td>26</td>
<td>2</td>
<td>2</td>
<td>-0.75</td>
<td>0.75</td>
<td>0.8875</td>
</tr>
<tr>
<td>9</td>
<td>13</td>
<td>2</td>
<td>0.75</td>
<td>0.75</td>
<td>0.2125</td>
<td>27</td>
<td>13</td>
<td>2</td>
<td>-0.75</td>
<td>0.75</td>
<td>0.2125</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>2</td>
<td>0.75</td>
<td>0.75</td>
<td>0.8875</td>
<td>28</td>
<td>2</td>
<td>13</td>
<td>-0.75</td>
<td>-0.75</td>
<td>0.2125</td>
</tr>
<tr>
<td>11</td>
<td>13</td>
<td>2</td>
<td>0.75</td>
<td>-0.75</td>
<td>0.8875</td>
<td>29</td>
<td>2</td>
<td>2</td>
<td>0.75</td>
<td>-0.75</td>
<td>0.2125</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>13</td>
<td>-0.75</td>
<td>0.75</td>
<td>0.2125</td>
<td>30</td>
<td>2</td>
<td>2</td>
<td>-0.75</td>
<td>0.75</td>
<td>0.8875</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>13</td>
<td>-0.75</td>
<td>0.75</td>
<td>0.8875</td>
<td>31</td>
<td>13</td>
<td>2</td>
<td>-0.75</td>
<td>0.75</td>
<td>0.2125</td>
</tr>
<tr>
<td>14</td>
<td>13</td>
<td>13</td>
<td>-0.75</td>
<td>-0.75</td>
<td>0.8875</td>
<td>32</td>
<td>2</td>
<td>2</td>
<td>-0.75</td>
<td>0.75</td>
<td>0.2125</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>13</td>
<td>0.75</td>
<td>0.75</td>
<td>0.2125</td>
<td>33</td>
<td>6</td>
<td>6</td>
<td>-0.25</td>
<td>0.25</td>
<td>0.4375</td>
</tr>
<tr>
<td>16</td>
<td>13</td>
<td>13</td>
<td>0.75</td>
<td>-0.75</td>
<td>0.2125</td>
<td>34</td>
<td>9</td>
<td>9</td>
<td>0.25</td>
<td>0.25</td>
<td>0.6625</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>13</td>
<td>0.75</td>
<td>-0.75</td>
<td>0.8875</td>
<td>35</td>
<td>9</td>
<td>6</td>
<td>0.25</td>
<td>-0.25</td>
<td>0.4375</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>13</td>
<td>0.75</td>
<td>-0.75</td>
<td>0.2125</td>
<td>36</td>
<td>6</td>
<td>9</td>
<td>-0.25</td>
<td>0.25</td>
<td>0.6625</td>
</tr>
</tbody>
</table>

Figure 3.7: Test points parameters values

error, ME, the root mean square error, RMSE, and the maximum error, MaxE, defined as

\[
\text{ME} = \frac{1}{M} \sum_{j=1}^{M} LE_j, \quad \text{RMSE} = \sqrt{\frac{1}{M} \sum_{j=1}^{M} LE_j^2}, \quad \text{MaxE} = \max_{j=1,\ldots,M} \{|LE_j|\}
\]
3.4. DRAG MODELS DESCRIPTION

![Figure 3.8: Test points location](image)

where the subindex \( j \) varies along the \( M \) test points in the subset of test points that is being considered, and the local error of the \( j \)-th test point, \( \text{LE}_j \), is defined in terms of the viscous drag values provided by the CFD tool and the surrogate, \( D_{j,\text{CFD}} \) and \( D_{j,\text{surrogate}} \), respectively, as

\[
\text{LE}_j = \frac{|D_{j,\text{CFD}} - D_{j,\text{surrogate}}|}{D_{j,\text{CFD}}}.
\]  

(3.13)

3.4.5 Direct models

Direct models are here considered as those that do not need any kink of previous calculation to estimate drag. Although they are semi-empirical methods, in the case of the it wetted area (section 3.4.5.1), many formulations are largely described in the literature. In the case of the drag coefficient integration, it is just needed to know the two-dimensional drag coefficient of the profile as function of the flight conditions.

3.4.5.1 Wetted area model

Wetted area methodology is the simplest way to estimate viscous drag. Some considerations must be taken into account when using this method in comparison with the panels plus boundary layer integration method described in section 3.3.3:

- No effect of the profile shape is taken into account.
- No swept or dihedral angles are considered.
CHAPTER 3. REDUCED MODELS AND COMPUTATIONAL FLUID DYNAMICS IN ENGINEERING DESIGN

<table>
<thead>
<tr>
<th>Case</th>
<th>Flight cond’ns # 1</th>
<th>Flight cond’ns # 2</th>
<th>Flight cond’ns # 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP1-TP32</td>
<td>ME 6.29%</td>
<td>RMSE 7.36%</td>
<td>MaxE 12.08%</td>
</tr>
<tr>
<td>TP33-TP36</td>
<td>ME 6.14%</td>
<td>RMSE 6.20%</td>
<td>MaxE 7.28%</td>
</tr>
<tr>
<td>Flight cond’ns # 1</td>
<td>RMSE 8.32%</td>
<td>MaxE 9.54%</td>
<td></td>
</tr>
<tr>
<td>Flight cond’ns # 2</td>
<td>RMSE 9.34%</td>
<td>MaxE 10.91%</td>
<td></td>
</tr>
<tr>
<td>Flight cond’ns # 3</td>
<td>RMSE 12.08%</td>
<td>MaxE 16.45%</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.6: Wetted area model errors

- No chord ratio is considered except for the calculation of the mean aerodynamic chord.
- No laminar-turbulent distinction is made.

Viscous drag is estimated as

\[ D = \frac{1}{2} c_f \rho_{\infty} U_{\infty}^2 W_A \]  

(3.14)

where \( \rho_{\infty} \) and \( U_{\infty} \) are the density and velocity at the free stream, \( W_A \) is the wetted area of the stabilizer, and \( c_f \) is the friction coefficient. The latter is estimated as [52]

\[ c_f = 0.455/\left(\log Re_m \right)^{2.588} \]  

(3.15)

where \( Re_m \) is the Reynolds number, based on the mean aerodynamic chord.

The mean and maximum relative errors in the test points defined above for the three cases considered are compared in table 3.6. The relative error is as defined in eq. (3.13).

These errors are compared with those obtained with the remaining methods in section 3.5.

3.4.5.2 Drag coefficient integration

The drag coefficient integration method is based on the assumption that the drag coefficient at each section of the stabilizer can be approximated by the two-dimensional profile coefficient. Therefore, it is sufficient to integrate the two-dimensional coefficient in the mean surface. Two-dimensional drag coefficient (in terms of the Mach and Reynolds numbers) must be provided. The viscous drag is then obtained as

\[ D_{c \sigma} = \frac{1}{2} \rho_{\infty} U_{\infty}^2 \int c_{d \sigma}(M, Re) c(s) \cos(\Lambda) ds \]
3.4. DRAG MODELS DESCRIPTION

### Drag coefficient integration

<table>
<thead>
<tr>
<th>Case</th>
<th>Flight cond'ns # 1</th>
<th>Flight cond'ns # 2</th>
<th>Flight cond'ns # 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ME</td>
<td>RMSE</td>
<td>MaxE</td>
</tr>
<tr>
<td>Flight cond'ns # 1</td>
<td>6.26%</td>
<td>8.21%</td>
<td>15.30%</td>
</tr>
<tr>
<td>Flight cond'ns # 2</td>
<td>5.24%</td>
<td>6.78%</td>
<td>12.61%</td>
</tr>
<tr>
<td>Flight cond'ns # 3</td>
<td>5.22%</td>
<td>7.01%</td>
<td>13.11%</td>
</tr>
</tbody>
</table>

Table 3.7: Drag coefficient integration model errors

where \( c(s) \cos(\Lambda) \, ds = dA \) is the differential of area and \( c_d_0 \) is the two-dimensional coefficient, which is considered to be a function of Mach and the Reynolds numbers. The latter is defined as

\[
Re = \frac{\rho_\infty U_\infty}{\nu} c(s)
\]

where the chord value is function of the section position in the stabilizer.

Some considerations must be taken into account when using this method instead previous methods.

- The profile shape is taken into account in the drag coefficient.
- No swept or dihedral angles dependence is considered.
- The chord ratio is taken into account when calculating the Reynolds number. Thus, it affects the used value of \( c_d_0 \) at each stabilizer section.
- No laminar-turbulent distinction is made.

The relative (mean, mean square, and maximum) errors provided by this method are presented in table 3.7; a further comparison with their counterparts for the remaining methods in this section are will be made in section 3.5.

#### 3.4.6 Corrected models

By corrected models it is meant here those models that are defined correcting the wetted area and drag coefficient integration methods described above. The process used to correct both methods is summarized as

1. A database is constructed using the panels plus boundary layer methodology described in section 3.3.4.
2. The dependence of viscous drag on some some free parameters associated with geometric properties specific of non-conventional configurations is analyzed.
3. Based on the analysis, a new formulation is proposed.

4. The free parameters are adjusted upon least squares fit.

Once the correction is defined, the results obtained using the new formulations are analyzed in the three cases in the three proposed in section 3.4.2.

### 3.4.6.1 Database computation

As anticipated above, a database can be used to correct the formulation described in section 3.4.5.1. Specifically, a database has been constructed using the same flight and geometry parameters used in case #1. The database parameters will be referred to as DB #1 hereafter and are defined in table 3.8.

The database consists in a structured net with five points per parameter, which are represented in figure 3.9. This yields a total number of 3125 snapshots that should be sufficient to adjust the formulation. In case of being necessary, the HOSVD+I method described in section 2.2 can be used to increase the database size without increasing the computational time too much.

<table>
<thead>
<tr>
<th>Database</th>
<th>Mach</th>
<th>Height (m)</th>
<th>Root chord (m)</th>
<th>Snapshots number</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>0.8</td>
<td>10,000</td>
<td>5.0</td>
<td>$5^5 = 3125$</td>
</tr>
</tbody>
</table>

**Table 3.8:** Database #1 parameters

**Figure 3.9:** Squares: Database #1 parameters values, Diamonds: Test points parameters values
3.4. DRAG MODELS DESCRIPTION

3.4.6.2 Corrected wetted area model

The influence of some parameters that are not considered in the formulation used in section 3.4.5.1 is analyzed here. Attending to the studied data, it can be assumed that:

1. Wetted area: it is checked that the drag dependence on the wetted area is approximately linear (as already assumed in the methodology described in section 3.4.5.1). Then, the viscous drag can be estimated as

\[ D = \frac{1}{2} \rho_{\infty} U_{\infty}^2 W A c_f. \] (3.16)

2. Reynolds number: it is also checked that the friction coefficient depends on Reynolds number as \( k_1 \cdot (\log Re)^{k_2} \).

3. Swept and dihedral angles: the friction coefficient is assumed to depend quadratically (namely, through second order polynomials) on both swept and dihedral angles.

4. Chord ratio: the friction coefficient is assumed to depend linearly on the chord ratio.

According to these observations, eq. (3.15) is corrected as

\[ c_f = \left[ \frac{0.455}{(\log Re_m)^{2.588}} \right] (1 + k_1 \lambda) \sum_{i,j=0}^{2} k_{ij} \lambda^i \tan^j (\Lambda) \] (3.17)

where \( k_1 \) and \( k_{ij} \forall i,j=0,1,2 \) are tunable constants (to be adjusted upon least squares fit using DB #1 data. This least squares adjustment consists in minimizing the function \( H \) defined in eq. (3.18). The minimization is performed using the Broiden method described in section 2.5. In that equation, \( D_{ref} \), defined as the mean value of DB #1, is used to non-dimensionalize the function \( H \), as

\[ H = \sum \left( \frac{D_{wac}}{D_{ref}} - \frac{D}{D_{ref}} \right)^2. \] (3.18)

The least squares adjustment yields some approximate values of the constants. Considering these values, which are summarized in table 3.9, the friction coefficient in equation (3.16) can be estimate as

\[ c_f = \left[ \frac{0.455}{(\log Re_m)^{2.588}} \right] (1 - 0.04\lambda)(0.93 + 0.02\lambda^2(1 + 5\tan^2\Lambda)]. \] (3.19)
### Table 3.9: Adjustment constants approximate values

<table>
<thead>
<tr>
<th>Case</th>
<th>TP1-TP32</th>
<th>TP33-TP36</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flight cond’ns # 1</td>
<td>ME 5.74%</td>
<td>RMSE 7.04%</td>
</tr>
<tr>
<td>Flight cond’ns # 2</td>
<td>ME 7.73%</td>
<td>RMSE 8.71%</td>
</tr>
<tr>
<td>Flight cond’ns # 3</td>
<td>ME 6.74%</td>
<td>RMSE 7.58%</td>
</tr>
</tbody>
</table>

### Table 3.10: Wetted area corrected model errors

By using eqs. (3.14) and (3.19), the three cases proposed in section 3.4.2 are computed to analyze the errors, which can be compared with those committed in section 3.4.5.1. Mean and maximum relative errors are presented here in table 3.10 and analyzed in section 3.5. Again, errors are defined as in eq. (3.13).

#### 3.4.6.3 Corrected drag coefficient integration model

In this case, viscous drag is obtained by integrating the drag coefficient (as done in section 3.4.5.2). It is known that the three-dimensional wing drag coefficient is not equal to the two-dimensional profile drag coefficient. This is due to those effects from the three-dimensional geometry. Taking into account the shortcomings of the panel method plus boundary layer integration described in section 3.3.3 and noticing that the method does not consider some of the three-dimensional effects, a new least square adjustment of the DB #1 drag values is performed to obtain a two-dimensional drag coefficient distribution along the generating line defined in chapter 4. The adjustment functions considered in this case are similar to those used in eq. (3.17).

The equation to be adjusted is

\[ D_{cd} = \frac{1}{2} \rho_{\infty} U_{\infty}^2 \int cdc(s) \cos(\Lambda) ds \]

where \( c \cos \Lambda ds = d\Lambda \) and \( cd \) is defined as

\[ cd = cd_0 (1 + k_1\lambda) \sum_{i,j=0}^{2} k_{ij} \left( \frac{d^2z}{dy^2} \right)^i \tan^j(\Lambda). \]
3.4. DRAG MODELS DESCRIPTION

Table 3.11: Adjustment constants approximate values

<table>
<thead>
<tr>
<th>Case</th>
<th>ME</th>
<th>RMSE</th>
<th>MaxE</th>
<th></th>
<th>ME</th>
<th>RMSE</th>
<th>MaxE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flight cond’ns # 1</td>
<td>3.77%</td>
<td>4.91%</td>
<td>9.26%</td>
<td></td>
<td>2.99%</td>
<td>3.03%</td>
<td>3.67%</td>
</tr>
<tr>
<td>Flight cond’ns # 2</td>
<td>3.19%</td>
<td>3.72%</td>
<td>6.59%</td>
<td></td>
<td>0.32%</td>
<td>0.41%</td>
<td>0.58%</td>
</tr>
<tr>
<td>Flight cond’ns # 3</td>
<td>3.18%</td>
<td>3.83%</td>
<td>7.13%</td>
<td></td>
<td>0.86%</td>
<td>0.91%</td>
<td>1.20%</td>
</tr>
</tbody>
</table>

Table 3.12: Drag coefficient integration corrected model errors

The objective function that is to be minimized is

\[ H = \sum \left( \frac{D_{cd}}{D_{ref}} - \frac{D}{D_{ref}} \right)^2. \]

As above, the Broiden method described in 2.5 is again used for such minimization task to obtain the fitted values of the free constants appearing in eq.(3.20). The adjusted values are summarized in table 3.11. Using these constants, the drag coefficient can be approximated as

\[ cd = cd_0 \left\{ 0.98 + 0.01 \left( \frac{d^2z}{dy^2} \right)^2 - 0.02\tan^2 \Lambda \left[ 1 - 4 \left( \frac{d^2z}{dy^2} \right)^2 \right] \right\}. \]

The errors of the resulting approximation are summarized in table 3.12 and compared with previous methods in section 3.5.

3.4.7 HOSVD drag description

The surrogate models considered above to calculate viscous drag are based on standard ideas, similar to the correlation formulae that is already used in conceptual design. Now, we consider a surrogate that is constructed in a different spirit, using the HOSVD+I methodology described in chapter 2, section 2.1. For illustration, the test case to be considered is simplified as follows:

- The number of design parameters is reduced.
- The VTP drag is assumed to be identical than an equivalent HTP drag.
CHAPTER 3. REDUCED MODELS AND COMPUTATIONAL FLUID DYNAMICS IN ENGINEERING DESIGN

- Only one singular HTP is run with symmetry condition in the root section.
- No fuselage contribution to drag is considered.
- No interaction between HTP, the VTP, and the fuselage is taken in account.

The process to construct drag surrogate using HOSVD+I methodology is reduced to the four steps described in chapter 4 which are briefed here:

2. Snapshots calculation using the methodology described in section 3.3.3.
3. HOSVD decomposition and database reduction using section 2.2 methodology (offline stage).
4. Drag estimation using section 2.3 multidimensional interpolation (on-line stage).

3.4.7.1 Database generation

Viscous drag is to be described in terms of the design parameters as

\[ D_f = D_f(\lambda_1, \ldots, \lambda_5). \]  (3.21)

Discretizing the five design parameters in a \( N_1 \times \ldots \times N_5 \)-structured grid, namely considering all combinations of \( \lambda_1, \ldots, \lambda_5 \), such that \( \lambda_k = \lambda_{k1}^{1}, \ldots, \lambda_{kN_k}^{N_k} \), for \( k = 1, \ldots, 5 \), we can construct the following 5-th order tensor from the function (3.21)

\[ A_{i_1 \ldots i_5} = f(\lambda_{1i_1}^{i_1}, \ldots, \lambda_{5i_5}^{i_5}). \]

This tensor builds a 5-dimensional database, whose elements could be obtained using either wind tunnel tests or CFD. As further explained below, this tensor will be obtained using a combination of a panel method and a boundary layer analogy.

Five different databases have been generated. Databases #1, #2 and #3 that correspond with flight conditions #1, #2 and #3 described in section 3.4.2, two additional databases to study the sensitivity of the method with the number of snapshots per parameter are used.

The databases consist in some structured nets with four, five or seven snapshots per parameter which are represented in figure 3.10. All these databases are resumed in table 3.13. Each database is associated to some...
3.4. DRAG MODELS DESCRIPTION

<table>
<thead>
<tr>
<th>Database</th>
<th>Flight conditions</th>
<th>Snapshots number</th>
<th>Time (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>#1</td>
<td>(4^3 = 1024)</td>
<td>9.95</td>
</tr>
<tr>
<td>#2</td>
<td>#2</td>
<td>(4^3 = 1024)</td>
<td>9.95</td>
</tr>
<tr>
<td>#3</td>
<td>#3</td>
<td>(4^3 = 1024)</td>
<td>9.95</td>
</tr>
<tr>
<td>#4</td>
<td>#1</td>
<td>(5^3 = 3125)</td>
<td>30.38</td>
</tr>
<tr>
<td>#5</td>
<td>#1</td>
<td>(7^5 = 16807)</td>
<td>163.40</td>
</tr>
</tbody>
</table>

Table 3.13: Databases description

flight conditions #1, #2, and #3. The required time to construct each database is also detailed in table 3.13.

All databases are analyzed in the same test points that the previous models defined in section 3.4.7.3.

\[\lambda_1\]

\[\lambda_2\]

\[\lambda_3\]

\[\lambda_5\]

\[\lambda_6\]

Figure 3.10: Circles: Databases #1, #2, and #3 parameters values, Squares: Database #4 parameters values, Triangles: Database #5 parameters values and Diamonds: Test points parameters values

3.4.7.2 Database reduction

After some calibration, it has been found that four snapshots per parameter is enough, which gives a total amount of \(4^5 = 1,024\) snapshots. Calculating these and applying HOSVD to the resulting tensor, the singular values along the five dimensions are as plotted in Fig.3.11. Note that the first singular values along the five dimensions coincide and are of the order of
100. Neglecting those modes singular values are smaller than $0.0005 \times 100$ requires retaining $S = s_1 + \ldots + s_5 = 2 + 2 + 2 + 1 + 2 = 9$ modes in the directions associated with the parameters $\lambda_1, \ldots, \lambda_5$, and invoking (2.7) provide an APREB smaller than $0.001$; in fact, the actual APREB is $5.68 \times 10^{-4}$.

![Figure 3.11: HOSVs of the baseline case decomposition as function of the considered mode for each parameter](image)

![Figure 3.12: Thick lines: relative errors of the surrogate as function of the number of retained modes in the baseline case ($M = 0.8$, $h = 10,000$ m, $Re = 3.35 \times 10^7$, 2,000 panels, and 1,024 snapshots). Thin lines: relative errors of the wetted area method.](image)

In order to confirm that using 1,024 snapshots and retaining 9 modes is a good choice, the various errors defined in (3.12)-(3.13) of the surrogate results are calculated in terms of the retained number of modes for the two groups of test points considered in subsection 3.4.3 and the baseline configuration (see Fig.3.12); for reference, the errors of the wetted area method (see eq.(3.14)) are also provided. Note that retaining $S = 9$ modes is a good strategy since increasing $S$ does not produce any benefit. Thus, 9 modes
3.4. DRAG MODELS DESCRIPTION

will be retained in all calculations below. Also:

- The errors in the TP1-TP32 test points are larger than those in the remaining TP33-TP36 test points, as was to be expected since the former set of points are near the boundary of the design parameter domain.

- The mean and root mean square errors remain somewhat close to each other, which means that errors are somewhat uniformly distributed. The maximum error is larger, especially in the first group of test points, due to the fact that some of these points, corresponding to extreme values of the parameter, are less correlated with the snapshots and thus they show a larger error.

- In any event, the errors remain acceptable since the RMS errors are of the order of 1% even in those test points located near the corners of the design parameter domain.

- The method highly improves the wetted area method, since all errors are roughly divided by 5.

Considering 9 modes, the database compression factor defined in eq. (3.22) is 0.051 for DBs #1 to #3, 0.020 for DB #4, and 0.005 for DB #5.

\[ K = \frac{\prod_{i=1}^{5} S_i + \sum_{i=1}^{5} S_i N_i}{\prod_{i=1}^{5} N_i} \]  

(3.22)

As already pointed out in chapter 2, the decomposition allows to obtain any result for intermediate values of the parameters by one-dimensional interpolations while memory needed to store the databases and the time to recover any drag value with an acceptable accuracy decrease with the database reduction.

3.4.7.3 Multidimensional interpolation

Using the methodology described in section 2.2, the values of the database snapshots can be recovered. Committed errors in this process are linked to the APREB value. But the real advantage of this methodology, as it is noted in section 2.3, is the simplification of the multi-dimensional interpolation that allow to easily estimate the results in some intermediate values of the parameters.

The test points defined in table 3.5 are used again to study the method accuracy. In figure 3.12 ME, RMSE, and MaxE when estimating test points drag as function of the retained modes is represented for DB #1.

Now, an analysis follows as how the method works for the other flight conditions described in (3.10) and (3.11).
CHAPTER 3. REDUCED MODELS AND COMPUTATIONAL FLUID DYNAMICS IN ENGINEERING DESIGN

The counterpart of Fig. 3.12 considering only RMSE for these new flight conditions is provided in Fig. 3.13, where flight conditions # 1, given in eq. (3.9) are also consider for reference. Note that:

- The surrogate works generally better as the Mach number decreases, which was to be expected.
- The three flight conditions yield results that are more similar among each other at those test points near the corners of the parameter domain (namely, TP1-TP32). This is due to the fact that the limitation for the surrogates in these points result from geometric complexity, not from aerodynamic complexity.
- Again, the surrogates significantly outperforms the wetted area method.

![Figure 3.13](image)

**Figure 3.13:** Thick lines: RMS error of the surrogate as function of the number of retained modes for the flight conditions 1, 2, and 3, defined in eqs. (3.9), (3.10), and (3.11). Thin lines: counterpart for the wetted area method.

Finally, a comparison for different numbers of snapshots is provided in Fig. 3.14 where, in addition to the case of $4^5 = 1,024$ snapshots, the cases of $5^5 = 3,125$ and $7^5 = 16,807$ snapshots are considered. Note that no

![Figure 3.14](image)

**Figure 3.14:** RMS error as function of the number of retained modes using 1,024, 3,125, and 16,807 snapshots.
significant improvement is obtained from increasing the number of snapshots, which means that 1,024 snapshots already catch well the main physical redundancies of the problem. Increasing the snapshots number catches further redundancies resulting from the rough CFD method itself, which are weaker than the physical redundancies and thus require increasing too much the modes number to obtain significant improvement of the method.

Summarizing the results above, the surrogate constructed with 1,024 snapshots, retaining 9 modes provides reasonably good results, since the RMS errors and maximum errors inside the parameter domain are less than 2%, and those near the corners of the parameter domain are less than 5%. Standard wetted area formulae, instead, yields errors that are five times larger. In section 3.5, errors are compared with their counterparts of the previous models. Concerning the computational cost, the offline CPU time needed to calculate each snapshot is 35 seconds, which gives a total CPU time of 10 hours to calculate the whole set of snapshots. And the online effort to calculate the viscous drag for each configuration using the surrogate is $1.5 \times 10^{-5}$ CPU seconds.

### 3.5 Results and discussion

Different models results are analyzed here. In previous sections, errors for each method have been anticipated. Those values are obtained by studying 36 test points defined in table 3.5.

The test points can be divided into two distinct zones of the parametric space. The test points #1 to #32 are test points near the corners of the parametric space (Zone #1) while the test points #33 to #36 are centered in the parametric space (Zone #2).

By studying the obtained results it is clear that greater errors appear in the extreme values of the parameters where the configurations are less conventional. Swept angles of 45° and large variations in the dihedral angle leading to the appearance of large values of the curvature in the stabilizer surface.

As it was anticipated in figure 3.1 errors diminish by correcting both formulations (wetted area and drag coefficient integration). Note that the minimum errors appear when HOSVD description is used. ME, RMSE, and MaxE for all methods are compared in figures 3.15, 3.16, and 3.17 considering the three flight conditions.

It must be noted that the obtaining results for corrected wetted area model, corrected drag coefficient integration and HOSVD description, are based on some databases obtained with the panels method plus boundary layer integration. Therefore, the validity of such descriptions for estimating
the drag values is conditioned to the validity of panels plus boundary layer integration method. Anyway the objective of this chapter is not to obtain a formulation for estimating the drag, but the application of a methodology that allow to define a description for estimate the drag value from a known
3.6 Concluding remarks

The basic/modular, global/local optimization tools developed in sections 3.1 and 3.2 are generic, in the sense that they are essentially independent of the particular engineering problem that is being treated. Only the number of design parameters and the type of functional relationship (not the relationship itself) between the objective function and its ingredients on the design parameters is necessary; the formulation of the various technical disciplines is also not needed. The particular engineering problem only appears implicitly through the snapshots, which in turn can be calculated using various fidelity solvers. Specific application of the basic and modular tools to the conceptual design of an aircraft empennage will be made below, in chapters 5 and 6.

The various skip-friction-drag surrogates considered in section 3.4 have been tested in section 3.5. The conclusion is that the completely new, HOSVD-based surrogate is the best compromise between precision and computational cost. The HOSVD surrogate is quite robust since it provides consistent results in connection with the flight conditions, the retained number of modes, and the number of snapshots. The combination of this method with the optimization tools will be tested in the specific applications considered below, in chapters 5 and 6.
In order to test the methodology developed in chapters 2 and 3, a test problem is described in the present chapter, namely the problem of optimizing a commercial aircraft empennage according to a prescribed multidisciplinary objective function and a set of restrictions. Appropriate description of the problem requires to define the objective function (section 4.1), the empennage geometry (section 4.2), the various objective function ingredients (section 4.3), including the technical disciplines associated with the latter, and the restrictions (section 4.4).

It is to be anticipated that the test problem definition has been selected to be reasonably realistic. In other words, it is intended to be conceptually similar to those used in the early stages of the conceptual design process. In particular, the objective function is based on geometrical complexity, weight, and drag, and the restrictions include stability, control, and stall restrictions. But, on the other hand, the problem is intended mainly for methodological purposes, and could be seen as a a fairly simplified exercise on conceptual design from the daily engineering practice perspective. In particular, the number of free parameters has been kept small enough, in such a way that the complete set of calculations, including computations of snapshots, can be run in a standard PC. In addition, our selection of hypotheses, ingredients of the objective function, and set of restrictions is rather simplified and debatable; and aero-elastic restrictions are not considered. Thus, we do not claim that the outcomes are configurations to be considered as actual candidates for future aircraft. Instead, what we intend is to illustrate and discuss here the practical implementation of our methodology, its advantages and drawbacks, and its potential as a tool to make quick trade-offs during the early phases of the design cycle.
CHAPTER 4. CASE STUDY: CONCEPTUAL DESIGN OF A COMMERCIAL AIRCRAFT EMPENNAGE

4.1 The objective function

Starting from a given reference empennage configuration to be specified below, the aim is to minimize the following objective function, which contains three terms related to three different criteria, namely the weight of the structure, the viscous drag, and the geometrical complexity (which is somewhat related to cost):

\[ \Phi = \epsilon_1 \frac{\text{weight}}{\text{weight}_{\text{ref}}} + \epsilon_2 \frac{\text{drag}}{\text{drag}_{\text{ref}}} + \epsilon_3 \frac{\text{complexity}}{\text{complexity}_{\text{ref}}} \]  

(4.1)

where the subscript \( \text{ref} \) denotes the reference configuration, and \( \epsilon_1, \epsilon_2, \) and \( \epsilon_3 \) are weight factors that are selected such that \( \epsilon_1 + \epsilon_2 + \epsilon_3 = 1 \). The three ingredients above will be calculated using surrogates, as explained in section 4.3.

In particular, the surrogate for weight (which will be the most computationally expensive ingredient) will be based on various lower order surrogates for the aerodynamic loads, constructed in subsection 4.3.3 from aerodynamics calculations based on AVL. The loads that dimension the structure are computed at the following flight conditions: altitude=1,500 m, and Mach=0.5. The horizontal tail plane (HTP) is dimensioned with the most critical among the following values of the angle of attack and sideslip angle, namely \((\alpha, \beta) = (0^\circ, 10^\circ), (5^\circ, 5^\circ), \) and \( (10^\circ, 0^\circ) \); the vertical tail plane (VTP) is sized at the less favorable conditions among the above mentioned, which is \((\alpha, \beta) = (0^\circ, 10^\circ) \). Viscous drag is calculated with a simple wetted area model at cruise conditions: altitude=15,000 m, Mach=0.8, and \( \alpha = \beta = 0^\circ \). Complexity is evaluated from purely geometrical properties.

A rigorous approach to find the Pareto optimal solutions of the multi-objective optimization process described by equation (4.1) would require the use of a methodology (based on multi-objective genetic algorithms MOGAs) of the type described by Eddy and Lewis [21] and Ferguson et al [22]. For the sake of simplicity, the simpler GA method described in section 2.4 will be used, although a possible future area of research would be to explore the combination of MOGAs and HOSVD.

4.2 Geometry

4.2.1 Design parameters

The fuselage and the wings are kept fixed in the optimization process. Each Horizontal Tail Plane (HTP) consists of an internal trapezoidal piece and an external piece in which a quadratic correction of the dihedral angle
is also allowed; the Vertical Tail Plane (VTP) is trapezoidal. The empennage geometry is defined with the following nine parameters:

1. The lengths of two pieces of the HTP, $\lambda_1, \lambda_2$
2. The HTP swept angle, $\lambda_3$
3. The dihedral angle of the HTP, $\lambda_4$, and a quadratic correction in the external piece of the HTP, $\lambda_5$
4. The taper ratio of the HTP, $\lambda_6$
5. The total height of the VTP, $\lambda_7$
6. The swept angle of the VTP, $\lambda_8$
7. The taper ratio of the VTP, $\lambda_9$

These parameters are allowed to vary in the following ranges:

- $0.0 \leq \lambda_1 \leq 20.0m$, $0.0 \leq \lambda_2 \leq 20.0m$, $-1.0 \leq \lambda_3 \leq 1.0$, $0.0 \leq \lambda_4 \leq 1.0$,
- $-1.0m^{-1} \leq \lambda_5 \leq 1.0m^{-1}$, $0.1 \leq \lambda_6 \leq 1.0$, $0.0 \leq \lambda_7 \leq 20.0$, $-1.0 \leq \lambda_8 \leq 1.0$,
- $0.1 \leq \lambda_9 \leq 1.0$.

With these ingredients, a reduced optimization tool, based on surrogates for the technical disciplines has been constructed with the HOSVD+Interpolation tool described in section 2.2 and section 2.3. To this end, each parameter is discretized with a total number of 6 equispaced values. For each technical discipline, the AVL method is run for all combinations of the resulting discretized values of the parameters, to obtain the snapshots that will be used to calculate the HOSVD step of the HOSVD+Interpolation tool. Note that the maximum number of runs occurs in the HTP, which depends on six free parameters and thus requires $6^6$ runs. The a priori relative error bound (2.7) for truncation of the HOSVD step is set to 1%; in addition, the maximum relative error in reconstructing all snapshots is calculated a posteriori and required to be smaller than 5%.

### 4.2.2 Parametrization of the geometry

#### 4.2.2.1 HTP generation

The HTP consists of two pieces (sketched in Fig.4.1), which are smoothly assembled. The internal piece is flat and trapezoidal but a quadratic correction of the dihedral angle is also included in the external piece. Each HTP is described using a generatrix that provides the centers of the chord wise
sections and is given by

\[ x = \lambda_3 y \quad \text{if} \quad 0 < y \]
\[ z = \lambda_4 y \quad \text{if} \quad 0 < y < \frac{\lambda_1}{\sqrt{1 + \lambda_3^2 + \lambda_4^2}} \]
\[ z = \lambda_4 y + \lambda_5 (y - y_1)^2 \quad \text{if} \quad \frac{\lambda_1}{\sqrt{1 + \lambda_3^2 + \lambda_4^2}} < y \]

in terms of the Cartesian coordinates \( x, y, \) and \( z, \) with origin at the center of the root section, the \( x \) axis along the fuselage axis, the \( z \) axis contained in the plane of the root section pointing upwards, and the \( y \) axis perpendicular to the fuselage symmetry plane pointing to the right. Thus, the generatrix consists of a straight segment whose length is \( \lambda_1, \) followed by a parabolic segment, whose length is \( \lambda_2. \) Imposing the latter requirement requires solving the equation \( s(y) - s(y_1) = \lambda_2, \) where \( y_1 = \frac{\lambda_1}{\sqrt{1 + \lambda_3^2 + \lambda_4^2}} \) and the arch length along the generatrix is defined in terms of the span-wise coordinate as

\[ s = \int_0^y \sqrt{1 + \lambda_3^2 + z'(y)^2} \, dy. \]

Thus, the HTP is defined in terms of six free parameters: \( \lambda_1, \lambda_2, \lambda_3 \) (proportional to the tangent of the swept angle), \( \lambda_4 \) (proportional to the tangent of the dihedral angle), \( \lambda_5 \) (quadratic correction to the dihedral angle), and \( \lambda_6 \) (the taper ratio).

The chordwise section is a NACA0012 airfoil profile, whose center moves along the generatrix in such a way that its mid line is always parallel to the \( x \) axis and the airfoil is always contained in the normal plane of the projection of the generatrix on the \( y - z \) plane (see Fig.4.1). In other words, the angle between the airfoil plane and the \( x - z \) plane is such that

\[ \sin \theta = z'(y)/\sqrt{1 + z'(y)^2}. \]

**Figure 4.1:** HTP generation

The (ruled) surface generated by the mid lines of the 2D airfoils will be called the mid-lines surface below. Both the chord and thickness of the 2D
4.3. OBJECTIVE FUNCTION INGREDIENTS

airfoil vary with the same linear law along the whole HTP, namely
\[
c = c_r + \frac{\lambda_6 - 1}{\lambda_1 + \lambda_2} c_r s(y)
\]
where \(c_r\) is the chord at the root section and \(\lambda_6 > 0\) is the taper ratio.

4.2.2.2 VTP generation

The VTP (see Fig.4.2) is trapezoidal, with a planform homothetic to that of the baseline HTP. The cross section is the same as that of the HTP (NACA0012). Thus the VTP shape depends only on three parameters, namely the height, \(\lambda_7\), the swept, \(\lambda_8\), and the taper ratio, \(\lambda_9\).

4.2.2.3 Some possible geometries

Some examples of the resulting empennage configurations are shown in figure 4.3. Note that the VTP can be present or not.

4.3 Objective function ingredients

Now, the calculation process of the various ingredients of the target function is defined. The construction of appropriate surrogate for loads calculation is also described.

4.3.1 Complexity

Using the parametrization defined above, the geometrical complexity of the empennage is calculated adding the contributions from the HTP and
Figure 4.3: Examples of empennage configurations

the VTP, as

$$C_{\text{EMPENNAGE}} = (1 + b)(2C_{\text{HTP}} + C_{\text{VTP}})$$

where the penalty $b$ is applied when a VTP is present. The complexity of the HTP is estimated as

$$C_{\text{HTP}} = \int_0^{\lambda_1 + \lambda_2} \sqrt{1 + \kappa(s)^m} \, ds$$

(4.2)

where $s$ is the arch length along the generatrix as above and $\kappa$ is the curvature of the generatrix. Since the generatrix of the VTP is a straight line, it
4.3. OBJECTIVE FUNCTION INGREDIENTS

exhibits zero curvature and the complexity of the VTP is defined as

\[
C_{VTP} = \int_0^\lambda \sqrt{1 + \lambda^2} \sqrt{1 + \kappa(s)^m} \, ds = \lambda_r \sqrt{1 + \lambda^2}
\]  

(4.3)

where \( \lambda_r \) is the VTP height and \( \lambda_8 \) is proportional to the tangent of the swept angle.

The exponent \( m \) and the penalty \( b \) are to be calibrated, noting that increasing \( m \) emphasizes the effect of high concentrated values of \( \kappa \), which occur in, e.g., sharp junctions between smooth pieces of the HTP. After some calibration that consisted in a campaign of GA runs (for various values of \( m \) and \( b \)) similar to the campaigns described in section 5.2, the following values have been selected

\[
b = 0.2, \quad m = 2.
\]

Since calculation of the integrals (4.2) and (4.3) is computationally inexpensive, no surrogate model is constructed to calculate complexity.

4.3.2 Drag

The total viscous drag is calculated at cruise conditions (altitude=15,000 m, Mach=0.8) using a flat plate boundary layer analogy, which allows for adding the contributions of the HTPs and the VTP,

\[
D = 2D_{HTP} + D_{VTP}.
\]

The contribution from each HTP (the VTP is treated similarly) is given by

\[
D_{HTP} = \frac{1}{2} \rho_\infty c_f \rho_\infty U_\infty^2 W_A
\]

where \( \rho_\infty \) and \( U_\infty \) are the density and velocity at the free stream, \( W_A \) is the wetted area of the HTP, and \( c_f \) is the friction coefficient. The latter is estimated as [52]

\[
c_f = \frac{0.455}{(\log Re_m)^{2.588}}
\]

where \( Re_m \) is the Reynolds number, based on the HTP mean chord.

As it happened with complexity, these calculations are computationally inexpensive and thus no surrogate model is constructed.
4.3.3 **Weight**

Structure is designed to withstand the loads that occur at Mach=0.5, altitude=1,500 m, and three combinations of the angle of attack and the sideslip angle, namely \((\alpha, \beta) = (0^\circ, 10^\circ), (5^\circ, 5^\circ), \) and \((10^\circ, 0^\circ)\). The steps needed to finally obtain the structure weight are: (a) calculate ultimate loads (which will be assumed the greater of these three combinations, (b) define the structure needed to withstand those loads and (c) calculate structure and not structure weight.

4.3.3.1 **Aerodynamic loads**

AVL provides the local pressure distribution at each chord-wise section of the HTP, \(p(x, s)\) (in terms of the stream-wise coordinate, \(x\), and the arch length along the generatrix, \(s\)), which is assumed as applied at the mid-lines surface. With this, the shear force, bending moment, and torsional moment on the span section \(s = s_0\) are calculated as

\[
\begin{align*}
Q(s_0) &= 1.5 \int_{s_0}^{\lambda_1+\lambda_2} \int_{x_{le}(s)}^{x_{te}(s)} p(x, s) \cos(\theta - \theta_0) \cos \varphi \, dxds \\
M(s_0) &= 1.5 \int_{s_0}^{\lambda_1+\lambda_2} \int_{x_{te}(s)}^{x_{le}(s)} p(x, s)(y - y_0) \cos \theta + p(x, s)(z - z_0) \sin \theta \cos \varphi \, dxds \\
T(s_0) &= 1.5 \int_{s_0}^{\lambda_1+\lambda_2} \int_{x_{te}(s)}^{x_{le}(s)} p(x, s)(x - x_0) \cos(\theta - \theta_0) \cos \varphi \, dxds
\end{align*}
\]

(4.4)

where \(x_{te}\) and \(x_{le}\) are the x-coordinates of the trailing and leading edges of each chord-wise section, respectively, and \(x_0\) is the chordwise coordinate of the mid point of the torsion box; \(\theta - \theta_0\) is the angle between the unit normals to the mid-lines surface at the sections \(s = s_0\) and \(s = s\), which are both contained in the plane \(y - z\) (see Fig.4.1); and \(\cos \varphi \, dxds = dA\) is the differential of area along the mid-lines surface, which is defined in terms of the angle \(\varphi\) between the tangent to the generatrix and the chordwise section (see Fig.4.2). In addition, a safety factor of 1.5 is applied.

Equations (4.4) must be applied to all considered configurations of the HTP, and the three above mentioned combinations of the angle of attack and the sideslip angle. Since the two pieces of the HTP must be considered independently, a total number of six distributions of \(Q\), \(M\), and \(T\) must be considered. These depend on \(s_0\) and the six free parameters of the HTP, as

\[
Q = Q_j(\lambda_1, ..., \lambda_6, s_0), \quad M = M_j(\lambda_1, ..., \lambda_6, s_0), \quad T = T_j(\lambda_1, ..., \lambda_6, s_0)
\]

for \(j=1,...,6\). The HOSVD+I method provides surrogates for these eighteen functions. The VTP is treated similarly but the load distributions are independent of the angle of attack. Thus, only one distribution of \(Q\), \(M\), and \(T\)
4.3. OBJECTIVE FUNCTION INGREDIENTS

results, which depends on the vertical position of the chordwise section $z_0$ and the three free parameters of the VTP, as

$$Q = Q_7(\lambda_7, \lambda_8, \lambda_9, z_0), \quad M = M_7(\lambda_7, \lambda_8, \lambda_9, z_0), \quad T = T_7(\lambda_7, \lambda_8, \lambda_9, z_0).$$

Again, the HOSVD+I methodology provides surrogates for these three functions.

4.3.3.2 Structure design

For the sake of brevity, only the HTP is considered below; the VTP is treated similarly. The main assumption is that aerodynamic loads on the HTP are completely supported by a longitudinal torsion box, which extends span wise along the HTP. The torsion box (Fig. 4.4) is a rectangular box formed by a leading and a trailing spars, which are located at 20% and 55% of the chord, respectively, and the cover joining both spars on their upper and lower edges, to complete the box. The height and width of the box will be denoted as $h$ and $d$, respectively.

![Torsion box sketch](image)

Figure 4.4: Torsion box sketch

Thus, the structural elements in the torsion are the cover and the spars, whose thicknesses are to be determined. Cover thickness includes the actual skin thickness plus the effect of stringers. The equivalent cover thickness, $t_{cov}$, is computed setting up the admissible stress $\sigma_{adm}$ as that of the baseline configuration, which is $\sigma_{adm} = 360$MPa. Assuming that the torsion box is symmetric, the stress in the upper cover can be defined in terms of the moment of inertia of the box and the bending moment calculated above, as

$$\sigma = \frac{M(s)h(s)}{2I_x(s)} \approx \frac{M(s)}{t_{cov}(s)d(s)h(s)}.$$
where $s$ is the arch length along the generatrix of the HTP, as above. This equation allows for calculating the thickness $t_{\text{cov}}$ (recall that both $d$ and $h$ are known, and $M$ is provided by a surrogate, as described in section 4.3.3.1) along the span.

In order to calculate the thickness of the spars, we consider the maximum shear flow, defined as

$$q_{\text{max}}(s) = \left| \frac{T(s)}{2dh(s)} \right| + \left| \frac{Q(s)}{2h(s)} \right|$$

where $d$ is the torsion box width, $T$ is the torsional moment, and $Q$ is the shear force. The latter two are provided by two surrogate models, as described in section 4.3.3.1. Now, a permissible shear stress is defined as that of the baseline configuration, which is $\tau_{\text{adm}}$. The shear stress is defined as the product of the spar thickness and the shear flow. The resulting spar thickness is

$$t_{\text{sp}}(s) = \frac{q_{\text{max}}}{\tau_{\text{adm}}}$$

where $q_{\text{max}}$ is as calculated above. Finally, no matter what the outcome of the computation above is, the minimum thickness considered in every component is 2.5mm.

### 4.3.3.3 Weight computation

Once the thickness of the structure has been calculated, the weight of each HTP is calculated as

$$W = 2 \int_{STB} \rho_{\text{cov}}d(s)\frac{t_{\text{cov}}(s)}{c(s)} + \rho_{\text{sp}}h(s)t_{\text{sp}}(s)\frac{dS}{c(s)} + \rho_S \left( A_S - \int_{STB} \frac{d(s)}{c(s)}\frac{dA}{dA} \right)$$

where the first integral yields both the weight of the equivalent upper and lower panels of the box and the spars weight, and the third term accounts for the remaining of the stabilizer. In these integrals, $STB$ is that part of the mean-lines surface of the HTP occupied to the torsion box, $A_S$ is the total area of the mean-lines surface, and $dA = \sin\varphi dx ds$ is the differential of area along the mid-lines surface, as above; $\rho_{\text{cov}}$ and $\rho_{\text{sp}}$ are the skin and spars material density, respectively, and $\rho_S$ is the density per unit surface of the stabilizer part that is not torsion box, defined in such a way that it accounts for the weight of other components and mechanical parts.

### 4.4 Aerodynamics restrictions

The search for the optimum configuration is subject to the following restrictions, which are calculated at altitude=1,500 m, and Mach=0.2:
4.4. AERODYNAMICS RESTRICTIONS

1. The conveniently scaled (see section 4.4.1) stability derivatives of the empennage with respect to the angle of attack and the sideslip angle, $C_{m\alpha}$ and $C_{n\beta}$, are not allowed to worsen their counterparts in the reference configuration.

2. Similarly, the horizontal and vertical control derivatives with respect to the horizontal and vertical control surfaces deflection angle (both denoted as $\delta$), $C_{m\delta}$ and $C_{n\delta}$, are required to be at least a 90% of their counterparts in the reference configuration.

3. Stall is assumed to occur, and the individual is discarded, if either the maximum or minimum (along the span) values of the chordwise lift coefficient reach some threshold values at maximum and minimum fixed angles of attack.

Surrogates for the stability and control derivatives, and the maximum and minimum span-wise values of the lift coefficient are constructed in section 4.4.1 and section 4.4.2. The approach that has been actually followed has been to discard those individuals that do not meet either of the restrictions R1, R2, or R3. A possible reasonable alternative could have been to retain these individuals in the GA population and apply a penalty function to their fitness value modifying eq. (4.1). The reason to choose the former criterion instead of the latter is that civil aircraft design and manufacturing tends to be a rather conservative activity and, normally, new concepts are accepted only if they improve previous ones both globally and, to some extent, locally. In this sense, aircraft handling and controllability are key technical issues and we decided to treat them as a series of clean cut restrictions.

4.4.1 Stability and control derivatives

Calculations of the stability and control derivatives are made running the AVL method at zero angle of attack and zero sideslip angle, altitude=1,500 m, and Mach=0.2. The derivatives of the vertical and lateral moments coefficients with respect to the angle of attack $\alpha$ and the sideslip angle $\beta$, $C_{m\alpha}$ and $C_{n\beta}$ provided by the AVL method are made dimensionless with the dynamic pressure at the free stream, a wing reference area, and the mean chord of the wing. Thus, the contributions of the HTP and VTP are re-scaled in the same way and they can be added to obtain the stability derivatives of the empennage. The contributions of the HTP and VTP depend on the six free parameters of the HTP, namely

$$C_{m\alpha}^{\text{HTP}} = F_1(\lambda_1, ..., \lambda_6), \quad C_{n\beta}^{\text{HTP}} = F_2(\lambda_1, ..., \lambda_6) \quad (4.5)$$
CHAPTER 4. CASE STUDY: CONCEPTUAL DESIGN OF A COMMERCIAL AIRCRAFT EMPENNAGE

Since the VTP is exactly vertical, it produces no vertical force. Its contribution to the lateral force stability derivative depends on the three VTP free parameters, namely

$$C_{n\beta}^{VTP} = F_3(\lambda_7, \lambda_8, \lambda_9) \quad (4.6)$$

The control derivatives are made dimensionless similarly. Again, the contributions of the HTP/VTP to the control derivatives depend on associated free parameters, namely

$$C_{m\delta}^{HTP} = F_4(\lambda_1, ..., \lambda_6), \quad C_{n\delta}^{HTP} = F_5(\lambda_1, ..., \lambda_6) \quad (4.7)$$

$$C_{n\delta}^{VTP} = F_6(\lambda_7, \lambda_8, \lambda_9) \quad (4.8)$$

The functions $F_1, ..., F_6$ defined in (4.5)-(4.8) are treated using as surrogate model that resulting from applying HOSVD+I to this function.

4.4.2 Stall restrictions

These restrictions are imposed at altitude=1,500 m, Mach =0.2, and $\beta = 0^\circ$, and require to calculate (using AVL) the span-wise maximum (at $25^\circ$) and minimum (at $-25^\circ$) values of the chordwise lift coefficients on the various cross sections, $C_{l_{max}}$ and $C_{l_{min}}$. These depend on the six free parameters that define the HTP geometry, namely

$$C_{l_{max}} = F_7(\lambda_1, ..., \lambda_6), \quad C_{l_{min}} = F_8(\lambda_1, ..., \lambda_6)$$

Again, these two functions are treated using as surrogate model that resulting from applying HOSVD+I to these functions.

4.5 Reference configuration

The reference configuration (see the sketch in Fig.4.5) is a highly simplified aircraft configuration characterized by the following parameters:

1. Fuselage (with a circular cross-section): maximum diameter= 8 m and length=65 m.

2. Wing: profile= NACA4412, swept angle=40$^\circ$, dihedral angle=5$^\circ$, taper ratio=0.2, and semi-span=30 m.

3. Horizontal tail plane (HTP): profile=NACA0012, root chord=10 meter, swept angle=0$^\circ$, dihedral angle=0$^\circ$, taper ratio=0.5, and semi-span=15 m.
4.5 REFERENCE CONFIGURATION

Figure 4.5: Reference configuration

<table>
<thead>
<tr>
<th>HTP parameters</th>
<th>VTP parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \lambda_1 )</td>
<td>( \lambda_7 )</td>
</tr>
<tr>
<td>( \lambda_2 )</td>
<td>( \lambda_8 )</td>
</tr>
<tr>
<td>( \lambda_3 )</td>
<td>( \lambda_9 )</td>
</tr>
<tr>
<td>( \lambda_4 )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Restriction</th>
<th>Reference</th>
<th>Optimization function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{m_a} )</td>
<td>-2.80</td>
<td>Weight</td>
<td>11,124</td>
</tr>
<tr>
<td>( C_{n_{\beta}} )</td>
<td>0.13</td>
<td>Drag</td>
<td>19,119</td>
</tr>
<tr>
<td>( C_{m_{\alpha}} )</td>
<td>-1.24</td>
<td>Complexity</td>
<td>57.1</td>
</tr>
<tr>
<td>( C_{n_{\alpha}} )</td>
<td>0.071</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( C_{l_{\text{max}}} )</td>
<td>1.57</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( C_{l_{\text{min}}} )</td>
<td>-1.57</td>
<td>Total fitness</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 4.1: Reference configuration values.

4. Vertical tail plane (VTP): profile=NACA0012, root chord=10 m, swept angle=30°, taper ratio=0.5, and semi-span=15 m.

Reference configuration parameters and values for optimization ingredients and restrictions are shown in Table 4.1.
Global Optimization

The methodology presented in chapters 2 and 3 (in particular, the global optimization tool) is now illustrated with the specific case study described in chapter 4, namely the problem of optimizing a commercial aircraft empennage according to a prescribed multidisciplinary objective function and a set of restrictions.

For the sake of brevity, only the modular optimization tool will be applied in detail, since it is very easily anticipated that the basic tool would be much more computationally expensive. This is because the total number of design parameters is 9 and the number of free tunable parameters in the objective function (namely, two of the three weights associated with the three ingredients) is 2; the latter must also be left free in principle, since they must be tuned as a part of the calibration of the tool itself. This means that if the basic optimization tool were used, the number of required snapshots (each of them to be calculated using all technical disciplines, including aerodynamics) would be $N_9 \cdot N_2$. Now, if each design parameter were discretized using $N_1$ values and the weights are discretized with $N_2$ values. And similarly, the HOSVD reconstruction of each specific case would require a number of algebraic operations that also scales with $N_1^9 \cdot N_2^2$. If, e.g., $N_1 = 6$ and $N_2 = 3$, then the computational effort (and the required CPU memory) would be proportional to $6^9 \times 3^2 = 90,699,264$; in the particular application developed in this chapter, this huge computational effort would offset the advantage of using surrogates instead of the standard CFD tools. In the modular tool instead, the worst technical discipline (namely, the aerodynamics of the HTP) exhibits 7 parameters, and the associated computational effort scales with $6^7 = 279,936$, yielding an acceleration factor of $6^2 \times 3^2 = 324!$. Thus, the advantage of the modular tool is clear. It is to be noted that in a more realistic situation, the difference between the total number of free parameters and those involved in the worst technical discipline will grow, making the advantage of the modular tool even more clear. An additional improvement of the offline computational effort would result if a sampling method were used, but this improvement is well
outside the scope of this thesis.

Application of the global optimization tool first requires to calibrate the genetic algorithm parameters, which will be made in section 5.1. The remaining of the chapter is devoted to performing two optimization campaigns to first fine tune some parameters of the method (in subsection 5.2.1), and then obtain some optimal and suboptimal solutions (in subsection 5.2.2). Comparison with the reference configuration is made in subsection 5.2.3 and some possible further improvements suggested this particular application of the method are commented in subsection 5.2.3. The chapter ends with some preliminary conclusions, in subsection 5.3.

5.1 Genetic algorithm parameters

The optimization platform selected is a genetic algorithm like the one described in section 2.4. In this section, some GA parameters are to be calibrated. In this case the values of the different genetic algorithm parameters are:

- The number of individuals per generation is $N_I = 300$
- The baseline configuration (seed) percentage is $\alpha_1 = 10\%$
- The elite percentage is $\alpha_E = 10\%$
- The crossover probability is $\alpha_C = 90\%$
- The mutation probability is $\alpha_M = 4\%$
- The total number of generations is $N_G = 500$

The optimization method is a genetic algorithm, which is chosen to be both simple and robust: 300 individuals per generation are considered, and a total number of 500 generations are run unless convergence occurs sooner. The crossover and mutation probabilities at bit level have been set to 90% and 4%, respectively. Additionally, an elite sub-population of 10% of the total population is made to pass unchanged to the next generation. Additional information regarding the details of the genetic algorithm can be found in the article by Alonso et al [3]. Nevertheless, this formulation can be certainly improved, most probably combined with a gradient-like method such as steepest descent, so as to generate a hybrid optimization method that runs significantly faster than the one used in this chapter. A first approach to this hybrid optimization is pointed in section 2.6 and analyzed in chapter 6.
5.2 Results and discussion

The performance of the reduced optimization tools is now analyzed. These have been run a total number of 36 times. Ten of them have been used to calibrate the complexity function, as explained in the section 4.2. The remaining 26 runs have been performed in two campaigns, considering seven and twelve combinations of the three weights appearing in the objective function (4.1), which are considered in section 5.2.1 and section 5.2.2 below.

5.2.1 First optimization campaign

Two sets of seven runs of the optimization tool have been performed for the seven combinations of the weights in the objective function (4.1) indicated in the table 5.1. The GA is run in a standard way for these combinations in the first seven runs. The resulting optimized individuals of the seven combinations are added as seeds in the subsequent seven combinations to ensure that new regions of the parameter space are considered in the GA runs. The values of the weight, drag, and complexity of the optimized individuals in the second run are also provided in table 5.1, and illustrated in the diagram of Fig.5.1, where their counterpart for the baseline configuration is also provided for reference. Note that:

1. The baseline drag and complexity is always improved, even in those combinations that do not take these into account. Weight is also improved except in combinations #4 and #6.

2. Weight and viscous drag show opposite trends, as do drag and complexity. This can be explained noting that increasing the span of the HTP decreases the wetted area and thus viscous drag, but increases both the bending moment and the length of the generatrix, which in turn increases both weight and complexity.

3. Weight and complexity instead show the same trend, which is further appreciated in the fact that combinations #3 and #5 provide almost the same result.

4. Combinations #1-5 and #7 produce qualitatively similar optimized configurations, which are always V-shaped. They differ only in the quantitative values of the various parameters of the HTP, such as the swept and dihedral angles. The VTP is absent in all combinations. Thus, lateral stability and control are more efficiently accounted for
Table 5.1: The seven combinations considered in the first optimization campaign (columns 2-4) and the associated values of the weight, drag, complexity, and fitness of the optimized configurations (columns 5-7); the baseline weight/drag/complexity is also given for reference.

<table>
<thead>
<tr>
<th>Comb. #</th>
<th>$\epsilon_1$</th>
<th>$\epsilon_2$</th>
<th>$\epsilon_3$</th>
<th>Weight (kg)</th>
<th>Drag (N)</th>
<th>Complexity</th>
<th>Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/3</td>
<td>1/3</td>
<td>1/3</td>
<td>9,759</td>
<td>18,070</td>
<td>39.6</td>
<td>0.84</td>
</tr>
<tr>
<td>2</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>10,836</td>
<td>15,734</td>
<td>45.2</td>
<td>0.90</td>
</tr>
<tr>
<td>3</td>
<td>1/2</td>
<td>0</td>
<td>1/2</td>
<td>9,521</td>
<td>18,964</td>
<td>38.6</td>
<td>0.76</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1/2</td>
<td>1/2</td>
<td>14,651</td>
<td>15,636</td>
<td>44.4</td>
<td>0.80</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>9,511</td>
<td>19,025</td>
<td>38.7</td>
<td>0.85</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>12,205</td>
<td>15,016</td>
<td>49.7</td>
<td>0.79</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>9,641</td>
<td>20,614</td>
<td>38.2</td>
<td>0.67</td>
</tr>
<tr>
<td>Reference</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>11,124</td>
<td>19,129</td>
<td>57.1</td>
<td>-</td>
</tr>
</tbody>
</table>

through the dihedral angle of the HTP than with the additional inclusion of a VTP, which penalizes the three ingredients in the objective function.

5. Combination #6 promotes a quadratic correction of the dihedral angle, which decreases viscous drag (subject to the imposed restrictions), the only objective function ingredient that is present in this combination. Such quadratic complexity penalizes both weight and complexity, and thus is absent in the optimized individuals resulting from the remaining combinations.

Since the results in this section are discussed as a matter of illustration (not intending to provide actual empennage configurations), only configuration #1 will be considered below in some detail; the main difference with configuration #6 will also be illustrated. Both configurations #1 and #6 are illustrated in Fig.5.2

Table 5.2 shows the parameter values of the optimized individual in combination #1; note that the GA provides values for the HTP and the VTP parameters, but the height of the VTP is $\lambda_7 = 0$, meaning that the VTP is in fact absent. The actual values of the restrictions are also indicated for both the baseline and the optimized configurations, as well as the optimized values of the weight, drag, and complexity, and the fitness. Note that the optimized individual improves the baseline in the three ingredients of the objective function. The generatrices of the HTP and VTP in the empennage along the optimization process in combination #1 and #6 are sketched in Fig.5.3, where the various generatrices are plotted. For reference, the baseline configuration is plotted with dashed lines. It is to be noted that the initial approach to the optimized configuration is quite fast. For instance,
5.2. RESULTS AND DISCUSSION

Figure 5.1: Weight/drag/complexity for the optimized individuals in the first optimization campaign.

Figure 5.2: Configurations #1 and #6.

the vertical stabilizer has been eliminated in the ninth generation of the GA.

A further illustration of the optimization process in combination #1 is provided in Fig.5.4, where the evolution of the taper ratio of the HTP and the fitness (global, as well as fitness for the three ingredients of the objective function) are provided. Note that optimization is essentially completed at generation 50th.

5.2.2 Second optimization campaign

Now, the optimization tool is run for the twelve combinations presented in table 5.3. As in the first campaign, the three ingredients of the objective function are improved in the optimized configurations, except for weight in combinations #2 and #4, which illustrates the consistency of the results.
CHAPTER 5. GLOBAL OPTIMIZATION

<table>
<thead>
<tr>
<th>HTP parameters</th>
<th>VTP parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \lambda_1 )</td>
<td>( \lambda_7 )</td>
</tr>
<tr>
<td>19.66</td>
<td>0.0</td>
</tr>
<tr>
<td>0.15</td>
<td>-0.3701</td>
</tr>
<tr>
<td>0.2598</td>
<td>0.3055</td>
</tr>
<tr>
<td>0.7559</td>
<td>0.3055</td>
</tr>
<tr>
<td>0.0236</td>
<td>0.3055</td>
</tr>
<tr>
<td>0.6669</td>
<td>0.3055</td>
</tr>
</tbody>
</table>

Restriction Baseline Optimized configuration

| Cm,α  | -2.80 | -2.81 |
| Cn,β  | 0.13  | 0.13  |
| Cm,δ  | -1.24 | -1.63 |
| Cn,δ  | 0.071 | 0.133 |
| Cl,max| 1.57  | 1.51  |
| Cl,min| -1.57 | -1.24 |

Optimization function Baseline Optimized configuration Fitness

| Weight | 11,124 | 9,759 | 0.88 |
| Drag   | 19,119 | 18,070| 0.94 |
| Complexity | 57.0 | 39.6 | 0.69 |
| Total fitness | | 1.00 | 0.84 |

Table 5.2: The optimized configuration in combination #1; the baseline configuration is also considered for reference.

Figure 5.3: Top and back views of the evolution of the generatrices along the optimization run for combinations #1 (left) and #6 (right). The reference configuration is plotted with dashed lines. Distances are in meters.

For the sake of brevity, the remaining results are omitted; we just point out that the trends in the first campaign are confirmed here. In particular, all optimized configurations are V-shaped, either exhibiting only the internal flat piece or with a short external curved piece, but with a quite small quadratic correction, which gives a maximum correction of 1 cm in
5.2. RESULTS AND DISCUSSION

Figure 5.4: a) HTP chord ratio evolution along the optimization with combination #1. b) Evolution of the optimization function.

the vertical deflection of the HTP.

5.2.3 Reduced optimization tool vs a conventional optimization tool

Results obtained with the reduced optimization tool in the first optimization campaign, combination #1, are compared in table 5.4 with its counterparts obtained with a conventional optimization tool (not using the surrogates for the various technical disciplines), based on the same genetic algorithm. As it can be seen, the relative error is always below 3%. Results for the remaining combinations in the first campaign and for all combinations in the second campaign are completely similar, except for the restriction on lateral stability, which shows larger errors (up to 7%) in some cases. The larger relative errors are associated with the fact that this restriction shows quite small values, and can be even due to round off errors in the AVL results. In any event, those errors could be decreased by both retaining more modes in the HOSVD (which also requires calculating a larger number of snapshots) and improving the precision of the AVL method (using a finer grid). This is out of the scope of the method performances illustration,
### Table 5.3: Counterpart of table 1 for the twelve combinations in the second optimization campaign.

<table>
<thead>
<tr>
<th>Combination</th>
<th>$\epsilon_1$</th>
<th>$\epsilon_2$</th>
<th>$\epsilon_3$</th>
<th>Weight (kg)</th>
<th>Drag (N)</th>
<th>Complexity</th>
<th>Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comb. #1</td>
<td>0</td>
<td>1/3</td>
<td>2/3</td>
<td>10,510</td>
<td>17,937</td>
<td>39.5</td>
<td>0.77</td>
</tr>
<tr>
<td>Comb. #2</td>
<td>0</td>
<td>2/3</td>
<td>1/3</td>
<td>15,181</td>
<td>15,223</td>
<td>46.2</td>
<td>0.80</td>
</tr>
<tr>
<td>Comb. #3</td>
<td>1/3</td>
<td>0</td>
<td>2/3</td>
<td>9,520</td>
<td>18,963</td>
<td>38.6</td>
<td>0.73</td>
</tr>
<tr>
<td>Comb. #4</td>
<td>1/3</td>
<td>2/3</td>
<td>0</td>
<td>11,158</td>
<td>15,426</td>
<td>46.7</td>
<td>0.87</td>
</tr>
<tr>
<td>Comb. #5</td>
<td>2/3</td>
<td>1/3</td>
<td>0</td>
<td>9,713</td>
<td>18,064</td>
<td>39.6</td>
<td>0.89</td>
</tr>
<tr>
<td>Comb. #6</td>
<td>2/3</td>
<td>0</td>
<td>1/3</td>
<td>9,520</td>
<td>18,963</td>
<td>38.6</td>
<td>0.79</td>
</tr>
<tr>
<td>Comb. #7</td>
<td>1/6</td>
<td>1/6</td>
<td>2/3</td>
<td>9,520</td>
<td>18,963</td>
<td>38.6</td>
<td>0.76</td>
</tr>
<tr>
<td>Comb. #8</td>
<td>1/6</td>
<td>2/3</td>
<td>1/6</td>
<td>11,149</td>
<td>15,436</td>
<td>46.7</td>
<td>0.84</td>
</tr>
<tr>
<td>Comb. #9</td>
<td>2/3</td>
<td>1/6</td>
<td>1/6</td>
<td>9,520</td>
<td>18,963</td>
<td>38.6</td>
<td>0.85</td>
</tr>
<tr>
<td>Comb. #10</td>
<td>1/4</td>
<td>1/4</td>
<td>1/2</td>
<td>9,520</td>
<td>18,963</td>
<td>38.6</td>
<td>0.80</td>
</tr>
<tr>
<td>Comb. #11</td>
<td>1/4</td>
<td>1/2</td>
<td>1/4</td>
<td>10,376</td>
<td>16,377</td>
<td>42.7</td>
<td>0.85</td>
</tr>
<tr>
<td>Comb. #12</td>
<td>1/2</td>
<td>1/4</td>
<td>1/4</td>
<td>9,520</td>
<td>18,963</td>
<td>38.6</td>
<td>0.84</td>
</tr>
<tr>
<td>Reference</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>11,124</td>
<td>19,129</td>
<td>57.1</td>
<td>-</td>
</tr>
</tbody>
</table>

Concerning reduction of the required computational time, which is what has been pursued with the reduced optimization tool, calculation of the properties of each configuration requires 0.0037 CPU seconds with the reduced tool, and 9.7 CPU seconds with the conventional tool that uses AVL to calculate the aerodynamics of each individual. Thus, if the GA optimization algorithm involves 300 individuals per generation and 500 generations, each optimization run requires 1.56 CPU hours using the reduced tool and 405.4 CPU hours using the conventional optimization tool, which means that the computational effort has is divided by a factor of 260 if only the optimization runs are considered. The reduced tool requires the previous AVL-calculation of the snapshots that are needed to generate the surrogates, which involve the following computational effort:

1. Stability and control: 17 CPU hours.
2. Aerodynamics loads: 52.7 CPU hours.
3. Stall calculations: 52.7 CPU hours.

If these 122.5 CPU hours are added to the optimization computational effort, the resulting CPU time is 124.1 hours, which means that the CPU time required by the conventional optimization tool to perform one optimization run (405.4 hours) has been decreased by a factor of 3.6 in just the first optimization run. But, as it happened in the illustration above, conceptual optimization usually requires a large number of additional optimization runs associated with both calibration and reformulation of the
5.2. RESULTS AND DISCUSSION

<table>
<thead>
<tr>
<th>Restriction</th>
<th>Reduced tool</th>
<th>Full tool</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cm_α</td>
<td>-2.81</td>
<td>-2.84</td>
<td>1.1%</td>
</tr>
<tr>
<td>Cn_β</td>
<td>0.13</td>
<td>0.13</td>
<td>0.0%</td>
</tr>
<tr>
<td>Cm_δ</td>
<td>-1.63</td>
<td>-1.68</td>
<td>3.1%</td>
</tr>
<tr>
<td>Cn_δ</td>
<td>0.133</td>
<td>0.135</td>
<td>1.5%</td>
</tr>
<tr>
<td>Cl_{max}</td>
<td>1.51</td>
<td>1.55</td>
<td>2.6%</td>
</tr>
<tr>
<td>Cl_{min}</td>
<td>-1.24</td>
<td>-1.21</td>
<td>2.5%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optimization function</th>
<th>Reduced tool</th>
<th>Full tool</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>9,759</td>
<td>10,052</td>
<td>3.0%</td>
</tr>
<tr>
<td>Drag</td>
<td>18,070</td>
<td>18,070</td>
<td>0.0%</td>
</tr>
<tr>
<td>Complexity</td>
<td>39.6</td>
<td>39.6</td>
<td>0.0%</td>
</tr>
<tr>
<td>Total fitness</td>
<td>2.51</td>
<td>2.53</td>
<td>0.8%</td>
</tr>
</tbody>
</table>

Table 5.4: Comparison between the full and reduced optimization tools in combination #1, first optimization campaign.

underlying assumptions (depending on intermediate results), which do not require recalculating the surrogates if flight conditions remain unchanged. As the number of additional optimizations increases, the benefit of the reduced model increases, approaching the asymptotic reduction factor of 260. For instance, in the 36 optimization runs performed in subsections V.A and V.B above, the reduction factor was 36x405.4/(122.5+36x1.56) =81.7. It is also to be noted that the CPU time required by the conventional computational tool (608 CPU days) would make this tool impractical in this task, while its counterpart for the reduced tool (7 days) is quite reasonable.

Note that all CPU times mentioned above scale with the number of AVL runs that are necessary using either the conventional tool and the reduced tool developed in above. Thus, the saving factors 3.6 and 260 mentioned above are independent of the fidelity of the CFD solver. In other words, the reduced tool would show the same advantages if a high fidelity CFD solver (such as a Reynolds averaged Navier Stokes solver) were used. The problem would be the huge amount of CFD runs that would be necessary (using either the conventional and the reduced tool), which is ultimately the reason for not using high-fidelity models in conceptual design, as explained in chapter 4.

5.2.4 Further improvements of the method

As developed above, the reduced optimization tool meets all requirements enumerated in chapter 4 above. In particular, it provides results comparable to the full optimization tool in a much smaller computational time, since the CPU time required to generate the surrogates was 3.6 times smaller than that required by the full optimization tool to perform one sin-
CHAPTER 5. GLOBAL OPTIMIZATION

gle optimization run. After that, each optimization run with the reduced model requires a CPU time that is 300 times smaller than that required by the conventional tool. Thus, the main advantage of the reduced tool is clearly illustrated. Further improvements, which are beyond the scope of this paper, could result in reducing the preprocess (to construct the surrogates) computational time, which required $6^6$ runs of the aerodynamic solver in a structured mesh in the parameter subspace associated with the HTP geometry (the VTP geometry depended on a fewer number of parameters). A large reduction of the preprocess CPU time could result from the following observations:

1. The geometrical constrains (which can be imposed a priory) are met by only 36.8% of the $6^6$ considered configurations; such reduction would exponentially increase when the number of design parameters is increased. Thus, considering only those configurations than meet the geometrical constrains reduces the computational effort, but requires considering non-structured meshes, which in turn requires to appropriately modify the HOSVD description. Such extension of HOSVD is the object of our current research.

2. Aerodynamic/aeroelastic/control constrains are met by only a 5.8% (this percentage decreases to 5.5% if geometrical constrains are also considered) of the $6^6$ configurations, which would lead to a stronger CPU time reduction; again, such reduction would exponentially increase when the number of design parameters is increased. These constrains are not known a priori, but could be either: (a) estimated using a lower fidelity aerodynamic solver/empirical construct, or (b) imposed sequentially when constructing the surrogates for the various technical disciplines (namely, those configurations that do not meet the constrain associated with the surrogate for a the technical disciplines are not considered when constructing the surrogates for the remaining technical disciplines). Again, considering only those configurations that meet all constrains requires the modification of the HOSVD description.

3. Constructing the HOSVD tool should not require computing all points of the structured mesh in the parameter space, but only the much smaller subset that is associated with compressed description provided by the HOSVD decomposition, which requires an appropriate selection of the configurations that are actually calculated. It is to be noted here that the compression factor increases exponentially as the dimension of the database increases [41]. The difficulty is of course to identify those points of the parameter space that should be calculated.
by the aerodynamic solver to obtain optimal results. This is related to
the so called sampling problem, and is currently the subject of active
research, see [6], [10], and references there in.

The necessity of reducing the preprocess CPU time associated with
POD/SVD based model reduction methods is an ubiquitous problem, whose
solution would be of interest in many applications of these methods, and is
the object of our current research. But this is also well beyond the object of
the present thesis and will not be pursued here any further.

5.3 Preliminary conclusions

A method has been presented for conceptual design in Aeronautics that
is made up of two levels. A genetic algorithm is in charge of optimizing
a target function that accounts for various multi-disciplinary ingredients
(e.g., weight, drag, and geometrical complexity). The various technical dis-
ciplines are modeled using surrogates (at various sub-levels) that are con-
structed using a combination of HOSVD and one dimensional interpo-
ation. The computation of the surrogates requires running the original com-
putational tools (for, e.g., Aerodynamics) at some structured, discretized
values of some subsets of the set of free design parameters. In fact, for
each technical discipline, the computational tool must be run only for the
free parameters involved in the discipline, which is always smaller than
the total number of free parameters. Thus, the advantages of the reduced
model become more evident as the number of free parameters increase.
For instance, if quadratic corrections are added to the swept angles of the
HTP and the VTP and a linear torsion of the HTP is considered, and four
free parameters (two for the HTPs and two for the VTP) were added, then
seven additional parameters must be added. This significantly increase
(from nine to sixteen) the dimension of the search space. With a conven-
tional tool, this requires in the GA to exponentially increase the number of
individuals per generation. With the reduced model instead, the number
of geometrical free parameters in the HTP and the VTP is increased from
six to eight and from three to four, respectively, which involves an afford-
able increase in the computational effort associated with constructing the
aerodynamics surrogates; a surrogate could be constructed for the struc-
ture (to account for the new structural free parameters) whose calculation
would be quite inexpensive. All these involve a much lower increase in the
required than that associated with the conventional tool.

The surrogates are much more computationally efficient than the con-
ventional computational tool (namely, the required CPU time was decreased
by a factor of 81.7 in the example considered above) and can be defined as
independent of the various modeling parameters that must be tuned in advance. Thus, tuning as well as optimization can be performed using the surrogate models. This makes the advantages of the reduced model even more evident. For instance, in the example considered in this chapter, the number of free design parameters was nine and five additional tunable parameters (the three weights in the objective function and the parameters and appearing in the definition of complexity) were present. For simplicity, no additional tunable parameters (e.g., the various safety factors and other parameters appearing in the structure calculation) were considered. The number of tunable parameters in actual conceptual design can be quite large, which could make it impractical an appropriate tuning using the conventional optimization tool; with the surrogates instead, the computational cost of tuning can be reasonable, even in a standard PC.

In fact, CPU time is a quite important issue in conceptual design, which must be performed within specific cost and time constrains. Therefore, an optimization method cannot be accepted unless those constrains are met, irrespectively of how robust, flexible, and reliable the method can be. In such scenario, we expect that the method presented in this thesis, which is flexible enough to be combined with any specific optimization method, is a promising alternative to current approaches.

For the sake of brevity, the illustration of the method in chapter 4 was rather simplified, but the advantages of the method would be even clearer in more realistic conceptual design problem, which could include:

1. More free parameters in the definition of the HTPs and the VTP, to account for, e.g., torsion in the HTP and higher order corrections in the various polynomial approximations.

2. A more detailed model of the structure.

3. A better modelization of viscous drag and stall, using a more detailed description of the boundary layer.

The method developed in this chapter is a basic version, which could be improved in various ways:

1. A multi-fidelity approach could be used, in which various aerodynamic models (based on, e.g., a panel method or a finite volume+turbulence model method) are sequentially used, in the whole parameter spaces and in subregions near pre-optimized configurations.

2. A multi-objective genetic algorithm would give a better performances than the more standard genetic algorithm that has been used.
3. The computational effort required to construct the surrogates for the various technical disciplines can be reduced strongly, as explained in section 2.2.

4. A gradient like method can be combined with the GA to speed up convergence of the elite individuals.
The results in the last chapter using the global optimization tool are now completed using the local tool. The same comments made at the beginning of last chapter apply here and are not repeated. In particular, the basic-local optimization tool can be discarded a priori, and the selected tool will be the modular-local one. The structure of this chapter is similar to that of the last one. The really new issue is the combination of the global and local tools, which can be performed in two ways:

- Using global optimization best individuals as initial guesses for local optimizations tuning the best individuals parameters
- Using local optimization tool within the global optimization method to diminish the required time to find an optimum solution

With these ideas in mind, the chapter is organized as follows. For convenience, the objective function, the geometry, and the restrictions are briefly recalled in section 6.1. Restrictions must be treated with care in the local optimization tool, as it will be explained and done in section 6.1.4. Similarly, the stability of the method must not be taken for granted. Instead, it will be tested in section 6.1.5. The above mentioned two combinations of the global and local tools will be developed in section 6.2. Results using the the two methods for the test problem will be provided and discussed in section 6.3, the influence on optimization results of using a higher fidelity model for viscous drag is discussed in section 6.4 and some conclusion and future research guidelines are suggested will be given in section 6.5.

6.1 Case study application

In order to facilitate the comprehension of the present chapter, let us summarize here the case study, described in chapter 4, which consists in the conceptual design of a rear-end empennage of commercial airplane. Conceptual design in this test problem is performed minimizing an objective
function subject to some restrictions, looking for possible non-conventional configurations.

6.1.1 Objective function

The objective function coincides with that used in the last chapter, namely

\[ \Phi = \varepsilon_1 \frac{\text{weight}}{\text{weight}_{\text{ref}}} + \varepsilon_2 \frac{\text{drag}}{\text{drag}_{\text{ref}}} + \varepsilon_3 \frac{\text{complexity}}{\text{complexity}_{\text{ref}}} \]

where subindex \( \text{ref} \) refers to the reference configuration, which is described in section 4.5. The three ingredients in the objective function are calculated in terms of the design variables as explained in the last chapter, section 4.3. It is to be recalled that the surrogate model is constructed from snapshots that are calculated using the AVL solver, which cannot be directly combined with a boundary layer analogy to calculate viscous drag. Thus, viscous drag will be calculated in most part of the chapter using a wetted area correlation. But the influence of using the higher fidelity surrogate for viscous drag derived in chapter 2 (instead of wetted area correlations) will be analyzed in section 6.4, by the end of the chapter.

6.1.2 Geometry

The geometrical description of the rear end is determined by nine parameters defining horizontal and vertical stabilizers. Six parameters are needed to describe the whole horizontal tail plane (HTP) and three more parameters to describe the vertical tail plane (VTP). Those parameters coincide with those used in the last chapter, section 4.2.1, namely

- The lengths of two pieces of the HTP, \( \lambda_1, \lambda_2 \).
- The HTP swept angle, \( \lambda_3 \).
- The dihedral angle of the HTP, \( \lambda_4 \), and a quadratic correction in the external piece of the HTP, \( \lambda_5 \).
- The taper ratio of the HTP, \( \lambda_6 \).
- The total height of the VTP, \( \lambda_7 \).
- The swept angle of the VTP, \( \lambda_8 \).
- The taper ratio of the VTP, \( \lambda_9 \).
6.1.3 Restrictions

Some restrictions are applied to the geometrical parameters that define the rear-end geometry. These restrictions must be included as constraints in the local optimization tool developed in section 2.5.3. Some additional restrictions will be imposed that related with aerodynamic characteristics.

6.1.3.1 Geometric restrictions

Two restrictions are imposed to each design variable ($\lambda_1, \ldots, \lambda_9$) that defined the span in which the design variable will be allowed to vary. The restrictions for the i-th design variable are

$$C^G_{ij} = (-1)^{i-1} |\lambda_i - \beta_{ij}| \geq 0$$

where $\beta_{i1}$ and $\beta_{i2}$ are the variable span lower and upper bounds, respectively. The actual values of these constants are summarized in table 6.1.

6.1.3.2 Aerodynamic restrictions

The aerodynamic restrictions consist in imposing that new configurations do not worsen some properties of the baseline configuration in connection with longitudinal and lateral stability, longitudinal and lateral controllability, and stall. The restrictions coincide with their counterparts in the last chapter, section 4.4. They are denoted as (using the same notation as in the last chapter)

- $C^A_1 \equiv c_{m_\alpha}^{\text{ref}} - c_{m_\alpha} \geq 0$,
- $C^A_2 \equiv c_{n_\beta} - c_{n_\beta}^{\text{ref}} \geq 0$,
- $C^A_3 \equiv c_{m_\delta}^{\text{ref}} - c_{m_\delta} \geq 0$,
- $C^A_4 \equiv c_{n_\delta} - c_{n_\delta}^{\text{ref}} \geq 0$,
6.1.4 Constraints particularity

The optimization process that will be used here is the gradient-like method described in the last chapter, section 2.5, which for convenience is summarized in figure 2.4. But before applying the local surrogate model, it is to be noted that some of the design variables, such as the HTP and VTP lengths and HTP and VTP chord ratio, cannot under any circumstance be negative. These is because negative values of these variables would make no physical sense, and neither the objective function ingredients and nor the aerodynamic constraints could be calculated at those unphysical values. Thus, the methodology described in chapter 2, subsection 2.5.3.3 is the appropriate one because it does not require the objective function to be defined when the one-sided constraints do not hold. This methodology was based on the orthogonal projection of the (straight) searching line on the border of the constraints, which required solving the set of nonlinear equations (2.26), which is solved using a Newton method and is rewritten here for convenience as

\[
C_j(\lambda^{(k)} + \alpha^{(k)}d^{(k)} + \sum \beta_j \nabla C_j) = 0, \tag{6.1}
\]

where the index \( j \) denotes only those constraints (either exact or one-sided) that are active. But some care must be taken when some of the problematic constraints are active because negative values of these problematic constraints cannot be allowed in any of the iterations of the above mentioned Newton method. In other words, there are three possible situations that must be distinguished in the algorithm:

- The active restrictions are all geometric constraints. This case does not need specific treatment because geometric constraints can always be evaluated.

- The active restrictions are all aerodynamic constraints. Again, no specific treatment is needed because all geometric constraints are met and then, aerodynamic constraints can be evaluated.

- Both geometric and aerodynamic restrictions active. It is only this case in which the above mentioned projection is needed.

The process (sketched in Fig.6.1) distinguishes between the three above mentioned cases as follows:
1. **Only geometric constraints are active.** In this case, the projection in eq.(6.1) must be made along the geometric constraints only, without any additional concern.

2. **Only aerodynamic constraints are active.** As above, the projection is performed using the aerodynamic constraints, with no additional concern.

3. **Both geometric and aerodynamic constraints are active.** Then, the projection based in the Newton method must be performed in such a way that the geometric constraints are always satisfied. This is done with a double iteration: (i) an outer iteration that imposes the aerodynamic constraints, and (ii) an inner iteration (which must be performed previously to each step of the outer iteration) that imposes the geometrical constraints. Doing that ensures that the geometric constraints are already satisfied each time the aerodynamics constraints are evaluated.

It is to be noted that the implementation of this process must be made with care, to obtain a robust algorithm that accounts for all possible scenarios.
6.1.5 Method stability tests

The local tool described in the previous subsections must be tested in connection with its stability. To begin with, some random configurations have been selected to test how the configuration parameters evolve as the local optimization process proceeds.

A total number of 500 optimizations are considered to study the stability of the method, which will be checked attending to both the value of the fitness, and the design parameters values of the minimizer. The former stability test in turn will be performed in terms of four different properties:

- The objective function ingredients, in table 6.2.
- The solutions repeatability, in table 6.3.
- The solutions parameters, in table 6.4.
- The active constraints, in tables 6.5 and 6.6.

Note that the best solution is not the most repeated one. This means that the basin of attraction of this solution is not the largest one, which in turn means that some care must be taken to ensure that the initial guess for the local optimization tool be in the appropriate basin of attraction, if the global minimum is sought. It should be noted that there are some cases that do not converge in the allowable number of iterations due to non-convergence of the projection process described in figure 6.1. 26 cases out of a total number of 500 cases that were run did not converge, which amounts to a 5.2% of the total number of cases.

<table>
<thead>
<tr>
<th>Solution</th>
<th>Fitness</th>
<th>Weight (Kg)</th>
<th>Drag (N)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>0.8267</td>
<td>9,881</td>
<td>17,117</td>
<td>39.70</td>
</tr>
<tr>
<td>#2</td>
<td>0.8282</td>
<td>9,903</td>
<td>17,137</td>
<td>39.79</td>
</tr>
<tr>
<td>#3</td>
<td>0.8321</td>
<td>10,271</td>
<td>16,199</td>
<td>41.36</td>
</tr>
<tr>
<td>#4</td>
<td>0.8332</td>
<td>10,077</td>
<td>16,485</td>
<td>41.69</td>
</tr>
<tr>
<td>#5</td>
<td>0.9646</td>
<td>10,897</td>
<td>18,051</td>
<td>55.30</td>
</tr>
<tr>
<td>#6</td>
<td>0.9668</td>
<td>10,856</td>
<td>18,272</td>
<td>55.22</td>
</tr>
<tr>
<td>#7</td>
<td>0.9682</td>
<td>10,876</td>
<td>18,138</td>
<td>55.76</td>
</tr>
<tr>
<td>#8</td>
<td>0.9683</td>
<td>10,877</td>
<td>18,115</td>
<td>55.84</td>
</tr>
<tr>
<td>#9</td>
<td>0.9685</td>
<td>10,798</td>
<td>18,270</td>
<td>55.82</td>
</tr>
<tr>
<td>#10</td>
<td>0.9708</td>
<td>10,892</td>
<td>18,150</td>
<td>56.09</td>
</tr>
<tr>
<td>#11</td>
<td>0.9948</td>
<td>11,562</td>
<td>16,602</td>
<td>61.37</td>
</tr>
<tr>
<td>#12</td>
<td>1.0350</td>
<td>12,027</td>
<td>16,866</td>
<td>65.08</td>
</tr>
<tr>
<td>#13</td>
<td>1.0383</td>
<td>12,179</td>
<td>16,566</td>
<td>65.76</td>
</tr>
<tr>
<td>#14</td>
<td>1.1557</td>
<td>14,018</td>
<td>17,095</td>
<td>74.83</td>
</tr>
</tbody>
</table>

Table 6.2: Stability solutions, objective function ingredients.
### 6.1. CASE STUDY APPLICATION

<table>
<thead>
<tr>
<th>Solution</th>
<th>Number</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>51</td>
<td>10.2</td>
</tr>
<tr>
<td>#2</td>
<td>22</td>
<td>4.4</td>
</tr>
<tr>
<td>#3</td>
<td>102</td>
<td>20.4</td>
</tr>
<tr>
<td>#4</td>
<td>194</td>
<td>38.8</td>
</tr>
<tr>
<td>#5</td>
<td>47</td>
<td>9.4</td>
</tr>
<tr>
<td>#6</td>
<td>24</td>
<td>4.8</td>
</tr>
<tr>
<td>#7</td>
<td>7</td>
<td>1.4</td>
</tr>
<tr>
<td>#8</td>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>#9</td>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>#10</td>
<td>19</td>
<td>3.8</td>
</tr>
<tr>
<td>#11</td>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>#12</td>
<td>3</td>
<td>0.6</td>
</tr>
<tr>
<td>#13</td>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>#14</td>
<td>1</td>
<td>0.2</td>
</tr>
</tbody>
</table>

**Table 6.3:** Stability solutions, solutions repeatability.

<table>
<thead>
<tr>
<th>Solution</th>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
<th>$\lambda_3$</th>
<th>$\lambda_4$</th>
<th>$\lambda_5$</th>
<th>$\lambda_6$</th>
<th>$\lambda_7$</th>
<th>$\lambda_8$</th>
<th>$\lambda_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>20.00</td>
<td>0.00</td>
<td>0.28</td>
<td>0.75</td>
<td>1.00</td>
<td>0.51</td>
<td>0.00</td>
<td>1.00</td>
<td>0.10</td>
</tr>
<tr>
<td>#2</td>
<td>20.00</td>
<td>0.00</td>
<td>0.27</td>
<td>0.76</td>
<td>-1.00</td>
<td>0.51</td>
<td>0.00</td>
<td>1.00</td>
<td>0.10</td>
</tr>
<tr>
<td>#3</td>
<td>14.52</td>
<td>5.60</td>
<td>0.31</td>
<td>0.59</td>
<td>0.38</td>
<td>0.51</td>
<td>0.00</td>
<td>1.00</td>
<td>0.10</td>
</tr>
<tr>
<td>#4</td>
<td>14.72</td>
<td>5.77</td>
<td>0.31</td>
<td>0.54</td>
<td>1.00</td>
<td>0.54</td>
<td>0.00</td>
<td>1.00</td>
<td>0.10</td>
</tr>
<tr>
<td>#5</td>
<td>15.31</td>
<td>0.00</td>
<td>0.18</td>
<td>0.15</td>
<td>1.00</td>
<td>0.38</td>
<td>15.29</td>
<td>0.23</td>
<td>0.36</td>
</tr>
<tr>
<td>#6</td>
<td>15.20</td>
<td>0.00</td>
<td>0.15</td>
<td>0.18</td>
<td>-1.00</td>
<td>0.41</td>
<td>15.33</td>
<td>0.23</td>
<td>0.36</td>
</tr>
<tr>
<td>#7</td>
<td>16.47</td>
<td>0.00</td>
<td>0.21</td>
<td>0.37</td>
<td>1.00</td>
<td>0.39</td>
<td>13.49</td>
<td>0.21</td>
<td>0.30</td>
</tr>
<tr>
<td>#8</td>
<td>16.72</td>
<td>0.00</td>
<td>0.22</td>
<td>0.40</td>
<td>1.00</td>
<td>0.39</td>
<td>13.08</td>
<td>0.21</td>
<td>0.29</td>
</tr>
<tr>
<td>#9</td>
<td>15.21</td>
<td>0.00</td>
<td>0.16</td>
<td>0.15</td>
<td>-0.48</td>
<td>0.40</td>
<td>15.45</td>
<td>0.23</td>
<td>0.36</td>
</tr>
<tr>
<td>#10</td>
<td>16.71</td>
<td>0.00</td>
<td>0.22</td>
<td>0.40</td>
<td>-1.00</td>
<td>0.39</td>
<td>13.20</td>
<td>0.21</td>
<td>0.29</td>
</tr>
<tr>
<td>#11</td>
<td>18.43</td>
<td>0.00</td>
<td>0.21</td>
<td>0.40</td>
<td>-1.00</td>
<td>0.10</td>
<td>14.15</td>
<td>0.21</td>
<td>0.10</td>
</tr>
<tr>
<td>#12</td>
<td>16.27</td>
<td>3.38</td>
<td>0.21</td>
<td>0.36</td>
<td>-1.00</td>
<td>0.16</td>
<td>13.97</td>
<td>0.21</td>
<td>0.10</td>
</tr>
<tr>
<td>#13</td>
<td>16.87</td>
<td>2.81</td>
<td>0.22</td>
<td>0.34</td>
<td>-1.00</td>
<td>0.10</td>
<td>14.53</td>
<td>0.22</td>
<td>0.10</td>
</tr>
<tr>
<td>#14</td>
<td>17.31</td>
<td>6.26</td>
<td>0.43</td>
<td>0.00</td>
<td>1.00</td>
<td>0.10</td>
<td>14.44</td>
<td>0.10</td>
<td>0.10</td>
</tr>
</tbody>
</table>

**Table 6.4:** Stability solutions parameters.

105
As it was expected, a finite number of solutions is obtained, namely 14 local minima. Most of the converged cases (73.8%) gave solutions that are close to the optimum solution, which is a "V-shaped" configuration without VTP, anticipated in chapter 4. Note that in solutions #1 and #2, the parameter $\lambda_5$ does not affect the configuration because $\lambda_2 = 0$ both cases. Thus, both solutions are quite similar among each other, which is consistent with the fact that both solutions exhibit a similar fitness.

Table 6.5 shows which geometric constraints are active for each solution. In this table, those parameters that do not affect the actual configuration (as it happened with the parameter $\lambda_5$ in the comment in the last paragraph), they are considered as non-active, in spite of the fact that the parameter value is in the constraint border.

Finally, the aerodynamic active constraints are summarized in table 6.6. Note that the stability restrictions are those that are more restrictive to the case study. However, longitudinal control restriction, $cm_{\delta}$, is also active in many solutions, in particular, in solutions #3 and #4, in which the straight piece of the HTP (where the controls are located) is smaller.

Then, it can be concluded that the method is stable enough as to be combined in a global optimization process, where the different basins of attraction were taken into account. This combination is considered next.
6.2 Global and local optimization tools combination

The local and global optimization tools are now combined, considering the same test case addressed above, namely that described in chapter 4. The combination will be done in in two different fashions:

- Applying first the global tool and then the local tool, in subsection 6.2.1).
- Combining both optimization tools in the same algorithm, in subsection 6.2.2).

In both methods, it is convenient to modify the calculation of the objective function ingredients to both (i) reduce the computational time and (ii) ensure that the ingredients be smooth enough, as required by the local optimization tool. Non-smoothness result from the necessity to apply the local optimization tool to some individuals selected by the global tool that are close to the constraint borders (for instance, individuals showing a quite small VTP). Those configurations did not appear in the application of the local tool alone.

In chapter 4, the weight computation was done using a HOSVD+I definition of the loads that sizes the required structure, but weight itself was calculated directly, without any HOSVD surrogate. This process was not determinant in the computational time required by the original tool, but after using the HOSVD surrogate models in the aerodynamics computations, comes to be the most computationally expensive. Moreover, it can

<table>
<thead>
<tr>
<th>Solution</th>
<th>(c_{m_a})</th>
<th>(c_{n_{\beta}})</th>
<th>(c_{m_{\phi}})</th>
<th>(c_{n_{\phi}})</th>
<th>(c_{l_{\max}})</th>
<th>(c_{l_{\min}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#2</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#6</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#7</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#8</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#9</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#10</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#11</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#12</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#13</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#14</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.6: Stability solutions, aerodynamic constraints. ✓ means that the restriction is active.
lead to the apparition of small peaks in the resulting weight calculations. Both difficulties are avoided at a time using the HOSVD methodology also to calculate the remaining ingredients of the objective function (in particular, weight). The three ingredients of the objective function are discretized as

\[ W_{HTP} = W(\lambda_1, \ldots, \lambda_6), \quad W_{VTP} = W(\lambda_7, \lambda_8, \lambda_9), \]
\[ D_{HTP} = D(\lambda_1, \ldots, \lambda_6), \quad D_{VTP} = W(\lambda_7, \lambda_8, \lambda_9), \]
\[ C_{HTP} = C(\lambda_1, \ldots, \lambda_6), \quad C_{VTP} = W(\lambda_7, \lambda_8, \lambda_9), \]

and the HOSVD+I method is apply to the resulting tensors.

In order to ensure smoothness, the cost function in equation 4.3.1 is also corrected by defining parameter \( b \) as

\[ b = 0.2(1 - 10^{-2\lambda_7}). \]

Note that (i) \( b = 0 \) is when VTP is not considered, (ii) \( b = 0.198 \) is quite close to the value considered in section 4.3.1 (namely, \( b = 0.2 \) when \( \lambda_7 = 1 \), (iii) and \( b \to 0.2 \) as the VTP height increases further.

These improvements reduce to the computational time required for each configuration from 37 CPU ms to 2 CPU ms. In addition, it is convenient to tune the GA parameters again, taking into account that now, the GA is not required to converge to an optimal solution, but to provide good initial estimates for the local tool. The new GA parameters (labeled as case #2) are provided in table 6.7, where the former GA parameters used in section 5.1 (labeled as case #1) are also provided for reference. The optimization process is considered as converged if the best individual remains unchanged for 30 generations.

Table 6.8 shows a comparison of the values of the objective function ingredients for the best individuals resulting from global optimizations performed with the old and new GA parameters. Both configurations are similar to each other, although they correspond to different values of the design parameters (table 6.9). As anticipated, the genetic algorithm does not necessarily provide the optimum solution. Thus, the difference between the best individuals provided by both GAs is not surprising.

The two above mentioned strategies for combining the global and local optimization tools are now performed. The resulting hybrid optimization
6.2. GLOBAL AND LOCAL OPTIMIZATION TOOLS COMBINATION

<table>
<thead>
<tr>
<th>Case</th>
<th>$\epsilon_1$</th>
<th>$\epsilon_2$</th>
<th>$\epsilon_3$</th>
<th>Weight (kg)</th>
<th>Drag (N)</th>
<th>Complexity</th>
<th>Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>$1/3$</td>
<td>$1/3$</td>
<td>$1/3$</td>
<td>9,759</td>
<td>18,070</td>
<td>39.6</td>
<td>0.84</td>
</tr>
<tr>
<td>#2</td>
<td>$1/3$</td>
<td>$1/3$</td>
<td>$1/3$</td>
<td>10,136</td>
<td>17,371</td>
<td>41.5</td>
<td>0.84</td>
</tr>
</tbody>
</table>

Table 6.8: Best individuals for global optimizations in cases #1 and #2; objective function ingredients comparison.

<table>
<thead>
<tr>
<th>Case</th>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
<th>$\lambda_3$</th>
<th>$\lambda_4$</th>
<th>$\lambda_5$</th>
<th>$\lambda_6$</th>
<th>$\lambda_7$</th>
<th>$\lambda_8$</th>
<th>$\lambda_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>19.66</td>
<td>0.15</td>
<td>0.26</td>
<td>0.76</td>
<td>0.02</td>
<td>0.67</td>
<td>0.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>#2</td>
<td>17.17</td>
<td>3.31</td>
<td>0.45</td>
<td>0.69</td>
<td>0.81</td>
<td>0.63</td>
<td>0.0</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6.9: Best individuals for global optimizations in cases #1 and #2; parameters comparison.

tools (HOMs) will be labeled as HOM1 and HOM2. These are now described, in the next two subsections.

6.2.1 Local optimization improving global solutions HOM1

The first strategy consists in first running the GA global tool and then applying the local tool, taking as initial guess some of the individuals selected by the GA. In fact, it has been tested that it is not enough to apply the local optimization tool to the GA-optimal solution only; in many cases, the best final result of the combined process is obtained from a suboptimal GA-solution, which thus turn out to be in the basin of attraction of the final best solution. GA-suboptimal solutions are defined here to be those that have been considered as best individuals in some generation of the GA-optimization process.

Seven different cases are considered here that correspond to those described in table 5.1. The results applying the HOM1 are summarized in table 6.11. Also, they will be compared in section 6.3 with their counterparts obtained using the second hybrid tool, which is described next.

6.2.2 Combined global/local optimization tool HOM2

The second strategy for combining the local and global tools consists in using the local optimization tool after each generation of the GA to optimize the best new individual of the generation and compare with the elite individuals. This process ensures that if any genes combination codifies an individual in the basin of attraction of the optimum solution, then the local optimization tool will for sure provide the optimum solution itself. In order to focusing on the largest basin of attraction, the elite is reduced to
CHAPTER 6. LOCAL OPTIMIZATION

<table>
<thead>
<tr>
<th>Combination</th>
<th>Method</th>
<th>Weight (Kg)</th>
<th>Drag (N)</th>
<th>Cost</th>
<th>Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>HOM1</td>
<td>9,881.6</td>
<td>17,116.8</td>
<td>39.7</td>
<td>0.8267</td>
</tr>
<tr>
<td></td>
<td>HOM2</td>
<td>9,863.0</td>
<td>16,980.2</td>
<td>39.7</td>
<td>0.8237</td>
</tr>
<tr>
<td>#2</td>
<td>HOM1</td>
<td>10,596.6</td>
<td>15,165.1</td>
<td>44.4</td>
<td>0.8729</td>
</tr>
<tr>
<td></td>
<td>HOM2</td>
<td>10,560.6</td>
<td>15,104.9</td>
<td>44.3</td>
<td>0.8696</td>
</tr>
<tr>
<td>#3</td>
<td>HOM1</td>
<td>9,353.3</td>
<td>20,444.2</td>
<td>38.2</td>
<td>0.7537</td>
</tr>
<tr>
<td></td>
<td>HOM2</td>
<td>9,322.6</td>
<td>20,386.7</td>
<td>38.1</td>
<td>0.7523</td>
</tr>
<tr>
<td>#4</td>
<td>HOM1</td>
<td>10,696.6</td>
<td>15,089.2</td>
<td>43.7</td>
<td>0.7718</td>
</tr>
<tr>
<td></td>
<td>HOM2</td>
<td>10,660.4</td>
<td>15,031.3</td>
<td>43.6</td>
<td>0.7702</td>
</tr>
<tr>
<td>#5</td>
<td>HOM1</td>
<td>9,353.1</td>
<td>20,422.1</td>
<td>38.2</td>
<td>0.8408</td>
</tr>
<tr>
<td></td>
<td>HOM2</td>
<td>9,322.3</td>
<td>20,364.5</td>
<td>38.1</td>
<td>0.8380</td>
</tr>
<tr>
<td>#6</td>
<td>HOM1</td>
<td>12,547.0</td>
<td>13,779.9</td>
<td>52.8</td>
<td>0.7207</td>
</tr>
<tr>
<td></td>
<td>HOM2</td>
<td>12,490.0</td>
<td>13,730.7</td>
<td>52.6</td>
<td>0.7181</td>
</tr>
<tr>
<td>#7</td>
<td>HOM1</td>
<td>9,610.8</td>
<td>20,298.5</td>
<td>37.0</td>
<td>0.6491</td>
</tr>
<tr>
<td></td>
<td>HOM2</td>
<td>9,579.2</td>
<td>20,240.6</td>
<td>36.9</td>
<td>0.6316</td>
</tr>
</tbody>
</table>

Table 6.10: The seven combinations considered in the actual optimization campaign.

Table 6.11: Best individuals for both local optimization tools in objective function parameters combinations #1 to #7

just one individual. Note that the use of the local algorithm increases the computational time required to run each generation, but accelerates the GA convergence, as pointed out in section 6.3.

6.3 Results and discussion

The results obtained applying the two hybrid methods described in section 6.2, for the seven combinations defined in table 6.10, are now discussed. The objective function values for the new optimum configurations are summarized in table 6.11.

As it was to be expected, the results from HOM2 are in all cases better than solutions obtained with HOM1. Concerning the computational time for each method, the required CPU time for each of the optimizations in
6.4. THE INFLUENCE OF USING A HIGHER FIDELITY MODEL FOR VISCOUS DRAG

<table>
<thead>
<tr>
<th>Combination</th>
<th>Number of generations</th>
<th>Number of LOs</th>
<th>Time GA (m)</th>
<th>Time LO (m)</th>
<th>Total time (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>125</td>
<td>10</td>
<td>41.7</td>
<td>3.2</td>
<td>44.9</td>
</tr>
<tr>
<td>#2</td>
<td>125</td>
<td>12</td>
<td>41.7</td>
<td>4.8</td>
<td>46.5</td>
</tr>
<tr>
<td>#3</td>
<td>91</td>
<td>9</td>
<td>30.3</td>
<td>17.2</td>
<td>47.5</td>
</tr>
<tr>
<td>#4</td>
<td>91</td>
<td>9</td>
<td>30.3</td>
<td>2.4</td>
<td>42.7</td>
</tr>
<tr>
<td>#5</td>
<td>124</td>
<td>12</td>
<td>41.3</td>
<td>26.0</td>
<td>67.3</td>
</tr>
<tr>
<td>#6</td>
<td>235</td>
<td>24</td>
<td>78.3</td>
<td>36.9</td>
<td>115.2</td>
</tr>
<tr>
<td>#7</td>
<td>111</td>
<td>10</td>
<td>37.0</td>
<td>20.1</td>
<td>57.1</td>
</tr>
</tbody>
</table>

Table 6.12: HOM1 computational time summary for objective function parameters combinations #1 to #7

<table>
<thead>
<tr>
<th>Combination</th>
<th>Number of generations</th>
<th>Total time (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>34</td>
<td>30.8</td>
</tr>
<tr>
<td>#2</td>
<td>31</td>
<td>19.3</td>
</tr>
<tr>
<td>#3</td>
<td>52</td>
<td>102.1</td>
</tr>
<tr>
<td>#4</td>
<td>30</td>
<td>16.8</td>
</tr>
<tr>
<td>#5</td>
<td>30</td>
<td>59.2</td>
</tr>
<tr>
<td>#6</td>
<td>30</td>
<td>40.0</td>
</tr>
<tr>
<td>#7</td>
<td>32</td>
<td>53.0</td>
</tr>
</tbody>
</table>

Table 6.13: HOM2 computational time summary for objective function parameters combinations #1 to #7

Table 6.11 are provided in tables 6.12 and 6.13. Note that the computational time for local optimizations depends on the case and the length of the path of the optimization process. As anticipated, the method HOM2 reduces significantly the number of generations needed to converge the GA, and thus it also reduces the overall required computational time (in spite of the fact that each generation is more computationally expensive than when calculated by the GA alone.

For illustration, the empennage configurations (mounted in a fuselage) of the best solutions for the various combinations considered above are schematized in figures 6.2-6.8. Note that the VTP is absent in all of them, which emphasizes the similar conclusion resulting from the application of the global optimization tool alone.

6.4 The influence of using a higher fidelity model for viscous drag

Finally, the fidelity level in viscous drag calculation is tested using the local optimization tool. The models for viscous drag calculation consid-
Figure 6.2: Best individual for combination #1

Figure 6.3: Best individual for combination #2
6.4. THE INFLUENCE OF USING A HIGHER FIDELITY MODEL FOR VISCOUS DRAG

Figure 6.4: Best individual for combination #3

Figure 6.5: Best individual for combination #4
**Figure 6.6:** Best individual for combination #5

**Figure 6.7:** Best individual for combination #6
6.4. THE INFLUENCE OF USING A HIGHER FIDELITY MODEL FOR VISCOUS DRAG

The methods used in the case study applications in chapters 4, 5, and 6 are low fidelity methods. These were used because of the necessity of considering a large number of design parameters, which requires to calculate a huge number of snapshots in the process. If the methods fidelity increases, the design process required time is not feasible for industrial purposes. However, it is possible to construct a HOSVD+I description of viscous drag using a higher fidelity method that allows to introduce the new information within the optimization process without increasing the design required time too much. In chapter 3, a medium fidelity level model to estimate some unconventional empennage viscous drag forces was described and tested. This method is used here to analyze how the fidelity level affects to the final solution.

For the sake of brevity, only combination #1 of the objective function parameters is considered here using the HOSVD+I description obtained with the panel plus boundary layer formulation in section 3.4.7. In this case, the optimum solution parameters for both the low (wetted area) and medium (panels method plus boundary layer) fidelity levels are compared in Table 6.14. Note that both the solutions obtained using the low and medium fidelity descriptions are similar to each other, except for the swept angle. This confirms that the method described in chapter 3 is appropriate for conceptual design. In any event, it must be taken into account that in
multi-fidelity optimization, different fidelity levels can lead to very different solutions.

6.5 Conclusions

The local optimization method developed in chapter 2 has been applied in the present chapter to improve the global optima obtained using a global method in chapter 5. The gradient-like plus Lagrange multipliers modified method with trust regions allows for obtaining the local optimum solution better than the genetic algorithm of previous chapter. Two different methods have been tested to combine both global and local optimizations. The study has lead that the introduction of a local optimization method within the global optimization process accelerates the global method convergence and most times the required time to obtain an optimum solution is reduced. In the same way accuracy is also greater if both methods are combined.

Both methods are fully compatible with both the basic and the modular configurations of the optimization tools developed in chapter 3. Thus, the local tool can be either basic or modular, but as explained at the beginning of the last chapter, the best option is the modular tool. Thus, only the modular local tool have been tested above.

Finally, the effects of increasing the fidelity have been illustrated by comparing the results obtained considering wetted area formulation and a panels method plus boundary layer formulation in the drag force computation. Despite, in this case, increasing the models fidelity does not improve the obtained results, it should not be ignored in general design phases. Thus, it can be easily guessed that using a multi-fidelity approach in local optimizations could further improve the designs. This further improvement is well beyond the scope of this thesis. Nevertheless, the results in the thesis suggest that a surrogate-based multifidelity approach could be developed that would only require a limited number of runs of the high fidelity CFD tools to calculate snapshots. This could be a quite advantageous alternative to current (non-surrogate), which require a huge (non-affordable) number of CFD runs.

Table 6.14: Optimum solutions for both fidelity levels (low: wetted area, and medium: panels method plus boundary layer formulation).

<table>
<thead>
<tr>
<th>Fidelity</th>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
<th>$\lambda_3$</th>
<th>$\lambda_4$</th>
<th>$\lambda_5$</th>
<th>$\lambda_6$</th>
<th>$\lambda_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>20.00</td>
<td>0.00</td>
<td>0.28</td>
<td>0.75</td>
<td>1.00</td>
<td>0.51</td>
<td>0.00</td>
</tr>
<tr>
<td>Medium</td>
<td>20.00</td>
<td>0.00</td>
<td>0.36</td>
<td>0.76</td>
<td>1.00</td>
<td>0.56</td>
<td>0.00</td>
</tr>
</tbody>
</table>

116
Various surrogate methods have been developed for the efficient optimization of conceptual design in engineering systems. As a general comment, even though the methods have been tested on an specific aeronautic test problem, (i) these methods are generic, not relying on specific engineering systems, and (ii) no intention have been made of developing a final industrial tool, amenable to be directly used in actual industrial environments. Instead, the main objective of the thesis is a methodological one, namely presenting and testing various new ideas in connection with the use of surrogate modeling methods in conceptual engineering systems design. A good part of the developed methods and tools could be industrialized right away, but this industrialization task remains to be done. And, what is more important, the thesis pretends to open new lines, more than closing any of them. As it could be appreciated in the presentation of the various chapters, there is space for further improvements in various points. As overall improvements:

- Calculating the snapshots in a structured mesh of the design parameter space, as has been done in the thesis, is not the best strategy. Redundancies among the snapshots (which is the basis of the whole surrogate model construction) should allow for only calculating a limited number of the snapshots in the structured mesh, and guessing the remaining snapshots using the redundancies. Selection of the sampling values of the design parameters could be done using a sampling method, whose development and application is well beyond the scope of the thesis. This improvement would be highly significant specially when a large number of design parameters are present.

- All snapshots have been calculated using the same CFD solver. A multifidelity approach in which the most interesting individuals are calculated using a higher fidelity method would highly improve the developed tools. This strategy would require appropriate updating (instead of fully calculating it) of the modes in the the dimensions
CHAPTER 7. CONCLUSIONS

associated with the various design parameters. This is a highly (and subtle) issue, which could be performed along the lines already developed in [49, 59]. Again, this improvement would be highly significant specially when a large number of design parameters are present.

Concerning the specific results of the thesis, and their possible improvements:

• The basic mathematical methodology was presented in chapter 2, where a HOSVD-tensor representation method has been selected to construct some surrogates of the different technical disciplines. Concerning optimization, a genetic algorithm has been selected for global optimization while a Broiden method combined with a trust regions and a flexible Lagrange multipliers method has been chosen for local optimization. The combination of surrogates models and both optimization methodologies are analyzed along the thesis.

• Using the basic methods described in chapter 2, various surrogate-based optimization tools were developed in chapter 3, along with some surrogates to calculate the skin friction drag force. The conclusion is that the HOSVD-based surrogate is the best compromise between precision and computational cost. Note that the HOSVD methodology and the optimization tools developed in this chapter are generic and independent of the particular engineering problem, which only appears implicitly through the snapshots, which can be calculated using various fidelity solvers.

• A case study is presented in chapter 4 where the various technical disciplines are modeled using a combination of HOSVD and one-dimensional interpolation. The computational tool must be run at some structured, discretized values of some subsets of the set of free design parameters. The number of parameters involve at each discipline is always smaller than the total number of free parameters, which allows to increase the design parameters number without increasing the required computational time.

• The previous case study is analyzed using a global optimization tool based on a genetic algorithm in chapter 5. In the example considered in that chapter, the use of HOSVD-based surrogate models allows to reduce the computational time by a factor of 81.7. Note that the number of free parameters in that example was nine and five additional tunable parameters were present. The number of tunable parameters in actual conceptual design can be quite large which could make impractical an appropriate tuning using the conventional optimization
tools; with the surrogates instead, the computational cost of tuning can be reasonable, even in a standard PC. The HOSVD-based surrogate method presented in this thesis is flexible enough to be combined with any specific optimization method as it is shown in chapters 5 and 6.

• A local optimization method is applied to the thesis case study to improve global approximations. This method allows to obtain local optimum solutions. Its combination with a global optimization tool allows to accelerate global convergence as it is shown in chapter 6. The benefits of increasing fidelity models are also outlined. Most times it is not possible to introduce high fidelity models in production design stages, but the use of surrogates opens a new development field in this important industrial area.
Bibliography


BIBLIOGRAPHY


BIBLIOGRAPHY


[56] K. SIDWELL, P. BARUAH, AND J. BUSSOLETI, *Pan air - A computer for predicting subsonic or supersonic linear potential flows about arbitrary configurations using a higher order panel method*, National Aeronautics and Space Administration.


