



HAL
open science

SURFACES DE REPONSE PAR KRIGEAGE POUR L'OPTIMISATION DE FORMES AERODYNAMIQUES.

Julien Laurenceau

► **To cite this version:**

Julien Laurenceau. SURFACES DE REPONSE PAR KRIGEAGE POUR L'OPTIMISATION DE FORMES AERODYNAMIQUES.. Sciences de l'ingénieur [physics]. Institut National Polytechnique de Toulouse - INPT, 2008. Français. NNT: . tel-00339863

HAL Id: tel-00339863

<https://theses.hal.science/tel-00339863>

Submitted on 19 Nov 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

N° d'ordre : 2625

présentée pour obtenir

le titre de

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE TOULOUSE

École doctorale : **Mécanique Énergétique Génie Civil et Procédés**

Spécialité : **Dynamique des Fluides**

par

Julien LAURENCEAU

Surfaces de réponse par krigeage pour l'optimisation de formes aérodynamiques.

soutenue le 26 juin 2008 devant le jury composé de :

G. CARRIER	Ingénieur de recherche à l'Onera, Châtillon	Examineur
J.-A. DÉSIDÉRI	Directeur de recherche, INRIA, Sophia-Antipolis	Rapporteur
M. MEAUX	Ingénieur de recherche à Airbus, Toulouse	Examineur
B. MOHAMMADI	Professeur à l'Université Montpellier II	Rapporteur
J. NOAILLES	Professeur émérite à l'ENSEEIH, Toulouse	Examineur
P. SAGAUT	Professeur à l'Université Pierre et Marie Curie, Paris	Directeur de thèse
M. VISONNEAU	Directeur de recherche, École Centrale, Nantes	Examineur

Thèse préparée à Airbus et au CERFACS (Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique)

Réf. CERFACS : TH/CFD/08/62

Remerciements

Je remercie M. J.-A. Désidéri et M. B. Mohammadi d'avoir accepté d'être rapporteur de cette thèse, ainsi que M. G. Carrier, M. M. Meaux, M. J. Noailles, M. P. Sagaut et M. M. Visonneau pour leur participation à mon jury de thèse.

Lors de cette thèse CIFRE, j'ai partagé mon temps entre le service méthodes aérodynamiques d'Airbus (EGAMT) et le service CFD (plus particulièrement l'équipe AAM) du CERFACS. Ce fut un réel plaisir d'un point de vue professionnel autant que personnel de faire partie de ces deux équipes aux compétences énormes et de bénéficier des moyens de ces deux structures (désolé pour les scripts qui font booter kali). Je salue donc P. Larrieu, G. Chevalier et J.-F. Boussuge d'être parvenu à une telle atmosphère de travail.

Je tiens aussi à remercier les nombreuses personnes d'origines diverses avec qui j'ai partagé du temps, des idées, des astuces ou du code lors de ces trois années et sans qui cette thèse ne serait pas ce qu'elle est.

Je voudrais saluer mon initiateur, J. Peter, qui m'a fait découvrir le monde merveilleux de l'optimisation de formes aérodynamiques lors d'un stage DEA-ingénieur à l'ONERA.

Je remercie P. Sagaut, mon directeur de thèse, pour sa confiance sans faille et ses conseils avisés.

Merci également à M. Montagnac, mon tuteur CERFACS, pour son aide tout au long de cette thèse. Son recul et sa logique m'ont beaucoup apporté, notamment lors de la rédaction.

Je remercie spécialement M. Meaux, mon tuteur Airbus. Sa vivacité d'esprit et son intérêt ont été un parfait stimulateur et nos nombreuses discussions débridées sur l'optimisation et bien d'autres sujets très enrichissantes.

Enfin, merci à mon entourage et particulièrement à Sarah qui m'a soutenue activement durant la dernière ligne droite.

Table des matières

Introduction	9
I Optimisation numérique et outils de conception aérodynamique à Airbus	13
I.1 Définition d'un problème d'optimisation	14
I.1.1 Optimisation multi-objectif de Pareto	15
I.1.2 Optimisation multi-objectif par combinaison	16
I.1.2.1 Equivalence des objectifs par leur valeur absolue	17
I.1.2.2 Equivalence des objectifs par leur variation admissible	18
I.2 Choix d'un algorithme d'optimisation	19
I.2.1 Problèmes abordés	20
I.2.2 Exploration, exploitation	21
I.2.3 Robustesse d'un optimum	21
I.2.4 Algorithmes d'optimisation globale	21
I.2.5 Algorithmes d'optimisation locale	22
I.2.6 Bilan comparatif entre algorithmes d'optimisation globale et locale	23
I.3 Présentation de la chaîne d'optimisation OPTaliA	23
I.3.1 Initialisation du problème et paramétrisation	23
I.3.2 Déformation de maillage	26
I.3.3 Simulation numérique	26
I.3.4 Post-traitement et évaluation des fonctions (traînée, portance)	26
I.3.5 Evaluation du gradient	27
I.3.5.1 Méthode des différences finies	27
I.3.5.2 Méthode de l'état adjoint discret	27
I.3.6 Quelques chiffres	28
I.3.7 Algorithme d'optimisation	29
I.3.8 Gestion du processus d'optimisation	29
I.4 Intérêt des surfaces de réponse	29
I.4.1 Surfaces de réponse globales	30
I.4.2 Surfaces de réponse locales	30
I.5 Bilan	31
II Construction efficace de surfaces de réponse par Krigeage et Cokrigeage	33
II.1 Introduction	33
II.2 Article « Building Efficient Response Surfaces of Aerodynamic Functions with Kriging and Cokriging »	34
II.3 Compléments	48

II.3.1	Krigeage et corrélation spatiale	48
II.3.2	L'échantillonnage	49
II.3.2.1	Taille de l'échantillonnage	49
II.3.2.2	Echantillonnage global	51
II.3.2.3	Critères d'échantillonnage pour la recherche de minimum	52
II.3.2.3.1	Critère LCB	52
II.3.2.3.2	Critère EI	52
II.3.2.3.3	Critère LCI	52
II.3.3	Cokrigeage et interpolation du gradient	52
II.3.3.1	Parallélisation du Cokrigeage	53
II.3.3.2	Cokrigeage indirect tronqué en direction	53
II.3.3.3	Cokrigeage indirect tronqué en échantillon	53
II.4	Bilan	53
III	Optimisation fluide structure biniveau par surfaces de réponse globales	55
III.1	Intégration aérodynamique d'une installation motrice	56
III.2	Optimisation multidisciplinaire	57
III.2.1	Introduction	57
III.2.2	Couplage géométrique et couplage physique	59
III.2.3	Optimisation intégrée, FIO	60
III.2.4	Optimisation intégrée biniveau, BLISS	61
III.3	Application à l'optimisation fluide-structure de la position d'un moteur	64
III.3.1	Fonctions objectifs et variables de forme, haut-niveau	65
III.3.2	Analyse bas-niveau structure du mât	65
III.3.3	Analyse bas-niveau aérodynamique de la configuration fuselage-aile-mât-nacelle	67
III.3.4	Optimisation haut-niveau	68
III.3.5	Résultats	70
III.3.5.1	Surface de réponse structure	70
III.3.5.2	Surface de réponse aérodynamique	73
III.3.5.3	Résultats de l'optimisation haut-niveau	73
III.3.6	Bilan	76
IV	Comparaison d'algorithmes d'optimisation par gradient et surface de réponse locale	79
IV.1	Introduction	79
IV.1.1	Environnement de l'étude comparative	80
IV.1.2	Optimiseurs par surfaces de réponse locales	81
IV.2	Acte de colloque « Comparison of Gradient and Response Surface Based Optimization Frameworks Using Adjoint Method »	82
IV.2.1	Résumé de l'optimisation à 6 variables du profil RAE2822	107
IV.2.2	Résumé de l'optimisation à 45 variables de l'aile AS28	108
IV.3	Optimisation à 48 variables d'un avion complet (configuration AS28)	109
IV.4	Compléments	116
IV.4.1	Discussion sur la définition du problème	116
IV.4.1.1	Déformations positives	116
IV.4.1.2	Traînée de frottement	116
IV.4.1.3	Portance et traînée induite	117
IV.4.1.4	Irrégularité du champ de déformation	117
IV.4.2	Validation des solutions obtenues	118
IV.5	Bilan	120

Conclusion	123
A Critères de raffinement d'échantillonnage à pondération dynamique	135
A.1 Les critères employés	135
A.1.1 Critères à pondération dynamique entre fonction et erreur	135
A.1.1.1 Critère LCB dynamique (LCB_b_{dyn})	136
A.1.1.2 Critère LCB par incrément de confiance (LCB_b_{dyntr})	136
A.1.1.3 Critère LCB par incrément de méfiance (LCB_b_{dyndtr})	137
A.2 Résultats	137
A Etude de la fonction de Hicks-Henne	141
A.1 Fonction de Hicks-Henne	141
A.2 Dérivée par rapport à l'amplitude	142
A.3 Dérivée par rapport au pole	142
A.4 Dérivée par rapport à l'exposant	143

Introduction

L'avion est un système complexe et plusieurs années sont nécessaires pour sa conception. Cette dernière est un travail collaboratif entre plusieurs milliers d'ingénieurs de disciplines très variées : aérodynamique, structure, systèmes, propulsion, etc. Elle représente pour l'industriel un coût important, en raison notamment des performances toujours accrues des nouveaux avions. La réduction des coûts de conception repose en partie sur la maîtrise de la durée du cycle de conception, d'où l'importance de développer des outils efficaces pour les concepteurs.

Pour les concepteurs aérodynamiciens, l'objectif n'est pas simplement de faire voler l'avion en générant une force de portance suffisante, mais surtout de minimiser la force de traînée s'opposant au mouvement de l'avion.

Une réduction de traînée entraîne directement une consommation de carburant inférieure. Cette économie est capitale puisqu'elle peut se traduire, soit par une distance franchissable plus importante, soit par une masse de carburant inférieure qui permet d'augmenter le nombre de passagers. Par exemple, sur un gros porteur, une réduction de 1% de la traînée est équivalente à une masse réduite de plusieurs tonnes. Ce pourcent de réduction de traînée peut aussi se traduire, après une année d'exploitation, par environ 300 000 litres de carburant économisés.

Il existe de nombreux moyens pour réduire la traînée d'un avion [113], dont les principaux sont :

- le profilage de la forme externe de l'avion,
- le profilage de la forme des excroissances (antennes, sondes, etc.),
- la réduction de la rugosité du carénage externe,
- l'ajout de dispositifs de contrôle d'écoulement (générateurs de tourbillons, fentes, jets, décharges électriques, etc.).

Le cadre de cette thèse est la conception de formes générant un minimum de traînée (profilage de l'avion).

Certaines relations simples liant traînée et forme existent. Par exemple, la traînée de frottement est générée par le frottement de l'air sur le carénage externe de l'avion. Ainsi, les aérodynamiciens savent que pour la réduire il faut diminuer la surface du carénage externe. Cependant, ce n'est pas l'unique source de traînée. Pour évaluer la traînée totale d'une forme, il faut déterminer l'écoulement à son voisinage. Cependant, il est impossible de maîtriser intuitivement les relations existantes entre forme et traînée. Le concepteur peut faire appel à son expérience pour proposer une déformation qui lui semble intéressante, mais doit s'en remettre à la simulation numérique pour évaluer le gain éventuel. C'est la méthode empirique par essai et erreur. Elle n'est pas adaptée aux problèmes de réduction de traînée car elle nécessiterait de nombreux essais pour parvenir à éventuellement améliorer la traînée. Or, plusieurs heures sont nécessaires pour évaluer la performance d'une seule forme.

Pour résoudre ce problème efficacement, il suffit de considérer la traînée comme une fonction mathématique dépendant de la déformation. Ainsi, il devient possible d'utiliser un algorithme d'optimisation

pour rechercher automatiquement le minimum de cette fonction. Cet algorithme s'inscrit dans un processus itératif nécessitant un certain nombre d'évaluations de fonction. Le savoir-faire du concepteur intervient alors au niveau de la définition du domaine d'étude. C'est l'optimisation numérique de formes aérodynamiques.

En plus de l'outil de simulation numérique, le processus d'optimisation de formes nécessite la mise en place d'un algorithme d'optimisation et d'outils de déformation de formes. L'efficacité du processus repose sur l'ensemble de tous ces outils et de la manière dont ils sont connectés. Cependant, l'algorithme d'optimisation occupe une place importante puisque c'est le chef d'orchestre du processus. Dans ce contexte, ce travail de thèse vise à améliorer un outil d'optimisation de formes aérodynamiques à travers l'étude de nouveaux algorithmes d'optimisation.

L'optimisation est une discipline très importante du monde de l'informatique. En effet, un classement des dix algorithmes les plus importants du XXème siècle en science et ingénierie a été établi en 2000 [32]. Le deuxième algorithme de ce classement est un algorithme d'optimisation, la méthode du simplexe pour les problèmes linéaires. Cette méthode, établie en 1947 par G. Dantzig, n'est bien entendue pas la seule. Il en existe de très nombreuses : simplexe, plus forte pente, Newton, quasi-Newton, programmation quadratique séquentielle, région de confiance, algorithme génétique, recuit simulé, essais particuliers, surfaces de réponse, etc. Cette diversité s'explique par la variété des problèmes que l'optimisation est capable de résoudre, par exemple :

- concevoir une forme aérodynamique optimale,
- trouver le plus court chemin entre deux points pour un GPS,
- établir l'ordonnancement d'objets de dimensions diverses de manière à en rentrer le maximum dans un volume restreint, etc.

Malheureusement, il n'existe pas d'algorithme d'optimisation qui permette de résoudre tous ces problèmes. Etant donné que le nombre de problèmes d'optimisation possibles est infini, un théorème [121] a montré qu'un algorithme adapté à une certaine classe de problèmes sera inadapté pour le reste des problèmes. Pour que l'outil d'optimisation aérodynamique soit efficace, son algorithme d'optimisation doit être adapté aux spécificités des problèmes de minimisation de traînée.

D'un point de vue général, les problèmes d'optimisation de formes aérodynamiques sont très fréquemment résolus par des méthodes de descente par gradient. Cette méthode a l'avantage de converger très rapidement. Comme l'évaluation de la traînée nécessite une simulation numérique coûteuse en temps de calcul, elle permet d'obtenir rapidement une forme plus performante. De plus, elle s'accommode bien des problèmes à grand nombre de variables, notamment grâce à l'utilisation du vecteur gradient. En pratique, son défaut majeur est qu'elle converge trop vite vers un optimum local. Elle améliore la performance de la forme mais, en général, un autre algorithme pourrait, sur le même problème, parvenir à une meilleure amélioration.

Parmi les algorithmes d'optimisation globale, ceux basés sur les surfaces de réponse semblent les plus prometteurs. Une surface de réponse est tout simplement un modèle approché de la fonction à optimiser. A partir de quelques évaluations, il est possible d'interpoler la fonction sur tout le domaine d'étude en utilisant, par exemple, une méthode de Krigage. Bien que cette interpolation mène à une perte de précision par rapport à la fonction obtenue par simulation numérique de l'écoulement autour de la forme, elle a l'avantage de pouvoir être évaluée à un coût de calcul dérisoire. L'emploi de surfaces de réponse permet, lors du processus d'optimisation, de substituer la fonction par son approximation et de réduire ainsi le coût de calcul. Toutefois, la fonction étudiée étant complexe, la surface de réponse est en général incapable de la reproduire de manière totalement fiable. Cela peut être évité en limitant la complexité de la fonction, par exemple en se limitant à un faible nombre de variables. La plupart des optimisations de formes aérodynamiques utilisant cette méthode dans le contexte de la simulation numérique haute fidélité se contentent de problèmes ainsi simplifiés.

Plus récemment, l'utilisation de surfaces de réponse employant l'approche de Cokrigage a été

suggérée pour améliorer leur précision. Le Cokrigeage permet de construire des surfaces de réponse interpolant non seulement l'information sur la fonction, mais aussi sur son gradient. Cette information additionnelle permet à la surface de réponse d'être plus précise. Cette thèse traite d'optimisation de formes par méthodes à base de surfaces de réponse par Krigeage et Cokrigeage.

Les formes étudiées sont principalement des ailes en régime de croisière. Les ailes ont pour fonction principale de générer la portance, mais sont aussi une importante source de traînée. Le régime de croisière a été choisi car un avion long-courrier y passe la majorité de son temps de vol. En outre, dans ces conditions, l'écoulement autour de l'avion y est complexe. L'avion vole à une vitesse élevée et la vitesse de l'air par rapport à l'avion est localement supérieure à la vitesse du son (écoulement transsonique), notamment sur l'extrados¹ de l'aile. Il se trouve que de petites déformations de l'aile se traduisent par de fortes variations de la traînée. Une variation d'un pourcent peut être la conséquence d'une déformation locale de quelques centimètres alors que l'aile mesure plusieurs mètres. Il faut donc déterminer en tout point de l'aile l'amplitude et la direction de la déformation à appliquer pour réduire la traînée. Ce champ surfacique de vecteurs est déterminé grâce à quelques dizaines de bosses de Hicks-Henne, au total le nombre de variables envisagé est de l'ordre de la cinquantaine.

L'outil d'optimisation à améliorer est bâti autour d'une méthode de descente par gradient (type quasi-Newton BFGS). C'est l'outil OPTaliA employé en conception à Airbus. Il utilise pour la simulation numérique de l'écoulement le logiciel *elsA* (développé par l'ONERA). L'écoulement est modélisé par le système d'équations RANS. La traînée est calculée par une méthode de type champ lointain. Cet outil est capable de calculer efficacement le vecteur gradient² par l'intermédiaire de la résolution du système adjoint discret d'*elsA*.

Le chapitre I introduit les notions d'optimisation essentielles à la bonne compréhension des applications qui suivront. Il décrit aussi globalement l'outil d'optimisation de formes aérodynamiques OPTaliA.

Le chapitre II porte sur la construction de surfaces de réponse pour modéliser efficacement les fonctions usuellement étudiées en aérodynamique. Pour cela, les formulations retenues sont le Krigeage pour interpoler une fonction à partir d'une base de données d'évaluations de la fonction seule, et le Cokrigeage pour interpoler une fonction à partir d'une base de données d'évaluations de la fonction et de son gradient. Les méthodes de distribution des échantillons (points de construction) dans le domaine d'étude y sont aussi abordées. Une étude de l'influence du nombre d'échantillons et du nombre de variables sur la précision des surfaces de réponse y est menée. Une nouvelle formulation de Cokrigeage indirect tronqué en échantillon est aussi introduite. Elle permet de réduire le coût de calcul, trop important, du Cokrigeage en dimension élevée.

Dans cette thèse, les surfaces de réponse sont classifiées en deux catégories : locales et globales. Une surface de réponse locale est considérée fiable seulement en ses points de construction. Une surface de réponse globale est considérée fiable sur le domaine d'étude complet.

Le chapitre III traite d'une optimisation par surfaces de réponse globales. Le problème étudié est de type multidisciplinaire, c'est à dire qu'une forme doit être conçue pour être performante du point de vue de plusieurs disciplines simultanément. Ici, deux disciplines sont abordées : l'aérodynamique et la structure. L'application traite de l'intégration d'une installation motrice sous l'aile d'un avion. L'optimisation multidisciplinaire de formes est une branche particulière de l'optimisation de formes pour laquelle des méthodes spécifiques sont utilisées afin de prendre en compte les contraintes liées à la répartition des tâches entre services experts. Chez un industriel tel qu'Airbus, l'aérodynamique et la structure sont du ressort de deux départements différents qui utilisent des outils différents. Les

¹L'extrados est la surface supérieure (orientée vers le ciel). L'intrados est la surface inférieure de l'aile (orientée vers le sol).

²Gradient : sensibilité de la fonction par rapport aux variables de forme.

méthodes d'optimisation multidisciplinaire permettent de coopérer efficacement tout en maintenant une certaine autonomie entre disciplines. Pour résoudre ce problème, une méthode d'optimisation biniveau est adoptée. Elle est basée sur des surfaces de réponse globales pour connecter les disciplines entre elles. La validité de cette application est discutée à travers, d'une part une étude d'erreur d'approximation et d'autre part à partir d'un front de Pareto.

Le chapitre IV est centré sur la discipline aérodynamique. L'étude menée dans ce chapitre part du constat que pour avoir un outil d'optimisation efficace, l'algorithme d'optimisation doit être particularisé aux problèmes traités en aérodynamique [121]. Un nouvel algorithme d'optimisation par surfaces de réponse locales est proposé. Il est mis en œuvre sous la forme de deux versions : une à base de Krigage et une à base de Cokrigage. Pour juger de la validité de ces algorithmes, ils sont comparés à un algorithme de référence utilisant une méthode de descente par gradient. Trois cas tests représentatifs des problèmes rencontrés en conception aérodynamique sont traités : un profil d'aile à 6 variables de forme, une aile isolée à 45 variables et un avion complet à 48 variables. Afin de juger des qualités de ces algorithmes, des critères de comparaison sont mis en place. Ils caractériseront notamment leurs performances en terme d'exploitation et d'exploration.

Optimisation numérique et outils de conception aérodynamique à Airbus

Nomenclature

Cd	Coefficient de traînée
Cd_f	Coefficient de traînée de frottement
Cd_p	Coefficient de traînée de pression
Cd_{vp}	Coefficient de traînée de pression visqueuse
Cd_w	Coefficient de traînée de choc
Cd_i	Coefficient de traînée induite
Cd_{sp}	Coefficient de traînée parasite
\mathcal{D}	Domaine d'étude des variables
E	Energie totale du fluide (n_X)
\mathcal{F}	Domaine des solutions admissibles
F	Vecteur des fonctions objectifs (n_F)
G	Vecteur des fonctions contraintes (n_G)
HH	Fonction de déformation de Hicks-Henne
k	Vecteur des coefficients d'équivalence (n_F)
n_α	Nombre de variables, dimension de \mathcal{D}
n_{calc}	Nombre de vecteurs variables à évaluer pour converger
n_F	Nombre de fonctions objectifs
n_G	Nombre de fonctions contraintes
n_{pop}	Taille de la population
n_X	Nombre d'éléments du maillage
R	Résidu des équations RANS discrétisées
U	Vecteur vitesse du fluide ($3n_X$)
W	Vecteur des variables conservatives ($5n_X$)
X	Maillage (n_X)
α	Vecteur des variables (n_α)
λ	Vecteur adjoint ($5n_X$)
ρ	Masse volumique du fluide (n_X)

Indices

i	$\in [1, n_F]$
j	$\in [1, n_\alpha]$

k k-ième itération $\in [1, +\infty[$

Exposants

* Valeur optimale
 NF Intégration champ proche
 FF Intégration champ lointain

Dans ce chapitre, nous allons aborder toutes les notions utiles à la bonne compréhension et à la reproductibilité des travaux effectués durant cette thèse. Etant donnée la variété des domaines connexes à l'optimisation de formes aérodynamiques par simulation numérique, seuls les points clés seront décrits avec mention de références bibliographiques pour certains points d'approfondissement.

L'optimisation automatique peut s'appliquer à n'importe quel système tant qu'il est possible de le formaliser par un problème d'optimisation et de le modéliser.

Dans cette thèse, l'optimisation est utilisée dans trois contextes différents. Tout d'abord, pour modéliser des données issues de simulations numériques par une surface de réponse, l'optimisation permet de choisir parmi une famille de surfaces celle qui minimise l'erreur ; le réglage des paramètres de construction d'une surface de réponse est un problème d'optimisation. Ensuite, lors de la conception de la structure primaire de l'avion, la masse doit être optimisée tout en respectant des contraintes de charge structurelle. Les variables de conception sont les dimensions et positions des poutres, panneaux et renforts. Enfin, d'un point de vue aérodynamique, il faut optimiser la traînée en respectant la contrainte de portance grâce à des déformations du carénage externe de l'avion.

Ces problèmes peuvent être traités de manière empirique par essai et erreur (« trial and error method »), mais leur résolution par l'optimisation procure de nombreux avantages. Elle permet :

- d'atteindre la meilleure performance pour un problème donné,
- de gagner du temps en augmentant la réactivité de la boucle de conception,
- d'aider le concepteur à mieux comprendre les phénomènes physiques en œuvre,
- de trouver une solution à un problème trop complexe à résoudre de manière empirique.

I.1 Définition d'un problème d'optimisation

L'optimisation est une branche des mathématiques, plus particulièrement de l'analyse fonctionnelle, qui étudie les méthodes permettant de trouver un minimum d'une fonction. C'est par la formulation du problème d'optimisation que le lien entre les mathématiques et la conception aérodynamique est réalisé. Un problème d'optimisation est défini par :

- L'espace des variables, $D \subset \mathbb{R}^{n_\alpha}$.
- La (les) fonction(s) contrainte(s) d'égalité ou inégalité à respecter déterminant l'espace d'étude, $\mathcal{D} \subset D$.
- La fonction objectif $F(\alpha)$, à minimiser sur \mathcal{D} .

Le problème consiste à trouver le minimum (global) α^* de la fonction F sur \mathcal{D} . La fonction objectif peut être multimodale et admettre des optima locaux respectant la condition suivante,

$$\alpha^* \in \mathcal{D} \text{ minimum local de } F \Leftrightarrow \exists \mathcal{S}_{\mathcal{D}} \subset \mathcal{D}, \mathcal{S}_{\mathcal{D}} \text{ voisinage de } \alpha^* \text{ tel que } \forall \alpha \in \mathcal{S}_{\mathcal{D}}, F(\alpha^*) \leq F(\alpha). \quad (\text{I.1})$$

Autrement dit, un minimum se trouve au fond d'une vallée notée $\mathcal{S}_{\mathcal{D}}$.

La fonction représentée en figure I.1 admet trois minima. Le meilleur des minima, α_2^* , est le minimum global, les autres sont des minima locaux.

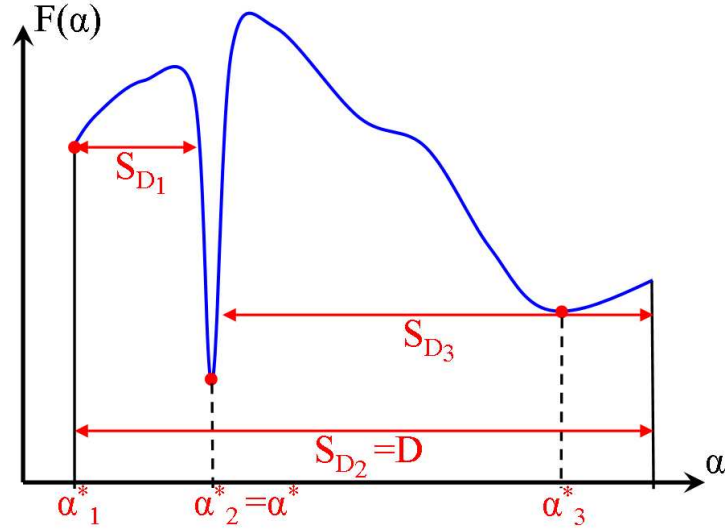


FIG. I.1: Illustration de la notion de minimum.

Les problèmes étudiés dans le cadre de cette thèse considèrent uniquement des contraintes de bornes sur la valeur des variables et aucunes contraintes générales. A partir d'une forme initiale, il s'agira de déterminer le champ de déformation permettant de minimiser la traînée de l'avion en condition de vol de croisière.

Lorsque la fonction objectif est dérivable, une condition nécessaire pour caractériser un minimum est connue sous la condition de Karush-Kuhn-Tucker (KKT [13]) qui en l'absence de contraintes s'écrit,

$$\alpha^* \in \mathcal{D} \text{ est un minimum} \Rightarrow \nabla F(\alpha^*) = 0. \quad (\text{I.2})$$

Si de plus la fonction est convexe sur \mathcal{D} , alors la réciproque est vraie [13]. Cette condition peut s'étendre aux problèmes à variables bornées en considérant le gradient projeté (élimination de la composante normale à la frontière).

Il est difficile de résumer la performance d'un système complexe tel qu'un avion à travers une seule fonction objectif. En général, le travail des concepteurs consiste à trouver un compromis entre plusieurs objectifs antagonistes. En optimisation, ces problèmes multi-objectifs peuvent être traités de deux manières :

- optimisation multi-objectif de Pareto,
- optimisation multi-objectif par combinaison.

I.1.1 Optimisation multi-objectif de Pareto

En considérant plusieurs fonctions objectifs, le problème n'admet plus une seule solution mais un ensemble de solutions appelé front de Pareto. Un optimum de Pareto est un état dans lequel un objectif ne peut pas être amélioré sans en détériorer un autre. Une telle solution est dite non dominée. La relation de dominance entre deux solutions s'exprime par la relation,

$$\forall (\alpha_1, \alpha_2) \in \mathcal{D}, \alpha_1 \prec \alpha_2 \Leftrightarrow \begin{cases} \forall i, F_i(\alpha_1) \leq F_i(\alpha_2) \\ \exists \tilde{i} / F_{\tilde{i}}(\alpha_1) < F_{\tilde{i}}(\alpha_2) \end{cases}. \quad (\text{I.3})$$

Le front de Pareto est défini comme l'ensemble des solutions non dominées. En général, c'est une hypersurface d'un espace de dimension n_F sans particularité. Il peut être discontinu (souvent à cause de contraintes), non borné ou non convexe.

Cette notion est très simple à appréhender de manière graphique en traçant le domaine des solutions admissibles. Le front de Pareto est un sous-ensemble de la frontière de ce domaine. Par exemple, la figure I.2 représente les solutions admissibles pour un problème de minimisation de deux objectifs. Dans ce cas, le front est une courbe définie par une fonction bornée, discontinue et convexe par morceaux.

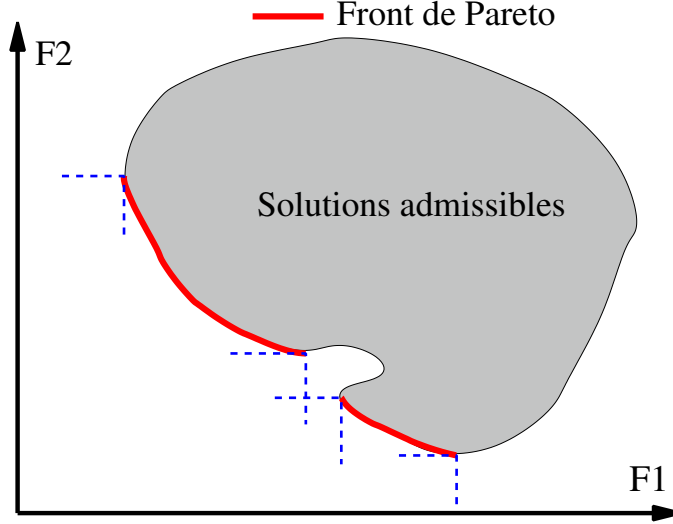


FIG. I.2: Front de Pareto et relation de dominance.

Les extrémités d'un front continu sont obtenues en résolvant chacun des problèmes mono-objectifs de manière isolée. Ce sont donc les points intérieurs au front qui correspondent aux compromis intéressants au sens de Pareto. Un optimum équilibré du point de vue de Pareto se situera au centre du front, équidistant des bornes du front.

Ce front de Pareto est une bonne base de discussion permettant d'étudier plusieurs solutions concurrentes. Il est utile pour définir *a posteriori* un bon compromis (ou plusieurs) entre les fonctions objectifs. Toutefois, l'obtention d'un tel ensemble de solutions optimales est beaucoup plus coûteux en évaluations de fonctions que la résolution d'un problème mono-objectif, d'autant plus coûteux que le nombre de fonctions est important. Il est très utilisé pendant la phase d'avant-projet, où les fonctions étudiées sont rapides à évaluer. Lorsque que le coût de calcul des fonctions est important, il faut se contenter de rechercher un point unique de ce front en effectuant *a priori* un compromis entre les objectifs [60, 87].

I.1.2 Optimisation multi-objectif par combinaison

De nombreux algorithmes d'optimisation ne traitent que les cas mono-objectifs. Un problème multi-objectif peut être reformulé en un problème mono-objectif en déterminant des coefficients d'équivalence k_i entre les objectifs F_i pour effectuer une combinaison linéaire (agrégation des objectifs).

$$F(\alpha) = \sum_i k_i F_i(\alpha). \quad (\text{I.4})$$

Le vecteur \vec{k} peut être interprété comme la normale à un hyperplan dans l'espace des solutions admissibles \mathcal{F} de dimension n_F . Un hyperplan sépare le domaine R^{n_F} en deux parties. L'hyperplan H de normale \vec{k} et passant par $s_0 \in \mathcal{F}$ est support du domaine convexe \mathcal{F} si et seulement si tous les points du domaine \mathcal{F} sont contenus d'un seul côté de cet hyperplan :

$$H_{\vec{k}} \text{ support de } \mathcal{F} \text{ en } s_0 \in \mathcal{F} \Leftrightarrow \begin{cases} \forall s \in \mathcal{F}, \overrightarrow{(s - s_0)} \cdot \vec{k} \geq 0 \\ \text{ou} \\ \forall s \in \mathcal{F}, \overrightarrow{(s - s_0)} \cdot \vec{k} \leq 0 \end{cases}. \quad (\text{I.5})$$

Le point s_0 fait forcément partie de la frontière du domaine \mathcal{F} . Les ouvrages de Boyd et Vandenberghe [13] ou de Hiriart-Urruty et Lemaréchal [46] décrivent plus exhaustivement ces notions d'optimisation convexe.

Il est possible qu'un minimum obtenu par résolution du problème multi-objectif par combinaison selon le vecteur \vec{k} appartienne au front de Pareto. Pour cela il faut qu'un hyperplan de normale \vec{k} support du domaine des solutions admissible \mathcal{F} existe. De plus, il faut que cet hyperplan soit support de \mathcal{F} en un point appartenant au front de Pareto. Finalement, l'optimum de Pareto correspondant au vecteur d'équivalence \vec{k} se trouve, lorsqu'il existe, à l'intersection entre le front et l'hyperplan support de normale \vec{k} . Par exemple, sur la figure I.3 les solutions s_1 , s_2 et s_3 appartiennent au front de Pareto. Pourtant, la solution s_2 ne pourra jamais être obtenue en résolvant le problème multi-objectif par combinaison.

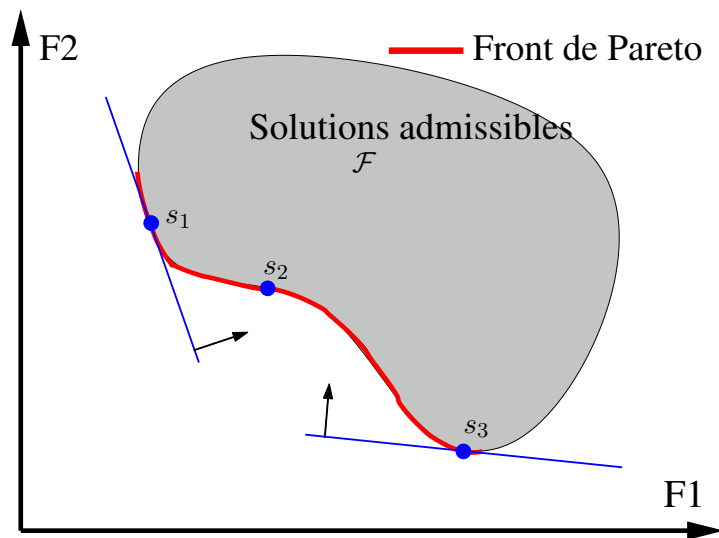


FIG. I.3: Hyperplan support et front de Pareto.

La figure I.4 montre que l'augmentation du poids associé à la fonction objectif $F1$ se traduit par un optimum plus performant vis-à-vis de cette fonction $F1$ (et moins performant vis-à-vis de $F2$). Ce résultat se généralise à tout front de Pareto convexe : lorsque l'on augmente le poids associé à un des objectifs, l'optimum global sera plus performant vis-à-vis de cet objectif.

Lorsque le domaine des solutions admissibles est convexe, chaque optimum constituant le front de Pareto correspond à la solution d'un problème résolu par méthode de combinaison en utilisant un certain vecteur de coefficients d'équivalence. La réciproque n'étant pas vraie, le choix de ces coefficients n'est pas trivial [27, 28] et des méthodes de détermination existent comme la programmation par objectif (« Goal Programming ») ou l'intersection de frontière selon la normale (« Normal Boundary Intersection ») [29].

I.1.2.1 Equivalence des objectifs par leur valeur absolue

Pour les fonctions physiques, il existe généralement des coefficients d'équivalence canoniques. Ainsi, lorsque la fonction objectif est la traînée totale d'une forme, les différentes composantes phénoménologiques de la traînée (frottement, pression) ont naturellement le même poids car la même unité physique. Or, ces coefficients d'équivalence physiques ne prennent en compte que la valeur absolue des objectifs et sont totalement indépendants du domaine d'étude \mathcal{D} . Cela correspond à une mise à l'échelle des

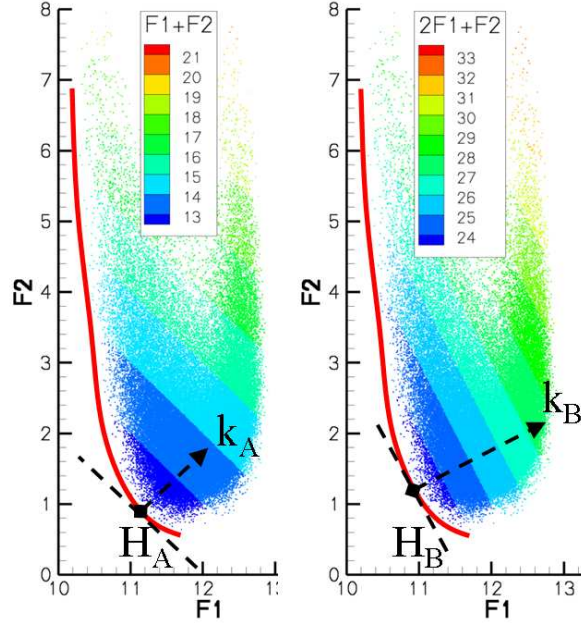


FIG. I.4: Front de Pareto et coefficients d'équivalence. $\vec{k}_A = (1; 1)$ et $\vec{k}_B = (2; 1)$.

objectifs par leur valeur absolue.

$$F(\alpha) = Cd(\alpha) = 1 \cdot Cd_f(\alpha) + 1 \cdot Cd_p(\alpha), \quad (I.6)$$

$$k_1 = k_2 = 1.$$

De la même manière, les problèmes aérodynamique-structure sont souvent résolus en utilisant des coefficients d'équivalence canoniques obtenus en multipliant la fonction objectif masse par la traînée initiale et la fonction objectif traînée par la masse initiale.

$$F(\alpha) = k_1 \cdot M(\alpha) + k_2 \cdot Cd(\alpha), \quad (I.7)$$

$$k_1 = 1, k_2 = \frac{M^{init}}{C_d^{init}}.$$

Le défaut de cette formulation est qu'elle conduit souvent à un optimum appartenant à une extrémité du front de Pareto. Le problème multi-objectif n'est donc pas équilibré au sens de Pareto. Une fonction objectif (ou plusieurs) dominant les autres. Ces fonctions dominées pourraient être complètement ignorées. Le problème ainsi simplifié serait équivalent, il aurait le même optimum global.

I.1.2.2 Equivalence des objectifs par leur variation admissible

La méthode décrite ici s'inspire de la méthode d'intersection de frontière selon la normale [29] qui permet de déterminer des coefficients d'équivalence correspondant à des optima équidistants sur le front de Pareto. Elle a pour objet de déterminer le plus simplement possible un compromis équilibré au sens de Pareto. Elle est issue des observations faites lors des différents cas d'application de cette thèse, notamment sur l'optimisation bi-objectif du chapitre III.

Si l'on cherche à obtenir un optimum équilibré entre les objectifs, on peut faire des suppositions sur la forme du domaine de Pareto pour essayer de déterminer *a priori* un compromis en son centre. Dans les cas abordés en ingénierie, où les variables sont bornées, le front de Pareto peut être considéré borné par les solutions aux problèmes mono-objectifs; aucune fonction objectif ne tend vers l'infini. Supposons de plus que le front soit convexe et dérivable. Après avoir déterminée l'amélioration admissible grâce aux variations de formes autorisées par le domaine d'étude, $\Delta F_i(\mathcal{D})$, soit en considérant n_F

optimisations mono-objectifs, soit en l'estimant, un ensemble de n_F points définissant un hyperplan $H_{\Delta F}$ est alors connu. Sur l'exemple bi-objectif de la figure I.5, l'hyperplan $H_{\Delta F}$ est défini par le vecteur \vec{h} correspondant aux solutions aux problèmes mono-objectifs, $F_{k=(1,0)}^*$ et $F_{k=(0,1)}^*$ ¹.

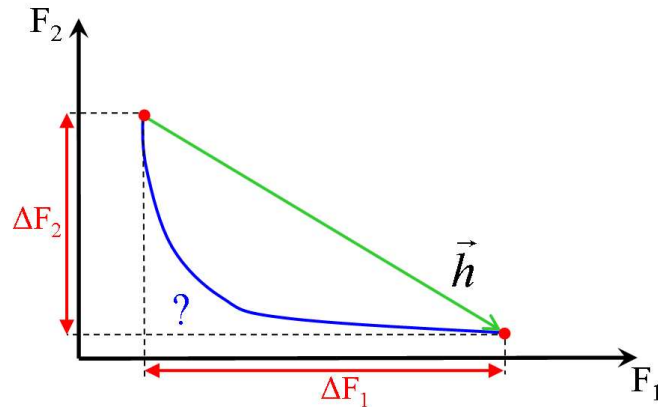


FIG. I.5: Equivalence des objectifs par leur variation admissible.

D'après le théorème des accroissements finis, tout front de Pareto dérivable admettra au moins un point dont la tangente correspondra au vecteur \vec{h} . Comme le front est convexe, cela permet d'affirmer qu'il existe un hyperplan parallèle à $H_{\Delta F}$ qui est effectivement support du front de Pareto. Or, le vecteur \vec{k} ,

$$\vec{k} = \begin{pmatrix} \frac{1}{\Delta F_1(\mathcal{D})} \\ \frac{1}{\Delta F_2(\mathcal{D})} \end{pmatrix}, \quad (\text{I.8})$$

est normal à cet hyperplan,

$$\vec{k} \cdot \vec{h} = \begin{pmatrix} \frac{1}{\Delta F_1(\mathcal{D})} \\ \frac{1}{\Delta F_2(\mathcal{D})} \end{pmatrix} \cdot \begin{pmatrix} \Delta F_1(\mathcal{D}) \\ -\Delta F_2(\mathcal{D}) \end{pmatrix} = 0. \quad (\text{I.9})$$

Finalement, la mise à l'échelle des objectifs par leur variation admissible correspond aux coefficients d'équivalence,

$$k_i = \frac{1}{\Delta F_i(\mathcal{D})}. \quad (\text{I.10})$$

I.2 Choix d'un algorithme d'optimisation

L'algorithme d'optimisation converge, selon un processus itératif, vers une solution. Les processus d'optimisation réels (cf. Figure I.7) peuvent être représentés de manière simplifiée (cf. Figure I.6) en considérant deux éléments : l'optimiseur et l'évaluateur. A chaque itération, l'optimiseur propose un ou plusieurs vecteurs de variables de forme à évaluer qui forment une population de taille n_{pop} . L'évaluateur doit alors calculer les fonctions objectifs ou contraintes et éventuellement leurs sensibilités par rapport aux variables (gradients).

Face à la diversité des problèmes pouvant être résolus par l'optimisation, de nombreuses méthodes existent. Malheureusement, aucune n'est capable de traiter efficacement tous les cas. En effet, Wolpert

¹La solution utopique du problème multi-objectif qui serait aussi optimale pour chacun des problèmes mono-objectifs est appelée point de Nadir.

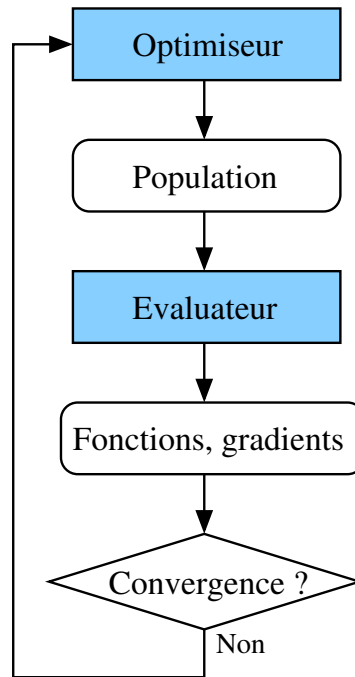


FIG. I.6: Processus général d'optimisation.

et Macready [121] ont démontré que si l'on considère l'ensemble de tous les problèmes d'optimisation possibles, alors aucun algorithme n'est meilleur qu'un autre (« no free lunch theorems for optimization »). Le choix d'un algorithme ne peut donc pas se faire par comparaison sur des cas tests généraux, mais doit aussi étudier des problèmes représentatifs des applications envisagées.

Pour choisir la méthode la plus adaptée à un problème, les caractéristiques principales prises en compte sont :

- la capacité à traiter des problèmes mono- ou multi-objectifs,
- la capacité à éviter les minima locaux pour converger vers le minimum global,
- la rapidité de convergence, c'est-à-dire le nombre de vecteurs variables à évaluer pour converger (qui peut être rapporté à la dimension du domaine d'étude),
- la nécessité du calcul de gradients.

I.2.1 Problèmes abordés

En optimisation haute-fidélité, l'évaluation des fonctions et de leurs gradients est très coûteuse ; le temps de restitution nécessaire à une évaluation est de plusieurs heures de calcul sur supercalculateur. Pour l'industriel, le temps de restitution total de l'optimisation doit être de l'ordre d'une semaine. Ainsi, la première contrainte du processus d'optimisation est d'obtenir une solution en moins de 100 itérations au total.

Le nombre de fonctions étudiées est faible (un ou deux objectifs et quelques contraintes). Le chapitre III traite d'une optimisation multidisciplinaire de la position d'une installation motrice résolue par combinaison linéaire entre fonctions objectifs aérodynamique (traînée) et structure (masse). Les optimisations décrites au chapitre IV sont mono-objectifs, elles ne considèrent que la traînée aérodynamique.

Le nombre de variables est souvent important, de l'ordre d'une centaine $n_\alpha = \mathcal{O}(100)$, mais les cas considérés ici en optimisation par surfaces de réponse ne dépasseront pas 48 variables.

Le facteur dominant dans la complexité des problèmes abordés est donc le nombre de variables.

La rapidité de convergence de l'algorithme ne doit pas dépendre trop fortement de la dimension du problème.

I.2.2 Exploration, exploitation

Pour un algorithme d'optimisation, l'exploration est sa capacité à explorer le domaine des variables pour rechercher la meilleure vallée, c'est à dire celle qui contient l'optimum global. A l'inverse, l'exploitation est sa capacité à converger rapidement vers le minimum d'une vallée donnée à partir d'un vecteur variables initial.

Ces deux notions sont très souvent employées pour décrire les algorithmes d'optimisation. Toutefois, il n'y a pas de définition communément admise permettant de les mesurer. La comparaison d'algorithmes d'optimisation du chapitre IV sera effectuée grâce à quelques critères représentant ces deux notions. Ces critères sont définis en IV.1.1.

I.2.3 Robustesse d'un optimum

La robustesse d'un optimum est aussi une notion déterminante pour les concepteurs et qui ne doit pas être négligée lors du choix de l'algorithme d'optimisation. La forme à optimiser est modélisée de manière déterministe par un maillage ; cette forme est donc optimisée en négligeant les incertitudes liées par exemple aux procédés de fabrication ou aux simplifications du problème (le couplage aéroélastique fluide-structure). Pour diminuer la probabilité que les performances de la forme optimisée ne soient pas vérifiées en réalité, on peut éliminer les minima non-robustes. Un minimum robuste est peu sensible aux incertitudes ; il est situé au fond d'une vallée large, de faible courbure. Sur l'exemple de la figure I.1, le minimum α_2^* n'est pas robuste et, bien qu'il soit plus performant que α_3^* , on pourra privilégier ce dernier.

De la même manière, en aérodynamique, on recherche un optimum robuste vis-à-vis des conditions de vol (Mach, incidence) [45,82,91]. Cela amène à la conception multipoint : au lieu d'optimiser l'avion uniquement en condition de croisière, on va plutôt l'optimiser sur tout un domaine de vol.

Pour l'algorithme d'optimisation, cela signifie qu'il n'est pas toujours nécessaire de chercher à atteindre l'optimum global. Une exploration limitée peut permettre de filtrer les optima non robustes en ignorant les vallées trop escarpées. Finalement, il faut trouver un bon compromis entre exploration et exploitation.

I.2.4 Algorithmes d'optimisation globale

Les algorithmes d'optimisation globale recherchent l'optimum global. En théorie, cette recherche nécessite l'exploration complète du domaine des solutions admissibles. Elle est coûteuse en nombre d'évaluations de fonctions. En pratique, il est impossible de vérifier le caractère globale d'un minimum à moins d'admettre des hypothèses sur la régularité de la fonction objectif.

Le gros désavantage des méthodes globales est que leur coût augmente, dans le pire des cas, exponentiellement avec la complexité du problème (nombre de variables n_α et nombre de fonctions n_F) [13]. Bellman [10] a décrit ce concept en tant que malédiction dimensionnelle (« curse of dimensionality »).

La distribution de points selon une grille est une méthode d'optimisation à exploration seule. Bellman nous indique que si l'on veut trouver le minimum global d'une fonction de n_α variables avec une erreur relative sur les variables inférieure à ϵ , alors il faudra effectuer $n_{calc} = \mathcal{O}((1/\epsilon)^{n_\alpha})$ évaluations. Par exemple, sur un problème de dimension $n_\alpha = 10$, si l'on veut localiser à 10% près les variables optimales, il faut $n_{calc} = \frac{1}{0.1}^{10}$ évaluations. Même pour une fonction évaluée en seulement une seconde, il faut plus de 10 000 jours pour obtenir la solution.

Bien que la plupart des méthodes globales (algorithme génétique, recuit simulé, séparation et évaluation, etc.) utilisent des méthodes stochastiques pseudo-aléatoires pour l'exploration, comme la mu-

tation des algorithmes génétiques, ce coût exponentiel est toujours latent [5,33]. Ainsi, pour garantir la diversité de population d'un algorithme génétique, Goldberg [41] recommande une taille de population croissant exponentiellement avec la dimension.

$$n_{pop} = \mathcal{O}(2^{n\alpha}) \quad (\text{I.11})$$

Les optimiseurs globaux ne sont pas adaptés aux problèmes de grande complexité et aux problèmes où l'évaluation des fonctions est coûteuse. Lorsque la fonction est multimodale, ils permettent l'obtention de meilleures solutions que les méthodes locales sans garantie sur le caractère globale de la solution obtenue.

I.2.5 Algorithmes d'optimisation locale

Les optimiseurs les plus rapides partent d'une solution initiale et l'exploitent en déterminant un chemin de descente vers le minimum le plus proche. La solution obtenue dépend donc de la solution initiale. L'utilisation du gradient, une information vectorielle, améliore beaucoup la rapidité de convergence et est primordiale pour limiter la dépendance au nombre de variables. Cependant, les méthodes de descente basées sur le gradient ne traitent que les problèmes mono-objectifs.

La méthode de la plus forte pente (« steepest descent ») utilise directement l'opposé du gradient comme direction de descente. Cette méthode simple n'est toutefois pas la plus rapide.

La méthode de référence en terme de rapidité de convergence est la méthode de Newton-Raphson de recherche de gradients nuls. C'est aussi la moins robuste et elle nécessite le calcul de la matrice Hessienne H (formée des dérivées secondes). De plus, le calcul de cette matrice est délicat en pratique car les fonctions étudiées ne sont par forcément \mathcal{C}^2 et le calcul analytique de cette matrice est fastidieux et coûteux.

$$\left[\begin{array}{l} \textbf{Itération de la méthode de Newton-Raphson :} \\ 1. \text{ Calculer le vecteur de descente} \\ \quad H(\alpha_k) \cdot d_k = -\nabla F(\alpha_k) \\ 2. \text{ Mise à jour des variables} \\ \quad \alpha_{k+1} = \alpha_k + d_k \end{array} \right. \quad (\text{I.12})$$

Afin d'améliorer la robustesse de la méthode de Newton, les algorithmes de type quasi-Newton construisent itérativement une approximation de la matrice Hessienne, en utilisant seulement l'évolution des gradients et en supposant que la fonction peut être approchée localement par un développement limité quadratique autour de l'optimum. La formulation itérative la plus courante est celle de Broyden-Fletcher-Goldfarb-Shanno (BFGS) [114].

$$\left[\begin{array}{l} \textbf{Itération d'une méthode de descente quasi-Newton :} \\ 1. \text{ Calculer une approximation de la hessienne } \hat{H}_k \text{ à partir de } \nabla F(\alpha_k) \text{ et } \nabla F(\alpha_{k-1}) \\ 2. \text{ Calculer la direction descente} \\ \quad \hat{H}_k(\alpha_k) \cdot d_k = -\nabla F(\alpha_k) \\ \quad \|d_k\| = 1 \\ 3. \text{ Recherche linéaire, déterminer } a_k > 0 \text{ en minimisant} \\ \quad \min_{a_k} (F(\alpha_k + a_k d_k)) \\ 4. \text{ Mise à jour des variables} \\ \quad \alpha_{k+1} = \alpha_k + a_k d_k \end{array} \right. \quad (\text{I.13})$$

Lorsque la fonction étudiée est bruitée, ces méthodes locales peuvent présenter des problèmes de convergence.

Ces algorithmes par gradients sont les plus rencontrés en optimisation aérodynamique de formes pour la conception détaillée, mais les méthodes de région de confiance (« trust region ») [2] ou de

programmation quadratique séquentielle (« Sequential Quadratic Programming ») [114] sont de plus en plus utilisées et d'efficacité semblable.

Jameson et Vassberg [52] ont montré et vérifié sur un problème analytique non linéaire que la convergence de l'algorithme de la plus forte pente dépend quadratiquement du nombre de variables,

$$n_{calc} = \mathcal{O}(n_{\alpha}^2), \quad (\text{I.14})$$

alors que le nombre d'évaluations nécessaires pour la convergence d'un algorithme de type quasi-Newton évolue linéairement avec le nombre de variables,

$$n_{calc} = \mathcal{O}(n_{\alpha}). \quad (\text{I.15})$$

I.2.6 Bilan comparatif entre algorithmes d'optimisation globale et locale

Les observations générales précédentes ont été confirmées sur des problèmes aérodynamiques, notamment par comparaison d'algorithmes [90, 119]. Le coût des algorithmes d'optimisation globale dépendant trop fortement de la complexité du problème, leur emploi est limité aux problèmes de faible complexité (faible nombre de variables et de fonctions) et pour lesquels les fonctions étudiées ne sont pas trop coûteuses [90]. Ces algorithmes sont adaptés à l'optimisation préliminaire de l'avion, car les fonctions étudiées sont rapides à évaluer et les paramètres définissant l'architecture globale de l'avion ne sont pas trop nombreux. De plus, cette phase de recherche du meilleur concept devant être exploratoire, les méthodes globales multi-objectifs sont particulièrement recommandées.

Les méthodes d'optimisation locale utilisant le gradient sont idéales pour obtenir le plus rapidement possible une solution à un problème mono-objectif dont le nombre de variables peut être important. Elles sont donc employées en conception détaillée.

D'autres classifications sont parfois utilisées pour différencier les algorithmes d'optimisation :

- stochastique/heuristique.

Les méthodes stochastiques sont celles qui font intervenir le pseudo-aléatoire. Par exemple, la procédure de mutation d'un algorithme génétique fait appel au pseudo-aléatoire. Ces méthodes sont généralement des méthodes globales.

- ordre 0/ordre 1/ordre 2.

Les méthodes d'ordre 0 utilisent l'information sur la fonction, celles d'ordre 1 requièrent aussi l'information du gradient et celles d'ordre 2 nécessitent en plus le calcul de la matrice Hessienne.

I.3 Présentation de la chaîne d'optimisation OPTaliA

La chaîne OPTaliA² [73, 82] d'optimisation automatique de formes aérodynamiques est utilisée pour la conception détaillée de l'avion à Airbus. C'est un outil d'optimisation haute-fidélité car aucun modèle réduit n'est employé entre le solveur aérodynamique et l'optimiseur. En conception détaillée, les paramètres de forme pilotant l'architecture globale de l'avion sont figés et les déformations envisagées sont généralement de faibles amplitudes afin de limiter l'impact sur les autres disciplines. La forme issue de la phase de projet antérieure (conception avant-projet) permet d'initialiser l'optimisation locale.

I.3.1 Initialisation du problème et paramétrisation

La définition du problème d'optimisation dans OPTaliA est représentée par la phase « Init » sur la figure I.7. Elle consiste à définir le maillage de la forme initiale et une paramétrisation associée. La paramétrisation définissant les degrés de liberté autorisés sur la forme s'effectue grâce à des fonctions analytiques de Hicks-Henne [43], chacune définie par trois variables, l'amplitude A , la position p et

²OPTaliA : OPTimisation en Aérodynamique

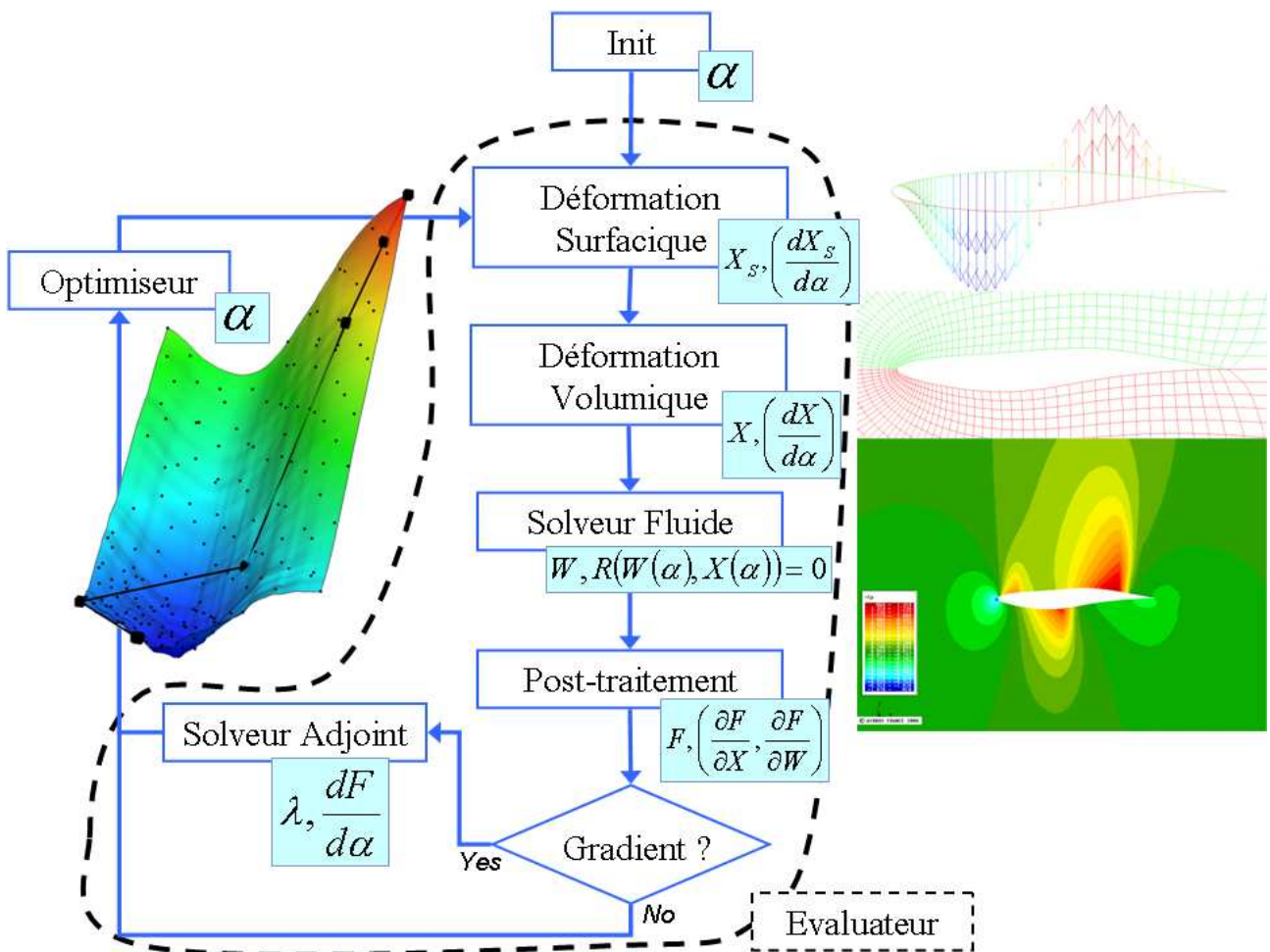


FIG. I.7: Processus d'optimisation par méthode de gradient dans OPTaliA.

l'expansion β (cf. Figure I.8) à travers l'équation I.16. Le vecteur contenant ces variables de forme est noté α .

$$HH(x) = A \cdot \sin^\beta \left(\pi x \frac{\ln 0,5}{\ln p} \right), \quad (I.16)$$

avec $x \in [0; 1]$, $A \in \mathfrak{R}$, $\beta \in [2; 10]$, $p \in]0; 1[$.

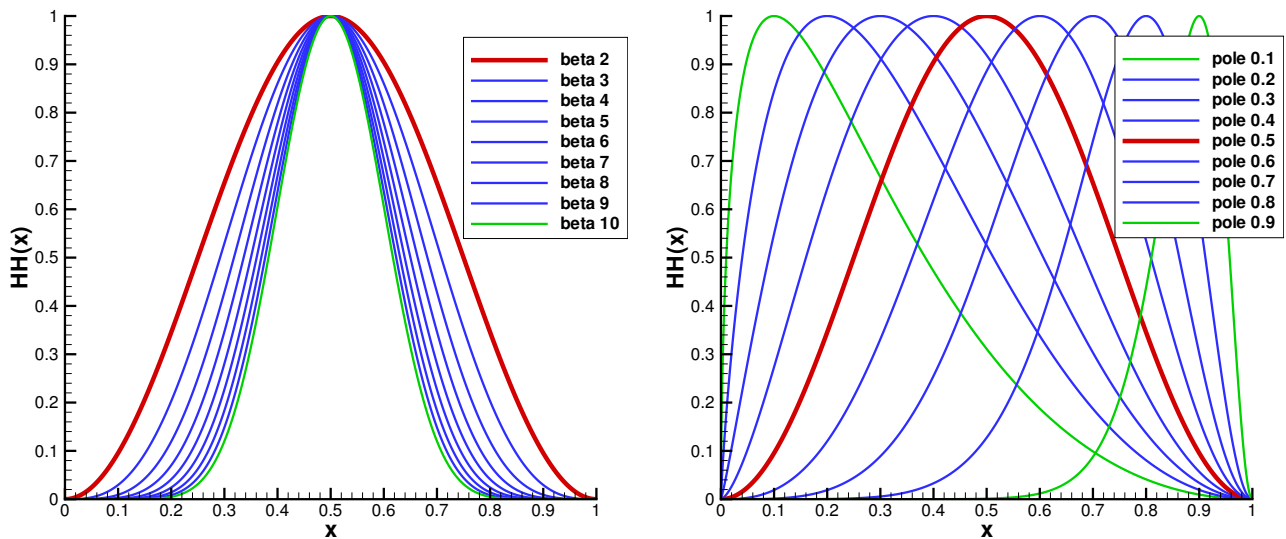


FIG. I.8: Bosses de Hicks-Henne, influence des paramètres de forme β et p .

Ces déformations bidimensionnelles (cf. Figure I.9) sont appliquées à des lignes de maillage dans la direction définie soit par la normale courante, soit par l'axe vertical z . Dans les cas tridimensionnels (cf. Figure I.10), on utilise une spline cubique d'amplitude maximale pour la section pilote et nulle en deux sections limites définies par l'utilisateur pour étendre la déformation. L'extension selon cette troisième direction ne fait pas l'objet de variables de forme supplémentaires. Dans tous les cas, la déformation devient progressivement nulle lorsque l'on se rapproche d'une frontière topologique du domaine d'application.

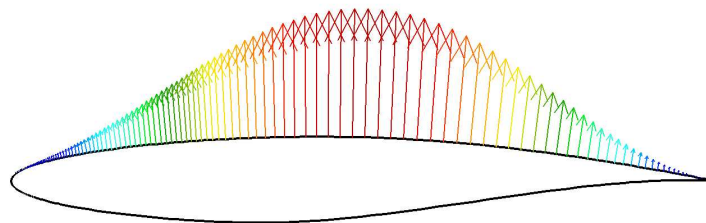


FIG. I.9: Vecteurs de déformation de la peau d'un profil d'aile par application d'une bosse de Hicks-Henne bidimensionnelle.

D'autres techniques de paramétrisation de maillage existent comme la déformation libre (« free form deformation »), le déplacement direct de la position des nœuds de maillage surfacique, ou la méthode PARSEC. Les comparaisons de ces approches [64, 122] donnent l'avantage aux bosses de Hicks-Henne. En effet, elles ont été spécialement créées pour l'optimisation aérodynamique de sections d'ailes et nécessitent beaucoup moins de variables qu'une paramétrisation considérant directement les nœuds de maillage surfacique et permettent d'atteindre des formes complexes tout en garantissant une déformation lisse (tangente nulle en zéro et un, cf. Figure I.8).

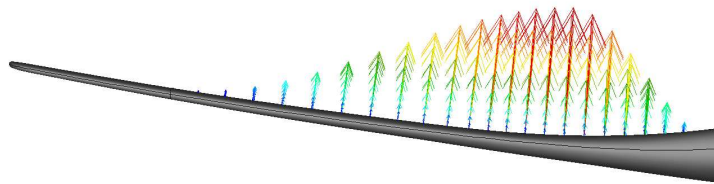


FIG. I.10: Vecteurs de déformation de la peau d'une aile par application d'une bosse de Hicks-Henne tridimensionnelle.

I.3.2 Déformation de maillage

Les bosses ainsi définies permettent de calculer un champ de déformation surfacique des parois du maillage. Cette déformation surfacique est ensuite propagée au maillage volumique, noté X , en utilisant une méthode mixte entre intégrale et interpolation transfinie [30, 82]. La méthode intégrale est la plus coûteuse en temps de calcul, mais c'est aussi celle qui conserve le mieux la qualité du maillage. Elle est utilisée pour déplacer les nœuds des frontières de chaque bloc. La méthode d'interpolation transfinie est très rapide, mais nécessite la connaissance préalable des déplacements des frontières. Elle est utilisée pour déplacer les nœuds intérieurs aux blocs.

Par rapport à une stratégie de génération de maillage, la déformation de maillage a plusieurs avantages. Tout d'abord, il est possible de calculer la sensibilité du maillage par rapport aux variables de forme, $\frac{dX}{d\alpha}$, par une méthode analytique. De plus, le temps nécessaire à la déformation est inférieur au temps nécessaire à la génération d'un maillage (surtout en l'absence de mailleur automatique).

I.3.3 Simulation numérique

Une fois le maillage $X(\alpha)$ obtenu, l'écoulement du fluide autour de la nouvelle forme doit être simulé. Le code de calcul *elsA* [17] développé par l'ONERA résout les équations de Navier-Stokes moyennées (approche RANS) pour la simulation d'écoulements compressibles. Dans le cadre de notre travail, le modèle de turbulence choisi pour compléter le système est le modèle de Spalart-Allmaras à une équation. La discrétisation de ce système d'équations s'effectue par une formulation volume fini dans laquelle les inconnues sont affectées aux centres des cellules de discrétisation [47]. Le domaine de calcul est modélisé par un maillage, X , multidomaine structuré par bloc.

Pour la discrétisation spatiale, le schéma décentré de Roe est utilisé. Afin d'étendre à l'ordre 2 ce schéma, la méthode MUSCL et le limiteur dérivable de Van Albada utilisés pour extrapoler les flux de Roe. Le schéma de résolution numérique temporel est le schéma Euler rétrograde (implicite). Sa résolution est effectuée par la méthode LU-SSOR (« Lower Upper Symmetric Successive Over Relaxation »). Afin d'améliorer la vitesse de convergence, les techniques multigrille et de pas de temps local sont employées.

La résolution de ce système d'équations discrétisées donne le vecteur des variables conservatives

$$W(\alpha) = (\rho, \rho U_x, \rho U_y, \rho U_z, \rho E). \quad (\text{I.17})$$

On note ce système discrétisé

$$R(W(\alpha), X(\alpha)) = 0. \quad (\text{I.18})$$

I.3.4 Post-traitement et évaluation des fonctions (traînée, portance)

Le champ volumique de l'écoulement n'est pas directement exploitable par un optimiseur. Pour obtenir une fonction objectif $F(\alpha)$ scalaire, il faut utiliser des grandeurs intégrées grâce au module de post-traitement,

$$F(\alpha) = F(W(\alpha), X(\alpha)). \quad (\text{I.19})$$

Le code *ffd41* [6,15,31] développé par l'ONERA emploie une approche champ-lointain pour calculer les coefficients aérodynamiques d'un avion : traînée induite Cd_i , traînée visqueuse Cd_{vp} , traînée de choc Cd_w , traînée de frottement Cd_f , traînée parasite Cd_{sp} , portance Cl et moments Cm . Le lien entre les valeurs de traînées en approche champ proche (NF) et champ lointain (FF) est,

$$Cd^{NF} = (Cd_p + Cd_f)^{NF} = (Cd_{vp} + Cd_w + Cd_i + Cd_{sp} + Cd_f)^{FF}, \quad (I.20)$$

d'où,

$$(Cd_p)^{NF} = (Cd_{vp} + Cd_w + Cd_i + Cd_{sp})^{FF}. \quad (I.21)$$

L'approche champ lointain par volume d'intégration, bien que plus complexe, a deux avantages par rapport à l'approche champ proche d'intégration à la peau :

- Elle est plus précise et moins sensible aux imprécisions numériques dues à un maillage trop grossier et à l'ajout de dissipation artificielle car elle isole leur influence par la détermination d'un terme de traînée parasite Cd_{sp} ,

$$\begin{aligned} Cd^{FF} &= Cd_{vp} + Cd_w + Cd_i + Cd_f \\ Cd^{NF} &= Cd^{FF} + Cd_{sp} \end{aligned} \quad (I.22)$$

- La décomposition de la traînée selon ses diverses sources physiques ainsi que l'identification spatiale des zones de création de traînée grâce aux volumes d'intégration permettent aux concepteurs de mieux comprendre le processus de création de traînée et ainsi d'améliorer les formes. Par exemple, l'évolution de la traînée de choc selon l'envergure est une information primordiale pour la conception d'une aile et l'intégration d'une installation motrice.

Ce code est aussi capable de calculer les sensibilités de la fonction objectif par rapport au maillage, $\frac{\partial F}{\partial X}$, et par rapport aux variables conservatives, $\frac{\partial F}{\partial W}$, nécessaires au calcul de gradient.

I.3.5 Evaluation du gradient

Plusieurs méthodes existent pour calculer le gradient, $\nabla F(\alpha)$.

I.3.5.1 Méthode des différences finies

La plus simple et la plus utilisée est la méthode des différences finies d'ordre un, estimant le gradient par les taux d'accroissement,

$$\frac{\partial F(\alpha)}{\partial \alpha_j} = \frac{F(\alpha + \delta \alpha_j) - F(\alpha)}{\delta \alpha_j}. \quad (I.23)$$

Cette méthode est extrêmement coûteuse puisqu'elle nécessite le calcul de $n_\alpha + 1$ écoulements. De plus, sa précision est fortement liée au choix du pas de différences finies $\delta \alpha_j$.

I.3.5.2 Méthode de l'état adjoint discret

Certaines méthodes analytiques permettent de calculer le gradient à un coût indépendant du nombre de variables. Ainsi, la méthode de différentiation automatique en mode inverse permet de calculer un gradient pour un coût maximal de cinq calculs de fonctions [88]. Toutefois, cette méthode est prohibitive d'un point de vue mémoire.

Dans *elsA*, une autre méthode dont le coût est indépendant du nombre de variables a été retenue : la méthode de l'état adjoint discret [39, 50]. Elle nécessite la résolution d'un système par fonctions étudiées, soit $n_F + n_G$ systèmes au total, impliquant la linéarisation du système discrétisé des équations de Navier-Stokes par rapport aux variables conservatives pour le calcul du terme $\frac{\partial R}{\partial W}$. Pour ce faire, il faut calculer le vecteur adjoint λ , solution du système linéaire

$$\lambda^t \frac{\partial R}{\partial W} = -\frac{\partial F}{\partial W} \text{ ou } \lambda^t = -\frac{\partial F}{\partial W} \left(\frac{\partial R}{\partial W} \right)^{-1}. \quad (I.24)$$

Comme $\forall \alpha, R(W, X) = 0$ (Equation I.18), il vient

$$\nabla R(W, X) = \frac{\partial R}{\partial W} \frac{dW}{d\alpha} + \frac{\partial R}{\partial X} \frac{dX}{d\alpha} = 0. \quad (\text{I.25})$$

Soit,

$$\frac{dW}{d\alpha} = - \left(\frac{\partial R}{\partial W} \right)^{-1} \frac{\partial R}{\partial X} \frac{dX}{d\alpha}. \quad (\text{I.26})$$

Or, le gradient recherché est

$$\nabla F(\alpha) = \frac{\partial F}{\partial X} \frac{dX}{d\alpha} + \frac{\partial F}{\partial W} \frac{dW}{d\alpha}, \quad (\text{I.27})$$

compte tenu de l'équation I.26,

$$\nabla F(\alpha) = \frac{\partial F}{\partial X} \frac{dX}{d\alpha} - \frac{\partial F}{\partial W} \left(\frac{\partial R}{\partial W} \right)^{-1} \frac{\partial R}{\partial X} \frac{dX}{d\alpha}. \quad (\text{I.28})$$

Finalement, le gradient calculé par la méthode de l'état adjoint est

$$\nabla F(\alpha) = \frac{\partial F}{\partial X} \frac{dX}{d\alpha} + \lambda^t \left(\frac{\partial R}{\partial X} \frac{dX}{d\alpha} \right). \quad (\text{I.29})$$

Le calcul de gradient par la méthode de l'adjoint discret fait intervenir un calcul de sensibilité par différences finies pour le terme $\left(\frac{\partial R}{\partial X} \frac{dX}{d\alpha} \right)$. Pour la linéarisation du flux visqueux, on utilise une approche « couche mince » pour négliger les termes de dérivation croisée et une approximation dite de « μ_t figé », considérant les inconnues déterminées par le modèle de turbulence indépendantes des variations de forme [96]. La précision du calcul de gradient par ce solveur adjoint a été établie [8, 82, 87, 95] par des comparaisons entre gradients calculés par trois méthodes différentes (état adjoint, différences finies et équation linéarisée) sur différents cas de validations (2D, 3D, non-visqueux et visqueux). Il s'avère que sur les cas les plus complexes (écoulements visqueux transsoniques), l'erreur est de l'ordre de quelques pourcents. Le système adjoint est résolu en utilisant la même méthode de résolution que le système direct, à l'exception des techniques d'accélération de convergence. Si l'on néglige le temps de calcul nécessaire aux calculs des sensibilités $\frac{\partial F}{\partial X}$ et $\frac{\partial F}{\partial W}$, le temps de calcul nécessaire à la résolution de l'équation adjointe (Equation I.24) par la méthode *LUSSOR* est environ identique au temps de calcul d'un écoulement seul (Equation I.18), car le schéma implicite de résolution des équations *RANS* implique le calcul de la sensibilité $\frac{\partial R}{\partial W}$.

I.3.6 Quelques chiffres

Concrètement, les maillages aujourd'hui utilisés en production sont constitués de plusieurs dizaines de millions de points répartis en plusieurs centaines de blocs. Dans le cadre de cette thèse, la géométrie la plus complexe traitée (cf. chapitre IV) est un maillage à sept millions de nœuds ($n_X = 7 \times 10^6$) d'une configuration fuselage aile installation motrice nécessitant six heures de calcul sur huit processeurs AMD Opteron 275 cadencés à 2.2 Ghz pour effectuer 800 itérations.

Afin d'éviter les possibles phénomènes d'hystérésis, aucune reprise n'est effectuée pour initialiser le calcul de l'écoulement sur la forme courante (α_k) par l'écoulement simulé sur la forme précédente (α_{k-1}). Comme le temps de calcul nécessaire à l'algorithme d'optimisation est négligé, la durée du processus d'optimisation dépend linéairement du temps nécessaire à une unique simulation numérique.

Avec l'augmentation de la taille des maillages utilisés en production, la sensibilité du maillage par rapport aux paramètres de formes, $\frac{dX}{d\alpha}$, est devenue une limitation du processus d'optimisation par gradient. En effet, la mémoire nécessaire pour stocker ce terme sur l'intégralité du maillage est équivalente à autant de maillages que de variables de forme ($n_X \times n_\alpha$). Pour un maillage à $n_X = 100 \times 10^6$ nœuds occupant environ deux giga-octets de mémoire, le terme de sensibilité par rapport à

cent variables de forme requiert environ deux cents giga-octets de mémoire. Même si ce terme n'existe en réalité que pour une liste de blocs impactés par les variations de forme, il nécessite un espace de stockage important dès lors que la géométrie étudiée est tridimensionnelle et le nombre de variables de forme important.

I.3.7 Algorithme d'optimisation

Historiquement, les premières optimisations numériques par Hicks *et al.* [44, 45] employèrent l'algorithme de descente par gradient *CONMIN* [115] et ce type d'algorithme est toujours largement employé dans la communauté scientifique [19, 39, 53, 66, 82, 84, 85]. La librairie d'optimisation utilisée dans OPTaliA est *DOT* [114], une évolution de la librairie *CONMIN*. Ces algorithmes d'optimisation locale sont toujours appréciés car ils convergent très rapidement. Ils sont très efficaces en terme d'exploitation, au détriment de leur capacité d'exploration. Ils nécessitent peu d'évaluations de la fonction ce qui permet de limiter la durée de l'optimisation. Bien que depuis les premières optimisations de formes aérodynamiques la puissance des ordinateurs a considérablement augmenté, le temps de calcul est toujours la principale limitation de l'optimisation.

En aérodynamique, la précision de la modélisation simulant l'écoulement évolue avec les capacités de calculs [58]. Le temps de restitution d'une simulation numérique reste grand en dépit de l'augmentation des performances informatiques. Ainsi, les simulations numériques d'écoulements potentiels utilisées en conception détaillée dans les années 80 furent remplacées progressivement par des simulations d'écoulements modélisés par les équations d'Euler puis par l'approche *RANS*. Et, on peut imaginer que dans quelques (dizaines ?) années, la simulation de tourbillons détachés (« Detached Eddy Simulation ») ou la simulation aux grandes échelles (« Large Eddy Simulation ») sera employée en production par les industriels. De la même manière, la géométrie est de plus en plus fidèle à la réalité grâce à l'augmentation du nombre de nœuds de maillage.

La pratique montre que les problèmes usuels en aérodynamique présentent de nombreux optima locaux auxquels les algorithmes d'optimisation locale ne peuvent pas échapper. En effet, la position et l'intensité du choc présent sur l'extrados d'une voilure en régime d'écoulement transsonique varient fortement avec les modifications de forme de l'extrados, même lorsqu'elles sont de faible amplitude. Dans ces conditions, la qualité de la solution obtenue par une optimisation locale dépend de la forme initiale choisie.

I.3.8 Gestion du processus d'optimisation

L'optimisation est un processus automatique ; il faut donc que chaque élément ou module impliqué soit suffisamment robuste pour être exécuté sans intervention de l'utilisateur (mode « batch »). Le logiciel OPTaliA se charge de gérer l'exécution successive de chacun des modules et l'échange de données entre ces modules. Sur la figure I.7, les données de sortie de chacun des modules sont représentées.

I.4 Intérêt des surfaces de réponse

Vis-à-vis de l'optimiseur, il apparaît sur la figure I.7 un élément virtuel, l'évaluateur, qui regroupe les éléments entourés en pointillé. Cet élément est en réalité un processus en lui-même chargé d'évaluer la fonction objectif en une entrée donnée en exécutant notamment le solveur haute-fidélité *elsA*. Comme signifié en section I.3.6, le coût d'un tel évaluateur limite les optimisations envisageables (cf. I.3.7 et I.2). Ainsi, les surfaces de réponse sont employées en optimisation pour substituer à cet évaluateur un modèle approché peu coûteux à évaluer. En effet, la surface de réponse prédit une fonction $F(\alpha)$ sur tout le domaine d'étude à partir d'une base de données d'échantillons de simulations numériques.

Les surfaces de réponse sont aussi appelées métamodèles lorsque la fonction modélisée est elle-même issue d'un autre modèle. On distinguera dans cette thèse les surfaces de réponse locales dont le domaine

de validité est limité à un voisinage des points de construction et les surfaces de réponse globales dont le domaine de validité est confondu avec le domaine d'étude.

L'implémentation de surfaces de réponse dans la chaîne d'optimisation par gradients OPTaliA peut d'une part, améliorer sa capacité d'exploration et donc mener à des optima plus performants et d'autre part, rendre le processus d'optimisation plus flexible afin de pouvoir facilement piloter des processus multidisciplinaires.

I.4.1 Surfaces de réponse globales

Les surfaces de réponse globales sont considérées fiables sur tout le domaine d'étude et peuvent donc se substituer à l'évaluateur.

Pour ce faire, leur construction doit être particulièrement soignée afin d'éviter les incohérences entre le métamodèle et la simulation numérique. Ainsi, leur construction est généralement complétée par :

- une analyse de sensibilité des variables afin d'éliminer les variables qui ont peu d'influence sur l'objectif (analyse de la variance),
- une réduction des bornes du domaine d'étude éliminant les zones où la fonction ne présente pas d'intérêt,
- une validation utilisant quelques simulations numériques additionnelles pour comparaison avec le métamodèle.

Comme les surfaces de réponse sont rapides à évaluer, le choix d'un algorithme d'optimisation pour trouver leur minimum n'est pas soumis aux mêmes restrictions que lorsque l'évaluateur est le solveur haute-fidélité. Il est alors possible d'utiliser des algorithmes d'optimisation globale pour trouver l'optimum prédit par la surface de réponse [21, 24, 48, 54, 97, 112]. En outre, Wilson *et al.* [120] ont montré que la recherche d'un front de Pareto sur des surfaces de réponse représentant un problème multi-objectif est pertinente et peut donc permettre d'identifier quelques compromis intéressants à analyser en détail par des optimisations par coefficients d'équivalence.

La surface de réponse globale est un évaluateur « boîte noire » beaucoup plus flexible que les codes de simulation numérique. Issue d'une formulation analytique, elle peut être stockée facilement et peut donc être utilisée comme interface de données entre disciplines. Dans le cadre d'optimisations multidisciplinaires, les surfaces de réponse évitent le couplage de codes hétéroclites et permettent de réutiliser les mêmes outils que lors d'optimisations monodisciplinaires. Elles facilitent donc la mise en place de processus de conception concourante [11, 69, 110, 123] tel que celui étudié au chapitre III. Sellar, Batill et Renaud [101] ont ainsi comparé la solution issue d'un processus d'optimisation par sous-domaine (méthode « Concurrent Sub Space Optimization ») utilisant un couplage par surfaces de réponse à la solution issue du processus d'optimisation complet (méthode « Fully Integrated Optimization ») et mis en avant l'économie en coût de calcul réalisée.

I.4.2 Surfaces de réponse locales

Les surfaces de réponse locales sont considérées fiables seulement au voisinage des échantillons. Moins précises, elles sont aussi plus faciles à construire car elles requièrent moins d'échantillons que les surfaces de réponse globales et moins de validation.

Au lieu de remplacer complètement l'évaluateur par une surface de réponse globale, le couplage (ou hybridation) entre une surface de réponse locale et un optimiseur permet d'améliorer ses performances. Par exemple, au lieu de toujours initialiser les optimisations locales par une déformation nulle, les tendances observées sur la surface de réponse locale peuvent guider vers une initialisation plus proche de l'optimum global [12] et donc améliorer l'exploration. Jin [56] et Ong, Nair et Keane [89] ont décrit différentes stratégies d'hybridation permettant d'améliorer la vitesse de convergence d'un algorithme génétique. L'une d'elle consiste à évaluer la plupart des générations sur surface de réponse et à n'utiliser l'évaluateur précis que lorsqu'une condition est vérifiée (par exemple toutes les 50 générations).

Les surfaces de réponse locales peuvent aussi être utilisées directement en tant qu'optimiseur. En effet, lorsque le minimum du métamodèle n'est pas situé sur un échantillon, il peut être proposé par l'optimiseur comme nouvel individu à évaluer par le code haute fidélité. Les performances des processus d'optimisation par surfaces de réponse de Booker *et al.* [12, 111] (« Surrogate Management Framework »), de Jones, Schonlau et Welch [59] (« SuperEGO ») et Cox et John [25] (« Sequential Design for Optimization »), ont permis de démontrer qu'une alternative aux algorithmes quasi-Newton jusqu'alors majoritairement utilisés pour la conception aérodynamique était possible. L'optimisation numérique de formes aérodynamiques par surfaces de réponse est aujourd'hui largement répandue dans la communauté scientifique [7, 54, 62, 92, 97, 100, 116], principalement en considérant des problèmes à faible nombre de paramètres (quelques dizaines au maximum). Le chapitre IV décrit un tel optimiseur dont l'une des originalités est l'utilisation de surfaces de réponse interpolant le gradient en plus de la fonction.

I.5 Bilan

Ce chapitre a permis d'aborder les notions d'optimisation utilisées dans la suite de ce texte, ainsi que les spécificités de la chaîne d'optimisation aérodynamique d'Airbus, OPTaliA. L'introduction de surfaces de réponse dans cet outil vise à :

- définir un algorithme d'optimisation à coût comparable à l'algorithme par gradient de référence, mais plus performant en terme d'exploration,
- permettre la résolution de problèmes multidisciplinaires en effectuant un couplage par surfaces de réponse.

En premier lieu, il convient de bien maîtriser les aspects relatifs à la construction de ces surfaces. Le chapitre II décrit les choix retenus pour le processus de construction de métamodèles.

Construction efficace de surfaces de réponse par Krigeage et Cokrigeage

Nomenclature

cf. II.2.Article

II.1 Introduction

Les problèmes envisagés en optimisation aérodynamique de formes sont souvent limités par le coût de calcul nécessaire à l'évaluation de la fonction objectif (cf. I.2). L'emploi de surfaces de réponse, rapides à évaluer, pour approcher cet objectif sur tout le domaine d'étude des variables est un moyen d'améliorer l'efficacité d'un outil d'optimisation (cf. I.4).

L'approximation d'une fonction s'effectue en trois étapes :

- La fonction étudiée étant définie sur un domaine borné \mathcal{D} , la première étape consiste à distribuer un nombre n_s de points, ou échantillons, dans ce domaine.
- La fonction doit ensuite être évaluée en ces points.
- Une fois cette base de données d'échantillons disponible, il est alors possible d'utiliser une méthode d'approximation pour modéliser la fonction sur tout l'espace.

Les techniques de modélisation de fonctions les plus courantes ont été comparées par Jin, Chen et Simpson [55] sur treize cas tests analytiques et un cas de conception automobile par simulation numérique. Ces fonctions tests représentent différents niveaux de complexité en terme d'ordre de non linéarité et de nombre de variables (1 à 16). Elles ont été modélisées à partir de distributions d'échantillons de tailles variables, cette taille dépendant linéairement ou quadratiquement de la dimension du problème. Il s'avère que le Krigeage [99] et le réseau de neurones à fonctions radiales de base [9] sont les méthodes les plus précises. Ce résultat a été confirmé par d'autres études comparatives [103, 104] et étendu à des cas tests considérant jusqu'à 101 variables [117]. En outre, Peter *et al.* [94] ont effectué une comparaison de méthodes d'approximation sur un cas à deux variables issu d'un problème de conception aérodynamique d'aube de turbine par simulation numérique. Au lieu d'étudier seulement la précision globale des différents modèles, la capacité à détecter avec précision le minimum de la fonction (précision locale) y est aussi prise en compte. Là aussi, le Krigeage est la méthode la plus performante.

Ainsi, bien que le Krigeage soit une méthode difficile à mettre en œuvre comparée au réseau de neurones ou à la régression polynômiale [117], elle permet de prédire précisément des fonctions issues de simulations numériques présentant de fortes non linéarités. De plus, cette méthode reste valide quel

que soit le nombre d'échantillons, contrairement à un modèle polynômial d'ordre deux qui nécessite un nombre minimal d'échantillons croissant quadratiquement avec la dimension du domaine d'étude, $n_s \geq \frac{(n_{dv}+2)(n_{dv}+1)}{2}$. Enfin, comme le Krigeage est issu d'une formulation statistique, il a l'avantage d'accompagner la prédiction de la fonction d'une prédiction d'erreur. Pour ces raisons, le Krigeage est la méthode d'approximation retenue dans cette thèse.

Avant d'utiliser le Krigeage en tant que boîte noire au sein d'un processus d'optimisation, il convient de bien connaître son comportement. Les deux points clés à approfondir pour construire une approximation robuste et précise sont :

- Les méthodes d'échantillonnage (« sampling methods »). La base d'échantillons étant le seul lien entre la fonction réelle et l'approximation, elle doit concentrer autant d'information que possible.
- Le réglage du modèle, souvent effectué par optimisation numérique comme lors de la phase d'apprentissage des réseaux de neurones, des moindres carrés des régressions polynômiales, ou de la maximisation de vraisemblance du Krigeage.

II.2 Article « Building Efficient Response Surfaces of Aerodynamic Functions with Kriging and Cokriging »

L'article suivant décrit la formulation et la méthode de construction mises en œuvre dans le code de Krigeage utilisé pour les applications décrites dans cette thèse. Il a été publié dans l'*AIAA Journal* (volume 46, numéro 2, pages 498-507) en février 2008. Il permet d'établir les limitations associées à cette formulation afin de cibler correctement les emplois possibles du Krigeage en optimisation. A partir d'un cas test aérodynamique représentatif des fonctions étudiées en optimisation de formes aérodynamiques (II.Article.III), cet article étudie l'influence des méthodes d'échantillonnage sur la précision globale du Krigeage pour déterminer si une méthode, plus efficace que les autres, est à privilégier (II.Article.IV).

L'utilisation de surfaces de réponse interpolant le gradient de la fonction en plus de la valeur de la fonction aux échantillons est un moyen d'éviter la malédiction dimensionnelle décrite en I.2.4. Afin d'interpoler ce gradient, une formulation de type Cokrigeage [23, 76], basée sur les travaux de Morris, Mitchell et Ylvisaker [86], est décrite (II.Article.II.D). La dernière partie (II.Article.V) traite de l'influence de la dimension du domaine d'étude sur la précision globale du Krigeage et du Cokrigeage à partir du cas test précédent.

Building Efficient Response Surfaces of Aerodynamic Functions with Kriging and Cokriging

J. Laurenceau *

CERFACS, Toulouse, 31057, France

P. Sagaut †

F-75252 Paris cedex 5, France

Preprint published February 2008,

doi: 10.2514/1.32308,

AIAA Journal,

vol.46, no.2, pp. 498-507.

In this paper, we compare the global accuracy of different strategies to build response surfaces by varying sampling methods and modeling techniques. The final application of the response surfaces being aerodynamic shape optimization, the test cases are issued from CFD simulations of aerodynamic coefficients. For comparisons, a robust strategy for model fit using a new efficient initialization technique followed by a gradient optimization was applied. Firstly, a study of different sampling methods proves that including 'a posteriori' information on the function to sample distribution can improve accuracy over classical space filling methods like Latin Hypercube Sampling. Secondly, comparing Kriging and gradient enhanced Kriging on two to six dimensional test cases shows that interpolating gradient vectors drastically improves response surface accuracy. Although direct and indirect Cokriging have equivalent formulations, the indirect Cokriging outperforms the direct approach. The slow linear phase of error convergence when increasing sample

size is not avoided by Cokriging. Thus, the number of samples needed to have a globally accurate surface stays generally out of reach for problems considering more than four design variables.

Nomenclature

$CV(x)$	Leave one out cross validation error
\mathcal{D}	Domain of design variables
$dKdS(x)$	Leave one out sensitivity error
$f(x)$	Regression vector $[n_p]$
F	Regression matrix $[N \times n_p]$
L	Model likelihood estimate
N	Order of the correlation matrix
n_{dv}	Number of design variables
n_p	Dimension of regression vectors
n_s	Number of samples
p	SCF power coefficients $[n_{dv}]$
R	Correlation matrix $[N \times N]$
$r(x)$	Correlation vector $[N]$
\mathcal{S}	Domain of sample points
$SCF(.,.)$	Spatial correlation function
$S(x)$	Standard error
s^i	i-th sample
x	Vector of design variables $[n_{dv}]$
$Y_{exa}(x)$	Exact function to approximate
Y_s	Exact function at samples $[N]$
$Y(x)$	Surrogate function
$Z(x)$	Gaussian process
β	Regression coefficients $[n_p]$
σ^2	Model variance
θ	SCF correlation coefficients $[n_{dv}]$

Subscripts

k	$\in [1, n_{dv}]$
l	$\in [1, n_{dv}]$
m	$\in [1, n_{dv}]$

Superscripts

i	$\in [1, n_s]$
$-i$	i-th sample left out for model build
j	$\in [1, n_s]$

I. Introduction

NUMERICAL shape optimization in the field of delta-tailed aircraft design is more and more used since high performance computers enable RANS flow simulation around complete aircraft in a short time scale. Before building a complex optimization process around a CFD suite considering a lot of design variables, some time must be spent to identify an efficient optimizer.

*Ph.D. Student, CFD, 42 Avenue Coriolis.

†Professor, Institut Jean Le Rond d'Alembert, Universit  Pierre et Marie Curie, 4 place Jussieu.

When dealing with computer expensive simulation, the main requirement is to converge to an optimal shape in as few simulations as possible excluding most of the global optimizers such as Genetic Algorithm or Simulated Annealing that are generally applied to conceptual or preliminary design. Assuming that the sensitivity of the functions with respect to the design variables is available at a comparative computational cost of a direct function evaluation (with an adjoint method), the quickest process is then gradient based. Using this strategy, the high-fidelity aerodynamic shape optimization suite *OPTaliA* of *Airbus* showed good results.¹ Despite their efficiency, gradient algorithms suffer from some limitations. They solve only single objective problems (multiple objective functions must be combined) and are easily blocked by local optima whereas aerodynamic functions are generally multimodal. Moreover, heavily constrained problems are hard to solve efficiently.

Promising alternative approaches use response surfaces (or surrogate models) to approximate the expensive computer code on the whole domain by inexpensive models. Numerous strategies exist² and have been applied to perform aerodynamic shape optimization.³⁻⁵ Firstly, response surfaces can lead the optimization process. The surrogate management framework⁶ and trust-region framework⁷ are generally the most favored methods, but refining the response surface at its predicted optimum is also efficient particularly when taking into account uncertainty (Expected Improvement or Lower Confidence Bounding Function criteria).^{4,5,8-10} Response surfaces can be reliable discipline independent models applied to solve Multi-Disciplinary Optimization (MDO) problems when no code coupling is possible.² When the model is sufficiently accurate, the predicted optimum on the response surface can be taken as the true optimum.^{11,12} Secondly, the convergence of existing optimizers can be improved by switching between optimization on expensive model and on response surface (variable fidelity).³ Thirdly, the exploration capabilities of local optimizers can be improved by using the optimum found on the response surface as initialization. Accordingly to these different applications, two concepts of response surfaces are defined. The global response surface is considered an accurate surrogate model of the true function on the whole domain, whereas the local response surface is reliable only near sampled locations but gives information on the overall tendencies.

In this study we wanted to define how to build accurate response surfaces and assess their domain of validity before any applications to shape optimization. Kriging model was chosen to approximate the objective function for its accuracy and robustness

with small datasets.¹³ An efficient methodology is detailed within this article including numerical issues and a new technique to fit the model (section II).

The sampling quality being essential to obtain an accurate model,¹⁴ section IV compares different distribution methods. For a given domain of study, classical sampling are space filling and locations are computed without knowing the function 'a priori'. Beside other modeling techniques, the uncertainty predictions available with Kriging are very useful. Thus, methods of 'a posteriori' sequential sampling based not solely on information on the domain but also on information on the function through predicted errors are also described. In addition to the standard error of Kriging and leave-one-out cross validation errors, two new predicted error criteria are introduced (section II-B-2) and tested (section IV-B). Methods are compared by varying sample size for the reconstruction of a bidimensional aerodynamic test function. 'A posteriori' samplings were already applied to polynomial test cases^{10,15} without in-depth comparison with classical methods. As 'a posteriori' samplings are tuned to the function studied, their performance is closely related to the test case. That is why it was necessary to perform the comparison directly on an aerodynamic function. More than a sampling criterion, these predicted error are very important when using the surrogate model as optimizer in order to have a balancing between exploration of unknown locations and exploration of predicted optima on the model.

As a very powerful gradient computation method was available (adjoint state solver^{1,16,17}), it was also interesting to evaluate Cokriging (gradient enhanced) model. General comparison between Kriging and Cokriging existed only on analytic functions.^{18,19} In the fields of aerodynamic, it was proven that refining iteratively at the predicted minimum of a Cokriging response surface improved the shape of a supersonic business jet depending on up to 15 design variables,^{20,21} but the gradient information was issued from finite differences analysis and no comparison with optimization on a Kriging-based response surface was made. A graphical validation of Cokriging versus Kriging was made on the same aerodynamic test case but considering only two design variables and small datasets.²¹ The benefits of Cokriging on global response surfaces accuracy cannot be assessed by only varying the sample size as the information added to the model through gradients is dimensional. Section V compares Cokriging and Kriging accuracy on an aerodynamic test case, the number of design variables increasing from two to six. A total of 5800 CFD computations were necessary to build reference surfaces.

II. Kriging and Cokriging for Computer Experiments

A. Sampling

A surrogate model only needs a sample distribution to approximate a function, thus it must concentrate as much information as possible. In this way, the sampling must be adapted to : known specificities of the function, modeling formulation and the final application of the surrogate. Without information on the function, design optimal distribution methods using for example entropy criterion²² (or DET-MAX) give space filling locations. Numerous methods exist to evenly distribute samples.

1. 'A priori' Sampling Methods

Four 'a priori' methods requiring only information on the domain and assuming that the function is unknown a priori were used: grid, Halton, Sobol and Latin Hypercube (LHS) sampling (source code of J. Burkardt <http://www.scs.fsu.edu/~burkardt/>).

The grid distribution is the most common in engineering and was thus included in the benchmark. However, it should be used with great cares as truncated grids are to be avoided and grids in high dimension require unreachable sample sizes. The LHS method is also well known. It roughly corresponds to random sampling with guaranteed space filling properties even for small sampling size. Among the four methods presented here only LHS is applicable to high dimensions.

Quasi-random sequences enable methods such as generalized Halton or Sobol²³ to give ordered space filling sampling. These methods are very efficient on low-dimensionnal problems but unapplicable to dimensions higher than 10 without special care. Halton and Sobol are sequential methods. A distribution of $n_s + 1$ samples inherits from the previously computed n_s samples. Latin Hypercube and grid sampling are not sequential methods.

2. 'A posteriori' Sequential Sampling Methods

Maximum error sequential sampling includes information on the true function in sample distribution, explaining the denomination of 'a posteriori' sampling.

It consists in iteratively refining the sample datasets where the model exhibits its maximum of error. By construction it ensures inheritance of samples. Its main drawback is that all intermediate models must be computed. To compute a sampling of a given size the CFD simulations must be launched one at a time, whereas with 'a priori' methods the n_s computations can be launched in parallel. The only requirement is the availability of

an error criterion null at already sampled locations. 'A posteriori' samplings are tuned to the function studied.

B. Kriging

The French mathematician Georges Matheron developed in the early sixties the theory of Kriging from the seminal work of D.G. Krige on mining data. In the field of computer experiments, the Kriging method refers to the DACE formulation²² (fitted by optimization of the likelihood estimate instead of using covariogram) described in the late eighties.

1. Function Prediction

Kriging method is a statistical prediction of a function at untried inputs. It requires fitting the correlation parameters of the model to each sample distribution by solving an optimization problem.

Since computer codes are deterministic, and therefore not subject to measurement error, the usual measures of uncertainty derived from least-square residuals have no obvious meaning. Consequently, statisticians have suggested approximating responses as a combination of a polynomial regression plus localized deviations interpolating samples

$$Y(x) = f^t(x)\beta + Z(x) \approx Y_{exa}(x) \quad (1)$$

$f^t(x)\beta$ is a low-order (constant, linear or quadratic) polynomial regression model. Solving a least-square regression problem gives

$$\beta = (F^t R^{-1} F)^{-1} F^t R^{-1} Y_s \quad (2)$$

In this paper a constant regression model is used, so $n_p = 1$, $f(x) = 1$ and $F = 1$.

$Z(x)$ is the realization of a normally distributed random process with mean zero, variance σ^2 ,

$$\sigma^2 = \frac{1}{n_s} (Y_s - F\beta)^t R^{-1} (Y_s - F\beta) \quad (3)$$

and co-variance

$$cov[Z(s^i), Z(s^j)] = \sigma^2 R_{ij} \quad (4)$$

The matrix of correlation between samples R is determined by a Spatial Correlation Function (SCF).

$$R_{ij} = SCF(s^i, s^j) = \prod_k scf_k(|s_k^i - s_k^j|) \quad (5)$$

This matrix is dense symmetric positive definite with ones along diagonal and become ill-conditioned when samples are too close. The order of the Kriging correlation matrix depends only on the number

of samples considered ($N = n_s$) and not on the number of variables.

The SCF can be any function reflecting the characteristics of the output function. Exponentials or cubic splines are the most common functions. The exponential function is adapted to a wide range of physical applications and is described below

$$scf_k(x) = \exp(-\theta_k x^{p_k}), \theta_k > 0, p_k \in [1, 2] \quad (6)$$

The $2n_{dv}$ correlation parameters (θ_k, p_k) need to be fitted through an optimization process (see section II-E-1)

The cubic spline function was also used

$$scf_k(x) = \begin{cases} 1 - 6(x\theta_k)^2 + 6(x\theta_k)^3, & x < \frac{1}{2\theta_k} \\ 2(1 - x\theta_k)^3, & \frac{1}{2\theta_k} \leq x < \frac{1}{\theta_k} \\ 0, & x \geq \frac{1}{\theta_k} \end{cases} \quad (7)$$

Only n_{dv} correlation parameters $\theta_k > 0$ must be fitted. Commonly, cubic splines are less adapted to very chaotic functions than exponentials.

To evaluate Kriging at an unknown location a vector of correlation r between sample points and unknown is computed

$$r_i(x) = SCF(x, s^i) \quad (8)$$

Finally, the Kriging prediction equation is

$$Y(x) = f^t(x)\beta + r^t(x)R^{-1}(Y_s - F\beta) \quad (9)$$

Practically, the Kriging model can be evaluated as a black-box function by storing vectors β and $rhs = R^{-1}(Y_s - F\beta)$. Then, the evaluation of the model at an unknown location costs only two dot products.

2. Error Prediction

With the statistical prediction of the function is also associated an uncertainty prediction. As by construction the model interpolates the function at samples, its error is null at samples.

The standard error $S(x)$ of the Kriging model grows with distance from samples, its maximum value being σ .

$$S^2(x) = \sigma^2 \left(1 - r^t(x)R^{-1}r(x) + u^t(x)(F^t R^{-1} F)^{-1}u(x) \right) \quad (10)$$

with $u(x)$ vector $n_p \times 1$

$$u(x) = F^t R^{-1}r(x) - f(x) \quad (11)$$

Other estimated errors can be computed with a technique of leave-one-out cross validation.⁹ The idea behind cross validation errors is to study the sensitivity of the model with respect to sample

dataset by altering it of one point. It implies n_s intermediate Kriging constructions and evaluations. Two predicted errors based on leave-one-out cross validation named sample sensitivity error ($dKdS(x)$) and cross validation error ($CV(x)$) are defined

$$dKdS(x) = \frac{1}{n_s} \sum_i |Y(x) - Y^{-i}(x)| \quad (12)$$

$$CV(x) = \frac{1}{n_s} \sum_i S^{-i}(x) \quad (13)$$

Kriging sensitivity error $dKdS(x)$ grows at locations where sample points give more information and is small at location where samples are redundant. Kriging standard sensitivity error $CV(x)$ is very close to the standard error. As cross validation errors are a mean of models built on altered sample datasets, they are not null at samples.

We decided to make a geometric mean between cross validation and standard errors to set them null at samples. That is why errors $dKdS(x)$ and $CV(x)$ are mixed with the standard error, and two new error predictions are introduced

$$dKdS_{mix}(x) = \sqrt{S(x)dKdS(x)} \quad (14)$$

$$CV_{mix}(x) = \sqrt{S(x)CV(x)} \quad (15)$$

Predicted errors are very useful to improve iterative process. When minimizing a function, iteratively refining at the minimum of the response surface usually leads to premature convergence.⁶ This can be improved by including exploration in the process using either a domain exploration ('Poll' step of the surrogate management framework⁶), either an infill criterion based on predicted error.¹⁰ When minimizing global error of a response surface, predicted errors can be used as sampling criteria (see MAXVAR,¹⁰ Application-driven Sequential Designs¹⁵ and IV-B).

C. Model Correlation Parameters Fit

Kriging spatial correlation parameters (θ_k, p_k) from equations 6 or 7 must always be fitted to the samples. Otherwise, if the correlation strength is underestimated (θ_k too large), the local deviation function $Z(x)$ looks like a peak function interpolating each sample and only the polynomial regression model gives information.

$$\text{for } \theta \rightarrow +\infty, \begin{cases} Z(x) \rightarrow 0 \\ Y(x) \rightarrow f^t(x)\beta \end{cases} \quad (16)$$

1. Maximization of Model Likelihood

The correlation parameters (θ_k, p_k) of the model are fitted by Maximizing its Likelihood Estimate (MLE method).

$$MLE = \max_{\theta, p} (\ln(L)) \quad (17)$$

$$\ln(L(\theta, p)) = -\frac{1}{2} \left(n_s (\ln(\sigma^2) + \ln(2\pi)) + \ln(|R|) \right) \quad (18)$$

The model likelihood L evaluation is usually very quick compared to a high-fidelity CFD computation, it only costs to compute the determinant of the correlation matrix R . Maximization of the likelihood estimate can sometimes lead to an incorrect model identifiable by high standard error or by comparing MLE to its asymptotical value.²⁴ This is due to the nature of the MLE function: quasi-null gradient on some zones, and multiple local optima.²⁵ This sometimes high-dimensional problem (from n_{dv} to $2n_{dv}$ parameters according to the correlation function) can be efficiently solved using a gradient optimization algorithm. By using a correct initialization of the correlation parameters as described in the next section (II-C-2), the optimizer converges to the global optimum in most cases. The gradient of the determinant of the correlation matrix is computed with a forward finite differences method.

2. Initialization of the Spatial Correlation Parameters

Estimating the Likelihood requires inverting the correlation matrix and for large datasets, as it is the case in V, the MLE problem becomes computer expensive. A method was developed to easily find an initial guess and the optimization boundaries for the correlation parameters. It enables to optimize the correlation parameters with a gradient optimizer (local) requiring few evaluations of the Likelihood. The parameters θ_k govern the directional strength of the correlation between two locations. The correlation decreases with distance between points. The log-Likelihood function is multimodal and has quasi null gradients for large θ_k .²⁵ The main idea is to define what a large correlation parameter (θ_k) is by introducing a scaling factor depending on the samples. It is then possible to overestimate the correlation strength with a small θ_k in order to directly study the region of the log-Likelihood exhibiting large gradients.

Considering the exponential SCF (Eqn. 6), p_k values close to 2 are best suited to smooth functions but less numerically robust that is why the initial value is $p_k = 1$.

First, assuming that all control points are correlated, the minimum correlation factor c is defined as

$$\forall (x^i, x^j) \in \mathcal{D}^2, \exists c \in]0; 1[/ \exp(-\theta_k |x_k^i - x_k^j|) \geq c \quad (19)$$

Thus, by fixing the strength of the correlation c between the two most directional-distant sample points $(\bar{s}_k^i, \bar{s}_k^j)$

$$\exists \theta_k > 0 / \exp(-\theta_k |\bar{s}_k^i - \bar{s}_k^j|) = c \quad (20)$$

Finally giving

$$\theta_k = \frac{-\ln(c)}{|\bar{s}_k^i - \bar{s}_k^j|} \quad (21)$$

According to our physical functions a good correlation factor is $c = 0.2$. Experience shows that underestimated θ_k parameters usually give better approximation than overestimated parameters, that is why a strong correlation factor was chosen. The initial parameters are determined by

$$\theta_k = \frac{\ln(5)}{|\bar{s}_k^i - \bar{s}_k^j|} \quad (22)$$

The optimization boundaries are $[10^{-3}\theta_k, 10^{+3}\theta_k]$ around the initial guess.

The same method can be applied to cubic spline functions.

D. Cokriging - Gradient Enhanced Kriging

A Cokriging model¹⁹ interpolates the function and the gradient at each sample location. As Cokriging models include more information on the true function than Kriging models, they need fewer samples to achieve a given level of accuracy. Moreover, the vectorial information provided by the gradient should be more beneficial for high-dimensional problems.

Two different formulations exist for Cokriging: direct or indirect. When using the same correlation parameters both formulations give the same results.

1. Indirect Cokriging

The indirect Cokriging method does not require Kriging source code modification. In fact, the original Kriging formulation is used with an augmented sample database. For each sample point, one point determined by a first order Taylor approximation is added in each direction. The size of the augmented sample set is $n_s(n_{dv} + 1)$. The step chosen to add those points must not be too low in order to avoid

ill-conditioning of the correlation matrix. A good compromise is $s^{n_s+ik} = s^i + 10^{-4}range_k(S)$.

$$y(s^{n_s+ik}) = y(s^i) + \frac{\partial y(s^i)}{\partial x_k} 10^{-4}range_k(S) \quad (23)$$

The indirect Cokriging applied to very dense datasets is expected to be problematic. If the delta used for the Taylor expansion is of the same order of magnitude than point density, the Taylor augmented datasets could degrade the model. Hopefully, it does not happen when approximating expensive computer codes.

2. Direct Cokriging

The direct Cokriging approach is equivalent to make one Kriging approximation for each direction of the gradient, and the link between function and gradient is conserved through an heritage of the correlation function and parameters. The vectors and matrix of size n_s are augmented to a size of $n_s(n_{dv} + 1)$.

$$Y_s = \begin{bmatrix} y(s^1), y(s^2), \dots, y(s^{n_s}), \\ \frac{\partial y(s^1)}{\partial x_1}, \frac{\partial y(s^1)}{\partial x_2}, \dots, \frac{\partial y(s^1)}{\partial x_{n_{dv}}}, \\ \frac{\partial y(s^2)}{\partial x_1}, \dots, \frac{\partial y(s^{n_s})}{\partial x_{n_{dv}}} \end{bmatrix} \quad (24)$$

For a constant regression model

$$f(x) = \begin{bmatrix} 1, 1, \dots, 1, \\ 0, 0, \dots, 0, \\ 0, \dots, 0 \end{bmatrix} \quad (25)$$

The augmented correlation matrix is assembled through 4 blocks, representing covariance of functions between functions, of functions between gradients, of gradients between functions and of gradients between gradients. Whereas the indirect formulation approximates the function on an augmented database, the direct formulation keeps the original database unchanged but approximates more functions ($n_{dv} + 1$).

$$\begin{cases} Cov(y(s^i), y(s^j)) = \sigma^2 SCF(s^i, s^j) \\ Cov(y(s^i), \frac{\partial y(s^j)}{\partial x_k}) = \sigma^2 \frac{\partial SCF(s^i, s^j)}{\partial x_k} \\ Cov(\frac{\partial y(s^i)}{\partial x_k}, y(s^j)) = -\sigma^2 \frac{\partial SCF(s^i, s^j)}{\partial x_k} \\ Cov(\frac{\partial y(s^i)}{\partial x_k}, \frac{\partial y(s^j)}{\partial x_l}) = \sigma^2 \frac{\partial^2 SCF(s^i, s^j)}{\partial x_k \partial x_l} \end{cases} \quad (26)$$

One should notice that the augmented correlation matrix is dense non-symmetric and its order $N = n_s(n_{dv} + 1)$ increases with the number of variables and the number of samples.

The 2 blocks of the new correlation vector are

$$r_i(x) = \begin{cases} SCF(x, s^i), [1, n_s] \\ \frac{\partial SCF(x, s^i)}{\partial x_k}, [n_s + 1, n_s(n_{dv} + 1)] \end{cases} \quad (27)$$

With the exponential function the distance (L1-norm) is elevated to a power lesser than two and is not twice differentiable. That is why cubic spline correlation functions are used with direct Cokriging. Derivatives of the cubic spline (Eqn. 7) are

$$\frac{\partial SCF(s^i, s^j)}{\partial x_k} = \begin{cases} \begin{aligned} &sign(s_k^i - s_k^j) \\ &\times (18\theta_k^3 |s_k^i - s_k^j|^2 - 12\theta_k^2 |s_k^i - s_k^j|) \\ &\times \prod_{m \neq k} [1 - 6(|s_m^i - s_m^j| \theta_m)^2 \\ &+ 6(|s_m^i - s_m^j| \theta_m)^3] \end{aligned} \\ -6\theta_k sign(s_k^i - s_k^j) \\ \times (1 - \theta_k |s_k^i - s_k^j|)^2 \\ \times \prod_{m \neq k} [2(1 - 6|s_m^i - s_m^j| \theta_m)^3] \\ 0 \end{cases} \quad (28)$$

Assuming $k \neq l$,

$$\frac{\partial^2 SCF(s^i, s^j)}{\partial x_k \partial x_l} = \begin{cases} \begin{aligned} &\prod_{m \in \{k, l\}} [sign(s_m^i - s_m^j) \\ &\times (18\theta_m^3 |s_m^i - s_m^j|^2 \\ &- 12\theta_m^2 |s_m^i - s_m^j|)] \\ &\times \prod_{m \notin \{k, l\}} [1 - 6(|s_m^i - s_m^j| \theta_m)^2 \\ &+ 6(|s_m^i - s_m^j| \theta_m)^3] \end{aligned} \\ \prod_{m \in \{k, l\}} [-6\theta_m^2 sign(s_m^i - s_m^j) \\ \times (1 - \theta_m |s_m^i - s_m^j|)^2] \\ \times \prod_{m \notin \{k, l\}} [2(1 - 6|s_m^i - s_m^j| \theta_m)^3] \\ 0 \end{cases} \quad (29)$$

$$\frac{\partial^2 SCF(s^i, s^j)}{\partial x_k^2} = \begin{cases} \begin{aligned} &(-12\theta_k^2 + 36\theta_k^3 |s_k^i - s_k^j|) \\ &\times \prod_{m \neq k} [1 - 6(|s_m^i - s_m^j| \theta_m)^2 \\ &+ 6(|s_m^i - s_m^j| \theta_m)^3] \end{aligned} \\ 12\theta_k^2 (1 - \theta_k |s_k^i - s_k^j|) \\ \prod_{m \neq k} [2(1 - 6|s_m^i - s_m^j| \theta_m)^3] \\ 0 \end{cases} \quad (30)$$

3. From Kriging to Cokriging

All Kriging functionalities are available when using Cokriging including error predictions. Cokriging models are also fitted by MLE (Eqn. 17). The MLE correlation parameters of Kriging, indirect Cokriging and direct Cokriging are different in general.

E. Numeric Implementation

1. Correlation Matrix Inversion and Robustness

Considering SCF formulation, close sample points can lead to quasi-identical columns in the correlation matrix R the matrix being then ill-conditioned²⁶ (especially with low θ_k values). The optimization of Kriging parameters implies computing very small determinants, and in order to avoid underflow truncature error the $\ln|R|$ term is computed as a sum of logarithms of eigenvalues with a LU method from LAPACK.²⁷

Typically, one can obtain a fitted Kriging with $\text{cond}(R) \approx 10^{16}$ and still a good interpolation of samples. That is why ill-conditioned matrix is not a stopping error criteria, but sample interpolation is checked after each correlation matrix inversion.

We noticed that when using the same correlation parameters the direct Cokriging correlation matrix (non-symmetric) is better conditioned than the indirect Cokriging correlation matrix (symmetric). It can be explained by the fact that the augmented sampling of the indirect method exhibits samples very close from each other.

2. Memory and Computation Time

The computational cost to build the model depends only on the correlation matrix order N . On a standard Intel Pentium4 2.8GHz processor, the estimated computation time is:

$$\begin{aligned} t_{user} &\approx 3.10^{-9} N^3 \text{ seconds} \\ \text{memsize} &\approx 8.10^{-6} N^2 \text{ Mbytes} \end{aligned} \quad (31)$$

For example, saying that the model must be built in one minute with 300 computations of the likelihood for the MLE fit, the maximum matrix order is then $N = 400$. For models approximating high-fidelity CFD solver the typical maximum number of computations authorized for an optimization is close to $n_s = 200$. The Kriging computation cost depending only on the number of samples ($N_{kriging} = n_s$) it is effectively negligible here. On the contrary, the computation cost of Cokriging models can be limiting on high dimensional problems ($N_{cokriging} = n_s(n_{dv} + 1)$). Considering an optimization problem with a hundred variables, the computation time for a single correlation matrix inversion would be $t_{user} > 6$ hours.

The ScaLAPACK user guide²⁸ (parallel version of LAPACK) recommends a maximum matrix order of 1000 per CPU.

In term of memory, a Kriging code only requires to allocate the inverted correlation matrix plus some vectors in double precision.

III. Definition of the Aerodynamic Test Case

The non-linearities observed on aerodynamic global coefficients when varying shape cannot be represented by an analytical test function. This is why a typical CFD test case was chosen. The aerodynamic function considered within this article is the far-field wave drag²⁹ of a RAE2822 airfoil in an Euler flow at a Mach number $M = 0.78$ and an angle of attack $AOA = 0.6^\circ$. The flow analysis was performed with the elsA CFD code³⁰ on a structured multiblock C-mesh (33×130). The second order Roe's upwind scheme with the Van Albada limiter is used as spatial scheme coupled with an implicit time resolution.

By modifying the shape of the transonic airfoil with Hicks-Henne sinusoidal bumps, the objective function exhibits multiple local optima and a non-linear behavior (Figure 1).

$$HH_k(u) = X_k \sin^4\left(\pi u \frac{\log 0.5}{\log \alpha_k}\right); u \in [0, 1] \quad (32)$$

The six design variables considered are amplitudes X_k of Hicks-Henne bumps on the upper part at $\alpha_k = 10\%, 80\%, 38\%, 66\%, 24\%, 52\%$ of the chord line so as the two first parameters correspond to deformations of the leading edge and of the zone where the shock occurs. The distribution is chosen so as when incrementally increasing the number of design variables, the bumps stay equidistributed on the upper surface. The amplitude of each bump is in the domain $[-5; +5]mm$, and the chord length of the airfoil is one meter.

At each update of the design variables the initial mesh is deformed using an improved integral method (iso-topology). The mesh deformation module is also able to compute the sensitivity of the mesh with respect to the shape variables needed to compute the gradient of the objective function.

The gradient information used to build the Cokriging models was computed using the discrete adjoint method.^{1, 16} In term of computational cost, a gradient calculation with the adjoint method costs about the same as one direct CFD computation, so a Cokriging model built with n_s samples costs the same as a Kriging model on $2n_s$ samples for any number of design variables. It should be noticed that the discrete adjoint formulation used within this paper ensures a coherence between direct flow

computation and gradient computation and thus a correlation between numerical residual errors at convergence.

The dimension of the problem varies from two to six, and for each problem a reference function considered exact is built using respectively 104, 208, 272, 1056 and 4160 samples. The indirect Cokriging formulation was used to build the reference functions, but as the five dimensional reference was already expensive to build with a matrix order $N = 6336$ the six dimensional reference was built with a Kriging formulation instead. The Cokriging code should be made parallel to extend this study to higher dimensions.

Due to the important number of CFD evaluations required to establish references, a coarse mesh was used. It is worth noting that this is not detrimental, since we are addressing inviscid flow simulations and that aerodynamic objective functions on finer meshes or more realistic problems exhibit the same kind of non-linearities.

A Sobol technique was used for the space filling sample distributions and the corners of the box domain were added to the distributions.

Only the first test surface with two variables can be plotted, but about the same level of complexity is expected when increasing the number of design variables.

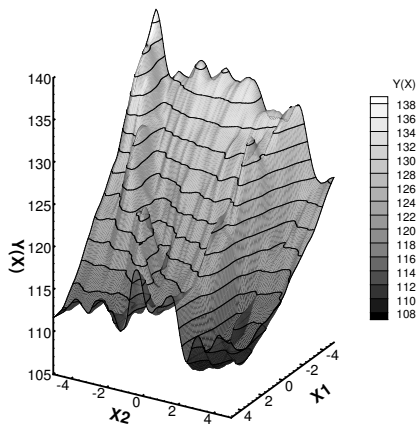


Figure 1. Aerodynamic reference surface for $n_{dv} = 2$

IV. Influence of Sampling Methods on Global Accuracy of Response Surfaces

As predicted errors are available with Kriging we wanted to determine if using them as 'a posteriori' sampling criteria could outperform 'a priori' methods. Maximum error 'a posteriori' sampling were

already applied to polynomial test cases^{10,15}, but without comparison with classical space filling designs.

The different techniques presented in II-A were compared to reconstruct our first reference function depending on two variables (Figure 1) with as few samples as possible. Figures 2 and 3 show the evolution of the exact relative root mean squared error (RMSE) when increasing the number of samples from 6 to 64. The target is the dashed line corresponding to the sampling refinement at maximum of exact error.

A. Performance of 'a priori' Sampling Methods (grid, Halton, Sobol, LHS)

The fact that LHS and grid method are not sequential explains why on Figure 2 the error converges unsmoothly when increasing sampling size. Otherwise, the lack of coherence on the convergence can also be explained by problems of correlation parameter optimization. The optimal parameters in the sense of MLE are not always the optimal parameters in the sense of minimum exact error. More particularly, grid sampling is unadapted to build Kriging because optimization of the directional correlation parameters is biased by the linear dependence existing within the spatial distribution.^{31,32}

Anyhow, regarding Figure 2 one can observe that the convergence of the error seems composed of two phases. Firstly, we observe a transitory phase of strong error reduction for small datasets with a negative power trend until reaching $n_s = 20$. Secondly, the error seems to reach a linear phase with a low negative slope. This linear phase is very costly if a low level of error is required.

The Halton method was taken as the reference for 'a priori' methods for its good overall performance with small and large datasets. However, the only method applicable to high dimensions is LHS.

B. Performance of 'a posteriori' Sequential Sampling Methods

Looking at Figure 3 it appears that all methods are quasi-equivalent to the Halton reference. The linear phase of convergence is not avoided using sampling at maximum predicted error.

Regarding 'a posteriori' methods, the $max(dKdS_{mix})$ error refinement is the best. It is understandable as the $dKdS_{mix}$ error is based on $dKdS$ error, the Kriging sensitivity to the samples. It performs better than the Halton reference for large datasets (Table 1). This is coherent with the fact that as the number of samples increases, the model is more precise and the predicted error is also more precise.

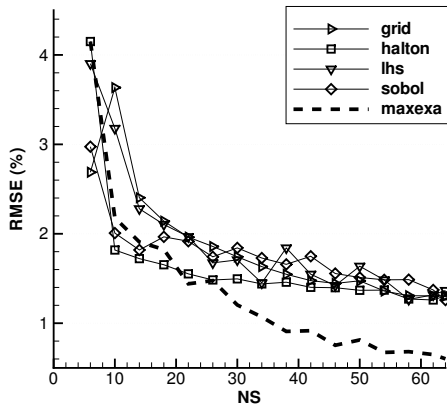


Figure 2. Evolution of error with sampling size for 'a priori' sampling methods, $n_{dv} = 2$

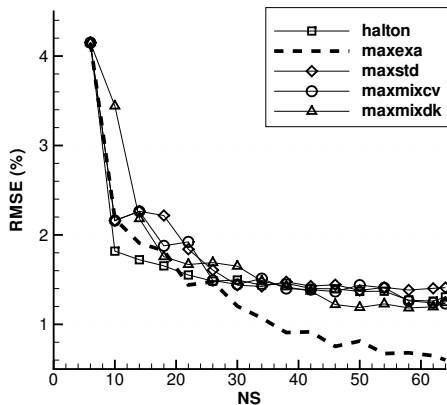


Figure 3. Evolution of error with sampling size for 'a posteriori' sampling methods, $n_{dv} = 2$

On small datasets, sampling refinement at maximum of error (even $max(Exa)$) does not outperform Halton but still produces very satisfying distributions.

When ignoring quasi-random sequence samplings, 'a posteriori' methods outmatch LHS and grid for small and large sampling size.

The fact that 'a posteriori' methods outperform only slightly space filling designs shows that the aerodynamic test function complexity is evenly distributed in the design space. In our sense this is always the case in aerodynamic optimization and global response surface are generally unaffordable.

method	$n_s = 6\dots 25$	$n_s > 25$	$n_s \geq 6$
<i>grid</i>	2.43	1.49	1.92
<i>Halton</i>	1.96	1.39	1.76
<i>LHS</i>	2.45	1.57	2.01
<i>Sobol</i>	2.11	1.61	1.85
<i>max(Exa)</i>	2.17	0.89	1.47
<i>max(S)</i>	2.36	1.43	1.87
<i>max(CV_{mix})</i>	2.32	1.38	1.84
<i>max(dKdS_{mix})</i>	2.43	1.36	1.86

Table 1. Integrated relative RMSE for all sampling methods, $n_{dv} = 2$

V. Influence of Dimension on Global Response Surface Accuracy and Benefits of Cokriging

No general rule can be made to assess a priori the number of samples needed to achieve a given accuracy as it depends on the complexity of the unknown function. It is generally admitted that to obtain a correct global response surface, 10 samples per direction is a good starting sample size ($n_s \approx 10^{n_{dv}}$). The sample size increasing exponentially it becomes then impossible to build a global response surface on high dimensional problems. The benefits of Cokriging beside Kriging cannot be assessed by only varying the sample size^{18,19,21} because the information added to the model through gradients is dimensional. Thus the accuracy of Kriging and Cokriging will be compared by varying the sample size and the number of design variables. The dependence between number of samples and dimension is expected to be reduced with Cokriging. When dealing with approximation of expensive computer analysis, the dataset size is generally very limiting and the reconstruction was made with less than 300 samples. The comparison was conducted on an aerodynamic function (section III) in order to use the same solvers as in aerodynamic shape optimization and gradient computed with the adjoint state formulation.

A. Results

Figures 4-8 show the evolution of the exact (with respect to the reference surface) root mean squared error (RMSE in percent) when increasing the number of samples for the five problems.

Firstly it appears that the indirect Cokriging method is the best. Despite the mathematical appeal of direct Cokriging formulation its performance

are deceiving. Overall, it is worse than Kriging on the two, three and five dimensional tests (Table 2). The spikes occurring on direct Cokriging convergence correspond to wrong MLE correlation parameters. The MLE problem has more local minima representing optimal correlation parameters for each of the $n_{dv} + 1$ functions (the function itself and its derivatives Eqn. 26). In fact, the MLE of direct Cokriging sometimes gives the same parameters as MLE of Kriging and sometimes gives the same parameters as MLE of indirect Cokriging, for example on Figure 7 between 150 and 200 samples. This does not discard the method as direct and indirect Cokriging models give the same results when using the same correlation parameters, but shows that direct Cokriging models fitted by MLE are easily ill-fitted.

The convergence in two phases described previously (in section IV) appears on all tests with Kriging or Cokriging models.

The number of samples needed to switch from transitory to linear phase is roughly the same for Kriging and Cokriging (Figures 4-8). The benefits of the Cokriging models on error is obvious during the transitory phase (Table 2).

During the second phase, the low negative linear slope is the same for Kriging and Cokriging models. However, the indirect Cokriging is still more efficient due to the advance taken during the first phase (Table 2).

In term of computation cost, the indirect Cokriging does effectively need fewer samples to reach a given accuracy. As the samples are twice more expensive to compute for Cokriging (cost of the adjoint solver), they should use at least twice less samples than Kriging to reach a given accuracy to be interesting. This seems true in general. By interpolating the gradient vector at each sample the surrogate models interpolate hyperspheres instead of points and as expected the best gain is achieved on the higher dimensional problem with small datasets.

It appears on table 2 that global error increases with the number of variables. It was expected since it was not possible to have a sampling size of ten samples per direction for all tests.

n_{dv}	$n_s = 3...25$			$n_s > 25$			$n_s \geq 3$		
	Krig.	D.Cok.	I.Cok.	Krig.	D.Cok.	I.Cok.	Krig.	D.Cok.	I.Cok.
2	3.45	2.87	2.42	2.20	1.76	1.57	2.57	2.09	1.82
3	5.23	6.18	3.36	2.44	2.60	1.66	2.84	3.11	1.91
4	8.40	8.16	3.91	3.14	3.58	2.57	3.75	4.11	2.72
5	17.9	15.7	8.68	5.59	6.05	4.73	6.62	6.85	5.05
6	22.8	20.0	11.6	7.25	6.71	6.71	8.45	7.74	7.08

Table 2. Integrated relative RMSE for all formulations and tests

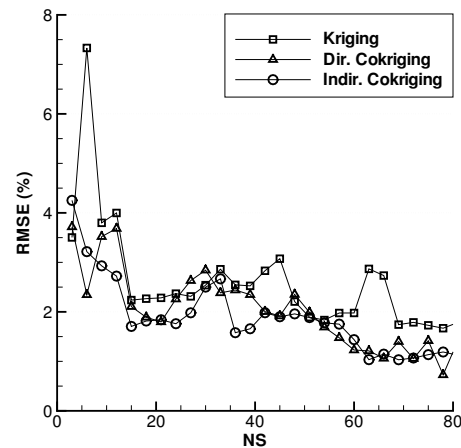


Figure 4. Evolution of Kriging and Cokriging errors with sampling size, $n_{dv} = 2$

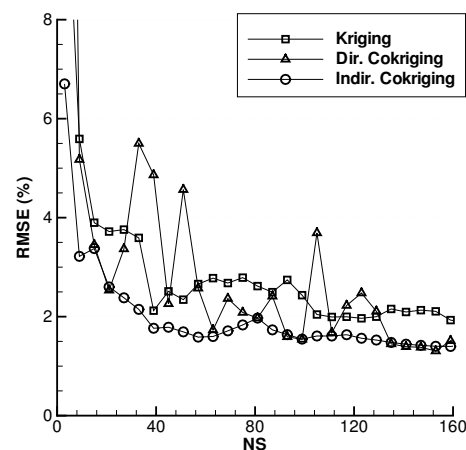


Figure 5. Evolution of Kriging and Cokriging errors with sampling size, $n_{dv} = 3$

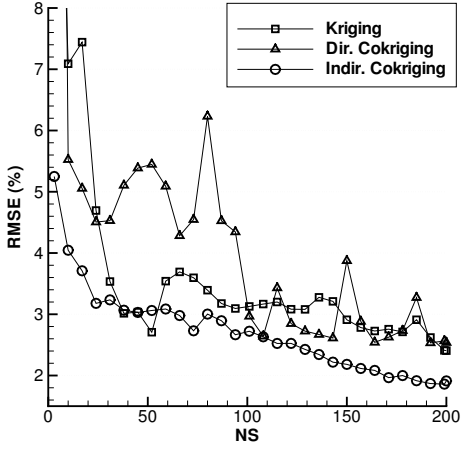


Figure 6. Evolution of Kriging and Cokriging errors with sampling size, $n_{dv} = 4$

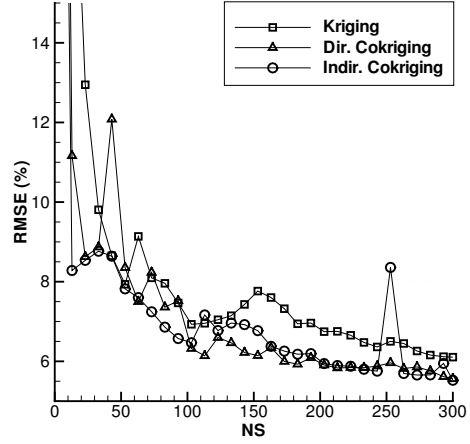


Figure 8. Evolution of Kriging and Cokriging errors with sampling size, $n_{dv} = 6$

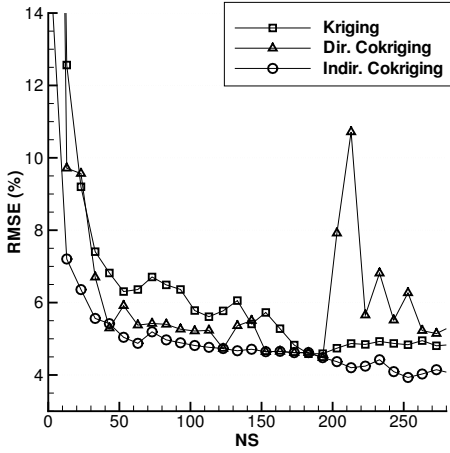


Figure 7. Evolution of Kriging and Cokriging errors with sampling size, $n_{dv} = 5$

VI. Conclusion

An efficient framework to fit Kriging correlation parameters by likelihood optimization was described. It consists in departing from a good initial guess for the parameters and then apply a gradient optimizer. It shows good overall robustness, even during the fit of computer expensive Kriging (up to order $N = 6336$).

Usually, comparison of modeling techniques are made on analytical test functions. Aerodynamic functions being highly non-linear, analytical test cases cannot represent the true level of complexity encountered in aerodynamic shape optimization.

Classical response surface frameworks use space filling 'a priori' sampling. It is possible to tune the

sample distribution to the function studied using 'a posteriori' refinement techniques to concentrate more information into the sampling. Compared to space filling designs, 'a posteriori' sampling at the maximum of exact error significantly increases accuracy of the model. Kriging being an interpolating model, any error criterion null at samples (standard error) can be used for sampling refinement. Two new predicted error criteria based on cross-validation, but satisfying this property were introduced.

The convergence of the global error when increasing the number of samples proceeds in a transitory phase of fast decrease followed by a linear phase of slow decrease. The sample size needed to reach this second phase cannot be foretold, but increases with dimension and complexity of the function studied.

Among 'a priori' techniques, quasi-random Halton give best results. Moreover, it also gives ordered samples. The new sequential sampling methods based on refinement at maximum error show fair results compared to classical sampling methods, but do not break the slow linear convergence as was expected regarding performance of refinement at maximum of exact error. From all predicted errors, the new leave one out sensitivity error introduced within this article enables to have the smallest exact error on large datasets, but needs to build all intermediate models. The fact that 'a posteriori' methods outperform only slightly space filling designs shows that the aerodynamic test function complexity is evenly distributed in the design space. In our sense this is always the case in aerodynamic optimization and global response surface are generally unaffordable.

The use of adjoint gradient information to build

Cokriging models drastically improves global accuracy, but is insufficient to break the slow convergence during the second linear phase. The computation cost induced by Cokriging formulation makes the response surface expensive to build when considering lots of design variables, but as the optimization process usually takes place on parallel computer architecture, it is possible to lessen this cost. As was stated in previous papers,¹⁹ indirect and direct Cokriging models are identical when built with the same correlation parameters. However, the sample database augmentation used to build indirect Cokriging induces bad condition number of the correlation matrix that can be avoided using the direct formulation. Despite its better conditioning, the direct Cokriging model cannot be easily fitted. In our view, the best framework to build gradient enhanced Kriging reuses correlation parameters from the indirect formulation to build a direct Cokriging. The best gain of Cokriging is observed during the transient phase, on small datasets with a large number of design variables.

Regarding the global error of reconstruction on all test cases considered within this paper, it appears that a global response surface accurate at one percent is out of reach with less than a few hundred samples except for dimension lesser than four. That is why, global response surfaces should be used only when it is possible to identify a very small set of interesting design variables or with low-fidelity solvers and large sample database.

However, physical functions generally exhibit a simple global tendency plus non-linear perturbations and it is still possible to find an optimum by iteratively refining the model at its predicted optimum. To overcome early convergence of this process it is interesting to reuse predicted error to introduce some exploration. The general tendencies found on the low-accuracy global response surface can also be used to initialize a local optimizer closer to the global optimum. As the domain of accuracy around sample locations is increased when interpolating gradient, an optimization process refining iteratively a local Cokriging response surface around interesting locations should be investigated on a high dimensional problem.

Acknowledgments

The authors would like to acknowledge M. Meaux (Mathieu.Meaux@airbus.com) and M. Montagnac (Marc.Montagnac@cerfacs.fr) for their support and suggestions. We also thank *Airbus France* for providing their shape optimization suite

OPTaliA. We used the computer implementation of J. Burkardt (<http://www.scs.fsu.edu/~burkardt/>) for 'a priori' sampling methods and thank him for sharing his source code.

References

- ¹M. Meaux, M. Cormery, and G. Voizard. Viscous aerodynamic shape optimization based on the discrete adjoint state for 3d industrial configurations. European Congress on Computational Methods in Applied Sciences and Engineering, Jul. 2004.
- ²Jaroslav Sobieszczanski-Sobieski and Raphael T. Haftka. Multidisciplinary aerospace design optimization: Survey of recent developments. In *4th Aerospace Science Meeting & Exhibit*, number AIAA-1996-0711, 1996.
- ³S. Tursi. Transonic wing optimization combining genetic algorithm and neural network. In *21st Applied Aerodynamics Conference*, number AIAA 2003-3787.
- ⁴S. Pierret, H. Kato, R. Coelho, and A. Merchant. Multi-objective and multi-disciplinary shape optimization. In *EUROGEN*, 2005.
- ⁵S. Jeong, M. Murayama, and K. Yamamoto. Efficient optimization design method using kriging model. *Journal of Aircraft*, 42(2):413–419, 2005.
- ⁶Andrew J. Booker, J. E. Dennis, Jr., Paul D. Frank, David B. Serafini, Virginia Torczon, Trosset, and W. Michael. A rigorous framework for optimization of expensive functions by surrogates. Technical Report TR-98-47, Center for Research on Parallel Computation, 1998.
- ⁷Natalia Alexandrov, J. E. Dennis, Jr., Robert Michael Lewis, Torczon, and Virginia. A trust region framework for managing the use of approximation models in optimization. Technical Report TR-97-50, 1997.
- ⁸D. Cox and S. John. Sdo: A statistical method for global optimization. In *Multidisciplinary design optimization (Hampton, VA, 1995)*, pages 315–329. SIAM, Philadelphia, PA, 1997.
- ⁹D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.
- ¹⁰M. J. Sasena. *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*. PhD thesis, University of Michigan, 2002.
- ¹¹J.C. Jouhaud, P. Sagaut, and B. Labeyrie. A kriging approach for CFD/Wind tunnel data comparison. *Journal of Fluid Engineering*, 128:847–855, 2006.
- ¹²J.-C. Jouhaud, P. Sagaut, M. Montagnac, and J. Laurenceau. A surrogate model based multi-disciplinary shape optimization method with application to a 2d subsonic airfoil. *Computers and Fluids*, 36(3):520–529, 2007.
- ¹³T. W. Simpson, J. D. Peplinski, P. N. Koch, and J. K. Allen. Metamodels for computer-based engineering design: Survey and recommendations. *Engineering with Computers*, 17:129–150, 2001.
- ¹⁴T. W. Simpson, D. K. Lin, and W. Chen. Sampling strategies for computer experiments: Design and analysis. *Journal of Reliability and Applications*, 2:209–240, 2001.
- ¹⁵W. C. M. van Beers. *Kriging Metamodeling for Simulation*. PhD thesis, Universiteit van Tilburg, 2005.
- ¹⁶A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3), 1988.
- ¹⁷J. Peter, Chi-Tuan Pham, and F. Drullion. Contribution to discrete implicit gradient and discrete adjoint method for aerodynamic shape optimisation. ECCOMAS, 2004.
- ¹⁸R. M. Lewis. Using sensitivity information in the construction of kriging models for design optimization. *AIAA Journal*, (AIAA-98-4799):730–737, 1998.

¹⁹W. Liu. *Development of Gradient-Enhanced Kriging Approximations for Multidisciplinary Design Optimization*. PhD thesis, University of Notre Dame, 2003.

²⁰H.-S. Chung and J. J. Alonso. Using gradients to construct cokriging approximation models for high dimensional design optimization problems. In *40th AIAA Aerospace Science Meeting & Exhibit*, number AIAA-2002-0317, 2002.

²¹H.-S. Chung and J. J. Alonso. Design of a low-boom supersonic business jet using cokriging approximation models. In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, number AIAA-2002-5598, 2002.

²²J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989.

²³L. Kocis and W. J. Withen. Computational investigations of low-discrepancy sequences. *ACM Transactions on Mathematical Software*, 23(2):155–294, 1997.

²⁴J. D. Martin. *A Methodology for Evaluating System-Level Uncertainty in the Conceptual Design of Complex Multidisciplinary Systems*. PhD thesis, Pennsylvania State University, 2005.

²⁵K. V. Mardia and A. J. Watkins. On multimodality of the likelihood in the spatial linear model. *Biometrika*, 2:289–295, 1989.

²⁶G. J. Davis and M. D. Morris. Six factors which affect the condition number of matrices associated with kriging. *Mathematical Geology*, 5:669–683, 1997.

²⁷E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*, third edition, 1999.

²⁸L. S. Blackford, J. Choi, A. Cleary, E. Dázevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley. *ScaLAPACK Users' Guide*, 1997.

²⁹D. Destarac. Far-field / near-field drag balance and applications of drag extraction in cfd. In *CFD-Based Aircraft Drag Prediction and Reduction*. von Karman Institute for Fluid Dynamics, 2003.

³⁰L. Cambier and M. Gazaix. elsA: an efficient object-oriented solution to cfd complexity. In *40th AIAA Aerospace Science Meeting & Exhibit*, number AIAA-2002-0108, 2002.

³¹M. Meckesheimer, R. B. Barton, T. W. Simpson, F. Limayem, and B. Yannou. Metamodeling of combined discrete/continuous responses. *AIAA Journal*, 39(10):1950–1959, 2001.

³²B. Wilson, D. J. Cappelleri, T. W. Simpson, and M. I. Frecker. Efficient pareto frontier exploration using surrogate approximations. In *8th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, number AIAA 2000-4895.

II.3 Compléments

II.3.1 Krigeage et corrélation spatiale

Le Krigeage vient de l'interprétation de la fonction à modéliser en tant que processus stochastique, $Z(x)$ (Equation II.Article.1). Il permet d'évaluer la fonction en un point inconnu en réalisant uniquement des calculs de corrélations entre couples de points (Equations II.Article.4, II.Article.8 et II.Article.9). L'hypothèse fondamentale, provenant des origines géostatistiques du Krigeage, est d'établir que la corrélation entre deux points est directement liée à leur distance par une fonction de corrélation spatiale (Equation II.Article.5). Ainsi, le réglage du Krigeage consiste à déterminer des paramètres de corrélation afin que les fonctions de corrélation représentent les relations les plus vraisemblables entre les échantillons (Equation II.Article.17). Une bonne compréhension du comportement des fonctions de corrélation spatiale est donc essentielle pour bien appréhender le Krigeage.

La fonction de corrélation, $scf_k(x)$, varie entre zéro et un en fonction de la distance, positive ou nulle, entre deux points. Elle est construite telle que deux points infiniment distants ne soient pas corrélés (fonction de corrélation nulle) et deux points identiques (distance nulle) aient une corrélation unitaire.

$$\begin{cases} \forall x \geq 0, scf_k(x) \in [0,1], \\ scf_k(0) = 1, \\ scf_k(x) \xrightarrow{x \rightarrow \infty} 0. \end{cases} \quad (\text{II.1})$$

C'est au niveau de la fonction de corrélation qu'interviennent les paramètres de corrélations, θ_k (Equations II.Article.6 et II.Article.7), qui permettent de régler le Krigeage en déterminant la vitesse selon laquelle la fonction tend vers zéro (Figures II.1 et II.2). Ces paramètres sont analogues aux hyperparamètres réglant le rayon d'influence des fonctions de base des réseaux de neurones à fonctions radiales de base. Ils établissent la sensibilité du métamodèle à chacune des variables. Nous utiliserons deux types de fonctions décrits ci-dessous :

- fonction de corrélation spatiale de type exponentiel,
- fonction de corrélation spatiale de type spline cubique.

La fonction de corrélation spatiale de type exponentiel (Equation II.Article.6 et Figure II.1) tend vers zéro sans jamais l'atteindre de sorte que tout couple de points du domaine d'étude présente une corrélation non nulle. Cette fonction fait intervenir un deuxième vecteur de paramètres de corrélation, p_k , qui agit sur le lissage de la fonction. La limite inférieure pour la puissance, $p_k = 1$, correspond à une fonction dont la dérivée est discontinue (fonction valeur absolue).

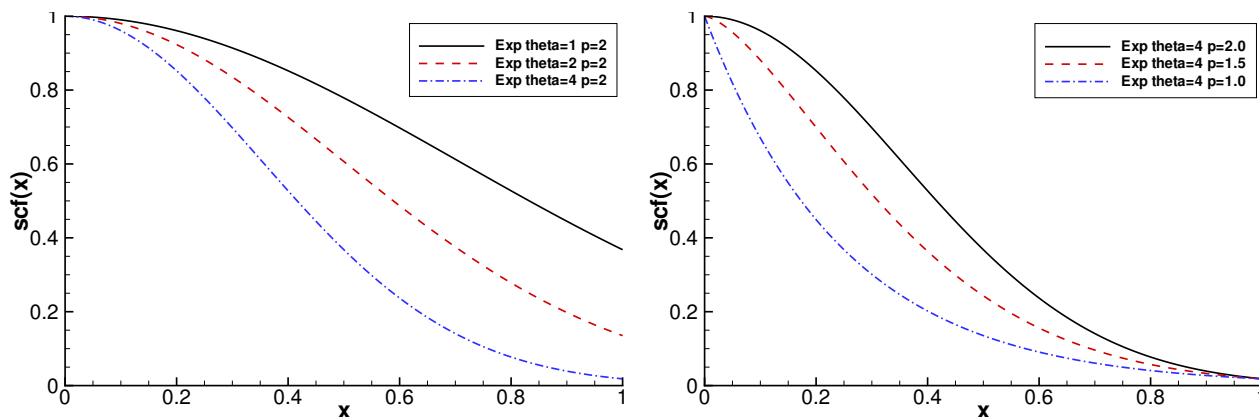


FIG. II.1: Influence des paramètres de corrélation θ, p de la fonction exponentielle.

La fonction de corrélation spatiale de type spline cubique est définie sur trois intervalles déterminés

par θ_k (Equation II.Article.7 et Figure II.2).

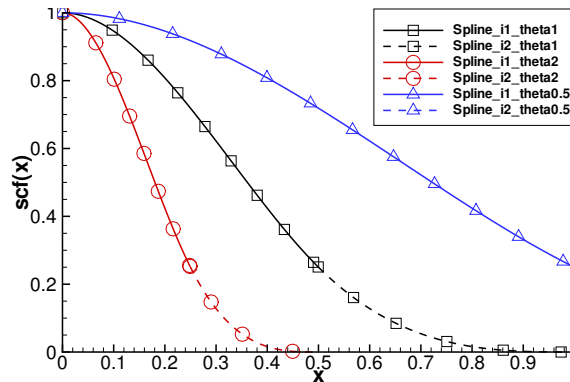


FIG. II.2: Influence du paramètre de corrélation θ de la fonction spline cubique.

Les paramètres de corrélation spatiale du modèle, θ_k , doivent être réglés à chaque échantillonnage. En effet, lorsque la corrélation entre les échantillons est sous-estimée (surestimation de θ_k), la fonction de déviation locale, $Z(x)$, tend trop vite vers zéro et le Krigage, $Y(x)$, ressemble alors au modèle de régression plus une série de fonctions de Dirac interpolant chaque échantillon (Equation II.Article.16 et Figure II.3).

Le coût de calcul d'une évaluation de cette fonction correspond au coût de calcul du déterminant de la matrice de corrélation, R (cf. II.Article.II.E).

Tel que suggéré par Mardia et Marshall [79], ce problème d'optimisation est traité par un algorithme local de type quasi-Newton. Toutefois, il est établi [80, 118] que la fonction de vraisemblance peut être multi-modale et admettre des zones plateau (où le gradient est quasi nul) proches de son optimum. Ainsi, comme le suggère Gano *et al.* [38], un algorithme global comme le recuit simulé est théoriquement préférable même s'il implique un plus grand nombre d'évaluations de la fonction objectif (d'autant plus que le nombre de paramètres est important).

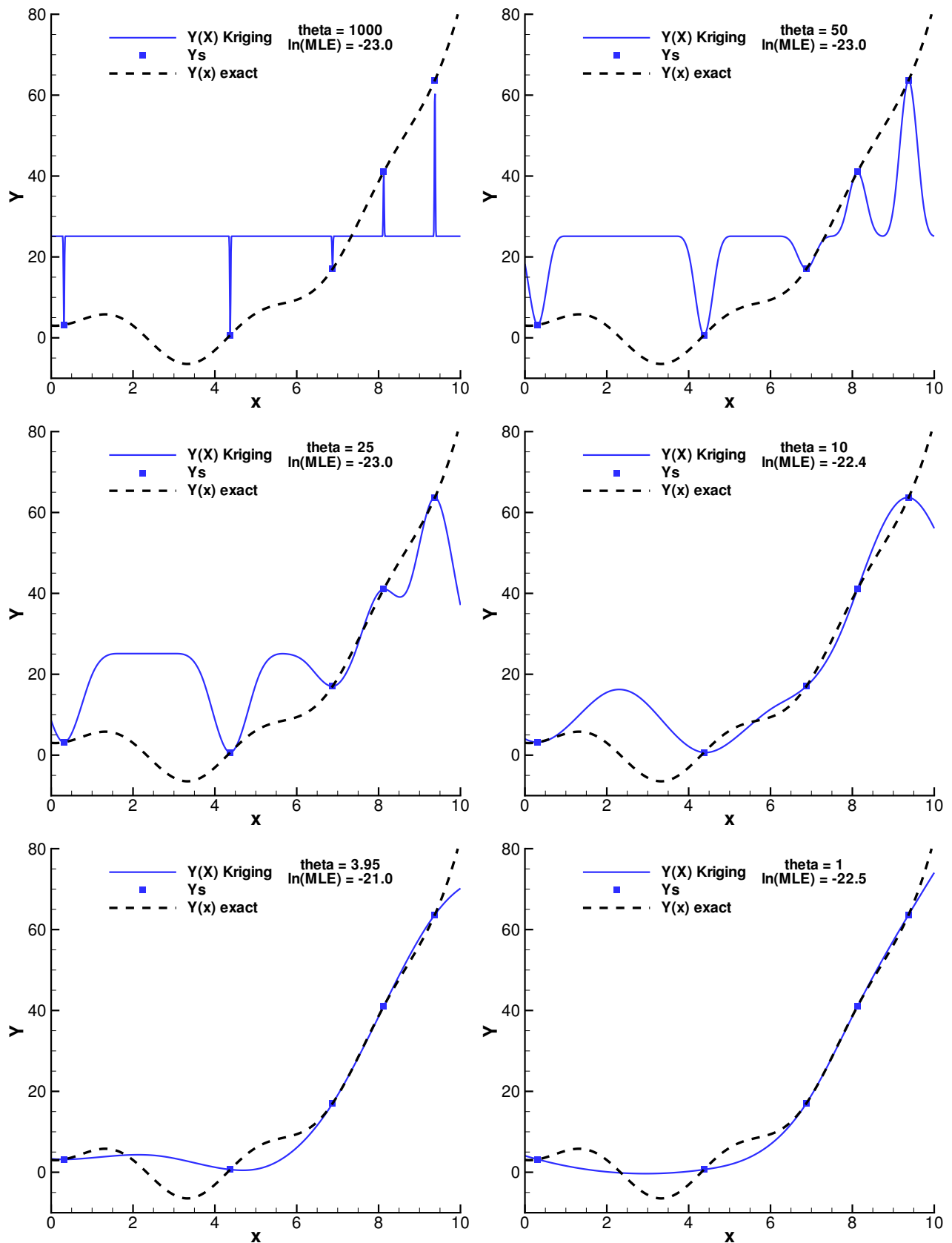
L'approche mise en place dans cette thèse pour faire face à ce problème est d'utiliser une méthode empirique de détermination des paramètres de corrélation (cf. II.Article.II.C.2) afin d'initialiser l'algorithme local par un jeu de paramètres proche de l'optimum recherché. Cette valeur initiale correspond à la courbe $theta=10$ de la figure II.3.

II.3.2 L'échantillonnage

II.3.2.1 Taille de l'échantillonnage

Pour le Krigage, un point de construction est une déviation par rapport au modèle de régression (terme constant $Z(x)$). Un point de construction peut donc s'interpréter comme un degré de liberté laissé au Krigage. L'augmentation du nombre d'échantillons permet de modéliser des fonctions de plus en plus complexes. Le nombre d'échantillons nécessaires pour correctement modéliser une fonction est donc déterminé par sa complexité. Cependant, comme la fonction à modéliser est inconnue, sa complexité l'est aussi. Il est donc impossible de déterminer *a priori* la taille d'échantillonnage nécessaire pour avoir une surface de réponse précise. Il existe tout de même quelques règles.

En général, les fonctions que l'on cherche à modéliser sont complexes, elles ne dépendent pas linéairement de leurs variables. On peut donc supposer que pour les modéliser, il faudra au moins autant d'échantillons que pour construire un modèle quadratique, c'est-à-dire $n_s \geq \frac{(n_{dv}+1)(n_{dv}+2)}{2}$ échantillons. En pratique, cela correspond à la limite inférieure de la taille d'échantillonnage. Théoriquement, le Krigage peut être construit avec moins d'échantillons, mais il risque alors d'être imprécis. En dessous

FIG. II.3: Optimisation des paramètres de corrélation θ du Krigage.

de cette taille, il conviendra donc de considérer le Krigeage comme une surface de réponse locale fiable seulement en ses points de construction.

Pour des fonctions de complexité inconnue, la taille d'échantillonnage recommandée est d'au moins dix points par direction, soit $n_s \geq 10^{n_{dv}}$. Cela n'est pas une taille maximale, mais une taille standard permettant d'au moins détecter les tendances et non linéarités principales. A partir de cette taille d'échantillonnage, on pourra envisager de remplacer complètement la fonction par sa surface de réponse. Il devient possible de considérer le Krigeage comme une surface de réponse globale. Cette dépendance exponentielle entre la dimension et la taille de l'échantillonnage correspond à la malédiction dimensionnelle de Bellman [10]. Mis à part en dimension deux (10^2 calculs) ou trois (10^3 calculs), une telle densité d'échantillons est en générale impossible à atteindre.

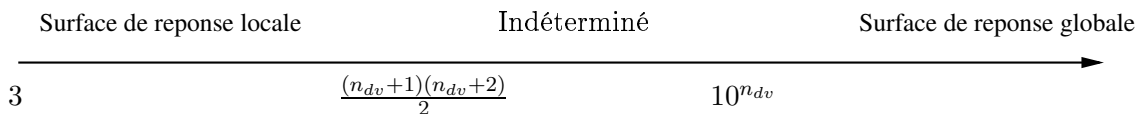


FIG. II.4: Recommandations sur la taille de l'échantillonnage.

II.3.2.2 Echantillonnage global

L'utilisation de méthodes d'échantillonnage séquentielles *a posteriori* (cf. II.Article.II.A) peut potentiellement apporter un gain de précision globale significatif par rapport aux méthodes *a priori* pour la construction de surfaces de réponse. Cela est vérifié par le critère de raffinement d'échantillonnage au maximum d'erreur exacte (cf. II.Article.IV). Un tel gain pourrait permettre d'utiliser moins d'échantillons, donc potentiellement moins de simulations numériques, pour atteindre un niveau de précision donné. Toutefois, il s'avère que dans notre cas, le gain apporté par les critères de raffinement d'échantillonnage au maximum d'erreur prédite est négligeable. Or, ces méthodes séquentielles *a posteriori* nécessitent la construction de chaque surface de réponse intermédiaire et sont donc contraignantes d'un point de vue pratique. En conséquence, les méthodes d'échantillonnage *a priori* (hypercube latin LHS, Sobol) ont finalement été choisies pour la construction de surfaces de réponse globales.

Afin d'éviter la dégénérescence des performances des méthodes de Sobol ou Halton en dimension élevée [68], la méthode LHS est utilisée lorsque la dimension du domaine d'étude dépasse 20. Cette dégénérescence est illustrée par l'exemple de la figure II.5. Elle représente 100 échantillons distribués selon trois méthodes différentes (Halton, Sobol, LHS) dans un espace de dimension 40. En projetant ces 100 échantillons sur un plan de coupe, il apparaît clairement que les méthodes Halton et Sobol ne parviennent pas à distribuer uniformément les échantillons contrairement à la méthode LHS.

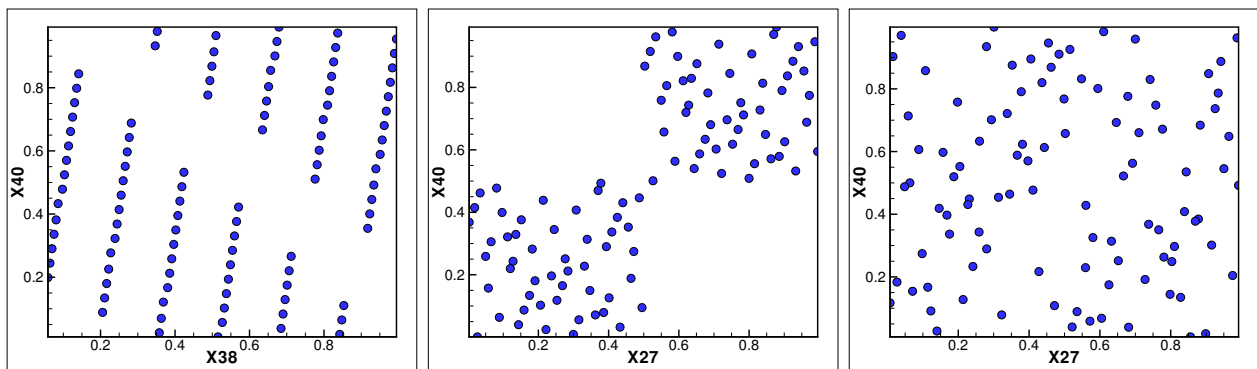


FIG. II.5: Projection bidimensionnelle de 100 points des distributions Halton, Sobol et LHS dans un domaine de dimension 40.

II.3.2.3 Critères d'échantillonnage pour la recherche de minimum

Lorsque la surface de réponse est utilisée au sein d'un processus d'échantillonnage séquentiel, dans le but de trouver le minimum de la fonction, il n'est pas nécessaire de rechercher une bonne précision globale. En effet, le minimum prédit s'impose alors comme critère de raffinement logique. Pour ne pas négliger les zones où la fonction prédite présentent une grande incertitude, il est possible d'utiliser des critères plus évolués. En établissant un équilibre entre exploration (grâce à la valeur de l'incertitude) et exploitation (grâce à la valeur de la fonction) ces critères permettent d'éviter une convergence prématurée. En se conformant à la nomenclature décrite en II.2, $Y(x)$ est la fonction prédite et $S(x)$ est l'erreur prédite.

II.3.2.3.1 Critère LCB La manière la plus évidente pour effectuer ce compromis est d'effectuer une combinaison linéaire entre fonction approchée et incertitude. Ainsi, Cox [25] a introduit le critère *LCB*, « Lower Confidence Bounding »,

$$LCB_b(x) = Y(x) - b \cdot S(x); b \in \mathcal{R}^*. \quad (\text{II.2})$$

Son minimum indique la position à échantillonner. Comme l'erreur prédite par le Krigeage, $S(x)$, est positive ou nulle, le coefficient de pondération d'erreur b peut être choisi positif ou négatif.

II.3.2.3.2 Critère EI Le critère *EI* (« Expected Improvement »), utilisé notamment par l'algorithme d'optimisation *EGO* de Jones [59], est répandu en optimisation aérodynamique de formes [54, 61, 83]. Ce critère calcule l'espérance d'amélioration du Krigeage par rapport au meilleur des échantillons, $\max(Y_s^{\min} - Y(x); 0)$. Pour ce faire, il suffit de traiter le Krigeage comme une variable aléatoire suivant une loi normale de moyenne $Y(x)$ et d'écart type $S(x)$. Les fonctions *cdf* et *pdf* désignant la fonction de répartition et la densité de la loi normale, le critère *EI* vaut,

$$EI(x) = (Y_s^{\min} - Y(x)) \cdot cdf\left[\frac{Y_s^{\min} - Y(x)}{S(x)}\right] + S(x) \cdot pdf\left[\frac{Y_s^{\min} - Y(x)}{S(x)}\right]. \quad (\text{II.3})$$

Son maximum indique la position à échantillonner. Cette espérance d'amélioration présente souvent des zones de type plateau (gradient nul) et un comportement de type fonction Dirac. Son maximum est donc parfois difficile à trouver.

II.3.2.3.3 Critère LCI La fonction de confiance d'amélioration, *LCI* (« Lower Confidence Improvement »), a été développée dans le même esprit que l'espérance d'amélioration, mais en utilisant le formalisme de l'optimisation plutôt que le formalisme statistique. Il a été développé pendant la thèse, mais son application n'est pas décrite dans ce document. L'idée est d'introduire une contrainte telle que l'amélioration prédite soit supérieure à l'erreur prédite. Cette contrainte est directement introduite dans le critère par un terme de pénalisation de type puissance.

$$LCI_b(x) = -Y(x) + \max(Y(x) - Y_s^{\min} - b \cdot S(x), 0)^{1.2}; b \in \mathcal{R}^*. \quad (\text{II.4})$$

Son maximum indique la position à échantillonner. Cette fonction d'amélioration est beaucoup plus régulière que l'espérance d'amélioration.

II.3.3 Cokrigeage et interpolation du gradient

Dans OPTaliA, la méthode adjointe (cf. I.3.5.2) permet de calculer le gradient des fonctions aérodynamiques par rapport aux variables de forme à un coût de calcul comparable à une simulation directe. L'inclusion de cette information vectorielle dans la base d'échantillons (II.Article.II.D) permet

d'améliorer significativement la précision globale des surfaces de réponse (II.Article.V), notamment lorsque la dimension du domaine d'étude est grande. Cependant, alors que le coût de calcul du Krigage ne dépend que du nombre d'échantillons, n_s , le Cokrigage direct comme le Cokrigage indirect ont un coût croissant avec le nombre de variables (Equation II.Article.II.E.2). De ce fait, le coût de construction du Cokrigage limite l'étendue des problèmes pouvant être abordés (dimension inférieure à 10). Pour contourner cette limitation, plusieurs solutions sont possibles.

II.3.3.1 Parallélisation du Cokrigage

Une parallélisation de l'inversion de la matrice de corrélation permettrait de garder un temps de calcul constant en augmentant le nombre de processeurs lorsque la dimension du problème croît. Cette voie n'a pas pu être explorée dans le cadre de cette thèse.

II.3.3.2 Cokrigage indirect tronqué en direction

Dans sa thèse, Liu [76] recommande d'utiliser un Cokrigage indirect tronqué en direction. Au lieu de rajouter en chaque échantillon un point par direction comme pour le Cokrigage indirect, cette formulation ne prend pas en compte toutes les directions du vecteur gradient mais seulement une seule direction par échantillon, quelque soit la dimension du vecteur gradient. Le coût de calcul reste donc maîtrisé, mais une grande partie de l'information disponible est inutilisée. Son argument principal vient du fait que l'agglomération de points lors de l'augmentation de la base d'échantillons du Cokrigage indirect induit un mauvais conditionnement de la matrice de corrélation (Equation II.Article.II.E.1). A en juger par les trois cas tests analytiques en dimension 1 à 15 effectués, ce mauvais conditionnement rend le Cokrigage indirect tronqué en direction plus précis que le Cokrigage indirect complet.

II.3.3.3 Cokrigage indirect tronqué en échantillon

La formulation de Cokrigage indirect tronqué en échantillon a été établie dans le cadre de cette thèse à travers les travaux présentés au chapitre IV. Cela consiste à inclure l'information complète du gradient, mais seulement en un nombre fixé, n_{aug} , d'échantillons appelés échantillons augmentés. Cette formulation a deux avantages :

- le coût de calcul du Cokrigage reste maîtrisé en agissant sur le nombre d'échantillons augmentés,
- le coût de calcul de l'évaluateur est diminué, car le calcul du gradient n'est pas nécessaire en tous les échantillons.

A partir de la règle recommandant une taille d'échantillonnage d'environ 10 points par direction (cf. II.3.2.1), il semble qu'une bonne valeur par défaut est de choisir le nombre d'échantillons augmentés égale à 10 ($n_{aug} = 10$). Par commodité, les échantillons qui sont augmentés sont les n_{aug} premiers de la base de données. Toutefois, il est possible de déterminer des schémas d'augmentation pour choisir avec précaution les échantillons qui seront augmentés. Par exemple, en II.Article.II.B.2 deux critères d'erreur non nuls aux points de construction ont été décrit. Il est donc possible d'augmenter les échantillons pour lesquels l'erreur semble maximale. Aussi, lors d'un processus de recherche de minimum, il est possible de choisir les échantillons à augmenter en prenant ceux pour lesquelles la fonction est minimale.

II.4 Bilan

Cette étude a tout d'abord permis d'implémenter une méthode d'interpolation de fonctions par Krigage robuste. D'autre part, différents moyens d'accroître la précision globale (erreur moyenne sur le domaine) d'une surface de réponse ont été comparés.

Il semble que l'utilisation de méthodes d'échantillonnage itératives n'apportent pas de gain significatif par rapport aux méthodes classiques de remplissage d'espace. De plus, aucune méthode ne permet d'obtenir une erreur moyenne inférieure à un pourcent, même en considérant une densité de huit points de construction par direction (64 échantillons en dimension deux).

L'interpolation de vecteurs gradients par une méthode de Cokrigage améliore nettement la précision. Toutefois, cette amélioration s'observe principalement pour les échantillonnages très peu denses, lorsque l'erreur globale est élevée. Comme le coût de calcul du Cokrigage augmente avec la quantité d'information contenue dans la base de données d'échantillons, il devient impossible de le calculer lorsque la dimension est grande. Pour pallier ce coût, il faut limiter la quantité d'information interpolée, soit en utilisant seulement certains coefficients du vecteur gradient (Cokrigage indirect tronqué en direction), soit en utilisant le vecteur gradient en un nombre très restreint d'échantillons (Cokrigage indirect tronqué en échantillon).

Finalement, la seule solution pour obtenir une surface de réponse globale (fiable sur tout le domaine) semble donc d'utiliser un nombre considérable d'échantillons. Il n'est pas possible d'établir une relation générale indiquant un nombre minimal d'échantillons en fonction de la dimension du problème. Des recommandations ont toutefois été décrites, telle que la règle empirique recommandant une taille minimale de 10 points par direction.

Lorsque cette règle n'est pas respectée, il faudra alors prendre soin de valider les résultats obtenus à partir de la surface de réponse. Pour cela, la prédiction d'erreur pourra être utilisée (cf. III). Aussi, lorsqu'il est possible d'identifier *a priori* certaines zones privilégiées, une distribution non uniforme pourra permettre de concentrer l'information là où elle est nécessaire [60]. Enfin, la mise en place d'un processus d'échantillonnage itératif *a posteriori* dans le but de trouver précisément le minimum de la fonction sera considéré au chapitre IV.

Optimisation fluide structure biniveau par surfaces de réponse globales

Nomenclature

C	Corde locale de l'aile
Cd	Coefficient de traînée
Cd_p	Coefficient de traînée de pression
Cl	Coefficient de portance
Cl_0	Coefficient de portance de référence
\mathcal{D}	Domaine d'étude des variables
F	Vecteur des fonctions objectifs
k	Coefficient d'équivalence masse/traînée
J	Fonction objectif (scalaire)
n_α	Nombre de variables, dimension de \mathcal{D}
n_G	Nombre de fonctions contraintes
n_{ss}	Nombre de sous-systèmes
n_s	Nombre d'échantillons
R	Système d'équations
S	Ensemble des échantillons
X	Position axiale de l'installation motrice
Y	Variables d'état
W	Largeur du mât
Z	Position verticale de l'installation motrice
α	Vecteur des variables de conception (n_α)
σ	Contrainte structurale
θ	Vecteur d'équivalence (aérodynamique ; structure)

Indices

<i>aero</i>	Sous-système aérodynamique
<i>glob</i>	Niveau global ou système
<i>i</i>	$\in [1, n_{ss}]$
$-i$	$\in [1, i - 1] \cup [i + 1, n_{ss}]$
<i>struct</i>	Sous-système structure

Exposants

ˆ	Valeur approchée
*	Valeur optimale

Le travail présenté ici est issu d'une collaboration entre l'ONERA, le CERFACS et Airbus dans le cadre du projet européen VIVACE (« Value Improvement through a Virtual Aeronautical Collaborative Enterprise », <http://www.vivaceproject.com/>) sur la conception collaborative ayant pour but d'améliorer l'efficacité du processus de conception d'un avion par la mise en œuvre de passerelles multidisciplinaires et d'outils d'optimisation numérique de formes.

Le composant étudié est une installation motrice à fort taux de dilution d'un avion de transport de type Airbus A3XX. L'intégration de ce composant sous une aile nécessite une étroite collaboration entre les domaines aérodynamique et structure (cf. III.1) afin de concevoir la configuration générant la pénalisation de traînée/portance et de masse minimale. Après une rapide introduction à l'optimisation multidisciplinaire (cf. III.2), une application est décrite (cf. III.3). Sylvain Mouton a présenté ces résultats lors du 42^{ème} colloque d'Aérodynamique Appliquée AAAF en 2007 [87].

III.1 Intégration aérodynamique d'une installation motrice

Afin d'améliorer la consommation spécifique de carburant¹ (« SFC, Specific Fuel Consumption »), l'avionneur peut agir sur la masse et l'efficacité aérodynamique de l'avion. De son côté, le motoriste propose des moteurs de plus en plus performants, notamment par une augmentation du taux de dilution. Cependant, l'augmentation de ce taux se traduit par un accroissement du diamètre du moteur. Ainsi, le diamètre des turboréacteurs actuels représente environ 50% du diamètre du fuselage pour un taux de dilution d'environ 9 : 1.

Or, le moteur est destiné à être installé sous une voilure basse, c'est-à-dire une voilure positionnée près du sol. Pour des raisons de garde au sol minimale et de surpoids d'un allongement du train d'atterrissage, cet encombrement important implique une grande proximité entre la nacelle et la voilure, d'où des interférences aérodynamiques importantes. Ainsi, Pendergraft *et al.* [93] et Godard *et al.* [40] ont mis en évidence expérimentalement sur des configurations de type avion de transport civil biréacteur une pénalisation de traînée au point de croisière (défini en Mach et portance) par rapport à la configuration lisse² de l'ordre de 5% à 15% selon la position et le diamètre de l'installation motrice considérée. Or, cet incrément de traînée ne correspond pas à la somme des traînées de la configuration lisse seule et de l'installation motrice isolée [6, 71, 93],

$$C_d^{conf. motorisee} \neq C_d^{conf. lisse} + C_d^{inst. motrice isolee}. \quad (III.1)$$

Il faut donc considérer un terme de traînée d'interférence tel que,

$$C_d^{interf.} = C_d^{conf. motorisee} - C_d^{conf. lisse} - C_d^{inst. motrice isolee}. \quad (III.2)$$

Cette production de traînée s'explique par le fait qu'en aérodynamique une modification locale de la forme va modifier globalement l'écoulement. L'intégration de l'installation motrice sous la voilure impacte en premier lieu l'écoulement sur l'intrados de l'aile. Cette forme défléchit l'écoulement et le mât le bloque dans la direction de l'envergure. Le jet du moteur modifie lui aussi l'écoulement intrados. En second lieu, l'écoulement l'extrados est lui aussi modifié, notamment en raison de la position de

¹Poussée par unité de volume par unité de temps.

²La configuration lisse comporte simplement le fuselage et les surfaces portantes.

l'entrée d'air en amont de l'aile [18]. Ainsi, l'ajout de l'installation motrice impacte la portance ainsi que la structure et la position du choc sur l'extrados de la voilure.

De plus, la proximité verticale entre la nacelle et la voilure peut générer une traînée importante par le biais de zones d'écoulements supersoniques et de décollements au niveau du mât. Les aérodynamiciens portent donc un intérêt particulier à la conception de la forme de la jonction mât/voilure [75], aux formes du mât [49] et à la position axiale, verticale et en envergure de la nacelle [37].

Afin de limiter la perte de pression sur l'intrados de la voilure engendrée par l'installation motrice, Ingraldi, Naik et Pendergraft [49] ont ainsi proposé un profil de mât dont l'épaisseur décroît avec l'épaisseur du profil d'aile, c'est-à-dire croît dans la direction axiale de l'écoulement. Le profil de ce mât ressemble, grossièrement, à un profil d'aile qui aurait été orienté dos à l'écoulement (bord de fuite face à l'écoulement). Toutefois, ce mât de compression à épaisseur minimale dans la zone amont n'est pas forcément compatible avec les exigences structurales. En effet, l'enveloppe externe du mât est un capot autour de la structure primaire dont le rôle est de résister aux contraintes engendrées par la poussée du turboréacteur. La largeur de cette structure primaire (notée W) dicte ainsi la largeur minimale du carénage externe du mât. Or, cette largeur est un paramètre agissant directement sur le poids de la structure primaire. C'est une variable de forme couplée aérodynamique-structure, comme le sont les positions axiale (notée X) et verticale (notée Z) de la nacelle.

L'intégration d'une installation motrice à une configuration portante est donc un problème multidisciplinaire. Dans le cadre de cette thèse nous traiterons seulement des interactions entre deux disciplines : l'aérodynamique et la structure. Nous étudierons l'influence de variables de forme couplées, la position de la nacelle, sur ces deux disciplines. Pour cela, une méthode d'optimisation multidisciplinaire est employée.

III.2 Optimisation multidisciplinaire

III.2.1 Introduction

Le cycle de conception d'un avion dure plusieurs années (environ 5 ans) et se décrit classiquement comme une succession de trois phases : avant-projet, conception préliminaire et conception détaillée. Le processus de conception regroupe ces trois phases et mène à fixer des paramètres de conception plus ou moins détaillés. La définition de l'avion s'affine ainsi progressivement avec l'avancement du projet. L'architecture globale de l'avion est d'abord décrite, puis les composants sont définis précisément et finalement toutes les pièces sont décrites.

Lors de la phase avant-projet, la faisabilité technique et économique de nouveaux concepts est étudiée. Les études de marchés permettent de déterminer les missions types que devra remplir le nouvel avion afin d'être concurrentiel. Cela aboutit ensuite à un cahier des charges précisant notamment le nombre de passagers, la quantité de fret transportable, la distance franchissable et les coûts d'exploitation. A partir de ces spécifications une configuration générale de l'avion est établie sous la forme d'un plan trois vues (haut, face, côté). Ce plan trois vues (ou forme en plan) fixe les dimensions principales de l'avion (dimension du fuselage, envergure, angle de flèche et rapport d'aspect des ailes, etc.), il définit l'architecture globale de l'avion. La forme en plan est déterminée à l'aide de modèles simplifiés et du savoir-faire de l'entreprise. Les données issues des modèles basse-fidélité sont ainsi recalées grâce aux données relatives aux avions existants. Typiquement, ce cahier des charges établit des performances cibles à atteindre pour chacune des équipes de concepteurs experts : une masse, une consommation spécifique de carburant, une finesse aérodynamique, une distance de décollage, etc.

Pendant les phases de conception préliminaire et de conception détaillée, l'habillage du plan trois vues par des formes extérieures précises est alors effectué par le biais de différentes boucles de conception s'appuyant sur des outils de simulation numérique haute fidélité ou des essais. La différence majeure entre la conception préliminaire et la conception détaillée ne vient pas des outils utilisés, mais des degrés

de liberté autorisés. Lors de la conception préliminaire, il est encore possible de faire des changements majeurs sur le concept et donc de mettre à jour le cahier des charges ou l'architecture de l'avion. Lors de la phase de conception détaillée, le projet est plus avancé. La mise en production de l'avion débute. Aussi, la production de données pour établir le comportement de l'avion en vol est en cours, de même que le processus de certification. Or, tout ce travail serait à refaire si une modification de l'architecture globale de l'avion était entreprise. Cela aurait un coût très important. Pendant les phases de conception avant-projet et préliminaire, le nombre d'ingénieurs impliqué dans le projet se chiffre en centaines et le coût de développement s'évalue en dizaines de millions d'euros. Pendant la phase de conception détaillée, plusieurs milliers d'ingénieurs travaillent en même temps sur le projet et le coût de développement s'évalue en milliards d'euros. En conception détaillée, les concepteurs se limitent à des modifications locales de l'avion. Ces modifications doivent impacter le minimum de domaines. En aérodynamique, on pourra par exemple étudier l'intérêt de déformations locales (quelques millimètres) de la peau externe de l'aile pour améliorer les performances.

Finalement, on constate [106] que plus le projet est avancé plus les degrés de liberté sur la conception de l'avion sont réduits. Or, paradoxalement, la connaissance de l'avion augmente avec le temps. Son comportement est d'abord prédit par des modèles basse-fidélité en phase avant-projet, puis est prédit plus précisément par des modèles haute-fidélité par la suite. C'est pour contrecarrer ce paradoxe que le formalisme de l'optimisation multidisciplinaire a été pensé. L'optimisation multidisciplinaire permet d'avoir plus tôt dans le projet une meilleure connaissance du système.

La NASA et Sobieszczanski-Sobieski [36,106,107] furent les principaux protagonistes de cette prise de conscience, menant à la fin des années 80 au formalisme de l'optimisation multidisciplinaire (OMD ou « MDO, Multidisciplinary Design Optimization »). A cette date, la simulation numérique commence à être largement utilisée en ingénierie. Aussi, l'optimisation numérique a déjà prouvé son intérêt. Par exemple en conception aérodynamique, grâce aux travaux de Hicks *et al.* [44,45]. Ainsi, il paraît possible de se rapprocher de la meilleure solution, l'optimum multidisciplinaire du système avion complet. L'approche naturelle consiste à piloter l'ensemble des codes de calculs représentant les différents domaines par un unique optimiseur. Toutefois, cette première approche n'est, en pratique, pas satisfaisante car elle exclut le savoir-faire accumulé et le facteur humain. Elle n'autorise aucune intervention extérieure puisque tout y est géré par un algorithme d'optimisation. En outre, le nombre de variables nécessaire pour décrire le système complet dépasse les capacités des algorithmes d'optimisation. La définition d'un tel problème d'optimisation serait aussi très difficile. Elle nécessiterait l'intervention des experts de tous les domaines.

L'optimisation multidisciplinaire consiste à mettre en place un processus d'optimisation permettant d'aboutir à un optimum du système, mais ce processus doit être suffisamment flexible pour s'adapter à l'organisation réelle du projet et donc autoriser une décomposition en sous-tâches. L'emploi de l'OMD est d'autant plus justifié que la connaissance sur le produit est faible. Il est particulièrement adapté à la conception avant-projet. Il est aussi employé en conception préliminaire, mais sur des systèmes plus restreint (fluide/structure, fluide/acoustique, etc.) car les analyses haute fidélité sont généralement coûteuses en temps.

Plus le projet diffère de la norme et des concepts déjà éprouvés, plus l'emploi de l'OMD est intéressant. Ainsi, son emploi n'est pas forcément essentiel pour un projet d'avion en configuration classique, car une grande quantité de données établies à partir des avions existants est connue. Par contre, il l'est pour un projet d'aile volante, d'avion de transport supersonique ou de navette spatiale.

La pratique a montré que pour les systèmes étudiés en aéronautique et en aérospatiale, la maîtrise des interactions entre disciplines est aussi importante que la maîtrise de la conception de chaque discipline prise individuellement. La manière classique de gérer ces interactions consiste à effectuer des boucles de conception au sein d'un processus séquentiel (aérodynamique/structure, charges/structure, etc.), mais il s'avère que cela ne conduit généralement pas à un optimum du système, notamment

pour des raisons de temps et de budget qui limitent le nombre d'itérations possibles. Ce type de processus autorise une grande autonomie entre domaines et laisse aux experts la décision de déterminer le meilleur compromis. Afin de conserver ces caractéristiques, Sobieszczanski-Sobieski [105] a proposé d'établir un diagramme de dépendance du processus complet décrivant les liens entre tous les domaines impliqués. Bien que dans certains cas, une décomposition en sous-tâches organisées selon une hiérarchie pyramidale soit possible, de nombreux problèmes couplés nécessitent d'établir des liens transversaux entre domaines. Il est alors possible de réorganiser ces sous-tâches de manière à minimiser le nombre de connexions, mais ces relations d'interdépendance impliquent la mise en place d'un processus itératif [69].

Les méthodes d'OMD décrivent des processus itératifs permettant de faire évoluer ces disciplines interdépendantes en parallèle, tout en s'assurant que le processus converge vers un optimum du système. Une des manières de casser ces liens transversaux entre disciplines consiste à mettre en place des contraintes de compatibilité. Lorsqu'une approche de type optimisation collaborative (« CO, Collaborative Optimization ») est employée, le processus est décomposé en deux niveaux et toutes les sous-tâches sont exécutées en parallèle. Le prix à payer pour parvenir à un processus aussi flexible est que la solution ne sera admissible que lorsque le processus aura convergé. Les étapes intermédiaires peuvent alors correspondre à des états non physiques où, à un instant donné, une même variable prend des valeurs différentes selon le sous-système considéré ; plusieurs instances de la variable coexistent. Pour obtenir une solution correcte à convergence, il faut alors ajouter au problème d'optimisation une contrainte de compatibilité imposant une égalité entre les différentes instances d'une même donnée.

Les caractéristiques principales des méthodes d'OMD sont :

- la rapidité et la robustesse de convergence vers un optimum du système,
- la stratégie de décomposition adoptée,
- la méthode retenue pour la communication de données entre sous-systèmes.

De nombreux processus d'OMD existent tels que l'optimisation intégrée (« FIO, Fully Integrated Optimization »), l'optimisation concurrente par sous-domaine [105] (« CSSO, Concurrent SubSpace Optimization »), l'optimisation collaborative [14] (« CO, Collaborative Optimization »), l'optimisation intégrée biniveau (« BLISS, Bi-Level Integrated System Synthesis ») et plusieurs articles de synthèse les détaillent [26, 63, 70, 110] et les comparent [3, 16]. Parmi toutes ces méthodes, seulement deux sont rappelées ici :

- la plus naturelle, l'optimisation intégrée (FIO),
- la plus récente, la méthode intégrée biniveau (BLISS).

III.2.2 Couplage géométrique et couplage physique

Les problèmes multidisciplinaires peuvent être de complexité variable, selon la nature des couplages entre disciplines. Si l'on considère la $i^{\text{ème}}$ discipline du système de manière isolée, elle est définie par un système d'équations d'état noté R_i . Ce système d'équations d'état dépend de la forme considérée, donc des variables de forme α_i . Sa résolution permet de calculer les variables d'état Y_i ,

$$R_i(\alpha_i, Y_i) = 0. \quad (\text{III.3})$$

Par exemple, en mécanique des fluides, les variables d'états sont la masse volumique, la vitesse et la pression du fluide. En réalité, il faut considérer les interactions de cette $i^{\text{ème}}$ discipline avec le reste du système.

En premier lieu, les disciplines peuvent être couplées au niveau de la géométrie. Lorsqu'une même forme est analysée par plusieurs disciplines, alors les paramètres de formes sont couplés. Le couplage géométrique se comprend facilement dans le cadre de problèmes fluide/structure. Par exemple, une

modification de l'enveloppe externe de l'avion en aérodynamique impacte l'encombrement intérieur disponible pour la conception de la structure primaire. L'épaisseur minimale d'une aile est dictée par des considérations structurales. Avec la notation $\alpha = \bigcup_i \alpha_i$, la prise en compte du couplage des variables de forme mène à la résolution du système,

$$\forall i, R_i(\alpha, Y_i) = 0. \quad (\text{III.4})$$

Pour le système *avion*, les disciplines sont interdépendantes et le partage de variables de forme, α , est en général insuffisant pour évaluer toutes les fonctions objectifs. Pour effectuer le couplage, il faut parfois aussi coupler les systèmes d'équations d'état de plusieurs disciplines, $R_i(\alpha, Y_i) = 0$, et échanger entre disciplines des variables d'état, Y_i . Par exemple, pour l'optimisation aéroélastique d'une aile en vol stationnaire, les systèmes d'équations fluide et structure sont couplés. La pression exercée par le fluide sur l'aile est une entrée du calcul de la déformation de la structure et inversement, la géométrie de l'aile est une entrée pour la simulation aérodynamique. Avec la notation $\alpha = \bigcup_i \alpha_i$ et $Y_{-i} = \bigcup_{j \neq i} Y_j$, la prise en compte du couplage des variables de forme mène à la résolution du système,

$$\forall i, R_i(\alpha, Y_i, Y_{-i}) = 0. \quad (\text{III.5})$$

Un problème d'optimisation multidisciplinaire général s'écrit donc :

$$\text{respectant : } \begin{cases} \min_{\alpha} (F(\alpha, Y)) \\ \forall i, R_i(\alpha, Y) = 0 \\ \alpha = \bigcup_j \alpha_j \\ Y = \bigcup_j Y_j \\ \alpha \in \mathcal{D} \end{cases} . \quad (\text{III.6})$$

Ce problème multi-objectif peut être résolu en recherchant un front de Pareto. La fonction objectif du système est alors un vecteur $F = \bigcup_i F_i$. Lorsque le vecteur d'équivalence entre discipline, θ , peut être fixé *a priori*, il est préférable d'utiliser une fonction obtenue par combinaison linéaire, $F = \sum_i \theta_i F_i$ (cf. I.1.1).

L'optimisation multidisciplinaire diffère de l'optimisation classique car l'évaluateur (cf. Figure I.6) n'est plus considéré comme une boîte noire, mais comme un ensemble de sous-systèmes interdépendants gérés de manière plus ou moins autonome. En outre, l'optimiseur ne sert pas uniquement à minimiser une ou plusieurs fonctions objectifs. Certaines méthodes d'OMD l'utilisent aussi pour synchroniser les sous-systèmes entre eux. Pour cela, une quantité de désynchronisation peut être introduite au problème sous la forme d'une contrainte d'égalité à respecter (contrainte de compatibilité).

III.2.3 Optimisation intégrée, FIO

La méthode la plus naturelle consiste à utiliser directement un évaluateur multidisciplinaire (« MDA, MultiDisciplinary Analysis ») capable de résoudre le système d'équations couplées et d'évaluer toutes les fonctions en une population de variables géométriques donnée. L'optimiseur système pilote alors toutes les variables de forme couplées du problème, c'est-à-dire l'union des variables de forme couplées de chaque discipline, de la même manière que pour une simple optimisation multi-objectif (cf. Figure I.6). C'est la méthode d'optimisation multidisciplinaire intégrée (« FIO, Fully Integrated Optimization »).

On appelle aussi parfois cette méthode l'optimisation multidisciplinaire réalisable (« MDF, Multidisciplinary Feasible ») [26] car chacun des vecteurs de variables de forme traités par l'optimiseur système correspond à une solution admissible du point de vue multidisciplinaire, c'est-à-dire à une forme réalisable même si non optimale. En pratique, lorsque les disciplines sont couplées, l'évaluateur multidisciplinaire ne se contentera pas d'effectuer une seule analyse pour chaque discipline. En effet, il ne suffit pas que chaque discipline respecte son propre système d'équations d'état (« IDF, Individual

Disciplinary Feasible ») pour que la solution soit réalisable du point de vue multidisciplinaire. Il faut en plus que toutes les variables couplées aient la même valeur au sein de chaque discipline. L'évaluateur multidisciplinaire peut donc nécessiter la mise en place d'un processus itératif interne effectuant plusieurs analyses, jusqu'à atteindre une solution multidisciplinaire réalisable.

Cette méthode n'autorise aucune désynchronisation entre les sous-systèmes, ni aucun parallélisme. C'est la méthode la plus rapide [3, 16], la plus économe en nombre d'évaluations de fonctions et donc en temps de calcul. En pratique, cette stratégie est applicable à de petits systèmes. En effet, l'implémentation informatique de l'évaluateur multidisciplinaire devient très complexe lorsqu'il faut coupler plusieurs logiciels experts entre eux. Elle s'accommode mal aux contraintes de répartition de tâches liées à la réalité de l'organisation du travail en industrie. Cette méthode est donc adaptée lorsque le problème est simple du point de vue multidisciplinaire (peu de disciplines couplées), mais que certaines disciplines nécessitent l'utilisation de logiciels coûteux d'analyse haute-fidélité. Elle est robuste et nécessite peu d'itérations pour converger [16]. Elle est souvent mise en œuvre pour réaliser des optimisations aéroélastiques d'ailes (couplage fluide/structure) en s'appuyant sur des logiciels experts [20]. Par exemple, Martins [81] l'a utilisé avec un algorithme d'optimisation par gradient et une méthode de calcul de gradient couplé par l'équation adjointe.

Enfin, dans le cas où l'on peut substituer l'évaluateur de chacune des disciplines par une surface de réponse globale, l'évaluateur multidisciplinaire (MDA) devient trivial et le problème multidisciplinaire se résume à une simple optimisation multi-objectif.

Pour mettre en perspective la méthode FIO, il est possible de la décrire comme l'extrême opposé de la méthode d'optimisation tout-en-un (« AAO, All-At-Once »). Dans le formalisme tout-en-un, l'optimiseur système gère directement toutes les variables du système, c'est-à-dire toutes les variables géométriques (couplées ou non) et toutes les variables d'états (couplées ou non). Les différents systèmes d'équations d'état de chaque discipline sont agglomérés en un système unique résolu par l'optimiseur. La solution sera une solution multidisciplinaire réalisable seulement à convergence du système. De plus, les solutions intermédiaires ne respecteront pas forcément les systèmes d'équations individuels à chaque discipline (solution IDF).

III.2.4 Optimisation intégrée biniveau, BLISS

Sobieszcanski-Sobieski [108] a introduit l'optimisation intégrée biniveau (« BLISS, Bi-Level Integrated System Synthesis ») en 1998 dans le but de décomposer le processus d'optimisation multidisciplinaire en un ensemble de sous-tâches de bas niveau synchronisées grâce à une optimisation haut niveau judicieusement posée. La modularité de cette méthode permet d'effectuer chaque analyse bas-niveau de manière concurrente en utilisant des ressources diverses (optimisation numérique, simulation numérique, abaque, loi de comportement) grâce à une information quantitative de l'influence de son domaine sur les autres domaines et sur le système.

L'articulation entre le haut-niveau et les bas-niveaux s'effectue par une décomposition de l'ensemble des variables de forme,

$$\alpha = \alpha_{glob} \cup_{i=1, n_{ss}} \alpha_{loc_i}. \quad (\text{III.7})$$

Les variables de forme communes à au moins deux sous-systèmes constituent l'ensemble des variables globales (ou haut-niveau ou système ou encore publiques), α_{glob} . Ces variables de forme globale décrivent les couplages géométriques du système. Au haut-niveau, l'optimiseur système a pour rôle de trouver les variables de forme globales optimales. Afin d'obtenir une convergence rapide de l'algorithme BLISS, il faut essayer de restreindre le nombre de variables globales.

Les autres variables de forme sont donc spécifiques aux n_{ss} sous-systèmes : elles forment des sous-ensembles indépendants de variables locales (ou bas-niveau ou privées), α_{loc_i} . Ces variables étant locales, leur nombre n'a pas d'impact sur la convergence du système et peut donc être élevé. Au sein de

chaque sous-système, une optimisation à variables globales figées permet de déterminer les variables locales optimales. Cette décomposition de l'ensemble des variables va permettre de traiter des problèmes couplés au niveau de la géométrie en parallèle.

Le problème d'optimisation biniveau intermédiaire ainsi défini s'écrit :

$$\left\{ \begin{array}{l} \bullet \min_{\alpha_{glob}} (F(\alpha, Y)) \quad \text{au haut-niveau} \\ \hline \bullet \min_{\alpha_{loc_i}} (F_i(\alpha_{glob}, \alpha_{loc_i}, Y)) \quad \text{au } i\text{-ème bas-niveau} \\ \text{respectant : } R_i(\alpha_{glob}, \alpha_{loc_i}, Y) = 0 \end{array} \right. \quad (III.8)$$

Concernant le couplage physique, ce système intermédiaire ne permet pas de traiter les sous-systèmes simultanément puisque les variables d'état, $Y = \bigcup_j Y_j$, sont directement partagées. Pour que les optimisations bas-niveau puissent être effectuées simultanément, il suffit de désynchroniser la valeur des variables d'état couplées. Pour cela, chaque sous-système construit un modèle approché de ses variables, $\hat{Y}_i(\alpha_{glob})$.

Pendant une itération du système, deux instances des variables d'état couplées, Y_i , coexistent. Les variables exactes Y_i respectent réellement le système d'équations d'état R_i ,

$$R_i(\alpha_{glob}, \alpha_{loc_i}, \hat{Y}_{-i}, Y_i) = 0 \quad (III.9)$$

Elles ne sont connues qu'après résolution de l'optimisation bas-niveau du $i^{\text{ème}}$ sous-système. Ce sont des variables privées, visibles seulement par leur sous-système d'origine. La résolution de ce système d'équations nécessite la connaissance des variables d'état des autres disciplines. Pour cela, les variables d'état approchées, $\hat{Y}_{-i} = \bigcup_{j \neq i} \hat{Y}_j$, sont utilisées. Après chaque analyse bas-niveau, les variables exactes Y_i sont stockées dans une base de données. A partir de cette base de données, le modèle approché \hat{Y}_i est construit. Seules les variables approchées $\hat{Y} = \bigcup_i \hat{Y}_i$ sont partagées entre sous-systèmes. Ce sont des variables publiques.

De la même manière, l'optimisation haut-niveau est découplée des analyses bas-niveau car elle est conduite sur des fonctions objectifs approchées, $\hat{F}_i(\alpha_{glob})$. Finalement, le problème d'optimisation intégrée biniveau BLISS est :

$$\left\{ \begin{array}{l} \bullet \min_{\alpha_{glob}} (\hat{F}(\alpha_{glob})) \quad \text{au haut-niveau} \\ \alpha_{loc} \text{ fixes} \\ \hline \bullet \min_{\alpha_{loc_i}} (F_i(\alpha_{glob}, \alpha_{loc_i}, \hat{Y}_{-i}, Y_i)) \quad \text{au } i\text{-ème bas-niveau} \\ \text{respectant : } R_i(\alpha_{glob}, \alpha_{loc_i}, \hat{Y}_{-i}, Y_i) = 0 \\ \alpha_{glob} \text{ fixes} \\ \bullet \text{ construire : } \hat{F}_i(\alpha_{glob}), \hat{Y}_i(\alpha_{glob}) \end{array} \right. \quad (III.10)$$

Ce processus BLISS est représenté en figure III.1. Avec cette méthode, il n'est pas nécessaire de mettre en œuvre une contrainte de synchronisation entre sous-systèmes du type $\hat{Y}_i = Y_i$. En effet, il suffit d'utiliser parmi les critères de convergence du système une condition d'égalité entre fonctions objectifs exactes et approchées, $\hat{F}_i(\alpha_{glob}^*) = F_i(\alpha_{glob}^*)$. En pratique, cela signifie que plus les modèles approchés utilisés sont précis plus l'optimisation BLISS converge rapidement. Ainsi, lorsqu'il est possible de construire des modèles approchés des fonctions objectifs valables sur tout le domaine des variables globales, il n'est pas nécessaire d'effectuer d'itérations. Ce sera le cas dans l'application considérée ensuite (cf. III.3) où l'on construira des surfaces de réponse globales pour chacun des sous-systèmes.

Par rapport à une boucle de conception classique, le processus BLISS nécessite la mise en œuvre de méthodes de construction de modèles réduits pour approcher les données bas-niveau couplées ($\hat{F}_i(\alpha_{glob})$)

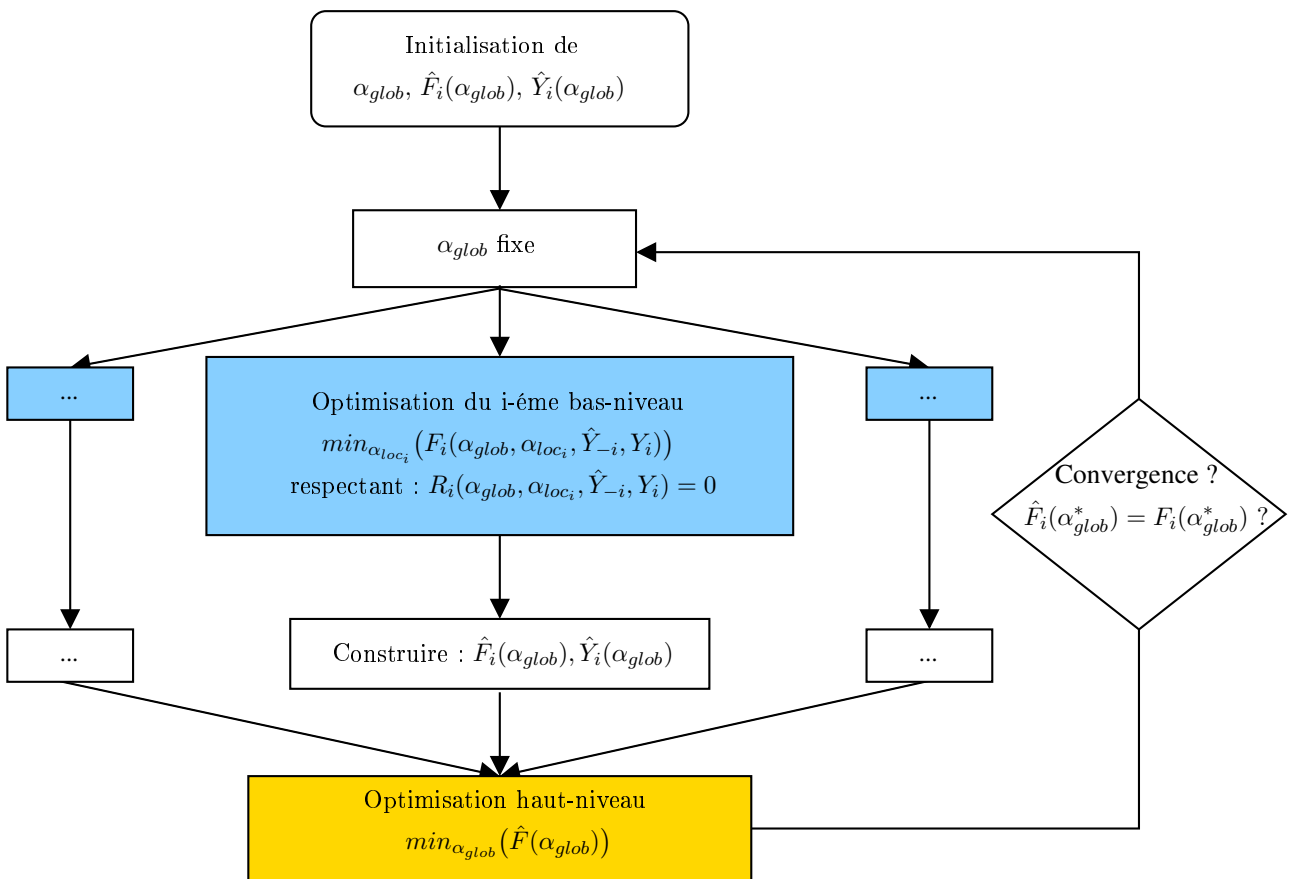


FIG. III.1: Processus d'optimisation multidisciplinaire intégrée biniveau BLISS.

et $\hat{Y}_i(\alpha_{glob})$). A l'origine, une formule de Taylor à l'ordre un, nécessitant un calcul de sensibilité à la suite de chaque analyse bas-niveau, était employée. Altus [4] a ensuite proposé la version BLISS-RS permettant de s'affranchir des calculs de sensibilités en utilisant des surfaces de réponse linéaires ou quadratiques, construites grâce à une base d'échantillons formée de plusieurs résolutions d'optimisation bas-niveau pour différentes valeurs de variables globales. Aujourd'hui, de nombreuses variantes existent (BLISS-2000) selon la méthode de modélisation des données bas-niveau choisie. En général, le nombre de fonctions couplées est relativement faible et les méthodes de type Krigeage, réseau de neurones ou régression polynomiale sont employées pour les modéliser.

Toutefois, ces méthodes peuvent être inadaptées pour approcher les variables d'état d'un sous-système, $\hat{Y}_i(\alpha_{glob})$. La quantité de données à interpréter pouvant être grande, l'emploi direct de telles formulations risquerait d'être coûteux. Par exemple, pour le cas d'une voilure aéroélastique, il faudrait créer un modèle d'ordre réduit de champs surfaciques pouvant contenir plusieurs centaines de milliers de points. Pour effectuer l'optimisation de bas-niveau structure il faut modéliser la variation du champ surfacique de pression en fonction des variables globales ; de même, pour effectuer l'optimisation de bas-niveau aérodynamique, il faut modéliser la variation du champ surfacique de déplacement en fonction des variables globales. Pour cela, il est donc nécessaire de filtrer ou réduire l'information. Ainsi, il semble préférable d'utiliser une méthode de type « POD, Proper Orthogonal Decomposition » qui est capable de détecter les structures les plus énergétiques d'un phénomène. A partir de la masse d'information disponible, une base ordonnée de vecteurs orthogonaux représentatifs des phénomènes prépondérants vis-à-vis du domaine de variation considéré est établie. Par exemple, un modèle POD n'utilisant qu'un seul vecteur propre approchera un champ surfacique inconnu par une simple moyenne de tous les champs surfaciques connus (« snapshots »).

Certains sous-systèmes sont couplés seulement au niveau des variables de forme globales. En l'absence de couplage des équations d'état, le problème BLISS simplifié devient,

$$\left\{ \begin{array}{l} \bullet \min_{\alpha_{glob}} (\hat{F}(\alpha)) \quad \text{au haut-niveau} \\ \alpha_{loc} \text{ fixes} \\ \hline \bullet \min_{\alpha_{loc_i}} (F_i(\alpha_{glob}, \alpha_{loc_i}, Y_i)) \quad \text{au } i\text{-ème bas-niveau} \\ \text{respectant : } R_i(\alpha_{glob}, \alpha_{loc_i}, Y_i) = 0 \\ \alpha_{glob} \text{ fixes} \\ \bullet \text{ construire : } \hat{F}_i(\alpha_{glob}) \end{array} \right. \quad (III.11)$$

La méthode BLISS, décrite en 1998, est l'une des plus récentes. Sobieski a toutefois introduit une nouvelle méthode « TLISS, Three Level Integrated System Synthesis » [109], qui est simplement une généralisation de la méthode BLISS aux cas où le problème peut être décomposé selon une architecture à trois niveaux : système de systèmes, systèmes et sous-systèmes.

D'après la comparaison de Brown et Olds [16], la stratégie BLISS est la plus prometteuse pour l'optimisation multidisciplinaire. Cette étude n'est bien entendu pas exhaustive. Les applications traitées sont représentatives des problèmes traités dans un contexte avant-projet.

III.3 Application à l'optimisation fluide-structure de la position d'un moteur

Les méthodes d'OMD sont généralement développées et appliquées dans le contexte de la phase de conception avant-projet où les codes d'analyses employés sont rapides à évaluer. Il semble qu'aucune application de la méthode intégrée biniveau à un problème de conception préliminaire, basée sur des logiciels d'analyse haute-fidélité, n'ait été publiée. En effet, la plupart des OMD haute-fidélité sont généralement résolues par la méthode intégrée FIO afin de réduire le temps de calcul.

L'application présentée ici a pour but d'étudier concrètement l'intérêt de BLISS en conception préliminaire sur un petit système (fluide/structure) où les variables d'état ne sont pas couplées, mais qui s'appuie sur des codes d'analyse haute-fidélité maîtrisés par deux équipes expertes différentes.

III.3.1 Fonctions objectifs et variables de forme, haut-niveau

Le système étudié est la position axiale, X , et verticale, Z , et l'épaisseur, W , du mât d'une nacelle d'un avion de transport de type Airbus A3XX tel que représenté en figure III.2. L'installation motrice choisie est la plus lointaine du fuselage (III.4). Ce problème multidisciplinaire fait intervenir les do-

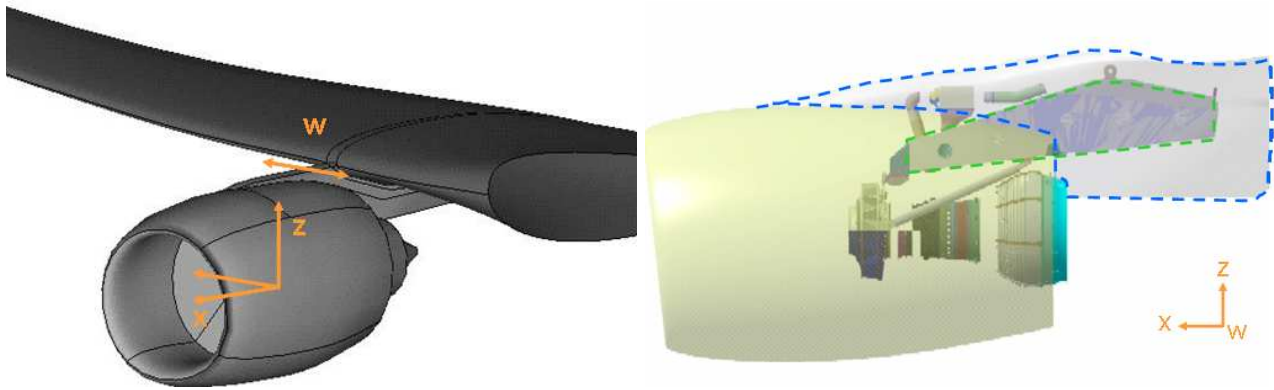


FIG. III.2: Variables globales pour l'optimisation de l'installation motrice. Carénage externe en bleu et structure primaire du mât en vert.

maines aérodynamique et structure. Les fonctions considérées sont la traînée de pression de l'avion, $J_{aero}(\alpha_{glob})$, et la masse de la structure primaire soutenant le moteur, $J_{struct}(\alpha_{glob})$.

Les variables de forme globales agissent sur les deux fonctions objectifs et sont adimensionnées par la corde locale de l'aile, C ,

$$\alpha_{glob} = \left(\frac{X}{C}, \frac{Z}{C}, \frac{W}{C} \right) \in [-0,024; +0,024]^3. \quad (\text{III.12})$$

La corde locale mesurant environ dix mètres, les déplacements autorisés sont de l'ordre de quelques dizaines de centimètres. Cela correspond à un domaine d'étude très restreint ; mais en aérodynamique, de telles variations de forme peuvent avoir énormément d'impact sur la traînée, notamment lors d'apparition de chocs.

En plus de ces variables globales agissant au niveau du système, des variables locales dont l'influence est limitée à un seul sous-système sont considérées grâce à un processus d'optimisation biniveau de type « BLISS » tel que décrit en III.2.4. Ce processus permet de réutiliser directement les outils d'optimisation monodisciplinaire spécifiques à chaque domaine au sein du processus d'optimisation multidisciplinaire.

III.3.2 Analyse bas-niveau structure du mât

L'analyse structure consiste à minimiser la masse, $J_{struct}(\alpha_{struct}) = M$, de la structure primaire du mât (représentée en figure III.2) en respectant un niveau de contrainte structurale maximum inférieur au seuil de tolérance du matériau. Cette contrainte maximale est testée pour 45 cas de chargements différents : 30 cas de fatigue et 15 cas de rupture de pale. Les variables locales au domaine structure, α_{struct} , sont 27 épaisseurs de panneaux constituant la structure primaire, représentées par des codes

couleur sur la Figure III.3.

$$\begin{cases} J_{struct}(\alpha_{struct}) = M \\ n_{G_{struct}} = 45 \\ n_{\alpha_{struct}} = 27 \end{cases} . \quad (III.13)$$

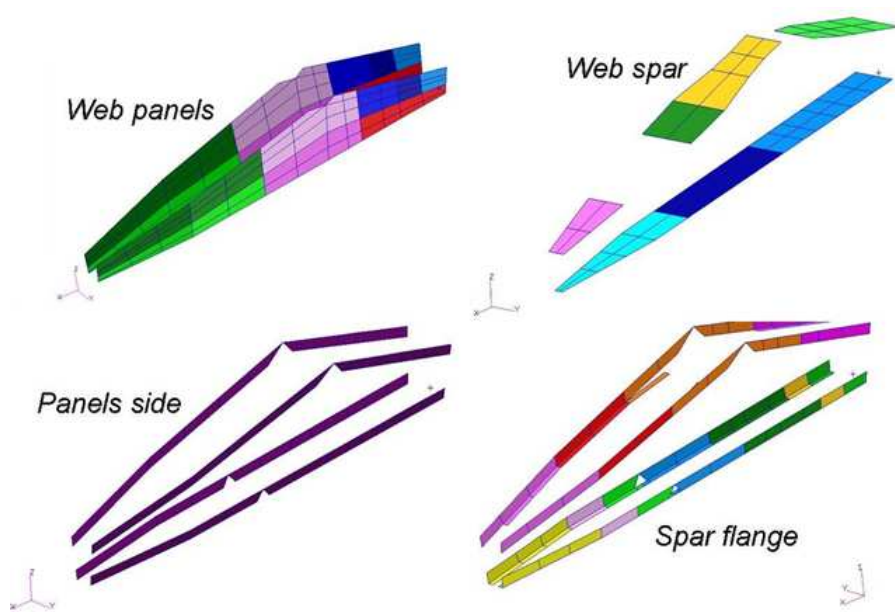


FIG. III.3: Variables locales pour les optimisations bas-niveau en structure.

La masse est évaluée par le logiciel NASTRAN SOL200. Les fonctions contraintes associées à chaque cas de chargement de la structure sont évaluées par des codes de calcul développés par Airbus. Le logiciel d'optimisation BOSS Quattro pilote ces éléments pour effectuer une optimisation locale par méthode de gradient et atteindre une structure de masse optimale respectant les contraintes.

Pour ce problème, un calcul de gradient par une méthode adjointe n'aurait pas de sens puisqu'il y a plus de fonctions (46) que de variables (27). Le gradient est calculé par la méthode des différences finies (cf. I.3.5.1). L'évaluation de la masse n'est pas coûteuse en temps de calcul puisqu'elle est obtenue par de simples mesures géométriques. Par contre, le calcul de la contrainte maximum (σ) nécessite la résolution d'équations aux dérivées partielles. Une optimisation par rapport aux variables structures dure quelques heures.

L'optimisation est utilisée ici d'un point de vue différent de celui des aérodynamiciens. En effet, en aérodynamique, l'optimisation est utilisée pour améliorer les performances d'une configuration en condition de vol normal. A l'heure actuelle, il n'est pas possible d'inclure dans une seule optimisation aérodynamique des contraintes liées au domaine de vol complet (flottement de la voilure, manœuvre à 2,5g, etc.). Or, dans le cas décrit ici, l'optimisation a pour but de réduire la masse, mais surtout de proposer un concept admissible, c'est-à-dire qui respecte au mieux les contraintes (σ) auxquelles la structure est soumise lors du fonctionnement normal (fatigue) et lors d'éventuels incidents (rupture de pale). C'est le respect de ces contraintes qui dimensionne la structure primaire. Etant donnée la faible amplitude de variation des variables haut-niveau, il s'avère qu'une solution admissible existe sur tout le domaine d'étude.

Une analyse bas-niveau structure est donc une optimisation par rapport aux variables structures. Les analyses structures ont été réalisées par le département structure d'Airbus (ESANT).

III.3.3 Analyse bas-niveau aérodynamique de la configuration fuselage-aile-mât-nacelle

Le problème spécifique au domaine de l'aérodynamique est de minimiser la traînée de l'avion au point de croisière, $Mach = 0,85$ et $Cl = Cl_0$, par des déformations de l'enveloppe externe de l'avion. Pour cela le logiciel OPTaliA (I.3) est employé.

En aérodynamique, la conception d'un composant isolé est impossible et l'on doit modéliser le moteur installé, donc une configuration (quasi-) complète. La demi-configuration étudiée est composée d'un demi-fuselage, d'une voilure et de deux installations motrices (« WBPn, Wing Body Pylon Nacelle »); elle est définie par un maillage formé de 1,5 millions de nœuds (Figure III.4). La faible résolution de ce maillage par rapport à la complexité de la géométrie ne permet pas aux simulations *RANS* de prédire les coefficients aérodynamiques avec précision en valeur absolue, mais elle permet de modérer le temps de calcul.

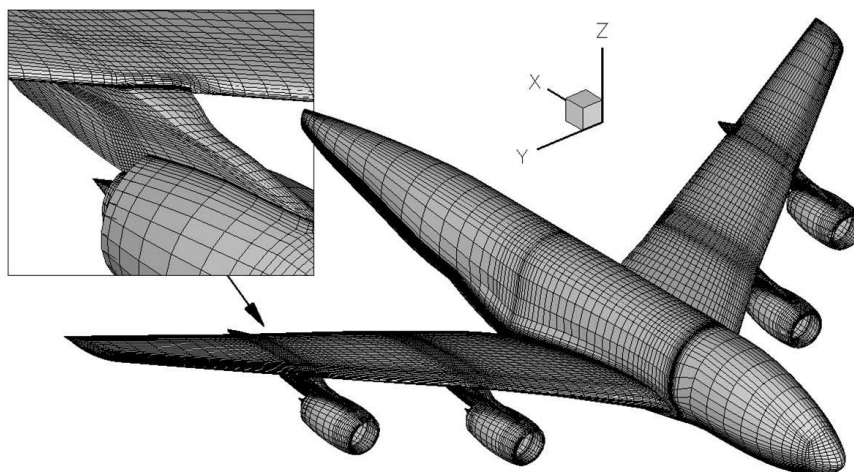


FIG. III.4: Maillage aérodynamique (symétrisé). Crédit ONERA.

La fonction objectif est la traînée de pression de l'avion à laquelle on ajoute un terme de pénalisation afin de garder la portance constante sans considérer de fonctions contraintes,

$$\begin{cases} J_{aero}(\alpha_{aero}) = Cd_p(\alpha_{aero}) - (Cl(\alpha_{aero}) - Cl_0) \left(\frac{dCd_p}{dCl} \right)_0 \\ n_{G_{aero}} = 0 \\ n_{\alpha_{aero}} = 19 \end{cases} \quad (III.14)$$

La sensibilité de la traînée de pression par rapport à la portance, $\left(\frac{dCd_p}{dCl} \right)_0$, est calculée préalablement à l'optimisation sur la configuration de référence.

De la même manière que pour une aile sont différenciées les surfaces intrados et extrados, le mât est décomposé en une surface interne orientée face au fuselage et une surface externe orientée vers le bout de l'aile. Les 19 variables de conception aérodynamique pilotent 17 bosses de Hicks-Henne réparties sur le mât en différentes zones :

- le bord d'attaque du mât,
- les surfaces latérales internes et externes du mât,
- les lignes d'intersection internes et externes entre le mât et la voilure.

La géométrie initiale du mât n'est bien entendu pas symétrique et les déformations autorisées peuvent aussi être dissymétriques entre le mât externe et interne. Le domaine de variation choisi autorise les bosses d'amplitude négative. La figure III.5 représente les formes correspondant aux déformations

d'amplitudes minimale, nulle et maximale. On peut constater sur la figure III.2 que le carénage externe

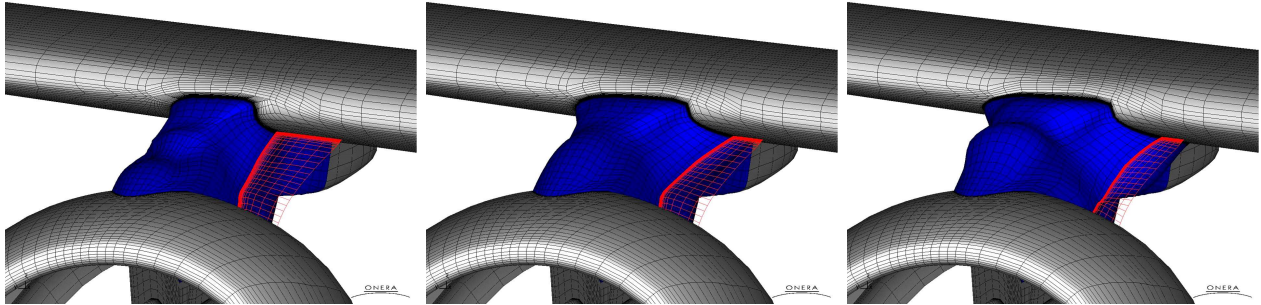


FIG. III.5: Domaine de variation des variables locales au bas-niveau en aérodynamique. Maillages correspondant aux déformations d'amplitude minimale (à gauche), nulle (au centre) et maximale (à droite). Crédit ONERA.

contient la structure primaire ainsi que de nombreux accessoires. Un problème d'optimisation mieux posé aurait donc dû inclure des contraintes géométriques pour respecter un encombrement intérieur minimal. On admettra par la suite que les déformations d'amplitude négative sont suffisamment faibles pour que cet encombrement soit toujours respecté.

Tel que décrit en I.3.2, à chaque vecteur de variables locales, α_{aero} , l'outil OPTaliA déduit le nouveau maillage par déformation d'un maillage initial. Toutefois, OPTaliA ne peut pas effectuer de déformation de maillage correspondant à un nouveau vecteur de variables haut-niveau. Pour chaque nouvelle position d'installation motrice, c'est-à-dire pour chaque vecteur de variable globale α_{glob} , le maillage est généré par un moyen externe puis fourni en tant que maillage initial en entrée d'OPTaliA.

La figure III.6 représente l'historique de convergence de l'optimisation pour la configuration initiale. La variable en abscisse est le numéro d'itération interne à l'optimiseur employé. Les deux courbes représentent l'évolution de la fonction objectif et du nombre d'évaluations de fonctions. La fonction objectif est réduite d'environ un demi point de traînée après 20 itérations. L'algorithme utilisé est *DOT* (méthode quasi-Newton BFGS, cf. I.3.7). Pour cet algorithme, une itération correspond à une étape de recherche linéaire dans une direction calculée à partir du gradient. En 20 itérations, 20 systèmes adjoints sont résolus, soit un coût équivalent à 20 simulations RANS. Au cours d'une itération de *DOT*, plusieurs évaluations de fonctions sont réalisées telles que représentées par chacun des carrés. Au total, la simulation RANS a été résolue sur 103 formes différentes. Le coût total d'une optimisation est équivalent au coût d'environ 123 simulations RANS. Cela représente 116 heures de calcul sur un processeur d'une machine NEC SX-8.

Une analyse bas-niveau aérodynamique est une optimisation par rapport aux variables aérodynamiques. Les analyses aérodynamiques ont été réalisées par le département aérodynamique appliquée (DAAP) de l'ONERA.

III.3.4 Optimisation haut-niveau

Au niveau du système, l'algorithme d'optimisation retenu est basé sur des surfaces de réponse globales modélisant chacun des sous-systèmes sur tout le domaine des variables (X,Z,W) . Comme seulement trois variables de conception sont considérées, il est possible d'effectuer un échantillonnage relativement dense du domaine \mathcal{D}_{glob} . Mais, pour des raisons de temps une densité de seulement deux points par direction ($2^3 = 8$) a finalement été adoptée. Les surfaces de réponse sont construites à partir de $n_s = 10$ échantillons pour la structure et $n_s = 9$ échantillons pour l'aérodynamique. Dans les deux cas, la méthode d'échantillonnage est un plan factoriel.

Après cette étape d'échantillonnage les analyses bas-niveau correspondantes sont réalisées en parallèle comme le montre la figure III.7. Lors de ces analyses, les variables globales sont figées et les

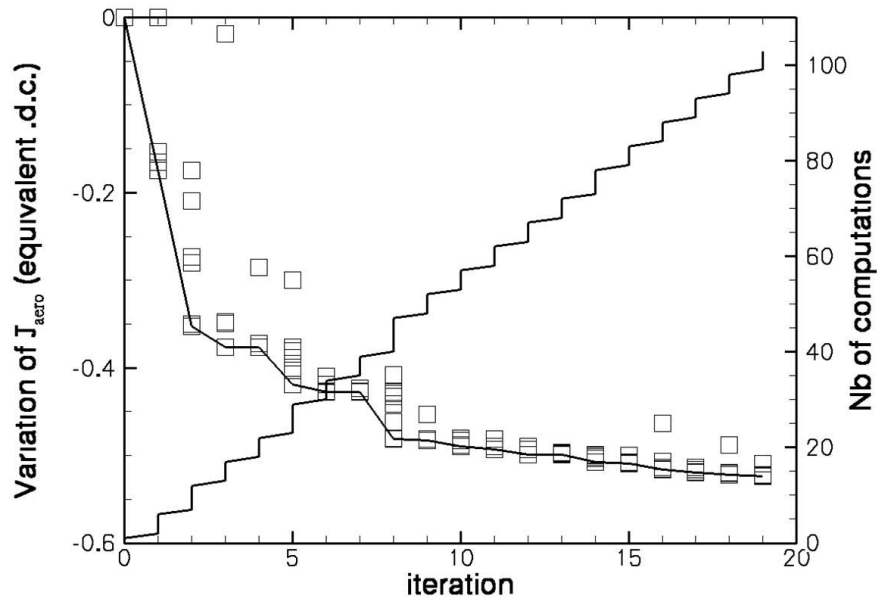


FIG. III.6: Convergence de l'optimisation aérodynamique à partir du maillage correspondant à la configuration initiale. Crédit ONERA.

optimisations par rapport aux variables bas-niveau sont résolues. Sur la figure III.7, ces résolutions ont été décrites selon un processus séquentiel au sein de chaque discipline. En réalité, comme une méthode d'échantillonnage *a priori* est utilisée, les échantillons sont tous générés en même temps et, au sein d'une discipline, les analyses sont toutes lancées en parallèle.

Lorsque la réponse du sous-système a été calculée pour l'ensemble des n_s échantillons, il est alors possible de modéliser ce sous-système par une surface de réponse. Ce modèle réduit est obtenu par une méthode de Krigage telle que décrite en II. Ces deux surfaces de réponse globales seront supposées exactes sur tout le domaine, de sorte que le processus ne fasse pas appel à une boucle itérative au niveau du système.

Au haut-niveau, l'optimiseur agit sur les variables globales et recherche l'optimum du système en s'appuyant uniquement sur les surfaces de réponse structure et aérodynamique. La fonction objectif est une simple agrégation entre fonctions aérodynamique et masse utilisant un coefficient d'équivalence, k , (cf. I.1.2.1)

$$J = J_{aero} + 2kJ_{struct}. \quad (\text{III.15})$$

Le facteur 2 sur la masse s'explique par le fait que, sur un avion, les mâts sont tous identiques. La traînée étant calculée pour un demi-avion, la masse des deux mâts est prise en compte (la configuration est quadrimoteur). Le coefficient k correspond au rapport du poids total de l'avion et de la force de traînée de l'avion à portance nulle. Par rapport aux méthodes décrites au chapitre I, ce coefficient est obtenu par une mise à l'échelle des objectifs par leur valeur absolue (cf. I.1.2.1) qui ne mènera pas forcément à un optimum équilibré au sens de Pareto (point intérieur au front de Pareto).

En résumé, le problème à résoudre est,

$$\left\{ \begin{array}{l}
 \bullet \min_{\alpha_{glob}} (\hat{J}_{aero}(\alpha) + 4k\hat{J}_{struct}(\alpha)) \quad \text{au haut-niveau} \\
 \alpha_{loc} \text{ fixes} \\
 \hline
 \bullet \min_{\alpha_{struct}} (J_{struct}(\alpha_{glob}, \alpha_{struct}, Y_{struct})) \\
 \text{respectant : } R_{struct}(\alpha_{glob}, \alpha_{struct}, Y_{struct}) = 0 \\
 \quad \forall i \in [1, n_{G_{struct}}], \sigma_i < \sigma_{admissible} \\
 \alpha_{glob} \text{ fixes} \\
 \bullet \text{ construire : } \hat{J}_{struct}(\alpha_{glob}) \\
 \bullet \min_{\alpha_{aero}} (J_{aero}(\alpha_{glob}, \alpha_{aero}, Y_{aero})) \\
 \text{respectant : } R_{aero}(\alpha_{glob}, \alpha_{aero}, Y_{aero}) = 0 \\
 \alpha_{glob} \text{ fixes} \\
 \bullet \text{ construire : } \hat{J}_{aero}(\alpha_{glob})
 \end{array} \right. . \quad (III.16)$$

III.3.5 Résultats

Les valeurs numériques pour les fonctions objectifs représentent des différences par rapport à la configuration de référence. Une amélioration par rapport à la référence correspondra à une fonction objectif négative et une fonction objectif nulle définira une solution équivalente à la solution de référence.

Le domaine des variables haut-niveau étant un cube, la visualisation des surfaces de réponse en trois dimensions n'est pas aisée. Il est tout de même possible de se représenter leurs tendances en traçant l'évolution de quelques iso-surfaces. Sur les figures III.8 et III.10, les surfaces de réponse calculées par le Krigeage sont ainsi tracées. Les points de construction y sont représentés par des cubes. L'erreur standard prédite par le Krigeage y est représentée en valeur absolue.

Comme le nombre de variables haut-niveau est faible, la recherche de l'optimum global de la fonction objectif est très facile. Une simple exploration par une grille suffisamment fine (100^3) permet de localiser la position du minimum global avec une précision de quelques millimètres.

III.3.5.1 Surface de réponse structure

La figure III.8 représente la réponse du sous-système structure sur le domaine des variables globales. Les iso-surfaces ressemblent à des surfaces quadratiques peu courbées. Le minimum de cette surface de réponse est situé dans un coin du domaine en,

$$\alpha_{glob}^{*struct} = \left(\frac{X}{C} = -0,024; \frac{Z}{C} = +0,024; \frac{W}{C} = +0,024 \right). \quad (III.17)$$

Cela correspond effectivement aux tendances prévues par l'expérience, à savoir que le mât s'alourdit lorsque :

- sa longueur augmente, c'est-à-dire si X augmente ou Z diminue,
- sa largeur W diminue.

En effet, lorsque sa longueur augmente, les dimensions augmentent, mais la charge à soutenir augmente aussi par un effet de bras de levier. Lorsque la largeur diminue, il faut ajouter de la matière pour soutenir la même charge, ce qui amène une pénalité de masse.

La prédiction d'erreur fournie par le Krigeage est révélatrice d'un échantillonnage mal distribué. En effet, aucun échantillon n'explore la borne supérieure pour la variable Z . Cela se traduit par une

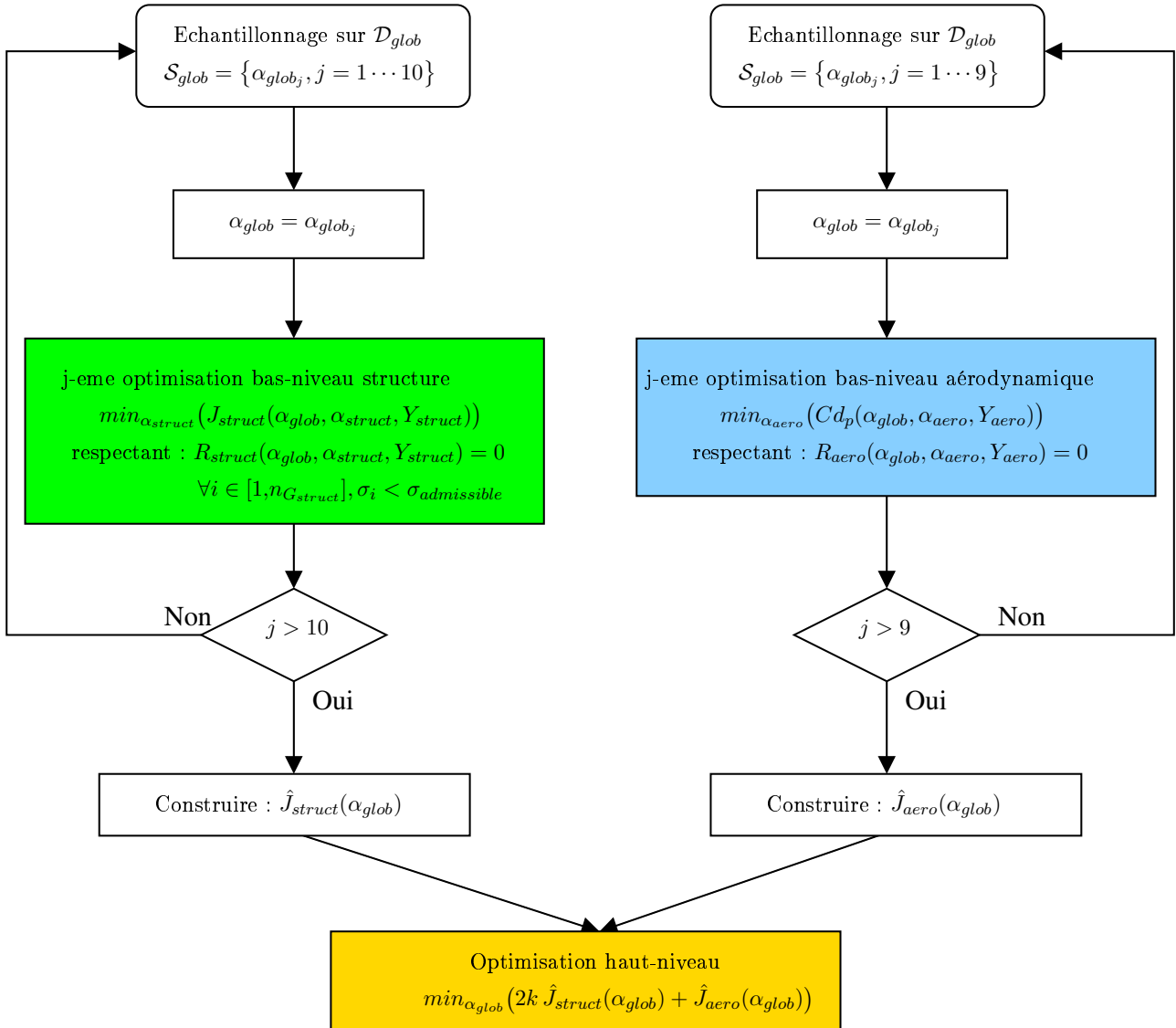


FIG. III.7: Processus de conception multidisciplinaire biniveau.

iso-surface d'erreur de l'ordre de 10 kilogrammes correspondant à un plan constant en $\frac{Z}{C} \approx 0,02$. Autrement dit, l'erreur absolue est supérieure à 10 kilogrammes dès lors que $\frac{Z}{C} > 0,02$. De plus, le maximum d'erreur correspond au minimum global sur la surface de réponse. Le niveau d'erreur y est non négligeable, d'environ 20 pourcents (20 kilogrammes d'erreur pour une amélioration de 100 kilogrammes) et l'erreur moyenne absolue sur le domaine vaut 7 kilogrammes. Toutefois, il s'avère que cet optimum est robuste vis-à-vis de l'erreur standard du Krigeage. Telle que représenté en figure III.9, une minimisation de la fonction moins l'erreur (LCB_1) ou de la fonction plus l'erreur (LCB_{-1}) ne change pas la localisation de l'optimum.

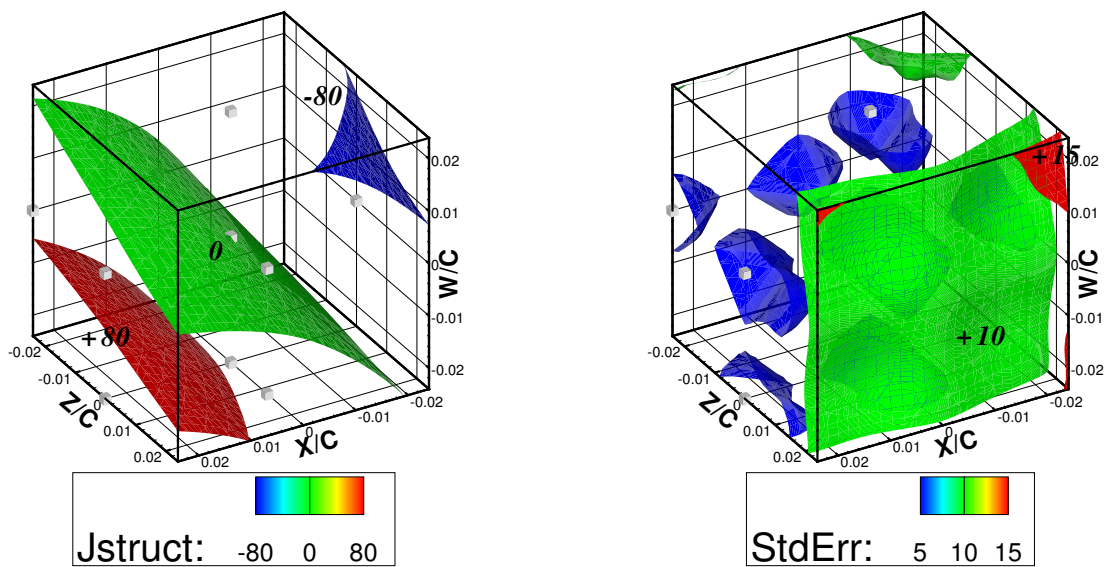


FIG. III.8: Surface de réponse structure et son erreur standard en valeur absolue.

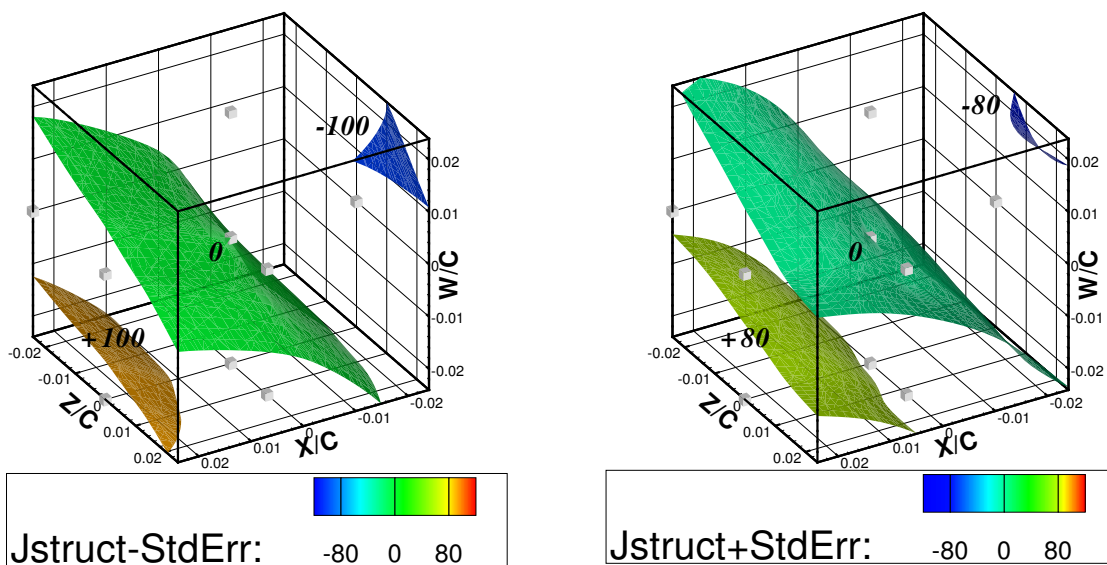


FIG. III.9: A gauche, surface de réponse structure plus son erreur standard. A droite, surface de réponse structure moins son erreur standard.

III.3.5.2 Surface de réponse aérodynamique

La figure III.10 représente la réponse du sous-système aérodynamique sur le domaine des variables globales. L'iso-surface $J_{aero} = -0,5$ en bleu est composée de deux éléments disjoints indiquant la présence de deux minima situés sur deux arêtes différentes du domaine. On admettra que ces deux minima sont les seuls minima de la surface de réponse aérodynamique. Le minimum local est situé en $(\frac{X}{C} = 0,024; \frac{Z}{C} = -0,024; \frac{W}{C} = -0,004)$, près d'un échantillon, donc dans une zone où l'erreur est assez faible. Le minimum global est situé en

$$\alpha_{glob}^{*aero} = \left(\frac{X}{C} = +0,011; \frac{Z}{C} = +0,024; \frac{W}{C} = +0,024 \right). \quad (III.18)$$

En terme de performance, le minimum local est moins performant d'environ quelques dizaines de pourcents. Malgré la disparité de localisation de ces deux optima, on peut observer globalement deux tendances assez marquées sur la surface de réponse :

- il semble nécessaire d'avancer la nacelle par rapport à l'aile afin d'améliorer la traînée ($\frac{X}{C} > 0$),
- les mâts plus larges semblent plus performants ($\frac{W}{C} > 0$).

Le minimum global est situé loin des points de construction de la surface de réponse, ce qui implique une erreur prédite importante dans cette zone. L'erreur au minimum est d'environ 20 pourcents. Elle est du même ordre de grandeur que sur la surface de réponse structure. Contrairement au cas structure où le minimum était insensible à cette erreur, ce n'est plus vrai ici. Le minimum de la fonction plus l'erreur (LCB_{-1}) est situé en $(\frac{X}{C} = +0,004; \frac{Z}{C} = +0,024; \frac{W}{C} = +0,024)$ et le minimum de la fonction moins l'erreur (LCB_1) est situé en $(\frac{X}{C} = +0,017; \frac{Z}{C} = +0,024; \frac{W}{C} = +0,024)$ (Figure III.11). Malgré la concurrence entre les deux minima, la prise en compte de l'erreur ne semble pas avoir d'influence sur la localisation de l'optimum sur l'arête ($\frac{Z}{C} = +0,024; \frac{W}{C} = +0,024$). Bien que le minimum local de la fonction soit situé dans une zone où l'erreur est faible, alors que le minimum global est situé dans une zone où l'erreur est importante, le minimum de la fonction plus l'erreur ne s'écarte que très peu du minimum global de la fonction. L'erreur moyenne absolue sur le domaine vaut 0,13 point de traînée.

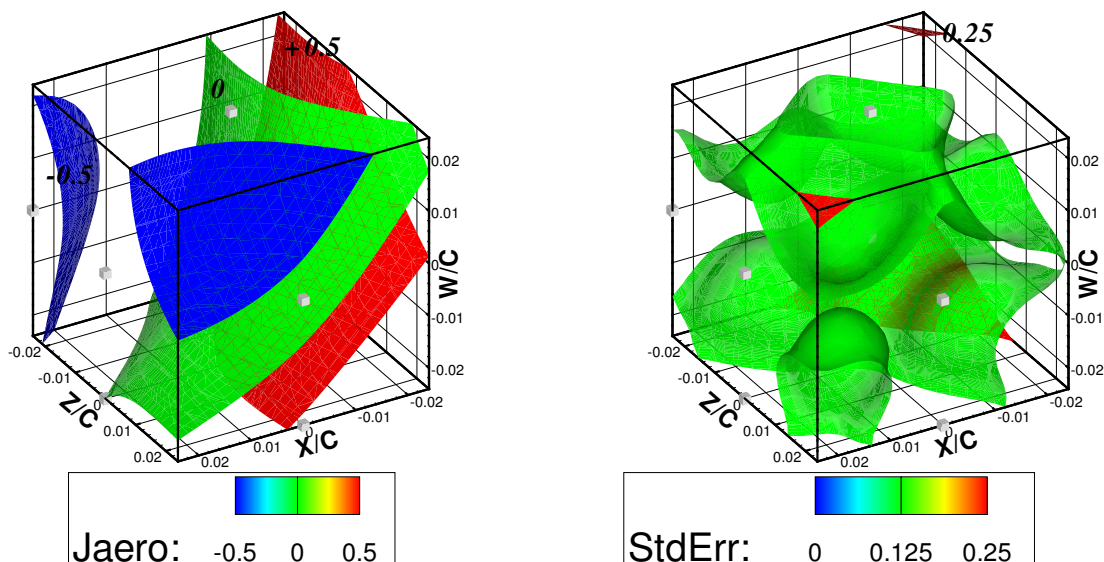


FIG. III.10: Surface de réponse aérodynamique et son erreur standard en valeur absolue.

III.3.5.3 Résultats de l'optimisation haut-niveau

Au niveau multidisciplinaire, le compromis entre les deux disciplines étudiées semble facile à déterminer puisque l'optimum structure et l'optimum aérodynamique sont situés sur la même arête du

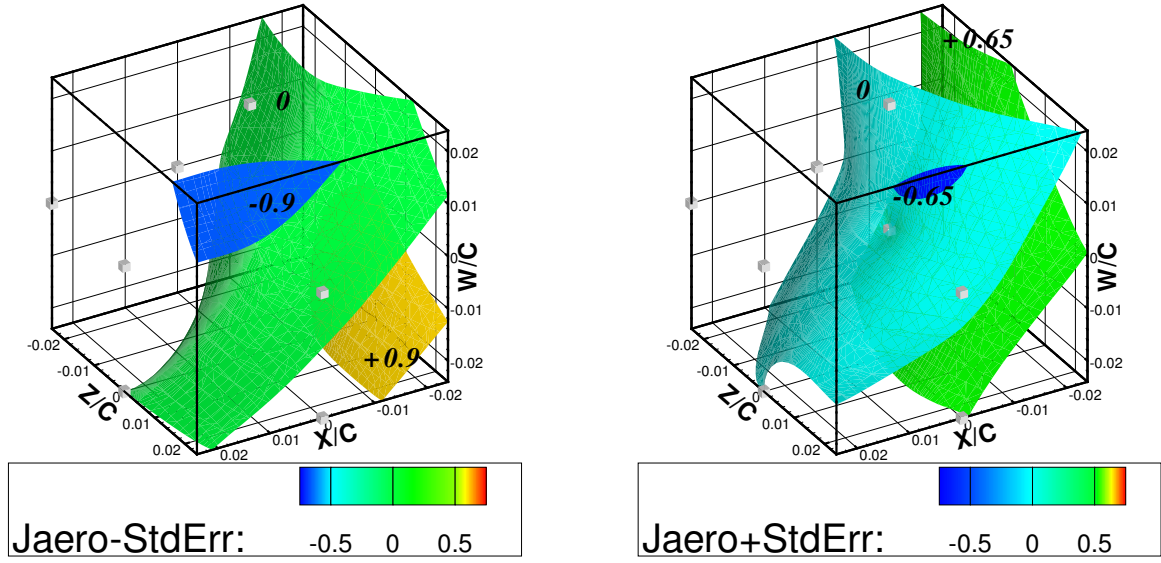


FIG. III.11: A gauche, surface de réponse aérodynamique plus son erreur standard. A droite, surface de réponse aérodynamique moins son erreur standard.

domaine, en $(\frac{Z}{C} = +0,024; \frac{W}{C} = +0,024)$. A priori, il suffit donc de trouver un compromis pour la position axiale de la nacelle $\frac{X}{C}$. La fonction objectif multidisciplinaire (Figure III.12), J , est obtenue par simple combinaison linéaire des surfaces de réponse structure et aérodynamique selon l'équation III.15. Le minimum global est situé en

$$\alpha_{glob}^* = \left(\frac{X}{C} = +0,011; \frac{Z}{C} = +0,024; \frac{W}{C} = +0,024 \right). \quad (\text{III.19})$$

L'optimum du problème couplé correspond à l'optimum du problème aérodynamique non-couplé. Aussi, au voisinage de l'optimum, la surface de réponse globale (Figure III.12) est semblable à la surface de réponse aérodynamique (Figure III.10). Cela indique que la fonction objectif choisie n'est pas équilibrée du point de vue multidisciplinaire. Le facteur d'équivalence masse/trainée, k , de cet avion est trop faible pour cela.

La valeur de référence de ce coefficient k est fixée par les caractéristiques de l'avion étudié. Le processus d'optimisation choisi permet de faire varier librement ce coefficient sans pour autant effectuer de calculs supplémentaires. Ainsi, la figure III.13 représente les surfaces de réponse correspondant aux formulations suivantes :

$$\begin{cases} J = J_{aero} + \frac{1}{5} (2k J_{struct}) \\ J = J_{aero} + 12 (2k J_{struct}) \\ J = J_{aero} + 20 (2k J_{struct}) \end{cases} . \quad (\text{III.20})$$

Si l'on se réfère à la théorie telle que décrite en I.1.2, l'équation initiale (Equation III.15) pour la fonction objectif haut-niveau est définie par le vecteur d'équivalence $\theta = (1; 2k)$ et la figure III.13 correspond aux vecteurs d'équivalence

$$\begin{cases} \theta = (1; 0,4k) \\ \theta = (1; 24k) \\ \theta = (1; 40k) \end{cases} . \quad (\text{III.21})$$

Le problème initial étant déjà dominé par l'aérodynamique, la diminution du poids accordé à la réponse structure par un facteur dix ne modifie pas l'allure de la surface de réponse haut-niveau et la position de l'optimum reste inchangée. A l'inverse, en multipliant le coefficient d'échange par un

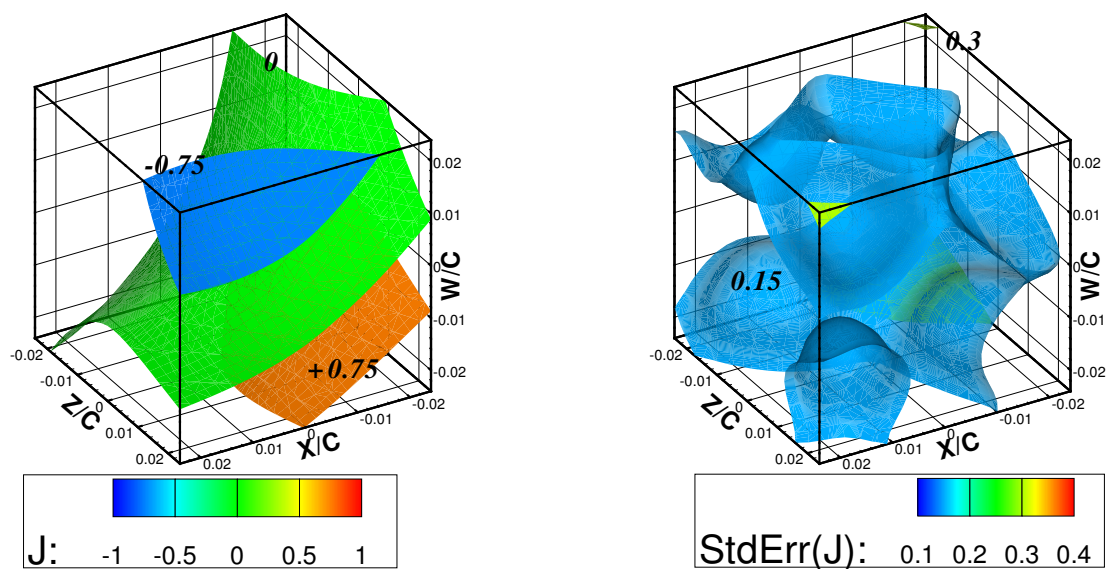


FIG. III.12: Surface de réponse globale et son erreur standard en valeur absolue.

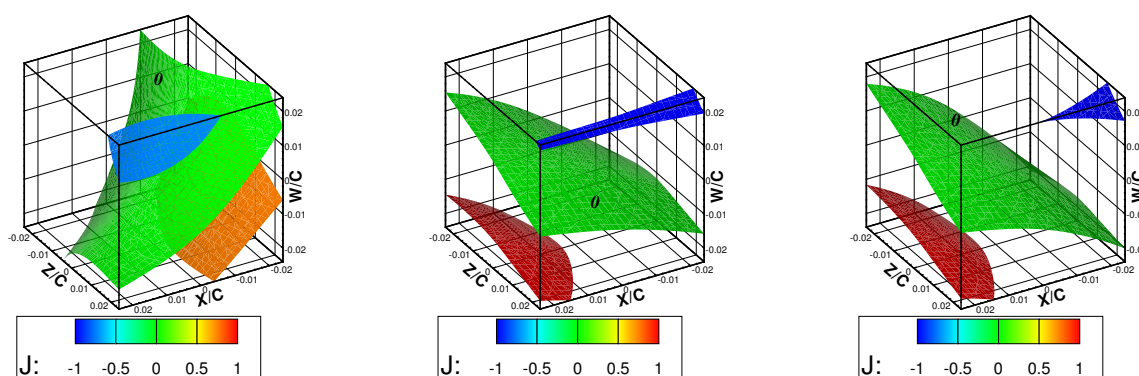


FIG. III.13: Influence du coefficient d'équivalence masse/trainée, k . A gauche, $k = \frac{k}{5}$, au centre $k = 12k$ et à droite $k = 20k$.

facteur dix, le problème est alors dominé par la réponse structure ; l'optimum haut-niveau correspond à l'optimum structure. Finalement, il semble que le coefficient d'échange doive être multiplié par un facteur six pour que les deux disciplines entrent vraiment en compétition.

Cela s'interprète aisément en traçant le domaine des solutions admissibles et le front de Pareto aérodynamique/structure (Figure III.14). L'optimum de Pareto défini par ce vecteur se trouve à l'intersection entre l'hyperplan³ support de normale $(1; 4k)$ et le front de Pareto (cf. I.1.2). Tel que représentée en figure III.14, cette intersection indique bien le minimum de la fonction aérodynamique seule. Le fait que le problème soit dominé par l'aérodynamique s'explique aussi par le fait que cet hyperplan n'est pas tangent au front de Pareto. Le front de Pareto est défini par un segment de normale $(1; 24k)$.

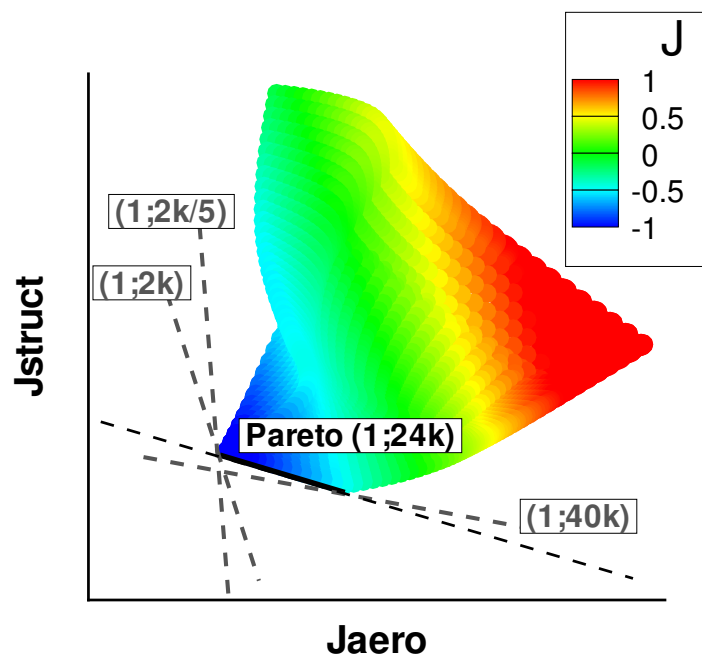


FIG. III.14: Domaine admissible, front de Pareto et hyperplans supports correspondant à 3 vecteurs d'équivalence différents.

III.3.6 Bilan

L'équation III.15 est aussi utilisée afin de calculer une erreur standard sur la fonction objectif globale, telle que représentée en figure III.12. L'erreur des deux surfaces de réponse étant cumulée, l'erreur prédite sur la fonction globale est très importante autour de l'optimum (environ 30 pourcents). Ainsi, cet optimum n'est pas robuste si l'on prend en compte l'erreur :

- Le minimum de la fonction plus l'erreur (LCB_{-1}) est situé en $(\frac{X}{C} = +0,004; \frac{Z}{C} = +0,024; \frac{W}{C} = +0,024)$.
- Le minimum de la fonction moins l'erreur (LCB_1) est situé en $(\frac{X}{C} = +0,02; \frac{Z}{C} = +0,024; \frac{W}{C} = +0,024)$.

L'erreur standard étant une erreur approchée, elle risque d'être d'autant plus inexacte que le Krigage est inexact. Ainsi, le niveau élevé de l'erreur approchée ne peut être négligé : 30 pourcents autour de l'optimum et 17 pourcents en moyenne.

³Pour un problème à deux objectifs, l'hyperplan est une droite.

Par exemple, en considérant le double de l'erreur standard pour la surface de réponse aérodynamique, il apparaît sur la figure III.15 que l'optimum indiqué par le critère « fonction plus erreur » (LCB_{-1}) change radicalement. L'optimum local devient alors l'optimum global.

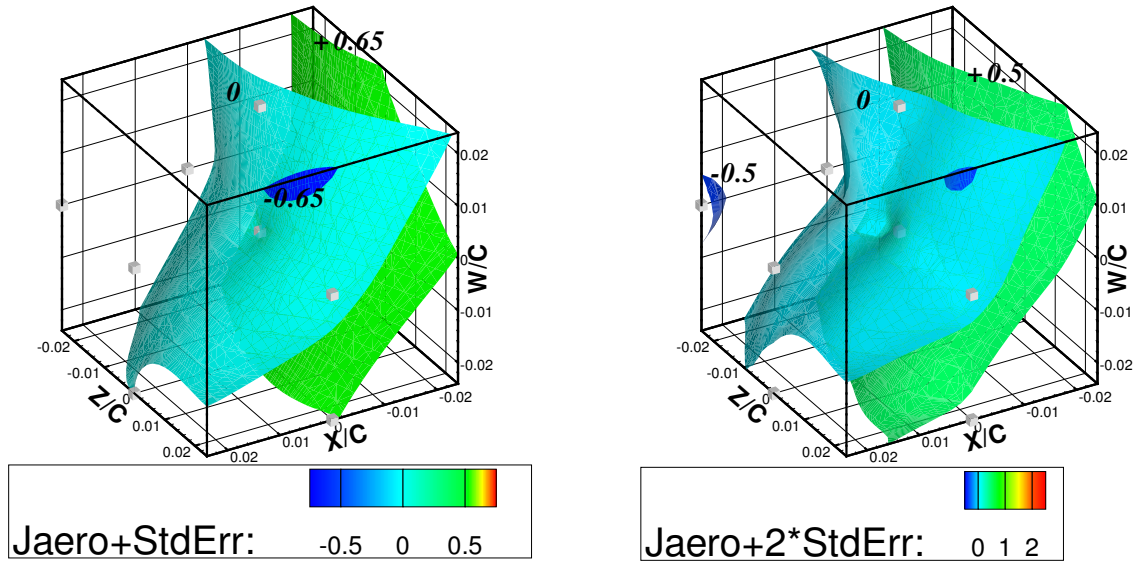


FIG. III.15: Surface de réponse aérodynamique : fonction plus erreur à gauche et fonction plus deux fois l'erreur à droite.

Les surfaces de réponses globales considérées exactes sur tout le domaine sont ici imprécises. Il semble donc nécessaire de valider la performance de l'optimum indiqué par les surfaces de réponses en le calculant par les logiciels d'analyse bas-niveau. Cela signifie qu'il faut effectuer une boucle itérative au niveau du système telle qu'effectuée par le processus BLISS original (Figure III.1). Cette boucle itérative n'a malheureusement pas pu être implémentée dans notre cas, car une des étapes du processus (Figure III.7) n'a pas pu être automatisée. La génération automatique du maillage aérodynamique correspondant à une nouvelle position de l'installation motrice (variables haut-niveau) n'était pas suffisamment robuste.

Pour éviter cette boucle itérative, il aurait fallu que, dès la première itération, les variables globales optimales fournies par l'optimiseur correspondent à une solution déjà calculée (un échantillon). C'est le critère de convergence du processus BLISS (Figure III.1). Pour cela, il aurait fallu construire des surfaces de réponses globales effectivement fiables sur tout le domaine. Par exemple en utilisant un échantillonnage plus dense. En général, la recommandation est de dix points par direction, soit 1000 échantillons (cf. II.3.2.1). Toutefois, le coût représenté par 1000 analyses bas-niveaux est inacceptable. Une autre manière d'éviter cette boucle aurait été d'inclure du savoir-faire, de l'intuition, dans la distribution des échantillons. Le domaine d'étude étant restreint, il était possible de prévoir que la solution se trouverait sur un bord du domaine. Avec 8 échantillons, il aurait été préférable d'échantillonner tous les coins du domaine au lieu d'utiliser une méthode de plan factoriel.

Finalement, la considération de la robustesse des résultats vis-à-vis de l'erreur approchée permet d'affirmer que l'optimum du système se trouve sur l'arête définie par les plans $\frac{Z}{C} = +0,024$ et $\frac{W}{C} = +0,024$.

La résolution de ce problème par la méthode BLISS a montré que les surfaces de réponse étaient un moyen simple et efficace de réaliser le couplage entre disciplines. La flexibilité du processus BLISS semble adaptée à une organisation industrielle, à condition que la boucle itérative soit effectivement possible.

Ce cas test multidisciplinaire s'est finalement avéré trop simplifié puisqu'il est largement dominé par l'aérodynamique. D'après le front de Pareto approché, il aurait fallu pouvoir obtenir un gain de masse 12 fois plus important pour que les disciplines entrent en concurrence. Cette variation plus importante de la masse pourrait être obtenue en considérant un système plus complexe. Par exemple, en incluant l'effet de la position du moteur sur le poids de la structure primaire de l'aile. Cela illustre le fait qu'un problème d'optimisation multi-objectif équilibré du point de vue physique, n'est pas nécessairement équilibré du point de vue optimisation. Par rapport aux méthodes de détermination d'équivalence entre objectifs décrites en I.1.2, la méthode de détermination d'équivalence par variation admissible aurait donnée un problème équilibré du point de vue optimisation.

Comparaison d'algorithmes d'optimisation par gradient et surface de réponse locale

Nomenclature

cf. IV.2.Colloque

IV.1 Introduction

Dans un contexte industriel, les concepteurs de formes aérodynamiques ont besoin d'outils leur permettant de réduire la traînée en un temps limité. L'optimisation permet cela au coût de plusieurs dizaines d'évaluations de traînée. Ce coût se justifie par la complexité de la relation liant une forme à sa traînée. Cette complexité met généralement en défaut la méthode de conception empirique par essais et erreurs (« trial and error method »).

L'évaluateur étant un logiciel de simulation numérique, la contrainte de temps se traduit par un nombre maximum d'évaluations de fonctions. Tel qu'introduit en I.2, cela signifie qu'il faut choisir l'algorithme d'optimisation proposant le meilleur compromis entre exploration et exploitation. L'optimiseur doit être adapté aux particularités des problèmes traités [121]. Etant donné nos problèmes, l'optimiseur doit parvenir à la meilleure amélioration (exploration) en respectant le temps qui lui est imparti (exploitation).

Dans le contexte de la conception de formes aérodynamiques, l'ensemble des contraintes suivantes doivent être prises en compte :

- Le coût de calcul requis pour une évaluation de fonction est important. Il faudra donc trouver une solution en moins de 200 évaluations.
- Le coût de calcul requis pour une évaluation de gradient est important (identique au coût d'une évaluation de fonction grâce à la méthode adjointe).
- Les fonctions admettent de multiples minima locaux et sont très sensibles. De petites perturbations de la forme peuvent générer de fortes variations de traînée (fort gradient). Il faudra donc, autant que possible, que l'optimiseur soit performant en terme d'exploration (cf. I.2.2).
- Le nombre de variables de forme est important.

Un avion en condition de vol de croisière évolue dans un écoulement transsonique. Dans ces conditions, la traînée est très sensible aux variations locales de la forme. Cela peut s'observer sur le cas d'optimisation d'avion complet en IV.3. L'impact de déformations locales de l'ordre de 50 millimètres sur une aile d'envergure 25 mètres et de corde 5,4 mètres peut aller jusqu'à une pénalisation en traînée totale

de l'avion de 35 pourcents (figure IV.9). La paramétrisation de la forme doit permettre de définir un champ surfacique de vecteurs déformations sur la forme. Ce champ doit pouvoir être suffisamment complexe pour pouvoir améliorer les performances de la forme, tout en évitant les formes trop irrégulières qui sont irréalistes. Il est défini par quelques dizaines de bosses de Hicks-Henne, soit en général de l'ordre de 100 variables.

Les méthodes d'optimisation multi-objectif de Pareto permettent, pour les problèmes de conception aérodynamique, d'optimiser simultanément la traînée et l'inverse de la portance. A partir du front de Pareto, il est possible de définir la forme offrant le meilleur compromis entre ces deux fonctions, c'est-à-dire la meilleure finesse aérodynamique. Toutefois, ces méthodes [34, 77, 90] sont exclues de ce travail. Etant donné le faible nombre de calculs autorisés pour réaliser l'optimisation (200) et la complexité du problème (nombre important de variables), il n'est pas possible de déterminer précisément un front de Pareto. Les méthodes d'optimisation étudiées sont mono-objectif.

L'algorithme de référence est DOT-BFGS (cf. I.3.7 et IV.Colloque.II.C) qui utilise une méthode de gradient type quasi-Newton BFGS. Cet optimiseur est très performant en terme d'exploitation. En général, il converge à un coût de calcul équivalent à 100 évaluations. Cet algorithme est purement séquentiel, il propose les nouvelles formes une par une ($n_{pop} = 1$). La durée d'une optimisation dépend donc linéairement du nombre de formes évaluées. Il s'accommode bien des problèmes à grand nombre de variables (quelques centaines). Toutefois, c'est un algorithme d'optimisation local. Sur nos problèmes, il a été vérifié qu'il converge vers un minimum local dépendant de la forme initiale et effectue peu d'exploration. Afin d'améliorer les capacités d'exploration du logiciel d'optimisation, deux nouveaux optimiseurs basés sur des surfaces de réponse ont été développés et sont décrits en IV.Colloque.III.

IV.1.1 Environnement de l'étude comparative

Un environnement générique d'optimisation a été mis en place afin de pouvoir connecter des optimiseurs hétéroclites à un unique évaluateur. Il permet de mesurer équitablement les performances d'optimiseurs variés sur des problèmes d'optimisation de formes aérodynamiques. Il est décrit en IV.Colloque.II.A et fait appel au logiciel OPTaliA (cf. I.3).

Cet environnement gère la soumission de calculs sur machines multiprocesseurs en parallèle. Pour ce faire, le processus utilise un système de synchronisation par fichier entre un gestionnaire d'évaluateurs et un optimiseur. Ainsi, les algorithmes utilisant une population de plusieurs individus par itération ($n_{aug} > 1$) seront avantagés en terme de temps de restitution. Il conviendra donc d'utiliser le nombre d'évaluations, n_{eval} , pour mesurer le coût de calcul et le nombre d'itérations, n_{iter} , pour mesurer le temps de restitution de l'optimisation. D'autre part, dans ce chapitre le nombre d'itérations, n_{iter} , se réfère aux itérations du processus d'optimisation (cf. I.6 ou IV.2.Colloque.Fig.1). Il est inutile de mesurer le nombre d'itérations interne d'un algorithme puisque ce nombre est spécifique à chaque méthode.

Les performances des optimiseurs ne seront pas seulement établies en considérant la valeur finale obtenue en fin d'optimisation. En effet, des critères de mesure de performance en terme d'exploration et d'exploitation sont introduits en IV.2.Colloque.II.D :

- Pour l'exploitation, on considérera le taux d'amélioration par évaluation (ou par itération), $\frac{F_{ini} - F_{ref}}{n_{eval}}$. Il est calculé par simple rapport entre l'amélioration totale et le nombre total d'évaluations (ou d'itérations) plus le nombre de calculs de gradients par la méthode adjointe.
- Pour l'exploration, on considérera la variance des paramètres de forme adimensionnés $\sum \|\bar{x}_c - \bar{x}_c^{ref}\|/n_{dv}$ (ou de la fonction, $\sum |F_c - F_c^{ref}|$) au cours de l'optimisation. Elle est obtenue en moyennant, à chaque évaluation, la différence entre la valeur courante proposée par l'optimiseur et la meilleure solution connue à cette étape du processus. L'exploration représente la capacité qu'a l'optimiseur de s'éloigner, dans l'espace de la fonction, de son optimum courant. Sur les historiques de convergence représentés en figures IV.2.Colloque.Fig.7, IV.2.Colloque.Fig.10 et IV.9, l'exploration se lit comme la moyenne des distances verticales entre les symboles et la

courbe. Généralement, l'exploration est une quantité mesurée dans l'espace des variables. Comme ici les variables sont de nature différentes (amplitude, position, expansion d'une bosse) et ne sont pas normalisées, l'exploration est mesurée dans l'espace des valeurs de la fonction. Sur les cas étudiés dans ce chapitre, il semble que l'exploration mesurée dans l'espace de la fonction soit effectivement corrélée à l'exploration mesurée dans l'espace des variables non normalisées (cf. tableaux IV.2.Colloque.Tab.4, IV.2.Colloque.Tab.4 et IV.2). La quantité d'exploration divisée par le nombre de formes explorées (nombre d'évaluations) sera aussi utilisée.

IV.1.2 Optimiseurs par surfaces de réponse locales

Il a été établi au chapitre II que l'utilisation de surfaces de réponse globales pour l'optimisation aérodynamique n'était pas adaptée, même lorsque le nombre de variables est faible. Cela a pu être vérifié sur l'application multidisciplinaire réalisée au chapitre III. La complexité des fonctions aérodynamiques semble nécessiter une taille d'échantillonnage trop importante pour être complètement modélisée. L'algorithme présenté ici utilise donc des surfaces de réponse locales (fiables seulement aux alentours des points de construction) au sein d'un processus itératif.

A chaque itération, un modèle approché de la fonction objectif est construit par une méthode de Krigeage. L'information fournie par ce modèle permet de trouver des points où la fonction n'a pas encore été calculée mais qui semblent prometteurs. Pour cela, trois critères de recherche sont utilisés (cf. II.3.2.3 et IV.2.Colloque.III.C.1) :

- le minimum de la fonction,
- le minimum du critère LCB_1 , c'est-à-dire la fonction moins l'erreur standard,
- le minimum du critère LCB_{-1} , c'est-à-dire la fonction plus l'erreur standard.

Le fait d'utiliser trois critères de recherche par itération permet, d'une part de combiner exploitation (minimisation de la fonction approchée) et exploration (prise en compte de l'incertitude) au sein d'une seule itération et d'autre part d'accélérer le temps de restitution de l'optimisation. Le minimum global de ces critères est recherché et aucune restriction sur le domaine des variables n'est effectué. Les optimiseurs par surfaces de réponse locales recherchent donc le minimum global de la fonction. Ce sont des algorithmes d'optimisation globale.

La procédure employée pour minimiser ces critères est détaillée en IV.2.Colloque.III.B. Elle vise à déterminer le minimum global du critère, même pour un nombre de variables important. Cette stratégie opère en construisant un ensemble des optima probables construit en répétant plusieurs optimisations globales par différents algorithmes : exploration aléatoire et algorithme génétique. Ensuite, le minimum de cet ensemble est retenu. Finalement, une optimisation locale par gradient est lancée à partir de ce point pour converger précisément vers le minimum détecté lors des étapes précédentes.

Le Krigeage devient impossible à calculer lorsque les points de construction sont trop proches. Cela s'explique par un mauvais conditionnement de la matrice de corrélation (cf. II.Article.II.E). Pour éviter cette situation, avant d'effectuer le raffinement de l'échantillonnage, une vérification de la distance par rapport aux points déjà échantillonnés est réalisée. Lorsque cette distance tombe en dessous d'un certain seuil, le point est rejeté (cf. IV.2.Colloque.III.C.2). En pratique, le nombre de points ajoutés à chaque itération est parfois inférieur à trois.

Au cours de la première itération, la base de données est initialisée par un échantillonnage de type remplissage d'espace (cf. II.3.2.3). Cet échantillonnage est très grossier (cf. II.3.2.1), sa densité moyenne est inférieure à un point par direction. Une méthode d'échantillonnage *a priori* est utilisée de sorte que tous les calculs de la première itération sont lancés en parallèle. Cet échantillonnage a pour objet de capturer la ou les principales tendances de la fonction. Il ne permet pas de construire une surface de réponse globale, fiable sur tout le domaine. Aucune validation de la précision de la surface de réponse n'est réalisée.

En pratique, les concepteurs partent d'une forme initiale qu'ils veulent améliorer. Cette forme initiale est donc incluse dans la base de données initiale. Elle sert aussi de point de départ pour

l'algorithme d'optimisation par gradient DOT-BFGS.

Afin de mieux capter les tendances de la fonction pour les problèmes à grand nombre de variables, une méthode de Cokrigeage est employée. Le processus reste alors inchangé, si ce n'est que certains des échantillons fournissent l'information sur le gradient de la fonction objectif. Pour les raisons de coût de calcul de la surface de réponse abordées en II.3.3, une formulation de Cokrigeage indirect tronqué en échantillon est employée (IV.2.Colloque.III.A.3). Le coût de calcul de la surface de réponse peut ainsi être adapté en fonction du problème en choisissant le nombre de points de construction utilisant le gradient, n_{aug} . L'intérêt de ce nouvel optimiseur, basé sur la formulation de Cokrigeage (« RS Cokriging »), est discuté à partir du gain de performance constaté en comparaison de l'optimiseur basé sur le Krigeage (« RS Kriging »).

IV.2 Acte de colloque « Comparison of Gradient and Response Surface Based Optimization Frameworks Using Adjoint Method »

L'acte de colloque numéro « AIAA 2008-1889 » ci-après décrit le travail présenté en avril 2008 à Schaumburg (USA) lors d'une conférence sur l'optimisation multidisciplinaire organisée par l'AIAA¹ (« 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference ») [72]. Ce papier introduit deux nouveaux optimiseurs mono-objectifs basés sur le Krigeage et le Cokrigeage et prouve leur efficacité en les comparant à un algorithme de référence, DOT-BFGS.

L'introduction du Cokrigeage est assez récente (2002-2003) [76] en optimisation de formes aérodynamiques, bien que la formulation ait été établie en 1993 [86]. A l'heure actuelle, une seule application du Cokrigeage couplé à un solveur adjoint a été effectuée par S. Kim et H.-S. Chung en 2006 [65]. Toutefois, le problème considéré était bi-objectif et le Cokrigeage était utilisé pour calculer un front de Pareto. Le processus décrit n'était pas itératif et les surfaces de réponse étaient globales. Les auteurs suggèrent que, sur leur cas aérodynamique à 6 variables, les surfaces de réponse par Cokrigeage construites avec 12 échantillons donnent des résultats comparables à ceux obtenus par des surfaces de réponses par Krigeage construites avec 29 échantillons. Les applications de Chung *et al.* [22, 23] ne décrivent pas non plus d'algorithme d'optimisation basé sur le Cokrigeage, puisqu'un raffinement itératif au minimum de la surface de réponse est simplement suggéré.

Très peu d'optimisations basées sur les surfaces de réponse ont été effectuées avec plus de 20 variables [12, 67, 98]. Or, les deux cas tests considérés en IV.2.Colloque.IV montrent les performances des optimiseurs en dimension 6 et 45. En outre, un autre exemple en dimension 48 est traité en IV.3.

Une autre originalité de ce travail vient du fait que différents optimiseurs sont comparés sur des cas tests représentatifs des problèmes de conception aérodynamique. Dans ce contexte, seulement deux études comparatives existent, celle de Widhalm *et al.* [119] (2007) et celle de Epstein *et al.* [35] (2008). Les comparaisons d'algorithmes d'optimisation sont généralement réalisées à partir de fonctions tests [78], par exemple grâce à celles disponibles dans l'environnement *CUTEr* [42]. Lorsque des problèmes de conception de formes aérodynamiques en régime d'écoulement transsonique sont considérés, la plupart des articles se contentent de montrer la validité d'un optimiseur particulier sur quelques cas donnés.

Epstein *et al.* [35] ont comparé trois logiciels d'optimisation aérodynamique différents pour minimiser la traînée d'une aile :

- Le logiciel SYN107 de Intelligent Aerodynamics² est basé sur un optimiseur par gradient et fait appel à une méthode d'adjoint continu pour le calcul de gradient [51].

¹American Institute of Aeronautics and Astronautics, <http://www.aiaa.org/~>.

²Stanford University

- Le logiciel MDOPT de Boeing est basé sur des surfaces de réponse construites par une méthode de Krigeage [74].
- Le logiciel OPTIMAS de Israel Aerospace Industries est basé sur un modèle d'ordre réduit couplé à un algorithme génétique [92].

Ces environnements d'optimisation utilisent chacun un code de calcul aérodynamique différent. La résolution du maillage est, de plus, très variables selon les codes, de $0,25 \cdot 10^6$ à $4 \cdot 10^6$ nœuds. La méthode de paramétrisation est différente dans chacun de ces logiciels. De ce fait, le nombre de variables de forme varie selon la méthode considérée. En raison de la grande diversité des outils utilisés, la cohérence de la comparaison est donc discutable. Elle ne fournit malheureusement pas de conclusions quant aux qualités respectives de chacun de ces outils. Par contre, elle met en évidence à quel point il est difficile, en pratique, de séparer l'évaluateur (le code CFD³) et l'optimiseur. C'est une spécificité des problèmes aérodynamiques, due à la complexité des codes de calcul utilisés. Le but de cette collaboration tripartite était de lancer un groupe de travail sur l'optimisation aérodynamique. Les auteurs proposent donc un cas test sur une géométrie appartenant au domaine public et des données d'optimisations réalisées à partir de cette géométrie. Ces travaux étant très récents (2008), il n'a pas été possible de résoudre ce problème avec l'outil OPTaliA.

L'étude comparative de Widhalm *et al.* [119] décrit une véritable comparaison de deux algorithmes d'optimisations. L'environnement d'optimisation y est effectivement commun (un seul évaluateur) et les optimiseurs peuvent être comparés équitablement. Un optimiseur de type simplex (Subplex, chemin de descente sans gradient) est comparé à un optimiseur par gradient (méthode « MMFD, Modified Method of Feasible Direction » de type gradient conjugué). Les problèmes traités consistent à minimiser la traînée, à portance constante, pour une aile (ONERA M6) et une aile volante. L'évaluateur résout le système d'équations d'Euler. Le vecteur gradient est calculé par la méthode de l'adjoint continu. Sur les 4 applications considérées, le nombre de variables est de 7, 12, 13 et 30. Le cas le plus complexe, avec 30 variables, n'a été traité qu'avec la méthode de gradient. Lors de leur comparaison, ils ont constaté que le simplex atteignait une meilleure solution, alors que l'optimiseur par gradient était beaucoup plus rapide à converger (grâce à l'adjoint). Autrement dit, l'optimiseur par gradient est très efficace en exploitation, mais la méthode de simplex semble faire plus d'exploration. Elle converge plus lentement, mais vers une meilleure solution. Aucune considération sur l'exploration n'est effectuée dans cette étude comparative. Vis-à-vis des formes en plan optimales pour l'aile volante, il semble que les formes données par le simplex soient plus éloignées de la forme initiale que celles données par l'optimiseur par gradient.

Enfin, le travail présenté ici se démarque des travaux existants par le processus de raffinement multi-critères adopté. La plupart des optimiseurs basés sur le Krigeage utilisent un seul critère de raffinement par itération, le plus souvent le critère EI ou directement le minimum de la fonction. Le fait d'ajouter plusieurs échantillons par itération n'est pas nouveau. Plusieurs exemples de couplage entre algorithme génétique et surfaces de réponse (algorithme hybride) l'ont déjà mis en œuvre [57, 89, 102, 112]. Par contre, ceux-ci n'utilisent qu'un seul critère de raffinement. Ils utilisent un algorithme de commutation pour effectuer certaines itérations de l'optimiseur sur le vrai évaluateur (le code de calcul) et les autres sur la surface de réponse. Cela est particulièrement adapté aux algorithmes qui font évoluer une population de plusieurs formes par itération comme les algorithmes génétiques.

³Computational Fluid Dynamics

Comparison of Gradient and Response Surface Based Optimization Frameworks Using Adjoint Method

J. Laurenceau ^{*}, M. Meaux [†]
Airbus, Toulouse, 31060, France

This paper deals with aerodynamic shape optimization using an high fidelity solver. Due to the computational cost and restitution time needed to solve the RANS equations, this type of optimization framework must improve the solution using very few objective function evaluations despite the high number of design variables. The choice of the optimizer is thus largely based on its speed of convergence. The quickest optimization algorithms use gradient information to converge along a descent path departing from the baseline shape to a local optimum. Within the past few decades, numerous design problems were successfully solved using this method. In our framework, the reference algorithm uses a quasi-Newton gradient method and an adjoint method to inexpensively compute the sensitivities of the functions with respect to shape variables. As usual aerodynamic functions show numerous local optima when varying shape, a more global optimizer can be beneficial at the cost of more function evaluations. More recently, the use of expensive global optimizers became possible by implementing response surfaces between optimizer and CFD code. In this way, a Kriging based optimizer is described. This optimizer proceeds in iteratively refining at up to three points per iteration by using a balancing between function minimization and error minimization. It is compared to the reference algorithm on two drag minimization problems. The test cases are 2D and 3D lifting bodies parameterized with six to more than forty design variables driving deformation of meshes with Hicks-Henne bumps. The new optimizer effectively proves to converge to lower function values without prohibitively increasing the cost. However, response surfaces are known to become inefficient when dimension increases. In order to efficiently apply this response surface based optimizer on such problems, a Cokriging method is used to interpolate gradient information at sample locations.

Nomenclature

$C(x)$	Sampling refinement criterion
Cd	Drag coefficient
Cl	Lift coefficient
Cp	Pressure coefficient
c	Mean chord length
\mathcal{D}	Domain of design variables
F_s	Exact function at samples $[N]$
$F(x)$	Objective function
$H(x)$	Hessian matrix of the objective function
L	Likelihood estimate
N	Order of the correlation matrix / quantity of information
n_{aug}	Number of gradient augmented samples

^{*}Ph.D. Student, Aerodynamics department, Airbus France, 316 route de Bayonne.

[†]Engineer, Aerodynamics department, Airbus France, 316 route de Bayonne.

n_{dv}	Number of design variables
n_{eval}	Number of function evaluations
n_{grad}	Number of gradient evaluations
n_{iter}	Number of iterations of the optimization process
n_{pop}	Optimizer population size
n_s	Number of samples
\mathcal{P}	Set of possible optima on the response surface
R	Correlation matrix [$N \times N$]
$r(x)$	Correlation vector [N]
\mathcal{S}	Domain of sample points
$SCF(.,.)$	Spatial correlation function
$S(x)$	Standard error
s^i	i-th sample
x	Vector of design variables [n_{dv}]
$Z(x)$	Stochastic process
β	Zero order regression model
σ^2	Model variance
θ	SCF correlation coefficients [n_{dv}]
$\ \cdot\ $	Euclidian norm; 2-norm

Subscripts

i	$\in [1, n_s]$
j	$\in [1, n_s]$
k	$\in [1, n_{iter}]$
c	$\in [1, n_{eval}]$
r	$\in [1, n_{pop}]$
v	$\in [1, n_{dv}]$

Superscripts

ini	initial sampling database
$\hat{\cdot}$	approximated value
ref	current best value

I. Introduction

In the field of aerodynamic aircraft design, the functions studied are very sensitive to small changes on the shape and it is then particularly hard for designers to reach an optimal solution by trial-and-error. Shape optimization tools are thus particularly favoured by aerodynamicians. These tools are completely automatic process capable of running by themselves computer expensive numerical simulation given some degrees of freedom on the geometry and a figure of merit qualifying the performance of each shape. In the context of detailed design, it was decided that the same level of fidelity of the analysis code should be used for absolute performance assessment and for shape optimization purpose. As computational cost of CFD problems generally increases with computational resources (size of meshes can be adapted to available computational power), the typical restitution time required for one flow analysis cannot be sufficiently decreased to enable the use of expensive global optimization algorithms requiring thousands objective function evaluations such as genetic algorithms. This drawback can be circumvented by using the shape given by studies performed during conceptual and preliminary design phases as a starting point for a gradient based local optimization. The first aerodynamic shape optimizations by Hicks et al.¹ demonstrate the efficiency of this type of process, still in use nowadays²⁻⁵ in combination

with efficient gradient calculation techniques based on adjoint method maintaining the cost of the optimization independent of the number of design variables.

Despite its speed of convergence, gradient based algorithms are known to lack design space exploration and are easily trapped by local optima. As the non-linear physical phenomena occurring in transonic flows imply numerous local optima on aerodynamic functions (drag, lift, momentum), a significant gain is expected by the use of more global optimizers. The use of such optimizers for applications driven by analysis of full Navier-Stokes equations was made possible more recently by the use of surrogate models (or response surfaces)⁶⁻⁹ approximating the expensive CFD function by an inexpensive to evaluate black-box model. Moreover, the use of response surfaces coupled with a multi-objective global optimizer enables to have a clearer view of possible trade-offs between concurrent objective functions¹⁰ and facilitates multidisciplinary analysis,^{11,12} whereas gradient based algorithms are limited to mono-objective problems and solved multi-objective problems only by using pre-determined equivalence coefficients between functions.

In the section II of this paper is presented the high-fidelity optimization suite OPTaliA and the gradient based optimizer *DOT*. This algorithm uses a classical quasi-Newton method and is taken as the reference optimizer. Numerous applications were successfully conducted using this optimization framework,^{5,12} but to tackle the limitations underlined previously a response surface based optimizer was developed using a Kriging method and is presented in section III. The performance of this optimizer should largely exceed the reference on low-dimensional problems by using a multi-criteria refinement process enabling to run multiple CFD runs in parallel. As surrogate models are known to become inefficient as dimension increased, a third optimizer using a Cokriging (gradient-enhanced Kriging) formulation is presented. It uses a sample limited Cokriging approach. This formulation was set up to overcome the large computational cost needed to build a surrogate model interpolating high dimensional gradient vectors.

The last section (section IV), compares the two new response surface based optimizers to the gradient reference on a low-dimensional drag reduction problem considering 6 design variables on a RAE2822 airfoil and a high dimensional test problem considering 45 design variables on a wing.

II. Optimization suite

The software OPTaliA, internally developed at Airbus, is used to perform aerodynamic shape optimization. This high-fidelity optimization suite can improve aerodynamic performance of an aircraft by changes in the external shape (planform variables fixed) and is adapted to the work done during the detailed design phases.

A. Common optimization framework

A general optimization framework, represented in Figure 1, has been set up in OPTaliA, in order to implement various type of optimizers (gradient, genetic, response surface). From the global point of view, the optimization process can be interpreted as a succession of two main tasks: evaluation and optimization. Within the evaluator, the function value and if needed the gradient value corresponding to n_{pop} shapes are computed. The population size depends on the optimization algorithm chosen, but if it is superior to one the evaluator performs the simulations of all shapes simultaneously by running multiple jobs on high performance computers. A large population can reduce the restitution time of the optimization process, at the risk of saturating computational resources. Once all shapes have been evaluated, the optimizer proposes a new population of shapes by using the new information on functions and the next iteration begins. In addition to optimizer internal stopping criteria, the convergence is forced at the OPTaliA level when the number of iterations or the number of function evaluations exceeds a given threshold, $n_{iter} \leq 100, n_{eval} \leq 200$.

One of the challenges in aerodynamic shape optimization is to manage running efficiently

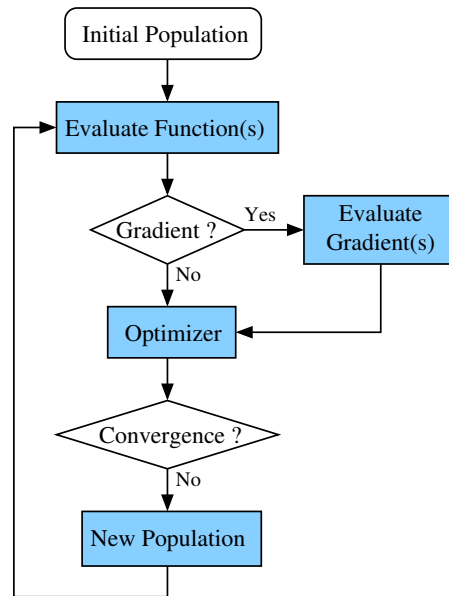


Figure 1. General Optimization Process

evaluator and optimizer automatically in batch mode. More particularly, the evaluator itself is a complex process requiring large computational resources.

B. Evaluator for CFD functions

1. Shape parameterization and mesh deformation

The shape parameterization consists in applying Hicks-Henne sinusoidal bumps on a surface skin of an initial block-structured mesh. Each bump is defined by three shape variables driving the amplitude, the position and the width expansion. The direction of the deformation can be either along the vector locally normal to the surface or along a fixed vector (vertical axis). This type of deformation was initially developed by Hicks et al.¹ for numerical optimization of airfoils. When applied on a bidimensional surface corresponding to a three dimensional shape, a linear propagation of the bump is done in the second direction using fixed propagation distances.

Once computed, the vector field of deformation at the surface skin is propagated to the volume mesh using a mixed integral / transfinite interpolation method.⁵ The integral method is used to compute deformation of nodes defining boundaries between blocks and then the transfinite interpolation computes deformation inside each block in parallel.

An analytical linearization of the shape parameterization and mesh deformation modules enables to inexpensively compute (in terms of CPU time) the sensitivity of the surface mesh and the sensitivity of the volume mesh with respect to design variables.

2. Flow simulation

Flow analysis were performed with the *elsA*¹³ software developed by Onera. The flow is simulated by solving the Reynolds Averaged Navier-Stokes (RANS) equations associated with the one-equation Spalart-Allmaras turbulence model on block structured meshes using a cell-centered finite volume approach. The second order Roe's upwind scheme with the Van Albada limiter is used as spatial scheme coupled with an implicit time resolution. Multigrid and local time stepping techniques are used to converge more quickly.

One of the main requirement from designers is to obtain the same results when using the CFD solver inside or outside the automatic optimization tool. As hysteresis phenomena are

common when dealing with transonic flows, it imposes that the same initial flow condition (uniform flow) are used for all simulations during the optimization. So, the computational cost of CFD simulations cannot be reduced by using a restart strategy using the flow solution corresponding to the previous shape. The computational cost of the optimization grows linearly with the number of function evaluations.

The sensitivity of the objective function with respect to the shape variables is computed using the discrete adjoint method¹⁴ of *elsA*. For an explicit presentation of the adjoint system solved the reader is referred to Peter et al.¹⁵ and Meaux et al.⁵ This method enables to compute the sensitivity of a single function with respect to n_{dv} design variables at the cost of one linear system resolution (same size as the RANS system). The gradient vector is thus computed using approximatively the same computational time (factor ≈ 1.5) as one direct flow simulation. For typical aerodynamic problems considering hundreds of design variables and a few functions (lift, drag), this is a considerable improvement over the classical method of finite differences requiring as many flow solutions as design variables.

3. Aerodynamic function computation

The objective function chosen is the far-field pressure drag,

$$F = Cd_p^{ff} = Cd_{vp} + Cd_w + Cd_i. \quad (1)$$

The friction drag, Cd_f , is excluded from the objective function as it does not significantly change considering small amplitudes of deformation. The wetted surface is kept unchanged (almost) by the shape deformation. The far-field code used is *ffd41*¹⁶ developed by Onera. The two main advantages of this approach are its ability to decompose pressure drag into physical components (wave drag, induced drag, viscous pressure drag) and its accuracy through a filtering of non-physical drag (spurious drag).

The post-processing module can also compute the sensitivity of the drag with respect to the flow variables and with respect to the mesh with an analytical formulation.

As aerodynamicians work at fixed lift rather than at fixed angle of attack, a more realistic problem should also take into account a fixed lift coefficient as a constraint for the optimizer and include the angle of attack as a design variable, but the present work compares two very different type of algorithms and the comparison must not be biased by different methods of constraints handling.

C. Gradient based optimizer, reference optimizer (DOT-BFGS)

The reference optimizer is gradient based and uses the classical quasi-Newton BFGS method (Broyden-Fletcher-Goldfarb-Shanno) from the DOT (Design Optimization Tools)¹⁷ library. Description of one internal step of this optimizer is given in Figure 2. As all gradient optimizers, it converges along a descent path until no improvement is achieved during one optimizer iteration or if the gradient norm is null. Firstly, the algorithm determines a descent direction, d_k , using the evolution of the gradient vector during the last two internal iterations. Once the direction is computed, a linear search aiming at computing the norm of displacement giving the best improvement is performed. The linear search is driven by a mono-dimensional polynomial interpolation and requires successive function evaluations. This type of optimizer is intrinsically sequential as it follows a single descent path. For the optimization framework, only one set of design variables is handled by process iteration. The process population contains only one individual ($n_{pop} = 1$).

In terms of quantity of information, the quasi-Newton gradient algorithm proposes the next set of variables by using only the information about the current internal iteration. The internal iteration contains information about the descent direction (computed using evolution of two gradient values) plus some function evaluations (usually no more than ten). This optimization algorithm proposes a new shape based on $N = 2n_{dv} + 10$ scalar informations on the unknown

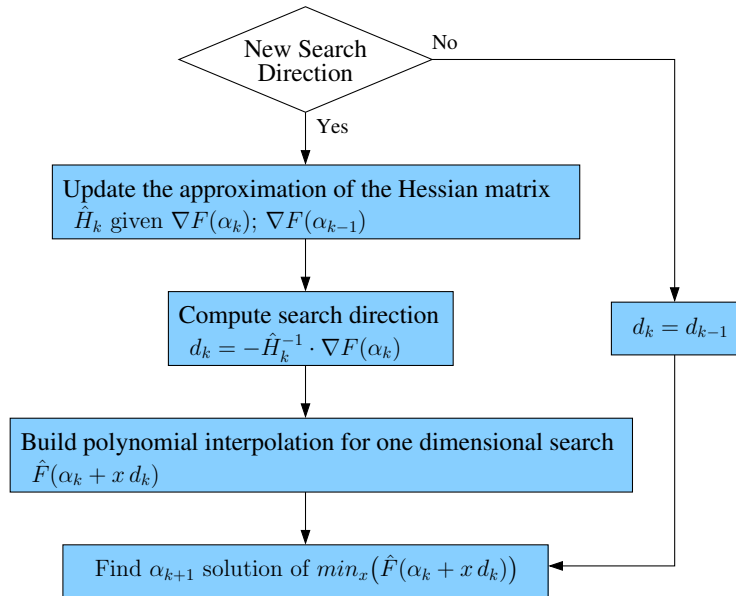


Figure 2. Inside the gradient based optimizer box

function. Even if the approximated Hessian matrix, \hat{H}_k , is more and more accurate as the number of internal iterations increases, the algorithm does not retain all the information known about the function but focuses on the information in the vicinity of the current shape.

D. Performance of optimization algorithms

The properties of optimization algorithms are described by two opposite notions, exploration and exploitation. The exploration denotes the ability of the optimizer to avoid being trapped by the nearest optimum, whereas the exploitation denotes how quickly the optimizer find the nearest optimal point. Gradient based algorithms are well known to be very efficient in exploitation and poor in exploration.

The figure of merit measuring the restitution time needed for an optimization to converge is the number of process iterations, n_{iter} . The figure of merit measuring the computational cost of an optimization is the total number of function and gradient evaluations, $n_{eval} + n_{grad}$. These quantities both qualify the exploitation properties of the optimizer. One has to notice that for more precision, a distinction is made between iteration at the process level and iteration at the optimizer level. The number of internal iterations of the optimizer does not directly intervene to assess performance and is not reported (for DOT it corresponds to the number of gradient calculations).

The values denoted by the superscript *ref* are reference values obtained by comparing the value(s) of the objective function at a given process iteration to all previous evaluations, so as the quantity $F_c - F_c^{ref}$ is positive or null. As gradient algorithms converge by following a descent path, the final *ref* point is effectively the last computation, but that is not true for all optimizers.

The exploration can be measured in terms of variation of the function value during the optimization, $\sum(F_c - F_c^{ref})$, and in terms of variation of the design variables during the optimization, $\sum \|x_c - x_c^{ref}\|$. In fact, as the variables are defined on very different ranges and are not scaled to measure the exploration, the measure of variation of the objective function is more representative.

III. Response Surface based optimizer (RS Kriging and RS Cokriging)

Surrogate modeling tools are now widely available within engineering. Basically, these tools enable to inexpensively approximate a function on a continuous domain \mathcal{D} given a database of n_s samples, \mathcal{S} . Once built, the surrogate model can give numerous information on the true function through: graphical plots giving trends, sensitivity of the function with respect to each variable, or the value and location of the minimum when coupled with an optimizer.

Although it is possible to build a globally reliable response surface when considering three or four design variables, a phenomenon described by Bellman¹⁸ as the 'curse of dimensionality' prevents the use of global response surfaces on high dimensional spaces. In fact, this phenomenon also discards the use of global optimization algorithms on such problems. A previous study¹⁹ varying dimension from one to six showed that the complexity of aerodynamic functions cannot be represented with as few as 200 function evaluations and recommends the use of local response surfaces. Local response surface based optimizers depart from a space filling sampling giving rough trends of the function and internally use an expensive global optimizer to optimize one or multiple sampling refinement criteria, $C_r(x)$, in order to iteratively increase the surface accuracy around locations of possible optima (cf. Figure 3).

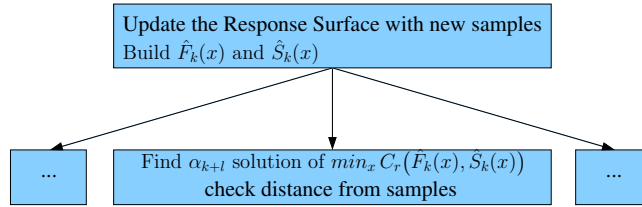


Figure 3. Inside a general response surface based optimizer box

This type of optimizer supposes that an efficient surrogate model can be built in high-dimensional search spaces (one hundred variables).

A. Building response surfaces with Kriging and Cokriging

Kriging²⁰ (DACE formulation) was chosen to build the response surface $\hat{F}(x)$ for its ability to approximate accurately non-linear functions and to guarantee a null error at samples.

1. Kriging

The Kriging method is from the statistical point of view the best linear interpolator. It is formed by a constant term, $\hat{\beta}$, representing a mean of the function at samples, F_s , plus a linear combination of basis function interpolating each sample built up as a stochastic process, $Z(x)$,

$$\hat{F}(x) = \hat{\beta} + Z(x) \approx F(x). \tag{2}$$

The basic assumption behind Kriging is that the covariance of the function is linked to the spatial correlation and this correlation is maximum when distance between points is null and decreases with distance,

$$cov[Z(s^i), Z(s^j)] = \hat{\sigma}^2 R_{ij}. \tag{3}$$

The correlation matrix R is dense symmetric positive definite (order $N = n_s$) with ones along the diagonal and results from evaluations of a Spatial Correlation Function,

$$R_{ij} = SCF(s^i, s^j) = \prod_v scf_v(|s_v^i - s_v^j|). \tag{4}$$

The basis function are the directional function (scf_v) linking the distance to the correlation between two points. A cubic spline function depending of one hyperparameter driving the

directional strength of the correlation, θ_v , was chosen,

$$scf_v(x) = \begin{cases} 1 - 6(x\theta_v)^2 + 6(x\theta_v)^3, & x < \frac{1}{2\theta_v} \\ 2(1 - x\theta_v)^3, & \frac{1}{2\theta_v} \leq x < \frac{1}{\theta_v} \\ 0, & x \geq \frac{1}{\theta_v} \end{cases} \quad (5)$$

In comparison to the traditional gaussian function, the cubic spline gives generally a better conditioned correlation matrix. In fact, the gaussian function decreases exponentially with the distance but tends toward zero without reaching it, whereas the spline function is null after a threshold value fixed by the hyperparameter.

Contrary to the radial basis function method using a different function radius for each basis function, the Kriging method defines this radius by direction. Thus, an appropriate comparative study (domain scaling) of each θ_v values enables to assess the influence of each variable on the function.

Spatial correlation functions must respect at least two properties, be equal to one for a null distance and decrease with distance. These properties lead Kriging models to be unstable when sample points are clustered. If the distance between two samples is too small, their corresponding columns in the correlation matrix will be almost the same implying an ill-conditioned matrix. Space filling sampling methods like Latin Hypercube Sampling (LHS) are thus recommended when using Kriging.

To evaluate Kriging at an unknown location a vector of correlation r between sample points and the unknown is computed,

$$r_i(x) = SCF(x, s^i). \quad (6)$$

Once the correlation matrix inverted, the Kriging, \hat{F} , can finally be evaluated at a new point x ,

$$\hat{F}(x) = \hat{\beta} + r^t(x)R^{-1}(F_s - 1\hat{\beta}). \quad (7)$$

One should notice, that only two samples are needed to build a Kriging response surface, whereas a linear polynomial interpolator requires at least $n_{dv} + 1$ samples to determine polynomial coefficients. In addition, the predicted uncertainty of the Kriging function or standard error, $\hat{S}(x)$, can be computed,

$$\hat{S}(x) = \hat{\sigma} \left(1 - r^t(x)R^{-1}r(x) + \frac{(1 - 1R^{-1}r(x))^2}{1^tR^{-1}1} \right)^{\frac{1}{2}} \geq 0. \quad (8)$$

2. Kriging fit

As Kriging models directly interpolate the samples, they are not fitted by minimizing the least square residual error at samples as polynomial regression models. The standard error of Kriging is null at samples and increases with distance. Kriging are fitted by minimizing their predicted errors, more precisely by maximizing the logarithm of their likelihood estimates. This optimization problem is known as the Maximum Likelihood Estimate (MLE) problem.

For the parameters β and σ^2 analytical expressions maximizing this value are known,

$$\hat{\beta} = (1^tR^{-1}1)^{-1}1^tR^{-1}F_s, \quad (9)$$

$$\hat{\sigma}^2 = \frac{1}{n_s}(F_s - 1\hat{\beta})^tR^{-1}(F_s - 1\hat{\beta}). \quad (10)$$

The correlation parameters (θ_v) are solution of the following MLE optimization problem,

$$MLE = \max_{\theta}(\ln(L)). \quad (11)$$

It is solved using a gradient based optimization algorithm initialized by an appropriate guess as described by Laurenceau et al.¹⁹ Each likelihood evaluation requires the computation of the determinant of the correlation matrix,

$$\ln(L(\theta)) = -\frac{1}{2} (n_s (\ln(\hat{\sigma}^2) + \ln(2\pi) + 1) + \ln |R|), \quad (12)$$

and the resolution of the optimization problems requires up to one thousand likelihood evaluations.

During the iterative refinement process, the hyperparameters are refitted at each update of the sample database.

3. *Sample limited indirect Cokriging, gradient enhanced Kriging*

A Cokriging model²¹ interpolates the function and the gradient at each sample location. As Cokriging models include more information on the true function than Kriging models, they need fewer samples to achieve a given level of accuracy. Moreover a comparison varying the dimension of the problem and the number of samples on an aerodynamic test case¹⁹ has proven that the vectorial information provided by the gradient is more beneficial for high-dimensional problems.

For its flexibility, the formulation retained to build gradient enhanced Kriging is the indirect Cokriging formulation.²¹ It does not change the Kriging formulation described previously because the gradient interpolation is performed through a sample database augmentation scheme. Instead of directly using the gradient information, it is used to add one point per direction at each sample using a first order Taylor approximation,

$$F(s^{n_s+iv}) = F(s^i) + \frac{\partial F(s^i)}{\partial x_v} 10^{-4} range_v(\mathcal{S}). \quad (13)$$

After using the gradient information at each sample, the augmented database contains $N = n_s(n_{dv} + 1)$ samples.

The computational cost of Kriging model fit or evaluation depends mainly on the order of the correlation matrix, N . On a standard Intel Pentium4 2.8GHz processor, the estimated computation time for one correlation matrix inversion is,

$$t_{user} \approx 3.10^{-9} N^3 \text{ seconds}, \quad (14)$$

and a total of several hundreds matrix inversions are done for each response surface.

The computational cost of the response surface (fit and evaluation) is considered negligible compared to CFD function evaluations for matrix order up to $N \approx 400$. However, as the computational cost of Cokriging depends on the number of variables, it becomes unusable for high-dimensional problems. It is then necessary to use altered augmentation schemes. Liu²¹ proposed a direction limited augmentation scheme using only one augmentation point per sample, $N = 2n_s$. Practically, this strategy implies a loss of information. As gradients of CFD functions are computed with an adjoint method, the complete gradient vector is computed even if only one term of the vector is needed.

This is why a sample limited augmentation scheme is preferred, as suggested by Kim et al.²² on their aerodynamic problem depending on six variables. It consists in using the complete gradient information but only for a fixed number of samples denoted n_{aug} . The augmented matrix order is then $N = n_s + n_{aug}n_{dv}$. Basically, using $n_{aug} = 10$ enables to abide by the rule of thumb recommending the use of at least 10 samples per direction to have a correct response surface. Practically, it is used to adapt the Cokriging computational cost to the number of variables.

B. Global optimization on the response surface

As stated in Figure 3, another optimization problem has to be solved inside the response surface based optimizer box. Some sampling refinement criteria, $C_r(x)$, must be optimized to exploit the information contained in the response surface and find a suitable new set of design variables. As the criteria are inexpensively evaluated, a global optimization strategy was adopted for their minimization. In order to effectively find the global minimum even for high dimensional problems, a strategy using multiple optimizers and multiple runs is used. This strategy proceeds in building a set, \mathcal{P} , of possible optima of the criterion given by various methods and finally retains only the minimum of this set.

Firstly, the set of possible optima, \mathcal{P} , is initialized by the minimum from the sample database.

Secondly, a binary coded genetic algorithm from David Carroll ^a is used with a population size of 100 individuals and a maximum number of generations equal to 1000. A convergence test was implemented to stop this algorithm when no improvement on the function is made during 10 consecutive iterations. This algorithm is ran 100 times using different seed values. Each run enables to increment a new point in the set of possible optima \mathcal{P} .

Thirdly, a pseudo-random exploration is performed on $100 \cdot 10^3$ sites. The minimum of the exploration phase is used to initialize a gradient descent (quasi-newton BFGS method). Once converged the minimum of the gradient algorithm is incremented in the set \mathcal{P} . This step is repeated 10 times with different seed values for the exploration.

Finally, a last gradient descent is performed departing from the minimum point of the set of possible optima \mathcal{P} .

The global optimization on the response surface is necessary to assess the efficiency of the response surface based optimizer. In fact less caution would have been possible during this internal optimization, but as this step should be as efficient on the 6 design variables test case as on the 45 design variables test case, robustness was privileged in detriment of convergence speed. As no bias can be attributed to this internal optimization, any loss of performance of the response surface based optimizer when increasing dimension can be attributed to a loss of accuracy of the response the surface.

C. Sampling refinement criteria

A variety of sampling refinement criteria exists. Some are designed to reduce the global error of the response surface,¹⁹ and some are designed to minimize the true function modelised by the response surface. Sasena²³ compared a variety of criteria.

1. Definition

The most basic criteria exploits only information on the function,

$$C_1(x) = \hat{F}(x). \quad (15)$$

As stated in the Surrogate Management Framework,^{24,25} optimization algorithms that refine iteratively only at the predicted minimum ('SEARCH' step) cannot efficiently find an optimum. A spatial exploration is necessary ('POLL' step) to ensure convergence. Using the standard error (Eqn. 8), it is possible to compute criteria able to locate minimum of the function taking into account uncertainty of the model.

Each refinement criterion proposes its own manner to balance exploitation of low function values with exploration of high uncertainty values. The Expected Improvement (EI)⁶ is the most common when using Kriging. It is a statistical criterion maximizing the probability of improvement over the best sample value. The maximum of the EI function indicates the predicted location of the global minimum of the true function. It has proven to be an efficient

^aCU Aerospace, <http://www.cuaerospace.com/carroll/ga.html>

refinement criterion on low dimensional problems.^{6,9,26} Despite that, the EI criterion was not retained here as it tends to cluster points around predicted minimum before exploring other locations. Moreover, it has quasi-null gradients on large parts of the domain making it impossible to use gradient algorithms to find its maximum (Figure 4). This criterion is then hard to maximize.

The Lower Confidence Bounding (LCB) criterion of Cox et al.²⁷ was preferred as it tends to give more weight to the uncertainty. The LCB directly uses a balancing coefficient b between function and error to perform a linear combination,

$$LCB_b(x) = \hat{F}(x) - b\hat{S}(x), \quad b \in \mathcal{R}^*. \quad (16)$$

The minimum of LCB indicates the expected location of the minimum of the true function. The standard error of Kriging being always positive, the error balancing coefficient, b , can be positive or negative. The values tested by Cox et al. are $b = 2.0$ and $b = 2.5$, using a different optimization process (single refinement criterion) on low-dimensional analytical functions. In this work, the values chosen are $b = 1.0$ and $b = -1.0$,

$$C_2(x) = LCB_1(x) = \hat{F}(x) - \hat{S}(x), \quad (17)$$

$$C_3(x) = LCB_{-1}(x) = \hat{F}(x) + \hat{S}(x). \quad (18)$$

On Figure 4, the minimum of the surrogate model F indicates an already sampled location, different from the optimum on the true function. Using the standard error (error bars), the minimum of the LCB_1 function will explore a zone of higher uncertainty whereas the EI criterion indicates a location very close to the minimum of the Kriging function corresponding to an already sampled location.

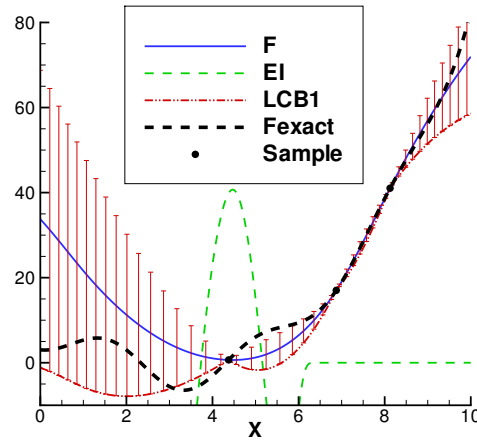


Figure 4. Expected Improvement and Lower Confidence Bounding Function on a one dimensional example approximated using three samples

2. Distance check

Each location indicated by the minimization of a sampling refinement criteria must be validated by performing a distance computation with respect to already computed locations. The main purpose of this validation is to avoid expensive CFD computations on shapes already well known. The secondary purpose is to ensure a good conditioning of the Kriging correlation matrix. The point is rejected if its minimal distance with respect to already computed samples fall below a fixed threshold,

$$x^{ref} \text{ accepted if, } \min_i \left(\sum_v \frac{|x_v^{ref} - s_v^i|}{range_v} \right) > n_{dv} \cdot \epsilon. \quad (19)$$

The value of the threshold, ϵ , should be adapted to the level of convergence expected from the optimizer i.e. the number of iterations authorized. The value chosen here is $\epsilon = 10^{-6}$.

D. Kriging/Cokriging based response surface optimizer

The new Kriging/Cokriging based optimizers (Figure 5) are initialized using a space filling sampling method for the first iteration. The refinement process begins at the second iterations after n_s^{ini} function evaluations. It uses for the exploitation of the best function value a refinement at the predicted minimum of the function. In order to ensure exploration of the domain and try to converge to the global optimum, the LCB function is used with two different error balancing coefficients. This framework runs simultaneously up to three CFD runs ($n_{pop} \leq 3$) per iteration. As the sampling refinement includes exploratory steps, this optimizer is able to manage very coarse initial sample database (10 samples for 45 variables) at the cost of more iteration of the process. The use of large initial sampling database can also save time as the n_s^{ini} evaluations are done simultaneously.

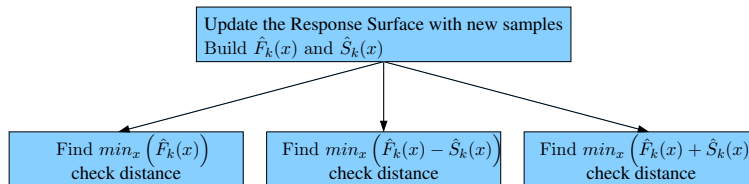


Figure 5. Inside the Kriging/Cokriging based optimizer box

Convergence is said to be reached when all three proposed refinements are close too already sampled locations or if the function value cannot be improved during 20 iterations.

IV. Results

In the following section, a comparison between the two new Kriging and Cokriging based optimizers and the reference gradient optimizer is made on two test cases. This comparison was not made on analytical test functions common in the optimization community because it cannot be representative of real CFD functions: multimodal (lots of local optima), high number of more or less independent variables and noise. The no-free-lunch theorem for optimization by Wolpert et al.²⁸ states that the ultimate optimization algorithm does not exist. If an optimizer is the most competitive for a given set of problems, it is always possible to find another set of problems to contradict this fact. Wolpert et al. recommendations are then to use a wide variety of test problems in order to compare general-purpose optimization algorithms, and to incorporate problem-specific knowledge into the optimizer as much as possible. Our test cases are then directly performed within OPTaliA. Moreover, the problems are defined as representative simplifications (no constraints, less design variables, coarse meshes) of real industrial problems handled by Airbus designers.

A. Airfoil drag minimization considering 6 design variables

The first test problem is about a RAE2822 airfoil at a Mach number $M = 0.729$, an angle of attack 2.31° . The chord length is one meter and the Reynolds number value is $Re = 7.10^6$. The C-mesh is formed of 73×458 nodes with its boundary layer refinement. The restitution time for one flow simulation is 400 seconds using two processors AMD Opteron 275 (2.2 Ghz) to perform 400 steady iterations. The objective function considered is the far field pressure drag. Two bumps ($n_{dv} = 6$) are applied to deform the upper surface in the direction of the vertical axis. Only positive deformations are authorized and the maximum amplitude of one bump is 5 millimeters. The domain of variation for the position of the bumps authorizes recovering.

The complexity of the optimization problem is low as only one function and six design variables are considered. Moreover, the ranges on variables are restrained to positive deformations in order to limit the complexity of the objective function on the domain and thus the number of possible optima and to limit the impact on lift.

The same initial database is used for Kriging and Cokriging response surfaces. It contains 10 samples, $n_s^{ini} = 10$, including the baseline shape and nine space filling samples distributed by a Sobol method. The sample database augmentation for the sample-limited Cokriging includes the information of 10 gradient vectors, $n_{aug} = 10$, providing a quantity of information equivalent to 60 additional samples.

The starting point for the gradient algorithm corresponds to large equidistributed bumps of null amplitudes.

On Figure 6 are represented the baseline and the optimized shapes and associated pressure coefficient values. The pressure distribution of the baseline shape demonstrates the presence of a shock wave (drag production) at the upper surface of the airfoil (between 50% and 60% of the chord length). This shock wave is attenuated (suppressed) by the deformation proposed by the different optimizers. It appears that the reference optimizer (DOT-BFGS) converged to a different optimal deformation. Whereas both response surface optimizers find an optimal shape by distributing the two bumps at two different locations (leading edge and location of the shock), the gradient optimizer has been trapped by the high sensitivity of the drag to deformation near the shock wave and has tried to superpose both bumps. This illustrates the fact that even considering simple shape variables (only two bumps), functions studied in aerodynamic shape optimization exhibit multimodal behaviour and strong trends (high gradient values) especially when considering transonic flows. The low complexity of this problem does not prevent the gradient to be trapped by a local optima.

In terms of final function value, both response surface optimizers largely outperform the gradient optimizer. It appears that even the purely exploratory space filling samples ($n_{eval} \leq 10$) managed to improve the baseline shape and to outperform the gradient. Both response surfaces algorithms give very close results in terms of shape deformation and final objective function value confirming their enhanced global exploration capabilities. One can see from Table 1 that contrary to the RS optimizers, DOT-BFGS does not manage to completely suppress the shock and slightly increases the viscous pressure drag by its larger deformation near the trailing edge.

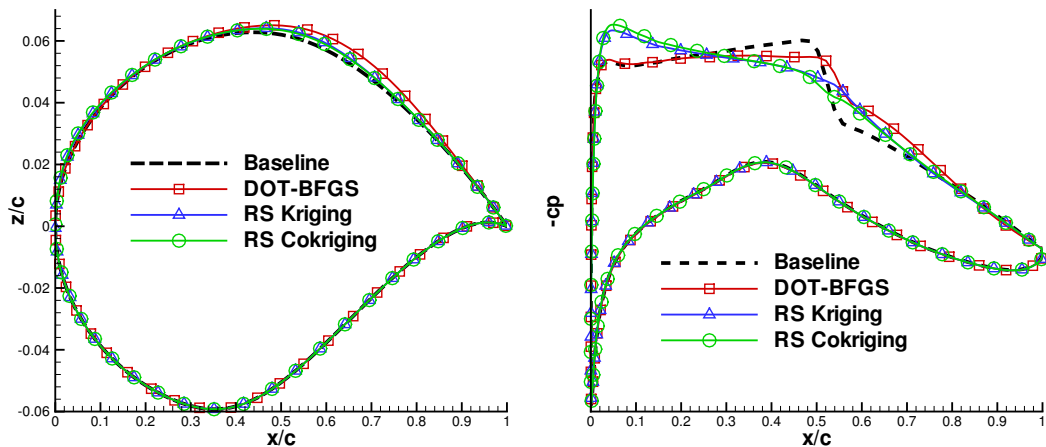


Figure 6. Optimized shape and pressure distribution of an RAE2822 airfoil

Before totally discarding the reference optimizer, it is necessary to look at the computational cost (total number of evaluations $n_{eval} + n_{grad}$) and restitution time (number of process iterations n_{iter}) of the different optimizations represented on the Figure 7 and summarized in Table 2. The worst algorithm in terms of final function value, DOT-BFGS, is also the quickest and less

	Baseline	DOT-BFGS	RS Kriging	RS Cokriging
$Cd_p = F$	100	84.6 (-15%)	76.7 (-23%)	76.9 (-23%)
Cd_w	19.3	3.3	0.0	0.6
Cd_{vp}	80.7	81.3	76.7	76.3
Cd_f	119.7	121.1	120.7	120.5
Cl	100	103.2	102.1	102.2

Table 1. Aerodynamic performance summary for RAE2822 optimizations

expensive. It seems coherent with the fact that this method performs very few exploration of the domain in terms of variation in function value and variables value. The additional cost of RS optimizers is attenuated by the parallelization of the optimization process and stays acceptable in an industrial context. It is important to note that the RS optimizers reached a plateau of convergence in as few iterations as the gradient optimizer, but the RS optimizers managed to continue domain exploration from this reference value. The additional gradient information used by the Cokriging RS seems not absolutely necessary, certainly due to the low dimensionality of the problem. It effectively produces a more accurate surrogate model as can be deduced from the lower values of exploration compared to the Kriging RS. When considering the the last process iteration of DOT-BFGS ($n_{iter} = 41$ on Figure 7), the function value given by the gradient optimizer is a lot larger than function values given by both RS based optimizers. Moreover both RS optimizers have reached a plateau of convergence. One could then say that in terms of resitition time the RS based optimizers are better than the gradient reference.

On figures Figure 7 and Figure 10, the lines stand for *ref* values of the objective function and the symbols represent the current values of the function.

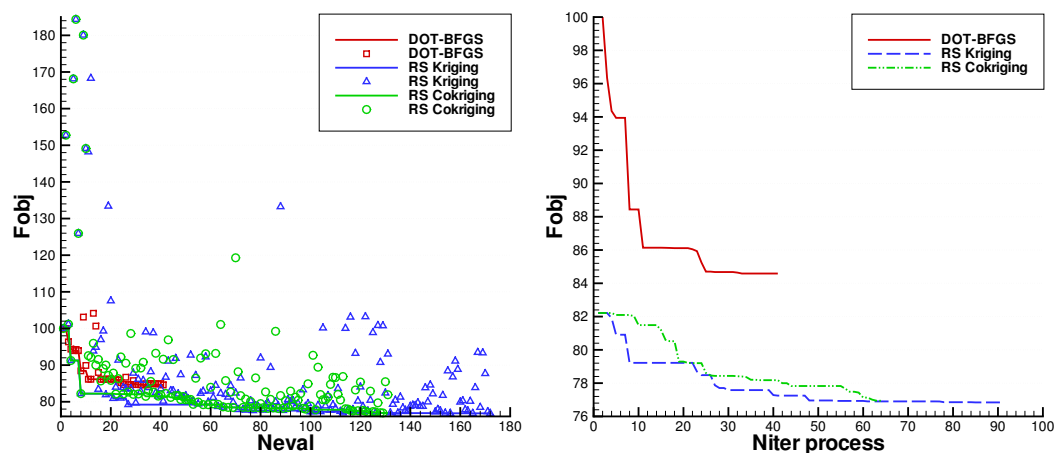


Figure 7. Optimization convergence history on the RAE2822 test case

The experience of numerous airfoil optimization with OPTaliA demonstrated that it is impossible to increase indefinitely the number of design variables on bidimensional airfoils without having highly correlated design variables. In fact, by using a decomposition of optimal shape deformation into a sum of Hicks-Henne bumps, one can see that the optimum of airfoil drag minimization problems are generally obtained through a sum of no more than five bumps. Increasing the number of bumps on a bidimensional wing section is generally equivalent to a parameterization using less bumps with higher ranges in amplitude of bumps. By considering only two bumps, one can guarantee that there is no redundancy between design variables and

	DOT-BFGS	RS Kriging	RS Cokriging
F^{ref} final value	84.6 (-15%)	76.7 (-23%)	76.9 (-23%)
$n_{iter} ; n_{eval} + n_{grad} ; n_s^{ini}$	41; 41+6; 0	91; 172+0; 10	64; 131+10; 10
$\sum F_c - F_c^{ref} $	56.2	1503	953
$\sum F_c - F_c^{ref} /n_{iter}$	1.37	16.5	14.9
$\sum F_c - F_c^{ref} /n_{eval}$	1.37	8.74	7.27
$\sum \ x_c - x_c^{ref}\ /n_{dv}$	0.565	65.4	41.0
$\sum \ x_c - x_c^{ref}\ /n_{iter}/n_{dv}$	$1.38 \cdot 10^{-2}$	$71.9 \cdot 10^{-2}$	$64.1 \cdot 10^{-2}$
$\sum \ x_c - x_c^{ref}\ /n_{eval}/n_{dv}$	$1.38 \cdot 10^{-2}$	$38.0 \cdot 10^{-2}$	$31.3 \cdot 10^{-2}$

Table 2. Optimizers performance summary on the RAE2822 test case

that the complexity of the optimization problem is effectively represented by the number of design variables. In Figure 8, a simple shape deformation obtained with 2 Hicks-Henne bumps of maximal amplitude equal to 3.0 millimeters is reconstructed using a very complex combination of 20 Hicks-Henne recovering bumps of maximum amplitude equal to 0.5 millimeters.

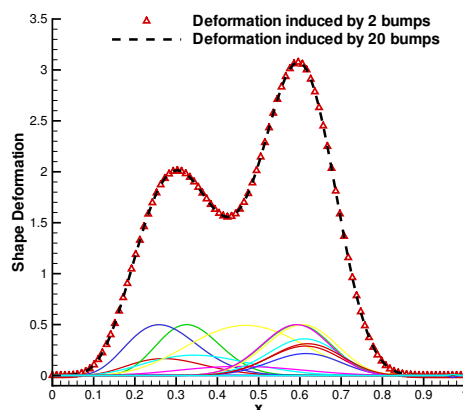


Figure 8. Equivalence between 2 Hicks-Henne bumps and 20 Hicks-Henne bumps of lower amplitude

B. Wing drag minimization considering 45 design variables

In order to increase the number of design variables with as less redundancy between variables as possible a three dimensional shape is considered, the AS28 wing in cruise condition, at Mach number $M = 0.8$ and angle of attack $AoA = 2.2^\circ$. The wing span is $b = 25$ meters and the mean chord value is $c = 5.4$ meters, giving a Reynolds number $Re = 40 \cdot 10^6$. The structured mesh (Figure 9) is formed of 4 blocks containing a total of $500 \cdot 10^3$ nodes with its boundary layer refinement. The restitution time for one flow simulation is 4400 seconds using two processors AMD Opteron 275 (2.2 Ghz) to perform 500 steady iterations.

The 45 shape parameters correspond to 15 Hicks-Henne bumps on the upper surface. The vertical bumps are distributed by group of three in five spanwise sections. At each section, the three bumps are equidistributed and a recovering is authorized by groups of two bumps. In order to ensure a wide diversity of possible shapes (maximum degree of freedom) with this parameterization, the linear spanwise expansion of each sectional deformation is stopped either at the closest boundary either at the closest deformed section. Only positive deformations are authorized and the maximum amplitude of one bump is 50 millimeters. The complexity of

this problem is thus increased by considering more independent design variables and also by increasing the relative range of possible deformation, $\frac{A^{max}}{c}$ from five to ten as represented by optimal shapes on Figures 14-17.

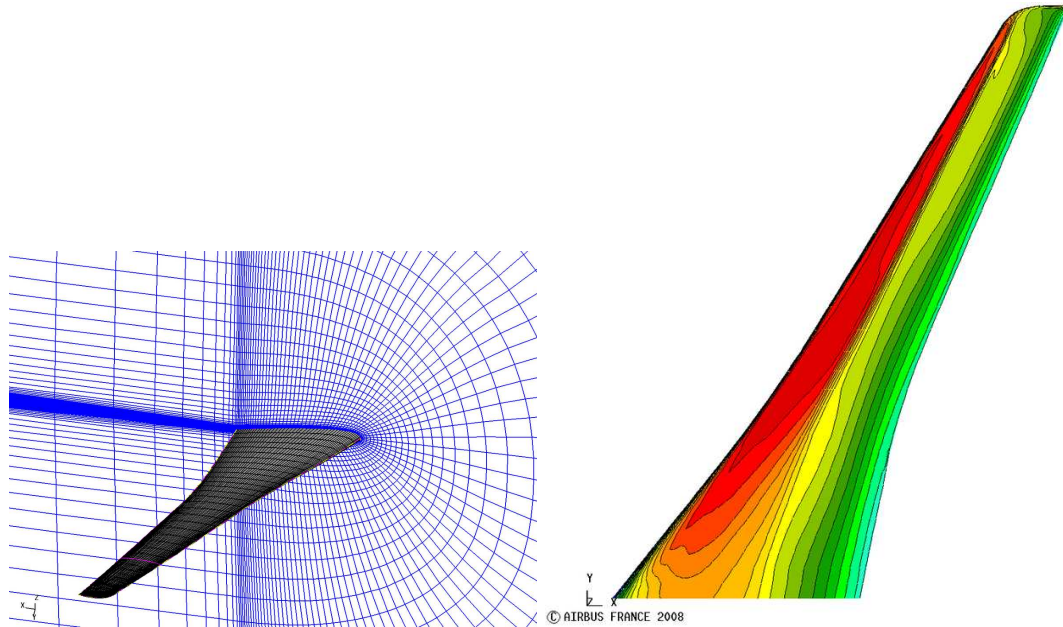


Figure 9. Mesh and pressure coefficient of the AS28 wing baseline shape

The RS based optimizers were initialized with only 11 samples, $n_s^{ini} = 11$, and the number of samples is even inferior to the number of variables. The initial database was formed of the baseline configuration plus 10 space filling samples (Latin Hypercube Sampling method). The Cokriging sampling was augmented by including the information of 5 gradient vectors, $n_{aug} = 5$, giving 225 additional scalar values. The starting point for the gradient algorithm corresponds to large equidistributed bumps of null amplitudes.

In cruise condition a shock wave appears on the upper surface of the baseline configuration. The Table 3 shows that most of optimizers drag reduction comes from minimization of the wave drag and leads to shapes with very weak shocks as can be verified on pressure distribution (cf. Figures 11-13).

	Baseline	DOT-BFGS	RS Kriging	RS Cokriging
$Cd_p = F$	100	95.1 (-4.9%)	94.6 (-5.4%)	93.5 (-6.5%)
Cd_w	9.0	2.6	2.6	1.0
Cd_{vp}	35.2	34.9	34.5	34.6
Cd_i	55.8	57.6	57.5	57.9
Cd_f	29.9	30.6	30.5	30.5
Cl	100	102.9	102.3	103.2

Table 3. Aerodynamic performance summary for AS28 wing optimizations

Figure 10 presents the convergence history of the optimizations. The gradient based algorithm keeps the same properties as with the previous test problem. It converges using few process iterations and very few objective function evaluations (even fewer than on the six dimensional problem). Despite its speed, it gives the lowest improvement of the objective function.

For the RS based optimizations, the influence of the very coarse initial sampling can directly be observed by comparing exploration values of Kriging and Cokriging optimizers (Table 4). As the Kriging response surface is less accurate, the Kriging based optimizer performs three times more exploration and needs more iterations to converge to its final value. The sampling refinement becomes almost similar to a space filling refinement, but a significant improvement of the function is achieved and the Kriging optimizer outperforms the gradient reference. Somehow, its convergence is stopped because no improvement is achieved during the last 20 iterations whereas the algorithm continues its domain exploration characterized by the large widespreading of symbols on Figure 10. The Kriging based optimizer seems to need more iterations to robustly converge. The Cokriging based algorithms does not suffer this problem, as very few improvement is achieved during the last 50 iterations despite the continuous exploration of the domain. By looking at the tridimensional views of optimal deformation fields given by Kriging (Figure 12) and Cokriging (Figure 13) based optimizers, it appears that both fields are very similar but the Kriging shapes corresponds in fact to an intermediate shape explored during the Cokriging based optimization.

The sample limited Cokriging based reached the best final objective function value at the cost of more function evaluations compared to DOT-BFGS. Despite the additionnal computational cost, the Cokriging based optimizer outperforms the gradient reference in terms of restitution time. When considering the last process iteration of DOT-BFGS ($n_{iter} = 26$ on Figure 10), the function value given by the gradient optimizer is larger than the function value given by the Cokriging based optimizer. However, the RS Cokriging optimizer has not yet reached its plateau at this iteration and needs three times more iterations to converge.

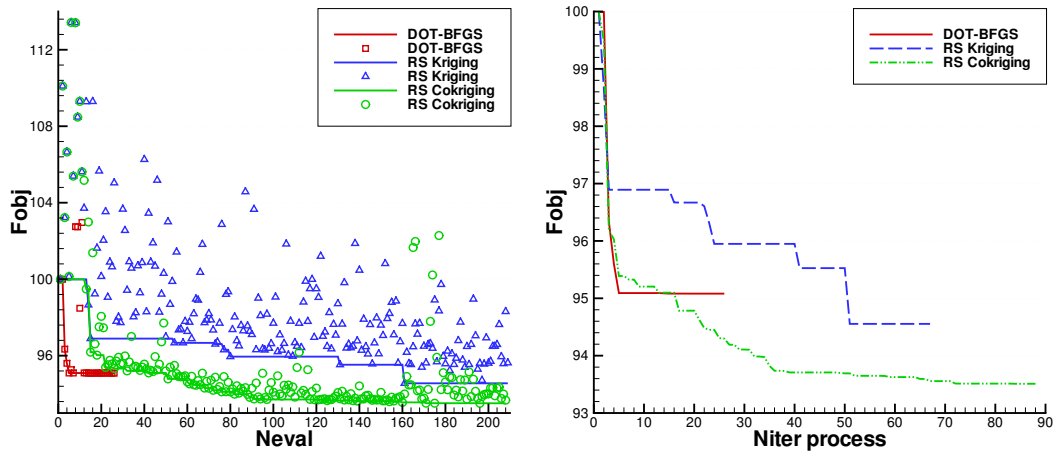


Figure 10. Optimization convergence history

On Figure 18 are plotted approximated drag polar for the baseline configuration and the three optimized shapes. The approximation is done using a second order polynomial method fitted with three samples for each shape represented by symbols on the figure. This graph shows that even after changing the angle of attack of the new shapes to obtain the same lift coefficient as with the baseline shape, the aerodynamic performance keep the same classification, gradient and Kriging based optimization give equivalent improvements and the Cokriging based optimal shape is best. Despite the improvement at cruise conditions, it is not surprising to observe that the performances at low lift flight conditions are decreased by all optimizations as it was not taken into account in the formulation of the optimization problem. As expected, the less the drag is reduced, the less low lift performances are impacted and the shape given by the gradient optimization is better than the shapes given by both RS based optimizations in these conditions.

	DOT-BFGS	RS Kriging	RS Cokriging
F^{ref} final value	95.1 (-4.9%)	94.6 (-5.4%)	93.5 (-6.5%)
n_{iter} ; $n_{eval} + n_{grad}$; n_s^{ini}	26; 26+4+0	67; 209+0; 11	88; 208+5; 11
$\sum F_c - F_c^{ref} $	26.8	534.4	205.5
$\sum F_c - F_c^{ref} /n_{iter}$	1.03	7.98	2.34
$\sum F_c - F_c^{ref} /n_{eval}$	1.03	2.56	0.99
$\sum \ x_c - x_c^{ref}\ /n_{dv}$	4.79	232.1	73.5
$\sum \ x_c - x_c^{ref}\ /n_{iter}/n_{dv}$	0.18	3.46	0.84
$\sum \ x_c - x_c^{ref}\ /n_{eval}/n_{dv}$	0.18	1.11	0.35

Table 4. Optimizers performance summary on the AS28 wing test case

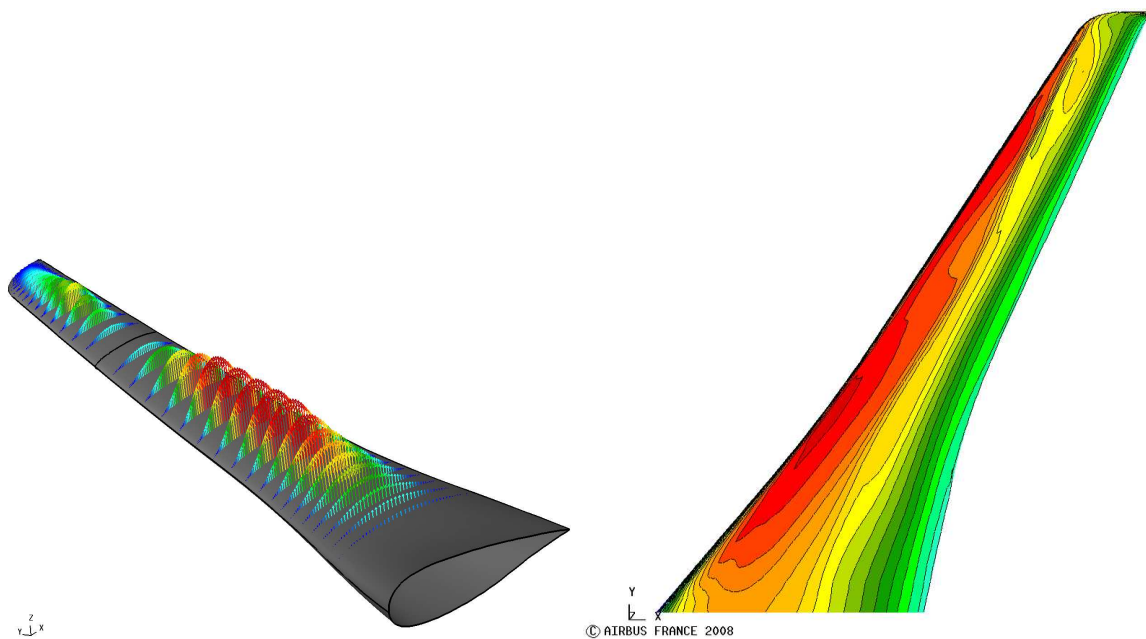


Figure 11. Optimum given by DOT-BFGS for the AS28 wing. Left: vectorial field of deformation. Right: pressure coefficient.

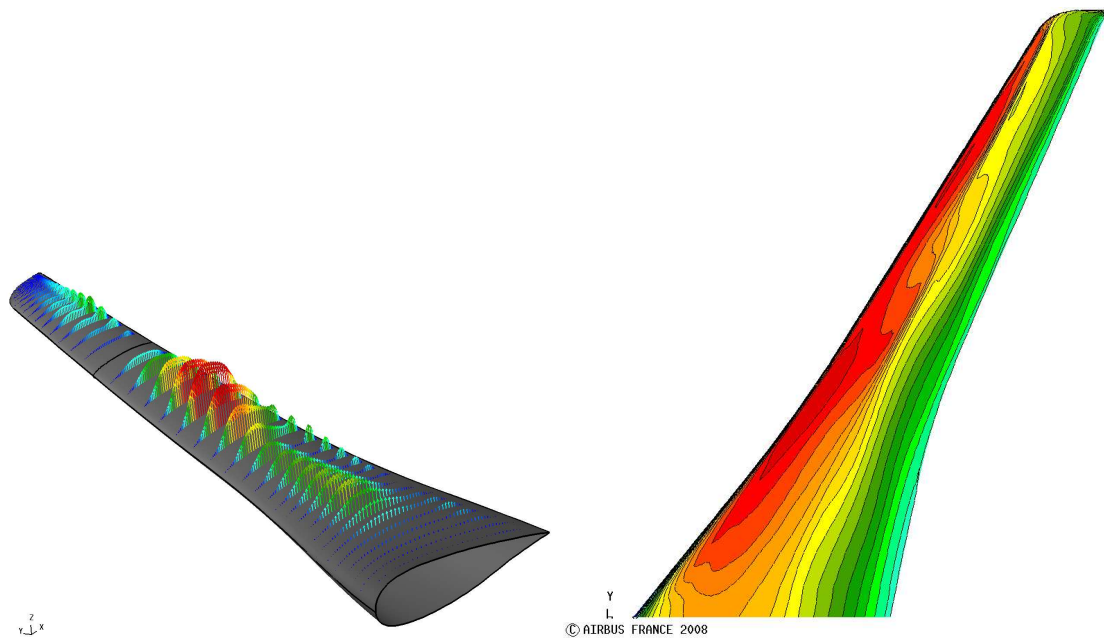


Figure 12. Optimum given by the Kriging based optimizer for the AS28 wing. Left: vectorial field of deformation. Right: pressure coefficient.

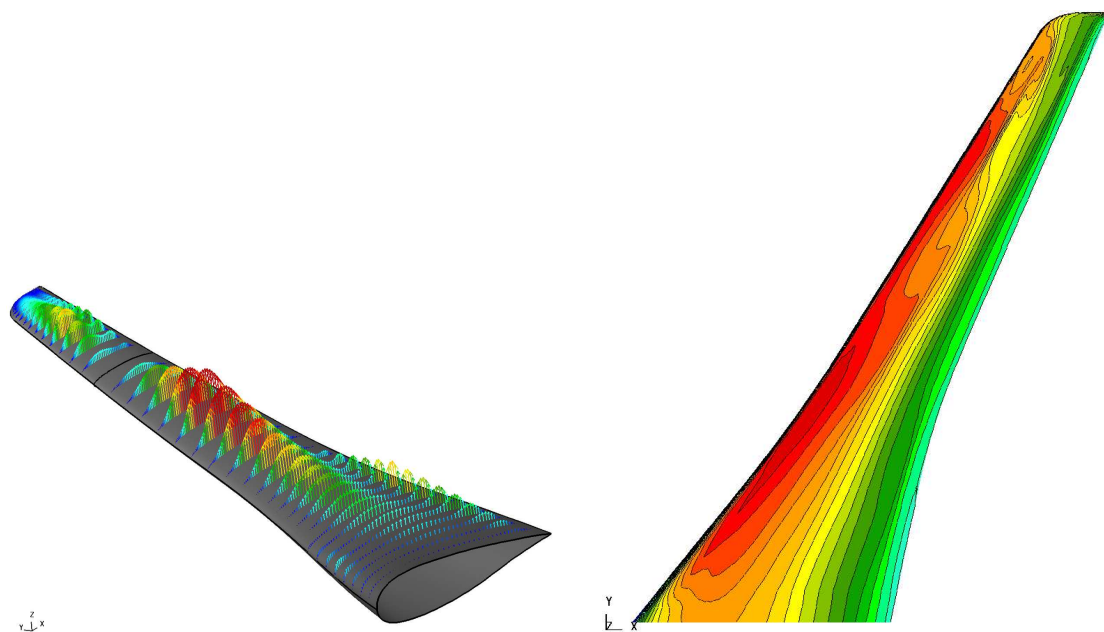


Figure 13. Optimum given by the Cokriging based optimizer for the AS28 wing. Left: vectorial field of deformation. Right: pressure coefficient.

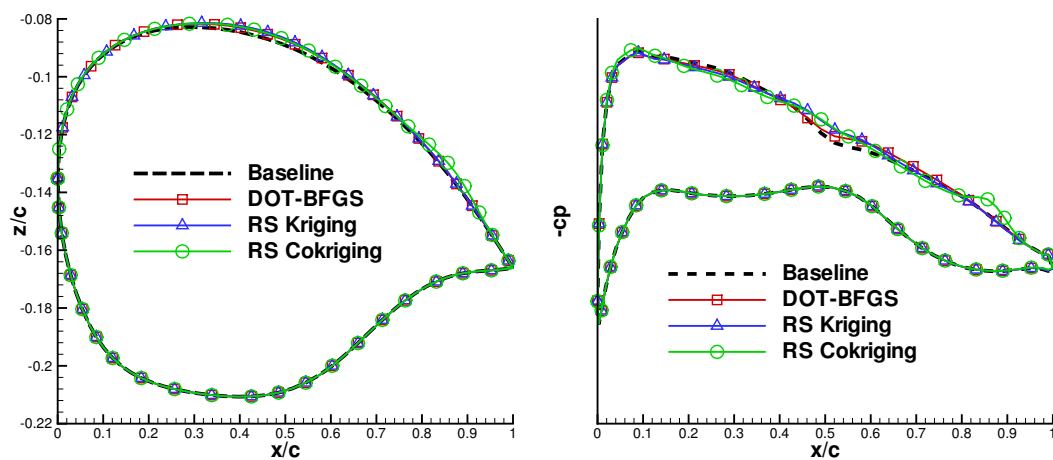


Figure 14. Optimized shape and pressure distribution at 6% of the wing span

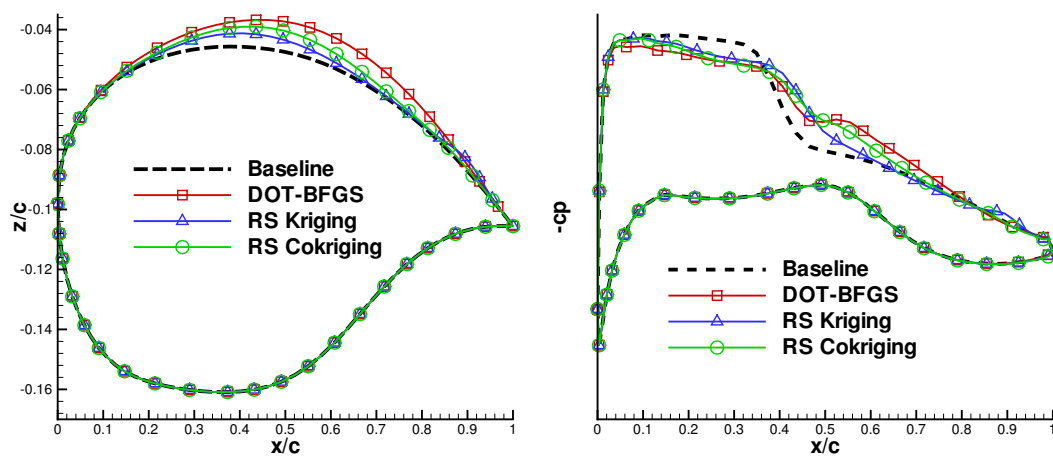


Figure 15. Optimized shape and pressure distribution at 19% of the wing span

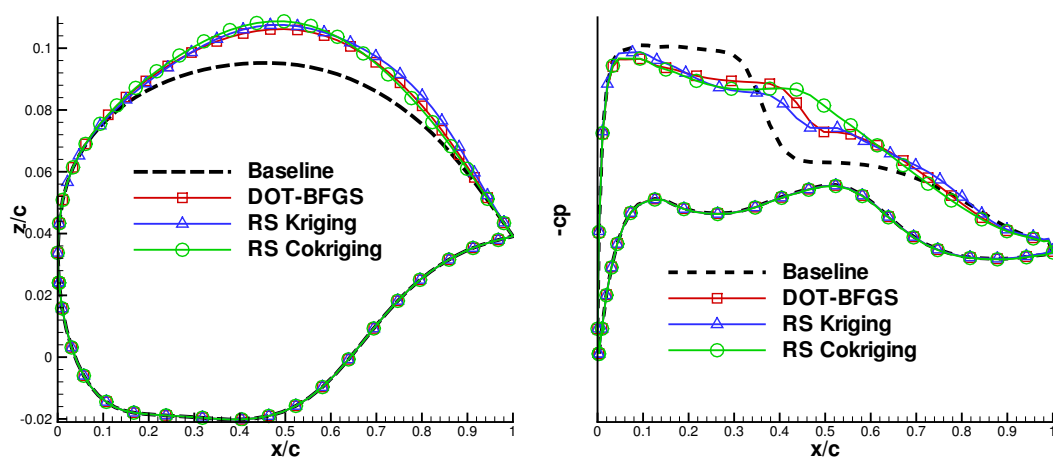


Figure 16. Optimized shape and pressure distribution at 35% of the wing span

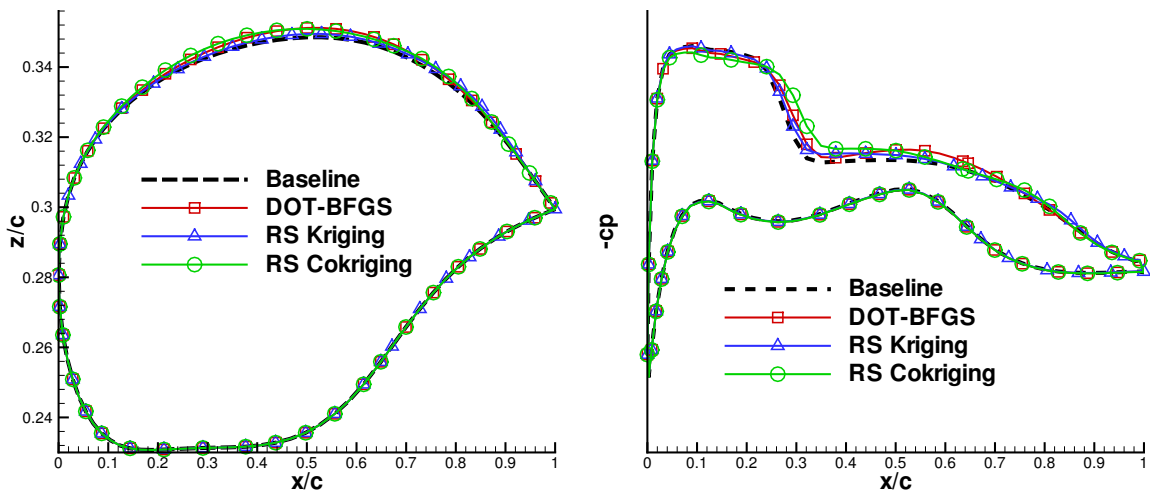


Figure 17. Optimized shape and pressure distribution at 60% of the wing span

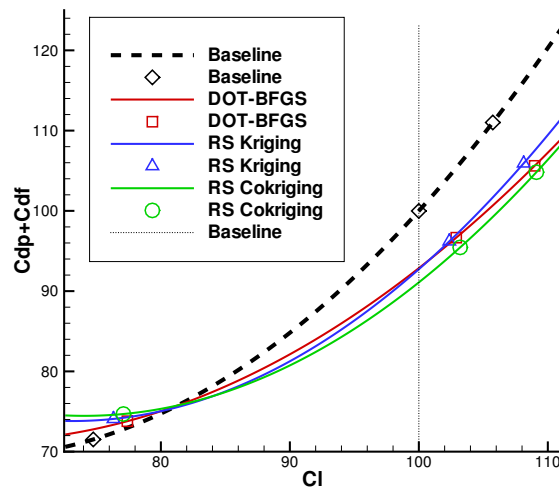


Figure 18. Approximated drag polars for the AS28 wing

V. Conclusion

A general framework for optimization enabling the use of various optimizers was set up. Two algorithms based on response surfaces built with Kriging and Cokriging were then implemented and compared to the quasi-Newton reference algorithm. These optimizers seem very promising as they achieve better function improvement and are complementary to the reference gradient optimizer converging very quickly to the nearest optimum. Best practices can be drawn from the results obtained on the two drag reduction test problems.

For the low dimensional problem, both response surface based optimizers largely outperform the gradient in terms of final function value, but also outperform it at equivalent computational cost and restitution time. Despite the fact that the gradient algorithm converges to a local optimum, it needs fewer functions evaluations to reach its final optimum. The interpolation of gradient information does not significantly improve Cokriging based optimizer performances, that is why in conclusion the gradient free Kriging based optimization algorithm should be preferred for low dimensional problems.

When increasing the complexity of the problem by considering 45 design variables on a wing, the Kriging based optimizer requires twice more iterations after gradient convergence before outmatching it. The lack of accuracy of the Kriging model makes it converge prematurely due to an excessive exploration of the domain. The sample limited Cokriging based optimizer does not suffer this problem and effectively converges to a better solution than the gradient reference. The additional cost needed is small in terms of restitution time due to the efficiency of the parallel framework, but is large in terms of total function evaluations. The cost of the gradient optimizer being totally independent of the number of design variables, it should be preferred for high dimensional problems when computational cost is the major issue. The Cokriging based optimizer can converge to a more global optimum and should be preferred when performance is the major issue.

These conclusions can be explained by considering the quantity of information, N , contained inside each optimizer. At a given iteration, the gradient algorithm only knows a search direction given by comparing two gradient vectors and a few function evaluations, $\forall n_{eval} N_{Gradient} \approx 10 + 2 \cdot n_{dv}$. Response surfaces are a mean to cumulate all known information on the function in a sample database and the sampling refinement process aims at having an accurate model in the vicinity of the global optimum. For the Kriging based algorithm, the quantity of information is independent of the number of variables $\forall n_{eval} N_{RSKriging} = n_{eval}$ explaining its domain of applicability. The Cokriging based algorithm interpolates a fixed number (n_{aug}) of gradient vectors $\forall n_{eval} N_{RSKriging} = n_{eval} + n_{aug} \cdot n_{dv}$ giving an efficient algorithm even for high dimensional problems.

Lastly, it was assessed that the use of global optimizers imply a careful definition of the optimization problem in order to avoid unrealistic shapes. The gradient based optimizer, departing from an already existing shape and converging to a close local optimum, is less prone to this problem.

Acknowledgments

The authors would like to acknowledge M. Montagnac from Cerfacs for its support and suggestions. We used the computer implementation of J. Burkardt (<http://www.scs.fsu.edu/~burkardt/>) to generate the initial space filling sampling and the genetic algorithm of D. Caroll for optimization on response surfaces and would to thank them for sharing their source codes.

References

- ¹Hicks, R. M. and Henne, P., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, 1978, pp. 407–412.

- ²Jameson, A., "Efficient Aerodynamic Shape Optimization," *10th Multidisciplinary Analysis and Optimization Conference*, No. AIAA-04-4369, AIAA/ISSMO, 2004.
- ³Liu, P.-Y. J. and Zingg, D. W., "Comparison of Optimization Algorithms Applied to Aerodynamic Design," *42nd Aerospace Science Meeting & Exhibit*, No. AIAA-04-454, AIAA, 2004.
- ⁴Moigne, A. L. and Qin, N., "Variable-Fidelity Aerodynamic Optimization for Turbulent Flows Using a Discrete Adjoint Formulation," *AIAA Journal*, Vol. 42, No. 7, 2004, pp. 1281–1292.
- ⁵Meaux, M., Cormery, M., and Voizard, G., "Viscous Aerodynamic Shape Optimization Based On The Discrete Adjoint State For 3D Industrial Configurations," *European Congress on Computational Methods in Applied Sciences and Engineering*, Jul. 2004.
- ⁶Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, 1998, pp. 455–492.
- ⁷Torczon, V. and Trosset, M. W., "Using approximations to accelerate engineering design optimization," Tech. Rep. TR-98-33, ICASE, Institute for Computer Applications in Science and Engineering, 1998.
- ⁸Peigin, S. and Epstein, B., "Robust Optimization of 2D Airfoils Driven by Full Navier-Stokes Computations," *Computers & Fluids*, Vol. 33, No. 9, 2003, pp. 1175–1200.
- ⁹Jeong, S., Murayama, M., and Yamamoto, K., "Efficient Optimization Design Method Using Kriging Model," *Journal of Aircraft*, Vol. 42, No. 2, 2005, pp. 413–419.
- ¹⁰Kanazaki, M., Tanaka, K., Jeong, S., and Yamamoto, K., "Multi-Objective Aerodynamic Exploration of Elements' Setting for High-Lift Airfoil Using Kriging Model," *Journal of Aircraft*, Vol. 44, No. 3, 2007, pp. 858–864.
- ¹¹Booker, A. J., Cramer, E. J., Frank, P. D., Gablonsky, J. M., and Dennis, J. E., "MoVars: Multidisciplinary Optimization Via Adaptive Response Surfaces," *48th Structures, Structural Dynamics, and Materials Conference*, AIAA/ASME/ASCE/AHS/ASC, 2007.
- ¹²Mouton, S., Laurenceau, J., and Carrier, G., "Aerodynamic and Structural Optimisation of Powerplant Integration under the Wing of a Transonic Transport Aircraft," *42ème colloque d'Aérodynamique Appliquée*, AAAF, 2007.
- ¹³Cambier, L. and Veillot, J.-P., "Status of the *elsA* CFD Software for Flow Simulation and Multidisciplinary Applications," *46th Aerospace Science Meeting & Exhibit*, No. AIAA-08-664, AIAA, 2008.
- ¹⁴Jameson, A., "Aerodynamic Design Via Control Theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1988.
- ¹⁵Peter, J., Pham, C.-T., and Drullion, F., "Contribution to Discrete Implicit Gradient and Discrete Adjoint Method for Aerodynamic Shape Optimisation," *European Conference on Computational Fluid Dynamics*, ECCOMAS, 2004.
- ¹⁶Destarac, D. and van der Vooren, J., "Drag/Thrust Analysis of Jet-Propelled Transonic Transport Aircraft; Definition of Physical Drag Components," *Aerospace Science and Technology*, Vol. 8, 2004, pp. 545–556.
- ¹⁷Vanderplaats, *D.O.T. Users Manual*, Vanderplaats Research & Development, Inc, 1995.
- ¹⁸Bellman, R., *Adaptive Control Processes: A Guided Tour*, Princeton University Press, 1961.
- ¹⁹Laurenceau, J. and Sagaut, P., "Building Efficient Response Surfaces of Aerodynamic Functions with Kriging and Cokriging," *AIAA Journal*, Vol. 46, No. 2, 2008, pp. 498–507.
- ²⁰Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409–435.
- ²¹Liu, W., *Development of Gradient-Enhanced Kriging Approximations for Multidisciplinary Design Optimization*, Ph.D. thesis, University of Notre Dame, Notre Dame, IN, USA, 2003.
- ²²Kim, S. and Chung, H.-S., *Computational Science and Its Applications - ICCSA 2006*, chap. Multiobjective Optimization Using Adjoint Gradient Enhanced Approximation Models for Genetic Algorithms, 2006, pp. 491–502.
- ²³Sasena, M. J., *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*, Ph.D. thesis, University of Michigan, Department of Mechanical Engineering, Ann Arbor, Michigan, USA, 2002.
- ²⁴Booker, A. J., Dennis, J. E., Jr., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M. W., "A Rigorous Framework for Optimization of Expensive Functions by Surrogates," Tech. Rep. TR-98-47, Center for Research on Parallel Computation, 1998.
- ²⁵Jones, D. R., "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, , No. 21, 2001, pp. 345–383.
- ²⁶Meunier, M., "Simulation and Optimization of Flow Control Strategies for Novel High-Lift Configurations," *25th Applied Aerodynamics Conference*, No. AIAA-07-4276, AIAA, 2007.
- ²⁷Cox, D. and John, S., "SDO: A Statistical Method for Global Optimization," *Multidisciplinary design optimization (Hampton, VA, 1995)*, SIAM, Philadelphia, PA, 1997, pp. 315–329.
- ²⁸Wolpert, D. H. and Macready, W. G., "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, April 1997, pp. 67–82.

IV.2.1 Résumé de l'optimisation à 6 variables du profil RAE2822

Le premier cas test est un profil RAE2822 en condition de vol transsonique (cf. IV.2.IV.A). Il est paramétré par deux bosses positives sur son extradós, soit 6 variables. Ce problème est de faible dimension. Le diagramme en rectangle IV.1 résume les principales performances des trois optimiseurs.

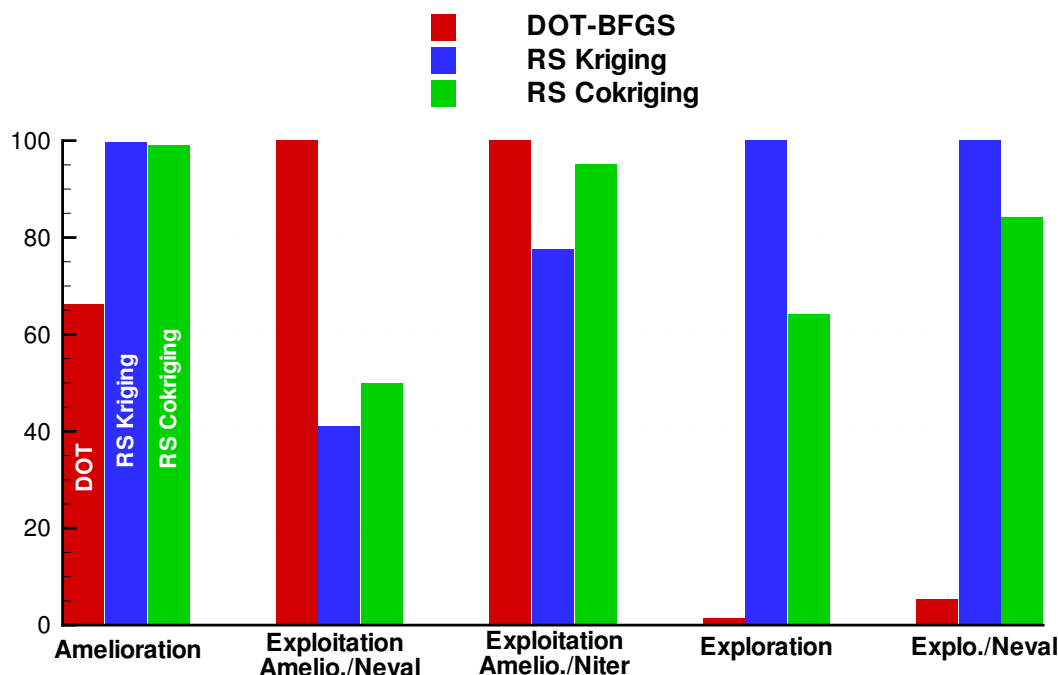


FIG. IV.1: Diagramme en rectangle résumant les performances des optimiseurs sur le cas du profil RAE2822 à 6 variables.

DOT converge très rapidement, mais vers un optimum local qui semble très lointain, en valeur de la fonction objectif, du minimum global. En effet, lors de la première itération des algorithmes par surfaces de réponse, la méthode de remplissage d'espace parvient à proposer deux formes plus performantes que la forme finale trouvée par l'optimiseur par gradient (IV.2.Colloque.Fig.7).

Lors des itérations suivantes, il apparaît que l'interpolation du vecteur gradient par la méthode RS Cokriging n'a pas un grand intérêt pour ce problème de faible complexité. Les deux nouveaux optimiseurs ont des performances équivalentes en terme de valeur finale et de coût de calcul. La méthode RS Cokriging est légèrement meilleure en terme d'exploitation, mais effectue moins d'exploration. Elle converge vers une valeur finale très légèrement moins bonne que celle de l'optimiseur RS Kriging.

La diminution de traînée de pression vient en grande partie d'une diminution de la traînée de choc (cf. IV.2.Colloque.Tab.1). L'optimiseur RS Kriging parvient à totalement supprimer la traînée de choc, en plaçant la première bosse au niveau du bord d'attaque et la deuxième bosse au niveau du choc, à 58% de la corde du profil (cf. IV.2.Colloque.Fig.6). Les amplitudes de déformation sont très faibles : environ deux millimètres alors que la corde du profil mesure un mètre. L'optimiseur RS Cokriging trouve une forme très proche, par contre la forme proposée par DOT est très différente. L'optimisation par gradient ne permet pas de supprimer totalement le choc et est de plus pénalisée par une traînée de pression visqueuse plus importante. En IV.2, les traînées sont mises à l'échelle par rapport à la configuration de référence. La solution à 3,3% de traînée de choc pour DOT (cf. IV.2.Colloque.Tab.1) correspond en valeur absolue à moins de 2 points de traînée.

Finalement, en faible dimension, la méthode RS Kriging présente le meilleur compromis entre exploration et exploitation.

IV.2.2 Résumé de l'optimisation à 45 variables de l'aile AS28

Le second cas test est une aile AS28 en condition de vol de croisière (cf. IV.2.IV.B). Elle est paramétrée par 15 bosses positives sur son extradados, soit 45 variables. Les bosses sont réparties par groupe de trois selon cinq sections en envergure. Ce problème est de grande dimension.

DOT converge en seulement 26 calculs vers son minimum local (IV.2.Colloque.Fig.10). C'est, de loin, la meilleure méthode en terme d'exploitation. En terme de valeur finale, les deux nouveaux optimiseurs donnent de meilleurs résultats mais effectuent plus de calculs et d'itérations. L'optimisation par Krigeage est stoppée car elle atteint le nombre de calcul maximum. On considère l'optimisation par Cokrigeage convergée car aucune amélioration n'est obtenue durant 20 itérations consécutives.

L'échantillonnage initial a été choisi très grossier afin de mettre en défaut le Krigeage. Il ne contient que 11 points. Par rapport aux tailles d'échantillonnages indicatives données en II.3.2.1, il faudrait au moins 1081 échantillons pour avoir une surface de réponse valide (capturer des tendances quadratiques) et idéalement 10^{45} échantillons pour considérer la surface de réponse fiable sur tout le domaine. Ces 11 points ne permettent même pas d'établir une régression linéaire de la fonction.

Toutefois, il s'avère que cela ne l'empêche pas de trouver une solution meilleure que celle proposée par l'optimiseur par gradient. Malgré la rapidité de convergence de DOT, l'optimiseur RS Cokriging devient plus performant après seulement 16 itérations. A convergence de DOT, l'optimiseur RS Cokriging propose une solution dont la traînée de pression est inférieure d'environ 0,5 pourcents⁴. La convergence de l'optimiseur RS Cokriging atteint un plateau après environ 40 itérations. Sur la figure IV.2, on peut constater que RS Kriging effectue deux fois plus d'exploration que RS Cokriging. En terme d'exploitation, les deux méthodes semblent équivalentes, mais RS Cokriging trouve une meilleure solution.

Finalement, en grande dimension, la méthode RS Cokriging présente le meilleur compromis entre exploration et exploitation.

⁴Toutes les valeurs de traînée sont mises à l'échelle par rapport à la traînée obtenue sur la forme non déformée.

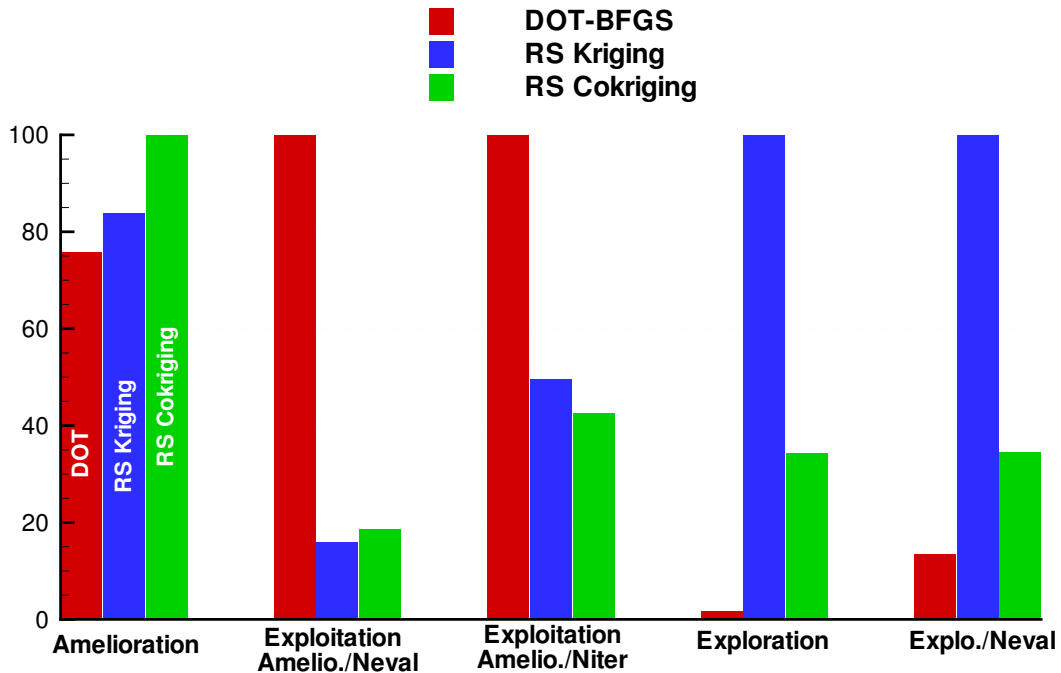


FIG. IV.2: Diagramme en rectangle résumant les performances des optimiseurs sur le cas de l'aile AS28 à 45 variables.

IV.3 Optimisation à 48 variables d'un avion complet (configuration AS28)

En plus des deux cas tests précédents, de nombreux autres cas ont été étudiés dont le plus coûteux en temps de calcul est une configuration AS28 complète. Cette configuration est constituée d'un demi-fuselage, une aile, un mât et une nacelle non motorisée et représente un avion de type Airbus. Elle sera étudiée en condition de croisière. Le nombre de Mach vaut $M = 0,8$ et l'angle d'incidence vaut $2,2^\circ$, l'écoulement est transsonique. Elle est modélisée par un maillage formé de 80 blocs contenant au total $7,5 \cdot 10^6$ nœuds (Figure IV.3). Tel que décrit en I.3.3, le système d'équations RANS complétée par le modèle de turbulence de Spalart-Allmaras est résolu pour simuler l'écoulement. La simulation numérique stationnaire est réalisée sur 8 processeurs AMD Opteron 275. Le temps de restitution pour un calcul direct est de 6 heures pour effectuer 800 itérations. Les figures IV.3 et IV.7 montrent le champ surfacique du coefficient de pression sur la forme de référence.

La fonction objectif est le coefficient de traînée totale calculé par l'approche champ lointain,

$$F = Cd_f + Cd_p = Cd_f + Cd_{vp} + Cd_i + Cd_w \quad (IV.1)$$

Elle est mise à l'échelle par la traînée de la forme initiale, de sorte qu'une forme ayant pour traînée indicielle 100 est aussi performante que la forme initiale. L'extrados de l'aile est paramétré par 16 bosses d'amplitude positive. Elles sont réparties par groupe de quatre selon quatre sections en envergure. L'amplitude maximale de déformation est de 50 millimètres, alors que l'envergure de l'aile est de 25 mètres et la corde moyenne mesure 5,4 mètres. Les bosses sont orientées selon l'axe vertical. Seule la partie frontale de l'extrados de l'aile est déformée. Environ un tiers de la surface de l'extrados est fixe, à savoir la partie proche du bord de fuite. Par rapport à la paramétrisation mise en place sur l'aile isolée (IV.2), les déformations peuvent s'étendre selon l'envergure complète de l'aile. La section fixe en bout d'aile a été supprimée afin de pouvoir obtenir des champs de déformation plus réguliers.

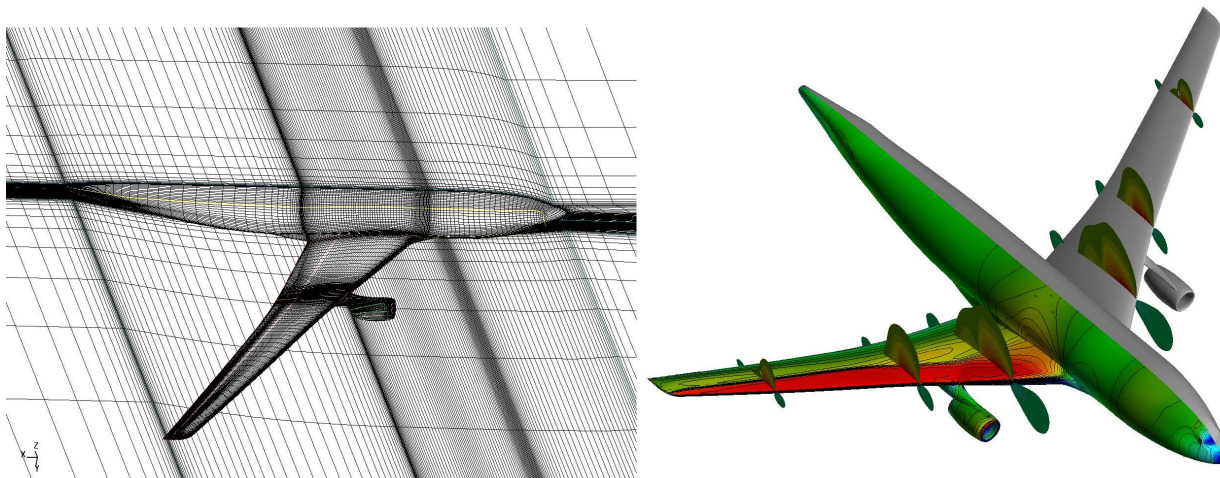


FIG. IV.3: Configuration AS28 non modifiée. Maillage à gauche et coefficient de pression à droite.

Le problème comporte 48 variables de forme. Il est de complexité équivalente au problème d'optimisation de l'aile isolée, mais l'évaluateur est beaucoup plus coûteux. Afin de réduire le temps de restitution des optimisations par surfaces de réponse, un échantillonnage initial plus important est employé. Alors que sur l'aile AS28, l'échantillonnage initial contenait 11 points, il en contient ici 51. Ces points sont distribués selon la méthode LHS et la configuration initiale est incluse dans cet échantillonnage. Avec une méthode de régression polynomiale, cette taille d'échantillonnage serait tout juste suffisante à effectuer une régression linéaire. Pour construire une surface de réponse quadratique, il faudrait au moins 1225 échantillons et pour une surface de réponse globale, il en faudrait 10^{48} (cf. II.3.2.1). Nous sommes donc toujours dans le cadre de l'optimisation par surfaces de réponse locales.

Durant la première itération des optimiseurs par surfaces de réponse, ces 51 calculs sont effectués simultanément⁵. Dans un contexte industriel, le coût représenté par ces 51 calculs effectués en parallèle n'a rien d'inhabituel ($51 \times 8 = 408$ processeurs pendant 6 heures). Ce coût est justifié car il permet d'accélérer le processus de conception.

Le Cokrigage indirect tronqué en échantillons, utilisé par l'optimiseur RS Cokriging, interpole $n_{aug} = 10$ vecteurs gradients. Cela correspond à une quantité d'information équivalente à 480 échantillons supplémentaires. L'optimiseur DOT est initialisé par des bosses larges et équidistribuées d'amplitude nulle.

Le tableau IV.1 résume les performances aérodynamiques des formes proposées par chacun des optimiseurs. Au regard du problème posé, la meilleure solution est obtenue par l'optimiseur RS Cokriging, avec une amélioration de 4,2% de la fonction objectif. Cela représente environ 10 points de traînée. Les deux autres optimiseurs donnent des solutions différentes. DOT réduit la fonction de 3,8% et RS Kriging de 2,2%.

Les figures IV.4-IV.6 présentent les champs de déformation optimale colorés par l'amplitude de la déformation, ainsi que la répartition surfacique du coefficient de pression sur la forme optimale. Comme cela a été observé sur l'optimisation de l'aile, DOT converge vers une déformation régulière, large et étendue. Cette déformation est obtenue avec 2 bosses non nulles par section, soit 8 bosses au total. Sur les 48 degrés de liberté seuls 24 sont réellement utilisées. Les déformations proposées par les optimiseurs par surfaces de réponse sont plus complexes. Elles sont effectivement définies à partir d'un plus grand nombre de bosses que ce qu'utilise DOT (15 bosses non nulles pour RS Kriging et 13 bosses non nulles pour RS Cokriging). Les méthodes RS Kriging et RS Cokriging profitent plus des

⁵En réalité, la base d'échantillonnage est calculée une seule fois. Les deux optimiseurs RS Kriging et RS Cokriging utilisent la même base de données initiale.

	Référence	DOT-BFGS	RS Kriging	RS Cokriging
$Cd = F$	100	96,2 (-3,8%)	97,8 (-2,2%)	95,8 (-4,2%)
Cd_f	36,5	36,6	36,6	36,5
Cd_p	63,5	59,6	61,5	59,3
Cd_w	7,2	2,9	3,8	1,9
Cd_{vp}	19,4	19,2	21,8	22,9
Cd_i	36,9	37,5	35,9	34,6
Cl	100	101,7	99,2	96,1
$\mathcal{O}(\Delta Cd_i)$ dû au ΔCl	0	-1	+0.5	+2.5

TAB. IV.1: Coefficients de traînée (mis à l'échelle) de la configuration AS28.

degrés de liberté autorisés.

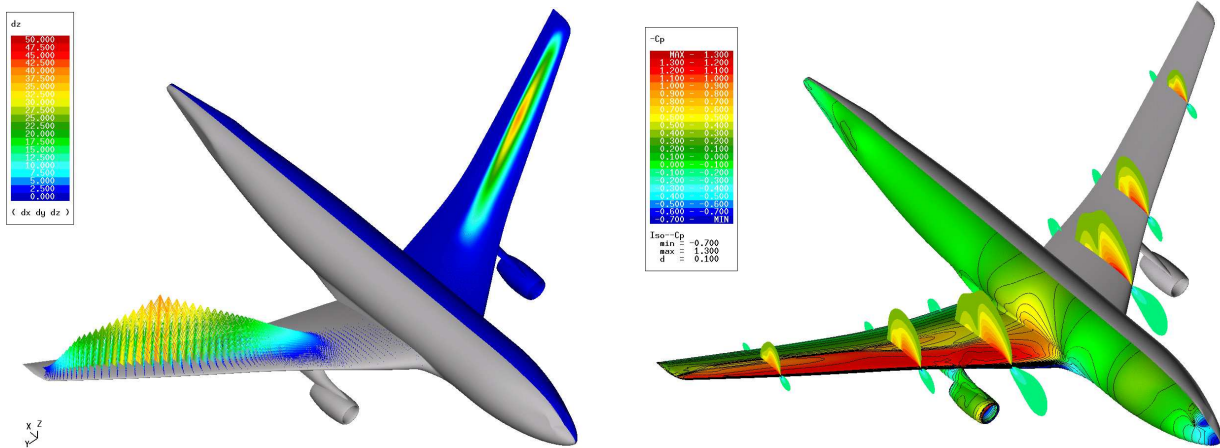


FIG. IV.4: Optimisation par la méthode DOT-BFGS. Champ de déformation à gauche, coefficient de pression à droite.

La figure IV.9 montre l'historique de convergence des trois optimiseurs. La fonction objectif y est tracée en fonction du nombre d'évaluations de fonctions et du nombre d'itérations du processus d'optimisation.

DOT converge après avoir évalué la fonction sur 60 formes différentes parce qu'il ne parvient plus à améliorer la fonction lors des deux dernières itérations. L'optimiseur RS Kriging converge après 143 évaluations car il ne parvient plus à améliorer la fonction lors des vingt dernières itérations. L'optimiseur RS Cokriging a été stoppé manuellement, car 20 itérations ont été suffisantes pour égaler l'amélioration de DOT. Au bout de 24 itérations la valeur finale obtenue par DOT est largement dépassée. Comme DOT atteint son plateau de convergence au bout de 26 itérations, on peut affirmer que l'optimiseur RS Cokriging est ici plus performant en terme d'exploitation (amélioration par itération). Ces deux optimiseurs convergent à une vitesse semblable en fonction du nombre d'itérations, mais DOT atteint son plateau de convergence plus tôt et pour une moins bonne valeur de la fonction. Cela se vérifie sur le tableau IV.2 et le diagramme récapitulatif en figure IV.10. Cette performance vient, en partie, du nombre d'évaluations (51) effectuées lors de la première itération des optimiseurs par surfaces de réponse. En effet, DOT reste meilleur en terme d'amélioration par évaluation.

Sur la figure IV.9, les 51 premières évaluations des méthodes RS Kriging et RS Cokriging correspondent à la phase de remplissage d'espace par la méthode LHS (la première itération). Lors de cette

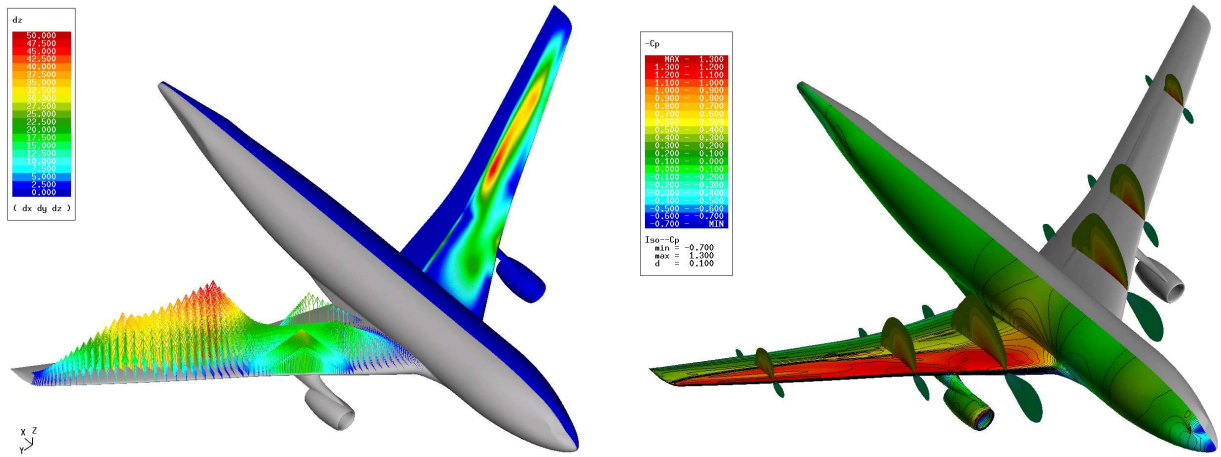


FIG. IV.5: Optimisation par la méthode RS Kriging. Champ de déformation à gauche, coefficient de pression à droite.

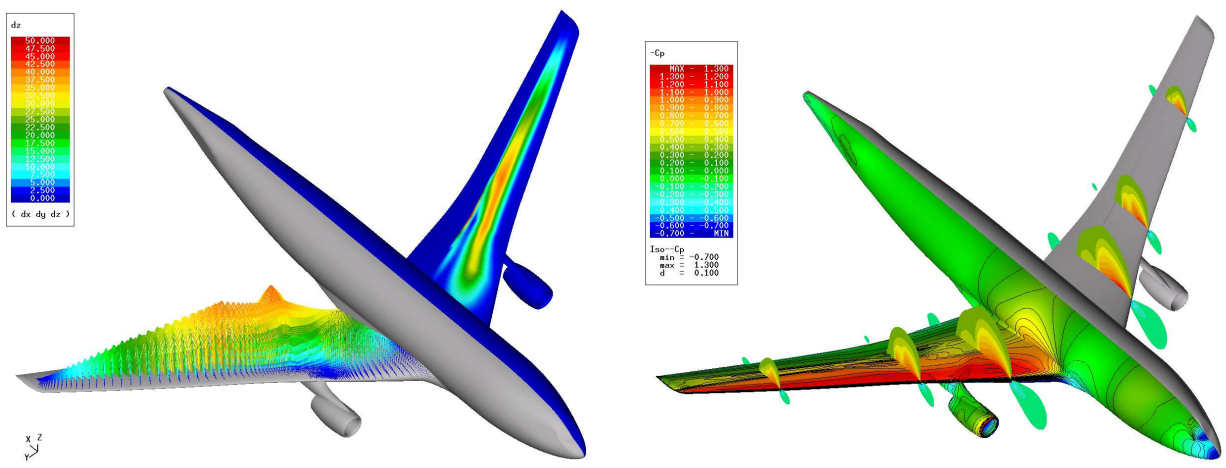


FIG. IV.6: Optimisation par la méthode RS Cokriging. Champ de déformation à gauche, coefficient de pression à droite.

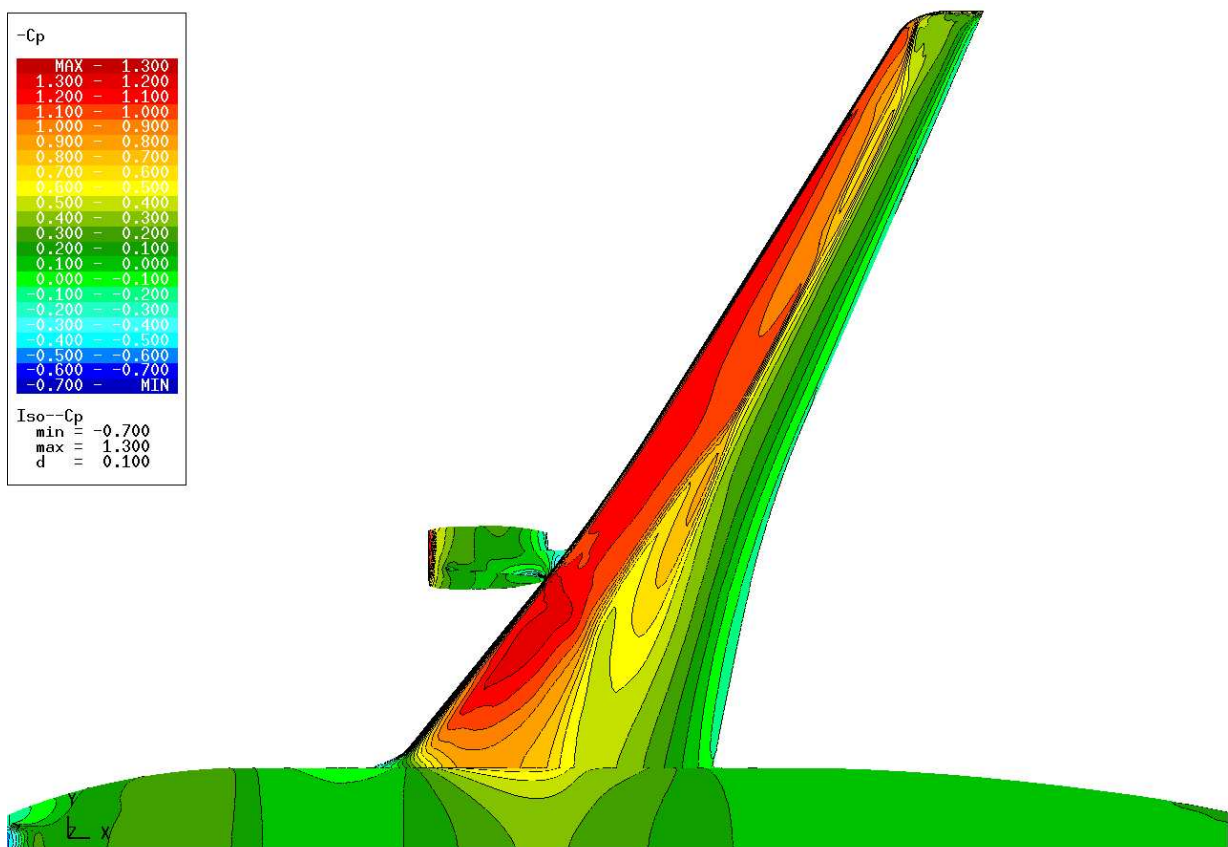
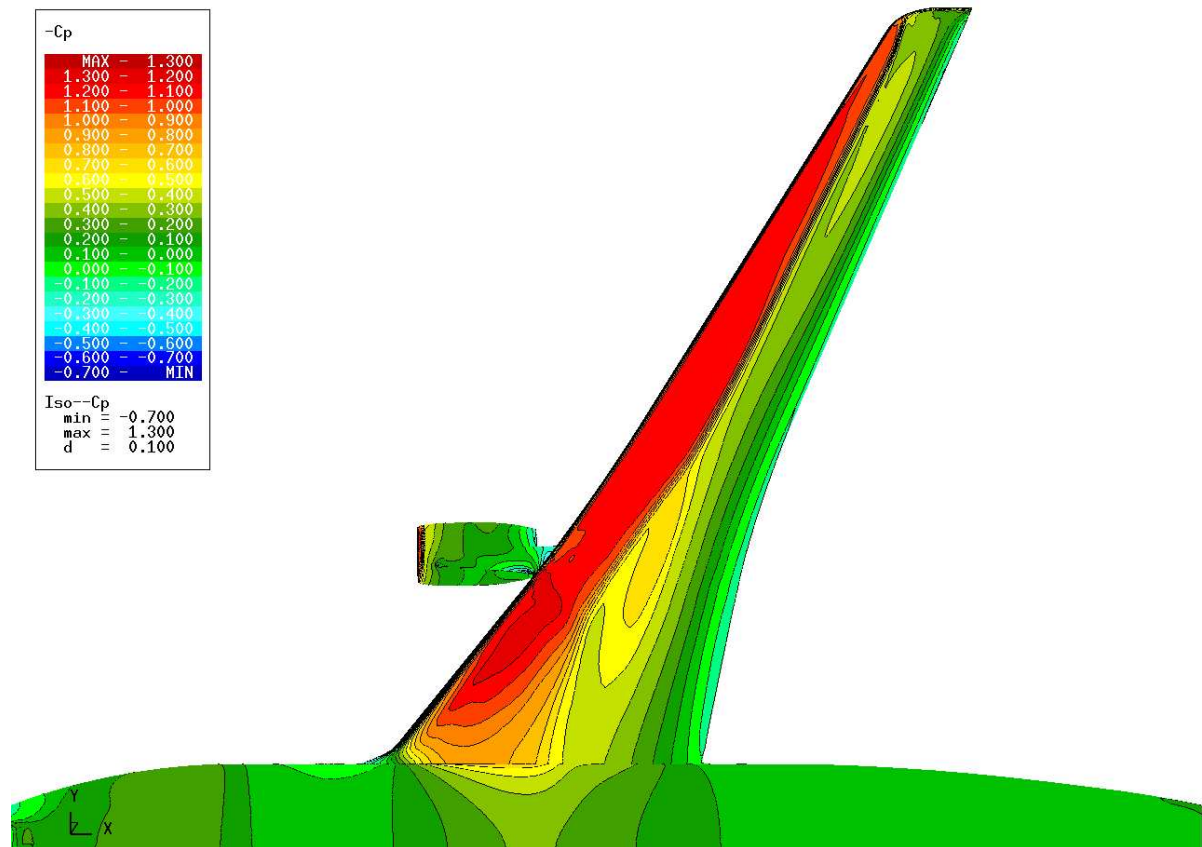


FIG. IV.7: Coefficients de pression de la forme initiale (haut) et de la solution trouvée par DOT (bas).

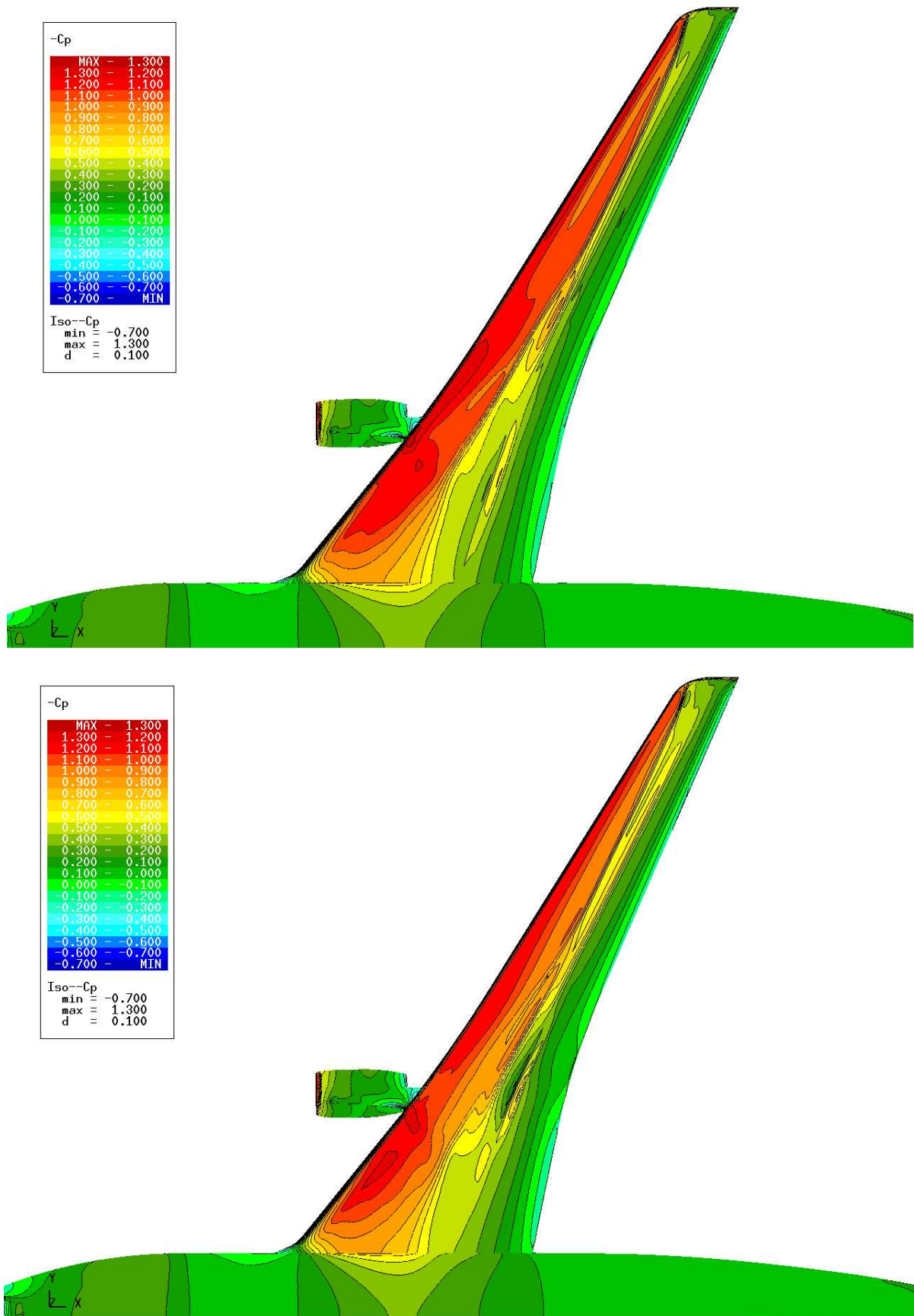


FIG. IV.8: Coefficients de pression des solutions trouvées par RS Kriging (haut) et RS Cokriging (bas).

phase, aucune amélioration de la fonction objectif n'est obtenue, bien que l'espace de conception soit assez largement exploré. Cela montre que le problème à résoudre n'est pas trivial et l'emploi de l'optimisation est donc justifié. En outre, les trois optimiseurs convergeant vers trois solutions différentes, ce problème semble admettre plusieurs minima.

Les optimiseurs RS Kriging et RS Cokriging sont à peu près équivalents en terme d'exploration. En revanche, DOT effectue très peu d'exploration, ce qui peut expliquer pourquoi il converge vers une forme simple. Malgré le fait que l'échantillonnage initial soit plus important, l'optimiseur RS Kriging reste peu satisfaisant en haute dimension par rapport à l'optimiseur basé sur le Cokrigage. Finalement, le nouvel optimiseur RS Cokriging présente, ici encore, le meilleur compromis entre exploration et exploitation.

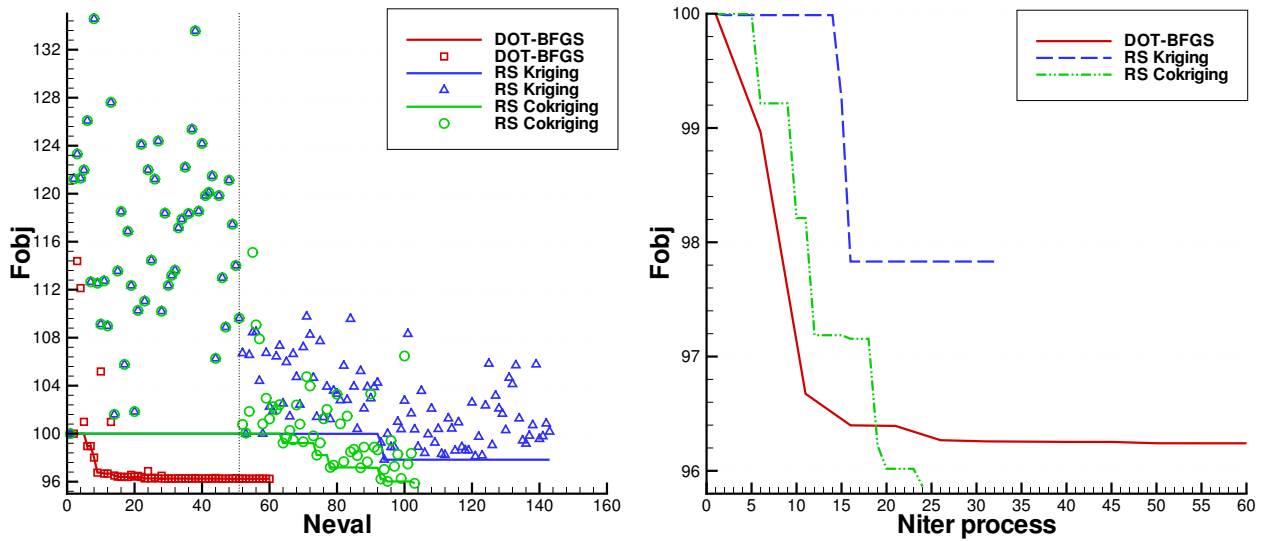


FIG. IV.9: Historique de convergence des optimisations sur le cas de la configuration AS28 à 48 variables.

	DOT-BFGS	RS Kriging	RS Cokriging
F_c^{ref}	96,2 (-3,8%)	97,8 (-2,2%)	95,8 (-4,2%)
$n_{iter} ; n_{eval} + n_{grad}; n_s^{ini}$	60 ; 60+11 ; 0	32 ; 143+0 ; 51	24 ; 103+10 ; 51
$\sum F_c - F_c^{ref} $	41,7	1178	970
$\sum F_c - F_c^{ref} / n_{iter}$	0,7	36,8	40,4
$\sum F_c - F_c^{ref} / n_{eval}$	0,7	8,2	9,4
$\sum \ \bar{x}_c - \bar{x}_c^{ref}\ / n_{dv}$	0,13	8,43	5,13
$\sum \ \bar{x}_c - \bar{x}_c^{ref}\ / n_{iter} / n_{dv}$	$0,2 \cdot 10^{-3}$	0,27	0,22
$\sum \ \bar{x}_c - \bar{x}_c^{ref}\ / n_{eval} / n_{dv}$	$0,2 \cdot 10^{-3}$	$59 \cdot 10^{-3}$	$50 \cdot 10^{-3}$

TAB. IV.2: Performance des optimiseurs DOT, RS Kriging et RS Cokriging sur le cas de la configuration AS28 à 48 variables.

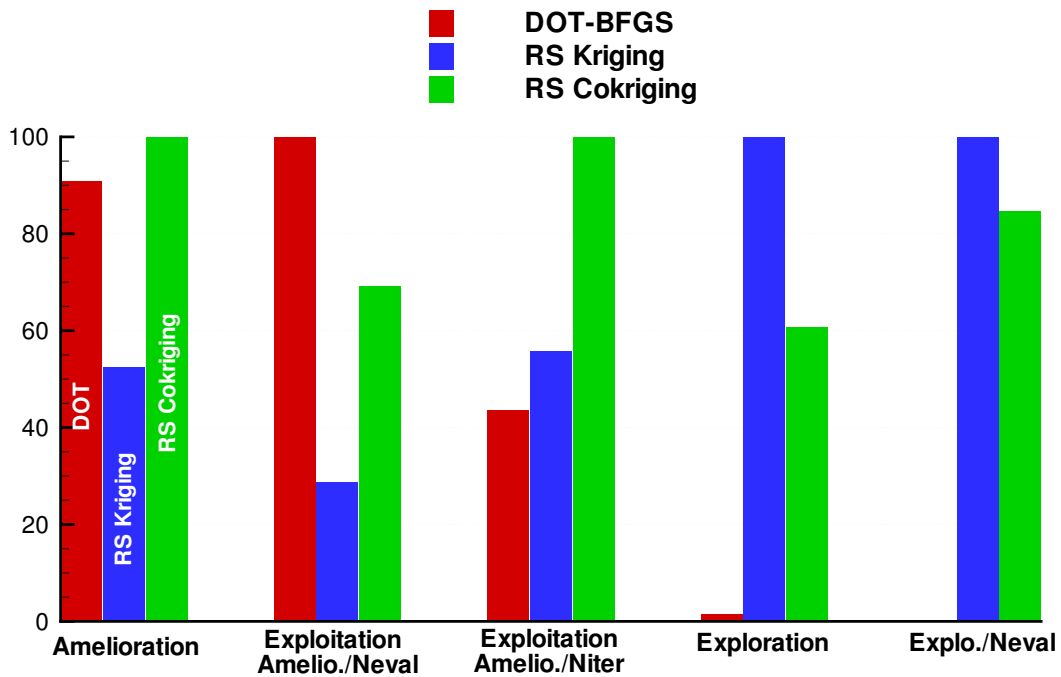


FIG. IV.10: Diagramme en rectangle résumant les performances des optimiseurs sur le cas de la configuration AS28 à 48 variables.

IV.4 Compléments

IV.4.1 Discussion sur la définition du problème

IV.4.1.1 Déformations positives

Sur les trois problèmes présentés ici, la paramétrisation est toujours formée de bosses d'amplitude positive. L'intérêt d'une déformation positive de l'aile est qu'elle ne réduit pas l'encombrement intérieur. L'impact sur la structure primaire interne à l'aile est très limité. Ce type de déformation est donc plus facile à intégrer lorsque le cycle de conception est déjà à un stade de maturité avancée (en conception détaillée). Cela réduit toutefois l'espace de conception. En effet, il est tout à fait possible d'obtenir un champ de déformation positif en combinant des bosses positives et négatives car les contributions de chacune des bosses s'ajoutent. Il serait donc préférable de considérer une contrainte géométrique sur l'amplitude du champ de déformation plutôt que sur l'amplitude de chacune des bosses prises individuellement. Toutefois, les problèmes d'optimisation sous contraintes ne seront pas abordés ici.

IV.4.1.2 Traînée de frottement

Sur les cas tests du profil RAE2822 et de l'aile AS28, la fonction objectif est la traînée de pression, Cd_p . En réalité, c'est la traînée totale de la forme ($Cd = Cd_p + Cd_f$) que le concepteur cherche à réduire. Logiquement, il faudrait donc inclure dans la définition de l'objectif un terme supplémentaire pour prendre en compte la traînée de frottement, Cd_f . En pratique, il est inutile d'inclure ce terme étant donnée l'amplitude de déformation autorisée.

La fonction objectif traînée définie sur le problème de l'avion complet est la traînée totale de la forme. Le tableau IV.1 résume les performances aérodynamiques des formes proposées par chacun des optimiseurs. On peut constater que la traînée de frottement, Cd_f , varie effectivement très peu (0,1%) lors des trois optimisations réalisées. Vis-à-vis des déformations autorisées, la surface totale de l'avion

change trop peu pour avoir une influence sur ce terme. C'est la traînée de pression, Cd_p , qui change notablement, surtout à travers une diminution de la traînée de choc, Cd_w .

IV.4.1.3 Portance et traînée induite

Sur le cas de l'avion complet, on peut observer sur le tableau IV.1 que pour les optimisations par surfaces de réponse, la réduction de traînée s'accompagne d'une diminution de portance non négligeable. En effet, les optimiseurs RS Kriging et RS Cokriging obtiennent une partie de leur diminution de traînée total (Cd , équation IV.1) par une diminution de la traînée induite Cd_i .

La quantité $\mathcal{O}(\Delta Cd_i)$ représente l'ordre de grandeur de l'augmentation de traînée à ajouter pour retrouver la portance initiale. Elle est calculée avec une simple relation $Cd_i = kCl^2$ [1], où la constante k est calculée sur la forme connue à incidence $2,2^\circ$. Du point de vue optimisation, l'optimiseur RS Cokriging converge vers la meilleure solution. Du point de vue aérodynamique, il semble que DOT fournit ici la meilleure solution puisqu'il réduit la traînée en augmentant la portance. Cela démontre une défaillance au niveau de la définition du problème d'optimisation. Elle s'explique par la manière de définir le point de vol. En simulation numérique, le point de vol est défini par le nombre de Mach et l'angle d'incidence de l'écoulement. En réalité, le point de vol est défini par le nombre de Mach et la portance Cl_0 qui compense le poids de la forme.

Pour remédier à cette défaillance, il faudrait inclure l'angle d'incidence parmi les variables de forme. Le niveau de portance à respecter pourrait ensuite être représenté par une contrainte d'égalité ($Cl = Cl_0$) ou d'inégalité ($Cl \geq Cl_0$) à respecter. Afin d'éviter de traiter un problème d'optimisation sous contrainte, il est possible de directement substituer à la contrainte un terme de pénalisation de la fonction objectif. Par exemple en utilisant une relation du type $F = Cd + k(Cl_0^2 - Cl^2)$. La constante k peut être calculée à partir d'une simulation numérique sur la forme initiale.

Sur le cas de l'optimisation de l'aile isolée, cette précaution n'a pas été prise et pourtant les optimiseurs fournissent des solutions qui réduisent la portance et augmentent la portance. L'inclusion de la portance au problème de minimisation de traînée n'est donc pas essentielle, mais recommandée.

Finalement, il apparaît ici que les optimiseurs par surfaces de réponse, qui recherchent l'optimum global du problème, sont plus sensibles à une définition défaillante du problème d'optimisation que l'algorithme DOT qui effectue une recherche locale.

IV.4.1.4 Irrégularité du champ de déformation

Sur les cas de l'aile AS28 et de l'avion complet, les déformations proposées par DOT sont larges et régulières. Sur les figures IV.2.Colloque.Fig.11 et IV.4, le champ de coefficient de pression ne paraît pas trop perturbé. Les optimiseurs par surfaces de réponse donnent des formes optimales beaucoup plus complexes (IV.2.Colloque.Fig.12 et IV.2.Colloque.Fig.13 pour l'aile et IV.5 IV.6 pour l'avion complet). Ces formes très irrégulières sont inhabituelles. C'est parce que les deux nouveaux optimiseurs effectuent plus d'exploration de l'espace de conception qu'elles ont pu être obtenues. Bien qu'elles soient très performantes vis-à-vis du problème d'optimisation posé, on peut supposer qu'elles soient irréalistes en pratique. Pour les qualifier complètement, il faudrait envisager deux étapes de validation.

D'une part, l'optimisation est effectuée sur le maillage, la forme discrétisée, dont la qualité a été adaptée vis-à-vis de la forme initiale. L'ajout de déformations locales qui n'affectent qu'un faible nombre de mailles risque d'entraîner une augmentation de l'erreur numérique de discrétisation. Pour quantifier cette erreur [8], il faudrait adapter le maillage à la forme optimale puis effectuer une nouvelle simulation numérique.

D'autre part, il faudrait considérer d'autres aspects de la conception, tels que les performances sur tout le domaine de vol (polaire aérodynamique, divergence en Mach, portance maximale) ou des charges structurales (répartition de charge selon l'envergure, tremblement) [1]. Cela n'est pas l'objet de cette étude, mais grâce à quelques calculs à différents angles d'incidence, on peut constater que les

performances des formes de l'aile AS28 optimisées changent selon le domaine considéré. Les polaires approchées en IV.2.Colloque.Fig.18 et IV.11 représentent l'évolution de la traînée totale (traînée de pression plus traînée de frottement) en fonction de la portance de l'aile AS28. Ces polaires sont obtenues avec une méthode de Krigeage dont les points de construction sont représentés par des symboles. Elles montrent que, pour les portances faibles, la forme initiale est légèrement plus performante, alors que la forme proposée par DOT est la meilleure pour les portances élevées.

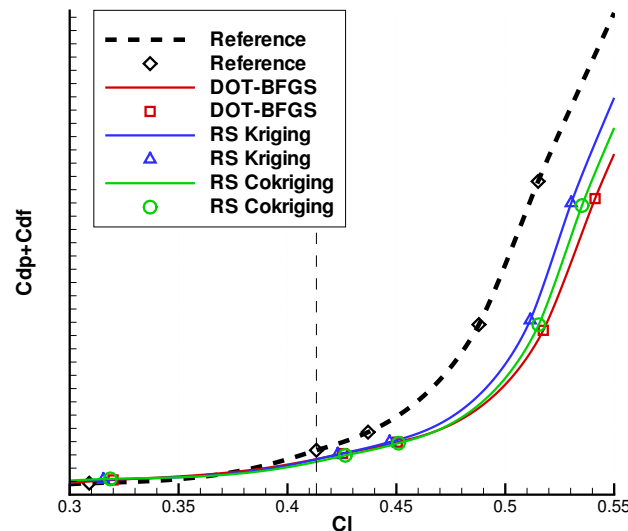


FIG. IV.11: Evolution de la traînée en fonction de la portance pour l'aile AS28. La forme de référence est comparée aux formes optimales.

Avec DOT, le contrôle *a posteriori* des éventuelles anomalies par le concepteur est généralement suffisant. Avec des optimiseurs recherchant le minimum global du problème, tels que les deux nouveaux optimiseurs par surfaces de réponse, le risque d'aboutir à une forme irrégulière est plus important. Il est donc primordial de définir soigneusement le problème d'optimisation *a priori*. Au vue des courbes IV.11, une optimisation multipoint réduisant la traînée en plusieurs points de vol n'aurait pas permis d'exclure les déformations irrégulières. Par contre, cela aurait pu être effectué en fixant la valeur de l'expansion des bosses de Hicks-Henne (cf. I.3.1) pour n'étudier que des combinaisons de bosses larges (exposant fixe et égale à $\beta = 2$). De plus, cela aurait permis de réduire le domaine de recherche (nombre de variables de forme) et donc facilité la tâche des optimiseurs.

IV.4.2 Validation des solutions obtenues

La fonction objectif n'étant pas analytique, il est difficile de vérifier si, du point de vue mathématique, la solution trouvée est bien un minimum (au moins local). Telle qu'introduite en I.1, la condition suffisante KKT (norme du gradient) permet de s'en assurer lorsque le gradient est connu. Elle fait partie des critères de convergence de DOT. Cependant, sur les cas présentés dans ce chapitre, DOT converge parce qu'il ne parvient pas à améliorer la fonction durant deux itérations successives. Avec les optimiseurs par surfaces de réponse, la condition KKT n'est pas calculée.

L'optimiseur DOT est un algorithme d'optimisation locale par gradient. Si après l'avoir fait converger une première fois, on utilise la solution obtenue pour initialiser une deuxième optimisation, alors DOT ne réussira pas à améliorer la solution. Il convergera vers la forme initiale de la deuxième optimisation qui correspond à la forme finale de la première optimisation. Cela a été vérifié pour les cas tests du profil RAE2822 et de l'aile AS28. Pour validation, les formes obtenues par les optimiseurs

RS Kriging et RS Cokriging ont aussi été utilisées comme points de départ d'optimisations effectuées avec DOT. Lorsque DOT parvient à améliorer une de ces solutions, cela signifie qu'elle ne correspond pas à un minimum local. Les figures IV.12 et IV.13 montrent les résultats obtenus.

Sur le cas du profil RAE2822, on constate que DOT ne parvient pas à améliorer la forme optimale trouvée par les optimiseurs RS Cokriging et DOT. Toutefois, l'optimiseur RS Kriging semble avoir convergé prématurément. DOT arrive à améliorer légèrement la solution obtenue par RS Kriging. La valeur indicielle de la traînée de pression diminue de 76,7 à 76,4.

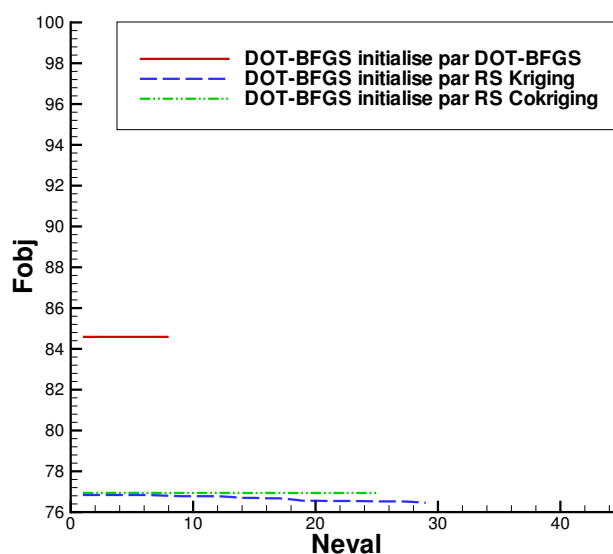


FIG. IV.12: Historique de convergence d'optimisations par DOT-BFGS sur le cas du profil RAE2822. Initialisation par les formes optimales obtenues par : DOT, RS Kriging et RS Cokriging.

Ce léger défaut de convergence de l'optimiseur RS Kriging peut être attribué aux critères de convergence employés. Dans ce cas, l'optimiseur a convergé car il n'a pas pu améliorer la fonction lors des 20 dernières itérations. Ce critère arbitraire n'a aucun fondement mathématique et a été implémenté dans le but d'éviter des calculs inutiles. Dans l'idéal, le critère de convergence pour les optimiseurs par surfaces de réponse devrait consister à s'arrêter lorsque les trois critères de recherche sur la surface de réponse ($\min(F)$, $\min(LCB_1)$ et $\min(LCB_{-1})$) indiquent le même point et que ce point a déjà été calculé. Cela nécessiterait d'avoir une faible erreur prédite sur le domaine, donc beaucoup d'échantillons (cf. II.3.2.1). Ce n'est pas possible étant donné les contraintes énoncées en introduction de ce chapitre (cf. IV.1).

Cela ne discrédite pas pour autant l'optimiseur RS Kriging. En pratique, les concepteurs veulent obtenir une réduction importante et rapide de la fonction objectif. La preuve que la solution est un minimum n'est pas nécessaire.

La figure IV.13 montre l'historique de convergence d'optimisations par DOT sur le cas de l'aile AS28. Les formes utilisés pour initialiser DOT sont les formes optimales obtenues par DOT, RS Kriging et RS Cokriging. On constate que DOT ne parvient pas à améliorer les formes trouvées par ces optimiseurs.

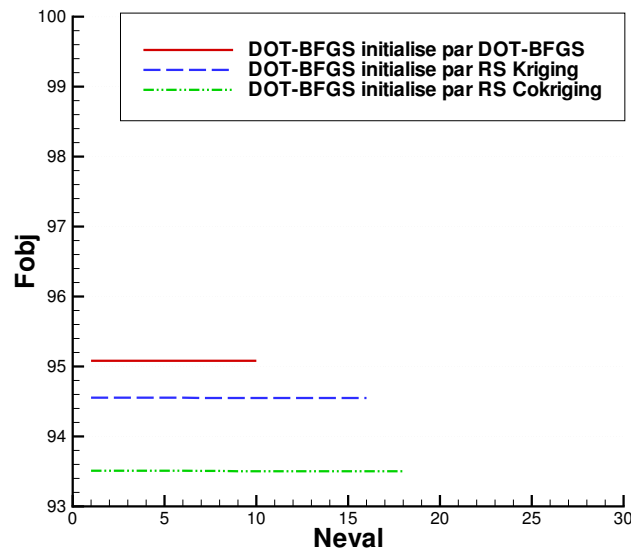


FIG. IV.13: Historique de convergence d'optimisations par DOT sur le cas de l'aile AS28. Initialisation par les formes optimales obtenues par : DOT, RS Kriging et RS Cokriging.

Chaque optimiseur semble avoir convergé vers son propre minimum local. Cela confirme le fait que les problèmes de minimisation de traînée admettent de nombreux minima locaux. Toutefois, cela ne signifie pas que les optimiseurs par surfaces de réponse locales sont des algorithmes d'optimisation locale. Ils effectuent une recherche de l'optimum global, mais ne peuvent converger vers celui-ci étant donné le nombre de calculs impartis.

IV.5 Bilan

D'après les trois cas de minimisation de traînée étudiés, en dimension 6, 45 et 48, les deux optimiseurs introduits (RS Kriging⁶ et RS Cokriging⁷) ont l'air très prometteur. Les performances de ces optimiseurs sont résumées sur les diagrammes en figure IV.1, IV.2 et IV.10. Comparés à la référence (DOT-BFGS), ils permettent d'atteindre une meilleure solution pour un surcoût modéré. Alors que DOT converge généralement vers une déformation assez simple de la forme initiale, ces nouveaux algorithmes recherchent l'optimum global du problème et semblent mieux profiter des degrés de liberté autorisés pour la déformation. Ils offrent donc une fonctionnalité complémentaire.

Leur surcoût, non négligeable en nombre d'évaluations, est amoindri en temps de restitution (nombre d'itérations du processus) par le lancement de calculs en parallèle. Par exemple sur le cas de l'avion complet, l'optimiseur RS Cokriging converge aussi rapidement que DOT mais vers une meilleure solution.

Finalement, il est possible d'identifier leurs domaines de prédilection :

- Lorsqu'un minimum local proche de la forme initiale est suffisant ou que l'amélioration de la fonction doit être atteinte en un nombre de calcul très faible (une cinquantaine), l'algorithme de descente par gradient est le mieux adapté.
- Lorsqu'il est prioritaire de trouver la meilleure amélioration ou d'explorer l'ensemble du domaine de conception pour trouver une solution innovante et qu'une centaine de calculs peuvent être envisagés, les optimiseurs par surfaces de réponse locales sont plus adaptés. En dimension élevée, l'optimiseur RS Cokriging à base de Cokrigeage est le plus performant. En faible dimension,

⁶RS Kriging : optimiseur basé sur le Krigeage.

⁷RS Cokriging : optimiseur basé sur le Cokrigeage.

l'emploi du gradient pour construire la surface de réponse par Cokrigeage n'est pas nécessaire et l'optimiseur RS Kriging à base de Krigeage est le plus performant.

Ce classement des optimiseurs peut être mis en relation avec la quantité d'information sur la fonction qu'ils utilisent pour décider quelle sera la nouvelle forme à évaluer. En quantité d'information, le gradient est un vecteur qui apporte une contribution égale au nombre de variables, n_{dv} . L'optimiseur DOT (cf. I.2.5) calcule sa direction de recherche par comparaison de deux vecteurs gradients. Lors de sa recherche linéaire, il effectue environ 10 évaluations de fonction. Au total, on peut dire que DOT utilise une quantité d'information de l'ordre de $N_{DOT} \approx 10 + 2 \cdot n_{dv}$. La quantité d'information utilisée par RS Kriging dépend du nombre d'évaluations, $N_{RSKriging} = n_{eval}$, ce qui explique pourquoi ses performances se dégradent lorsque la dimension du problème augmente. Comme l'optimiseur RS Cokriging interpole n_{aug} vecteurs gradients, il reste performant en grande dimension, $N_{RSCokriging} = n_{eval} + n_{aug} \cdot n_{dv}$.

Algorithme	DOT	RS Kriging	RS Cokriging
Quantité d'information	$\approx 10 + 2 \cdot n_{dv}$	n_{eval}	$n_{eval} + n_{aug} \cdot n_{dv}$

TAB. IV.3: Quantité d'information utilisée pour définir la forme suivante.

Les optimiseurs par surfaces de réponse gardent en mémoire toute l'information connue sur la fonction. L'algorithme de descente par gradient DOT n'utilise qu'une information locale sur la fonction. Lorsqu'une nouvelle itération interne de l'algorithme DOT commence, seule l'information du gradient de l'itération précédente est conservée.

Conclusion

L'objectif principal des travaux présentés dans cette thèse était d'améliorer un outil d'optimisation de formes aérodynamiques, à savoir l'outil OPTaliA utilisé par les concepteurs d'Airbus en raison de sa rapidité de convergence. Son algorithme d'optimisation, une méthode de descente par gradient (quasi-Newton BFGS), est performant en terme d'exploitation. Bien qu'il soit rapide, cet algorithme converge vers un minimum local. Il n'est pas performant en terme d'exploration. Cet outil permet d'améliorer la traînée d'une forme à un faible coût de calcul, en dépit du nombre important de variables de forme. Pour cela, il s'appuie sur une méthode efficace pour le calcul de la sensibilité de la fonction objectif par rapport aux variables de forme, une méthode adjointe.

Deux axes d'amélioration ont été envisagés :

- l'extension de l'outil aux problèmes d'optimisation multidisciplinaire,
- l'emploi d'algorithmes d'optimisation plus performants en terme d'exploration afin de trouver de meilleures solutions aux problèmes aérodynamiques (optimum global).

Les surfaces de réponse ont semblé pouvoir répondre à ces deux attentes.

En optimisation multidisciplinaire, elles permettent d'effectuer le couplage entre disciplines très facilement. Après avoir construit une surface de réponse pour chacune des disciplines, il est possible de substituer les codes de calcul experts, difficiles à manipuler, par leurs modèles réduits. Ces surfaces de réponse, considérées fiables sur tout le domaine, sont appelées surfaces de réponse globales. Elles permettent d'éviter la mise en place d'un évaluateur multidisciplinaire pour calculer les fonctions objectifs de chacune des disciplines ainsi que leurs gradients. Cet évaluateur multidisciplinaire nécessiterait le couplage de plusieurs codes experts, mais cela n'est pas adapté au travail en industrie où les différentes équipes expertes doivent pouvoir travailler indépendamment.

En optimisation de formes aérodynamiques, il n'est pas envisageable de construire une surface de réponse globale pour des raisons de coût de calcul. Etant donné que le nombre de variables de forme est important, le nombre de points de construction devrait lui aussi être important. Or, chacun de ces points nécessite une simulation numérique coûteuse en temps de calcul. Il faut donc considérer la surface de réponse fiable seulement en ses points de construction (surface de réponse locale). L'emploi de surfaces de réponse locales au sein d'un processus itératif peut permettre de limiter ce coût tout en garantissant de meilleures performances en terme d'exploration.

Avant de montrer l'intérêt de ces surfaces de réponse globales et locales à travers deux types d'applications (multidisciplinaire en III et aérodynamique en IV), il a paru nécessaire de définir leur bon usage.

La formulation retenue pour construire les surfaces de réponse est le Krigeage.

Une étude comparative sur des cas tests aérodynamiques simples a permis de mieux comprendre l'influence de l'échantillonnage et du nombre de variables sur la précision du Krigeage. Des critères d'échantillonnage séquentiel par raffinement au maximum de différentes erreurs prédites ont été comparés à des méthodes d'échantillonnage classiques de type remplissage d'espace. Alors que le raffinement d'échantillonnage au maximum d'erreur exacte amène un gain de précision significatif, ce gain n'est pas

observable lorsqu'une erreur prédite est utilisée. Les différentes méthodes de distribution sont toutes à peu près équivalentes et la méthode d'échantillonnage retenue est la méthode LHS de remplissage uniforme d'espace. Il semble donc que le seul moyen d'améliorer la précision d'une surface de réponse est d'ajouter de l'information sur la fonction en augmentant la taille de la base de données d'échantillons. Il est impossible d'établir une relation générale indiquant *a priori* la taille d'échantillonnage nécessaire pour avoir une surface de réponse globale en fonction de la dimension du problème. Un domaine de validité a tout de même pu être défini à partir de règles empiriques. Du fait de la malédiction dimensionnelle, il est recommandé d'utiliser une taille d'échantillonnage croissant exponentiellement avec le nombre de variables pour construire une surface de réponse globale.

Comme l'outil d'optimisation est capable de calculer le gradient de la fonction au même coût de calcul qu'une évaluation de fonction, il a paru intéressant de tirer parti de cette information pour construire la surface de réponse. Pour cela, la méthode de Cokrigeage a été employée. Elle permet d'obtenir une surface de réponse plus précise notamment lorsque le nombre de variables est grand. Toutefois, du fait de la quantité d'information à interpoler, le coût de calcul du Cokrigeage est inacceptable dès lors que la dimension dépasse dix. Pour limiter ce coût, une nouvelle méthode de Cokrigeage indirect tronqué en échantillon a été développée.

La première application présentée traitait de l'utilisation de surfaces de réponse globales pour l'optimisation multidisciplinaire. Le problème fluide/structure considéré n'a pas fait intervenir de couplage au niveau des équations d'état mais seulement au niveau des variables de forme. Il a été résolu par une approche biniveau de type BLISS. Au haut niveau, les fonctions objectifs traînée et masse ont été modélisées sur le domaine des variables de formes couplées grâce au Krigeage. Le nombre de variables couplées étant faible (trois variables), la taille d'échantillonnage a été jugée suffisamment importante pour substituer ces fonctions par des surfaces de réponse considérées globales. Au bas niveau, les codes de calcul experts, coûteux en temps de calcul, ont été employés en totale indépendance. Ils utilisaient des variables de forme expertes dont l'influence restait limitée à une seule discipline. Le processus BLISS est habituellement utilisé dans le cadre de la conception avant-projet où les codes de calcul sont rapides. Ce cas test a permis de vérifier son applicabilité à la conception préliminaire.

Toutefois, deux faiblesses ont été mises en lumière. En premier lieu, il est apparu que ce problème d'optimisation bi-objectif était dominé par la fonction objectif aérodynamique. Il a été résolu en utilisant une méthode de combinaison des objectifs grâce à un coefficient d'équivalence masse/traînée déterminé *a priori*. Ce coefficient d'équivalence entre objectifs a été obtenu par considération physique (mise à l'échelle en valeur absolue). En traçant un front de Pareto *a posteriori* et en calculant un coefficient d'équivalence par la méthode de mise à l'échelle des objectifs par leur variation admissible, il s'est avéré que ce coefficient était douze fois trop faible. En second lieu, une validation de l'optimum obtenu a été entreprise à partir de l'erreur prédite du Krigeage. L'optimum multidisciplinaire, situé loin des points de construction des surfaces de réponse globales, s'est révélé ne pas être robuste. En effet, pour limiter le coût de calcul, le nombre d'échantillons a été sous-estimé par rapport aux recommandations. De ce fait, les surfaces de réponse globales sont imprécises. Finalement, le processus biniveau semble adapté au cas test envisagé à condition de mettre en œuvre un processus itératif pour raffiner l'échantillonnage au niveau de la solution obtenue.

Un tel processus de raffinement itératif d'échantillonnage a été mis en place dans le cadre de l'optimisation de formes aérodynamiques (monodisciplinaire). Cela a mené au développement d'un nouvel algorithme d'optimisation à base de surfaces de réponse locales. Cet algorithme, décliné en deux versions utilisant soit le Krigeage soit le Cokrigeage indirect tronqué en échantillon, a prouvé son efficacité par comparaison à l'algorithme DOT utilisant une méthode de descente par gradient. Les trois cas tests exposés ont été choisis pour représenter les problèmes de minimisation de traînée traités par les concepteurs de formes aérodynamiques. A un surcoût modéré, ces nouvelles méthodes permettent

d'atteindre une meilleure solution car elles sont plus performantes en terme d'exploration. Alors que la performance de l'algorithme à base de Krigeage se dégrade en grande dimension, l'algorithme à base de Cokrigeage a été capable de converger aussi rapidement que l'algorithme DOT tout en atteignant une meilleure solution sur le cas à 48 variables sur un avion complet. Cette performance est en partie attribuable à l'évaluation en parallèle de plusieurs échantillons par itération. Elle s'explique aussi par le fait que les surfaces de réponse exploitent toute l'information connue sur la fonction, alors que l'algorithme de descente par gradient se concentre sur une information locale. Au cours de la convergence de l'optimisation, les différentes évaluations de la fonction objectif peuvent être regroupées pour former une base de données. Vis-à-vis de cette base de données, la méthode de descente par gradient filtre l'information et n'utilise qu'une information locale (les dernières itérations). Au contraire, la surface de réponse est un moyen d'interpréter sans perte l'information contenue dans cette base de données.

Perspectives

Finalement, les surfaces de réponse ont permis de mieux comprendre et d'analyser les spécificités des problèmes d'optimisation en aérodynamique. Elles ont été utilisées dans des contextes différents (aérodynamique, structure, faible/grande dimension, globale/locale), ce qui démontre leur flexibilité. Elles semblent un outil essentiel à la compréhension de fonctions complexes.

Les nombreux cas tests étudiés ont permis d'identifier quelques règles pour mieux définir les problèmes d'optimisation. Par exemple, la définition d'un trop grand nombre de bosses par section s'est avérée en pratique équivalente à un problème défini à partir de moins de bosses mais de plus grande amplitude. Aussi, alors qu'avec l'algorithme d'optimisation locale DOT il était souvent possible de minimiser la traînée efficacement sans pour autant considérer de contraintes sur la portance, cela ne semble plus le cas lorsque l'optimiseur recherche l'optimum global.

Concrètement, les surfaces de réponse ont apporté une nouvelle fonctionnalité à l'outil d'optimisation de formes aérodynamiques à travers deux nouveaux algorithmes d'optimisation. Ces algorithmes peuvent bénéficier de deux améliorations. Premièrement, la recherche du minimum global sur la surface de réponse a été effectuée selon un processus nécessitant de trop nombreuses évaluations. Il faudrait donc envisager le couplage des surfaces de réponse avec un algorithme d'optimisation globale efficace. Par exemple, avec l'algorithme génétique NSGA2⁸. En outre, ce couplage pourrait être réalisé de manière à ce que l'algorithme génétique multi-objectif puisse s'appuyer sur plusieurs surfaces de réponse modélisant chacune des fonctions objectifs et contraintes. Deuxièmement, étant donné les bonnes performances de l'algorithme d'optimisation basé sur le Cokrigeage indirect tronqué en échantillon par rapport à celui basé sur le Krigeage, il pourrait être envisagé de paralléliser le code de Cokrigeage afin de pouvoir l'appliquer dans sa formulation complète. De plus, cela permettrait de résoudre des problèmes à plus de 100 variables.

Lorsque la paramétrisation est définie directement sur la maquette numérique (la forme CAO⁹) plutôt que sur le maillage, il peut être difficile de calculer le terme de sensibilité du maillage par rapport aux variables de forme nécessaire pour le calcul du gradient de la fonction objectif. Dans ce cas, il faut utiliser un algorithme d'optimisation d'ordre 0, sans gradient. L'optimiseur basé sur le Krigeage, très efficace en faible dimension, peut donc être employé pour ce type de processus d'optimisation. Comme il est beaucoup plus facile de paramétrer une maquette numérique plutôt qu'un maillage multibloc, ce type de processus d'optimisation est de plus en plus courant.

Dans un contexte d'optimisation multidisciplinaire, les surfaces de réponse ont permis de coupler efficacement deux disciplines. Grâce à l'expérience acquise sur ce cas test multidisciplinaire simple, un cas plus complexe faisant intervenir un double couplage géométrique et physique pourrait être résolu

⁸Nondominated Sorting Genetic Algorithm 2 de Kalyanmoy Deb.

⁹Conception Assistée par Ordinateur

selon un processus biniveau itératif. Dans ce cadre, un code de Krigage parallélisé permettrait de construire les modèles réduits de champs surfaciques des variables d'état.

La mise en place de l'optimiseur par surfaces de réponse a impliqué le développement d'une architecture totalement nouvelle de l'outil d'optimisation. Cette architecture a été pensée de manière à pouvoir s'adapter à tous types d'algorithmes (avec ou sans gradients, évaluation de plusieurs formes par itération, etc.). En outre, l'interface définie permet aussi de mettre en place très facilement de nouveaux évaluateurs. Cela permettra d'une part de développer facilement de nouvelles méthodes d'optimisation et d'autre part d'appliquer les optimiseurs disponibles à une grande variété de problèmes.

Bibliographie

- [1] *Pilot's Handbook of Aeronautical Knowledge*, chapter 1-4. Number FAA-H-8083-25. 2003.
- [2] N. M. Alexandrov, J. E. Dennis, R. M. Lewis, and V. Torczon. A trust region framework for managing the use of approximation models in optimization. Technical Report TR-97-50, ICASE, 1997.
- [3] N. M. Alexandrov and S. Kodiyalam. Initial results of an MDO method evaluation study. In *7th Symposium on Multidisciplinary Analysis and Optimization*, number AIAA-98-4884. AIAA/USAF/NASA/ISSMO, 1998.
- [4] T. R. Altus. A response surface methodology for bi-level integrated system synthesis (BLISS). Technical Report 211652, NASA, 2002.
- [5] G. Alvarez. Can we make genetic algorithms work in high-dimensionality problems? Stanford Exploration Project 112, Stanford, 2002.
- [6] S. Amant. *Calcul et Décomposition de la Traînée Aérodynamique des Avions de Transport à Partir de Calculs Numériques et d'Essais en Soufflerie*. PhD thesis, SUPAERO, 10, avenue Édouard-Belin, BP 54032, 31055 Toulouse cedex 4, France, 2002.
- [7] C. Audet, J. E. Dennis, and D. W. Moore. A surrogate-model-based method for constrained optimization. In *8th Symposium on Multidisciplinary Analysis and Optimization*, number AIAA 2000-4891. AIAA/ISSMO, 2000.
- [8] A. Barthet. *Amélioration de la prévision des coefficients aérodynamiques autour de configurations portantes par méthode adjointe*. PhD thesis, INPT, 6, allée Emile Monso - ZAC du Palays, BP 34038, 31029 Toulouse cedex 4, France, 2007.
- [9] B. J. C. Baxter. *The Interpolation Theory Of Radial Basis Functions*. PhD thesis, Cambridge University, UK, 1992.
- [10] R. Bellman. *Adaptive Control Processes : A Guided Tour*. Princeton University Press, 1961.
- [11] A. J. Booker, E. J. Cramer, P. D. Frank, J. M. Gablonsky, and J. E. Dennis. MoVars : Multidisciplinary optimization via adaptive response surfaces. In *48th Structures, Structural Dynamics, and Materials Conference*. AIAA/ASME/ASCE/AHS/ASC, 2007.
- [12] A. J. Booker, J. E. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. Technical Report TR-98-47, Center for Research on Parallel Computation, 1998.
- [13] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [14] R. D. Braun. *Collaborative Architecture for Large-Scale Distributed Design*. PhD thesis, Stanford University, Stanford, CA, USA, 1996.

- [15] O. Brodersen, M. Rakowitz, S. Amant, P. Larrieu, D. Destarac, and M. Sutcliffe. Airbus, Onera, and DLR results from the 2nd AIAA drag prediction workshop. In *42th Aerospace Science Meeting & Exhibit*, number AIAA-04-0391. AIAA, 2005.
- [16] N. F. Brown and J. R. Olds. Evaluation of multidisciplinary optimization (MDO) techniques applied to a reusable launch vehicle. In *43th Aerospace Science Meeting & Exhibit*, number AIAA-05-707. AIAA, 2005.
- [17] L. Cambier and J.-P. Veillot. Status of the *elsA* CFD software for flow simulation and multi-disciplinary applications. In *46th Aerospace Science Meeting & Exhibit*, number AIAA-08-664. AIAA, 2008.
- [18] J. R. Carlson and M. Lamb. Integration effects of pylon geometry and rearward mounted nacelles for a high-wing transport. In *23rd Joint Propulsion Conference*, number AIAA-87-1920. AIAA/ASME/ASAE/SAE, 1987.
- [19] G. Carpentieri, M. J. L. van Tooren, and B. Koren. Aerodynamic shape optimization by means of sequential linear programming techniques. In *European Conference on Computational Fluid Dynamics*. ECCOMAS, 2006.
- [20] G. Carrier. Multi-disciplinary optimisation of a supersonic transport aircraft wing planform. In *European Conference on Computational Fluid Dynamics*. ECCOMAS, 2004.
- [21] S. Choi, J. J. Alonso, S. Kim, and I. M. Kroo. Two-level multi-fidelity design optimization studies for supersonic jets. In *43rd Aerospace Science Meeting & Exhibit*, number AIAA-05-0531. AIAA, 2005.
- [22] H.-S. Chung and J. J. Alonso. Design of a low-boom supersonic business jet using cokriging approximation models. In *9th Symposium on Multidisciplinary Analysis and Optimization*, number AIAA-02-5598. AIAA/ISSMO, 2002.
- [23] H.-S. Chung and J. J. Alonso. Using gradients to construct cokriging approximation models for high dimensional design optimization problems. In *40th Aerospace Science Meeting & Exhibit*, number AIAA-02-0317. AIAA, 2002.
- [24] H.-S. Chung and J. J. Alonso. Multiobjective optimization using approximation model-based genetic-algorithms. In *10th Symposium on Multidisciplinary Analysis and Optimization*, number AIAA-04-4325. AIAA/ISSMO, 2004.
- [25] D. Cox and S. John. SDO : A statistical method for global optimization. In *Multidisciplinary design optimization (Hampton, VA, 1995)*, pages 315–329. SIAM, Philadelphia, PA, 1997.
- [26] E. J. Cramer, J. E. Dennis, P. D. Frank, R. M. Lewis, and G. R. Shubin. Problem formulation for multidisciplinary optimization. *SIAM J. Optimization*, 4(4) :754–776, 1994.
- [27] E. J. Cramer, V. X. Du, and J. M. Gablonsky. Multi-objective optimization for complex computer simulations. In *44th Aerospace Science Meeting & Exhibit*, number AIAA-06-729. AIAA, 2006.
- [28] 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, 1996.
- [29] I. Das and J. E. Dennis. Normal-Boundary Intersection : a new method for generating Pareto optimal points in multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3) :631–657, 1998.
- [30] J. Delbove. *Contribution aux outils de simulation aéroélastique des aéronefs : Prédiction du flottement et déformation statique des voilures*. PhD thesis, SUPAERO, 10, avenue Édouard-Belin, BP 54032, 31055 Toulouse cedex 4, France, 2005.
- [31] D. Destarac and J. van der Vooren. Drag/thrust analysis of jet-propelled transonic transport aircraft ; definition of physical drag components. *Aerospace Science and Technology*, 8 :545–556, 2004.

-
- [32] J. Dongarra and F. Sullivan. Guest editors' introduction : The top 10 algorithms. *Computing in Science and Engineering*, 02(1) :22–23, 2000.
- [33] D. L. Donoho. High-dimensional data analysis : The curses and blessings of dimensionality. In *Math Challenges of the 21st Century*. American Math. Society, 2000. Department of Statistics, Stanford University.
- [34] F. Duchaine. *Optimisation de Forme Multi-Objectif sur Machines Parallèles avec Méta-Modèles et Coupleurs. Application aux Chambres de Combustion Aéronautiques*. PhD thesis, INPT, 6, allée Emile Monso - ZAC du Palays, BP 34038, 31029 Toulouse cedex 4, France, 2007.
- [35] B. Epstein, S. Peigin, H. Harrison, A. Jameson, D. Roman, and J. Vassberg. Comparative study of 3D wing drag minimization by different optimization techniques. In *46th Aerospace Science Meeting & Exhibit*, number AIAA 2008-326. AIAA, 2008.
- [36] AIAA Technical Committee for MDO. Current state of the art in multidisciplinary design optimization. *AIAA White Paper*, 1991.
- [37] M. Fujino and Y. Kawamura. Wave-drag characteristics of an over-the-wing nacelle business-jet configuration. *Journal of Aircraft*, 40(6) :1177–1184, 2003.
- [38] S. E. Gano, J. E. Renaud, J. D. Martin, and T. W. Simpson. Update strategies for kriging models for use in variable fidelity optimization. In *46th Structures, Structural Dynamics, and Materials Conference*, number AIAA-05-2057. AIAA/ASME/ASCE/AHS/ASC, 2005.
- [39] R. Glowinski and O. Pironneau. Towards the computation of minimum drag profiles in viscous laminar flow. *Appl. Math. Modelling*, 1 :58–66, 1976.
- [40] J. L. Godard, H. Hoheisel, C. C. Rossow, and V. Schmitt. Investigation of interference effects for different engine positions on a transport aircraft configuration. In Heinz Hoheisel, editor, *Aspects of Engine-Airframe Integration for Transport Aircraft*. DLR, 1996.
- [41] D. Goldberg. Sizing populations for serial and parallel genetic algorithms. In *Third International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers, 1989.
- [42] N Gould, D. Orban, and P. Toint. CUTer (and SifDec), a constrained and unconstrained testing environment, revisited. Technical Report TR/PA/01/04, CERFACS, 2004.
- [43] R. M. Hicks and P. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15 :407–412, 1978.
- [44] R. M. Hicks, E. M. Murman, and G. N. Vanderplaats. An assessment of airfoil design by numerical optimization. Technical Report TM X-3092, NASA, 1974.
- [45] R. M. Hicks and G. N. Vanderplaats. Application of numerical optimization to the design of supercritical airfoils without drag-creep. In Society of Automotive Engineers, editor, *Business Aircraft Meeting*. Society of Automotive Engineers, 1977.
- [46] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer, 1993. Two volumes.
- [47] C. Hirsch. *Numerical and Computation of Internal and External Flows, Volume 1*. John Wiley & Sons, New York, NY, 1988.
- [48] C.-J. Huang, J. Galuski, and C. L. Bloebaum. Multi-objective Pareto concurrent subspace optimization for multidisciplinary design. *AIAA Journal*, 45(8) :1894–1906, 2007.
- [49] A. M. Ingraldi, D. A. Naik, and O. C. Pendergraft. Effect of pylon cross-sectional geometries on propulsion integration for a low wing transport. Technical Report Technical Paper 3333, NASA, 1993.
- [50] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3), 1988.

- [51] A. Jameson, S. Kim, S. Shankaran, and K. Leoviriyakit. Aerodynamic shape optimization : Exploring the limits of design. In *KSAS 1st International Sessions*. KSAS, 2003.
- [52] A. Jameson and J. C. Vassberg. Studies of alternative numerical optimization methods applied to the brachistochrone problem. In *OptiCON'99*, Newport Beach, CA, October 1999.
- [53] A. Jameson and J. C. Vassberg. Computational fluid dynamics for aerodynamic design : Its current and future impact. In *39th Aerospace Science Meeting & Exhibit*, number AIAA-01-0538. AIAA, 2001.
- [54] S. Jeong, M. Murayama, and K. Yamamoto. Efficient optimization design method using kriging model. *Journal of Aircraft*, 42(2) :413–419, 2005.
- [55] R. Jin, W. Chen, and T. W. Simpson. Comparative studies of metamodelling techniques under multiple modelling criteria. *Struct Multidisc Optim*, 23(1) :1–13, 2001.
- [56] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 9(1) :3–12, 2005.
- [57] Y. Jin, M. Olhofer, and B. Sendhoff. Managing approximate models in evolutionary aerodynamic design optimization. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 592–599. IEEE, 2001.
- [58] F. T. Johnson, E. N. Tinoco, and N. Jong Yu. Thirty years of development and applications of CFD at Boeing commercial airplanes, seattle. In *16th Computational Fluid Dynamics Conference*, number AIAA-03-3439. AIAA, 2003.
- [59] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13 :455–492, 1998.
- [60] J.-C. Jouhaud, P. Sagaut, M. Montagnac, and J. Laurenceau. A surrogate model based multi-disciplinary shape optimization method with application to a 2D subsonic airfoil. *Computers and Fluids*, 36(3) :520–529, 2007.
- [61] M. Kanazaki, K. Tanaka, S. Jeong, and K. Yamamoto. Multi-objective aerodynamic exploration of elements' setting for high-lift airfoil using kriging model. *Journal of Aircraft*, 44(3) :858–864, 2007.
- [62] A. J. Keane. Wing optimization using design of experiment, response surface, and data fusion methods. *Journal of Aircraft*, 40(4) :741–750, 2003.
- [63] A. J. Keane and P. B. Nair. *Computational Approaches for Aerospace Design : The Pursuit of Excellence*. John-Wiley and Sons, 2005.
- [64] M. S. Khurana, H. Winarto, and Sinha A. K. Airfoil geometry parameterization through shape optimizer and computational fluid dynamics. In *46th Aerospace Science Meeting & Exhibit*, number AIAA-08-295. AIAA, 2008.
- [65] S. Kim and H.-S. Chung. *Computational Science and Its Applications - ICCSA 2006*, chapter Multiobjective Optimization Using Adjoint Gradient Enhanced Approximation Models for Genetic Algorithms, pages 491–502. 2006.
- [66] S. Koc, Kim H.-J., and K. Nakahashi. Aerodynamic design optimization of wing-body configurations. In *43rd Aerospace Science Meeting & Exhibit*, number AIAA-05-0331. AIAA, 2005.
- [67] P. N. Koch, T. W. Simpson, J. K. Allen, and F. Mistree. Statistical approximations for multi-disciplinary design optimization : the problem of size. *Journal of Aircraft*, 36(1) :275–286, 1999. Special Issue on Multidisciplinary Design Optimization.
- [68] L. Kocis and W. J. Withen. Computational investigations of low-discrepancy sequences. *ACM Transactions on Mathematical Software*, 23(2) :155–294, 1997.

-
- [69] I. Kroo, S. Altus, R. Braun, P. Gage, and J. Sobieszczanski-Sobieski. Multidisciplinary optimization methods for aircraft preliminary design. In *5th Symposium on Multidisciplinary Analysis and Optimization*, number AIAA-94-4325-CP. AIAA/USAF/NASA/ISSMO, 1994.
- [70] I. M. Kroo. Distributed multidisciplinary design and collaborative optimization. In *lecture series on Optimization Methods & Tools for Multicriteria/Multidisciplinary Design*. VKI, 2004.
- [71] M. Laban. Aircraft drag and thrust analysis (AIRDATA). Technical Report NLR-TP-2000-473, NLR, 2000.
- [72] J. Laurenceau and M. Meaux. Comparison of gradient and response surface based optimization frameworks using adjoint method. In *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, number AIAA 2008-1889. AIAA, 2008.
- [73] J. Laurenceau and P. Sagaut. Building efficient response surfaces of aerodynamic functions with kriging and cokriging. *AIAA Journal*, 46(2) :498–507, 2008.
- [74] S. T. LeDoux, W. W. Herling, and G. J. Fatta. MDOPT - a multidisciplinary design optimization system using higher order analysis codes. In *10th Multidisciplinary Analysis and Optimization Conference*, number AIAA-04-4567. AIAA/ISSMO, 2004.
- [75] E. E. Lee and O. C. Pendergraft. Installation effects of long-duct pylon-mounted nacelles on a twin-jet transport model with swept supercritical wing. Technical Report Technical Paper 2457, NASA, 1985.
- [76] W. Liu. *Development of Gradient-Enhanced Kriging Approximations for Multidisciplinary Design Optimization*. PhD thesis, University of Notre Dame, Notre Dame, IN, USA, 2003.
- [77] B. Mantel, J. Périaux, B. Stoufflet, M. Sefrioui, J.-A. Désidéri, S. Lanteri, and N. Marco. Evolutionary computational methods for complex design in aerodynamics. In *36th Aerospace Science Meeting & Exhibit*, number AIAA-98-222. AIAA, 1998.
- [78] N. Marco, C. Godart, J.-A. Désidéri, B. Mantel, and J. Périaux. Optimisation aérodynamique par algorithmes génétiques hybrides : application à la réduction d'un critère de bang sonique. Technical Report 4884, INRIA, 2003.
- [79] K. V. Mardia and R.J. Marshall. Maximum likelihood estimation of models for residual covariance in spatial regression. *Biometrika*, 71(1) :135–146, 1984.
- [80] K. V. Mardia and A. J. Watkins. On multimodality of the likelihood in the spatial linear model. *Biometrika*, 2 :289–295, 1989.
- [81] J. R. R. A. Martins. *A Coupled-Adjoint Method for High-Fidelity Aero-Structural Optimization*. PhD thesis, Stanford University, Stanford, CA, USA, 2002.
- [82] M. Meaux, M. Cormery, and G. Voizard. Viscous aerodynamic shape optimization based on the discrete adjoint state for 3D industrial configurations. European Congress on Computational Methods in Applied Sciences and Engineering, Jul. 2004.
- [83] M. Meunier. Simulation and optimization of flow control strategies for novel high-lift configurations. In *25th Applied Aerodynamics Conference*, number AIAA-07-4276. AIAA, 2007.
- [84] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. Oxford Science Publications, 2001.
- [85] A. Le Moigne and N. Qin. Variable-fidelity aerodynamic optimization for turbulent flows using a discrete adjoint formulation. *AIAA Journal*, 42(7) :1281–1292, 2004.
- [86] M. D. Morris, T. J. Mitchell, and D. Ylvisaker. Bayesian design and analysis of computer experiments : Use of derivatives in surface prediction. *Technometrics*, 35(3) :243–255, 1993.
- [87] S. Mouton, J. Laurenceau, and G. Carrier. Aerodynamic and structural optimisation of power-plant integration under the wing of a transonic transport aircraft. In *42ème colloque d'Aérodynamique Appliquée*. AAAF, 2007.

- [88] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.
- [89] Y. S. Ong, P. B. Nair, and A. J. Keane. Evolutionary optimization of computationally expensive problems via surrogate modelling. *AIAA Journal*, 41(4) :687–696, 2003.
- [90] L. Oulladji, A. Janka, J.-A. Désidéri, and A. Dervieux. Optimisation aérodynamique par algorithmes génétiques hybrides : application à la réduction d'un critère de bang sonique. Technical Report 4884, INRIA, 2003.
- [91] S. Peigin and B. Epstein. Robust drag minimization of aerodynamic wings in engineering environment. *Journal of Aircraft*, 43(4) :1195–1204, 2006.
- [92] S. Peigin and B. Epstein. Multipoint aerodynamic design of wing-body configurations for minimum drag. *Journal of Aircraft*, 44(3) :971–980, 2007.
- [93] O. C. Pendergraft, A. M. Ingraldi, R. J. Re, and T. T. Kariya. Installation effects of wing-mounted turbofan nacelle-pylons on a 1/17-scale, twin-engine low wing transport model. Technical Report Technical Paper 3168, NASA, 1992.
- [94] J. Peter, M. Marcelet, S. Burguburu, and V. Pediroda. Comparison of surrogate models for the actual global optimization of a 2D turbomachinery flow. In *7th WSEAS International Conference on Simulation, Modelling and Optimization*. WSEAS, 2007.
- [95] J. Peter, C.-T. Pham, and F. Drullion. Contribution to discrete implicit gradient and discrete adjoint method for aerodynamic shape optimisation. In *European Conference on Computational Fluid Dynamics*. ECCOMAS, 2004.
- [96] C.-T. Pham. *Linéarisation du flux visqueux des équations de Navier-Stokes et de modèles de turbulence pour l'optimisation aérodynamique en turbomachines*. PhD thesis, ENSAM, 151 bd de l'hôpital, 75013, Paris, France, 2006.
- [97] S. Pierret, H. Kato, R. Coelho, and A. Merchant. Multi-objective and multi-disciplinary shape optimization. In *EUROGEN*, 2005.
- [98] S. Pierret, H. Kato, R. Filomeno Coelho, and A. Merchant. Aero-mechanical optimization method with direct cad access - application to counter rotating fan design. Number GT2006-90505. ASME Turbo Expo, 2006.
- [99] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4) :409–435, 1989.
- [100] M. J. Sasena. *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*. PhD thesis, University of Michigan, Department of Mechanical Engineering, Ann Arbor, Michigan, USA, 2002.
- [101] R. Sellar, S. Batill, and J. Renaud. Response surface based, concurrent subspace optimization for multidisciplinary system design. In *34th Aerospace Sciences Meeting and Exhibit*, number AIAA 96-0714. AIAA, 1996.
- [102] S. Shahpar, A. Polynkin, and V. Toropov. Large scale optimization of transonic axial compressor rotor blades. In *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, number AIAA 2008-2056. AIAA, 2008.
- [103] T. W. Simpson, J. J. Korte, T. M. Mauery, and F. Mistree. Comparison of response surface and kriging models for multidisciplinary design optimization. In *Symposium on Multidisciplinary Analysis and Optimization*, number AIAA-98-4755. AIAA/USAF/NASA/ISSMO, 1998.
- [104] T. W. Simpson, D. K. Lin, and W. Chen. Sampling strategies for computer experiments : Design and analysis. *Journal of Reliability and Applications*, 2 :209–240, 2001.
- [105] J. Sobieszcanski-Sobieski. Optimization by decomposition : A step from hierarchic to non-hierarchic systems. In *Second Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*, number Conference Publication 3031, Part 1. NASA/Air Force, 1988.

-
- [106] J. Sobieszczanski-Sobieski. Multidisciplinary optimization for engineering systems : Achievements and potential. Technical Memorandum 101566, NASA, 1989.
- [107] J. Sobieszczanski-Sobieski. Multidisciplinary design optimization : An emerging new engineering discipline. Technical Memorandum 107761, NASA, 1993.
- [108] J. Sobieszczanski-Sobieski. Bi-level integrated system synthesis (BLISS). Technical Memorandum 208715, NASA, 1998.
- [109] J. Sobieszczanski-Sobieski. Integrated system-of-systems synthesis (ISSS). In *11th Symposium on Multidisciplinary Analysis and Optimization*, number AIAA-06-7064. AIAA/ISSMO, 2006.
- [110] J. Sobieszczanski-Sobieski and Raphael T. Haftka. Multidisciplinary aerospace design optimization : Survey of recent developments. In *4th Aerospace Science Meeting & Exhibit*, number AIAA-96-0711. AIAA, 1996.
- [111] V. Torczon and M. W. Trosset. Using approximations to accelerate engineering design optimization. Technical Report TR-98-33, ICASE, Institute for Computer Applications in Science and Engineering, 1998.
- [112] S. Tursi. Transonic wing optimization combining genetic algorithm and neural network. In *21st Applied Aerodynamics Conference*, number AIAA 2003-3787. AIAA, 2003.
- [113] C. P. van Dam. Aircraft design and the importance of drag prediction. In *lecture series on CFD-Based Aircraft Drag Prediction and Reduction*. VKI, 2003.
- [114] G. N. Vanderplaats. *D.O.T. Users Manual*. Vanderplaats Research & Development, Inc, 1995.
- [115] G. N. Vanderplaats. CONMIN : A fortran program for constrained function minimization : User's manual. Technical Report TM-X-62282, NASA, 19973.
- [116] A. Vavalle and N. Qin. Iterative response surface based optimization scheme for transonic airfoil design. *Journal of Aircraft*, 44(2) :365–376, 2007.
- [117] L. Wang, D. Beeson, G. Wiggs, and M. Rayasam. A comparison of meta-modeling methods using practical industry requirements. In *47th Structures, Structural Dynamics, and Materials Conference*, number AIAA-06-1811. AIAA/ASME/ASCE/AHS/ASC, 2006.
- [118] J. J. Warnes and B. D. Ripley. Problems with likelihood estimation of covariance functions of spatial gaussian processes. *Biometrika*, 74(3) :640–642, 1987.
- [119] M. Widhalm, A. Ronzheimer, and M. Hepperle. Comparison between gradient-free and adjoint based aerodynamic optimization of a flying wing transport aircraft in the preliminary design. In *25th Applied Aerodynamics Conference*, number AIAA-07-4060. AIAA, 2007.
- [120] B. Wilson, D. J. Cappelleri, T. W. Simpson, and M. I. Frecker. Efficient Pareto frontier exploration using surrogate approximations. In *8th Symposium on Multidisciplinary Analysis and Optimization*, number AIAA 2000-4895. AIAA/ISSMO, 2000.
- [121] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1) :67–82, April 1997.
- [122] H.-Y. Wu, S. Yang, F. Liu, and H.-M. Tsai. Comparison of three geometric representations of airfoils for aerodynamic optimization. In *16th Computational Fluid Dynamics Conference*, number AIAA-03-4095. AIAA, 2003.
- [123] B. A. Wujek, J. E. Renaud, and S. Batill. Concurrent subspace optimization using design variables sharing in a distributed computing environment. *Concurrent Engineering*, 4(4) :361–377, 1996.

Critères de raffinement d'échantillonnage à pondération dynamique

Cette annexe compare différents critères de raffinement d'échantillonnage pour la recherche de minimum et introduit des critères de raffinement à pondération dynamique entre fonction et erreur. Au chapitre IV l'optimiseur par surfaces de réponse utilise une méthode multi-critère. A chaque itération de l'optimiseur, l'échantillonnage est raffiné aux positions indiquées par trois critères :

- le minimum de la fonction Y ,
- le minimum du critère LCB_1 ,
- le minimum du critère LCB_{-1} .

Ce processus de raffinement de référence va être comparé à 9 processus de raffinement monocritères. Le cas test utilisé est le profil RAE2822 à 6 variables variables tel que décrit au chapitre IV. Pour cette comparaison les surfaces de réponse sont construites par Krigeage (pas d'information sur le gradient) et utilisent la même base de données initiale. L'algorithme de référence à trois critères est noté « RS Kriging y lcb1 lcb-1 ».

A.1 Les critères employés

En plus des critères Y , EI , LCB_1 , LCB_{-1} , LCI_1 , LCI_{-1} présentés au chapitre II, trois critères dits dynamique sont employés. A partir d'un critère LCB, ils permettent une évolution automatique du coefficient de pondération entre fonction et erreur au cours de l'optimisation.

A.1.1 Critères à pondération dynamique entre fonction et erreur

Par rapport au critère LCB classique,

$$LCB_b(x) = Y(x) - b \cdot S(x), \quad (\text{A.1})$$

les critères à pondération dynamique entre fonction et erreur introduisent un coefficient de pondération dynamique noté c ,

$$LCB_b(x) = Y(x) - b \cdot c \cdot S(x). \quad (\text{A.2})$$

Ce coefficient c est recalculé à chaque itération de l'optimiseur. Trois algorithmes de détermination dynamique de ce coefficient ont été développés (dynamique, par incrément de confiance, par incrément de méfiance) tel que décrit ci-après.

A.1.1.1 Critère LCB dynamique (LCB_b $_{dyn}$)

A l'itération i , on dispose d'une surface de réponse, $Y_i(x)$, construite grâce à un échantillonnage de taille N_i et de son erreur prédite, $S_i(x)$. Admettons que cet échantillonnage soit raffiné en un seul point ($N_{i+1} = N_i + 1$) noté M . A l'itération $i + 1$, il est possible de calculer *a posteriori* l'erreur exacte au point de raffinement,

$$S_i^{exa}(M) = Y_{i+1}(M) - Y_i(M). \quad (\text{A.3})$$

Il est alors possible de déterminer le coefficient d'échange, c^{dyn} , entre fonction et erreur prédite de manière à ce qu'au point de raffinement l'erreur prédite soit égale à l'erreur exacte,

$$c_{i+1}^{dyn} = \frac{Y_{i+1}(M) - Y_i(M)}{S_i^{exa}(M)}. \quad (\text{A.4})$$

Ce coefficient c_{i+1}^{dyn} n'est en réalité exacte qu'à l'itération i . Il n'est calculable qu'*a posteriori*, mais il sera considéré valable à l'itération $i + 1$.

Le coefficient est initialisé égal à un, $c_0^{dyn} = 1$. Lorsqu'un raffinement en plusieurs points est utilisé, le coefficient d'échange est obtenu par une simple moyenne des coefficients indiqués en chacun des points de raffinement. Finalement, une borne inférieure non nulle est appliquée à ce coefficient,

$$c_{i+1}^{dyn} = \max(0,01; c_{i+1}^{dyn}). \quad (\text{A.5})$$

En pratique, le coefficient dynamique c^{dyn} peut varier de manière importante entre deux itérations successives (figure A.6). Pour éviter cela, deux critères de pondération par incrément ont été développés.

A.1.1.2 Critère LCB par incrément de confiance (LCB_b $_{dyntr}$)

Au lieu de directement utiliser le coefficient dynamique décrit précédemment, la méthode par incrément de confiance applique un facteur multiplicatif d'expansion ou de réduction à partir d'un coefficient initial. Ce facteur est déterminé à partir de l'évolution lors des deux dernières itérations de l'optimiseur du coefficient de pondération dynamique c^{dyn} . Lorsque le coefficient dynamique croît, le facteur multiplicatif est supérieur à un ce qui correspond à une phase d'expansion. À l'inverse, lorsque le coefficient dynamique décroît, le facteur multiplicatif est inférieur à un ce qui correspond à une phase de réduction. L'algorithme est initialisé par un coefficient égal à un, $c_0^{dyntr} = 1$.

$$\left[\begin{array}{l} \text{Itération de la méthode par incrément de confiance :} \\ 1. \text{ Calculer le coefficient dynamique de référence} \\ \quad c_{i+1}^{dyn} = \frac{Y_{i+1}(M) - Y_i(M)}{S_i^{exa}(M)} \\ 2. \text{ Calculer le coefficient d'échange par incrément} \\ \quad \text{Si } |c_{i+1}^{dyn} - c_i^{dyn}| < \epsilon \text{ alors } c_{i+1}^{dyntr} = c_i^{dyntr} \\ \quad \text{Si } c_{i+1}^{dyn} - c_i^{dyn} \leq -\epsilon \text{ alors } c_{i+1}^{dyntr} = c_i^{dyntr} \cdot (1 - \delta) \quad \text{REDUCTION} \\ \quad \text{Sinon } c_{i+1}^{dyntr} = c_i^{dyntr} \cdot (1 + \delta) \quad \text{EXPANSION} \end{array} \right. \quad (\text{A.6})$$

En pratique le seuil d'activation vaut $\epsilon = 0,1$ et l'incrément vaut $\delta = 0,3$.

A.1.1.3 Critère LCB par incrément de méfiance (LCB_b $dyndtr$)

Cette méthode est similaire à la méthode par incrément de confiance, mis à part une inversion du critère d'expansion/réduction.

$$\left[\begin{array}{l}
 \text{Itération de la méthode par incrément de méfiance :} \\
 1. \text{ Calculer le coefficient dynamique de référence} \\
 \quad c_{i+1}^{dyn} = \frac{Y_{i+1}(M) - Y_i(M)}{S_i^{exp}(M)} \\
 2. \text{ Calculer le coefficient d'échange par incrément} \\
 \quad \text{Si } |c_{i+1}^{dyn} - c_i^{dyn}| < \epsilon \text{ alors } c_{i+1}^{dyndtr} = c_i^{dyndtr} \\
 \quad \text{Si } c_{i+1}^{dyn} - c_i^{dyn} \leq -\epsilon \text{ alors } c_{i+1}^{dyndtr} = c_i^{dyndtr} \cdot (1 + \delta) \quad \text{EXPANSION} \\
 \quad \text{Sinon } c_{i+1}^{dyndtr} = c_i^{dyndtr} \cdot (1 - \delta) \quad \text{REDUCTION}
 \end{array} \right. \quad (A.7)$$

A.2 Résultats

Les historiques de convergence présentés en figure A.1 permettent tout d'abord de constater que les optimisations par critère dynamique (LCB_b dyn) et par incrément de confiance (LCB_b $dyntr$) ne parviennent pas à améliorer la fonction. Après la première itération, qui correspond à la phase de remplissage d'espace quasi-aléatoire, la fonction n'est pas réduite alors que plusieurs points sont évalués. Cela s'explique par un excès d'exploration tel que constaté sur la figure A.4.

Il apparaît aussi clairement que les coefficients de pondération d'erreur négatifs ($b < 0$) impliquent une convergence prématuré de l'optimisation par défaut d'exploration. En effet, la méthode LCI_{-1} converge après seulement 5 itérations et la méthode LCB_{-1} converge immédiatement car le premier raffinement proposé correspond à un point appartenant à l'échantillonnage initial. Ainsi, la performance de la méthode LCI_{-1} en exploitation (figure A.3) s'explique surtout par ce très faible nombre d'itérations avant convergence.

L'amélioration obtenue (figures A.1 et A.2) est équivalente pour les critères Y , EI , LCB_1 , LCI_1 et LCB_1 $dyndtr$. Comme chacun de ces critères converge après environ 50 itérations, ils ont des performances semblables en terme d'exploitation (figure A.3). Comparés au processus multi-critère de référence (Y LCB_1 LCB_{-1}), ils convergent plus rapidement mais vers une solution nettement moins bonne (plus de 10% d'écart en amélioration). Parmi ces cinq méthodes les critères LCB_1 et LCB_1 $dyndtr$ se démarquent par une meilleure performance en terme d'exploration (figure A.4). En terme d'exploration, le critère Y LCB_1 LCB_{-1} se situe au même niveau que les critères LCB_1 et LCB_1 $dyndtr$, tout en étant légèrement moins performant en terme d'exploitation.

Finalement, étant données les performances prometteuses du nouveau critère dynamique par incrément de méfiance (LCB_1 $dyndtr$), il a été incorporé au sein d'un processus de raffinement multi-critère en complément d'un raffinement au minimum de la fonction (critère noté Y LCB_1 $dyndtr$). L'historique de convergence de la figure A.5 permet de constater que ce critère parvient à une solution comparable à la solution du critère de référence en un nombre d'itérations inférieur.

L'évolution des critères de pondération dynamique (c^{dyn} , c^{dyntr} et c^{dyndtr}) au cours de l'optimisation en figure A.6 permet de constater une tendance du coefficient dynamique par incrément de méfiance (c^{dyndtr}) vers zero avant la convergence du processus. Cela permet de réduire le poids accordé à l'erreur prédite et donc de limiter l'exploration du domaine en fin d'optimisation ce qui facilite la convergence.

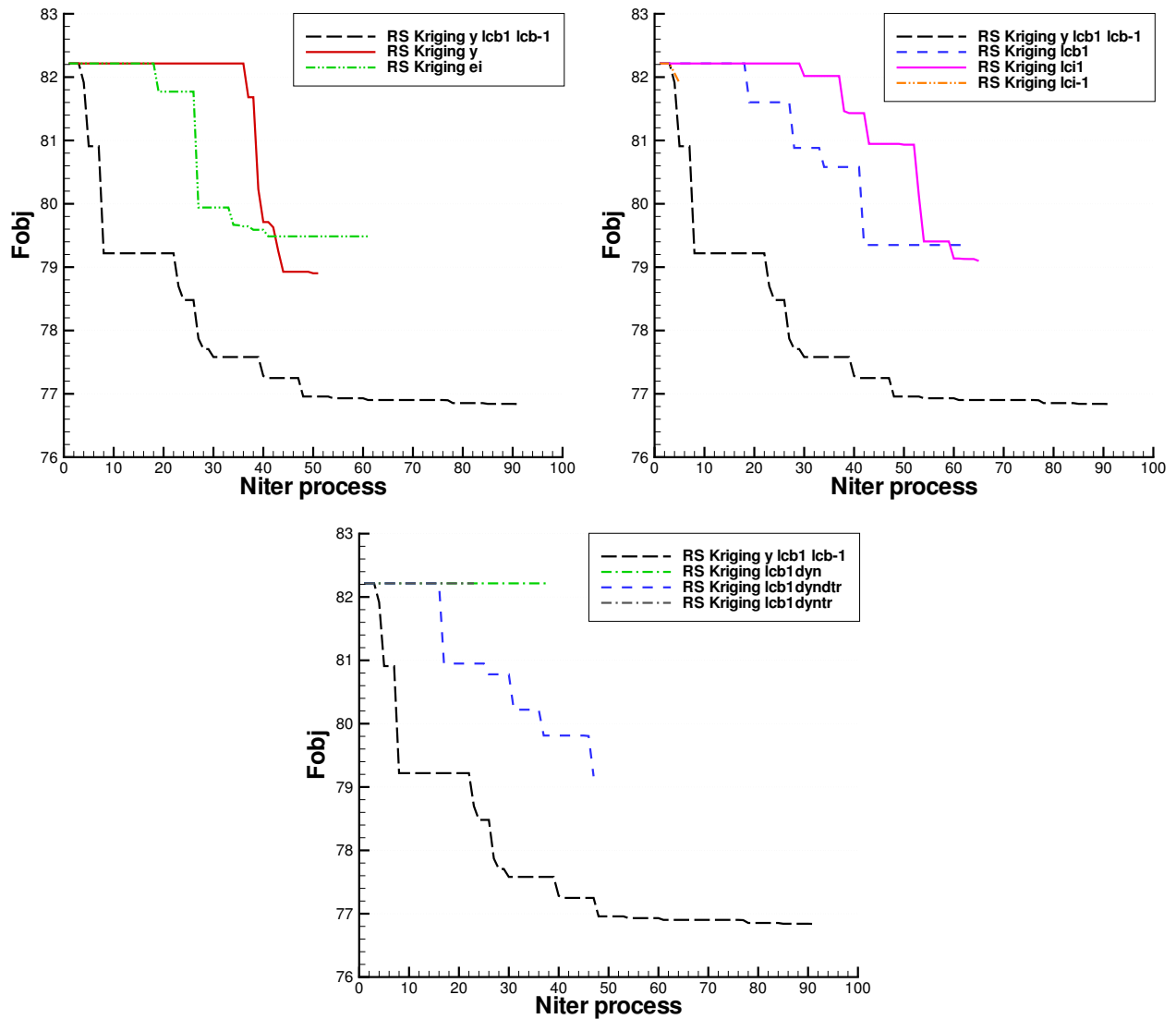


FIG. A.1: Historiques de convergence des 9 méthodes de raffinement d'échantillonnage sur le cas du profil RAE2822 à 6 variables.

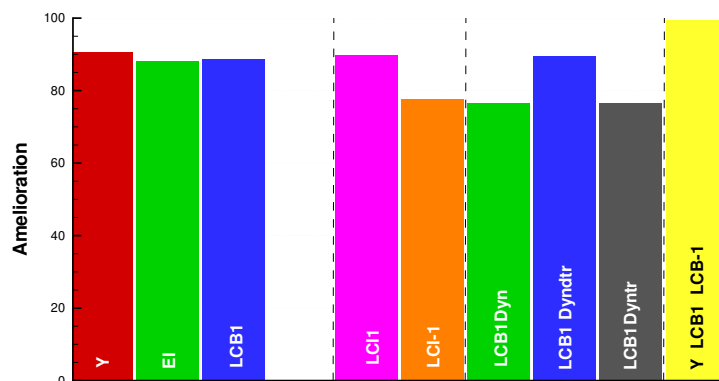


FIG. A.2: Performances en terme d'amélioration obtenues par les 9 méthodes de raffinement d'échantillonnage sur le cas profil RAE2822 à 6 variables.

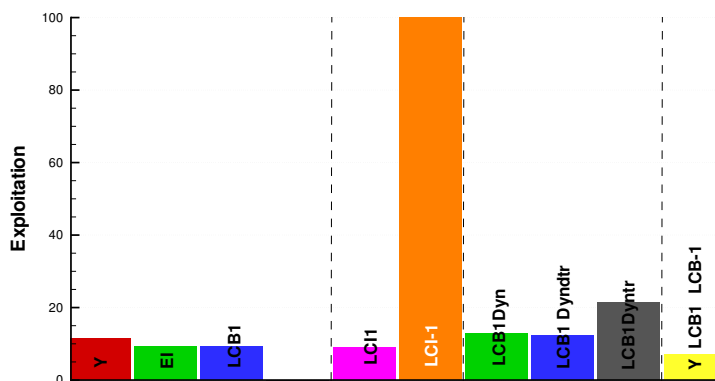


FIG. A.3: Performances en terme d'exploitation obtenues par les 9 méthodes de raffinement d'échantillonnage sur le cas profil RAE2822 à 6 variables.

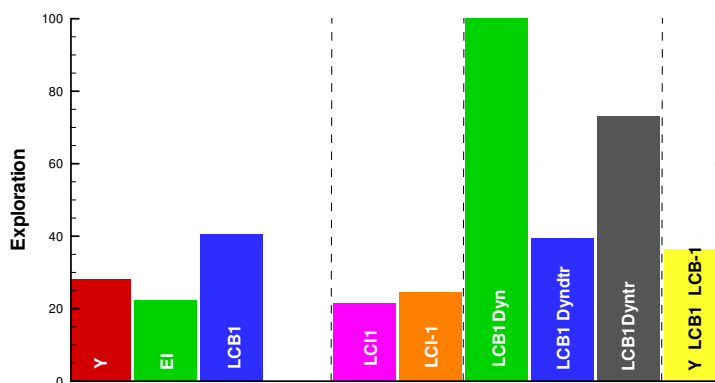


FIG. A.4: Performances en terme d'exploration obtenues par les 9 méthodes de raffinement d'échantillonnage sur le cas profil RAE2822 à 6 variables.

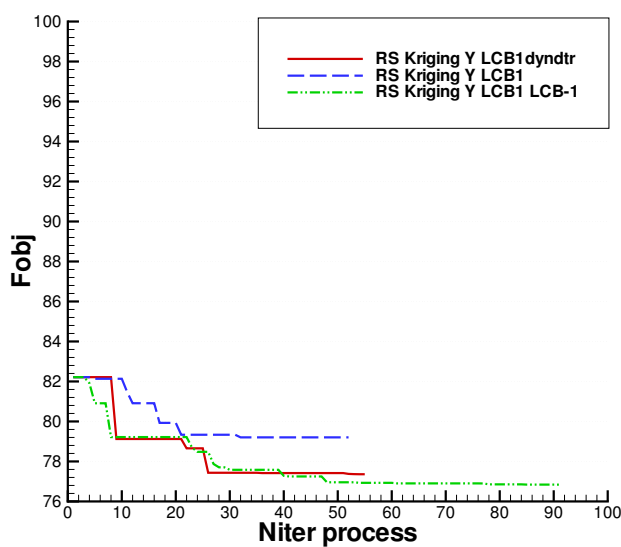


FIG. A.5: Historique de convergence de 3 méthodes de raffinement d'échantillonnage sur le cas du profil RAE2822 à 6 variables.

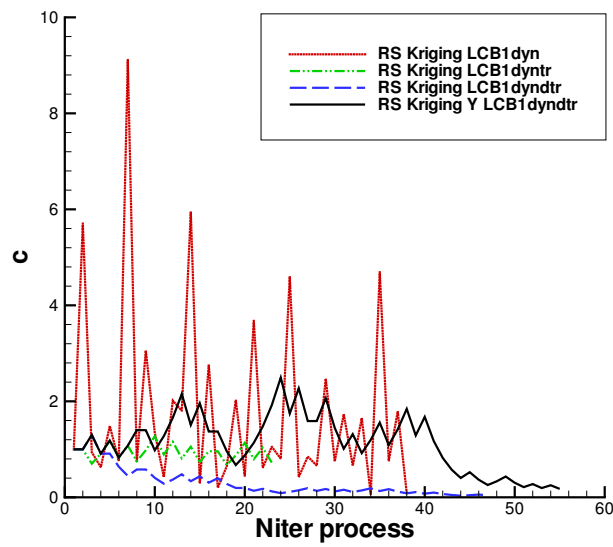


FIG. A.6: Evolution de critères de pondération dynamique au cours de l'optimisation, 4 processus différents, sur le cas du profil RAE2822 à 6 variables.

Etude de la fonction de Hicks-Henne

A.1 Fonction de Hicks-Henne

La fonction de déformation de Hicks-Henne a déjà été introduite en I.3.1 et l'influence des paramètres α pour la position et β pour l'expansion y est décrite. La variable x représente l'abscisse curviligne sur la forme. Elle est adimensionnée et varie entre zéro et un.

$$HH(x) = A \cdot \sin \left[\pi x^{-\frac{\ln[2]}{\ln[\alpha]}} \right]^\beta \quad (\text{A.1})$$

avec $x \in [0; 1]$, $A \in \mathfrak{R}$, $\beta \in [2; 10]$, $p \in]0; 1[$.

La largeur de la bosse de Hicks-Henne dépend en premier lieu de son exposant β , mais aussi de son pole α tel que l'on peut le constater sur la figure A.1.

La tangence avec la fonction de déformation nulle ($f(x) = 0$) est assurée en 0 et 1 dès lors que la puissance appliquée au sinus est strictement supérieure à un ($\beta = 1$). Afin d'obtenir des fonctions de déformation suffisamment lisses la borne inférieure pour cette puissance vaut $\beta = 2$. La fonction est \mathcal{C}^∞ sur l'ouvert $]0; 1[$ et $\mathcal{D}^{\beta-1}$ sur $]0; 1]$.

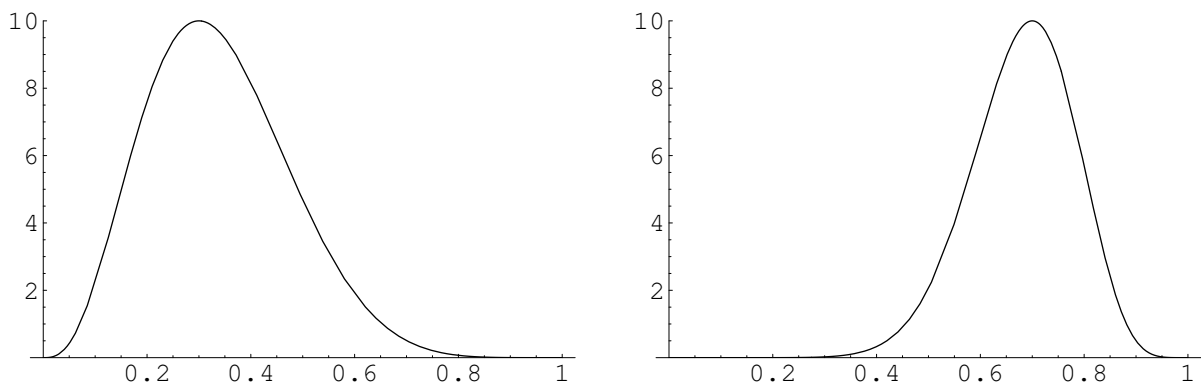


FIG. A.1: Bosses de Hicks-Henne. A gauche $A = 10$ $\alpha = 0,3$ et $\beta = 5$, à droite $A = 10$ $\alpha = 0,7$ et $\beta = 5$.

Pour calculer la sensibilité du maillage par rapport aux variables de forme, $\frac{dX}{d\alpha}$, la fonction de Hicks-Henne est dérivée selon les trois équations décrites ci-dessous.

A.2 Dérivée par rapport à l'amplitude

La dérivée de la fonction de Hicks-Henne par rapport à l'amplitude est toujours positive. Elle est comprise entre zéro et un.

$$\frac{dHH}{dA}(x) = \sin \left[\pi x^{-\frac{\ln[2]}{\ln[\alpha]}} \right]^\beta \quad (\text{A.2})$$

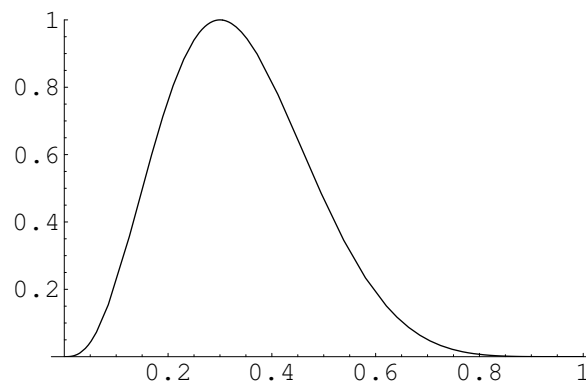


FIG. A.2: Bosse de Hicks-Henne, dérivée par rapport à l'amplitude ($A = 10$ $\alpha = 0,3$ et $\beta = 5$).

A.3 Dérivée par rapport au pole

La dérivée de la fonction de Hicks-Henne par rapport au pole est nulle aux bornes (influence du sinus). Elle admet aussi une racine entre ces deux bornes (influence du cosinus). Si l'amplitude est nulle, alors cette dérivée est nulle.

$$\frac{dHH}{d\alpha}(x) = \frac{A\pi x^{-\frac{\ln[2]}{\ln[\alpha]}} \beta \cos \left[\pi x^{-\frac{\ln[2]}{\ln[\alpha]}} \right] \ln[2] \ln[x] \sin \left[\pi x^{-\frac{\ln[2]}{\ln[\alpha]}} \right]^{-1+\beta}}{\alpha \ln[\alpha]^2} \quad (\text{A.3})$$

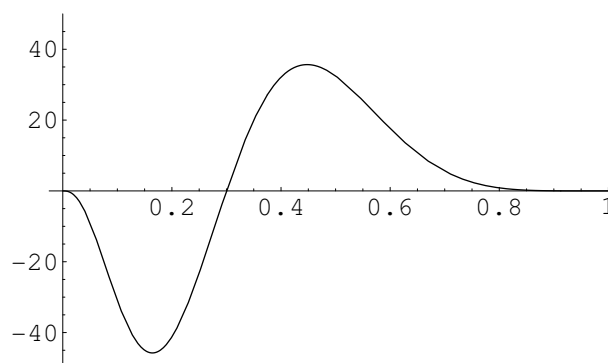


FIG. A.3: Bosse de Hicks-Henne, dérivée par rapport au pole ($A = 10$ $\alpha = 0,3$ et $\beta = 5$).

A.4 Dérivée par rapport à l'exposant

La dérivée de la fonction de Hicks-Henne par rapport à l'exposant est du même signe que l'amplitude. Si l'amplitude est nulle, alors cette dérivée est nulle.

$$\frac{dHH}{d\beta}(x) = A \ln \left[\sin \left[\pi x - \frac{\ln[2]}{\ln[\alpha]} \right] \right] \sin \left[\pi x - \frac{\ln[2]}{\ln[\alpha]} \right]^\beta \quad (\text{A.4})$$

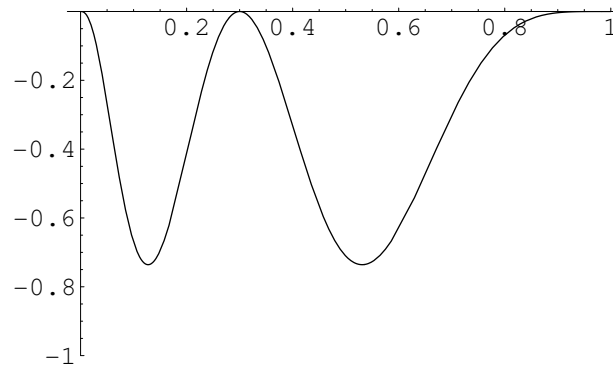


FIG. A.4: Bosse de Hicks-Henne, dérivée par rapport à l'exposant ($A = 10$ $\alpha = 0,3$ et $\beta = 5$).

Finalement, lorsque qu'un algorithme de descente par gradient est initialisé à partir d'un champ de déformation nul, seules les sensibilités de la fonction objectif par rapport aux amplitudes de bosses sont non nulles.

Résumé

Les simulations numériques RANS, utilisées en aérodynamique pour évaluer la performance d'une forme, sont coûteuses en temps de calcul. Comme une optimisation nécessite l'étude de plusieurs dizaines de formes différentes, ce coût limite le champ des problèmes envisageables. C'est donc en employant une technique efficace pour le calcul de gradients (méthode adjointe) et un algorithme d'optimisation par gradient (méthode quasi-Newton) que sont habituellement traités les problèmes de minimisation de traînée en conception à Airbus. Cet algorithme converge très rapidement, mais vers un optimum local.

Dans ce contexte, un nouvel optimiseur basé sur des surfaces de réponse construites par une méthode de Krigeage est proposé. Appliqué sur différents cas tests (profil, voilure, configuration complète), celui-ci semble effectivement apporter de nouvelles possibilités aux concepteurs : à un surcoût modéré, la solution obtenue est meilleure. De plus, cet optimiseur semble aussi capable de traiter des problèmes de grande dimension (48 variables) en interpolant le vecteur gradient aux points de construction du Krigeage.

En optimisation multidisciplinaire, les surfaces de réponse sont largement employées pour échanger facilement des données entre différentes disciplines. Ainsi, une approche d'optimisation bi-niveau avec couplage fluide/structure par surface de réponse est étudiée. L'application considérée traite de l'intégration d'une installation motrice (positionnement) sur un avion de transport civil.

Mots clés : optimisation, krigeage, cokrigeage, gradient, adjoint, conception, traînée, aérodynamique, multidisciplinaire.

Abstract

Aerodynamic shape optimization problems based on flow simulations solving RANS equations are computationally expensive. In order to keep this cost acceptable, optimization problems are usually solved using a gradient based optimization algorithm and an adjoint solver to efficiently compute the gradient of the objective function. This framework, implemented at Airbus, converges quickly but to a local optimum.

In this thesis, a response surface based optimizer using kriging model is described and applied to aerodynamic test cases. Compared to the gradient based optimizer, it achieves better function values at a moderate additional cost. Moreover, the use of gradient enhanced kriging (or cokriging model) enables to efficiently apply this new optimizer to high dimensional problems (48 design variables).

In a multidisciplinary context, a bilevel optimization framework was applied to solve a problem of fluid/structure powerplant integration. Using this framework, both disciplines considered were handled independently by discipline specific solvers and the coupling was achieved thanks to response surfaces.

Keywords : optimization, kriging, cokriging, gradient, adjoint, design, drag, aerodynamic, multidisciplinary.