

UNIVERSITÉ DE MONTRÉAL

OPTIMISATION À FIDÉLITÉ VARIABLE

XAVIER BERTRAND MARDUEL
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE MÉCANIQUE)
NOVEMBRE 2002



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-81711-3

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

OPTIMISATION À FIDÉLITÉ VARIABLE

présenté par : MARDUEL Xavier Bertrand

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. PELLETIER Dominique, Ph.D., président

M. TRÉPANIÉ Jean-Yves, Ph.D., membre et directeur de recherche

M. AUDET Charles, Ph.D., membre

REMERCIEMENTS

Je tiens à exprimer ma gratitude à l'égard de mon directeur de recherche, le professeur Jean-Yves Trépanier, pour ses conseils avisés. Je souhaite également le remercier pour m'avoir laissé une grande liberté dans mon travail, en soutenant mes initiatives par des aides humaines ou matérielles.

Je tiens à remercier tous les membres du CERCA et tout particulièrement Christophe Tribes, Julien Dompierre, Paul Labbé et Michel Béland du RQCHP pour m'avoir fait profiter de leur expérience dans le domaine de la recherche. Je remercie également le personnel administratif et technique du CERCA, efficace et chaleureux.

Je remercie enfin la Fondation de l'X et le département de génie mécanique pour m'avoir aidé par leur soutien financier à effectuer ma maîtrise à l'École Polytechnique de Montréal.

RÉSUMÉ

Ce mémoire se penche sur le problème de l'optimisation à fidélité variable, c'est-à-dire de l'optimisation par utilisation conjointe de deux résolveurs numériques de précisions et donc de coûts distincts et ce, dans le but de profiter de la fiabilité de l'un et de la rapidité de l'autre. Ce travail évalue l'opportunité d'une telle méthode sur l'optimisation aérodynamique d'un profil d'aile.

Un algorithme a été choisi pour sa rigueur mathématique, qui garantit la convergence vers un point stationnaire du problème haute fidélité, et pour son adaptation aux méthodes à gradient, qui sont efficaces pour notre problème. Nous le présentons et le modifions légèrement pour lui donner une architecture modulaire afin de faciliter la programmation et toute substitution de résolveur ou d'optimiseur.

Le programme est ensuite testé sur des fonctions analytiques et des améliorations sont apportées sur la méthode pour certains cas problématiques.

L'algorithme est finalement appliqué pour trois formulations du problème aérodynamique, chacune avec deux couples de résolveurs distincts. Nous constatons, sur ce cas pratique, l'inefficacité des améliorations validées au cours de l'étude analytique. Nous concluons néanmoins sur une réduction du nombre d'appels au résolveur haute fidélité par un facteur deux ou trois, comparativement à une optimisation directe sur ce même solveur.

Nous abordons enfin le manque de robustesse de l'algorithme, observé sur de nombreux cas pratiques, et désignons le bruit numérique comme responsable très probable grâce à une simulation analytique dans laquelle nous introduisons artificiellement du bruit.

Nous concluons enfin sur l'intérêt certain de la méthode en terme de réduction du temps de calcul et donnons un certain nombre de conseils pour améliorer sa robustesse. L'algorithme est donc prometteur mais il est encore nécessaire d'en prolonger l'étude.

ABSTRACT

This thesis deals with variable fidelity optimization which consists in using the advantages of high and low fidelity solvers in an optimization process. The high fidelity solver will provide solution reliability while the low fidelity solver will reduce the computational cost. This research discusses the feasibility and the benefit of such a method on a 2D airfoil optimization.

An algorithm is described which converges to a solution of the high fidelity problem and which is well adapted to gradient-based optimization suitable for our particular case. A modular architecture is proposed in order to facilitate implementation and any substitution of solver or optimizer.

The program is tested on analytical functions and improved for a few problematic cases.

Results are then presented for three kinds of 2D airfoil optimization, each based on two distinct pairs of solvers. Improvements achieved during the analytical study do not materialize in these cases. Nevertheless, the algorithm leads to a threefold savings in high fidelity solver calls, compared to a direct optimization using the high fidelity solver.

Finally, the observed robustness deficiency is tackled. Numerical noise is introduced in an analytical simulation and appears to cause many premature stops. Numerical noise observed with our real solvers is therefore suspected to cause the robustness problems encountered.

A real benefit can be obtained from this method and we conclude on future improvements to avoid premature stops. Results are promising and point towards further studies.

TABLE DES MATIÈRES

REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vii
TABLE DES MATIÈRES	ix
LISTE DES TABLEAUX	xiii
LISTE DES FIGURES	xvi
LISTE DES SIGLES ET ABRÉVIATIONS	xix
INTRODUCTION	1
CHAPITRE 1 PRÉSENTATION THÉORIQUE DE L'AMF ET DESCRIPTION DE L'ALGORITHME UTILISÉ	4
1.1 Méthode classique du rayon de confiance pour un problème sans contraintes	4
1.1.1 Présentation de l'algorithme	5
1.1.1.1 Principe de la méthode	5
1.1.1.2 Critères d'acceptation du pas	6
1.1.1.3 Critères de mise-à-jour du rayon de confiance	7
1.1.1.4 Algorithme de la méthode du rayon de confiance	8
1.1.2 Hypothèses et propriétés de convergence	9
1.1.2.1 Hypothèses sur la nature de la fonction coût	9
1.1.2.2 Hypothèses sur la fonction $q_k(x_k + s)$	9

1.1.2.3	Hypothèses sur la résolution du sous-problème d'optimisation	11
1.1.2.4	Propriétés de convergence	13
1.2	Application de la méthode du rayon de confiance à la fidélité variable	14
1.2.1	Introduction du modèle basse fidélité dans la méthode du rayon de confiance par l'intermédiaire d'une nouvelle approximation	15
1.2.2	Hypothèses sur la nouvelle approximation	15
1.2.3	Construction de la nouvelle approximation par la β -correction	16
1.3	Méthode du rayon de confiance appliquée à un problème d'optimisation avec contraintes	17
1.3.1	Lagrangien augmenté	18
1.3.1.1	Méthode classique du Lagrangien augmenté	18
1.3.1.2	Lagrangien augmenté associé à la méthode du rayon de confiance	20
1.3.2	Méthodes SQP	24
1.3.2.1	Principe de la méthode	24
1.3.2.2	Méthode SQP à rayon de confiance	26
1.3.2.3	Algorithme de la méthode SQP à rayon de confiance appliquée à la fidélité variable	30
1.3.3	Présentation de l'algorithme de l'AMF implanté	32
1.3.3.1	Algorithme de l'AMF	32
1.3.3.2	Schéma de l'AMF	34
CHAPITRE 2	ETUDE SUR DES CAS ANALYTIQUES ET AMÉLIORATION DE LA BÊTA-CORRECTION	37
2.1	Validation du comportement	38
2.1.1	Vérification de la convergence	38

2.1.2	Apport de la bêta correction avec constante et première intuition sur la correction additive	39
2.2	Problème de convergence et nouvelle correction	42
2.2.1	Mise en évidence du problème de convergence	43
2.2.2	Améliorations de la bêta correction	45
2.2.2.1	Bêta correction d'un ordre supérieur	45
2.2.2.2	Delta correction	46
2.2.2.3	Démonstration de l'équivalence entre une bêta correction avec une forte constante et une delta correction	48
2.3	Résultats analytiques obtenus avec les corrections améliorées	49
2.3.1	Fonctions x^4 et $(x - 0, 1)^4$	49
2.3.2	Fonction de Rosenbrock	52
2.3.3	Problème de Hock et Schittkowski numéro 100	54
2.4	Conclusions sur l'intérêt des nouvelles corrections	55
CHAPITRE 3 RÉSULTATS SUR UN CAS D'OPTIMISATION AÉRO-DYNAMIQUE DE PROFIL D'AILE		57
3.1	Description de l'expérience et de la formulation du problème d'optimisation	57
3.1.1	Paramétrisation	57
3.1.2	Résolveurs	59
3.1.2.1	BGK	59
3.1.2.2	NSC2KE	60
3.1.2.3	NSU2D	61
3.1.3	Optimiseur	62
3.1.4	Fonctions coût sélectionnées	62
3.1.4.1	Maximisation de la finesse de l'aile	62

3.1.4.2	Minimisation de la traînée à portance constante . . .	63
3.1.4.3	Optimisation avec courbe de C_p cible	64
3.2	Résultats expérimentaux	65
3.2.1	Maximisation de la finesse de l'aile	66
3.2.1.1	Résultats obtenus avec BGK comme modèle faible	66
3.2.1.2	Résultats obtenus avec NSC2KE comme modèle faible	71
3.2.2	Minimisation de la traînée sous contrainte de portance . . .	75
3.2.2.1	Résultats obtenus avec BGK comme modèle faible	75
3.2.2.2	Résultats obtenus avec NSC2KE comme modèle faible	77
3.2.3	Optimisation avec courbe de C_p cible	80
3.2.3.1	Résultats obtenus avec BGK comme modèle faible	80
3.2.3.2	Résultats obtenus avec NSC2KE comme modèle faible	82
3.2.4	Synthèse des résultats expérimentaux	85
CHAPITRE 4	ROBUSTESSE DE L'AMF	89
4.1	Hypothèses sur l'influence du bruit	89
4.1.1	Effet du bruit numérique du modèle fort	90
4.1.2	Effet du bruit numérique du modèle faible	91
4.2	Simulation du bruit numérique	93
4.2.1	Influence du pas de différenciation fini	95
4.2.2	Influence du bruit provenant du modèle fort	96
4.2.3	Influence du bruit provenant du modèle faible	97
4.2.4	Influence du bruit provenant des deux modèles	97
4.2.5	Influence du bruit sur l'ordre de la correction	97
4.3	Synthèse de l'étude de robustesse de l'AMF	98
CONCLUSION		100
RÉFÉRENCES		103

LISTE DES TABLEAUX

Tableau 2.1	Comparaison du nombre d'appels au modèle haute fidélité pour le problème Hock et Schittkowski numéro 5	41
Tableau 2.2	Comparaison du nombre d'appels au modèle haute fidélité pour une minimisation de x^4	51
Tableau 2.3	Comparaison du nombre d'appels au modèle haute fidélité pour la fonction de Rosenbrock	53
Tableau 2.4	Comparaison du nombre d'appels au modèle haute fidélité pour le problème Hock et Schittkowski numéro 100	55
Tableau 3.1	Coût de calcul de diverses stratégies et caractéristiques des solutions pour une maximisation de la finesse C_l/C_d en utilisant BGK comme modèle faible	66
Tableau 3.2	Nombre d'appels nécessaires à l'optimisation directe pour battre l'optimum de diverses stratégies d'AMF, pour une maximisation de la finesse C_l/C_d en utilisant BGK comme modèle faible.	68
Tableau 3.3	Coût de calcul de diverses stratégies et caractéristiques des solutions pour une maximisation de la finesse C_l/C_d en utilisant NSC2KE comme modèle faible	72

Tableau 3.4	Nombre d'appels nécessaires à l'optimisation directe pour battre l'optimum de diverses stratégies d'AMF, pour une maximisation de la finesse C_l/C_d en utilisant NSC2KE comme modèle faible.	72
Tableau 3.5	Coût de calcul de diverses stratégies et caractéristiques des solutions pour une minimisation de la traînée à portance constante en utilisant BGK comme modèle faible	75
Tableau 3.6	Nombre d'appels nécessaires à l'optimisation directe pour battre l'optimum de diverses stratégies d'AMF, pour une minimisation de la traînée à portance constante en utilisant BGK comme modèle faible.	76
Tableau 3.7	Coût de calcul de diverses stratégies et caractéristiques des solutions pour une minimisation de la traînée à portance constante en utilisant NSC2KE comme modèle faible	77
Tableau 3.8	Nombre d'appels nécessaires à l'optimisation directe pour battre l'optimum de diverses stratégies d'AMF, pour une minimisation de la traînée à portance constante en utilisant NSC2KE comme modèle faible.	79
Tableau 3.9	Coût de calcul et fonction coût finale de diverses stratégies pour une optimisation avec C_p cible en utilisant BGK comme modèle faible	80
Tableau 3.10	Nombre d'appels nécessaires à l'optimisation directe pour battre l'optimum de diverses stratégies d'AMF, pour une optimisation avec C_p cible en utilisant BGK comme modèle faible.	81

Tableau 3.11	Coût de calcul et fonction coût finale de diverses stratégies pour une optimisation avec C_p cible en utilisant NSC2KE comme modèle faible	83
Tableau 3.12	Nombre d'appels nécessaires à l'optimisation directe pour battre l'optimum de diverses stratégies d'AMF ou à l'AMF pour battre l'optimisation directe, pour une optimisation avec C_p cible en utilisant NSC2KE comme modèle faible. . .	83
Tableau 3.13	Tableau récapitulatif des expériences menées avec l'AMF . .	86
Tableau 4.1	Influence du bruit sur l'AMF pour le problème de fidélité variable basé sur la fonction de Rosenbrock	94

LISTE DES FIGURES

Figure 1.1	Schéma de l'AMF implanté	36
Figure 2.1	Modèle haute fidélité du problème HS5	40
Figure 2.2	Modèle basse fidélité du problème HS5	40
Figure 2.3	Modèle basse fidélité du problème HS5 après correction par la bêta correction sans constante	42
Figure 2.4	Modèle basse fidélité du problème HS5 après correction par la bêta correction avec constante	42
Figure 2.5	Modèle haute fidélité x^4 et modèle basse fidélité $(x - 0, 1)^4$.	43
Figure 2.6	Modèle faible $(x - 0, 1)^4$ corrigé par une bêta correction li- néaire à la première itération	44
Figure 2.7	Modèle faible $(x - 0, 1)^4$ corrigé par une bêta correction li- néaire à la seconde itération	44
Figure 2.8	Modèle faible $(x - 0, 1)^4$ corrigé par une bêta correction li- néaire à la troisième itération	44
Figure 2.9	Modèle basse fidélité du problème HS5 après correction par la delta correction	48
Figure 2.10	Modèle faible $(x - 0, 1)^4$ corrigé par une bêta correction qua- dratique à la troisième itération	50

Figure 2.11	Modèle faible $(x - 0, 1)^4$ corrigé par une delta correction cubique à la troisième itération	51
Figure 2.12	Courbes de niveau de la fonction de Rosenbrock	52
Figure 3.1	Représentation d'un profil RAE2822 par une NURBS déterminée par quinze points de contrôle	59
Figure 3.2	Maillage mixte développé pour le solveur NSC2KE	61
Figure 3.3	Distance entre deux courbes C_p évaluée par une surface	65
Figure 3.4	Courbes C_p des solutions de l'optimisation directe et de l'AMF utilisé avec BGK sur le problème de maximisation de C_l/C_d	70
Figure 3.5	Profils solutions de l'optimisation directe et de l'AMF utilisé avec BGK sur le problème de maximisation de C_l/C_d	70
Figure 3.6	Courbes C_p des solutions de l'optimisation directe et de l'AMF utilisé avec NSC2KE sur le problème de maximisation de C_l/C_d	74
Figure 3.7	Profils solutions de l'optimisation directe et de l'AMF utilisé avec NSC2KE sur le problème de maximisation de C_l/C_d	74
Figure 3.8	Courbes C_p des solutions de l'optimisation directe et de l'AMF utilisé avec BGK sur la minimisation de traînée à portance constante	77

Figure 3.9	Courbes C_p des solutions de l'optimisation directe et de l'AMF utilisé avec NSC2KE sur la minimisation de traînée à portance constante	80
Figure 3.10	Courbes C_p des solutions de l'optimisation directe et de l'AMF utilisé avec BGK sur l'optimisation avec C_p cible . . .	82
Figure 3.11	Courbes C_p des solutions de l'optimisation directe et de l'AMF utilisé avec NSC2KE sur l'optimisation avec C_p cible	84
Figure 4.1	Influence du bruit numérique du solveur haute fidélité sur l'AMF	91
Figure 4.2	Influence du bruit numérique du solveur basse fidélité sur l'AMF	92

LISTE DES SIGLES ET ABRÉVIATIONS

AMF	Approximation Management Framework ; architecture proposée par Alexandrov <i>et al.</i> (1997) pour l'optimisation à fidélité variable
BFGS	Broyden, Fletcher, Goldfarb et Shanno ; méthode d'approximation de la matrice hessienne pour l'optimisation par la méthode quasi-Newton
BGK	Bauer, Garabedian et Kron ; méthode de résolution des caractéristiques aérodynamiques de profils d'aile par la méthode potentielle
CERCA	Centre de Recherche en Calcul Appliqué
FCD	Fraction of Cauchy Decrease : fraction de la réduction prévue par le pas de Cauchy
FOD	Fraction of Optimal Decrease
HS	Problème Hock et Schittkowski
MDO	Multidisciplinary Design Optimization ; Optimisation sur des problèmes impliquant plusieurs disciplines couplées
NSC2KE	Navier-Stokes Compressible 2D k- ϵ : Solveur Volumes-Finis avec modèle de turbulence k- ϵ pour le calcul d'écoulements 2D sur un maillage non structuré, utilisé dans notre cas pour résoudre les équations d'Euler
NSU2D	Navier-Stokes Unstructured 2D ; Résolveur Navier-Stokes en 2D avec modélisation de la turbulence sur un maillage non structuré

NURBS	Non-Uniform Rational B-splines ; fonction paramétrique permettant notamment la représentation précise de profils avec un nombre restreint de variables de design
RQCHP	Réseau Québécois de Calcul Haute Performance
SAO	Sequential Approximate Optimization ; optimisation utilisant des approximations de la fonction coût

Caractères usuels

c	Longueur de la corde du profil
C_d	Coefficient de traînée
C_l	Coefficient de portance
C_p	Coefficient de pression à la surface de l'aile
$f(x)$	Fonction coût ou fonction objectif de l'optimisation
$f_{hi}(x)$	Valeur attribuée à la fonction coût par le solveur haute fidélité
$f_{lo}(x)$	Valeur attribuée à la fonction coût par le solveur basse fidélité
$g(x)$	Vecteur des contraintes d'inégalité
$g_{hi}(x)$	Valeur attribuée au vecteur des contraintes d'inégalité par le solveur haute fidélité
$g_{lo}(x)$	Valeur attribuée au vecteur des contraintes d'inégalité par le solveur basse fidélité
$h(x)$	Vecteur des contraintes d'égalité

$h_{hi}(x)$	Valeur attribuée au vecteur des contraintes d'égalité par le solveur haute fidélité
$h_{lo}(x)$	Valeur attribuée au vecteur des contraintes d'égalité par le solveur basse fidélité
H	Matrice des dérivées secondes de la fonction coût (Hessien)
K	Constante additive utilisée dans la formulation de la bêta correction
N	Nombre de variables de design
N_g	Nombre de contraintes d'inégalité
N_h	Nombre de contraintes d'égalité
x	Vecteur de design
X	Abscisse d'un point
Y	Ordonnée d'un point
W	Poids attribué à un point de contrôle dans une NURBS

Caractères grecs

$\beta(x)$	fonction bêta, définie par le quotient de la réponse du solveur haute fidélité par la réponse du solveur basse fidélité
β^1	bêta correction linéaire
β^2	bêta correction quadratique

β^3	béta correction cubique
β_0	béta correction sans constante additive
$\delta(x)$	delta correction, définie par la différence de la réponse du solveur haute fidélité et de la réponse du solveur basse fidélité
δ^1	delta correction linéaire
δ^2	delta correction quadratique
δ^3	delta correction cubique

Symboles

∇	opérateur gradient
----------	--------------------

INTRODUCTION

En aéronautique, comme dans de nombreux autres secteurs de haute technologie évoluant dans un contexte de concurrence élevée, les problèmes d'optimisation sont de plus en plus fréquents. Ces problèmes concernent de nombreux départements et en particulier le bureau de conception où l'intégration des procédures dans une boucle d'optimisation a été facilitée par le niveau élevé d'automatisation des calculs et l'omniprésence de l'outil informatique. Les ingénieurs s'appliquent ainsi à améliorer sans cesse leurs produits en optimisant la traînée, le poids ou toute autre caractéristique de l'avion.

Ce critère de qualité, appelé fonction coût ou fonction objectif, qu'ils cherchent justement à optimiser est rarement trivial et doit être évalué par une analyse numérique. Ainsi si l'on cherche à minimiser la traînée d'une aile, il est nécessaire d'utiliser un résolveur numérique capable de calculer l'écoulement et d'en déduire les caractéristiques du design testé. Il suffit ensuite de relier cette évaluation automatique à l'un des nombreux optimiseurs existant sur le marché pour espérer trouver un design optimal.

Or il existe une très grande variété de solveurs numériques qui diffèrent selon :

- les équations physiques qu'ils résolvent : des petites perturbations aux équations de Navier-Stokes avec modèle de turbulence en passant par les équations d'Euler.
- la finesse et l'adaptation du maillage ainsi que le niveau de convergence.

Il est même courant d'utiliser en guise de solveur une approximation ou une surface de réponse capable de représenter grossièrement les comportements d'un résolveur numérique.

Il est alors nécessaire de choisir un de ces solveurs pour le relier à l'optimiseur.

Idéalement, l'ingénieur souhaiterait utiliser son résolveur le plus précis afin d'être certain de la solution à laquelle aboutit l'algorithme. Toutefois un processus d'optimisation requiert souvent de nombreuses évaluations. Ainsi, si chacune des évaluations nécessite plusieurs heures de calcul, l'optimisation sur un solveur haute fidélité pourrait facilement prendre quelques mois ce qui est impensable dans un bureau d'étude...

Les concepteurs se sont donc résolu à utiliser des résolveurs plus grossiers qui font converger l'optimiseur vers des solutions moins fiables qu'il est alors nécessaire de corriger «manuellement» en se basant sur quelques évaluations effectuées avec un résolveur haute fidélité.

Les designers se sont donc penchés depuis deux décennies (voir Schmit et Farshi (1974), Schmit et Miura (1976), Schmit et Fleury (1980) et Barthelemy et Haftka (1993) pour les problèmes en analyse et optimisation structurale) sur le problème de la fidélité variable, c'est-à-dire de l'utilisation combinée de modèles haute et basse fidélité dans le but de profiter de la fiabilité du premier et de la rapidité du second. Toutes ces méthodes sont regroupées sous le terme SAO pour Sequential Approximation and Optimization. Elles consistent toutefois pour la plupart en la fabrication d'une très bonne approximation, soit avant l'optimisation, soit pendant, reliée directement à l'optimiseur. Les procédures sont souvent heuristiques et la convergence vers un optimum du modèle haute fidélité est très rarement garantie.

Dans ce contexte, deux algorithmes mathématiquement rigoureux et qui s'appliquent à deux résolveurs numériques de qualités distinctes ont retenu notre attention. Le premier, proposé par Dennis et Torczon (1996) et Booker *et al.* (1999), est basé sur une stratégie de type «Pattern Search» ne nécessitant pas de gradient tandis que le second, proposé par Alexandrov *et al.* (1997), s'inscrit dans le cadre des optimisations à gradient. Ces deux méthodes garantissent la convergence

vers un optimum du solveur haute fidélité. Les solveurs que nous utilisons étant assez fiables pour calculer des gradients par différences finies, nous avons décidé de bénéficier de l'efficacité des méthodes à gradient et avons donc choisi d'étudier l'algorithme de l'AMF (Approximation Management Framework) proposé par Alexandrov.

Notre étude a donc porté sur la faisabilité et l'intérêt d'une telle méthode pour des problèmes d'optimisation aérodynamique de profils d'aile. Pour cela nous développons dans le premier chapitre la théorie de l'AMF ainsi que l'algorithme que nous avons implanté. Nous validons dans le second chapitre le comportement global de l'AMF sur des cas analytiques de référence et tentons de l'améliorer. Le troisième chapitre est dédié aux résultats expérimentaux obtenus sur l'optimisation aérodynamique d'un profil d'aile. Le dernier chapitre s'attarde finalement sur les problèmes de robustesse rencontrés durant les expériences et présente ainsi les limites de la méthode.

CHAPITRE 1

PRÉSENTATION THÉORIQUE DE L'AMF ET DESCRIPTION DE L'ALGORITHME UTILISÉ

Afin de présenter l'AMF et de démontrer ses propriétés de convergence, nous présenterons dans une première partie la méthode classique du rayon de confiance, sur laquelle se base l'AMF, ainsi que ses propriétés de convergence. Nous expliquerons dans une seconde partie comme nous pouvons appliquer cette méthode à la fidélité variable et nous détaillerons finalement les modifications à apporter pour résoudre un problème avec contraintes.

1.1 Méthode classique du rayon de confiance pour un problème sans contraintes

Nous présentons dans cette partie la méthode du rayon de confiance. Nous recommandons vivement la lecture du livre de Conn *et al.* (2000) pour une présentation exhaustive et pour les démonstrations détaillées des méthodes à rayon de confiance. Les premiers algorithmes utilisant cette stratégie d'optimisation ont été développés en 1944 par Levenberg et en 1963 par Marquardt. Les travaux de Powell (1975) ont permis de démontrer la convergence de ce type d'algorithme vers un point stationnaire du problème en supposant une série d'hypothèses peu contraignantes. Par mesure de simplification et pour la clarté du propos nous commencerons par exposer la méthode en l'appliquant à un problème d'optimisation sans contraintes.

1.1.1 Présentation de l'algorithme

1.1.1.1 Principe de la méthode

Soit le problème d'optimisation suivant :

$$\min_x f(x),$$

où $f(x)$, fonction de $\mathbb{R}^N$ dans $\mathbb{R}$, est la fonction objectif à minimiser et x le vecteur de design. La méthode du rayon de confiance est un procédé itératif qui consiste, à chaque itération k , à créer une approximation quadratique de la fonction coût $f(x)$ autour du point courant x_k . On définit donc à chaque itération k la fonction suivante :

$$q_k(x_k + s) = f(x_k) + \nabla^T f(x_k)s + \frac{1}{2}s^T H_k s$$

où H_k est le hessien de la fonction coût et où $s \in \mathbb{R}^N$.

L'idée consiste ensuite à transférer le problème d'optimisation de la fonction coût vers son approximation quadratique. La fonction $q_k(x_k + s)$ étant une approximation locale, il nous faut définir une zone, dite de confiance, dans laquelle elle prédit avec justesse le comportement de la fonction coût. Il s'agit donc, à chaque itération, de résoudre le sous-problème d'optimisation suivant :

$$\begin{aligned} \min_s \quad & q_k(x_k + s) \\ \text{t.q.} \quad & \|s\| \leq \delta_k \end{aligned} \tag{1.1}$$

où δ_k est appelé rayon de confiance. Nous présenterons un peu plus loin comment sa valeur est déterminée de manière dynamique. Nous n'avons pas précisé le type de norme utilisé puisque la théorie s'applique à tout type de norme. En ce qui nous concerne, nous avons utilisé la norme infinie.

1.1.1.2 Critères d'acceptation du pas

Le pas s_k , solution du sous-problème d'optimisation 1.1, est accepté si et seulement si il conduit à une diminution de la valeur de la fonction coût puisque c'est elle qui doit être minimisée. Ainsi :

$$x_{k+1} = \begin{cases} x_k + s_k & \text{si } f(x_k + s_k) < f(x_k) \\ x_k & \text{sinon} \end{cases} \quad (1.2)$$

Si le pas proposé est rejeté, en supposant qu'il s'agit bien du minimum du sous-problème (1.1), cela signifie que l'on a surestimé la faculté de l'approximation quadratique à bien représenter le comportement de la fonction coût dans la zone de confiance définie par δ_k . Il faut alors réduire le rayon de confiance jusqu'à ce que la solution s_k du sous-problème d'optimisation conduise à une amélioration de la fonction coût. Intuitivement, nous pouvons prévoir qu'à force de diminuer le rayon, un pas s_k finira bien par être accepté, puisque la fonction $q_k(x_k + s)$ est tangente à la fonction coût $f(x)$ en x_k . Nous devons alors préciser, et nous le détaillerons plus tard, que f est différentiable par hypothèse.

1.1.1.3 Critères de mise-à-jour du rayon de confiance

Nous allons présenter maintenant la règle systématique d'adaptation du rayon de confiance qui dépend directement de la solution du sous-problème d'optimisation. Il est nécessaire en effet de quantifier la qualité de l'approximation quadratique pour faire varier intelligemment la valeur du rayon de confiance. Pour cela on a défini une variable r , quotient de l'amélioration réelle mesurée par $f(x)$ entre x_k et $x_k + s_k$, et de l'amélioration prévue par $q_k(x_k + s)$:

$$r = \frac{f(x_k) - f(x_k + s_k)}{f(x_k) - q_k(x_k + s_k)}. \quad (1.3)$$

Nous pouvons remarquer d'emblée que le dénominateur du quotient est toujours positif. Concrètement, plus r est proche de 1, plus $q_k(x_k + s)$ est de bonne qualité sur le domaine défini par δ_k . Si r est positif cela signifie que $q_k(x_k + s)$ a permis de déterminer un pas s_k qui améliore la fonction coût $f(x)$. Si r est négatif, le pas s_k conduit malheureusement à une augmentation de la fonction coût réelle. Si r est supérieur à 1 l'approximation n'est pas bonne mais elle a permis par chance de déterminer un très bon pas s_k .

Le rayon de confiance sera donc adapté selon les règles suivantes :

Soient $0 < r_1 < r_2 < 1$ et $0 < c_1 < 1 < c_2$,

$$\delta_{k+1} = \begin{cases} c_1 \|s_k\| & \text{si } r < r_1 \\ c_2 \delta_k & \text{si } r > r_2 \\ \|s_k\| & \text{sinon} \end{cases} \quad (1.4)$$

les valeurs typiques des coefficients sont $r_1 = 0,1$, $r_2 = 0,75$, $c_1 = 0,5$ et $c_2 = 2$.

1.1.1.4 Algorithme de la méthode du rayon de confiance

L'algorithme de la méthode du rayon de confiance peut se présenter de la manière suivante :

jusqu'à convergence :

Construire l'approximation q_k et trouver une solution s_k au sous-problème d'optimisation :

$$\begin{array}{ll} \min_s & q_k(x_k + s) \\ \text{t.q.} & \|s\| \leq \delta_k. \end{array}$$

Calculer r and mettre à jour x_k and δ_k selon (1.2) et (1.4).

Les critères d'arrêt de ce type d'algorithme sont discutés dans Gay (1982) et Dennis et Schnabel (1983). Succinctement, la boucle est arrêtée si :

- la norme du gradient de la fonction coût est inférieure à un ϵ_∇ proche de zéro.
- la norme du pas, lorsqu'il est accepté, est inférieure à un ϵ_s proche de zéro. Cela signifie alors que l'on progresse très peu dans l'espace de design.
- l'amélioration de la fonction coût entre deux itération est inférieure à un ϵ_f proche de zéro. Cela signifie alors que l'on améliore très peu la valeur de la fonction coût.

1.1.2 Hypothèses et propriétés de convergence

Il s'agit maintenant de présenter les propriétés de convergence de la méthode du rayon de confiance, issues du théorème de convergence globale de Powell (1975). Pour cela il nous faut vérifier un certain nombre d'hypothèses.

1.1.2.1 Hypothèses sur la nature de la fonction coût

La fonction coût f est bornée inférieurement, elle est différentiable et sa dérivée est uniformément continue. Le hessien est lui aussi uniformément borné. Si la condition sur le hessien est difficile à remplir, elle peut être remplacée par une condition sur la suite des hessiens par où passe l'algorithme. On exige alors que la suite $(H_k)_{k \in \mathbb{N}^+}$ soit uniformément bornée.

1.1.2.2 Hypothèses sur la fonction $q_k(x_k + s)$

Les hypothèses suivantes sont triviales à vérifier lorsqu'un modèle quadratique de la fonction coût est réellement utilisé pour $q_k(x_k + s)$. Cependant, elles nous permettent de distinguer, parmi toutes les propriétés de l'approximation quadratique, celles qui entraînent effectivement la convergence de l'algorithme.

- Les valeurs de q_k et de f en x_k doivent être égales.
- Les gradients de q_k et de f en x_k doivent être égaux. Cette propriété est évidente si q_k est un véritable modèle quadratique. Nous ajoutons cependant que si le gradient de q_k n'est qu'une approximation de celui de f , les propriétés de convergence sont maintenues et nous renvoyons le lecteur à l'article de Carter (1991) pour plus de précisions.
- La suite des hessiens de q_k , $(\nabla^2 q_k)_{k \in \mathbb{N}^+}$, doit être uniformément bornée. Cette

condition est la seule qui porte sur le hessien de q_k si bien qu'il n'est pas nécessaire de calculer exactement le hessien de f pour construire l'approximation quadratique. Nous pouvons ainsi nous contenter d'une approximation B_k de H_k . Cette approximation peut être évaluée par une méthode de type BFGS qui utilise les valeurs des précédents gradients de f pour générer une estimation du hessien (voir par exemple Vanderplaats (1999)). Nous pourrions même remplacer H_k dans l'expression du modèle quadratique par la matrice nulle ou identité. Evidemment, nous espérons une vitesse de convergence accrue lorsque B_k représente bien H_k , propriété qui se retrouve dans les méthodes quadratiques de type SQP.

Remarque : Le concept de borne uniforme d'un ensemble de matrice $(B_k)_{k \in K}$ où K n'est pas nécessairement dénombrable est plus connu dans la littérature francophone à travers le théorème de Banach-Steinhaus. Il consiste juste à dire qu'il existe une constante c pour laquelle

$$\|B_k x\| \leq c \|x\| \quad \forall x \in \mathbb{R}^N \quad \forall k \in K.$$

Toutefois si l'on se muni de la norme des applications linéaires de $\mathbb{R}^N$ dans $\mathbb{R}^N$ définie par :

$$\|T\|_{\mathcal{L}(\mathbb{R}^N, \mathbb{R}^N)} = \sup_{x \in \mathbb{R}^N, \|x\| \leq 1} \|Tx\|,$$

celà revient à dire que l'ensemble $(B_k)_{k \in K}$ est borné.

1.1.2.3 Hypothèses sur la résolution du sous-problème d'optimisation

Il n'est pas nécessaire de résoudre de manière exacte le sous-problème d'optimisation portant sur la fonction $q_k(x_k + s)$. En effet la solution s_k doit juste vérifier la condition connue sous le nom de *Fraction of Cauchy Decrease* ou FCD. Cette condition exige que le gain prédit par le pas s_k soit au moins égal à une fraction du gain prédit par le pas de Cauchy. Cette condition nous permet d'affirmer que, même si s_k n'est pas le véritable minimum du problème (1.1), il existe toujours, pour un x_k qui n'est pas un point stationnaire du problème, un $\delta > 0$ en deçà duquel le pas s_k sera accepté.

Le pas de Cauchy est défini comme le pas, parallèle au gradient de f , qui minimise l'approximation quadratique de f en x_k . Il correspond en fait à la solution du problème unidimensionnel suivant :

Soit m_k , fonction de $\mathbb{R}^N$ dans $\mathbb{R}$, définie comme l'approximation quadratique de f en x_k ,

$$\begin{aligned} \min_{\lambda} \quad & m_k(x_k - \lambda \nabla f(x_k)) = f(x_k) - \lambda \|\nabla f(x_k)\|^2 + \frac{1}{2} \lambda^2 \nabla f(x_k)^T B_k \nabla f(x_k) \\ \text{t.q.} \quad & \|\lambda \nabla f(x_k)\| \leq \delta_k. \end{aligned}$$

Si $\nabla f(x_k)^T B_k \nabla f(x_k) > 0$ alors la fonction est convexe et le minimum λ_* vérifie :

$$\lambda_* = \frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^T B_k \nabla f(x_k)} \quad \text{si } \lambda_* \|\nabla f(x_k)\| \leq \delta_k.$$

Dans le cas où le minimum de la fonction convexe n'est pas à l'intérieur du rayon de confiance ou si la fonction est concave, la solution du problème est à la frontière de la zone de confiance.

Ainsi le pas de Cauchy vaut :

$$s_{Cauchy} = \begin{cases} \lambda_* \nabla f(x_k) & \text{si } \lambda_* \|\nabla f(x_k)\| \leq \delta_k \text{ et } \nabla f(x_k)^T B_k \nabla f(x_k) > 0 \\ \frac{\delta_k}{\|\nabla f(x_k)\|} \nabla f(x_k) & \text{sinon} \end{cases}$$

Pour pouvoir satisfaire la condition FCD il faut donc vérifier que le pas s_k proposé comme solution du sous-problème d'optimisation vérifie pour un $\kappa > 0$ la propriété suivante :

$$q_k(x_k + s_k) - q_k(x_k) \leq \kappa(q_k(x_k + s_{Cauchy}) - q_k(x_k)) \quad (1.5)$$

La condition FCD peut aussi s'exprimer sous la forme suivante : il existe $\beta > 0$ et $C > 0$ indépendants de k pour lesquels le pas s_k vérifie :

$$q_k(x_k) - q_k(x_k + s_k) \geq \beta \|\nabla f(x_k)\| \min(\delta_k, \frac{\|f(x_k)\|}{C}). \quad (1.6)$$

L'une ou l'autre de ces conditions est nécessaire pour démontrer théoriquement la convergence de la méthode. Powell (1975) et More (1983) ont démontré que la propriété (1.5) entraîne automatiquement (1.6).

Cette condition nous permet d'affirmer que si nous ne sommes pas sur un point stationnaire du problème initial, alors il existe un rayon $\delta > 0$ en deçà duquel non seulement le pas sera accepté mais en plus avec un facteur $r > r_2$. Nous renvoyons le lecteur au théorème 6.4.2 établi dans le livre de Conn *et al.* (2000) pour la démonstration.

Concrètement, cette condition FCD est grossière et très facile à vérifier. Les optimiseurs courant cherchent en effet à minimiser la fonction coût ce qui les amène

à produire des solutions qui vérifient des conditions bien plus fortes que la FCD comme la *Fraction of Optimal Decrease* (FOD). Cette dernière permet même d'assurer la convergence de l'algorithme vers une solution au second ordre du problème initial à un rythme quadratique. Les preuves ont été établies par Sorensen (1982) mais nous ne nous attarderons sur les propriétés de l'algorithme général que dans le cas où la FCD est vérifiée.

1.1.2.4 Propriétés de convergence

Les hypothèses précédentes étant vérifiées, le théorème de convergence globale de Powell (1975) nous assure que la suite des points x_k générés par l'algorithme du rayon de confiance vérifie :

$$\liminf_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0.$$

Toutefois si on remplace le critère d'acceptation du pas s_k (1.2) par un critère un peu plus sévère qui s'énonce ainsi : Soit $1 > \mu > 0$,

$$x_{k+1} = \begin{cases} x_k + s_k & \text{si } r > \mu \\ x_k & \text{sinon} \end{cases} \quad (1.7)$$

alors la suite des points x_k générés par l'algorithme du rayon de confiance vérifie :

$$\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0.$$

Cela signifie que l'algorithme du rayon de confiance converge vers un point stationnaire du problème initial. Cet algorithme constitue donc une méthode mathématiquement robuste qui converge quel que soit le point initial.

Pour conclure et en guise de synthèse du paragraphe 1.1.2 nous énonçons le théorème de Powell de convergence globale.

Théorème 1 (de convergence globale) *Si f est bornée inférieurement, uniformément et continuellement différentiable, si son hessien est uniformément borné, si la suite des hessiens $(\nabla^2 q_k)_{k \in \mathbb{N}^+}$ est bornée, si le pas proposé s_k vérifie le critère FCD et si le critère d'acceptation du pas utilisé correspond à (1.7), alors la suite des points x_k générés par l'algorithme présenté en 1.1.1.4 vérifie :*

$$\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0,$$

et converge donc vers un point stationnaire du problème initial.

1.2 Application de la méthode du rayon de confiance à la fidélité variable

Maintenant que nous avons présenté la méthode du rayon de confiance pour des cas d'optimisation sans contraintes et que nous connaissons ses propriétés de convergence et de robustesse, nous allons voir comment nous pouvons l'utiliser dans le cadre de l'optimisation à fidélité variable.

Nous considérons un problème d'optimisation sans contraintes, soit

$$\min_x f(x),$$

pour lequel nous disposons de deux solveurs capables d'évaluer la valeur de f , l'un de haute fidélité qui retourne $f_{hi}(x)$, et l'autre de basse fidélité qui retourne $f_{lo}(x)$. Nous cherchons bien entendu à minimiser la valeur de $f_{hi}(x)$.

1.2.1 Introduction du modèle basse fidélité dans la méthode du rayon de confiance par l'intermédiaire d'une nouvelle approximation

Dans la méthode du rayon de confiance, nous utilisons une approximation purement locale de la fonction coût. Cependant, l'utilisateur a une certaine intuition sur le comportement global de la fonction coût grâce à un solveur de basse fidélité ou modèle faible $f_{lo}(x)$. L'idée proposée par Alexandrov *et al.* (1997) consiste à modifier ce modèle afin qu'il puisse vérifier les mêmes hypothèses que l'approximation quadratique dans la méthode du rayon de confiance. Nous pourrions alors remplacer le modèle quadratique par cette approximation et bénéficier alors des propriétés de convergence et de robustesse de l'algorithme présenté au paragraphe 1.1.1.4. De plus, et c'est là tout l'intérêt de la méthode, nous transférerons le poids de l'optimisation, qui porte originellement sur le modèle haute fidélité, sur le modèle basse fidélité utilisé à chaque itération dans le sous-problème d'optimisation. Nous espérons ainsi accélérer le processus en diminuant le nombre d'appel au modèle haute fidélité, ce qui est propre à la méthode du rayon de confiance, et en augmentant la portée de l'approximation grâce au modèle faible.

1.2.2 Hypothèses sur la nouvelle approximation

Les propriétés de convergence de la méthode du rayon de confiance découlent du fait que les fonctions q_k et f ont les mêmes valeurs et les mêmes gradients en x_k à chaque itération. C'est cette propriété qui nous assure qu'en diminuant le rayon

de confiance, nous améliorons la qualité de l'approximation.

Ainsi, il nous suffit de construire, à chaque itération, une approximation $a_k(x_k + s)$ basée sur le modèle faible $f_{lo}(x)$ et égale au modèle fort $f_{hi}(x)$ au premier ordre en x_k . Soit :

$$\begin{aligned} a_k^f(x_k) &= f_{hi}(x_k) \quad \text{et} \\ \nabla a_k^f(x_k) &= \nabla f_{hi}(x_k). \end{aligned} \tag{1.8}$$

Nous pourrions alors bénéficier de toutes les propriétés de convergence et de robustesse de la méthode du rayon de confiance en remplaçant la fonction $q_k(x_k + s)$ par la fonction $a_k(x_k + s)$.

1.2.3 Construction de la nouvelle approximation par la β -correction

Nous présentons dans cette partie une méthode pour construire l'approximation $a_k^f(x)$, basée sur le modèle basse fidélité $f_{lo}(x)$ et vérifiant les équations (1.8).

Cette stratégie proposée par Chang *et al.* (1993) et citée à maintes reprises dans la littérature (voir Alexandrov *et al.* (1997)) est connue sous le nom de «bêta correction» et permet de manière simple et systématique de construire $a_k^f(x)$.

Soit la fonction $\beta(x)$ définie de $\mathbb{R}^N$ dans $\mathbb{R}$ par l'équation :

$$\beta(x) = \frac{f_{hi}(x)}{f_{lo}(x)}.$$

Cette fonction est à priori inconnue de l'utilisateur puisque ce dernier ne connaît pas l'allure de $f_{hi}(x)$ et de $f_{lo}(x)$. En revanche, à chaque itération x_k , nous pouvons calculer $f_{hi}(x_k)$, $\nabla f_{hi}(x_k)$, $f_{lo}(x_k)$ et $\nabla f_{lo}(x_k)$. Nous pouvons alors en déduire directement les valeurs de $\beta(x_k)$ et de $\nabla\beta(x_k)$.

Nous définissons ensuite la fonction $\beta_k(x)$ comme l'approximation linéaire de $\beta(x)$ en x_k , soit :

$$\beta_k(x) = \beta(x_k) + \nabla^T \beta(x_k)(x - x_k).$$

Nous sommes alors en mesure de construire l'approximation $a_k^f(x)$ en la définissant par :

$$a_k^f(x) = \beta_k(x)f_{lo}(x).$$

Le lecteur attentif pourra vérifier que, ainsi définie, l'approximation vérifie bien les équations (1.8).

1.3 Méthode du rayon de confiance appliquée à un problème d'optimisation avec contraintes

Nous avons vu au paragraphe précédent en quoi consistait la méthode du rayon de confiance appliquée à la fidélité variable pour un problème d'optimisation sans contraintes. Il s'agit maintenant d'expliquer comment les contraintes peuvent être prises en compte dans l'algorithme pour que ce dernier puisse bénéficier des mêmes

propriétés de convergence. Nous présenterons deux méthodes qui permettent de traiter un problème d'optimisation général sous contraintes et qui utilisent la méthode du rayon de confiance.

Nous nous plaçons dans le cas du problème général formalisé ainsi :

$$\begin{aligned} \min_x \quad & f(x) \\ \text{t.q.} \quad & g(x) \geq 0 \\ & h(x) = 0, \end{aligned}$$

où $f(x)$ est la fonction coût, $g : \mathbb{R}^N \rightarrow \mathbb{R}^{N_g}$, le vecteur des N_g contraintes d'inégalité et $h : \mathbb{R}^N \rightarrow \mathbb{R}^{N_h}$ le vecteur des N_h contraintes d'égalité. $f_{hi}(x)$, $g_{hi}(x)$, $h_{hi}(x)$ et $f_{lo}(x)$, $g_{lo}(x)$, $h_{lo}(x)$ sont leurs valeurs estimées reciproquement par les modèles haute et basse fidélité.

1.3.1 Lagrangien augmenté

La façon la plus triviale et la plus directe pour ramener un problème sous contraintes à un problème sans contraintes est d'utiliser la méthode du Lagrangien augmenté. Nous présenterons rapidement l'algorithme qui lie le Lagrangien augmenté au rayon de confiance comme Conn *et al.* (1991) l'ont implémenté.

1.3.1.1 Méthode classique du Lagrangien augmenté

La méthode classique du Lagrangien augmenté consiste dans un premier temps à transformer toutes les contraintes d'inégalité du problème général en contraintes

d'égalité.

$$h(x, z) = g(x) + z,$$

où $z \in \mathbb{R}^{N_g}$. On se rapporte alors au problème suivant :

$$\begin{aligned} \min_{x, z} \quad & f(x) \\ \text{t.q.} \quad & h(x, z) = 0 \\ & z \geq 0. \end{aligned}$$

Le vecteur $h(x, z)$ est alors le fruit de la concaténation des contraintes d'égalité $h_i(x)$ et des contraintes d'inégalité modifiées $h_j(x, z)$. On pose alors le Lagrangien augmenté comme étant la fonction :

$$L(x, z, \lambda, \mu) = f(x) + \lambda^T h(x, z) + \frac{1}{2\mu} \|h(x, z)\|_2^2$$

où λ est le vecteur des multiplicateurs de Lagrange et μ la pénalité. L'algorithme classique s'écrit alors ainsi :

Initialiser $(x_i, z_i, \lambda_i, \mu_i)$ et jusqu'à convergence :

Avec λ_i et μ_i fixés, trouver la solution (x^+, z^+) du sous-problème d'optimisation suivant :

$$\begin{array}{ll} \min_{x,z} & L(x, z, \lambda_i, \mu_i) \\ \text{t.q.} & z \geq 0. \end{array}$$

Mettre à jour $(x_i, z_i) = (x^+, z^+)$ et (λ_i, μ_i) en fonction des nouveaux (x^+, z^+) .

L'étape de mise-à-jour de (λ_i, μ_i) en fonction de la solution (x^+, z^+) du sous-problème d'optimisation est clairement définie dans la littérature, on pourra par exemple se référer au livre de Gauvin (1995).

1.3.1.2 Lagrangien augmenté associé à la méthode du rayon de confiance

La première façon d'intégrer les deux méthodes consiste à résoudre le sous-problème d'optimisation du Lagrangien par la méthode du rayon de confiance. Ce qui revient à créer deux niveaux de sous-problèmes. L'algorithme ainsi produit a la forme ci-contre :

Initialiser $(x_i, z_i, \lambda_i, \mu_i)$ et jusqu'à convergence :

Pour (λ_i, μ_i) fixé, minimiser la fonction $L_i(x, z) = L(x, z, \lambda_i, \mu_i)$ par la méthode du rayon de confiance en initialisant $(x_{k=0}, z_{k=0})$ à (x_i, z_i) et en effectuant les opérations suivantes jusqu'à convergence :

- Construire une approximation $a_k^{L_i}(x_k + s^x, z_k + s^z)$ du Lagrangien égale au premier ordre au Lagrangien en (x_k, z_k) . L'approximation vérifie donc les égalités suivantes :

$$\begin{aligned} a_k^{L_i}(x_k, z_k) &= L_i(x_k, z_k) \\ \nabla a_k^{L_i}(x_k, z_k) &= \nabla L_i(x_k, z_k) \end{aligned}$$

- Trouver la solution (s_k^x, s_k^z) du sous-problème d'optimisation suivant :

$$\begin{aligned} \min_{s^x, s^z} \quad & a_k^{L_i}(x_k + s^x, z_k + s^z) \\ \text{t.q.} \quad & z_k + s^z \geq 0 \\ & \|s^x\| \leq \delta_k \\ & \|s^z\| \leq \delta_k \end{aligned}$$

- Calculer $r = \frac{L_i(x_k, z_k) - L_i(x_k + s_k^x, z_k + s_k^z)}{L_i(x_k, z_k) - a_k^{L_i}(x_k + s_k^x, z_k + s_k^z)}$
- Mettre à jour (x_k, z_k) et δ_k en fonction de r

Soit (x^+, z^+) la solution issue de la dernière itération k , mettre à jour $(x_i, z_i) = (x^+, z^+)$ et (λ_i, μ_i) en fonction des nouveaux (x^+, z^+) .

Nous insistons sur le fait que cet algorithme permet de découpler complètement le problème en ce sens que l'optimisation avec la méthode du rayon de confiance peut être effectuée par une boîte noire qui résoud, elle, le sous-problème d'optimisation propre à la boucle de la méthode du Lagrangien augmenté. Cette formulation se retrouve notamment dans le livre de Conn *et al.* (2000) au paragraphe 14.4.1.

Ce type de formulation à deux niveaux de sous-problèmes peut s'apparenter aux problèmes multi-niveaux apparaissant dans les optimisations qui impliquent un couplage entre plusieurs disciplines. Les articles sur ce sujet (voir Alexandrov et Lewis (2000)b et Alexandrov et Lewis (2000)a), désigné dans la littérature sous le sigle MDO (Multidisciplinary Design Optimization), concluent souvent sur une meilleure performance des formulations de type intégré. Nous pouvons en effet observer, dans l'algorithme précédent, qu'il n'est pas très utile au cours des premières itérations de la boucle propre au Lagrangien augmenté de résoudre précisément le problème de minimisation de $L_i(x, z)$. Pour éviter ces itérations coûteuses et peu intéressantes, une formulation intégrée où les deux boucles seraient fusionnées semble être une voie intéressante à explorer.

Un tel algorithme a été proposé par Alexandrov *et al.* (2000). Nous pouvons effectivement observer que les étapes de mise-à-jour du rayon de confiance et d'acceptation du pas, propres à la méthode du rayon de confiance, s'effectuent en même temps que celles de mise-à-jour des multiplicateurs de Lagrange et de la pénalité, propres à la méthode du Lagrangien augmenté.

Initialiser $(x_k, z_k, \lambda_k, \mu_k)$

jusqu'à convergence :

Calculer le Lagrangien, son gradient et si possible son hessien par rapport aux variables (x, z) afin de construire l'approximation quadratique de $L(x, z, \lambda_k, \mu_k)$, soit $q_k^L(x_k + s^x, z_k + s^z, \lambda_k, \mu_k)$.

Trouver la solution (s_k^x, s_k^z) du sous-problème d'optimisation suivant :

$$\begin{aligned} \min_{s^x, s^z} \quad & q_k^L(x_k + s^x, z_k + s^z, \lambda_k, \mu_k) \\ \text{t.q.} \quad & z \geq 0 \\ & \|s^x\| \leq \delta_k \\ & \|s^z\| \leq \delta_k \end{aligned}$$

$$\text{Calculer } r = \frac{L(x_k, z_k, \lambda_k, \mu_k) - L(x_k + s_k^x, z_k + s_k^z, \lambda_k, \mu_k)}{L(x_k, z_k, \lambda_k, \mu_k) - q_k^L(x_k + s_k^x, z_k + s_k^z, \lambda_k, \mu_k)}$$

Mettre à jour (x_k, z_k) et δ_k en fonction de r

Mettre à jour (λ_k, μ_k) .

Ces deux algorithmes permettent donc d'utiliser de manière assez simple la méthode du rayon de confiance et de l'appliquer à la fidélité variable pour des problèmes d'optimisation avec contraintes. Toutefois, en utilisant ces stratégies, nous nous attendons à hériter aussi des problèmes inhérents à la méthode du Lagrangien augmenté, soient notamment les problèmes de conditionnement qui apparaissent à mesure que la pénalité μ tend vers 0.

1.3.2 Méthodes SQP

Nous présentons en second lieu la méthode SQP parce qu'elle permet de résoudre un problème sous contraintes et qu'elle peut être liée naturellement à la méthode du rayon de confiance. Cette démarche nous permet donc de présenter une autre façon de résoudre un problème d'optimisation à fidélité variable avec contraintes.

Les stratégies d'optimisation SQP (Sequential Quadratic Programming) sont très populaires parce qu'elles bénéficient d'une rapidité de convergence rarement égalée. En fait, dès que l'utilisateur peut avoir une bonne évaluation du gradient de la fonction coût et pour peu que cette dernière soit assez régulière, il a tout intérêt à utiliser une méthode SQP, à fortiori lorsque le nombre de variables de design est élevé. Un aperçu des méthodes SQP est développé dans le livre de Gill *et al.* (1981).

1.3.2.1 Principe de la méthode

La méthode SQP ressemble beaucoup à la méthode du rayon de confiance en ce sens qu'elle résout une suite de sous-problèmes d'optimisation qui portent non pas sur la fonction coût mais sur une approximation quadratique de cette dernière. Dans cette méthode, une approximation B_k du hessien H_k est mise-à-jour à chaque itération grâce aux informations que l'on acquiert sur les gradients $\nabla f(x_k)$. En guise d'exemple nous pouvons citer la méthode BFGS (voir par exemple Vanderplaats (1999)) qui sert à approximer le hessien. Le sous-problème d'optimisation tient compte en plus d'une approximation linéaire de chacune des contraintes. Il s'écrit ainsi de la manière suivante :

$$\begin{aligned}
\min_s \quad & \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s \\
\text{t.q.} \quad & g(x_k) + \nabla g(x_k) s \geq 0 \\
& h(x_k) + \nabla h(x_k) s = 0
\end{aligned}$$

On peut alors montrer (voir Papalambros et Wilde (2000)) que résoudre ce sous-problème d'optimisation revient à utiliser une méthode de Newton pour mettre à jour les variables de design ainsi que les multiplicateurs de Lagrange. La solution de chaque itération se calcule donc directement en résolvant un système linéaire. La résolution de ce système est bien documentée (voir Conn *et al.* (2000)) et s'effectue directement par factorisation ou par un procédé itératif comme la méthode des gradients conjugués ou la méthode de Lanczos.

Cette stratégie pose cependant des problèmes de convergence globale inhérents à la méthode de Newton. En effet, loin de l'optimum, l'algorithme peut diverger ou encore le sous-problème d'optimisation peut avoir une solution non-bornée ou non-réalisable. Il est alors nécessaire d'adopter une stratégie de stabilisation et ce, dans toute méthode SQP. On distingue alors principalement deux approches pour stabiliser une méthode SQP (voir Gould et Toint (1999)), la recherche linéaire et la méthode du rayon de confiance. Nous présenterons bien évidemment un algorithme lié à la méthode du rayon de confiance puisque nous montrerons ainsi comment généraliser la stratégie sans contraintes du paragraphe 1.1.1 au problème avec contraintes.

1.3.2.2 Méthode SQP à rayon de confiance

L'introduction des contraintes dans l'algorithme présenté au paragraphe 1.1.1 ne se fait pas de manière triviale. En effet, il n'est plus possible en présence de contraintes d'utiliser un critère de mise-à-jour du rayon de confiance basé uniquement sur l'amélioration de la fonction coût comme présenté en (1.3). Il faut en effet tenir compte des progrès réalisés tant dans la diminution de la fonction coût que dans celle de l'infaisabilité. Il nous faut alors utiliser une fonction «mérite» capable de tenir compte des deux objectifs de l'optimisation à savoir trouver un point réalisable qui minimise la fonction coût.

Présentation de la fonction mérite

Il existe plusieurs façons de considérer l'influence des contraintes dans une méthode SQP à rayon de confiance (voir Gould et Toint (1999)). Nous retiendrons celle qui introduit la fonction de pénalité l_2 et qui est notamment explicitée dans l'article d'Alexandrov (1998).

Pour cela nous introduisons une fonction mérite, $P(x, \rho)$, qui va jouer le même rôle que la fonction coût dans l'optimisation sans contraintes :

$$P(x, \rho) = f(x) + \rho \|I(x)\|^2,$$

où $\rho > 0$ est une pénalité et $I(x)$ représente le vecteur d'infaisabilité du problème. Soient N_g et N_h respectivement le nombre de contraintes d'inégalité et d'égalité,

on définit le vecteur d'infaisabilité par :

$$I_i(x) = \begin{cases} \max(0, -g_i(x)) & \text{si } i \leq N_g \\ |h_i(x)| & \text{si } N_g < i \leq N_g + N_h. \end{cases}$$

La pénalité ρ traduit l'importance qu'accorde le processus d'optimisation à trouver un point réalisable, il faut donc définir une méthode de mise-à-jour dynamique de cette variable.

Mise-à-jour de la pénalité dans la fonction mérite

Alexandrov (1998) propose un algorithme simple mais rigoureux de mise-à-jour de la pénalité ρ . Cet algorithme s'assure que la prévision de réduction de la fonction mérite, évaluée par les approximations à la fin de chaque sous-optimisation, est au moins positive. Cet algorithme est une extension de celui proposé par El-Alem (1988).

On initialise ρ_0 à 1 et on choisit $\beta \in (0, 1)$. La mise-à-jour de la pénalité s'effectue à chaque itération en fonction du pas s_k proposé.

Calculer la prévision de réduction de l'infaisabilité

$$Ipred_k = \|I(x_k)\|^2 - \|I(x_k) + \nabla I(x_k)^T s_k\|^2,$$

Calculer la prévision de réduction de la fonction mérite

$$pred_k = (f(x_k) - (f(x_k) + \nabla f(x_k)^T s_k + \frac{1}{2} s_k^T B_k s_k)) + \rho_k Ipred_k,$$

Calculer ρ_+ :

$$\rho_+ = \begin{cases} \rho_k & \text{si } pred_k \geq \frac{\rho_k}{2} Ipred_k \\ \beta + 2 \frac{(f(x_k) + \nabla f(x_k)^T s_k + \frac{1}{2} s_k^T B_k s_k) - f(x_k)}{Ipred_k} & \text{sinon} \end{cases}$$

Mettre à jour $\rho_k = \rho_+$

Concrètement, cette mise-à-jour permet de modifier la pénalité en fonction du travail qui a été réalisé lors de la résolution du sous-problème d'optimisation. Nous partons du principe que cette optimisation se comporte de manière logique, à savoir que si le point de départ est infaisable, alors le pas solution s_k tente, sinon de trouver un point réalisable, au moins d'en trouver un qui réduit l'infaisabilité. De même, si le point de départ est réalisable, on s'attend à une solution qui l'est aussi. Ceci nous permet d'affirmer que $Ipred_k > 0$.

Ainsi dans le cas où le travail d'optimisation a consisté à réduire l'infaisabilité, ce qui se fait souvent au détriment de la fonction coût, la pénalité ρ_k est augmentée de manière à obtenir *in fine* une prévision de réduction de la fonction mérite positive. Sans cette augmentation, la pénalité pourrait être trop faible par rapport à l'augmentation de la fonction coût ce qui aboutirait à une augmentation de la fonction mérite alors qu'elle est justement sensée avoir été réduite durant l'étape de sous-optimisation...

Nouveau critère de mise-à-jour du rayon de confiance

Maintenant que nous sommes certains que $pred_k > 0$, nous pouvons utiliser la fonction mérite $P(x, \rho)$ pour déterminer la qualité des approximations et donc la mise-à-jour du rayon de confiance δ_k .

ρ_k étant fixé pour cette itération, on pose alors r comme le quotient de la réduction de la fonction mérite effective et de la réduction de la fonction mérite prévue, soit

$$r = \frac{P(x_k, \rho_k) - P(x_k + s_k, \rho_k)}{pred_k}, \quad (1.9)$$

ou encore de manière plus explicite :

$$\frac{(f(x_k + s_k) + \rho \|I(x_k + s_k)\|^2) - (f(x_k) + \rho \|I(x_k)\|^2)}{(f(x_k) + \nabla f(x_k)^T s_k + \frac{1}{2} s_k^T B_k s_k + \rho \|I(x_k) + \nabla I(x_k)^T s_k\|^2) - (f(x_k) + \rho \|I(x_k)\|^2)}.$$

Ce facteur r remplace dorénavant celui représenté en (1.3) et servira donc à mettre à jour δ_k .

Algorithme de la méthode SQP à rayon de confiance

Un problème d'optimisation général peut se résoudre avec la méthode SQP à rayon de confiance en respectant l'algorithme suivant :

Jusqu'à convergence :

Résoudre le sous-problème d'optimisation :

$$\begin{aligned} \min_s \quad & \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s \\ \text{t.q.} \quad & g(x_k) + \nabla g(x_k) s \geq 0 \\ & h(x_k) + \nabla h(x_k) s = 0 \\ & \|s\| \leq \delta_k, \end{aligned}$$

Mettre à jour ρ_k , calculer la fonction mérite $P(x_k + s_k)$

et déterminer le quotient r ,

Mettre à jour δ_k et x_k en fonction de r ,

Il ne nous reste plus qu'à montrer comment on introduit la fidélité variable dans cette méthode SQP à rayon de confiance.

1.3.2.3 Algorithme de la méthode SQP à rayon de confiance appliquée à la fidélité variable

Nous présentons l'algorithme de ce que nous appellerons dorénavant l'AMF (Approximation Management Framework), basé sur la méthode SQP, et tel qu'il a été proposé par Alexandrov *et al.* (2000).

Jusqu'à convergence et pour chaque itération k :

Construire $a_k^f(x)$, $a_k^g(x)$ et $a_k^h(x)$ respectant (1.8) pour chaque fonction,

Résoudre le sous-problème d'optimisation :

$$\begin{aligned} \min_s \quad & a_k^f(x_k + s) \\ \text{t.q.} \quad & a_k^g(x_k) + \nabla a_k^g(x_k)^T s \geq 0 \\ & a_k^h(x_k) + \nabla a_k^h(x_k)^T s = 0 \\ & \|s\| \leq \delta_k, \end{aligned}$$

Calculer la fonction mérite $P(x_k + s_k)$ et déterminer le quotient r ,

Mettre à jour δ_k et x_k en fonction de r ,

Tout comme pour le Lagrangien augmenté, Alexandrov propose ici une formulation intégrée plutôt qu'une formulation à plusieurs niveaux. La mise-à-jour de la pénalité pour la fonction mérite, du rayon de confiance et du point de design se font donc au cours de la même boucle.

Nous pouvons aussi remarquer dans cette méthode que, conformément à la méthode SQP, le sous-problème d'optimisation ne prend en compte qu'une approximation linéaire de chacune des approximations $a_k^g(x)$ et $a_k^h(x)$ ce qui revient à linéariser directement les fonctions $g_{hi}(x)$ et $h_{hi}(x)$ en x_k . Ainsi nous pouvons affirmer que cette formulation n'utilise aucune information sur les contraintes provenant du modèle basse fidélité. Le seul apport du modèle faible se fait au niveau de la fonction coût.

D'autre part, nous nous sommes éloignés de la méthode SQP parce que, dans le sous-problème d'optimisation, nous minimisons dorénavant une fonction $a_k^f(x)$ qui

n'est plus quadratique, ce qui nous empêche d'utiliser les méthodes classiques de résolution du sous-problème, qui se posait précédemment sous forme de système linéaire (voir Papalambros et Wilde (2000) et Conn *et al.* (2000)). Il n'est en effet plus question de construire un hessien au fur et à mesure des évaluations du gradient puisqu'à chaque évaluation, la fonction $a_k^f(x)$ est modifiée!

Néanmoins, cette formulation inspirée de la méthode SQP à rayon de confiance nous assure, pour peu que les fonctions $f_{hi}(x)$ et $f_{lo}(x)$ soient aussi régulières que les hypothèses de la méthode du rayon de confiance l'exigent (paragraphe 1.1.2), que la suite des points x_k converge vers un point de Karush Kuhn Tucker, ou point critique, du modèle haute fidélité (voir Alexandrov (1998)).

Cette dernière formulation qui ne peut plus être qualifiée de méthode SQP nous a servie de point de départ pour formuler notre propre algorithme.

1.3.3 Présentation de l'algorithme de l'AMF implanté

1.3.3.1 Algorithme de l'AMF

Quitte à nous éloigner de la formulation SQP à rayon de confiance, nous avons formulé notre propre algorithme, largement inspiré de celui d'Alexandrov, à ceci prêt que nous avons choisi une formulation à deux niveaux contrairement à la formulation intégrée présentée au paragraphe précédent.

Nous avons en effet préféré utiliser toute l'information que nous pouvions tirer du modèle faible en utilisant réellement les approximations $a_k^g(x)$ et $a_k^h(x)$.

Jusqu'à convergence :

Construire $a_k^f(x), a_k^g(x), a_k^h(x)$ respectant (1.8),

Résoudre le sous-problème d'optimisation :

$$\begin{aligned} \min_s \quad & a_k^f(x_k + s) \\ \text{t.q.} \quad & a_k^g(x_k + s) \geq 0 \\ & a_k^h(x_k + s) = 0 \\ & \|s\| \leq \delta_k, \end{aligned}$$

Calculer la fonction mérite $P(x_k + s_k)$ et déterminer le quotient r ,

Mettre à jour δ_k et x_k en fonction de r ,

Nous remarquons que le sous-problème d'optimisation se pose alors comme n'importe quel problème général d'optimisation sous contraintes. Nous pouvons alors adopter une approche modulaire qui consiste à faire résoudre le sous-problème d'optimisation par un optimiseur qui sera pour nous une boîte noire. Nous ne nous préoccupons pas de la méthode qu'il emploie pour converger. Cette vision a l'avantage de permettre l'incorporation de codes commerciaux sans aucune difficulté. Nous pourrions alors aussi changer complètement d'optimiseur selon la nature du problème que nous voulons traiter.

Nous constatons que nous n'utilisons plus un algorithme SQP à rayon de confiance même si nous en gardons la structure pour nous assurer toujours de la convergence et de la robustesse de la méthode.

Cet algorithme a cependant le désavantage, propre aux optimisations multi-niveaux, de résoudre le problème de sous-optimisation de manière poussée, ce qui permet certes de vérifier le critère FCD (1.5), voire FOD mais ce qui coûte relative-

ment cher en évaluations du modèle faible. Il faut alors veiller à ne pas trop reserrer les critères de convergence de cette sous-optimisation, surtout dans les premières boucles de la méthode à rayon de confiance.

Grâce à ce dernier algorithme nous avons l'assurance théorique de converger vers un point de Karush Kuhn Tucker du problème initial, résolu avec le modèle haute fidélité, et ce de manière robuste. Nous espérons diminuer ainsi le nombre d'appel au modèle fort nécessaire à la découverte de l'optimum.

1.3.3.2 Schéma de l'AMF

Le schéma 1.1 représente la structure du programme que nous avons écrit.

L'algorithme choisi nous permet de considérer l'optimiseur comme une boîte noire responsable de la résolution du sous-problème d'optimisation. Il reçoit donc en entrée les coordonnées de départ ainsi que les bornes des variables de design, il conduit ensuite l'optimisation avec le modèle faible corrigé et renvoie en sortie les coordonnées de l'optimum trouvé, ainsi que les multiplicateurs de Lagrange.

De la même façon, les solveurs peuvent être aussi considérés comme des boîtes noires qui reçoivent en entrée les variables de design et écrivent en sortie les valeurs de la fonction coût et des contraintes correspondantes.

Tout ceci nous a permis d'écrire un programme modulaire, utilisant des fichiers d'entrée et de sortie standards. Il suffit donc de remplacer une boîte par une autre selon que l'on désire changer de modèle fort, de modèle faible ou d'optimiseur. Le travail d'adaptation nécessaire pour un nouveau problème porte donc sur l'«enrobage» des solveurs ou de l'optimiseur pour qu'ils puissent traduire le fichier standard d'entrée et écrire le fichier standard de sortie, travail qui doit être effectué de toutes façons, quel que soit le problème d'optimisation que l'on désire résoudre. Ainsi, grâce à cette approche, l'algorithme de l'AMF ne change pas et doit juste

être réglé pour la nouvelle taille du problème.

La première étape effectuée par l'AMF consiste à calibrer la correction à apporter au modèle faible. Il s'agit de calculer la valeur de la fonction coût et de son gradient au point courant grâce aux deux solveurs disponibles. La comparaison des deux réponses permettra de construire la bêta correction (1.2.3). Une fois écrits les coefficients de cette bêta correction, l'AMF lance l'optimiseur à partir du point de design courant. Afin d'évaluer la fonction coût et les contraintes pour tous les points de design qu'il souhaite essayer, l'optimiseur utilise le modèle faible suivi de l'approximation qui modifie, par la bêta correction, les valeurs de sortie du modèle faible. Une fois cette boucle interne propre au sous-problème d'optimisation effectuée, l'optimiseur renvoie la solution à l'AMF qui vérifie s'il s'agit effectivement d'un meilleur point aux yeux du solveur haute fidélité. Si le point est accepté, la première étape est répétée pour le nouveau point de design, sinon l'optimisation est effectuée de nouveau avec un rayon de confiance plus restreint.

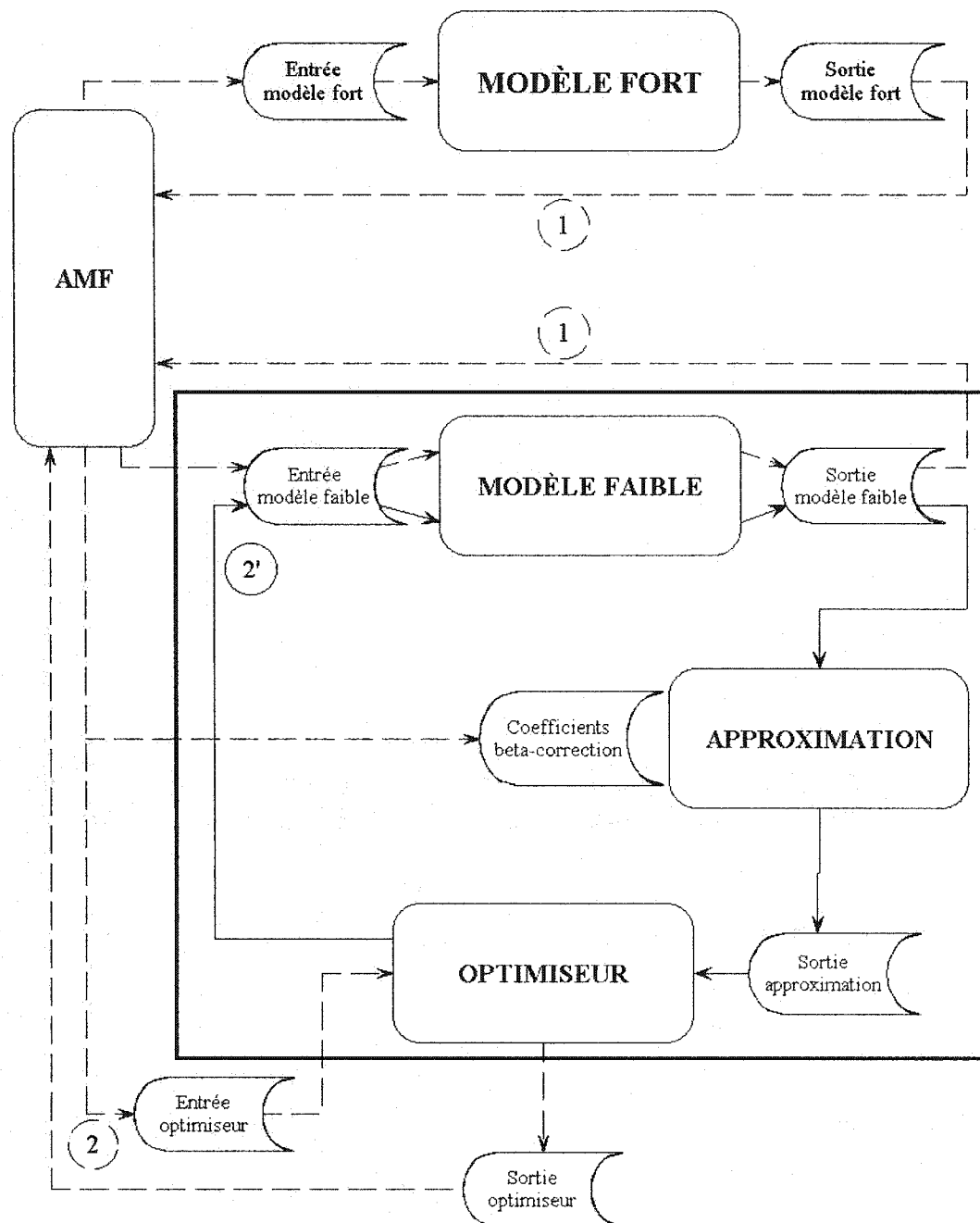


Figure 1.1 Schéma de l'AMF implanté

CHAPITRE 2

ETUDE SUR DES CAS ANALYTIQUES ET AMÉLIORATION DE LA BÊTA-CORRECTION

Afin de valider son comportement, nous avons testé notre programme sur des cas analytiques de référence. Ces problèmes présentent l'avantage d'être réguliers, rapides à calculer et de posséder des solutions bien répertoriées.

Se pose alors la question de la modélisation de la fidélité variable dans un contexte analytique. La fidélité variable est naturelle lorsqu'il s'agit d'évaluer la fonction objectif par l'intermédiaire de solveurs plus ou moins précis. Cependant, dans un problème analytique, la fonction coût est parfaitement connue. Comment construire alors un modèle fort et un modèle faible représentant avec plus ou moins de précision ce modèle analytique ?

Nous avons choisi de représenter le modèle haute fidélité par le problème analytique initial et exact. Ainsi nous connaissons parfaitement l'optimum vers lequel l'algorithme doit converger. Le modèle basse fidélité correspond, lui, à ce même problème analytique mais perturbé cette fois-ci par d'autres fonctions de manière à conserver une certaine ressemblance avec le problème initial tout en lui étant différent.

Cette formulation est incertaine parce qu'il est difficile de connaître à priori quelle est la différence entre un modèle fort et un modèle faible puisque cela est propre à chaque problème et à chaque solveur employé. Cependant, ces premiers essais nous ont permis de valider l'algorithme choisi et le programme implanté ainsi que d'observer son comportement.

2.1 Validation du comportement

Cette partie est à placer dans un contexte chronologique, elle retrace en effet nos premiers essais dans le but de vérifier le bon fonctionnement global de l'AMF ainsi que sa convergence vers l'optimum du modèle fort.

2.1.1 Vérification de la convergence

Les premiers essais ont été conduits sur les problèmes standards¹ suivants :

- Hock et Schittkowski numéro cinq (HS5) : Optimisation sur $\mathbb{R}^2$ sans contraintes.
- Fonction de Rosenbrock : Optimisation sur $\mathbb{R}^2$ sans contraintes.
- Hock et Schittkowski numéro sept (HS7) : Optimisation sur $\mathbb{R}^2$ avec une contrainte d'égalité.
- Hock et Schittkowski numéro quatorze (HS14) : Optimisation sur $\mathbb{R}^2$ avec une contrainte d'égalité et une contrainte d'inégalité.

Les deux premiers exemples ont bien convergé vers l'optimum du modèle fort quel que soit le modèle faible. En revanche, les deux derniers n'ont pas pu se rendre jusqu'à la solution. Après étude, nous avons constaté que les coefficients de la bêta correction propre à la contrainte d'égalité étaient énormes dans les deux cas. Il existe en effet un problème de conditionnement de cette correction lorsque le dénominateur $h_{lo}(x)$ tend vers zéro. Ce problème peut d'ailleurs survenir pour toutes les autres fonctions, contraintes ou objectif, dès lors que le dénominateur devient très petit devant le numérateur.

Pour remédier à cela nous avons décidé d'ajouter une constante dans la formulation de la bêta correction, qui ne modifie en rien les propriétés nécessaires à la

¹<http://www.princeton.edu/~rvdb/ampl/nlmodels/>

convergence, mais qui évite les problèmes de conditionnement. La nouvelle bêta correction est ainsi définie par

$$\beta(x) = \frac{f_{hi}(x) + K}{f_{lo}(x) + K},$$

où K est une constante. $a_k^f(x)$ est alors définie par

$$a_k^f(x) = \beta_k(x)(f_{lo}(x) + K) - K.$$

K est choisie assez grande par rapport à $f_{lo}(x)$. Typiquement, lorsque $f_{lo}(x)$ est de l'ordre de l'unité, nous avons pris $K = 1000$.

Cette légère modification nous a permis de constater que l'algorithme convergeait bien vers l'optimum du modèle fort dans tous nos cas de figure.

2.1.2 Apport de la bêta correction avec constante et première intuition sur la correction additive

Une fois la convergence vérifiée, notre attention s'est portée sur l'économie que représentait l'utilisation de l'AMF en terme d'appels au modèle haute fidélité. C'est là que nous avons remarqué l'effet bénéfique de la bêta correction avec constante sur l'efficacité de l'AMF. Nous présentons, pour illustrer notre propos, les résultats obtenus sur le modèle HS5, défini comme suit :

Le modèle fort correspond au véritable problème, soit

$$f_{hi}(x) = \sin(x + y) + (x - y)^2 - 1, 5x + 2, 5y + 1,$$

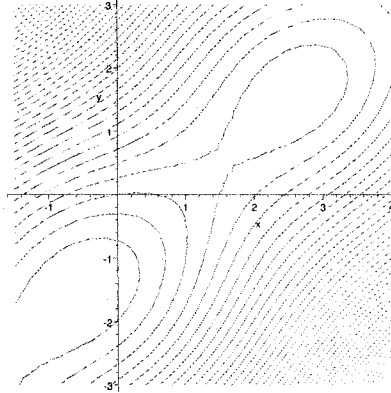


Figure 2.1 Modèle haute fidélité du problème HS5

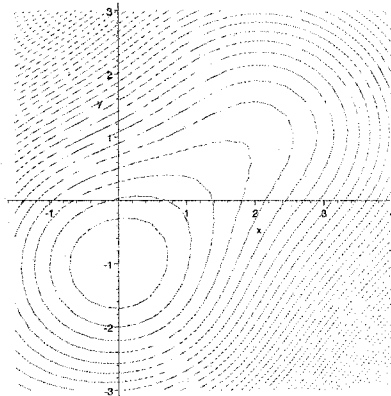


Figure 2.2 Modèle basse fidélité du problème HS5

tandis que le modèle faible est légèrement perturbé de la manière suivante :

$$f_{lo}(x) = \sin(x + y) + (x - y)^2 - 1,5x + 2,5y + 1 - (1 - x)y.$$

Les courbes de niveau des deux fonctions sont présentées sur les figures 2.1 et 2.2.

Nous présentons dans le tableau 2.1 le nombre d'appels au modèle fort et à son gradient requis par trois stratégies d'optimisation : l'optimisation directe sur le

modèle fort, l'utilisation de l'AMF avec bêta correction sans constante (β_0) et l'utilisation de l'AMF avec bêta correction et constante (β_{1000}).

Tableau 2.1 Comparaison du nombre d'appels au modèle haute fidélité pour le problème Hock et Schittkowski numéro 5

appels	directe	β_0	β_{1000}
f_{lo}	0	11578	95
∇f_{lo}	0	6325	73
f_{hi}	12	1000	14
∇f_{hi}	9	710	14

Nous constatons sur ce problème que l'introduction de la constante a considérablement réduit le nombre d'appels au modèle fort sans toutefois rendre l'AMF intéressant par rapport à une optimisation directe, pour ce cas particulier.

Pour comprendre ce gain, nous avons tracé les approximations issues des corrections avec et sans constante (figure 2.4 et 2.3). Nous pouvons constater que le modèle faible original est complètement déformé lorsqu'il n'y a pas de constante alors qu'avec, il est encore reconnaissable. En déformant moins le modèle faible, l'AMF peut utiliser plus d'information et converger plus vite vers l'optimum.

Cette faible déformation provient du fait, et nous le démontrerons au paragraphe 2.2.2.3, qu'une correction multiplicative comme la bêta correction ressemble à une correction additive quand on lui ajoute une forte constante. Or, comme nous le montrerons au paragraphe 2.2.2.2, nous pensons qu'une correction additive perturbe moins le modèle faible.

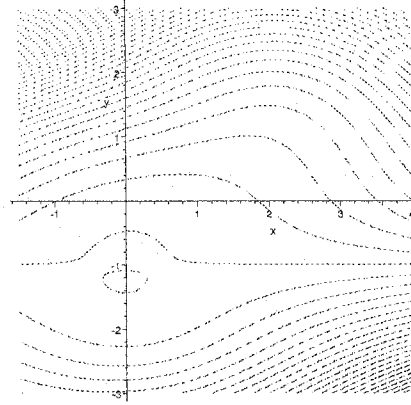


Figure 2.3 Modèle basse fidélité du problème HS5 après correction par la bêta correction sans constante

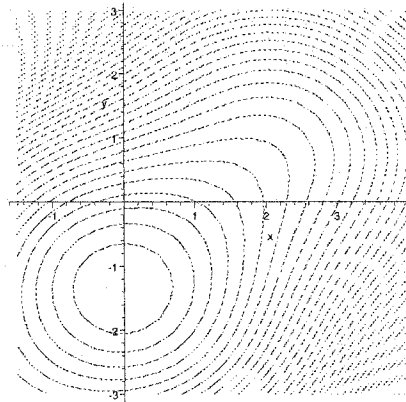


Figure 2.4 Modèle basse fidélité du problème HS5 après correction par la bêta correction avec constante

2.2 Problème de convergence et nouvelle correction

Une fois la bêta correction avec constante adoptée, nous avons pu comparer le nombre d'appels au modèle fort avec ou sans utilisation de l'AMF. Là encore, certains résultats nous dévoilèrent un taux de convergence de plus en plus lent sur la fin. Plus l'algorithme se rapproche de la solution et plus les pas sont petits. Le nombre de recalibrations sur le modèle fort est alors plus important que le

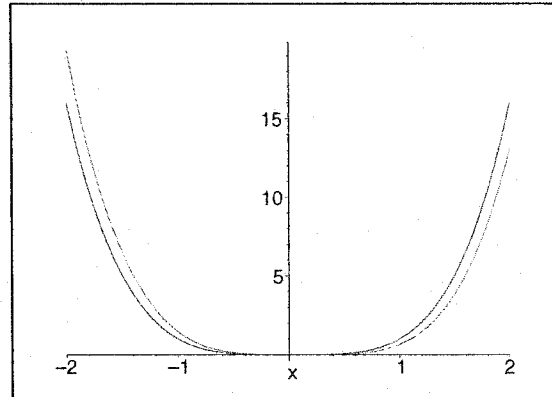


Figure 2.5 Modèle haute fidélité x^4 et modèle basse fidélité $(x - 0, 1)^4$

nombre d'appels au modèle fort requis par une optimisation directe ! Ce problème est détaillé et corrigé pour les exemples étudiés, dans cette partie.

2.2.1 Mise en évidence du problème de convergence

Afin d'illustrer la lenteur de convergence de l'algorithme, nous avons construit un exemple simple. Nous avons choisi comme modèle haute fidélité la fonction x^4 et comme modèle basse fidélité la fonction $(x - 0, 1)^4$ représentées sur la figure 2.5. Le point de départ de l'optimisation est situé en $x = 2$.

Nous n'utilisons dorénavant que la bêta correction avec constante. Nous avons représenté sur les figures 2.6, 2.7 et 2.8 la séquence d'approximations effectuées par l'AMF et issues de la correction du modèle faible. A chaque itération, l'optimiseur trouve le minimum de l'approximation, l'AMF l'accepte et recalibre la bêta correction au nouveau point. Cependant les progrès sont de plus en plus petits (nous n'avons même pas un rythme de convergence linéaire), si bien que le nombre d'appels au modèle fort devient plus important que si l'on avait effectué l'optimisation directement sur le modèle fort !

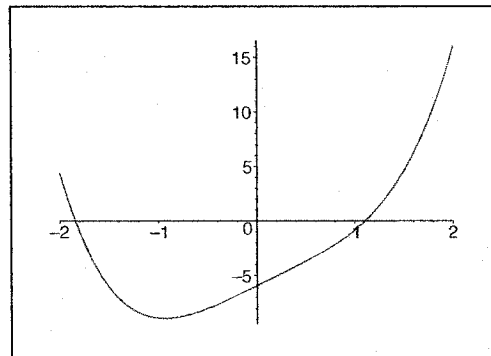


Figure 2.6 Modèle faible $(x - 0, 1)^4$ corrigé par une bêta correction linéaire à la première itération

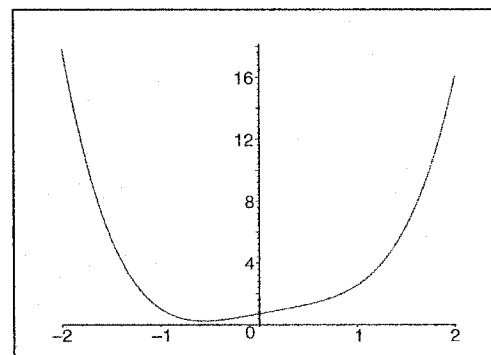


Figure 2.7 Modèle faible $(x - 0, 1)^4$ corrigé par une bêta correction linéaire à la seconde itération

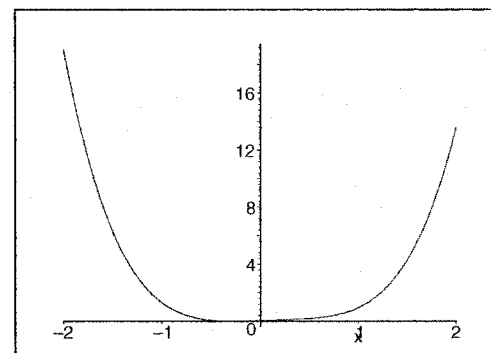


Figure 2.8 Modèle faible $(x - 0, 1)^4$ corrigé par une bêta correction linéaire à la troisième itération

2.2.2 Améliorations de la bêta correction

Deux stratégies d'amélioration de la bêta correction ont alors été envisagées :

- La première consiste à utiliser les valeurs précédentes des modèles fort et faible pour que l'approximation soit égale au modèle fort au premier ordre non seulement au meilleur point actuel (x_k) mais aussi aux anciens points. Nous espérons améliorer la qualité de l'approximation et ainsi diminuer le nombre d'appels au modèle fort.
- La seconde consiste à utiliser une correction non plus multiplicative mais additive, comme nous l'évoquions au paragraphe 2.1.2, en espérant ainsi déformer encore moins le modèle faible.

Evidemment toutes ces nouvelles formulations vérifient toujours l'équation 1.8 sans laquelle nous ne bénéficierions pas des propriétés de convergence de l'AMF.

2.2.2.1 Bêta correction d'un ordre supérieur

Cette bêta correction demeure une expression polynomiale qui n'est plus forcément linéaire mais qui peut maintenant être quadratique ou cubique :

$$\beta_k(x) = \beta(x_k) + \nabla^T \beta(x_k)(x - x_k) + (x - x_k)^T H_k^\beta (x - x_k) \dots$$

Les termes linéaires sont toujours évalués de la même manière afin de vérifier l'équation 1.8. Les termes d'ordre supérieur sont calculés de manière à correspondre aux précédentes valeurs de $\beta(x_k)$ et de $\nabla^T \beta(x_k)$. Plus précisément, nous approximons la fonction $\beta(x)$, qui nous est inconnue, par un polynôme d'interpolation d'ordre 2 ou 3 construit à partir des valeurs connues de $\beta(x_k)$ et de $\nabla^T \beta(x_k)$.

Pour cela nous résolvons un système linéaire. Lorsque le système est mal condi-

tionné nous diminuons l'ordre de la bêta correction en supprimant progressivement les différents termes du polynôme. Au fur et à mesure des itérations, si le système devient surdéterminé, nous résolvons un sous-système pour passer le plus proche possible des anciennes valeurs de $\beta(x)$ au sens des moindres carrés. Enfin, lorsque trop d'anciennes valeurs sont accumulées, nous les supprimons de notre base de données.

2.2.2.2 Delta correction

On définit la fonction $\delta(x)$ par l'équation suivante :

$$\delta(x) = f_{hi}(x) - f_{lo}(x).$$

A chaque itération x_k , une approximation linéaire de $\delta(x)$ est construite :

$$\delta_k(x) = (f_{hi}(x_k) - f_{lo}(x_k)) + (\nabla^T f_{hi}(x_k) - \nabla^T f_{lo}(x_k))(x - x_k).$$

On définit alors l'approximation $a_k^f(x)$ de la manière suivante :

$$a_k^f(x) = f_{lo}(x) + \delta_k(x).$$

Le lecteur assidu pourra constater que l'approximation ainsi définie vérifie bien les conditions de l'équation 1.8.

A l'instar de la bêta correction, la delta correction peut être d'un ordre plus élevé si l'on souhaite tenir compte des valeur précédentes de $\delta(x_k)$. Les coefficients du

polynôme sont calculés de la même façon que pour la bêta correction.

Corriger le modèle faible par addition plutôt que par multiplication nous a paru plus intuitif. Nous espérons ainsi garder de fortes ressemblances entre l'approximation et le modèle faible.

Plus concrètement, quelle que soit la forme du modèle faible, ce dernier peut toujours être correctement représenté par un polynôme, d'un ordre plus ou moins élevé selon la complexité de la fonction. Ainsi, en le multipliant par une fonction linéaire, quadratique ou cubique, on ne fait qu'augmenter son «ordre» et éventuellement le nombre de ses minima locaux. En revanche, lui ajouter une fonction linéaire, quadratique ou cubique n'augmente pas son ordre, sauf si le modèle faible est moins complexe (d'un ordre moindre) que la correction qu'on lui applique. Par exemple, si nous sommes à proximité d'un bassin de convergence de forme quadratique du modèle faible, une bêta correction quadratique produira une approximation quartique alors qu'une delta correction quadratique produira une approximation quadratique. La delta correction semble donc, à priori, moins perturbatrice.

De plus, l'allure globale de l'approximation et en particulier les directions de descente sont dictées par ses dérivées :

Si la bêta correction construit une approximation $(a + b(x - x_k))f_{lo}(x - x_k)$, sa dérivée est égale à $b f_{lo}(x - x_k) + f'_{lo}(x - x_k)(a + b(x - x_k))$. On est donc assez loin de la dérivée de départ $f'_{lo}(x - x_k)$, à fortiori lorsqu'on s'éloigne du point courant x_k et que le terme $b(x - x_k)$ devient important.

Inversement la delta correction $a + b(x - x_k) + f_{lo}(x - x_k)$ aura une dérivée égale à $f'_{lo}(x - x_k) + b$, qui non seulement est plus proche de la dérivée originale mais en plus ne se déforme pas plus à mesure que l'on s'éloigne du point courant, d'où le respect de l'allure globale du modèle faible.

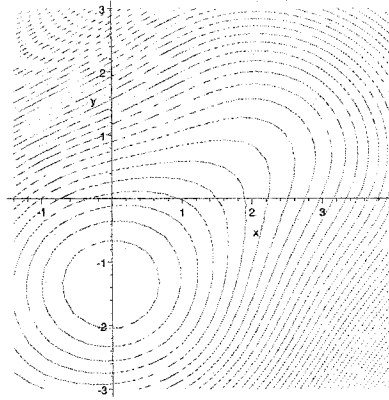


Figure 2.9 Modèle basse fidélité du problème HS5 après correction par la delta correction

Ces considérations qualitatives peuvent être illustrées par la figure 2.9 qui reprend le problème de Hock et Schittkowski numéro cinq. Nous y avons représenté le modèle faible déformé par la delta correction à la première itération. Nous pouvons constater que le modèle basse fidélité est beaucoup moins déformé qu'avec la bêta correction originale. De plus, nous illustrons ainsi la très forte ressemblance entre la bêta correction avec une forte constante et la delta correction puisque cette figure ressemble de très près à la figure 2.4. Nous allons d'ailleurs le démontrer dans le prochain paragraphe.

2.2.2.3 Démonstration de l'équivalence entre une bêta correction avec une forte constante et une delta correction

Pour cette démonstration, nous partirons de l'hypothèse que $\frac{f_{lo}(x)}{K}$ et $\frac{f_{hi}(x)}{K}$ sont très petits devant 1. Nous évaluons alors $a_k^f(x)$ en négligeant les termes de l'ordre de $\frac{f}{K}$.

$$a_k^f(x) = \left(\frac{f_{hi}(x_k) + K}{f_{lo}(x_k) + K} + \frac{\nabla f_{hi}(x_k)(f_{lo}(x_k) + K) - \nabla f_{lo}(x_k)(f_{hi}(x_k) + K)}{(f_{lo}(x_k) + K)^2} (x - x_k) \right) (f_{lo}(x) + K) - K$$

$$a_k^f(x) \simeq \left(K \frac{1 + \frac{f_{hi}(x_k)}{K}}{1 + \frac{f_{lo}(x_k)}{K}} + \frac{\nabla f_{hi}(x_k) \left(1 + \frac{f_{lo}(x_k)}{K}\right) - \nabla f_{lo}(x_k) \left(1 + \frac{f_{hi}(x_k)}{K}\right)}{\left(1 + \frac{f_{lo}(x_k)}{K}\right)^2} (x - x_k) \right) \left(1 + \frac{f_{lo}(x)}{K}\right) - K$$

$$a_k^f(x) \simeq K \left(1 + \frac{f_{hi}(x_k)}{K} - \frac{f_{lo}(x_k)}{K} + \frac{f_{lo}(x)}{K}\right) - K + (x - x_k) \times \\ \left(\nabla f_{hi}(x_k) \left(1 + \frac{f_{lo}(x_k)}{K} - 2 \frac{f_{lo}(x_k)}{K} + \frac{f_{lo}(x)}{K}\right) - \nabla f_{lo}(x_k) \left(1 + \frac{f_{hi}(x_k)}{K} - 2 \frac{f_{lo}(x_k)}{K} + \frac{f_{lo}(x)}{K}\right) \right)$$

Ainsi, à l'ordre 0 :

$$a_k^f(x) \simeq f_{lo}(x) + \left(f_{hi}(x_k) - f_{lo}(x_k)\right) + (x - x_k) \left(\nabla f_{hi}(x_k) - \nabla f_{lo}(x_k)\right)$$

On reconnait alors l'expression de la delta correction.

2.3 Résultats analytiques obtenus avec les corrections améliorées

Nous avons évalué l'apport de la delta et de la bêta correction à des ordres plus ou moins élevés sur trois cas analytiques. Les résultats sont développés dans les trois prochains paragraphes.

2.3.1 Fonctions x^4 et $(x - 0, 1)^4$

La première analyse a été effectuée sur le cas qui nous a permis d'illustrer les problèmes de convergence à la section 2.2.1.

Nous avons représenté les approximations successives produites par la bêta correction au cours de l'optimisation. Afin d'observer l'apport de la bêta correction à

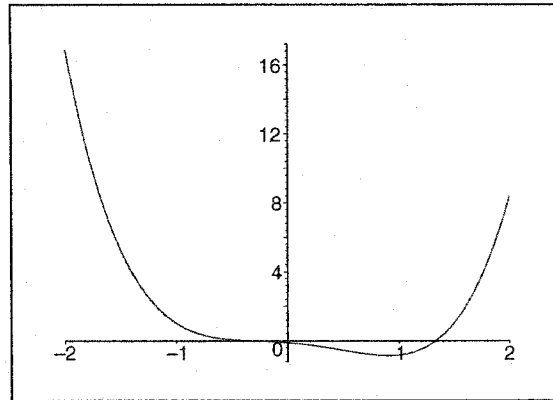


Figure 2.10 Modèle faible $(x - 0, 1)^4$ corrigé par une bêta correction quadratique à la troisième itération

un ordre supérieur, nous avons tracé (figure 2.10) l'approximation correspondant à la troisième itération de l'AMF lorsqu'une bêta correction quadratique est utilisée. Nous pouvons constater que la forme du modèle faible a malheureusement été considérablement modifiée par la bêta correction quadratique. Cependant, cette nouvelle forme force l'optimiseur à se rendre dans la demi-droite définie par les x positifs et grâce à quelques adaptations de rayon, elle évite la lente succession de minima locaux observée précédemment. Le nombre d'appels au modèle basse fidélité est alors moindre que pour une bêta correction linéaire. Cependant, nous espérons mieux représenter la bassin de convergence avec une correction d'un ordre supérieur, ce n'est pas du tout le cas ici.

L'apport d'une delta correction cubique est illustré par la figure 2.11 qui représente l'approximation correspondant à la troisième itération de l'AMF lorsqu'une delta correction cubique est utilisée. Nous pouvons observer sur cet exemple que le modèle faible est non seulement peu déformé puisque sa forme globale est conservée dans l'approximation mais qu'en plus cette dernière représente bien le modèle fort. Dans ce cas, une delta correction cubique est bien plus efficace qu'une bêta correction

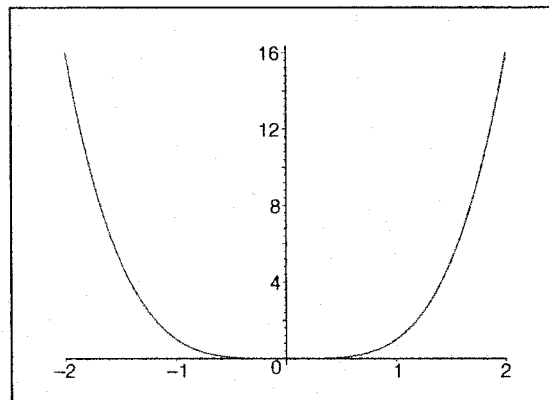


Figure 2.11 Modèle faible $(x - 0, 1)^4$ corrigé par une delta correction cubique à la troisième itération

linéaire.

Tous les résultats obtenus sur ce cas analytique sont répertoriés dans le tableau 2.2. Notre point de comparaison pour toutes ces expériences est le nombre d'appels au solveur haute fidélité requis par une optimisation directe. Nous pouvons constater que pour les deux types de correction, plus l'ordre est élevé, moins l'AMF a besoin de consulter le modèle haute fidélité. De plus, sur cet exemple, la delta correction linéaire est pire que la bêta correction du même ordre mais à l'ordre 3 la tendance s'inverse. Nous remarquons enfin que l'AMF utilisé avec une delta correction cubique permet de résoudre le problème avec moins d'appels au modèle haute fidélité qu'avec une optimisation directe, ce qui est le but recherché.

Tableau 2.2 Comparaison du nombre d'appels au modèle haute fidélité pour une minimisation de x^4

appels	directe	β^1	β^2	β^3	δ^1	δ^2	δ^3
f_{lo}	0	200	162	66	590	183	35
∇f_{lo}	0	190	70	34	406	76	28
f_{hi}	19	40	16	4	100	21	3
∇f_{hi}	5	40	11	5	101	12	4

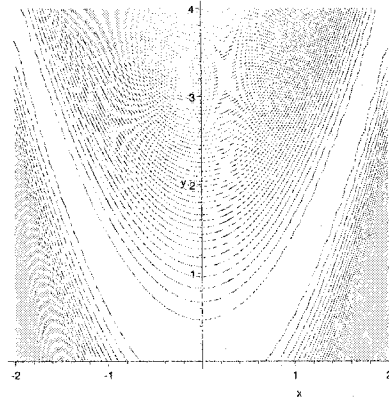


Figure 2.12 Courbes de niveau de la fonction de Rosenbrock

2.3.2 Fonction de Rosenbrock

Dans cet exemple, le rôle du modèle haute fidélité est rempli par la fonction de Rosenbrock représentée sur la figure 2.12. Cette fonction est définie par

$$f_{hi}(x, y) = 100(y - x^2)^2 + (1 - x)^2,$$

et possède un minimum en $(1; 1)$. Le modèle faible correspondant a été construit en décalant la vallée et le minimum qui s'y trouve. Il correspond à la fonction

$$f_{lo}(x, y) = 100\left((y + 0,1) - x^2\right)^2 + (1,5 - x)^2$$

qui possède un minimum en $(1,5; 2,15)$.

La comparaison des stratégies d'optimisation est détaillée sur le tableau 2.3. Nous avons ajouté une stratégie séquentielle $(f_{lo} + f_{hi})$ qui consiste à faire courir le modèle basse fidélité puis le modèle haute fidélité.

Pour la delta et la bêta correction, le passage du linéaire au quadratique est très bénéfique en terme d'appels au modèle fort. En revanche le passage du quadratique au cubique est pénalisant pour la bêta correction. En ce qui concerne la delta correction, nous n'avons pas assez d'itérations pour créer une véritable delta correction cubique, elle ne monte en effet qu'à l'ordre 2. De manière beaucoup plus significative que pour l'exemple précédent, la delta correction devient plus intéressante que la bêta correction lorsque l'on augmente l'ordre. Nous remarquons enfin qu'à partir de l'ordre 2, la bêta et la delta correction requièrent respectivement trois et neuf fois moins d'appels au modèle fort qu'une optimisation directe.

Toutefois il est très important de noter ici que la perturbation que nous avons introduite dans le modèle faible est additive et quadratique donc parfaitement représentable par la delta correction quadratique. Cette dernière est donc avantagée artificiellement. Nous avons donc plus de foi dans les résultats de la bêta correction qui montrent eux aussi une nette amélioration au passage à l'ordre 2.

Nous remarquons enfin que l'optimisation séquentielle est moins efficace que l'AMF utilisé avec des corrections d'ordre 2 ou 3.

Tableau 2.3 Comparaison du nombre d'appels au modèle haute fidélité pour la fonction de Rosenbrock

appels	directe	$f_{lo} + f_{hi}$	β^1	β^2	β^3	δ^1	δ^2	δ^3
f_{lo}	0	79	972	338	431	1041	82	82
∇f_{lo}	0	36	431	148	194	469	39	39
f_{hi}	72	41	61	16	21	68	2	2
∇f_{hi}	27	19	44	9	13	49	3	3

2.3.3 Problème de Hock et Schittkowski numéro 100

Il s'agit d'une optimisation sur $\mathbb{R}^7$ avec 4 contraintes d'inégalité. Le problème initial remplit le rôle de modèle haute fidélité et est décrit par les fonctions

$$\begin{aligned}
 f_{hi}(x) &= (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 \\
 &\quad + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \\
 g_{hi}^1(x) &= 127 - 2x_1^2 - 3x_2^4 - x_3 - 4x_4^2 - 5x_5 \\
 g_{hi}^2(x) &= 282 - 7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5 \\
 g_{hi}^3(x) &= 196 - 23x_1 - x_2^2 - 6x_6^2 + 8x_7 \\
 g_{hi}^4(x) &= -4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^2 - 5x_6 + 11x_7,
 \end{aligned}$$

dont le minimum est situé en (2,33 ; 1,95 ; -0,48 ; 4,37 ; -0,62 ; 1,04 ; 1,59).

Ce modèle a été perturbé afin de construire un modèle basse fidélité qui s'y rapporte :

$$\begin{aligned}
 f_{lo}(x) &= 7(x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 \\
 &\quad + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 - 4\mathbf{x}_1\mathbf{x}_3\mathbf{x}_7 \\
 g_{lo}^1(x) &= 127 - 2x_1^2 + 5x_2^4 - x_3 - 4x_4^2 - 5x_5 \\
 g_{lo}^2(x) &= 282 - 7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5 - 5\mathbf{x}_2\mathbf{x}_4\mathbf{x}_6 \\
 g_{lo}^3(x) &= 196 - 23x_1 - x_2^2 - 6x_6^2 + 8x_7 + 4(1 - \mathbf{x}_4)(3 - 7\mathbf{x}_5) \\
 g_{lo}^4(x) &= -4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^2 - 5x_6 + 11x_7,
 \end{aligned}$$

dont le minimum est situé en (3,33 ; 5 ; 1,60 ; 5 ; 0 ; 0,52 ; 2,45).

Le lecteur attentif aura remarqué que la perturbation est additive et cubique mais nous n'aurons jamais assez d'itération pour monter jusqu'à cet ordre avec la delta correction.

Les résultats obtenus avec les différentes corrections sur cet exemple sont consignés dans le tableau 2.4. Là encore le passage d'une correction linéaire à une correction quadratique est très bénéfique mais celui d'une correction quadratique à une correction cubique n'est pas intéressant. Bêta et delta correction sont équivalentes et permettent une économie sur le nombre d'appels au modèle fort dès que leur ordre est supérieur ou égal à 2.

Tableau 2.4 Comparaison du nombre d'appels au modèle haute fidélité pour le problème Hock et Schittkowski numéro 100

appels	directe	$f_{lo} + f_{hi}$	β^1	β^2	β^3	δ^1	δ^2	δ^3
f_{lo}	0	14	972	349	606	1041	104	104
∇f_{lo}	0	12	431	156	227	469	52	52
f_{hi}	64	41	61	15	26	68	5	5
∇f_{hi}	28	19	44	10	12	49	5	5

2.4 Conclusions sur l'intérêt des nouvelles corrections

Le lecteur doit garder à l'esprit que les problèmes analytiques que nous avons résolus n'ont à priori aucun rapport avec les solveurs que nous rencontrerions dans des problèmes réels. Il est difficile d'inventer un modèle faible à partir d'un modèle fort puisque nous ne savons pas ce qui les différencie. De plus, les perturbations introduites dans les modèles faibles étant des polynômes semblables aux corrections, nous pourrions avoir généré des cas adaptés à l'AMF mais qui ne se présentent pas dans la réalité. Cette précaution prise, nous pouvons tout de même déduire un certain nombre de tendances des expériences analytiques conduites.

- La première concerne l'efficacité de la bêta correction avec constante. Non seulement elle a permis d'éviter les problèmes de conditionnement de la bêta correction qui empêchaient l'algorithme de converger mais en plus elle a diminué le nombre d'appels au modèle fort utilisés par l'AMF pour converger. Cette constatation a été confirmée dans l'immense majorité des expériences conduites si bien que l'utilisation de la bêta correction avec constante s'est généralisée dans toutes nos études, le terme «bêta correction» signifiant dorénavant implicitement qu'elle est calculée avec constante.
- La seconde observation porte sur l'influence de l'ordre de la correction. L'AMF devient plus efficace en augmentant l'ordre de la correction utilisée. Le nombre d'appels au modèle fort est sensiblement diminué et devient intéressant par rapport à une optimisation directe à partir de l'ordre 2 dans les exemples présentés. Toutefois le passage de l'ordre 2 à l'ordre 3 n'est pas toujours de bon aloi et lorsque qu'il apporte une amélioration, cette dernière n'est pas très significative.
- La dernière tendance porte sur la comparaison entre la bêta correction avec constante et la delta correction. Il semble qu'à l'ordre 1 la bêta correction est de meilleure qualité que la delta correction puisque l'AMF appelle moins souvent le modèle fort. Cette tendance semble s'inverser en augmentant l'ordre.

Ces expériences ne constituent en rien des démonstrations mais nous ont permis de réfléchir aux améliorations à apporter à l'AMF ainsi qu'aux problèmes qui pouvaient survenir.

Puisque nous bloquons toujours sur la pertinence des modèles choisis, nous allons maintenant aborder notre principal objectif en étudiant l'apport de l'AMF sur des cas réels d'optimisation aérodynamique de profils d'aile d'avion.

CHAPITRE 3

RÉSULTATS SUR UN CAS D'OPTIMISATION AÉRODYNAMIQUE DE PROFIL D'AILE

3.1 Description de l'expérience et de la formulation du problème d'optimisation

Nous avons cherché à optimiser la forme d'un profil d'aile. Pour cela, nous nous sommes dotés d'une paramétrisation, nous avons évalué la qualité de nos profils grâce à différents résolveurs de précision variable et nous avons relié le tout à un optimiseur de type SQP.

3.1.1 Paramétrisation

La paramétrisation d'un profil consiste à définir une méthode capable d'associer à un ensemble de nombre réels une géométrie. Elle doit être choisie en fonction :

- de l'espace de design qu'elle représente, c'est-à-dire de l'ensemble des profils qu'elle est capable de générer en modifiant les paramètres. Idéalement une bonne paramétrisation doit pouvoir représenter la plupart des profils déjà existant et éviter de produire des profils extravagants.
- du nombre de paramètres (ou variables de design) qu'elle requiert. Plus la paramétrisation nécessite un grand nombre de variables de design, plus l'optimisation sera coûteuse.
- de la complexité de la relation qui existe entre la fonction coût que l'on va choisir

de minimiser et les variables de design. Si l'influence des variables de design sur la fonction coût est obscure et si le couplage entre les variables de design est très important, l'optimisation nécessitera un grand nombre d'itérations.

Nous avons choisi de représenter notre profil d'aile par une NURBS (Non-Uniform Rational B-Spline), fonction mathématique capable de générer une courbe régulière. Cette courbe est déterminée par un vecteur nodal, définissant le rythme de parcours de la courbe, ainsi que par un ensemble de points de contrôle.

Ces derniers sont définis chacun par leurs coordonnées dans le plan (X,Y) et par un poids W qui détermine l'influence de ce point sur la courbe. Ainsi, la courbe est incluse dans le polygone convexe qui relie les points de contrôle et elle passe d'autant plus près d'un point de contrôle que le poids de ce dernier est important.

Lépine (1999) a montré dans son mémoire la possibilité de représenter la grande majorité des profils courants avec la précision de manufacture en utilisant seulement une quinzaine de points de contrôle, soit 45 variables de design.

Les NURBS ont cependant le désavantage d'être capable de générer des profils complètement abérants. L'espace de design est tellement vaste que de nombreuses formes géométriques qui n'ont aucun rapport avec un profil d'aile peuvent être produites. Il faut donc être très vigilant quant aux bornes qui s'appliquent aux variables de design.

Pour conduire notre expérience, nous avons choisi comme profil initial un profil RAE2822. Ce profil est représenté par une NURBS déterminée par quinze points de contrôle (figure 3.1). Afin de limiter la complexité du problème et pour éviter de générer des formes extravagantes, nous avons sélectionné en guise de variables de design les coordonnées Y des 6 points de contrôle situés sur l'extrados. Ces paramètres nous permettent de modifier sensiblement le profil sur une zone qui est très influente aérodynamiquement à cause du choc qui s'y produit en régime

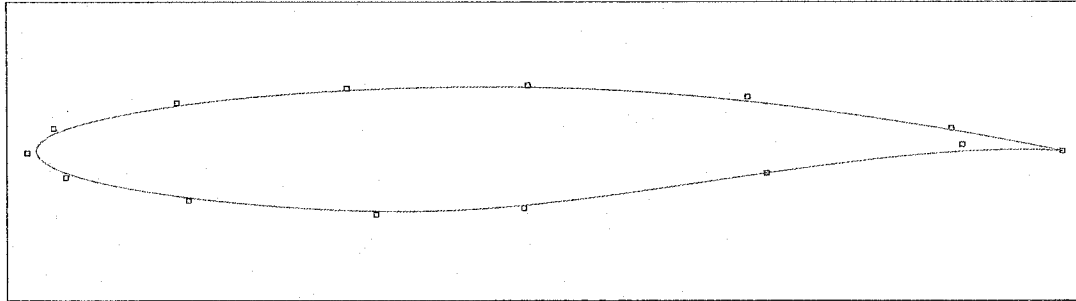


Figure 3.1 Représentation d'un profil RAE2822 par une NURBS déterminée par quinze points de contrôle

transsonique. Nous avons ajouté une septième variable de design qui correspond à l'angle d'attaque.

Nous verrons lors de la présentation des résultats expérimentaux l'opportunité d'utiliser une telle paramétrisation pour chacune des fonctions coût que nous avons utilisée.

3.1.2 Résolveurs

Afin d'analyser les propriétés aérodynamiques des profils générés nous avons utilisé un certain nombre de solveurs. Nous les présentons ici par ordre de qualité croissante.

3.1.2.1 BGK

Ce programme résout les équations du potentiel avec l'hypothèse des petites perturbations et ajoute une correction pour tenir compte de l'effet de la couche limite. Utilisant un maillage très grossier, le calcul prend environ deux secondes à être complété. Pour plus de précision sur le solveur, le lecteur peut se référer à l'article

de Bauer *et al.* (1977). Même si sa qualité est grossière, ce solveur permet de dégager les grandes tendances pour améliorer la fonction coût. Historiquement, des profils ont été optimisés avec ce solveur. Il constitue pour nous un modèle basse fidélité.

3.1.2.2 NSC2KE

Ce programme a été développé par l'INRIA (voir Mohammadi (1994) et Mohammadi et Pironneau (1994)) et résoud dans notre cas les équations d'Euler sur un maillage d'environ 2500 noeuds en 4 ou 5 minutes.

Nous avons développé pour ce solveur un maillage mixte (figure 3.2), structuré proche du profil et non structuré lorsqu'on en est éloigné pour limiter le nombre de noeuds. Ce choix nous a permis d'obtenir de très bons gradients. En effet, proche du profil, le maillage suit les déformations que subit la géométrie et est donc très légèrement déplacé lorsqu'on modifie légèrement la forme du profil pour calculer les dérivées partielles par différences finies. Ce n'est pas le cas lorsque, par exemple, nous utilisons un mailleur automatique non structuré qui peut changer considérablement le maillage même si on modifie très peu le profil.

Ainsi, l'analyse de sensibilité conduite sur NSC2KE nous a montré une grande constance des dérivées partielles pour une large plage de pas de différenciation. Ce solveur est donc de bonne qualité et indique des directions de descente assez fiables pour l'optimiseur. Il nous a permis aussi, grâce à son maillage structuré, d'effectuer lors de nos calculs de gradient des départs à chaud, c'est-à-dire prenant comme solution initiale de l'écoulement la solution de l'itération précédente. Il constitue pour nous un modèle basse fidélité.

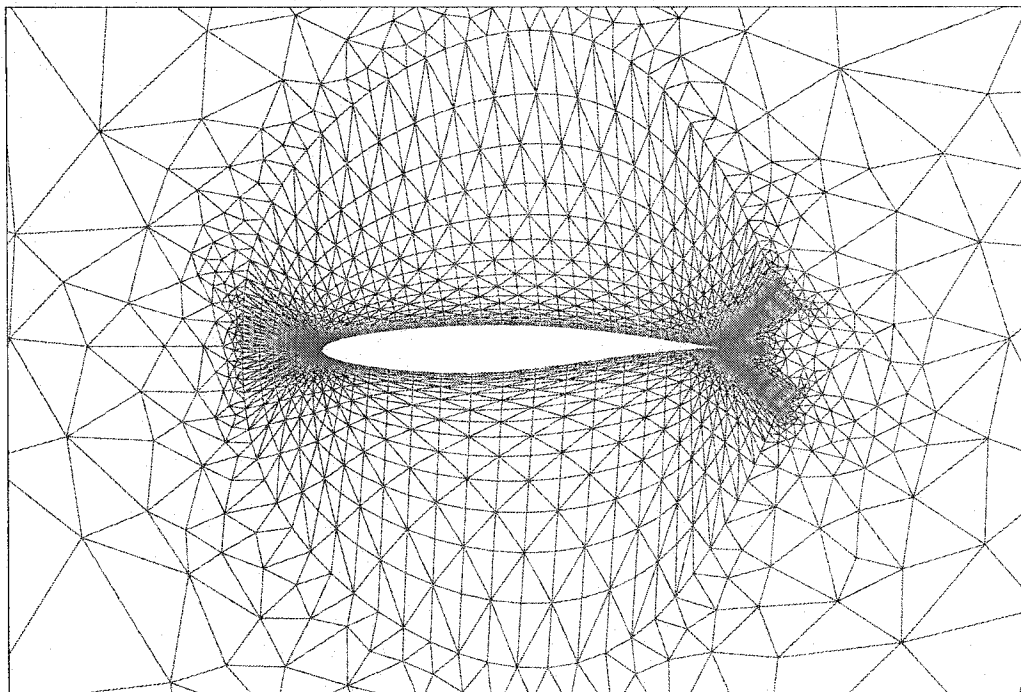


Figure 3.2 Maillage mixte développé pour le solveur NSC2KE

3.1.2.3 NSU2D

Ce solveur commercial développé par Mavriplis¹ résout les équations de Navier-Stokes avec un modèle de turbulence sur un maillage de 20000 noeuds en 20-25 minutes. Comme le maillage est non structuré et généré automatiquement, nous obtenons des dérivées moins stables qu'avec NSC2KE et nous n'avons pas la possibilité d'effectuer des départs à chaud. Ce solveur de haute fidélité est utilisé actuellement dans l'industrie et produit des résultats fiables lorsqu'il est inséré dans une boucle d'optimisation. Il constitue notre modèle haute fidélité.

¹Voir [http ://www.icas.edu/~dimitri](http://www.icas.edu/~dimitri)

3.1.3 Optimiseur

L'AMF a été développé dans le cas où des gradients fiables peuvent être obtenus avec le modèle fort et le modèle faible. Nous avons donc voulu profiter de la possibilité d'obtenir ces gradients avec notre optimiseur. C'est pourquoi nous avons choisi une méthode à gradient de type SQP (pour Sequential Quadratic Programming) développée par Spelluci (1998). Ce programme, DONLP2, est identique à celui utilisé pour nos cas analytiques. Il est à noter que même si la correction se base sur les gradients, la structure de notre AMF peut supporter un optimiseur quelconque. Nous avons par exemple effectué un certain nombre de validations avec l'optimiseur NOMAD de type «pattern search» développé par Audet et Dennis (2000).

3.1.4 Fonctions coût sélectionnées

Nous avons choisi trois types de formulation d'un problème d'optimisation aérodynamique.

3.1.4.1 Maximisation de la finesse de l'aile

C'est la formulation la plus directe en aérodynamique puisque qu'elle vise à maximiser la finesse globale de l'aile soit le quotient de la portance par la traînée. Pour un profil d'aile en deux dimensions cela se ramène à maximiser le rapport du coefficient de portance sur le coefficient de traînée. Cette problématique est parfois appelée «optimisation directe» par les aérodynamiciens puisque qu'elle exprime directement le souhait de ces derniers : minimiser la traînée et augmenter la portance de l'aile. La fonction objectif à minimiser est donc :

$$f = -\frac{C_l}{C_d},$$

où C_l est le coefficient de portance (lift) et C_d le coefficient de traînée (drag).

L'un des désavantages de cette formulation toute intégrée est qu'elle amène souvent le profil à être considérablement aminci pour diminuer la traînée, ce qui diminue aussi la portance mais dans une proportion moindre pour toujours améliorer la fonction coût. Or, avoir une faible portance sur un profil n'est pas très intéressant puisqu'il y a toujours un fuselage au bout de l'aile qu'il faut bien soulever si l'on souhaite décoller une fois arrivé au bout de la piste...

C'est pour cela que le problème d'optimisation aérodynamique est souvent posé avec une contrainte.

3.1.4.2 Minimisation de la traînée à portance constante

Ce problème se rapproche le plus de ce que l'on demande à un profil : être capable de soulever l'avion tout en minimisant la traînée. Les ingénieurs au cours du processus de design ont évalué de manière précise le poids de l'avion et les conditions de vol ce qui leur fixe un coefficient de portance. Le problème s'énonce alors ainsi :

$$\begin{aligned} \min_x \quad & f(x) = C_d \\ \text{t.q.} \quad & C_l \geq C_{lmin} \end{aligned}$$

Dans notre cas le C_{lmin} que nous avons choisi correspond au C_l initial du profil RAE2822 que nous améliorons. Cette formulation produit souvent des profils plus réalistes que la première. Elle a aussi l'avantage dans notre cas de s'exprimer avec une contrainte ce qui nous permettra de valider l'architecture de l'AMF.

Nous ferons remarquer que ces deux premières formulations, directes puisqu'elles expriment explicitement le souhait de l'ingénieur aérodynamicien, produisent une fonction coût complexe et fortement non-linéaire lorsqu'on utilise une paramétrisation de type NURBS. Le couplage entre les variables de design est très fort et le lien entre le déplacement d'un point de contrôle et l'évolution de la fonction coût est bien obscur.

Ayant remarqué ces difficultés, les ingénieurs ont souvent recours à une formulation indirecte, plus difficile à exprimer, mais qui a l'avantage d'être mieux adaptée à l'optimiseur.

3.1.4.3 Optimisation avec courbe de C_p cible

Cette formulation consiste à définir une courbe de C_p idéale dont nous voudrions trouver le profil correspondant. Il s'agit donc pour chaque configuration testée de tracer la courbe de C_p du profil et de calculer une «distance» entre cette courbe et la courbe cible. Dans notre cas, cette distance est définie par l'aire du polygone délimité par les deux lignes brisées qui relient les points des deux courbes C_p (figure 3.3).

Cette formulation, souvent utilisée dans l'industrie, a l'avantage de produire une fonction coût plus «intuitive» pour l'optimiseur. En déplaçant un point de contrôle de l'extrados vers le haut, la courbe de C_p devrait localement être déformée d'une façon similaire.

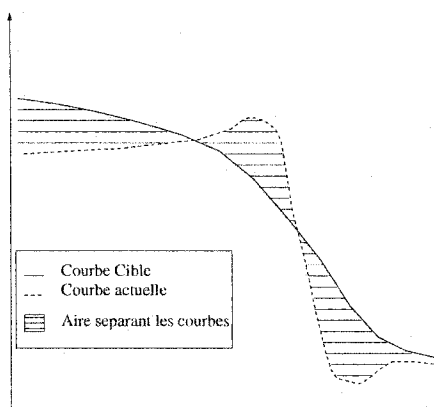


Figure 3.3 Distance entre deux courbes C_p évaluée par une surface

Évidemment, il y a dans cette formulation un transfert de la difficulté vers l'ingénieur puisque c'est dorénavant à lui de déterminer ce qu'est une bonne courbe de C_p .

Dans notre cas et dans un but académique, nous avons choisi une courbe de C_p cible calculée par NSU2D et qui correspond à un profil d'aile représentable par notre paramétrisation avec une certaine combinaison des variables de design. La courbe cible, d'assez bonne qualité, pourrait s'envisager dans un cas réel et nous connaissons, par construction, l'optimum global du problème.

3.2 Résultats expérimentaux

Les résultats sont classés en fonction de la formulation du problème et du modèle basse fidélité choisi.

3.2.1 Maximisation de la finesse de l'aile

3.2.1.1 Résultats obtenus avec BGK comme modèle faible

Nous avons consigné dans le tableau 3.1 les caractéristiques du profil initial et pour chacune des expériences conduites les caractéristiques de l'optimum trouvé ainsi que le nombre d'appels au modèle fort et au modèle faible qui ont été nécessaires à la découverte de cet optimum. Nous avons testé une optimisation directe sur le modèle fort qui constitue notre référence et notre score à battre, ainsi que l'utilisation de l'AMF avec une bêta correction, une bêta correction sans constante (β_0) et une delta correction. Ces trois dernières expériences ont été conduites aux ordres 1 et 2, l'amélioration par passage à l'ordre 3 ne nous ayant pas vraiment convaincu sur les cas analytiques.

Tableau 3.1 Coût de calcul de diverses stratégies et caractéristiques des solutions pour une maximisation de la finesse C_l/C_d en utilisant BGK comme modèle faible

appels et propriétés	initial	directe	β^1	β^2	δ^1	δ^2	β_0^1	β_0^2
f_{lo}	-	0	129	107	305	190	378	295
∇f_{lo}	-	0	25	22	56	42	61	53
f_{hi}	-	83	4	4	8	7	9	11
∇f_{hi}	-	19	2	2	3	4	4	4
C_l	0,6884	0,7191	0,7030	0,7030	0,7103	0,7093	0,711	0,7088
$C_d \times 10^3$	12,4	10,74	11,1	11,1	10,86	10,85	10,83	10,77
C_l/C_d	55,48	66,96	63,18	63,18	65,366	65,363	65,65	65,80

Afin d'analyser soigneusement nos résultats, nous organiserons notre propos selon trois thèmes : la comparaison des tendances prévues analytiquement et de celles constatées expérimentalement, l'étude de l'apport de l'AMF par rapport à une optimisation directe et enfin la vérification de la pertinence physique des résultats.

Comparaison avec les résultats analytiques

Cette expérience, représentée dans le tableau 3.1, a mis à mal les grandes tendances observées analytiquement et retenues en conclusion du chapitre précédent.

Tout d'abord, la bêta correction sans constante se retrouve être la meilleure correction puisqu'elle permet à l'AMF d'aboutir à une fonction coût bien supérieure à celle obtenue avec les autres corrections. En fait, les autres stratégies sont trop rapidement interrompues, à l'instar de la bêta correction qui n'effectue qu'une itération avec succès, soit deux appels à ∇f_{hi} . Ces arrêts prématurés, caractérisés par 2 appels seulement à ∇f_{hi} , constituent une faiblesse de l'AMF car ils limitent considérablement l'ampleur de l'optimisation. Nous en reparlerons dans le chapitre suivant.

D'autre part, augmenter l'ordre de la correction ne semble ni permettre d'obtenir une meilleure fonction coût ni diminuer le nombre d'appels au modèle fort. Ainsi la delta correction quadratique nous amène à un optimum légèrement moins bon que celui de son homologue linéaire et avec un appel au gradient du modèle fort en plus ! La seule correction qui semble s'améliorer avec l'ordre est la bêta correction sans constante.

La comparaison des différentes corrections va donc complètement à l'encontre des premières observations analytiques. Nous verrons si elles se confirment dans les autres expériences.

Apport de l'AMF en terme de coût par rapport à une optimisation directe

Le but de l'AMF est de nous conduire à une bonne solution en économisant le nombre d'appels au modèle fort. Il s'agit donc de comparer les différentes solutions et le coût de leur découverte. Toutefois, dans toutes nos expériences, nous ne pouvons pas parler réellement d'optimum puisqu'aucune des solutions ne correspond

à un critère d'arrêt «propre» comme nous en obtenions avec les problèmes analytiques. Dans le cas de l'optimisation directe, le processus s'arrête parce que les gradients ne sont plus d'assez bonne qualité et ce, à cause du bruit numérique de NSU2D. Dans le cas de l'AMF, le programme s'arrête parce qu'aucun meilleur point n'a été trouvé à l'intérieur du rayon de confiance et ce, à cause du bruit du modèle faible BGK ou du modèle fort NSU2D. Il ne s'agit donc même pas de différents minimums locaux, ce sont justes des points d'arrêt distincts. Pour les comparer il faudra donc tenir compte et de la valeur finale de la fonction coût et du nombre d'appels au modèle fort nécessaires au processus. C'est pourquoi nous avons choisi de comparer (tableau 3.2) le nombre d'appels au modèle fort effectués par l'AMF (à gauche de la barre oblique) au nombre d'appels nécessaires à l'optimisation directe pour dépasser la solution de l'AMF (à droite de la barre oblique). Ce critère de comparaison nous permet ainsi de réellement mesurer l'apport de l'AMF en terme de réduction d'appels au modèle fort.

Tableau 3.2 Nombre d'appels nécessaires à l'optimisation directe pour battre l'optimum de diverses stratégies d'AMF, pour une maximisation de la finesse C_l/C_d en utilisant BGK comme modèle faible.

appels AMF / appels opt. directe	β^1	β^2	δ^1	δ^2	β_0^1	β_0^2
f_{hi}	4/14	4/14	8/45	7/45	9/49	11/52
∇f_{hi}	2/3	2/3	3/10	4/10	4/11	4/13
C_l/C_d AMF/ C_l/C_d au dépassement	63,18/ 64,56	63,18/ 64,56	65,36/ 65,63	65,36/ 65,63	65,65/ 65,67	65,80/ 65,86

Grâce à cette échelle de comparaison nous pouvons observer que l'AMF n'apporte quasiment rien si une bêta correction avec constante est utilisée. La ligne du nombre d'appels au gradient — la plus importante — nous montre un appel seulement en moins par rapport à l'optimisation directe et il reste au moment du dépassement une grande différence en terme de fonction coût. Si l'on ajoute à l'AMF le coût de l'utilisation du modèle faible, nous avons perdu notre temps avec cette stratégie.

Ce n'est heureusement pas le cas avec la delta correction linéaire qui permet une réduction d'un facteur 3 du coût de calcul sur le fort. En tenant compte du temps de calcul sur le faible, l'AMF a permis de diviser le temps de calcul par un facteur 2,5 en comparaison à l'optimisation directe. De même, la bêta correction quadratique sans constante a permis une économie d'un facteur 2,5 sur le temps de calcul.

Nous prendrons enfin une dernière précaution, valable tout au long du mémoire, en faisant remarquer au lecteur que la comparaison des résultats de l'optimisation directe et de l'AMF reste délicate. En effet, l'efficacité de l'AMF dépend du programme que nous avons écrit et de la correction utilisée alors que celle de l'optimisation directe dépend de la qualité de l'optimiseur. Ainsi dans un cas extrême, si nous utilisons un optimiseur très gourmand en évaluations, le nombre d'appels au modèle fort de l'optimisation directe serait très grand alors que l'AMF, lui, ne le consulterait pas plus. Un certain nombre de comparaison avec des logiciels commerciaux nous a cependant conforté dans la qualité de l'optimiseur que nous utilisons et nous ne devrions donc pas favoriser artificiellement l'AMF.

Pertinence physique des résultats

Nous avons tracé les courbes C_p du profil initial, de la solution de l'optimisation directe et de la solution de l'AMF avec δ^1 et avec β_0^2 sur la figure 3.4. Nous pouvons constater que les trois stratégies retenues ont permis d'obtenir des solutions pertinentes physiquement puisque toutes ont limité l'amplitude du choc sur l'extrados. Ces solutions sont aussi allées chercher un surcroît de portance au niveau du bord d'attaque. L'optimisation ne nous a donc pas mené vers une incohérence du solveur et les résultats ont donc une certaine valeur.

Les profils solutions sont représentés sur la figure 3.5. Nous constatons que les modifications sont toutes légères et qu'elles ont toutes consisté en un amincissement

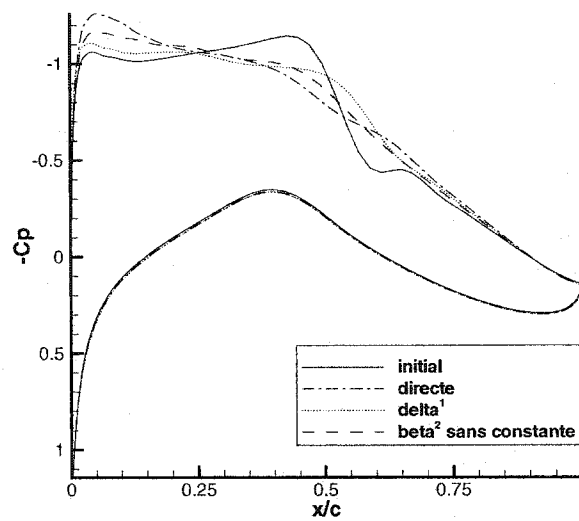


Figure 3.4 Courbes C_p des solutions de l'optimisation directe et de l'AMF utilisé avec BGK sur le problème de maximisation de C_l/C_d

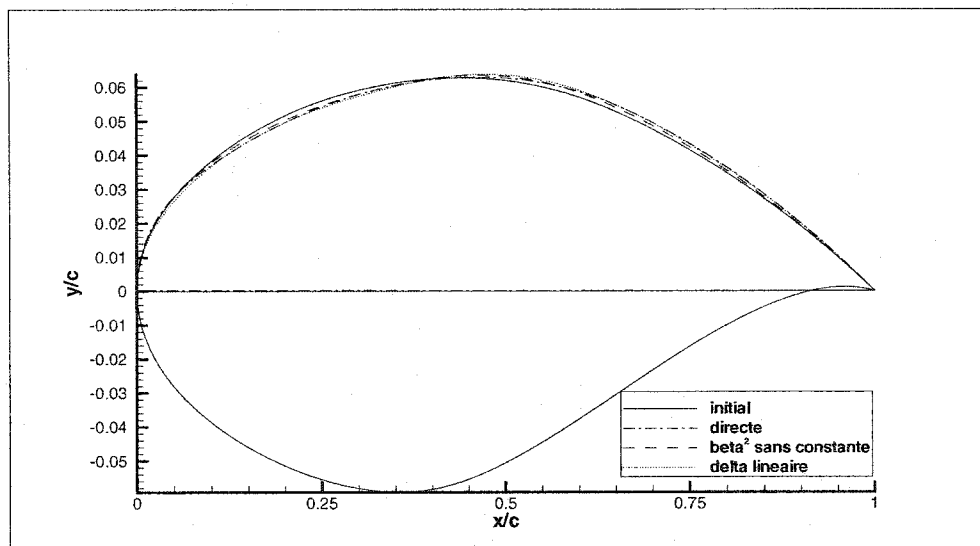


Figure 3.5 Profils solutions de l'optimisation directe et de l'AMF utilisé avec BGK sur le problème de maximisation de C_l/C_d

de l'épaisseur sur la première moitié du profil et un élargissement sur la seconde. Après grossissement au niveau du bord d'attaque, nous pouvons constater aussi un léger épaissement de ce dernier.

Pour conclure sur ces premiers résultats, nous dirons qu'ils sont mitigés puisqu'ils vont à l'encontre de nos résultats analytiques mais qu'ils restent néanmoins intéressants quant à l'apport de l'AMF en terme de réduction d'appel au modèle fort.

Nous précisons cependant que le solveur BGK nous a été déconseillé pour une formulation directe par des ingénieurs de Bombardier. En effet, les prédictions de BGK quant au coefficient de traînée sont particulièrement hasardeuses et certainement très éloignées de ce que peut produire un solveur comme NSU2D. Il s'agit donc d'un modèle faible particulièrement mal adapté au problème, ce qui entraîne une réduction rapide du rayon de confiance et donc des arrêts prématurés comme nous le montrerons au chapitre 4. Cela peut donc être à l'origine de certains problèmes rencontrés et nous permet de nuancer les conclusions tirées de ces premiers résultats.

3.2.1.2 Résultats obtenus avec NSC2KE comme modèle faible

Les résultats sont consignés dans le tableau 3.3. Nous constatons que les différentes stratégies d'AMF ont effectué plus d'itérations qu'avec BGK. Nous n'avons donc pas eu de problèmes d'arrêt prématuré avec ce solveur.

Tableau 3.3 Coût de calcul de diverses stratégies et caractéristiques des solutions pour une maximisation de la finesse C_l/C_d en utilisant NSC2KE comme modèle faible

appels & propriétés	initial	directe	β^1	β^2	δ^1	δ^2	β_0^1	β_0^2
f_{lo}	-	0	91	249	70	102	269	322
∇f_{lo}	-	0	38	71	32	23	57	58
f_{hi}	-	83	14	13	12	6	10	10
∇f_{hi}	-	19	6	7	4	4	5	5
C_l	0,6884	0,7191	0,7041	0,7023	0,7057	0,7044	0,7259	0,7171
$C_d \times 10^3$	12,4	10,74	10,70	10,73	10,70	10,77	11,19	11,13
C_l/C_d	55,48	66,96	65,82	65,46	65,92	65,38	64,84	64,44

Comparaison avec les résultats analytiques

Cette fois-ci, la bêta correction sans constante donne bien des résultats pire que ceux obtenus avec la bêta et la delta correction, conformément à notre analyse analytique. D'autre part, nous constatons qu'en élevant l'ordre de la correction nous dégradons la qualité de la solution, ce qui va totalement à l'encontre de nos résultats analytiques. Nous essayerons d'en déterminer la cause ultérieurement.

Apport de l'AMF en terme de coût

Tableau 3.4 Nombre d'appels nécessaires à l'optimisation directe pour battre l'optimum de diverses stratégies d'AMF, pour une maximisation de la finesse C_l/C_d en utilisant NSC2KE comme modèle faible.

appels AMF / appels opt. directe	β^1	β^2	δ^1	δ^2	β_0^1	β_0^2
f_{hi}	14/52	13/45	12/53	6/45	10/34	10/14
∇f_{hi}	6/13	7/10	4/14	4/10	5/7	5/3
C_l/C_d AMF/	65,82	65,46	65,92	65,38	64,84	64,44
C_l/C_d au dépassement	65,86	65,64	66,01	65,64	64,86	64,56

De la même façon qu'au paragraphe précédent, nous produisons le tableau 3.4 pour comparer effectivement optimisation directe et AMF. Si l'on considère le

temps passé sur le modèle faible, l'utilisation de l'AMF avec une bêta correction sans constante est pénalisante en terme de temps, tandis que la bêta correction quadratique n'apporte rien. En revanche, les delta corrections et la bêta correction linéaire rendent l'AMF efficace. Dans le meilleur des cas, nous divisons le nombre d'appels au fort par 3,5, ce qui revient à diviser le temps de calcul par un peu plus qu'un facteur 2. Dans le pire des cas, pour la bêta correction linéaire, nous divisons le temps global de calcul par un facteur inférieur à 1,5.

Nous attirons ici l'attention du lecteur sur la difficile comparaison des temps de calculs entre NSC2KE et NSU2D. Pour le premier, nous utilisons des départs à chaud ce qui l'avantage par rapport au second. Inversement, NSC2KE est moins efficace et optimisé que NSU2D qui tournerait cinq à dix fois plus rapidement sur un cas Euler. (Toutefois, nous n'avons pas choisi NSU2D en Euler comme modèle faible parce que la génération automatique de maillage donnait de piètres gradients par différences finies, contrairement à NSC2KE). Ainsi il est difficile de se baser sur le temps réel de calcul pour conclure à l'efficacité générale de l'AMF avec NSC2KE. Nous la donnerons toutefois à titre indicatif tout en gardant comme premier critère d'efficacité le nombre d'appels au modèle haute fidélité.

Pertinence physique des résultats

Les courbes C_p des solutions de l'optimisation directe et de β^1 et δ^1 sont tracées sur la figure 3.6. Nous constatons là encore que l'optimisation nous a conduit à des solutions qui ont un sens physique puisqu'elles ont toutes trois diminué la force du choc et sont allées chercher un regain de portance au bord d'attaque. Il est aussi intéressant de constater que les courbes solutions sont beaucoup plus proches les unes des autres que lorsque BGK était utilisé, ce qui suggère qu'elles sont toutes situées dans un même bassin de convergence.

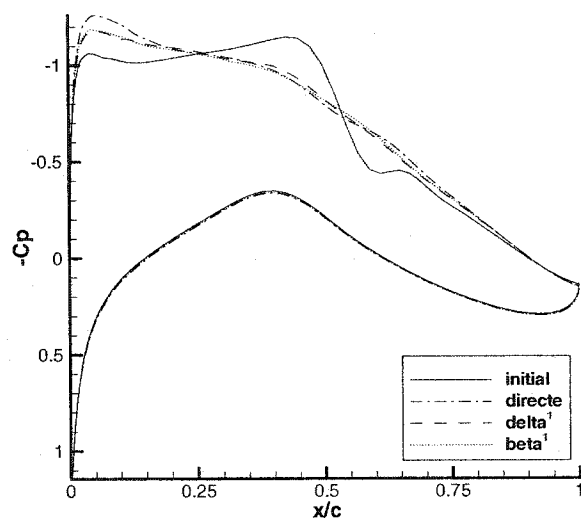


Figure 3.6 Courbes C_p des solutions de l'optimisation directe et de l'AMF utilisé avec NSC2KE sur le problème de maximisation de C_l/C_d

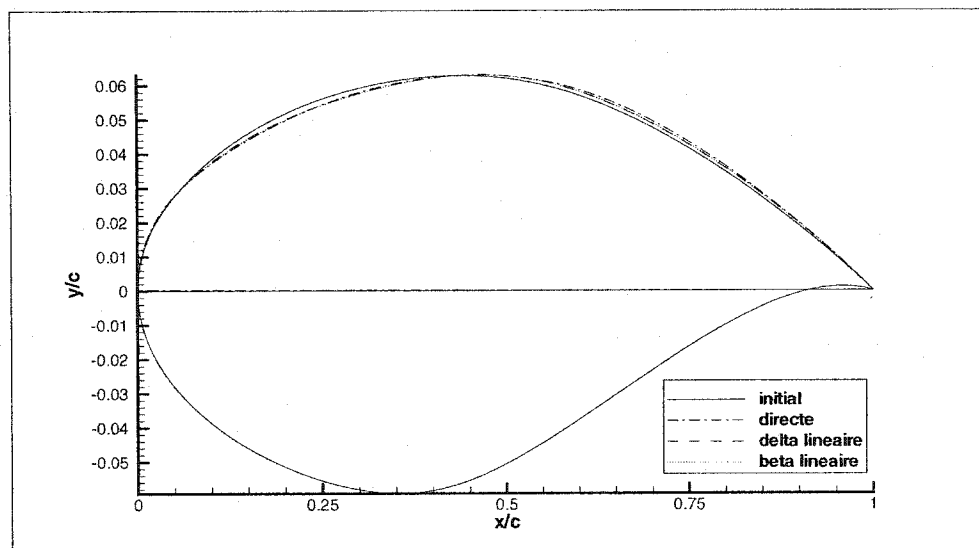


Figure 3.7 Profils solutions de l'optimisation directe et de l'AMF utilisé avec NSC2KE sur le problème de maximisation de C_l/C_d

Les profils solutions sont représentés sur la figure 3.7. Nous constatons que les profils solutions sont très proches les uns des autres comme le suggérerait la comparaison des C_p . Ici aussi, les modifications sont légères et ont toutes consisté en un amincissement de l'épaisseur sur la première moitié du profil et un élargissement sur la seconde. Là-encore une observation plus poussée nous montre un léger épaissement du bord d'attaque.

3.2.2 Minimisation de la traînée sous contrainte de portance

3.2.2.1 Résultats obtenus avec BGK comme modèle faible

Tableau 3.5 Coût de calcul de diverses stratégies et caractéristiques des solutions pour une minimisation de la traînée à portance constante en utilisant BGK comme modèle faible

appels et propriétés	initial	directe	β^1	β^2	δ^1	δ^2	β_0^1	β_0^2
f_{lo}	-	0	56	54	213	64	97	375
∇f_{lo}	-	0	15	15	33	11	25	76
f_{hi}	-	86	2	2	4	2	4	11
∇f_{hi}	-	22	2	2	3	2	3	4
$C_d \times 10^3$	12,36	10,38	11,42	11,42	10,76	11,11	10,73	10,69
C_l	0,6885	0,6882	0,6882	0,6882	0,7000	0,6968	0,6996	0,7098

Nous constatons (tableau 3.5) que l'AMF avec bêta correction ou avec delta correction quadratique a subi un arrêt prématuré puisque le programme n'a effectué qu'une seule itération. Nous rappelons que BGK nous a été déconseillé pour une telle formulation ce qui nous permet de nuancer les résultats de cette expérience.

Comparaison avec les résultats analytiques

A part dans le cas de la bêta correction sans constante, augmenter l'ordre de la correction réduit la performance de l'AMF. De plus, contrairement à nos prévisions, l'AMF semble ici meilleur sans constante qu'avec constante.

Apport de l'AMF en terme de coût

Nous évaluons le nombre d'évaluations nécessaires à l'optimisation directe pour battre les résultats de l'AMF (tableau 3.6). Malgré des arrêts prématurés pour β^1 , β^2 et δ^2 , l'apport de l'AMF est sensible puisque le nombre d'appels est divisé par deux ce qui diminue le temps de calcul par 1,5. Pour δ^1 , le nombre d'appels est divisé par trois. Les meilleurs résultats sont obtenus avec la bêta correction sans constante qui réduit le nombre d'appels au fort par un facteur 4.

Tableau 3.6 Nombre d'appels nécessaires à l'optimisation directe pour battre l'optimum de diverses stratégies d'AMF, pour une minimisation de la traînée à portance constante en utilisant BGK comme modèle faible.

appels AMF et appels opt. directe	β^1	β^2	δ^1	δ^2	β_0^1	β_0^2
f_{hi}	2/25	2/25	4/37	2/25	4/45	11/47
∇f_{hi}	2/4	2/4	3/10	2/4	3/13	4/14
$C_d \times 10^3$ AMF/ $C_d \times 10^3$ au dépassement	11,42/ 11,05	11,42/ 11,05	10,76/ 10,75	11,11/ 11,05	10,73/ 10,71	10,69/ 10,62

Pertinence physique des résultats

Les courbes C_p des solutions de l'optimisation directe et de β_0^1 et δ^1 sont tracées sur la figure 3.8. Là encore, même si les courbes solutions ne correspondent pas à des courbes C_p habituellement rencontrées sur de vrais profils, les solutions ont un certain sens physique puisqu'elles ont toutes trois diminué la force du choc et sont

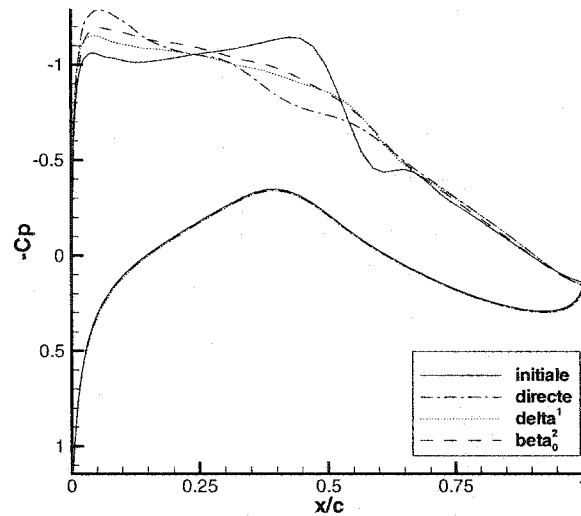


Figure 3.8 Courbes C_p des solutions de l'optimisation directe et de l'AMF utilisé avec BGK sur la minimisation de traînée à portance constante

allées chercher un regain de portance au bord d'attaque.

3.2.2.2 Résultats obtenus avec NSC2KE comme modèle faible

Tableau 3.7 Coût de calcul de diverses stratégies et caractéristiques des solutions pour une minimisation de la traînée à portance constante en utilisant NSC2KE comme modèle faible

appels et propriétés	initial	directe	β^1	β^2	δ^1	δ^2	β_0^1	β_0^2
f_{lo}	-	0	82	82	17	17	72	240
∇f_{lo}	-	0	17	17	7	7	42	63
f_{hi}	-	86	2	2	2	2	15	12
∇f_{hi}	-	22	2	2	2	2	7	5
$C_d \times 10^3$	12,36	10,38	10,68	10,68	10,86	10,86	10,88	10,90
C_l	0,6885	0,6882	0,6931	0,6931	0,6875	0,6875	0,7149	0,7170

Nous constatons (tableau 3.7) que toutes les stratégies s'arrêtent prématurément

sauf pour la bêta correction sans constante mais c'est pour obtenir un moins bon résultat que les autres corrections. Nous observons aussi que les solutions obtenues par les delta corrections sont irréalisables puisqu'elles ne vérifient pas la contrainte d'inégalité. Il s'agit là de la conséquence d'un arrêt prématuré dans le cas d'une optimisation sous contraintes. Le jeu de mise-à-jour de la pénalité permet en effet à l'AMF d'effectuer des itérations dans des zones irréalisables pour à terme converger vers une solution réalisable. Lorsque l'AMF s'arrête trop tôt, cela peut donc se produire dans une zone qui ne vérifie pas les contraintes.

Comparaison avec les résultats analytiques

Augmenter l'ordre ne se perçoit même pas. En effet, l'AMF ne dispose que d'une seule itération pour fabriquer une correction d'un ordre supérieur et les quelques coefficients qui auraient pu être calculés ne l'ont pas été à cause du mauvais conditionnement du système linéaire. Ainsi, corrections linéaires et quadratiques donnent des résultats similaires. Le seul effet tangible de l'augmentation de l'ordre s'observe pour la bêta correction sans constante et il est néfaste. Enfin le résultat obtenu avec la bêta correction sans constante est extrêmement proche des autres solutions mais a nécessité beaucoup plus d'itérations ce qui va dans le sens des conclusions de l'étude analytique.

Apport de l'AMF en terme de coût

Nous constatons (tableau 3.8) que la bêta correction sans constante réduit de manière négligeable le temps de calcul. En revanche, les autres stratégies permettent de diviser le nombre d'appels au modèle fort jusqu'à un facteur 7 pour la bêta correction. Même si l'AMF s'arrête rapidement, le gain est significatif puisque c'est pendant les toutes premières itérations que l'on profite réellement des grandes

Tableau 3.8 Nombre d'appels nécessaires à l'optimisation directe pour battre l'optimum de diverses stratégies d'AMF, pour une minimisation de la traînée à portance constante en utilisant NSC2KE comme modèle faible.

appels AMF et appels opt. directe	β^1	β^2	δ^1	δ^2	β_0^1	β_0^2
f_{hi}	2/48	2/48	2/34	2/34	15/34	12/32
∇f_{hi}	2/14	2/14	2/8	2/8	7/8	5/7
$C_d \times 10^3$ AMF/ $C_d \times 10^3$ au dépassement	10,68/ 10,62	10,68/ 10,62	10,86/ 10,83	10,86/ 10,83	10,88/ 10,83	10,90/ 10,90

tendances données par le modèle faible. En terme de temps de calcul, la bêta correction divise le temps de calcul par 4 et la delta correction par 2,5. Toutefois, il serait nécessaire de prolonger le résultat de la delta correction par une optimisation directe ne serait-ce que pour obtenir une solution réalisable.

Pertinence physique des résultats

Les courbes C_p des solutions de l'optimisation directe et de β^1 et δ^1 sont tracées sur la figure 3.9. Nous constatons là encore que l'optimisation nous a conduit à des solutions qui ont un sens physique puisqu'elles ont toutes trois diminué la force du choc et sont allées chercher un regain de portance au bord d'attaque. En revanche, elles sont très différentes les unes des autres contrairement aux résultats de l'optimisation de la finesse C_l/C_d obtenus avec le même solveur. Le jeu de la pénalité dû à la contrainte envoie certainement l'algorithme dans des parties très différentes de l'espace de design et ce, un peu au hasard. L'AMF s'étant arrêté tout de suite, les solutions n'ont pas pu se rapprocher comme dans le cas de la première formulation.

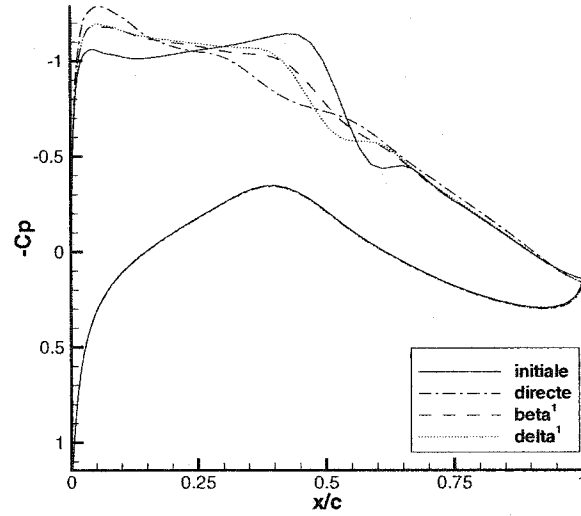


Figure 3.9 Courbes C_p des solutions de l'optimisation directe et de l'AMF utilisé avec NSC2KE sur la minimisation de traînée à portance constante

3.2.3 Optimisation avec courbe de C_p cible

3.2.3.1 Résultats obtenus avec BGK comme modèle faible

Tableau 3.9 Coût de calcul et fonction coût finale de diverses stratégies pour une optimisation avec C_p cible en utilisant BGK comme modèle faible

appels & f_{hi} finale	initial	directe	β^1	β^2	δ^1	δ^2	β_0^1	β_0^2
f_{lo}	-	0	148	418	217	254	97	161
∇f_{lo}	-	0	45	96	48	62	50	75
f_{hi}	-	91	14	9	10	8	16	10
∇f_{hi}	-	23	6	5	5	4	9	4
$f_{hi} = \text{Aire}$	8,34	0,401	0,701	0,816	0,799	0,866	1,39	4,08

Les résultats obtenus sont présentés dans le tableau 3.9. Nous constatons que toutes les stratégies ont effectué un certain nombre d'itérations. Nous n'avons donc pas

eu d'arrêt prématuré ce qui est en outre imputable à la nature du problème, bien adapté à la paramétrisation et aux solveurs.

Comparaison avec les résultats analytiques

Nous constatons que dans tous les cas l'augmentation de l'ordre de la correction détériore les résultats. Nous observons aussi les piètres résultats obtenus avec une bêta correction sans constante.

Apport de l'AMF en terme de coût

Tableau 3.10 Nombre d'appels nécessaires à l'optimisation directe pour battre l'optimum de diverses stratégies d'AMF, pour une optimisation avec C_p cible en utilisant BGK comme modèle faible.

appels AMF / appels opt. directe	β^1	β^2	δ^1	δ^2	β_0^1	β_0^2
f_{hi}	14/63	9/63	10/63	8/59	16/44	10/8
∇f_{hi}	6/15	5/15	5/15	4/12	9/9	4/2
Aire finale AMF/ Aire au dépassement	0,701/ 0,66	0,816/ 0,66	0,799/ 0,66	0,866/ 0,85	1,39/ 1,06	4,08/ 2,58

Les comparaisons sont consignées dans le tableau 3.10. Nous constatons peu de gain, voire une perte de temps lorsque l'on utilise une bêta correction sans constante. En revanche l'utilisation de l'AMF permet de diviser le nombre d'appel au modèle fort par un facteur 2,5 pour β^1 et par un facteur 3 pour β^2 , δ^1 et δ^2 ce qui correspond respectivement à une réduction du temps de calcul par un facteur 2 et un facteur 2,5. Ainsi, même si β^1 va plus loin en terme de fonction coût, le gain est moins important.

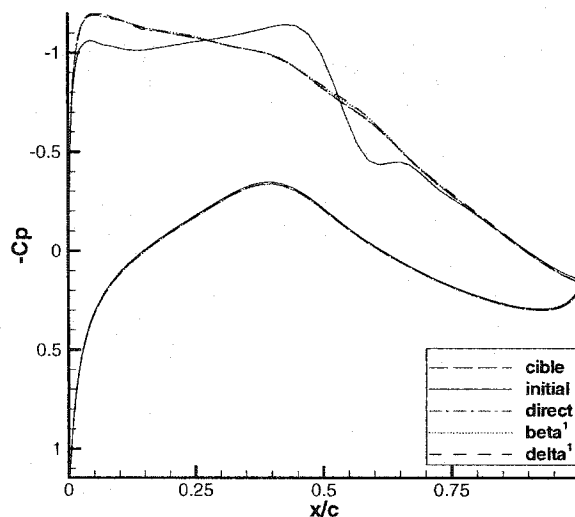


Figure 3.10 Courbes C_p des solutions de l'optimisation directe et de l'AMF utilisé avec BGK sur l'optimisation avec C_p cible

Pertinence physique des résultats

Ont été représentées sur la figure 3.10, les courbes C_p initiale, cible, et les résultats de l'optimisation directe et de l'AMF avec β^1 et δ^1 .

Nous pouvons constater que toutes les optimisations se sont globalement bien déroulées et qu'elles ont réussi à atteindre leur cible avec une précision relativement bonne.

3.2.3.2 Résultats obtenus avec NSC2KE comme modèle faible

Là encore (tableau 3.11), les AMF ont pu effectuer un certain nombre d'itérations sans arrêt prématuré. Nous constatons de plus que les solutions obtenues avec δ^1 et β^1 sont même meilleures que celles obtenues avec l'optimisation directe !

Tableau 3.11 Coût de calcul et fonction coût finale de diverses stratégies pour une optimisation avec C_p cible en utilisant NSC2KE comme modèle faible

appels & f_{hi} finale	initial	directe	β^1	β^2	δ^1	δ^2	β_0^1	β_0^2
f_{lo}	-	0	258	676	214	279	96	130
∇f_{lo}	-	0	72	158	58	79	54	64
f_{hi}	-	91	17	19	14	16	17	14
∇f_{hi}	-	23	9	11	5	9	7	5
$f_{hi} = \text{Aire}$	8,34	0,401	0,241	0,655	0,304	0,466	0,486	0,736

Comparaison avec les résultats analytiques

Contrairement aux résultats analytiques, accroître l'ordre des corrections détériore les résultats. En revanche conformément à ce qui était prévu, la bêta correction sans constante est toujours la plus mauvaise.

Apport de l'AMF en terme de coût

Tableau 3.12 Nombre d'appels nécessaires à l'optimisation directe pour battre l'optimum de diverses stratégies d'AMF ou à l'AMF pour battre l'optimisation directe, pour une optimisation avec C_p cible en utilisant NSC2KE comme modèle faible.

appels AMF / appels opt. directe	β^1	β^2	δ^1	δ^2	β_0^1	β_0^2	$f_{lo} + f_{hi}$
f_{hi}	11/91	19/66	8/91	16/81	17/81	14/63	123/76
∇f_{hi}	6/23	11/16	4/23	9/22	7/22	5/15	34/20
Aire AMF/ Aire directe	0,388/ 0,401	0,655/ 0,654	0,304/ 0,401	0,466/ 0,401	0,486/ 0,401	0,736/ 0,66	0,563/ 0,54

Les comparaisons en terme de coût sont consignées dans le tableau 3.12. Pour les corrections δ^1 et β^1 qui ont fait mieux que l'optimisation directe, nous avons compté le nombre d'appels nécessaires à l'AMF pour battre l'optimisation directe. Nous avons aussi ajouté à ce tableau de comparaison les résultats obtenus avec une optimisation séquentielle ($f_{lo} + f_{hi}$).

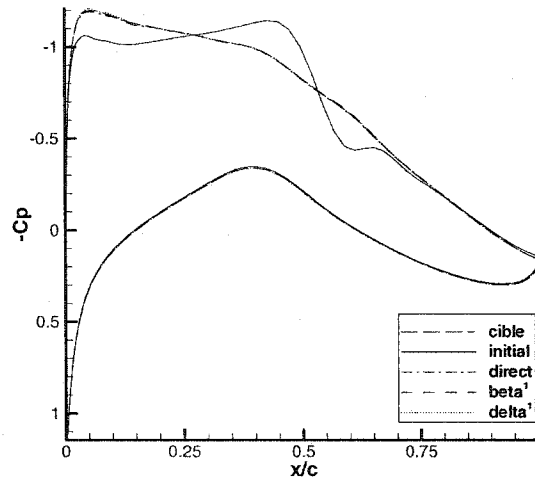


Figure 3.11 Courbes C_p des solutions de l'optimisation directe et de l'AMF utilisé avec NSC2KE sur l'optimisation avec C_p cible

Nous constatons que β^1 permet de diviser le nombre d'appels au modèle fort par un facteur 4 ce qui correspond à un temps de calcul divisé par 2,5. De même δ^1 permet de diviser le nombre d'appels par 6 et le temps par 3,5.

Nous observons que la stratégie qui consiste à optimiser successivement le faible et le fort est perdante. Cela vient du fait que les modèles physiques des deux solveurs sont différents. NSC2KE, basé sur les équations d'Euler, a tendance à situer le choc en amont du profil, comparativement à la solution des équations de Navier-Stokes donnée par NSU2D. La solution à laquelle aboutit l'optimisation sur le faible est alors très décalée, ce qui revient à donner un point de départ désavantageux à l'optimisation directe sur le modèle fort. Mieux vaut donc opérer des corrections régulièrement avec l'AMF.

Pertinence physique des résultats

Ont été représentées sur la figure 3.11, les courbes C_p initiale, cible, et les résultats de l'optimisation directe et de l'AMF avec β^1 et δ^1 . Nous pouvons constater que toutes les optimisations se sont bien déroulées et qu'elles ont réussi à atteindre leur cible avec une bonne précision.

3.2.4 Synthèse des résultats expérimentaux

Afin de tirer des conclusions générales de toutes ces expériences, nous avons dressé un tableau récapitulatif (tableau 3.13). Le tableau est organisé comme suit :

- Dans la première colonne, nous avons noté le bon déroulement de l'AMF, caractérisé par une absence totale d'arrêt prématuré ($++$), l'apparition de quelques arrêts prématurés ($+-$) ou l'omniprésence des arrêts prématurés ($--$).
- La seconde colonne vérifie si oui ($+$) ou non ($-$) la bêta correction est meilleure avec que sans constante.
- La troisième colonne vérifie si les corrections sont plus efficaces ($+$) ou non ($-$) en augmentant l'ordre.
- A chaque itération de l'AMF, l'optimiseur effectue un certain nombre d'itérations qui lui sont propres sur le modèle faible corrigé. Il est donc possible pour chaque expérience de calculer un nombre moyen d'itérations nécessaires à la résolution du sous-problème d'optimisation. Nous avons représenté dans la quatrième colonne l'accroissement, au passage d'une correction linéaire à une correction quadratique, de ce nombre d'itérations par sous-optimisation pour la bêta et la delta correction (respectivement à gauche et à droite de la barre oblique).
- Enfin nous avons consigné dans la dernière colonne les facteurs de réduction en nombre d'appels (ité) et en temps (tps) des corrections linéaires ainsi que les

valeurs finales des fonctions coût (f_{hi}).

Tableau 3.13 Tableau récapitulatif des expériences menées avec l'AMF

formulation et modèle faible	AMF bien déroulé	β > β_0	effet de $\nearrow$ ordre	$\nearrow$ itérations par $\nearrow$ ordre β/δ	gains ité/tps/ f_{hi} pour β^1	gains ité/tps/ f_{hi} pour δ^1
C_l/C_d + BGK	+-	-	-	-12%/-14%	1,5/1/63,2	3/2,5/64,6
C_l/C_d + NSC2KE	++	+	-	+101%/+44%	2/1,5/65,8	3,5/2/65,9
C_d tq C_l + BGK	+-	-	-	0%/-33%	2/1,5/11,42	3/2,5/10,8
C_d tq C_l + NSC2KE	--	+	-	0%/0%	7/4/10,7	4/2,5/10,9
C_p cible + BGK	++	+	-	+232%/+61%	2,5/2/0,7	3/2,5/0,8
C_p cible + NSC2KE	++	+	-	+96%/+19%	4/2,5/0,2	6/3,5/0,3

Voici les conclusions que nous pouvons tirer :

- La moitié des problèmes sont concernés par des arrêts prématurés. Il s'agit donc d'un problème très courant. On notera qu'il ne dépend pas uniquement de la qualité du modèle faible puisqu'alors même que le bruit de NSC2KE est très limité, des arrêts prématurés surviennent pour la minimisation de la traînée à portance constante.
- Pour tous les problèmes qui n'ont pas subi d'arrêt prématuré, la bêta correction avec constante a été plus efficace que sans constante, conformément aux résultats analytiques. Des résultats contradictoires ont été cependant obtenus avec les deux premières formulations appliquées sur BGK car dans ces deux cas, l'AMF basé sur une bêta correction avec constante n'a effectué qu'une seule itération et a donc subi un arrêt prématuré. Nous pouvons donc en partie écarter ces résultats contradictoires.
- Dans tous les cas, augmenter l'ordre de la correction est pénalisant en terme de coût et de qualité de la solution. Ce résultat est à l'opposé des prévisions analytiques.
- Pour toutes les expériences qui n'ont pas subi d'arrêt prématuré, le processus de sous-optimisation requiert plus d'évaluations lorsqu'une correction d'un ordre

supérieur est utilisée. En effet l'approximation se complexifie avec l'ordre de la correction et cela est encore plus pénalisant avec une correction multiplicative comme nous l'expliquions dans notre analyse analytique. Ainsi, non seulement l'ordre supérieur est plus coûteux au niveau du modèle fort, mais en plus il l'est aussi au niveau du modèle faible. Il n'y a donc aucun intérêt à augmenter l'ordre.

- En augmentant la qualité du modèle faible, on aboutit à une meilleure solution finale. Ce résultat s'observe dans toutes nos expériences quand on passe de BGK à NSC2KE, à l'exception de l'optimisation sous contrainte où, pour la bêta correction, remplacer BGK par NSC2KE est très légèrement pénalisant. Toutefois, l'optimisation sous contrainte avec NSC2KE a subi des arrêts prématurés pour la bêta correction ce qui nous permet encore d'écarter ce résultat.
- La formulation avec C_p cible, qui est la mieux adaptée à notre paramétrisation, nous donne les résultats les plus encourageants pour notre algorithme, sans arrêts prématurés.
- Nous pouvons constater que si une stratégie d'AMF va plus loin que les autres en terme de fonction coût, cela n'entraîne pas forcément de meilleurs gains. Ainsi dans la formulation avec C_p cible, même si la fonction coût obtenue par la bêta correction est meilleure que celle de la delta correction, le gain qu'elle apporte en terme d'itérations est moins important. Cela provient du fait que l'AMF est très intéressant au début puisqu'il profite des directions de descente du modèle faible mais qu'il devient moins efficace lorsqu'il s'approche de l'optimum et que de nombreuses itérations-corrrections sont nécessaires.
- Les corrections linéaires (sauf la bêta correction sans constante) ont toujours divisé le nombre d'appels au modèle fort, par 2,7 en moyenne pour β^1 et par 3,7 en moyenne pour δ^1 . Nous soulignons enfin que même en cas d'arrêt prématuré, l'AMF est intéressant par rapport à l'optimisation directe.

Les résultats sont donc encourageants, en particulier pour les problèmes bien posés (C_p cible). Cependant il nous reste à analyser le problème récurant des arrêts prématurés et à expliquer pourquoi, contrairement à nos prévisions analytiques, l'augmentation de l'ordre est pénalisante.

CHAPITRE 4

ROBUSTESSE DE L'AMF

Nous avons constaté au cours de nos expériences sur de véritables résolveurs numériques que l'AMF s'arrêtait en général beaucoup plus tôt que l'optimisation directe alors que les deux stratégies devraient théoriquement être aussi robuste l'une que l'autre. Or nous avons constaté cette robustesse équivalente durant notre étude sur des cas analytiques. Le problème se situe donc certainement dans la différence que l'on peut observer entre les modèles analytiques et les expériences réelles.

Intuitivement, le facteur qui vient modifier les données du problème quand on passe d'un modèle analytique à un résolveur numérique est la présence de bruit. C'est pourquoi nous allons bâtir un certain nombre d'hypothèses sur l'influence du bruit numérique pour ensuite les tester sur une simulation.

4.1 Hypothèses sur l'influence du bruit

La théorie de convergence de l'AMF a été établie en supposant que toutes les fonctions étaient C^1 . Cette hypothèse permet en fait de toujours trouver un rayon assez petit pour que le pas, solution du sous-problème d'optimisation, soit accepté. Or dans le cas réel, les fonctions sont perturbées par du bruit et un petit rayon devient rapidement problématique dès lors que la variation caractéristique des fonctions dans la zone de confiance est du même ordre de grandeur que le bruit numérique. En somme, les sous-optimisations sont fiables et cohérentes si le rayon de confiance n'est ni trop gros, afin que l'approximation ne nous mène pas n'importe où, ni trop

petit, afin que les variations observées aient un sens réel. Dans nos expériences sur des cas réels, le rayon devait être inférieur à 0,25 (limite supérieur des rayons qui ont borné une optimisation fructueuse) et supérieur à 0,01 (ordre de grandeur du pas de différenciation).

Nous développerons dans cette partie deux hypothèses portant sur l'influence du bruit provenant respectivement du solveur haute fidélité et du solveur basse fidélité.

4.1.1 Effet du bruit numérique du modèle fort

Nous supposerons dans cette partie que le solveur basse fidélité est non bruité pour nous concentrer uniquement sur les effets du bruit numérique du modèle fort.

La taille caractéristique du rayon de confiance en-deçà de laquelle les problèmes de bruit deviennent dangereux est directement donnée par le pas de différenciation car il a justement été déterminé de manière à limiter les effets du bruit sur la valeur du gradient.

Imaginons alors, comme cela arrive lorsque le solveur basse fidélité indique de mauvaises directions de descente, que le rayon de confiance s'atrophie et soit plus petit que le pas de différenciation (figure 4.1). Dans ce cas, le sous-problème d'optimisation est alors très simple à résoudre dans la mesure où le solveur basse fidélité est non bruité. En effet, le rayon de confiance étant plus petit que le pas de différenciation, le sous problème d'optimisation porte sur une fonction quasi-linéaire et le minimum est alors souvent l'intersection de la direction de descente et des bornes du rayon de confiance.

Ce pas-solution est alors soumis à l'approbation du modèle fort pour vérifier qu'il conduit bien à une amélioration de f_{hi} . C'est à ce moment que le bruit numérique du solveur haute fidélité peut devenir fatal. En effet, d'après la figure 4.1, le pas

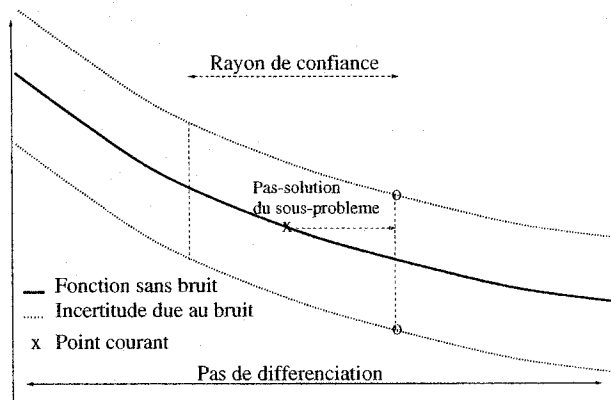


Figure 4.1 Influence du bruit numérique du solveur haute fidélité sur l'AMF

solution peut donner une valeur de fonction coût situé entre les deux «O» à cause du bruit numérique. Si la valeur que donne le solveur haute fidélité est proche du «O» supérieur, l'AMF constatera une augmentation de fonction coût, le pas sera donc rejeté et le rayon encore diminué. Si quelques itérations se succèdent ainsi, le rayon atteindra sa limite inférieure et l'AMF s'arrêtera. Ainsi, alors même que le sous-problème d'optimisation produit une solution qui est tout-à-fait correcte, le pas est si petit que la baisse de la fonction coût lissée est noyée dans le bruit et peut passer inaperçue au niveau de l'AMF.

4.1.2 Effet du bruit numérique du modèle faible

L'arrêt prématuré pourrait aussi survenir à cause du bruit numérique qui pollue le modèle faible.

Dans notre étude, nous avons utilisé pour le sous-problème d'optimisation une méthode SQP avec recherche par ligne. Une fois le gradient évalué au point courant, l'optimiseur calcule une direction de descente, ou ligne de descente, sur laquelle il va faire plusieurs évaluations. Si le rayon de confiance est très petit, l'optimiseur ne fera

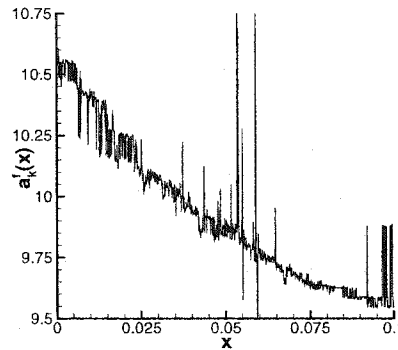


Figure 4.2 Influence du bruit numérique du solveur basse fidélité sur l'AMF

que très peu d'évaluations de l'approximation. Or, si cette dernière est bruitée, il se peut tout-à-fait qu'il ne trouve aucun meilleur point sur cette ligne. L'optimiseur s'arrête donc et renvoie en solution le point de départ. L'AMF l'interprète alors comme un pas infiniment petit et s'arrête.

La figure 4.2 illustre le bruit numérique que l'on peut rencontrer sur l'approximation $a'_k(x)$ et qui provient du modèle faible (ici BGK). Le lecteur pourra constater que si le rayon de confiance devient dans ce cas inférieur à 0,01 (taille caractéristique du pas de différenciation), ce qui arrive très souvent dans l'utilisation de l'AMF, le bruit devient plus important que les variations caractéristiques de la fonction. Les quelques points évalués par l'optimiseur peuvent alors être supérieurs au point courant à cause du bruit alors même qu'ils sont situés sur une véritable ligne de descente. L'optimiseur ne trouve alors pas de meilleur point que le point de départ qui lui a été donné et l'AMF s'arrête prématurément.

4.2 Simulation du bruit numérique

Nous avons supposé au paragraphe précédent l'influence que peut avoir le bruit numérique sur l'AMF. Nous avons voulu tester nos hypothèses sur une simulation.

Nous avons repris l'exemple de la fonction de Rosenbrock développé au cours de l'étude analytique et nous avons perturbé les modèles par du bruit artificiel du même ordre de grandeur que celui rencontré expérimentalement. Plus précisément, la valeur que retourne le modèle est :

$$f_{\text{bruit}} = f_{\text{analytique}}(1 + \text{bruit}),$$

où le bruit est égal à :

$$\begin{aligned} \text{bruit} = & \text{amplitude} \left(\sin(w_1(2x_1 + x_2)) + \cos(w_2(x_1 + 2x_2)) \right. \\ & - \left| \cos(w_3(2x_1 + x_2)) \sin(w_4(x_1 + 2x_2)) \right| \\ & + \left. \left| \sin(w_6(2x_1 + x_2)) \right| \right), \end{aligned}$$

où les w sont des pulsations de très haute fréquence (10^9) différentes d'un modèle à l'autre. Cela rend la fonction non lisse aux yeux de l'optimiseur.

L'amplitude du bruit artificiel a été réglée en fonction de ce que nous avons observé sur NSU2D. D'ailleurs, l'analyse de sensibilité effectuée sur le modèle analytique bruité nous a permis de sélectionner un pas de différenciation égal à celui choisi pour NSU2D.

Nous avons effectué un certain nombre d'expériences afin de distinguer les effets de différents facteurs : bruit du solveur haute fidélité, bruit du solveur basse fidélité et pas de différenciation. Nous obtenons en effet des résultats différents, même en absence de bruit, selon que les gradients sont calculés avec un pas de différenciation fini ou mathématiquement.

Tableau 4.1 Influence du bruit sur l'AMF pour le problème de fidélité variable basé sur la fonction de Rosenbrock

f_{hi} finale et appels à ∇f_{hi}	directe	β^1	β^2	δ^1	δ^2
modèle mathématique	$4, 2.10^{-11}$ 27	$1, 0.10^{-10}$ 44	$2, 2.10^{-11}$ 9	$9, 9.10^{-11}$ 49	$3, 4.10^{-14}$ 3
aucun bruit pas = 0,01	$3, 8.10^{-4}$ 30	$1, 0.10^{-6}$ 26	$1, 8.10^{-5}$ 8	$8, 7.10^{-7}$ 28	$3, 8.10^{-4}$ 4
bruit sur le fort	$1, 4.10^{-4}$ 37	$3, 0.10^{-5}$ 47	6, 2 4	$1, 7.10^{-3}$ 21	5, 9 9
bruit sur le faible	$3, 8.10^{-4}$ 30	$4, 1.10^{-1}$ 5	$4, 1.10^{-1}$ 14	$3, 1.10^{-2}$ 6	$7, 7.10^{-1}$ 8
bruit sur les deux	$1, 4.10^{-4}$ 37	1, 2 4	1, 4 6	$2, 2.10^{-1}$ 8	1, 0 3

Nous avons consigné dans le tableau 4.1 la fonction coût finale et le nombre d'appels au gradient du modèle fort pour cinq configurations faible-fort :

- La première ligne correspond à des modèles non-bruités différenciés mathématiquement.
- La seconde correspond à des modèles non bruités mais dont les gradients sont calculés par différences finies avec un pas de 0,01.
- Pour la troisième, nous avons bruité uniquement le modèle fort et calculons tous les gradients avec un pas différenciation de 0,01.
- Dans la quatrième, nous avons bruité uniquement le modèle faible et calculons tous les gradients avec un pas de 0,01.
- Dans la dernière expérience, les deux modèles sont bruités et leurs gradients calculés avec un pas de 0,01.

Pour chacune d'entre elle nous avons effectué une optimisation directe et utilisé l'AMF avec bêta et delta corrections aux ordres 1 et 2. Nous aurons ainsi accès à la fonction lissée et à la fonction bruitée et serons donc capables de déterminer les véritables raisons de l'arrêt de l'AMF pour ces exemples.

Nous analysons l'effet de l'introduction du pas de différenciation fini et du bruit dans les parties suivantes.

4.2.1 Influence du pas de différenciation fini

La première constatation vient de la différence apportée par un pas de différenciation qui n'est plus infinitésimal. Nous pouvons voir que les solutions auxquelles aboutissent les différentes stratégies sont bien moins bonnes que celles obtenues sur de purs modèles mathématiques. L'AMF et l'optimiseur s'arrêtent parce que les gradients calculés par différences finies ne sont plus fiables. En effet, proche d'un minimum, les erreurs de troncature dans l'évaluation du gradient, dues aux termes négligés dans le développement de Taylor, génèrent de mauvais gradients et donc de fausses directions de descente. Les solutions des optimisations sont donc moins précises.

Nous constatons de plus que l'AMF aboutit à une valeur de la fonction coût moins basse avec les ordres quadratiques qu'avec les ordres linéaires alors qu'avec les modèles mathématiques les solutions sont aussi bonnes. Heureusement, conformément à notre analyse du chapitre 2, les ordres supérieurs sont toujours plus rapides car il faut en effet 22 itérations à la bêta correction linéaire pour battre les 8 itérations de la quadratique et 18 itérations pour la delta linéaire pour dépasser les 4 de la quadratique.

4.2.2 Influence du bruit provenant du modèle fort

Nous constatons en bruitant le modèle fort que la qualité de la solution de l'optimisation directe n'est pas affectée alors qu'elle a baissé pour toutes les stratégies.

Les arrêts sont survenus pour différentes raisons :

- L'AMF avec bêta correction linéaire est allé très loin et a rencontré le même critère d'arrêt que les optimisations sans bruit. La dernière sous-optimisation n'a pas trouvé de meilleur point que le point de départ à cause des erreurs de troncature dans l'évaluation du gradient du modèle faible non bruité. Même si le bruit a rendu l'AMF moins efficace en terme de fonction coût finale et de nombre d'évaluations, il n'est pas responsable de son arrêt.
- Les deux corrections quadratiques se sont arrêtées alors que dans le dernier gradient du modèle fort, l'une des dérivées partielles est de signe opposé à ce qu'il devrait être. En revanche si la direction de descente est loin d'être parfaite, elle n'en est pas moins une véritable direction de descente. Les nouveaux points proposés sont meilleurs quand ils sont évalués avec la fonction lisse mais refusés car ils apparaissent moins bons avec le bruit numérique. Il s'agit donc d'un arrêt qui illustre bien l'hypothèse que nous avançons.
- L'AMF avec delta correction linéaire s'est arrêté quant à lui alors que le gradient indiquait une direction de descente qui n'en était vraiment pas une. En effet, près de l'optimum, les dérivées partielles sont de plus en plus faibles et de plus en plus noyées dans le bruit numérique. Sans adaptation graduelle du pas de différenciation, le gradient fini par être complètement faux.

Le dernier critère d'arrêt est très instructif car il apporte un nouvel élément à l'étude de la robustesse de l'AMF : les gradients deviennent effectivement de plus en plus mauvais car nous n'adaptions pas graduellement la taille du pas de différenciation. Or, l'AMF y est beaucoup plus sensible que l'optimiseur. En effet ce dernier est

capable de faire des redémarrages lorsqu'il ne trouve pas de meilleurs points dans la direction de descente (à ce moment il abandonne le hessien qu'il avait calculé). Ces nouveaux départs le rendent plus robuste et lui permettent d'aller plus loin que l'AMF qui ne peut que s'arrêter si un gradient faux lui est donné.

4.2.3 Influence du bruit provenant du modèle faible

Dans tous les cas, le processus de sous-optimisation n'a pas trouvé de meilleur point dans le rayon de confiance à cause du bruit numérique présent dans le modèle faible, conformément à l'hypothèse que nous avons bâtie.

4.2.4 Influence du bruit provenant des deux modèles

Nous obtenons un résultat assez intuitif : lorsque le bruit affecte les deux solveurs, la fonction coût finale est du même ordre de grandeur que la pire des solutions entre le modèle fort bruité et le modèle faible bruité.

4.2.5 Influence du bruit sur l'ordre de la correction

Un nouvel aspect de l'influence du bruit numérique apparaît avec cette expérience. En effet, alors même que sans bruit les corrections d'un ordre supérieur étaient plus efficaces, elles deviennent très mauvaises dès que l'on ajoute du bruit numérique.

Cela peut s'expliquer par le fait que les coefficients que l'on rajoute au polynôme de correction pour passer de l'ordre 1 à l'ordre 2 sont basés sur les valeurs des fonctions, et surtout des gradients, des précédentes itérations. Ces gradients sont plus ou moins erronés à cause du bruit des deux solveurs et par conséquent, si

l'on tente d'en déduire un ordre supérieur, les coefficients résultant peuvent être complètement faux. Il est alors très hasardeux d'élever l'ordre de la correction.

4.3 Synthèse de l'étude de robustesse de l'AMF

Il apparaît de la simulation que nous avons menée que le bruit numérique perturbe considérablement l'AMF.

- Si le gradient du modèle fort est faux, l'AMF ne peut que s'arrêter alors que l'optimiseur peut essayer de repartir.
- Lorsque le rayon de confiance devient petit, l'AMF peut s'arrêter à cause du bruit du modèle fort ou du modèle faible alors même que les directions de descente sont bonnes. Nous faisons remarquer cependant qu'il serait possible de s'affranchir en partie de l'arrêt prématuré dû au modèle faible en renvoyant à l'AMF, comme solution du sous-problème d'optimisation lorsque ce dernier ne trouve pas mieux que le point actuel, l'intersection de la droite dirigée par le gradient du modèle fort et des bornes de confiance.
- Les erreurs des deux modèles s'ajoutent quand il s'agit de calculer les corrections et le résultat devient fatal quand nous voulons passer à un ordre supérieur.

Toutes ces tendances peuvent très bien expliquer le comportement constaté dans nos expériences. Il nous est cependant impossible d'en être certain puisque nous n'avons pas accès aux fonctions réelles lissées ! Pour être certain de l'influence des différents facteurs, il faudrait prendre un cas réel, supprimer le bruit numérique en interpolant les fonctions par des surfaces de réponse et observer le comportement de l'AMF. Nous pourrions alors savoir si c'est effectivement le bruit qui a condamné les corrections d'un ordre supérieur ou si nous avons bâti des cas particuliers dans le chapitre 2.

L'enseignement qu'il faut retenir de cette étude est qu'il faut faire extrêmement attention au bruit numérique quand on désire employer l'AMF car il y est plus sensible qu'un optimiseur classique. Nous suggérerons d'accroître la taille des pas de différenciation pour le calcul du gradient, par rapport à une optimisation classique, pour gagner en robustesse quitte à perdre en précision. Enfin nous ferons remarquer que le problème de l'arrêt prématuré n'est pas si mauvais car il apparaît souvent quand le rayon de confiance est petit, signe que les solveurs ont des comportements de plus en plus différents. Il indique donc assez naturellement qu'il est temps d'abandonner le solveur basse fidélité pour prolonger éventuellement avec le solveur haute fidélité.

CONCLUSION

L'objectif de ce travail était d'étudier l'opportunité d'utiliser deux solveurs de qualités distinctes pour un problème aérodynamique dans le but d'opérer une optimisation de qualité et à moindre coût. Parmi les stratégies proposées dans la littérature, nous avons choisi l'AMF, proposé par Alexandrov, pour sa rigueur mathématique et son adaptation évidente aux méthodes d'optimisation à gradient, ces dernières étant performantes pour notre problème.

Après avoir présenté la théorie assurant la convergence de l'algorithme, nous avons proposé une formulation légèrement différente de celle suggérée par Alexandrov pour tenir compte au maximum des informations du solveur basse fidélité, en particulier au niveau des contraintes. De plus, nous avons pu ainsi aborder l'architecture de notre programme sous un aspect modulaire afin de pouvoir modifier à volonté solveurs et optimiseur.

Nous avons validé le comportement général de l'AMF sur des cas analytiques afin de nous assurer de la convergence de l'algorithme choisi et implanté. L'introduction d'une constante dans la formulation de la bêta correction nous a en outre permis d'éviter les problèmes de conditionnement. Nous avons ensuite évoqué la lenteur de convergence sur certains problèmes et avons tenté d'y remédier en introduisant une nouvelle correction, non-plus multiplicative comme le suggérait la littérature mais additive, ainsi qu'en augmentant l'ordre de cette correction à partir des données accumulées au cours du processus d'optimisation. Nous avons conclu que l'augmentation de l'ordre était très efficace pour les deux types de correction envisagées.

Nous avons ensuite appliqué ce programme sur l'optimisation aérodynamique d'un

profil d'aile. Trois formulations ont été testées, chacune sur deux couples de résolveurs basse et haute fidélité, soit BGK-NSU2D et NSC2KE-NSU2D. Les résultats ont tous contredit l'intérêt des corrections d'ordre élevé observé sur les cas analytiques. Les expériences ont cependant montré une réduction d'un facteur 3 en moyenne du nombre de calculs effectués sur le résolveur haute fidélité lorsque des corrections linéaires sont employées avec l'AMF, comparativement à une optimisation directe sur le solveur haute fidélité. Nous avons pu constater d'autre part le manque de robustesse de l'algorithme de par le nombre élevé d'arrêts prématurés.

Au cours du dernier chapitre, nous avons identifié les causes très probables du manque de robustesse de l'AMF. Ainsi, grâce à une simulation, nous avons désigné le bruit numérique des solveurs comme un facteur très probable d'arrêt prématuré. En effet, l'AMF tel qu'implanté est beaucoup plus sensible au bruit qu'un optimiseur classique notamment parce qu'il subit le bruit de deux solveurs qui peuvent le faire échouer à deux étapes distinctes. Nous avons pu constater de plus que l'introduction du bruit enlève toute efficacité aux corrections d'ordre élevé ce qui peut tout-à-fait expliquer les résultats expérimentaux.

Nous pouvons donc conclure sur un certain nombre de mesures à prendre lorsque l'on désire utiliser l'AMF :

- La première concerne le bruit qu'il faut minimiser au maximum pour bien conduire l'optimisation. L'abandon d'un mailleur automatique pour un maillage se déplaçant avec le profil est, par exemple, très recommandé.
- Il est de plus conseillé d'utiliser un pas de différenciation plus élevé que pour une optimisation classique afin d'obtenir des gradients plus fiables et ce, pour éviter certains arrêts prématurés.
- L'optimisation sur le modèle faible peut être court-circuitée, si elle n'aboutit pas à cause du bruit, en renvoyant comme solution l'intersection de la droite dirigée

par le gradient du modèle fort avec les bornes de confiance. Nous pourrions ainsi éviter les arrêts prématurés dûs au bruit du résolveur basse fidélité.

- Pour une première utilisation, le choix d’une constante pour la b ta correction n’est pas r solu. Il est donc conseill  d’utiliser la delta correction qui ne requiert pas de r glages.

Il reste cependant quelques aspects m ritant d’ tre explor s :

- Le r glage de la constante dans la b ta correction a  t  tr s peu  tudi . Elle repr sente pourtant une transition entre une correction purement multiplicative ($K = 0$) et une correction purement additive ($K = \infty$). Il serait peut- tre int ressant de faire ainsi un compromis entre ces deux corrections qui ne d forment pas le mod le faible avec la m me intensit .
- Afin de distinguer r ellement l’influence du bruit sur notre probl me r el, il serait int ressant de lisser les r ponses des r solveurs par des m thodes d’interpolation. Nous pourrions ainsi esp rer une convergence aussi pouss e avec l’AMF qu’avec l’optimisation directe et  tudier le r el apport de l’AMF en terme de r duction d’appels au mod le fort. Nous pourrions aussi v rifier que le bruit est bien le facteur qui condamne l’utilisation des corrections d’ordre  lev .
- Il reste enfin   appliquer la m thode sur d’autres probl mes. Nous pourrions par exemple  voluer vers des probl mes 3D. Il serait int ressant aussi d’ tudier d’autres conjonctions de solveurs basse et haute fid lit , par exemple en gardant le m me r solveur mais en modifiant la pr cision du maillage ou le niveau de convergence. L’AMF doit enfin  tre test  sur des probl mes r els avec beaucoup de contraintes pour v rifier l’efficacit  de la fonction m rite introduite en 1.3.2.2.

Pour conclure nous dirons que l’AMF est prometteur puisqu’il permet de diviser par trois le nombre d’appels au r solveur haute fid lit  pour des probl mes bien pos s. Il est cependant moins robuste qu’un optimiseur classique et m rite encore d’ tre  tudi  sur d’autres probl mes.

RÉFÉRENCES

- ALEXANDROV, N. M. (1998). On managing the use of surrogate in general nonlinear optimization and MDO. *7th Symposium on Multidisciplinary Analysis and Optimization*. AIAA/USAF/NASA/ISSMO, no. 98-4798.
- ALEXANDROV, N. M., DENNIS, J. E., LEWIS, R. M. ET TORCZON, V. (1997). A trust region framework for managing the use of approximation models in optimization. Rapport technique 97-50, ICASE.
- ALEXANDROV, N. M. ET LEWIS, R. M. (2000a). Algorithmic perspective on problem formulations in MDO. *8th Symposium on Multidisciplinary Analysis and Optimization*. AIAA/USAF/NASA/ISSMO, no. 2000-4719.
- ALEXANDROV, N. M. ET LEWIS, R. M. (2000b). Analytical and computational properties of distributed approaches to MDO. *8th Symposium on Multidisciplinary Analysis and Optimization*. AIAA/USAF/NASA/ISSMO, no. 2000-4718.
- ALEXANDROV, N. M., LEWIS, R. M., GUMBERT, C. R., GREEN, L. L. ET NEWMAN, P. A. (2000). Optimization with variable-fidelity models applied to wing design. *Proceedings of the 38th Aerospace Sciences Meeting and Exhibit*. AIAA, no. 2000-0841.
- AUDET, C. ET DENNIS, J. (2000). Analysis of generalized pattern searches. *SIAM Journal on Optimization*. To appear.
- BARTHELEMY, J.-F. M. ET HAFTKA, R. T. (1993). Approximation concepts for optimal structural design. *Structural Optimization*, 5, 129-144.

BAUER, F., GARABEDIAN, P. ET KORN, D. (1977). *Supercritical Wing Section III*. Springer-Verlag, Berlin.

BOOKER, A. J., CONN, A. R., DENNIS, J. E., FRANK, P. D., SERAFINI, D., TORCZON, V. ET TROSSET, M. (1999). A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, 17, 1–13.

CARTER, R. G. (1991). On the global convergence of trust region algorithms using inexact gradient information. *SIAM Journal on Numerical Analysis*, 28, 283–288.

CHANG, K. J., HAFTKA, R. T., GILES, G. L. ET KAO, P. J. (1993). Sensitivity-based scaling for approximating structural response. *Journal of Aircraft*, 30, 283–288.

CONN, A. R., GOULD, N. I. M. ET TOINT, P. L. (1991). A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis*, 28, 545–572.

CONN, A. R., GOULD, N. I. M. ET TOINT, P. L. (2000). *Trust-Region Methods*. Series on Optimization. SIAM.

DENNIS, J. E. ET SCHNABEL, R. B. (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, NJ.

DENNIS, JR., J. E. ET TORCZON, V. (1996). Managing approximation models in optimization. *Multidisciplinary Design Optimization — State-of-the-Art — Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization*. SIAM.

EL-ALEM, M. (1988). *A Global Convergence Theory for a Class of Trust Region Algorithms for Constrained Optimization*. Thèse de doctorat, Rice University, Department of Mathematical Science.

GAUVIN, J. (1995). *Leçons de programmation mathématique*. Éditions de l'École polytechnique de Montréal.

GAY, D. M. (1982). On convergence testing in model/trust-region algorithms for unconstrained optimization. Computing Science Technical Report 104, Bell Laboratories.

GILL, P. E., MURRAY, W. ET WRIGHT, M. H. (1981). *Practical Optimization*. Academic Press.

GOULD, N. I. M. ET TOINT, P. L. (1999). SQP methods for large-scale nonlinear programming. Rapport technique, Department of Mathematics, University of Namur.

LÉPINE, J. (1999). *Optimisation de la Représentation de Profils d'Ailes pour Application au Design Aérodynamique*. Mémoire de maîtrise, École polytechnique de Montréal.

MOHAMMADI, B. (1994). Fluid dynamics computation with NSC2KE, a user-guide, release 1.0. Rapport technique RT-0164, Institut National de Recherche en Informatique et en Automatique.

MOHAMMADI, B. ET PIRONNEAU, O. (1994). *Analysis of the K-Epsilon Turbulence Model*. John Wiley & Sons and Masson.

MORE, J. J. (1983). *Mathematical Programming : The State of the Art*, Springer-Verlag, chapitre Recent Developments in Algorithms and Software for Trust Re-

gion Methods.

PAPALAMBROS, P. A. ET WILDE, D. J. (2000). *Principles of Optimal Design*, Cambridge University Press, chapitre 7. Seconde édition, 313–324.

POWELL, M. J. D. (1975). *Nonlinear Programming*, Academic Press, New York, chapitre Convergence Properties of a Class of Minimization Algorithms. Seconde édition, 1–27.

SCHMIT, L. A. ET FARSHI, B. (1974). Some approximation concepts for structural synthesis. *AIAA Journal*, 12, 692–699.

SCHMIT, L. A. ET FLEURY, C. (1980). Structural synthesis by combining approximation concepts and dual methods. *AIAA Journal*, 18, 1252–1260.

SCHMIT, L. A. ET MIURA, H. (1976). Approximation concepts for efficient structural synthesis. Rapport technique CR-2552, NASA.

SORENSEN, D. C. (1982). Newton's method with a model trust region modifications. *SIAM Journal of Numerical Analysis*, 19, 409–426.

SPELLUCI, P. (1998). An SQP method for general nonlinear programs using only equality constrained subproblems. *Mathematical Programming*, 82, 413–448.

VANDERPLAATS, G. N. (1999). *Numerical Optimization Techniques for Engineering Design*. Vanderplaats Research & Development, Inc.