# Aircraft shape optimization for mission performance

François Gallard


# THÈSE

**En vue de l'obtention du**

# DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

**Délivré par :**

Institut National Polytechnique de Toulouse (INP Toulouse)

**Discipline ou spécialité :**

A®ronautique, Astronautique

---

## Présentée et soutenue par :

M. FRANCOIS GALLARD

le lundi 26 mai 2014

## Titre :

OPTIMISATION DE FORME D'UN AVION POUR SA PERFORMANCE SUR UNE MISSION

---

**Ecole doctorale :**

A®ronautique-Astronautique (AA)

**Unité de recherche :**

CERFACS

**Directeur(s) de Thèse :**

M. BIJAN MOHAMMADI

M. MARC MONTAGNAC

**Rapporteurs :**

M. JEAN-ANTOINE DESIDERI, INRIA SOPHIA ANTIPOLIS

M. MICHEL VISONNEAU, ECOLE CENTRALE NANTES

**Membre(s) du jury :**

M. CHRISTOPHE AIRIAU, UNIVERSITE TOULOUSE 3, Pr®sident

M. BIJAN MOHAMMADI, UNIVERSITE MONTPELLIER 2, Membre

M. CHRISTOPHE BLONDEAU, ONERA, Membre

M. MATTHIEU MEAUX, AIRBUS FRANCE, Membre

# Contents

# Remerciements

En premier lieu, je remercie mes encadrants de thèse, Matthieu Meaux, Marc Montagnac et Bijan Mohammadi. Je pense avoir eu beaucoup de chance, votre encadrement étant tout à fait complémentaire, vous avez su m'aider et me guider efficacement dans le dédale de problèmes ouverts que constitue l'optimisation aérodynamique. Je remercie en particulier Marc pour son souci de la précision scientifique, qui nous a permis de débusquer le diable dans les détails, et de grandement améliorer la qualité scientifique de notre travail. Merci aussi à Bijan pour m'avoir toujours poussé à creuser l'aspect mathématique des questions que nous avons soulevées, et de m'avoir aidé à en résoudre. Merci beaucoup à Matthieu pour son aide, le temps passé à me former, merci pour les discussions passionnantes sur l'optimisation, l'idée et le montage de la thèse, les nombreux concepts que tu m'as transmis et les journées de développement logiciel partagées et fort fructueuses.

Je souhaite également remercier chaleureusement les personnes avec qui j'ai eu la chance de travailler à Airbus et au CERFACS, et qui m'ont fourni une aide précieuse. Je remercie donc Loïc Tourrette, Thierry Druot, Simon Trapier, Pascal Larrieu, Marie-Josèphe Estève, Julien Cliquet, Cyril Gacherieu, Alain Soulard, Jean-Pierre Gualina, Frédéric Barrois, Anne Gazaix, Renaud Sauvage, Joël Brézillon, Julien Delbove, Florian Blanc, Steeve Champagneux, Ekaterina Makarova, Antoine Dechaume, Jérôme Tournier, Adrien Gomar, Guillaume Puigt, Xavier Pinel, Jean-François Boussuge et Selime Gürol ; vous contribuez à un environnement exceptionnellement riche en compétences dans lequel les problèmes restent rarement sans solution. Merci en particulier à Joel Brézillon pour le partage de sa riche expérience en optimisation et ses conseils sans lesquels les applications finales de la thèse fonctionnaient pas. Merci de même à Renaud Sauvage pour son aide, pour les discussions, parfois nocturnes et souvent animées, à propos des outils en CFD, l'optimisation, la géométrie, maillage. Je remercie aussi Christophe Blondeau, Antoine Dumont, Itham Salah el Din et Gérald Carrier de l'Onera pour m'avoir fourni des données et pour les échanges d'idées. Je remercie Christian Bulgubure, Eric Chaput, Thierry Surply et Eric Duceau pour votre intérêt porté à nos travaux et la qualité de vos remarques.

Merci à tous ceux qui ont contribué au développement de la chaine d'optimisation, sans laquelle je n'aurais pas pu réaliser cette thèse, merci notamment à Yann Mérillac, Arnaud Barthet, Loïc Boussouf, Maxime Pivetta et Guillaume Voizard. Merci aussi aux différents stagiaires avec qui j'ai travaillé (ce fût un plaisir que j'espère partagé), en particulier à Hadrien Montanelli pour l'optimisation en turbomachines ; à Pedro Langenegger-Videiro et Pierre-Jean Barjhoux, pour leurs contributions avec succès dans le développement du code structure adjoint, et merci à Anne

Gazaix pour ces travaux passionnants sur la MDO. Merci également aux utilisateurs de nos outils, notamment à Guillaume Drochon et Angel, pour votre patience et vos retours constructifs.

I would also like to greatly acknowledge Joaquim R.R.A. Martins and his team for their inspiring contributions to the multidisciplinary and aerodynamic optimization fields, and for the very rich exchanges of ideas.

Un grand merci au service informatique du CERFACS, pour la qualité exceptionnelle des services qu'ils nous fournissent. Merci également à Marie Labadens, Chantal Nasri, Michèle Campassens et l'équipe administrative de nous faciliter la vie au quotidien.

Merci aux amis qui m'ont hébergé au début de la thèse : Marie Hazet, Thomas Le Boulicaut & Marie Girier, Julia Bignon & Olivier Arnaud.

Pour finir, je garde un excellent souvenir de ces trois années, en bonne partie grâce à la fine équipe "détente et plus si affinités", merci à Jean-Christophe, Charlotte, Sophie, Xavier, Hugues, Nicolas, Sébastien, Frédéric, Gaëlle, Adrien, Thomas, Yannis, Lokman, Fabien, Marcel, Quentin, Sebastian, Benjamin, et bien sûr Flore.

# Nomenclature

**Roman letters**

| | |
|---|---|
| $\overline{\mathbf{U}}$ | Favre averaged eulerian flow particle speed |
| $\mathbf{I}$ | Identity matrix |
| $\mathcal{L}$ | Augmented Lagrangian function |
| $\overline{E}$ | Total energy per mass unit |
| $AoA$ | Angle of Attack |
| $c$ | Speed of sound |
| $C_s$ | Jet engines fuel specific consumption |
| $Cd$ | Drag coefficient |
| $Cd_p$ | Pressure drag coefficient |
| $Cd_w$ | Wave drag coefficient |
| $Cd_{ind}$ | Induced drag coefficient |
| $Cd_{vp}$ | Viscous pressure drag coefficient |
| $Cl$ | Lift coefficient |
| $Cl_{eq}$ | Lift coefficient at equilibrium |
| $D$ | Structural displacements |
| $F$ | Structural flexibility matrix |
| $g$ | Earth gravity acceleration |
| $H_k$ | Hessian matrix approximation by BFGS formula |
| $J$ | Optimization objective function |

| | |
|---|---|
| $j$ | Single operating condition objective function |
| $L$ | Aerodynamic loads |
| $L_a$ | Aircraft characteristic length |
| $LoD$ | $Cl_{eq}/Cd$: Lift over drag ratio |
| $M$ | Mach number |
| $m$ | Number of operating conditions |
| $n$ | Number of design variables |
| $n_s(n,p)$ | $n_s(n,p)$ : Required number of points to evenly sample a $n$-dimensional design space with a precision $10^{-p}$ |
| $p$ | Fluid static pressure |
| $R$ | Discrete form of the flow state equations |
| $S$ | Aircraft surface shape |
| $s$ | Line-search step |
| $S_{ref}$ | Aircraft reference surface |
| $T$ | Engines thrust |
| $t$ | Time |
| $t_0$ | Begin of cruise time |
| $T_f$ | Convection characteristic time on the aircraft |
| $t_f$ | End of cruise time |
| $T_w$ | Fuel consumption characteristic time |
| $T_{eq}$ | Thrust at equilibrium |
| $U_\infty$ | Aircraft air speed |
| $V$ | Aircraft air speed |
| $W$ | Flow state variables $[\overline{\rho}, \overline{\rho}\overline{\mathbf{U}}, \overline{\rho}\overline{E}, \mu_t]$ |
| $w$ | Aircraft mass |
| $X$ | Computational mesh |

**Greek letters**

| | |
|---|---|
| $\tau_{wf}$ | Non-dimensional ratio between convection and fuel consumption time scales |
| $\alpha$ | Operating condition, typically a (Mach number, Angle of Attack, Reynolds) tuple |
| $\chi$ | Design variables vector |
| $\Delta_w$ | Fuel overall consumption |

| | |
|---|---|
| $\lambda$ | Discrete adjoint vector |
| $\lambda_l$ | Laminar thermal conductivity |
| $\lambda_t$ | Turbulent thermal conductivity |
| $\overline{\phi_{\mathbf{t}}}$ | Constraint tensor due to turbulent heat flux |
| $\overline{\phi}$ | Constraint tensor due to heat flux |
| $\overline{\tau}$ | Favre averaged viscous constraint tensor |
| $\tau_{\mathbf{r}}$ | Favre stress tensor |
| $\nabla_\chi$ | Gradient vector of total derivatives with respect to $\chi$ |
| $\omega$ | Weights of an aggregate objective function |
| $\overline{\rho}$ | Favre averaged fluid mass fraction |
| $\rho$ | Fluid density |

**Shortcuts**

| | |
|---|---|
| $k$ | Optimization iteration index |

# Introduction

Air traffic has doubled between 1997 and 2012. According to the Airbus Global Market Forecast of 2013 [1], this number should double again between 2012 and 2027. 14 000 new aircraft would have to be delivered in the next ten years to sustain this growth.

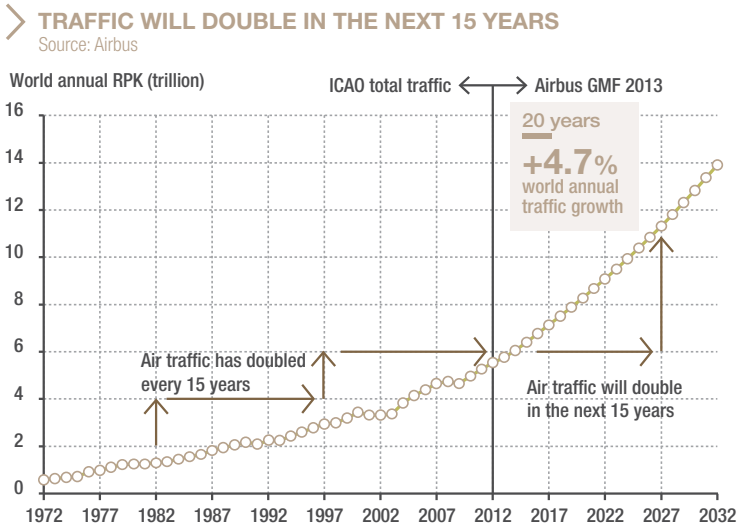



Figure 1: Air traffic growth history and forecast [1]

Aircraft fuel consumption has been reduced by 70% in the last forty years, and new progresses have been done recently. The Advisory Council for Aeronautics Research in Europe (ACARE) sets the objective of reducing further the $CO_2$ emissions by 50% between 2000 and 2020. A strong fuel price rise has been observed in the mean time, and is expected to be maintained, resulting in higher airline operating expenses. In addition to cost issues, the resulting kerosene demand increase may face production limits, according to Nygren et al. [100]. Besides, air traffic has already a non-negligible environmental impact with 2% of overall $CO_2$ emissions in 2008. Consequently, reducing fuel consumption in the next decades is a key challenge for aviation.

Continuous improvements of the existing technologies, as well as technology ruptures, will con-

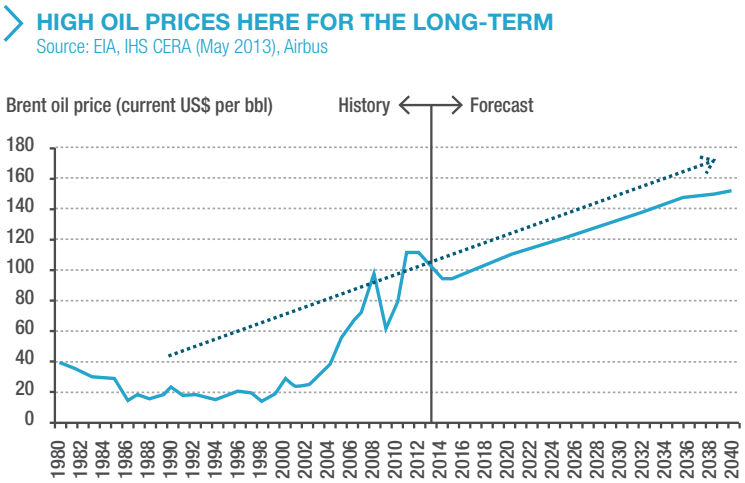



Figure 2: Oil prices history and forecast [1]

tribute to the construction of more fuel-efficient aircraft. Innovative methodologies should as well lead to better performance. For instance, numerical optimization in the fields of aerodynamics helps engineers to create more efficient wings. Optimization automates the generation of shapes and the simulation of the resulting system, using numerical physics-based models, in the aim of finding the best design.

It has now been 25 years since A. Jameson demonstrated the possibility of performing aerodynamic shape design in transonic conditions, using an automated optimization process, and in a way that is scalable to industrial cases [55]. Optimization processes require Computational Fluid Dynamics (CFD) simulation codes to predict aircraft aerodynamic performance, which are available in the industry since nearly 40 years [58]. Optimization is today less used than CFD, even when the final purpose of the simulation is to improve the shape performance. This technology is facing industrialization issues, already identified by industrial experts, such as Shahpar from Rolls Royce [121] and Johnson from Boeing commercial planes [58]. In particular, the gaps between academic demonstrators and industrial tools are not only a question of software programming, but also a matter of open scientific questions, among which the formulation of engineering problems as mathematical optimization problems. Above all, optimization has to demonstrate clear interests compared to manual design.

In its original form, aerodynamic optimization was performed for a single operating condition, i.e. for a single instant in the flight mission. Rapidly, academics realized that the resulting shapes are strongly inadapted to other flight conditions [30]. This specialization phenomena is amplified by the use of an optimization algorithm that takes advantage of the model properties. In reality, aircraft operate at an infinite set of operating conditions during a flight, such as fuel mass, altitude and Mach number, so aircraft should perform as well as possible at all of them. In addition, the important dimensions of transport aircraft make them relatively flexible, which implies that the shape is actually changing in flight, due to the combined effects of aeroelasticity and the evolution in time of operating conditions. Therefore, to obtain an aircraft shape with optimal fuel consumption on all its missions, one needs to predict how flexible shapes modifications impact the aircraft performance, for thousands of missions, which seems out of reach for human comprehension [76]. Classically, the mission performance is analyzed for a given design, and not especially for a design purpose, which avoids the latter issue. A solution may come from computers, that are well-suited to the analysis of large amount of data. Automated optimization relies on the complete translation of

the engineering design problem into mathematics. Once this step is achieved, the design strategy can be assisted by computers and only the essential information is provided to the user. For instance, adjoint sensitivity maps that can be calculated by optimization processes, highlight the areas of the shape to which the performance is particularly sensitive. This information can then be extended to multiple operating conditions, providing an essential tool towards the design for optimal mission-performance. Besides, the detection of the critical operating conditions that mostly drive the mission performance, similarly to critical load cases for structural design, can be addressed by adjoint methods used in optimization processes, as we demonstrate in the present study. Mission-optimal design and the associated fuel consumption gains may then be another key motivation for the spreading of optimization tools in daily aerodynamic design.

The present thesis focuses on the formulation of aircraft mission performance fuel minimization into a mathematical robust optimization problem whose cost of resolution is not prohibitive. Demonstrations of optimizations on academic cases and a civil transport aircraft are provided as examples. In Part I, the fuel consumption equations on a mission are written, and analyzed in a numerical optimization perspective, which leads to the choice of an optimization strategy. The resulting robust and multi-objective optimization problem raises open mathematical questions, that are addressed in Part II, with demonstrated theorems. Part III shows that the theory proposed in Part II is applicable to airfoil and aircraft wing aerodynamic design.

## Associated communications

- An Adaptive Formulation for Robust Parametric Optimization, Gallard F. Mohammadi B., Montagnac M., Meaux M., Journal of Optimization Theory and Applications, 2014)
- Aerodynamic aircraft design for mission performance by multipoint optimization, Gallard F., Meaux M., Montagnac M., Mohammadi B., proceedings of the 21st AIAA applied CFD conference, San Diego, USA, 2013
- Aerodynamic aircraft design for mission performance by multipoint optimization AIAA Journal. Gallard F., Meaux M., Montagnac M., Mohammadi B., (in preparation)
- Non-uniform time sampling for multiple-frequency harmonic balance computations, Guédenay T., Gomar A., Gallard F., , Sicot F., Dufour G., Puigt G., Journal of Computational Physics, 2013
- Constrained Adjoint-Based Multipoint Aerodynamic Shape Optimization of a Turbine Nozzle Guide Vane, Montagnac M., Gallard F., Montanelli H., Journal of Turbomachinery (in preparation)

# Part I

# Formulation of the optimization problem

# Table of Contents

# CHAPTER 1

## Aircraft mission performance

***Summary*** In this chapter, the problem of aircraft aerodynamic optimization for its mission-performance is described. In the first section, equations for the fuel overall consumption are detailed. In the context of gradient-based optimization, these equations are derived with respect to design variables. The resulting expression leads to a robust optimization formulation based on the weighted sum of lift over drag ratios at multiple operating conditions, which raises the question of the operating conditions to be included in the optimization problem. In the second section, the question of the aeroelasticity in the context of aerodynamic design optimization for mission performance is discussed. As a result, taking fluid-structure interaction into account in a multiple operating conditions wing optimization procedure is required.

***Résumé*** Dans ce chapitre, la question de l'optimisation d'un avion pour une performance maximale sur une mission est adressée. Dans la première section, les équations régissant la consommation totale sont établies. Dans un contexte d'optimisation par méthode de gradient, ces mêmes équations sont dérivées par rapport aux variables de design. Les expressions résultantes mènent à la formulation d'un problème d'optimisation robuste basée sur la pondération d'objectifs. Celui-ci soulève la question des conditions d'opérations à inclure dans le problème d'optimisation. Dans une seconde section, nous décrivons les impacts du phénomène d'aéro-élasticité sur l'optimisation de la performance aérodynamique sur une mission. Il en résulte la nécessité de la prise en compte de l'interaction fluide-structure dans une procédure d'optimisation de voilure en de multiples conditions d'opération.

## 1.1 From mission fuel consumption to a robust optimization formulation

### 1.1.1 Mission fuel overall consumption

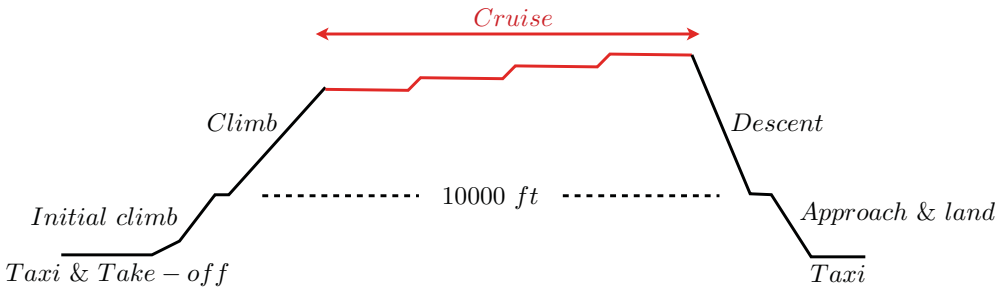



Figure 1.1: Typical civil transport aircraft mission profile.

As shown in Fig. 1.1, the aircraft mission is composed of successive phases among which: taxi, take-off, climb, cruise, descent, approach, land and a taxi phase again. Consequently, during a flight, an aircraft encounters a wide range of operating conditions. For example the Mach number, Reynolds number, and angle of attack vary and have then to be taken into account during aerodynamic shape design. Figure 1.2 shows that the performance of the aircraft is strongly influenced by these operating conditions. While the aircraft mass decreases due to fuel burn, the lift force is adjusted, so as the altitude and then the aerodynamic efficiency given by the lift over drag ratio (LoD), varies. Aircraft altitude is usually constrained to discrete values called Flight Levels (FL), measured in hundreds of feet, for air traffic control reasons. The engine efficiency, measured in terms of specific fuel consumption ($C_s$), also depends on the aircraft operating conditions.

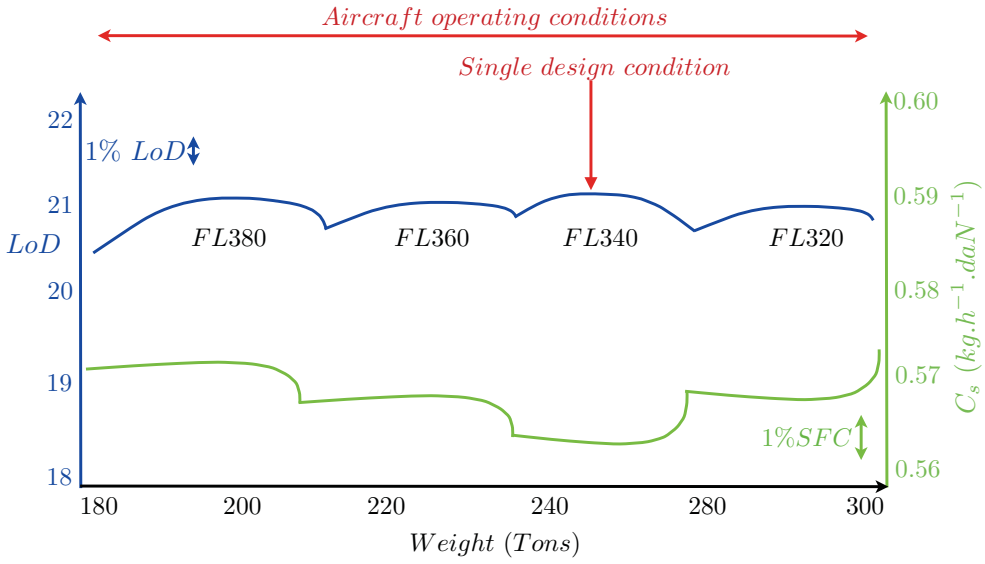



Figure 1.2: Aircraft operating conditions encountered during a mission and the associated aerodynamic and engine performance.

Equations for the minimization of the fuel consumption can be written considering all the phases. For the sake of simplicity we will only consider cruise in the following demonstration as most of the fuel is burnt in cruise conditions [2], except for a short-range aircraft. In addition, the conclusion we draw does not rely on the accuracy of the mission consumption calculation, but on the general form of the equations.

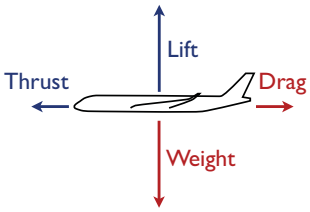



Figure 1.3: Aircraft balance of forces in cruise

In steady flight at constant altitude, the lift compensates the weight and the thrust compensates the drag as displayed in Fig. 1.3. In aerodynamics, forces are scaled by dynamic air pressure and wing reference surfaces to obtain non-dimensional lift and drag coefficients $Cl$ and $Cd$. Comparison between different shapes is made easier with this convention.

$$\begin{cases} \frac{1}{2}\rho(\alpha(t))Cl(\chi,\alpha(t))S_{ref}V^2 = w(t,\chi)g(\alpha(t)), \\ \frac{1}{2}\rho(\alpha(t))Cd(\chi,\alpha(t))S_{ref}V^2 = T(\chi,\alpha(t)). \end{cases} \tag{1.1}$$

Lift and drag coefficients are influenced by the aircraft shape and therefore depend on design variables. Mass decreases in time due to fuel burn, and fuel burn is influenced by the shape, so mass is both a function of time and design variables. Required thrust is a result of all the previously cited phenomena, so is a function of time, operating conditions and design variables. As seen in Fig. 1.1, the aircraft operating conditions are a function of time.

By combination of Eqs (1.1):

$$T(\chi,\alpha(t))\frac{Cl(\chi,\alpha(t))}{Cd(\chi,\alpha(t))} = w(t,\chi)g(\alpha(t)). \tag{1.2}$$

Also, we define $Cl_{eq}$ the required lift coefficient to compensate the weight:

$$Cl_{eq}(\chi,\alpha(t),w(t,\chi)) = \frac{2w(t,\chi)g(\alpha(t))}{\rho(\alpha(t))S_{ref}V^2}, \tag{1.3}$$

and $T_{eq}$ the required thrust to compensate the drag:

$$T_{eq}(\chi,\alpha(t),w(t,\chi)) = \frac{Cd(\chi,\alpha(t))w(t,\chi)g(\alpha(t))}{Cl_{eq}(\chi,\alpha(t))}. \tag{1.4}$$

For a jet engine, at first order, the fuel mass flow rate is proportional to the thrust. The proportionality coefficient $C_s$ is named fuel specific consumption.

$$\frac{dw(t,\chi)}{dt} = -C_s(\chi,\alpha(t))T_{eq}(\chi,\alpha(t),w(t,\chi)). \tag{1.5}$$

We define now the $LoD = Cl_{eq}/Cd$ factor as the lift-to-drag ratio, and the Breguet equation is obtained in Eq. (1.6). Lift-to-drag ratio is a major aerodynamic performance indicator for an

aircraft as it drives the fuel consumption at first order. We consider $LoD$ as a function of the mass in the following, since $Cl_{eq}$ is a function of the mass through Eq. (1.3), and we keep only one equation from the system of Eq. (1.1).

$$\frac{dw(t,\chi)}{dt} = -\frac{C_s(\chi,\alpha(t))g(\alpha(t))}{LoD(\chi,\alpha(t),w(t,\chi))}w(t,\chi). \tag{1.6}$$

Figure 1.4 plots typical mass evolution in time for different values of $\chi$. Constant values for the $LoD$ and $C_s$ coefficients are taken.

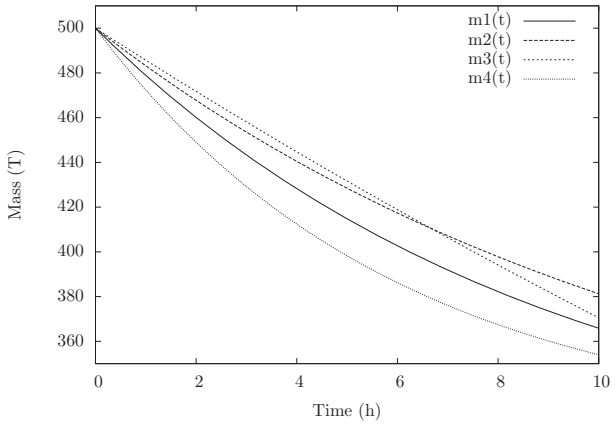



Figure 1.4: Typical evolution of aircraft mass over time for different designs

Aircraft mass can be decomposed into fuel mass $w_f(t)$, payload $w_p$ and structural mass $w_s(\chi)$. Eq. (1.6) is a first-order differential equation that can be integrated once on the time variable, over the range $[t_0, t_f]$. The fuel overall consumption is the difference between the beginning and the end of cruise $\Delta_w = w(t_0) - w(t_f)$.

$$\Delta_w(\chi) = -\int_{t=t_0}^{t=t_f} \frac{dw(t,\chi)}{dt}dt = \int_{t=t_0}^{t=t_f} \frac{C_s(\chi,\alpha(t))(w_f(t)+w_s(\chi)+w_p)g(\alpha(t))}{LoD(\chi,\alpha(t),w_f(t)+w_s(\chi)+w_p)}dt. \tag{1.7}$$

Equation (1.7) shows that multiple disciplines affect the overall aircraft performance and that problem is multi-disciplinary.

### 1.1.2 Aerodynamic optimization for multiple missions fuel consumption

When a high number of design variables are involved such as in 3D aircraft configurations, gradient-based algorithms prove to be the most efficient approach (see Chapter 2), so the derivative of the objective function with respect to the design variables has to be computed. The vector constituted by the total derivative of any scalar quantity with respect to the design variables will be called "gradient" in the following, and the associated operator is noted $\nabla_\chi$. Assuming that the function $\Delta_w$ of Eq.(1.7) is piecewise continuously differentiable, and bounded on $t \in [0, t_f]$, which is the case

in practice, then the Leibniz Theorem allows to swap derivation and integration operators:

$$\begin{aligned}\nabla_\chi \Delta_w(\chi) = &- \int_{t=t_0}^{t=t_f} \frac{C_s(\chi,\alpha(t))(w_f(t)+w_s(\chi)+w_p)g(\alpha(t))}{LoD(\chi,\alpha(t),w_f(t)+w_s(\chi)+w_p)^2} \\ &\left(\frac{\partial LoD(\chi,\alpha(t),w_f(t)+w_s(\chi)+w_p)}{\partial\chi} + \frac{\partial LoD(\chi,\alpha(t),w_f(t)+w_s(\chi)+w_p)}{\partial w_s}\ \nabla_\chi\ w_s(\chi)\right) dt \\ &+ \int_{t=t_0}^{t=t_f} \frac{(w_f(t)+w_s(\chi)+w_p)g(\alpha(t))}{LoD(\chi,\alpha(t),w_f(t)+w_s(\chi)+w_p)}\ \nabla_\chi\ C_s(\chi,\alpha(t))dt \\ &+ \int_{t=t_0}^{t=t_f} \frac{C_s(\chi,\alpha(t))g(\alpha(t))}{LoD(\chi,\alpha(t),w_f(t)+w_s(\chi)+w_p)}\ \nabla_\chi\ w_s(\chi)dt.\end{aligned} \tag{1.8}$$

Equation (1.8) shows that gradients of each discipline's main performance criteria contribute to the performance total gradient. The meaning of these terms is detailed here:

- The term $\frac{\partial LoD(\chi,\alpha(t),w_f(t)+w_s(\chi)+w_p)}{\partial\chi}$ is the derivatives of the lift over drag ratio with respect to the design variables, for a given structure, engine and mission.
- The term $\frac{\partial LoD(\chi,\alpha(t),w_f(t)+w_s(\chi)+w_p)}{\partial w_s}\nabla_\chi\ w_s$ is the derivatives of lift over drag with respect to the design variables, due to the impact of structural mass variation on the lift coefficient that ensures the balance with the weight (Eq. (1.1)).
- The term $\nabla_\chi\ C_s(\chi,\alpha(t))$ is the derivative of the engine performance with respect to the design variables.
- The term $\nabla_\chi\ w_s(\chi)$ is the derivative of the structural mass with respect to the design variables.

The present study focuses on the aerodynamic design optimization, so does not consider the impacts of design variables changes on the other disciplines. Therefore only the pure aerodynamic term $\frac{\partial LoD(\chi,\alpha(t),w_f(t)+w_s(\chi)+w_p)}{\partial\chi}$ is kept and the equation is simplified to Eq. (1.9). Also the engine design is not considered, $C_s$ is not a function of $\chi$ any more. This assumption implies to choose design variables that have a high influence on the aerodynamic discipline relatively to its influence on other disciplines, by comparison of the terms in Eq. (1.8) for instance, so that the pure aerodynamic optimization decreases the multidisciplinary $\Delta_w$ function. When it is not possible, or when the aerodynamic optimum is known to degrade another discipline's key design feature, constraints have to be added so as not to degrade the multidisciplinary performance. For instance, an elliptic lift repartition on a wing leads to minimal induced drag, while it creates a high wing root bending moment and leads to a high structural mass, and degrades $\Delta_w$. Constraining the root bending moment in the pure aerodynamic optimization is an option, but the question of its optimal value is a multidisciplinary problem. An aggregation of single-discipline approaches consequently requires to add constraints to the optimization problem to manage multidisciplinary coupling variables and may then over constrain the aircraft design problem so lead to a higher fuel consumption than a fully multidisciplinary approach that lets interaction variables evolve. However, a Multi-disciplinary Design Optimization (MDO) process relies on single disciplines methods and tools and the present work is in this sense a contribution of the aerodynamic discipline to the aircraft MDO process.

Under the single-discipline assumption we obtain:

$$\nabla_\chi \Delta_w(\chi) = - \int_{t=t_0}^{t=t_f} \frac{C_s(\alpha(t))w(t)g(\alpha(t))}{LoD(\chi,\alpha(t),w(t))^2} \nabla_\chi LoD(\chi,\alpha(t),w(t))dt. \tag{1.9}$$

In an aerodynamic optimization process based on numerical simulation of steady flows, the calculations are performed at a given angle of attack, mass and altitude, so a time-discretization of

Eq. (1.9) is required. A Riemann sum approach is proposed, with uniform time discretization in $n_t$ intervals of length $\delta t_i$ such that $\delta t_i = \frac{t_f - t_0}{n_t}$.

$$\nabla_\chi \Delta_w(\chi) = -\sum_{i=1}^{n_t} \frac{C_s(\alpha(t_i))w(t_i)g(\alpha(t_i))}{LoD(\chi, \alpha(t_i), w(t_i))^2} \nabla_\chi LoD(\chi, \alpha(t_i), w(t_i, \chi))\delta t_i. \tag{1.10}$$

Finally, a set of scalar positive functions $\omega_i(\chi)$ can be defined such that:

$$\nabla_\chi \Delta_w(\chi) = -\sum_{i=1}^{n_t} \omega_i(\chi) \nabla_\chi LoD(\chi, \alpha(t_i), w(t_i)). \tag{1.11}$$

It is interesting to note that when $n_m$ missions of an aircraft fleet are considered, the total fuel consumption is the sum of consumptions on each mission.

$$\Delta_{w\ fleet}(\chi) = -\sum_{p=1}^{n_m} \int_{t=t_{0,p}}^{t=t_{f,p}} \frac{C_s(\chi, \alpha(t))w(t)g(\alpha(t))}{LoD(\chi, \alpha(t), w(t))} dt, \tag{1.12}$$

and similarly for the gradients:

$$\nabla_\chi \Delta_{w\ fleet}(\chi) = -\sum_{p=1}^{n_m} \sum_{i=1}^{n_t} \omega_{i,p}(\chi) \nabla_\chi LoD(\chi, \alpha(t_{i,p}), w(t_{i,p}, \chi)). \tag{1.13}$$

So there exists a set of operating instants $(\bar{t}_1, \ldots, \bar{t}_k, \ldots, \bar{t}_{n_m \times n_t})$ and a set of weights $(\bar{\omega}_1, \ldots, \bar{\omega}_k, \ldots, \bar{\omega}_{n_m \times n_t})$ such that:

$$\nabla_\chi \Delta_{w\ fleet}(\chi) = -\sum_{k=1}^{n_m \times n_t} \bar{\omega}_k(\chi) \nabla_\chi LoD(\chi, \alpha(\bar{t}_k), w(\bar{t}_k)). \tag{1.14}$$

Eq.(1.14) and Eq. (1.11) are similar. A multi-mission fuel consumption gradient only differs from a single-mission one by the operating conditions to be considered and the associated weights. The formulation strategy for multiple mission optimization problems can then be identical to the single-mission ones.

Eq. (1.11) can be viewed as an aggregate objective function at selected operating conditions with non-constant weights. Steuer [126] interprets such a quantity as a non-linear decision maker's preference function for the weighted-sum method. From the aerodynamic optimization point of view, the question that rises is how to choose the operating conditions and weights so that the optimization process converges to an interesting engineering solution. However, the set of missions the aircraft is going to fly is not known at the time of the aerodynamic design. Expected missions profiles can be estimated using data from a similar aircraft which is in operation [76]. But for a given aerodynamic design, the optimal mission profile changes, so it is also the case for the weights and operating conditions to be included in the multipoint optimization. Therefore two imbricated and coupled optimization processes should be used to resolve the coupling between the mission and the aerodynamic performance optimization.

For a single-discipline aerodynamic optimization, weights that do not depend on other disciplines are required, their specification should only rely on pure aerodynamic analysis. In this way, the multipoint formulation does not require mission and aircraft performance models in order to explicitly compute the weighting function $\omega_i(\chi)$. Operating conditions and mass conditions are translated into aerodynamic variables: Mach number, Reynolds number and lift coefficient. A Pareto frontier based weighting estimation to obtain a good aerodynamic compromise is proposed in the next chapters. It is based on pure aerodynamic physical considerations. Such an approach

aims at maximizing the lift over drag function $\chi \to LoD(\chi, \alpha, w)$ for a continuous range of operating conditions $\alpha$ and masses $w$.

We have now turned the aerodynamic mission-performance optimization problem into an aerodynamic polar optimization. Figure 1.5 displays the performance of the RAE2822 airfoil for a range of Mach numbers and angle of attack. We aim at finding a formulation for the control of such a surface though an optimization problem, in other words to give a mathematical sense to the "maximization of such a surface", at minimal computational cost.

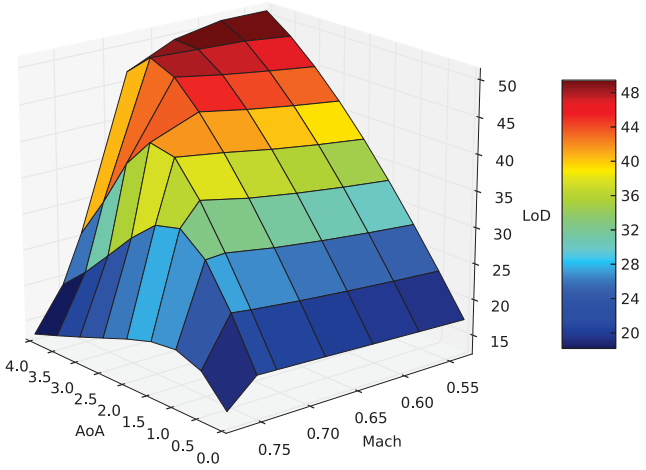



Figure 1.5: Lift over drag polar of the RAE2822 airfoil, depending on angle of attack and Mach number

## 1.2 Aeroelasticity

In the previous section, mission performance aerodynamic optimization has been shown to require flow calculations around the aircraft for a set of operating conditions. A closer look is taken at these numerical simulations, and in particular at the fluid-structure interaction. Operating conditions impact the aero-elastic interactions, so the aerodynamic shape depends on the operating conditions, which has an impact on performance. Aeroelasticity is therefore a key point of multiple operating condition aerodynamic design.

### *1.2.1 The aeroelastic phenomenon*

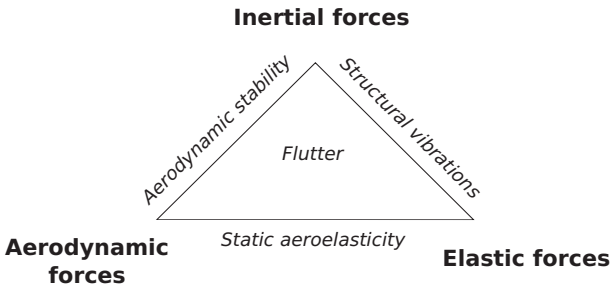



Figure 1.6: Collar's aeroelasticity triangle

Aeroelasticity addresses the effect of combined aerodynamic, inertial and elastic forces on a system. The "Collar triangle" [23] summarizes the possible interaction cases in Fig. 1.6. Aerodynamic forces are applied to the aircraft structure that deforms in response. Structural deformations modify the aircraft surface, so the air flow and therefore the aerodynamic forces. Inertial forces can also apply on the structure when unsteady phenomena such as vibrations or flutter occur. When dynamics effects are negligible, the interaction called static aeroelasticity can lead to an equilibrium or divergence. Figure 1.7 describes the situation.

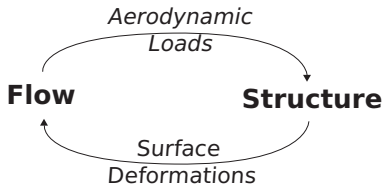



Figure 1.7: Disciplines and coupling variables in static aeroelasticity

Only static aeroelasticity is considered for aircraft performance. Flutter and structural vibrations have to be avoided with margins during the design. In the case of properly designed aircraft wings, static aeroelasticity reaches an equilibrium. Typical twist and bending deformations due to aeroelasticity can be seen in Figs. 1.8 and 1.9.

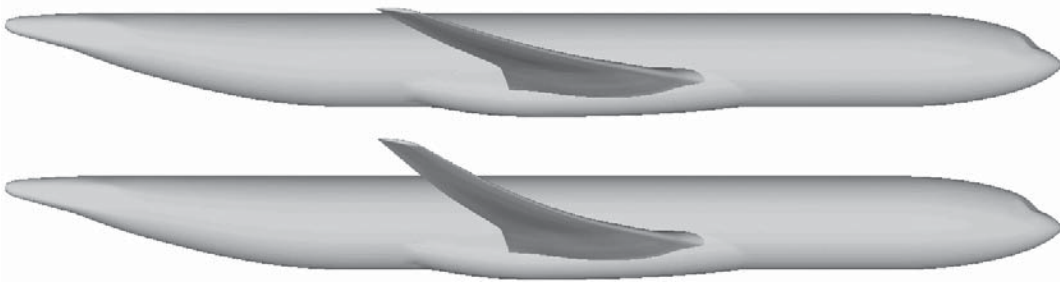


Figure 1.8: Side view of the rigid XRF-1 (top) and typical deformations due to aeroelasticity (bottom)

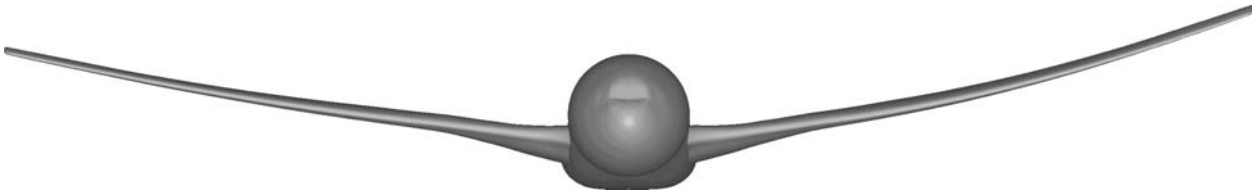


Figure 1.9: Front view of the rigid XRF-1(left) and typical deformations due to aeroelasticity (right)

### *1.2.2 Some consequences of fluid-structure interaction on multiple operating conditions performance*

Twist modifications impact the local angle of attack of the airfoils, as shown in Fig. 1.10, and in a different way depending on the operating conditions, as shown in Fig. 1.11. As a consequence,

the lift repartition on the wing changes depending on the operating conditions, so as the drag induced by the lift [110, 109]. The optimal lift repartition for a single flight condition is known to be the elliptical one [110, 109]. One can take a look at the single point and multipoint lift repartition optimizations in Chapter 4.4 for more details on that subject. For a flexible wing, it is then impossible to have an optimal span lift repartition that minimizes induced drag at all flight conditions [35, 125]. A compromise is then required. Also, airfoils local angles of attacks depend on the wing twist so they are modified by aeroelastic deformations. Drag depends on the angle of attack of each wing section, the wing drag is then also impacted in this way by aeroelasticity. Therefore, an airfoil design that is robust to angle of attack perturbations is required.

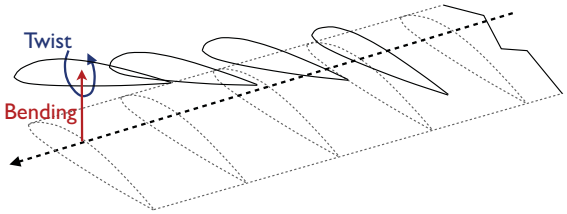
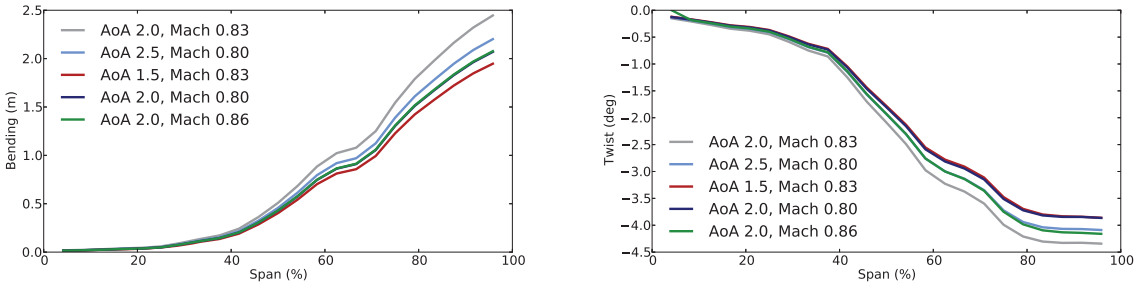



Figure 1.10: Schematic main wing aeroelastic deformations




(a) Bending deformations

(b) Twist deformations

Figure 1.11: Example of deformations of the XRF-1 wing due to aeroelasticity, at multiple operating conditions, from coupled fluid-structure CFD computations

Under the hypothesis that the structural deformations $D$ are small and that the deformations depend linearly on the loads $L$ through the structural stiffness matrix $F$, one can write:

$$D = FL. \tag{1.15}$$

The aerodynamic shape $S$ is a function of the design variables and the structural deformations:

$$S = S(\chi, D). \tag{1.16}$$

The loads are due to weight, inertial and aerodynamic forces. We obtain:

$$L = L(w(t), W(S(\chi, D), \alpha(t))), \tag{1.17}$$

where $W$ is the aerodynamic field, that depends on the aerodynamic surface and the operating conditions. The coupling variables between aerodynamics and structure are loads $L$ and displacements $D$. In steady cases, Eq. (1.15), representing the structural residuals, to be solved in addition

to flow state equations, becomes:

$$D - FL(w(t), W(S(\chi, D), \alpha(t))) = 0. \tag{1.18}$$

Classically, Eq. (1.18) is solved using a fixed-point iteration assuming constant time. From a starting shape, a given amount of flow solver iterations are performed, loads are then computed from the pressure field, structural equations are then solved which leads to a deformed structure and a new aircraft shape on which the flow has to be updated by the flow solver. The whole coupling procedure is stopped when both flow and structural residuals are sufficiently decreased to reach the required accuracy. The approach is very efficient and costs only about 20% more than a non-aeroelastic calculation. A final deformed surface and the flow around it are obtained, aircraft surface and structural displacements are then implicit functions of the design variables, the operating conditions and the mass; respectively $S = S(\chi, D(w(t), \chi, \alpha(t)))$ and $D = D(w(t), \chi, \alpha(t))$. In Eq. (1.7), the dependency of the lift over drag ratio to the design variable is due to the shape, the effect of aeroelastic deformations is added in Eq. (1.19) expressing that the shape is a function of aeroelastic deformations.

$$\Delta_w(\chi) = \int_{t=t_0}^{t=t_f} \frac{C_s(\chi, \alpha(t)) w(t, \chi) g(\alpha(t))}{LoD(S(\chi, D(w(t, \chi), \chi, \alpha(t))), \alpha(t), w(t, \chi))} dt. \tag{1.19}$$

In a single operating condition optimization, the aerodynamic design can be performed on the loaded shape. It means that to remove the coupling between aerodynamics and structure, the loads $L$ are a consequence of the rigid aerodynamic shape. As a result, the structural design has to be performed with target deformations $D_t$ for the design point loads $L_t$. For a manufactured shape called "jig" shape $S_0$, the deformed shape at design point $S_{flight}$ has to be such that $S_{flight} = S_0 + D_t$, ensuring structure state equation $D_t = FL_t$. Flexibility and jig shape are outputs of this process.

In a mission-oriented approach, the aerodynamic shape becomes a function of the operating conditions through the term $S(\chi, D(w(t), \chi, \alpha(t)))$ in Eq. (1.19). Structural flexibility is an input of the process since loads and displacements depend on the operating condition. Figure 1.2 shows that the aerodynamic performance varies when the operating condition varies. This fact opens the opportunity of taking advantage of passive aeroelastic shape adaptation. Active control has already been proposed to address this problem, with techniques such as variable camber airfoils [35, 125]. Passive control is complementary as it requires no energy or dedicated systems.

### 1.2.3 On the unsteadiness of the problem

Through the aeroelastic coupling equation (1.18), the aircraft shape is a function of time. Therefore the aeroelastic interaction calculation during a mission is an unsteady problem. This raises the question of how to consider the flow for a given shape. Are the shape changes fast enough to make the information propagation time in the computational domain non-negligible? Or is the fluid-structure equilibrium reached much faster than the characteristic time scale of mass variation, which means that two different time scales prevent the strong interaction?

To answer that question, we compare two time scales: the characteristic time of the aircraft mass variation $T_w$, and the characteristic time of the flow adaption to shape changes $T_f$. We approximate the last one to the time that a perturbation of the flow takes to move from the nose of the airplane to the tail by convection. For transonic flight, this time scale is also close to the sound propagation time as the Mach number is close to 1.

$$T_f = \frac{L_a}{M.c}. \tag{1.20}$$

In order to estimate a characteristic time for the mass variation, we take Eq. (1.7), and assume an average and constant lift over drag ratio $\overline{LoD}$, and do the same for the fuel specific consumption

$\overline{C_s}$ and gravity acceleration $\overline{g}$. Then the integral can be calculated analytically:

$$\overline{\Delta_w}(t) = w_0 \; exp\left(-\frac{\overline{C_s}\overline{g}}{\overline{LoD}}t\right). \tag{1.21}$$

Equation (1.21) provides a characteristic time for the mass variation.

$$T_w = \frac{\overline{LoD}}{\overline{g}\overline{C_s}} \tag{1.22}$$

The ratio between the two time scales is:

$$\tau_{wf} = \frac{T_f}{T_w} = \frac{L_a\overline{g}\overline{C_s}}{M.c.\overline{LoD}}. \tag{1.23}$$

Finally, for a standard civil transport aircraft we have typically: $\overline{LoD} = 20$, $\overline{g} = 9.8 \; m.s^{-2}$, $\overline{C_s} = 10^{-5} \; kg.N^{-1}.s^{-1}$, $M = 0.85$ , $c = 300 \; m.s^{-1}$ and $L_a = 80 \; m$. Consequently $\tau_{wf} \approx 10^{-6} \ll 1$. In our case, the dynamics of the flow is much faster than the dynamics of the structure driven by fuel consumption. Flow calculations can then assume steadiness of the problem.

# CHAPTER 2

## Multiple operating conditions strategies for aerodynamic design optimization

***Summary*** In this chapter, we present a review of approaches addressing aerodynamic shape optimization and robustness to operating conditions perturbations. In a first part, optimization algorithms classically used in the aerodynamic field are listed. Given their respective advantages, the high number of design parameters, and the objective function properties of our aerodynamic high-speed shape drag minimization problem, the class of gradient-based algorithms is elected. In a second part, robust and multi-objective optimization formulations in engineering are explored. Besides, the links between robust and multipoint optimization are explained. The specific properties of our problem led to the choice of the weighted sum formation. However the weighted sum formulation has a major theoretical drawback that is dismissed in Part II.

***Résumé*** Dans ce chapitre, nous effectuons une revue des travaux portant sur l'optimisation de formes aérodynamiques robustes aux perturbations des conditions d'opération. Dans une première section, les algorithmes d'optimisation classiquement utilisés en aérodynamique sont listés. Etant donné leurs avantages et inconvénients respectifs, ainsi que le nombre de variables de design et propriétés de notre fonction objective en aérodynamique, la classe des algorithmes par gradient est choisie. Une seconde partie aborde les formulations en optimisation robuste et multi-objective en ingénierie. Le lien entre l'optimisation robuste et l'optimisation multi-objectif est expliqué. Une fois encore, les propriétés de notre problème ont mené au choix d'une formulation de somme pondérée d'objectifs. Néanmoins, la pondération d'objectif a un défaut théorique majeur qui sera levé dans la Partie II.

## 2.1 Common optimization algorithms in aerodynamic shape optimization

The literature is very rich on the subject of optimization algorithms and strategies, so we wish not to be exhaustive. Because there is "no free lunch theorem for optimization" [141], all optimization algorithms have the same average performance on all possible problems, election of an algorithm has to rely on specific properties of the problem to be solved. We propose here arguments for the selection of the best suited optimization algorithm to our high-dimensional multipoint 3D aerodynamic shape drag minimization problem.

### *2.1.1 Evolutionary algorithms*

Using a classical single-objective optimization algorithm to solve a multi-objective problem requires to aggregate the multiple objectives into a single composite function. The approach is also called scalarization. The main drawbacks of aggregation is the choice of a weighting strategy between the objectives [24]. Alternative approaches that do not rely on scalarization were also proposed. For instance the minmax optimization optimizes one objective at a time and is theoretically equivalent to the weighted sum. Li et al. [138] proposed an adaptive min max approach for robust multi Mach optimization. Probabilistic approaches have also been employed [50].

On the other hand, some optimization algorithms are multi-objective by construction. Evolutionary algorithms and Genetic Algorithms (GA) [41] handle populations of design variables with associated performances. Since they propose a set of possible solution of multi-objective problems they could consequently be adapted to our problem. The main mechanisms of GA are inspired from evolution: mutation, reproduction, recombination and selection. The population evolves by successive action of the above operators, usually involving randomness for at least mutation and selection. Multiple variants of algorithms were created by variations on the way these operators are built.

In the aerodynamic field for instance, Marco et al. [83] performed Euler airfoil optimization using GA, a Bezier parametrization of the shape, and obtained a set of Pareto-optimal solutions. The aim of the study was to find compromise solutions between high lift and low drag. Vicini et al. also performed inverse and direct airfoil design using a multi-objective genetic algorithm [137].

Duvigneau et al. performed aerodynamic optimization of a transonic wing in multiple Mach conditions [31]. A Radial Basis Function was used to create a meta-model of the drag depending on the Mach number to perform a Monte-Carlo analysis, and compute the probability density function (PDF) of the drag, given a gaussian PDF of the Mach number. Particle Swarm Optimization, which is an Evolutionary algorithm, was used to drive the optimization process. The lift constraint deals a particular issue as it was not achieved for all the points used to compute the PDF.

Since populations of solutions and randomness are key features, GA require a large number of objective function evaluations to converge. The complexity in terms of evaluations grows at least with the square of the design variables number [79]. For instance, the NSGA2 algorithm scales with the number of objectives times the square population size, and the population size scales linearly with the number of design variables [26]. For a representative 3D CFD optimization problem with 5 objectives and a hundred design variables, this represents approximately 50 000 function evaluations, which is prohibitive in real-life applications using physics-based expensive models.

### 2.1.2 Meta-models

An optimization history represents a database of function evaluations for multiple design vectors. Such global information about the objective can be exploited to detect potentially interesting regions of the design space. A natural strategy for that is to build models of the objectives and constraints, called meta-models since they are models of functions that are already modeling the behavior of a system. If the design vector set finely samples the design space, these meta-models can be global, also since they are cheap to evaluate compared to the original functions they may enable the use of global optimization strategies.

For instance, Gratton et al. developed a derivative-free algorithm based on a polynomial meta-model of the function, with trust region to control the confidence on the model. The Bounds Constrained Derivative-Free Optimization (BCDFO) algorithm [43] was benchmarked on airfoil aerodynamic optimization problems and behaves well for a moderate number of design variables. Peigin et al. [103] used a local polynomial meta-model, on which GA algorithm are used, and benchmarked against the multipoint profile optimization method from Li et al[138, 75].

Similarly, Artificial Neural Networks can also be used for interpolation of data coming from CFD, and then hybridized with GA to find a global minimum, see the work by Visonneau et al. [32].

Chung et al. [22] used a Cokriging approach to enhance the Krigging meta-model with a gradient information coming from adjoint. The Cokriging approach drastically improves the surface quality, however maintaining a global accuracy of the surface is out of reach for more than 4 design variables [72]. However, a local refinement can be used during optimization to moderate the computational cost by avoiding to refine the models in regions of the design space that are unlikely to contain the minimum. A coarse response surface can be built and refined in the areas that are promising for the purpose of optimization. A higher number, but still moderate, of design variables can then be successfully used in an aerodynamic optimization process [73].

### 2.1.3 Gradient-based algorithms

The use of derivatives and tangents to find functions extrema was already proposed by P. Fermat in the 17th century [128], and J.L Lagrange in the 18th century with the calculus of variations. Basing an iterative process on gradients to decrease a function value and reach a minimum is natural, since tangent is a key information about local variations of a function. Gradient-based algorithms follow a one-dimensional path in the $n$-dimensional design space, driven by the $n$-dimensional gradient information. It is a key property, as opposed to GA and meta-model based approaches that explore the design space. Therefore, only local convergence is guaranteed, but at a moderate cost.

To give an idea of the mechanisms involved in gradient optimization, a Taylor expansion of the objective function is made. For all perturbation direction $\mathbf{d} \in \mathbb{R}^n$, $||\mathbf{d}|| = 1$, and for all perturbation step lengths $s \in \mathbb{R}$, we have:

$$j(\chi + s\ \mathbf{d}, \alpha) = j(\chi, \alpha) + s\ \langle \mathbf{d}, \nabla_\chi j(\chi, \alpha) \rangle + \mathbf{O}(s^2), \tag{2.1}$$

where $\langle , \rangle$ is the dot product.

Equation (2.1) gives a first order estimation of function variation $\langle \mathbf{d}, \nabla_\chi j(\chi, \alpha) \rangle$ for any t-normed perturbation in the direction $D$. The steepest descent algorithm takes a step $\mathbf{d} = -s \nabla_\chi j(\chi, \alpha)$ that is collinear to minus the gradient so that $\langle \mathbf{d}, \nabla_\chi j(\chi, \alpha) \rangle < 0$, and aims to obtain $j(\chi + s\ \mathbf{d}, \alpha) < j(\chi, \alpha)$. The step length $s$ can be either fixed or determined using a line-search algorithm. However, it is common to all coordinates of the step. For non-isotropically

curved objectives, this leads to poor convergence rates, and many objective function evaluations.

To overcome the issue, second order methods use the curvature information. By Taylor expansion at second order, we have:

$$j(\chi + s\ \mathbf{d}, \alpha) = j(\chi, \alpha) + s\ \langle \mathbf{d}, \nabla_\chi j(\chi, \alpha) \rangle + \frac{s^2}{2} \mathbf{d}^T \frac{d^2 j(\chi, \alpha)}{d\chi^2} \mathbf{d} + \mathbf{O}(s^3). \tag{2.2}$$

Since the Hessian matrix $\frac{d^2 j(\chi,\alpha)}{d\chi^2}$ is expensive to compute, an approximation of it can be iteratively built from the optimization history. The BFGS (Broyden Fletcher Goldfarb Shanno) approximation is a typical case. Quasi-Newton approaches with Hessian matrix approximation are very efficient at minimizing functionals like $\chi \to -LoD(\chi)$ [13, 89], and are extensively used in the present thesis. The DOT [82] and l-bfgs-b [145] libraries are available implementations of BFGS based algorithms.

Some gradient methods can handle general constraints, for instance the CONMIN [135] methods searche for descent direction that is also feasible using gradients from both constraints and the objective. The SLSQP [68] algorithm extends the methodology of the l-bfgs-b algorithm to general constraints, by successive minimizations of a quadratic model under linearized constraints. A review of aerodynamic optimization studies using gradient-based algorithms is proposed in the next sections.

In the next paragraph, a closer look is taken at a specific Quasi-Newton algorithm, as an illustrative example of quasi second-order methods. Its iteration scheme and properties are also used in the next part on robust optimization.

#### *The L-BFGS-B algorithm*

Bounds constrained limited memory BFGS [145] algorithm is used in the next chapters, the algorithm updates the design variables using the gradient of the function but not the function value itself. This feature has to be kept in mind when formulating the optimization problem, and will be used in Part II in the GSA approach that focuses on the gradients of the objectives. One iteration of the algorithm is sketched here:

1. Define the local quadratic model: $m_k(\chi) = J(\chi_k) + \nabla_\chi J(\chi_k)(\chi - \chi_k) + \frac{1}{2}(\chi - \chi_k)^T H_k (\chi - \chi_k)$
2. Project the steepest descent direction $-\nabla_\chi J(\chi)$ onto the feasible region: $\chi(t) = P(\chi_k - t\nabla_\chi J(\chi))$ where $P(\chi_i) = \chi_i$ if and only if $l_i \leq \chi_i \leq u_i$ otherwise $P(\chi_i) = l_i$ if $\chi_i < l_i$ or $P(\chi_i) = u_i$ if $\chi_i > u_i$
3. Along this direction, find the first local minimizer of the quadratic model: $\chi^c = argmin\ m_k(\chi(t))$, freezing coordinates at active bounds. Since $H_k$ is positive definite, the minimum of the unconstrained problem is the generalized Cauchy point $\chi^c = \chi_k - H_k^{-1} \nabla_\chi J_k(\chi)$. An iterative process is required to take the bounds into account and gives an approximate solution $\bar{\chi}$
4. Define the descent direction $d_k = \bar{\chi} - \chi_k$ and perform a line search in this direction to satisfy the sufficient decrease condition: $J(\chi_{k+1}) \leq J(\chi_k + \alpha s_k d_k^T \nabla_\chi J(\chi))$, and try to enforce the curvature conditions: $|d_k^T \nabla_\chi J(\chi_k)| < \beta |d_k^T \nabla_\chi J(\chi_k)|$ with $\beta = 0.9$ and $\alpha = 10^{-4}$, and $s_k$ the step length.
5. Update BFGS Hessian approximation: $H_{k+1} = H_k + \frac{y_k {y_k}^T}{{y_k}^T p_k} - \frac{H_k p_k {p_k}^T H_k}{{p_k}^T H_k p_k}$ where $y_k = \nabla_\chi J_{k+1}(\chi) - \nabla_\chi J_k(\chi)$, and $p_k = \chi_{k+1} - \chi_k$.

Three stop criteria are implemented: norm of projected gradient lower than a tolerance, design variables and function variation between two iterates lower than a tolerance.

To summarize, the main idea of the algorithm is to minimize at each step a local quadratic model built on the gradient and Hessian matrix ($\left[\frac{d^2J(\chi)}{d^2\chi}\right]$) approximation by the BFGS formula of Eq. (2.3).

$$H_{k+1} = I + \sum_{i=1}^{k} \frac{y_i {y_i}^T}{{y_i}^T s_i} - \frac{H_i s_i {s_i}^T H_i}{{s_i}^T H_i s_i}, \tag{2.3}$$

This minimum is reached by the classical Newton step for an unconstrained problem of Eq. (2.4)

$$\chi^c = \chi_k - H_k^{-1} \nabla_\chi J_k(\chi). \tag{2.4}$$

If the step is too large, we can obtain $f(\chi^c) > f(\chi_k)$, therefore a line search step is used to ensure a sufficient decrease of the function.

## 2.2 Choice of the best suited algorithm

In this section, arguments for the choice of the best suited algorithm to our problem are provided. The question of computational cost relatively to the solutions quality is discussed.

### *2.2.1 The question of dimensionality*

From the outlook above of commonly used algorithms in aerodynamic optimization, we can say that the number of design variables is a driving constraint for the selection of an appropriate optimization strategy.

From a mathematical point of view, this raises the question of the minimal number of design variables that are required to perform an adequate optimization of a given shape. The short answer is that in theory, only one design variable is required. Indeed, for a parametrization with any number of design variables $n$, for $\chi_0$ the initial design vector and $\chi^* = argmin(J(\chi))$, one can define the function $\mathbf{P}_m : t \to t(\chi^* - \chi_0) + \chi_0$ and obtain $\mathbf{P}_m(0) = \chi_0$ and $1 = argmin(J(\mathbf{P}_m(t)))$. The $\mathbf{P}_m$ function is a minimal re-parametrization of the problem. In other words, for any parametrization, there always exists a single design variable re-parametrization which gives the same optimal shape. In the present demonstration, the $H_m$ function requires the solution $\chi^*$ of the high dimensional problem to be constructed, so is only a theoretical argument. In practice, see Chapter 5 section 2 on parametrization, wing geometry parametrization leads to a high number of design variables.

Besides, the type of algorithmic approach determines the relation between the number of required function evaluations, or design attempts, with the design space dimension. For costly real-life CFD applications, the computational cost and restitution time are important factors, and proportional to the number of function evaluations. "Curse of dimensionality" or "combinatorial explosion" is the fundamental reason why the computational effort needed to explore the design space or build a meta-model of functions steeply increases with the number of design variables, so has to be kept in mind for the choice of the best suited algorithm. When no hypothesis is made on the observed function, no region of the design space has a higher probability to contain the optimum than other regions, so the design space should be evenly explored.

For a uniformly sampled $n$-dimensional unit cube, the Euclidian distance $d$ between two samples is:

$$d = \left[ \sum_{i=1}^{n} \delta^2 \right]^{\frac{1}{2}}, \tag{2.5}$$

for a constant step $\delta$ in each direction. To ensure a sampling precision of $10^{-p}$, one should set $d = 10^{-p}$ and then:

$$\delta = 10^{-p} n^{-\frac{1}{2}}. \tag{2.6}$$

So the required number of samples $n_s(n, p)$, for evenly sampling a $n$ dimensional space with a precision $10^{-p}$, and without the bounds of the domain is such that:

$$n_s(n, p) \approx \prod_{i=1}^{n} \frac{1}{\delta} = \frac{1}{\delta^n}, \tag{2.7}$$

and finally:

$$n_s(n, p) \approx 10^{np} n^{\frac{n}{2}}. \tag{2.8}$$

For instance, by application of Eq. (2.8) sampling the $[0, 1]$ segment with a $10^{-1}$ precision requires $n_s(1, 1) = 10$ samples. In dimension 2, with the same precision we have $n_s(2, 1) = 200$, $n_s(10, 1) = 10^{16}$, and for a representative dimensions of our problems (see Part III) $n_s(10, 1) = 10^{16}$, $n_s(100, 1) = 10^{185}$. The extreme increase of $n_s(n, p)$, shown in Fig. 2.1 with $n$ is known as "Curse of Dimensionality" as invented by Bellman [6]. As a consequence, an

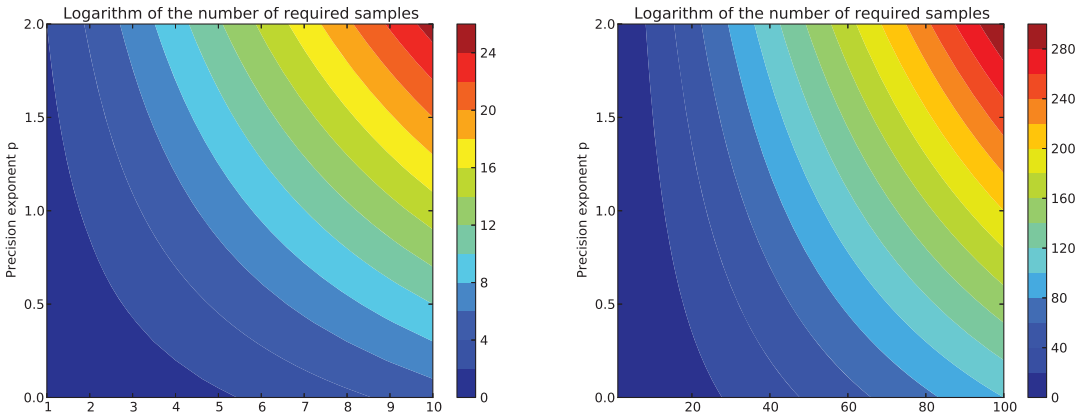


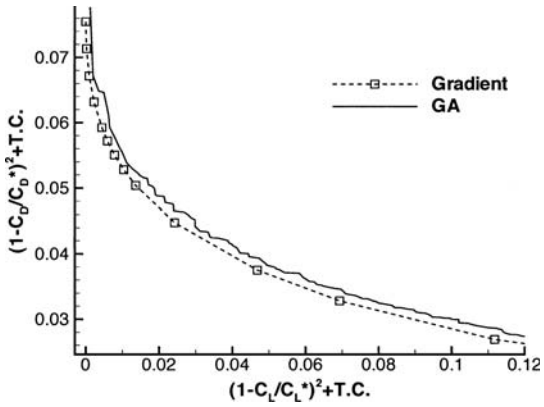

Figure 2.1: Number of required samples to keep an euclidian distance between samples of $10^{-p}$ in a $n$-dimensional space

optimization strategy that aims at being global tends to be very costly for CFD optimization cases in high dimension. Also, any meta-model requires sampling the design space and is therefore subject to the curse of dimensionality, so suitable for a moderate number of design variables. Laurenceau et al. [71] compared meta-model based with gradient-based optimization on a wing optimization problem. The Cokriging optimization, using the gradient of the objective function, outperformed the gradient-based algorithm with forty five design variables. But none of the solutions were shock-free with 11 remaining wave drag counts, probably due to the Hicks-Henne bumps parametrization of the 3D shape while single-point wing optimization should reach the zero wave drag. Hicks-Henne bumps tend to create very local shape deformations and therefore local minimums of the function. This raises the questions of convergence of the algorithms and highlights the importance of the parametrization for benchmarking algorithms.

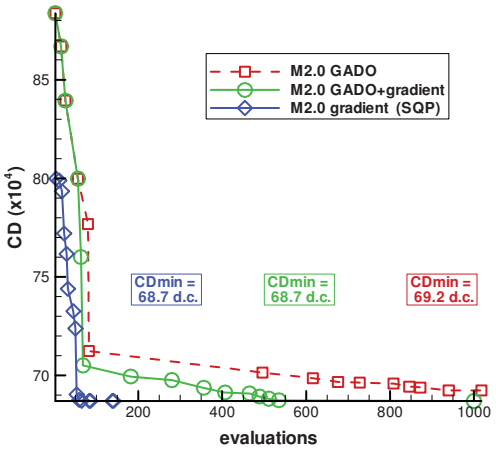
Finally, Shahpar, from Rolls Royce, shared his experience on the interest of a high dimensional parametrization in Ref [121]: "a reduced degree of freedom could translate to a stiffer problem for the optimizer to produce any improved solutions, in reference [129] it is shown that increasing the design space (spline points defining an annulus line of a compressor by 5 points) would lead to better results that is actually better than a hand design one, whilst the reduced design space could only produce results that were inferior to those obtained by the hand-design process."

### 2.2.2 Comparison between algorithms

Since GA explore the design space, involve randomness in the generation of design variables, and are subject to curse of dimensionality, the question of their efficiency in terms of objective decrease per function call arises. On the other hand, since gradient-based algorithms have only local convergence properties, the question of their ability to reach the Pareto frontier for drag optimizations also arises, since it is known that this approach can fail at discretizing non-convex Pareto fronts [24]. As a consequence, Nemec et al. checked that gradient-based optimizations [97] were able to reach a Lift / Drag Pareto front obtained using GA. 10 gradient optimizations were run with different weighting factors, leading to 10 points on the Pareto front, representing 250 calculations in total. The quasi-Newton algorithm required 25 iterations to attain convergence to engineering precision on the objective function. By comparison, GA required more than 3000 iterations and provided higher drag solutions due to the lack of exploitation as shown in Fig. 2.2a.




(a) Multipoint drag and lift polar optimization by Nemec et al. [97]




(b) Supersonic business jet aerodynamic optimization by Carrier [18]

Figure 2.2: Comparisons between genetic algorithm and gradient-based optimization for pure aerodynamic problems

In multidisciplinary aerodynamic and sonic boom optimization, Carrier notes that gradient-based algorithm converges to a local minimum, while GA gave better solutions [18]. To mitigate the computational cost while keeping efficiency in exploration, the GA was stopped before convergence and a gradient-based algorithm followed, in a sequential way and starting from the best solution proposed by the GA. For a single-discipline supersonic business jet aerodynamic optimization, the same author compared GA, the sequential approach and gradient-based optimization in Fig. 2.2b. Similarly to the study by Nemec et al. [97], the gradient-based algorithm was able to reach better results in six times less iterations than the GA, and three times less than the sequential GA-

Gradient algorithm for an identical solution. More recently, Chernukhin and Zingg performed transonic wing and airfoil optimizations, with gradient-based and hybrid algorithms, and concluded that such problems were unimodal [19]. However, they also addressed aerodynamic optimization problems, such as a Blended Wing-body optimization, on which hybrid algorithms performed better than gradient. A similar observation was made by Wild et al. [140] on a high-lift configuration optimization, where GA gave best results compared to local DFO and Gradient-based algorithms.

Lyu et al. [80] also investigated the question of flatness of the wing optimization design space by randomly perturbing the NASA CRM model, and using these shapes as starting point for single point optimizations in cruise condition. Initial and final pressure coefficients are shown in Fig. 2.3. The three shock-free optimums have the same pressure drag more or less 0.05 drag counts, which justifies the use of local optimization algorithms for this class of problems.

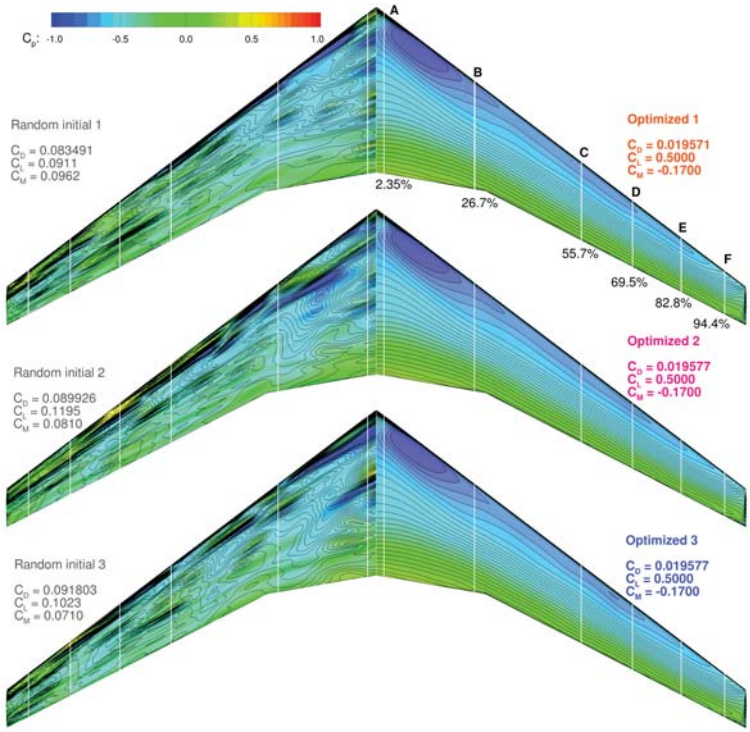



Figure 2.3: Three randomly deformed NASA CRM wings used as starting point for constrained single point optimizations by Lyu et al. [80]

To conclude, gradient-based algorithms have converged to a better or identical solution than genetic algorithms in multiple examples of transonic airfoil and wing drag minimization in cruise conditions. These problems are probably either unimodal, or have a flat design space with very close objective values at multiple minimums.

### *2.2.3 Conclusion: the choice of gradient-based algorithm*

Finally, given:

- The high computational cost of a 3D CFD computation of the flow around an aircraft.
- The hundred of design variables required to parametrize a 3D wing geometry.

- The computational cost of genetic algorithms scales with the square number of design variables times the number of objectives.
- The use of meta-model based algorithms restricted to a moderate number of design variables due to Curse of dimensionality.
- The computational cost of gradient-based algorithms scales with the number of design variables, when an efficient approach is used for the gradient calculation.
- The ability of gradient-based algorithm to find better solutions than genetic algorithms on drag minimization problems in cruise conditions, which is not necessarily the case in other flight conditions such as high-lift configurations in take-off.

We conclude that gradient-based algorithms are more adapted to a multipoint lift over drag maximization optimization.

## 2.3 A state of the art in multiple-operating conditions aerodynamic shape optimization

### 2.3.1 A brief history of gradient-based wing optimization

Given a set of partial differential equations describing the flow physics around a shape, Lions and Mitter [78] proposed in 1971 a formulation to control a criteria through modification of the boundaries of model using numerical optimization. These theoretical results opened a wide range of potential applications for the systematic design of airplanes, since at the same time, CFD was gaining in precision.

A few years later, Hicks et al. [112, 48] demonstrated that an airfoil performance could be optimized using a numerical procedure. Euler equations were used to estimate performance, controlled with a few design parameters. Finite differences allowed the computation of the derivatives of the objective function with respect to design variables. Finite differences is a straightforward approach, but very costly due to the number of required calculations that is proportional to the design variables number.

Based on these studies, Jameson proposed to use an adjoint formulation to make the derivatives calculation cost dependent to the number of objectives and constraints, instead of the number of design variables [55]. Since in 3D aerodynamic design the number of design variables is usually much higher than the number of objectives and constraints, the formulation is advantageous.

### 2.3.2 The single-point optimization effect

Drela, in Ref. [30], listed the pros and cons of airfoil optimization. He noticed that the optimizer is systematically taking advantage of all properties of the numerical model to improve the objective. The flow around an airfoil is very sensitive to shape modifications, so a given design can be very efficient at a given angle of attack, and behave badly when the angle of attack is modified, as shown in Fig. 2.4. Since the aircraft flies under a variety of operating condition, this is unacceptable. The phenomena has been reported as drag-creep [48], single-point optimization effect [30], or localized optimization [75].

Therefore, he proposed to add more operating conditions to the optimization problem using a weighted-sum approach. The results of a two and six-points optimizations are given in Fig. 2.5.

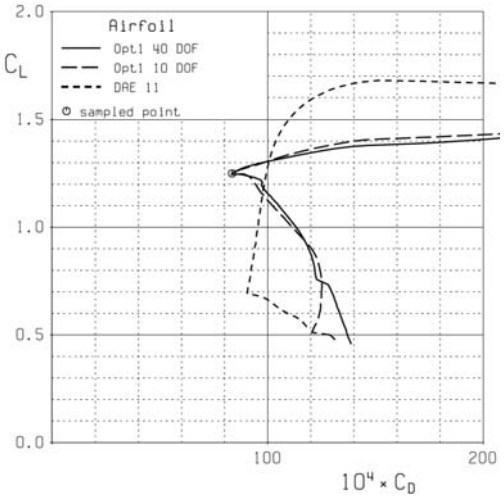
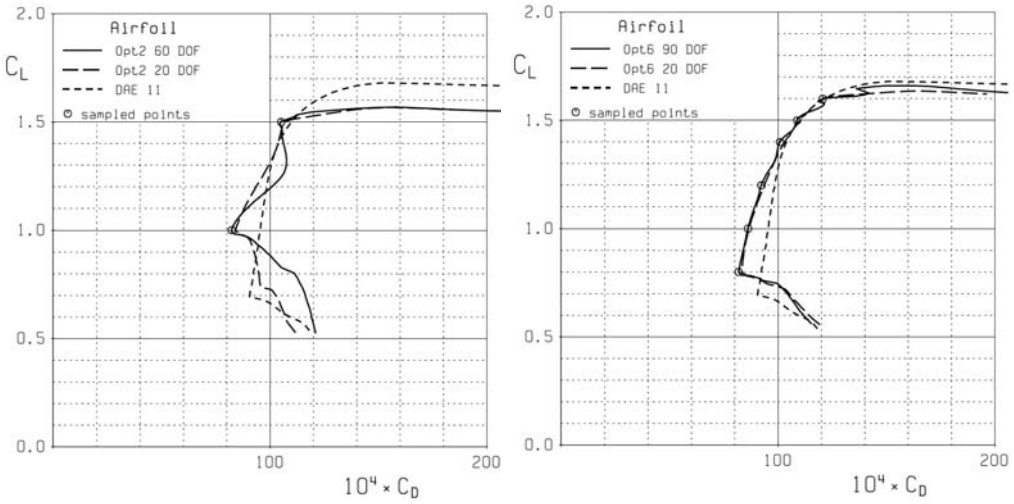



Figure 2.4: Single point airfoil optimization by Drela [30]: resulting polar.




(a) 2 points optimization (b) 6 points optimization

Figure 2.5: Two-points and six-points airfoil optimizations by Drela [30]: resulting polars.

An improvement of the performance on the added operating conditions was observed, but the problem remained for the other off-design conditions. A strong degradation of the performance was observed even for lift coefficients that are close to the design ones, as the shape was not adapted to the flow any more. Figure 2.5 is extremely demonstrative, as the phenomenon is amplified by the use of Hicks-Henne bumps to parametrize shape deformations. In Fig. 2.6, the comparison of the obtained shapes clearly shows the local shape modifications that makes the shape adapted to a local flow feature for each design operating condition.

The use of a geometry parametrization that generates more global shape deformations would not have created such local deformation, but the theoretical question remained open. In this same study, a first guess on a mathematical necessary condition to obtain a smooth polar, was given in Eq. (2.9). The main idea was that if for each operating condition, the optimizer can dedicate a design variable to locally adapt the shape to the flow features, then the number of operating conditions incorporated into the optimization problem should be of the same order of magnitude

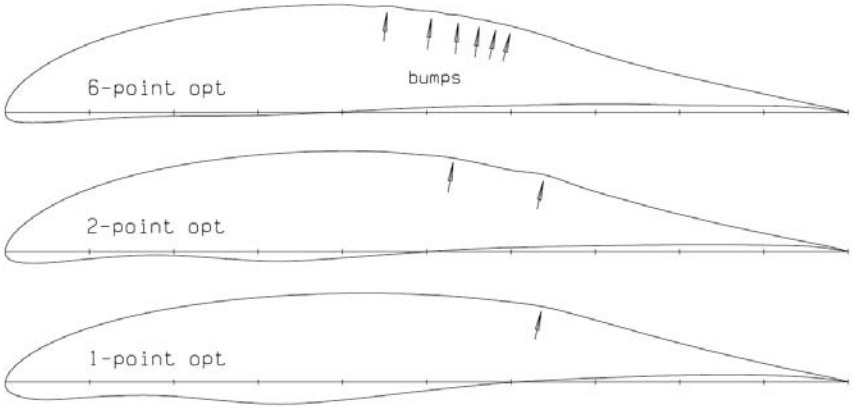



Figure 2.6: Single and multipoint airfoil optimizations by Drela [30]: the optimal geometries.

as the design variable number.

$$m = \mathcal{O}(n) \tag{2.9}$$

As Drela mentioned, many design variables are required to keep degrees of freedom and satisfy constraints. Also, for 3D applications, the description of surface deformations naturally involves many design variables. As a consequence, the computational cost of a robust optimization is "number of design variables" times more computationally expensive that a single-point optimization. The control of localized optimization effect is therefore a major challenge for real-life applications.

### 2.3.3 From multipoint to multi-mission

The class of aerodynamic shape optimization problems taking into account robustness to operating conditions is classically tackled with multipoint optimization methods in the literature. Many studies were achieved, most of the times with an arbitrary set of operating conditions. Campbell et al. for instance, show an example of two-points wing reverse design with target pressure on localized slices in [17]. Reuther et al. performed a 3D, 3-points multiple Mach and lift coefficient optimization, with geometrical thickness constraints [114, 115]. The robust design had a penalty of 1 drag count compared to the single point optimized operating condition, but unfortunately the optimization was not fully converged as stopped after 5 iterations. A similar study was also performed by Elliot and Peraire [36].

Buckley et al. noticed in [13] that because aircraft design is multidisciplinary, aerodynamic optimization needs to take into account for constraints coming from other disciplines, such as thickness constraints. Also, the multiple operating conditions encountered by the aircraft led to the use of 18 design operating conditions, involving multiple Mach, lift coefficients and Reynolds numbers. The high associated cost raises the issue of the minimality of the operating condition set. Finally the question of weights was addressed, and the distinction between hard points where the performance is difficult to improve and the preference of the designer is highlighted. Their confusion leads to missed performance opportunities when a too high weight is associated to the hard point that are not necessarily the point that influences the most the aircraft overall performance. They used an empirical formula for updating the weights from the work of Zingg et Elias [146].

Zingg and Elias [146] proposed a way for achieving a constant drag over a Mach range. The iterative process detects drag-creep at off-design conditions during the optimization and adds such a condition to the multipoint formulation. They noticed that in the case of multiple lift coefficients,

achieving a constant drag seems not reasonable due to the physics of airfoils. A multipoint formulation should therefore take into account both the designer preferences and the physical phenomena that are inherent to the system being designed.

In Ref. [75], Li et al. proposed three types of approaches to tackle multiple operating conditions optimization, and in particular to create a scalar objective. The first approach is a worst case one, where the maximum drag when the operating conditions varies is minimized with respect to design variables. A second approach is a search for constant performance over the operating conditions, which can be reached in multiple Mach but maybe not in multiple lift optimization, as stated in [12]. The last approach is a minimization of a weighted integral of the drag over the operating conditions. In this same study [75], the mathematical equivalence between the two first approaches is demonstrated. And a constant performance can be reached using a weighted integral, if the weights are well chosen. These mathematically equivalent approaches differ in the way the optimization problem is setup, so differ from the user point of view. In practice, the worst-case approach is rarely used in this class of problem.

Since thickness constraints are an indirect way to address the structural design effects on wing design, Jameson et al. [56] incorporated a structural model to the 2 and 3-points optimization for a redesign of the Boeing 747 wing. This leads to more degrees of freedom, and was already explored in the Multidisciplinary Design Optimization (MDO) context by Reuther et al. [113] following the emergence of the MDO field in engineering [123]. The weights between aerodynamics and structure disciplines was based on the Breguet range equation, see Chapter 1.

In the industry, multipoint optimizations were performed with a Euler-Boundary layer code at Boeing [143]. Since a Newton approach was used to resolve the coupled problem with exact linearization of the residuals, the adjoint equation for the computation of the derivatives only required the linearization of the post-processing routines, which is straightforward. Adjoint-based multipoint optimization was available by 1997 in this framework and the TRANAIR optimization massively contributed to Boeing airplanes design, including the GE90-115B engine integration on the B777 for instance [58].

One of the most advanced demonstration in terms of taking the mission into account during optimization, and complexity of the addressed case is the work by Liem et al [76] that extends the study by Kenway et al. [63]. The main idea of the study is that operating conditions and associated weights for a NASA Common Research Model multipoint optimization can be deduced from flight history of a similar aircraft. To this aim, they browsed the flight history of the Boeing 777-200ER, and extracted payload and range statistics. A mission model can estimate fuel burn from payload and range, as well as probability density functions of operating conditions, here Mach, altitude, lift coefficient and tail trimming angle. A continuous model of lift and lift-over-drag is required for that, see Eq. (1.7). To this aim, a Krigging surrogate model was used to interpolate CFD data. Operating conditions to be included in the multipoint optimization were selected to evenly fill the operating condition space, in a prescribed number. The surrogate model is rebuilt from these 25 new points, and the associated weights are computed from flight statistics. The multipoint problem setup is summarized in Fig. 2.7.

Results are displayed in Fig. 2.8. The optimizer decreased the weighted average fuel burn by 8.8%. It was achieved by an increase in wingspan and aspect ratio, while a more inboard loading allowed a slightly lighter structure. Higher payload and range mission's fuel burn was decreased to the detriment of shorter missions and the ones with lower payload.

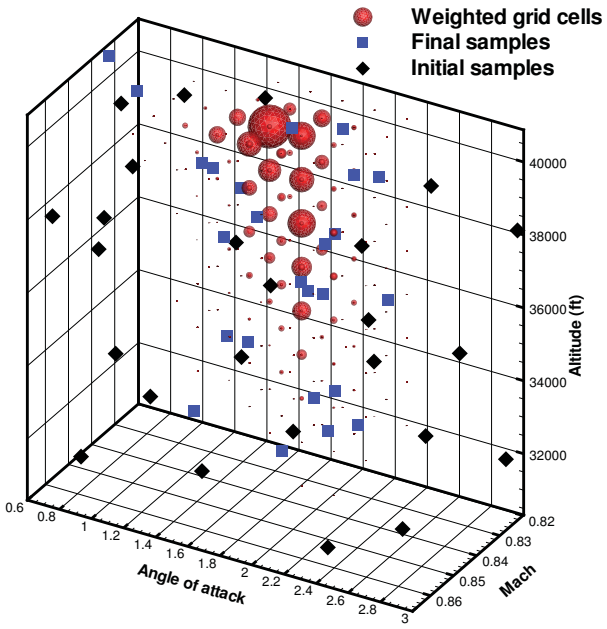
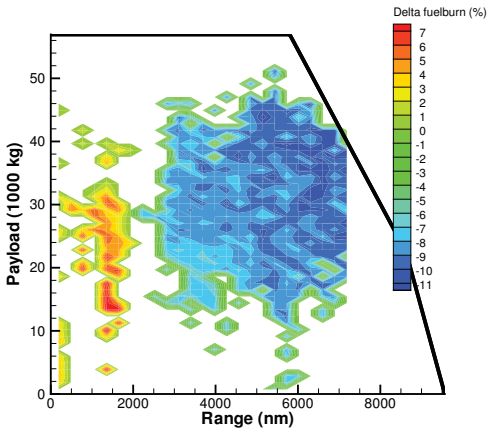
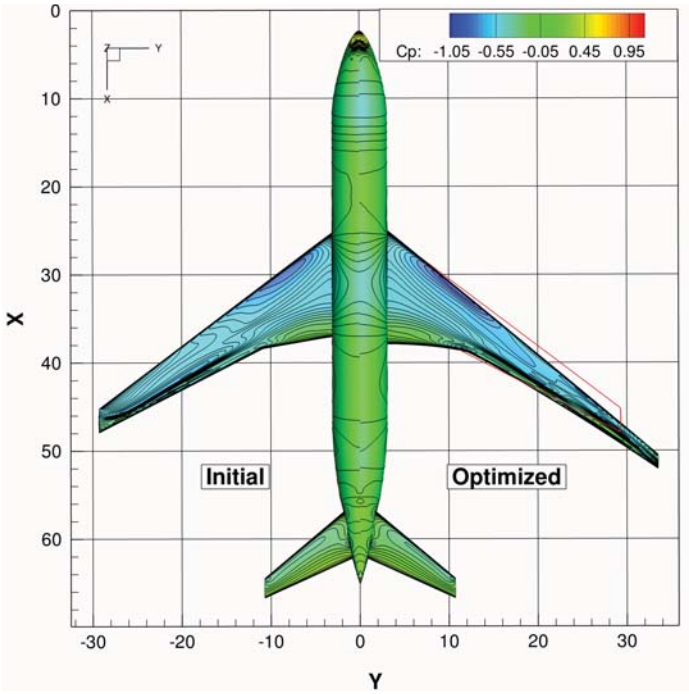



Figure 2.7: Operating conditions and associated weights used by Liem et al. [76] for a multi-point, multi-mission wing aero-structural optimization




(a) Fuel burn of optimized geometry for targeted missions




(b) Optimal shape and pressure distribution

Figure 2.8: Multipoint, multi mission optimized CRM performance and shape by Liem et al. [76]

## 2.4 Robust and multiobjective optimization

### 2.4.1 Robust optimization in engineering

Taguchi initiated a three steps design process that takes into account for uncertainties [131]. Those steps are:

1. System design: aims at finding the main parameters that drive the performance of the system.
2. Parameter design: aims at finding the parameters that give the maximal performance.
3. Tolerance design: aims at finely tuning the parameters of the system to optimize the performance. There is no conceptual difference with the previous step. The distinction usually relies on methods and geometrical details.

The main idea of Taguchi is that low effort should be put when the parameters have a high associated uncertainty, so at the beginning of the design process, and a higher effort is put at the end. The methodology is broadly used. For instance, a typical aircraft design process that starts with low fidelity models and high level trade-offs between disciplines, and at the end takes care of a fine tuning of the systems is similar to Taguchi's process.

The origins of uncertainties to which a design procedure should be robust at were listed by Beyer [7]:

- Operating conditions of the system are subject to uncertainties. The performance of the system usually depends on operating conditions. The real product may therefore not achieve the expected performance.
- Production tolerances and actuator imprecision are such that the real system is different from the designed system.
- The models used to design the system are uncertain so their outputs and performance estimations may not reflect reality.
- Constraints may not be feasible on the real system, due to the previous reasons.

For aerodynamic optimization robust to the operating conditions encountered during the aircraft's missions, only the first source of uncertainty is treated.

Beyer also listed possible measures of uncertainty:

- The worst case: the maximum of all realizations.
- Probabilistic threshold measure: the number of realizations that perform better than a threshold.
- Statistical feasibility: the probability of constraint violation to be lower than a threshold.
- Expectancy: the integral of realizations times the probability of realization.

To address the problem of aircraft mission fuel consumption, the worst case may not be adapted measure, since the aircraft physics prevents from achieving a constant performance: at some point

the LoD decreases when the angle of attack or the Mach increases. The probabilistic threshold has the same drawback as the worst case, and the statistical feasibility aims at treating uncertain constraints, while we have an uncertain performance and certain constraints.

The work by Mulvey et al. on robust optimization of large scale systems [92] has largely contributed to the development of Robust Optimization. A characterization of the desirable properties of a system when problem data is defined by a set of scenarios for their values is proposed. The solution is declared robust if it remains close to optimum for all data. It is "model robust" if it remains "almost" feasible for all data scenarios. They penalized the performance objective with a robustness measure in order to formulate a robust optimization problem. For instance, one can minimize the mean value of the possible realizations, weighted by probability of each realization. The formulation also allows to incorporate higher moments. An alternative approach is the worst case scenario, the so-called "min max". A feasibility penalty function is added to the objective, to penalize the designs that violate constraints. Applications to aircraft scheduling and minimum weight structural design were presented.

In our case, all the realizations are always feasible whatever the uncertain parameter value, since we can make the constraints independent of the uncertain variables (see Part III). A robust optimization formulation taken from Mulvey et al. applied to the problem presented in Chapter 1, would be a maximization of the mean Lift over Drag ratio, without feasibility penalty: $min \quad \sum_i LoD_i \xi_i$, where $\xi_i$ is a probability density function. By identification, the weighted sum approach of Eq. (1.11) is a robust optimization formulation that aims at maximizing the expectancy of the LoD with associated probability density function of : $\xi_i = \frac{C_s(\alpha(t_i))w(t_i,\chi)g(t_i)}{LoD(\chi,\alpha(t_i),w(t_i,\chi))^2}\delta t_i$.

### 2.4.2 The algorithm point of view

In this section, we recall some key features of the l-bfgs-b algorithm presented in Section 2.1.3, and analyze the consequence of these properties on the mission-performance aircraft optimization problem formulation.

The main idea of quasi-Newton optimization algorithms is to apply a Newton iteration to the derivative of the objective function, which can be summarized by the equation: $\chi^c = \chi_k - H_k^{-1}\nabla_\chi J_k(\chi)$ (see Section 2.1.3). It is then interesting to note that equations (2.4) and (2.3), driving the l-bfgs-b iterations, do not depend on the function value but only on the gradients. As a consequence, for the optimization problem formulation, the focus should be made on the gradients of the objective function rather than on the objective value itself.

As highlighted by Eq. (1.11), the gradient of the mission-fuel consumption is a linear combination of the gradients of the lift over drag at multiple operating conditions. This result relies on the fact that an aggregate objective function is used, which naturally appears in the addressed problem.

We note $LoD(\chi, \alpha_i, w_i) = LoD_i(\chi)$, for any aggregation function $\mathcal{A}$, $\mathcal{A} : (\chi, LoD_1(\chi), \ldots, LoD_m(\chi)) \rightarrow \mathcal{A}(\chi, LoD_1(\chi), \ldots, LoD_m(\chi))$, by the chain rule we obtain:

$$\frac{d\mathcal{A}}{d\chi}(\chi, LoD_1(\chi), \ldots, LoD_m(\chi)) = \frac{\partial \mathcal{A}}{\partial \chi} + \sum_{i=1}^{m} \frac{\partial \mathcal{A}}{\partial LoD_i}\nabla_\chi LoD_i(\chi). \tag{2.10}$$

By identification with Eq. (1.11), we can deduce the aggregation coefficients for the mission fuel consumption objective function, we have $\frac{\partial \mathcal{A}}{\partial \chi} = 0$ and:

$$\frac{\partial \mathcal{A}}{\partial LoD_i}(\chi) = -\omega_i(\chi) = -\frac{C_s(\alpha(t_i))w(t_i,\chi)g(t_i)}{LoD(\chi,\alpha(t_i),w(t_i,\chi))^2}\delta t_i, \tag{2.11}$$

A similar analysis was proposed by Marler [85] to interpret the weights of the weighted sum method as derivatives of the user preference function.

Given the remark that the algorithm bases the search direction on the gradient information, and given Eq. (2.10), the $m$ terms $\nabla_\chi LoD_i(\chi), i \in (1 \dots m)$ define the search directions in the design space, and the $\frac{\partial \mathcal{A}}{\partial LoD_i}$ terms define the compromise between these directions.

Since $\forall i \in (1 \dots m)$, $\frac{\partial \mathcal{A}}{\partial LoD_i} < 0$ for the $\Delta_w$ function, any optimization process that decreases the $LoD_i$ values for a subset of operating conditions without degrading the others also decreases the mission fuel consumption.

To conclude, the point of view given by Eq. (2.10) reveals the two main questions to control the continuous function $t \to LoD(\chi, \alpha(t), w(t))$:

- The choice of the subset of operating conditions to be included in the optimization problem $((\alpha_0, w_0), \dots, (\alpha_m, w_m))$. That discretization problem actually reveals two questions. What should be $m$, the number of samples? And what should be the samples values, keeping in mind that the cost of the optimization is proportional to $m$?
- The choice of a preference function $\frac{\partial \mathcal{A}}{\partial LoD_i}$, i.e. the weights.

The first point is addressed in the next section and relies on a multi-objective analysis. The latter is addressed through an original approach in Chapter 4.

### 2.4.3 Link between robust and multi-objective optimization

Robust optimization can be addressed in a deterministic way. Instead of minimizing the effect of perturbations on a system's performance, one can aim at maximizing the performance of the system on the range of possible operating conditions. The realizations of perturbations may be hard to predict, so are often seen as non-deterministic, while the realistic range of realizations is generally known. For instance, we know in advance that any civil transport aircraft will fly under Mach 0.9 and above Mach 0.3, and realizations out of this range are so rare that they have no influence on the average fuel consumption on the missions. Giving a mathematical meaning to "maximizing the performance of the system on the range of possible operating conditions" is required, since the performance on a range of operating conditions is a continuous function, and the choice of an order relationship to compare two functions is not trivial. However, we have seen in Chapter 1 that in the end, a discretization of these functions is necessary to compute the performance and its gradient. Then, these discrete performance values can be compared. That is the scope of multi-objective optimization.

In multi-objective optimization, the Pareto frontier is a very important concept. We apply it here to multi-operating conditions optimization with a parametric objective function $j(\chi, \alpha)$, to compare the performance $j$ of possible designs represented by $\chi$, on discrete sets of operating conditions $\alpha \in (\alpha_1, \dots, \alpha_m)$. For all feasible choices of design variables $\chi$, we plot the performance $j(\chi, \alpha_i)$ in a $m$-dimensional space, for $\alpha \in (\alpha_1, \dots, \alpha_m)$. A point $\chi_0$ is said dominated if there exists at least one $\chi^*$ and one $i^*$ such that for all $i$, $j(\chi^*, \alpha_i) \leq j(\chi_0, \alpha_i)$, and $j(\chi^*, \alpha_{i^*}) < j(\chi_0, \alpha_{i^*})$. A non-dominated solution $\chi^*$ is then such that for all $k$, the objective $j(\chi, \alpha_k)$ cannot be decreased without increasing some $j(\chi, \alpha_i)$ for some $i \neq k$. The Pareto frontier is the set of all non-dominated solutions. Since all solutions on the Pareto frontier are feasible, by definition, all interesting solution from the engineering point of view are on the Pareto Frontier. In general, the Pareto Frontier can

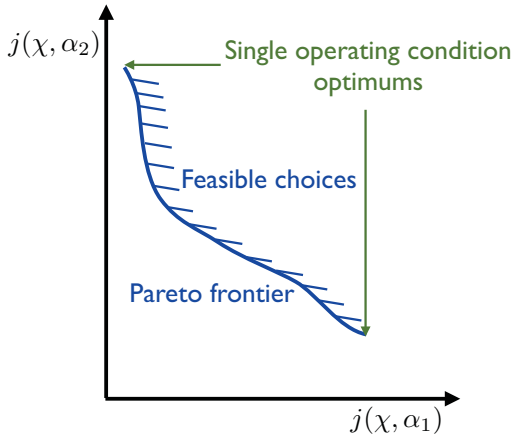



Figure 2.9: A representation of the Pareto frontier for a two-points multi-objective problem

be non-convex and non-continuous. It is also not generally a single point, meaning that, in contrast to single-point optimization, multi-objective optimization problems do not have a single solution.

Taking a deterministic multi-objective optimization point of view on multi-operating conditions optimization can provide a methodology to setup the optimization problem. For instance, specifying user preferences on the different objectives is a key point in multi-objective optimization, which has been addressed by many authors. Marler et al. achieved a review on multi-objective optimization methods for engineering [84] and classified the approaches in three categories: methods with a priori articulation of preferences, methods with a posteriori articulation of preferences, and methods with no articulation of preferences. The drawback of methods with no articulation of preferences is that the obtained solution has an arbitrary position on the Pareto frontier. Consequently, algorithms that provide multiple solutions on the Pareto frontier may be preferred by the decision maker when a preference criteria is not known. Methods with a posteriori articulation of preferences propose a set of solutions to the designer that choses one. The literature is abundant on the topic, Normal Boundary Intersection [25], and Normal Constraint [88], Successive Pareto Optimization [91], Directed Search Domain [37] methods provide an even distribution of Pareto Optimal points. Multiple-Gradient Descent Algorithm [27] is an hybrid approach that finds a common descent direction to all objectives to reach the Pareto frontier, and then a Nash game to move on the Pareto frontier. However these approaches require to solve multiple single-objective problems so are costly.

Methods with a priori articulation of preferences are the most efficient computationally speaking [84]. Specifying the preference is hard because it usually depends on the attainable solutions that are unknown at the time of the problem setup. Since the starting point of the optimization is usually not on the Pareto frontier, an estimation of the Pareto frontier's width for instance, is difficult. We present here a computationally efficient method for the specification of user preferences.

For each criteria $j(\chi, \alpha_i)$, we note ${j_i}^o$, the minimum of the single objective optimization problem ${j_i}^o = \ min\ j(\chi, \alpha_i)$. The "utopia point" or "Nadir point" has coordinates in the Pareto diagram defined by $({j_1}^o, \ldots, {j_n}^o)$. Figure 2.10 represents the Pareto frontier and the utopia point. Since the existence of a design vector that reaches effectively the utopia point's performance is unrealistic for most multi-objective optimization problems, the second best solution is the closest point (in

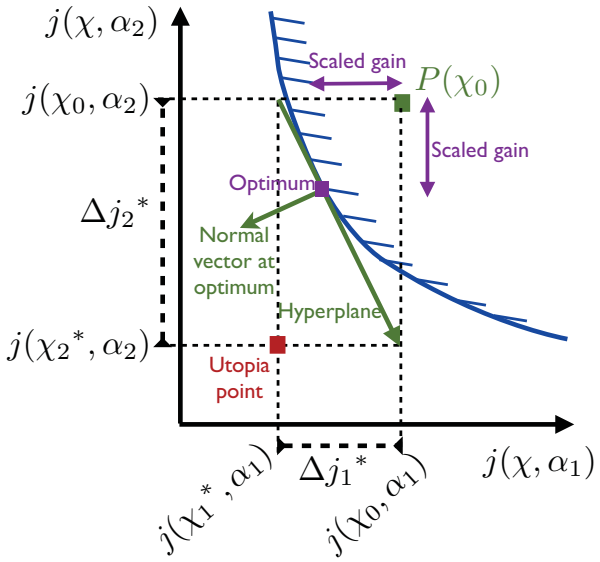



Figure 2.10: Utopia point and the Pareto front in multi-objective optimization

the Euclidian norm sense for instance) to the utopia point. A solution that is close to the utopia point represents a good compromise between objectives. By definition, the utopia point calculation requires $m$ single-point optimizations, where $m$ is the number of operating conditions. To avoid such a costly calculation, an approximation of it called "unattainable aspiration" point is a reasonable substitution [84]. The approximation should be provided by the user and therefore requires specific knowledge on the problem. In the case of aerodynamic optimization, for instance we know that in single-point, shock-free configuration can be obtained if the relative thickness of the shape is not too high [47]. Also, we know how to estimate the potential gains on the other sources of drag such as induced drag and viscous pressure drag, as will be shown in Chapter 7. In the end it is possible to estimate the unattainable aspiration at the cost of $m$ CFD calculations and far field drag analysis, which is about two order of magnitudes lower than the cost of $m$ single-point optimizations. The method was used by Laurenceau [52] in aerodynamic optimization, with weights $\omega_i = \frac{1}{\Delta j_i{}^*} = \frac{1}{j(\chi_0,\alpha_i) - j_i{}^o}$, where $\chi_0$ is the initial value of design variables. As shown with two objectives in Fig. 2.10, the vector $(\frac{1}{\Delta j_1{}^*}, \frac{1}{\Delta j_2{}^*})$ is the tangent hyperplane to the Pareto front of normal $(\Delta j_1{}^*, -\Delta j_2{}^*)$, at a point which balances the gains between the objectives. Therefore the minimization of a weighted sum of objectives with weights $\omega_i = \frac{1}{\Delta j_i{}^*}$ converges to a point that balances the gains between the multiple objectives.

The weighted sum method, first proposed by Zadeh [144], is often used in multi-objective optimization. Its simplicity and the fact that the minimum of a weighted sum of objectives is a Pareto optimal point [144, 84] when the weights are positive, are appreciated. A major drawback [24] is that it cannot reach non-convex regions of the Pareto frontier [66, 49]. To summarize, being the minimum of a weighted sum is a sufficient but not necessary condition for Pareto optimality. It must be noted that in general, finding such a minimum requires the use of a global optimization algorithm. Besides, the weighted sum method has an useful property: at the convergence point, the tangent hyperplane to the Pareto frontier has a normal vector that is collinear to the weighting vector [25], see also [10] pages 46-59 for more details about separating hyperplanes. Since the utopia point is an interesting target, this enables to select the weights to obtain a good compromise on the gains at each objective, with the help of the graphical interpretation in Fig. 2.10.

In multi-objective optimization, the scaling of objectives aims at balancing the gains on each

objective so that none naturally dominates the objective function. The user preference is specified afterwards by multiplication by a user specified factor. To this aim, Marler gave an interpretation of the weights: "how much of a loss in an objective is one willing to sacrifice for a gain in another objective" [85] that the user should keep in mind when specifying the weights. This interpretation is compatible with a Lagrange multipliers interpretation. Indeed, the zero gradient condition at the optimum $\nabla_\chi j(\chi, \alpha_k) + \sum_{i=m, i\neq k}^{m} \frac{w_i}{w_1} \nabla_\chi j(\chi, \alpha_i) = 0$, can be seen as the KKT condition of a minimization of the $k$-th objective under constraint of off-design performance on the $m-1$ other objectives. The Lagrange multipliers are equal to the derivatives of the objective value at the optimum with respect to the constraint value (from the envelope theorem [130], see Chapter 5 section 9.3, envelope theorem). In other words, the weights ratio $w_i/w_k$ represents how much the optimum value of the $k$-th condition is degraded for a unitary gain on the $i$-th condition.

Given the computational efficiency of the weighted sum method and the fact that we have an efficient approach for the choice of the weights, this method will be used in the rest of the present study. A practical way of selecting the weights for the specific case of aerodynamic optimization is described in Chapter 7 section 6.

## 2.5 Conclusion and way forward: the need for a mathematical formalism

As seen at the beginning of this chapter in the review of multipoint optimization studies in aerodynamics, the question of the weights specification is quasi-systematically addressed in the literature, since it is naturally interpreted as a way to specify the user preference. However, in most of the studies, the set of operating points to be included in the optimization is stated as an input of the problem. The impact of the choice of the operating conditions used on the obtained solution is rarely questioned, while there is no a priori reason for the selection of a particular operating condition in the ranges of operating conditions in which the real aircraft flies. A possible reason for that is the lack of theoretical framework on the operating condition choice.

Drela [30], as already mentioned, highlighted first the link between the design space and the operating conditions space, stating that for a $n$-dimensional design space a number of operating conditions $m$ to be included in the optimization problem should be of the same order of magnitude: $m = O(n)$. Li et al. [75] formulated the "localized optimization", or "drag-creep effect" into a mathematical condition: the optimized polar is not robust if there exists an operating condition at which the performance can be improved without degrading the performance at one of the operating conditions involved in the weighted sum objective. The reformulation allowed them to refine the relationship, and they demonstrated that one should use $m > n$, as a necessary condition to avoid drag-creep. For high dimensional problems, this inequality makes the robust optimization unaffordable.

However, there are multiple reasons why the inequality $m > n$ should be questioned, as it is:

1. **Not goal-oriented**: it does not take into account for the specificities of the objective function. For instance, if this function does not depend on the operating condition, or in a negligible way, one operating condition is sufficient to obtain a robust solution. From this example, we can state that the inequality $m > n$ may not lead to the minimal number of required operating condition. In other words, it is not a necessary condition.

2. **Parametrization independent**: it does not take into account for the parametrization effects on the optimization problem. For instance, as shown by Drela, the use of local defor-

mation amplifies the localized optimization effect, leading to non-robust designs, while global and smooth deformations can lead to more robust designs. With the inequality $m > n$, two optimization problems setup with different parametrizations, a smooth one and a non-smooth one, but with the same number of design variables, lead to the same number of required operating conditions.

3. **Sampling independent**: it does not take into account for the value of the selected operating conditions. For instance, if all the selected operating conditions are clustered in a region of the operating conditions space, the optimization can lead to the drag creep effect. Such an example is provided in the RAE2822 multipoint optimization section. In other words, the inequality $m > n$ is not a sufficient condition.

4. **Operating condition range independent**: if one performs two robust optimizations on operating conditions $\mathbf{I_1}$ and $\mathbf{I_2}$ such that $\mathbf{I_1} \in \mathbf{I_2}$, one would expect that the required numbers of operating conditions, respectively $m_1(\mathbf{I_1})$ and $m_2(\mathbf{I_2})$, are such that $m1(\mathbf{I_1}) \leq m2(\mathbf{I_2})$. At least, the operating conditions ranges should be taken into account in the choice of operating conditions to be included in the optimization problem.

As a consequence of the lack of theoretical framework for multipoint aerodynamic optimization problem setup, which computational cost is affordable, practical approaches were proposed to detect drag-creep and correct the optimization problem. For instance, in [146], Zingg and Elias performed a multi-Mach optimization, and checked during the optimization that the drag was a monotonous function of the Mach number. If not, an operating condition was added to the optimization problem. The method was then tested on problems with constraints coming from aircraft aerodynamic design and more types of operating conditions in [12]. The convergence of such a procedure that iteratively modifies the optimization problem and solves it, is an open question. In addition, an extension to multi-dimensional (Mach, Cl, Reynolds...) operating conditions spaces is not straightforward. It is also the case for objective functions which desired variation with respect to operating conditions is not known.

To conclude, there is a need for a more adapted theoretical framework to setup multipoint aerodynamic optimization problems. In the next Chapter, we demonstrate necessary and sufficient conditions that links parametric and multipoint optimizations together with a new sampling method of the operating condition space. When respected, these mathematical conditions lead to a minimal cost optimization problem such that at the optimum, there does not exist an operating condition at which the performance can be improved without degrading the performance at an operating condition involved in the weighted sum objective. It is an extension of the non-domination condition, in the Pareto sense, to continuous sets of concurrent objectives. The formulation is goal-oriented, takes into account for the parametrization smoothness and operating conditions ranges.

# Part II

# The Gradient Span Analysis method

# Table of Contents

# CHAPTER 3

## GSA: the theory

***Summary*** In this chapter, a theoretical framework for the choice of operating conditions to be included in a multipoint optimization problem is proposed. This deterministic approach is based on the analysis of the descent directions generated by the parametric objective gradient when the operating conditions vary. Necessary and sufficient conditions for the well poisedness of the optimization problem are demonstrated. An algorithm called Gradient Span Analysis (GSA) is also proposed, ensuring a minimal computational cost of the resulting optimization process. This work was submitted to the Journal of Optimization Theory and Applications.

***Résumé*** Dans ce chapitre, des bases théoriques sont proposées pour le choix des conditions d'opérations à inclure dans un problème d'optimisation multi-point. Cette approche déterministe est basée sur l'analyse des directions de descentes potentielles générées par le gradient de la fonction objective paramétrique quand les conditions d'opération varient. Des conditions nécessaires et suffisantes pour que le problème d'optimisation multi-point soit bien posé sont démontrées. Un algorithme appelé Gradient Span Analysis (GSA) est également proposé, assurant un coût minimal du problème d'optimisation résultant. Ce travail a été soumis au Journal of Optimization Theory and Applications.

## 3.1 Multipoint objective function

Given the parameter $\alpha \in \mathbf{I} \subset \mathbb{R}$, the typical single-point aerodynamic optimization problem is to find the vector of design variables $\chi \in \mathbf{O}_{ad} \subset \mathbb{R}^n$ that minimizes the function $\mathcal{J}(\mathcal{U}, \chi, \alpha)$ subject to the equality constraint $R(\mathcal{U}, \chi, \alpha) = 0$. $\mathcal{U}$ is a solution of the discretized non-linear steady-state flow equations. Design variables come from the parameterization of the aerodynamic shape. The objective function is typically the drag, due to dissipative phenomena. Fluid dynamics equations (3.1) are Euler or Reynolds-Averaged Navier-Stokes equations. $\mathcal{U}$ represents the density, velocity and energy of the flow. The parameter $\alpha$ is a specified operating condition in the single-point design problem such as Mach number, Reynolds number or angle of attack.

$$R(\mathcal{U}, \chi, \alpha) = 0. \tag{3.1}$$

$R$ is supposed to be a regular function of $\chi$. The implicit function theorem allows to write:

$$\mathcal{U} = U(\alpha, \chi). \tag{3.2}$$

The scalar objective function $\mathcal{J}(U(\alpha, \chi), \chi, \alpha)$ can also be written:

$$\mathcal{J}(U(\alpha, \chi), \chi, \alpha) = j(\alpha, \chi), \tag{3.3}$$

and is assumed to be continuously differentiable in the region,

$$\chi_L \leq \chi \leq \chi_U,$$

where $\chi_L$ and $\chi_U$ are the specified lower and upper bounds.

When $\alpha$ is no longer fixed but varies continuously in the set $\mathbf{I}$, the optimization problem can no longer be formulated as previously because for a given $\chi$, $j(\chi, \alpha)$ is a function of $\alpha$. Eq. (3.1) is now required to be a regular function of $\chi$ and $\alpha$. Note that the variable $\alpha$ is not considered as a design variable.

A strategy must be defined to keep a single-objective optimization problem. The approach presented in this paper uses the inherent properties of the state equation described in Eq. (3.1) to adequately sample the set $\mathbf{I}$ of operating conditions. From this sampling and the computation of the associated objectives, a weighted sum is used to build an aggregate objective function,

$$J(\chi) = \sum_{k=1}^{m} \omega_k \; j(\chi, \alpha_k). \tag{3.4}$$

This strategy is similar to the Proper Orthogonal Decomposition (POD) method that has been developed first by Kosambi [65] and then applied to aerodynamics in Refs. [93, 132, 11]. POD takes a set of solutions of the state equation as inputs, from various operating conditions for instance. From these solutions, a set of basis vectors is computed using singular value decomposition. Then a reduced model is built by selection of the highest singular values. This projection can be used for approximation of the states when the operating conditions vary. In the present case, the aim is not to approximate the solution of state equations but to take advantage of the linear dependencies between the gradients of functionals computed from the states with respect to design variables. In both cases, the underlying idea is that there exists an appropriate lower dimension approximation of the system states for the purpose of interpolation or gradient computation.

Another similar approach exists in the machine learning and perceptrons fields, where the Stochastic Gradient Descent (SGD) algorithm also aims at minimizing a weighted sum of objectives [9] in parametric optimization. The numerical cost of the gradient evaluation is also an issue, and the straightforward approach of SGD is to randomly select the operating condition at which the gradient is evaluated. Quasi-Newton algorithms with BFGS approximation are also used in this field [8]. Considering the very high cost of CFD calculations with adjoint, the number of optimization iterations affordable for a real life design case (lower than 100), and the need for repeatability for comparison between designs, introducing randomness in the optimization process is not acceptable so the SGD algorithm is discarded for our study.

Introduction of the aggregate objective function allows the formulation of a multipoint optimization problem. Given $m$ parameters $(\alpha_1, \ldots, \alpha_m) \in \mathbf{I}^m$ and the associated weights $(\omega_1, \ldots, \omega_m) \in \mathbb{R}^m$, the problem is to find the vector of design variables $\chi \in \mathbf{O}_{ad} \subset \mathbb{R}^n$ that minimizes the function:

$$J(\chi) = \sum_{k=1}^{m} \omega_k \; j(\chi, \alpha_k),$$

subject to the equality constraints:

$$R(\chi, \alpha_k) = 0, \qquad k = 1, \ldots, m.$$

The choice of the set of conditions will be discussed in Sec. 3.2. Specifying the weights relies on multi-objective optimization analyses, this question is addressed for instance by Normal Constraint [88], Successive Pareto Optimization [91], Directed Search Domain [37], and Multiple-Gradient Descent Algorithm [27] and is not treated in the present study.

Aggregate objective functions in Eq. (3.4) show a natural parallelism through the possible independent solutions of the state equation $R$ and evaluations of the function $j$ that represent the major part of the computational effort. This is sketched in algorithm 1, classically used with variants on the minimization step for a given aggregate objective gradient [17, 114, 115, 13, 56, 146]. The main steps consist of solving the state equations for all the operating conditions, computing the objectives $j(\chi, \alpha_k)$ and their gradients with respect to the design variables $\chi$, aggregating the function and the gradients, and the use of a gradient-based minimization algorithm to update the design variables.

**Algorithm 1** Scheme of the parallel property of the multipoint optimization

For given operating conditions $(\alpha_1, \ldots, \alpha_m) \in \mathbf{I}^m$, weights $(\omega_1, \ldots, \omega_m) \in \mathbb{R}^m$, and a starting point $\chi_0 \in \mathbf{O}_{ad}$,

- Optimization iterations: $l = 1, \ldots$
  - $m$ parallel resolutions of $R(\mathcal{U}_k, \chi, \alpha_k) = 0$,
  - $m$ parallel evaluations of $j(\chi_l, \alpha_k)$,
  - $m$ parallel evaluations of $\nabla_\chi j(\chi_l, \alpha_k)$,
  - Define the descent direction: $d = \sum_{k=1}^{m} \omega_k \nabla_\chi j(\chi_l, \alpha_k)$,
  - $\rho_{opt} = \operatorname{argmin}_{\rho \in \mathbb{R}_*^+} \{J(\chi_l - \rho d)\}$,
  - $\chi_{l+1} = \chi_l - \rho_{opt} d$.

A general background is now defined and the next sections focus on the specific issues raised by the parameters $(\alpha_1, \ldots, \alpha_m)$. In particular, a link is made between these parameters and some properties of the objective function of Eq. (3.4).

## 3.2 Choice of a set of conditions

In this section, the minimization of parametric objective functions that depend on a set of operating conditions in addition to design variables is formulated with the weighted sum method of Eq. (3.4).

The objective is to control the performance $j(\chi, \alpha)$ over a range $\mathbf{I}$ of operating conditions while only a finite number $m$ of conditions are explicitly used in the formulation Eq. (3.4). The underlying idea to achieve this goal is that the state equation of the system in Eq. (3.1) links the different values $j(\chi, \alpha)$ computed at each operating condition $\alpha \in \mathbf{I}$. In the aircraft design example, the Navier-Stokes equations are solved for each operating condition $\alpha$ to compute the flow around the same shape defined by $\chi$.

Because any two conditions can be either cooperative, independent or concurrent, improving the performance at one operating condition by modifying the shape either leads to an improvement, a degradation or no modification of the other one. This point can be exploited to generate the minimal set of conditions $\alpha_k$ that gives the shape design with the same performance as if the whole set $\mathbf{I}$ was considered. Reasoning on the set formed by the gradients of the objective function with respect to the design variables at each sampled condition, Li et al. [75] established that the required number of conditions $m$ in the multipoint optimization is linked to the dimension of the design space $n$. From now on and throughout the rest of the thesis, the term gradient will mean gradient of the performance function with respect to the design variable unless otherwise stated.

To summarize Ref. [75], a parametric function $j(\chi, \alpha)$, $\alpha \in \mathbf{I}$ is adequately controlled by an aggregate objective function $J(\chi) = \sum_{k=1}^{m} \omega_k \; j(\chi, \alpha_k)$ over the whole operating conditions set $\mathbf{I}$ if no new descent direction of $j(\chi, \alpha)$ can be found when $\alpha$ varies in $\mathbf{I}$ compared to the descent directions given by $-\nabla_\chi j(\chi, \alpha)$ when $\alpha$ varies in $(\alpha_1, \ldots, \alpha_m)$. Because the gradient at each condition is a vector of dimension $n$, it is stated in Ref. [75] that taking at least $m \geq n$ is a necessary condition. At convergence of the gradient-based algorithm to a local minimum, the sum of the weighted gradients is zero in unconstrained cases, so the gradients are linearly dependent. If operating conditions are chosen such that the gradients are linearly independent at the initial guess, this tends to provide poor solutions. As a consequence, at least $n + 1$ conditions are taken in practice as expressed by Eq. (3.5),

$$m \geq n + 1. \tag{3.5}$$

Also in Ref. [75], Li et al. proposed an heuristic method with variable weights to avoid this costly dependence to the number of design variables.

Here new necessary and sufficient conditions for controlling the shape performance on $\mathbf{I}$ are given, and they are less restrictive than the condition of Eq. (3.5) that appears consequently to be neither necessary nor sufficient. These theorems provide the basis for a selection strategy for the condition set $\mathbf{I_m}$.

### 3.2.1 Geometrical approach of the gradient span

As an example, a lift-to-drag polar as a function of lift coefficient is plotted in Fig. 3.1. The optimization of this polar belongs to the class of multiple operating conditions problems. If two design parameters are considered, the gradients are then in the $\mathbb{R}^2$ plane. Thus, Fig. 3.1 also shows the gradient vectors of four conditions named A, B, C, and D. A basic graphical approach shows that if two gradients are linearly independent as the ones in A and C, then any other gradient such as the one of the condition B can be linearly decomposed on the previous ones. This is due to the fact that the set formed by the gradients at conditions A and C is a basis of the $\mathbb{R}^2$ plane. So

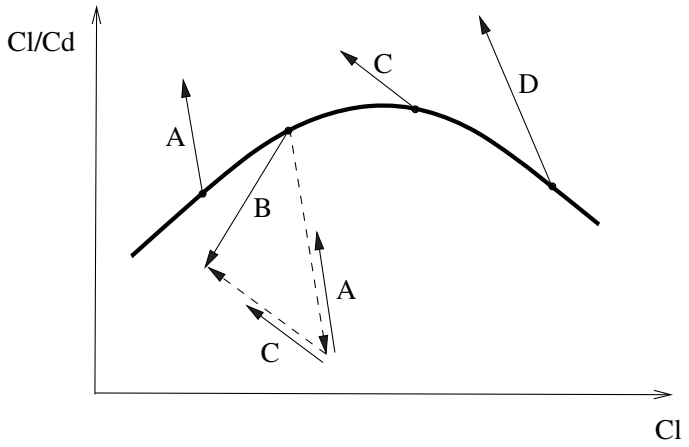



Figure 3.1: Lift over Drag / Lift polar and the gradients at four conditions in a design space of dimension two.

using the optimization algorithm with the contribution of the gradient at the condition B or of its linear decomposition on the other conditions is identical.

In a general context with $n$ design variables, the design space that contains any gradient is $\mathbb{R}^n$. The point is that it may be impossible to build a basis of $\mathbb{R}^n$ from the gradients computed at all the conditions in $\mathbf{I}$. When the set of gradients from all the conditions is a basis of $\mathbb{R}^n$, the condition of Eq. (3.5) becomes effective and it is our worst case. For this reason, we should focus on the vector space spanned by all descent directions of the objective when the operating conditions vary. In fact, it is sufficient to select a subset of gradients that is a basis of this vector space to setup the optimization problem, as we will demonstrate in the next sections.

### 3.2.2 Implicit additional operating conditions

In this section, the basic theorem on the equivalence between operating conditions modifications and weights modifications in aggregated objective functions gradients is demonstrated. The main hypothesis is that the operating conditions contained in the initial aggregate objective are well chosen so that no descent direction is missed in the operating condition ranges. Equality between two vector spaces is used to formulate it. On the other hand, the hypothesis appears to be minimal to obtain the present result as the reverse implication is proven. To the author's knowledge, the hypothesis formulation is new compared to the state of the art and leads to an original result.

Let us start with the definition of the different required ingredients. For the sake of simplicity, we consider only one operating condition variable and $\mathbf{I} \subset \mathbb{R}$, but the analysis can be extended to more than one dimension.

We first suppose the functional $j(\chi, \alpha) : \mathbb{R}^n \times \mathbf{I} \to \mathbb{R}$ only continuously differentiable, then the analysis is extended to twice continuously differentiable functionals.

We consider this functional and its gradients at points, $\mathbf{I_m} = (\alpha_1, \ldots, \alpha_m) \in \mathbf{I}^m$.

**Definition 1.** *We define two spanned vector spaces:*

$$\mathbf{K}_{I_m,\chi} = \mathrm{Span}\{\nabla_\chi j(\chi, \alpha),\ \alpha \in \mathbf{I_m}\},$$

*and*

$$\mathbf{K}_{I,x} = \mathrm{Span}\{\nabla_\chi j(\chi, \alpha),\ \alpha \in \mathbf{I}\}.$$

**Definition 2.** *Let us introduce two composite functionals:*

$$J_{m+1}(\chi) = \sum_{k=1}^{m+1} \omega_k j(\chi, \alpha_k), \qquad \text{and} \qquad J_m(\chi) = \sum_{k=1}^{m} \bar{\omega}_k j(\chi, \alpha_k),$$

*with* $(\omega_1, \ldots, \omega_{m+1}) \in \mathbb{R}^{m+1}$ *and* $(\bar{\omega}_1, \ldots, \bar{\omega}_m) \in \mathbb{R}^m$.

Now, let us analyze the relations between the two vector spaces $\mathbf{K}_{I_m,\chi}$ and $\mathbf{K}_{I,x}$ and, in particular, the equivalence between the following propositions:

**Proposition 1.**

$$\mathbf{K}_{I_m,\chi} = \mathbf{K}_{I,x}.$$

**Proposition 2.**

*Given* $\alpha_{m+1} \in \mathbf{I}$, $\omega_{m+1} \in \mathbb{R}^*$ *and* $(\omega_1, \ldots, \omega_m) \in \mathbb{R}^m$, *there exists* $(\bar{\omega}_1, \ldots, \bar{\omega}_m) \in \mathbb{R}^m$, *such that*

$$\sum_{k=1}^{m+1} \omega_k \ \nabla_\chi j(\chi, \alpha_k) = \sum_{k=1}^{m} \bar{\omega}_k \ \nabla_\chi j(\chi, \alpha_k).$$

**Proposition 3.**

*Given* $(\omega_1, \ldots, \omega_m) \in \mathbb{R}^m$, *the condition* $\sum_{k=1}^{m} \omega_k \ \nabla_\chi j(\chi, \alpha_k) = 0$ *implies that given* $\alpha_{m+1} \in \mathbf{I}$, *there exists* $(\bar{\omega}_1, \ldots, \bar{\omega}_{m+1}) \in (\mathbb{R}^m \times \mathbb{R}^*)$, *such that for all* $D \in \mathbb{R}^n$, *and* $t \in \mathbb{R}$, *we have* $\sum_{k=1}^{m+1} \bar{\omega}_k \ j(\chi + t \ D, \alpha_k) = \sum_{k=1}^{m+1} \bar{\omega}_k \ j(\chi, \alpha_k) + \mathbf{O}(t^2)$.

**Proposition 4.**

$$\textit{Given} \ \ \alpha \in \mathbf{I}, \ \textit{we have} \ \ \frac{\partial \nabla_\chi j(\chi, \alpha)}{\partial \alpha} \in \mathbf{K}_{I_m,\chi}.$$

**Theorem 1.** *If* $j \in C^1(\mathbb{R}^n \times \mathbf{I}, \mathbb{R})$, *then Propositions 1 and 2 are equivalent.*

*Proof.* Let us first prove that Proposition 1 ⇒ Proposition 2. As $\alpha_{m+1} \in \mathbf{I}$, by definition of $\mathbf{K}_{I,x}$, we have $\omega_{m+1} \nabla_\chi j(\chi, \alpha_{m+1}) \in \mathbf{K}_{I,x}$. And, by hypothesis, $\mathbf{K}_{I_m,\chi} = \mathbf{K}_{I,x}$, then $\omega_{m+1} \nabla_\chi j(\chi, \alpha_{m+1})$ is also in $\mathbf{K}_{I_m,\chi}$. As a consequence, this vector can be decomposed on the gradient set of $\mathbf{K}_{I_m,\chi}$.

$$\exists (\rho_1, \ldots, \rho_m) \in \mathbb{R}^m, \ \omega_{m+1} \nabla_\chi j(\chi, \alpha_{m+1}) = \sum_{k=1}^{m} \rho_k \nabla_\chi j(\chi, \alpha_k),$$

and

$$\sum_{k=1}^{m+1} \omega_k \ \nabla_\chi j(\chi, \alpha_k) = \sum_{k=1}^{m} \omega_k \ \nabla_\chi j(\chi, \alpha_k) + \omega_{m+1} \ \nabla_\chi j(\chi, \alpha_{m+1}),$$

so

$$\sum_{k=1}^{m+1} \omega_k \ \nabla_\chi j(\chi, \alpha_k) = \sum_{k=1}^{m} (\omega_k + \rho_k) \ \nabla_\chi j(\chi, \alpha_k).$$

Taking $\bar{\omega}_k = \omega_k + \rho_k$ gives Proposition 2.

Let us prove now that Proposition 2 $\Rightarrow$ Proposition 1. With the hypothesis of Proposition 2,

$$\sum_{k=1}^{m+1} \omega_k \ \nabla_\chi j(\chi, \alpha_k) = \sum_{k=1}^{m} \bar{\omega}_k \ \nabla_\chi j(\chi, \alpha_k),$$

is true for any $\omega_{m+1} \in \mathbb{R}^*$. In particular for some $\omega_{m+1} = 1$, and for any $\alpha_{m+1} \in \mathbf{I}$,

$$\nabla_\chi j(\chi, \alpha_{m+1}) + \sum_{k=1}^{m} \omega_k \ \nabla_\chi j(\chi, \alpha_k) = \sum_{k=1}^{m} \bar{\omega}_k \ \nabla_\chi j(\chi, \alpha_k).$$

Hence, for any $\alpha_{m+1} = \bar{\alpha}$:

$$\nabla_\chi j(\chi, \bar{\alpha}) = \sum_{k=1}^{m} (\bar{\omega}_k - \omega_k) \ \nabla_\chi j(\chi, \alpha_k).$$

But, by the definition of the span,

$$\mathbf{K}_{I,x} = \{\mathbf{g}; \mathbf{g} = \sum_{i=1}^{p} \lambda_i \nabla_\chi j(\chi, \bar{\alpha}_i) : p \in \mathbb{N}, \ (\lambda_1, \ldots, \lambda_p) \in \mathbb{R}^p, \ (\bar{\alpha}_1, \ldots, \bar{\alpha}_p) \in \mathbf{I}^p\},$$

for all $\mathbf{g} \in \mathbf{K}_{I,x}$:

$$\mathbf{g} = \sum_{i=1}^{p} \lambda_i \nabla_\chi j(\chi, \bar{\alpha}_i) = \sum_{i=1}^{p} \lambda_i \sum_{k=1}^{m} (\bar{\omega}_{i,k} - \omega_{i,k}) \ \nabla_\chi j(\chi, \alpha_k).$$

We have then, for all $\mathbf{g}$ in $\mathbf{K}_{I,x}$, $\mathbf{g}$ is a linear combination of elements from the $\mathbf{K}_{I_m,\chi}$ vector space and consequently is in this vector space, so $\mathbf{K}_{I,x} \subset \mathbf{K}_{I_m,\chi}$.

In addition because $\mathbf{I_m} \subset \mathbf{I}$, we already have $\mathbf{K}_{I,x} \supset \mathbf{K}_{I_m,\chi}$ and finally

$$\mathbf{K}_{I,x} = \mathbf{K}_{I_m,\chi}.$$

□

**Remark 1.** *Theorem 1 makes a counter-intuitive link between adding a condition to the aggregate objective function and perturbing its weights. This link is counter-intuitive since the weights selection problem are treated in the literature by completely different approaches from the operating conditions selection methods. The proof also gives the modification to the weights $\rho_k$, for building the new weighted sum in Proposition 2 in the argument:*
*$\omega_{m+1}\nabla_\chi j(\chi, \alpha_{m+1}) \in \mathbf{K}_{I_m,\chi}$. So, this vector can be decomposed on the gradient set of $\mathbf{K}_{I_m,\chi}$.*

**Remark 2.** *The argument is purely algebraic and independent of the properties of the functional $j : \chi \to j(\chi, \alpha)$. This is due to the fact that no variation of $\chi$ is considered.*

**Remark 3.** *In a worst case scenario, the dimension of $\mathbf{K}_{I,x}$ is $n$, which implies that $\mathbf{I_m}$ has to be of cardinal at least $n$ to ensure $\mathbf{K}_{I_m,\chi} = \mathbf{K}_{I,x}$. That depends on the properties of the functional $\nabla_\chi j : \alpha \to \nabla_\chi j(\chi, \alpha)$.*

### 3.2.3 Induced optimality condition on implicit additional conditions

The following theorem expresses how the zero gradient condition on a weighted sum of $m$ well chosen operating conditions implies the optimality of any derived multipoint problems built on $m+1$ conditions but specific weights. By optimality, we mean here no first order variation of the functional induced by any potential descent direction $D$ in the neighborhood of the design vector $\chi$, without distinction whether or not $\chi$ is effectively an extremum. It is a direct consequence of Theorem 1.

**Theorem 2.** *If $j \in C^1(\mathbb{R}^n \times \mathbf{I}, \mathbb{R})$, then Propositions 1 and 3 are equivalent.*

*Proof.* Theorem 1 has established the equivalence between the Proposition 1 and the Proposition **??**. The equivalence between Proposition 2 and Proposition 3 is proved here, which gives the equivalence between Proposition 1 and Proposition 3.

First, let us make the following assumptions.

$\forall(\omega_1, \ldots, \omega_m) \in \mathbb{R}^m$ and $\forall(\alpha_1, \ldots, \alpha_m) \in \mathbf{I}^m$, let us suppose that we have $\chi \in \mathbb{R}^n$ such that

$$\sum_{k=1}^{m} \omega_k \ \nabla_\chi j(\chi, \alpha_k) = 0. \tag{3.6}$$

Then observe that, $\forall(\bar{\omega}_1, \ldots, \bar{\omega}_m) \in \mathbb{R}^m$, $\forall \alpha_{m+1} \in \mathbf{I}$, $\forall D \in \mathbb{R}^n$ and $\forall t \in \mathbb{R}$, by Taylor expansion we have:

$$\sum_{k=1}^{m+1} \bar{\omega}_k \ j(\chi + t\ D, \alpha_k) = \sum_{k=1}^{m+1} \bar{\omega}_k \ j(\chi, \alpha_k) + t\ \langle D, \sum_{k=1}^{m+1} \bar{\omega}_k \ \nabla_\chi j(\chi, \alpha_k) \rangle + \mathbf{O}(t^2), \tag{3.7}$$

where $\langle ., . \rangle$ is the scalar product.

Let us first prove that Proposition 2 $\Rightarrow$ Proposition 3. Thanks to the hypothesis of Proposition 2,

$$\sum_{k=1}^{m+1} \omega_k \ \nabla_\chi j(\chi, \alpha_k) = \sum_{k=1}^{m} \bar{\omega}_k \ \nabla_\chi j(\chi, \alpha_k),$$

$$\sum_{k=1}^{m} \omega_k \ \nabla_\chi j(\chi, \alpha_k) = \sum_{k=1}^{m} \bar{\omega}_k \ \nabla_\chi j(\chi, \alpha_k) - \omega_{m+1} \nabla_\chi j(\chi, \alpha_{m+1}).$$

Let us choose $\bar{\omega}_{m+1} = -\omega_{m+1}$ then,

$$\sum_{k=1}^{m} \omega_k \ \nabla_\chi j(\chi, \alpha_k) = \sum_{k=1}^{m+1} \bar{\omega}_k \ \nabla_\chi j(\chi, \alpha_k).$$

Using Eq. (3.6), one can write:

$$\forall(\omega_1, \ldots, \omega_m) \in \mathbb{R}^m, \ \exists(\bar{\omega}_1, \ldots, \bar{\omega}_{m+1}) \in \mathbb{R}^m \times \mathbb{R}^*, \ \sum_{k=1}^{m+1} \bar{\omega}_k \ \nabla_\chi j(\chi, \alpha_k) = 0.$$

Replacing the above expression in Eq. (3.7), we get:

$$\sum_{k=1}^{m+1} \bar{\omega}_k \ j(\chi + t\ D, \alpha_k) = \sum_{k=1}^{m+1} \bar{\omega}_k \ j(\chi, \alpha_k) + \mathbf{O}(t^2).$$

Let us prove now that Proposition 3 ⇒ Proposition 2. Assuming Proposition 3, we have:

$$\sum_{k=1}^{m} \omega_k \ \nabla_\chi j(\chi, \alpha_k) = 0,$$

and

$$\sum_{k=1}^{m+1} \bar{\omega}_k \ j(\chi + t \ D, \alpha_k) = \sum_{k=1}^{m+1} \bar{\omega}_k \ j(\chi, \alpha_k) + \mathbf{O}(t^2).$$

So Eq. (3.7) becomes:

$$0 = t \ \langle D, \sum_{k=1}^{m+1} \bar{\omega}_k \ \nabla_\chi j(\chi, \alpha_k) \rangle + \mathbf{O}(t^2).$$

Due to the uniqueness of the Taylor expansion, every term of this $t$ polynomial is null, in particular

$$0 = \langle D, \sum_{k=1}^{m+1} \bar{\omega}_k \ \nabla_\chi j(\chi, \alpha_k) \rangle.$$

This equality is true for all $D$ in $\mathbb{R}^n$ so in particular for $D = \sum_{k=1}^{m+1} \bar{\omega}_k \ \nabla_\chi j(\chi, \alpha_k)$,

$$\left\| \sum_{k=1}^{m+1} \bar{\omega}_k \ \nabla_\chi j(\chi, \alpha_k) \right\|^2 = 0,$$

and then

$$\sum_{k=1}^{m+1} \bar{\omega}_k \ \nabla_\chi j(\chi, \alpha_k) = 0.$$

But, by hypothesis

$$\sum_{k=1}^{m} \omega_k \ \nabla_\chi j(\chi, \alpha_k) = 0,$$

so finally

$$\sum_{k=1}^{m+1} \bar{\omega}_k \ \nabla_\chi j(\chi, \alpha_k) = \sum_{k=1}^{m} \omega_k \ \nabla_\chi j(\chi, \alpha_k),$$

and this proposition is equivalent to Proposition 2. □

*Interpretation:* Theorem 1 shows that a Pareto stationary point for a problem associated with conditions $\{\alpha_k\}_{k\in[1...m]} \in \mathbf{I}^m$ such that $\mathbf{K}_{I_m,\chi} = \mathbf{K}_{I,x}$, is also Pareto stationary for any other problem with additional conditions taken from $\mathbf{I}$, as long as the weights of this new problem are positive. In Ref. [27], Désidéri defines Pareto stationarity as the existence of a convex combination of the gradient vectors that is null. When the objective is a smooth function of the design variables, it is a necessary condition for Pareto optimality.

In other words, the theorem gives a necessary condition for the choice of the discrete set $\mathbf{I_m}$ for Pareto stationarity on the continuous set $\mathbf{I}$. If the weights are positive, it is also a sufficient condition.

*On the sign of $\bar{\omega}_k$:* In these theorems, it is not ensured that the weights $\bar{\omega}_k$ are positive. We have $\bar{\omega}_k = \omega_k + \rho_k$, and $\rho_k$ is the component of the $m+1^{th}$ condition on the $k^{th}$ vector of the minimal gradient set. Then, a negative $\rho_k$ means that the $m+1^{th}$ condition is concurrent with the $k^{th}$

one. When $\bar{\omega}_k$ is negative, increasing the objective on the $k^{th}$ condition will decrease the aggregate objective function. This means that locally the condition is concurrent with the objective and its degradation leads to a better global performance in the sense defined by the initial problem.

To the author's knowledge, this result is new and shows that Eq. (3.5) proposed in [75] is over restrictive. As a matter of fact, if the two vector spaces $\mathbf{K}_{I_m,\chi}$ and $\mathbf{K}_{I,x}$ are equal, then, when $\chi$ is a local minimum of $J_m(\chi)$, there exists no perturbation of the design variables that can improve the performance of any operating condition at first order without degrading the others. When the dimension of $\mathbf{K}_{I,x}$ is less than the number of design variables, our result implies that incorporating more operating conditions than design variables in the multipoint objective function is not a necessary condition for this property.

#### 3.2.4 Second-order derivatives

Theorem 3 gives a more intuitive interpretation for the equality in Proposition 1. It expresses that if the derivative of the gradient vector with respect to $\alpha$ lies in the subspace $\mathbf{K}_{I_m,\chi}$, then the same subspace contains all gradients for $\alpha$ in $\mathbf{I}$. In addition, the two conditions are equivalent.

We need now to assume that $j$ is twice continuously differentiable.

**Theorem 3.** *If $j \in C^2(\mathbb{R}^n \times \mathbf{I}, \mathbb{R})$, then Proposition 1 and Proposition 4 are equivalent.*

*Proof.* Let us first prove that Proposition 1 $\Rightarrow$ Proposition 4. By definition of the derivative:

$$\forall \alpha \in \mathbf{I},\ \forall t \in \mathbb{R}\ t+\alpha \in \mathbf{I},\ \nabla_\chi j(\chi, \alpha+t) = \nabla_\chi j(\chi,\alpha) + t\ \frac{\partial \nabla_\chi j(\chi,\alpha)}{\partial \alpha} + \mathbf{O}(t^2),$$

$$\lim_{t\to 0} \frac{\nabla_\chi j(\chi,\alpha+t) - \nabla_\chi j(\chi,\alpha)}{t} = \frac{\partial \nabla_\chi j(\chi,\alpha)}{\partial \alpha}.$$

We have $\nabla_\chi j(\chi,\alpha+t) \in \mathbf{K}_{I,x}$, and $\nabla_\chi j(\chi,\alpha) \in \mathbf{K}_{I,x}$. Also $\mathbf{K}_{I,x}$ being a vector space, it contains any sum of two of its vectors.

$$\frac{\nabla_\chi j(\chi,\alpha+t) - \nabla_\chi j(\chi,\alpha)}{t} \in \mathbf{K}_{I,x}.$$

Assuming Proposition 1, we have $\mathbf{K}_{I_m,\chi} = \mathbf{K}_{I,x}$, so:

$$\frac{\nabla_\chi j(\chi,\alpha+t) - \nabla_\chi j(\chi,\alpha)}{t} \in \mathbf{K}_{I_m,\chi}.$$

The limit of a continuous function ($j \in \mathbf{C}^2(\mathbb{R}^n \times \mathbf{I}, \mathbb{R})$) in a space being in the closure of this space,

$$\lim_{t\to 0} \frac{\nabla_\chi j(\chi,\alpha+t) - \nabla_\chi j(\chi,\alpha)}{t} \in \overline{\mathbf{K}}_{I_m,\chi},$$

so we have now:

$$\frac{\partial \nabla_\chi j(\chi,\alpha)}{\partial \alpha} \in \overline{\mathbf{K}}_{I_m,\chi}.$$

Because $\mathbf{K}_{I_m,\chi} \subset \mathbb{R}^n$, it is a finite dimensionnal subspace of the normed vector space $\mathbb{R}^n$, thus $\mathbf{K}_{I_m,\chi}$ is closed and $\overline{\mathbf{K}}_{I_m,\chi} = \mathbf{K}_{I_m,\chi}$. That finally gives the Proposition 4.

$$\frac{\partial \nabla_\chi j(\chi,\alpha)}{\partial \alpha} \in \mathbf{K}_{I_m,\chi}.$$

Let us prove now that Proposition 4 ⇒ Proposition 1. Since the derivatives of the cost functional $j$ are continuous by hypothesis, for all $(\alpha, \alpha_0) \in \mathbf{I}^2$

$$\int_{\alpha_0}^{\alpha} \frac{\partial \nabla_\chi j(\chi, t)}{\partial t} dt + \nabla_\chi j(\chi, \alpha_0) = \nabla_\chi j(\chi, \alpha).$$

For all $k \in \mathbb{N}^*$, we define the Riemann sum $S_k$:

$$S_k = \frac{\alpha - \alpha_0}{k} \sum_{i=1}^{k} \frac{\partial \nabla_\chi j}{\partial \alpha}(\chi, \alpha_0 + i \ \frac{\alpha - \alpha_0}{k}),$$

and because, $j \in \mathbf{C}^2(\mathbb{R}^n \times \mathbf{I}, \mathbb{R})$, the sequence converges to the integral:

$$\lim_{k \to +\infty} S_k = \int_{\alpha_0}^{\alpha} \frac{\partial \nabla_\chi j(\chi, t)}{\partial t} dt.$$

The Proposition 4 expresses that for all $\alpha$,

$$\frac{\partial \nabla_\chi j}{\partial \alpha} \in \mathbf{K}_{I_m, \chi}.$$

Because $\mathbf{K}_{I_m,\chi}$ is a vector space, we also have:

$$S_k = \frac{\alpha - \alpha_0}{k} \sum_{i=1}^{k} \frac{\partial \nabla_\chi j}{\partial \alpha}(\chi, \alpha_0 + i \ \frac{\alpha - \alpha_0}{k}) \in \mathbf{K}_{I_m, \chi}.$$

If we take now $\alpha_0 \in \mathbf{I_m}$, $\nabla_\chi j(\chi, \alpha_0)$ is in $\mathbf{K}_{I_m,\chi}$, and then:

$$S_k + \nabla_\chi j(\chi, \alpha_0) \in \mathbf{K}_{I_m, \chi}.$$

The series $S_k + \nabla_\chi j(\chi, \alpha_0)$, with the same argument as in the first part of the proof, converges in the close finite dimensional vector space $\mathbf{K}_{I_m,\chi}$:

$$\lim_{k \to +\infty} S_k + \nabla_\chi j(\chi, \alpha_0) = \int_{\alpha_0}^{\alpha} \frac{\partial \nabla_\chi j(\chi, t)}{\partial t} dt + \nabla_\chi j(\chi, \alpha_0) = \nabla_\chi j(\chi, \alpha) \in \mathbf{K}_{I_m, \chi}.$$

□

Theorem 3 shows that the variations of the function with respect to the operating conditions link the gradients at each operating condition. Exploiting this link when the condition of Proposition 4 is ensured gives the necessary and sufficient Proposition 1 for the Theorem 1 and Theorem 2. That shows how the parametric optimization problem is specific compared to a multi-objective problem when the aggregated objectives come from different disciplines, with their own state equations, so when the multipoint problem does not raise from a continuous function $j : (\chi, \alpha) \to j(\chi, \alpha)$, but $m$ from different $j_k, x :\to j_k(\chi)$ functions.

#### 3.2.5 Additional properties

**Proposition 5.** *For any two operating condition ranges $\mathbf{I_1} \subset \mathbb{R}$ and $\mathbf{I_2} \subset \mathbb{R}$ such that $\mathbf{I_1} \subset \mathbf{I_2}$, the required operating conditions number to setup a multipoint optimization problem on $\mathbf{I_1}$ and $\mathbf{I_2}$, are respectively $m_1$ and $m_2$ such that $m_1 \leq m_2$.*

*Proof.* By definition $m_1 = dim\left(\text{Span}\{\nabla_\chi j(\chi,\alpha),\, \forall\alpha \in \mathbf{I_1}\}\right)$ and $m_2 = dim\left(\text{Span}\{\nabla_\chi j(\chi,\alpha),\, \forall\alpha \in \mathbf{I_2}\}\right)$.

Because $\mathbf{I_1} \subset \mathbf{I_2}$ we have, $\text{Span}\{\nabla_\chi j(\chi,\alpha),\, \forall\alpha \in \mathbf{I_1}\} \subset \text{Span}\{\nabla_\chi j(\chi,\alpha),\, \forall\alpha \in \mathbf{I_2}\}$,

we obtain $dim\left(\text{Span}\{\nabla_\chi j(\chi,\alpha),\, \forall\alpha \in \mathbf{I_1}\}\right) \leq dim\left(\text{Span}\{\nabla_\chi j(\chi,\alpha),\, \forall\alpha \in \mathbf{I_2}\}\right)$ and finally $m_1 \leq m_2$ □

**Proposition 6.** *For an objective function with separated design variables and operating conditions, ie. when there exist two functions $g$ and $h$ such that for all $\chi$ and for all $\alpha$, $j(\chi,\alpha) = g(\chi)h(\alpha)$, then we have, for all operating condition ranges $\mathbf{I} \subset \mathbb{R}$, the number of operating conditions required to setup a multipoint optimization problem on the range $\mathbf{I}$ for the function $j$ is 1.*

*Proof.* By definition $m = dim\left(\text{Span}\{\nabla_\chi g(\chi)h(\alpha),\, \forall\alpha \in \mathbf{I}\}\right) = dim\left(\text{Span}\{h(\alpha)\nabla_\chi g(\chi),\, \forall\alpha \in \mathbf{I}\}\right)$.

Since for all $\alpha$, $h(\alpha) \in \mathbb{R}$, we have $\text{Span}\{\nabla_\chi g(\chi)h(\alpha),\, \forall\alpha \in \mathbf{I}\} = \text{Span}\{\nabla_\chi g(\chi),\, \forall\alpha \in \mathbf{I}\}$.

Besides, $\text{Span}\{\nabla_\chi g(\chi),\, \forall\alpha \in \mathbf{I}\} = \text{Span}\{\nabla_\chi g(\chi)\}$ and $\nabla_\chi g(\chi)$ being a single vector, $m = dim\left(\text{Span}\{\nabla_\chi g(\chi)\}\right) = 1$ □

## 3.3 Gradient Span Analysis Algorithm

To apply the previous theorems in practical applications, a minimal set of conditions $\mathbf{I_m}$ that verifies the condition $\mathbf{K}_{I_m,\chi} = \mathbf{K}_{I,x}$ for the multipoint optimization has to be built. Thus, the optimization problem requires a minimum amount of expensive computations of the functions $j(\chi,\alpha_k)$ and their gradients $\nabla_\chi j(\chi,\alpha_k)$. The Gradient Span Analysis (GSA) method described in algorithm 4 is proposed for building such a set. This algorithm is used to obtain an orthonormal basis of $\mathbf{K}_{I_m,\chi}$, such that $\mathbf{K}_{I_m,\chi}$ contains every vector of $\mathbf{K}_{I_m,\chi}$, where $\mathbf{I_M}$ is an initial sample of size $M$ in $\mathbf{I}$, with a maximal relative error of $\epsilon$, and $m$ as small as possible.

The GSA algorithm is based on the repeated use of modified Gram-Schmidt (MGS) processes shown by the algorithm 2. The MGS process gives an orthonormal basis of a vector set, e.g. the gradient set $\{\nabla_\chi j(\chi,\alpha_k),\ \alpha_k \in \mathbf{I}_M\}$. $\pi_u$ represents the projection operator. The GSA algorithm

**Algorithm 2** Modified Gram-Schmidt process

**for** $j = 1 \to M$ **do**
  $q_j \leftarrow \nabla_\chi j(\chi,\alpha_j)$
  **for** $i = 1 \to j-1$ **do**
    $q_j \leftarrow q_j - \pi_{q_i}(q_j)$ with $\pi_u(v) = \frac{\langle v,u\rangle}{\langle u,u\rangle}\, u$
  **end for**
  $q_j \leftarrow \frac{q_j}{\|q_j\|}$
**end for**

relies on two rules which are:

- to take into account more vectors than the dimension of the gradient space, and
- to find the minimal subset of the gradient set that spans $\mathbf{K}_{I,x}$ for a given projection error.

The algorithm starts by building a fine sample $\mathbf{I_M}$ of the condition set $\mathbf{I}$ that will help to get an estimation of $\mathbf{K}_{I,x}$. Then conditions are chosen and added to the set $\mathbf{I_m}$ until $\mathbf{K}_{I_m,\chi} = \mathbf{K}_{I,x}$. At

each step, the condition that maximizes the quantification of the intersection $\mathbf{K}_{I_m,\chi} \cap \mathbf{K}_{I_m,\chi}$ given by Eq. (3.8) is taken from available conditions in $\mathbf{I_M} \setminus \mathbf{I_m}$.

$$c(\mathbf{I_m}) = \text{Card}\{\alpha \in \mathbf{I_M}, ||\nabla_\chi j(\chi,\alpha) - Proj(\nabla_\chi j(\chi,\alpha), \mathbf{K}_{I_m,\chi})|| < \epsilon ||\nabla_\chi j(\chi,\alpha)||\}. \tag{3.8}$$

A potential set $\mathbf{I_{m1}}$ will then be preferred to $\mathbf{I_{m2}}$ if $c(\mathbf{I_{m1}}) > c(\mathbf{I_{m2}})$.

As for the choice of $\epsilon$, two strategies are possible. The threshold $\epsilon$ can be:

- the stopping criteria of the optimization algorithm:

  $\chi = \chi_{opt} \Leftrightarrow ||\nabla J(\chi)|| < \epsilon$, or

- the norm of the objective function gradient times a small constant:

  $\epsilon = 10^{-3}||\nabla_\chi J(\chi)||$.

For an easier understanding, an algorithmic description of GSA is proposed in Alg. 3, followed by a pseudo-code description in Alg. 4.

**Algorithm 3** Gradient Span Analysis Algorithm (GSA)

Let $\epsilon \in \mathbb{R}$, $\epsilon > 0$ and $\mathbf{I_M} = \{\alpha_1, \ldots, \alpha_M\} \subset \mathbf{I}^M$ be a set of operating conditions.
**Step 0.** Initialize the set of available operating conditions: set $\mathbf{A} = \mathbf{I_M}$.
Initialize the set of selected operating conditions: set $\mathbf{I_m} = \{\}$, set $j = 1$.
**Step 1.** Initialize the counter: set $\bar{c} = 0$. Initialize the set of tested operating conditions at the $j$-th iteration: $\mathbf{T_j} = \{\}$.
**Step 2.** Take a non-tested operating condition $\alpha \in \mathbf{A} \backslash \mathbf{T_j}$, update the set of tested operating conditions $\mathbf{T_j} = \mathbf{T_j} \cup \{\alpha\}$, set $q_j^1 = \nabla_\chi j(\chi,\alpha)$, and $i = 1$.
**Step 4.** Apply a MGS iteration on $q_j^1$:
while $i < j$, set $q_j^{i+1} = q_j^i - \pi_{q_i}(q_j^i)$ and set $i = i + 1$.
**Step 5.** Set $c = 1$, set $p = 1$.
If $\|q_j^j\| = 0$, go to step 2; otherwise set $q_j = \frac{q_j^j}{\|q_j^j\|}$.
**Step 6.** Set $v_{j,p}^0 = \nabla_\chi j(\chi,\alpha_p)$ and set $k = 0$.
**Step 7.** Project $\nabla_\chi j(\chi,\alpha_p)$ on $\text{Span}\{q_1, \ldots, q_j\}$:
while $k < j$, set $v_{j,p}^{k+1} = v_{j,p}^k - \pi_{q_k}(v_{j,p}^k)$ and set $k = k + 1$.
**Step 8.** If $\|v_{j,i}^j\| < \epsilon \|\nabla_\chi j(\chi,\alpha_p)\|$, set $c = c + 1$.
**Step 9.** If $p < M$, set $p = p + 1$ and go to step 6.
**Step 10.** If $c > \bar{c}$, set $\bar{c} = c$ and set $\bar{q} = q_j$ and set $\bar{\alpha} = \alpha$.
**Step 11.** If $\mathbf{A} \backslash \mathbf{T_j} \neq \{\}$ go to Step 2; otherwise set $q_j := \bar{q}$, and update the conditions sets via $\mathbf{A} = \mathbf{A} \backslash \{\bar{\alpha}\}$ and $\mathbf{I_m} = \mathbf{I_m} \cup \{\bar{\alpha}\}$.
**Step 12.** If $c_m = M$, stop; otherwise set $j = j + 1$ and go to Step 1.
Step 10. can be replaced by
**Step 10'.** If $c = \bar{c}$ and $min\{\|\alpha - \tilde{\alpha}\|, \tilde{\alpha} \in \mathbf{I_m}\} > min\{\|\bar{\alpha} - \tilde{\alpha}\|, \tilde{\alpha} \in \mathbf{I_m}\}$, or $c > \bar{c}$, set $\bar{c} = c$ and set $\bar{q} = q_j$ and set $\bar{\alpha} = \alpha$. This gives the preference to operating conditions that are more distant to the already selected ones when they lead to identical gradient spans.

**Algorithm 4** Gradient Span Analysis Algorithm (GSA)

$indexes \leftarrow \{1 \dots M\}$
**for** $j = 1 \rightarrow M$ **do**
  $c_m \leftarrow 0$
  **for** $n \in indexes$ **do**
    $q_j \leftarrow \nabla_\chi j(\chi, \alpha_j)$
    **for** $i = 1 \rightarrow j-1$ **do**
      $q_j \leftarrow q_j - \pi_{q_i}(q_j)$
    **end for**
    $q_j \leftarrow \frac{q_j}{\|q_j\|}$
    $c \leftarrow 0$
    **for** $i = 1 \rightarrow M$ **do**
      $v \leftarrow \nabla_\chi j(\chi, \alpha_i)$
      **for** $k = 1 \rightarrow j$ **do**
        $v \leftarrow v - \pi_{q_k}(v)$
      **end for**
      **if** $\|v\| < \epsilon \|\nabla_\chi j(\chi, \alpha_i)\|$ **then**
        $c \leftarrow c + 1$
      **end if**
    **end for**
    **if** $c > c_m$ **then**
      $q_m \leftarrow q_j$
      $c_m \leftarrow c$
      $n_m \leftarrow n$
    **end if**
  **end for**
  $q_j \leftarrow q_m$
  $indexes \leftarrow indexes \backslash \{n_m\}$
  **if** $c_m = M$ **then**
    END
  **end if**
**end for**

*Minimality of the set* $\mathbf{I_m}$*:* It is not proven that the set of operating conditions $\mathbf{I_m}$ given by the GSA algorithm is minimal. Nevertheless, it is guaranteed that the added condition is optimally chosen at each step. But at the end of the process, another set of lower dimension that also spans $\mathbf{K}_{I,x}$ with the same maximal error could exist. To ensure this global optimality, all combinations of vectors should be tested, which would lead to the orthonormalization of $\binom{M}{n}$ sets. This number quickly becomes too high to use the method. For instance, the analysis of a typical 3D wing optimization problem with 100 variables and 10 conditions would require $10^{13}$ Gram-Schmidt processes.

## 3.4 On the similarities and differences between GSA and MGDA

The MGDA (*Multiple-Gradient Descent Algorithm*) is a multi-objective optimization algorithm for which several variants have been proposed [27, 28]. MGDA has similarities with the GSA algorithm. The main characteristics of the two algorithms are briefly outlined in the following.

- MGDA is a family of multi-objective optimization descent methods; thus, it is intended as an

optimization solution method.

- MGDA is based on a concept of Pareto-stationarity related to the convex hull of the gradients associated with with an arbitrary set of differentiable objective-functions of the same design vector.
- Based on these gradients, a descent direction common to all objective-functions is identified and employed in a descent step.
- One variant of MGDA uses a Gram Schmidt process applied the gradients to this aim; as a result, the objective functions are split into two subgroups: for the first subgroup, the directional derivative is equal to some positive constant $\sigma$; for the complementary subgroup, the directional derivatives are at least equal to $a\sigma$, where $a$ is a user-specified cut-off constant $(0 < a < 1)$.

About GSA:

- GSA is used to setup optimization problems. It aims at finding operating conditions to be incorporated in a multipoint optimization problem.
- GSA takes a set of gradients as input and gives a set of operating conditions as output.
- GSA uses Modified Gram Schmidt processes on the gradients in order to detect of the linear dependancies between the objectives gradients at the different operating conditions.

## 3.5 Computing an equivalent problem

In this section, for an arbitrary given multipoint problem with a set of $M$ operating conditions, we aim at building the weights of an equivalent problem with a minimal set of $m$ operating conditions selected by the GSA algorithm 4. Thanks to the GSA algorithm, $\mathbf{I_m} = (\bar{\alpha}_1, \ldots, \bar{\alpha}_m)$, a subset of $\mathbf{I_M}$ is built such that $\mathbf{K}_{\mathbf{I_m},\chi} = \mathbf{K}_{\mathbf{I_m},\chi}$. The original objective function gradient is:

$$\nabla_\chi J(\chi) = \sum_{k=1}^{M} \omega_k \ \nabla_\chi j(\chi, \alpha_k).$$

That quantity can now be computed from less operating conditions and modified weights.

$$\forall \alpha_k \in \mathbf{I},\ \forall \omega_k \in \mathbb{R},\ \exists (\omega_{k,1}, \ldots, \omega_{k,m}) \in \mathbb{R}^m,\ \omega_k \nabla_\chi j(\chi, \alpha_k) = \sum_{p=1}^{m} \omega_{k,p} \ \nabla_\chi j(\chi, \bar{\alpha}_p).$$

A set of $M$ similar linear systems are defined by Eq. (3.9). For all $k \in (1, 2, \ldots, M)$,

$$\omega_k \nabla_\chi j(\chi, \alpha_k) = \begin{bmatrix} \frac{\partial}{\partial \chi_1} j(\chi, \bar{\alpha}_1) & & \cdots & & \frac{\partial}{\partial \chi_1} j(\chi, \bar{\alpha}_m) \\ & \ddots & & & \\ & & \frac{\partial}{\partial \chi_i} j(\chi, \bar{\alpha}_p) & & \\ & & & \ddots & \\ \frac{\partial}{\partial \chi_n} j(\chi, \bar{\alpha}_1) & & \cdots & & \frac{\partial}{\partial \chi_n} j(\chi, \bar{\alpha}_m) \end{bmatrix} \begin{bmatrix} \omega_{k,1} \\ \cdots \\ \omega_{k,p} \\ \cdots \\ \omega_{k,m} \end{bmatrix}. \tag{3.9}$$

The matrix is not necessarily square ($m \leq n$) so a least-square solver is used. However, as by hypothesis the GSA algorithm builds $\mathbf{K}_{I_m,\chi}$ such that $\mathbf{K}_{I_m,\chi} = \mathbf{K}_{I,x}$:

$$\forall \alpha_k \in \mathbf{I},\ \nabla_\chi j(\chi, \alpha_k) \in \text{Span}\{\nabla_\chi j(\chi, \alpha_1), \ldots, \nabla_\chi j(\chi, \alpha_m)\},$$

the problem has at least one solution. Solving the $M$ linear systems of Eq. (3.9) gives finally:

$$\nabla_\chi J(\chi) = \sum_{k=1}^{M} \sum_{p=1}^{m} \omega_{k,p}\ \nabla_\chi j(\chi, \bar{\alpha}_p).$$

Because $\nabla_\chi j(\chi, \bar{\alpha}_p)$ does not depend on $k$, a new set of weights can be defined in Eq. (3.10).

$$\bar{\omega}_p = \sum_{k=1}^{M} \omega_{k,p}. \tag{3.10}$$

The gradient of the objective function can be computed with Eq. (3.11).

$$\nabla_\chi J(\chi) = \sum_{p=1}^{m} (\sum_{k=1}^{M} \omega_{k,p})\ \nabla_\chi j(\chi, \bar{\alpha}_p) = \sum_{p=1}^{m} \bar{\omega}_p\ \nabla_\chi j(\chi, \bar{\alpha}_p) = \sum_{k=1}^{M} \omega_k\ \nabla_\chi j(\chi, \alpha_k). \tag{3.11}$$

We have $m \leq M$, as a consequence the method saves $M - m$ costly evaluations of the gradient. It can be noted again that the case $dim(\mathbf{K}_{I_m,\chi}) = n$, that leads to $m = M$ is the worst one.

# CHAPTER 4

## Application of GSA to a quasi-analytic wing optimization

***Summary*** In this chapter, applications of the GSA approach to a quasi-analytic wing twist optimization using a lifting line model with adjoint is performed. An optimization problem involving finely sampled operating conditions is solved. An identical solution is found by solving an alternative optimization problem proposed by the GSA method, at lower computational cost, demonstrating the interest of the approach.

***Résumé*** Dans ce chapitre, des applications de l'approche GSA à l'optimisation de vrillage d'une voilure sont réalisées en se basant sur un modèle de ligne portante quasi-analytique disposant de l'adjoint discret. Un problème d'optimisation dont les conditions de vol ont été finement discrétisées est résolu. Une solution identique est ensuite atteinte par la résolution d'un problème d'optimisation équivalent, obtenu par la méthode GSA, à un coût moindre et démontrant ainsi l'intérêt de l'approche.

## 4.1 The lifting line model

In this section, Theorem 1 and Theorem 2 are applied to a multipoint wing optimization problem. The objective is to minimize the drag of a wing over a range of lift coefficients using a gradient-based algorithm. A lifting line model estimates the lift and drag [109], the design variables are the twist angles of airfoils and the gradients are computed using an adjoint method.

The lifting line theory is a linearized fluid model that gives an estimation of wing's performance. More precisely, it enables the computation of the lift and drag induced by the lift. The flow around the wing is assumed to be potential so non-viscous, in low Mach number and the rotational of the velocity is supposed to be null. Potential sources are associated with wing sections (i.e. airfoils). Their interference models the interactions between airfoils. In this theory, the circulations defined by Eq. (4.1) are the state variables, we have then $U = \gamma$ in the formalism of Eq. (3.1).

$$\gamma(y) = \frac{1}{2} C_l(y, \alpha_{\text{eff}}(y))\, c(y) U_\infty, \tag{4.1}$$

where $C_l(y, \alpha_{eff}(y))$ is the lift coefficient of the airfoil of chord $c(y)$ at span $y$ and angle of attack $\alpha_{eff}(y)$. The effective angle of attack given by Eq. (4.2) is the difference between the geometrical angle of attack and the angle of attack induced by the lift as shown in Fig. 4.1. Infinite angle of attack is the angle between the airfoil and the infinite air speed. Effective angle of attack is the angle between the local air speed and the airfoil. The geometrical angle of attack is the sum of wing angle of attack and local twist of the airfoil that is a local rotation of the airfoil around $y$ axis, see Fig. 4.2.

$$\alpha_{eff} = \alpha_\infty + \alpha_{\text{twist}} - \alpha_{\text{ind}}. \tag{4.2}$$

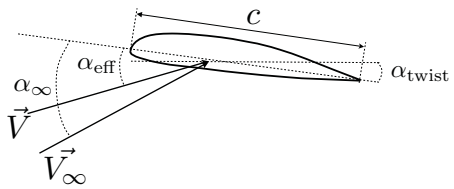



Figure 4.1: An airfoil of the lifting line model

Induced angles of attack due to downwash velocities given by Eq. (4.3) are computed by summing the effects of all airfoil circulations. The downwash velocity is the difference between air speed of the aircraft and local air speed of wing sections, $\vec{V} - \vec{V}_\infty$ as shown in Fig. 4.1. $s$ is the half wingspan, $y_0$ is the current airfoil position.

$$\alpha_{\text{ind}}(y_0) = \frac{1}{4\pi} PV \int_{-s}^{s} \frac{d\gamma(y)}{dy} \frac{dy}{y - y_0}, \tag{4.3}$$

where PV represents the Cauchy principal value of the integral. The circulation on each airfoil must then be computed.

In that aim, we express with a residual function Eq. (4.4) that the difference between a given circulation $\gamma$ and the circulation given by the lifts under effective angles of attack due to that circulation should be zero. To solve this coupled problem, a Newton method can be used taking circulation as the state vector.

$$R(\gamma, y_0) = \frac{1}{2} C_l \left( y_0, \alpha_\infty + \alpha(y_0) - \frac{1}{4\pi} PV \int_{-s}^{s} \frac{d\gamma(y)}{dy} \frac{dy}{y - y_0} \right) c(y_0) U_\infty - \gamma(y_0), \tag{4.4}$$

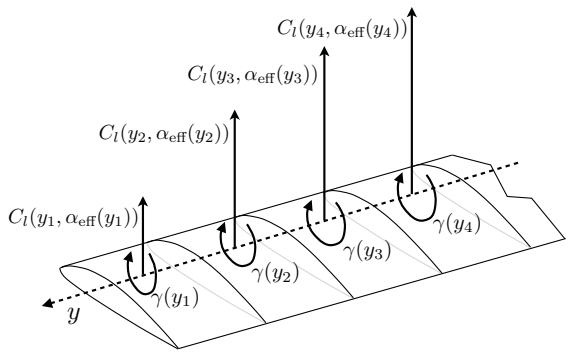



Figure 4.2: A wing in the lifting line model

where the local twist $\alpha_{\text{twist}}$ is taken as the design variable $\chi(y_0)$. We take here the twist angles as design variables because they drive the lift repartition for a given planform, i.e. fixed chords and airfoil positions. Finding the optimal twist vector is a typical lift repartition design problem in aerodynamics and one of the main lifting line model application.

## 4.2 The discrete model and its computation

A cartesian discretization of $y$ with a step $\Delta y$ gives discrete vectors $\chi$, $c$ and $\Gamma$ of size $p = 2s/\Delta y$, respectively in Eqs. (4.5), (4.6) and (4.7).

$$\chi = \Big[\chi_1, \ldots, \chi_i, \ldots, \chi_p\Big]^T = \Big[x(y_1), \ldots, x(y_i), \ldots, x(y_p)\Big]^T, \tag{4.5}$$

$$c = \Big[c_1, \ldots, c_i, \ldots, c_p\Big]^T = \Big[c(y_1), \ldots, c(y_i), \ldots, c(y_p)\Big]^T, \tag{4.6}$$

$$\Gamma = \Big[\gamma_1, \ldots, \gamma_i, \ldots, \gamma_p\Big]^T = \Big[\gamma(y_1), \ldots, \gamma(y_i), \ldots, \gamma(y_p)\Big]^T. \tag{4.7}$$

Induced angles of attack and residuals are obtained in Eqs. (4.8) and (4.9).

$$\alpha_{\text{ind},i} = \frac{1}{4\pi} \sum_{k=0, k\neq i}^{p} \frac{d\gamma_k}{dy} \frac{\Delta y}{y_k - y_i} \tag{4.8}$$

$$R(\Gamma, \chi) = \begin{bmatrix} R_1 \\ \ldots \\ R_i \\ \ldots \\ R_p \end{bmatrix} = \frac{1}{2} U_\infty \begin{bmatrix} C_{l1}(\alpha_\infty + \chi_1 - \frac{1}{4\pi} \sum_{k=0,k\neq 1}^{p} \frac{d\gamma_k}{dy} \frac{\Delta y}{y_k - y_1}) c_1 \\ \ldots \\ C_{li}(\alpha_\infty + \chi_i - \frac{1}{4\pi} \sum_{k=0,k\neq i}^{p} \frac{d\gamma_k}{dy} \frac{\Delta y}{y_k - y_i}) c_i \\ \ldots \\ C_{lp}(\alpha_\infty + \chi_p - \frac{1}{4\pi} \sum_{k=0,k\neq p}^{p} \frac{d\gamma_k}{dy} \frac{\Delta y}{y_k - y_p}) c_p \end{bmatrix} - \Gamma. \tag{4.9}$$

The Newton method of Eq. (4.10) gives the next iterate using the derivatives of the residual with respect to the state variables. A relaxation factor $\beta$ is used to numerically stabilize the method.

$$\Gamma_{n+1} = \Gamma_n - (1 - \beta) \left[\frac{\partial R}{\partial \Gamma}(\Gamma, \chi)\right]^{-1} \Gamma. \tag{4.10}$$

## 4.3 Discrete adjoint and computation of gradients

The gradient of the residual with respect to the state variables is used in the discrete adjoint formulation in order to compute the gradient of the objective function $J(\Gamma, \chi)$ with respect to the design variables $\chi$. Geometrical parameters such as chords, twist angles or airfoil positions can be used as design parameters. The linear adjoint equation is defined in Eq. (4.11) which solution gives the adjoint vector $\lambda_J$.

$$\frac{\partial J(\Gamma,\chi)}{\partial \Gamma} + \lambda_J^T \frac{\partial R(\Gamma,\chi)}{\partial \Gamma} = 0. \tag{4.11}$$

Finally the gradient is assembled as in Eq. (4.12).

$$\nabla_\chi J(\Gamma,\chi) = \frac{\partial J(\Gamma,\chi)}{\partial \chi} + \lambda_J^T \frac{\partial R(\Gamma,\chi)}{\partial x}. \tag{4.12}$$

The computation of the derivatives of the residuals with respect to the circulation is done in Eq. (4.13).

$$\begin{aligned} &\forall (i,j) \in [1..p]^2,\ \frac{\partial R_i(\Gamma,\chi)}{\partial \gamma_j} = \\ &\frac{1}{2} U_\infty \frac{\partial \alpha_{\text{ind},i}}{\partial \gamma_j} \frac{dC_{li}}{d\alpha} (\alpha_\infty + \chi_i - \frac{1}{4\pi} \sum_{k=0,k\neq i}^{p} \frac{d\gamma_k}{dy} \frac{\Delta y}{y_k - y_i}) c_i - \delta_{i,k}. \end{aligned} \tag{4.13}$$

Derivation of the induced angle of attack with respect to the circulation is given by Eq. (4.14).

$$\frac{\partial \alpha_{\text{ind},i}}{\partial \gamma_j} = \frac{1}{4\pi} \sum_{k=0,k\neq i}^{p} \frac{d}{d\gamma_j} (\frac{d\gamma_k}{dy}) \frac{c_i}{y_j - y_i}, \tag{4.14}$$

where the derivative $\frac{d}{d\gamma_j}(\frac{d\gamma_k}{dy})$ depends on the discretization scheme used for the computation of $\frac{d\gamma_k}{dy}$. Here, second-order centered finite differences are used. The derivatives of the residuals with respect to the design variables, required in the gradient assembly is computed in Eq. (4.15).

$$\frac{\partial R_i(\Gamma,\chi)}{\partial \chi} = \frac{1}{2} U_\infty \frac{dC_{li}}{d\alpha} (\alpha_\infty + \chi_i - \sum_{k=0,k\neq i}^{p} \frac{d\gamma_k}{dy} \frac{\Delta y}{y_k - y_i}) c_i, \tag{4.15}$$

where the functions of interest: the lift and the drag, are given by Eq. (4.16) and Eq. (4.17) respectively.

$$C_l = \frac{2\Delta y}{S} \sum_{i=0}^{p} \gamma_i, \tag{4.16}$$

$$Cd_{\text{ind}} = \frac{2\Delta y}{S} \sum_{i=0}^{p} \gamma_i \alpha_{\text{ind},i}. \tag{4.17}$$

The source terms of the adjoint equations for lift and drag are Eqs. (4.18)and (4.19).

$$\frac{\partial C_l}{\partial \gamma_p} = \frac{2\Delta y}{S}, \tag{4.18}$$

$$\frac{\partial Cd_{\text{ind}}}{\partial \gamma_p} = \frac{2\Delta y}{S} [\alpha_{\text{ind},p} + \sum_{i=0}^{p} \gamma_i \frac{\partial \alpha_{\text{ind},i}}{\partial \gamma_k}]. \tag{4.19}$$

The so-called geometrical terms required for the gradients computation $\frac{\partial J(\Gamma,\chi)}{\partial \chi}$ in Eq. (4.12) are null because the functions Eq. (4.16) Eq. (4.17) have no direct dependance to the design variables.

## 4.4 GSA and lifting line model

Aircraft fuel burn rate is proportional to the drag, while the lift is imposed by the aircraft mass. From the lifting line model, a parametric objective function $Cd_{\text{ind}}(\chi, \alpha_\infty)$, a constraint $C_l(\chi, \alpha_\infty)$ and their gradients $\nabla_\chi Cd_{\text{ind}}(\chi, \alpha_\infty)$ and $\nabla_\chi C_l(\chi, \alpha_\infty)$ are obtained. It is possible to apply the GSA approach to this particular problem, with here 40 design variables. According to the lifting line theory, the circulation distribution at the optimum of a single-point problem is elliptical. This is observed in Fig. 4.3 for the problem Eq. (4.20).

$$\min_{\chi \in \mathbb{R}^n} Cd_{\text{ind}}(\chi, \alpha_\infty) \quad \text{subject to} \quad C_l(\chi, \alpha_\infty) = C_{l0}. \tag{4.20}$$

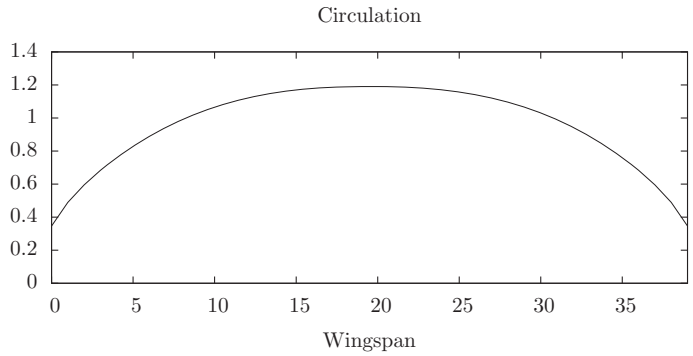



Figure 4.3: Optimal circulation for a single-point wing design at $C_l$=0.5

With a null sideslip angle, the problem is symmetric and only half the model needs to be considered. The whole model is kept in our example.

Three numerical experiments are conducted with the lifting line model. In the first one, the lift range $[0.3, 0.9]$ is uniformly discretized with 500 samples. Then, the gradients of the induced drag are computed at each of these operating conditions, and a GSA analysis is performed on the resulting gradient set. GSA detects 2 independent dimensions of the gradient span, and therefore 3 operating conditions are required to formulate the robust optimization problem. Since the design space is of dimension 40, this approach shows a gain of 38 computations compared to an aggregate objective function with a number of operating conditions under the hypothesis Eq. (3.5), which represents a cost cut of 93%. The same calculations are then performed with a 80 design variables model obtained by doubling the span-wise discretization of the wing. The gradient span dimension for the drag is also 2. In a general case, increasing the number of design variables can increase the gradient span dimension.

In the second experiment, a multipoint design problem is built with 7 operating conditions and a uniform weighting. The operating conditions are defined by lift constraints: $Cl = 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$ and the angle of attack at each condition is adjusted by the optimization algorithm to ensure this minimal lift. This optimization problem is solved with the SLSQP (Sequential Least Squares Programming) algorithm [69], the results are summarized in Fig. 4.4 for the circulation distributions and in Fig. 4.5 for the optimal twist vector $\chi$. 7 optimization iterations were required, corresponding to 49 evaluations of the lifting line model.

In the last experiment, GSA is used to analyze the gradient span for each iteration of the second experiment and the results are shown in Table 4.1. Finally, the equivalent problem given by the

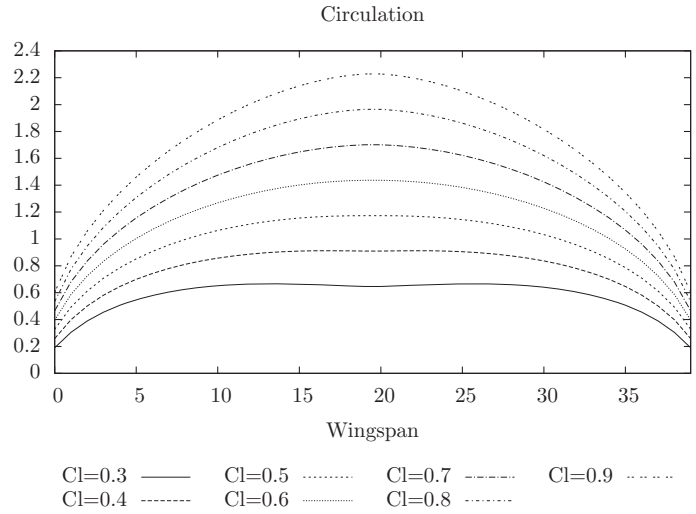



Figure 4.4: Optimal circulation for a multipoint wing design from $C_l$=0.3 to $C_l$=0.9

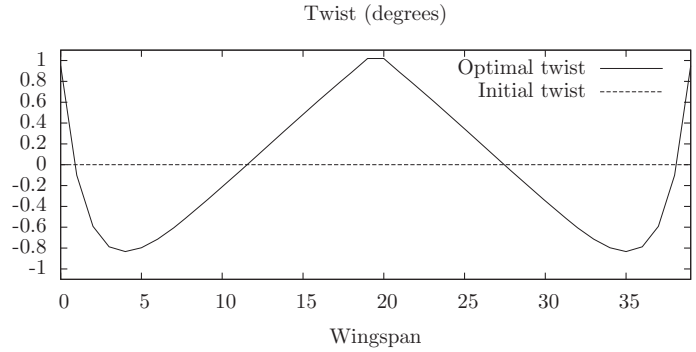



Figure 4.5: Optimal twist repartition for a multipoint wing design from $C_l$=0.3 to $C_l$=0.9

Table 4.1: Original and equivalent multi conditions lifting line optimization problem

| Iteration | $\omega_1$ | $\omega_2$ | $\omega_3$ | $\omega_4$ | $\omega_5$ | $\omega_6$ | $\omega_7$ |
|---|---|---|---|---|---|---|---|
| Baseline | 1. | 1. | 1. | 1. | 1. | 1. | 1. |
| Optimum | 0.7509 | 3.2988 | 0. | 0. | 0. | 0. | 2.9501 |

weights at convergence of the previous experiment is solved. That required 7 optimization iterations. The difference between the optimal circulations $\Gamma$ of the original situation and equivalent problem are of the same order of magnitude as the machine precision ($10^{-14}$) for the three conditions 1, 2 and 7. In other words the optimizer found the same physical solution to the two problems.

The second experiment required 49 functions and gradients evaluations, and the third one required only 21 evaluations, which represents a gain of 57%. The computational cost of the GSA algorithm and the equivalent weights are negligible compared to the model, and this is even more true for high fidelity models.

On the other hand, the equivalent weights computation must be performed at the optimum because the theorems and GSA are applied at a given $\chi$. As a consequence, this requires the solution of the initial problem. Hence, this last experiment has only a theoretical and demonstrative

interest.

However, it is important to notice that the dimension of the gradient span does not vary during the optimization. And even more interesting, the three conditions given by the GSA do not change during the optimization process, except at the first iteration, even if the equivalent weights are not constant. As a consequence, analysis of the problem needs not to be performed for each iteration of the optimization process but only after several ones, preserving the computational cost reduction. It means that for such problems, the gradient span is an invariant of the objective function during the optimization process, that is used to efficiently setup the optimization problem as described in this section.

The first step consists in performing a fine sampling of the operating condition ranges as in experiment 1. Latin Hypercube Sampling [87] or other Design analysis of computer experiment [118] methods can be employed for an efficient initial sampling. In the lifting line case, a uniform sampling is selected. The initial sampling is followed by a GSA calculation that selects the required operating conditions to be included in the optimization problem. After that, a method for adequately choosing the weights is used. As a reminder, Normal Constraint [88], Successive Pareto Optimization [91], Directed Search Domain [37], or Multiple-Gradient Descent Algorithm [27] are possible approaches. Weights usually dependent on the optimization iteration so are computed after the choice of the operating conditions. At the end of the optimization or during it, the gradient span dimension should be checked to be sure that Propositions 1 is still valid.

# Part III

# Applications to CFD-based optimization

# Table of Contents

# CHAPTER 5

## Gradient optimization for steady-state Navier-Stokes control problems

***Summary*** The aim of this chapter is to present the main components of the CFD based optimization chain, and the way they are linearized to provide the gradient of the cost function to the optimization algorithm. The parametrization, mesh update strategy, flow solver and post processing are presented. Besides, some programming contributions to the industrial optimization chain achieved during the present study are given, in particular the ones that are specific to multipoint optimization issues.

***Résumé*** L'objectif de ce chapitre est de présenter les composants principaux de la chaine d'optimisation basée sur la CFD et la façon dont ils sont linéarisés pour fournir le gradient de la fonction coût à l'algorithme d'optimisation. Les méthodes de paramétrisation, de mise à jour du maillage, de post-traitement ainsi que le code CFD sont expliquées. De plus, certaines contributions de programmation logicielle à la chaine d'optimisation industrielle réalisées au cours du présent travail sont détaillées, en particulier celles spécifiques à l'optimisation multi-point.

## 5.1 Principle

In 1956 Boltyanskiy et al. [136] gave a general approach for the study of optimal processes. They established necessary conditions known as "Pontriagin's minimum principle" for control laws to move the state of a system from a starting point to a desired state in a minimum amount of time.

In 1968, Lions has shown that systems governed by Partial Differential Equations (PDE) can be controlled by their boundary conditions [77, 78]. The latter work opened the opportunity of automated shape design of complex systems, by automation of iterative shape modifications and performance calculations. Hicks et al. [112, 48] achieved a demonstration on airfoil numerical optimization using Euler equations. In 1988, A. Jameson used an adjoint formulation and greatly lowered the cost of the gradients calculation [55]. There have been numerous demonstrations in the aerodynamics field since then, with contributions towards incorporation of industrial constraints and requirements such as complex geometries, multiple operating conditions, multi physics, unsteady flows, algorithms and complex computer architectures.

The gap between an academic demonstrator and an industrial tool that is able to address multiple cases in their formulations is not only about robustness in the software implementation, but also in the appropriate split of the problem into elementary ones. Each of these sub-problems should then be addressed in a generic way, which requires a clear formalization of inputs, outputs and processes, enabling then automation of the overall optimization process. The seven main problematics are the "optimization seven pillars" described by Shahpar from Rolls Royce [121]:

- Shape parametrization: Complex parametric geometries have to be generated, to perform shape updates when the design variables are modified by the optimization algorithm. Industrial problems often deals with geometrical constraints due to other disciplines such as systems or structures. The formulation of these constraints can depend on the design variables so the parametrization engine has to be able to formulate parametric geometrical constraints. The output of the optimization process is an optimal shape to be included in a wider aircraft design process. The mathematical standard for exchanging geometry is today Non-Uniform Rational B-Splines (NURBS) and all Computed Aided Design (CAD) software are able to import NURBS. NURBS are then a good candidate to base a parametrization tool on.

- Mesh update: The computational domain mesh has to be modified when the shape is updated. The mesh quality must not be degraded when updated. This operation must also keep consistent the surface geometry description given by the parametrization and the discretized surface of the mesh, otherwise the wrong shape performance is calculated and the optimization is biased. Usually, mesh deformation is used as it is possible to make it a derivable process for any geometry, as opposed to automated re-meshing.

- Flow solver: the flow has to be computed on updated meshes at each optimization step. The quality of the simulation has to be at the state of the art for the phenomena that is aimed to be controlled by the optimization process. If the aim is to control shocks on a wing for example, then the mesh has to be fine enough to allow the numerical scheme to resolve them precisely. The boundary layers have also to be discretized adequately to enable the viscous pressure drag calculation, flow separation prediction, and because they have an impact on the shocks position.

- Post processing: objective functions and constraint of the optimization have to be calculable from the flow solution. Various design problems are treated by design teams, which leads to a large variety of problem formulations and objectives functions. The post processing must

then be very flexible. In the context of adjoint-based optimization, it is differentiated to provide any source term for the adjoint equation.

- Optimization algorithms: driving the shape modifications strategy, optimization algorithms are at the heart of the process and have a major impact on the results. Also, a link between optimization algorithm, that are developed by mathematicians, and the numerical simulation has to be addressed. Formulation of design problems into optimization problems is closely linked to algorithms and the functional properties. The ability of available algorithms to handle general constraints, the flexibility of the optimization framework in terms of optimization formulations are key points.

- Workflow management: previous capabilities are provided by five different codes, on different machines within a network, consequently a workflow management tool is required to sequence these operations and manage the data. Besides, computing codes can be used in different contexts, and chained in different orders, so their interfaces with the workflow manager must be reusable and context independent. High fidelity simulations are used, with typically hundreds of inputs, and developed by other teams with frequent update to provide new simulation capabilities. Data is shared by multiple codes and is used in different contexts depending on the addressed problem. In data-oriented workflow managers, the programmer drives the execution through the data path, which leads to high maintenance for high fidelity simulation. On the other hand, in workflow-oriented managers, programmers only describes general sharing rules for the data, and the workflow manager deduces the data exchanges sequence from the execution sequence.

- Information Technology is an important issue. Since all services must be automated, no human intervention is possible. Programmed solutions must be treated in a very general and robust way otherwise the optimization software is very hard to maintain, too much dedicated to simplified problems and cannot be used in an industrial environment. The question of robustness is particularly important for multipoint optimization because much more calculations are executed than during single point optimizations, so all the bricks have to be more reliable to give a chance to the multipoint optimization to end successfully. IT issues remain one of the most time-consuming and least recognized part of numerical optimization.

All these points are linked together because the geometry deformations impact the mesh deformation and the flow simulations, the flow and adjoint codes are linked with the post processing for instance. To obtain satisfactory results, the whole strategy has to be consistent. It means that the architects must have an overall view of the problem, from the mathematical aspects, to the simulation and programming ones.

A simplified view of the overall process is given in Fig. 5.1.

During this work, programming contributions to all the bricks of the optimization chain were achieved. The most important ones were on the parametrization software PADGE (Parametric and Differentiated Geometrical Engine), the workflow manager WORMS (WORkflow Management System) and the optimization framework OpenDACE (Design and Analysis of Computer Experiements).

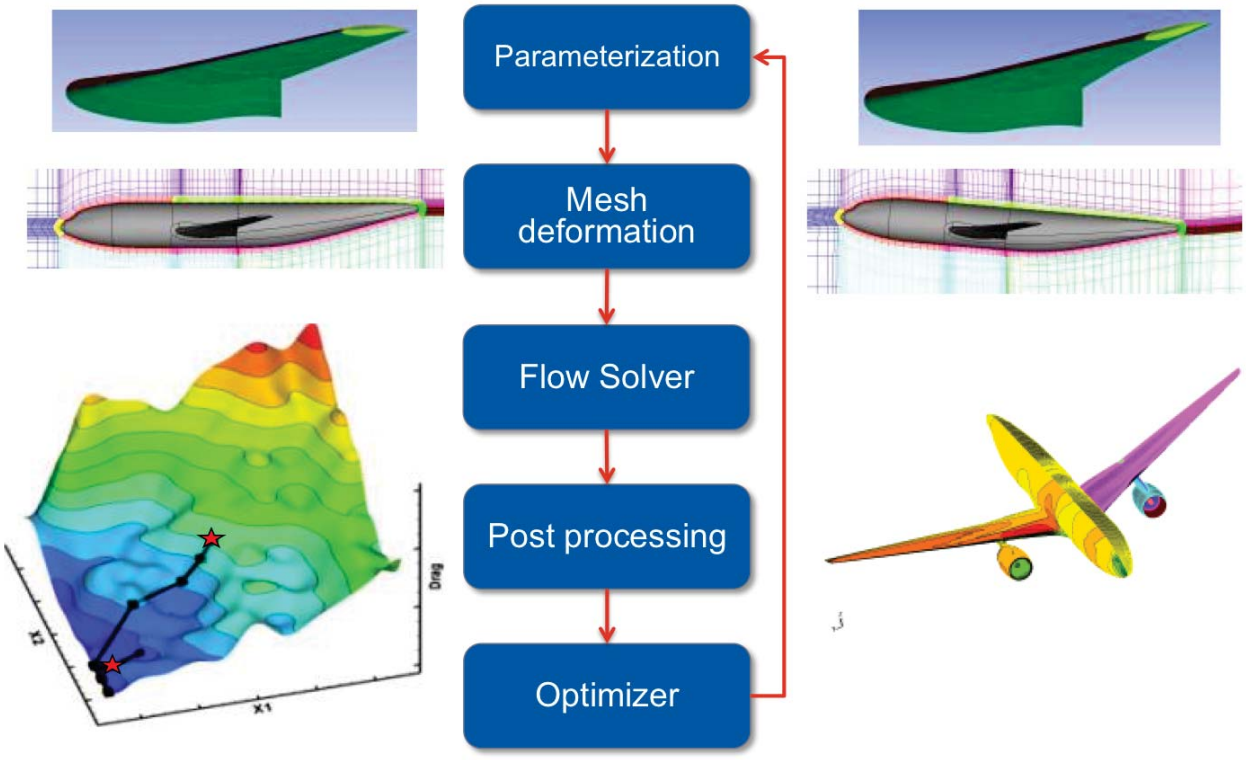



Figure 5.1: General principle scheme of an aerodynamic optimization chain

## 5.2 Parametrization

### 5.2.1 Description of the tool

In a review article [119], Samareh lists the possible approaches for parametrization of shapes dedicated to high fidelity multidisciplinary optimization: direct nodes manipulation, or using a basis vector, an elliptic partial differential equation, splines and polynoms, free-form deformation that also updates the mesh, and CAD (Computed Aided Design) manipulation. Since in the end, the manufacturing standards are based on CAD, an industrial optimization process must provide an optimized CAD as output. The most straightforward way of doing it is to have a parametric CAD in the optimization loop. Samareh gives two main challenges for CAD-based parametrization: the computation of analytical derivatives for gradient optimization, in an efficient way, and the use of design variables that make sense from a designer point of view. PADGE (Parametric and Differentiated Geometrical Engine) is the CAD engine that is used to produce the surface shapes in the present study. The above requirements drive the main features of PADGE:

- A fully parametric kernel: every geometrical entity and transformation is parametric. Each input variable can depend on other parametric elements and is updated dynamically.
- NURBS-based: Geometrical elements, 1D, 2D and 3D are based on NURBS (Non-Uniform Rational B-Splines) that are a standard for representing geometry in most CAD kernels. For a reference book on NURBS, see "the NURBS book" by Piegl et al.[106].
- Full differentiation: the derivative with respect to design variables of any shape point linked with the CFD mesh is analytically calculated in an efficient way. By comparison, Nemec et

al. [95] achieved a 3D aerodynamic optimization based on the CAPRI CAD software. Since no analytical derivatives were available, finite differences dealt computational cost issues.

- Shape smoothness: $C^n$ continuity [39], for any integer $n$. At least $C^2$ geometry description is required for aerodynamic optimization, as the Navier-Stokes equations involve second order space derivatives. For instance, it is known that parametrization strategies that directly handle the shape points coordinates require a smoothing to maintain the shape quality [64], since adjoint gradients are usually not even $G0$. Such a smoothing requires integration with the optimization algorithm. Here the shape quality is maintained by construction of the parametric shape.

A RAE 2822 profile parametrization based on the Computer Aided Design (CAD) formalism of Non-Uniform Rational B-Splines (NURBS) is displayed in Fig. 5.2. A complex 3D shape such as the XRF-1 wing can be parametrized, see Fig. 5.3.

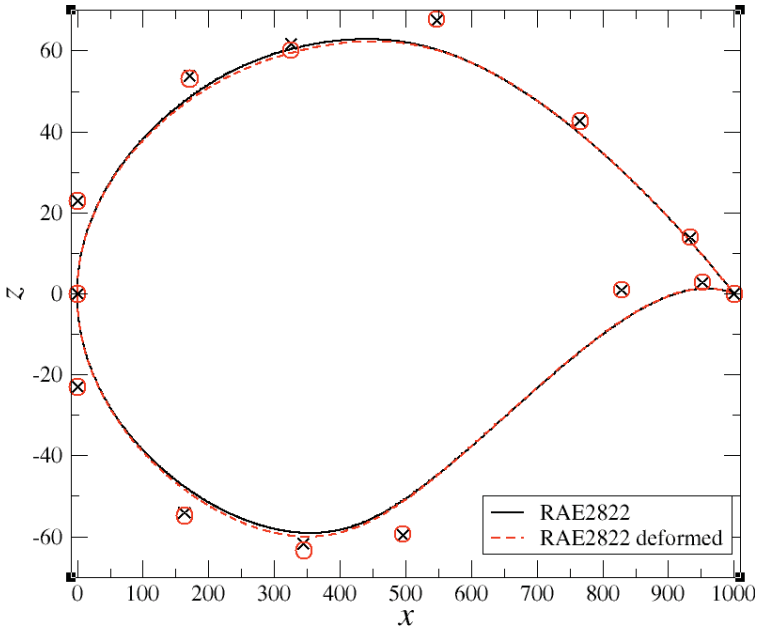



Figure 5.2: A NURBS parametrization of the RAE 2822 profile, with 16 NURBS poles.

The mathematical expression of a NURBS curve is:

$$C(u) = \frac{\sum_{i=1}^{npu} N_{i,d}(u) w_i P_i}{\sum_{i=1}^{k} N_{i,d}(u) w_i}, \tag{5.1}$$

where $d$ is the degree of the curve, $npu$ is the number of poles $P_i$, $u$ is the curvilinear abscissa, $N_{i,d}(u)$ are polynomial B-Splines basis functions. The poles are in the geometrical space, $R^3$ for instance. For surfaces, the expression is:

$$S(u,v) = \frac{\sum_{i=1}^{npu} \sum_{j=1}^{npv} N_{i,du}(u) N_{j,dv}(v) w_{i,j} P_{i,j}}{\sum_{i=1}^{npu} \sum_{j=1}^{npv} N_{i,du}(u) N_{j,dv}(v) w_{i,j}}, \tag{5.2}$$

Parametrization in the NURBS formalism consists in the creation of relations to displace the poles of NURBS in the geometrical space, which means making the NURBS poles depend on design variables $P_i : \chi \rightarrow P_i(\chi)$, so that curves and surface become parametric functions. The motion of

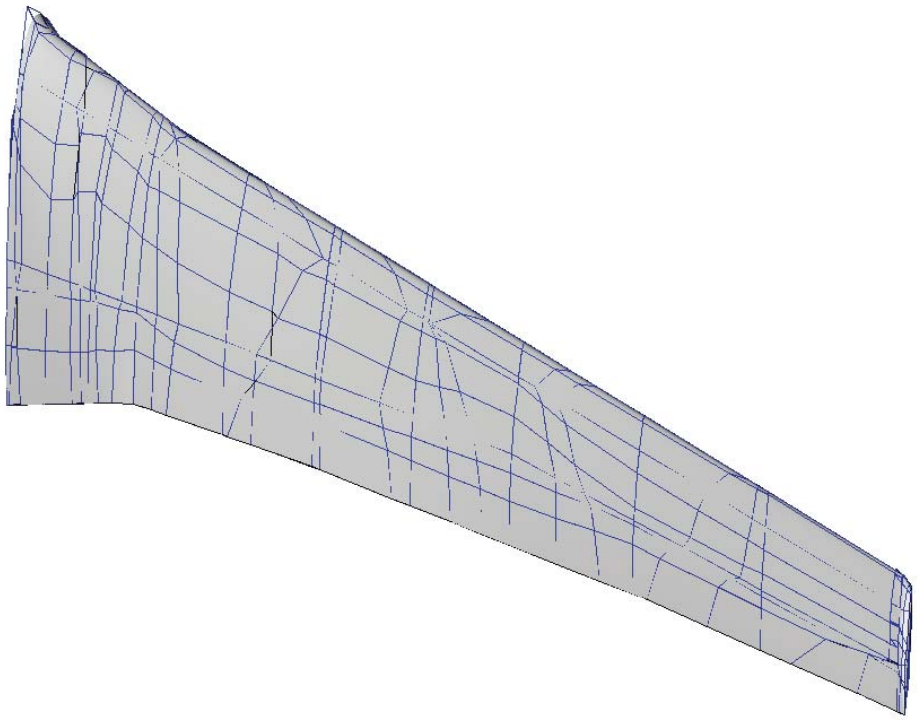


Figure 5.3: A NURBS parametrization of the XRF-1 configuration, with NURBS control polygon.

poles through a design variable change, locally modifies the shape, as displayed in Fig. 5.3. This can also be seen through eqs. (5.1) and (5.2).

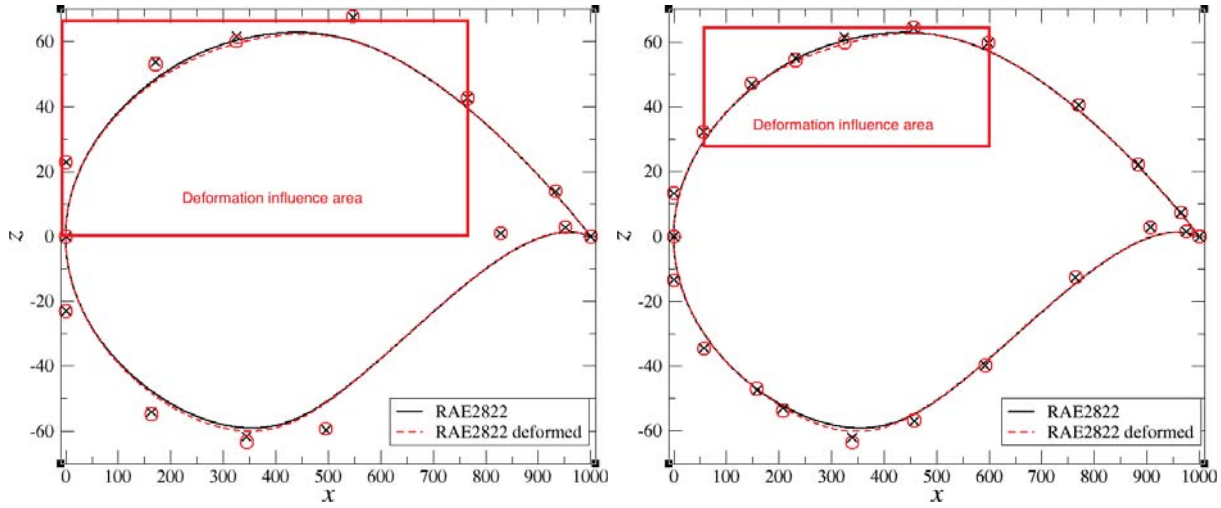
An important programming contribution to this tool has been produced during the present thesis, in the kernel, parametric airfoil, wing and wingtip templates. For instance, the author contributed to the development of a smart update system of parametric elements based on automatic detection of dependancies, including coordinates systems, to perform all geometry calculations in the correct order, with up to date information. Since object-oriented programming separates the description of the objects from the way the instantiated geometrical entities are calculated, there is no direct link between instantiation order and calculations order. For instance, a user can instantiate a wing, create a "Sweep" design variable, and control the sweep angle of the wing with the "Sweep" variable. The execution and instantiation sequence is "Create Wing, Create Sweep design variable, Control Wing sweep with Sweep design variable" while the correct update sequence is "Update Sweep variable, Update Wing sweep". In addition, the gradients propagation by the chain rule also depends on the update sequence. These aspects were interesting challenges.

### 5.2.2 The question of dimensionality in geometry parametrization

In the Chapter 2, it has been shown that the number of design variables has a high impact on the choice of the optimization algorithm, when the objective function is costly to evaluate. This argument justifies the use of gradient-based algorithms and the adjoint approach. The number of parameters used to parametrize a 3D shape is driven by technological and geometrical reasons detailed in this section.

In practice, designers use geometrical design variables to describe shapes or deformations in a

geometrical space since this allows to incorporate knowledge of the problem into the optimization problem. For a fixed NURBS degree, the higher the number of poles is, the more local the deformations are. This is illustrated in Fig. 5.4a and Fig. 5.4b, where the same geometry is described with two different NURBS curves. Since B-Splines basis functions have are non-null on a segment only, they localize the influence of the poles positions on the shape, see [106] for more details about NURBS. Therefore, the higher the number of poles is, the finer is the control of shape modifications.




(a) 16 NURBS poles parametrization

(b) 24 NURBS poles parametrization

Figure 5.4: Comparison of two NURBS parametrizations of the RAE 2822 profile.

Placing poles of NURBS in a geometrical space is a sampling problem so is subject to Curse of Dimensionality (see Chapter 2 section 2.2.1). Typically, 5 to 20 design variables are used for 2D airfoil optimizations, which corresponds to 5 to 15 poles for the upper profile side, and the same amount for the profile wing side. For instance Nemec et al. [97] used 25 control points and 19 design variables. In our example of Fig. 5.2, the vertical position of 7 poles are driven by the design variables, the abscissa remaining frozen. Consequently, placing poles for an airfoil parametrization can be seen as sampling a 1D space. The curse of dimensionality formula, giving the number of required samples $n_s$, in a $n$-dimensional space and for a precision $10^{-p}$ is $n_s(n,p) \approx 10^{np} n^{\frac{n}{2}}$, see Chapter 2, Eq. (2.8).

Since $n_s(1, 0.84) \approx 7$, the precision of a 7 points sampling in a 1-dimensional space is $10^{-0.84}$. The $n_s(n,p)$ function can be used to calculate the sampling cost when the dimension of the space varies, by fixing $p$ and varying $n$. Then, to maintain a similar geometry control in a 2-dimensional space, one should use $n_s(2, 0.84) \approx 96$ sampling points. A possible interpretation is that if one wants to create a 3D wing parametrization from the 2D parametrization of Fig. 5.2, he should use approximately 96 parametrized poles. In Fig. 5.3, the XRF-1 wing is parametrized with about 100 design variables, which illustrates the variation of the number of design variables with respect to the dimension of the geometrical space.

## 5.3 Mesh deformation

A mesh deformation approach is used to update the mesh when the CAD model is updated by new design parameters. Subtraction of the baseline shape CAD to the updated one provides a surface deformation field that has to be propagated into the volume mesh. Here, the method relies on an algebraic integral formulation by Meaux et al. [81] given in Eq. (5.3), based on Shepard's

method of inverse distances for data interpolation [122]. Deformation field in the volume mesh $\delta X(M)$ is explicitly extrapolated from the surface deformation field $\delta X^*(P)$ prescribed by the parametrization. A visibility factor $exp\left(\frac{2}{3}\left(1-\left(\frac{\overrightarrow{n_p}.\overrightarrow{PM}}{||\overrightarrow{PM}||}\right)\right)\right)$, where $\overrightarrow{n_p}$ is the normal to the surface at the point $P$, enhances the formulation to avoid upper wing deformations to influence lower wing volume mesh points.

$$\delta X(M) = \frac{\sum_{P\in\Omega} \delta X^*(P)\left[||\overrightarrow{PM}||exp\left(\frac{2}{3}\left(1-\left(\frac{\overrightarrow{n_p}.\overrightarrow{PM}}{||\overrightarrow{PM}||}\right)\right)\right)\right]^{-a}}{\sum_{P\in\Omega}\left[||\overrightarrow{PM}||exp\left(\frac{2}{3}\left(1-\left(\frac{\overrightarrow{n_p}.\overrightarrow{PM}}{||\overrightarrow{PM}||}\right)\right)\right)\right]^{-a}}, \tag{5.3}$$

where $\Omega$ is the set of surface mesh points with prescribed deformations. The algorithmic complexity of such a calculation "is number of surface points" times "number of volume points", and becomes expensive for meshes with dozens of millions nodes. The integral method is hybridized with transfinite interpolation for the inner nodes of the mesh blocks.

Given adjoint volume mesh derivatives $\frac{dJ(\chi,\alpha)}{dX}$ analytical differentiation is used to provide a reverse mode for the computation of the surface mesh sensitivity $\frac{dJ(\chi,\alpha)}{dS}$ through Eq. (5.4),

$$\frac{dJ(\chi,\alpha)}{dS} = \frac{dJ(\chi,\alpha)}{dX}\frac{dX}{dS}. \tag{5.4}$$

We note the attenuation function $Att(P,M) = \left[||\overrightarrow{PM}||exp\left(\frac{2}{3}\left(1-\left(\frac{\overrightarrow{n_p}.\overrightarrow{PM}}{||\overrightarrow{PM}||}\right)\right)\right)\right]^{-a}$, and with the formulation of Eq. (5.3), we obtain:

$$\frac{dJ(\chi,\alpha)}{dS}(P) = \sum_{M\in V}\frac{dJ(\chi,\alpha)}{dX}(M)\frac{dAtt(P,M)}{dP}\frac{1}{\sum_{Q\in\Omega} Att(M,Q)}, \tag{5.5}$$

where $V$ is the set of volume mesh points.

In this way, the impact of the mesh deformation on the flow solution is derived and taken into account in the final gradient provided to the optimizer, at a similar cost as the direct mesh deformation. We obtain a surface sensitivity field $\frac{dJ(\chi,\alpha)}{dS}$ displayed in Fig. 5.5, that highlights the areas of the shapes which deformation has a great impact on the function of interest.

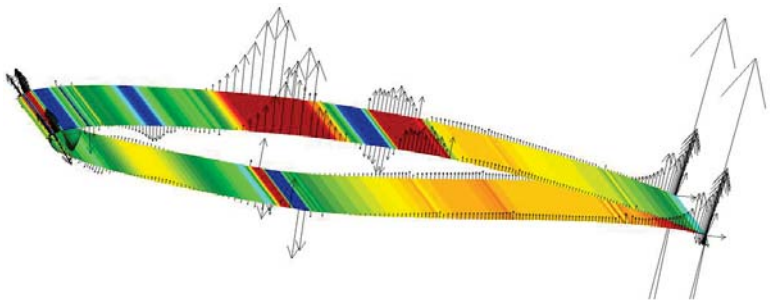


Figure 5.5: Pressure drag surface sensitivity vectors $\frac{dCdp(S,\alpha)}{dS}$ on the RAE2822 airfoil at Mach 0.72, Cl 0.7 Re 6.5M

## 5.4 Flow solver

### 5.4.1 Flow state equations

The flow solver computes the flow around a shape, given as input through an updated computational volume mesh. To this aim, the Reynolds-Averaged Navier-Stokes equations, modeling the flow physics, are solved. The Navier-Stokes equations are a set of three partial-derivatives equations that describe the viscous newtonian fluids motion in the continuum hypothesis. Naviers-Stokes equations describe all the scales of turbulence. We wish not to resolve these scales due to the very high computational cost [21]. Consequently, the Reynolds-Averaged approach was developed, aiming at solving the averaged state of the flow. The Reynolds-average is an average on a set of experiments, not to be confused with a time-average. For compressible flows, the Favre average extends the Reynolds average with the aim of keeping the mass conservation equation free from any source term due to the averaging process. Reynolds Averaged Navier-Stokes equations (RANS) have proven to provide sufficient precision for the aerodynamic design [74] and is therefore well suited for aerodynamic shape optimization. A steady state differential version of these equations is presented here.

$$div(\overline{\rho}\overline{\mathbf{U}}) = 0 \tag{5.6}$$

$$div(\overline{\rho}\overline{\mathbf{U}} \otimes \overline{\mathbf{U}} + \overline{p}\mathbf{I} - \overline{\tau} - \tau_{\mathbf{r}}) = 0 \tag{5.7}$$

$$div\left[\overline{\rho}\overline{E}\ \overline{\mathbf{U}} + \overline{p}\overline{\mathbf{U}} - (\overline{\tau} + \tau_{\mathbf{r}})\overline{\mathbf{U}} + C_p(\overline{\phi} + \phi_{\mathbf{t}})\right] = 0 \tag{5.8}$$

$\overline{\rho}$: Favre averaged fluid mass fraction

$\overline{\mathbf{U}}$: Favre averaged eulerian flow particle speed.

$p$: Fluid static pressure

$\mathbf{I}$: identity matrix

$\overline{\tau}$: Favre averaged viscous constraint tensor

$\tau_{\mathbf{r}} = -\overline{\rho('\mathbf{U}) \otimes ('\mathbf{U})}$ : Reynolds stress tensor. This term appears when the non-linear convective term $div(\rho\mathbf{U}\otimes\mathbf{U})$ is averaged. To close the equations, this term requires a dedicated transport model.

$\overline{E}$ : Total energy per mass unit.

$\overline{\phi}$: Constraint tensor due to heat flux.

$\overline{\phi_{\mathbf{t}}}$: Constraint tensor due to turbulent heat flux.

Equation (5.6) is called continuity equation and translates the fluid mass conservation. Equation (5.7) represents the conservation of momentum and Eq. (5.8) represents the energy conservation.

The ideal gas law is added:

$$p = \rho RT \tag{5.9}$$

Two comportmental laws are supposed. The deformation tensor is supposed to be proportional to the constraints gradient for Newtonian fluids.

$$\overline{\tau} = -\frac{2}{3}\mu(div\overline{\mathbf{U}})\mathbf{I} + 2\mu\mathbf{D}, \tag{5.10}$$

with $\mu$ the dynamic molecular viscosity of the fluid and where the deformations tensor $\mathbf{D}$ is given by:

$$\mathbf{D} = \mathbf{grad}(\overline{\mathbf{U}}) + \mathbf{grad}^t(\overline{\mathbf{U}}). \tag{5.11}$$

The Fourier's law for thermal heat flux is given in Eq.(5.12):

$$\overline{\phi} = -\lambda_l \mathbf{grad}(\overline{T}). \tag{5.12}$$

where the molecular thermal conductivity coefficient is given by $\lambda_l = C_p\mu/Pr$. $C_p$ and $Pr$ are the heat capacity at constant pressure and the Prandtl number.

To close the RANS equations, the turbulent Reynolds stress tensor needs to be modeled. There are two options, either writing transport equations for the six components of the tensor, as in the Reynolds Stress Model approach, or making an analogy with laminar viscosity as in Eq. (5.10) and Eq. (5.11), also called the Boussinesq hypothesis:

$$\tau_r = -\frac{2}{3}\mu_t(div\overline{\mathbf{U}})\mathbf{I} + 2\mu_t\mathbf{D} \tag{5.13}$$

In the latter case, the modeled turbulence effects are isotropic, because it is modeled by a single scalar $\mu_t$, the turbulent viscosity. A transport equation for the turbulent viscosity or a set of equations from which the turbulent viscosity is defined, called "turbulence model", must then be solved. Details are not given here on turbulence modeling, but can be found in [139]. The turbulent heat flux also needs to be modeled. We take $\phi_\mathbf{t} = -\lambda_t \mathbf{grad}(\overline{T})$, with the turbulent thermal conductivity: $\lambda_t = C_p\mu_t/Pr_t$.

The finite volume method allows to discretize the continuous equations. The aim is to put it under the form $R(W, X\alpha) = 0$, where a discrete representation of the space given by the mesh $X$. Discrete flow state variables at each cell $W = [\overline{\rho}, \overline{\rho}\overline{\mathbf{U}}, \overline{\rho}\overline{E}, \mu_t]$ are the unknown. In that objective, the equations are integrated over each control volume given by all the mesh cells, the Green-Ostrogradsky theorem leads to a flux balance at each of the interfaces. Finite volumes is well-suited for the Navier-Stokes equations because of the numerous divergence operators of Equations. (5.6), Eq. (5.7) and Eq. (5.8) that are removed when integrated on the cells volumes. Approximations result in the calculation of fluxes at the interface, knowing averaged flow state variables at the center of the cells for instance, by the numerical scheme. Then, either a linear system is resolved or an explicit iteration scheme is used to obtain an updated $W$, or an hybrid of the two. The procedure is iterative, because RANS equations are non-linear. Norm of the residuals $R$ are used to monitor the convergence of the solution, and quantify how precisely the state equations are respected.

### 5.4.2 Numerical aspects

Numerical computations were performed with the *elsA* Onera software [15, 111, 16]. This code manages both the flow analysis and the flow sensitivity aspects. It solves the 3D compressible

Reynolds-Averaged Navier-Stokes equations using a cell-centered finite-volume method on structured grids as well as the associated discrete adjoint equations. The turbulence model chosen is the one-equation Spalart-Allmaras [124] model.

The spatial convective fluxes of the mean flow are discretized with the upwind Roe scheme [116] with the Harten's entropic correction. A MUSCL scheme (Monotone Upstream-centered Schemes for Conservation Laws) [134] associated with a Van Albada limiter [133] provides a second-order accurate scheme. The spatial convective fluxes of the turbulent flow are discretized with the first-order upwind Roe scheme. Spatial diffusive fluxes are approximated with a second-order central scheme. The turbulent equations are solved separately from the mean flow equations at each time step with the same time-marching method. The backward Euler implicit scheme drives the time integration. The resulting linear systems are solved with the scalar Lower-Upper Symmetric Successive Over-Relaxation (LU-SSOR) method [142]. A standard nonlinear multigrid algorithm [53] combined with local time stepping accelerates the convergence to steady-state solutions.

## 5.5 Flow adjoint solver

All of the methods previously described in the flow solver are differentiated by hand. The turbulent eddy viscosity and thermal conductivity are assumed constant [54, 99] during the differentiation.

The resolution method to solve the discrete adjoint equation is the preconditioned first-degree iterative method, similar to an approximate Newton method, of Eq. (5.14).

$$\widetilde{A}(\lambda^{(l+1)} - \lambda^{(l)}) = -A\lambda^{(l)} - b, \tag{5.14}$$

where the matrix $\tilde{A}$ is an approximation of the jacobian matrix $A = \frac{\partial R}{\partial W}$.

The approximation matrix $\tilde{A}$ derives from the linearization of a first-order Steger-Warming flux-vector splitting scheme for the convective flux and the linearization of the diffusive flux neglecting the spatial cross derivatives. The resulting linear systems are solved with a few steps of a block LU-SSOR algorithm at each iteration of the iterative method.

## 5.6 Post-processing

Post processing is performed with an in-house code called Zapp, that computes objective functions and constraints. Zapp's philosophy is to provide a generic post-processing of quantities through their expression depending on flow state variables $W$ and geometrical integration areas. Resulting integrals are computed, and the differentiation for creating the flow discrete adjoint source terms $\frac{\partial j(W,X)}{\partial W}$, and the geometrical term $\frac{\partial j(W,X)}{\partial X}$, see Eq. (5.30) are also generic. In optimization applications, many different objectives and constraints are encountered (center of lift, target load span repartition, combinations of lift, drag and moment), it is then important to provide a generic capability for adjoint source terms computation and gradient assembly.

For instance, the pressure drag is the objective function used in the next chapters, it is computed with Eq. (5.15).

$$Cdp(X,W) = \iint_S \frac{p(W) - p_\infty}{\frac{1}{2}\rho {V_\infty}^2} \overrightarrow{n(X)}.\vec{x} ds(X), \tag{5.15}$$

, where $\overrightarrow{n(X)}$ is the normal to the surface, $\vec{x}$ is the coordinate system's first axis, and $ds(X)$ is the elementary surface integration area, typically the facet of a mesh cell. Since the pressure $p$ is as a

function of flow state variables $W = (\rho, \rho U_x, \rho U_y, \rho U_z, \rho E)$:

$$p = (\gamma - 1)\frac{\rho E - 0.5((\rho U_x)^2 + (\rho U_y)^2 + (\rho U_z)^2)}{\rho}. \tag{5.16}$$

The source term for the adjoint equation is computed by combining Eq.(5.17) and Eq.(5.17):

$$\frac{\partial Cdp(X,W)}{\partial W} = \iint_S \frac{\partial p(W)}{\partial W} \frac{1}{\frac{1}{2}\rho {V_\infty}^2} \overrightarrow{n(X)}.\vec{x}ds(X). \tag{5.17}$$

$$\frac{\partial p(W)}{\partial W} = (\gamma - 1) \begin{bmatrix} -\frac{\rho E - 0.5((\rho U_x)^2 + (\rho U_y)^2 + (\rho U_z)^2)}{\rho^2} \\ \frac{\rho U_x}{\rho} \\ \frac{\rho U_y}{\rho} \\ \frac{\rho U_z}{\rho} \\ \frac{\rho E}{\rho} \end{bmatrix}. \tag{5.18}$$

The geometrical term of the objective gradient $\frac{\partial Cdp(X,W)}{\partial X}$ requires the derivation $\frac{\partial \overrightarrow{n(X)}.\vec{x}ds(X)}{\partial X}$.

## 5.7 Workflow manager

Running multiple high fidelity computer codes using large data on a computer grid is a challenge from a software point of view. Workflow management tools formalize and facilitate tasks execution sequencing, data exchanges between codes in memory, on the disk and through the network. They also check the data provided by users and the coherency of workflow before execution, strongly decreasing the probability of failure at execution. Chaining geometry generation, mesh deformation, CFD softwares and post processing involves a high number of inputs and outputs, making the debugging of the classical scripting approach very time consuming. Optimization requires to perform these tasks multiples times, and multipoint optimization multiplies again this number by a factor of typically 3 to 10, making the use of adequate workflow manager more critical.

The advantage of such tool is that they address the problems of parametric analysis, optimization, gradient validation by finite differences in a general way from the software point of view : running multiples times in parallel a code with varying inputs. These bricks are modular: one can check gradients by finite differences (FD), check the gradients of a multipoint chain by FD, perform a multipoint optimization using gradient method on with a solver that does not provide gradients using the FD module, etc., saving a lot of programming effort compared to the scripting approach.

Data management raises multiple issues. There are two main paradigms in workflow management: data oriented and workflow oriented paradigms. In data oriented softwares, programmers typically describes execution tasks (processes), their inputs, outputs, and links the inputs either to a user interface or to the outputs of another process. The execution sequence is driven by the data: when a process has up-to-date inputs, it is executed and return outputs. In the workflow oriented paradigm, programmers describe processes inputs and outputs, the execution sequence, and rules about data such as sharing policy with other processes. The order of magnitude of the number of inputs of high fidelity simulation softwares is a hundred, making the explicit programming of data exchanges time consuming, so the workflow-oriented approach is advantageous. It also enables the modularity of the processes such as finite differences, multipoint, optimization, which require different input data depending on the sub-processes. For instance, performing a multipoint on the

CFD software typically requires from the user to specify Mach, Reynolds, Angle of attack ranges or sets of values, while if one achieves a multipoint analysis on an analytic function, the inputs will be different. This means that the input data of the multipoint chain has to be dynamic: it depends on the workflow, here the sub-process. Data dynamism in workflow management is rarely addressed and a specific issue in simulation [3]. A methodology for comparing workflow management tools was proposed by Stoilova et al. [127].

In the present thesis, generic finite differences and multipoint chains for parametric analysis and gradient validation have been developed. Besides, improvements to the WORMS (WORkflow Management System) for managing data dynamism, symmetric multiprocessing, and data management through the network have been programmed.

## 5.8 Connecting optimizers to the simulation environment

Optimization algorithms are developed by mathematicians, and benchmarked against analytical functions libraries such as CUTEr [42], while CFD codes are developed by physicists and software engineers. Therefore, turning the CFD simulation environment into a function "drag, lift = CFD($\chi$)" as required by the algorithm point of view is challenging. Workflow management tool are a step towards it, since it automates the CFD process. Many libraries of optimization algorithms already exists and should be used, such as scipy [60], DAKOTA [34], nlopt [59], DOT [82] and snopt[40]. However these libraries do not necessarily provide standardized algorithm interfaces, and never provide the link to the simulation.

The OpenDACE software, used in the present thesis, makes the link between algorithms libraries and the workflow manager. Similar initiatives such as pyOpt [104] and OpenMDAO [45], that also provides workflow management, have been developed. Programming contributions to the tool were achieved during the present study, such as functions management, link with the simulation, the integration of optimization algorithms and design of experiments strategies.

## 5.9 Gradient computation by discrete adjoint method

Understanding the discrete adjoint state for RANS equations is disconcerting. There are multiples interpretations and definitions of discrete adjoint. Two of them are presented here. First, the algebraic point of view is useful to mechanically derive the equations and compute the gradient of functions of interest. Then the Lagrangian point of view is presented, that is very useful for reasoning on the optimization formulations and optimizer behavior. Both of them lead to the same equations in the end.

Dedicated Computational Fluid Dynamics solvers compute the flow solution $W$ by solving the fluid state equations $R(\mathcal{W}, X(\chi), \alpha) = 0$. Any functions of interest $j$ required for the optimization is computed from the flow $W$ and the computational mesh $X$. The discrete adjoint method enables to compute the derivatives of $j$ with respect to design variables $\chi$ at moderate cost. It relies on the fact that the state equation imposes a necessary condition on the derivatives of $W$ and $X$ at convergence and therefore avoids to compute explicitly the costly derivatives of the flow solution with respect to the design variables $\frac{dW(\chi)}{d\chi}$.

### 5.9.1 The algebraic point of view

We aim at computing the derivatives of any function of interest with respect to design variables ($\frac{dj(\chi)}{d\chi}$). We assume that the design variables parametrize the shape only, which implies that only the mesh $X$ has a direct dependance to them, through the mesh deformation process. The flow state $W$ is assumed to respect the state equation Eq.(5.19).

$$R(\mathcal{W}, X(\chi), \alpha) = 0. \tag{5.19}$$

With Eq.(5.19), the implicit functions theorem allows to write:

$$\mathcal{W} = W(\chi, \alpha). \tag{5.20}$$

And also:

$$R(W(\chi, \alpha), X(\chi), \alpha) = 0. \tag{5.21}$$

Then, the functional $j(\chi, \alpha)$ can also be written as:

$$j(\chi, \alpha) = \mathcal{J}(W(\chi, \alpha), X(\chi), \alpha), \tag{5.22}$$

and derived with respect to the design variables:

$$\frac{dj(\chi, \alpha)}{d\chi} = \frac{\partial \mathcal{J}(W(\chi, \alpha), X(\chi), \alpha)}{\partial W} \frac{dW(\chi)}{d\chi} + \frac{\partial \mathcal{J}(W(\chi, \alpha), X(\chi), \alpha)}{\partial X} \frac{dX(\chi)}{d\chi}, \tag{5.23}$$

In Eq.(5.23), the partial derivatives with respect to the state $W$ and the mesh $X$ represent the derivation of the post-processing routines, typically the integration of pressure forces on the shape. The term $\frac{dW(\chi,\alpha)}{d\chi}$ represents the effect of the shape changes on the flow solution. They are computed with a linearized version of the flow solver. Therefore, we derive Eq.(5.21), and suppose that the state equation is always resolved when the design variables change, so is always equal to zero:

$$\frac{dR(W(\chi, \alpha), X(\chi), \chi, \alpha)}{d\chi} = \frac{\partial R}{\partial W} \frac{dW}{d\chi} + \frac{\partial R}{\partial X} \frac{dX}{d\chi} = 0. \tag{5.24}$$

Equation (5.24) provides a necessary condition on $\frac{dW(\chi,\alpha)}{d\chi}$ where the dependency on $\chi$ only appears in the linearization of the mesh deformation process $\frac{dX}{d\chi}$, that can be computed efficiently in adjoint mode and for a cost that is quasi independent of the number of design variables. We can then write:

$$\frac{dW}{d\chi} = -\left[\frac{\partial R}{\partial W}\right]^{-1} \frac{\partial R}{\partial X} \frac{dX}{d\chi}, \tag{5.25}$$

to replace $\frac{dW(\chi,\alpha)}{d\chi}$ in Eq.(5.23).

$$\frac{dj(\chi, \alpha)}{d\chi} = \left[\frac{\partial \mathcal{J}(W(\chi, \alpha), X(\chi), \alpha)}{\partial X} - \frac{\partial \mathcal{J}(W(\chi, \alpha), X(\chi), \alpha)}{\partial W} \left[\frac{\partial R}{\partial W}\right]^{-1} \frac{\partial R}{\partial X}\right] \frac{dX(\chi)}{d\chi}, \tag{5.26}$$

Equation (5.26) is the key to understand the mechanisms that links shape modifications to functionals variations and the linearization approaches. A fully differentiated CFD evaluation of for a given design vector provides much more information than the functional and its gradient on the design problem, the numerical model and the derivation strategy. A good understanding of the meaning of each term is therefore important to design and operate adjoint-based shape optimization workflows.

- $\frac{dX(\chi)}{d\chi}$ Represents the impact of design variables modification on the volume mesh.

- $\frac{\partial \mathcal{J}(W(\chi,\alpha),X(\chi),\alpha)}{\partial X}$ Represents the linearization of the functional post-processing routines with respect to the mesh, for a given flow $W$.
- Therefore, $\frac{\partial \mathcal{J}(W(\chi,\alpha),X(\chi),\alpha)}{\partial X}\frac{dX(\chi)}{d\chi}$ is a pure geometrical term, and represents, for a frozen flow solution, the impact of a shape modification on the geometrical part of the functional post processing integration.
- $\frac{\partial R}{\partial X}$ represents, for a frozen flow, the sensitivity of the residuals calculation to the volume mesh.
- Therefore, $\frac{\partial R}{\partial X}\frac{dX(\chi)}{d\chi}$ represents, for a frozen flow, the impact of the volume mesh deformation due to a design variable change, on the residual calculation.
- $\left[\frac{\partial R}{\partial W}\right]^{-1}$ represents the sensitivity of the flow solution $W$ to any perturbation of the residuals, for a frozen mesh.
- And then, $\left[\frac{\partial R}{\partial W}\right]^{-1}\frac{\partial R}{\partial X}\frac{dX(\chi)}{d\chi}$ represents the derivative of the flow solution with respect to the design variables, due to combined effects of the volume mesh deformation on the residuals calculation and then on the flow solution.
- $\frac{\partial \mathcal{J}(W(\chi,\alpha),X(\chi),\alpha)}{\partial W}$ represents the linearization of the post-processing routine of the functional with respect to the flow, for a frozen geometry.
- $\frac{\partial \mathcal{J}(W(\chi,\alpha),X(\chi),\alpha)}{\partial W}\frac{dX(\chi)}{d\chi}$ represents the sensitivity of the functional to design variables modifications through combined effects of mesh deformation that impacts the residuals calculations, then the flow and finally the calculation of the functional due to the flow change.
- In $\frac{dj(\chi,\alpha)}{d\chi}$, it is now clear how all the sensitivities are combined to obtain the total derivatives of the functional with respect to design variables, taking into account effects of design variables modifications on the mesh that then impacts the residuals calculation, so the flow solution that respects the state equation, and finally the post-processing of the functional.

In Eq. (5.26), the high dimensional terms are $\left[\frac{\partial R}{\partial W}\right]^{-1}$ and $\frac{\partial R}{\partial X}$. Two main options exists to calculate $-\frac{\partial \mathcal{J}(W(\chi,\alpha),X(\chi),\alpha)}{\partial W}\left[\frac{\partial R}{\partial W}\right]^{-1}\frac{\partial R}{\partial X}\frac{dX(\chi)}{d\chi}$. To assemble the gradient, one can first derive the state variables with respect to the design variables. Eq. (5.25) tells us that $\frac{dW}{d\chi}$ is the solution of Eq. (5.27).

$$\left[\frac{\partial R}{\partial W}\right] x = \frac{\partial R}{\partial X}\frac{dX}{d\chi}. \tag{5.27}$$

One linear system is solved per design variable, and the solution is replaced $\left[\frac{\partial R}{\partial W}\right]^{-1}\frac{\partial R}{\partial X}\frac{dX(\chi)}{d\chi}$ in Eq (5.26). This approach called tangent linear model or direct linearization in the literature is efficient when less design variables are used than functionals. On the other hand, if less functionals are used than design variables, then the calculation should be performed in "reverse mode", so that the calculation depends on the functionals number, instead of the design variables number. $-\frac{\partial \mathcal{J}(W(\chi,\alpha),X(\chi),\alpha)}{\partial W}\left[\frac{\partial R}{\partial W}\right]^{-1}$ is a vector of the same size as the flow state variables, and the solution of the linear system of Eq. (5.28):

$$-\frac{\partial \mathcal{J}(W(\chi,\alpha),X(\chi),\alpha)}{\partial W}\left[\frac{\partial R}{\partial W}\right]^{-1} = \lambda^T, \tag{5.28}$$

often rewritten as the so-called "adjoint equation":

$$\frac{\partial \mathcal{J}(W(\chi,\alpha),X(\chi),\alpha)}{\partial W} + \lambda^T\left[\frac{\partial R}{\partial W}\right] = 0. \tag{5.29}$$

Eq.(5.29) is solved, usually by Krylov methods [117, 96]. The method was recently extended by Pinel et al. to take advantage of common information when solving multiple adjoint equations in constrained aerodynamic optimization for instance [107]. Equation (5.26) is reformulated using the adjoint vector in Eq. (5.30). To give the final formula for adjoint derivatives.

$$\frac{dj(\chi,\alpha)}{d\chi} = \left[\frac{\partial \mathcal{J}(W(\chi,\alpha),X(\chi),\alpha)}{\partial X} + \lambda^T \frac{\partial R}{\partial X}\right] \frac{dX(\chi)}{d\chi}. \tag{5.30}$$

One can also compute the total derivative of the functional with respect to the mesh with Eq. (5.31). Far from the aircraft skin, the $\frac{dj(\chi,\alpha)}{dX}$ field represents a goal-oriented mesh dependency of the solution and can then be used for mesh adaptation techniques [105].

$$\frac{dj(\chi,\alpha)}{dX} = \frac{\partial \mathcal{J}(W(\chi,\alpha),X(\chi),\alpha)}{\partial X} + \lambda^T \frac{\partial R}{\partial X}. \tag{5.31}$$

When parameterization and mesh deformation are preformed by two different approaches, contrary to free-form deformation for instance [120], the derivative $\frac{dX(\chi)}{d\chi}$ is decomposed. Volume sensitivities can be composed with surface sensitivities to compute the derivative of functions with respect to surface coordinates $S$,

$$\frac{dj(S,\alpha)}{dS} = \frac{dj(X,\alpha)}{dX}\frac{dX(S)}{dS}. \tag{5.32}$$

Finally the surface is parametrized with design variables $\chi$ and the gradient of the function is calculated,

$$\nabla_\chi j(\chi,\alpha) = \frac{dj(S,\alpha)}{dS}\frac{dS(\chi)}{d\chi}. \tag{5.33}$$

The functions and gradient calculation at multiple operating conditions by the different codes of the optimization chain are summarized in Fig. 5.6.

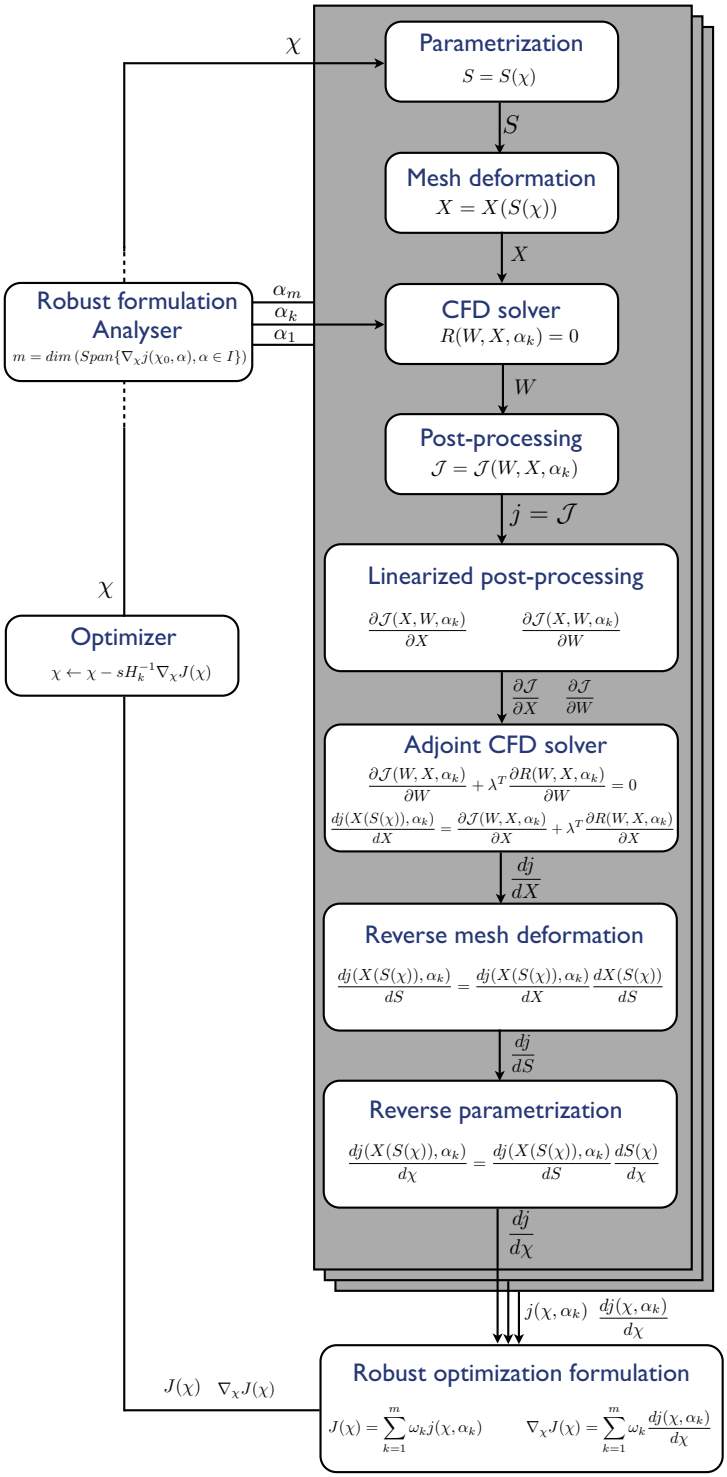



Figure 5.6: Multipoint optimization workflow with key formulas

### 5.9.2 Discrete adjoint: the Lagrangian point of view

Another point of view for the discrete adjoint interpretation is the Lagrange multiplier approach. The purpose of the Lagrangian is to turn a constrained optimization problem into a unconstrained

one that has the same solution. A Newton iteration can be used to minimize the unconstrained problem and give the solution of the original problem for instance. The main idea is to penalize the objective function by all constraints, and to find the penalty coefficient so that the two problem are equivalent. Actually no equivalence is achieved, but the Lagrange equation is a necessary condition. To illustrate the approach, we plot the contours of the objective $J(\chi)$ and the constraint $g(\chi)$ in Fig. 5.7. If the contours of $J$ and $g$ are not parallel, more precisely if their tangent planes are

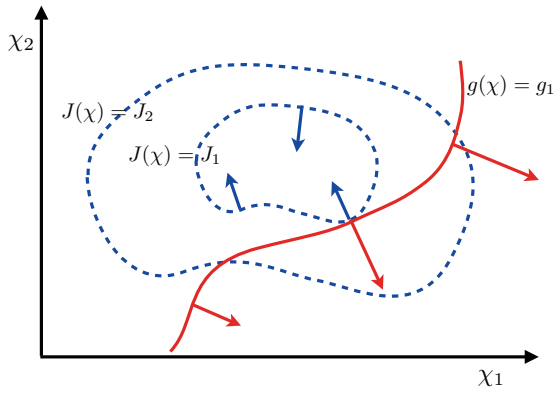



Figure 5.7: A graphical interpretation of the Lagrange multipliers

not parallel, then when following an isoline $g(\chi) = g_1$ in red, the objective $J(\chi)$ varies. On the opposite, at the optimum of the constrained problem the constraint is valid, so staying on the isoline $g(\chi) = g_1$, the objective does not vary at first order, otherwise it would be possible to decrease it while keeping constant the constraint. Therefore the contours of $J$ and $g$ are parallel at the optimum. As the gradients are normal to the isolines, the gradients of the objective and the constraints are necessary collinear at the optimum. There should exist a scalar $\lambda$ such that $\nabla_\chi J(\chi) + \lambda \nabla_\chi g(\chi) = 0$. The latter expression can be viewed as the gradient of the augmented function $\mathcal{L}(\chi, \lambda) = J(\chi) + \lambda g(\chi)$, and the zero of its gradient can be viewed as an optimality condition. The Lagrangian and the derived optimality conditions are presented here. We suppose that the residuals are computed on a $n_c$ cells mesh, and that there are 7 scalar state equations.

$$
\begin{gathered}
\chi^* = \text{argmin} \;\; J(\chi), \\
\text{subject to:} \\
g(\chi) = 0, \\
\Rightarrow \\
\exists \lambda \in \mathbb{R}^{7n_c}, \\
\mathcal{L}(\chi, \lambda) = J(\chi) + \lambda^T g(\chi) \\
\nabla_\chi \mathcal{L}(\chi^*, \lambda) = 0 \quad (\Leftrightarrow \nabla_\chi J(\chi^*) + \lambda^T \nabla_\chi g(\chi^*) = 0) \\
\nabla_\lambda \mathcal{L}(\chi^*, \lambda) = 0 \;\; (\Leftrightarrow g(\chi^*) = 0)
\end{gathered}
$$

The Karush Kuhn Tucker (KKT) conditions [70, 62] are a generalization of the Lagrangian to general equality and inequality constraints. We can apply now the Lagrangian approach to the CFD-based optimization problem, with the satisfaction of the flow state equation as equality con-

straint:

$$\begin{gathered}\chi^*, W^* = \text{argmin} \;\; \mathcal{J}(X(\chi), W), \\ \text{subject to:} \\ \mathcal{R}(X(\chi), W) = 0, \\ \Rightarrow \\ \exists \lambda \in \mathbb{R}^{7n_c}, \\ \mathcal{L}(X(\chi), W, \lambda) = \mathcal{J}(\chi) + \lambda^T \mathcal{R}(X(\chi), W) \\ \nabla_W \mathcal{J}(\chi^*, W^*) + \lambda^T \nabla_W \mathcal{R}(X(\chi^*), W^*) = 0 \\ \nabla_\chi \mathcal{J}(\chi^*, W^*) + \lambda^T \nabla_\chi \mathcal{R}(X(\chi^*), W^*) = 0 \\ \nabla_\lambda \mathcal{L}(X(\chi), W, \lambda) = 0\end{gathered}$$

We wish not to let the optimizer find the solution of the state equation, because efficient CFD solvers find $W^*$ such that $\mathcal{R}(X(\chi), W^*) = 0$ for any design vector $\chi$. Therefore the problem is splitted in two steps: finding $W$ for each $\chi$ provided by the optimizer. $W$ becomes an implicit function of $\chi$ as in the previous part. This is why the equation $\nabla_W \mathcal{J}(\chi^*, W^*) + \lambda^T \nabla_W \mathcal{R}(X(\chi^*), W^*) = 0$ is written in the classical form $\frac{\partial \mathcal{J}(\chi^*, W^*)}{\partial W} + \lambda^T \frac{\partial \mathcal{R}(X(\chi^*), W^*)}{\partial W} = 0$, and is ensured at each optimization step. Equation $\nabla_\chi \mathcal{J}(\chi^*, W^*) + \lambda^T \nabla_\chi \mathcal{R}(X(\chi^*), W^*) = 0$ represents the zero gradient condition, so is ensured only at convergence of the optimization algorithm. We have now the same equations as in the previous section reached through the algebraic approach.

#### 5.9.3 Discrete adjoint: the Envelope Theorem

The envelope Theorem provides a general interpretation of Lagrange multipliers. Given the optimization problem:

$$\begin{gathered}\chi^* = \text{argmin} \;\; J(\chi), \\ \text{subject to: } g(\chi) = \bar{g}, \\ \Rightarrow \\ \exists \lambda \in \mathbb{R}, \nabla_\chi J(\chi^*) + \lambda \nabla_\chi g(\chi^*) = 0,\end{gathered}$$

given the optimal value function $V(\bar{g}) = J(\chi*)$ such that $g(\chi^*) = \bar{g}$, the envelope Theorem, see [20] page 428 and [130] page 484, allows to compute the rate of change of the objective function when the value of the constraint $\bar{g}$ changes:

$$\frac{dV(\bar{g})}{d\bar{g}} = \lambda. \tag{5.34}$$

In our RANS-based aerodynamic optimization case, see Eq. (5.34), the constraint translates conservation laws $\mathcal{R}(X(\chi), W) = 0$ has to be rewritten $\mathcal{R}(X(\chi), W, \alpha) = \bar{R}$, evaluated for $\bar{R} = 0$, and the $V$ function is the value of the drag $V : \bar{R} \rightarrow J(X(\chi^*), W, \alpha)$, at the optimum $\chi^* = argmin_\chi(J(X(\chi^*), W, \alpha)$, and under constraint $\mathcal{R}(X(\chi^*), W, \alpha) = 0$ . Then Eq. (5.34) becomes:

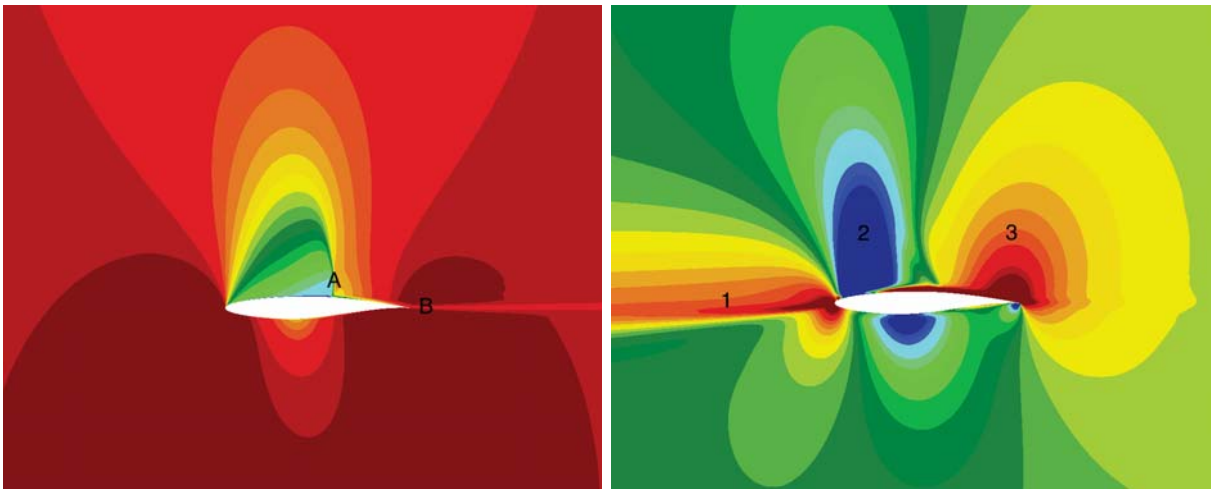
$$\frac{dV(\bar{R})}{d\bar{R}} = \lambda. \tag{5.35}$$

Consequently, the adjoint field in CFD is the rate of change in terms of objective, drag for instance, when the residuals of the RANS equations change. If the residual is perturbed and equal to $R_0$,

one can expect, at first order, a variation of the objective function of $\frac{dV}{dR}^T.(R_0 - 0) = \lambda^T.(R_0 - 0)$. Such property can be used to estimate goal-oriented convergence error in CFD [5], or the impact of any perturbation of the residuals on a specific function at a moderate cost. Besides, with the help of Eq. (5.34), one can directly state that a residual rate of change $\frac{\partial R(X(\chi))}{\partial \chi}$ due to the shape modification by design variables, changes changes the objective function at a rate $\lambda^T \frac{\partial R}{\partial \chi}$. If the objective function has also a direct dependance to the geometry, such as integration areas for aerodynamic function, adding the term $\frac{\partial \mathcal{J}(W,X(\chi),\alpha)}{\partial \chi}$ provides the full objective function gradient of Eq. (5.36):

$$\frac{dj(\chi,\alpha)}{d\chi} = \frac{\partial \mathcal{J}(W(\chi,\alpha),X(\chi),\alpha)}{\partial \chi} + \lambda^T \frac{\partial R}{\partial \chi}. \tag{5.36}$$

Figure 5.8 shows the density state variable and the first component of the adjoint to drag (dual




(a) $\rho$ (b) $\lambda_\rho$

Figure 5.8: First components of flow state and adjoint to pressure drag around the RAE2822 at Mach 0.74 Cl 0.782

state of the density state variable) around the RAE2822 airfoil. One can note that the main features of the adjoint state are not situated at the same areas as the main features of the flow such as the shock (Fig. 5.8a area A) and the wake (Fig. 5.8a area B). The rate of change in the drag due to the flow density (i.e. first component of adjoint) is mostly affected by the generating conditions of the shock: on the right (Fig. 5.8b area 3) and the left (Fig. 5.8b area 2) to the shock, where the adjoint has the high positive and negative values. In addition, the state of the fluid arriving to the leading edge of the airfoil is strongly affecting the flow around the airfoil, leading to high adjoint to drag values (Fig. 5.8b area 1).

# CHAPTER 6

## RAE2822 airfoil optimizations

***Summary*** In this chapter, single point and multipoint optimizations of the RAE2822 airfoil in transonic and viscous flow conditions are presented. Since these single point optimizations converge to shock-free airfoils, they validate the optimization chain. While the airfoils resulting of single point-optimizations present strong degradations of the performance in off-design conditions, it is shown that adequately formulated multi-Mach and multi-lift optimizations present much more interesting performance compromises. In addition, the GSA method is assessed against a multipoint optimization on a finely sampled operating condition space. The limit of the constant gradient span dimension hypothesis on which GSA relies for effective optimization is also tested, and a solution is proposed.

***Résumé*** Dans ce chapitre, des optimisations mono-point et multi-points du profil RAE2822 sont menées en conditions transsoniques et pour un écoulement visqueux. Les optimisations mono-points, convergeant vers une solution sans choc, valident la chaine d'optimisation. Ces profils issus d'optimisations mono-points montrent une forte dégradation de la performance en des conditions de vol pour lesquel ils n'ont pas été optimisés. A l'inverse, les profils issus d'optimisation multi-points montrent de bien meilleurs compromis de performance. De plus, la méthode GSA est validée par comparaison à une optimisation multi-point finement résolue dans l'espace des conditions de vol. La limite de l'hypothèse de constance de la dimension de l'espace vectoriel engendré par les gradients de la fonction objective au cours de l'optimisation, utilisée en pratique dans la méthode GSA, est montrée et une solution y remédiant est proposée.

## 6.1 Presentation of the test case

The RAE2822 transonic airfoil is a common use case for aerodynamic optimization. It has been used to illustrate a wide range of methodologies such as meta models [57, 73], Radial Basis Functions for mesh deformation and parametrization [90], practical design constraints handling [14], as well as multipoint optimization [30, 146]. Besides, 3D cases such as wing-body optimization presented in the next chapter are difficult to reproduce, the numerous optimization ingredients difficult to master, and the expected optimal configuration is not easy to guess, making harder the assessment of methods. On the other hand, it is known that single-point airfoil pressure drag minimization should lead to shock-free solutions, more generally, the numerous RAE2822 optimizations available in the literature represent an important knowledge about this problem. Besides, airfoil optimization reveals real-life applications properties such as the single-point optimization effect [30], making the RAE2822 profile a well-suited intermediate test case between analytic models and aircraft optimization.

### 6.1.1 Parametrization

A CAD-based parametrization of the RAE2822 is build using PADGE, see Chapter 5 section 2. 16 design variables representing curvatures, tangent angles, and airfoil camber are used. The thicknesses at 30% and 60% of chord are fixed. Figure 6.1 shows the CAD generated by PADGE, with the control polygon in blue, representing the poles of the $4^{th}$ degree NURBS. Figure 6.2 shows two examples of deformations of the shape, a camber increase and a leading edge radius increase. The shape quality is maintained in this case up to $G^2$ continuity, meaning that on the parametric CAD surface defined by $x, y, z = f(u, v)$, we obtain an implicit function $z = g(x, y)$ in the geometrical space that is twice differentiable. PADGE handles arbitrary orders of continuity on demand.

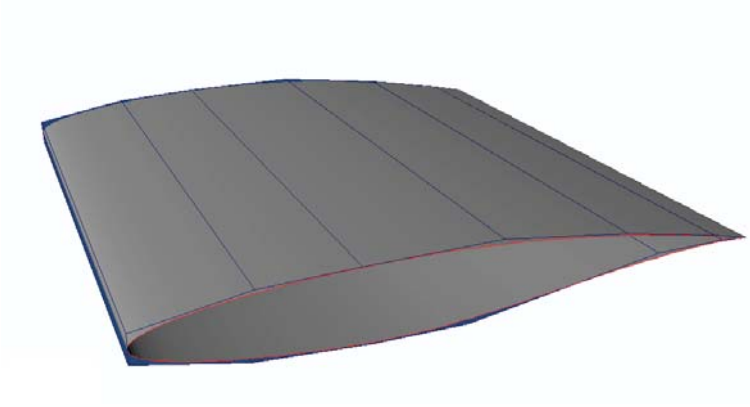
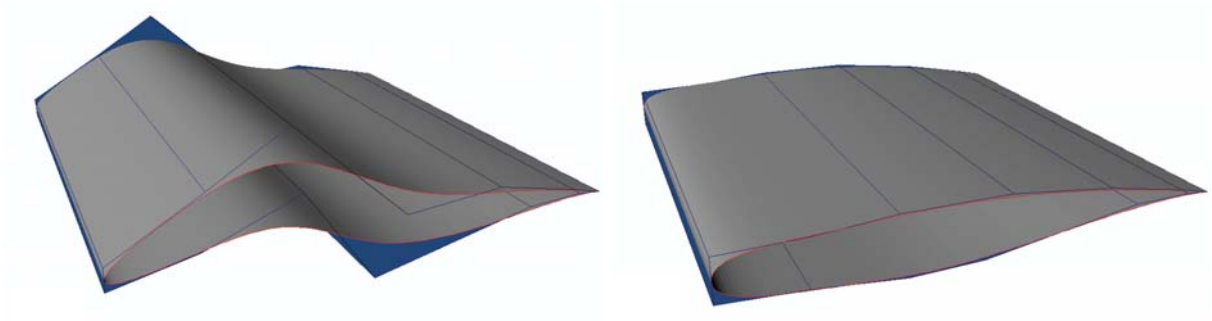


Figure 6.1: CAD of the RAE2822 and its control polygon

### 6.1.2 Flow and adjoint calculations

A fine mesh of 300 000 cells and 1.5 millions degrees of freedom is used and displayed in Fig. 6.3, the domain extends over 80 chord lengths. The height of the first layer of cells is such that $y^+ = 1$. Numerical methods presented in the previous chapter are used within the elsA code, the Roe scheme is employed with the Spalart-Allmaras turbulence model. Each calculation is performed on 8 CPUs,



(a) 10% chord camber increase at 30% of chord (b) Leading edge radius large increase

Figure 6.2: Extreme deformations in the parametrization of the RAE2822 airfoil

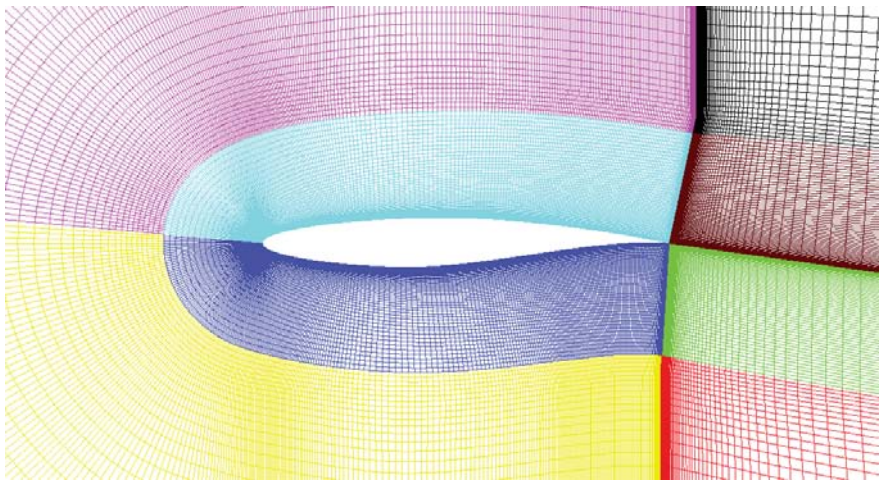
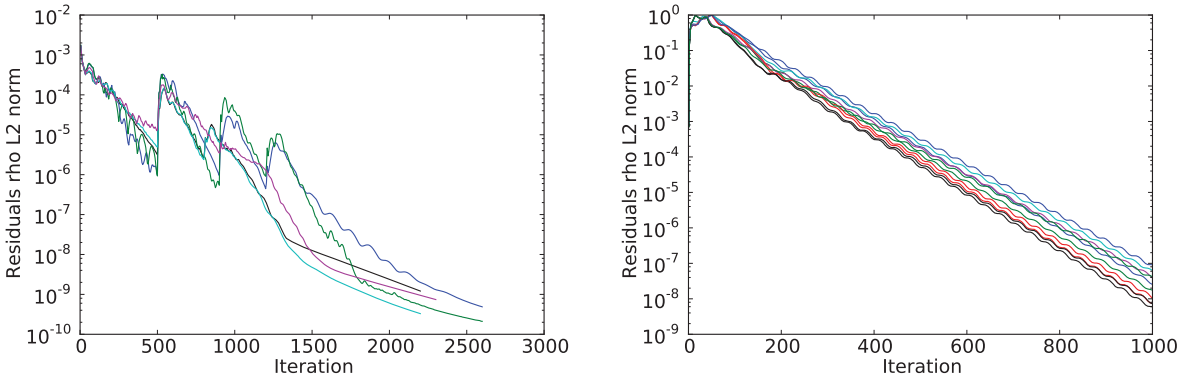


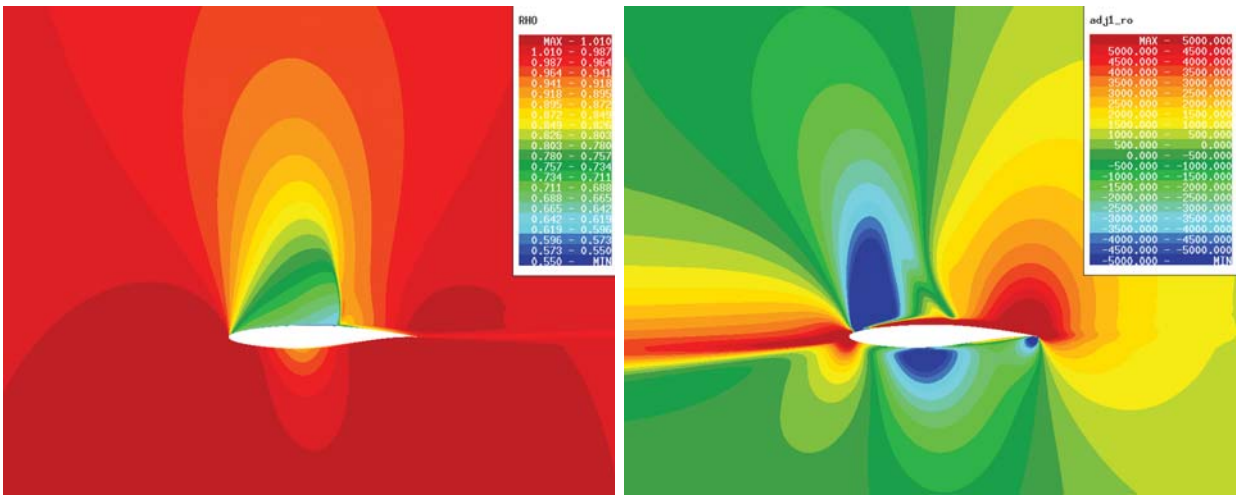
Figure 6.3: Mesh of the RAE2822

with no load imbalance. The flow and adjoint calculations convergences at five operating conditions: Re 6.5M, Cl 0.782, Mach 0.5, 0.62, 0.68, 0.72, 0.74 on the RAE2822 are displayed in Fig. 6.4. The residuals for both direct and adjoint calculations are reduced by 7 orders of magnitude. In the following optimizations, only 500 adjoint solver iterations are used to save computational time, since no substantial ($< 0.1$ drag count) difference in the optimization results have been observed. The restitution time of a full shape performance evaluation with pressure lift and drag gradients calculations is 19 minutes. The obtained flow and adjoint states are displayed in Fig. 6.5, Fig. 6.6, Fig. 6.7 and Fig. 6.8.



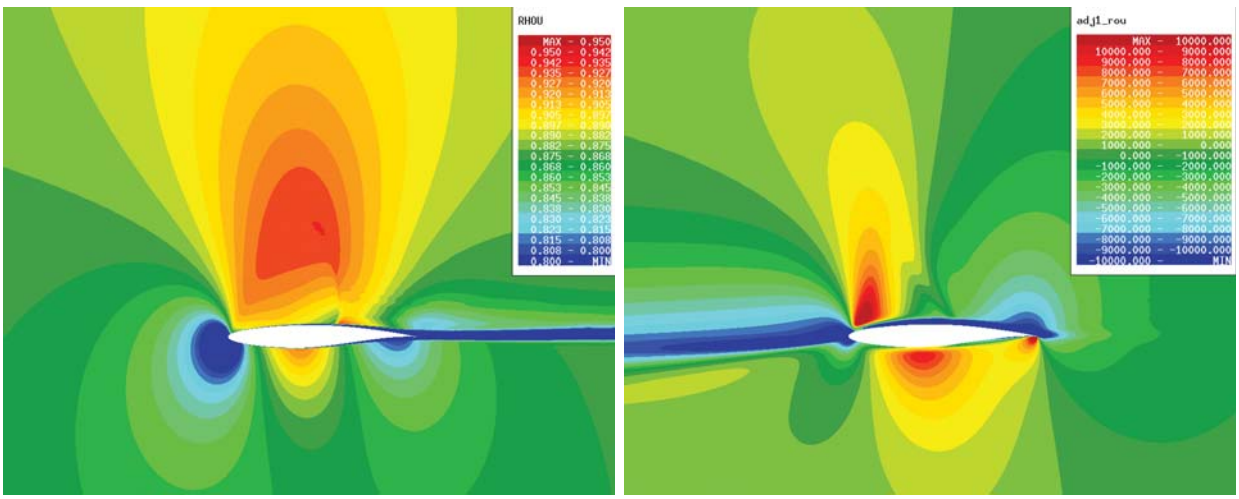

(a) Direct calculation convergence (b) Adjoint calculation convergence

Figure 6.4: Direct and adjoint calculations convergence history on the RAE2822 baseline at the 5 operating conditions



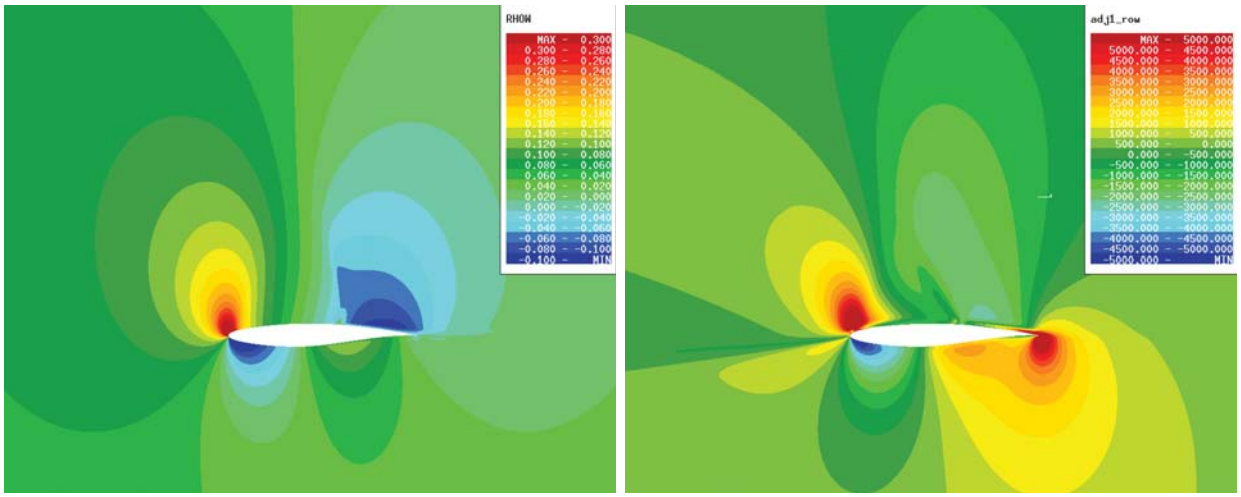

(a) $\rho$ (b) $\lambda_\rho$

Figure 6.5: Flow state and adjoint around the RAE2822 at Mach 0.74 Cl 0.782



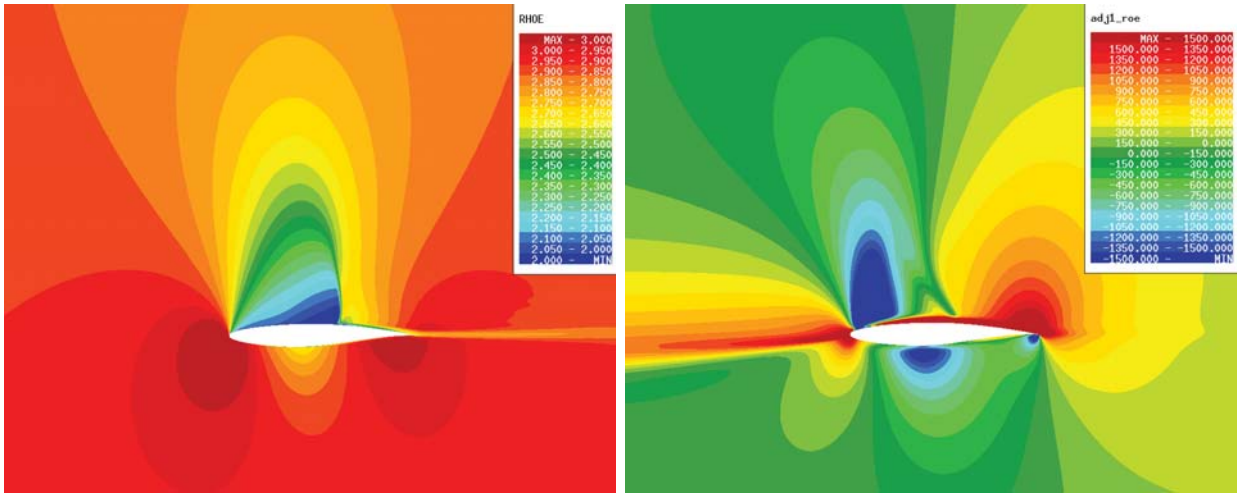

(a) $\rho U$ (b) $\lambda_{\rho U}$

Figure 6.6: Flow state and adjoint around the RAE2822 at Mach 0.74 Cl 0.782




(a) $\rho W$ (b) $\lambda_{\rho W}$

Figure 6.7: Flow state and adjoint around the RAE2822 at Mach 0.74 Cl 0.782




(a) $\rho E$ (b) $\lambda_{\rho E}$

Figure 6.8: Flow state and adjoint around the RAE2822 at Mach 0.74 Cl 0.782

## 6.2 Single-point optimization

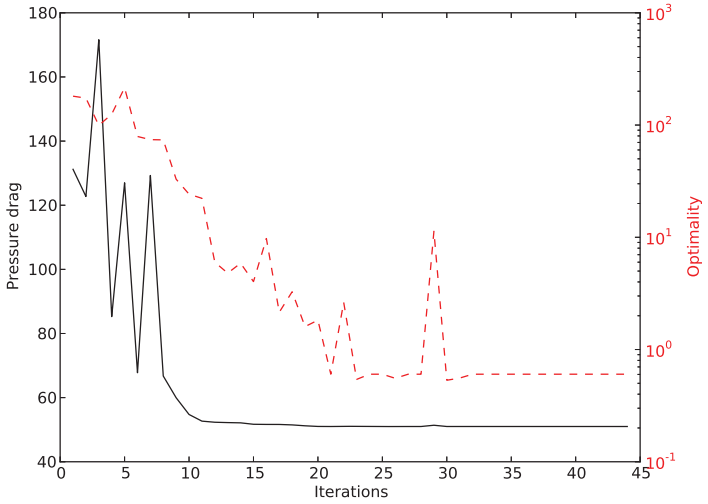
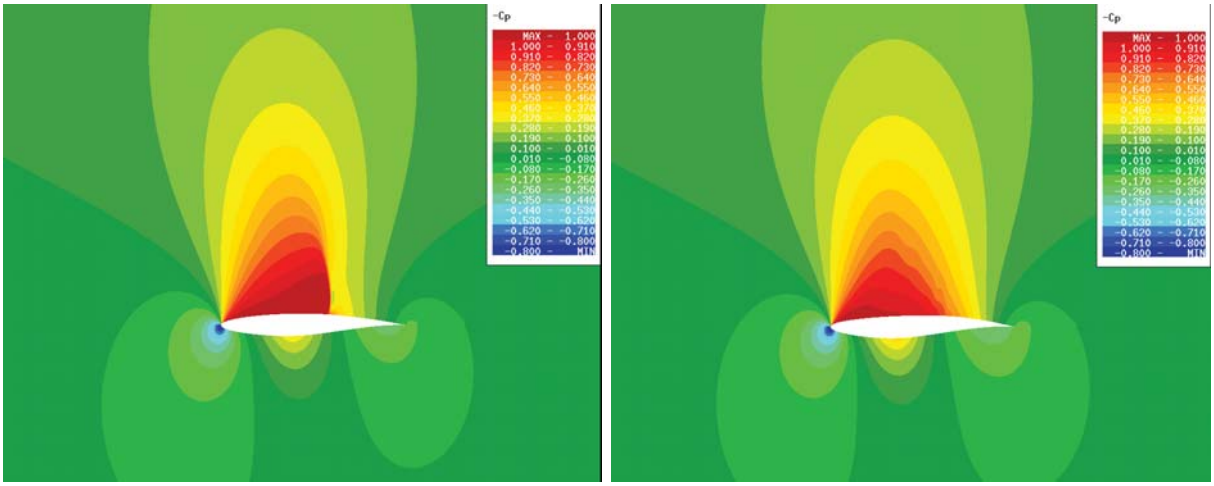



Figure 6.9: Optimality and objective function convergence on the single point RAE 2822 optimization at Mach 0.74, Cl 0.782, Re 6.5 M

Figure 6.9 shows an optimization convergence at Mach 0.74, Reynolds 6.5M and Cl 0.782. A target lift approach calculation finds the adequate angle of attack to achieve the lift constraint while the flow is converging. The optimality is controlled here by the norm of the Lagrangian, see Chapter 5. The constraints taken into account in this Lagrangian are the lift constraint and bounds on the design variables. It is interesting to note that the optimizer achieves 2 orders of magnitude convergence in terms of optimality and does not converge further with more iterations. This is due to the noise on the gradient, since the turbulence model is not linearized in the elsA code, and that a thin shear layer approximation is used in the linearization. Numerous finite differences validation of the gradient have shown that this hypothesis leads to an average noise of 1%, and up to 10% errors on a couple of variables. The L-BFGS-B optimizer [145] (Low memory Broyden Fletcher Goldfarb Shanno - Bounds constrained) drives the process, see 2.3.1 for more details. Quasi Newton algorithms using line-search are tolerant to noise on the gradient up to a certain level, as demonstrated by Gratton et al.[44]. The present optimization case justifies the frozen turbulence assumption since the associated noise on the gradient did not prevent the l-bfgs-b algorithm to converge and provide important gains. Similarly, Dwight et al. [33] shown on a RAE2822 optimization test case and by comparison with a fully differentiated code that both the frozen turbulence and the thin shear layer approximation can have negligible impacts on the optimization results. The Congugate Gradient algorithm was less sensitive to noise than the Quasi-Newton algorithm, but provided significantly less drag gains using both exact and approximate gradients.

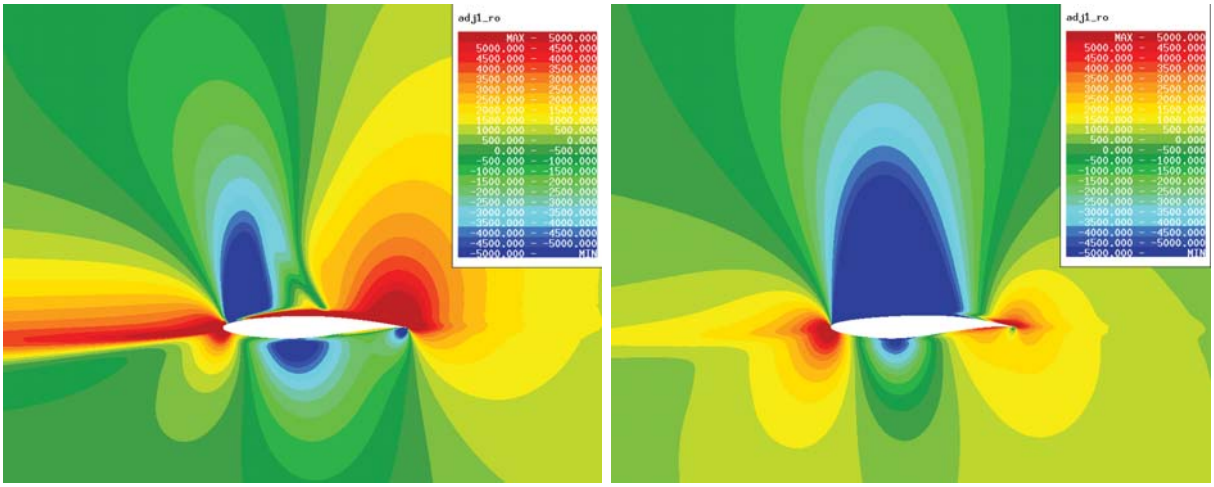
Figure 6.10 shows the pressure coefficient around the RAE2822 baseline and single point optimum at Mach 0.74, Cl 0.782 and Reynolds 6.5M. The strong shock visible on Fig.6.10a has been completely removed in Fig. 6.10b. Some pressure oscillations are visible. It was shown that shock free airfoils can be obtained by optimization of the RAE2822, however not at Mach 0.74 and Cl 0.782, only Cl 0.7 was reached in the study by Harbeck et al. [47]. Figure 6.11a shows the density component of the adjoint to pressure drag, around the RAE2822 baseline. Figure 6.11b displays the same quantity around the optimized profile. Keeping in mind that the adjoint field represent



(a) Baseline (b) Single-point optimum

Figure 6.10: Pressure coefficient around baseline RAE2822 and optimized at Mach 0.74 Cl 0.782

the sensitivity of the drag to the flow residuals value (see Chapter 5, envelope Theorem), one can notice that the influence of the flow arriving on the airfoil on the pressure drag has been reduced. In addition, the flow around the optimum being shock-free, the features of the adjoint state on the suction side are also greatly modified. Interpretation of adjoint fields is not easy, and there is room for improvement on that topic. Being able to analyze such an information should provide insights on the flow control problem, so has a potential interest from a practical point of view.



(a) Baseline (b) Single-point optimum

Figure 6.11: Adjoint to drag $\rho$ component, around baseline RAE2822 and optimized at Mach 0.74 Cl 0.782

## 6.3 Multipoint optimizations

In this section, multiple lift coefficient and multiple-Mach coefficients optimizations are conducted. They aim at validating the GSA approach, and is compared to an alternative formulation by Zingg et al. [146].

### 6.3.1 Weights calculations

The weighting strategy chosen to aggregate the objective functions is the inverse of an estimation of the utopia point distance [67], called "unattainable aspiration" [84], see Chapter 2, section 4. Consequently, potential gains expected by the optimization process have to be provided. To this aim, the pressure drag is decomposed into physical components: induced drag $Cd_{ind}$, wave drag $Cd_w$ and viscous pressure drag $Cd_{vp}$ using a far-field decomposition provided by the ONERA FFD code [29]. The potential gain at each condition is estimated as a fraction of these decompositions as shown in Eq. (6.1).

$$\omega_i = \frac{1}{\omega_w Cd_{wi} + \omega_{ind} Cd_{ind_i} + \omega_{vp} Cd_{vp_i}}, \tag{6.1}$$

where $i$ is the operating condition index. The idea behind this hypothesis is that it is easier to guess potential gains in terms of fractions of the drag components, which can be linked to physical phenomena, than to guess an absolute drag gain value. Wave drag is a good illustration since it should be completely removed by single-point optimizations. Gains obtained after a single point optimization provide a good estimation of the potential gain fractions $\omega_w$, $\omega_{ind}$, $\omega_{vp}$ in Eq. (6.1). For instance, the single point optimization usually provides shock-free airfoils, meaning that 100% of the wave drag can be reduced, so we take $\omega_w = 1$. When 20% of the viscous pressure drag and 5% of the induced drag are reduced during the single point optimization, we obtain $\omega_{ind} = 0.05$, and $\omega_{vp} = 0.2$. Fractions are here supposed independent of the operating condition, this approximation saves "number of operating conditions -1" single point optimizations. In this way, the optimization formulation is fully determined by an automated physical analysis of the problem, with one single point optimization, a GSA analysis on the baseline and a drag decomposition per selected operating condition. It is also possible to perform a single point optimization at each operating condition, that are not computationally expensive in 2D cases, to accurately estimate the utopia point.

For 2D cases like the RAE2822, we take in the following: $\omega_w = 1$, $\omega_{ind} = 0$, since there is no induced drag in 2D, and a shock-free solution is expected in single-point. $\omega_{vp}$ is determined by the single point optimization of the previous section.

### 6.3.2 Calculations convergence

Multipoint optimizations typically require hundreds of direct and adjoint calculations. Monitoring the convergence of such a high number of calculations is made easier with a statistical approach. Figures 6.12 and 6.13 plot the number of direct and adjoint calculations versus their convergence magnitude order. In 99% of the direct calculations, the residual L2-norm is reduced by 8 orders of magnitude. Such a convergence ensures that the RANS equations are well-resolved, which is a necessary condition for the gains obtained by optimizations to have a physical origin. In the worst cases they are reduced by 5.5 orders. In adjoint calculations, the residual L2-norms are reduced by 7 orders of magnitude in more than 99% of the cases. A low level of convergence in adjoint calculations creates noise in the gradients, degrades the optimizer performance, and perturbs the GSA algorithm. Here, such a noise is many orders of magnitude under the one due to the frozen turbulence assumption [33].

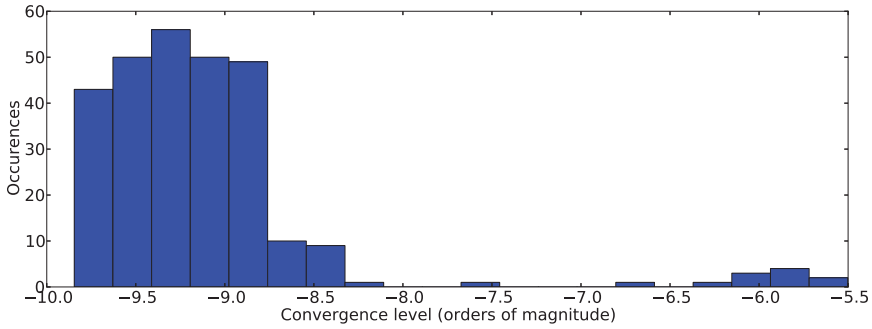



Figure 6.12: Direct calculations convergence statistics during the RAE2822 multi-Mach 5 points optimization

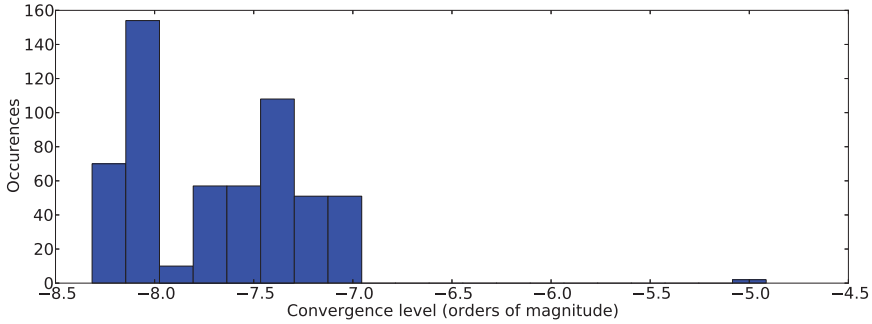



Figure 6.13: Adjoint calculations convergence statistics during the RAE2822 multi-Mach 5 points optimization

### 6.3.3 Multi-Cl optimizations

A multiple lift coefficient optimization on the range $Cl \in [0.5, 0.782]$ is run at Reynolds 6.5M, Mach 0.74. The GSA algorithm is used with a tolerance $\epsilon = 0.05$, and gives three required operating conditions: $Cl = 0.5, Cl = 0.62, Cl = 0.72$. For comparison, three single point optimizations are performed at $Cl = 0.5, Cl = 0.6, Cl = 0.782$. Resulting polars plotted in Fig. 6.14 show the advantages of multipoint optimization on single point ones. Indeed, the single-point optimization provides the best performance at their design point, but the neighborhood of operating conditions in which it occurs can be as small as 3 lift counts (1 lift count = 0.01 Cl amount) for the $Cl = 0.6$ optimization. In addition, the average gains of single point optimizations on the range $Cl \in [0.3, 0.89]$ are of 1.6 L/D for the $Cl = 0.5$ optimization, 2.7 L/D for the $Cl = 0.6$ optimization, which is an order of magnitude lower than the average gain (8.95 L/D) of the robust design. The single-point optimization at Cl=0.782 provides an average gain of 7.0 L/D, at the price of an unequal distribution, with a loss of 1.5 L/D on the range $Cl \in [0.3, 0.6]$, and a gain of 8.49 on the range $Cl \in [0.6, 0.89]$. The multipoint optimization provides both gains on these two intervals, with a 0.43 L/D gain on the range $Cl \in [0.3, 0.6]$, and a gain of 8.51 on the range $Cl \in [0.6, 0.89]$. It is noticeable that the single point optimal airfoil designed at $Cl = 0.5$ and $Cl = 0.6$ do not exhibit peaky polars, while the one designed at $Cl = 0.782$ is. A possible explanation is the presence of stronger shocks at high lift, which generates non-linear dependancies between the gradients and the operating conditions, and therefore specialized designs. Besides, none of these polars exhibit extremely peaky polars such as the ones observed by Drela in Fig. 2.5. The reason is the smooth CAD parametrization employed here, compared to the Hicks-Henne bumps used by Drela.

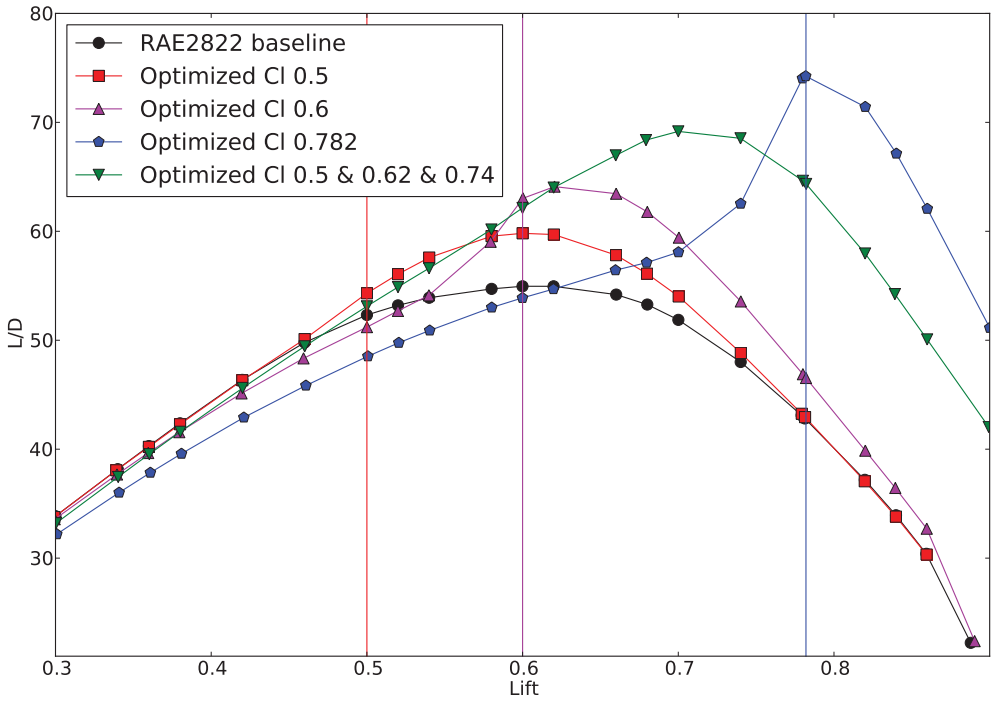



Figure 6.14: Comparison of lift over drag vs lift coefficient for the RAE2822 multiple-lift optimizations at mach 0.74

Parametrization is then a key point for robust optimization.

The present exercise shows the advantages of multipoint optimization on single-point optimization, but does not specifically validates the GSA method, and in particular, the fact that no descent directions were missed, leading to missed performance opportunities. To this aim, a 10 points optimization is performed on the range $Cl \in [0.5, 0.782]$, on which the GSA algorithm was used. Results are displayed in Fig. 6.15. Contrary to Chapter 4, section 4, where the purpose was to find weights so that the 3-points optimization problem gives the same solution as the 7 points optimization problem, in the present case, for a given operating condition, the associated weight is identical in the 3 points and the 10 points optimizations. While in Section 4.4, the problem setup of the 3-points optimization problem required the resolution of the 7 points one, it is not the case of the present test case, that is representative of real-life applications. Therefore, we do not expect the two problems to have identical solutions. Instead, we obtain close polars when compared to Fig. 6.14, and, also important, the 3-points optimization resulting polar shows no localized performance degradation at some operating conditions when compared to the 10 points one. The discrepancies are then partially explained by the fact that the two problems are not mathematically equivalent. The impacts of this effect can be reduced by changing the weights. One can notice that the 10 points solution performs better at the high Cl, but is slightly less performant in $Cl \in [0.6, 0.7]$, meaning that they are two different Pareto-optimal compromises.

Table 6.1 displays the dimensions of the gradient span estimated by the GSA algorithm during the 10 points multi-lift RAE2822 optimization. For 26 of the 30 optimization iterations the dimension of the gradient span for the drag is 4. Iteration 20 corresponds to a restart of the BFGS approximation, which led to an increase of drag and an additional independent descent direction. The dimension at the first iteration on the original RAE2822 is 3, while it is of 4 at convergence. This under-estimation of the number of required operating conditions can explain a part of the

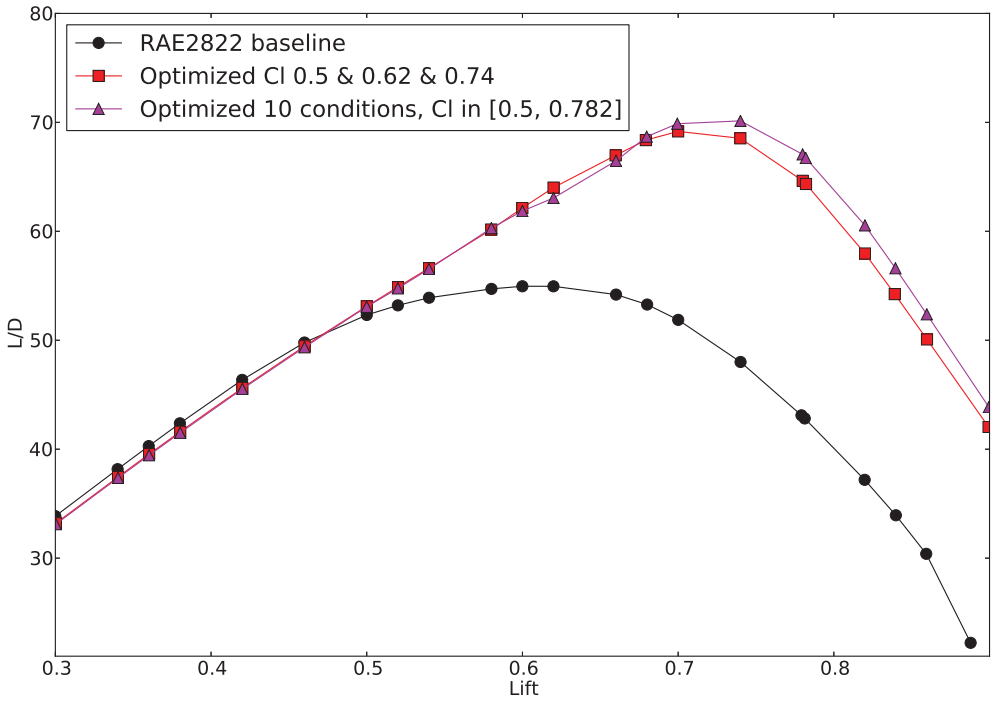



Figure 6.15: Comparison of lift over drag vs lift coefficient for the RAE2822 multiple-lift optimizations at mach 0.74, with 10 and 3 operating conditions.

discrepancies between the 3 and 10 points optimizations shown in Fig. 6.15.

| Optimization iteration | Gradient span dimension |
|---|---|
| 1 | 3 |
| 2...8 | 4 |
| 9 | 5 |
| 10 | 6 |
| 11...19 | 4 |
| 19 | 4 |
| 20 | 5 |
| 21...30 | 4 |

Table 6.1: Gradient span dimension of the pressure drag at constant lift, provided by GSA with $\epsilon = 0.05$, on the Multi-lift RAE2822 optimization at Mach 0.74, Reynolds 6.5M

### 6.3.4 Multi-Mach optimizations

Multiple-Mach optimization are performed and summarized in Fig. 6.16 on the range $Mach \in [0.68, 0.74]$, at Reynolds 6.5M and Cl 0.782. Four required operating conditions were detected by the GSA algorithm. First, a unitary weight optimization (red) is compared with an utopia point weighting (purple) with the same potential gains fractions as in the last section. It is noticeable that the utopia-weighting gives a better distribution of the pressure drag gains, at the price of an

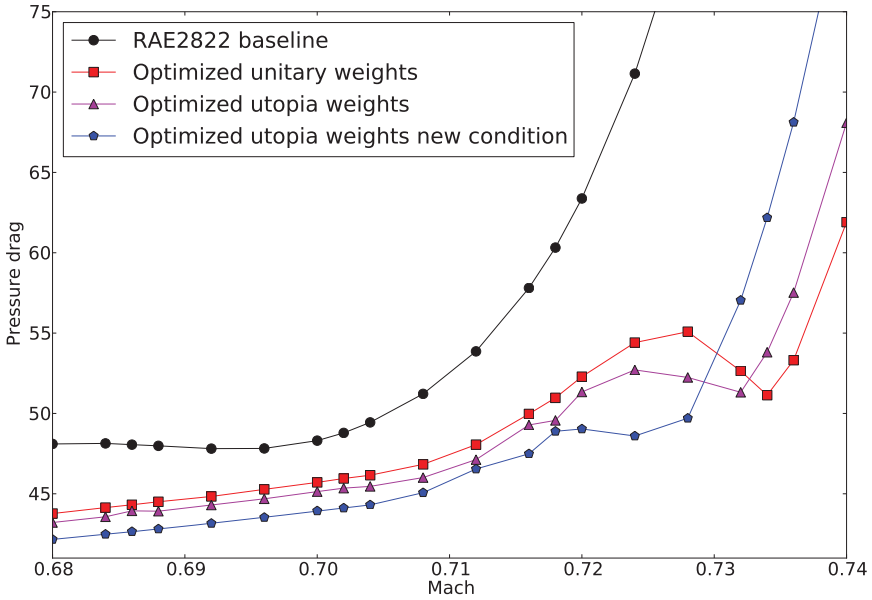



Figure 6.16: Comparison of pressure drag vs Mach number for the RAE2822 multipoint optimizations

earlier drag rise. Besides, both resulting polars exhibit a drag-rise effect around $Mach = 0.725$, clearly lower for the utopia weighting: $Cd(Mach = 0.723) - Cd(Mach = 0.732) = 1$ *d.c.* than for the unitary weights optimum: $Cd(Mach = 0.728) - Cd(Mach = 0.735) = 2.5$ *d.c.*. The drag-creep effect is due to missed descents directions in the gradient span, see Part II. To conclude, the weights of the problem also has an importance and is strongly linked with the operating conditions choice.

Since the GSA algorithm was used on the RAE2822 baseline, and that the shape has changed during the optimization, it seems that the vector basis that generates the drag gradient span has changed, which leads to the drag creep. In theory, the GSA algorithm should be used at each iteration of the optimization process because all the theory was demonstrated at a given set of design variable, but since it would be too costly, it is used only on the baseline, with the hypothesis that the selected operating conditions are still sufficient to generate the gradient span during the optimization. Since it seems not to be the case here, a GSA analysis is performed on the shape resulting of the multipoint optimization with utopia weights. Four operating conditions are also detected, but now the conditions $Mach = 0.724$ is detected. The updated set of operating conditions leads to an optimal shape which polar is displayed in blue in Fig. 6.16. It is noteworthy that the drag-creep is now negligible compared to the one observed on the previous optimizations. Once again, drag-creep removal is achieved at the price of an earlier drag divergence, since these solutions are Pareto compromises, one cannot gain drag at each Mach.

The present optimizations illustrate how multipoint optimization results depends on the weights and the operating conditions choice. These are linked together since their modification can have a comparable effect on the results, i.e. performance degradation in some areas of the operating conditions. A GSA analysis can be used to distinguish if the origin of such a degradation is due to a missing operating condition or not. It also shows that an analysis of the resulting polars is always required, and that the GSA step followed by an optimization can be used in an iterative way. The convergence of such a process is an open question. However, from a practical point of view, the designer can select a preferred solution from multiple steps of the process.

Such results can be compared with the ones obtained by Zingg et al. in Fig. 6.17. They also proposed an automated procedure for selection of weights and operating conditions, aiming at obtaining a constant drag on the Mach range. In the study of Ref [146], the RAE2822 optimization also requires two analysis phases, and the addition of an operating condition after a first optimization. However the process leads to a polar that exhibits a drag-creep of 1 dc, even if it is lower than the one observed at intermediate solutions which is of 4 d.c. Besides, as explained by Zingg et al., searching for a constant drag may not be a practical objective and is not generalizable to multiple-lift optimizations, see the previous section, and to multi-dimensional operating conditions spaces. Finally, such a heuristic method does not provide a mathematical explanation of the mechanism behind drag-creep, while the GSA method relying on the detection of independent descent directions does.

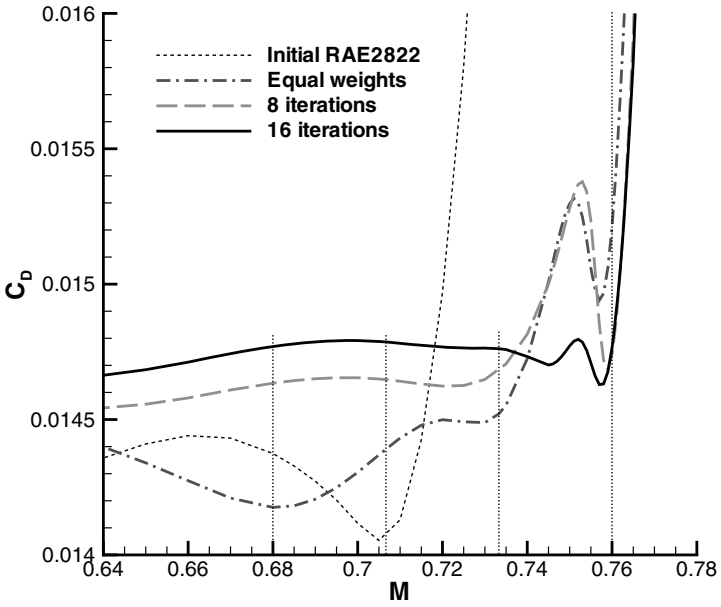



Figure 6.17: RAE2822 multi-Mach optimization by Zingg et al.[146] using an automated selection procedure of weights and operating conditions

# CHAPTER 7

# XRF-1 wing optimizations

***Summary*** In this chapter, the wing of the XRF-1 wing-body configuration, representative of modern civil transport aircraft, is optimized in viscous and transonic conditions with the adjoint method. A single point optimization and a multi-lift-multi-Mach optimization are performed. The polars of the resulting designs are compared with the ones of the XRF-1 baseline, and the single-point optimum design. The test case aims at showing the applicability of the robust optimization method proposed in the present thesis to real-life cases. The effect of the noise on the gradients of the objective functions on both the multipoint optimization and the GSA algorithm are explained and solutions are proposed. Finally, the XRF-1 is optimized taking into account for aeroelasticity, using coupled aero-elastic CFD simulations. The robust rigid multipoint optimum is taken as starting point for a twist derivative-free multipoint optimization.

***Résumé*** Dans ce chapitre, l'aile de la configuration voilure-fuselage XRF-1, représentative des avions de transport modernes, est optimisée en écoulement visqueux transsonique à l'aide de la méthode adjointe. Une optimisation mono-point et une optimisation multi-portance-multi-Mach sont réalisées. Les polaires des différents designs ainsi obtenus sont comparées. Ce cas test montre l'applicabilité de la méthodologie d'optimisation robuste présentée dans cette thèse à des cas réels. Les effets du bruitage des gradients de la fonction coût sur l'optimisation multi-point et l'algorithme GSA sont expliqués, et des solutions proposées. Finalement, le XRF-1 est optimisé avec prise en compte de l'aéro-élasticité, en utilisant des simulations CFD couplées aéro-élastiques. L'optimum robuste obtenu par optimisation multi-point rigide est utilisé comme point de départ pour une optimisation de twist, sans gradient.

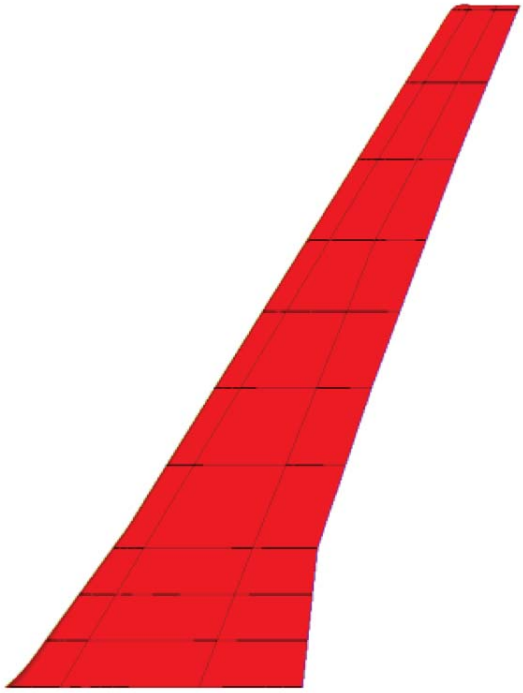
## 7.1 Parametrization

The in-house parametric and differentiated CAD engine PADGE presented in Chapter 5 section 2, is used to build a parametric model of the XRF-1 wing shown in Fig. 7.1 and Fig. 7.2. Eleven wing sections are parametrized by parametric B-splines in a similar way as the RAE2822 airfoil is parametrized in Chapter 6. Coons patches [106] are used to create parametric surfaces between the parametric airfoils. Since the most outboard wing section is modified by the optimizer, the wing tip has also to be deformed, so a parametric wing tip is also modeled. These parametric templates were programmed for the present project.

Decreasing the wing thickness would decrease the drag, but it would increase the mass of the internal structure. The compromise is multidisciplinary. Because no structural sizing is involved in the pure aerodynamic optimization process, the two structural beams are built in the wing CAD model, see Fig. 7.1 in red, and their thickness are geometrical constraints. The wing sweep is kept constant, also for its impact on the structure that is not simulated in our case. Similarly, the planform of the wing is fixed, the wingspan, sweep and reference surface are frozen. Their choice require the multidisciplinary compromise between high-speed fuel burn, low-speed performance such as take-off field length, weight balance, max take-off weight, operating cost etc.; which are not addressed in the present project.

The whole model, displayed in Fig. 7.1, contains 91 design variables.


Figure 7.1: Parametric CAD model of the XRF-1 wings



(a) Top view

(b) Side view

Figure 7.2: XRF-1 wing PADGE model

## 7.2 CFD simulation setup

### 7.2.1 Initial mesh choice

The Reynolds-Averaged-Navier-Stokes solution and its discrete adjoint are computed on the structured mesh shown in Fig. 7.3, made of 140 blocks with 22 millions cells. The height of the first layer of cells is such that $y^+ = 1$ at Reynolds 49.9M. The same numerical settings as for the RAE2822 optimizations in Chapter 6 are used, to recall the main ones, we used a second-order implicit Roe scheme with the Spalart-Allmaras turbulence model. A mesh convergence study on the XRF-1

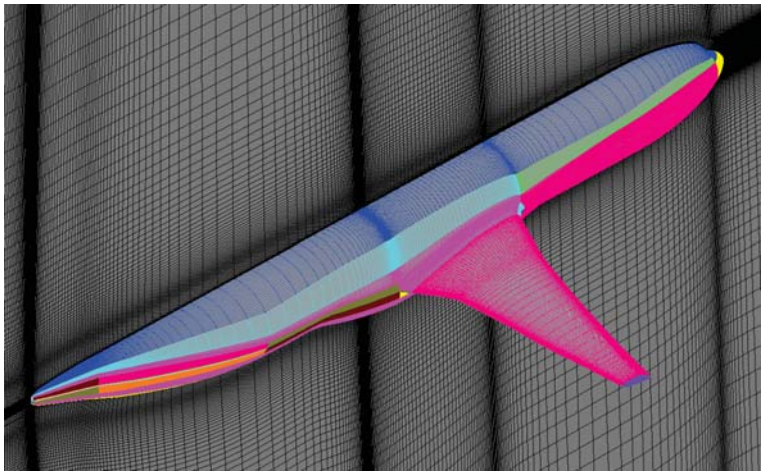


Figure 7.3: Multiblocks structured mesh of the XRF-1 model

configuration was performed by Nguyen-Dinh [98]. The mesh hierarchy of [98] is consistent in topology. The results obtained on our 22 millions cells mesh are compared to it in Tables 7.1 and 7.2. The angle of attack is frozen, which leads to different lift coefficients depending on the mesh. The drag and lift values obtained on the finest mesh of 100 millions cells are taken as reference,

| Mesh size | $y^+$ | $CL_p$ | $Cd\ Error\%$ | $CD_p\ Error\%$ | $CD_f\ Error\%$ |
|---|---|---|---|---|---|
| $100M$ | $\frac{8}{27}$ | 0.536 | 0. | 0. | 0. |
| $74M$ | $\frac{4}{9}$ | 0.534 | $-0.610$ | $-0.3913$ | $-1.01$ |
| $22M$ | 1 | 0.527 | 1.32 | 1.50 | 1.00 |
| $13.5M$ | $\frac{2}{3}$ | 0.527 | 4.37 | 5.84 | 1.65 |
| $10M$ | 1 | 0.527 | 4.78 | 6.56 | 1.55 |
| $3.2M$ | 1 | 0.517 | 19.0 | 29.03 | 0.657 |

Table 7.1: XRF-1 near field analysis; $Mach = 0.83$ ; $AoA = 2.607^o$ ; $Re = 49.9\ 10^6$. $Cl$, $CL_p$, $Cd$, $CD_p$ and $CD_f$ on a mesh hierarchy from Nguyen-Dinh's phd thesis [98].

| Mesh size | $CD_{spurious}(\times 10^{-4})$ |
|---|---|
| $100M$ | 2.68 |
| $74M$ | 2.78 |
| $22M$ | 3.41 |
| $13.5M$ | 14.3 |
| $10M$ | 14.9 |
| $3.2M$ | 55.2 |

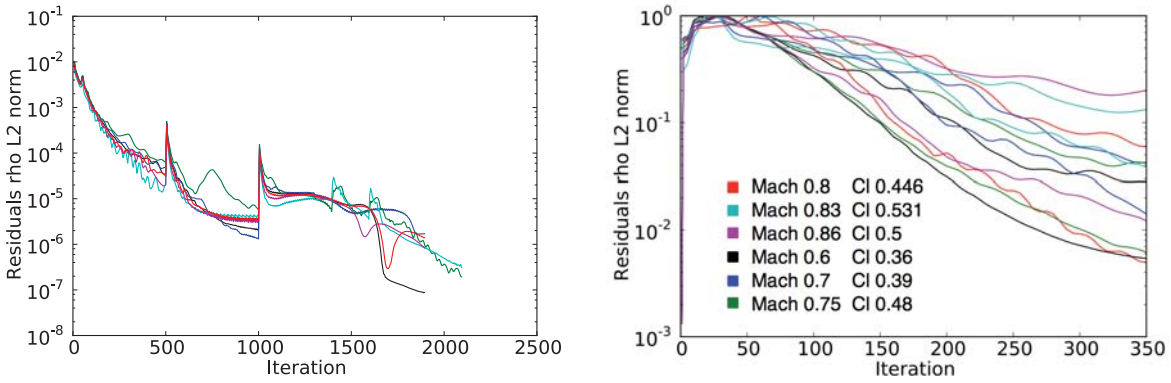
Table 7.2: XRF-1 ; $Mach = 0.83$ ; $AoA = 2.607^o$ ; $Re = 49.9\ 10^6$. Far-field spurious drag on a mesh hierarchy from Nguyen-Dinh's phd thesis [98].

relative errors in percentage are computed on each mesh. In Table 7.1 the near field analysis shows that the 22 millions cells mesh provides a relative error of about 1.% on the drag compared to the finest mesh, both in the near-field and far-field analysis, and 1.7% on the lift, while the 74 millions cells mesh divides this error by two for a more than three times higher CPU cost. The 13.5 millions cells mesh provides a relative precision of 4.3% on the drag, which is the order of magnitude of the gains expected by the optimizations. In addition, Tab. 7.2 shows that the spurious drag quantifying non-physical dissipation, is of 14.3 d.c. on the 13.5M cells mesh, which is also the order of magnitude of the expected gains by the optimization. On the other hand, it is of 3.4 d.c., on the 22M cells mesh. One could still perform a numerical optimization using the 13.5 cells mesh, but it has to be kept in mind that the aim of optimization is to modify a shape in order to improve its in-flight performance. Consequently, the observed performance variations have to be related to physical phenomena modifications and not to a numerical behavior of the model. If the measured performance variations due to optimization are of a similar order of magnitude as the precision of the numerical model, then it can be difficult to justify. So we consider that the 13.5M cells mesh is not the best-suited to the demonstration.

### 7.2.2 Convergence of direct and adjoint calculations

In addition to mesh convergence, the residuals convergence is a critical point in optimization. The norms of the residuals quantifies how much the flow state equations are satisfied, so having them as close to zero as possible is required to link the optimization gains to a real performance improvement, and not a violation of the state equations due to modification of the numerical model during the optimization. Besides, in Chapter 5, it has been shown that the adjoint to the flow supposes that the residuals of the state equation are null. If it is not the case, then the obtained gradients are inexact, which is known to slow down the optimization process and degrade the results [33]. Direct and adjoint calculations convergence are displayed in 7.4, on the XRF-1 baseline. The residuals L2-norm of the direct calculations are reduced to $10^{-6}$ for all the flight conditions.

The residuals of the adjoint calculations are plotted against LU-SSOR iterations, each of them using 10 LU relaxation iterations. The optimal number of LU relaxations has been chosen to minimize the CPU cost for a given convergence level. The levels of convergence of the adjoint calculations are between 0.8 and 2.1 orders of magnitude. The number of iterations that provides such a level of convergence has been chosen by comparison of the obtained gradients and the level of noise due to the linearization hypotheses: frozen turbulence, thin layer approximation. Besides, we noted that the L2-norm being the sum of all square residuals, the quantity can be greatly affected by a poor convergence in some areas of the shape that have a low influence on optimization, far from parametrized shape areas. The situation for adjoint is quite different from a direct calculation, a high residual in the RANS equation means that the flow is ill-resolved, so that the solution is non-physical. In the adjoint approach, the objective is to obtain sufficiently precise gradients to achieve optimizations, and gradient-based optimizers are robust to noise [44]. To this aim, the adjoint field has to be adequately resolved close to the parametrized surfaces. Ideally, of course, the adjoint problem should be solved up to machine precision. In practice however, the algorithms should be robust to noise, since industrial applications with complex geometrical details often lead to non-ideal meshes and poor convergence. We will see that this noise also has impacts on the multipoint optimization strategy, detailed in the following sections.




(a) Direct calculations convergence at multiple operating conditions

(b) Adjoint calculations convergence at multiple operating conditions

Figure 7.4: Direct and adjoint calculations convergence history on the XRF-1 baseline at the 6 operating conditions of the multipoint optimization

## 7.3 Single point optimization strategy

A single point optimization of the configuration is run at Mach 0.83, Reynolds 49.9M and Cl 0.531. This enables to check the behavior of the optimization process, and to compare the results with multipoint optimizations. The L-BFGS-B code from Zhu et al. [145] is used as optimizer. The lift constraint $Cl - Cl_0 = 0$ is handled in the flow solver by a target lift approach. Newton iterations with finite differences on the function $Cl(AoA)$ are performed during the convergence of the flow and the cost is about 20% more than a fixed $AoA$ calculation. A Lagrangian is derived to compute the derivative of the drag at constant lift in Eq. (7.1) and Eq. (7.2), the derivatives being calculated by discrete adjoints of $Cl$ and $Cd$:

$$\frac{dL(Cd, Cl - Cl_0)(\chi)}{d\chi} = \frac{dCd(\chi, AoA)}{d\chi} + \lambda_{Cl}\frac{dCl(\chi, AoA)}{d\chi}. \tag{7.1}$$

The adjoint equation associated to the $Cl(\chi, AoA) = Cl_0$ constraints is:

$$0 = \frac{dCd(\chi, AoA)}{dAoA} + \lambda_{Cl}\frac{dCl(\chi, AoA)}{dAoA}. \tag{7.2}$$

Since this equation is scalar, we get:

$$\lambda_{Cl} = -\frac{\frac{dCd(\chi,AoA)}{dAoA}}{\frac{dCl(\chi,AoA)}{dAoA}}, \tag{7.3}$$

and therefore the gradient of the Lagrangian is:

$$\frac{dL(Cd, Cl - Cl_0)(\chi)}{d\chi} = \frac{dCd(\chi, AoA)}{d\chi} - \frac{\frac{dCd(\chi,AoA)}{dAoA}}{\frac{dCl(\chi,AoA)}{dAoA}}\frac{dCl(\chi, AoA)}{d\chi}. \tag{7.4}$$

This Lagrangian gradient is provided to the l-bfgs-b algorithm. Figure 7.5 shows that the target

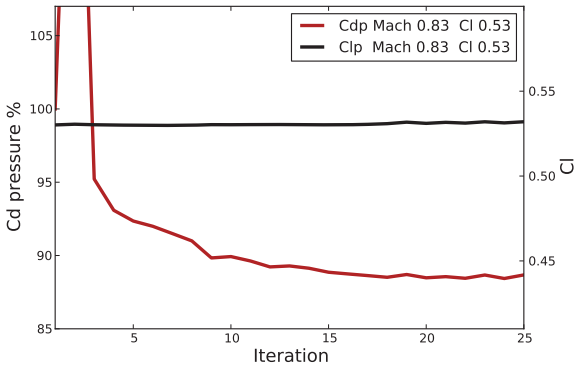
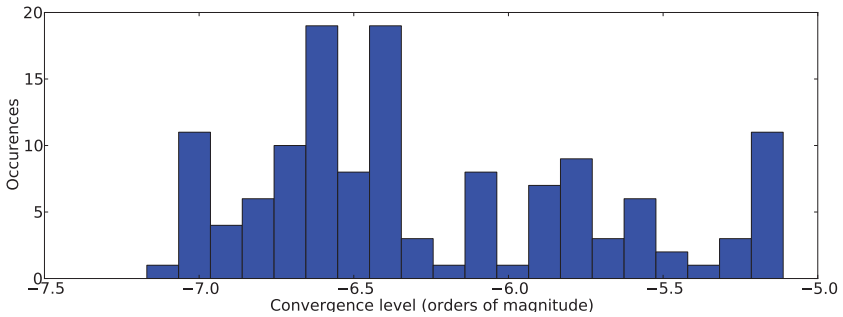



Figure 7.5: XRF-1 single point optimization convergence history

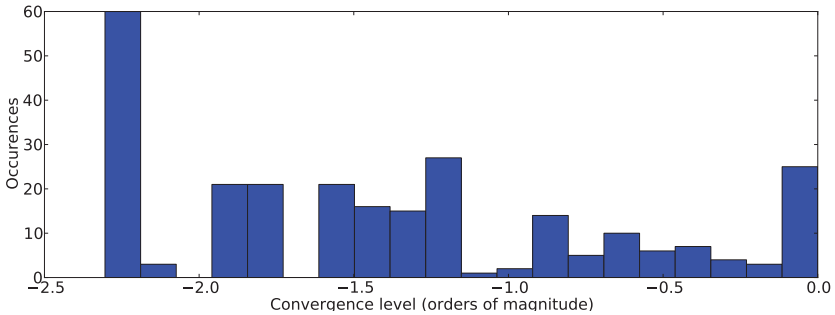
lift approach succeeds in maintaining the lift during the optimization by automatic adjustment of the angle of attack. The overall drag reduction of 11% is mainly due to the shock smoothing as shown in Fig. 7.8, which also leads to a viscous pressure drag reduction. Fig. 7.7 also shows smooth pressure coefficient iso-lines and the global view of the configuration.

Figure 7.6, similarly to figs. 6.12 and 6.13, shows the convergence statistics for the single-point optimization. The direct calculation convergence levels are satisfactory, 100% of them being higher

that 5 orders of magnitude. 200 adjoint calculations have a convergence level higher or equal to 1 order of magnitude, and 60 of them have a convergence level higher than 2 orders of magnitude. However 25 adjoint calculations did not converge adequately, which represents 8% of them and are mostly at the second iteration of the algorithm, which represents 12 calculations. This second step was rejected in the line-search phase of the L-BFGS-B algorithm (see more details in the section 2.1.3), so the gradient was not taken into account in the BFGS approximation. The second iteration of the algorithm is always a special case, since the Hessian approximation is the identity matrix, meaning that the full steepest descent step is used, often leading to too important design variables modification and then a drag increase, which is observed in Fig. 7.5. A solution for that is to modify the L-BFGS-B algorithm to decrease the first step length, modification that we accomplished during the present project.




(a) Direct calculations convergence statistics




(b) Adjoint calculations convergence statistics

Figure 7.6: Direct and adjoint calculations convergence statistics during the XRF-1 6 operating conditions optimization

## 7.4 Single point optimization detailed results

In Fig. 7.8 the displayed deformations are coherent with the pressure coefficients modifications: material is added behind the upper-wing shock position, to lower the re-compression, and removed in the supersonic area to lower the acceleration. The resulting wave drag is almost zero at the optimum. The lower and upper deformation curves cross at four points: leading and trailing edge, and the two structural beams intersections, which means that the geometrical constraint of constant beam thickness are respected. The lift repartition across the wingspan is also slightly moved outboard to reduce the induced drag. Resulting gains are summarized in Fig. 7.9.

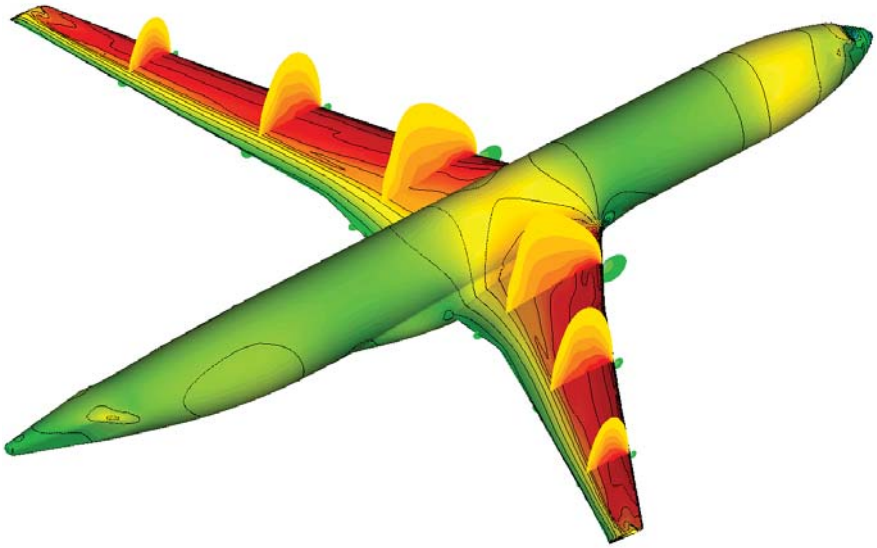
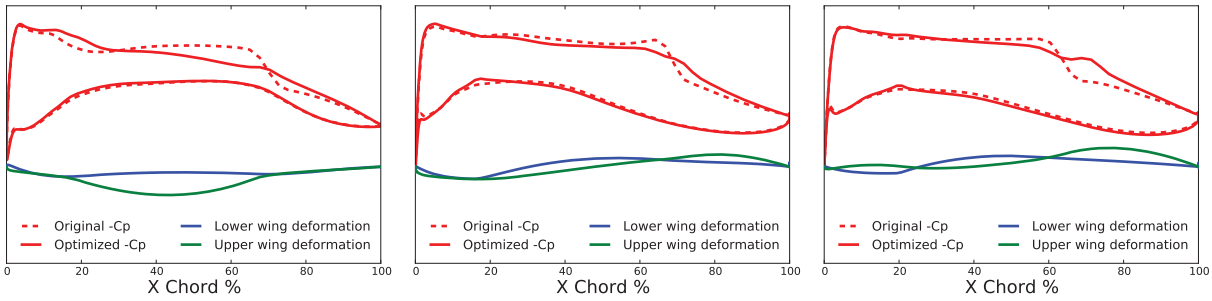


Figure 7.7: Pressure coefficient at single point optimum




(a) 15 % Span (b) 40 % Span (c) 80 % Span

Figure 7.8: Cp distributions and airfoil deformations at single point convergence, Mach 0.83 Cl 0.53

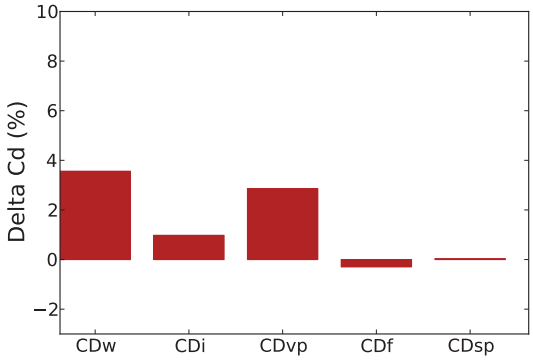



Figure 7.9: Drag gain as percentages of total initial drag for the single point XRF-1 optimization case

## 7.5 The operating conditions choice

### 7.5.1 GSA on the XRF-1

In flight, the aircraft encounters varying aerodynamic flight conditions : angles of Attack, Mach and Reynolds numbers. A robust design should therefore take into account these three ranges. In order to perform the selection of operating conditions used in the optimization process, the GSA algorithm is used (see Chapter 3). First, a multiple Mach problem is analyzed in the range $Mach \in [0.6, 0.86]$, at Cl 0.531 and Re 49.9M. 15 samples are computed, and the gradient span for the pressure drag appears to be of dimension 5. Then, a multiple Angle of attack problem is analyzed for the range $AoA \in [2.2, 2.6]$ at Mach 0.83, Cl 0.531, with the same number of samples. The gradient span is of dimension 2. The Reynolds effect is also studied, and a sampling of 15 operating conditions are selected in $Re \in [25 * 10^6, 71 * 10^6]$ at Mach 0.83, Cl 0.531. The gradient span dimension is 1, which means that any multiple Reynolds optimization problem on this range is equivalent to a single Reynolds optimization problem. These results suggests that the main non-linearities driving the pressure drag gradient span arise from the Mach number variations. In addition, the angle of attack has a lower influence on the gradient span dimension, and the Reynolds number has no influence at all. This last information is also interesting in the context of manual design, as a single Reynolds design should not show a poor performance when the Reynolds varies.

Finally, to setup the multipoint optimization problem, 32 samples are taken in the operating conditions space $(Mach, AoA) \in [0.6, 0.86] \times [2.2, 2.6]$. From this original sampling of the gradient space, 6 operating conditions are selected by GSA and displayed in fig. 7.10. First of all, the design space being of dimension 91, it is noteworthy that only 5 conditions are sufficient to build a basis of the spanned 38 initial conditions drag gradients. It shows the interest of the GSA approach in terms of computation time saving.

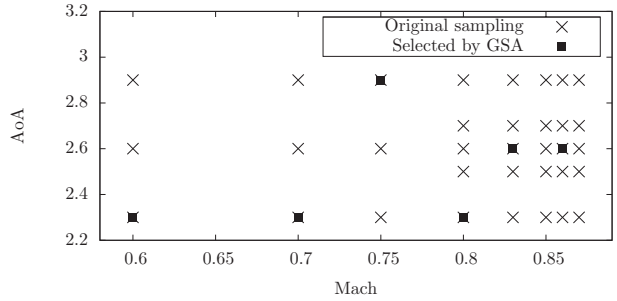



Figure 7.10: GSA operating conditions selection

The GSA algorithm analyses the linear dependancies between the gradients $\nabla_\chi j(\chi, \alpha)$, when $\alpha$ varies. Since the gradient of the objective function is given by : $\nabla_\chi j(\chi, \alpha) = \frac{dj(S,\alpha)}{dS} \frac{dS(\chi)}{d\chi}$ (Eq. 5.33), and that in a rigid simulation $\frac{dS(\chi)}{d\chi}$ does not depend on $\alpha$, the non-linear dependancies between the gradients $\nabla_\chi j(\chi, \alpha)$ necessarily come from the $\frac{dj(S,\alpha)}{dS}$ terms, displayed in Figs. 7.11 to 7.16. Since those derivatives represent the total derivative of the drag with respect to the Z coordinates of the aircraft skin, they give the effect of adding or removing material, at first order. Basically, to reduce the drag at a given condition, on the upper wing, one should add material in the blue areas and remove some in the red areas. Such an analysis should be made in conjunction with a lift sensitivity map analysis, in order to keep the lift constant. Even better, one can use a Lagrangian sensitivity map of the drag at constant lift. On Figs. 7.11 to 7.16, it is observed that maps displays different

patterns when the operating conditions varies. Shock have a particularly large influence on the areas of the shapes to which the drag is sensitive, and their position changes when the Mach number changes. Each of these patterns lead to a linearly independent drag gradient when assembled with the CAD sensitivity. This links the non-linear physical phenomena of shocks displacement when the operating condition varies, to the gradient span dimension and in the end the choice of the operating conditions for the optimization problem by the GSA algorithm.

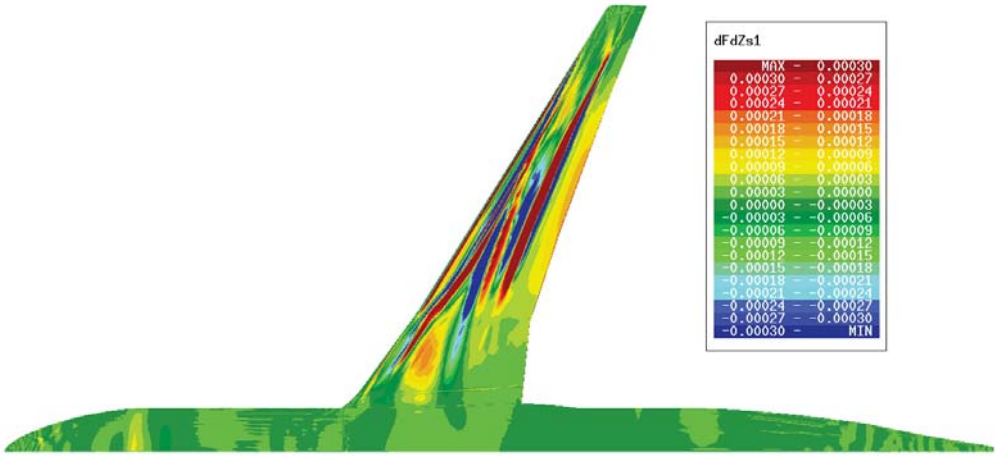



Figure 7.11: XRF1 pressure drag derivative wrt Z coordinate at Mach 0.8 Cl 0.46

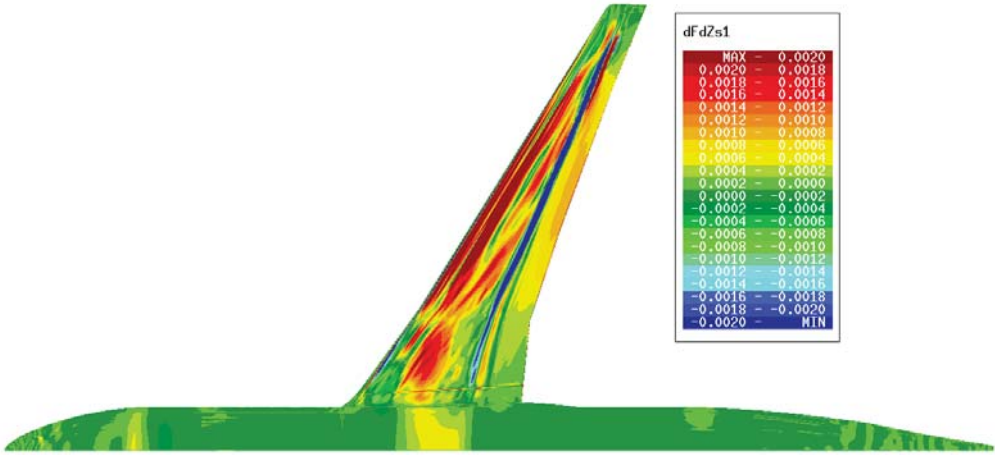



Figure 7.12: XRF1 pressure drag derivative wrt Z coordinate at Mach 0.83 Cl 0.53

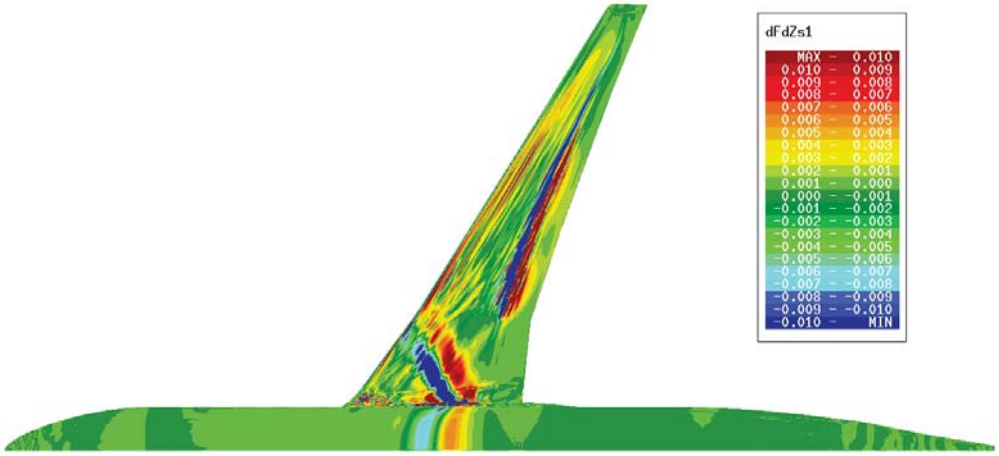



Figure 7.13: XRF1 pressure drag derivative wrt Z coordinate at Mach 0.86 Cl 0.5

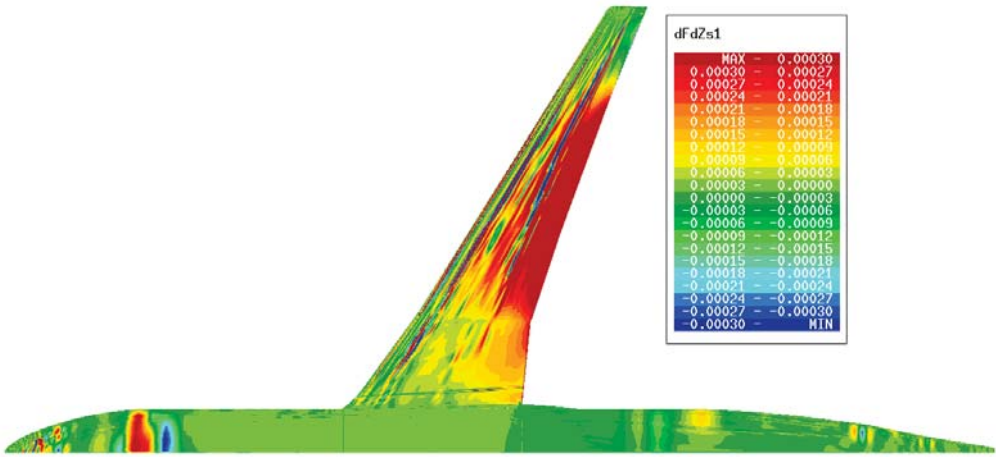



Figure 7.14: XRF1 pressure drag derivative wrt Z coordinate at Mach 0.6 Cl 0.36

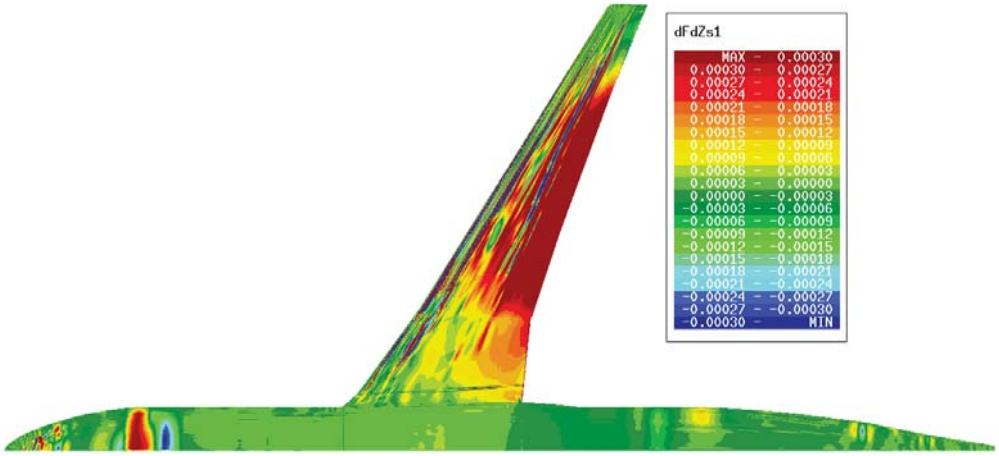



Figure 7.15: XRF1 pressure drag derivative wrt Z coordinate at Mach 0.87 Cl 0.39

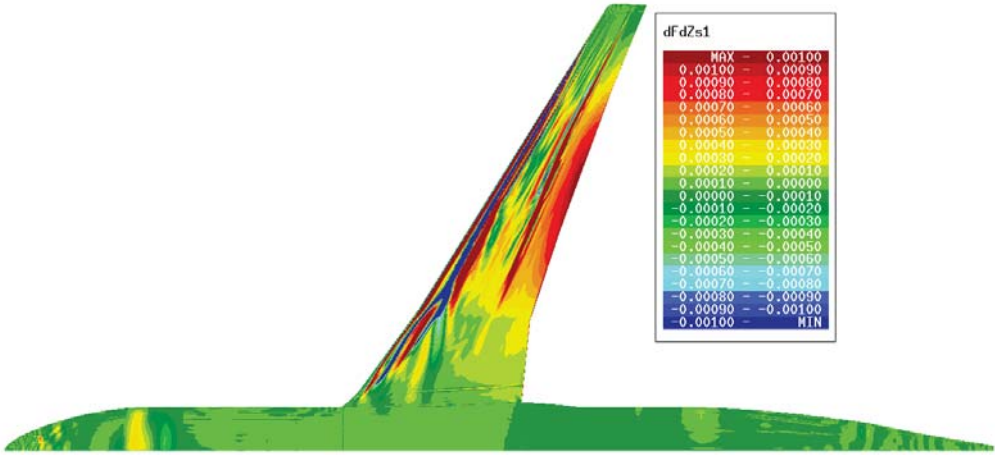



Figure 7.16: XRF1 pressure drag derivative wrt Z coordinate at Mach 0.75 Cl 0.48

In multi-point optimizations, the gradient of the objective is computed by:

$$\nabla_\chi J(\chi) = \sum_{k=1}^{m} \omega_k \frac{dj(S,\alpha_k)}{dS} \frac{dS(\chi)}{d\chi}, \tag{7.5}$$

which can also be re-written algebraically,

$$\nabla_\chi J(\chi) = \left[ \sum_{k=1}^{m} \omega_k \frac{dj(S,\alpha_k)}{dS} \right] \frac{dS(\chi)}{d\chi} \tag{7.6}$$

Therefore, we can define the drag surface sensitivity in a multi-point context by:

$$\frac{dj(S)}{dS}_{MP} = \sum_{k=1}^{m} \omega_k \frac{dj(S,\alpha_k)}{dS}, \tag{7.7}$$

which, gives the gradient of the aggregated objective function $\nabla_\chi J(\chi) = \frac{dj(S)}{dS}_{MP} \frac{dS}{\chi}$, similarly to the single-point case where $\nabla_\chi j(\chi,\alpha) = \frac{dj(S,\alpha)}{dS}\frac{dS(\chi)}{d\chi}$. Figure 7.17 plots $\frac{dj(S)}{dS}_{MP}$ on the XRF-1, using the 6 operating conditions selected by the GSA algorithm and the weights calculated in the following section. $\frac{dj(S)}{dS}_{MP}$ combines the weighted sensitivities at multiple operating conditions in a single field which interpretation is similar to single-point sensitivity fields. Such a synthetic information makes easier the understanding of the multipoint aerodynamic design.

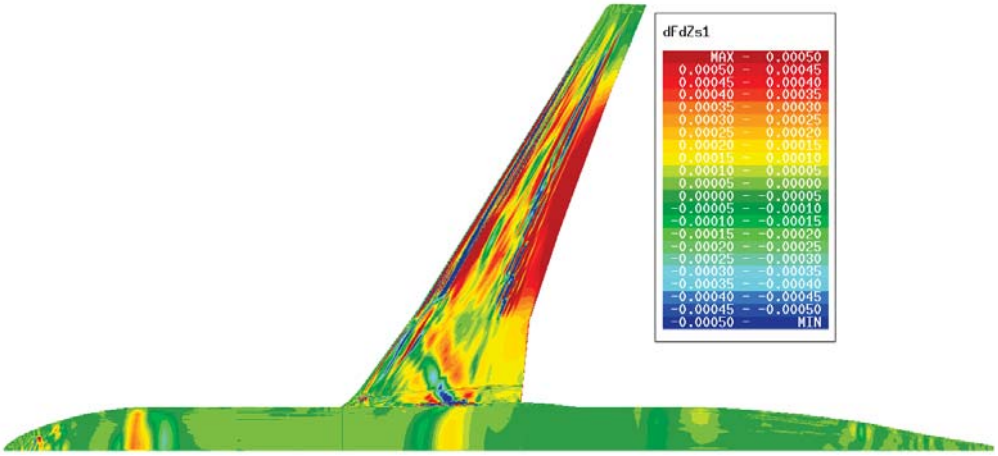



Figure 7.17: XRF-1, weighted sum of pressure drag derivatives wrt. Z coordinates at the operating conditions selected by the GSA algorithm.

### 7.5.2 The noisy gradients issue

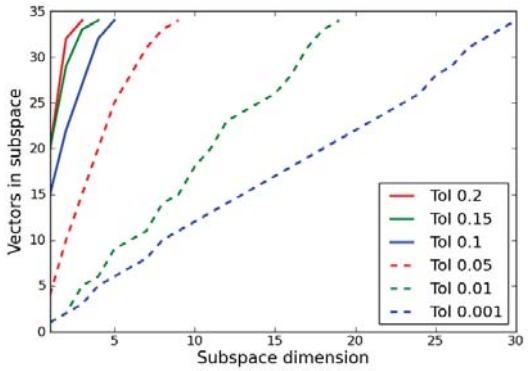
The GSA algorithm uses a tolerance $\epsilon$ as input, which must be chosen adequately. For quasi analytical problems of Part II, it is not an issue because the problem is non-noisy, so the alternatives proposed in Chapter 3 such as machine precision or the optimization algorithm stop criteria are viable. In the case of our 3D CFD simulations, the gradient has a noise level of 5 to 10 %, which is a real issue. The next figures show different behaviors of the algorithm depending on the $\epsilon$ parameter for the last multiple Mach and AoA problem. In Fig. 7.18a, the gradient count $c(\mathbf{I_m})$ Eq. (7.8) during the GSA process is plotted at each iteration $m$, for the current gradient subspace $\mathbf{I_m}$.

$$c(\mathbf{I_m}) = \text{Card}\{\alpha \in \mathbf{I_M}, ||\nabla_x j(x,\alpha) - Proj(\nabla_x j(x,\alpha), \mathbf{K}_{I_m,x})|| < \epsilon ||\nabla_x j(x,\alpha)||\}. \tag{7.8}$$

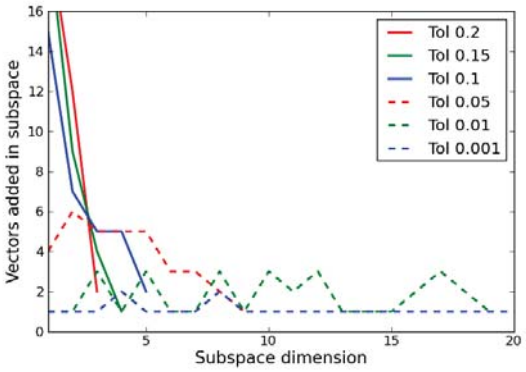
An asymptote of equation $c(\mathbf{I_m}) = m$ is observed when $\epsilon$ goes to zero. In the latter case, only one operating condition is added at each iteration, so the gradient span dimension is estimated to be the number of samples, and at most the number of design variables. When $\epsilon$ is higher than the noise level, the slope of the curve is high, and the estimated gradient span dimension is much lower than in the case of null tolerance. On the other hand, a too high $\epsilon$ increases the risks of underestimating the gradient span dimension. Plotting the slope of the curves in Fig. 7.8, gives a better view on the problem. At each iteration, a certain number of vectors are considered as added into the gradient subspace by GSA. This number is the slope, or increment count, of the previous curve and is displayed in Fig. 7.18b. GSA is searching for the main descent directions, using $c(\mathbf{I_m})$ as quantifier for the discrimination between two possible sets. So the algorithm is supposed to generate a decreasing suite $c(\mathbf{I_m})_{k+1} - c(\mathbf{I_m})_k$, where $k$ is the main GSA iteration index. The choice of $\epsilon$ is much easier with this information, and 0.1 is the best compromise given Fig. 7.8.

Besides, there is a mathematical explanation of why noise artificially adds dimensions to the measured gradient span. The GSA algorithm selects $m$ operating conditions using an initial set of $M$ gradients, such that:

$$m = Rank(\mathbf{G}) + 1, \tag{7.9}$$




(a) History of added vectors to the gradient span vector space




(b) Growth of subspace dimension history

Figure 7.18: GSA algorithm history

where the gradient matrix is:

$$\mathbf{G} = \begin{bmatrix} \frac{\partial}{\partial \chi_1} j(\chi, \alpha_1) & & \cdots & & \frac{\partial}{\partial \chi_1} j(\chi, \alpha_M) \\ & \ddots & & & \\ & & \frac{\partial}{\partial \chi_i} j(\chi, \alpha_p) & & \\ & & & \ddots & \\ \frac{\partial}{\partial \chi_n} j(\chi, \alpha_1) & & \cdots & & \frac{\partial}{\partial \chi_n} j(\chi, \alpha_M) \end{bmatrix}. \tag{7.10}$$

When a noise is applied to the coefficients of the gradient matrix $\mathbf{G}$, the matrix can be considered as randomly perturbed. The rank of random matrices is full with probability 1, see [38], because the space of singular matrices has a null Lebesgue measure in $\mathbb{R}^{M \times n}$. This means that a noisy numerical estimation of the gradients matrix has a higher rank than its exact value, leading to an over-estimation of the required number of operating conditions by the GSA algorithm.

To conclude, setting an adequate tolerance for the GSA algorithm is a balance between robustness to the gradients noise, and avoiding an under-estimation of the gradient span dimension, which leads to less robust shapes to operating conditions changes. Running multiple times the GSA algorithm with increasing tolerances enables to draw the plot Fig. 7.18. Its analysis provides an adequate tolerance values.

## 7.6 Weights calculation

The weights $\omega_k, k \in (1..m)$ of the objective function are an estimation of the utopia point distance [67], see Chapter 2, section 2.4.3. As a consequence, an estimation of the potential gains expected by the optimization process has to be provided. To this aim, the pressure drag objective is decomposed in physical components: induced drag $Cd_{ind}$, wave drag $Cd_w$ and viscous pressure drag $Cd_{vp}$ using the ONERA FFD code [29]. The potential gain at each condition is estimated as a fraction of these decompositions as shown in Eq. (7.11). Table 7.3 summarizes the obtained weights.

$$\omega = \frac{1}{0.9Cd_w + 0.2Cd_{ind} + 0.1Cd_{vp}} \tag{7.11}$$

The gains obtained after a single point optimization can provide a good estimation for the potential gain fractions in Eq. (7.11), but a rough estimate can be sufficient. In this way, the optimization formulation is fully determined by an automated physical analysis of the problem.

| Mach | Cl | weight |
|---|---|---|
| 0.8 | 0.446 | 0.161 |
| 0.83 | 0.531 | 0.071 |
| 0.86 | 0.5 | 0.0219 |
| 0.6 | 0.36 | 0.313 |
| 0.7 | 0.39 | 0.278 |
| 0.75 | 0.48 | 0.155 |

Table 7.3: XRF-1 weights for the multipoint optimization

## 7.7 Multipoint optimization strategy

### 7.7.1 On the effect of the gradient noise in multipoint optimization

In a first attempt, multipoint optimizations of the XRF-1 case were run with one angle of attack per operating condition, and one associated lift constraint. The CFD calculations were then performed at a fixed angle of attack. A typical optimization history is displayed in Fig. 7.19, where the pressure drag is plotted as a percentage of its initial value on the baseline. We notice that the SLSQP (Sequential Least Squares Quadratic Programming) [69] algorithm converges to the initial configuration, while in 2D, similar optimizations do improve the performance.

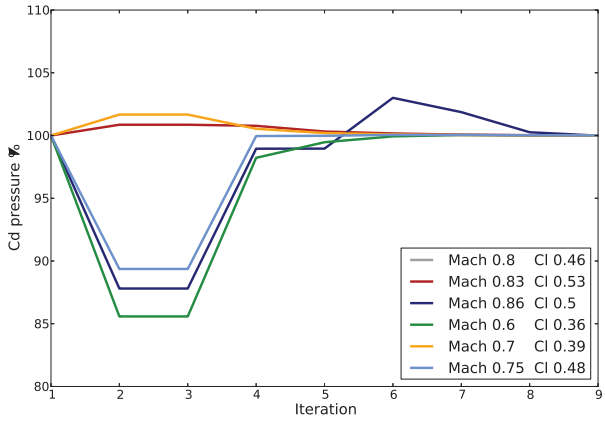



Figure 7.19: An attempt of multipoint optimization with 6 operating conditions and explicit lift constraints handled by the SLSQP optimizer [69]

Since the algorithm failed to reach and maintain the lift constraints, we intuited that the error on the gradients in 3D were too high, or somewhat unfavorable. The target lift approach, in the same conditions, converges, see next section. Therefore, a noise propagation study is proposed in the following.

### 7.7.2 Analysis of the error propagation on the Lagrangian gradients

In a target lift formulation, the Lagrangian gradient provided to the optimizer is given in Eq. (7.4): $0 = \frac{dCd(\chi,AoA)}{dAoA} + \lambda_{TCL}\frac{dCl(\chi,AoA)}{dAoA}$. A sensitivity to the noise of this Lagrangian can be run to compare the formulations. We note $\Delta x$ a noise perturbation on the variable $x$.

$$\frac{\Delta dL(Cd, Cl - Cl_0)(\chi)}{d\chi} = \frac{\Delta dCd(\chi, AoA)}{d\chi} + \Delta\lambda_{TCL}\frac{dCl(\chi, AoA)}{d\chi} + \lambda_{TCL}\Delta\frac{dCl(\chi, AoA)}{d\chi}, \quad (7.12)$$

where $\Delta\lambda_{TCL}$ is the perturbation of the Lagrange multiplier of Eq. (7.3), we obtain

$$\Delta\lambda_{TCL} = \Delta\frac{\frac{dCd(\chi,AoA)}{dAoA}}{\frac{dCl(\chi,AoA)}{dAoA}}. \quad (7.13)$$

By derivation of Eq. (7.13) we obtain, at first order

$$\Delta\lambda_{TCL} = \frac{\Delta\frac{dCd(\chi,AoA)}{dAoA}}{\frac{dCl(\chi,AoA)}{dAoA}} - \frac{\frac{dCd(\chi,AoA)}{dAoA}\Delta\frac{dCl(\chi,AoA)}{dAoA}}{\frac{dCl(\chi,AoA)}{dAoA}^2}. \quad (7.14)$$

We norm Eq. (7.14),

$$||\Delta\lambda_{TCL}|| \leq \left\|\frac{\Delta\frac{dCd(\chi,AoA)}{dAoA}}{\frac{dCl(\chi,AoA)}{dAoA}}\right\| + \left\|\frac{\frac{dCd(\chi,AoA)}{dAoA}\Delta\frac{dCl(\chi,AoA)}{dAoA}}{\frac{dCl(\chi,AoA)}{dAoA}^2}\right\|, \quad (7.15)$$

and note the relative errors $\Delta_{rel}\frac{dCd(\chi,AoA)}{dAoA} = \left\|\frac{\Delta\frac{dCd(\chi,AoA)}{dAoA}}{\frac{dCd(\chi,AoA)}{dAoA}}\right\|$ and $\Delta_{rel}\frac{dCl(\chi,AoA)}{dAoA} = \left\|\frac{\Delta\frac{dCl(\chi,AoA)}{dAoA}}{\frac{dCl(\chi,AoA)}{dAoA}}\right\|$ to finally obtain an upper bound for the Lagrange multiplier noise propagation in a target lift formulation.

$$||\Delta\lambda_{TCL}|| \leq \left\|\frac{\frac{dCd(\chi,AoA)}{dAoA}}{\frac{dCl(\chi,AoA)}{dAoA}}\right\| \left(\Delta_{rel}\frac{dCd(\chi, AoA)}{dAoA} + \Delta_{rel}\frac{dCl(\chi, AoA)}{dAoA}\right). \quad (7.16)$$

And the relative error $\Delta_{rel}\lambda_{TCL} = \frac{\Delta\lambda_{TCL}}{\lambda_{TCL}}$:

$$\Delta_{rel}\lambda_{TCL} \leq \Delta_{rel}\frac{dCd(\chi, AoA)}{dAoA} + \Delta_{rel}\frac{dCl(\chi, AoA)}{dAoA}. \quad (7.17)$$

In the case of explicit constraints given to the optimizer, with one angle of attack $AoA_i$ per operating conditions as design variable, the Lagrange multiplier at each iteration is given by Eq. (7.21), where at each operating condition we note the vector of design variables $\chi_i = (\chi_1, \ldots, \chi_n, AoA_i)$.

$$0 = \nabla_{\chi_i}Cd(\chi_i) + \lambda_{CST}\nabla_{\chi}Cd(\chi_i) \quad (7.18)$$

Since the Lagrangian equation is valid at the optimum only, at each optimization step, we take $\lambda_{CST} = argmin_{\lambda}\|\nabla_{\chi_i}Cd(\chi_i) + \lambda\nabla_{\chi}Cl(\chi_i)\|$, by expansion of the norm we have

$$\|\nabla_{\chi_i}Cd(\chi_i) + \lambda\nabla_{\chi}Cd(\chi_i)\| = \|\nabla_{\chi_i}Cd(\chi_i)\|^2 + \lambda^2\|\nabla_{\chi_i}Cl(\chi_i)\|^2 + 2\lambda\langle\nabla_{\chi_i}Cd(\chi_i), \nabla_{\chi}Cl(\chi_i)\rangle \quad (7.19)$$

By derivation with respect to $\lambda$, it follows that

$$0 = 2\lambda_{CST}\|\nabla_{\chi_i}Cl(\chi_i)\|^2 + 2\langle\nabla_{\chi_i}Cd(\chi_i), \nabla_{\chi_i}Cl(\chi_i)\rangle, \quad (7.20)$$

and finally

$$\lambda_{CST} = -\frac{\langle \nabla_{\chi_i} Cd(\chi_i), \nabla_{\chi_i} Cl(\chi_i) \rangle}{\|\nabla_{\chi_i} Cl(\chi_i)\|^2}. \tag{7.21}$$

Similarly to the target lift approach, we calculate the sensitivity to the noise of Eq. (7.21), and obtain by derivation, at first order

$$\begin{aligned} \Delta\lambda_{CST} = &- \frac{\Delta\left(\langle \nabla_{\chi_i} Cd(\chi_i), \nabla_{\chi_i} Cl(\chi_i) \rangle\right)}{\|\nabla_{\chi_i} Cl(\chi_i)\|^2} \\ &+ 2\frac{\langle \nabla_{\chi_i} Cl(\chi_i), \Delta\nabla_{\chi_i} Cl(\chi_i) \rangle \langle \nabla_{\chi_i} Cd(\chi_i), \nabla_{\chi_i} Cl(\chi_i) \rangle}{\|\nabla_{\chi_i} Cl(\chi_i)\|^4}. \end{aligned} \tag{7.22}$$

With Eq. (7.21), we have $\|\nabla_{\chi_i} Cl(\chi_i)\|^2 = -\frac{\langle \nabla_{\chi_i} Cd(\chi_i), \nabla_{\chi_i} Cl(\chi_i) \rangle}{\lambda_{CST}}$, and then

$$\begin{aligned} \Delta\lambda_{CST} = &- \frac{\Delta\left(\langle \nabla_{\chi_i} Cd(\chi_i), \nabla_{\chi_i} Cl(\chi_i) \rangle\right)}{\langle \nabla_{\chi_i} Cd(\chi_i), \nabla_{\chi_i} Cl(\chi_i) \rangle} \lambda_{CST} \\ &+ 2\frac{\langle \nabla_{\chi_i} Cl(\chi_i), \Delta\nabla_{\chi_i} Cl(\chi_i) \rangle}{\|\nabla_{\chi_i} Cl(\chi_i)\|^2} \lambda_{CST}. \end{aligned} \tag{7.23}$$

which can be re-arranged to compute the relative error propagation,

$$\Delta_{rel}\lambda_{CST} \leq \Delta_{rel}\langle \nabla_{\chi_i} Cd(\chi_i), \nabla_{\chi_i} Cl(\chi_i) \rangle + 2\Delta_{rel}\|\nabla_{\chi_i} Cl(\chi_i)\|. \tag{7.24}$$

Eq. (7.17) shows that in the target lift approach, the relative error on the Lagrangian is the sum of the relative errors of drag and lift derivatives with respect to the angle of attack only, while Eq. (7.24) shows a much more complex expression involving the noise of the derivatives with respect to all the design variables in the case where explicit constraints are used.

The errors on the gradients are evaluated by second order finite differences in the aim of comparing the two approaches. Finite differences are expensive, so the 2D RAE2822 case is used, at Mach 0.74 and Cl 0.782 Re 6.5M. The average errors on the gradient are of 0.36% for the lift, and 1.9% for the drag. Numerical evaluations of Eq. (7.17) and Eq. (7.24) gave $\Delta_{rel}\lambda_{TCL} = 15.2$ and $\Delta_{rel}\lambda_{CST} = 43.9$. It means that the Lagrangian of target lift approach amplifies the noise on the gradients 2.9 times less than the lagrangian of an explicit constraints formulation. In this 2D case, there are only 16 design variables, including the angle of attack, but on the XRF-1 case, the parametrizations have between 100 and 300 design variables. It is interesting to note that $\Delta_{rel}\lambda_{TCL}$ does not depend on the design variables number, while $\Delta_{rel}\lambda_{CST}$ does. It is possible to extrapolate the gradient errors estimation from 2D to the 3D case by copying the gradients to reach the targeted number of design variables, since the wing parametrization uses copies of airfoil parametrization along wingspan. Of course, this is not accurate to estimate the XRF-1 true gradient error since the adjoint convergence is not the same and the flows are different, but we aim here at obtaining tendencies for the variations of $\Delta_{rel}\lambda_{CST}$ with the number of design variables. For 96 design variables, we obtain $\Delta_{rel}\lambda_{CST} = 236.3$, and $\Delta_{rel}\lambda_{CST} = 736.$ for 304 design variables.

The SLSQP algorithm used to solve the drag minimization under explicit lift constraints does not directly minimize the Lagrangian of the optimization problem. Quadratic Programming minimizes a second order model of the function under feasibility constraints. However, the second order terms are modeled by the BFGS approximation computed using the gradients of the Lagrangian. Similarly, the L-BFGS-B algorithm, used to minimize the drag at constant lift ensured with the target lift approach also uses the BFGS approximation of the Lagrangian of the drag at constant lift. Therefore, the previous comparisons of the gradients noise propagation on the Lagrangians is

an argument that contributes to the explanation of why we obtained better optimization results using target lift than explicit lift constraints.

## 7.8 Multipoint optimization detailed results

A multipoint optimization is run, with the 6 required operating conditions. The target lift approach is employed, and, as in the single point case, succeeds in ensuring a constant lift during the process. In Fig. 7.20, the pressure drag history for each operating condition is displayed. Each condition contributes to the overall drag reduction of the overall mission fuel consumption of Eq. (1.10). As a consequence, the mission fuel burn has been reduced. It also means that the initial design was not on the Pareto front formed by the pressure drag at these 6 operating conditions. The CPU cost of

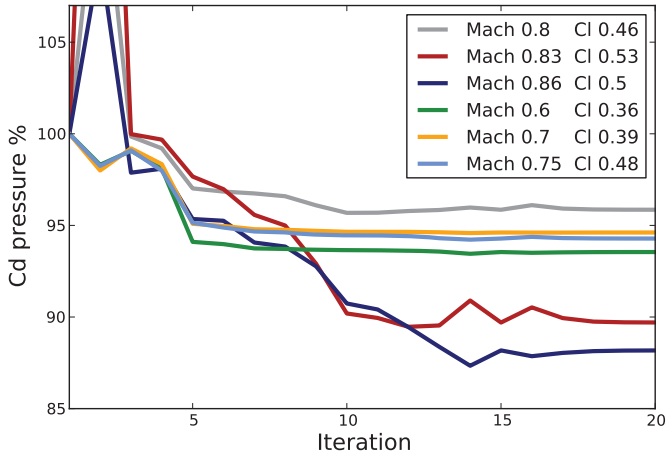
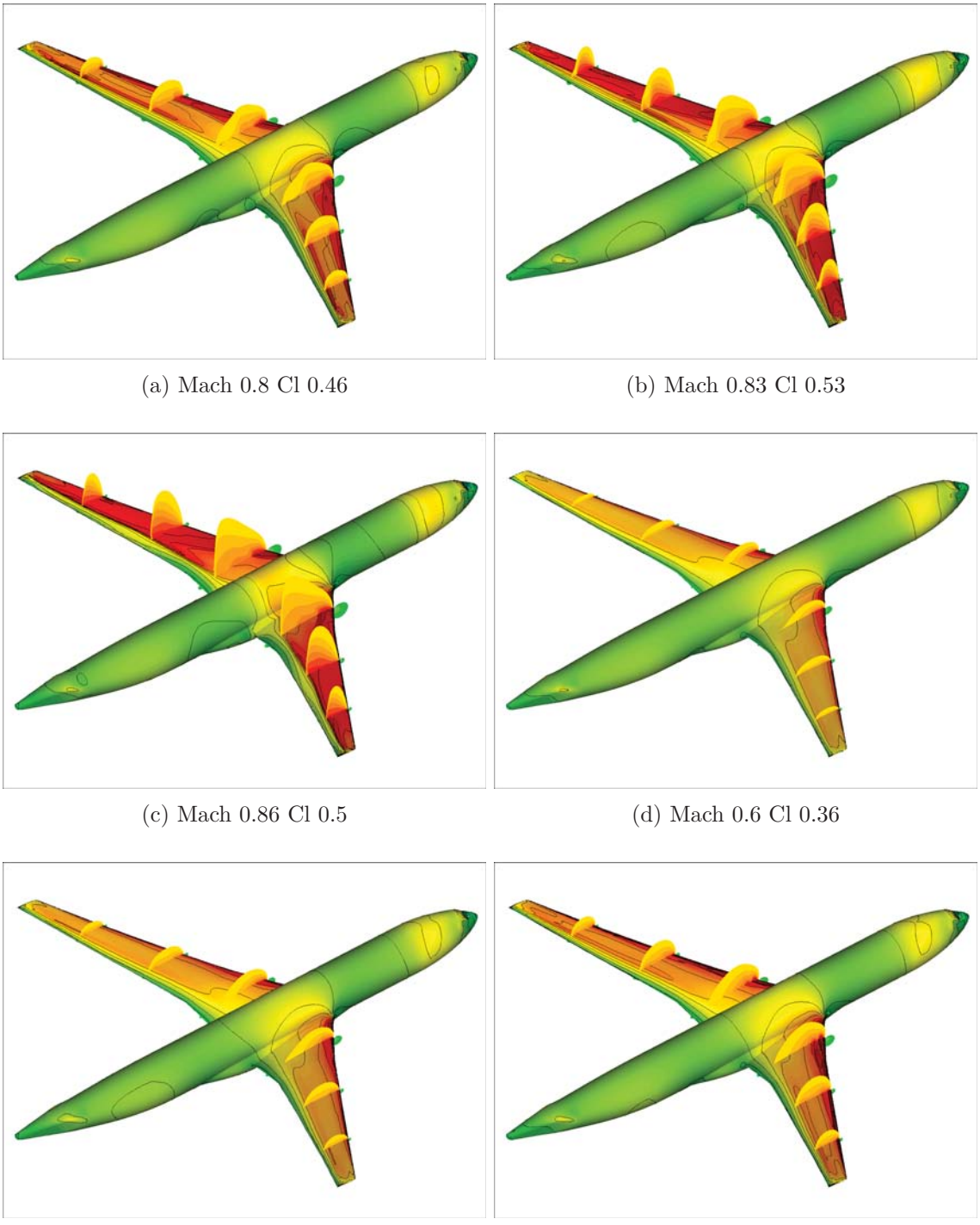



Figure 7.20: 6 points multi-Mach multi-Lift XRF-1 optimization history.

the multipoint optimization was 80 000 hours on Intel(R) Xeon(R) X5670 @ 2.93GHz processors, taking into account the initial sampling for the GSA algorithm and the optimization run itself. The restitution time is 8 days on 400 CPU. It represents a 90% cost cut compared to an approach where "number of design variables plus one" samples are selected in the operating condition ranges [75]. The robust optimization cost 8 times more than the single point one.

In Figures 7.21 to 7.27 the pressure coefficient plots give a global view of the flow at the optimum. As expected, the multipoint optimum being a compromise, the shocks at the same condition are stronger than at the single point optimum, which almost achieves a shock-free flow. However, the wave drag is reduced in a significant way as shown in Fig. 7.28. The other components of the pressure drag are also reduced; the induced drag is decreased by moving outboard the center of lift, as in the single point optimization. The reduction of the shock intensity lowers the boundary layers thicknesses behind the shocks, that reduces the viscous pressure drag. Therefore the viscous pressure drag reduction is a consequence of the wave drag reduction, even if the gradient of this component is not exact in the adjoint solver that assumes frozen turbulence.

In Fig. 7.29, the multi-lift pressure drag polars of the XRF-1 baseline, single-point optimum (Mach 0.83 Cl 0.531), and multipoint optimum are plotted. First, it is noticeable that the highest drag gains are obtained at the highest lift coefficients for both the single and multiple operating conditions optimizations. For instance, 15% of pressure drag are gained at $Cl = 0.55$ for both optimized configurations. Then, for $Cl < 0.5$, the XRF-1 baseline is better-performing than the



(a) Mach 0.8 Cl 0.46

(b) Mach 0.83 Cl 0.53

(c) Mach 0.86 Cl 0.5

(d) Mach 0.6 Cl 0.36

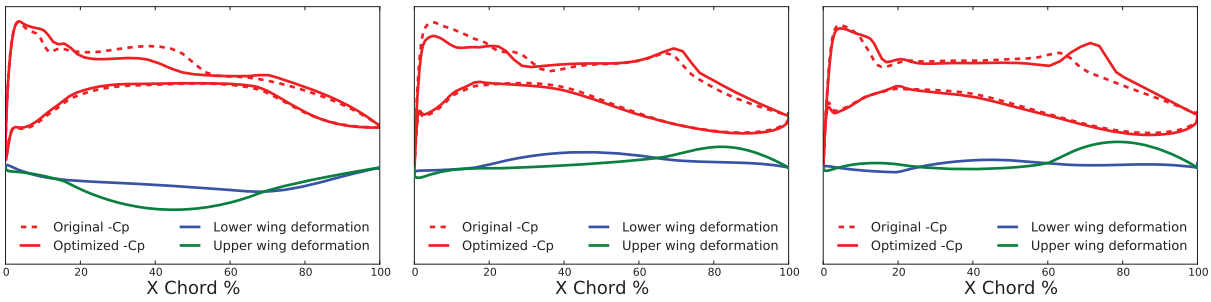
(e) Mach 0.7 Cl 0.39

(f) Mach 0.75 Cl 0.48

Figure 7.21: Pressure coefficient at multiple operating conditions optimum

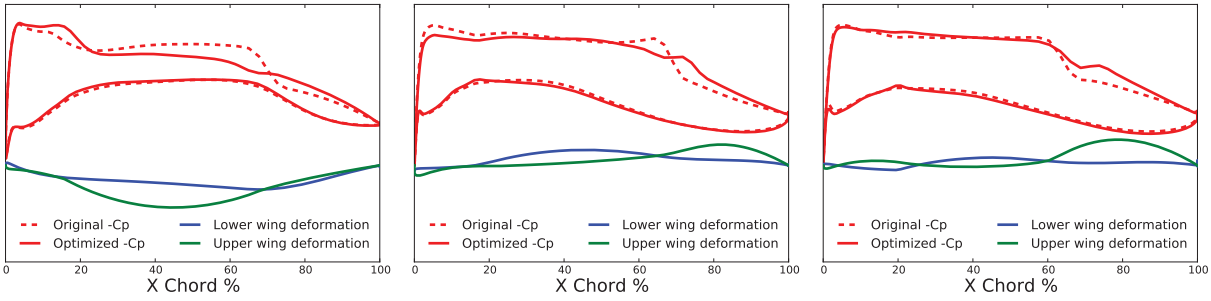
single-point optimum (between 2% and 4%), while the pressure drag of the baseline is significantly lower (about 2%) than the one of the multipoint optimum for $0.4 < Cl < 0.48$. The single point optimum is better performing than the multipoint one only on the range $0.52 < Cl < 0.55$, and for less than 1% of pressure drag in average, at the price of a more than 2% higher drag on the range $0.35 < Cl < 0.52$.

In Fig. 7.30, the multi-Mach pressure drag polars of the XRF-1 baseline, single-point optimum



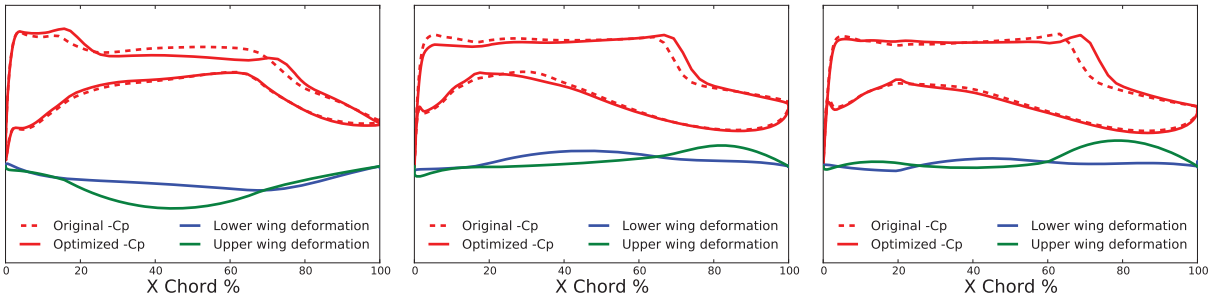

(a) 15 % Span (b) 40 % Span (c) 80 % Span

Figure 7.22: Cp distributions and airfoil deformations at multipoint convergence, Mach 0.8 Cl 0.46




(a) 15 % Span (b) 40 % Span (c) 80 % Span

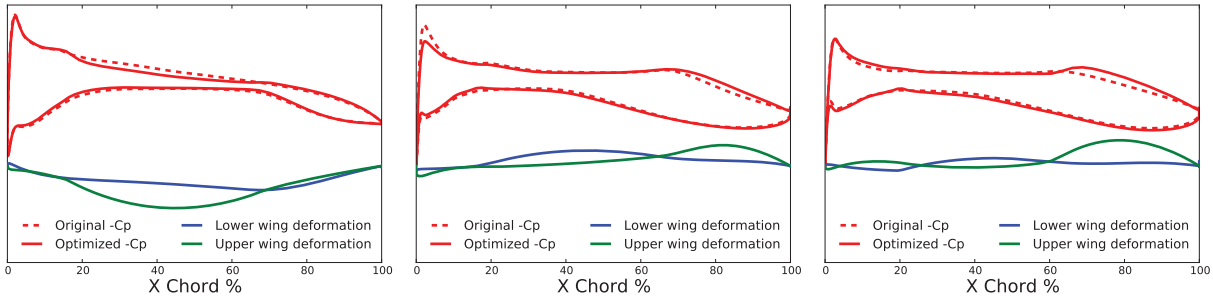
Figure 7.23: Cp distributions and airfoil deformations at multipoint convergence, Mach 0.83 Cl 0.53




(a) 15 % Span (b) 40 % Span (c) 80 % Span

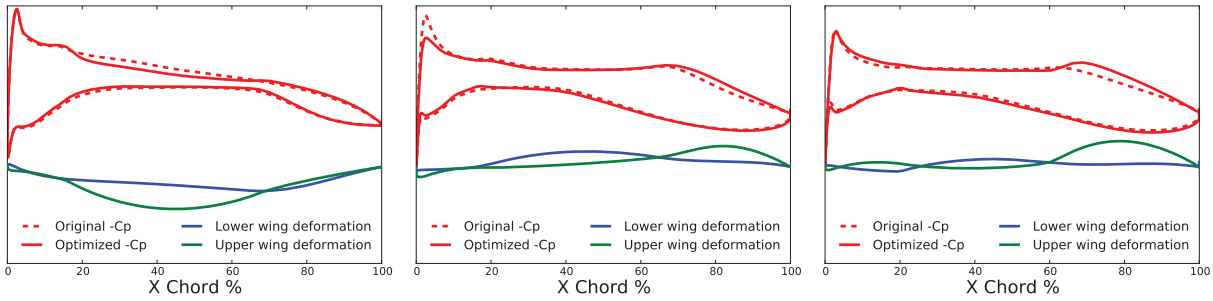
Figure 7.24: Cp distributions and airfoil deformations at multipoint convergence, Mach 0.86 Cl 0.5

and multipoint optimum are plotted. Contrary to the previous case, the two optimums perform better than the baseline on the whole Mach range. Similarly to the multi-Cl comparison, the single point optimum performs better than the multipoint one locally on the range $0.825 < Mach < 0.86$ i.e. close to the design Mach of the single point optimum (Mach 0.83). From these two figures, we conclude that the multipoint optimum is a much better compromise than the single-point optimum and than the XRF-1 baseline on the ranges of operating conditions $0.35 < Cl < 0.55$ and $0.7 < Mach < 0.86$. In average, the single point optimum has a 2% higher pressure drag than the multipoint optimum on the considered operating conditions, showing the interest of the robust optimization in terms of shape performance.



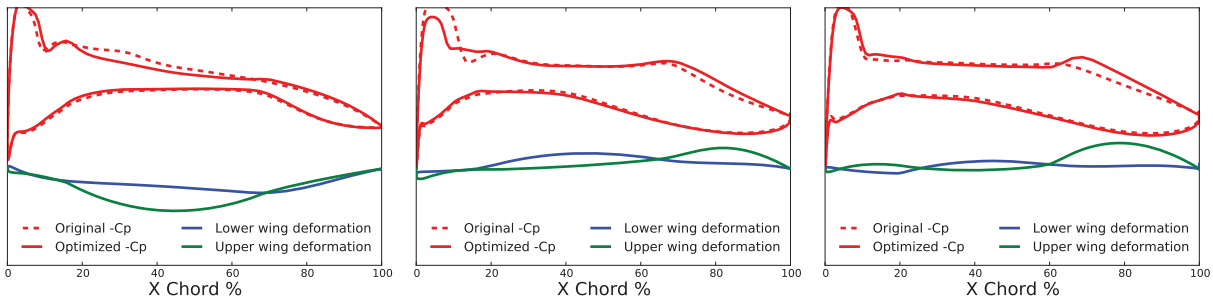

(a) 15 % Span (b) 40 % Span (c) 80 % Span

Figure 7.25: Cp distributions and airfoil deformations at multipoint convergence, Mach 0.6 Cl 0.36




(a) 15 % Span (b) 40 % Span (c) 80 % Span

Figure 7.26: Cp distributions and airfoil deformations at multipoint convergence, Mach 0.7 Cl 0.39




(a) 15 % Span (b) 40 % Span (c) 80 % Span

Figure 7.27: Cp distributions and airfoil deformations at multipoint convergence, Mach 0.75 Cl 0.48

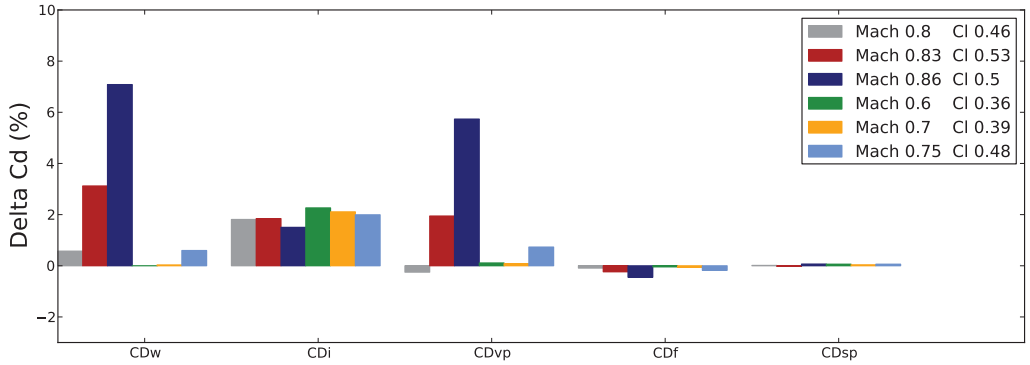



Figure 7.28: Drag gain as percentages of total initial drag on the multipoint case

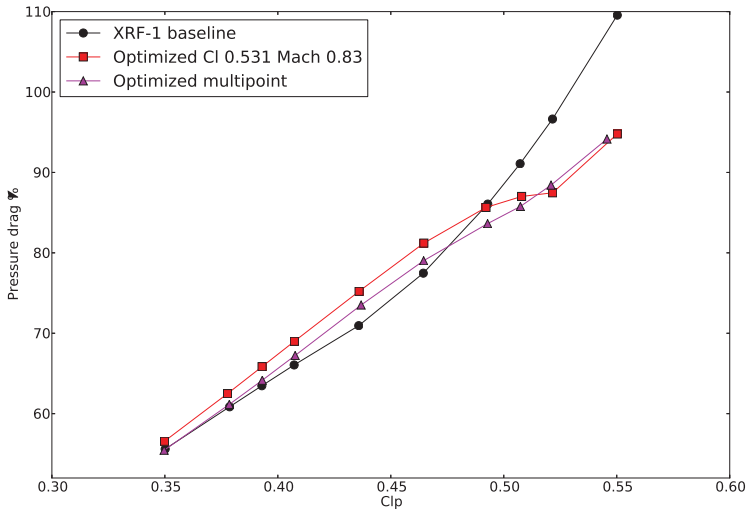



Figure 7.29: Multi-Lift polars of baseline and optimized XRF-1 at Mach 0.83

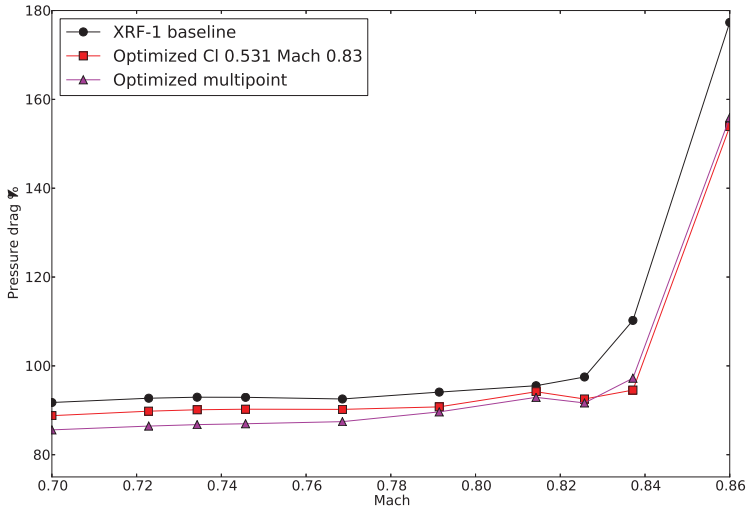



Figure 7.30: Multi-Mach polars of baseline and optimized XRF-1at Cl 0.531

## 7.9 Multipoint aeroelastic gradient-free optimization

In the previous section, the XRF-1 wing is optimized in 6 operating conditions, assuming a rigid structure. In this section, the XRF-1 wing is optimized taking into account for the flexibility of the structure. Consequently, the wing shapes at each operating condition is now different, due to different aerodynamic loads. Since the multipoint optimum of the previous section is robust to angle of attack changes, and that aeroelasticity only creates twist and bending deformations in the model considered here, see Chapter 1, the main impact of aeroelasticity will be a wingspan load repartition modification due to the twist modifications, the bending having a lower order of influence on the drag. The wing sections will remain unchanged. Therefore, we only parametrize the twists of the 10 wing sections, with 4 design variables: $(\chi_0, \chi_1, \chi_2, \chi_3)$, which should be sufficient to control the lift span repartition to balance aeroelasticity effects. Arbitrary linear laws summarized in Tab. 7.4 are used to aggregate the 10 twists in the 4 control parameters. The structural model is adjusted so that the flexible shape, for $(\chi_0, \chi_1, \chi_2, \chi_3) = (0, 0, 0, 0)$ and at $Mach = 0.83$ and $Cl = 0.531$ is the shape of the multipoint rigid optimum. The low number of design variables enables to use a

| Twist | Formula |
|---|---|
| $Twist_1$ | $\chi_0$+0.01$(5\chi_3 + 10^2\chi_2 + 1^3\chi_1)$ |
| $Twist_2$ | $\chi_0$+0.01$(6\chi_3 + 9^2\chi_2 + 2^3\chi_1)$ |
| $Twist_3$ | $\chi_0$+0.01$(7\chi_3 + 8^2\chi_2 + 3^3\chi_1)$ |
| $Twist_4$ | $\chi_0$+0.01$(8\chi_1 + 7^2\chi_3 + 4^3\chi_2)$ |
| $Twist_5$ | $\chi_0$+0.01$(9\chi_1 + 6^2\chi_3 + 5^3\chi_2)$ |
| $Twist_6$ | $\chi_0$+0.01$(10\chi_1 + 5^2\chi_3 + 6^3\chi_2)$ |
| $Twist_7$ | $\chi_0$+0.01$(9\chi_1 + 4^2\chi_3 + 7^3\chi_2)$ |
| $Twist_8$ | $\chi_0$+0.01$(8\chi_2 + 3^2\chi_1 + 8^3\chi_3)$ |
| $Twist_9$ | $\chi_0$+0.01$(7\chi_2 + 2^2\chi_1 + 9^3\chi_3)$ |
| $Twist_{10}$ | $\chi_0$+0.01$(6\chi_2 + 1^2\chi_1 + 10^3\chi_3)$ |

Table 7.4: Parametrization for aeroelastic twist optimization of the XRF-1 model

derivative-free algorithm. In the present case, the COBYLA [108] algorithm relies on successive linear approximations of the function and eventually the constraints. This is why the aggregation of the twist variables is linear: we wish not to introduce a non-linearity in the parametrization that would slow the convergence of the algorithm based on linear approximations. The lift constraints are handled with a target lift approach as in the previous applications, so not by the optimizer. Weights and operating conditions are used to build the aggregated objective function, identically to the rigid multipoint optimization of the previous section, to keep the comparison possible. Results are summarized in Fig. 7.31, which plots the relative variations of the pressure drag at each operating condition during the optimization. Since the starting point of this optimization was the shape obtained after the rigid multipoint optimization, the new aspect of the physical model is only the aeroelastic static coupling. Gains in terms of drag are obtained for five of the six considered operating conditions. Drag is reduced by 0.2% to 0.5% for the low Mach operating conditions, ie. for $0.7 < Mach < 0.8$, and of 1.35% for the operating condition with the highest Mach number (Mach 0.86, Cl 0.5). The only condition affected by a performance degradation is the Mach 0.83, Cl 0.48 operating condition, with an increase of 0.4%. In average on the 6 operating conditions, the drag gain is of 0.38% compared to the rigid optimum.

Multiple conclusions can be drawn from this experiment. First, aeroelasticity introduces a new

degree of freedom in the multipoint optimization. This degree of freedom can be exploited to gain in performance, and the optimal rigid design is not optimal any more when aeroelasticity is taken into account, even if it enabled to design the wing sections. Second, the weights used for the rigid optimization, computed by an estimation of the utopia point with a rigid structure, in the aim of gaining drag at all the operating conditions, are not fully adapted to the problem any more, since the drag has increased at one operating condition. This means that aeroelasticity is changing the trade-offs of gains between the operating conditions. The weight calculations for multipoint optimizations must then take aeroelasticity into account to well balance the gains between operating conditions. Figure 7.32 summarizes the process used for the XRF-1 optimization, leading to a wing

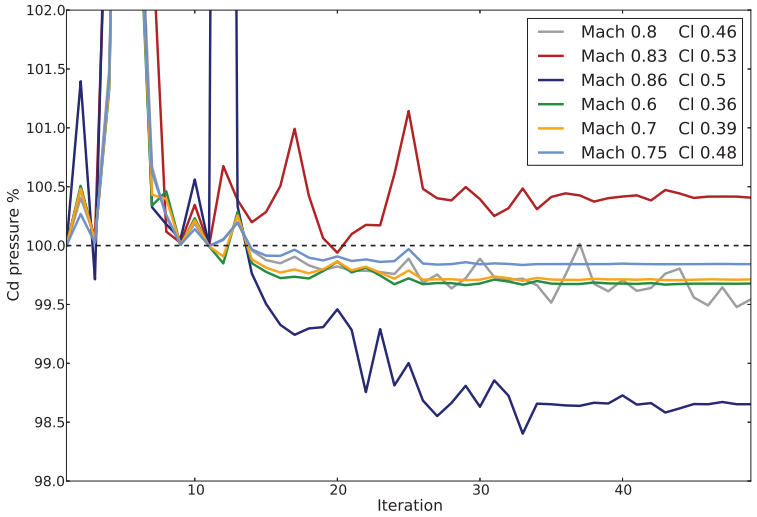



Figure 7.31: Multipoint aeroelastic twist optimization of the XRF-1 model, starting from the multipoint rigid optimum

optimized for a multi-Mach and multi-Lift operating conditions, and accounting for aeroelastic effects. These steps are automated and integrated in the workflow manager, which shows that the methodology can be scaled to an industrial level.

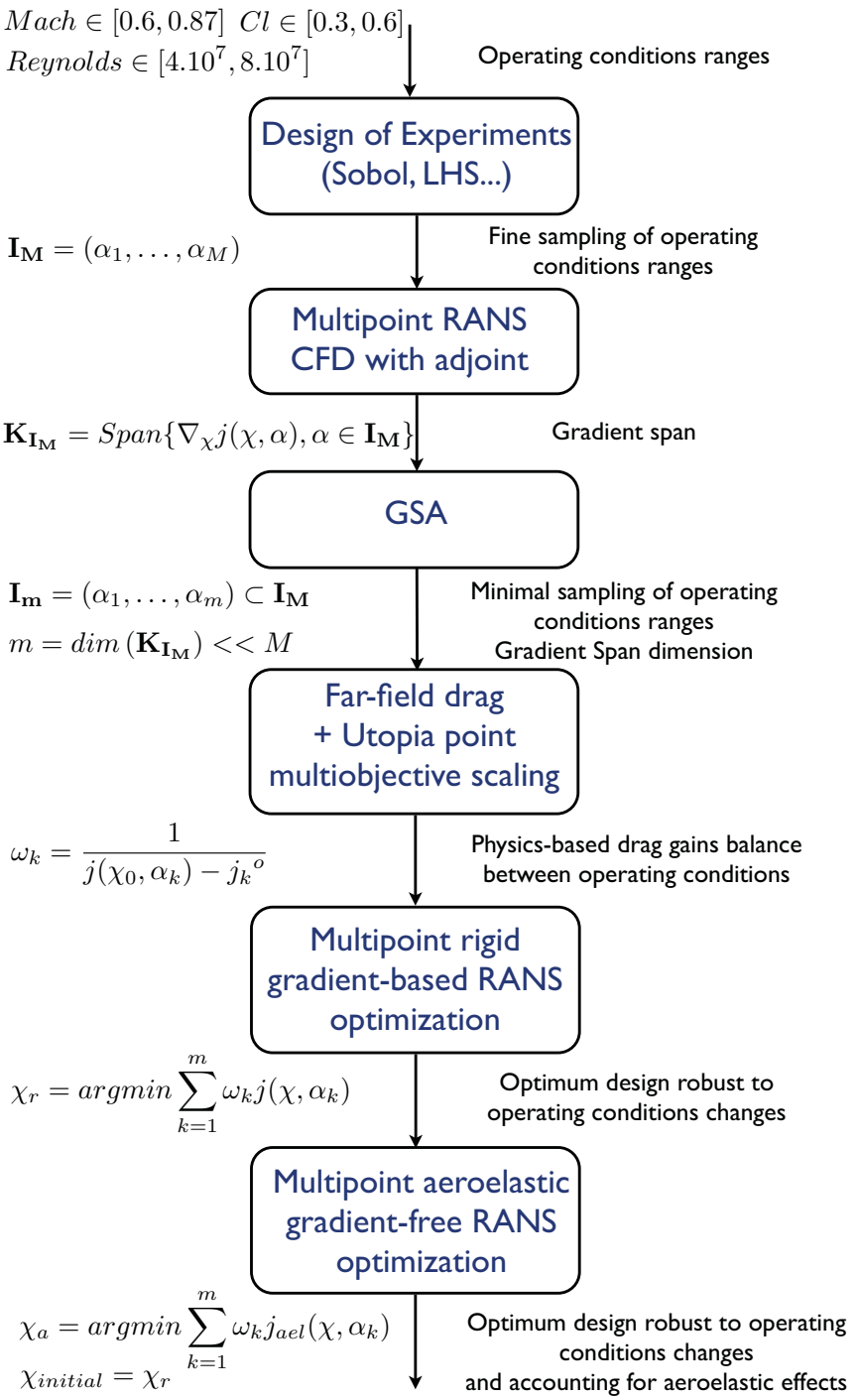



Figure 7.32: Steps of the XRF-1 multipoint aeroelastic optimization

# Conclusions

## Outcomes

Aircraft design for optimal mission performance necessitates the optimization of aerodynamic shapes at all the operating conditions encountered during the flights. Such optimizations are made possible by the new Gradient Span Analysis (GSA) method, which drastically reduces their computational cost. Robust optimization by weighted sum minimization is a simple and efficient way of addressing the design under multiple operating conditions. In this case, the selection of operating conditions to be included in the composite objective function has a major impact on the cost of the resolution, and the quality of the solution. The GSA algorithm is proposed to determine the required conditions and their number, in order to minimize the computational cost of the optimization. Combined with a weighting estimation method based on multi-objectives analysis, it leads to an automated way for setting up multipoint optimization problems. The originality of the method comes from its deterministic point of view, while robustness is often addressed through statistics that may require a high number of calculations.

We have demonstrated that the operating conditions are not just an input for a parametric optimization, but that the choice of the operating conditions is part of the optimization problem setup. The operating conditions incorporated in a weighted sum of objectives must be selected after the analysis of the objective gradients. Ranges of operating conditions being the inputs, the choice of operating conditions is a sampling problem.

To summarize the main theoretical results, the multipoint optimization problem must combine $m$ operating conditions, chosen such that the two gradient spanned vector spaces $\mathbf{K}_{I,\chi}$ and $\mathbf{K}_{I_m,x}$ are equal, otherwise one of the two following situations appears. If $m < dim(\mathbf{K}_{I,\chi})$, then, at the optimum, there exists at least one operating condition at which the objective can be improved without degrading the others, so the performance of the system can be enhanced. If $m > dim(\mathbf{K}_{I,\chi})$, then there exists an equivalent problem with less operating conditions, and modified weights, that gives the same solution, so computational resources are wasted. A counter-intuitive outcome is that when the operating ranges are adequately sampled, i.e. $\mathbf{K}_{I,\chi} = \mathbf{K}_{I_m,x}$, adding an operating condition to the ones used in the optimization problem is equivalent to changing the weights of the aggregate objective function.

The methodology is first validated on a simple wing optimization with a quasi-analytic lifting line

code and its discrete adjoint. Three numerical experiments have been successfully conducted with this model. They show the feasibility of the method, and its interest in terms of computational time savings, compared to state-of-the-art approaches where the number of operating conditions involved in the weighted sum is number of design variables plus one. Then, RAE2822 airfoil optimizations using Reynolds-Averaged Navier-Stokes computations are presented. This academic case validates the single-point optimization chain, and shows the advantages of the GSA method compared to the state-of-the-art in multipoint aerodynamic optimization automated setup. Besides, the limits of the hypothesis stating that the operating conditions generating the gradient span are invariant during the optimization process was tested on this case. When the hypothesis is false, an iterative process made of an optimization followed by a GSA analysis is used. However, the convergence of this algorithm is an open question.

A modern transport aircraft multipoint wing is optimized, using high-fidelity Reynolds-Averaged Navier-Stokes simulations. It confirms the range of applications with an average 2% drag gain compared to the single point design. An interpretation is provided to link the polar control dimension with the non-linear physical phenomena occurring on the aircraft such as shock waves. The successful usage of the GSA algorithm shows that the adjoint sensitivities can not only enhance the gradient-based optimization effectiveness but also enable a new way of improving the multiple operating conditions optimization procedure. The impacts of noise on gradients of the objective function is discussed, both on the GSA algorithm and the robust optimization formulation and results.

Shape parametrization has been shown to play a major role in the generation of robust designs. Non-smooth shapes can perform very well at a given operating condition, and very badly at others. The use of Hicks-Henne bumps for airfoil parametrization led in the past to the false idea that more operating conditions than design variables had to be used for multipoint optimization problems to avoid the drag-creep effect, while what was observed was mainly driven by the parametrization. Using a smooth parametrization is very computationally efficient from that point of view, and certainly the only alternative for 3D cases.

Finally, it is shown that aeroelasticity has to be taken into account during multipoint wing optimization, first, for the fidelity of the simulations to be representative, and second, because it adds additional degrees of freedom to the multipoint optimization, making the shape different at each operating condition, opening the possibility of passive adaption.

## Key contributions

One of the key contributions is a formalization of the aerodynamic shape optimization for mission performance. In addition, a methodology for solving the resulting aerodynamic optimization problem at affordable computational cost on industrial configurations has been proposed. The mathematical theorems on which the methodology relies have been demonstrated. A better understanding of the single-point optimization effect, thanks to an improved mathematical formalization, is also a key academic contribution of this thesis.

Programming contributions to the optimization chain represent an important part of the time spent on the project. The main ones are detailed in the following.

- A multipoint module was programmed, providing a generic way of setting up parametric analyses, for any numerical simulation code. The GSA algorithm being embedded into it, this module automates multipoint optimization problem setup and resolution.

- Generic parametric analyses imply specific requirements for the workflow manager, in charge of sequencing simulation codes. To this aim, the tool was improved with enhanced dynamic data and workflow capabilities for instance, and a higher reliability required to complete the numerous calculations of multipoint optimization.
- A lifting line code with adjoint was created, in order to test multipoint formulations, and validate the optimization modules.
- Similarly, a quasi-analytic structural code for structural sizing and mass estimation, with adjoint, was written with two interns, in the aim of testing MDO formulations and to be coupled with the lifting line code.
- Since high quality and differentiated parametrization is required for robust optimization, an important effort was made on the re-programming of the CAD engine that manages the shapes in the optimization chain. This team work led to a much faster tool using a new smart update system, a more user friendly way of parametrizing shapes, $C^n$ continuity connection between CAD elements for any $n$, and a more efficient differentiation engine.
- Another team work has been the programming of a platform for the integration of optimization algorithm, design of experiments, interpolation methods and their link to the simulation environment and the workflow manager.

## Some lessons learnt

Discrete adjoint is the key enabler to make optimization affordable, however it is also often a blocking point in the software implementation. Frequently, adjoint is not maintained at the same level as the direct code, when a basic hand differentiation approach is used. This leads to errors on the calculated gradients, and as we observed with the frozen turbulence assumption, degrades the optimization algorithms behavior and can even prevent it to converge. The GSA algorithm is also affected by noise, and implies the use of a higher tolerance for the detection of required operating conditions, leading potentially to missed performance gains opportunities. Fortunately, alternatives to hand derivation of codes considerably reduce the programming effort of discrete adjoint. Two main technologies exist to automate the generation of a discrete adjoint state, or reverse mode : automatic differentiation with tools like Tapenade [102], and operator overloading [94]. They enable to compute partial derivatives of softwares, which can then be assembled to compute a total derivative, or create adjoint linear systems. In the latter case, there exists efficient matrix-free linear solver libraries such as Petsc [4] which is used in the aerodynamics community. None of these strategies provides out of the box, memory and computationally efficient derivatives. The code architecture needs to be adapted. Consequently, when the differentiation constraint is taken into account at the conception stage of the software, we noticed that an adequate object-oriented architecture combined with a high level description of the algorithm in Numpy and Scipy [61, 101] for instance, takes in charge most of the differentiation task. The programming of derivatives is made much easier both because the assembly of derivatives by the chain rule, corresponding to composition of functions, is taken in charge by the architecture; and the derivation of algorithms written in Numpy is very similar to the algorithm itself. Such architectures have shown the ability to compute coupled adjoints of for instance 6 main disciplines splitted into 42 components [51], with millions of state variables, 25 000 design variables, in 6 operating conditions.

Making adjoint calculations cheaper, more precise and robust is an appropriate way of reducing the cost of optimizations, especially when the number of constraints is increasing leading to a multiplication of the number of adjoint calculations. There are multiple ways of breaking the

proportionality of the computational cost to the functions of interest in the adjoint approach. First, calculating the gradient of Lagrangians directly instead of functions and constraints separately leads to a single adjoint calculation. When it is not possible, some linear solvers such as block-Generalized Minimum Residual (GMRES) can take advantage of multiple right-hand sides, and solving multiple linear systems at the same time is less costly than solving them separately [107]. Interpretation of adjoint fields should give insights on the way to control the physics of the optimization problem. Although some arguments such as the envelope theorem are provided in the present thesis, there is room for improvement in the field.

## Future work

Efforts and progresses can be made on the main aspects developed in the three parts of the present work.

**On the applications side** At the current status, optimization algorithms are very much separated from the simulation. The only information provided to the optimizers is the functions of interests and their gradient, which is very little information compared to data generated by the simulations. Besides, we noticed that some issues are typically raised by missing information about the models, on the algorithm side. For instance, when different types of design variables, i.e. parameterizing different geometry properties such as curvatures, positions and tangent angles, we observed that the optimizer sometimes fixes the values of one category of variables while they have a strong influence on the objective function, leading to a non-optimal solution. On the other hand there is unexploited information in the simulations. The geometry Hessian is made available by differentiated CAD kernels, and could be used for pre-conditioning algorithms as it represents information on the relative curvatures associated to the different design variables. Similarly, in data assimilation, background error matrix can be used to precondition adjoint-based conjugate gradient algorithms [46]. If one wants to take more advantage of the specificities of a particular problem, because similar ones will be solved thousands of times, such as airfoil optimization, then the optimization strategy has to be dedicated to it, see the No Free Lunch Theorem for optimization [141]. Therefore, optimizers should not come as executables, but libraries of elementary bricks such as line-search algorithms, trust regions management, BFGS approximation, and be easily tuned. Constraints handling in aerodynamic optimization is another typical example where the structure of the problem can be used to simplify its resolution, typically lift and momentum constraints can be satisfied by adjusting dedicated variables such as Angle of Attack and pitch angle.

**On multipoint optimization problems and the GSA theorems**, there are open theoretical questions remaining with potential for new demonstrations, and linked with industrial applications. First, the use of constraints in the optimization process may require extension of the existing theory. With the Lagrangian approach, the reformulation into an unconstrained problem provides, as in the XRF-1 application for instance, a way of directly using the actual GSA approach. In the case of explicit constraints handled by the optimizer, the Lagrangian can be used to adapt the existing theorems. Indeed, the Lagrangian gradient is a linear combination of constraints and objectives gradients; since GSA also relies on linear dependencies between the objective gradients, the approaches are compatible. Second, we have seen that an iterative procedure made of GSA and optimizations may be required, and that there is no proof of convergence for such a process. Third, the weighted sum method being known to have limitations [24], the use of alternatives to solve the multi-objective optimization problem deserves further research. In particular, Theorems demonstrated in the present thesis suppose that a weighted function is being minimized while not all multi-objective optimization formulation rely on aggregation of objectives by weighting.

**On the aircraft mission fuel consumption minimization formulation as an optimization problem**, Multi-Disciplinary Optimization (MDO) is a way of addressing interactions between multiple disciplines. For instance, in wing design, we have seen that flexibility couples the aerodynamics, mass and structural sizing. To address mission-performance optimization in the pure aerodynamic field, we have seen that constraints such as beam thicknesses are added to the optimization to do not increase the wing mass, since no wing box sizing is simulated. Such a constraint leads to sub-optimal designs in the multidisciplinary sense. To avoid this, structural optimization has to be in the loop, and MDO offers multiple formulations to this aim [86]. Besides, the way aero-elasticity is taken into account is a key question for mission-oriented performance optimization. There is clearly a potential for aeroelastic coupled adjoint, which has already been used in the literature since it is the only way to take into account strong coupling sensitivities, and is helpful for efficient MDO. Calculation of the mission performance could also be a discipline in the MDO point of view comparably to aerodynamics and structure.

The methodology presented in this work is based on a one way coupling between the mission and the robust optimization : the mission gives ranges for the operating conditions, that are then incorporated into the optimization. Actually, weights of the multipoint formulation should depend on the mission, since they basically represent the fuel burn rate per unit of drag counts. Besides, the aircraft trajectory optimization is made for a given aircraft design, so the coupling should be two-way. This raises a problem, since the models of the aircraft performance depending typically on Mach, Reynolds, Angle of Attack and aircraft mass properties require numerous CFD calculations to maintain the reduced oder models quality, while we only use a minimal amount of them in the optimization, thanks to GSA. The feasibility in terms of computational cost of a coupled mission and optimization problem is an open question that could be addressed by MDO.

Aircraft multidisciplinary optimization is intrinsically multipoint, for instance the critical loads are multiple, besides the operating conditions driving the aircraft aerodynamic performance are multiple as well as the one driving the engines performance. The adaptation of multipoint formulations to MDO architectures is an open question, with associated potential for both computational cost and aircraft performance gains.

# List of Figures

# Bibliography


[1] Global Market Forecast, Future Journeys 2013. Technical report, Airbus, 2013. URL http://www.airbus.com/company/market/forecast/. *Cited in pages 7, 8, and 147*

[2] John D Anderson. Aircraft performance & design. 1999. *Cited in page 19*

[3] Emir Mahmut Bahsi. *Dynamic Workflow Management for Large Scale Scientific Applications*. PhD thesis, Louisiana State University, 2008. *Cited in page 93*

[4] S Balay, J Brown, K Buschelman, V Eijkhout, W Gropp, D Kaushik, M Knepley, L Curfman McInnes, B Smith, and H Zhang. Petsc users manual revision 3.4. 2013. *Cited in page 143*

[5] Arnaud Barthet. Amélioration de la prévision des coefficients aérodynamiques autour de configurations portantes par méthode adjointe. 2007. *Cited in page 100*

[6] Richard Bellman. *Adaptive control processes: a guided tour*, volume 4. Princeton university press Princeton, 1961. *Cited in page 34*

[7] Hans-Georg Beyer and Bernhard Sendhoff. Robust optimization – A comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33–34):3190–3218, 2007. ISSN 0045-7825. doi: 10.1016/j.cma.2007.03.003. URL http://www.sciencedirect.com/science/article/pii/S0045782507001259. *Cited in page 42*

[8] Léon Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17:9, 1998. *Cited in page 55*

[9] Léon Bottou and Olivier Bousquet. The tradeoffs of large-scale learning. *Optimization for Machine Learning*, page 351, 2011. *Cited in page 55*

[10] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. *Cited in page 46*

[11] Thierry Braconnier, Marc Ferrier, Jean-Christophe Jouhaud, Marc Montagnac, and Pierre Sagaut. Towards an adaptive POD/SVD surrogate model for aeronautic design. *Computers & Fluids*, 40:195–209, January 2011. doi: 10.1016/j.compfluid.2010.09.002. *Cited in page 54*

[12] Howard P. Buckley and David W. Zingg. Approach to aerodynamic design through numerical optimization. *AIAA Journal*, 51(8):1972–1981, 2014/06/25 2013. doi: 10.2514/1.J052268. URL http://dx.doi.org/10.2514/1.J052268. *Cited in pages 40 and 48*

[13] Howard P. Buckley, Beckett Y. Zhou, and David W. Zingg. Airfoil Optimization Using Practical Aerodynamic Design Requirements. *Journal of Aircraft*, 47(5):1707–1719, September-October 2010. doi: 10.2514/1.C000256. *Cited in pages 32, 39, and 55*

[14] Howard P Buckley, Beckett Y Zhou, and David W Zingg. Airfoil optimization using practical aerodynamic design requirements. *Journal of Aircraft*, 47(5):1707–1719, 2010. *Cited in page 102*

[15] L. Cambier, M. Gazaix, S. Heib, S. Plot, M. Poinot, J.-P. Veuillot, J.-F. Boussuge, and M. Montagnac. CFD Platforms and Coupling : An Overview of the Multi-Purpose elsA Flow Solver. *Aerospace Lab*, 2, March 2011. *Cited in page 90*

[16] L. Cambier, S. Heib, and S. Plot. The Onera ElsA CFD Software: Input from Research and Feedback from Industry. *Mechanics & Industry*, 14(03):159–174, January 2013. doi: 10.1051/meca/2013056. *Cited in page 90*

[17] Richard L. Campbell. Efficient Viscous Design of Realistic Aircraft Configurations. In *AIAA Paper*, number 98-2539, 1998. *Cited in pages 39 and 55*

[18] G Carrier. Single and multi-point aerodynamic optimizations of a supersonic transport aircraft wing using optimization strategies involving adjoint method and genetic algorithm. In *Proceedings of ERCOFTAC Workshop" Design optimization: methods and applications", Las Palmas*, 2006. *Cited in page 35*

[19] Oleg Chernukhin and David W Zingg. Multimodality and global optimization in aerodynamic design. *AIAA Journal*, 51(6):1342–1354, 2013. *Cited in page 36*

[20] AC Chiang and Kevin Wainwright. Fundamental methods of mathematical economics. *McGraw-Hill, New York*, 2005. *Cited in page 99*

[21] Haecheon Choi and Parviz Moin. Grid-point requirements for large eddy simulation: Chapman's estimates revisited. *Physics of Fluids*, 24:011702, 2012. *Cited in page 89*

[22] Hyoung-Seog Chung and Juan J Alonso. Using gradients to construct cokriging approximation models for high-dimensional design optimization problems. *AIAA paper*, 317:14–17, 2002. *Cited in page 31*

[23] A.R. Collar. The expanding domain of aeroelasticity. *J. Royal Aero Society*, 1:613–634, 1946. *Cited in page 24*

[24] I. Das and J. E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural and Multidisciplinary Optimization*, 14:63–69, 1997. ISSN 1615-147X. doi: 10.1007/BF01197559. URL http://dx.doi.org/10.1007/BF01197559. *Cited in pages 30, 35, 46, and 144*

[25] Indraneel Das and J. E. Dennis. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM J. on Optimization*, 8(3):631–657, March 1998. ISSN 1052-6234. doi: 10.1137/S1052623496307510. URL http://dx.doi.org/10.1137/S1052623496307510. *Cited in pages 45 and 46*

[26] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002. *Cited in page 30*

[27] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathématiques*, 350(56):313–318, 2012. ISSN 1631-073X. doi: 10.1016/j.crma.2012.03.014. URL http://www.sciencedirect.com/science/article/pii/S1631073X12000738. *Cited in pages 45, 55, 61, 66, and 75*

[28] Jean-Antoine Désidéri. Mgda variants for multi-objective optimization. Technical Report 8068, Rapport de Recherche INRIA, 2012. URL http://hal.archives-ouvertes.fr/docs/00/73/28/81/PDF/RR-8068.pdf. *Cited in page 66*

[29] D. Destarac. Far-field/near field drag balance and applications of drag extraction in cfd. *VKI Lecture Series*, pages 3–7, 2003. *Cited in pages 108 and 128*

[30] Mark Drela. *Pros and cons of airfoil optimization*, chapter Frontiers of Computational Fluid Dynamics, pages 363–381. World Scientific, Singapore, 1998. *Cited in pages 8, 37, 38, 39, 47, 102, and 148*

[31] R Duvigneau. Robust design of a transonic wing with uncertain mach number. *EUROGEN 2007 Evolutionary Methods for Design, Optimization and Control*, 2007. *Cited in page 30*

[32] R Duvigneau and M Visonneau. Hybrid genetic algorithms and artificial neural networks for complex design optimization in cfd. *International journal for numerical methods in fluids*, 44 (11):1257–1278, 2004. *Cited in page 31*

[33] Richard P Dwight and Joël Brezillon. Effect of approximations of the discrete adjoint on gradient-based optimization. *AIAA Journal*, 44(12):3022–3031, 2006. *Cited in pages 106, 108, and 119*

[34] Michael S Eldred, Anthony A Giunta, Bart G van Bloemen Waanders, Steven F Wojtkiewicz, William E Hart, and Mario P Alleva. *DAKOTA, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 4.1 reference manual*. Sandia National Laboratories Albuquerque, NM, 2007. *Cited in page 93*

[35] David Eller and Sebastian Heinze. Approach to induced drag reduction with experimental evaluation. *Journal of Aircraft*, 42(6):1478–1485, 2005. *Cited in pages 25 and 26*

[36] J Elliott and J Peraire. Constrained, multipoint shape optimisation for complex 3d configurations. *Aeronautical Journal*, 102(1017):365–376, 1998. *Cited in page 39*

[37] Tohid Erfani and Sergei V. Utyuzhnikov. Directed search domain: a method for even generation of the Pareto frontier in multiobjective optimization. *Engineering Optimization*, 43(5): 467–484, 2011. doi: 10.1080/0305215X.2010.497185. URL http://www.tandfonline.com/doi/abs/10.1080/0305215X.2010.497185. *Cited in pages 45, 55, and 75*

[38] Xinlong Feng and Zhinan Zhang. The rank of a random matrix. *Applied mathematics and computation*, 185(1):689–694, 2007. *Cited in page 128*

[39] James D Foley, Andries Van Dam, Steven K Feiner, John F Hughes, and Richard L Phillips. *Introduction to computer graphics*, volume 55. Addison-Wesley Reading, 1994. *Cited in page 85*

[40] Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005. *Cited in page 93*

[41] David E Goldberg and John H Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988. *Cited in page 30*

[42] Nicholas I. M. Gould, Dominique Orban, and Philippe L. Toint. Cuter and sifdec: A constrained and unconstrained testing environment, revisited. *ACM Trans. Math. Softw.*, 29(4):373–394, December 2003. ISSN 0098-3500. doi: 10.1145/962437.962439. URL http://doi.acm.org/10.1145/962437.962439. *Cited in page 93*

[43] Serge Gratton, Philippe L Toint, and Anke Tröltzsch. An active-set trust-region method for derivative-free nonlinear bound-constrained optimization. *Optimization Methods and Software*, 26(4-5):873–894, 2011. *Cited in page 31*

[44] Serge Gratton, Philippe L Toint, and Anke Tröltzsch. How much gradient noise does a gradient-based linesearch method tolerate? Technical report, CERFACS, 2011. *Cited in pages 106 and 119*

[45] Justin Gray, Kenneth T Moore, and Bret A Naylor. Openmdao: an open source framework for multidisciplinary analysis and optimization. In *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, TX, USA*. MIT Press, 2010. *Cited in page 93*

[46] S Gürol, AT Weaver, AM Moore, A Piacentini, HG Arango, and S Gratton. B-preconditioned minimization algorithms for variational data assimilation with the dual formulation. *Quarterly Journal of the Royal Meteorological Society*, 2013. *Cited in page 144*

[47] M. Harbeck and A. Jameson. Exploring the limits of shock-free transonic airfoil design. In *AIAA 43rd aerospace sciences meeting and exhibition*, 2005. *Cited in pages 46 and 106*

[48] Raymond M. Hicks and Garret N. Vanderplaats. Application of Numerical Optimization to the Design of Supercritical Airfoils without Drag Creep. In *Business Aircraft Meeting*, number Paper 770440, Wichita, Kansas, March 29 - April 1 1977. Society of Automotive Engineers. URL http://papers.sae.org/770440. *Cited in pages 37 and 82*

[49] Chen-Hung Huang, Jessica Galuski, and Christina L Bloebaum. Multi-objective pareto concurrent subspace optimization for multidisciplinary design. *AIAA Journal*, 45(8):1894–1906, 2007. *Cited in page 46*

[50] Luc Huyse, Sharon L Padula, R Michael Lewis, and Wu Li. Probabilistic approach to free-form airfoil shape optimization under uncertainty. *AIAA Journal*, 40(9):1764–1772, 2002. *Cited in page 30*

[51] John T Hwang, Dae Young Lee, James W. Cutler, and Joaquim R. R. A. Martins. Large-scale multidisciplinary optimization of a small satellite's design and operation. *Journal of Spacecraft and Rockets*, 2013. (Accepted Nov. 17, 2013). *Cited in page 143*

[52] Laurenceau J. *Surfaces de réponse par krigeage pour l'optimisation de formes aérodynamiques - TH/CFD/08/62*. PhD thesis, Institut National Polytechnique de Toulouse, France, 2008. URL http://www.cerfacs.fr/~cfdbib/repository/TH_CFD_08_62.pdf. phd. *Cited in page 46*

[53] A. Jameson. *Steady State Solutions of the Euler Equations for Transonic Flow by a Multigrid Method*, chapter Advances in Scientific Computing, pages 37–70. Academic press, 1982. *Cited in page 91*

[54] A. Jameson, L. Martinelli, and N. A. Pierce. Optimum Aerodynamic Design Using the Navier-Stokes Equations. *Theoretical and Computational Fluid Dynamics*, 10(1-4):213–237, 1998. doi: 10.1007/s001620050060. *Cited in page 91*

[55] Antony Jameson. Aerodynamic design via control theory. *J. Sci. Comput.*, 3(3):233–260, 1988. *Cited in pages 8, 37, and 82*

[56] Antony Jameson, Kasidit Leoviriyakit, and Sriram Shankaran. Multi-point Aero-Structural Optimization of Wings Including Planform Variations. In *45th Aerospace Sciences Meeting and Exhibit*, number AIAA Paper 2007-764, Reno, Nevada, January 8-11 2007. *Cited in pages 40 and 55*

[57] Shinkyu Jeong, Mitsuhiro Murayama, and Kazuomi Yamamoto. Efficient optimization design method using kriging model. *Journal of Aircraft*, 42(2):413–420, 2005. *Cited in page 102*

[58] Forrester T. Johnson, Edward N Tinoco, and N Jong Yu. Thirty years of development and application of CFD at boeing commercial airplanes, seattle. *Computers & Fluids*, 34(10): 1115–1151, 2005. *Cited in pages 8 and 40*

[59] S Johnson. The nlopt nonlinear-optimization package (2011). *URL http://ab-initio. mit. edu/nlopt. Cited in page 93*

[60] Eric Jones, Travis Oliphant, and Pearu Peterson. Scipy: Open source scientific tools for python. *http://www. scipy. org/*, 2001. *Cited in page 93*

[61] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL http://www.scipy.org/. *Cited in page 143*

[62] William Karush. *Minima of functions of several variables with inequalities as side constraints*. PhD thesis, Masterś thesis, Dept. of Mathematics, Univ. of Chicago, 1939. *Cited in page 98*

[63] Gaetan K. W. Kenway and Joaquim R. R. A. Martins. Multi-point high-fidelity aerostructural optimization of a transport aircraft configuration. *Journal of Aircraft*, 2013. (In Press). *Cited in page 40*

[64] Sangho Kim, Kaveh Hosseini, Kasidit Leoviriyakit, and Antony Jameson. Enhancement of adjoint design methods via optimization of adjoint parameters. In *43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV*, pages 10–13, 2005. *Cited in page 85*

[65] D.D. Kosambi. Statistics in Function Space. *JIMS*, 7:76–88, 1943. *Cited in page 54*

[66] Juhani Koski. Defectiveness of weighting method in multicriterion optimization of structures. *Communications in applied numerical methods*, 1(6):333–337, 1985. *Cited in page 46*

[67] Juhani Koski and Risto Silvennoinen. Norm methods and partial weighting in multicriterion optimization of structures. *International Journal for Numerical Methods in Engineering*, 24 (6):1101–1121, 1987. ISSN 1097-0207. doi: 10.1002/nme.1620240606. URL http://dx.doi.org/10.1002/nme.1620240606. *Cited in pages 108 and 128*

[68] D. D. Kraft. A software package for sequential quadratic programming. Technical Report DFVLR-FB 88-28, DLR German Aerospace Center – Institute for Flight Mechanics, Koln, Germany, 1988. *Cited in page 32*

[69] D. D. Kraft. A software package for sequential quadratic programming. Technical report, DLR German Aerospace Center – Institute for Flight Mechanics, Koln, Germany, 1988. *Cited in pages 73, 129, and 150*

[70] Harold W Kuhn and Albert W Tucker. Nonlinear programming. In *Proceedings of the second Berkeley symposium on mathematical statistics and probability*, volume 5. California, 1951. *Cited in page 98*

[71] J Laurenceau and M Meaux. Comparison of gradient and response surface based optimization frameworks using adjoint method. *AIAA Paper*, 1889:2008, 2008. *Cited in page 34*

[72] J Laurenceau and P Sagaut. Building efficient response surfaces of aerodynamic functions with kriging and cokriging. *AIAA Journal*, 46(2):498–507, 2008. *Cited in page 31*

[73] J Laurenceau, M Meaux, M Montagnac, and P Sagaut. Comparison of gradient-based and gradient-enhanced response-surface-based optimizers. *AIAA Journal*, 48(5):981–994, 2010. *Cited in pages 31 and 102*

[74] David Levy, Kelly Laflin, John Vassberg, Edward Tinoco, Mortaza Mani, Ben Rider, Olaf Brodersen, Simone Crippa, Christopher Rumsey, Richard Wahls, Joe Morrison, Dimitri Mavriplis, and Mitsuhiro Murayama. Summary of data from the fifth aiaa cfd drag prediction workshop. American Institute of Aeronautics and Astronautics, 2013/10/08 2013. *Cited in page 89*

[75] Wu Li, Luc Huyse, and Sharon Padula. Robust Airfoil Optimization to Achieve Consistent Drag Reduction Over a Mach Range. *Structural and Multidisciplinary Optimization*, 24(1): 38–50, 2002. *Cited in pages 31, 37, 40, 47, 56, 62, and 132*

[76] Rhea P. Liem, Gaetan K. W. Kenway, and Joaquim R. R. A. Martins. Multi-point, multi-mission, high-fidelity aerostructural optimization of a long-range aircraft configuration. In *14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Indianapolis, IN, 09/2012 2012. *Cited in pages 8, 22, 40, 41, and 148*

[77] Jacques-Louis Lions and Pierre Lelong. *Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles*, volume 1. Dunod Paris, 1968. *Cited in page 82*

[78] Jacques-Louis Lions and Sanjoy K Mitter. *Optimal control of systems governed by partial differential equations*, volume 1200. Springer Berlin, 1971. *Cited in pages 37 and 82*

[79] Fernando G. Lobo, David E. Goldberg, and Martin Pelikan. Time complexity of genetic algorithms on exponentially scaled problems. In *Proceedings of the genetic and evolutionary computation conference*, pages 151–158. Morgan-Kaufmann, 2000. *Cited in page 30*

[80] Zhoujie Lyu, Gaetan KW Kenway, and JRRA Martins. Rans-based aerodynamic shape optimization investigations of the common research model wing. In *AIAA Science and Technology Forum and Exposition (SciTech), National Harbor, MD*, 2014. *Cited in pages 36 and 148*

[81] Meaux M., Cormery M., and Voizard G. Viscous aerodynamic shape optimization based on the discrete adjoint state for 3D industrial configurations. In *European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS*, 2004. *Cited in page 87*

[82] DOT Users Manual. Vanderplaats research & development. *Inc., Version*, 4, 1995. *Cited in pages 32 and 93*

[83] Nathalie Marco, Jean-Antoine Désidéri, and Stéphane Lanteri. Multi-objective optimization in cfd by genetic algorithms. Technical report. *Cited in page 30*

[84] R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004. *Cited in pages 45, 46, and 108*

[85] R Timothy Marler and Jasbir S Arora. The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization*, 41(6):853–862, 2010. *Cited in pages 44 and 47*

[86] Joaquim R. R. A. Martins and Andrew B. Lambe. Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51:2049–2075, 2013. doi: 10.2514/1.J051895. *Cited in page 145*

[87] Michael D McKay, Richard J Beckman, and William J Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979. *Cited in page 75*

[88] A. Messac, A. Ismail-Yahaya, and C.A. Mattson. The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization*, 25:86–98, 2003. ISSN 1615-147X. doi: 10.1007/s00158-002-0276-1. URL http://dx.doi.org/10.1007/s00158-002-0276-1. *Cited in pages 45, 55, and 75*

[89] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. Oxford Science Publications, 2001. *Cited in page 32*

[90] A. M. Morris, C. B. Allen, and T. C. S. Rendall. Cfd-based optimization of aerofoils using radial basis functions for domain element parameterization and mesh deformation. *International Journal for Numerical Methods in Fluids*, 58(8):827–860, 2008. ISSN 1097-0363. doi: 10.1002/fld.1769. URL http://dx.doi.org/10.1002/fld.1769. *Cited in page 102*

[91] Daniel Mueller-Gritschneder, Helmut E. Graeb, and Ulf Schlichtmann. A Successive Approach to Compute the Bounded Pareto Front of Practical Multiobjective Optimization Problems. *SIAM Journal on Optimization*, 20(2):915–934, 2009. *Cited in pages 45, 55, and 75*

[92] John M Mulvey, Robert J Vanderbei, and Stavros A Zenios. Robust optimization of large-scale systems. *Operations research*, 43(2):264–281, 1995. *Cited in page 43*

[93] D. My-Ha, K.M. Lim, B.C. Khoo, and K. Willcox. Real-time optimization using proper orthogonal decomposition: Free surface shape prediction due to underwater bubble dynamics. *Computers & Fluids*, 36(3):499–512, 2007. ISSN 0045-7930. doi: 10.1016/j.compfluid.2006.01.016. *Cited in page 54*

[94] Uwe Naumann. *The Art of Differentiating Computer Programs*, volume 24. SIAM, 2012. *Cited in page 143*

[95] M. Nemec, M. J Aftosmis, and T. H. Pulliam. Cad-based aerodynamic design of complex configurations using a cartesian method. *AIAA Paper*, 113, 2004. *Cited in page 85*

[96] Marian Nemec and David W Zingg. Newton-krylov algorithm for aerodynamic design using the navier-stokes equations. *AIAA Journal*, 40(6):1146–1154, 2002. *Cited in page 96*

[97] Marian Nemec, David W Zingg, and Thomas H Pulliam. Multipoint and multi-objective aerodynamic shape optimization. *AIAA Journal*, 42(6):1057–1065, 2004. *Cited in pages 35 and 87*

[98] M Nguyen-Dinh. *Qualification des simulations numériques par adaptation anisotropique de maillages*. PhD thesis, Université de Nice-Sophia Antipolis, 2004. *Cited in page 118*

[99] E. J. Nielsen and W. K. Anderson. Aerodynamic Design Optimization on Unstructured Meshes Using the Navier-Stokes Equations. *AIAA Journal*, 37(11):1411–1419, 1999. doi: 10.2514/2.640. *Cited in page 91*

[100] Emma Nygren, Kjell Aleklett, and Mikael Höök. Aviation fuel and future oil production scenarios. *Energy Policy*, 37(10):4003–4010, 2009. *Cited in page 7*

[101] Travis E. Oliphant. Python for scientific computing. *Computing in Science & Engineering*, 9(3):10–20, 2007. doi: http://dx.doi.org/10.1109/MCSE.2007.58. URL http://scitation.aip.org/content/aip/journal/cise/9/3/10.1109/MCSE.2007.58. *Cited in page 143*

[102] Valérie Pascual and Laurent Hascoët. Extension of tapenade toward fortran 95. In *Automatic Differentiation: Applications, Theory, and Implementations*, pages 171–179. Springer, 2006. *Cited in page 143*

[103] Sergey Peigin and Boris Epstein. Robust optimization of 2d airfoils driven by full navier-stokes computations. *Computers and Fluids*, 33:1175–1200, 2004. *Cited in page 31*

[104] Ruben E Perez, Peter W Jansen, and Joaquim RRA Martins. pyopt: a python-based object-oriented framework for nonlinear constrained optimization. *Structural and Multidisciplinary Optimization*, 45(1):101–118, 2012. *Cited in page 93*

[105] Jacques Peter, Maxime Nguyen-Dinh, and Pierre Trontin. Goal oriented mesh adaptation using total derivative of aerodynamic functions with respect to mesh coordinates âĂŞ with applications to euler flows. *Computers & Fluids*, 66(0):194 – 214, 2012. ISSN 0045-7930. doi: http://dx.doi.org/10.1016/j.compfluid.2012.06.016. URL http://www.sciencedirect.com/science/article/pii/S0045793012002393. *Cited in page 96*

[106] Le. A Piegl and W. Tiller. *The NURBS book*. Springer, 1997. *Cited in pages 84, 87, and 116*

[107] Xavier Pinel and Marc Montagnac. Block krylov methods to solve adjoint problems in aerodynamic design optimization. *AIAA Journal*, 51(9):2183–2191, 2013. *Cited in pages 96 and 144*

[108] Michael JD Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, pages 51–67. Springer, 1994. *Cited in page 137*

[109] L. Prandtl. Theory of lifting surfaces. Technical report, NACA, 1920. URL http://naca.larc.nasa.gov/reports/1920/naca-tn-9/. *Cited in pages 25 and 70*

[110] Ludwig Prandtl. Tragflügeltheorie. i. mitteilung. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1918:451–477, 1918. *Cited in page 25*

[111] G. Puigt, M. Gazaix, M. Montagnac, M.-C. Le Pape, M. de la Llave Plata, M. Marmignon, J.-F. Boussuge, and V. Couaillier. Development of a new hybrid compressible solver inside the CFD elsA software. In *20th AIAA Computational Fluid Dynamics Conference*, number AIAA–2011–3379, Honolulu (HI), USA, June 27-30 2011. *Cited in page 90*

[112] Hicks R. M., Murman E. M., and Vanderplaats G. N. An assessment of airfoil design by numerical optimization. Technical Report TMX-3092, NASA Ames Research Center, Moffett Field, California, 1974. *Cited in pages 37 and 82*

[113] J. Reuther, J.J. Alonso, Joaquim R. R. A. Martins, and S. C. Smith. A coupled aero-structural optimization method for complete aircraft configurations. In *Proceedings of the 37th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 1999. AIAA 99-0187. *Cited in page 40*

[114] James J. Reuther, Antony Jameson, Juan J. Alonso, Mark J. Rimlinger, and David Saunders. Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, Part 1. *Journal of Aircraft*, 36(1):51–60, January-February 1999. doi: 10.2514/2.2413. *Cited in pages 39 and 55*

[115] James J. Reuther, Antony Jameson, Juan J. Alonso, Mark J. Rimlinger, and David Saunders. Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, Part 2. *Journal of Aircraft*, 36(1):61–74, January-February 1999. doi: 10.2514/2.2414. *Cited in pages 39 and 55*

[116] P. L. Roe. Approximate Riemann Solvers, Parameter Vectors and Difference Schemes. *Journal of Computational Physics*, 43(2):357–372, October 1981. doi: 10.1016/0021-9991(81)90128-5. *Cited in page 91*

[117] Youcef Saad and Martin H Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986. *Cited in page 96*

[118] Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. Design and analysis of computer experiments. *Statistical science*, 4(4):409–423, 1989. *Cited in page 75*

[119] Jamshid A Samareh. Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. *AIAA Journal*, 39(5):877–884, 2001. *Cited in page 84*

[120] Jamshid A Samareh. Aerodynamic shape optimization based on free-form deformation. *AIAA paper*, 4630:2004, 2004. *Cited in page 96*

[121] S Shahpar. Challenges to overcome for routine usage of automatic optimisation in the propulsion industry. *Aeronautical Journal*, 115(1172):615–625, 2011. *Cited in pages 8, 35, and 82*

[122] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524. ACM, 1968. *Cited in page 88*

[123] Jaroslaw Sobieszczanski-Sobieski and Raphael T Haftka. Multidisciplinary aerospace design optimization: survey of recent developments. *Structural optimization*, 14(1):1–23, 1997. *Cited in page 40*

[124] P. R. Spalart and S. R. Allmaras. A One-Equation Turbulence Transport Model for aerodynamic flows. In *30th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 1992. AIAA Paper 92-0439. *Cited in page 91*

[125] E. Stanewsky. Adaptive wing and flow control technology. *Progress in Aerospace Sciences*, 37(7):583–667, 2001. *Cited in pages 25 and 26*

[126] Ralph E. Steuer. *Multiple criteria optimization: theory, computation, and application*. Krieger Malabar, 1989. *Cited in page 22*

[127] Krasimira Stoilova and Todor Stoilov. Comparison of workflow software products. In *International Conference on Computer Systems and Technologies-CompSysTech*, pages 15–16, 2006. *Cited in page 93*

[128] Per Strømholm. Fermat's methods of maxima and minima and of tangents. a reconstruction. *Archive for History of Exact Sciences*, 5(1):47–69, 1968. ISSN 0003-9519. doi: 10.1007/BF00328112. URL http://dx.doi.org/10.1007/BF00328112. *Cited in page 31*

[129] Marius Swoboda, André Huppertz, Akin Keskin, Dierk Otto, and Dieter Bestle. Multidisciplinary compressor blading design process using automation and multi-objective optimization. In *25th Congress of International Council of the Aeronautical Sciences. Paper number ICAS2006-5.6 S*, 2006. *Cited in page 35*

[130] Knut Sydsaeter and Peter J Hammond. *Essential mathematics for economic analysis*. Pearson Education, 2008. *Cited in pages 47 and 99*

[131] Genichi Taguchi. *Introduction to quality engineering: designing quality into products and processes*. 1986. *Cited in page 42*

[132] Bui T. Thanh, M. Damodaran, and K. Willcox. Proper Orthogonal Decomposition Extensions for Parametric Applications in Transonic Aerodynamics (AIAA Paper 2003-4213). In *Proceedings of the 15th AIAA Computational Fluid Dynamics Conference*, 2003. *Cited in page 54*

[133] G. D. van Albada, B. van Leer, and W. W. Roberts. A Comparative Study of Computational Methods in Cosmic Gas Dynamics. *Astronomy and Astrophysics*, 108:76–84, 1982. *Cited in page 91*

[134] B. van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Journal of Computational Physics*, 32(1):101–136, 1979. doi: 10.1016/0021-9991(79)90145-1. *Cited in page 91*

[135] GN Vanderplaats. A robust feasible directions algorithm for design synthesis. In *Proc., 24th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conf*, 1983. *Cited in page 32*

[136] R.V. Gamkrelidze. L.S. Pontryagin V.G. Boltyanskii. *Towards a theory of optimal processes (Russian)*, volume 110(1). Reports Acad. Sci. USSR, 1956. *Cited in page 82*

[137] Alessandro Vicini and Domenico Quagliarella. Inverse and direct airfoil design using a multiobjective genetic algorithm. *AIAA Journal*, 35(9):1499–1505, 1997. *Cited in page 30*

[138] Li W. and Padula S. Performance trade study for robust airfoil shape optimization. In *21th AIAA Applied Aerodynamics Conference*, 2003. *Cited in pages 30 and 31*

[139] David C Wilcox. *Turbulence modeling for CFD*, volume 2. DCW industries La Canada, 1998. *Cited in page 90*

[140] Jochen Wild, Joël Brézillon, Olivier Amoignon, Jürgen Quest, Frederic Moens, and Domenico Quagliarella. Advanced high-lift design by numerical methods and wind tunnel verification within the european project eurolift ii. In *25th AIAA applied aerodynamics conference*, 2007. *Cited in page 36*

[141] David H Wolpert and William G Macready. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67–82, 1997. *Cited in pages 30 and 144*

[142] S. Yoon and A. Jameson. Lower-Upper Symmetric-Gauss-Seidel Method for the Euler and Navier-Stokes Equations. *AIAA Journal*, 26(9):1025–1026, 1988. doi: 10.2514/3.10007. *Cited in page 91*

[143] D. P. Young, W. P. Huffman, R. G. Melvin, C. L. Hilmes, and F. T. Johnson. Nonlinear elimination in aerodynamic analysis and design optimization. In Lorenz T. Biegler, Matthias Heinkenschloss, Omar Ghattas, Bart Bloemen Waanders, Timothy J. Barth, Michael Griebel,

David E. Keyes, Risto M. Nieminen, Dirk Roose, and Tamar Schlick, editors, *Large-Scale PDE-Constrained Optimization*, volume 30 of *Lecture Notes in Computational Science and Engineering*, pages 17–43. Springer Berlin Heidelberg, 2003. ISBN 978-3-642-55508-4. URL http://dx.doi.org/10.1007/978-3-642-55508-4_2. *Cited in page 40*

[144] L Zadeh. Optimality and non-scalar-valued performance criteria. *Automatic Control, IEEE Transactions on*, 8(1):59–60, 1963. *Cited in page 46*

[145] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. L-bfgs-b - fortran subroutines for large-scale bound constrained optimization. Technical report, ACM Trans. Math. Software, 1994. *Cited in pages 32, 106, and 120*

[146] David Zingg and Samy Elias. Aerodynamic Optimization Under a Range of Operating Conditions. *AIAA Journal*, 44(11):2787–2792, 2006. doi: 10.2514/1.23658. *Cited in pages 39, 48, 55, 102, 107, 113, and 149*

### Aircraft shape optimization for mission performance

An aircraft encounters a wide range of operating conditions during its missions, i.e. flight altitude, Mach number and angle of attack, which consideration at the design phase enhances the system robustness and consequently the overall fleet consumption. Numerical optimization of aerodynamic shapes contributes to aircraft design, and relies on the automation of geometry generation and numerical simulations of the flight physics. Minimization of aerodynamic shapes drag must take into account multiple operating conditions, since optimization at a single operating condition leads to a strong degradation of performance when this operating condition varies. Besides, structural flexibility deforms the wings differently depending on the operating conditions, so has to be simulated during such optimizations.

In the present thesis, the mission fuel consumption minimization is formulated as an optimization problem. The focus is made on the choice of operating conditions to be included in the optimization problem, since they have a major impact on the quality of the results, and the computational cost is proportional to their number. A new theoretical framework is proposed, overcoming and giving new insights on problematic situations revealed by state-of-the-art methods for multipoint optimization problem setup. An algorithm called Gradient Span Analysis is proposed to automate the choice of operating conditions. It is based on a reduction of dimension of the vector space spanned by adjoint gradients obtained at the different operating conditions. Programming contributions to the optimization chain enabled the evaluation of the new method on the optimizations of the academic RAE2822 airfoil, and the XRF-1 wing-body configuration, representative of a modern transport aircraft. While the shapes resulting of single-point optimizations present strong degradations of the performance in off-design conditions, adequately formulated multi-Mach-multi-lift optimizations present much more interesting performance compromises. It is finally shown that fluid-structure interaction adds new degrees of freedom, and has consequences on multiple flight conditions optimizations, opening the perspective of passive shape adaptation.



**Keywords:** Computational Fluid Dynamics, Robust Optimization, Adjoint, Aerodynamics



### Optimisation de forme d'un avion pour sa performance sur une mission

Les avions rencontrent de nombreuses conditions d'opérations au cours de leurs vols, comme le nombre de Mach, l'altitude et l'angle d'attaque. Leur prise en compte durant la conception améliore la robustesse du système et finalement la consommation des flottes d'avions. L'optimisation de formes aérodynamiques contribue à la conception des avions, et repose sur l'automatisation de la génération de géométries ainsi que la simulation numérique de la physique du vol. La minimisation de la trainée des formes aérodynamiques doit prendre en compte de multiples conditions d'opération, étant donné que l'optimisation à une unique condition de vol mène à des formes dont la performance se dégrade fortement quand cette condition de vol est perturbée. De plus, la flexibilité structurelle déforme les ailes différemment selon la condition de vol, et doit donc être simulée lors de telles optimisations.

Dans cette thèse, la minimisation de la consommation de carburant au cours d'une mission est formulée en problème d'optimisation. Une attention particulière est apportée au choix des conditions d'opérations à inclure dans le problème d'optimisation, étant donné que celles-ci ont un impact majeur sur la qualité des résultats obtenus, et que le coût de calcul est proportionnel à leur nombre. Un nouveau cadre théorique est proposé pour adresser cette question, offrant un point de vue original et surmontant des difficultés révélées par les méthodes à l'état-de-l'art en matière de mise en place de problèmes d'optimisation multi-points. Un algorithme appelé Gradient Span Analysis (GSA), est proposé pour automatiser le choix des conditions d'opération. Il est basé sur la réduction de dimension de l'espace vectoriel engendré par les gradients adjoints aux différentes conditions de vol. Des contributions de programmation à la chaine d'optimisation ont permis d'évaluer les méthodes aux optimisations du profil académique RAE2822 et de la configuration voilure-fuselage XRF-1, représentative des avions de transport modernes. Alors que les formes résultant d'optimisation mono-point présentent de fortes dégradation de performance hors du point de conception, les optimisations multi-points adéquatement formulées fournissent de bien meilleurs compromis. Il est finalement montré que les interactions fluide-structure ajoutent de nouveaux degrés de liberté, et ont un impact sur les optimisations en de multiples conditions de vol, ouvrant des perspectives en matière d'adaptation passive de forme.



**Mots clés :** Mécanique des fluides numérique, Optimisation Robuste, Adjoint, Aérodynamique