



**HAL**  
open science

# Algorithmes hiérarchiques et stratégies de jeux pour l'optimisation multidisciplinaire Application à l'optimisation de la voilure d'un avion d'affaires

Badr Abou El Majd

► **To cite this version:**

Badr Abou El Majd. Algorithmes hiérarchiques et stratégies de jeux pour l'optimisation multidisciplinaire Application à l'optimisation de la voilure d'un avion d'affaires. Mathématiques [math]. Université Nice Sophia Antipolis, 2007. Français. NNT: . tel-00529309

**HAL Id: tel-00529309**

**<https://theses.hal.science/tel-00529309>**

Submitted on 25 Oct 2010

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

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

**UNIVERSITÉ DE NICE-SOPHIA ANTIPOLIS  
FACULTÉ DES SCIENCES**

**THÈSE**

pour obtenir le titre de  
**Docteur en SCIENCES**  
de l'Université de Nice-Sophia Antipolis

Discipline : Mathématiques Appliquées

BADR ABOU EL MAJD

**Algorithmes hiérarchiques et stratégies de jeux  
pour l'optimisation multidisciplinaire  
Application à l'optimisation de la voilure d'un avion d'affaires**

Thèse préparée à l'INRIA Sophia Antipolis  
Soutenue le 20 septembre 2007 devant le jury composé de :

JACQUES BLUM	: président
JACQUES PÉRIAUX	: rapporteur
ANTOINE HENROT	: rapporteur
CATHERINE VAYSSADE	: examinatrice
JEAN-ANTOINE DÉSIDÉRI	: directeur
ABDERRAHMANE HABBAL	: co-directeur
ALFONSO RUGGIERI	: invité



A mes parents,





Pour jouir de ce bonheur qu'on cherche tant et qu'on trouve si peu, la sagesse vaut mieux que le génie, l'estime que l'admiration, et les douceurs du sentiment que le bruit de la renommée.

Jean le Rond d'Alembert - Extrait des Eloges



# Remerciements

Je tiens tout d'abord à adresser mes plus vifs remerciements aux directeurs de cette thèse, Jean Antoine désidéri et Abderrahmanne Habbal, pour leur direction avisée et exigeante, leurs qualités humaines et scientifiques, et leur soutien constant tout au long de cette thèse.

Je remercie également Ales Janka de son aide précieux au début de la thèse et de la qualité du travail qu'il m'a transmis, et j'exprime toute ma gratitude à Regis Duvigneau qui a contribué fortement à l'aboutissement de cette thèse et surtout l'aide considérable qu'il m'a apporté pour la rédaction de ce mémoire.

Je tiens à remercier sincèrement Antoine Henrot et Jacques Périaux qui m'ont fait l'honneur d'accepter de juger ce travail et d'en être les rapporteurs.

Je remercie également Jacques Blum qui m'a honoré en acceptant d'être le président de mon jury de thèse et Catherine Vayssade d'avoir accepté d'examiner mon mémoire et de faire partie du jury.

Je remercie l'INRIA Sophia Antipolis et son personnel qui ont mis à ma disposition un bon environnement pour la réalisation de ce travail. Je remercie également la région PACA et l'entreprise Piaggio Aero pour leur participation financière dans cette thèse. Et toute ma gratitude encore à Jean Antoine Désidéri de m'avoir assuré un financement supplémentaire pour finir la thèse dans de bonnes conditions.

Je tiens à remercier l'assistante du projet Opale Montserrat Argente de son aide administrative et sa sympathie, et tous mes collègues et amis que j'ai côtoyé à l'INRIA et ailleurs, et qui m'ont fait part de leurs remarques et leur encouragement pendant la période de la thèse.

J'exprime enfin un geste modeste de remerciement et de reconnaissance à toutes les personnes qui me sont les plus chères pour tous ce qu'elles m'ont apporté, en particulier mon père qui m'a fait aimer les mathématiques «à sa manière originelle» dès mon jeune âge, et ma mère pour son encouragement sans égal, ma femme Rachida pour son infaillible soutien et son incroyable patience, et mon fils Ilyas, qui a été conçu et né pendant la réalisation de cette thèse, de son calme et son sourire magnifique.



# Table des matières

<b>1</b>	<b>Introduction générale</b>	<b>13</b>
<b>2</b>	<b>Principaux éléments de la méthodologie numérique</b>	<b>17</b>
2.1	Introduction	17
2.2	Résolution des équations d'Euler par une méthode de volumes finis	18
2.2.1	Lois de conservation	18
2.2.2	Approximation spatiale décentrée	19
2.2.3	Conditions aux bords	21
2.2.4	Intégration en temps	22
2.3	Techniques de paramétrisation	23
2.3.1	Approche discrète	23
2.3.2	Approche polynomiale	24
2.3.2.1	Paramétrisation de Bézier	24
	Les courbes polynomiales :	24
	Algorithme de De Casteljau :	25
	Les propriétés des courbes de Bézier :	25
	La surface et le volume de Bézier :	27
2.3.2.2	Généralisations de la paramétrisation de Bézier	28
2.3.3	Approche Free-form deformation (FFD)	29
2.4	Outils de déformation de maillage	30
2.4.1	Conditions de transpiration	32
2.4.2	Free-form deformation	34
2.5	Méthodes d'optimisation	35
2.5.1	Généralités	35
2.5.2	Méthodes de type simplexe	36
2.5.2.1	Méthode de Spendley, Hext et Himsworth	37
2.5.2.2	Méthode de Nelder-Mead	38
2.5.2.3	Méthode MSA (Multi-directional Search Algorithm)	40
2.6	Conclusion	41
<b>3</b>	<b>Algorithmes hiérarchiques en optimisation de Forme</b>	<b>43</b>
3.1	Introduction	43
3.2	Algorithmes multigrilles	43
3.2.1	Modèle à résoudre	44
3.2.2	Opérateurs de transfert	45
3.2.3	Enrichissement progressif de maillage	45

3.2.4	Méthode bigrille idéale . . . . .	47
3.2.5	Algorithme multigrille (MG) . . . . .	49
3.2.6	Méthode multigrille complète (FMG) . . . . .	49
3.3	Approches hiérarchiques en optimisation de forme . . . . .	49
3.3.1	Motivations et illustration par un problème modèle . . . . .	50
3.3.2	Méthode multiniveau basée sur la paramétrisation . . . . .	52
3.3.2.1	Représentation de forme . . . . .	52
3.3.2.2	Construction des paramétrisations emboîtées . . . . .	53
3.3.2.3	Enrichissement progressif de la paramétrisation . . . . .	55
3.3.2.4	Algorithme d'optimisation V-cycle . . . . .	56
3.3.2.5	Algorithme multiniveaux d'optimisation de forme complet (FMOSA) . . . . .	57
3.4	Conclusion . . . . .	58
<b>4</b>	<b>Expérimentation numérique d'algorithmes multiniveaux</b>	<b>59</b>
4.1	Introduction . . . . .	59
4.2	Problème modèle . . . . .	59
4.2.1	Problème variationnel . . . . .	59
4.2.2	Résultats numériques . . . . .	63
4.2.2.1	Description du cas test . . . . .	63
4.2.2.2	Résultats . . . . .	67
4.3	Application à l'optimisation de la voilure d'un avion d'affaires . . . . .	71
4.3.1	Description du cas test . . . . .	71
4.3.2	Résultats numériques . . . . .	74
4.4	Optimisation appliquée à un jet d'affaires supersonique . . . . .	85
4.4.1	Description du cas test . . . . .	85
4.4.2	Résultats numériques . . . . .	87
4.4.2.1	Optimisation de la voilure . . . . .	87
4.4.2.2	Optimisation du nez . . . . .	94
4.5	Conclusion . . . . .	101
<b>5</b>	<b>Adaptation de la paramétrisation</b>	<b>103</b>
5.1	Introduction . . . . .	103
5.2	Approche 1D : cas d'une paramétrisation par courbe de Bézier . . . . .	104
5.2.1	Formulation du problème . . . . .	104
5.2.1.1	Contraintes géométriques . . . . .	104
5.2.1.2	Contrainte de meilleure approximation . . . . .	106
5.2.1.3	Normalisation des équations . . . . .	107
5.2.1.4	Calcul du critère de régularisation et de son gradient . . . . .	108
5.2.2	Développement formel des calculs . . . . .	111
5.2.3	Résultats numériques . . . . .	115
5.2.3.1	Validation du calcul formel du gradient de $J_2$ . . . . .	115
5.2.3.2	Auto-adaptation de la paramétrisation . . . . .	117
5.3	Approche 3D : cas d'une paramétrisation par Free-Form Deformation . . . . .	121
5.3.1	Première approche . . . . .	122
5.3.1.1	Procédure d'adaptation . . . . .	122
	Algorithme de couplage optimisation-adaptation : . . . . .	122

	Adaptation du support : . . . . .	123
	Contrainte de meilleure approximation : . . . . .	123
	Critère de régularisation : . . . . .	124
5.3.1.2	Résultats numériques . . . . .	124
5.3.2	Deuxième approche . . . . .	130
5.3.2.1	Procédure d'adaptation . . . . .	130
5.3.2.2	Résultats numériques . . . . .	131
	Optimisation avec une paramétrisation grossière ( $3 \times 1 \times 1$ ) : . .	132
	Optimisation avec une paramétrisation moyenne ( $6 \times 1 \times 1$ ) : .	136
	Optimisation avec une paramétrisation fine ( $9 \times 1 \times 1$ ) : . . . . .	140
5.4	Conclusion . . . . .	144
<b>6</b>	<b>Stratégies de jeux en optimisation multidisciplinaire : Aérodynamique versus Coques minces</b>	<b>145</b>
6.1	Introduction . . . . .	145
6.2	Deux problèmes d'optimisation de forme . . . . .	148
6.2.1	Optimisation aérodynamique . . . . .	148
6.2.2	Optimisation structurale . . . . .	148
6.3	Deux approches d'optimisation multidisciplinaire . . . . .	148
6.3.1	Jeu de Nash . . . . .	149
6.3.2	Jeu de Stackelberg . . . . .	149
6.4	Stratégies d'optimisation . . . . .	150
6.4.1	Paramétrisation de forme . . . . .	150
6.4.2	Partage de territoires entre les disciplines . . . . .	151
6.4.3	Algorithmes de couplage . . . . .	151
6.5	Expérimentation numérique . . . . .	153
6.5.1	Données aérodynamiques . . . . .	153
6.5.2	Données structurales . . . . .	153
6.5.3	Résultats . . . . .	154
6.5.3.1	Première stratégie d'optimisation . . . . .	155
6.5.3.2	Seconde stratégie d'optimisation . . . . .	173
6.6	Conclusion . . . . .	182
<b>7</b>	<b>Conclusions et perspectives</b>	<b>183</b>
<b>8</b>	<b>Bibliographie</b>	<b>185</b>
<b>A</b>	<b>Continuité des transferts du hessien dans une optimisation de forme multiniveau</b>	<b>195</b>
A.1	Introduction . . . . .	195
A.2	Opérateurs de transfert . . . . .	195
A.3	Optimisation à support fixe . . . . .	197
A.4	Auto-adaptation de la paramétrisation . . . . .	200
A.5	Algorithme d'optimisation . . . . .	202
A.6	Expérimentations numériques . . . . .	203
A.7	Conclusion . . . . .	207
<b>B</b>	<b>Paramétrisations non emboîtées</b>	<b>209</b>



<b>C</b>	<b>Analyse structurale</b>	<b>215</b>
C.1	Test de validation . . . . .	215
C.1.1	Géométrie . . . . .	215
C.1.2	Résultats . . . . .	216
C.2	Étapes d'analyse avec le Code Aster . . . . .	217
<b>D</b>	<b>Fonctions radiale de base et Krigeage</b>	<b>223</b>
D.1	Problème d'interpolation . . . . .	223
D.2	Méthode des fonctions radiale de base . . . . .	223
D.3	Modèle de Krigeage . . . . .	224

# Chapitre 1

## Introduction générale

La conception et l'optimisation d'un avion en milieu industriel est un processus complexe, multidisciplinaire et itératif que l'on illustre schématiquement à la figure (1.1). À partir d'un concept initial d'avant-projet proposé par les bureaux d'études, on réalise une succession d'optimisations et de vérifications par des analyses scientifiques fines se rapportant à plusieurs disciplines. Ces études plus basiques s'appuient de nos jours sur les techniques les plus récentes de la simulation numérique et du calcul haute-performance et sont le plus souvent réalisées par les industriels en collaboration avec les laboratoires de recherche.

Par exemple, dans le cas de l'optimisation d'une voilure d'avion, on optimise d'abord la géométrie pour atteindre les performances aérodynamiques visées. Cette partie s'effectue en utilisant une procédure d'optimisation de forme qui inclut plusieurs éléments : des codes numériques de résolution des équations de la mécanique des fluides, des outils de paramétrisation de forme, des techniques de mise à jours du maillage, des optimiseurs appropriés, etc. Le concept est ensuite "vérifié" (éventuellement corrigé) par des calculs statiques en "chargements extrêmes" et de mécanique du vol. Dans une deuxième partie, on ré-optimise la forme vis-à-vis des critères structuraux en faisant appel à un code numérique de mécanique des structures. De nouvelles vérifications sont ensuite effectuées pour valider le concept vis-à-vis de critères de stabilité aéroélastique, fatigue, flambage, etc. Toute la boucle est refaite, éventuellement plusieurs fois, pour satisfaire au mieux toutes les contraintes ci-dessus, et d'autres encore, notamment de fabrication.

L'automatisation de cette boucle d'optimisation multidisciplinaire par la simulation numérique est un sujet vaste et complexe qui mobilisera de nombreuses équipes de recherche et développement en ingénierie concurrente, analyse numérique et informatique. Cette thèse vise à contribuer à la partie scientifique associée au développement des méthodes innovantes en analyse numérique pour la réalisation des optimisations aérodynamiques et couplées entre l'aérodynamique et la mécanique des structures.

Dans le premier volet, on s'intéresse à des méthodologies numériques en optimisation de forme pour des applications où chaque évaluation de la fonction coût nécessite la résolution d'équations aux dérivées partielles. Notre objectif est d'améliorer la convergence itérative et la précision de la solution optimale des algorithmes d'optimisation de forme en aérodynamique pour lesquels une EDP est résolue (ex. les équations d'Euler) comme une

équation d'état pour évaluer les coefficients aérodynamiques de la géométrie à optimiser. Pour cela, on a développé des algorithmes d'optimisation multiniveaux, à l'instar des méthodes multigrilles particulièrement performantes en convergence itérative, qui s'appuient sur une hiérarchie de paramétrisations emboîtées. On a étudié ensuite le concept d'adaptation de paramétrisation, notion analogue à l'adaptation de maillage, pour améliorer la précision et la convergence itérative.

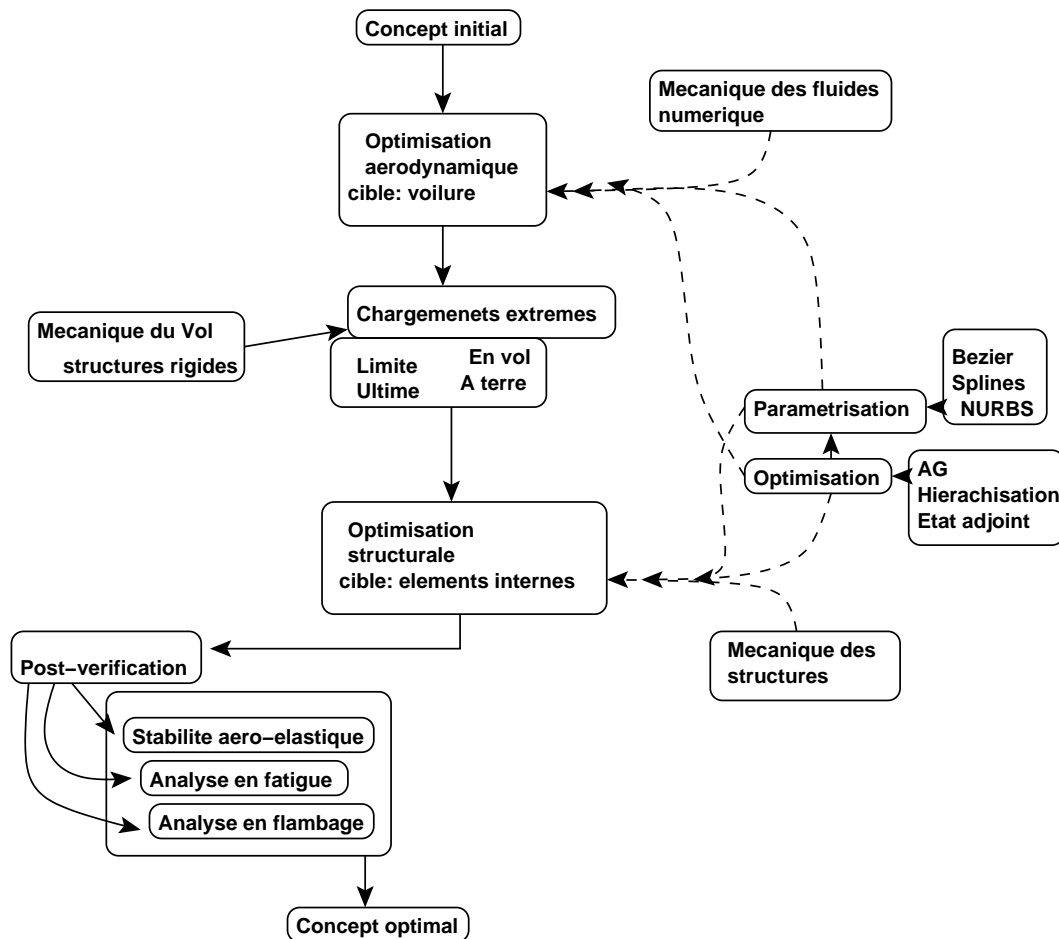


FIGURE 1.1 – Schéma d'optimisation d'un concept de voilure d'avion

L'optimisation multidisciplinaire (OMD) concerne le processus de conception des matériaux et des structures qui doivent être conformes à plusieurs critères, parfois antagonistes. En raison de la nature de la concurrence des critères, des méthodes raisonnables non-arbitraires devraient être employées pour conduire l'OMD, et clairement le paradigme de la théorie des jeux offre un tel cadre. On résout, dans le deuxième volet, un problème d'optimisation concurrente où le concepteur aérodynamique interagit avec le concepteur structural parallèlement dans un jeu symétrique de Nash ou hiérarchiquement dans un jeu de Stackelberg. L'objectif est d'étudier des algorithmes d'optimisation pour cette optimisation couplée fluide-structure dans un régime où le critère aérodynamique est prépondérant.

L'organisation de la thèse est comme suit. Au deuxième chapitre, on présente les principaux éléments de notre méthodologie numérique pour traiter un problème d'optimisation de forme en aérodynamique. On décrit brièvement les équations d'Euler qui régissent l'écoulement et leur résolution par la méthode des volumes finis. On expose ensuite les techniques de représentation de la forme et de la mise à jour du maillage. On présente enfin les méthodes d'optimisation utilisées pour minimiser la fonctionnelle coût.

Au troisième chapitre, On rappelle le formalisme et les principales propriétés des algorithmes hiérarchiques en résolution pure (méthodes multigrilles). Ces algorithmes se basent sur une hiérarchie de maillages correspondant aux différents domaines de fréquences et des transferts d'informations de grille à grille. On présente ensuite notre approche hiérarchique en optimisation de forme, en s'inspirant des algorithmes multigrilles, qui s'appuie sur une hiérarchie de paramétrisations en utilisant des opérateurs de transfert entre les espaces paramétriques.

Le quatrième chapitre est consacré à l'expérimentation numérique de nos stratégies d'optimisation hiérarchique sur différents cas tests d'optimisation. On a utilisé, dans la première application, un problème simple de calcul des variations formulé comme un problème d'optimisation géométrique. On a testé par la suite notre approche hiérarchique sur des problèmes industriels plus réalistes ; le problème d'optimisation d'une voilure d'avion d'affaire en régime transsonique (réduction de la traînée avec une contrainte sur la portance), et un problème d'optimisation d'un avion d'affaire supersonique (réduction d'un critère de bang sonique avec une contrainte sur la portance et sur la traînée).

Le cinquième chapitre présente un processus d'adaptation de paramétrisation, notion analogue à l'adaptation du maillage, dans deux contextes différents : quand la forme est représentée par une paramétrisation de Bézier classique, le processus d'adaptation se traduit par la régularisation du polygone de contrôle en minimisant un critère de variation totale. Ce concept est testé en utilisant le problème modèle de calcul des variations. Le deuxième contexte est quand la forme est représentée par la méthode Free-Form Deformation ; le processus d'adaptation consiste à modifier la carte FFD afin de repositionner les points de contrôle autour de la forme d'une façon régulière. On teste cette approche sur un problème d'optimisation d'une voilure d'avion d'affaire en régime transsonique.

Au sixième chapitre, on traite un problème d'optimisation couplée entre l'aérodynamique et la mécanique des structures dans un régime où l'aérodynamique est le plus important. Ce problème d'optimisation multidisciplinaire est résolu en utilisant la théorie des jeux. On décrit brièvement les deux problèmes d'optimisation aérodynamique et structurale, et ensuite notre approche d'optimisation concurrente où les deux critères à minimiser sont considérés par des joueurs qui interagissent entre eux d'une façon symétrique (Jeu de Nash) ou d'une façon hiérarchique (Jeu de Stackelberg). On évoque la question de partage de territoires dont le choix a une importance cruciale vu la nature des critères qui sont antagonistes. On termine le chapitre par plusieurs expérimentations numériques sur un problème d'optimisation d'une voilure d'avion d'affaire.

On expose enfin au dernier chapitre quelques conclusions et perspectives.



## Chapitre 2

# Principaux éléments de la méthodologie numérique

### 2.1 Introduction

Au milieu des années 70, les chercheurs commencent à explorer l'utilité des techniques d'optimisation numérique pour les composantes d'avions [54][55][70][113][114]. Ces premières études se sont focalisées sur les profils d'ailes d'avions (en deux dimensions d'espace) en utilisant des modèles simplifiés pour l'analyse et une évaluation du gradient par la différence finie. L'incapacité de ces modèles à traiter les problèmes non linéaires a limité leur application.

L'évolution de l'architecture des ordinateurs, ainsi que les améliorations considérables de la vitesse et de la capacité de la mémoire, ont motivé la mise en oeuvre de plusieurs travaux qui traitent le cycle complet de l'optimisation de forme en aérodynamique, en combinant la dynamique des fluides numérique (CFD), l'analyse des sensibilités et l'optimisation numérique. Malgré ces avancées, le temps CPU des simulations reste très grand lorsqu'on considère des modèles non linéaires. Plusieurs méthodologies ont été développées pour réduire le temps de calcul excessif, on cite par exemple l'analyse simultanée en aérodynamique et l'optimisation de forme (SAADO) [52], les méthodes multigrilles [17][19][28][50][118], les méthodes one-shot [7][64][107], les méthodes de pas de temps fictif [57], et le calcul parallèle [95].

Pour traiter des configurations complexes, l'utilisation d'un maillage structuré est très difficile, s'il n'est pas impossible. Donc, pour traiter de telles géométries complexes, plusieurs méthodes de décomposition de domaine ont été proposées pour réaliser un calcul multi-bloc. Une alternative, pour étudier des formes complexes, consiste à utiliser des maillages non-structurés [13][39]. En effet, le maillage non-structuré permet une grande flexibilité et son enrichissement (ou son adaptation) se fait localement sans affecter les autres régions du maillage. Plusieurs approches pour l'adaptation du maillage ont été proposées en optimisation de forme, on cite par exemple [15][21][48][82].

On se focalise dans ce chapitre sur la présentation des principaux éléments de notre méthodologie numérique, à savoir la résolution des équations d'Euler, les méthodes de mise

à jour du maillage, les techniques de paramétrisation, et enfin les méthodes d'optimisation utilisées. Ces éléments vont servir à étudier un problème d'optimisation de forme en aérodynamique qui consiste à chercher une forme optimale en minimisant une fonctionnelle coût vis-à-vis des contraintes géométriques et aérodynamiques. La procédure d'optimisation de forme fait appel à une technique de paramétrisation pour représenter la géométrie, dont le choix doit satisfaire à plusieurs exigences au niveau de l'efficacité et de la flexibilité, et à une technique de déformation de maillage pour la mise à jour du maillage. On note aussi que la fonctionnelle coût dépend des variables physiques, ce qui nécessite la résolution d'une équation d'état régissant l'écoulement.

## 2.2 Résolution des équations d'Euler par une méthode de volumes finis

On décrit brièvement dans cette section la méthode numérique de résolution des équations d'Euler compressible qui a été développée au sein du projet SINUS (INRIA Sophia Antipolis) [36][28]. On présente seulement le cas 2D pour des raisons de simplicité.

### 2.2.1 Lois de conservation

Soit  $\Omega$  le domaine de calcul dont la frontière est  $\Gamma = \Gamma_B \cup \Gamma_\infty$ .  $\Gamma_B$  représente les parois et  $\Gamma_\infty$  est la frontière à l'infini délimitant le domaine de l'écoulement étudié. Le système des équations d'Euler en deux dimensions s'écrit en formulation conservative :

$$\frac{\partial W}{\partial t} + \vec{\nabla} \cdot \vec{F}(W) = 0 \quad (2.1)$$

avec

$$W = \begin{pmatrix} \rho \\ \rho \vec{U} \\ E \end{pmatrix}$$

et  $\vec{F}(W) = (F_1(W), F_2(W))^T$  est le vecteur flux dont les composantes sont données par

$$F_1(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E + p) \end{pmatrix}, \quad F_2(W) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E + p) \end{pmatrix}.$$

$\rho$  est la masse volumique,  $\vec{U} = (u, v)^T$  est le vecteur vitesse,  $E$  est l'énergie totale par unité de volume, et la pression  $p$  vérifie la loi d'état des gaz parfaits :

$$p = (\gamma_p - 1) \left( E - \frac{1}{2} \rho \|\vec{U}\|^2 \right)$$

où  $\gamma_p$  est le rapport des capacités calorifiques à pression et à volume constant ( $\gamma_p = 1.4$  pour l'air).

## 2.2.2 Approximation spatiale décentrée

Le domaine de calcul  $\Omega$  est discrétisé par une triangulation  $\mathcal{T}_h$ , où  $h$  est la longueur maximale des arêtes de  $\mathcal{T}_h$ . On considère un sommet  $s_i$  de  $\mathcal{T}_h$  et ses voisins  $N(i)$ .

L'approximation repose sur un "maillage dual" obtenu en traçant les médianes des triangles, et en identifiant ensuite autour de chaque noeud  $i$  (ou sommet  $s_i$ ) une cellule de contrôle  $C_i$  dont les arêtes sont des portions des médianes des triangles dont le noeud  $i$  est un sommet (voir Fig. 2.1).

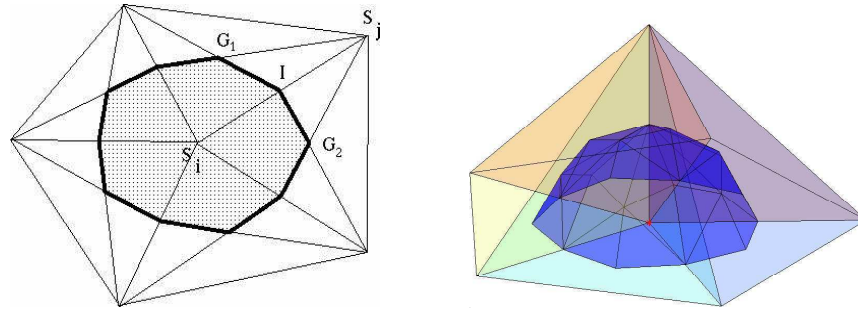


FIGURE 2.1 – Cellule de contrôle  $C_i$  d'un maillage triangulaire 2D (à gauche), demi-cellule de contrôle en 3D (à droite)

Le bord de la cellule  $C_i$  est noté par  $\partial C_i$  et le vecteur  $\vec{v}_i = (v_{ix}, v_{iy})$  est la normale unitaire extérieure à  $\partial C_i$ . L'union de tous ces cellules constitue une discrétisation de  $\Omega$  souvent qualifié de maillage dual de  $\mathcal{T}_h$  :

$$\Omega_h = \bigcup_{i=1}^{N_v} C_i,$$

où  $N_v$  est le nombre des sommets de  $\mathcal{T}_h$ .

En appliquant la formule de Green à l'intégrale sur  $C_i$  de l'équation (2.1), il vient :

$$\begin{aligned} \int \int_{C_i} \frac{\partial W}{\partial t} d\vec{x} + \sum_{j \in N(i)} \int_{\partial C_{ij}} \vec{F}(W) \cdot \vec{v}_i d\sigma \\ + \int_{\partial C_i \cap \Gamma_B} \vec{F}(W) \cdot \vec{n}_i d\sigma + \int_{\partial C_i \cap \Gamma_\infty} \vec{F}(W) \cdot \vec{n}_i d\sigma = 0, \end{aligned} \quad (2.2)$$

où  $\partial C_{ij} = \partial C_i \cap \partial C_j$ .

L'approximation volumes finis au premier ordre du premier terme de (2.2) s'écrit :

$$\frac{W_i^{n+1} - W_i^n}{\Delta t} + \sum_{j \in N(i)} \phi_F(W_i^n, W_j^n, \vec{v}_{ij}) \quad (2.3)$$



avec  $\phi_F$  la fonction de flux numérique telle que :

$$\phi_F(W_i, W_j, \vec{v}_{ij}) \approx \int_{\partial C_{ij}} \vec{F}(W) \cdot \vec{v}_i d\sigma \quad , \quad \vec{v}_{ij} = \int_{\partial C_{ij}} \vec{v}_i d\sigma \quad (2.4)$$

Le flux numérique (2.4) rend le schéma conservatif, si pour toutes les arêtes  $[s_i, s_j]$  la condition suivante est vérifiée :

$$\phi_F(W_i, W_j, \vec{v}_{ij}) = -\phi_F(W_j, W_i, \vec{v}_{ji}). \quad (2.5)$$

En utilisant le solveur de Riemann approché de Roe [96], on obtient :

$$\phi_F(W_i, W_j, \vec{v}_{ij}) = \frac{\vec{F}(W_i) + \vec{F}(W_j)}{2} \cdot \vec{v}_{ij} - |A_R(W_i, W_j, \vec{v}_{ij})| \frac{(W_j - W_i)}{2}, \quad (2.6)$$

où  $A_R(W_i, W_j, \vec{v}_{ij}) = \left( \frac{\partial F}{\partial W}(W_i, W_j, \vec{v}_{ij}) \right)_R$  est la matrice de Roe qui vérifie la condition suivante :

$$A_R(W_i, W_j, \vec{v}_{ij})(W_j - W_i) = F(W_j, \vec{v}_{ij}) - F(W_i, \vec{v}_{ij}) \quad (2.7)$$

avec :

$$F(W, \vec{v}_{ij}) = \vec{F}(W) \cdot \vec{v}_{ij} \quad (2.8)$$

Donc, le flux numérique (2.6) peut être formulé comme :

$$\phi_F(W_i, W_j, \vec{v}_{ij}) = F(W_j, \vec{v}_{ij}) - A_R^+(W_i, W_j, \vec{v}_{ij})(W_j - W_i) \quad (2.9)$$

ou comme :

$$\phi_F(W_i, W_j, \vec{v}_{ij}) = F(W_j, \vec{v}_{ij}) + A_R^-(W_i, W_j, \vec{v}_{ij})(W_j - W_i) \quad (2.10)$$

où  $A_R(W_i, W_j, \vec{v}_{ij}) = A_{\vec{v}_{ij}}(\tilde{W})$ , avec  $\tilde{W}$  est tel que :

$$\begin{cases} \tilde{W} &= (\tilde{\rho}, \tilde{\rho}\tilde{u}, \tilde{\rho}\tilde{v}, \tilde{E})^T \\ \tilde{\rho} &= (\sqrt{\rho_1}\rho_1 + \sqrt{\rho_2}\rho_2)/(\sqrt{\rho_1} + \sqrt{\rho_2}) \\ \tilde{u} &= (\sqrt{\rho_1}u_1 + \sqrt{\rho_2}u_2)/(\sqrt{\rho_1} + \sqrt{\rho_2}) \\ \tilde{v} &= (\sqrt{\rho_1}v_1 + \sqrt{\rho_2}v_2)/(\sqrt{\rho_1} + \sqrt{\rho_2}) \\ \tilde{H} &= (\sqrt{\rho_1}H_1 + \sqrt{\rho_2}H_2)/(\sqrt{\rho_1} + \sqrt{\rho_2}) \end{cases} \quad (2.11)$$

où  $H = \frac{\gamma p}{(\gamma-1)\rho} + \frac{u^2+v^2}{2}$  est l'enthalpie totale par unité de volume.

D'autre part, on a :

$$A^\pm(\tilde{W}) = T(\tilde{W})\Lambda^\pm(\tilde{W})T^{-1}(\tilde{W}) \quad (2.12)$$

avec :

$$\Lambda = \left( \vec{U} \cdot \vec{v} - c, \vec{U} \cdot \vec{v}, \vec{U} \cdot \vec{v}, \vec{U} \cdot \vec{v} + c \right)^T, \quad (2.13)$$

où  $c = \sqrt{\gamma \frac{p}{\rho}}$  est la vitesse du son.

Le passage à l'ordre 2 se réalise par une adaptation de la technique MUSCL de Van Leer [68], dans laquelle les vecteurs  $W_i$  and  $W_j$  sont remplacés dans (2.4) par des vecteurs  $W_{ij}$  et  $W_{ji}$  obtenus par extrapolation linéaire à l'interface ( $I_{ij}$ ) entre les cellules  $C_i$  et  $C_j$  :

$$\tilde{W}_{ij} = \tilde{W}_i + \frac{1}{2}(\vec{\nabla}\tilde{W})_i \cdot s_i \vec{s}_j \quad , \quad \tilde{W}_{ji} = \tilde{W}_j - \frac{1}{2}(\vec{\nabla}\tilde{W})_j \cdot s_i \vec{s}_j, \quad (2.14)$$

où  $\tilde{W} = (\rho, \vec{U}, p)^T$  (voir Fig 2.2).

Si nous supposons que  $\tilde{W}$  est P1 sur chaque tétraèdre  $T$ , nous pouvons donc calculer des

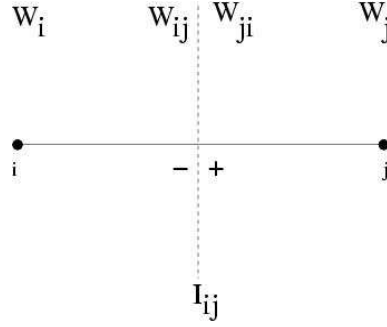


FIGURE 2.2 – Positions des états  $W_{ij}$  and  $W_{ji}$  entre les cellules  $C_i$  et  $C_j$

gradients P1-Galerkin constants sur chaque tétraèdre  $T$ . On en déduit les gradients nodaux ( $\vec{\nabla}\tilde{W}_i$  et  $\vec{\nabla}\tilde{W}_j$ ) en moyennant les gradients calculés sur l'ensemble des tétraèdres contenant le sommet considéré, c'est-à-dire :

$$(\vec{\nabla}\tilde{W})_i = \frac{\int \int_{C_i} \vec{\nabla}\tilde{W}|_T d\vec{x}}{\int \int_{C_i} d\vec{x}} = \frac{1}{\text{aire}(C_i)} \sum_{T \in C_i} \frac{\text{area}(T)}{3} \sum_{k=1, k \in T}^3 (\tilde{W})_k \vec{\nabla}\phi_k^T \quad (2.15)$$

où  $\phi_k^T(x, y, z)$  est la fonction de base de type P1 définie au sommet  $S_k$  du tétraèdre  $T$ .

### 2.2.3 Conditions aux bords

Les deux termes de la seconde ligne de l'équation (2.2) correspondent aux conditions aux limites du problème.

On considère les deux situations suivantes :

- *paroi*. On impose à  $\Gamma_B$  la condition de glissement  $\vec{U} \cdot \vec{n} = 0$ , avec  $\vec{n}$  est le vecteur normal sortant à  $\Gamma_B$ .

Le terme de bord glissant est les suivant :

$$\int_{\partial C_i \cap \Gamma_w} \vec{F}(W) \cdot \vec{n}_i d\sigma = p_i \cdot \text{aire}(\partial C_i) \begin{pmatrix} 0 \\ \tilde{n}_{ix} \\ \tilde{n}_{iy} \\ 0 \end{pmatrix} \quad (2.16)$$

où  $p_i$  est la pression moyenne sur  $\partial C_i$

- *domaine à l'infini*. On évalue des flux à travers cette frontière à l'aide d'une technique de décentrage introduite par l'utilisation du flux numérique de Steger-Warming. L'écoulement uniforme à l'infini ( $\Gamma_\infty$ ) a les caractéristiques suivantes :

$$\rho_\infty = 1, \vec{U}_\infty = (\cos \alpha, \sin \alpha)^T \quad \text{avec} \quad \|\vec{U}_\infty\| = 1, p_\infty = \frac{1}{\gamma M_\infty^2} \quad (2.17)$$

avec  $\alpha$  est l'angle d'incidence et  $M_\infty$  est le nombre de Mach à l'infini.

## 2.2.4 Intégration en temps

On choisit la méthode d'Euler implicite, décrite par Fezoui et Stoufflet [45], pour intégrer en temps l'équation suivante :

$$\text{aire}(C_i) \frac{dW_i^n}{dt} + \Psi(W_i^n) = 0 \quad , \quad i = 1, \dots, N_v, \quad (2.18)$$

où  $W_i^n = W(\vec{x}_i, t^n)$ ,  $t^n = n\Delta t^n$ , et :

$$\Psi(W_i^n) = \sum_{j \in N(i)} \phi_F(W_{ij}, W_{ji}, \vec{v}_{ij}) + \int_{\partial C_i \cap \Gamma} \vec{F}(W) \cdot \vec{n}_i d\sigma. \quad (2.19)$$

Le schéma en temps s'écrit comme suit :

$$\frac{\text{aire}(C_i)}{\Delta t^n} (W_i^{n+1} - W_i^n) + \Psi(W_i^{n+1}) = 0 \quad , \quad i = 1, \dots, N_v \quad (2.20)$$

En linéarisant au premier ordre le terme  $\Psi(W_i^{n+1})$  :

$$\Psi(W_i^{n+1}) = \Psi(W_i^n) + \Psi'(W_i^n)(W_i^{n+1} - W_i^n) \quad (2.21)$$

où  $\Psi'$  est le jacobien de l'approximation spatiale.  
on obtient la formulation suivante :

$$\left( \frac{\text{aire}(C_i)}{\Delta t^n} + \Psi'(W_i^n) \right) (W_i^{n+1} - W_i^n) = -\Psi(W_i^n) \quad (2.22)$$

Ceci est équivalent à la résolution du système linéaire suivant :

$$A_h \delta W = f_h \quad (2.23)$$

avec  $A_h = \frac{\text{aire}(C_i)}{\Delta t^n} + \Psi'(W_i^n)$ ,  $f_h = -\Psi(W_i^n)$  et  $\delta W = W_i^{n+1} - W_i^n$ .

Ce schéma simplifié d'intégration en temps s'identifie à une méthode de Newton lorsque le pas de temps tend vers l'infini. En effet, la convergence itérative est alors quadratique [106] lorsque le jacobien de l'approximation décentré est d'ordre 2. Cependant, lorsque l'approximation du jacobien est d'ordre 1, il s'agit seulement d'une méthode de quasi-Newton ; néanmoins, la matrice  $A_h$  est alors à diagonale dominante permettant l'utilisation des méthodes de relaxation pour la résolution du système linéaire. On utilise en pratique l'itération de Jacobi ou de Gauss-Seidel.

## 2.3 Techniques de paramétrisation

La paramétrisation de la forme est l'un des points essentiels dans la phase préliminaire de conception d'une géométrie. Le choix de la technique de paramétrisation en optimisation de forme, lorsqu'elle est couplée avec le code d'analyse aérodynamique dans un processus itératif pour obtenir la forme optimale vis-à-vis d'un critère aérodynamique et des contraintes, a un impact majeur sur l'efficacité du résultat final. Plusieurs travaux ont étudié l'effet de la paramétrisation de forme dans un processus de conception. Parmi d'autres, on cite le travail de Wu et al [119], où on trouve une comparaison entre trois représentations géométriques dans le cas d'un problème inverse en turbomachine. On cite aussi le travail de Song et Keane [104] qui utilisent deux méthodes de paramétrisations (les courbes de B-splines et des fonctions de base orthogonales), en cherchant à estimer leurs capacités à représenter trois formes différentes d'ailes d'avion.

Une telle paramétrisation doit répondre en général à trois exigences. La première est la flexibilité, donc la capacité de représenter un large spectre de différentes formes. La seconde est l'efficacité de la méthode en terme de convergence vers la forme optimale. Enfin, la méthode considérée doit s'appuyer sur un petit nombre de paramètres, car le nombre de paramètres affecte considérablement le temps CPU du processus d'optimisation.

On présente ci-dessous un aperçu global des techniques de paramétrisation les plus utilisées [98].

### 2.3.1 Approche discrète

L'approche discrète utilise les coordonnées des points du bord (voir figure (2.3)) comme paramètres d'optimisation. Cette approche est facile à implémenter. Cependant, il est difficile de conserver la régularité de la géométrie. En plus, le nombre de variables de conception varie selon la nature des problèmes, il peut atteindre, par exemple, des milliers pour des problèmes tridimensionnels [81], ce qui conduit à considérer une méthode de gradient avec une équation adjointe qui connaît plusieurs limitations.

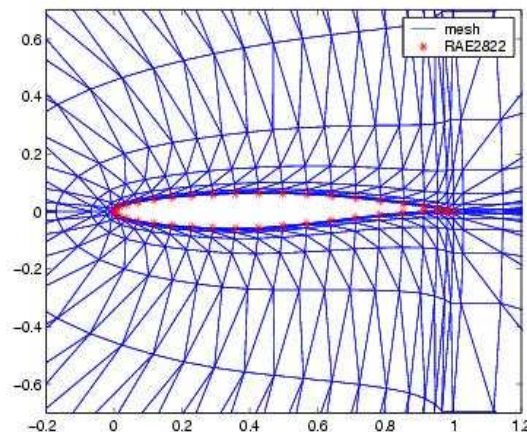


FIGURE 2.3 – Points de contrôle du RAE2822 (approche discrète)

### 2.3.2 Approche polynomiale

L'utilisation de ce genre d'approche permet de réduire considérablement le nombre de variables d'optimisation, en décrivant la forme d'une manière compacte. Dans la figure (2.4), on montre la paramétrisation de l'extrados et l'intrados du profil d'aile d'avion NACA0012 par 16 points de contrôle. En particulier, les courbes de Bézier et B-splines sont les plus utilisées pour l'optimisation de forme de profils d'aile.

On présente ci-dessous les représentations polynomiales les plus connues, à savoir les B-splines, les NURBS (Non Uniform Rational B-Splines), et avec plus de détails les courbes de Bézier qui ont été utilisées comme un outil de paramétrisation pendant la thèse. On trouve dans [42][100] plus de détails à ce sujet.

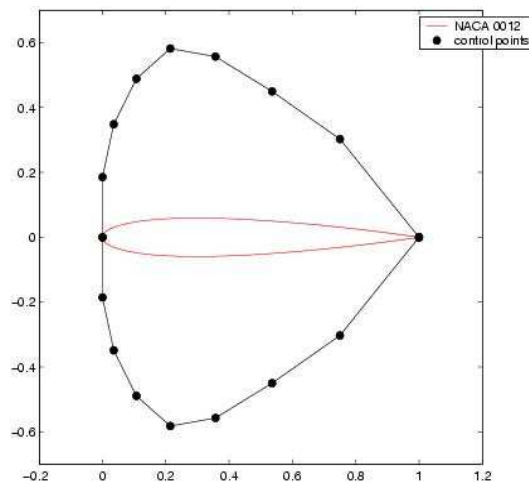


FIGURE 2.4 – Points de contrôle du NACA0012 (approche polynomiale)

#### 2.3.2.1 Paramétrisation de Bézier

La paramétrisation de Bézier est une formulation mathématique utilisant une représentation polynomiale, basée sur des polynômes de Bernstein au lieu de la base canonique. Elle est fréquemment utilisée en infographie et en modélisation géométrique. Cette représentation de Bézier a été développée par l'ingénieur Pierre Bézier en 1962 dans les bureaux d'étude de Renault.

**Les courbes polynomiales :** En général, une courbe peut être décrite comme un polynôme sous la forme,

$$P_n(t) = \sum_{i=0}^n a_i t^i \quad (2.24)$$

avec  $n$  est le degré de la paramétrisation,  $\{t^k\}(k = 0, \dots, n)$  est la base canonique, et  $\{a_i\}(i = 0, \dots, n)$  sont les coefficients du polynôme.

La représentation de Bézier représente une formulation équivalente, en exprimant le polynôme  $P_n(t)$  dans la base des polynômes de Bernstein. Donc, on aboutit à ce qu'on appelle par la courbe de Bézier,

$$P_n(t) = \sum_{i=0}^n B_i^n(t) P_i \quad (2.25)$$

avec  $n + 1$  est le nombre des points de contrôles, et  $B_i^n(t)$  est le polynôme de Bernstein de degré  $n$ . Les  $\{P_i\}(i = 0, \dots, n)$  sont les points de contrôle qui sont utilisés comme variables d'optimisation.

Les polynômes de Bernstein sont définis comme suit,

$$B_i^n(t) = C_n^i t^i (1-t)^{n-i} \quad (i = 0, \dots, n) \quad (2.26)$$

avec  $C_n^i = \frac{n!}{i!(n-i)!}$

Et, ils ont les propriétés suivantes :

1.  $\sum_{i=0}^n B_i^n(t) = (t + (1-t))^n = 1$
2.  $B_i^n(t) \geq 0, t \in [0, 1]$
3.  $B_i^n(t) = t \cdot B_{i-1}^{n-1}(t) + (1-t) \cdot B_i^{n-1}(t)$
4.  $B_i^n(t) = B_{n-i}^n(1-t)$
5.  $B_i^n(t)$  atteint son maximum au point  $t = \frac{i}{n}$

**Algorithme de De Casteljau :** Comme le calcul des coefficients binomiaux est numériquement instable, les points de la courbe de Bézier  $P_n(t)$  peuvent être évalués différemment à l'équation (2.25) en utilisant l'algorithme de De Casteljau.

Étant donnés les points de contrôle  $P_0, \dots, P_n$  de la courbe de Bézier et  $\tau \in [0, 1]$ , l'algorithme est comme suit :

$$P_i^r(\tau) = (1-\tau)P_i^{r-1}(\tau) + \tau P_{i+1}^{r-1}(\tau) \quad \begin{cases} r = 1, \dots, n \\ i = 0, \dots, n-r \end{cases}$$

$$P_i^0(\tau) = P_i.$$

Alors,  $P_0^n(\tau)$  est le point de la courbe correspondant au paramètre  $\tau$ .

On illustre dans la figure (2.5) un exemple d'application de l'algorithme de De Casteljau, avec  $n = 3$  et  $\tau = 0.4$  (resp.  $\tau = 0.6$ ). Le point  $P_0^3$  est le point de la courbe qui correspond à  $\tau = 0.4$  ( resp.  $\tau = 0.6$ ).

**Les propriété des courbes de Bézier :** L'algorithme de De Casteljau permet de déduire plusieurs propriétés importantes des courbes de Bézier.

**L'invariance affine :** Les courbes de Bézier sont affinement invariantes, ce qui implique que les deux procédures sont équivalentes :

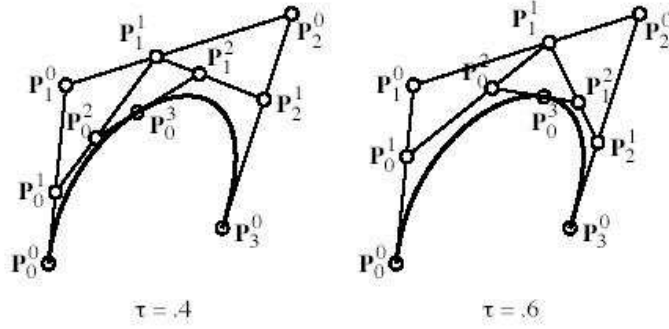


FIGURE 2.5 – Algorithme de De Casteljau

1. calculer le point de la courbe  $P_n(t)$  et puis appliquer la transformation affine.
2. appliquer la transformation affine aux points de contrôle et puis évaluer  $P_n(t)$ .

L'invariance affine est une conséquence directe de l'algorithme de De Casteljau car c'est une répétition de plusieurs interpolations linéaires.

**Propriété d'enveloppe convexe :**  $\forall t \in [0, 1]$ , la courbe  $P_n(t)$  se trouve à l'intérieur de l'enveloppe convexe formée par les points de contrôle  $P_i$ . Ceci résulte du fait que le point intermédiaire  $b_i^r$  de l'algorithme de Casteljau est une combinaison barycentrique convexe des points précédents  $b_j^{r-1}$ .

**Interpolation d'extrémités :** La courbe de Bézier passe par les points  $P_0$  et  $P_n$ . Autrement dit,  $P_n(0) = P_0$  et  $P_n(1) = P_n$ .

**Symétrie.** Puisque les polynômes de Bézier sont symétriques par rapport à  $t$  et  $1 - t$ , alors

$$\sum_{j=0}^n P_j B_j^n(t) = \sum_{j=0}^n P_{n-j} B_j^n(1-t)$$

**Derivabilité :** La dérivée d'une courbe de Bézier est une autre courbe de Bézier, obtenue par application de l'opérateur de différence successive sur le polygone de contrôle original. Elle s'écrit comme suit

$$\frac{d^r}{dt^r} P_n(t) = \frac{n!}{(n-r)!} \sum_{j=0}^{n-r} \Delta^r P_j B_j^{n-r}(t).$$

où

$$\begin{aligned} \Delta^r P_j &= \Delta^{r-1} P_{j+1} - \Delta^{r-1} P_j \\ \Delta P_j &= P_{j+1} - P_j \end{aligned}$$

En particulier, la courbe admet  $P_0P_1$  (resp.  $P_{n-1}P_n$ ) comme tangente au point  $t = 0$  (resp.  $t = 1$ ); de même, la courbure peut être contrôlée par  $P_2$  à l'origine  $P_0$ , et par  $P_{n-2}$  au point finale  $P_n$ .

**Intégrale :** L'intégrale d'une courbe de Bézier est une autre courbe de Bézier.

**Élévation du degré :** Étant donné une courbe de Bézier de degré  $n$ , correspondant à  $n + 1$  points de contrôle  $P_k$  ( $k = 0, 1, \dots, n$ ), les  $n + 2$  points de contrôle  $P_k^*$  ( $k = 0, 1, \dots, n + 1$ ) obtenus par

$$\begin{cases} P_0^* &= P_0, \\ P_k^* &= \frac{k}{n+1}P_{k-1} + \left(1 - \frac{k}{n+1}\right)P_k, \quad 1 \leq k \leq n, \\ P_{n+1}^* &= P_n, \end{cases} \quad (2.27)$$

définit la même courbe géométrique, vue comme une courbe de Bézier de degré  $n + 1$ .

Cette propriété d'élévation de degré, illustrée par la figure (2.6), permet de tirer quelques conclusions :

1. le processus d'élévation de degré garde la courbe de Bézier inchangée.
2. la courbe de Bézier admet une infinité de représentations pour différents degrés. En fait, même pour un degré fixé, la représentation n'est pas unique [108].
3. si le polygone de contrôle initial est convexe, alors tous les polygones de contrôles obtenus par le processus d'élévation de degré seront convexes. Á la limite, le polygone de contrôle converge vers la courbe, qui sera par conséquent convexe.

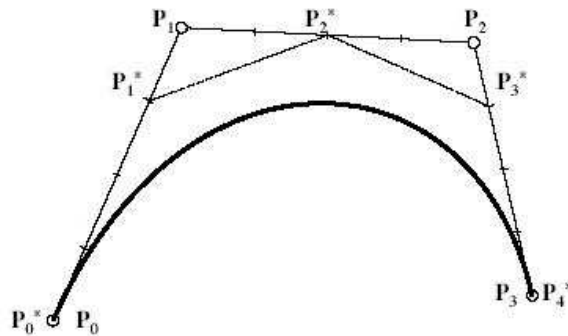


FIGURE 2.6 – Illustration du processus d'élévation de degré : à partir de la courbe de Bézier de degré 3, associée au polygone de contrôle qui relie les points  $P_k$ , le nouveau polygone de contrôle qui relie les points  $P_k^*$  de degré 4, obtenu par combinaison convexe, correspond à la même courbe géométrique

**La surface et le volume de Bézier :** La surface de Bézier est une extension directe de la courbe de Bézier. On définit la surface de Bézier de la façon suivante

$$P(s, t) = \sum_{i=0}^m \sum_{j=0}^n B_i^m(s) B_j^n(t) P_{i,j}. \quad (2.28)$$

avec  $\{P_{i,j}\} (0 \leq i \leq n, 0 \leq j \leq m)$  sont les points de contrôle,  $B_i^m(s)$  (resp.  $B_j^n(s)$ ) est le polynôme de Bernstein de degré  $m$  (resp.  $n$ ), et  $(s, t) \in [0, 1]^2$ .



Les points de contrôle associés à la surface de Bézier sont aussi les points de contrôle correspondant à un ensemble de courbes de Bézier (cf figure ( 2.7)). En effet, si on prend  $t = 0$  dans l'équation ci-dessus, on obtient

$$\begin{aligned} P(s, 0) &= \sum_{i=0}^m \sum_{j=0}^n B_i^m(s) B_j^n(0) P_{i,j} \\ &= \sum_{i=0}^m B_i^m(s) P_{i,0} \end{aligned}$$

Donc, les propriétés des surfaces de Bézier sont analogues à celles des courbes de Bézier.

Le principe se généralise similairement au volume de Bézier qui représente une extension directe de la surface de Bézier. On le définit comme suit

$$P(s, t, u) = \sum_{i=0}^m \sum_{j=0}^n \sum_{k=0}^l B_i^m(s) B_j^n(t) B_k^l(u) P_{i,j,k} \quad ,$$

avec  $\{P_{i,j,k}\} (0 \leq i \leq m, 0 \leq j \leq n, 0 \leq k \leq l)$  sont les points de contrôle,  $B_i^m(s)$  (resp.  $B_j^n(s)$  et  $B_k^l(u)$ ) est le polynôme de Bernstein de degré  $m$  (resp.  $n$  et  $l$ ), et  $(s, t, u) \in [0, 1]^3$ .

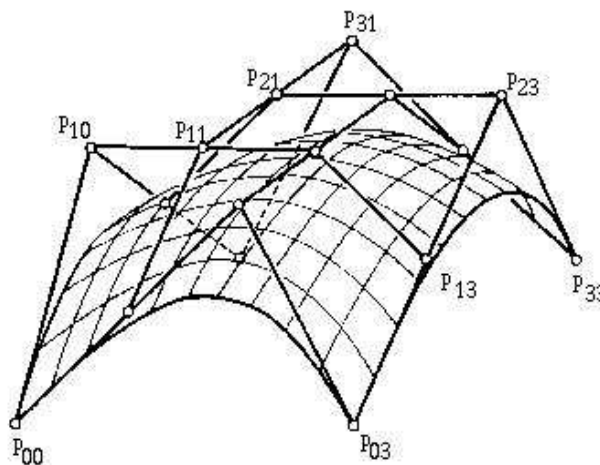


FIGURE 2.7 – Surface de Bézier de degré (3,3) et ses points de contrôle

### 2.3.2.2 Généralisations de la paramétrisation de Bézier

L'un des inconvénients de la paramétrisation de Bézier est qu'elle ne peut représenter que des géométries suffisamment régulières, puisque les paramètres de contrôle agissent d'une manière globale sur la forme. Une manière efficace pour représenter des géométries

complexes consiste à utiliser un ensemble de représentations par patch pour couvrir entièrement la forme. C'est ce qu'on nomme par la représentation B-spline. Par exemple, une forme bidimensionnelle peut être décrite par la courbe B-spline comme suit :

$$P_n(t) = \sum_{i=0}^n N_i^p(t) P_i \quad (2.29)$$

avec  $t \in [t_0, t_{n+p+1}] (t_0 \leq \dots \leq t_{n+p+1})$ ,  $P_i$  sont les points de contrôle, et  $\{N_i^p(t)\} (i = 0, \dots, n)$  sont les fonctions de base B-splines de degré  $p$  définie par :

$$N_i^0(t) = \begin{cases} 1 & \text{si } t_i \leq t \leq t_{i+1} \\ 0 & \text{sinon} \end{cases}$$

$$N_i^p(t) = \frac{t - t_i}{t_{i+n} - t_i} N_i^{p-1}(t) + \frac{t_{i+n+1} - t}{t_{i+n+1} - t_{i+1}} N_{i+1}^{p-1}(t) \quad (2.30)$$

L'extension de cette formulation pour représenter des surfaces ou des volumes se fait de la même façon que la représentation de Bézier.

Un inconvénient de cette représentation réside dans son incapacité à représenter des sections coniques. Cependant, en utilisant une extension des B-splines, qu'on désigne par NURBS (non-uniform rational B-spline), on peut représenter exactement des coniques (ex. cylindres, cônes,...). En 2D, une courbe de NURBS peut être décrit par :

$$P_n(t) = \frac{\sum_{i=0}^n N_i^p(t) W_i P_i}{\sum_{i=0}^n N_i^p(t) W_i} \quad (2.31)$$

où les  $W_i$  sont des fonction de poids.

Les NURBS ont été utilisées dans [69] au départ pour approcher des profils d'ailes existants, et ensuite comme un outil de paramétrisation dans la procédure d'optimisation.

### 2.3.3 Approche Free-form deformation (FFD)

En optimisation de forme aérodynamique, on commence souvent par un profil déjà existant, et le but est d'améliorer ses performances vis-à-vis d'un critère aérodynamique. L'écart entre la forme initiale et la forme optimisée est petit, mais la différence entre leurs performances peut être substantielle. Donc, un meilleur moyen pour réduire le nombre des variables de conception, surtout pour des configurations complexes, consiste à paramétriser la déformation plutôt que la forme elle-même.

La Free-form deformation est l'une des techniques utilisées en informatique graphique pour le morphing d'images et la déformation d'objets. Elle consiste à déformer l'espace contenant la forme, plutôt que la forme elle-même. Au lieu de manipuler l'objet directement, On déforme la boîte englobante contenant l'objet. La FFD à été proposée à l'origine par Sederberg et Parry [101]. Dans leur approche, l'espace à l'intérieur de la boîte est représenté par un produit tensoriel de Bézier, et les points de contrôle sont les coefficients de ces polynômes. L'objet à l'intérieur de la boîte se déforme en bougeant les points de contrôle. Coquillart [24]

a étendue ensuite le concept dans le but d'avoir des boîtes de contrôle de forme arbitraire. Cette technique à été adoptée aussi en optimisation de forme, vu sa flexibilité et son efficacité à représenter des configurations complètes d'avion. On cite, parmi d'autres, le travail de Samareh [97] qui a utilisé les NURBS, qui ont été introduits par Lamousin et al [66], pour paramétrer des surfaces en aérodynamique pour optimiser des formes d'avion complet, et celui de Duvigneau [37] qui a utilisé l'approche FFD avec les B-splines, et à l'INRIA le travail de Andreoli et al [6] dans lequel l'approche FFD a été adaptée à la paramétrisation de Bézier afin d'optimiser une voilure d'avion d'affaire.

Soit  $(\xi, \eta, \zeta)$  un système de coordonnées lié à la boîte englobant la forme, avec  $(\xi, \eta, \zeta) \in [0, 1]^3$ . Pour chaque noeud  $q$  du maillage, on pose :

$$\mathbf{m}_q = \mathbf{m}_q^{init} + \sum_{i=0}^l \sum_{j=0}^m \sum_{k=0}^n B_i^l(s_q) B_j^m(t_q) B_k^n(u_q) \delta \mathbf{p}_{ijk}$$

où  $\mathbf{m}_q$  (resp.  $\mathbf{m}_q^{init}$ ) représente les coordonnées de  $q$  dans le maillage courant (resp. dans la configuration initiale),  $\delta \mathbf{p}_{ijk}$  sont les paramètres de conception qui servent à manipuler la déformation, et  $\mathbf{t}_q = (s_q, t_q, u_q)$  représente les coordonnées FFD de  $q$  dans la boîte englobante. On désigne dans ce cas par  $i \times j \times k$  le degré de la représentation, ce qui correspond à  $i + 1$  (resp.  $j + 1$  et  $k + 1$ ) points de contrôle selon  $\xi$  (resp.  $\eta$  et  $\zeta$ ).

On montre, dans la figure (2.8)(en haut), la boîte englobante initiale et la forme associée. La figure (2.8)(en bas) montre l'effet de la déformation de la boîte englobante sur la forme. Les points rouge de la figure désignent les points de contrôle.

Le succès de la méthode FFD en optimisation de forme aérodynamique est dû à plusieurs raisons :

1. La topologie du maillage reste fixe au cours de l'optimisation, donc le maillage peut être régénéré automatiquement avec une grande robustesse (en évitant les repliements).
2. Puisque la technique proposée est independante de la topologie du maillage, on peut traiter similairement les maillages structurés et non-structurés pour des configurations plus ou moins complexe.
3. Les changements de la géométries sont petits.
4. Le nombre des paramètres est petit en général, ceci dépend du choix de l'utilisateur.

## 2.4 Outils de déformation de maillage

La précision de la solution, dans le contexte de l'optimisation de forme, dépend de la qualité du maillage et aussi des propriétés du flux. Ces propriétés sont évaluées par un code CFD en s'appuyant sur un maillage. Comme la géométrie se déforme au cours du processus d'optimisation, en faisant appel à une technique de paramétrisation, le maillage doit être ajusté également. Il s'agit donc de développer des algorithmes robustes de remaillage qui prennent en compte la déformation de la forme et peu coûteux.

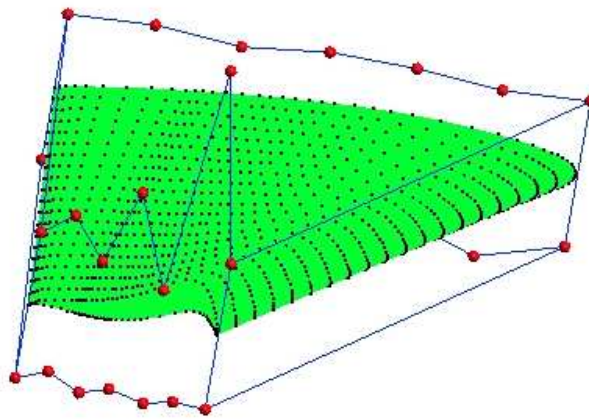
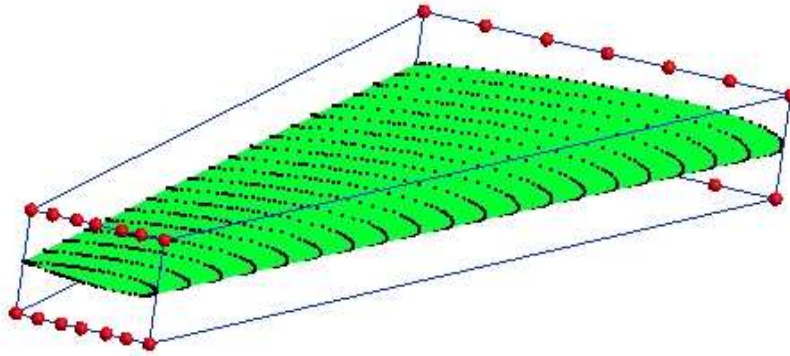


FIGURE 2.8 – Exemple d’une boîte englobant la forme dans le cas d’une représentation de Bézier de degré 6-1-1 : avant déformation (en haut), après déformation (en bas)

On cite brièvement quelques techniques de déformation du maillage utilisée en optimisation de forme [83], et avec plus de détails les méthodes adoptées durant cette thèse.

- **Déformation explicite** : La déformation est décrite en utilisant les noeuds du maillage [76]. La déformation pour chaque noeud est proportionnelle à sa distance par rapport à la forme. Pour chaque noeud intérieur  $i$ , on écrit :

$$\begin{cases} (\delta x_m)_i &= \frac{1}{\alpha_i} \sum_{k \in \Gamma_w} w_k \alpha_{ki} \delta \tilde{x}_i \\ \delta \tilde{x}_m &= \delta \tilde{x} \quad \text{sur} \quad \Gamma_w \end{cases}$$

où  $\delta x_m$  est la variation cherchée des noeuds du maillage,  $w_k$  est le poids de contribution

pour chaque noeud  $k$  de la forme qui dépend de la longueur des segments voisins,  $\alpha_{ki} = \frac{1}{|\bar{x}_k - \bar{x}_i|^\beta}$  avec  $\beta$  un paramètre positif arbitraire,  $\alpha_i = \sum_{k \in \Gamma_w} w_k \alpha_{ki}$  est le paramètre de normalisation et  $\Gamma_w$  est la forme après le déplacement  $\delta \bar{x}_m$ .

Cet algorithme est robuste mais très coûteux puisque sa complexité est proportionnelle au nombre des noeuds du maillage.

- **Lissage elliptique** : on résout le système suivant :

$$\begin{cases} (I - \eta \Delta) \delta x_m = \bar{\delta \bar{x}} \\ \bar{\delta \bar{x}} = \delta \bar{x} \quad \text{sur} \quad \Gamma_w \\ \bar{\delta \bar{x}} = 0 \quad \text{dans} \quad D \end{cases}$$

où  $\delta x_m$  est la variation du maillage,  $\eta$  est le coefficient de viscosité dans le processus de déformation du maillage,  $\Gamma_w$  est la forme après le déplacement  $\delta \bar{x}_m$ , et  $D$  est l'ensemble des noeuds intérieurs au maillage.

Cet algorithme n'est pas coûteux mais n'est pas vraiment robuste, il ne conserve pas la conformité des éléments pour les maillages fins.

- **Déformation par analogie avec les ressorts** : Cette méthode, qui est très proche de la précédente, est un autre moyen pour déformer le maillage autour de la frontière [40][41]. Chaque segment du maillage étant assimilé à un ressort de compression. Le déplacement des noeuds au cours de la déformation est la solution d'un système linéaire :

$$\begin{cases} Kq = 0 \\ q = \bar{q} \quad \text{sur} \quad \Gamma_m \end{cases}$$

Avec  $K$  est la matrice de rigidité,  $q$  est le vecteur de déplacement, et  $\bar{q}$  le vecteur déplacement des noeuds aux limites du domaine  $\Gamma_m$ .

## 2.4.1 Conditions de transpiration

En optimisation de forme aérodynamique, on initialise généralement l'algorithme d'optimisation par une forme déjà existante, et le but consiste à améliorer sa performance. La perturbation de la géométrie entre la forme initiale et la forme optimisée est petite. Il s'agit d'utiliser, dans ces circonstances, les conditions de transpiration qui modifient les conditions aux limites, ce qui évite de modifier le maillage.

Le concept de la transpiration suppose que l'aire des éléments reste constant, puisqu'il s'agit des petites perturbations de forme. Après déformation de la surface, la normale change d'orientation, et par conséquent on obtient un nouveau vecteur vitesse aux noeuds de la surface déformée (voir figure (2.9)).

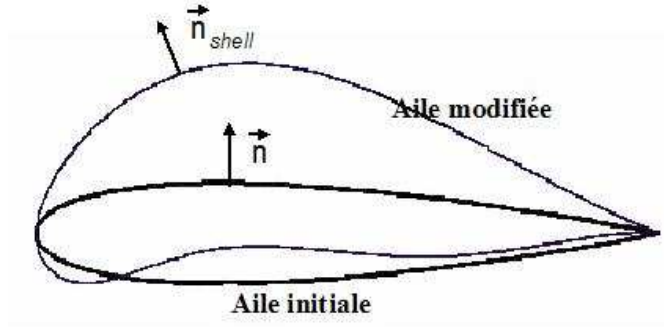


FIGURE 2.9 – Illustration du concept de la transpiration

Soit  $\vec{n}_{shell}$  (resp.  $\vec{n}$ ) la normale unitaire à la surface déformée (resp. la surface initiale). Alors, le flux de bord  $\Psi(W)_{\text{bord glissant}}$  est défini par :

$$\Psi(W)_{\text{bord glissant}} = q \cdot W + \begin{pmatrix} 0 \\ P \cdot n_x \\ P \cdot n_y \\ P \cdot n_z \\ P \cdot q \end{pmatrix} \quad (2.32)$$

avec

$$q = \vec{V} \cdot (\vec{n} - \vec{n}_{shell})$$

où  $\vec{V}$  est la vitesse du fluide,  $P$  est la pression, et

$$W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{pmatrix}$$

Il se produit donc un transfert de matière, de quantité de mouvement et d'énergie de la surface initiale à la surface transpirée.

On note que la condition de soufflage à la paroi (2.32) permet de prendre en compte la condition de glissement,

$$\vec{V} \cdot \vec{n}_{shell} = 0$$

La transpiration offre l'avantage d'être rapide et relativement simple à implémenter. L'idée est simple, il s'agit de résoudre l'écoulement uniquement sur le maillage initial, et une légère modification des conditions aux limites permet de traduire la déformation par un faible déplacement suivant la normale unitaire à la surface initiale. Ce concept a été utilisé avec succès par plusieurs auteurs dans le contexte de l'optimisation de forme [56][75]. En plus, Il s'est avéré aussi efficace pour des perturbations de la surface plutôt importantes dans [46][93].

## 2.4.2 Free-form deformation

La méthode Free-form deformation, qu'on a expliqué dans la section précédente en tant que technique de paramétrisation, permet de déformer la forme et le maillage simultanément. Cette démarche a pour avantage de conserver la qualité du maillage en évitant un remaillage global qui peut être coûteux.

On définit initialement un volume de contrôle qui englobe la forme initiale. La modification de la boîte englobante au cours du processus d'optimisation agit sur la forme emboîtée. Le maillage doit par conséquent suivre les déformations appliquées à la forme.

Cette procédure de remaillage peut produire certains problèmes de continuité au bord de la boîte englobante. D'où l'idée de partitionner la boîte en trois coquilles (voir figure (2.4.2)). Celle du milieu correspond à la zone déformée qui englobe la forme initiale. Et autour se trouve la zone influencée, qui est affectée par le mouvement des points de contrôle de la zone déformée. Au bords, la zone virtuelle, où tous les points de contrôles sont fixés, évite le problème des conditions aux limites (comme l'a souligné Bartels et al [10]).

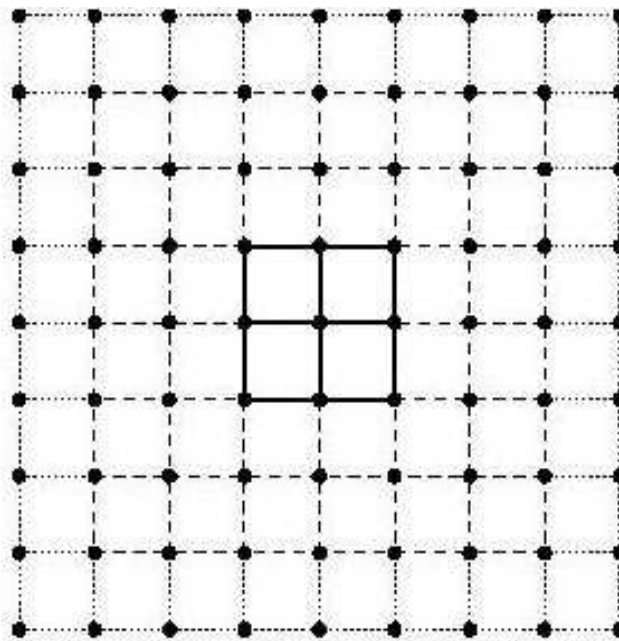


FIGURE 2.10 – La boîte FFD montrant trois zones : déformée, influencée et virtuelle

## 2.5 Méthodes d'optimisation

### 2.5.1 Généralités

Un problème d'optimisation de forme en aérodynamique se formule généralement de la façon suivante :

$$\begin{cases} \min \mathcal{J}(p, W(p), X(p)) \\ R(p, W(p), X(p)) = 0 \\ c_1(p, W(p), X(p)) = 0 \\ c_2(p, W(p), X(p)) < 0 \end{cases} \quad (2.33)$$

où  $J$  est la fonctionnelle coût à minimiser,  $c_1$  et  $c_2$  sont les contraintes du problème, qui dépendent des paramètres de conception  $p$ , des variables physiques  $W$  et des éléments géométriques  $X$ , et  $R$  représente l'équation d'état régissant l'écoulement.

On a présenté dans les sections précédentes quelques éléments de la méthodologie numérique : l'équation d'état qui régit l'écoulement, les techniques de mise à jour du maillage, et les différentes méthodes de paramétrisation. On s'intéresse maintenant à la résolution numérique du problème d'optimisation de forme (2.33). Différentes méthodes sont possibles pour minimiser la fonction objectif  $\mathcal{J}$ , qu'on peut subdiviser généralement en deux catégories. La première classe des méthodes d'optimisation, dite de type gradient, se base sur la valeur de la fonction objectif et de ses dérivées. Le schéma itératif s'écrit, à l'itération  $k$ , comme suit :

$$p^{k+1} = p^k + \rho^k d^k \quad (2.34)$$

où  $p$  est le vecteur de conception,  $\rho^k > 0$  et  $d^k$  sont respectivement le pas et la direction de descente à l'itération  $k$ .

La plus simple des méthodes d'optimisation est la méthode de descente à pas fixe ( $\rho^k = \rho^0$ ) qui utilise seulement la dérivée d'ordre 1 :

$$d^k = -\nabla \mathcal{J}(p^k, W(p^k), X(p^k)) \quad (2.35)$$

Cette méthode converge lentement puisque le pas de descente n'évolue pas au cours des itérations. La méthode dite de plus forte descente propose une alternative pour calculer le pas de descente optimal, associé à la direction de descente choisie, en utilisant une méthode de recherche mono-dimensionnelle.

Pour accélérer le processus itératif, il s'agit d'utiliser les dérivées d'ordre 2. C'est ce que propose la méthode de Newton qui utilise une direction de descente comme suit :

$$d^k = -\mathcal{H}^{-1}(p^k, W(p^k), X(p^k)) \nabla \mathcal{J}(p^k, W(p^k), X(p^k)) \quad (2.36)$$

où  $\mathcal{H}$  est le hessien de  $\mathcal{J}$ .

L'inconvénient de cette méthode est la difficulté et le coût important du calcul de  $\mathcal{H}^{-1}$  surtout pour des applications complexes. D'où l'idée d'utiliser une approximation de l'inverse du hessien au cours des itérations. Les méthodes de type quasi-Newton adoptent cette stratégie, qui consiste à prendre :

$$d^k = -\mathcal{H}^k \nabla \mathcal{J}(p^k, W(p^k), X(p^k)) \quad (2.37)$$



où  $\mathcal{H}^k$  est une approximation de l'inverse du hessien.

Les formules les plus connues de  $\mathcal{H}^k$  sont dû à Davidon-Fletcher-Powell (DFP) :

$$\mathcal{H}^k = \mathcal{H}^{k-1} + \frac{\delta\delta^T}{\delta^T\gamma} - \frac{\mathcal{H}^{k-1}\gamma\gamma^T\mathcal{H}^{k-1}}{\gamma^T\mathcal{H}^{k-1}\gamma} \quad (2.38)$$

et à Broyden-Fletcher-Goldfarb-Shanno (BFGS) :

$$\mathcal{H}^k = \mathcal{H}^{k-1} + \left(1 + \frac{\gamma^T\mathcal{H}^{k-1}\gamma}{\delta^T\gamma}\right) \frac{\delta\delta^T}{\delta^T\gamma} - \frac{\delta\gamma^T\mathcal{H}^{k-1} + \mathcal{H}^{k-1}\gamma\delta^T}{\delta^T\gamma} \quad (2.39)$$

où  $\delta := p^k - p^{k-1}$  et  $\gamma = \mathcal{J}(p^k, W(p^k), X(p^k)) - \mathcal{J}(p^{k-1}, W(p^{k-1}), X(p^{k-1}))$

Ces méthodes d'optimisation nécessitent le gradient de la fonction objectif dont le calcul, particulièrement en aérodynamique, est très délicat. En effet, on utilise généralement trois techniques pour le calcul du gradient : les différences finies, surtout lorsqu'on utilise des codes d'optimisation comme *boîte noire*, demandent un nombre d'évaluations qui dépend linéairement de la dimension de l'espace de recherche ; la résolution d'une approximation discrète d'un problème adjoint comme dans [58] est coûteux et difficile à mettre en oeuvre surtout pour des problèmes complexes ; et la différentiation automatique comme dans [25][53] demande un aménagement spécifique dans les codes de simulation pour qu'elle soit appliquée, ce qui n'est facile à réaliser dans des cas complexes. On trouve plus de détails sur ces méthodes d'optimisation dans les références suivantes [80][47].

La seconde catégorie comporte les méthodes d'optimisation sans gradient qui utilisent seulement les valeurs de la fonction objectif sans faire appel à ses dérivées. La méthode classique du simplexe, et les méthodes dites stochastiques comme les algorithmes génétiques, les algorithmes évolutionnaires, les algorithmes PSO (Particle Swarm Optimization), sont souvent préférés et se caractérisent par une grande robustesse dans des situations non-linéaires complexes. On a choisit d'utiliser dans nos applications la méthode du simplexe de Nelder-Mead, présentée dans la section suivante, qui permet de réaliser, pour un problème modérément complexe, un compromis entre la simplicité, la robustesse et la rapidité de convergence.

## 2.5.2 Méthodes de type simplexe

La méthode du simplexe est une méthode d'optimisation déterministe qui utilisent directement la valeur de la fonction objectif, sans faire appel à ses dérivées. C'est une méthode robuste, puisqu'elle est capable de minimiser des fonctions qui présentent un bruitage, simple à programmer, et rapide pour des problèmes de dimensions petites. La méthode du simplexe utilise une notion géométrique, qui se nomme simplexe, pour guider la recherche vers l'optimum. Un simplexe de  $\mathbb{R}^n$  est un polyèdre de  $n + 1$  sommets. Par exemple, un simplexe en 2D est un triangle, et en 3D est un tétraèdre. L'algorithme utilise seulement les simplexes dits non-dégénérés où les arêtes adjacentes à chaque sommet forment une base de  $\mathbb{R}^n$ .

Le processus d'optimisation démarre, contrairement aux méthodes d'optimisation classiques, par  $n + 1$  points qui définissent un simplexe initial non-dégénéré dont les sommets

$x_i (i = 0, \dots, n)$  sont définis comme suit :

$$x_i = x_0 + \lambda e_i, \quad i = 1, \dots, n \quad (2.40)$$

avec  $e_i (i = 1, \dots, n)$  sont des vecteurs unitaires linéairement indépendants,  $\lambda$  est une constante à adapter selon la nature du problème, et  $x_0$  est un point initial.

La stratégie d'optimisation se base sur la comparaison entre les évaluations de la fonction objectif aux sommets du simplexe. Si on désigne par  $f$  la fonction à minimiser, alors à chaque étape de l'algorithme, on ordonne les points  $x_i (i = 0, \dots, n)$  de telle sorte que  $f(x_0) \leq f(x_1) \leq \dots \leq f(x_n)$ . Le meilleur sommet dans ce cas est  $x_n$  et le plus mauvais est  $x_0$ . L'algorithme fait évoluer ensuite la recherche de l'optimum en utilisant des opérations algébriques qui se traduisent par des transformations géométriques élémentaires. On distingue par la suite plusieurs stratégies de recherche.

### 2.5.2.1 Méthode de Spendley, Hext et Himsworth

La première méthode de type simplexe à été proposée initialement par Spendley, Hext et Himsworth en 1962 [105], et qui se base seulement sur le mouvement de réflexion qui consiste à calculer le symétrique du plus mauvais sommet du simplexe par rapport au barycentre (voir figure (2.11)). Cet algorithme évalue initialement la fonction  $f$  aux sommets du simplexe initial afin de définir le plus mauvais point, qui correspond à la valeur de la fonction la plus haute, on fait appel ensuite à l'opérateur de réflexion. Si le sommet réfléchi est encore le plus mauvais, on choisit le deuxième plus mauvais et ainsi de suite. Au voisinage de l'optimum, la stratégie de recherche aboutit à un cycle limite suite au seul mouvement de réflexion (voir figure (2.12)). On propose donc une autre alternative pour éviter ce genre de situation.

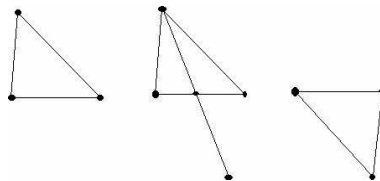


FIGURE 2.11 – Illustration de l'opérateur de réflexion

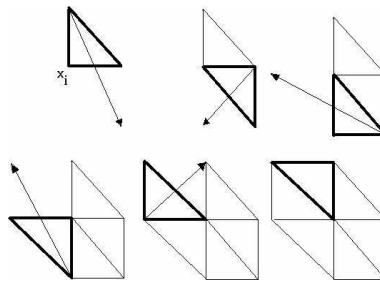


FIGURE 2.12 – Comportement au voisinage de l'optimum

### 2.5.2.2 Méthode de Nelder-Mead

On propose ici une extension de la méthode précédente qui est due à Nelder et Mead [85]. Cette méthode utilise deux mouvements supplémentaires, l'expansion et la contraction, qui sont destinés à accélérer la stratégie de recherche et assurer la convergence.

À l'itération  $k$ , soit  $x_j^k$  le plus mauvais sommet du simplexe. Le processus d'optimisation effectue initialement un mouvement de réflexion (voir figure (2.13)). Le sommet réfléchi  $x_r^k$  est défini par :

$$x_r^k = (1 + \alpha)\bar{x}^k - \alpha x_j^k, \quad (2.41)$$

où  $\alpha$  est une constante positive, généralement égale à 1, appelée *constante de réflexion*, et  $\bar{x}^k$  est le centre de gravité des points différents de  $x_j^k$ .

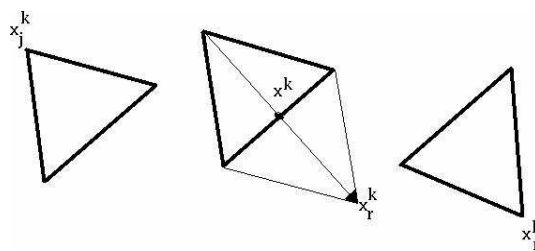


FIGURE 2.13 – Réflexion

Si  $x_j^k$  est le meilleur sommet du simplexe, le simplexe subit ensuite une expansion (voir figure (2.14)). Il en résulte un autre sommet  $x_e^k$  défini par :

$$x_e^k = \gamma x_r^k + (1 - \gamma)\bar{x}^k, \quad (2.42)$$

où  $\gamma$  est la *constante d'expansion*, dont la valeur est supérieure à 1, et qui désigne le rapport de la distance  $[x_e^k, \bar{x}^k]$  à la distance  $[x_r^k, \bar{x}^k]$ .

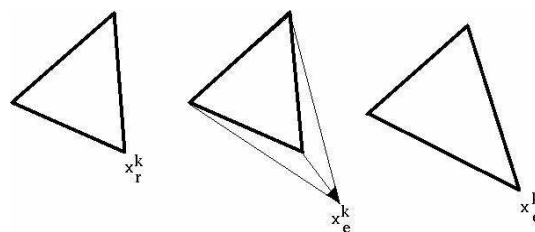


FIGURE 2.14 – Expansion

Si le point obtenu  $x_e^k$  est meilleur que le sommet réfléchi  $x_r^k$ , on le garde pour l'étape suivante, sinon on garde seulement le sommet  $x_r^k$ .

Si le sommet réfléchi  $x_r^k$  est le plus mauvais, on effectue une contraction avec le meilleur point parmi  $x_r^k$  et  $x_j^k$  pour obtenir un nouveau point  $x_c^k$ . Si  $x_j^k$  est meilleur que  $x_r^k$ , i.e.  $f(x_r^k) \leq$

$f(x_j^k)$ , on effectue une contraction dite *interne* comme suit (voir figure (2.15)) :

$$x_c^k = \beta x_r^k + (1 - \beta) \bar{x}^k \quad (2.43)$$

sinon, on effectue une contraction dite *externe* comme suit (voir figure (2.16)) :

$$x_c^k = \beta x_j^k + (1 - \beta) \bar{x}^k \quad (2.44)$$

où  $\beta \in [0, 1]$  est le coefficient de contraction qui désigne le rapport de la distance  $[x_c^k, \bar{x}^k]$  à la distance  $[x_j^k, \bar{x}^k]$ .

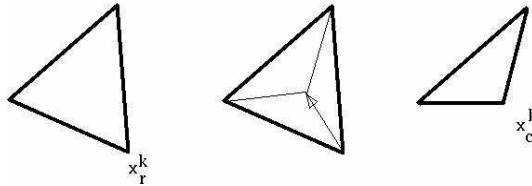


FIGURE 2.15 – Contraction, si  $f(x_r^k) < f(x_j^k)$

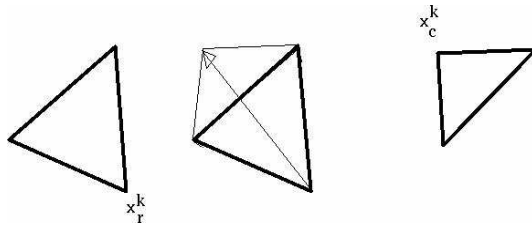


FIGURE 2.16 – Contraction, si  $f(x_r^k) > f(x_j^k)$

Soit  $x_l^k$  tel que :

$$\begin{cases} x_l^k = x_r^k, & \text{si } f(x_r^k) < f(x_j^k) \\ x_l^k = x_j^k, & \text{si } f(x_j^k) < f(x_r^k) \end{cases} \quad (2.45)$$

Si le point  $x_c^k$  est meilleur que  $x_l^k$ , i.e.  $f(x_c^k) \leq f(x_l^k)$ , on remplace  $x_l^k$  par  $x_c^k$  dans la prochaine itération. Sinon, le simplexe subit une réduction (voir figures (2.17) et (2.18)), ce qui revient à remplacer tous les points par

$$x_i^{k+1} = \frac{x_i^k + x_m^k}{2}, \quad (2.46)$$

où  $x_m^k$  est le meilleur sommet.

L'algorithme du simplexe de Nelder-Mead a été utilisé avec succès dans plusieurs domaines. Cependant, aucune preuve de convergence n'a été publiée jusqu'à présent pour les dimensions supérieures. La première analyse a été fournie en [65], qui montre que l'algorithme converge dans  $\mathbb{R}^1$  vers un point stationnaire en considérant des hypothèses classiques. Par contre pour les dimensions supérieures, plusieurs exemples ont été présentés dans [79] qui prouvent que l'algorithme peut converger dans  $\mathbb{R}^2$  vers un point non-stationnaire.

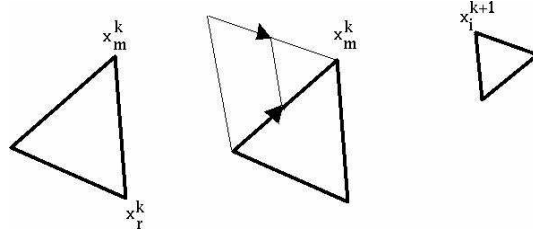


FIGURE 2.17 – Réduction, si  $f(x_r^k) > f(x_j^k)$

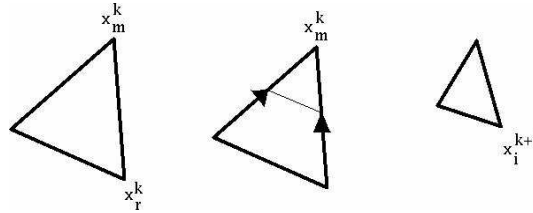


FIGURE 2.18 – Réduction : si  $f(x_r^k) < f(x_j^k)$

### 2.5.2.3 Méthode MSA (Multi-directional Search Algorithm)

La méthode MSA, qui a été développée en 1991 par Torczon [111][27], est une méthode de type simplexe destinée à une utilisation sur architecture parallèle. La méthode MSA est, en quelque sorte, une version parallèle de la méthode de Nelder-Mead. Elle utilise les mouvements de réflexion, de contraction et d'expansion, autour du meilleur sommet du simplexe, pour guider la recherche de l'optimum. Contrairement à la méthode précédente, la méthode MSA possède une preuve de convergence avec des hypothèses classiques.

À l'itération  $k$ , soit  $x_0^k$  le meilleur sommet du simplexe. On fait subir un mouvement de réflexion à l'ensemble des sommets par rapport à  $x_0^k$  comme suit :

$$\widehat{x}_i^k = (1 + \alpha)x_0^k - \alpha x_i^k, \quad i = 0, \dots, n \quad (2.47)$$

où  $\alpha$  est une constante positive, généralement égale à 1, appelée *constante de réflexion*. On évalue la fonction objective en  $\widehat{x}_i^k, (i = 1, \dots, n)$ . Ces évaluations peuvent être effectuées simultanément en faisant appel à  $n$  processeurs en parallèle. Donc, si l'un des sommets obtenu est meilleur que  $x_0^k$ , on poursuit la recherche en faisant un mouvement d'expansion :

$$x_i^{k+1} = \gamma \widehat{x}_i^k + (1 - \gamma)x_0^k, \quad i = 0, \dots, n \quad (2.48)$$

où  $\gamma$  est la *constante d'expansion*, dont la valeur est supérieure à 1. Sinon, on fait subir à tous les sommets du simplexe un mouvement de contraction pour se rapprocher du meilleur sommet  $x_0^k$  :

$$x_i^{k+1} = \beta \widehat{x}_i^k + (1 - \beta)x_0^k, \quad i = 0, \dots, n \quad (2.49)$$

où  $\beta \in [0, 1]$  est le *coefficient de contraction*. Et, on débute ensuite une nouvelle itération avec un autre simplexe  $x_i^{k+1}, (i = 0, \dots, n)$  obtenu soit par expansion ou par contraction. On évalue

la fonction objectif en  $x_i^{k+1}, (i = 0, \dots, n)$  d'une façon simultanée pour déterminer le meilleur sommet, et on réitère le procédé.

## 2.6 Conclusion

On a présenté dans ce chapitre quelques ingrédients de notre méthodologie numérique pour traiter un problème d'optimisation de forme. Ces outils vont nous servir pour optimiser une géométrie initiale (par exemple une aile d'avion) en minimisant une fonctionnelle coût vis-à-vis des contraintes aérodynamiques et géométriques, comme par exemple la réduction de la traînée aérodynamique avec une contrainte sur la portance. Ce processus d'optimisation est généralement coûteux, puisque chaque évaluation de la fonction objectif passe par la résolution d'une équation d'état (par exemple les équations d'Euler). Pour remédier à cela, le prochain chapitre propose une méthodologie numérique basée sur une hiérarchie de paramétrisations de la forme, et qui s'inspire des propriétés intéressantes des méthodes multigrilles qui peuvent atteindre une complexité proportionnelle au nombre de degrés de liberté.



## Chapitre 3

# Algorithmes hiérarchiques en optimisation de Forme

### 3.1 Introduction

L'objectif majeur de notre travail est de développer des algorithmes numériques en optimisation de forme pour des applications, dans lesquelles chaque évaluation de la fonction coût est très coûteuse, puisqu'elle dépend de l'évaluation de coefficients aérodynamiques provenant de la résolution d'un système d'EDP (équations d'Euler).

Durant les dernières années, plusieurs études ont démontré le grand potentiel des méthodes multigrilles qui permettent d'atteindre une complexité linéaire avec la méthode multigrille complète (FMG). En s'inspirant de ce concept intéressant, nous avons étendu le concept multiniveau en optimisation de forme en s'appuyant sur une hiérarchie de paramétrisations.

Nous rappelons brièvement dans ce chapitre les méthodes hiérarchiques en résolution pure, et nous présentons ensuite notre approche hiérarchique en optimisation de forme.

### 3.2 Algorithmes multigrilles

Les méthodes multigrilles ont été développées depuis les années 60. Le premier schéma multigrille a été développé par Fedorenko [43][44] pour résoudre l'équation de Poisson dans un carré unité. Plusieurs auteurs ont étendu par la suite l'idée de Fedorenko à des problèmes aux limites elliptique (voir par exemple [43]). Cependant, la grande efficacité de l'approche multigrille a été réalisée après les travaux de Brandt [16][17] et Hackbusch[50] qui ont introduit les méthodes multigrilles (MG) pour des problèmes non linéaires comme la méthode FAS (Full Approximation Scheme) [17][50]. Un nouveau développement dans la formulation des méthodes multigrilles est la méthode multigrille complète (Full MultiGrid) [17][50] qui se base sur le couplage des méthodes MG ou FAS à une technique d'enrichissement de maillage.

Le principe de base des algorithmes repose sur l'analyse modale des systèmes linéaires issus de la discrétisation d'équations aux dérivées partielles ; l'idée de base est la suivante : un signal peut se décomposer en série de Fourier. Chaque mode de Fourier est associé à une fréquence. Un signal basse fréquence sur la grille fine devient un signal haute fréquence



sur une grille grossière. De plus, les méthodes itératives classiques, telles que l'itération de Jacobi, sont de bons lisseurs haute fréquence, c'est à dire elles permettent d'atténuer les composantes haute fréquence de l'erreur itérative. Par contre, les composantes basse fréquence, qui sont l'origine de la raideur du système, sont bien représentées sur une grille moins fine et sont efficacement traitées par une étape dite de "correction grille-grossière".

Les méthodes multigrilles se basent sur une hiérarchie de maillages correspondant aux différents domaines de fréquences et des transferts d'informations de grille à grille. Ces algorithmes permettent d'atteindre dans le cas idéal une complexité optimale proportionnelle au nombre de degrés de liberté. Dans cette section, on présente un aperçu sur les méthodes multigrilles en s'inspirant de [28][118].

### 3.2.1 Modèle à résoudre

On se restreint dans cette brève présentation à la résolution d'un système linéaire :

$$A_h u_h = f_h \quad (3.1)$$

avec,

$u_h$  est le vecteur des  $N$  inconnues  $u_k$  ( $k = 1, \dots, N$ ),  $A_h$  est la matrice d'approximation de dimension  $N \times N$ , et  $f_h$  est le vecteur associé au terme source.

Ce système linéaire représente la discrétisation en espace du modèle continu  $Au = f$ , issu d'une EDP elliptique, sur un maillage (ou une grille)  $M_h$ ; l'indice  $h$  caractérise la finesse du maillage.

Pour définir et analyser les méthodes multigrilles pour des problèmes linéaires, on introduit un problème modèle simple. On considère le problème unidimensionnel avec conditions aux limites de Dirichlet :

$$\begin{cases} -u_{xx} = f, & \text{dans } [0, 1] \\ u(x) = 0, & x \in \{0, 1\} \end{cases} \quad (3.2)$$

La discrétisation de ce modèle par différences finies sur un maillage uniforme de finesse  $h$  conduit au système discret dans lequel la matrice d'approximation (de dimension  $N \times N$ ) à la structure suivante :

$$A_h = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} \quad (3.3)$$

Les valeurs propres de  $h^2 A_h$  sont données par

$$\lambda_k^h = 2 - 2 \cos \theta_k, \quad k = 1, \dots, N \quad (3.4)$$

et les vecteurs propres correspondants sont

$$v_k^h = \left( \sqrt{2h} \sin(j\theta_k) \right)_{j=1}^N, \quad k = 1, \dots, N \quad (3.5)$$

avec

$$\theta_k = \frac{k\pi}{N+1} = k\pi h \quad (3.6)$$

On peut résoudre le système discret par l'itération de Jacobi définie comme suit :

$$u_h^{n+1} = G_J u_h^n + D_h^{-1} f_h \quad (3.7)$$

où  $G_J = I - D_h^{-1} A_h$  avec  $D_h$  est la matrice diagonale de  $A_h$ .

On utilise un résultat bien connu sur le critère de convergence : La méthode (3.7) converge si et seulement si  $\rho(G_J) < 1$  ( $\rho(G_J)$  est le rayon spectral de  $G_J$ ).

Or, les valeurs propres de  $G_J$  sont données par :

$$\mu_k = \cos \theta_k, \quad k = 1, 2, \dots, N \quad (3.8)$$

Alors, le rayon spectral s'écrit :

$$\rho(G_J) = \cos \frac{\pi}{N+1} = \cos \pi h = 1 - \frac{\pi^2}{2} h^2 + \dots \quad (3.9)$$

On a  $\rho(G_J) < 1$ , ce qui garantit la convergence. En particulier, on déduit que la convergence est d'autant plus lente ( $\rho(G_J)$  proche de 1) que le maillage devient de plus en plus fin ( $h$  tend vers 0). Ceci explique la raideur numérique du système. On note aussi que les valeurs propres de la matrice d'amplification (ou plutôt d'atténuation)  $G_J$  mesurent le taux d'atténuation des composantes fréquentielles durant l'itération. Une observation basique est que  $|\mu_k|$  croît quand  $k$  décroît. Ceci signifie que, les composantes basses fréquences ( $k$  proche de 1) s'atténuent lentement ( $|\mu_k|$  proche de 1), à l'inverse, les composantes hautes fréquences s'atténuent rapidement. Donc, le principe basique des méthodes multigrilles consiste à approcher les composantes BF de l'erreur au niveau grossier. Les composantes HF de l'erreur sont atténuées par un nombre petit (independent de  $h$ ) d'itérations avec une méthode itérative au niveau fin.

### 3.2.2 Opérateurs de transfert

On considère deux types de discrétisation : une discrétisation fine avec  $M_h$  degrés de liberté, notée  $\mathcal{M}_h$ , et une discrétisation grossière avec  $M_{2h}$  degrés de liberté, notée  $\mathcal{M}_{2h}$ , qui est généralement emboîtée avec le maillage fin (i.e  $\mathcal{M}_{2h} \subset \mathcal{M}_h$ ). Les méthodes multigrilles utilisent les opérateurs de restriction et de prolongement qui permettent le transfert d'informations d'un niveau à un autre. On note par la suite l'opérateur de prolongement par :  $P = \mathcal{I}_{2h}^h : \mathcal{M}_{2h} \rightarrow \mathcal{M}_h$ , et l'opérateur de restriction par :  $R = \mathcal{I}_h^{2h} : \mathcal{M}_h \rightarrow \mathcal{M}_{2h}$ . On schématise dans la figure (3.1) ces deux opérateurs.

### 3.2.3 Enrichissement progressif de maillage

Cette méthode, connue dans la littérature sous le nom de "Nested Iteration" [17], est la plus ancienne et aussi la plus intuitive des méthodes qui utilisent une hiérarchie de maillages. Le processus consiste à résoudre d'abord le modèle discret sur une grille grossière, en faisant appel à une méthode itérative, on prolonge ensuite le résultat sur une grille plus fine, on itère

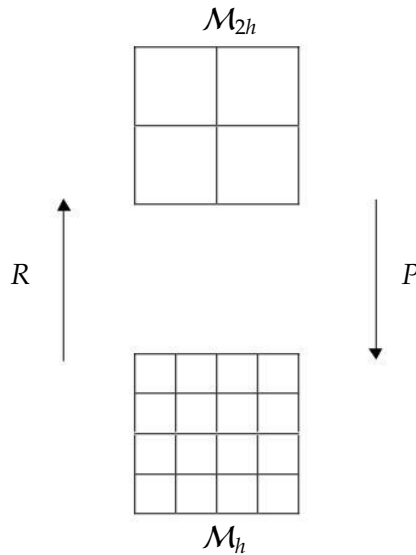


FIGURE 3.1 – Opérateurs de transfert,  $P$  : opérateur de prolongement,  $R$  : opérateur de restriction

à nouveau, puis on interpole le résultat, par prolongement, sur une grille encore plus fine, et ainsi de suite.

Comme démontré par exemple dans [28], l'algorithme d'enrichissement progressif de maillage permet de réduire le coût de calcul d'un facteur de  $\log N$ , où  $N$  est le nombre de degrés de liberté ; ce qui correspond au nombre de niveaux considérés. Les mérites de cet méthode sont dûs à trois raisons principales :

1. L'utilisation d'un maillage grossier au début du processus permet d'accélérer la convergence, car avec un nombre faible de paramètres le problème est moins raide.
2. Pour atténuer, d'une façon efficace, les composantes hautes fréquences de l'erreur, on exige un bon lisseur (une méthode itérative) et des critères d'arrêt bien réglés.
3. Une bonne initialisation par prolongement du résultat par passage d'un niveau à un autre.

On schématise ce processus dans la figure suivante dans le cas de 3 niveaux :

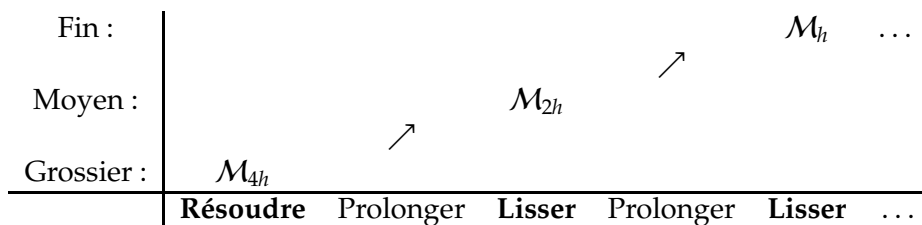


FIGURE 3.2 – Schéma de l'algorithme d'enrichissement progressif de maillage

### 3.2.4 Méthode bigrille idéale

Nous décrivons ici la partie basique des méthodes multigrilles. Le processus démarre, contrairement à l'intuition, par un lissage sur la grille fine afin d'atténuer les composantes hautes fréquences de l'erreur, seules les composantes basses fréquences persistent et sont la manifestation de la raideur du système. Or, ces modes sont correctement représentés sur une grille grossière. On résout alors exactement (à convergence complète) le problème de correction sur la grille grossière. On prolonge ensuite le résultat sur la grille fine et on procède à un lissage pour éliminer à nouveau les modes hautes fréquences de l'erreur qui sont réapparues lors du prolongement du résultat de correction grille grossière.

On se limite ci-dessous à la description d'un algorithme bigrille idéal linéaire :

1. **Lissage sur la grille fine** : Soit  $u_h^0$  l'approximation initiale sur la grille fine, on a :

$$u_h^{k+\frac{1}{2}} = G_h^{\alpha_1} u_h^0 + b_h \quad (3.10)$$

où  $G_h^{\alpha_1}$  est la matrice d'amplification (ou plutôt d'atténuation),  $b_h$  un vecteur fonction du second membre  $f_h$  indépendant de  $u_h$ , et  $\alpha_1$  est le nombre d'itérations de l'algorithme de lissage.

A l'issue de cette phase, on évalue le résidu :

$$r_h^{k+\frac{1}{2}} = A_h u_h^{k+\frac{1}{2}} - f_h \quad (3.11)$$

L'erreur s'exprime par :

$$e_h^{k+\frac{1}{2}} = u_h^{k+\frac{1}{2}} - u_h \quad (3.12)$$

On déduit alors les deux équations suivantes :

$$A_h e_h^{k+\frac{1}{2}} = r_h^{k+\frac{1}{2}} \quad (3.13)$$

qui exprime la relation entre l'erreur et le résidu, et,

$$u_h = u_h^{k+\frac{1}{2}} - e_h^{k+\frac{1}{2}} \quad (3.14)$$

qui exprime l'écart à la solution exacte du modèle discret.

Les composantes hautes fréquences de l'erreur  $e_h^{k+\frac{1}{2}}$  ont été atténuées lors de cette première phase de lissage; ce sont les composantes basse fréquence qui persistent, mais elles sont mieux représentées sur la grille grossière; d'où la deuxième phase.

2. **Correction sur la grille grossière** :

On cherche à calculer une fonction discrétisée sur la grille grossière, notée  $e_{2h}^{k+\frac{1}{2}}$ , tel que :

$$e_h^{k+\frac{1}{2}} \approx P e_{2h}^{k+\frac{1}{2}} \quad (3.15)$$

On aboutit alors au système suivant :

$$R A_h P e_{2h}^{k+\frac{1}{2}} = R r_h^{k+\frac{1}{2}} \quad (3.16)$$

Donc, on résout exactement (3.16) sur la grille grossière. On prolonge ensuite le résultat obtenu sur la grille fine :

$$e_h^{k+\frac{1}{2}} = P e_{2h}^{k+\frac{1}{2}} \quad (3.17)$$

d'où l'approximation corrigée de  $u_h$  :

$$u_h^{k+\frac{3}{2}} = u_h^{k+\frac{1}{2}} - e_h^{k+\frac{1}{2}} \quad (3.18)$$

Cette phase permet d'atténuer les composantes basse fréquence de l'erreur, mais d'autres modes basse fréquence peuvent réapparaître. d'où l'intérêt de finir l'algorithme par une phase de lissage.

### 3. Lissage sur la grille fine :

Cette phase consiste à calculer  $u_h^{k+1}$  tel que :

$$u_h^{k+1} = G_h^{\alpha_1} u_h^{k+\frac{3}{2}} + b_h \quad (3.19)$$

En rassemblant les résultats précédents, on obtient :

$$u_h^{k+1} = G_h^{\alpha_2} u_h^0 + g_h \quad (3.20)$$

Où la matrice d'amplification a la structure suivante :

$$G_h^{\alpha_2} = G_h^{\alpha_1} (I - P(RA_h P)^{-1} RA_h) G_h^{\alpha_1} \quad (3.21)$$

Pour démontrer l'efficacité du cycle bigrille, on se place dans le cadre du problème modèle présenté précédemment.

Supposons que la matrice  $G_h^{\alpha_2}$  s'exprime comme le polynôme suivant la matrice  $A_h$  :

$$G_h^{\alpha_2} = \mathcal{P}(h^2 A_h) \quad (3.22)$$

En particulier (voir [28]), dans le cas d'un lissage par la méthode de Richardson avec 3 pseudo-pas de temps optimisés  $(\tau_l)_{l=1}^3$ , on a :

$$\mathcal{P}(\lambda) = \prod_{l=1}^3 (1 - \tau_l \lambda) \quad (3.23)$$

Et comme démontré dans [28], le rayon spectrale du cycle bigrille idéal s'exprime comme suit :

$$\rho(G_h^{\alpha_2}) = \max_{m=1, \dots, M_{2h+1}} \left( \frac{d_m^2 + d_{M_{h+1}+1_m}^2}{4} \right) \quad (3.24)$$

avec

$$d_m = d(\lambda_m^h) \quad (3.25)$$

où  $(\lambda_m^h)_{m=1}^{M_h}$  sont les valeurs propres de la matrice  $h^2 A_h$ , et  $d$  est une fonction définie par :

$$d(\lambda) = \sqrt{(\lambda) \mathcal{P}(\lambda)} \quad (3.26)$$

admet la borne supérieure suivante indépendamment de  $h$  :

$$B_0 = \max_{\lambda \in [0,2]} \left( \frac{d(\lambda)^2 + d(4-\lambda)^2}{4} \right) \quad (3.27)$$

En particulier,  $B_0 \approx 0.04$  dans le cas du cycle de Richardson précédent. On déduit que la vitesse de convergence est indépendante de  $h$  (la finesse du maillage). Ceci confirme l'efficacité du cycle bigrille idéal.

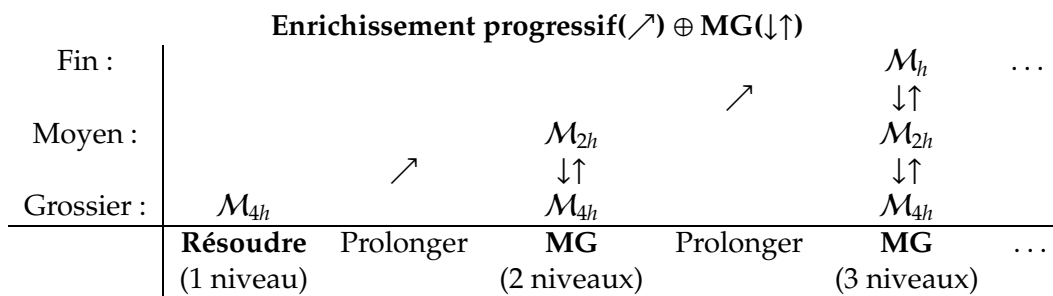
### 3.2.5 Algorithme multigrille (MG)

Si on remplace dans l'algorithme précédent la phase de correction grille grossière par un algorithme bigrille idéal, on obtient l'algorithme trigrille idéal. En généralisant ce concept à plusieurs niveaux, on aboutit à l'algorithme multigrille idéal.

On démontre dans [28] que le coût globale de la résolution par l'algorithme multigrille idéal est proportionnel à  $N \log N$ , avec  $N$  est le nombre de degrés de liberté.

### 3.2.6 Méthode multigrille complète (FMG)

On aboutit enfin à un algorithme de convergence optimale ; il est de complexité proportionnelle au nombre de degrés de liberté ou d'une manière équivalente, sa vitesse de convergence est indépendante de la densité du maillage. Cette approche combine l'algorithme multigrille à une technique d'enrichissement progressif du maillage, et s'appelle méthode multigrille complète (en anglais Full Multigrid Method (FMG)). Elle consiste à résoudre le problème sur une grille grossière par un algorithme monogrille, on prolonge ensuite le résultat sur une grille plus fine, puis on résout le problème par un algorithme bigrille, on prolonge le résultat sur une grille encore plus fine, on résout itérativement le problème par un algorithme trigrille et ainsi de suite. On schématise dans la figure suivante l'algorithme multigrille complète dans le cas de 3 niveaux :



## 3.3 Approches hiérarchiques en optimisation de forme

Le grand potentiel des méthodes multigrilles et leur efficacité à résoudre des problèmes divers ont motivé plusieurs auteurs à appliquer les principes basiques des méthodes multigrilles en optimisation de forme. Au départ, de telles méthodes ont été appliquées avec succès en optimisation de forme en aérodynamique par Jameson [59] pour la résolution des équations d'état et les équations adjointes continues. Ces équations ont été combinées avec les méthodes multigrilles en utilisant le concept one-shot par Ta'asan [107] en dimension finie. Arian et Ta'asan [7] ont étendu par la suite cette méthode en dimension infinie. Une autre approche multiniveau en optimisation de forme a été proposée par Dervieux et Beux [14] qui consiste à introduire un préconditionneur dans l'algorithme de descente, vu comme un opérateur de lissage, pour que la forme et le gradient soient dans le même espace de régularité, et, inspiré par les propriétés des méthodes multigrilles, cet opérateur peut être construit par la composition d'une suite symétrique d'opérateurs de lissage et de projection. Ensuite, Marco et Beux [74] ont combiné cette nouvelle approche avec l'idée de Ta'asan [107] avec succès. En se basant sur la théorie des approximations consistantes en théorie du contrôle

introduites par Polak [92], qui étudie comment les paramètres de discrétisation peuvent être raffinés dans un algorithme d'optimisation, Pironneau et al [35][91] ont développé un processus hiérarchique de raffinement de maillage en optimisation de forme.

On présente ici, suite aux articles [29][31][32][33][73], notre approche multiniveau en optimisation de forme basée sur la paramétrisation de Bézier. On définit des espaces emboîtés de paramétrisation par l'intermédiaire du processus d'élévation du degré. On considère au départ le cas 2D où la forme est paramétrisée directement. Le concept est ensuite généralisé au cas 3D en utilisant une paramétrisation de la forme par FFD (Free-Form Deformation).

### 3.3.1 Motivations et illustration par un problème modèle

Soit  $\gamma$  une forme associée à la courbe paramétrée  $(x(t), y(t))$  telle que :

$$\begin{cases} x(t) = B_n(t)^T X \\ y(t) = B_n(t)^T Y \end{cases} \quad (3.28)$$

où  $t \in [0, 1]$ ,  $B_n(t)^T = (B_n^k(t))_{0 \leq k \leq n}$  est le vecteur ligne des polynômes de Bernstein, et  $(X, Y)$  est le polygone de contrôle de degré  $n$  qui paramétrise la forme  $\gamma$ .

On suppose que l'arc cible, constitué par le graphe d'équation  $y = \bar{y}(x)$ , est aussi une courbe de Bézier de degré  $n$  associé au polygone de contrôle  $(X^0, \bar{Y})$ .

On définit la fonction paramétrique à minimiser, pour un support fixé  $X = X^0$ , comme suit :

$$J_n(Y) = \mathcal{J}(\gamma)$$

avec  $\mathcal{J}(\gamma)$  une fonctionnelle intrinsèque telle que :

$$\mathcal{J}(\gamma) = \int_0^1 \frac{1}{2} [y(x) - \bar{y}(x)]^2 dx \quad (3.29)$$

Alors,

$$J_n(Y) = \int_0^1 \frac{1}{2} \left[ \underbrace{y(t)}_Y - \underbrace{\bar{y}(t)}_{\bar{Y}} \right]^2 \underbrace{x^0(t)'}_{X^0} dt \quad (3.30)$$

On déduit que le gradient de  $J_n$  dépend linéairement du vecteur  $Y$  (voir chapitre (5)) :

$$J'_n(Y) = A(X^0) Y - b(X^0, \bar{Y}) \quad (3.31)$$

où :

$$A(X^0) = \int_0^1 \underbrace{B_n(t) B_n(t)^T}_{\substack{\text{matrice} \\ (n+1) \times (n+1)}} \underbrace{n B_{n-1}(t)^T \Delta X}_{\substack{\text{forme linéaire} \\ \text{en } X^0}} dt \quad (3.32)$$

Donc, le problème de minimisation de la fonctionnelle  $J_n$  est équivalent à la résolution du système linéaire suivant :

$$A(X^0) Y = b(X^0, \bar{Y}) \quad (3.33)$$

Dorénavant, on considère les notations suivantes :  $A = A(X^0)$  et  $b = b(X^0, \bar{Y})$ .

On résout le système linéaire par une méthode itérative ; on choisit par exemple l'itération de Jacobi, qui s'écrit :

$$Y^{k+1} = G_J Y^k + D^{-1} b \quad (3.34)$$

avec

$$G_J = I - D^{-1} A$$

où  $D$  est la matrice diagonale de  $A$ . Ceci est équivalent à une méthode de gradient classique.

Dans l'absence d'un calcul formel des valeurs propres de la matrice  $G_J$ , et en guise d'illustration, le calcul est effectué numériquement. Il s'agit dans un premier temps de regarder la dépendance du degré de paramétrisation avec le comportement de la convergence de l'algorithme. La figure (3.3) montre que le rayon spectral s'approche de 1 au fur et à mesure que le degré de paramétrisation augmente. À la figure (3.4), le nombre de conditionnement augmente exponentiellement avec le degré. On conclut donc que la convergence est d'autant plus lente avec l'augmentation du degré, c'est ce qu'on appelle la raideur numérique.

**Illustration de la raideur numérique**

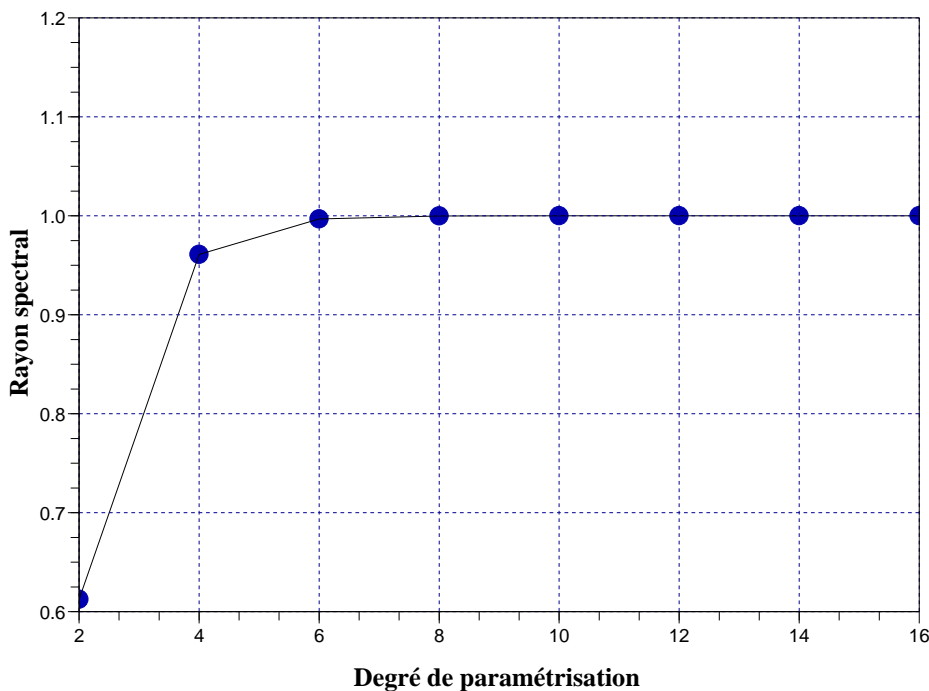


FIGURE 3.3 – Rayon spectral en fonction du degré de paramétrisation



### Illustration de la raideur numérique

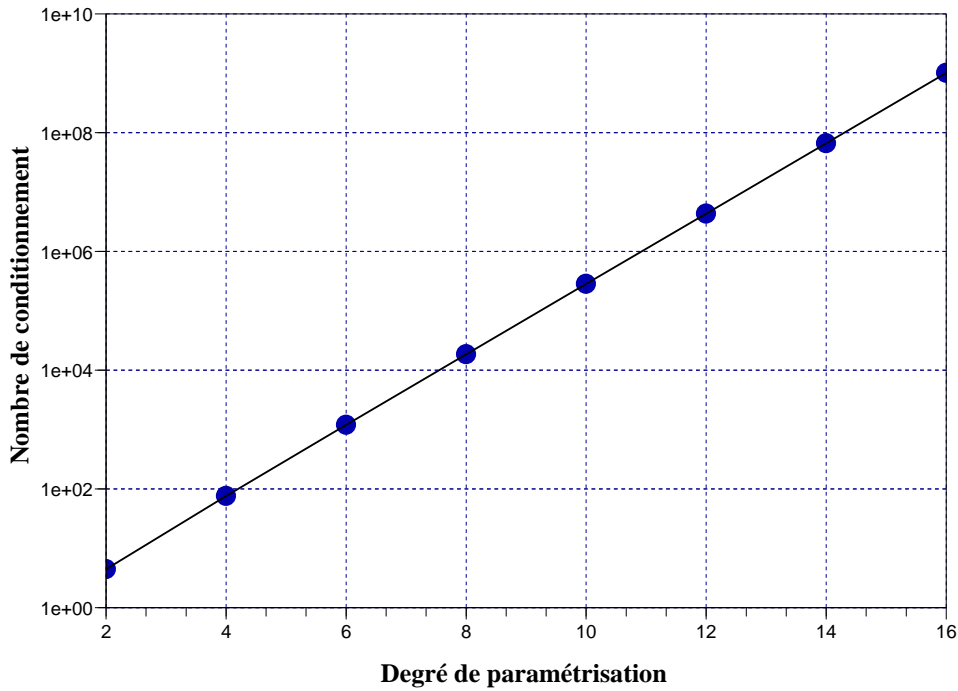


FIGURE 3.4 – Nombre de conditionnement en fonction du degré de paramétrisation

Ce constat très intéressant permet une analogie avec la théorie multigrille en raisonnant sur le paramètre  $h$ , qui désigne la finesse du maillage, au lieu du degré de la paramétrisation. D'où l'idée d'aborder le problème d'optimisation multiniveau à l'instar des méthodes multigrilles en s'appuyant sur une hiérarchie de paramétrisations.

### 3.3.2 Méthode multiniveau basée sur la paramétrisation

#### 3.3.2.1 Représentation de forme

Nous représentons la forme, c'est à dire en 2D une courbe plane, par la paramétrisation de Bézier, donnée par :

$$x(t) = \sum_{k=0}^n B_n^k(t) x_k, \quad y(t) = \sum_{k=0}^n B_n^k(t) y_k \quad (3.35)$$

avec  $t \in [0, 1]$ ,  $n$  est le degré de la paramétrisation, et

$$B_n^k(t) = C_n^k t^k (1-t)^{n-k} \quad (3.36)$$

est le polynôme de Bernstein,  $C_k^n = \frac{n!}{k!(n-k)!}$ , et

$$P_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix} \quad (k = 0, 1, \dots, n) \quad (3.37)$$

est le point de contrôle. Les coordonnées de  $P_k$  se divisent en deux vecteurs

$$X = \{x_k\}, \quad Y = \{y_k\}, \quad k = 0, 1, \dots, n, \quad (3.38)$$

et on désigne par le vecteur  $X$  le support de la paramétrisation, et par  $Y$  le vecteur de forme. Si on prend  $P(t) = (x(t), y(t))$ , l'équation (3.35) se réécrit comme suit :

$$P(t) = \sum_{k=0}^n B_n^k(t) P_k \quad (3.39)$$

Le choix de la base des polynômes de Bernstein comme base de représentation au lieu de la base canonique, à deux principales mérites :

- elle définit une courbe lisse même si le polygone de contrôle est irrégulier.
- elle facilite le calcul explicite des dérivées successives, et donc un contrôle des éléments de contact. En effet, en dérivant (3.39) par rapport à  $t$   $m$ -fois, il vient :

$$\begin{cases} P'(t) = n \sum_{k=0}^{n-1} B_{n-1}^k(t) (P_{k+1} - P_k) \\ P''(t) = n(n-1) \sum_{k=0}^{n-2} B_{n-2}^k(t) (P_{k+2} - 2P_{k+1} + P_k) \\ \dots \\ P^{(m)}(t) = \frac{n!}{(n-m)!} \sum_{k=0}^{n-m} B_{n-m}^k(t) (\Delta^m P_k) \end{cases} \quad (3.40)$$

où  $\Delta$  est l'opérateur de différence finie avancée ( $\Delta P_k = P_{k+1} - P_k$ ) et  $\Delta^m$  est la puissance  $m$  de l'opérateur  $\Delta$  ( $\Delta^2 P_k = P_{k+2} - 2P_{k+1} + P_k, \dots$ )

Donc, par l'intermédiaire des points de contrôle  $P_0, P_1, \dots, P_n$  on peut spécifier l'extrémité de l'arc  $P(0) = P_0$  (resp.  $P(1) = P_n$ ) et les éléments de contact  $P'(0), P''(0), \dots$  (resp.  $P'(1), P''(1), \dots$ ). En particulier, la courbe admet  $P_0 P_1$  (resp.  $P_{n-1} P_n$ ) comme tangente à  $t = 0$  (resp.  $t = 1$ ). La courbure, quant à elle, se contrôle par  $P_2$  à l'origine, et par  $P_{n-2}$  à l'extrémité de la courbe, etc. La tangente verticale peut être imposée en prenant ( $x_1 = x_0 = 0$  et  $y_1 \neq y_0 = 0$ ), ceci représente une contrainte standard au bord d'attaque d'un profil d'aile.

On illustre dans la figure (3.5) le profil d'aile d'avion RAE2822, approché par une courbe de Bézier de degré 16, et les deux polygones de contrôle associés à son extradados et son intrados.

### 3.3.2.2 Construction des paramétrisations emboîtées

L'élévation du degré, présentée dans le chapitre précédent, est la plus importante des propriétés de la paramétrisation de Bézier ; elle représente la base même de notre construction des algorithmes hiérarchiques. Ce processus, illustré dans la figure (3.6), nous permet de construire des espaces de paramétrisation rigoureusement emboîtés. Soit  $(n_k)_{k=1}^p$  une suite des entiers strictement croissante. Donc, à partir d'une paramétrisation grossière de degré  $n_1$ , associée au polygone de contrôle  $P_k = (x_k^{(1)}, y_k^{(1)}) (k = 1, \dots, n_1)$ , on peut construire une paramétrisation plus fine de degré  $n_2$  en appliquant  $n_2 - n_1$  élévations du degré au polygone initial, en appliquant encore  $n_3 - n_2$  élévations du degré, on peut obtenir une paramétrisation encore plus fine de degré  $n_3$ , etc.

La figure (3.7) illustre la construction de trois paramétrisations emboîtées de degrés  $(n_1, n_2, n_3) = (4, 8, 16)$ . Chaque courbe possède une tangente verticale à l'origine, cette contrainte est imposée par la condition  $x_0^{(m)} = x_1^{(m)} = 0$ , pour chaque niveau ( $m = 1, 2, 3$ ).

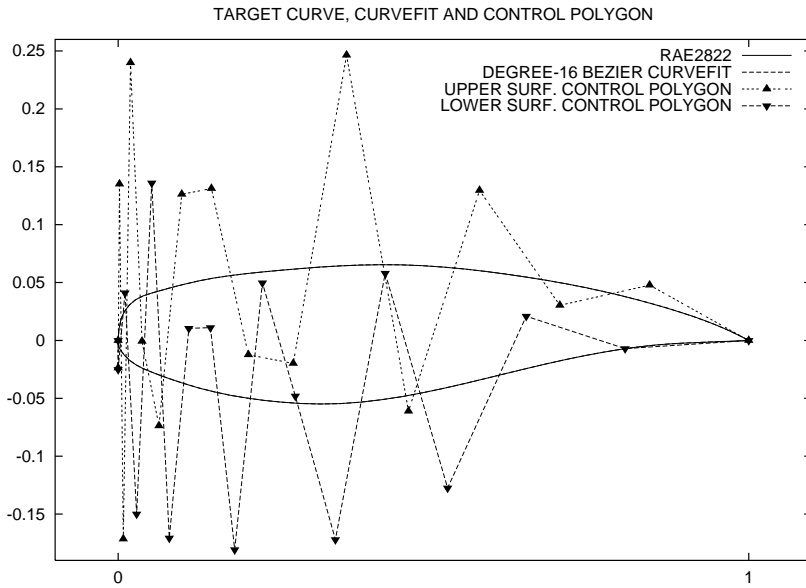


FIGURE 3.5 – La représentation du profil d’aile RAE2822 par la paramétrisation de Bézier (de [29]).

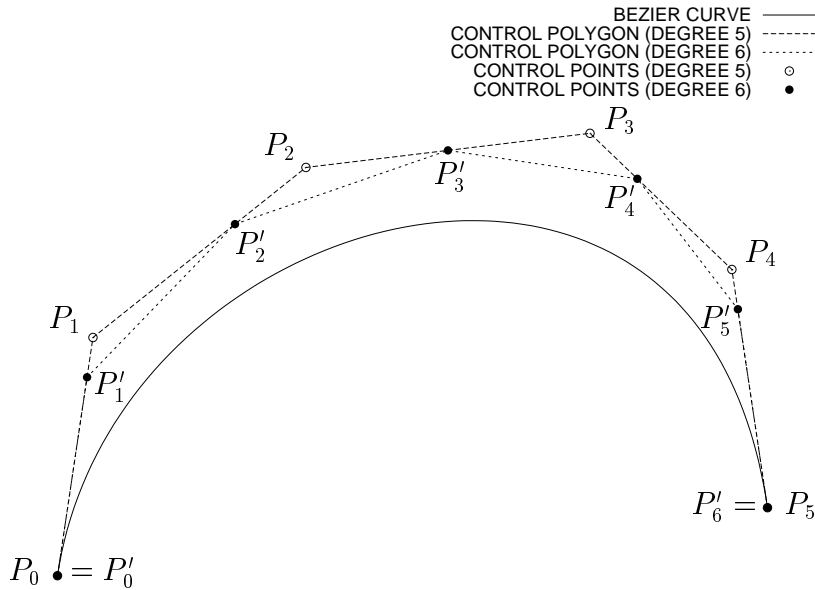


FIGURE 3.6 – A partir de la courbe de Bézier de degré 5, associée au polygone de contrôle qui relie les points  $P_k$ , le nouveau polygone de contrôle qui relie les points  $P'_k$  de degré 6, obtenu par combinaison convexe, correspond à la même courbe géométrique (de [29]).

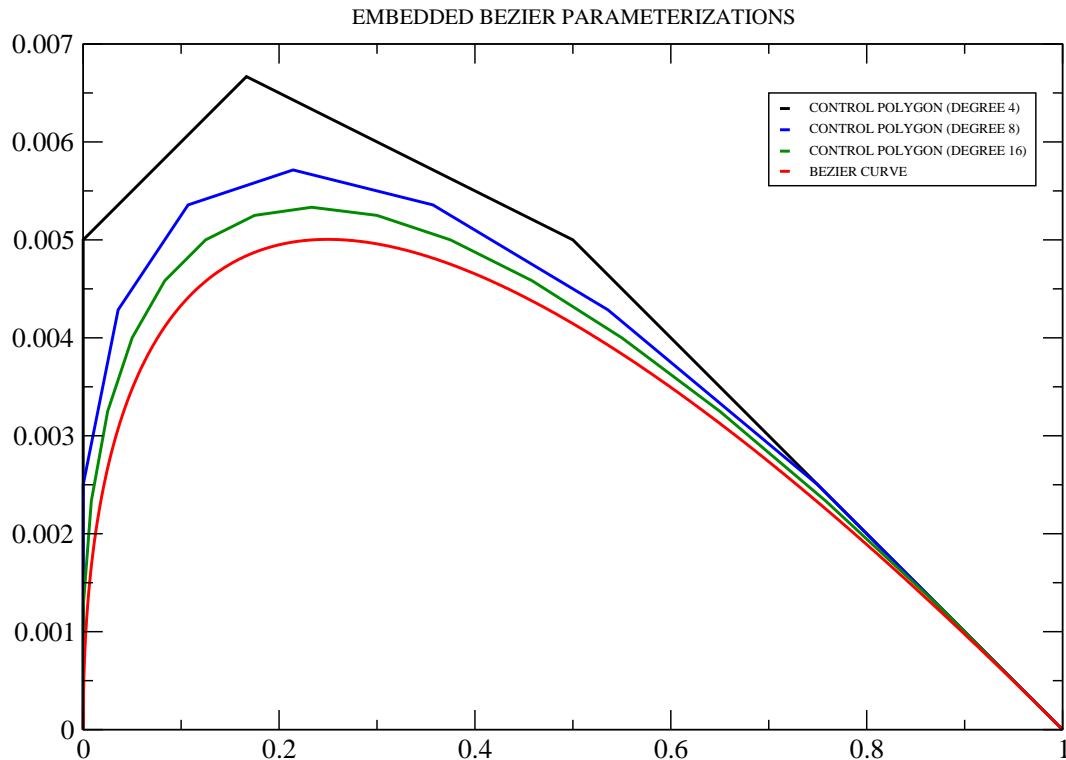


FIGURE 3.7 – Exemple de trois paramétrisations de Bézier emboîtées ( $n = 4, 8, 16$ )

Par conséquent, chaque courbe de Bézier de degré  $n_m$  ( $m < p$ ) correspondant au polygone de contrôle  $P_k = (x_k^{(m)}, y_k^{(m)}) (k = 1, \dots, n_m)$ , peut s'exprimer exactement comme une courbe de Bézier de degré plus élevé  $n_h$  ( $h > m$ ). Nous définissons, par la suite, différents algorithmes d'optimisation multiniveaux, en utilisant notre construction d'une suite d'espaces de paramétrisations rigoureusement emboîtés.

### 3.3.2.3 Enrichissement progressif de la paramétrisation

Le plus simple des algorithmes d'optimisation hiérarchique est celui correspondant à un schéma "montant sans retour". Il est analogue, dans le contexte de la résolution pure d'EDP en utilisant plusieurs maillages, à la méthode multigrille cascadique (En anglais Nested Iteration) dans laquelle on effectue seulement des transferts par prolongement vers une grille plus fine. Il consiste, initialement, à résoudre le problème d'optimisation en utilisant une paramétrisation grossière (ex.  $n_1 = 4$ ). Cette première phase permet d'accélérer la convergence, car avec un nombre faible de paramètres, le processus n'est pas raide. On interpole ensuite la solution obtenue, exactement, dans un espace de paramétrisation de degré plus élevé par le processus d'élévation du degré (ex.  $n_2 = 8$ ). En réinitialise alors l'optimisation avec plus de paramètres, et avec une bonne condition initiale. Le résultat est encore rigoureusement exprimé dans un espace de paramétrisation plus fine (ex.  $n_3 = 16$ ), pour servir de condition initiale à une nouvelle optimisation, etc.

Ce processus est schématisé à la figure (3.8) dans l'hypothèse de 3 niveaux ( $\mathcal{B}_k$  est l'espace de recherche associé à la paramétrisation de Bézier de degré  $n_k, k = 1, 2, 3$ ) :

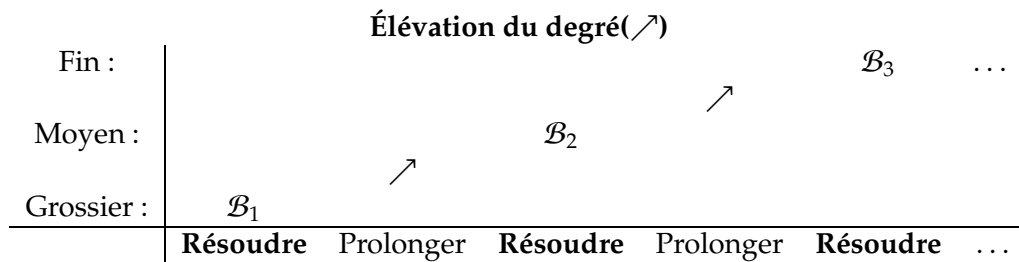


FIGURE 3.8 – Schéma de l'algorithme d'enrichissement progressif de paramétrisation, dans le cas de 3 niveaux

Les mérites de cet algorithme sont dues à deux principales raisons :

1. La résolution du problème d'optimisation avec plusieurs niveaux de paramétrisation, en commençant par le grossier, permet d'accélérer la convergence, et de résoudre la raideur du système progressivement.
2. L'opérateur de prolongement utilisé (l'élévation du degré) permet un transfert exact de paramétrisations par passage d'un niveau à un autre.

### 3.3.2.4 Algorithme d'optimisation V-cycle

Les algorithmes hiérarchiques pour la résolution des EDP ont prouvé plus d'efficacité, en utilisant des algorithmes V-cycle, qui commencent, contrairement à l'enrichissement progressif du maillage, par un maillage fin [28]. Ce concept peut être repris en optimisation de forme, en utilisant la perturbation de la forme au niveau grossier à partir de la dernière estimation au niveau fin.

Soit  $\mathcal{B}_1$  (resp.  $\mathcal{B}_2$ ) l'espace de recherche associé à la paramétrisation de Bézier de degré  $n_1$  (resp.  $n_2$ ) ; dans cette présente approche  $\mathcal{B}_1 \subset \mathcal{B}_2$ . Nous définissons ci-dessous l'algorithme d'optimisation V-cycle à deux niveaux ( $n_1 < n_2$ ) :

1. En effectue quelques itérations d'optimisation dans l'espace fin  $\mathcal{B}_2$  ; soit  $\Gamma_1$  la forme obtenue.
2. On résout ensuite le problème d'optimisation (à convergence complète) dans  $\mathcal{B}_1$  ; il consiste à perturber  $\Gamma_1$  dans l'espace grossier  $\mathcal{B}_1$  ; Soit  $\Gamma_2 = \Gamma_1 + E_{n_1}^{n_2} \delta\Gamma_1$  la forme obtenue, où  $E_{n_1}^{n_2}$  est l'opérateur de l'élévation du degré du niveau grossier au niveau fin et  $\delta\Gamma_1$  appartient à  $\mathcal{B}_1$ .
3. la forme  $\Gamma_2$ , exprimée dans l'espace fin  $\mathcal{B}_2$ , va servir comme une condition initiale à une nouvelle optimisation au niveau fin ; on effectue seulement quelques itérations.

On schématise dans la figure (3.9) l'algorithme d'optimisation V-cycle à deux niveaux :

Cet algorithme peut se généraliser facilement à un V-cycle de plus de deux niveaux, dans lequel la convergence complète est nécessaire seulement au niveau grossier.

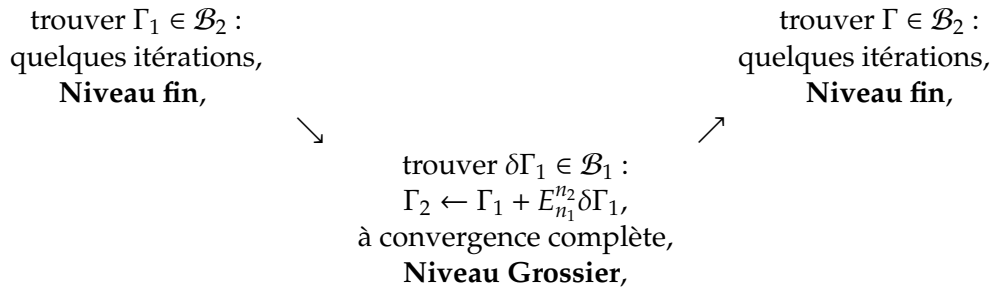


FIGURE 3.9 – Algorithme d’optimisation V-cycle à deux niveaux

### 3.3.2.5 Algorithme multiniveaux d’optimisation de forme complet (FMOSA)

L’algorithme complet est défini en combinant l’algorithme d’enrichissement progressif de la paramétrisation et l’algorithme V-cycle. Nous définissons ci-dessous l’algorithme dans le cas de 3 niveaux de paramétrisation de degrés respectifs  $n_1, n_2$  et  $n_3$  ( $n_1 < n_2 < n_3$ ); soit  $\mathcal{B}_1$  (resp.  $\mathcal{B}_2, \mathcal{B}_3$ ) l’espace de recherche associé à la paramétrisation de Bézier de degré  $n_1$  (resp.  $n_2, n_3$ ); dans cette présente approche  $\mathcal{B}_1 \subset \mathcal{B}_2 \subset \mathcal{B}_3$ .

1. Nous résolvons le problème d’optimisation dans l’espace grossier  $\mathcal{B}_1$ ;
2. La solution obtenue est exprimée rigoureusement dans un espace plus fin  $\mathcal{B}_2$ , pour servir comme condition initiale à une nouvelle optimisation;
3. Nous résolvons ensuite le problème d’optimisation dans un espace plus fin  $\mathcal{B}_2$  en utilisant l’algorithme V-cycle à deux niveaux, schématisé dans la figure (3.9);
4. Le résultat est encore exprimé rigoureusement dans l’espace le plus fin  $\mathcal{B}_3$ , pour servir comme condition initiale à une nouvelle optimisation;
5. Nous résolvons enfin le problème d’optimisation dans l’espace le plus fin  $\mathcal{B}_3$  en utilisant l’algorithme V-cycle à 3 niveaux.

On schématise dans la figure (3.10) l’algorithme d’optimisation FMOSA (Full and Multi-level Optimum-Shape Algorithm) à 3 niveaux :

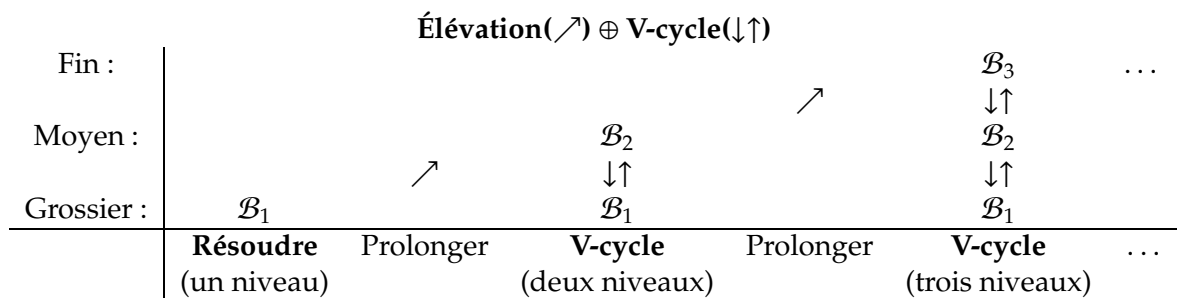


FIGURE 3.10 – Algorithme d’optimisation FMOSA à trois niveaux

### 3.4 Conclusion

On a rappelé à la première partie de ce chapitre les principes des méthodes multigrilles pour résoudre les systèmes linéaires. Les mérites et les propriétés intéressantes de ces méthodes ont été une source d'inspiration pour développer une approche similaire, en optimisation de forme, qui se base sur une hiérarchie de paramétrisations.

La nature des transferts d'informations d'un niveau à un autre est l'un des points essentiels qui garantit l'efficacité des méthodes multiniveaux. C'est d'ailleurs la clé de réussite de notre approche hiérarchique qui utilise le processus d'élévation de degré, qui permet un transfert exact dans le sens montant, et une correction au niveau grossier, qui permet de perturber la solution précédemment optimisée.

Plusieurs tests numériques vont être présentés aux chapitre suivant pour expérimenter notre approche hiérarchique en optimisation de forme.

## Chapitre 4

# Expérimentation numérique d'algorithmes multiniveaux

### 4.1 Introduction

On se propose dans cette partie de tester numériquement nos stratégies d'optimisation hiérarchique qui ont été présentées dans le chapitre précédent. Le but est de montrer que les algorithmes multiniveaux en optimisation de forme se révèlent efficace pour résoudre la raideur numérique, qui augmente avec le nombre des paramètres d'optimisation, et pour accélérer la convergence itérative. Pour cela, on considère trois problèmes d'optimisation :

1. un problème modèle de calcul des variations formulé comme un problème d'optimisation géométrique. Le but est de tester les algorithmes multiniveaux sur un problème simple afin de se familiariser avec le concept hiérarchique en optimisation de forme et d'avoir une idée sur le réglage des paramètres d'optimisation,
2. un problème d'optimisation d'une voilure d'avion d'affaire en régime transsonique qui consiste à réduire la traînée avec une contrainte sur la portance,
3. un problème d'optimisation du nez et de la voilure d'un avion d'affaire supersonique qui consiste à réduire le bang sonique.

### 4.2 Problème modèle

Ce problème modèle à été étudié en détail dans [34] où les auteurs ont introduit et analysé une classe de problèmes d'optimisation géométrique formulée comme un problème de calcul des variations. Ici on utilise ce modèle pour tester des algorithmes numériques en optimisation de forme dans un contexte simplifié ayant les caractéristiques d'un autre problème plus complexe [33][11].

#### 4.2.1 Problème variationnel

On considère un arc de courbe reliant l'origine  $(0, 0)$  au point  $(1, 0)$  et admettant la paramétrisation ci-dessous :

$$\begin{cases} x = x(t) \\ y = y(t) \end{cases} \quad (4.1)$$



où  $0 \leq t \leq 1$ ,  $x(t)$  et  $y(t)$  sont  $C^2$ ,  $x(t)$  est monotone croissante et  $y(t) \geq 0$ , avec les conditions aux limites suivantes :

$$\begin{cases} x(0) = 0, x(1) = 1 \\ y(0) = y(1) = 0 \end{cases} \quad (4.2)$$

On introduit les quantités suivantes :

$$p = \int_0^1 \sqrt{x'^2(t) + y'^2(t)} \omega(t) dt \quad (4.3)$$

$$\mathcal{A} = \int_0^1 y(t) x'(t) \omega(t) dt \quad (4.4)$$

La fonction poids  $\omega(t)$  est positive et réglable. Dans le cas particulier où  $\omega(t) = 1$ , les quantités  $p$  et  $\mathcal{A}$  sont respectivement la longueur de l'arc et l'aire de la région entre l'arc et l'axe des abscisses.

On considère la fonctionnelle suivante :

$$\mathcal{J}(y) = \frac{p^\alpha}{\mathcal{A}} \quad (4.5)$$

où  $\alpha$  est une constante positive.

Cette fonctionnelle possède des propriétés intéressantes qui ont été démontrées dans [34] :

1. Pour  $\alpha \geq 2$ , la fonctionnelle  $\mathcal{J}$  est strictement convexe.
2. Si  $\alpha > 1$  et si la fonctionnelle  $\mathcal{J}(y)$  admet un point de stationnarité en  $y = y(t)$ , alors ce point est unique et réalise le minimum global de la fonctionnelle  $\mathcal{J}(y)$ , qui est unimodale.

On cherche par la suite, pour un choix particulier de la constante  $\alpha$  et de la fonction de poids  $\omega$ , à reproduire une géométrie particulière en minimisant la fonctionnelle  $\mathcal{J}$  par rapport à  $y(t)$  avec  $x(t)$  fixé.

**Condition de stationnarité :** La fonctionnelle  $\mathcal{J}(y)$  est stationnaire ssi :

$$\alpha \frac{\delta p}{p} - \frac{\delta \mathcal{A}}{\mathcal{A}} = 0 \quad (4.6)$$

ceci est équivalent à :

$$\alpha \mathcal{A} \delta p = p \delta \mathcal{A} \quad (4.7)$$

Or,

$$\delta p = \int_0^1 \frac{1}{2} (x'^2 + y'^2)^{-\frac{1}{2}} 2 y' \delta y' \omega(t) dt \quad (4.8)$$

donc si  $\delta y' = (\delta y)'$ , on obtient par intégration par partie que :

$$\begin{aligned} \delta p &= \left[ \omega y' (x'^2 + y'^2)^{-\frac{1}{2}} \delta y \right]_0^1 - \int_0^1 \frac{d}{dt} \left( \omega y' (x'^2 + y'^2)^{-\frac{1}{2}} \right) \delta y dt \\ &= \int_0^1 \left[ -(\omega' y' + \omega y'') (x'^2 + y'^2)^{-\frac{1}{2}} + \frac{\omega y'}{2} (x'^2 + y'^2)^{-\frac{3}{2}} (2 x' x'' + 2 y' y'') \right] \delta y dt \end{aligned} \quad (4.9)$$

si  $\delta y(0) = \delta y(1) = 0$ . Après simplification, on déduit que :

$$\begin{aligned}\delta p &= \int_0^1 \phi \delta y dt \\ \delta \mathcal{A} &= \int_0^1 \psi \delta y dt\end{aligned}\tag{4.10}$$

avec

$$\begin{aligned}\phi(t) &= \omega \frac{x'(x''y' - x'y'')}{(x'^2 + y'^2)^{\frac{3}{2}}} - \omega' \frac{y'}{\sqrt{x'^2 + y'^2}} \\ \psi(t) &= \omega(t) x'(t)\end{aligned}\tag{4.11}$$

La condition de stationnarité s'écrit alors comme suit :

$$\forall \delta y, \int_0^1 [\alpha \mathcal{A} \phi - p \psi] \delta y dt = 0\tag{4.12}$$

ceci est équivalent à l'équation différentielle suivante :

$$\forall t, \frac{x''y' - x'y''}{(x'^2 + y'^2)^{\frac{3}{2}}} - \frac{\omega'}{\omega} \frac{y'}{x' \sqrt{x'^2 + y'^2}} = \frac{p}{\alpha \mathcal{A}} (= \text{const.})\tag{4.13}$$

**Cas particulier** ( $\omega(t) = 1$ ) : Dans cet exemple, l'équation différentielle intégrale (4.17) se simplifie en :

$$-\frac{1}{r} = \frac{p}{\alpha \mathcal{A}}\tag{4.14}$$

avec  $r$  est le rayon de courbure.

Cette équation caractérise un arc circulaire concave de rayon  $|r| = -r$ , dont le centre, compte tenu des conditions aux limites, a pour abscisse  $x = \frac{1}{2}$ . On introduit les paramètres  $h$  et  $\theta_0$  comme dans la figure (4.1), on déduit les relations géométriques suivantes :

$$\begin{cases} \frac{1}{2} = |r| \cos(\theta_0) \\ h = |r| \sin(\theta_0) \\ p = (\pi - 2\theta_0) |r| \\ \mathcal{A} = \frac{1}{2}(\pi - 2\theta_0)r^2 - \frac{1}{2}h\end{cases}\tag{4.15}$$

Il en résulte que :

$$\frac{1}{|r|} = \frac{\gamma |r|}{\frac{1}{2}\alpha(\gamma - \sin \gamma)r^2}\tag{4.16}$$

avec  $\gamma = \pi - 2\theta_0$

En particulier, pour  $\alpha = 2$  et  $\gamma = \pi$ ; donc  $\theta_0 = 0$  et  $|r| = \frac{1}{2}$ . Il s'agit alors d'un demi cercle de rayon  $\frac{1}{2}$  et de centre  $(\frac{1}{2}, 0)$  (voir figure (4.2)).

Il s'ensuit que :

$$\mathcal{J} = \frac{p^2}{\mathcal{A}} = \frac{(\pi/2)^2}{\frac{1}{2}\pi(\frac{1}{2})^2} = 2\pi\tag{4.17}$$

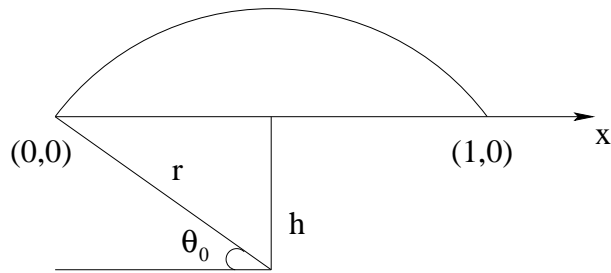


FIGURE 4.1 – Arc optimal (notations) [34]

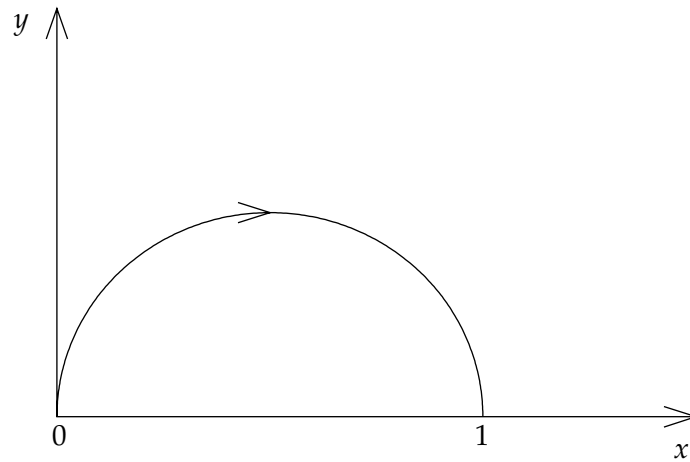


FIGURE 4.2 – Demi cercle correspondant au cas  $\alpha = 2$  et  $\gamma = \pi$ .

**Problème inverse** ( $\omega(t) \neq 1$ ) : Il est possible avec un choix judicieux de la fonction  $\omega(t)$  et de la constante  $\alpha$  d'obtenir des caractéristiques géométriques intéressantes, par exemple un profil aérodynamique.

On introduit les conditions de régularités suivantes :

**Définition 4.2.1**

La paramétrisation  $(x(t), y(t))$  ( $0 \leq t \leq 1$ ) est admissible si elle satisfait les conditions aux limites :

$$x(0) = y(0) = x(1) - 1 = y(1) = 0 \tag{4.18}$$

et les hypothèses suivantes :

- $x(t)$  et  $y(t)$  sont de classe  $C^2$  et  $x'(t)^2 + y'(t)^2 > 0$  uniformément.
- $x(t)$  est monotone croissante,  $y(t)$  est unimodale et à l'unique point  $t = t_0$  ( $0 < t_0 < 1$ ) où  $y(t)$  atteint son maximum, on a  $y'(t_0) = 0$  et  $y''(t_0) < 0$ .

Dans le théorème suivant, on se restreint à ce choix des paramétrisations admissibles.

**Théorème 4.2.1 (démontré dans [34])**

Soit  $(x(t), y(t))$  ( $0 \leq t \leq 1$ ) une paramétrisation admissible.

Soit  $C$  tel que :

$$C = \frac{p}{\alpha \mathcal{A}} = \left( \frac{x''y' - x'y''}{(x'^2 + y'^2)^{\frac{3}{2}}} \right)_{|t=t_0} = \left( \frac{-y''}{x'^2} \right)_{|t=t_0} = \left( \frac{-d^2y}{dx^2} \right)_{|t=t_0} > 0 \quad (4.19)$$

avec  $t = t_0$  est le point où  $y = y(t)$  atteint son maximum.

La fonction  $\omega(t)$ , déduit de l'équation différentielle (??) comme suit,

$$\ln \omega(t) = \int \left( \frac{x''y' - x'y''}{(x'^2 + y'^2)^{\frac{3}{2}}} - C \right) \cdot \left( \frac{x' \sqrt{x'^2 + y'^2}}{y'} \right) dt, \quad (\omega(0) = 1) \quad (4.20)$$

est unique, de classe  $C^1$  au moins et uniformément positive. En plus, si  $\alpha$  est calculé à partir de la formule :

$$\alpha = \frac{p}{C \mathcal{A}} \quad (4.21)$$

la fonctionnelle  $\mathcal{J}(y)$  est stationnaire par rapport à  $y = y(t)$ , pour  $x = x(t)$  fixé.

D'un point de vu pratique, ce théorème implique qu'une forme arbitraire, comme par exemple un profil d'aile, qui peut être décrit par une paramétrisation admissible, est celle où la fonctionnelle  $\mathcal{J}$ , qui est unimodale, atteint son minimum global, pour une fonction poids spécifique  $\omega(t)$  et une valeur spécifique de  $\alpha$  qui peuvent être calculés numériquement à partir de la paramétrisation.

**4.2.2 Résultats numériques****4.2.2.1 Description du cas test**

Notre construction numérique utilise une géométrie classique en aérodynamique, celle d'un profil d'aile RAE2822, connu pour sa faible traînée en régime transsonique, documenté et tabulé dans le cadre du Projet Européen ECARP [1]. Les données correspondantes seront utilisées dans nos tests numériques.

Ce profil, comme on peut le constater dans la figure (4.3)(en haut), ne répond pas a nos exigences expliquées dans la première partie de ce chapitre à cause de la non-unimodalité de son intrados. Pour surmonter cette difficulté, il a été remplacé par un profil symétrisé, qui garde la même distribution de l'épaisseur le long de la corde (voir la figure (4.3)(en bas)).

Initialement, On paramétrise l'extrados du profil symétrisé par la paramétrisation de Bézier de degré  $n$  associée aux points de contrôles  $P_k = (x_k, y_k)$  ( $0 \leq k \leq n$ ). On utilise spécialement les données de la table (4.1) correspondant à un polygone de contrôle approprié de degré 16. Il a été obtenu par la procédure suivante : étant donnés deux paramètres  $x_2^{(4)}$  et  $x_3^{(4)}$  satisfaisant la condition ( $0 \leq x_2^{(4)} \leq x_3^{(4)} \leq 1$ ), le support  $X^{(4)}$  de la paramétrisation de Bézier de degré  $n = 4$  est construit en imposant les conditions aux limites comme suit :

$$X^{(4)} = \begin{pmatrix} 0 \\ 0 \\ x_2^{(4)} \\ x_3^{(4)} \\ 1 \end{pmatrix} \quad (4.22)$$

On applique ensuite 12 fois le processus d'élévation du degré, on obtient le vecteur  $X^{(16)}$  des abscisses des points de contrôle de la paramétrisation de Bézier de degré 16. Le vecteur  $Y^{(16)}$  des ordonnées des points de contrôle est tel que la courbe de Bézier correspondante soit une approximation de la courbe cible au sens des moindres carrés. Parmi toutes les paramétrisations  $(X^{(16)}, Y^{(16)})$ , on choisit le support  $X^{(16)}$  qui réalise la moindre variation totale

$$TV(Y^{(16)}) = \sum_{k=0}^{16-1} |y_{k+1}^{(16)} - y_k^{(16)}| \quad (4.23)$$

La courbe et le polygone de contrôle correspondant sont montrés dans la figure (4.3) (en bas).

$k$	$x_k$	$y_k$
0	0.00000000	0.00000000
1	0.00000000	0.00901285
2	0.01000000	0.01567691
3	0.02735714	0.02838362
4	0.05013187	0.02125740
5	0.07708791	0.04556686
6	0.10769231	0.04674385
7	0.14211538	0.01152551
8	0.18123077	0.11801427
9	0.22661538	-.01510481
10	0.28054945	0.11016884
11	0.34601648	0.03575430
12	0.42670330	0.09234295
13	0.52700000	0.04909957
14	0.65200000	0.05107346
15	0.80750000	0.01403431
16	1.00000000	0.00000000

TABLE 4.1 – Les coordonnées des points de contrôle pour la paramétrisation de degré 16 pour un profil d'aile symétrisé RAE2822

Un simple programme en langage Maple a été développé pour calculer la fonction de poids  $\omega(t)$  et l'exposant  $\alpha$  correspondant au profil d'aile symétrisé RAE2822 (voir figure (4.4)). En utilisant les résultats du théorème (4.2.1), on résout tout d'abord l'équation

$$y'(t_0) = 0 \quad (4.24)$$

ceci permet de calculer la constante  $C$  et l'exposant  $\alpha$  selon l'équation (4.19). Selon l'équation (4.20), on déduira la valeur de la fonction de poids  $\omega(t)$  en intégrant l'expression suivante

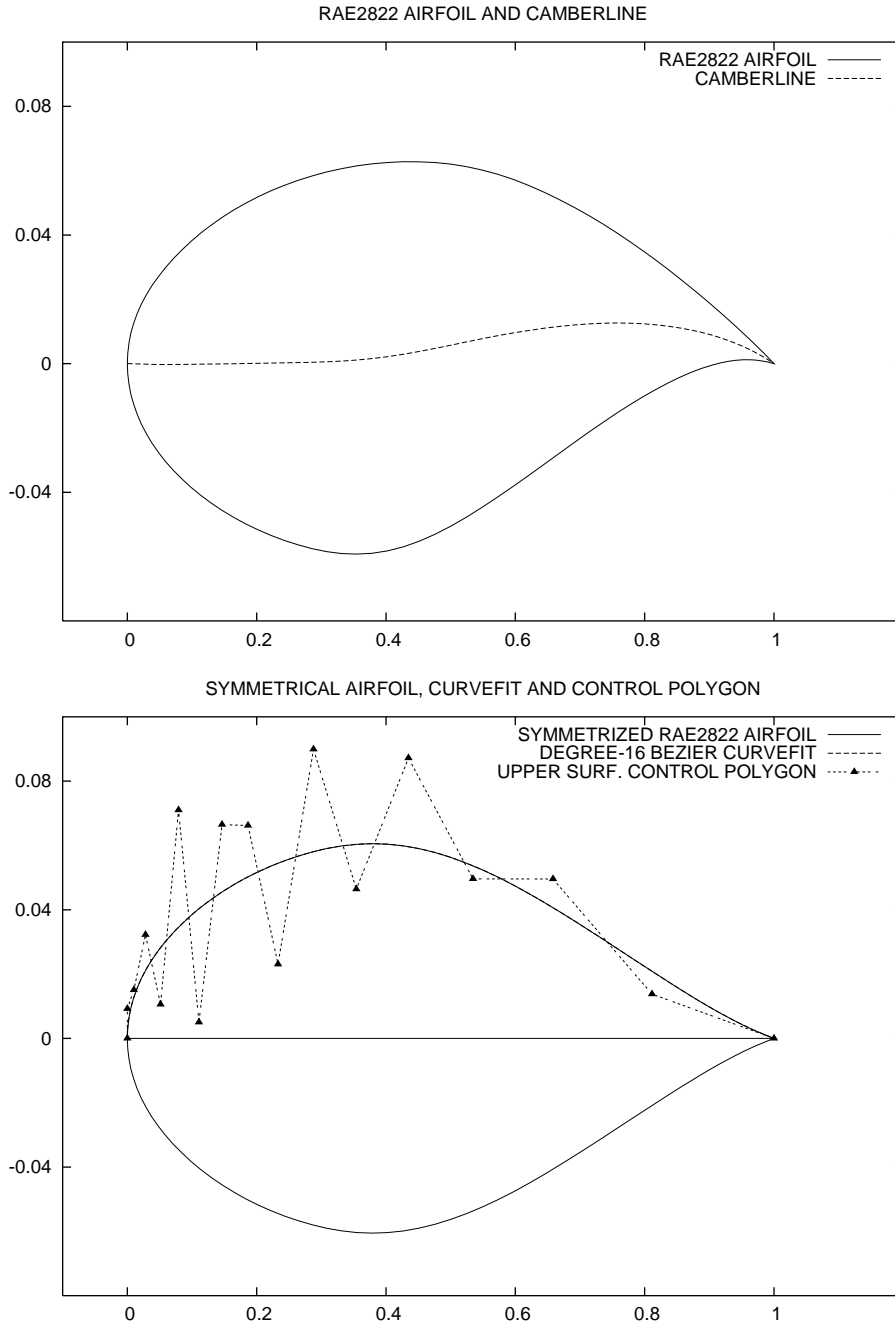


FIGURE 4.3 – RAE2822 (en haut) , Profil symétrisé avec le polygone de contrôle associé (en bas)

pour  $t \in [0, 1]$ .

$$\theta(t) = \left( \frac{x''(t)y'(t) - x'(t)y''(t)}{(x'^2(t) + y'^2(t))^{\frac{3}{2}}} - C \right) \left( \frac{x'(t) \sqrt{x'^2(t) + y'^2(t)}}{y'(t)} \right) \quad (4.25)$$

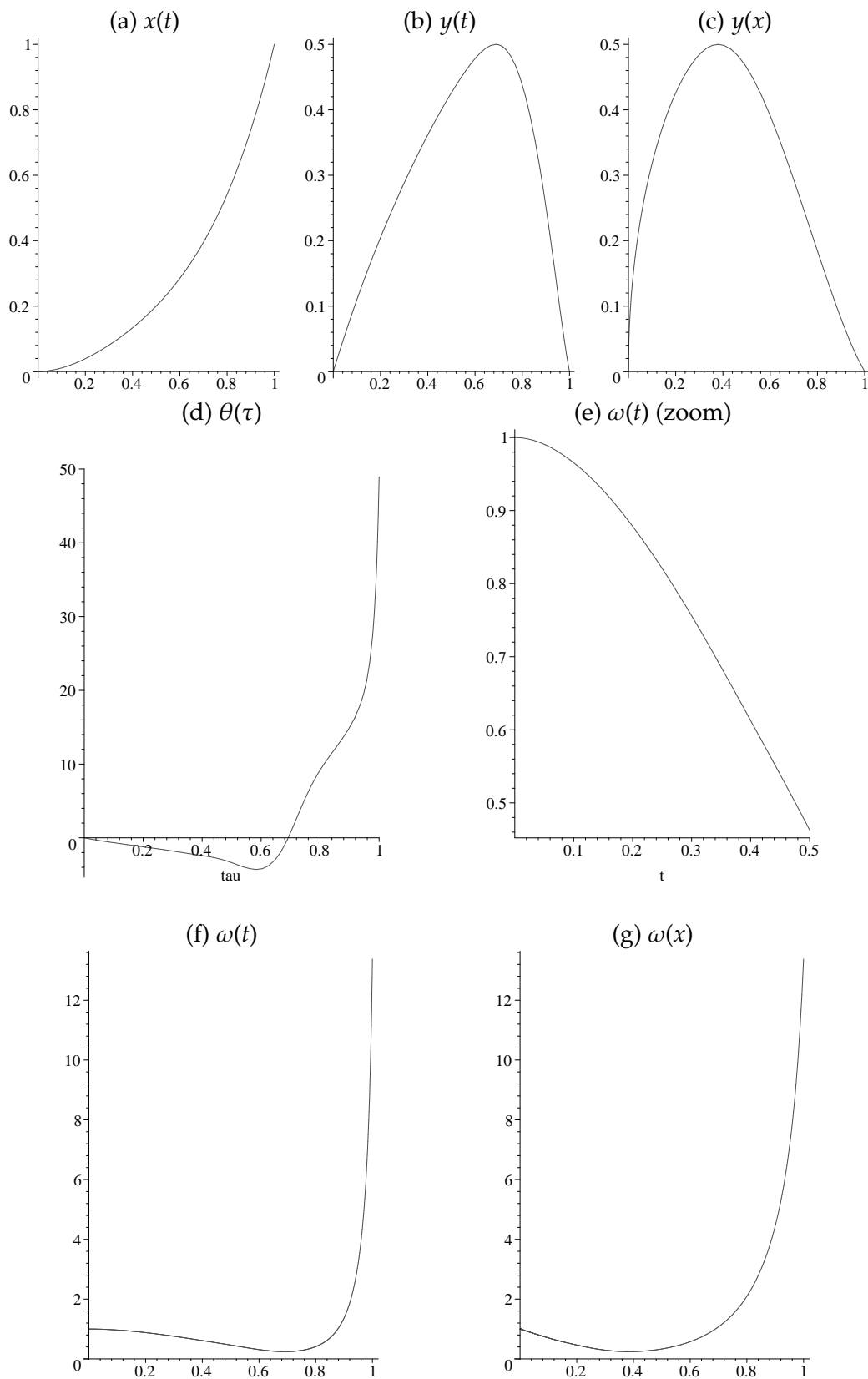


FIGURE 4.4 – Résultat de la visualisation (MAPLE) : Les données normalisées pour le RAE2822 symétrisée

Etant donnée la fonction  $\omega(t)$  (ou  $\omega(x)$ ) et l'exposant  $\alpha$ , calculés comme ci-dessus, correspondant à un support de paramétrisation fixé,  $\{x_k^{(n)}\}(k = 0, \dots, n)$ , notre procédure d'optimisation consiste à minimiser la fonctionnelle  $\mathcal{J}$ , définie par l'équation (4.5), par rapport aux paramètres de design  $\{y_k^{(n)}\}$ , en utilisant un code pour la programmation non-linéaire SCIP [122] de type gradient.

On note que la fonction  $\omega(t)$  (ou  $\omega(x)$ ) dépend de  $x(t)$ , et par conséquent du support  $\{x_k^{(n)}\}$ . Par suite, à chaque changement de support il faut recalculer la fonction  $\omega(t)$  qui lui correspond.

Soit  $\bar{\omega}(\tau)$  la fonction correspondant au support  $\{\bar{x}_k^{(n)}\}(k = 0, \dots, n)$ .

Pour calculer la fonction  $\omega(t)$  correspondant au support  $\{x_k^{(m)}\}(k = 0, \dots, m)$ , on résout l'équation suivante par la méthode de Newton :

$$\bar{x}(\tau) = x(t) \quad (4.26)$$

où

$$\bar{x}(\tau) = \sum_{k=0}^n B_n^k(\tau) \bar{x}_k \quad (4.27)$$

Enfin, on déduira par une interpolation linéaire les valeurs de  $\omega(t)$ .

#### 4.2.2.2 Résultats

On utilise le cas test précédent pour expérimenter, dans un contexte simplifié, nos algorithmes hiérarchiques d'optimisation de forme. Le but est d'évaluer le gain réalisé non seulement en convergence itérative mais aussi en précision par rapport à l'algorithme basique.

Dans une première expérience, on a réalisé plusieurs optimisations classiques en utilisant différents degrés de paramétrisation de Bézier. La figure (4.6) montre pour différentes valeurs du degré  $n$ , la valeur optimale obtenue par la fonctionnelle  $\mathcal{J}$  à convergence. Comme la dimension de l'espace de recherche de l'optimisation augmente, la valeur minimale de la fonctionnelle  $\mathcal{J}$  décroît, ce qui implique l'amélioration de l'efficacité du processus d'optimisation. Il apparaît en plus, pour ce problème simple, que le degré 10 est suffisant pour atteindre une bonne approximation de la forme optimale.

On examine ensuite l'influence du degré de la paramétrisation  $n$  sur la convergence itérative. On effectue trois optimisations séparément, associées aux degrés de paramétrisation  $n = 2, 5$  et  $10$ , en commençant par la même condition initiale. On montre dans la figure (4.5) les polygones de contrôle initiaux qui sont rigoureusement emboîtés en utilisant le processus d'élévation du degré. On peut voir clairement dans la figure (4.7) que la précision s'améliore avec l'augmentation du degré, mais la raideur numérique augmente aussi, ce qui nécessite plus d'itérations pour atteindre la convergence. Quant à la figure (4.8), elle indique les formes optimales correspondantes, avec en particulier la forme partiellement optimisée à l'itération 50 pour  $n = 10$ . À proximité de l'optimum, on constate que la variation de la forme est



plus substantielle que la fonctionnelle; on note en plus qu'à l'itération 50, la convergence incomplète dégrade visiblement la forme obtenue pour  $n = 10$ .

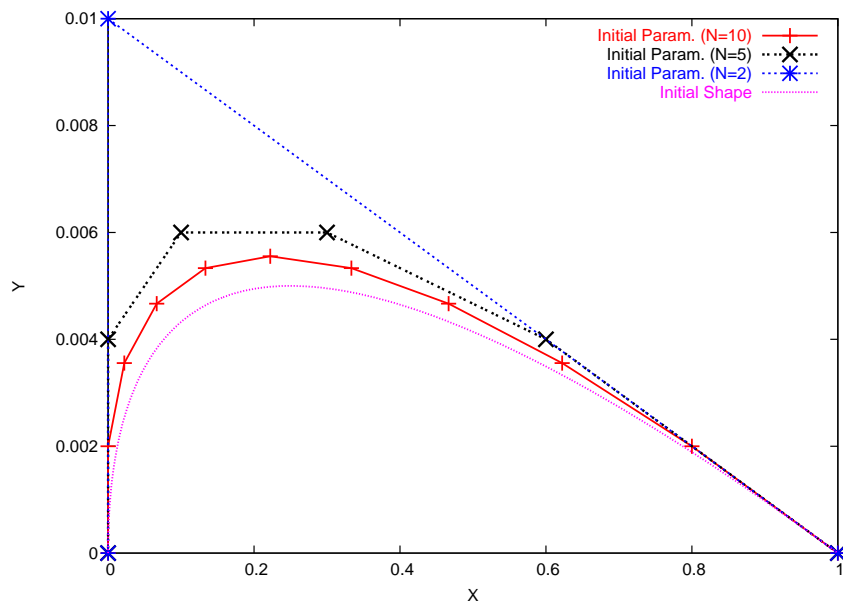


FIGURE 4.5 – Polygones de contrôles initiaux pour  $n = 2, 5, 10$

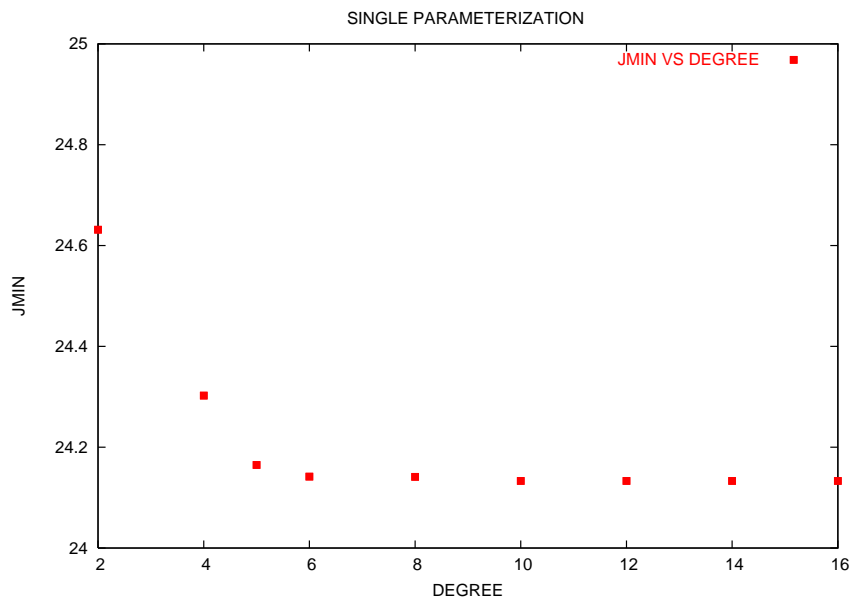


FIGURE 4.6 – Degré de paramétrisation de Bézier en fonction de la valeur minimale de  $\mathcal{J}$

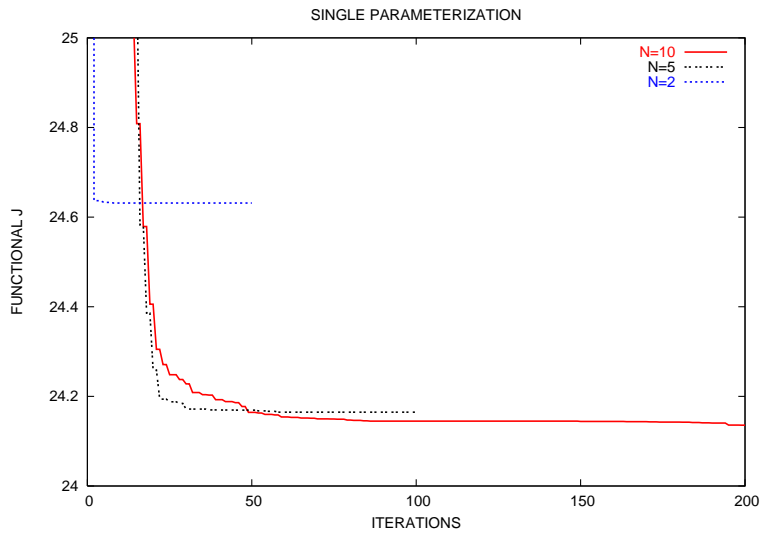


FIGURE 4.7 – Algorithme basique pour  $n = 2, 5$  et  $10$

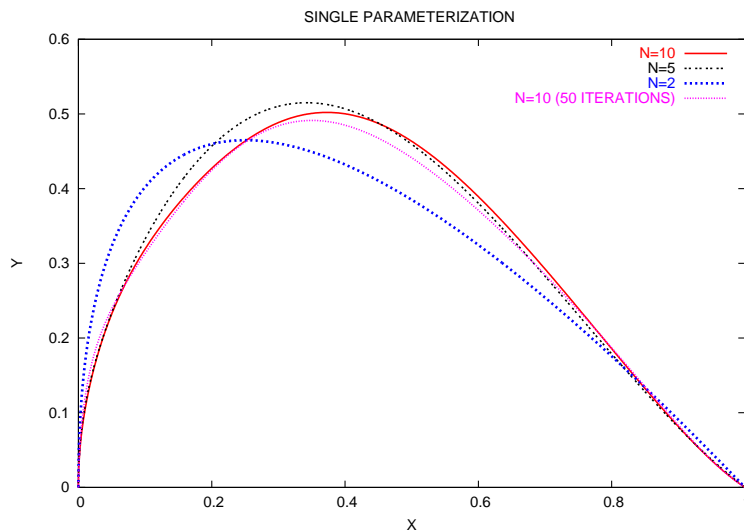


FIGURE 4.8 – Formes optimales obtenues pour  $n = 2, 5$  et  $10$

On présente maintenant les résultats obtenus avec notre approche hiérarchique. La figure (4.9) donne une comparaison entre les trois méthodes ci-dessous :

1. Méthode basique : pour une paramétrisation fine de degré  $n = 10$ , on effectue une optimisation jusqu'à convergence ;
2. Élévation du degré : on considère 3 niveaux de paramétrisation  $n = 2, 5$  et  $10$ , l'optimisation s'effectue par enrichissement progressif en utilisant le processus d'élévation du degré comme opérateur de transfert ;
3. FMOSA : en utilisant les mêmes niveaux de paramétrisation, on couple la méthode d'enrichissement progressif avec un cycle non-symétrique de paramétrisation en *dent de scie*.

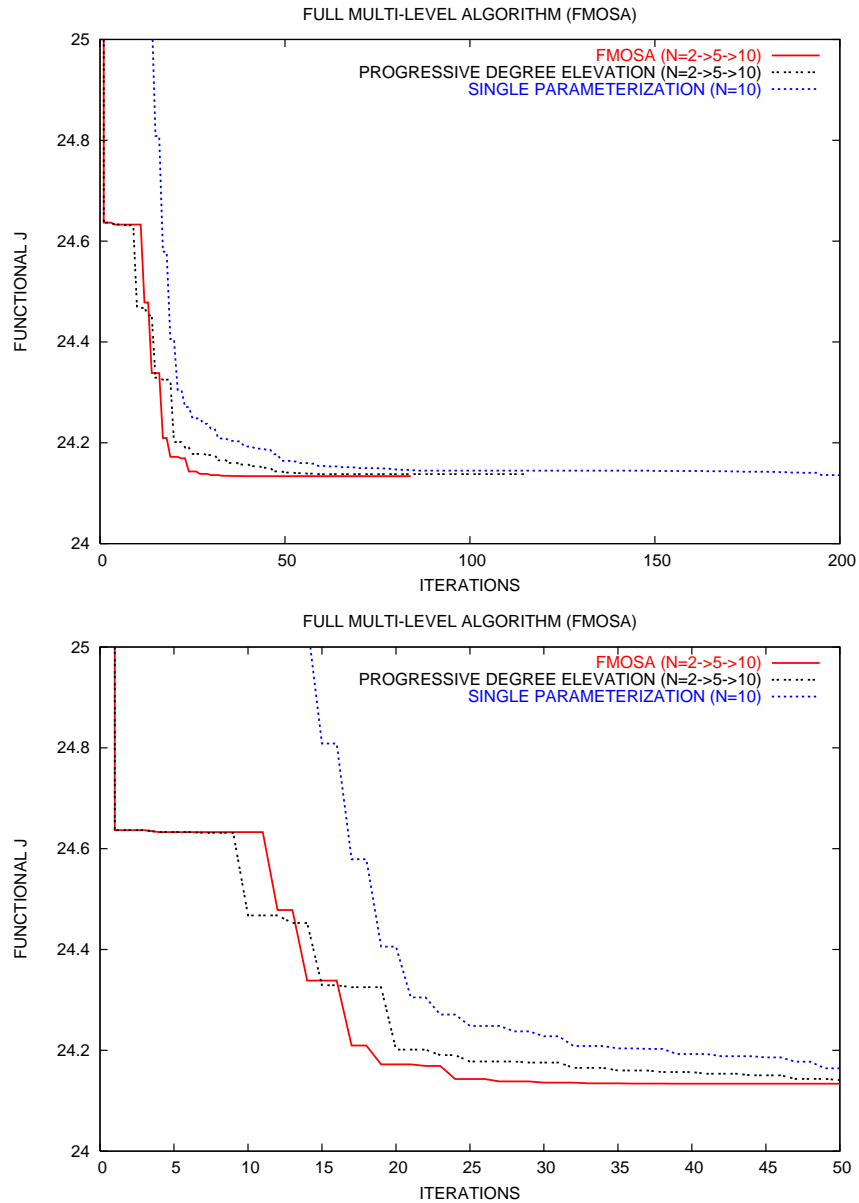


FIGURE 4.9 – Comparaison entre 3 stratégies d’optimisation : méthode basique, méthode avec enrichissement progressif et FMOSA

Avec les trois niveaux considérés, la méthode basée sur le processus d’élévation du degré est beaucoup plus rapide que la méthode basique. FMOSA est encore plus performante que les deux autres méthodes.

Enfin, comme nous l’avons signalé dans le chapitre précédent, la nature des transferts d’un niveau à un autre joue un rôle important dans notre approche multiniveau. Ceci est démontré par la figure (4.10) qui montre une comparaison entre deux algorithmes basés sur l’enrichissement progressif de paramétrisation en utilisant deux types d’opérateurs de transfert : l’un est basé sur le processus d’élévation du degré, présenté dans le chapitre précédent,

et l'autre sur une projection  $L^2$  (voir Annexe B). Ce dernier a dégradé considérablement la convergence de l'algorithme, à l'inverse de l'opérateur d'élévation du degré, puisque la paramétrisation n'est pas emboîtée par interpolation.

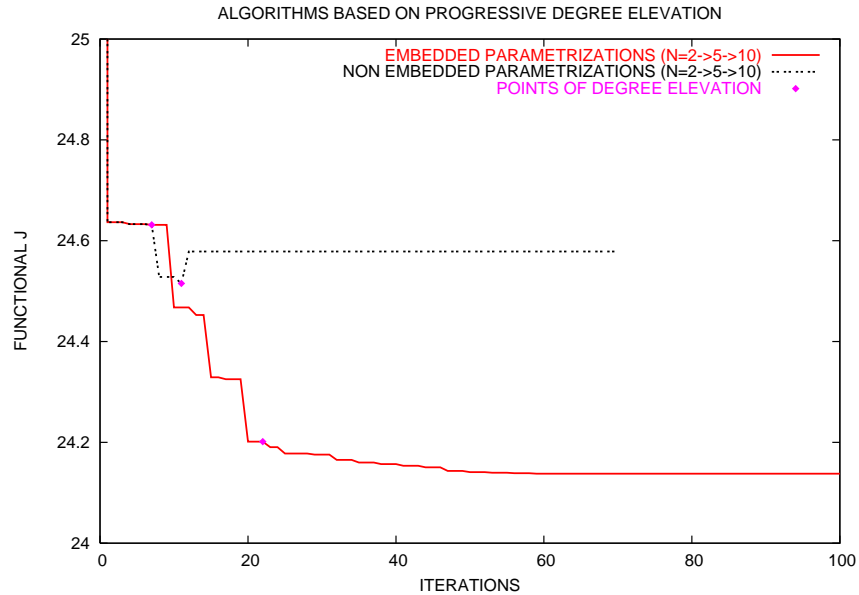


FIGURE 4.10 – L'algorithme d'enrichissement progressif pour deux types d'opérateurs de transfert

En conclusion, ces expériences ont montré le potentiel des algorithmes multiniveaux quant à l'amélioration significative de la convergence itérative et à la précision de la solution. Le choix du processus d'élévation du degré est strictement recommandé afin de garantir l'exactitude des transferts.

### 4.3 Application à l'optimisation de la voilure d'un avion d'affaires

#### 4.3.1 Description du cas test

Dans ce cas test, on optimise une voilure d'avion d'affaire en régime transsonique. La forme en plan est fournie par Piaggio Aero Industries (voir figure (4.11)). La géométrie 3D est composée par des sections correspondant au profil NACA0012 équidistantes le long de l'aile (voir figure (4.12)).

Notre solveur utilise un maillage non-structuré composé par 31124 noeuds et 173445 éléments, et qui est adapté à un écoulement transsonique avec un raffinement à l'emplacement du choc (voir figure (6.2)). L'écoulement est simulé en résolvant les équations d'Euler compressible avec une méthode de volumes finis. Les conditions d'écoulement à l'infini sont les suivantes :

Nombre de Mach	0.83
Angle d'incidence	2°

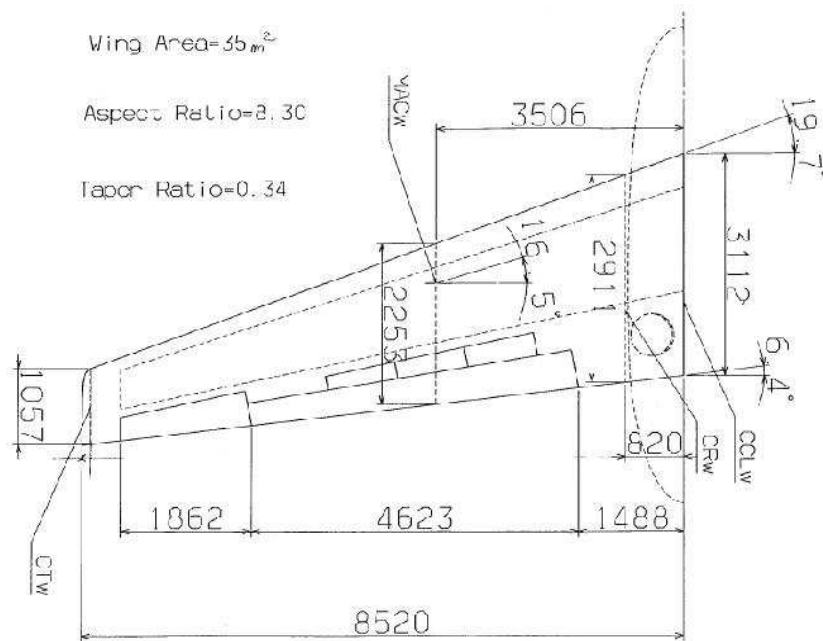


FIGURE 4.11 – Forme en plan fournit par Piaggio Aero Industries

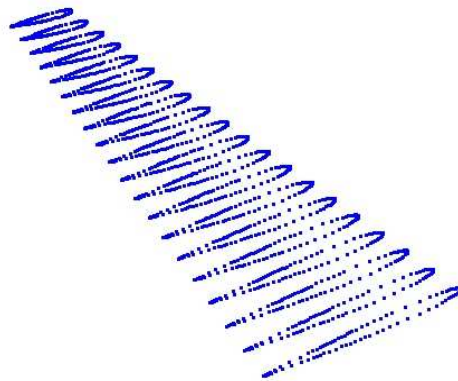


FIGURE 4.12 – Construction de la géométrie 3D

La procédure d'optimisation consiste à réduire le coefficient de la traînée  $C_D$  avec une contrainte sur la portance  $C_L$  qui ne doit pas diminuer de 0.1%. Cette contrainte est imposée par une pénalisation.

$$J_A = \frac{C_D}{C_{D_0}} + 10^4 \cdot \max\left(0, 0.999 - \frac{C_L}{C_{L_0}}\right) \quad (4.28)$$

où  $C_{D_0}$  et  $C_{L_0}$  sont respectivement la traînée et la portance de la configuration initiale.

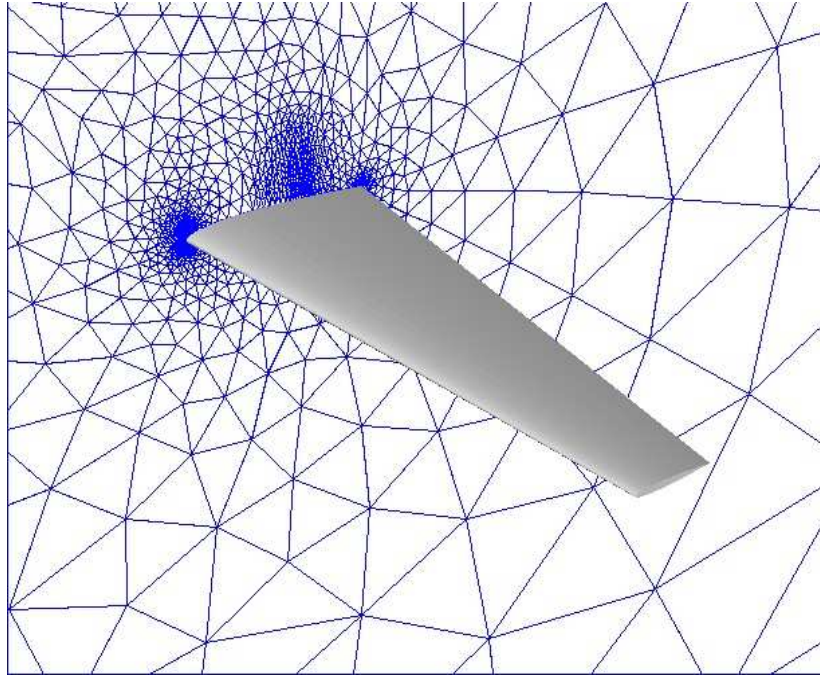


FIGURE 4.13 – Forme initiale et maillage dans le plan de symétrie

On considère au départ une boîte FFD construite à partir du volume de contrôle correspondant à la forme initiale. On note par  $n_i \times n_j \times n_k$  le degré du produit tensoriel de Bézier, qui correspond à  $n_i + 1$  (resp.  $n_j + 1$  et  $n_k + 1$ ) points de contrôle dans la direction  $x$  (resp.  $y$  et  $z$ ) (voir figure (4.14)). On fixe, durant le processus d'optimisation, le bord de fuite et le bord d'attaque, ce qui revient à fixer les sommets de la boîte englobante ( $i = 0$  et  $i = n_i$ ). On bouge seulement les points de contrôle dans la direction verticale.

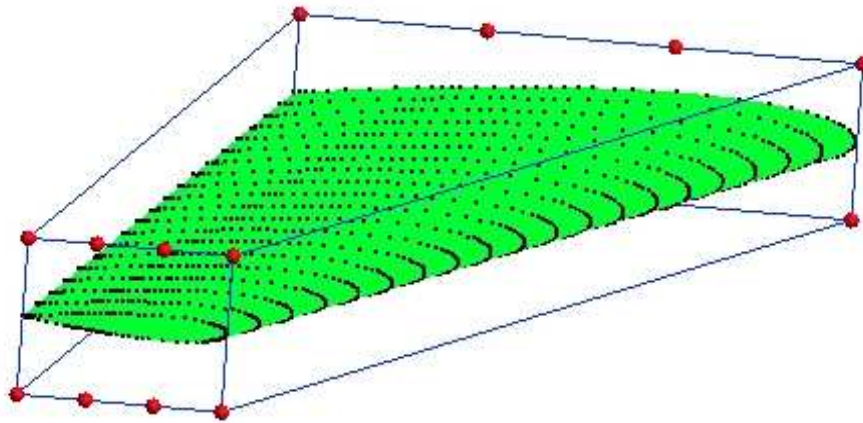


FIGURE 4.14 – Boîte FFD associée à la représentation de Bézier de degré  $3 \times 1 \times 1$

On considère dans notre cas test trois types de paramétrisation : grossière de degré  $3 \times 1 \times 1$  qui correspond à 8 paramètres d'optimisation, moyenne de degré  $6 \times 1 \times 1$  qui correspond à 20 paramètres d'optimisation, et fine de degré  $9 \times 1 \times 1$  qui correspond à 32 paramètres d'optimisation. La méthode du simplexe de Nelder-Mead a été utilisée comme optimiseur dans tous les cas tests numériques.

### 4.3.2 Résultats numériques

Pour tester l'influence du degré de la paramétrisation sur la convergence et aussi sur la précision de la solution, on a effectué des tests d'optimisation avec la méthode basique pour trois types de paramétrisation : grossière, moyenne et fine. On montre dans la figure (4.15) une comparaison entre les courbes de convergence obtenues par ces trois optimisations. On constate que la convergence itérative est plus lente mais le résultat plus précis avec l'augmentation du degré de paramétrisation. On atteint, avec une paramétrisation fine, une solution plus précise mais au prix d'un nombre important d'itérations.

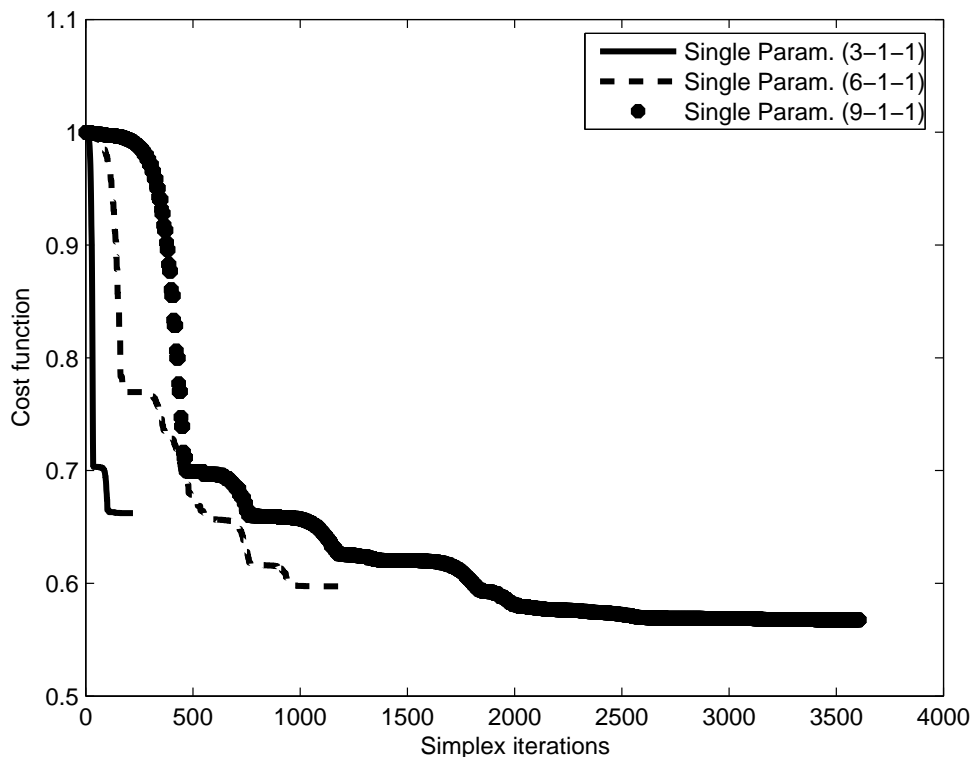


FIGURE 4.15 – Méthode basique associée à trois types de paramétrisation : grossière, moyenne et fine

On résume dans la table (4.2) les valeurs des coefficients aérodynamiques et de la fonction objectif associés à la solution optimale obtenue par trois stratégies d'optimisation. On constate que la contrainte sur la portance a été respectée et que la réduction de la traînée est d'autant plus importante que le degré de paramétrisation augmente. Ceci est illustré par les

figures (4.16)-(4.19) et la figure (4.20) qui montrent respectivement le champ de Mach sur la surface de l'aile et le coefficient de pression associé à trois sections de l'aile (l'emplanture, 50% de l'envergure et l'extrémité) pour la configuration initiale et les trois stratégies d'optimisation. On constate que la réduction du choc sur l'extrados de l'aile optimisée augmente avec la finesse de la paramétrisation. Cette dernière a aussi un impact sur les formes optimales comme on peut le voir dans la figure (4.21).

	$C_L$	$C_D$	Fonction coût
Config. initiale	0.319192893	0.026352608	1.
$3 \times 1 \times 1$	0.318874966	0.017450289	0.662184501
$6 \times 1 \times 1$	0.318873704	0.015741397	0.597337341
$9 \times 1 \times 1$	0.318876190	0.014955670	0.567521127

TABLE 4.2 – Comparaison entre les valeurs optimales des coefficients aérodynamiques et de la fonction objectif obtenus avec la méthode basique pour différents degrés de paramétrisation

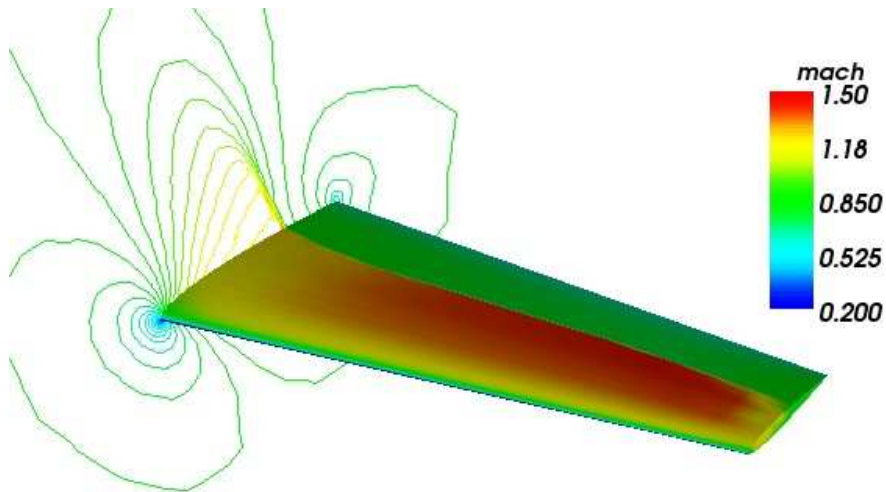


FIGURE 4.16 – Champs et isovaleurs de nombre de Mach : configuration initiale



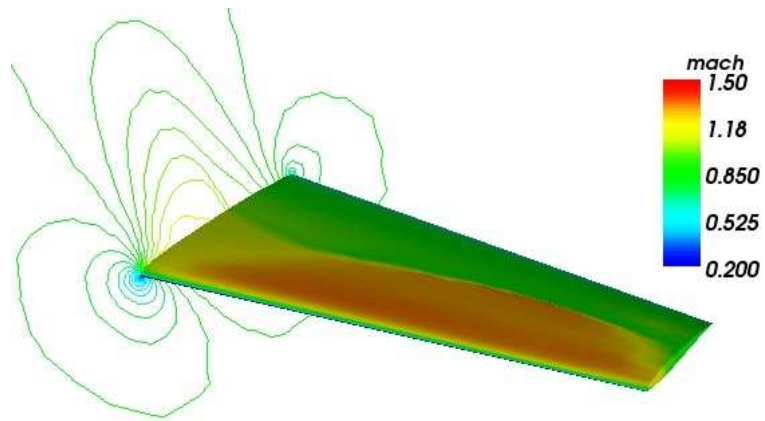


FIGURE 4.17 – Champs et isovalues de nombre de Mach : Paramétrisation grossière ( $3 \times 1 \times 1$ )

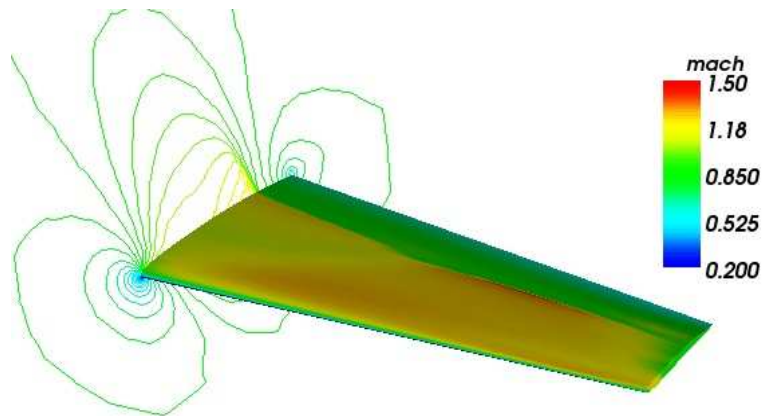


FIGURE 4.18 – Champs et isovalues de nombre de Mach : Paramétrisation moyenne ( $6 \times 1 \times 1$ )

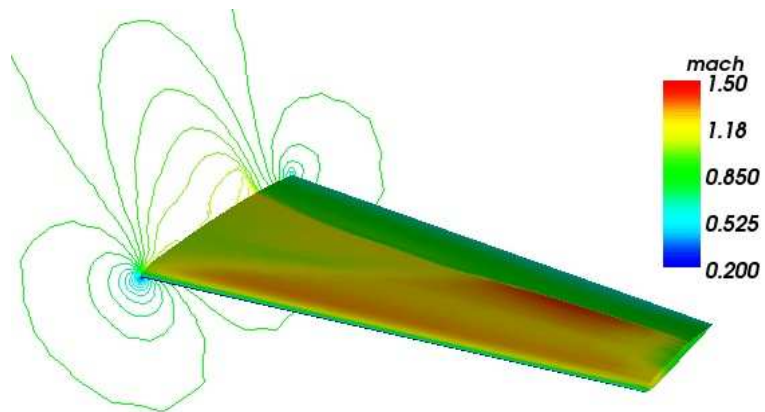


FIGURE 4.19 – Champs et isovalues de nombre de Mach : Paramétrisation fine ( $9 \times 1 \times 1$ )

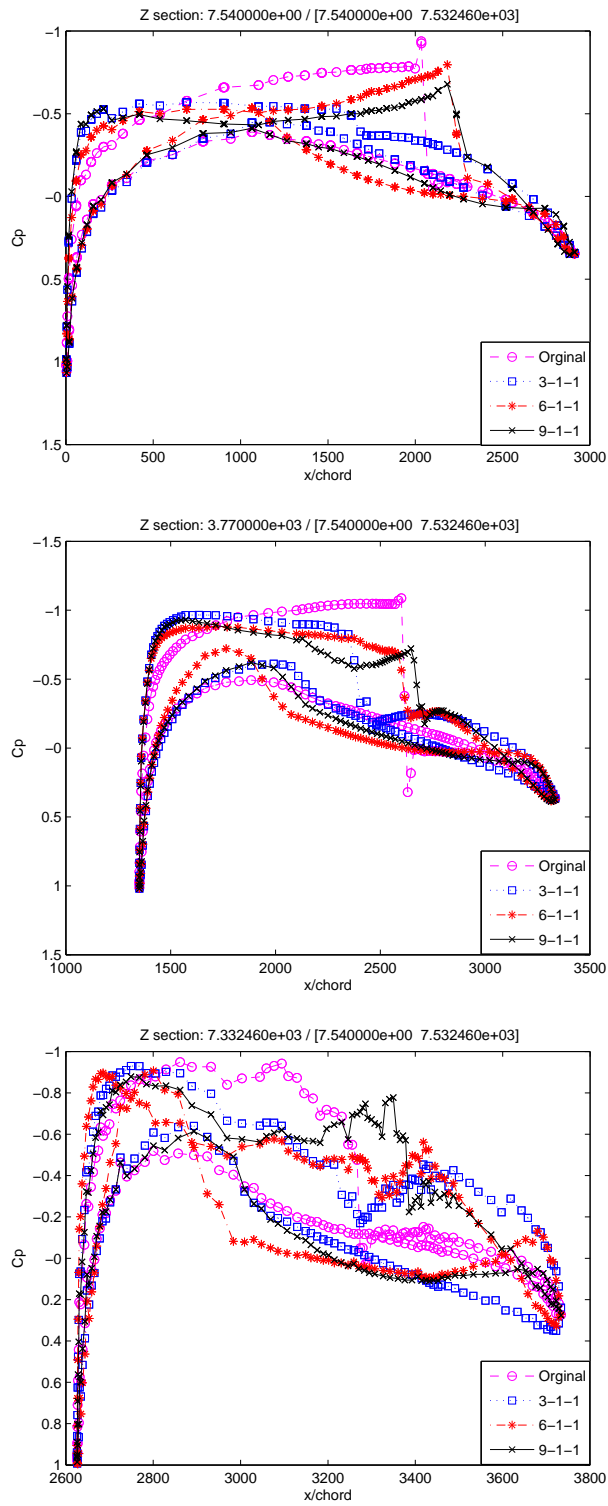


FIGURE 4.20 – Coefficient de pression associé aux trois sections de l’aile : l’emplanture (haut), 50% de l’envergure (milieu) et l’extrémité (bas)

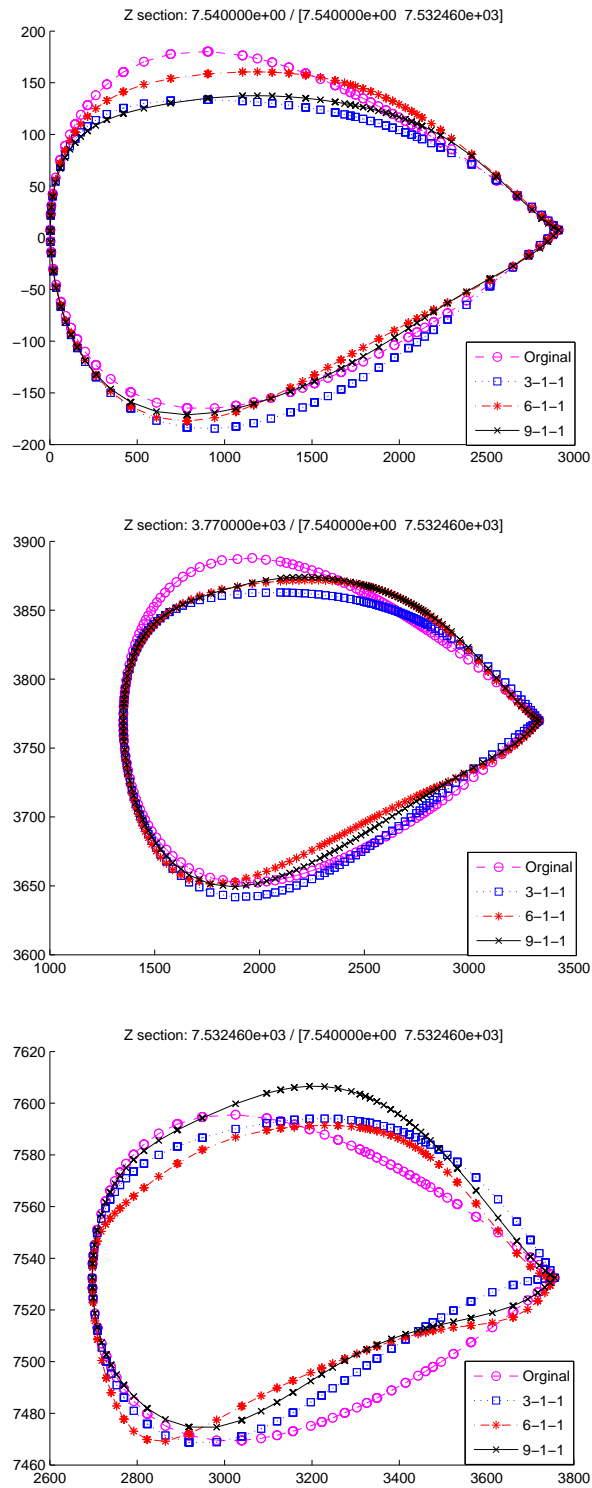


FIGURE 4.21 – Forme des sections de l’aile à l’emplanture (haut), à 50% de l’envergure (milieu) et à l’extrémité (bas)

Le but du cas test précédent est de clarifier la dépendance entre la finesse de la paramétrisation, la précision de la solution et la vitesse de convergence. On a démontré que la vitesse de convergence de la méthode basique associée à une paramétrisation grossière est plus rapide mais la solution optimale n’est pas satisfaisante. Ce qui est tout à fait l’inverse en utilisant une paramétrisation fine. A partir de ce constat, il paraît judicieux d’inclure dans la boucle d’optimisation différents niveaux de paramétrisation afin de combiner à la fois la rapidité de convergence et aussi la précision de la solution optimale. C’est d’ailleurs l’objet du cas test suivant qui consiste à tester nos stratégies multiniveaux décrites à la section précédente. Pour cela on compare trois stratégies d’optimisation :

1. Stratégie A : Méthode basique associée à la paramétrisation fine de degré  $9 \times 1 \times 1$ .
2. Stratégie B : Enrichissement progressif de paramétrisation associé à trois niveaux de paramétrisation ( $3 \times 1 \times 1$ ,  $6 \times 1 \times 1$  et  $9 \times 1 \times 1$ ).
3. Stratégie C : Algorithme FMOSA, associée aux mêmes niveaux de paramétrisation, qui combine la technique d’enrichissement progressif avec le V-cycle symétrique de paramétrisation (voir la table (4.3)).

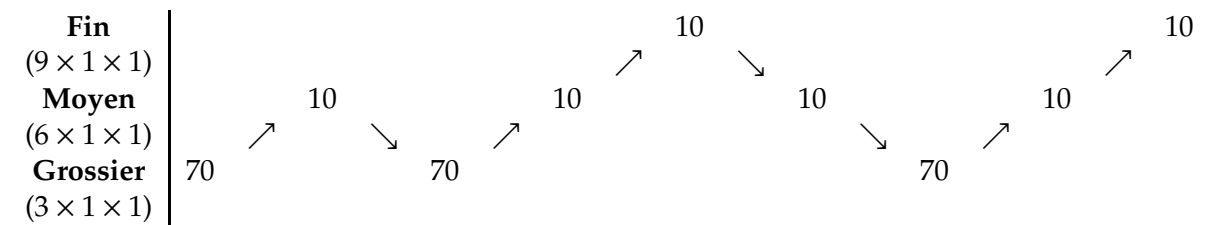


TABLE 4.3 – Algorithme FMOSA ; les entiers désignent le nombre d’itérations d’optimisation ; ↗ indique élévation du degré + relaxation ; ↘ correspond à la reformulation du problème de correction au niveau grossier.

La figure (4.22) montre les courbes de convergence obtenues par les stratégies A, B et C. On constate que les stratégies multiniveaux ont une convergence rapide et atteignent une meilleure performance en comparaison avec la méthode basique associée à une paramétrisation fine. La dernière méthode (FMOSA) est davantage efficace en terme de convergence et de précision, réalisant une performance meilleure avec un nombre faible d’itérations.

On résume dans la table (4.3.2) les valeurs des coefficients aérodynamiques et de la fonction objectif associés à la solution optimale obtenue par chacune des trois stratégies A, B et C. On constate une réduction du coefficient de traînée avec une portance presque constante pour toutes les stratégies d’optimisation avec un écart considérable en faveur des stratégies multiniveaux. Ceci se traduit par une réduction de l’intensité du choc sur l’extrados de l’aile comme on peut le voir dans les figures (4.23)-(4.26), qui représentent les champs et les isovaleurs de nombre de Mach respectivement sur la surface et à l’emplanture de l’aile obtenus avec les trois stratégies. On constate que la réduction du choc n’est pas importante en utilisant la méthode de base associée à une paramétrisation fine après 700 itérations. Par contre, avec les stratégies multiniveaux l’intensité du choc a été atténuée et la zone supersonique a été réduite sur l’extrados de l’aile. Ceci est confirmé aussi par la figure (4.27) qui montre les coefficients de pression associés aux trois sections de l’aile (l’emplanture, 50% de l’envergure et l’extrémité) pour les mêmes stratégies. Quant à la figure (4.28), elle montre les

formes de trois sections de l'aile (l'emplanture, 50% de l'envergure et l'extrémité) associées aux stratégies A, B et C.

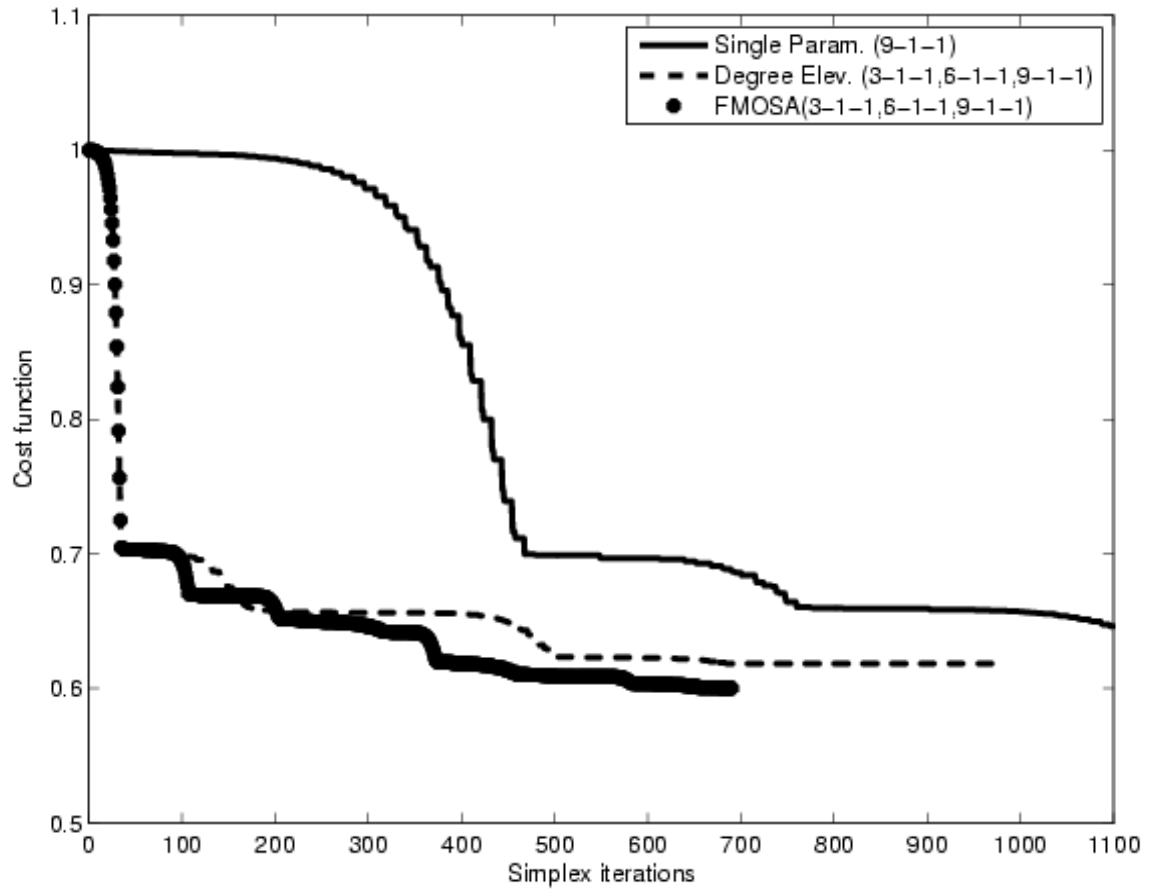


FIGURE 4.22 – Comparaison entre les courbes de convergence associées aux stratégies A, B et C

	$C_L$	$C_D$	Fonction coût
Config. initiale	0.319192893	0.026352608	1.
Test A	0.324961051	0.018084091	0.6862353246
Test B	0.318911102	0.016295193	0.6183521866
Test C	0.318900321	0.015818799	0.6002755598

TABLE 4.4 – Comparaison entre les valeurs optimaux des coefficients aérodynamiques et de la fonction objective obtenues par les stratégies A, B, et C après 700 itérations

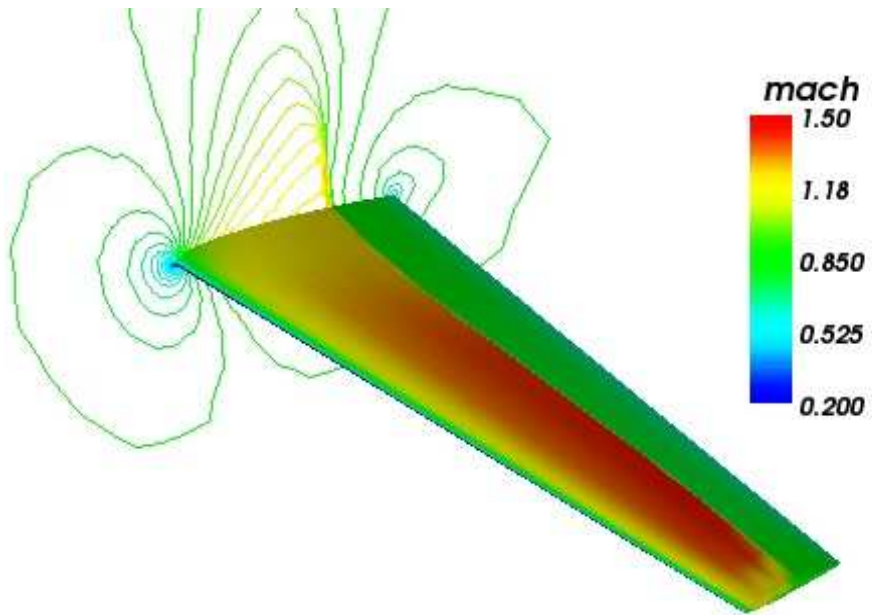


FIGURE 4.23 – Champs et isovalues de nombre de Mach : configuration initiale

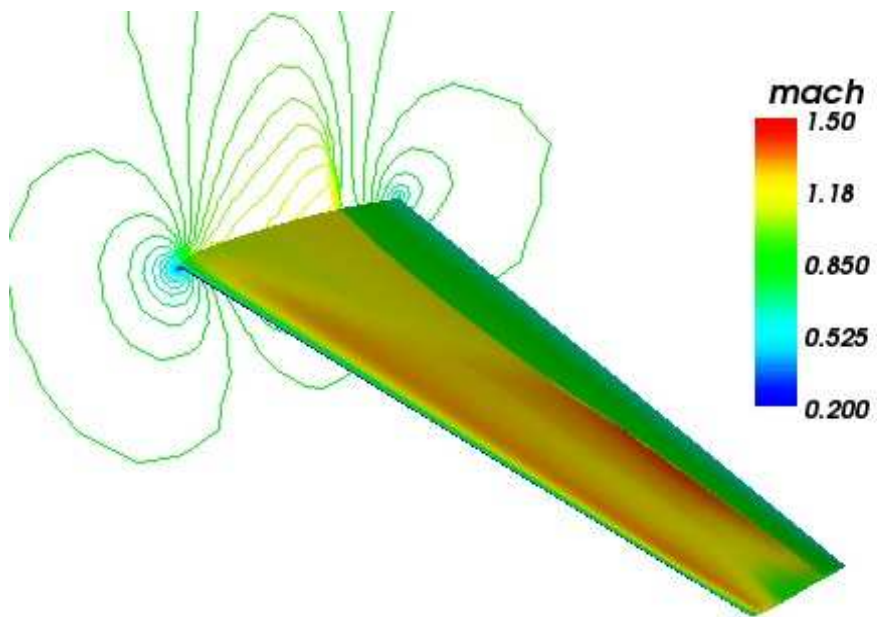


FIGURE 4.24 – Champs et isovalues de nombre de Mach : test A

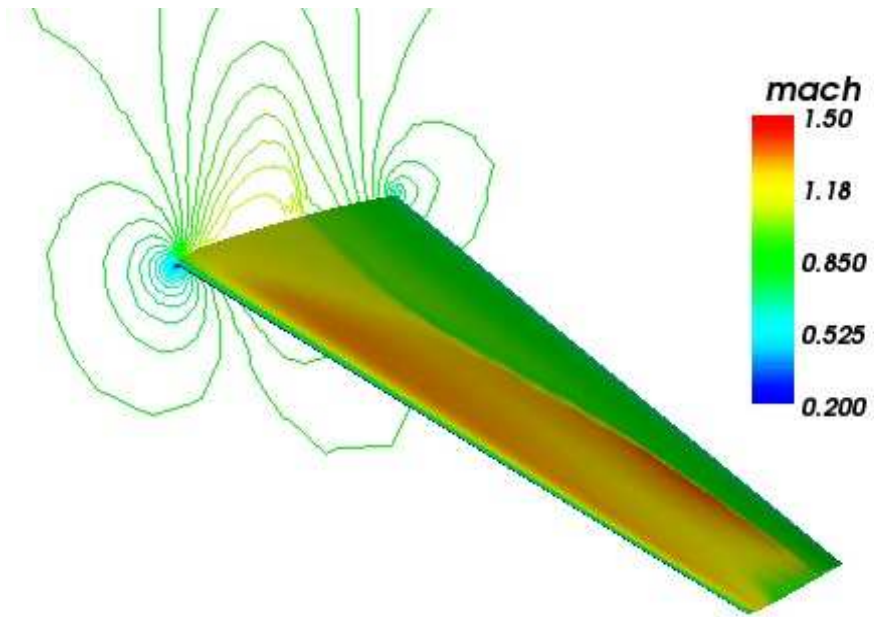


FIGURE 4.25 – Champs et isovaleurs de nombre de Mach : test B

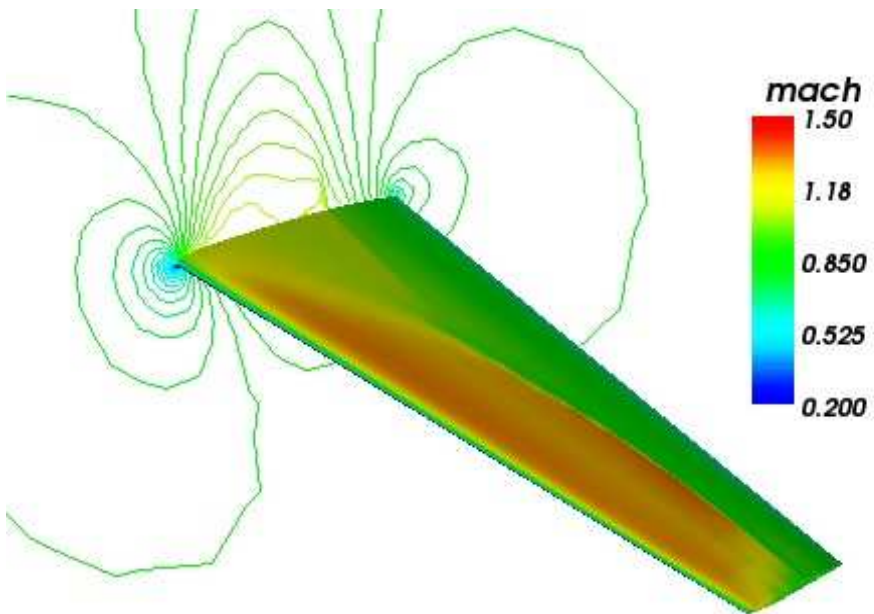


FIGURE 4.26 – Champs et isovaleurs de nombre de Mach : test C

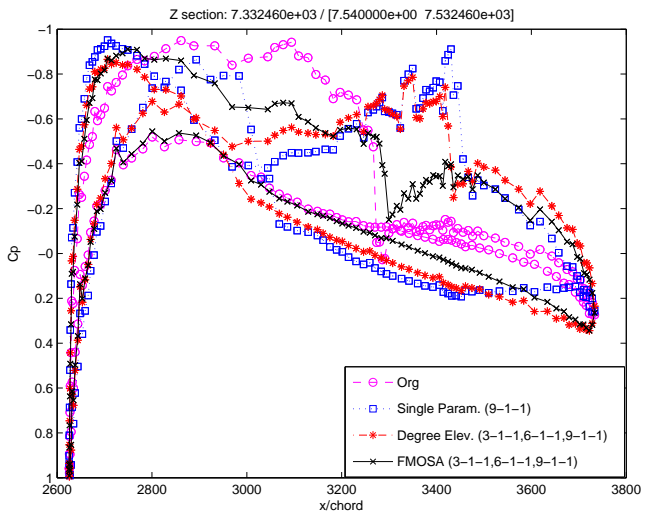
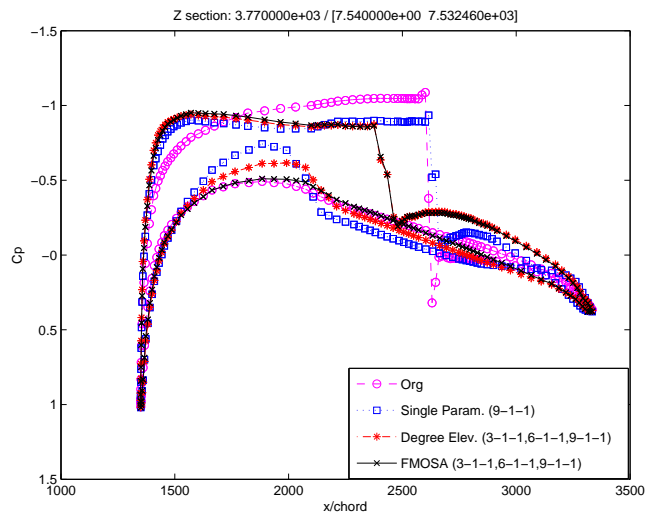
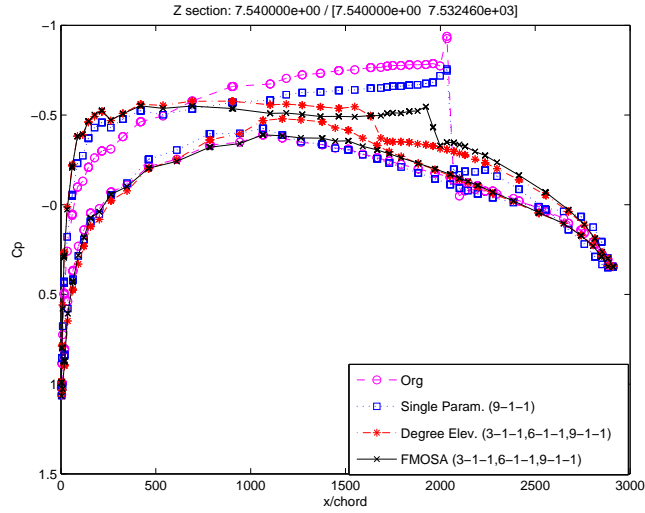


FIGURE 4.27 – Coefficient de pression associé aux trois sections de l’aile : l’emplanture (haut), 50% de l’envergure (milieu) et l’extrémité (bas)



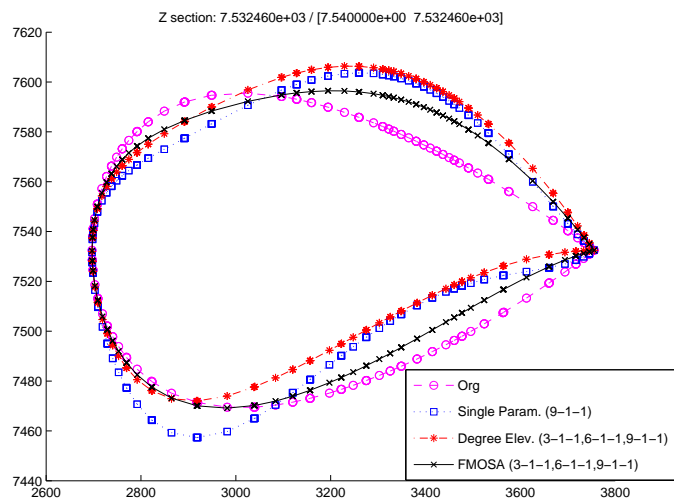
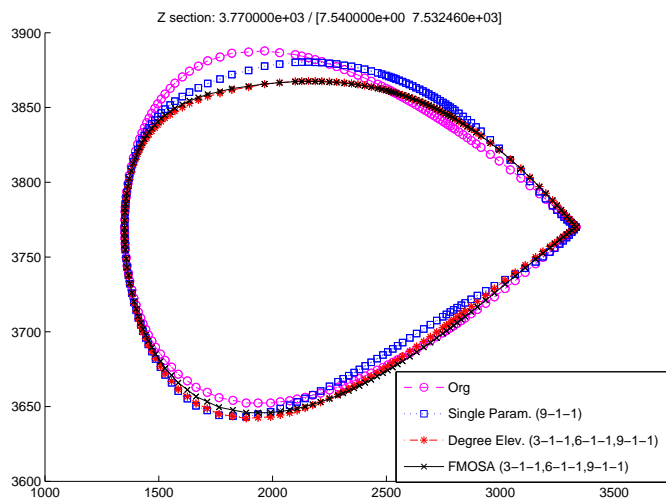
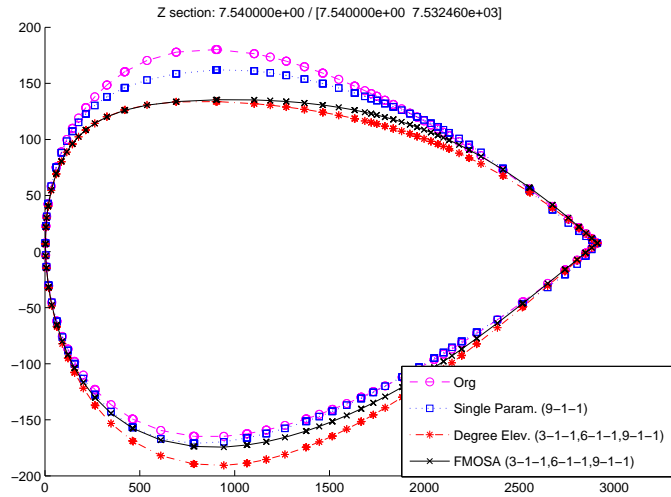


FIGURE 4.28 – Forme des sections de l’aile à l’emplacement (haut), 50% de l’envergure (milieu) et à l’extrémité (bas)

## 4.4 Optimisation appliquée à un jet d'affaires supersonique

Contrairement au régime transsonique où on optimise la forme vis-à-vis des critères et contraintes aérodynamiques et géométriques, l'optimisation de forme pour des avions civils supersoniques inclut un critère supplémentaire qui concerne la minimisation du bang sonique généré par l'avion (voir figure 4.29). Le bang sonique représente le son entendu au sol dû à une brusque variation de pression (ou onde de choc), il est généré lors du déplacement de l'avion dans l'atmosphère à une vitesse supersonique.

### 4.4.1 Description du cas test

On cherche à tester notre approche hiérarchique en optimisation de forme en régime supersonique. Ce cas test consiste à améliorer, dans deux cas différents, l'aile et le nez d'un jet d'affaires supersonique fourni par Dassault aviation en réduisant une fonctionnelle liée au bang sonique tout en préservant les performances aérodynamiques. Les conditions à l'infini sont les suivantes :

Nombre de Mach	1.8
Angle d'incidence	1°

Dans cette étude, on ne simule pas le bang sonique proprement dit, car le problème nécessite un couplage avec les équations de propagation pour obtenir la signature de pression au sol. On reprend l'idée proposée dans [115][32] qui consiste à quantifier l'intensité des nuisances du bang sonique par l'intégrale, sur le volume  $\Omega_{SB}$ , de la norme du gradient de la pression au carré. Donc en réduisant la forme des chocs dans  $\Omega_{SB}$ , la signature de pression au sol se réduit aussi.

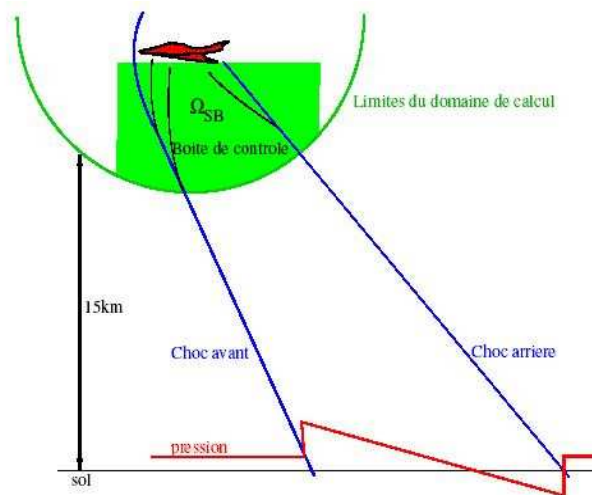


FIGURE 4.29 – Bang sonique au-dessous de l'avion

La géométrie considérée à été optimisée en minimisant la fonctionnelle coût suivante :

$$J = J_{SB} + J_{C_L} + J_{C_D} + J_{Vol} \quad (4.29)$$

qui consiste à minimiser la nuisance du bang sonique :

$$J_{SB} = \frac{\int_{\Omega_{SB}} (\nabla p)^2 d\Omega}{\int_{\Omega_{SB}} (\nabla p^0)^2 d\Omega} \quad (4.30)$$

avec des contraintes aérodynamiques traitées par pénalisation, on n'autorise que 0.1% de réduction de la portance et d'augmentation de la traînée.

$$J_{C_L} = \begin{cases} 10^4 \cdot |\Delta C_L| & \text{si } \Delta C_L < -10^{-3} \\ 0 & \text{si } \Delta C_L \geq -10^{-3} \end{cases} \quad (4.31)$$

$$J_{C_D} = \begin{cases} 0 & \text{si } \Delta C_D \leq -10^{-3} \\ 10^4 \cdot |\Delta C_D| & \text{si } \Delta C_D > -10^{-3} \end{cases} \quad (4.32)$$

et des contraintes géométriques sur le changement de volume introduites par pénalisation :

$$J_{Vol} = 10^4 \cdot \max(-\Delta Vol, 0) \quad (4.33)$$

avec  $\Delta X = (X - X^0)/X^0$  pour  $X = C_L, C_D$  ou  $Vol$ ,  $X^0$  étant la valeur de  $X$  pour la configuration initiale.

On résout les équations d'Euler compressible par la méthode de volume finis en s'appuyant sur un maillage non-structuré. Pour le nez, on utilise un maillage de 173526 noeuds et 981822 éléments (voir figure (4.30)), et pour la voilure, un maillage de 37375 noeuds et 184249 éléments (voir figure (4.31)).

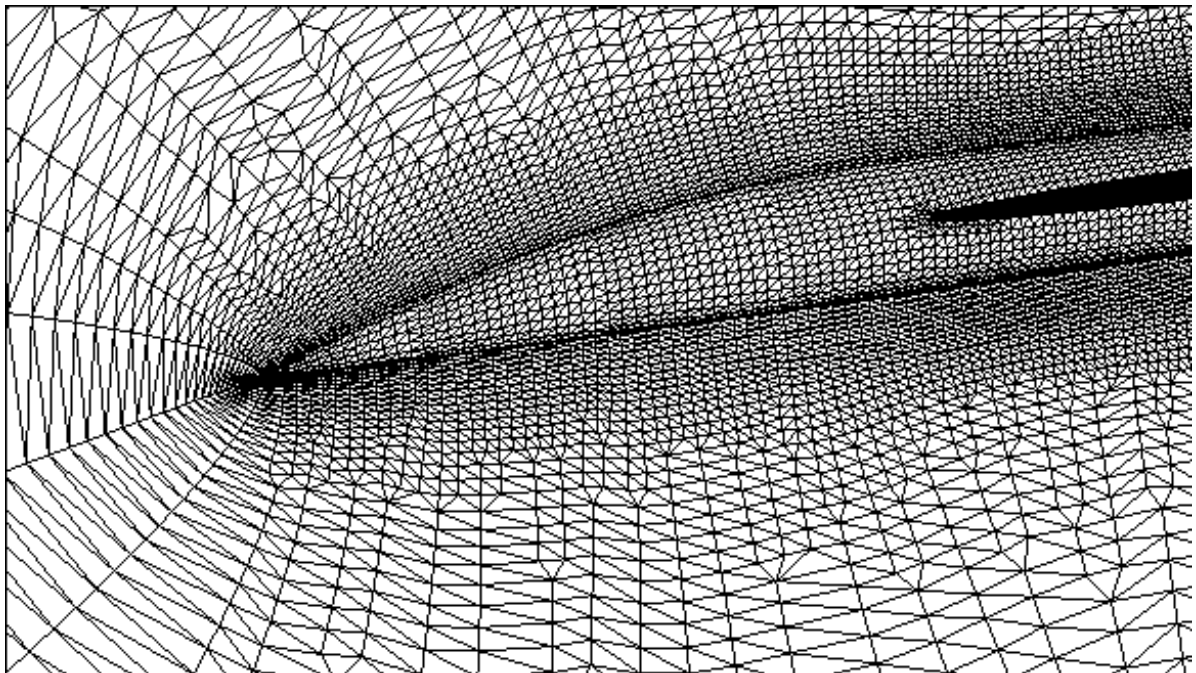


FIGURE 4.30 – Maillage initial dans le plan de symétrie : Optimisation du nez

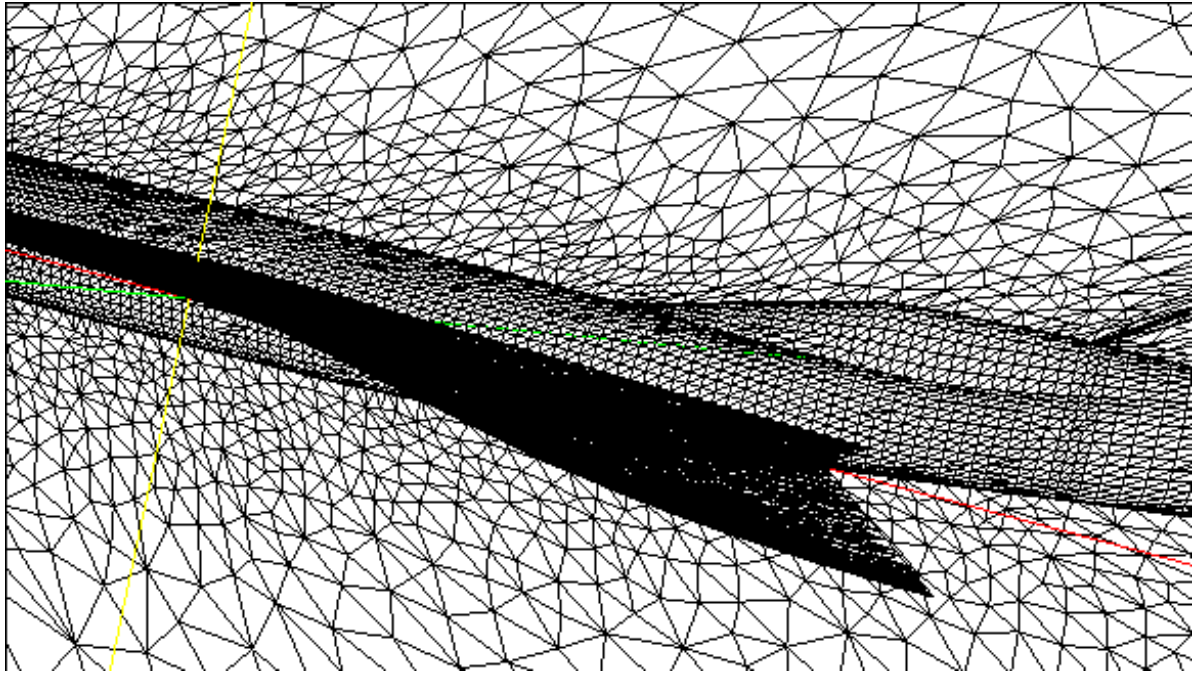


FIGURE 4.31 – Maillage initial dans le plan de symétrie : Optimisation de la voilure

## 4.4.2 Résultats numériques

### 4.4.2.1 Optimisation de la voilure

On paramétrise l'aile du jet d'affaire en utilisant la méthode de Free-Form Deformation. On considère une boîte englobante, contrairement au cas test précédent, qui contient des points de contrôle répartis autour de l'aile comme dans la figure (4.32)). On bouge les points de contrôle dans la position verticale à l'exception des points désignés dans la figure suivante par les symboles  $\bullet$ , qu'on maintient fixe au cours du processus d'optimisation. On choisit trois types de paramétrisations de degrés  $3 \times 1 \times 2$  (grossière),  $3 \times 1 \times 4$  (moyenne) et  $6 \times 1 \times 4$  (fine).

Le but de cette optimisation de forme est de tester trois différentes stratégies, comme le cas test précédent, qu'on note également par A, B et C. On schématise dans la table (4.5) les différents niveaux utilisés en indiquant le nombre d'itérations pour l'algorithme FMOSA.

<b>Fin</b> ( $6 \times 1 \times 4$ )				15					15
<b>Moyen</b> ( $3 \times 1 \times 4$ )			15		15		15		15
<b>Grossier</b> ( $3 \times 1 \times 2$ )	70		70		70		70		70

TABLE 4.5 – Algorithme FMOSA ; les entiers désignent le nombre d'itérations d'optimisation ;  $\nearrow$  indique élévation du degré + relaxation ;  $\searrow$  correspond à la reformulation du problème de correction au niveau grossier.

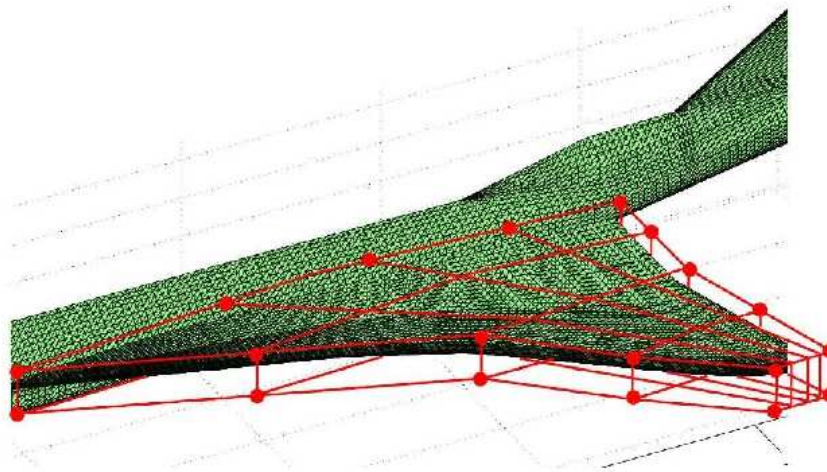


FIGURE 4.32 – La boîte FFD englobant la forme pour une paramétrisation de degré  $4 \times 1 \times 4$ ; les symboles • désignent les points de contrôle qui sont fixe; les autres points bougent verticalement (de [33]).

La figure (4.33) montre une comparaison entre les courbes de convergence obtenues par les trois stratégies. On constate que les deux stratégies multiniveaux B et C sont plus rapides et plus précises que la méthode de base associée à la paramétrisation fine. La dernière stratégie (FMOSA), qui utilise une correction au niveau grossier, est significativement meilleure que la stratégie hiérarchique B qui utilise un simple enrichissement de la paramétrisation.

On résume dans la table (4.6) les valeurs des coefficients aérodynamiques, du bang sonique et de la fonction objectif associés à la solution optimale obtenue par chacune des trois stratégies d'optimisation. On constate que les contraintes sur la portance et sur la traînée ont été respectées et le critère de bang sonique a été réduit. En particulier la méthode FMOSA réalise un écart important en bang sonique par rapport aux autres méthodes. Ceci est confirmé par les figures ((4.34)-(4.36)) qui montrent la distribution de pression au-dessous de la voilure pour les trois stratégies d'optimisation en comparaison avec la configuration initiale. Les profils de trois sections de l'aile optimisée par les stratégies A, B et C sont données par la figure (4.37).

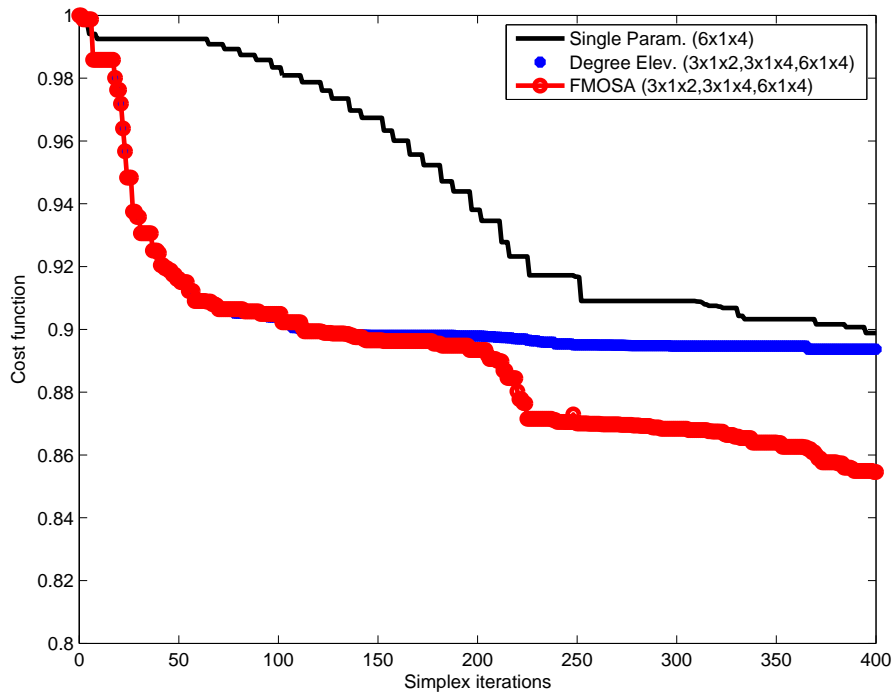


FIGURE 4.33 – Comparaison entre les courbes de convergence obtenues par les stratégies d’optimisation A, B et C

	C_L	C_D	J_SB	Fonction coût
Config. initiale	0.00424370	0.000813619	91.981413	1.
Test A	0.00427057	0.000813141	74.309577	0.8988192353
Test B	0.00422259	0.000814257	73.459053	0.8936606304
Test C	0.00433770	0.000812811	67.142264	0.8545980704

TABLE 4.6 – Comparaison entre les coefficients aérodynamiques, le bang sonique, et la valeur de la fonction objectif associés aux stratégies A, B, et C après 400 itérations



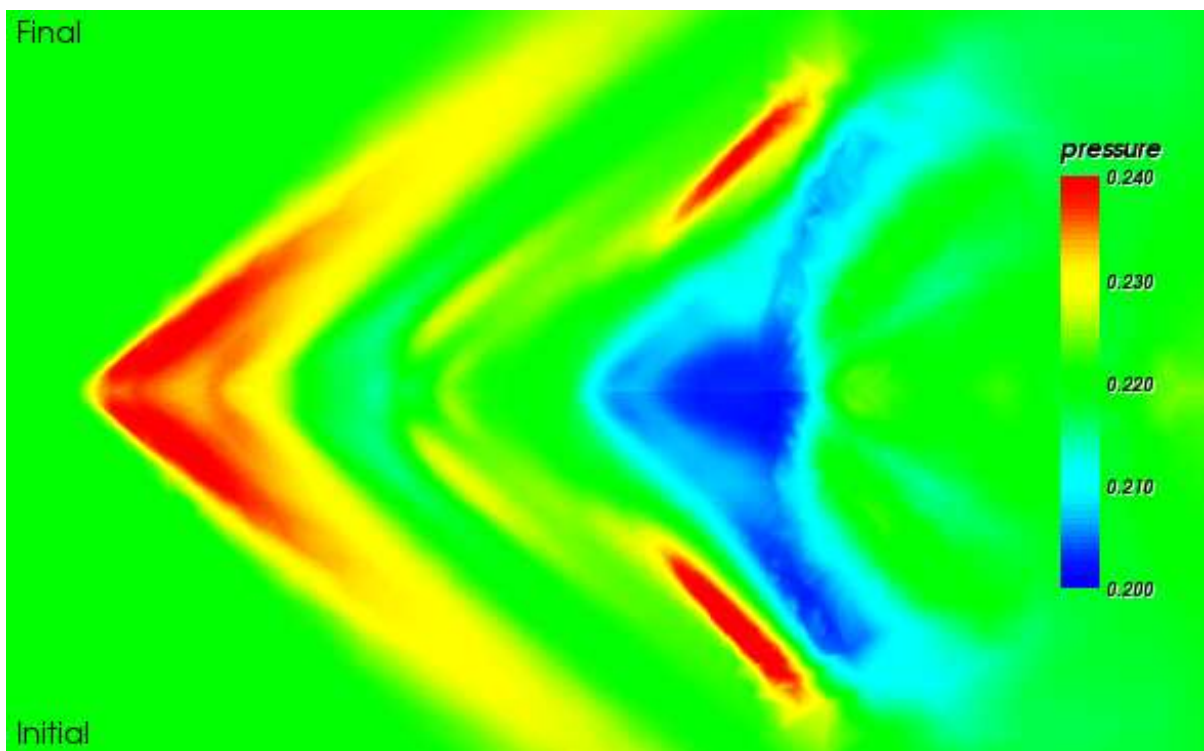
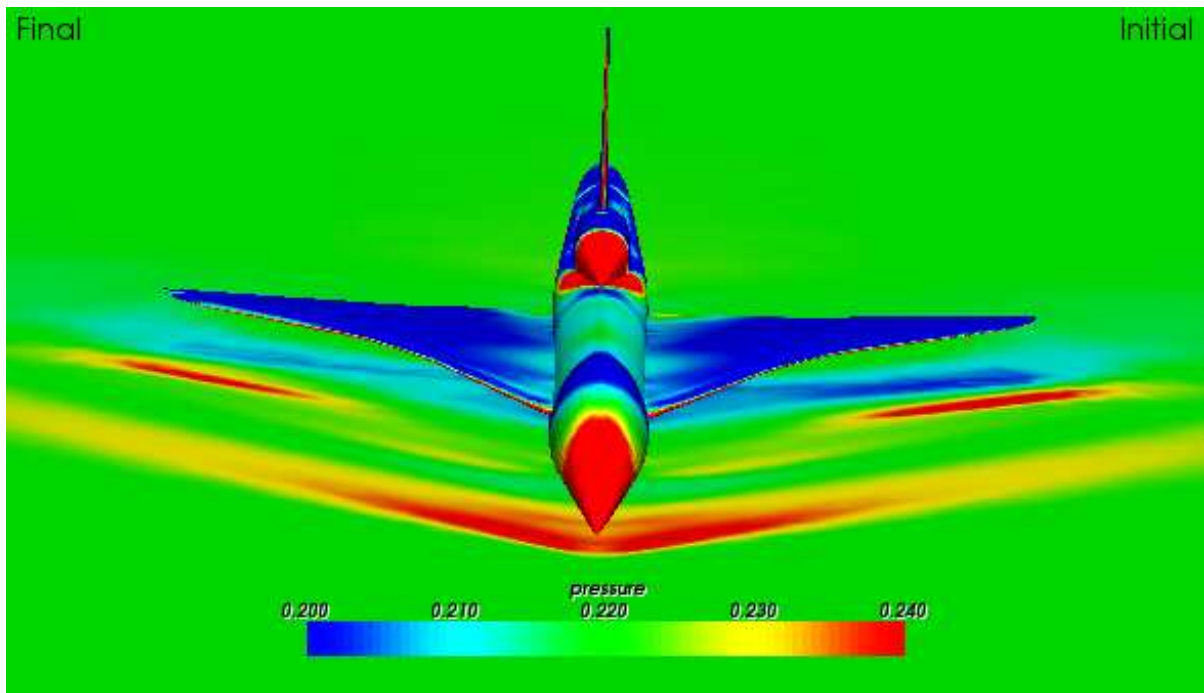


FIGURE 4.34 – Distribution de pression au-dessous de l’avion obtenue par la stratégie A (final) en comparaison avec la configuration initiale (initial); (en haut) une vue globale, (en bas) une trace au-dessous de l’avion

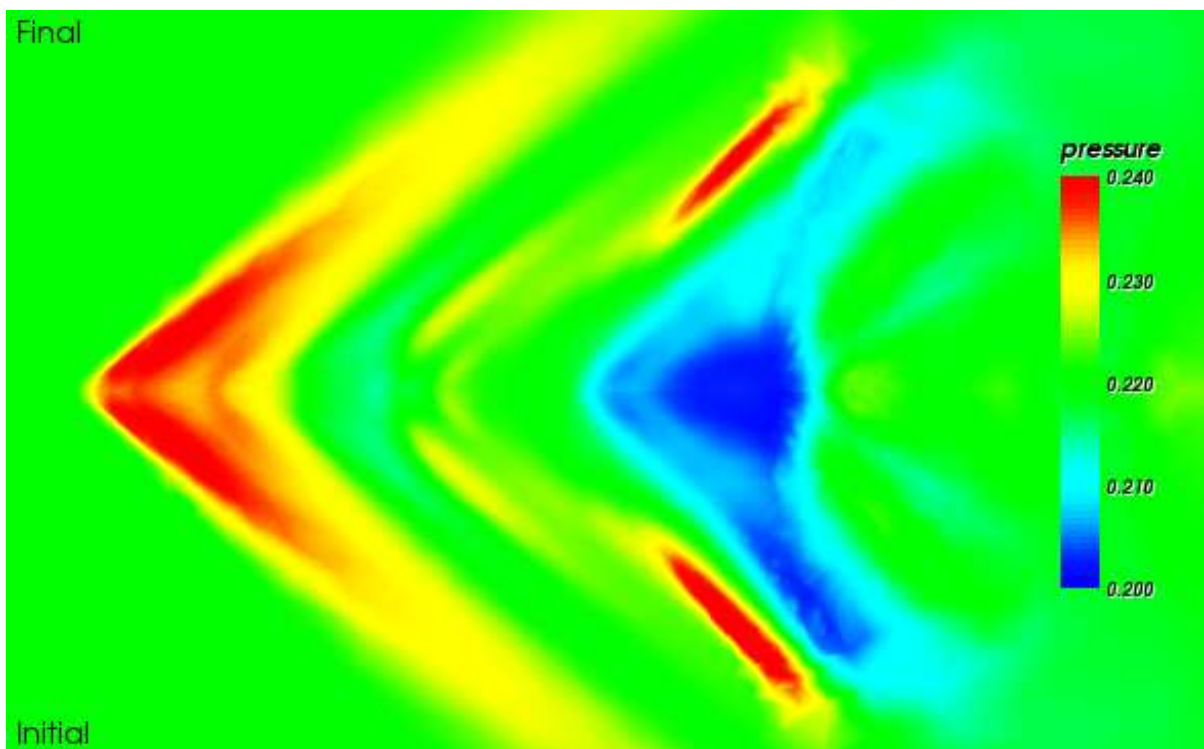
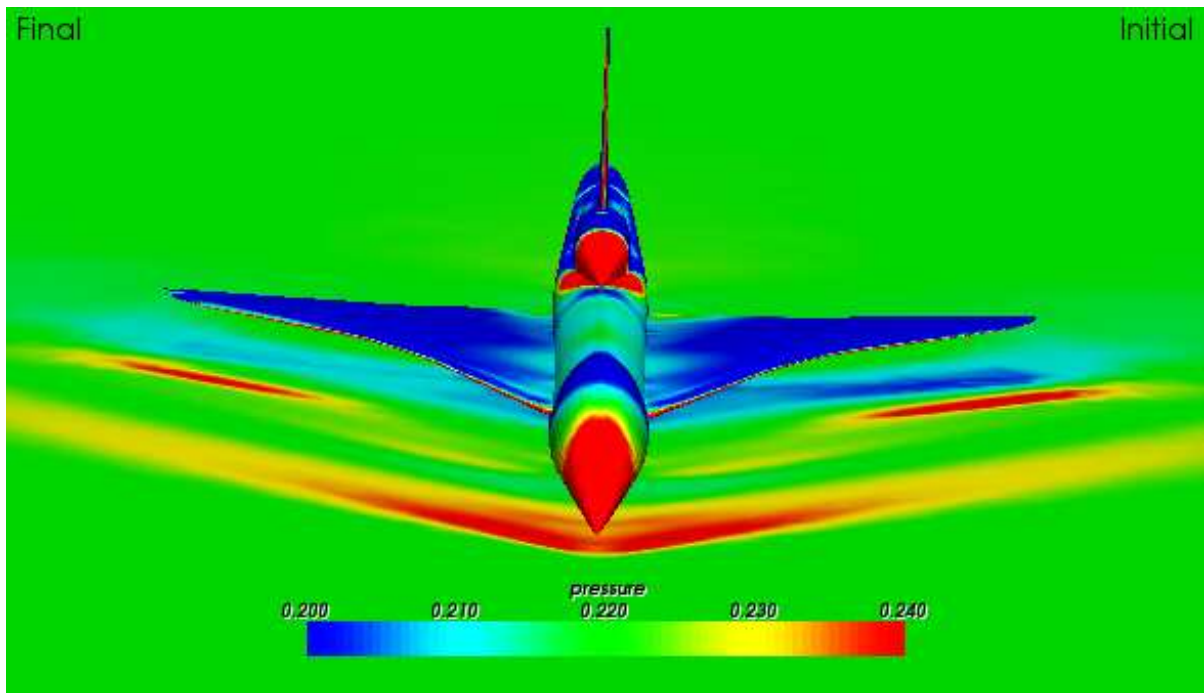


FIGURE 4.35 – Distribution de pression au-dessous de l'avion obtenue par la stratégie B (final) en comparaison avec la configuration initiale (initial); (en haut) une vue globale, (en bas) une trace au-dessous de l'avion



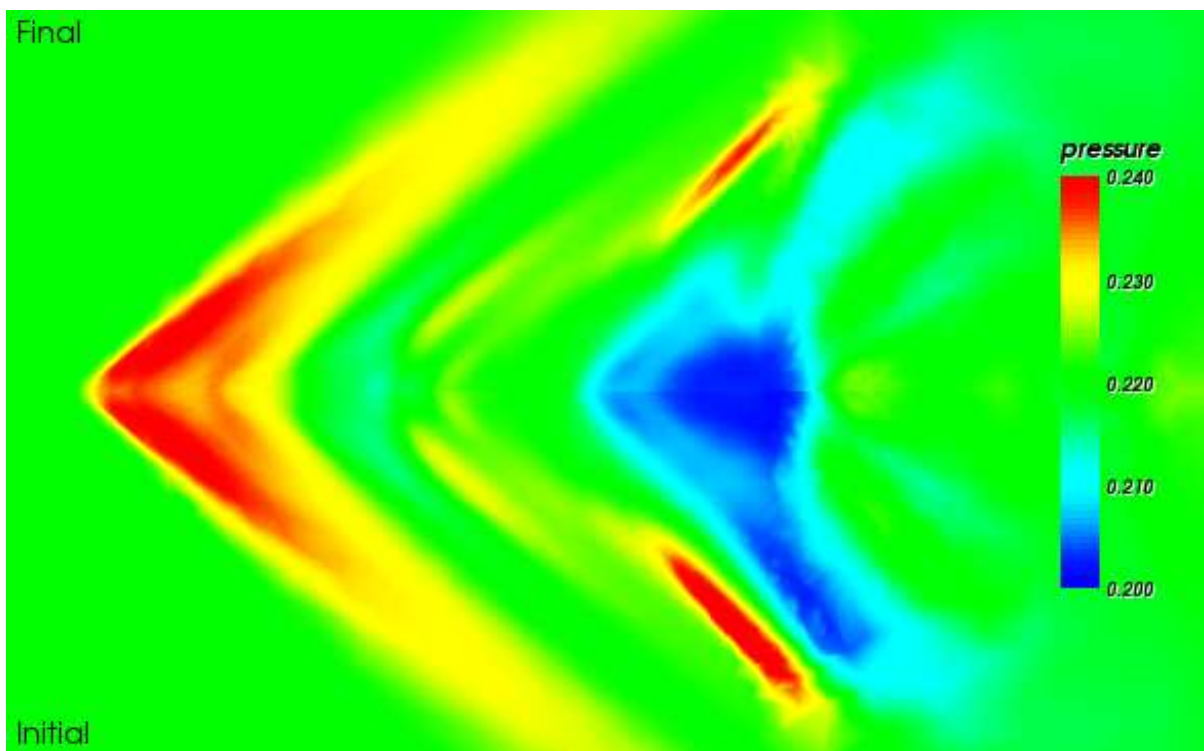
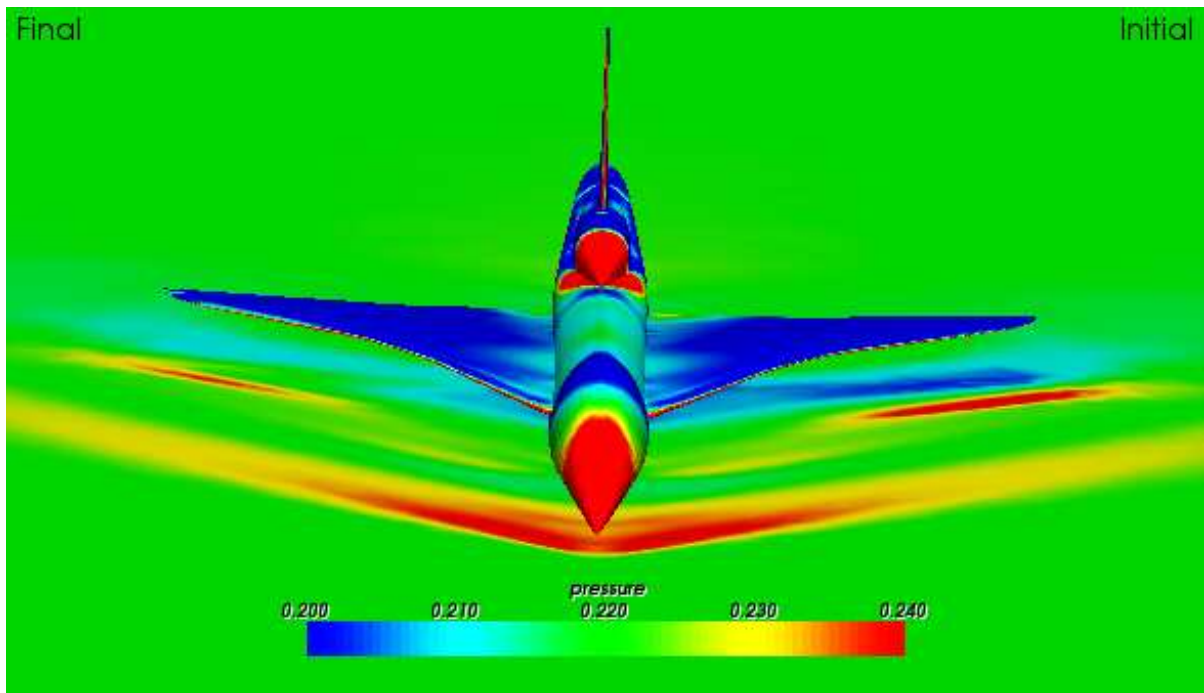


FIGURE 4.36 – Distribution de pression au-dessous de l’avion obtenue par la stratégie C (final) en comparaison avec la configuration initiale (initial); (en haut) une vue globale, (en bas) une trace au-dessous de l’avion

Black: Initial Config.  
Red : Optimal Config.



Black: Initial Config.  
Red : Optimal Config.



Black: Initial Config.  
Red : Optimal Config.



FIGURE 4.37 – Profils de trois sections de l’aile optimisée par les stratégies *A*, *B* et *C* (respectivement du haut en bas)

#### 4.4.2.2 Optimisation du nez

On s'intéresse maintenant à un autre cas test qui consiste à optimiser le nez du jet d'affaires supersonique afin de réduire son impact sur le bang sonique avec des contraintes sur la traînée et sur la portance. On paramétrise le nez du jet d'affaire en considérant trois types de paramétrisations :  $2 \times 1 \times 1$  (6 d.d.l),  $3 \times 1 \times 1$  (12 d.d.l) et  $3 \times 2 \times 2$  (30 d.d.l). On montre dans la figure (4.38) la boîte englobant la forme et les déplacements des points de contrôle.

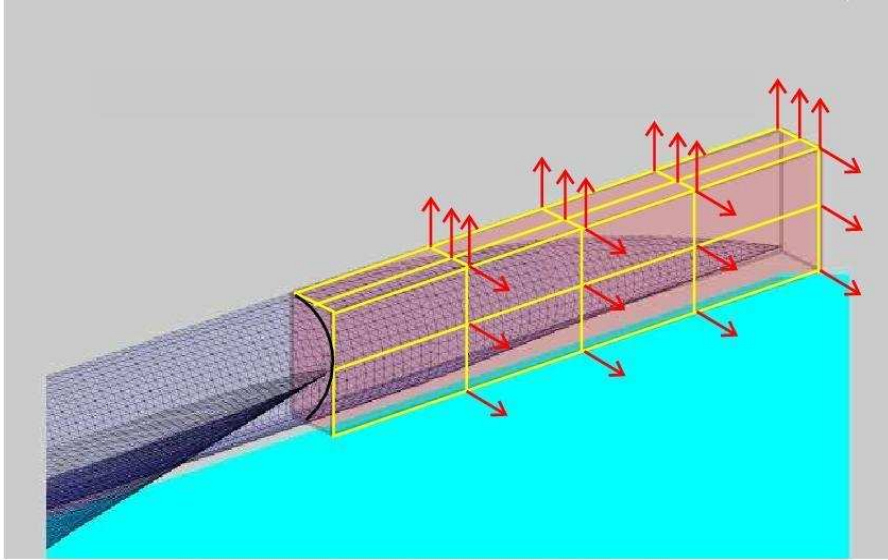


FIGURE 4.38 – La boîte englobant le nez d'un jet d'affaires supersonique ; les flèches désignent les déplacements des points de contrôle (de [32]).

On cherche à démontrer le potentiel de notre approche multiniveau en comparant trois stratégies d'optimisation comme les cas tests précédents, qu'on désigne aussi par *A*, *B* et *C*. On schématise dans la table (4.7) les différents niveaux utilisés en indiquant le nombre d'itérations pour l'algorithme FMOSA.

<b>Fin</b> ( $3 \times 1 \times 2$ )								
<b>Moyen</b> ( $3 \times 1 \times 1$ )								
<b>Grossier</b> ( $2 \times 1 \times 1$ )	70		70		70		70	
		↗		↘	↗		↘	↗
			15		15		15	
								15

TABLE 4.7 – Algorithme FMOSA ; les entiers désignent le nombre d'itérations d'optimisation ; ↗ indique élévation du degré + relaxation ; ↘ correspond à la reformulation du problème de correction au niveau grossier.

Dans la figure (4.39) on compare les courbes de convergence obtenues par les trois stratégies d'optimisation. On constate que les deux méthodes basées sur un processus hiérarchique

permettent une convergence rapide et atteignent une meilleure performance. Quant à la méthode de base qui est associée à la paramétrisation fine de degré  $3 \times 2 \times 2$ , la convergence est lente à cause du nombre important des points de contrôle (28 d.d.l). La dernière stratégie (FMOSA) réalise un gain de 5% en fonctionnelle coût après 400 itérations par rapport à la deuxième méthode qui se base sur un simple enrichissement de la paramétrisation. On note que la courbe de convergence associée à la meilleure stratégie présente un grand palier, ceci est dû peut être à la convergence vers un minimum local au milieu du processus. Donc, la dernière phase de l'algorithme FMOSA a permis d'éviter le minimum local.

On résume dans la table (4.8) les valeurs des coefficients  $C_L$  et  $C_D$ , du critère de bang sonique  $J_{SB}$  et de la fonctionnelle coût  $J$  associés à la solution optimale obtenue par chacune des trois stratégies d'optimisation  $A$ ,  $B$  et  $C$ . Les méthodes  $B$  et  $C$ , qui sont basées sur un processus hiérarchique, permettent de réduire mieux le critère de bang sonique par rapport à la méthode de base associée à la paramétrisation fine. On note que les contraintes sur la traînée et sur la portance ont été respectées. L'algorithme FMOSA est encore efficace par rapport à l'algorithme basé sur l'enrichissement progressif de paramétrisation. Ceci est illustré dans les figures ((4.40)-(4.42)) qui montrent la distribution de pression au-dessous de l'avion obtenue par chacune des stratégies d'optimisation en comparaison avec la configuration initiale. Quant à la figure (4.43), elle montre les profils de trois sections du nez optimisées par les trois stratégies  $A$ ,  $B$  et  $C$ . On remarque les formes obtenues ne sont pas réalistes, ceci est dû au manque de contrainte géométrique.

	$C_L$	$C_D$	$J_{SB}$	Fonction coût
Config. initiale	0.0039533115	0.00079464753	73.925929	1.
Test A	0.0039541543	0.00079502392	69.977041	0.97292662
Test B	0.0041089977	0.00079336231	60.237911	0.90268568
Test C	0.0039801960	0.00079486575	53.285429	0.84899637

TABLE 4.8 – Comparaison entre les coefficients aérodynamiques, le bang sonique, et la valeur de la fonction objectif associés aux stratégies  $A$ ,  $B$ , et  $C$

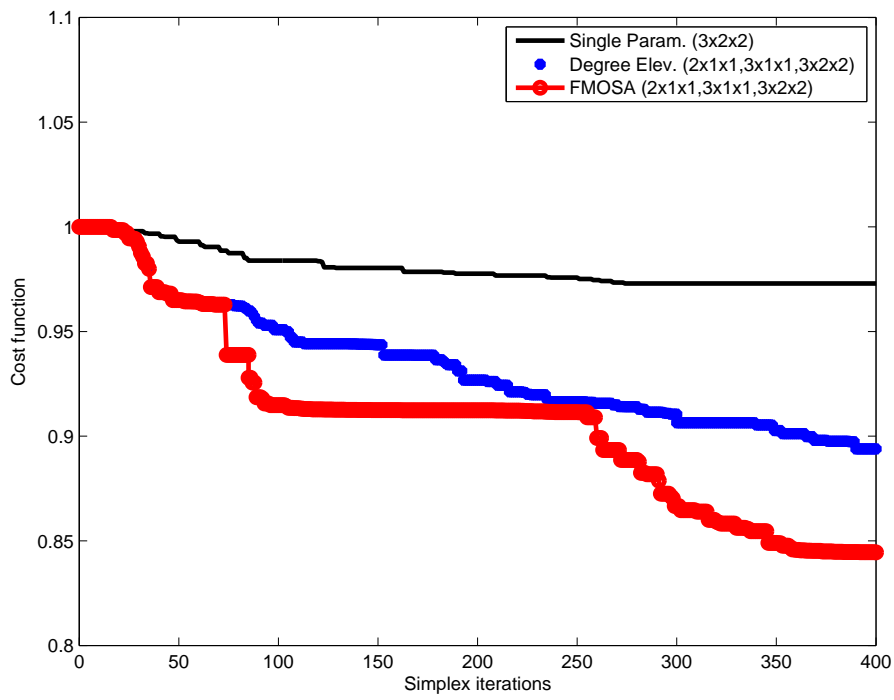


FIGURE 4.39 – Comparaison entre les courbes de convergence obtenues par les stratégies d’optimisation A, B et C

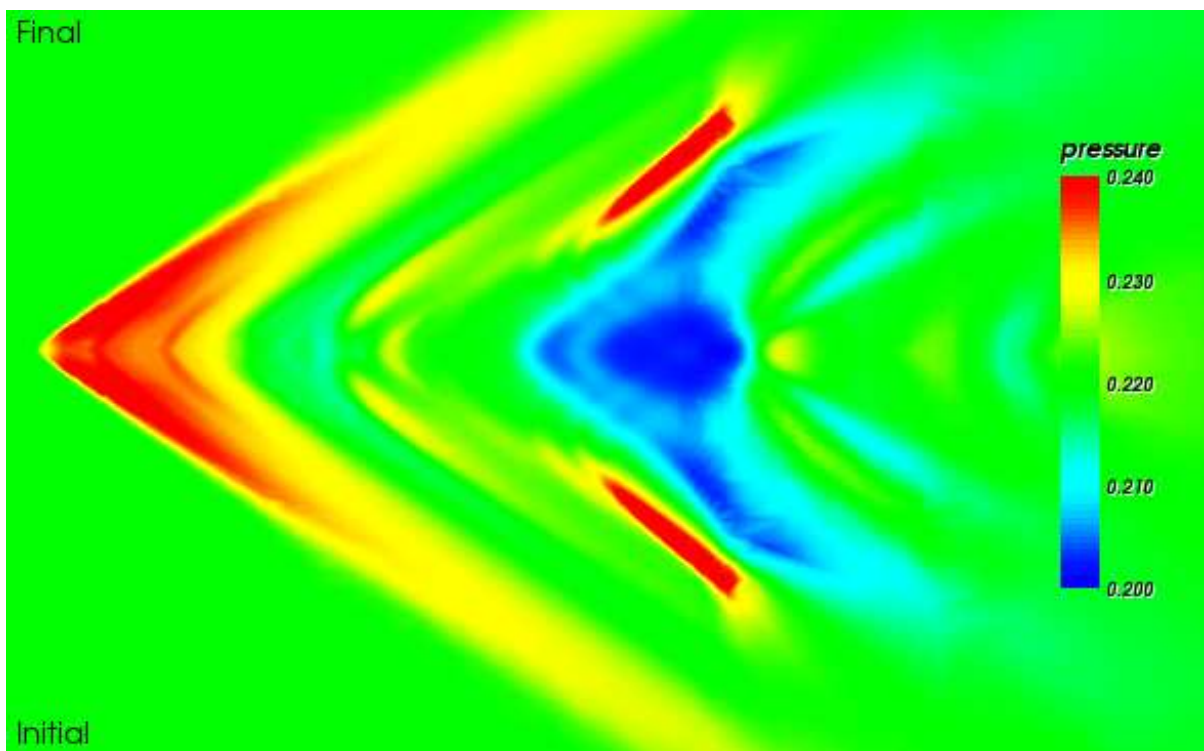
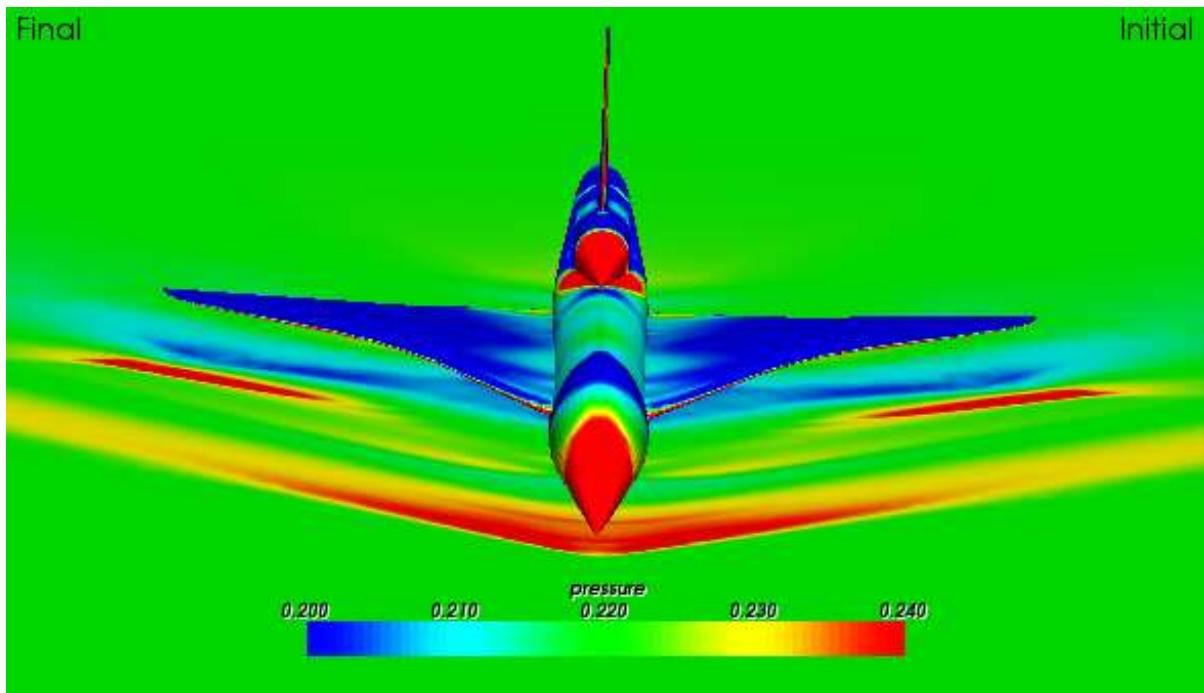


FIGURE 4.40 – Distribution de pression au-dessous de l'avion obtenue par la stratégie A (final) en comparaison avec la configuration initiale (initial); (en haut) une vue globale, (en bas) une trace au-dessous de l'avion

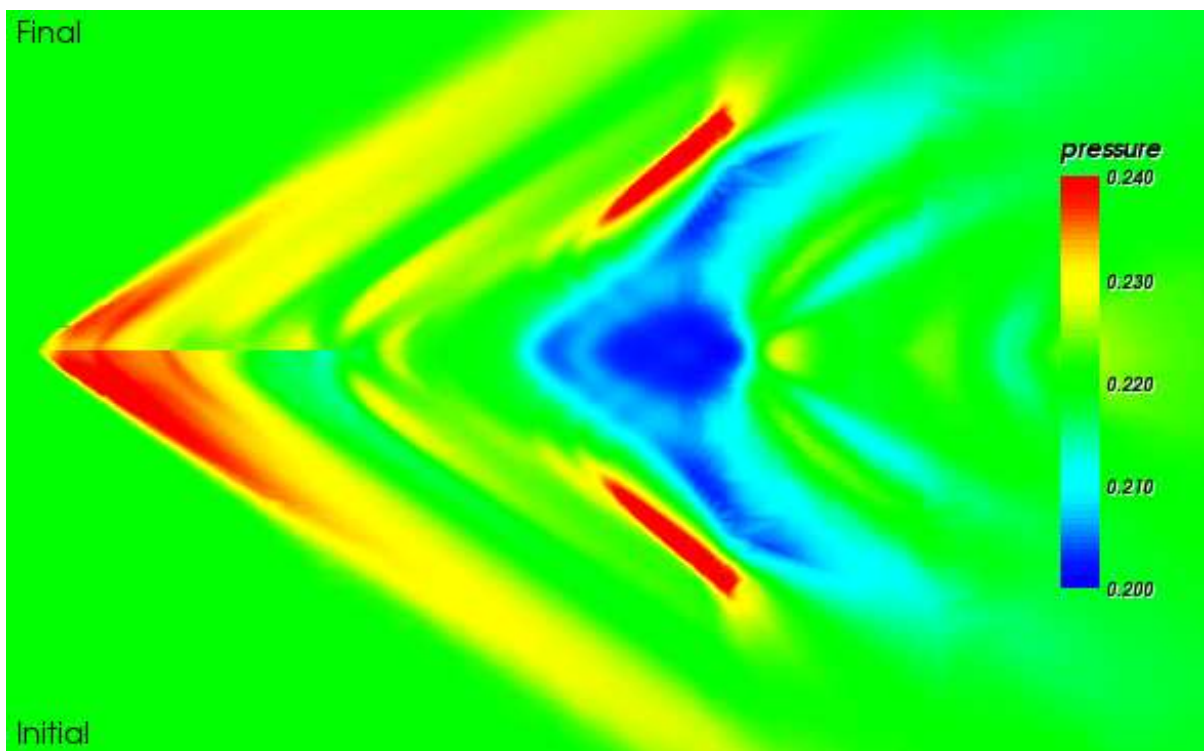
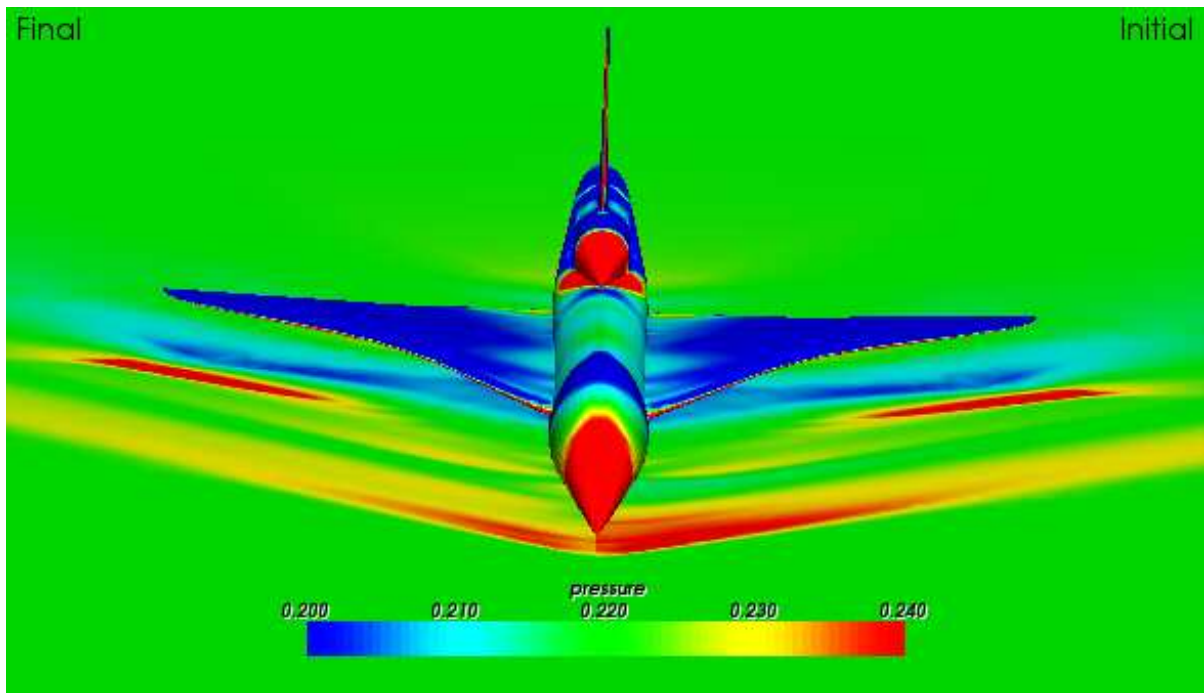


FIGURE 4.41 – Distribution de pression au-dessous de l'avion obtenue par la stratégie B (final) en comparaison avec la configuration initiale (initial); (en haut) une vue globale, (en bas) une trace au-dessous de l'avion



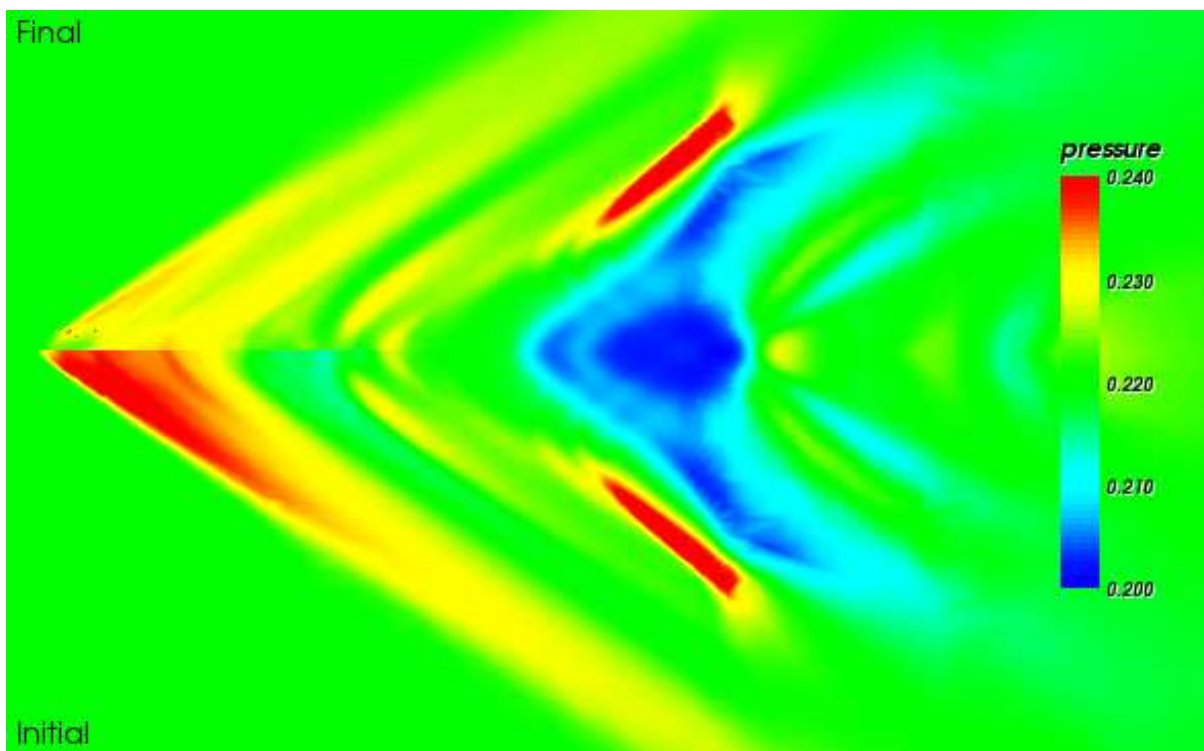
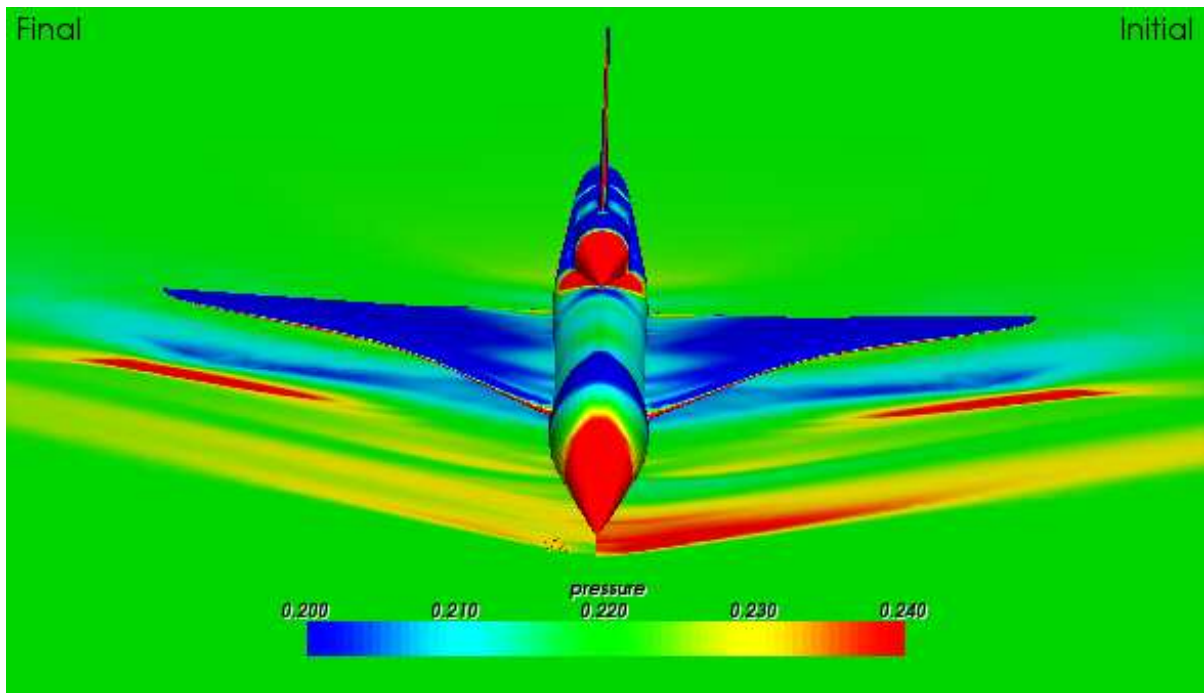
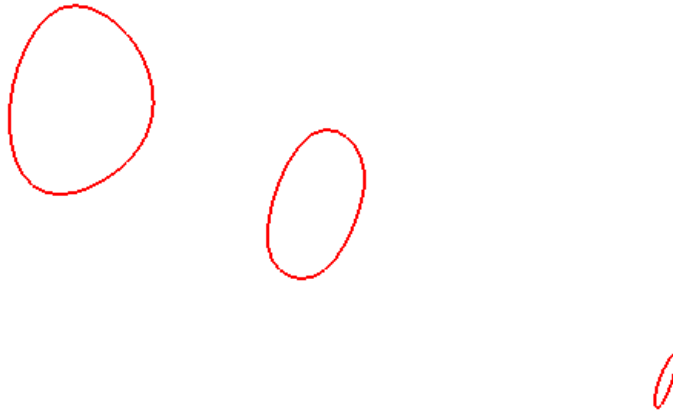


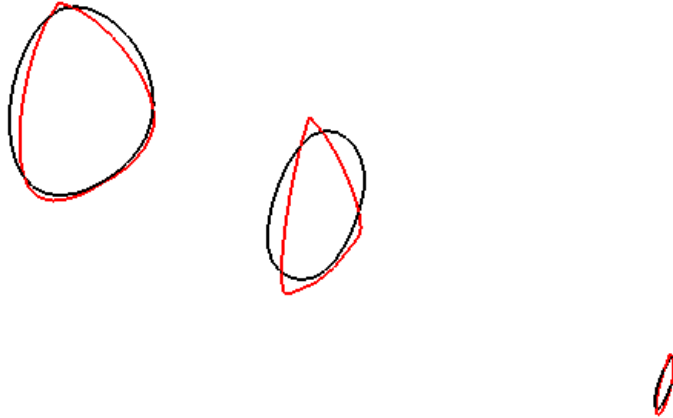
FIGURE 4.42 – Distribution de pression au-dessous de l’avion obtenue par la stratégie C (final) en comparaison avec la configuration initiale (initial); (en haut) une vue globale, (en bas) une trace au-dessous de l’avion



Black: Initial Config.  
Red : Optimal Config.



Black: Initial Config.  
Red : Optimal Config.



Black: Initial Config.  
Red : Optimal Config.

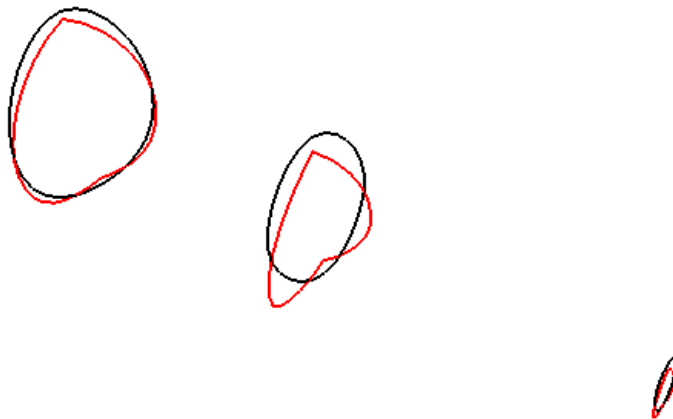


FIGURE 4.43 – Profils de trois coupes de la forme optimale du nez associés aux stratégies *A*, *B* et *C* (respectivement du haut en bas)

## 4.5 Conclusion

Différents tests numériques, qui correspondent à des exemples académiques ou industriels, ont été présentés dans ce chapitre pour expérimenter notre approche multiniveau en optimisation de forme. Toutes ces expériences ont montré le grand potentiel des algorithmes multiniveaux pour améliorer la convergence itérative en évitant les minima locaux et la précision de la solution optimale. En particulier, l'algorithme FMOSA qui utilise une correction au niveau grossier est davantage performant. La clé de la réussite réside dans l'aspect hiérarchique de l'algorithme qui se base sur une hiérarchie de paramétrisations de finesses différentes, et aussi sur l'exactitude des transferts entre les espaces de paramétrisation.

On a observé au cours du processus itératif que la paramétrisation (ou le volume de contrôle) devient irrégulier, ce qui peut augmenter la raideur numérique. Il s'agit donc d'intégrer dans la boucle d'optimisation un processus d'adaptation de paramétrisation, notion analogue à l'adaptation du maillage, pour accélérer la convergence itérative.



# Chapitre 5

## Adaptation de la paramétrisation

### 5.1 Introduction

Au cours de la résolution numérique d'un problème d'EDP, on fait appel généralement à un processus d'adaptation du maillage pour accélérer la convergence de l'approximation. En optimisation de forme, on utilise une notion similaire qui concerne la paramétrisation ou la représentation de forme. Cette dernière devient irrégulière au cours du processus d'optimisation, ce qui par conséquent peut rendre le problème d'optimisation numériquement raide. En effet, On a vu au chapitre (3) qu'à partir d'une paramétrisation de Bézier associée à une forme donnée, on peut construire, en utilisant une infinité d'élévations de degré, une autre paramétrisation équivalente superposée à la même forme. Donc, avec une paramétrisation irrégulière, on est loin de la convergence. Il s'agit donc d'introduire une procédure d'adaptation de la paramétrisation dans la boucle d'optimisation afin que la paramétrisation soit la plus lisse possible, ceci en minimisant un critère purement géométrique de régularisation ; donc à un coût marginal.

Nous présentons dans ce chapitre notre processus d'adaptation de la paramétrisation dans deux contextes différents :

1. La forme (une courbe plane) est représentée par une paramétrisation de Bézier [42] ; le processus d'adaptation consiste à régulariser le polygone de contrôle en minimisant un critère de variation totale qui mesure la régularité de la paramétrisation. Cette approche sera testée en utilisant un problème modèle de calcul des variations présenté dans le chapitre précédent. On trouve dans [12] [108] d'autres études similaires.
2. La forme (aile 3D) est représentée par la méthode Free-Form Deformation [99] ; le processus d'adaptation consiste à modifier la carte FFD afin de repositionner les points de contrôle autour de la forme d'une façon régulière. Cette approche va être testée sur un problème d'optimisation d'une voilure d'avion d'affaire en régime transsonique.

## 5.2 Approche 1D : cas d'une paramétrisation par courbe de Bézier

### 5.2.1 Formulation du problème

On considère la paramétrisation de Bézier de degré  $n$  :

$$\begin{cases} x(t) = B_n(t)^T X \\ y(t) = B_n(t)^T Y \end{cases} \quad (5.1)$$

où  $t \in [0, 1]$ ,  $B_n(t)^T = (B_n^k(t))_{0 \leq k \leq n}$  est le vecteur des polynômes des Bernstein, et

$$X = \{x_k\}, \quad Y = \{y_k\}, \quad (k = 0, 1, \dots, n)$$

On note par,

$$P_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix} \quad (k = 0, 1, \dots, n)$$

les points constituant le polygone de contrôle.

On désigne par le vecteur  $X$  le support de la paramétrisation, et par  $Y$  le vecteur de conception. En pratique, on fixe le support  $X$  et on optimise  $Y$  vis-à-vis d'un critère aérodynamique ou géométrique, comme par exemple la réduction de la traînée. En procédant ainsi, au lieu d'optimiser globalement les deux vecteurs  $X$  et  $Y$ , on réduit la raideur du système et la dimension de l'espace de recherche, mais on soulève la question du choix judicieux du support  $X$ . Donc, on propose dans cette partie une méthodologie qui consiste, après chaque phase d'optimisation à support fixe, à choisir un autre support afin que le polygone de contrôle, qui paramétrise la forme résultante, soit régulier et que l'approximation par rapport à la forme cible soit la meilleure possible.

On considère un arc de courbe régulier dans le plan  $(x, y)$ , dit *arc cible*, constituant un graphe d'équation :

$$y = y_T(x) \quad (0 \leq x \leq 1) \quad (5.2)$$

On peut se restreindre au cas particulier où la cible est elle-même une courbe de Bézier de degré  $n$ , donnée implicitement par la paramétrisation initiale suivante :

$$\begin{cases} x_T^0(\tau) = B_n(\tau)^T X^0 \\ y_T^0(\tau) = B_n(\tau)^T Y^0 \end{cases} \quad (5.3)$$

C'est le cas notamment si on adapte le support  $X$  après une phase d'optimisation à support fixé  $X^0$  conduisant au vecteur de conception  $Y^0$ .

#### 5.2.1.1 Contraintes géométriques

On suppose que les points limites de l'arc sont imposés :

$$x_0 = y_0 = 0 \quad x_n - 1 = y_n = 0 \quad (5.4)$$

De plus, on impose que la tangente à l'origine soit verticale par la condition

$$x_1 = x_0 (= 0) \quad (5.5)$$

qui implique que  $x'(0) = 0$ .

Ces spécifications permettent de réduire le jeu  $(X, Y)$  de  $2(n + 1)$  paramètres à un sous-ensemble  $(\xi, \eta)$  tel que :

$$\begin{cases} X = E \xi + e \\ Y = F \eta + f \end{cases} \quad (5.6)$$

où

$$\xi = \begin{pmatrix} x_2 \\ x_3 \\ \vdots \\ x_{n-1} \end{pmatrix}, \quad \eta = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

sont des vecteurs-colonnes de dimension  $n - 2$  et  $n - 1$  respectivement, alors que

$$e = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \quad f = 0$$

sont des vecteurs-colonnes donnés de dimension  $n + 1$ , et les matrices

$$E = \widehat{E}_{n-2} = \begin{pmatrix} \dots & 0 & \dots \\ \dots & 0 & \dots \\ & I_{n-2} & \\ \dots & 0 & \dots \end{pmatrix}, \quad F = \widehat{F}_{n-1} = \begin{pmatrix} \dots & 0 & \dots \\ & I_{n-1} & \\ \dots & 0 & \dots \end{pmatrix}$$

sont de dimension  $(n + 1) \times (n - 2)$  et  $(n + 1) \times (n - 1)$  respectivement ( $I_m$  est la matrice identité de dimension  $m$ ).

Le système (5.6) sans les définitions qui suivent, correspond également au cas où le vecteur  $X$  des abscisses des points de contrôle de la paramétrisation de Bézier de degré  $n$ , est le résultat de  $p$  applications du processus d'élévation du degré (qui est un opérateur linéaire), à partir des abscisses des points de contrôle d'une paramétrisation de Bézier de moindre degré,  $n - p$ . Dans ce cas, le vecteur colonne  $\xi$  qui est associé à la paramétrisation de Bézier de degré  $n - p$  a pour dimension  $n - p - 2$ , et la matrice  $E$  admet la définition différente suivante :

$$E = \mathcal{E}_{n-1} \mathcal{E}_{n-2} \dots \mathcal{E}_{n-p} \widehat{E}_{n-p-2} \quad (5.7)$$

où d'une manière générale,  $\mathcal{E}_m$  est la matrice de dimension  $(m + 2) \times (m + 1)$  qui correspond à l'élévation du degré d'une unité appliquée aux abscisses des points de contrôle d'une



On souhaite imposer la contrainte suivante : à  $X$  fixé (i.e. à  $\xi$  fixé),  $Y$  (i.e.  $\eta$ ) minimise  $J$ , ce qui équivaut à la condition suivante :

$$\frac{\partial J}{\partial \eta} = \frac{\partial J}{\partial Y} F = 0 \quad (5.15)$$

Or,

$$\begin{aligned} \frac{\partial J}{\partial Y} &= \int_0^1 [y(t) - \bar{y}_T(t, X)] \frac{\partial y}{\partial Y} x'(t) dt \\ &= \int_0^1 [B_n(t)^T Y - \bar{y}_T(t, X)] B_n(t)^T nB_{n-1}(t)^T \Delta X dt \end{aligned} \quad (5.16)$$

et

$$\left( \frac{\partial J}{\partial Y} \right)^T = \int_0^1 B_n(t) [B_n(t)^T Y - \bar{y}_T(t, X)] nB_{n-1}(t)^T \Delta X dt = A(X) Y - b(X) \quad (5.17)$$

où la matrice  $A(X)$  est de dimension  $(n + 1) \times (n + 1)$  et admet la définition suivante :

$$A(X) = \int_0^1 B_n(t) B_n(t)^T nB_{n-1}(t)^T \Delta X dt \quad (5.18)$$

et le vecteur

$$b(X) = \int_0^1 B_n(t) \bar{y}_T(t, X) nB_{n-1}(t)^T \Delta X dt \quad (5.19)$$

dont la non-linéarité en  $X$  provient de la dépendance de  $X$  du facteur  $\bar{y}_T(t, X)$ .

On note que la matrice  $A(X)$  est symétrique et dépend linéairement de  $X$  :

$$A(X) = \mathcal{A} \otimes X \quad (5.20)$$

où  $\mathcal{A} = A'(X)$  est un tenseur d'ordre 3, indépendant de  $X$ , et  $\otimes$  correspond à un certain produit contracté.

En définitive, la contrainte de moindres carrés s'exprime comme suit :

$$\left( \frac{\partial J}{\partial \eta} \right)^T = F^T (A(X) Y - b(X)) = 0 \quad (5.21)$$

### 5.2.1.3 Normalisation des équations

Les éléments diagonaux de la matrice  $A(X)$  augmentent fortement lorsque  $n \rightarrow \infty$ , ce qui pose des problèmes d'implémentation pour les grandes valeurs de  $n$ . Pour remédier à ce problème, on normalise cette matrice par sa partie diagonale par la procédure qui suit.

La matrice  $A(X)$  est symétrique définie-positive. La matrice  $F^T A(X) F$  est le bloc diagonal de dimension  $(n - 1) \times (n - 1)$  obtenu à partir de cette matrice en supprimant les quatre bords. Il s'agit donc également d'une matrice réelle symétrique définie-positive que l'on peut décomposer comme suit :

$$F^T A(X) F = U + D^2 + U^T \quad (5.22)$$



où les matrices  $U$  et  $D$  sont de dimension  $(n-1) \times (n-1)$ ,  $U$  est triangulaire supérieure stricte, et  $D$  diagonale réelle positive. Posons :

$$\bar{A} = D^{-1} F^T A(X) F D^{-1} \quad (5.23)$$

et observons que cette matrice est également symétrique définie-positive. On lui applique la décomposition de Choleski :

$$\bar{A} = W^T W \quad (5.24)$$

où la matrice  $W$  est réelle et triangulaire. Il vient :

$$F^T A(X) F = D W^T W D \quad (5.25)$$

Cette expression sera utilisée ultérieurement pour l'inversion de systèmes faisant intervenir la matrice  $F^T A(X) F$ .

#### 5.2.1.4 Calcul du critère de régularisation et de son gradient

On introduit le critère de régularisation (la variation totale) à minimiser sous les contraintes précédentes (contraintes géométriques et contrainte de meilleure approximation), où la dépendance implicite en  $\xi$  est le fait de ces contraintes.

$$TV(Y) = \sum_{k=1}^n |y_k - y_{k-1}| = J_2(Y(\xi)) \quad (5.26)$$

La dérivée  $\frac{\partial J_2}{\partial \xi}$  se calcule par applications successives de la règle de dérivation des fonctions composées. On pose d'abord :

$$\frac{\partial J_2}{\partial Y} = p(Y)^T \quad (5.27)$$

et on remarque que cette dérivée n'est pas définie lorsque  $y_k - y_{k-1}$  change de signe. On peut contourner cette difficulté en utilisant plutôt (formellement ou numériquement) le critère régularisé suivant :

$$J_2^\varepsilon(Y) = \sum_{k=1}^n \sqrt{(y_k - y_{k-1})^2 + \varepsilon} \quad (5.28)$$

où  $\varepsilon$  est un nombre positif petit. Pour tout indice  $k = 1, 2, \dots, n-1$ , il vient :

$$p_k^\varepsilon(Y) = \frac{\partial J_2^\varepsilon}{\partial y_k} = \frac{y_k - y_{k-1}}{\sqrt{(y_k - y_{k-1})^2 + \varepsilon}} - \frac{y_{k+1} - y_k}{\sqrt{(y_{k+1} - y_k)^2 + \varepsilon}} \quad (5.29)$$

Par passage à la limite ( $\varepsilon \rightarrow 0$ ), il vient :

$$p_k(Y) = \begin{cases} +2 \text{ si } y_k \text{ est un maximum local strict de la suite } \{y_k\}, \\ \quad \text{i.e. } y_k > \max(y_{k-1}, y_{k+1}), \\ +1 \text{ si } y_{k-1} < y_k = y_{k+1}, \\ \quad \text{ou si } y_{k-1} = y_k > y_{k+1}, \\ 0 \text{ si } \min(y_{k-1}, y_{k+1}) < y_k < \max(y_{k-1}, y_{k+1}), \\ \quad \text{ou si } y_{k-1} = y_k = y_{k+1}, \\ -1 \text{ si } y_{k-1} = y_k < y_{k+1}, \\ \quad \text{ou si } y_{k-1} > y_k = y_{k+1}, \\ -2 \text{ si } y_k \text{ est un minimum local strict de la suite } \{y_k\}, \\ \quad \text{i.e. } y_k < \min(y_{k-1}, y_{k+1}). \end{cases} \quad (5.30)$$

Noter que cette définition peut s'appliquer partout, mais n'est pas pour autant continue. Par conséquent, le gradient de la fonctionnelle  $J_2$  peut avoir des sauts.

Il vient :

$$\frac{\partial J_2}{\partial \eta} = \frac{\partial J_2}{\partial Y} \frac{\partial Y}{\partial \eta} = p(Y)^T F \quad (5.31)$$

puis

$$\frac{\partial J_2}{\partial \xi} = \frac{\partial J_2}{\partial \eta} \frac{\partial \eta}{\partial \xi} = p(Y)^T F \frac{\partial \eta}{\partial \xi} \quad (5.32)$$

Le facteur

$$q = \frac{\partial \eta}{\partial \xi} \quad (5.33)$$

correspond à la dérivée des ordonnées lorsque le support change, sous la contrainte d'approximation par moindres carrés, (5.21). On l'obtient en dérivant cette contrainte par rapport à  $\xi$ . A cette fin, prenons d'abord la différentielle de la contrainte correspondant à la perturbation  $dX = E d\xi$  :

$$F^T [dA(X) Y + A(X) dY - b'(X) dX] E = 0 \quad (5.34)$$

où l'indice supérieur ' indique une dérivation par rapport au vecteur  $X$ . Or,

$$dA(X) = A'(X) \otimes dX \quad (5.35)$$

par définition des symboles, où  $A'(X)$  est un tenseur d'ordre 3, en fait indépendant de  $X$ , car la matrice  $A(X)$  en dépend linéairement. Il vient donc :

$$dA(X) = A(dX) \quad (5.36)$$

Notons  $\{e_k\}$  ( $k = 0, 1, \dots, n$ ) la base canonique de  $\mathbb{R}^{n+1}$ . L'équation (5.36) fournit en particulier l'expression suivante de dérivée partielle :

$$\frac{\partial A(X)}{\partial x_k} Y = A(e_k) Y \quad (5.37)$$

Dans ce qui suit, on notera  $\tilde{A}_Y$  la matrice de dimension  $(n + 1) \times (n + 1)$  dont la colonne  $k$  est égale au vecteur  $A(e_k) Y$ . L'équation (5.34) équivaut donc à la suivante :

$$F^T \left[ \tilde{A}_Y - b'(X) + A(X) \frac{\partial Y}{\partial X} \right] E = 0 \quad (5.38)$$

Par ailleurs :

$$\frac{\partial Y}{\partial X} E = F \frac{\partial \eta}{\partial X} \frac{\partial X}{\partial \xi} = F q \quad (5.39)$$

L'inconnue  $q$  est donc la solution du système linéaire suivant :

$$F^T \left[ \tilde{A}_Y - b'(X) \right] E + F^T A(X) F q = 0 \quad (5.40)$$

En résumé, le calcul de  $J_2$  et de sa dérivée par rapport à  $\xi$  s'effectue par la suite des opérations suivantes :

**1. Calcul du critère :**

- Calculer les vecteurs

$$X = E \xi + e \quad (5.41)$$

et  $b(X)$ , et identifier les matrices  $D$  et  $W$  de la décomposition de Choleski (5.25).

- Résoudre le système linéaire

$$F^T A(X) F \eta = F^T [b(X) - A(X) f] \quad (5.42)$$

par la procédure détaillée ci-dessous s'appuyant sur la décomposition de Choleski.

- Calculer le vecteur

$$Y = F \eta + f \quad (5.43)$$

puis la valeur du critère  $J_2 = J_2(Y)$ .

**2. Calcul du gradient du critère :**

- Calculer le vecteur

$$p = p(Y) \quad (5.44)$$

et les matrices  $\tilde{A}_Y$  et  $b'(X)$ .

- Résoudre le système linéaire

$$F^T A(X) F q = F^T [b'(X) - \tilde{A}_Y] E \quad (5.45)$$

par la même procédure que ci-dessus.

- Calculer la dérivée

$$\frac{\partial J_2}{\partial \xi} = p^T F q. \quad (5.46)$$

Il s'agit de résoudre plusieurs systèmes linéaires de la forme :

$$F^T A(X) F z = r \quad (5.47)$$

où  $z$  est le vecteur inconnu et  $r$  un vecteur préalablement calculé. En vertu de (5.25), le système ci-dessus équivaut à :

$$D W^T W D z = r \quad (5.48)$$

ou encore :

$$W^T W \zeta = D^{-1} r \quad (5.49)$$

où  $\zeta = Dz$ . Le système (5.49) se résout classiquement par une "descente-remontée". Enfin, il vient :

$$z = D^{-1} \zeta \quad (5.50)$$

En particulier, si  $n$  est grand, le calcul du membre de droite de (5.45) peut être problématique. Il faudra évaluer directement le produit  $D^{-1} F^T \tilde{A}_Y E$ .

## 5.2.2 Développement formel des calculs

– Calcul de la matrice  $A(X)$  :

$$A(X) = \int_0^1 B_n(t) B_n(t)^T n B_{n-1}(t)^T \Delta X dt \quad (5.51)$$

Pour  $0 \leq i, j \leq n$  :

$$\begin{aligned} A(X)(i, j) &= \int_0^1 B_n^i(t) B_n^j(t) n \sum_{k=0}^{n-1} B_{n-1}^k(t) (x_{k+1} - x_k) dt \\ &= n \sum_{k=0}^{n-1} (x_{k+1} - x_k) \int_0^1 B_n^i(t) B_n^j(t) B_{n-1}^k(t) dt \\ &= n \sum_{k=0}^{n-1} (x_{k+1} - x_k) C_n^i C_n^j C_{n-1}^k \int_0^1 t^{i+j+k} (1-t)^{3n-1-i-j-k} dt \\ &= n \sum_{k=0}^{n-1} (x_{k+1} - x_k) \frac{C_n^i C_n^j C_{n-1}^k}{3 n C_{3n-1}^{i+j+k}} \end{aligned} \quad (5.52)$$

Finalement :

$$A(X)(i, j) = \frac{1}{3} \sum_{k=0}^{n-1} (x_{k+1} - x_k) \frac{C_n^i C_n^j C_{n-1}^k}{C_{3n-1}^{i+j+k}} \quad (5.53)$$

– Calcul du vecteur  $b(X)$  :

$$b(X) = \int_0^1 B_n(t) \bar{y}_T(t, X) n B_{n-1}(t)^T \Delta X dt \quad (5.54)$$

Pour  $0 \leq i \leq n$  :

$$b(X)(i) = \int_0^1 B_n^i(t) \bar{y}_T(t, X) n \sum_{k=0}^{n-1} B_{n-1}^k(t) (x_{k+1} - x_k) dt \quad (5.55)$$

$$b(X)(i) = n \sum_{k=0}^{n-1} (x_{k+1} - x_k) C_n^i C_{n-1}^k \int_0^1 \bar{y}_T(t, X) t^{i+k} (1-t)^{2n-1-i-k} dt \quad (5.56)$$

On pose :  $f_{i,k}(t) = \bar{y}_T(t, X) t^{i+k} (1-t)^{2n-1-i-k}$

$$b(X)(i) = n \sum_{k=0}^{n-1} (x_{k+1} - x_k) C_n^i C_{n-1}^k \int_0^1 f_{i,k}(t) dt \quad (5.57)$$

On approche numériquement l'intégrale de  $f_{i,k}(t)$  par une formule de quadrature (la formule du trapèze). Soit  $M$  un entier strictement positif et  $\sigma = (t_l)_{(0 \leq l \leq M)}$  une subdivision de  $[0,1]$ . Alors :

$$b(X)(i) \approx n \sum_{k=0}^{n-1} (x_{k+1} - x_k) C_n^i C_{n-1}^k \sum_{l=0}^{M-1} (t_{l+1} - t_l) \frac{(f_{i,k}(t_{l+1}) + f_{i,k}(t_l))}{2} \quad (5.58)$$

et enfin :

$$b(X)(i) \approx n \sum_{k=0}^{n-1} \sum_{j=0}^{m-1} (x_{k+1} - x_k) C_n^i C_{n-1}^k (t_{j+1} - t_j) \frac{(f_{i,k}(t_{j+1}) + f_{i,k}(t_j))}{2} \quad (5.59)$$

**Remarque :** Classiquement, lorsque l'intégrande est suffisamment régulière ( $C^2$ ), la formule d'intégration du trapèze qui s'appuie sur une interpolation linéaire, est (au moins) du second ordre par rapport au pas de discrétisation. La règle du trapèze corrigée [23] s'appuie sur une interpolation hermitienne du 3<sup>e</sup> ordre et son application rajoute un terme correctif proportionnel à la variation de dérivée par sous-intervalle. Lorsque le pas d'intégration est constant, la somme de ces termes correctifs est proportionnelle à la variation de dérivée entre les limites d'intégration. Par conséquent, si fortuitement, la dérivée de l'intégrande a la même valeur en ces limites, la correction est nulle, et la formule standard est du 4<sup>e</sup> ordre (en supposant l'intégrande  $C^4$ ). Or la dérivée de la fonction  $f_{i,k}(t)$  s'exprime comme suit :

$$f'_{i,k}(t) = \bar{y}_T(t, X)' t^{i+k} (1-t)^{2n-1-i-k} + \bar{y}_T(t, X) \frac{d}{dt} t^{i+k} (1-t)^{2n-1-i-k} \quad (5.60)$$

Où  $0 \leq i \leq n$  et  $0 \leq k \leq n-1$ , ce qui implique  $0 \leq i+k \leq 2n-1$ . A l'exception des valeurs extrêmes  $i+k = 0$  ou  $2n-1$ , le premier terme dans l'expression de  $f'_{i,k}(t)$  s'annule en  $t = 0$  et  $1$ , à condition que la dérivée de la fonction  $\bar{y}_T(t, X)$  reste bornée ; pour le cas d'un profil d'aile, cette question pose problème en  $t = 0$ , mais on fait l'hypothèse que la paramétrisation assure cette bornitude par régularisation. Le deuxième terme est toujours nul aux limites. On prévoit donc que la formule d'intégration de la fonction  $f_{i,k}(t)$  est précise au 4<sup>e</sup> ordre, sauf peut-être pour  $(i, k) = (0, 0)$  et  $(n, n-1)$ .

– **Calcul de la matrice  $A(Y)$  :**

Pour  $0 \leq i, j \leq n$  :

$$A(Y)(i, j) = \frac{1}{3} \sum_{k=0}^{n-1} (y_{k+1} - y_k) \frac{C_n^i C_n^j C_{n-1}^k}{C_{3n-1}^{i+j+k}} \quad (5.61)$$

– **Calcul de la matrice  $b'(X)$  :**

$$b'(x) = b'_1(X) + b'_2(X) \quad (5.62)$$

où :

$$\begin{cases} b'_1(X) = \int_0^1 B_n(t) \bar{y}_T(t, X) n B_{n-1}(t)^T \Delta dt \\ b'_2(X) = \int_0^1 B_n(t) \bar{y}_T'(t, X) \underbrace{n B_{n-1}(t)^T \Delta X}_{x'(t)} dt \end{cases} \quad (5.63)$$

**Remarque :** le calcul des deux premières lignes et de la dernière colonne de  $b'(X)$  n'est pas nécessaire car elles n'interviennent pas dans le calcul du gradient.

Pour  $0 \leq i \leq n$  et  $2 \leq j \leq n-1$  on a :

$$\begin{aligned} b'_1(X)(i, j) &= \int_0^1 \bar{y}_T(t, X) n B_n^i(t) (B_{n-1}^{j-1}(t) - B_{n-1}^j(t)) dt \\ &= n C_n^i C_{n-1}^{j-1} \int_0^1 \bar{y}_T(t, X) t^{i+j-1} (1-t)^{2n-i-j} dt \\ &\quad - n C_n^i C_{n-1}^j \int_0^1 \bar{y}_T(t, X) t^{i+j-1} (1-t)^{2n-1-i-j} dt \\ &= C_n^i C_n^j \int_0^1 \bar{y}_T(t, X) t^{i+j-1} (1-t)^{2n-1-i-j} (j-nt) dt \end{aligned} \quad (5.64)$$

On pose :  $g_{i,j}(t) = (j-nt)\bar{y}_T(t, X) t^{i+j-1} (1-t)^{2n-1-i-j}$  de sorte que finalement :

$$b'_1(X)(i, j) = C_n^i C_n^j \int_0^1 g_{i,j}(t) dt \quad (5.65)$$

On approche numériquement l'intégrale de  $g_{i,j}(t)$  par une formule de quadrature (la formule du trapèze). Soient  $M$  un entier strictement positif et  $\sigma = (t_l)_{(0 \leq l \leq M)}$  une subdivision de  $[0, 1]$  ; il vient :

$$b'_1(X)(i, j) \simeq C_n^i C_n^j \sum_{l=0}^{M-1} (t_{l+1} - t_l) \frac{(g_{i,j}(t_{l+1}) + g_{i,j}(t_l))}{2} \quad (5.66)$$

**Remarque :** Ici  $0 \leq i \leq n$  et  $2 \leq j \leq n-1$ , de sorte que  $2 \leq i+j \leq 2n-1$ . Par conséquent, en faisant la même hypothèse de régularité qu'à la remarque précédente, il vient :

$$g'_{i,j}(0) = g'_{i,j}(1) = 0 \quad (5.67)$$

La formule d'intégration ci-dessus est donc du 4<sup>e</sup> ordre dans le cas d'une partition uniforme  $t_l = l/M$ .

La dérivée  $b'_1(X)$  fait donc intervenir comme précédemment la fonction nonlinéaire  $\bar{y}_T(t, X)$  sans nouvelle difficulté par rapport au calcul de la fonctionnelle  $J_2$ . Par contre, le terme  $b'_2(X)$  fait intervenir la dérivée de cette fonction nonlinéaire par rapport à  $X$ . Au vu de la définition (5.11), il vient :

$$\bar{y}_T(t, X)' = \frac{\partial \bar{y}_T(t, X)}{\partial X} = \frac{dy_T}{dx}(x(t)) \frac{\partial x(t)}{\partial X} = \frac{dy_T}{dx}(x(t)) B_n(t)^T \quad (5.68)$$

où  $x(t) = B_n(t)^T X$ .

Dans le cas particulier où la cible est elle-même une courbe de Bézier de degré  $n$ , donnée implicitement par la paramétrisation suivante :

$$\begin{cases} x_T^0(\tau) = B_n(\tau)^T X^0 \\ y_T^0(\tau) = B_n(\tau)^T Y^0 \end{cases} \quad (5.69)$$

on a :

$$\frac{d y_T}{d x}(x(t)) = \frac{y_T^{0'}(\tau)}{x_T^{0'}(\tau)}(x(t)) \quad (5.70)$$

où l'indice supérieur ' désigne ici la dérivée par rapport au paramètre  $\tau$ , dont la valeur est une fonction implicite de  $t$  et de  $X$  par la condition :

$$x_T^0(\tau) = x(t) \quad (5.71)$$

soit encore :

$$B_n(\tau)^T X^0 = B_n(t)^T X \quad (5.72)$$

Enfin, les dérivées par rapport à  $\tau$  s'expriment par (??), par substitution de symboles (notamment  $t \rightarrow \tau$ ). Finalement :

$$\bar{y}_T(t, X)' = \frac{B_{n-1}(\tau)^T \Delta Y^0}{B_{n-1}(\tau)^T \Delta X^0} B_n(t)^T \quad (5.73)$$

où  $\tau = \tau(t, X)$  est la solution de l'équation (5.72).

**Cas particulier d'une tangente verticale à l'origine :** Dans ce cas, les vecteurs  $X^0$  et  $X$  satisfont les conditions suivantes :

$$x_0^0 = x_1^0 = 0, \quad x_0 = x_1 = 0 \quad (5.74)$$

L'équation (5.72) se développe alors comme suit :

$$\begin{aligned} \underbrace{(1-\tau)^n x_0^0}_{=0} + \underbrace{n\tau(1-\tau)^{n-1} x_1^0}_{=0} + \frac{n(n-1)}{2} x_2^0 \tau^2 (1-\tau)^{n-2} + \dots = \\ \underbrace{(1-t)^n x_0^0}_{=0} + \underbrace{nt(1-t)^{n-1} x_1^0}_{=0} + \frac{n(n-1)}{2} x_2^0 t^2 (1-t)^{n-2} + \dots \end{aligned} \quad (5.75)$$

ce qui donne :

$$\tau \underset{(t \rightarrow 0)}{\sim} \sqrt{\frac{x_2}{x_2^0}} t \quad (5.76)$$

Par conséquent :

$$\begin{aligned} \bar{y}_T(t, X)' &= \frac{y_1^0 + O(\tau)}{0 + (n-1)\tau x_2^0 + O(\tau^2)} \left( (1, 0, \dots, 0) + O(t) \right) \\ &\underset{(t \rightarrow 0)}{\sim} \frac{y_1^0}{(n-1)\sqrt{x_2 x_2^0} t} (1, 0, \dots, 0) \end{aligned}$$

Par ailleurs,

$$B_{n-1}(t) \Delta X \underset{(t \rightarrow 0)}{\sim} (n-1)x_2 t \quad (5.77)$$

En définitive, l'intégrande  $\phi_2(t)$  de  $b'_2(X)$  admet la limite finie suivante :

$$\phi_2(t) \rightarrow C B_n(0) B_n(0)^T \quad (5.78)$$

où :

$$C = n y_1^0 \sqrt{\frac{x_2}{x_2^0}} \quad (5.79)$$

est une constante numérique, et

$$B_n(0) B_n(0)^T = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \quad (5.80)$$

est une matrice constante de dimension  $(n+1) \times (n+1)$ , dont seul l'élément diagonal de la première ligne n'est pas nul.

### 5.2.3 Résultats numériques

#### 5.2.3.1 Validation du calcul formel du gradient de $J_2$

Dans un premier temps, on a fait un test de validation du calcul formel du gradient du critère  $J_2$ , présenté dans la section précédente, en utilisant deux outils : Tapenade, qui est un logiciel de différenciation automatique développé par le projet TROPICS (INRIA Sophia Antipolis) [?] [?], et la méthode des différences finies :

$$\frac{\partial J_2}{\partial x_2} \simeq \frac{J_2(\xi + \epsilon \vec{e}_1) - J_2(\xi)}{\epsilon} \quad (5.81)$$

et

$$\frac{\partial J_2}{\partial x_3} \simeq \frac{J_2(\xi + \epsilon \vec{e}_2) - J_2(\xi)}{\epsilon} \quad (5.82)$$

avec  $\epsilon = 10^{-5}$ .

Les figures (5.1) et (5.2) illustrent le calcul du gradient du critère  $J_2$  avec trois calculs indépendants : le calcul formel, la différenciation automatique, et les différences finies. Dans ce cas test, on choisit  $n = 4$ ,  $\xi = (x_2, x_3)^T$  ( $x_0 = x_1 = 0, x_4 = 1$ ), et  $\eta = (y_1, y_2, y_3)^T$  ( $y_1 = y_4 = 0$ ) est calculée en considérant comme arc cible,

$$y = \bar{y}(x) = \sqrt{x}(1-x)/6 \quad (5.83)$$

qui représente une courbe de Bézier de degré 3 associée aux points de contrôle suivants :

$$P_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, P_2 = \begin{pmatrix} 0 \\ 1/18 \end{pmatrix}, P_3 = \begin{pmatrix} 1/3 \\ 1/9 \end{pmatrix}, P_4 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (5.84)$$

On constate que les trois calculs, qui présentent plusieurs fronts de discontinuités, sont parfaitement superposés.



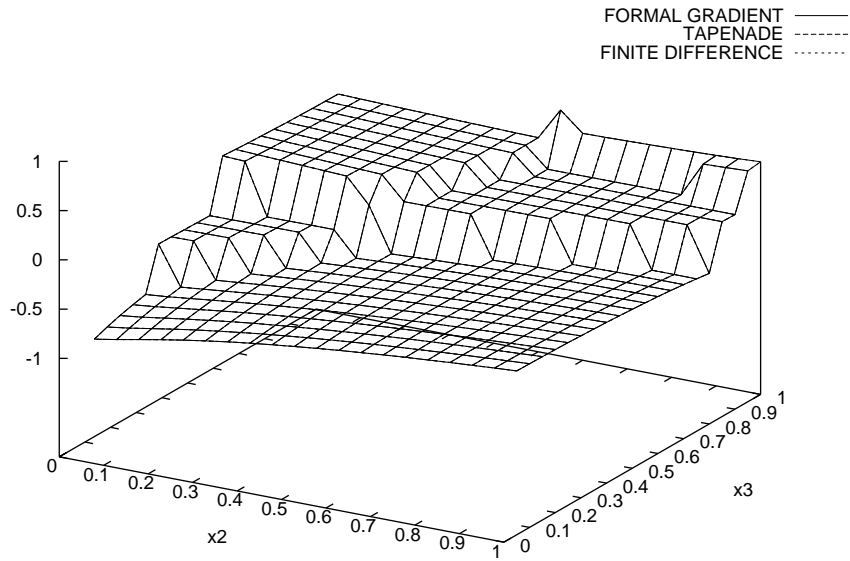


FIGURE 5.1 – Gradient du critère  $J_2 : \frac{\partial J_2}{\partial x_3}$

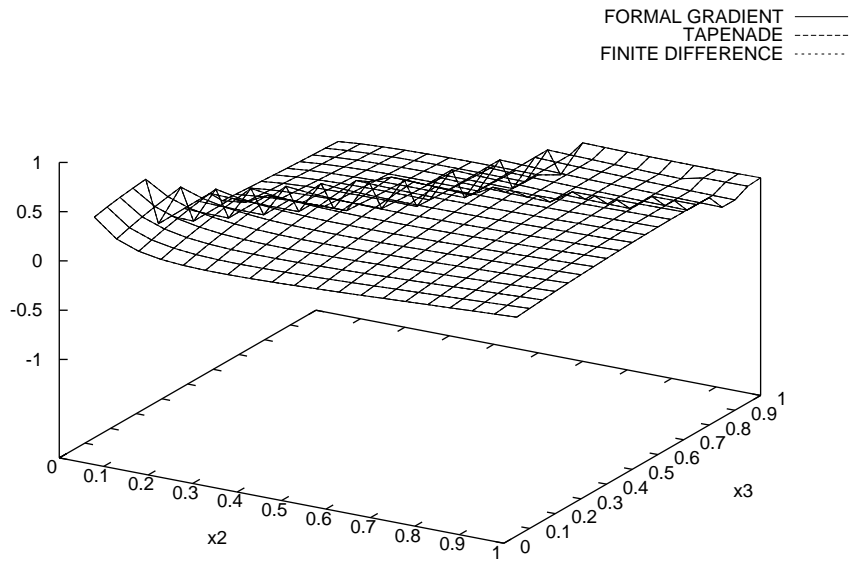


FIGURE 5.2 – Gradient du critère  $J_2 : \frac{\partial J_2}{\partial x_2}$

### 5.2.3.2 Auto-adaptation de la paramétrisation

On fait appel maintenant au gradient  $\partial J_2/\partial \xi$ , qui a été validé dans la section précédente, dans la procédure d'adaptation de paramétrisation. On utilise dans ce cas test un optimiseur de type gradient (FFSQP) [121] pour minimiser le critère de régularisation  $J_2$ .

La paramétrisation de Bézier initiale est une approximation par moindre carrée de la courbe cible (5.83) associée à un support de degré 12, obtenu après 9 élévations du degré à partir de  $(0, 0, 1/3, 1)$ . Le polygone correspondant est celui qui présente des irrégularités comme dans la figure (5.3a). La courbe résultante, après la procédure d'adaptation, est superposée à la courbe initiale, tandis que le polygone associé est lisse. La figure (5.3b) indique la convergence itérative du critère  $J_2$ .

Enfin, on mesure le gain réalisé par l'auto-adaptation de la paramétrisation. On utilise le problème modèle, exposé dans le chapitre précédent, dans le cas particulier où  $\omega(t) = 1$  et  $\alpha = 2$ , la fonctionnelle est convexe et la forme optimale est un arc de cercle  $(x - \frac{1}{2})^2 + y^2 = \frac{1}{4}$ . On considère un couplage optimisation-adaptation répété séquentiellement, pour trois degrés de paramétrisation ( $n = 3, 6, 12$ ), et une condition initiale qui correspond à la paramétrisation (5.84).

Dans la figure(5.4a), on constate que le minimum atteint par chacune des optimisations, à partir d'une même forme initiale, est de plus en plus faible quand le degré  $n$  augmente. Par contre, après une suite de couplages optimisation-adaptation, le minimum atteint devient presque indépendant du degré de la paramétrisation. On peut donc atteindre avec un nombre petit de paramètres, par ce processus, des résultats similaires au cas d'une paramétrisation de degré élevé. Néanmoins, le minimum théorique ( $2\pi$ ) n'est atteint par aucune des optimisations. Ceci est dû généralement à la raideur numérique du système quand le degré de la paramétrisation augmente. Pour remédier à cela, on couple l'optimisation et l'adaptation dans un processus hiérarchique. On effectue un enrichissement progressif de paramétrisation à trois niveaux ( $n = 3, 6, 12$ ) en prenant comme condition initiale la solution obtenue par le couplage optimisation-adaptation associé à la paramétrisation grossière de degré 3 (voir figure(5.4a)). On se rapproche du minimum théorique avec cette nouvelle stratégie hiérarchique, comme le montre la figure (5.4b), qui est visiblement efficace et moins coûteuse que la méthode basique associée à la paramétrisation de degré 12.

On montre dans la figure (5.5) les formes obtenues par une optimisation pure (sans adaptation) pour trois degrés de paramétrisations ( $N=3, 6$  et  $12$ ). Quant à la figure (5.6), on montre les formes qui résultent d'un couplage optimisation-adaptation pour les mêmes degrés de paramétrisations. On constate que la forme associée à une paramétrisation grossière (deux paramètres d'optimisation) dans une optimisation couplée avec l'adaptation est significativement proche de la forme associée à une paramétrisation plus fine (10 paramètres d'optimisation).

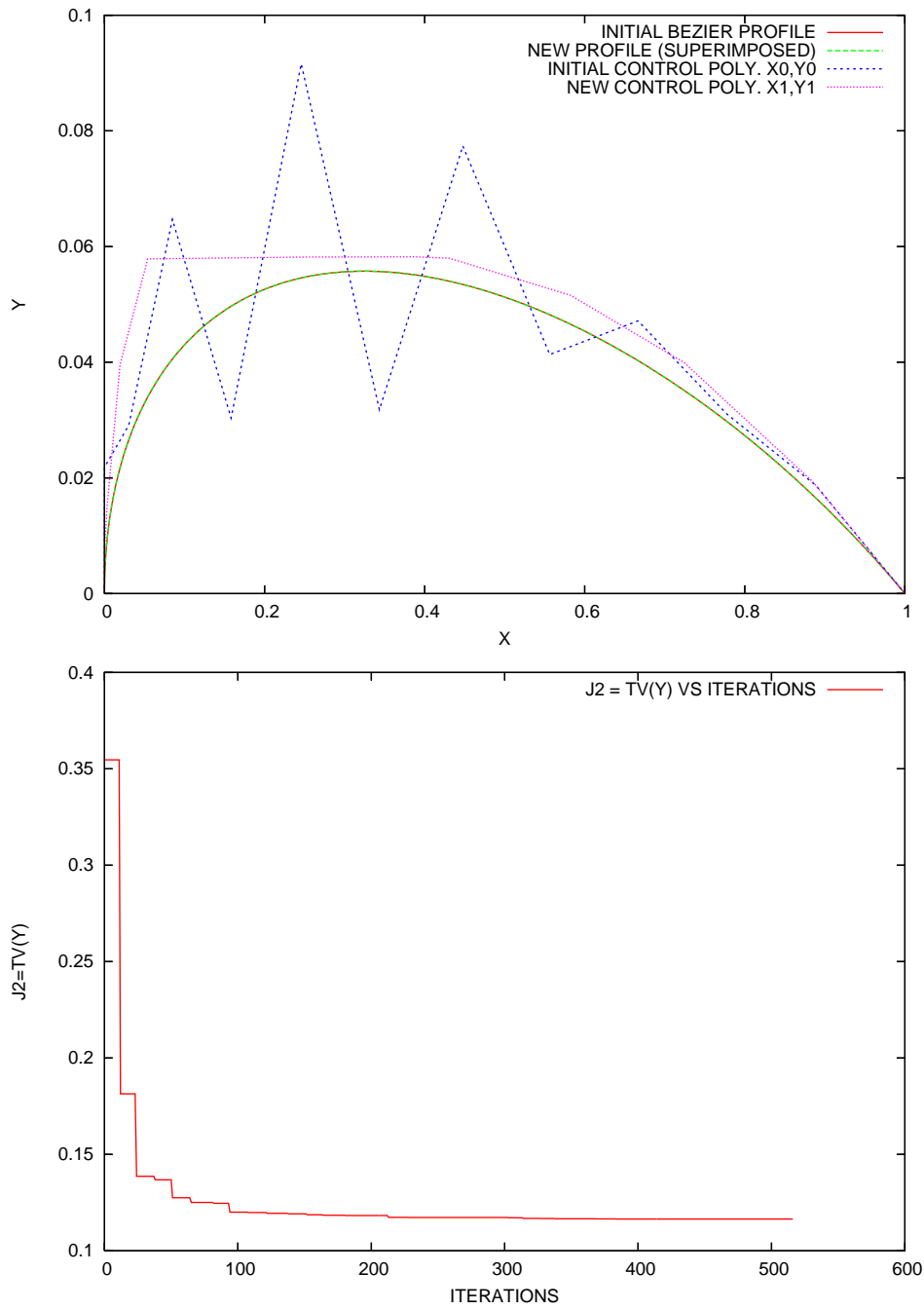


FIGURE 5.3 – Adaptation de la paramétrisation : (a) en haut : Effet de la régularisation de la paramétrisation de Bézier par adaptation, (b) en bas : la convergence itérative du critère  $J_2$

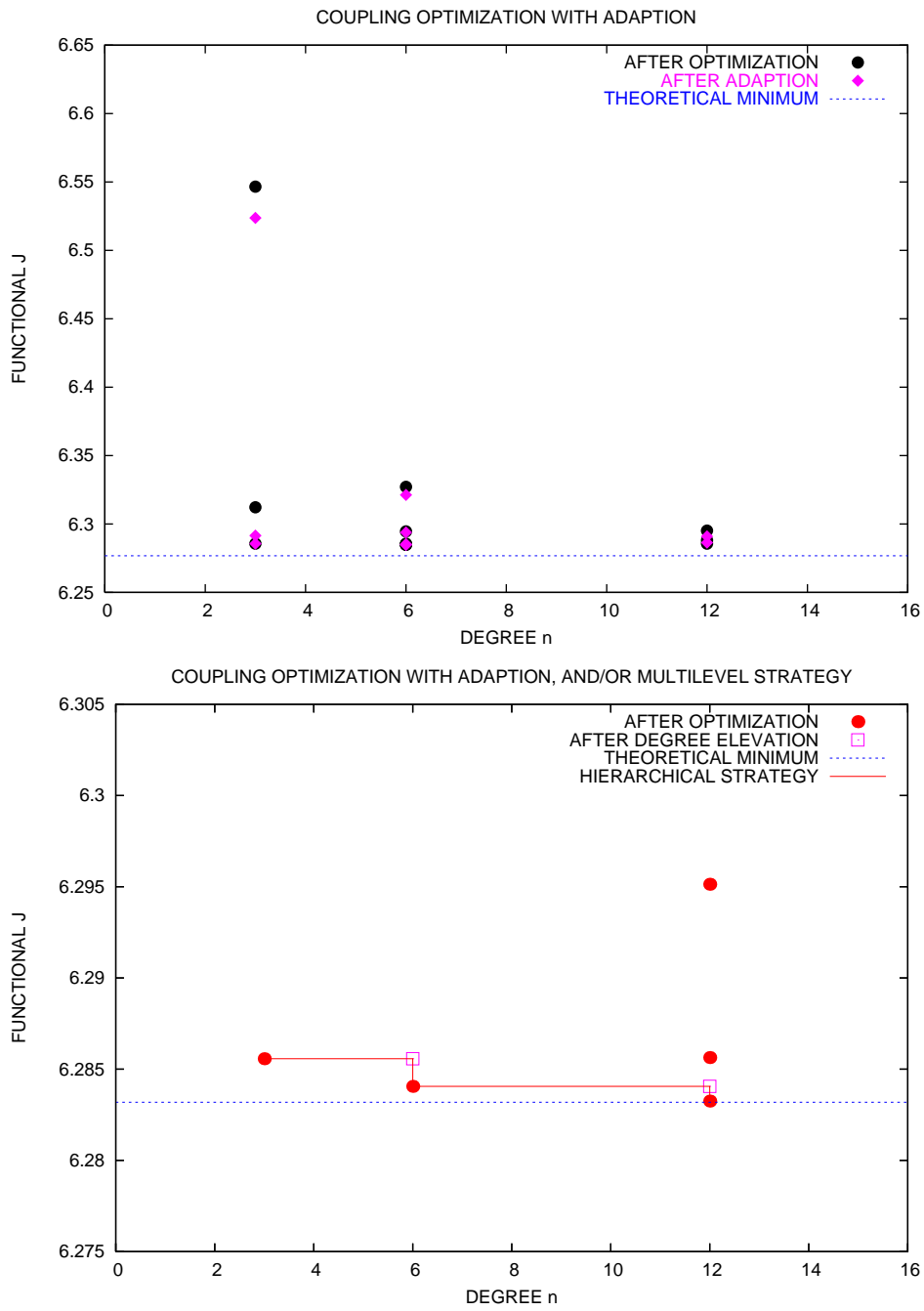


FIGURE 5.4 – (a) en haut : Couplage optimisation-adaptation, (b) en bas : (a) + une stratégie hiérarchique

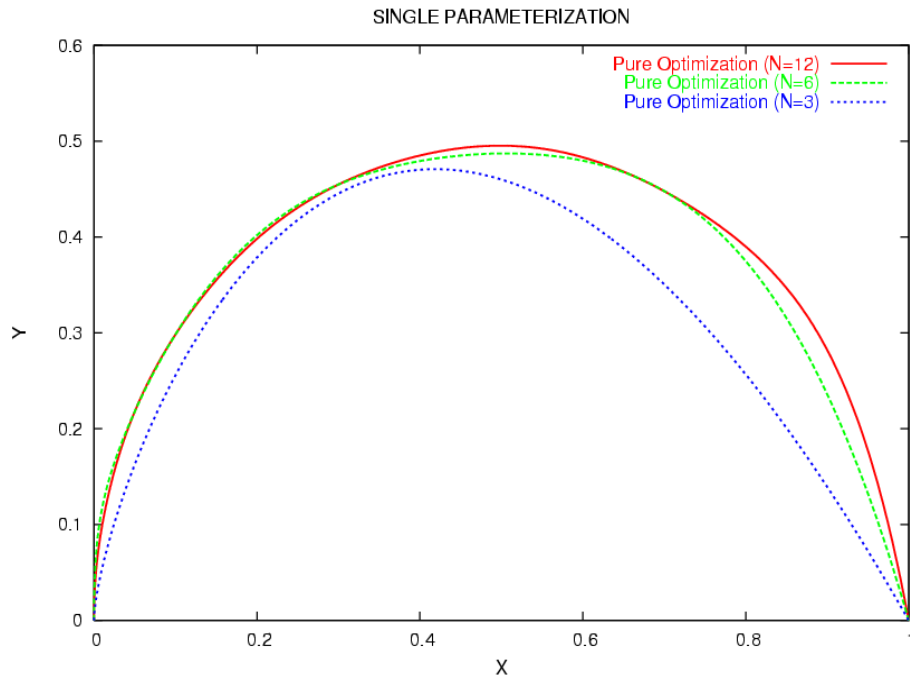


FIGURE 5.5 – Optimisation pure (sans adaptation) pour  $N=3, 6$  et  $12$

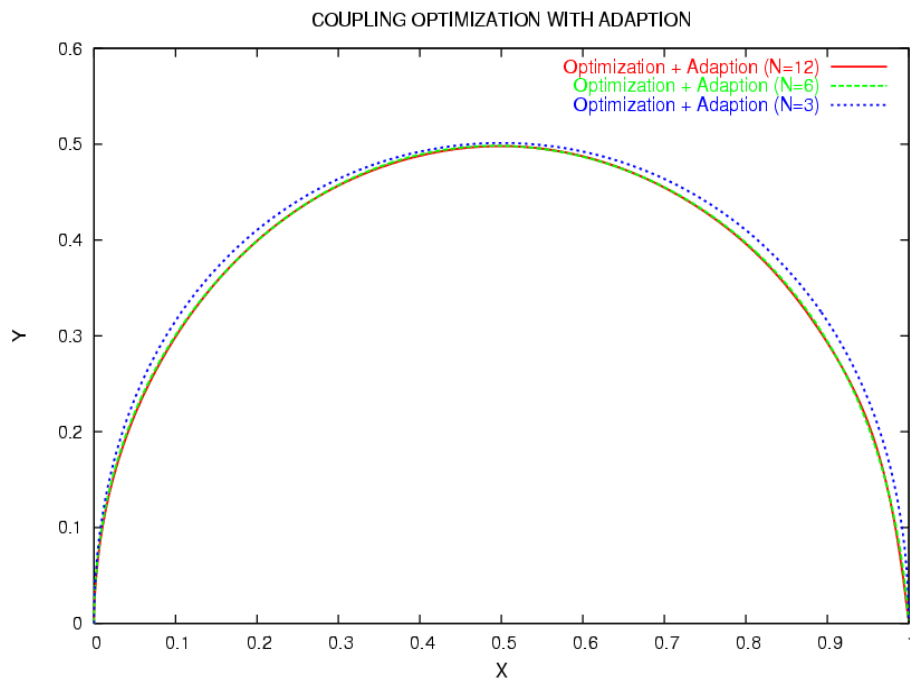


FIGURE 5.6 – Couplage optimisation-adaptation : pour  $N=3, 6$  et  $12$ , et avec une stratégie hiérarchique ( $N=3 \rightarrow 6 \rightarrow 12$ )

### 5.3 Approche 3D : cas d'une paramétrisation par Free-Form Deformation

On a présenté dans la section précédente une procédure d'adaptation de paramétrisation qui consiste à régulariser le polygone de contrôle au cours du processus itératif en minimisant un critère de variation totale. Les résultats numériques montrent combien cette notion d'adaptation de paramétrisation est efficace pour réduire la raideur du système. Cette première approche utilise une représentation classique de la géométrie par la paramétrisation de Bézier.

On cherche maintenant à étendre ce concept d'adaptation de paramétrisation dans le cas d'une représentation de la géométrie par Free-Form deformation (FFD). Cette dernière consiste à déformer l'espace contenant la forme plutôt que la forme elle-même. Il s'agit donc, par analogie avec l'approche précédente, de régulariser la boîte FFD (ou le volume de contrôle) qui devient irrégulière au cours du processus d'optimisation. Deux approches seront présentées ; la première utilise le support du volume de contrôle, qui représente la forme, comme paramètre d'adaptation, et la seconde se base sur une propriété simple et intéressante expliquée par Sederberg et Parry [101]. Tous les deux ont pour objectif la régularisation du volume de contrôle (ou la boîte englobant la forme) en changeant la carte FFD qui représente les coordonnées locales des noeuds du maillage qui sont à l'intérieur de la boîte FFD.

On rappelle ici brièvement la technique de Free-form deformation en 3D. Soit  $q$  un noeud du maillage, on pose :

$$T_q = S_q + \sum_{i=0}^n \sum_{j=0}^m \sum_{k=0}^l B_i^n(u_q) B_j^m(v_q) B_k^l(w_q) \delta P_{ijk} \quad (5.85)$$

où  $T_q = (x_q, y_q, z_q)$  (resp.  $S_q = (x_q^0, y_q^0, z_q^0)$ ) représente les coordonnées de  $q$  dans le maillage courant (resp. dans la configuration initiale),  $(u_q, v_q, w_q)$  représente les coordonnées locales de  $q$  dans la boîte englobant la forme, et  $\delta P_{ijk} = (\delta X_{ijk}, \delta Y_{ijk}, \delta Z_{ijk})$  sont les paramètres de conception.

On introduit les notations suivantes :

$$\delta U^T = (\delta U_{ijk})_{0 \leq i \leq n, 0 \leq j \leq m, 0 \leq k \leq l} \quad (5.86)$$

avec  $U = P, X, Y, Z$

$$R/C = \{R(u_q, v_q, w_q) / (u_q, v_q, w_q) \in C\}$$

avec,

$$R(u_q, v_q, w_q) = (B_i^n(u_q) B_j^m(v_q) B_k^l(w_q))_{0 \leq i \leq n, 0 \leq j \leq m, 0 \leq k \leq l} \quad (5.87)$$

pour tout noeud  $q$  du maillage.

$C$  est la carte FFD, qui représente l'ensemble des coordonnées locales  $(u_q, v_q, w_q)$  de  $q$  dans la boîte englobant la forme.

### 5.3.1 Première approche

#### 5.3.1.1 Procédure d'adaptation

On rappelle que notre stratégie d'optimisation consiste à bouger les paramètres de contrôle seulement dans la direction verticale ; ce qui implique que  $\delta X = \delta Z = 0$ . On se ramène alors au système suivant :

$$\begin{cases} x = x^0 = R/C X^0 \\ y - y^0 = R/C \delta Y \\ z = z^0 = R/C Z^0 \end{cases} \quad (5.88)$$

où  $(x, y, z)$  (resp.  $(x^0, y^0, z^0)$ ) sont les noeuds du maillage courant (resp. du maillage initial), qui sont à l'intérieur de la boîte FFD, et  $C$  est la carte FFD associée.

Au cours du processus d'optimisation, le vecteur  $\delta Y$ , qui désigne le vecteur de design, présente des irrégularités ; la boîte de paramétrisation devient par conséquent irrégulière. Il s'agit donc de choisir judicieusement la carte FFD, qui permet de déterminer le positionnement des points de contrôle autour de la forme, afin de régulariser la boîte englobant la forme. Notre procédure d'adaptation, qui utilise le support de la paramétrisation comme paramètre d'adaptation, consiste à calculer, pour chaque choix de support, une nouvelle carte, et ensuite minimiser un critère de variation totale sous la contrainte que la forme résultante soit une bonne approximation de la forme cible. On trouve dans [38], une autre approche d'adaptation de paramétrisation qui utilise plutôt la carte FFD comme paramètre d'adaptation. On présente par la suite l'algorithme général d'optimisation couplée avec l'adaptation, et en détail la procédure d'adaptation de paramétrisation.

#### Algorithme de couplage optimisation-adaptation :

1. **Initialisation** : On considère une forme initiale  $F_{C_0}^{(0)}$  associée au volume de contrôle  $P_{C_0}^{(0)} = (X^0, Y^0, Z^0)$  ( $\delta P_{C_0}^{(0)} = (\delta X^0, \delta Y^0, \delta Z^0) = 0$ ) qui correspond à une carte initiale  $C_0$ .
2. **Couplage optimisation-adaptation** :  $k \leftarrow 0$
3. **Optimisation de forme** : Après  $K$  itérations, on obtient la forme optimale  $F_{C_0}^{(K)} = (x^{(K)}, y^{(K)}, z^{(K)})$  associée au volume de contrôle  $P_{C_0}^{(K)} = (X^0 + \delta X^0, Y^0 + \delta Y^{(K)}, Z^0 + \delta Z^0)$ . On note que  $\delta X^0$  et  $\delta Z^0$  sont nuls durant l'optimisation.
4. **Adaptation de la paramétrisation** : On cherche un nouveau support  $X^{(K)}$  correspondant à une nouvelle carte  $C_{ad}$ , en minimisant un critère qui mesure l'irrégularité de  $\delta Y^{(K)}$ , sous la contrainte que la forme obtenue soit une meilleure approximation de la forme cible  $F_{C_0}^{(K)}$ .
5. **Réinitialisation** : On réinitialise alors les paramètres de déformation pour servir comme condition initiale à une nouvelle optimisation.

$$\begin{cases} C_0 \leftarrow C_{ad} \\ \delta X^0 \leftarrow \delta X_{ad}^{(K)} \\ \delta Y^0 \leftarrow \delta Y_{ad}^{(K)} \end{cases}$$

$k \leftarrow k + 1$   
retour à (3)

**Adaptation du support :** Au cours du processus d'optimisation, on maintient fixe les deux supports de la paramétrisation, qu'on note initialement par  $X^0$  et  $Z^0$ , et on bouge seulement les paramètres d'optimisation selon l'axe des ordonnées. Or, tous les points de contrôle dans la direction  $z$  (uniquement les sommets de la boîte FFD) sont fixes ; alors le support  $Z^0$  restera fixe après l'adaptation.

Dans ce cas particulier, on prend seulement le support de la paramétrisation comme paramètre d'adaptation. La nouvelle carte  $C$  est liée au changement de support ( $X^0 \rightarrow X$ ) par la condition :

$$x^{(K)} = \tilde{R} X \quad (5.89)$$

où  $\tilde{R} = R_{/C}$  avec  $C = \{(\vec{u}, \vec{v}, \vec{w}), \vec{u} \text{ et } \vec{w} \text{ sont fixe}\}$

Le vecteur  $\vec{v}$  est calculé en résolvant le système des équations (5.89) par la méthode de Newton.

Mais dans le cas général où le support  $Z_0$  n'est pas fixe, la nouvelle carte FFD sera obtenu en résolvant le système des équations suivant :

$$\begin{cases} x^{(K)} = \tilde{R} X \\ z^{(K)} = \tilde{R} Z \end{cases}$$

où  $\tilde{R} = R_{/C}$  avec  $C = \{(\vec{u}, \vec{v}, \vec{w}), \vec{u} \text{ et/ou } \vec{w} \text{ sont fixe}\}$

**Contrainte de meilleure approximation :** Étant donnée une nouvelle carte FFD, qu'on désigne par  $C_{ad}$ , obtenue après adaptation du support de la paramétrisation. On cherche à calculer un nouveau vecteur  $\delta Y^{(K)}$  tel que :

$$\Delta y = y^{(K)} - y^{(0)} = \bar{R} \delta Y^{(K)} \quad (5.90)$$

avec  $\bar{R} = R_{/C_{ad}}$

En multipliant l'équation (5.90) par  $\bar{R}^T$ , ce qui est équivalent à une projection, on obtient :

$$\bar{R}^T \Delta y = \bar{R}^T \bar{R} \delta Y^{(K)} \quad (5.91)$$

Or la matrice carrée  $\bar{R}^T \bar{R}$  est symétrique définie positive, alors :

$$\delta Y^{(K)} = \left( \bar{R}^T \bar{R} \right)^{-1} \bar{R}^T \Delta y \quad (5.92)$$



**Critère de régularisation :** Dans l'approche précédente, on a choisie un critère de variation totale comme fonctionnelle d'adaptation, qui consiste à lisser le polygone de contrôle. Or en 3D, le volume de Bézier n'est qu'une extension directe des surfaces de Bézier, et aussi des courbes de Bézier. Autrement dit, un volume de Bézier est un ensemble de courbe de Bézier. Ce constat nous amène à choisir une fonctionnelle d'adaptation comme étant la somme des variations totales associées aux polygones composant le volume de contrôle. Donc, si on pose :

$$\delta Y^{(K)} = \bigcup_{l=1}^N \delta Y_l^{(K)} \quad (5.93)$$

avec  $N$  est le nombre des polygones de contrôle.

La fonctionnelle de régularisation s'écrit comme suit :

$$\mathcal{J}_{reg} = \sum_{l=1}^N TV(\delta Y_l^{(K)}) \quad (5.94)$$

Ce qui se traduit par le fait que la régularisation du volume de contrôle revient, tout simplement, à lisser les polygones de contrôle qui le compose.

### 5.3.1.2 Résultats numériques

on utilise le cas test, présenté dans le chapitre précédent, qui consiste à optimiser une voilure d'avion d'affaire en régime transsonique. On rappelle que le but de l'optimisation consiste à réduire le coefficient de la traînée  $C_D$  avec une contrainte sur la portance  $C_L$  en minimisant la fonctionnelle coût suivante :

$$J_A = \frac{C_D}{C_{D_0}} + 10^4 \cdot \max\left(0, 0.999 - \frac{C_L}{C_{L_0}}\right) \quad (5.95)$$

avec  $C_{D_0}$  et  $C_{L_0}$  sont respectivement la traînée et la portance de la configuration initiale.

On choisit une paramétrisation de degré  $3 \times 1 \times 1$  qui correspond à 8 paramètres d'optimisation. Le but est de montrer l'efficacité de l'adaptation de paramétrisation dans la boucle d'optimisation. Pour cela on compare deux types de stratégies : une optimisation pure (sans adaptation), et une optimisation couplée avec la procédure d'adaptation.

On montre dans la figure (5.7) les courbes de convergence des deux stratégies en utilisant la méthode du simplexe de Nelder-Mead. La procédure d'adaptation intervient après 100 itérations, suivie d'un autre cycle d'optimisation avec une nouvelle carte FFD et une nouvelle paramétrisation régulière. On constate que la procédure d'adaptation, qui a un coût marginal, permet d'accélérer la convergence itérative de l'optimisation. Ceci est dû au choix judicieux d'une nouvelle carte FFD qui permet de repositionner les points de contrôle d'une façon régulière. Ce constat est illustré par la figure (5.8) qui montre que la boîte FFD après adaptation présente moins d'irrégularités que celle avant adaptation. Ce passage est dû à la minimisation d'un critère de régularisation par la méthode du simplexe de Nelder-Mead ; on montre dans la figure (5.9) la courbe de convergence du critère  $\mathcal{J}_{reg}$ .

La table (5.1) donne une comparaison entre les valeurs des coefficients aérodynamiques et de la fonction objectif associés à la solution optimisée par les deux stratégies. On constate que la contrainte sur la portance a été bien respectée, et que le coefficient de la traînée a été réduit ; c'est ce qui explique le gain réalisé d'environ 3% en fonctionnelle coût. Cette amélioration se traduit par la réduction de l'intensité de choc sur l'extrados de l'aile, c'est ce qu'on peut voir dans les figures (5.10) et (5.12) qui montrent respectivement le champ de Mach sur la surface de l'aile et la distribution de pression à l'extrémité, à 50% de l'envergure, et à l'emplanture de l'aile pour la configuration initiale et les stratégies avec et sans adaptation. Les formes associées aux mêmes sections sont données par la figure (5.11).

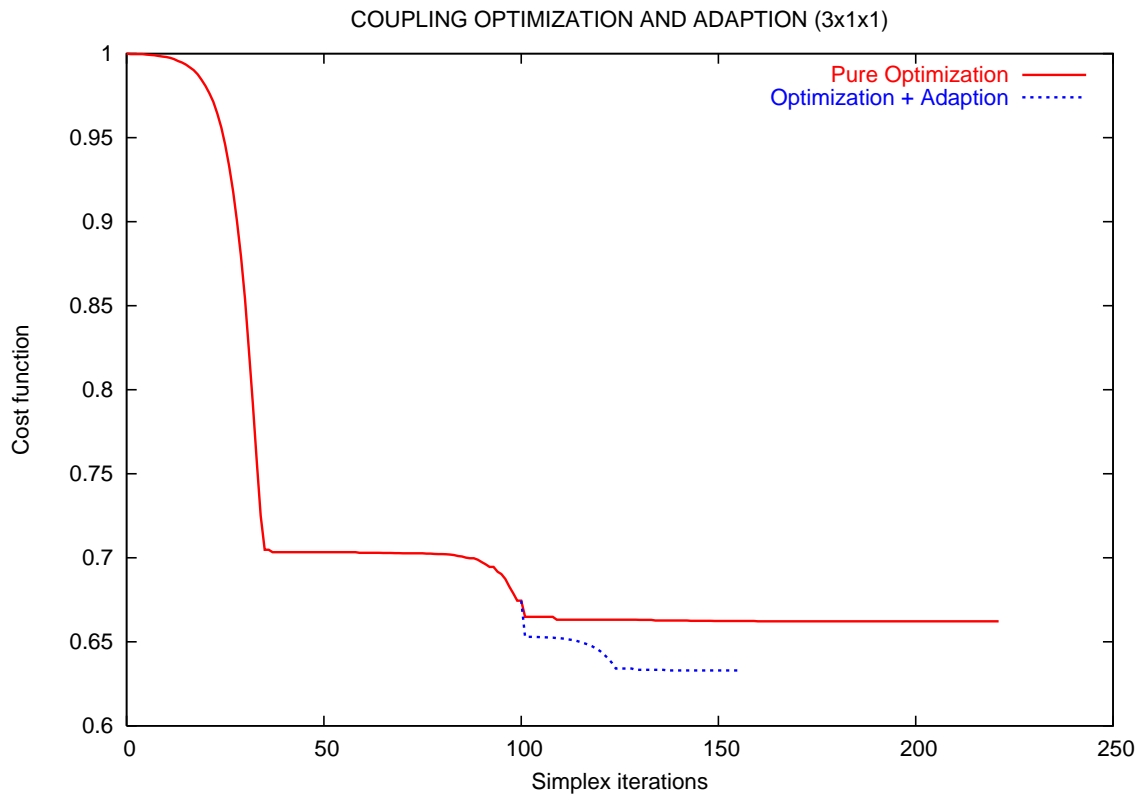


FIGURE 5.7 – Courbes de convergence : optimisation pure et optimisation adaptative

	C_L	C_D	Fonction coût
Config. initiale	0.319192893	0.026352608	1.
Optimisation Pure	0.318874966	0.017450289	0.662184501
Optimisation + Adaptation	0.319485647	0.0167084668	0.632927111

TABLE 5.1 – Comparaison entre les coefficients aérodynamiques obtenues par les deux stratégies avec et sans adaptation

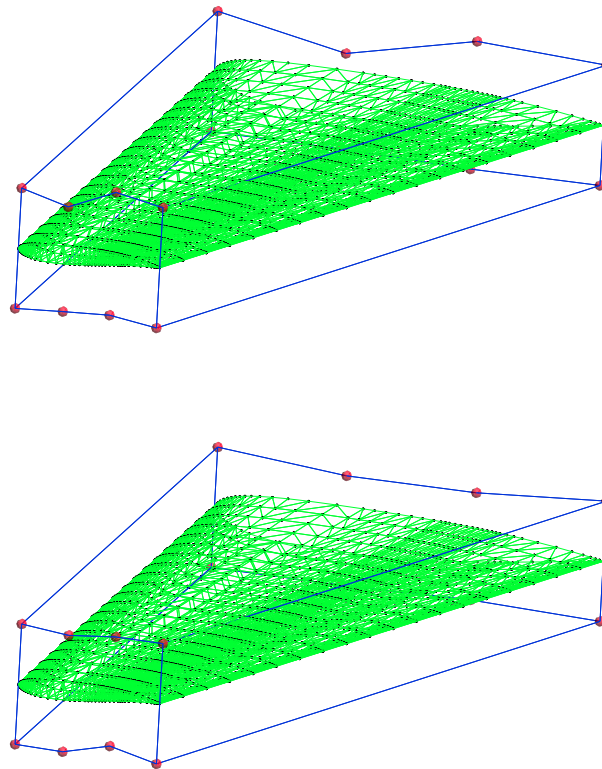


FIGURE 5.8 – Régularisation de la boîte FFD par adaptation après 100 itérations : avant adaptation (en haut), après adaptation (en bas)

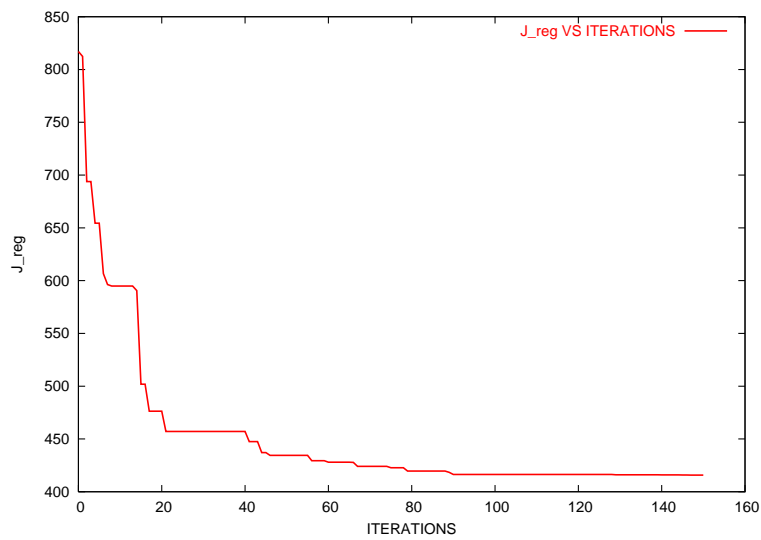


FIGURE 5.9 – Courbe de convergence du critère de régularisation

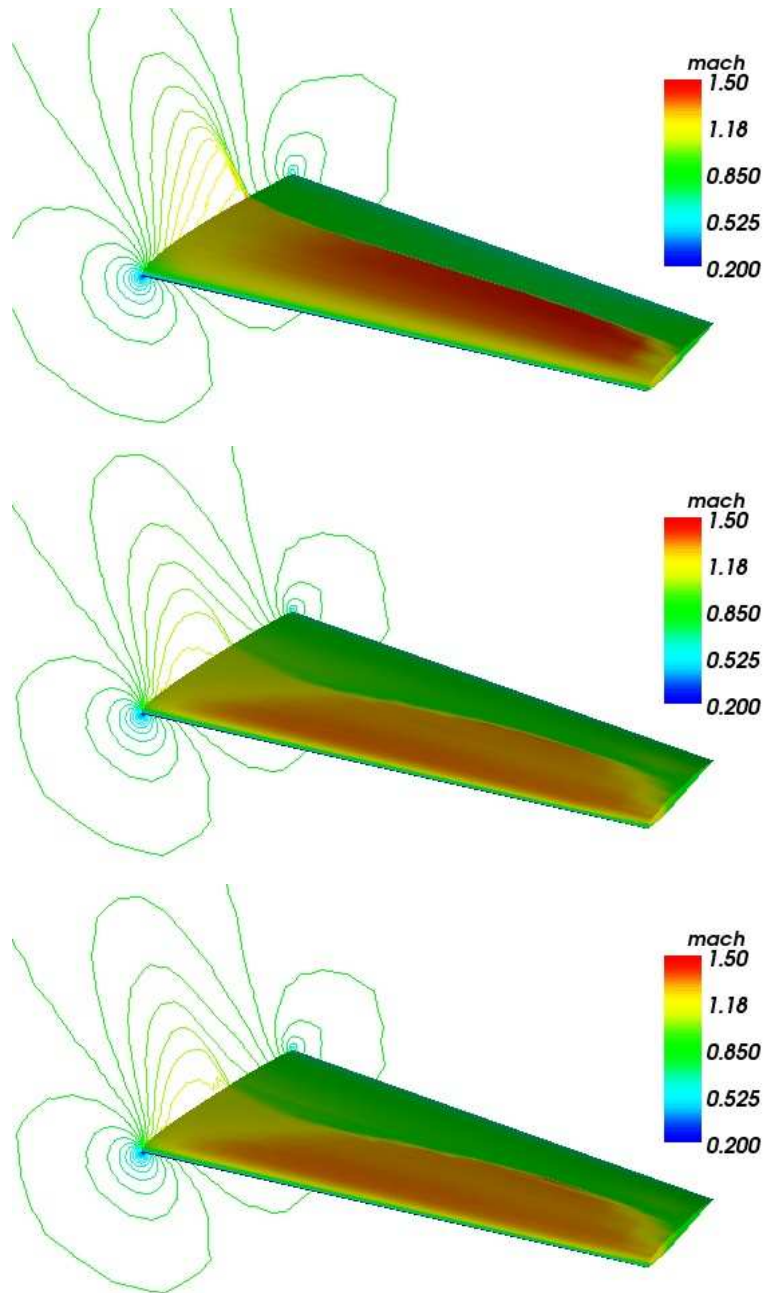


FIGURE 5.10 – Champ de Mach sur la surface de l’aile : configuration initiale (en haut), stratégie sans adaptation (au milieu), et stratégie avec adaptation (en bas)

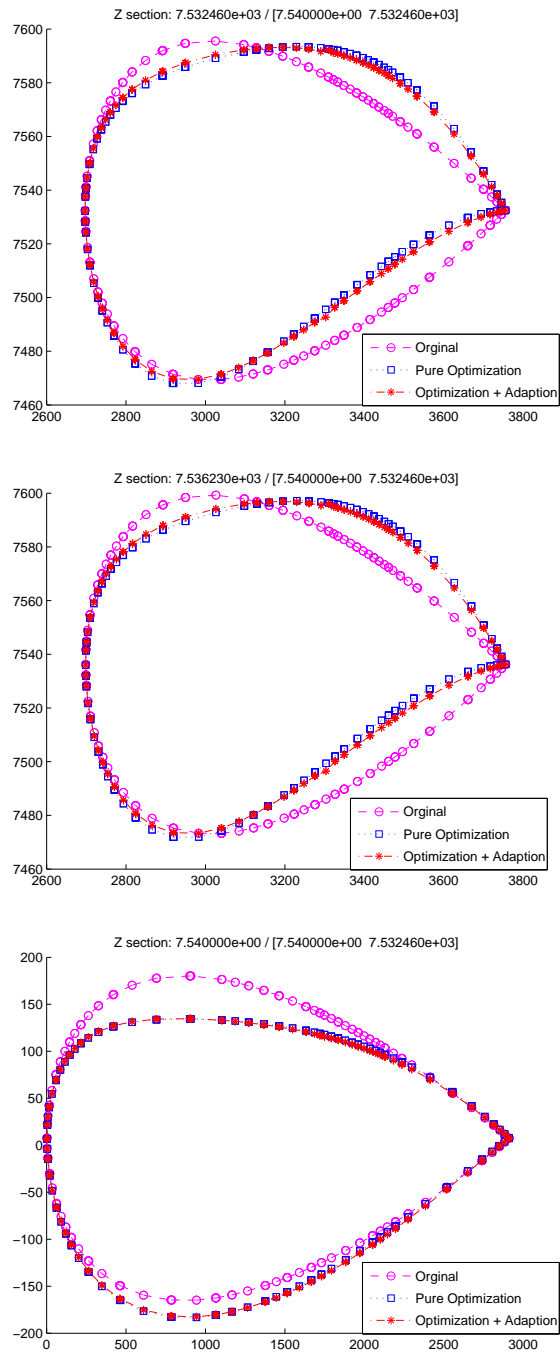


FIGURE 5.11 – Comparaison entre les formes, pour la configuration initiale et les deux stratégies d'optimisation, à trois sections de l'aile : l'extrémité (en haut), 50% de l'envergure (au milieu), et l'emplanture (en bas)

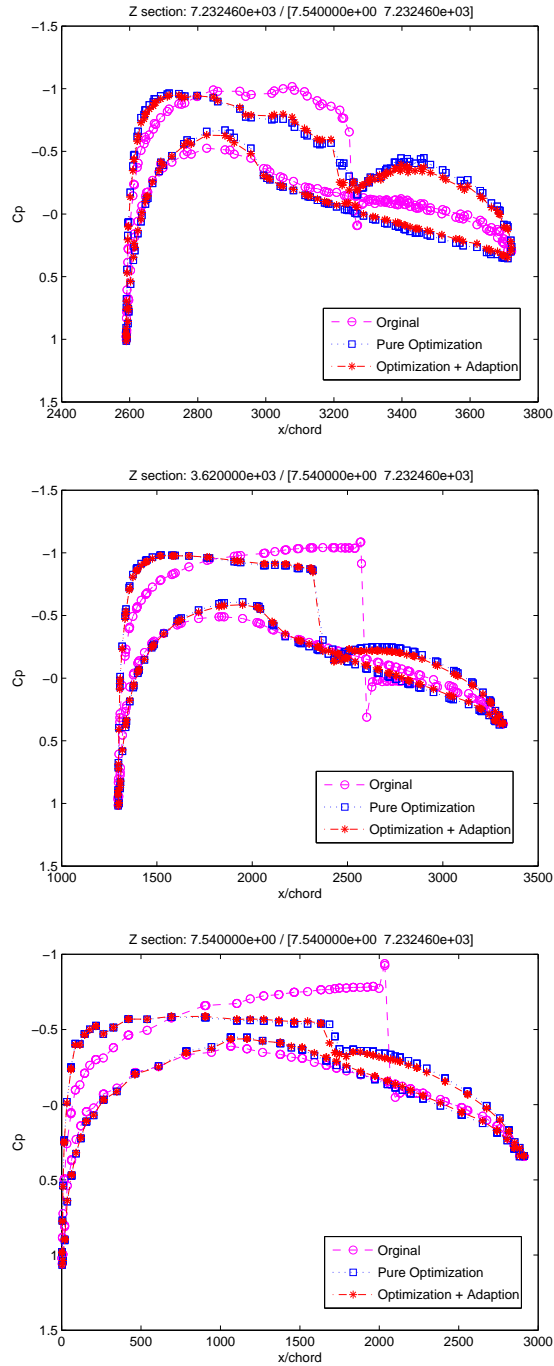


FIGURE 5.12 – Comparaison entre les coefficients de pression, pour la configuration initiale et les deux stratégies d’optimisation, à trois sections de l’aile : l’extrémité (en haut), 50% de l’envergure (au milieu), et l’emplanture (en bas)

## 5.3.2 Deuxième approche

### 5.3.2.1 Procédure d'adaptation

On présente maintenant une autre approche d'adaptation de paramétrisation qui consiste à faire un "restart" de la paramétrisation en utilisant une propriété intéressante de la méthode Free-Form Deformation [101], et qu'on peut formuler par la proposition suivante dont la démonstration est triviale :

#### Proposition 5.3.1

On considère une boîte définie par deux sommets  $(X_{min}, Y_{min}, Z_{min})$  et  $(X_{max}, Y_{max}, Z_{max})$ .

Soit  $Q$  un noeud arbitraire de coordonnées  $(x, y, z)$  à l'intérieur de la boîte.

Alors, il existe un volume de contrôle  $(P_{ijk})_{0 \leq i \leq n, 0 \leq j \leq m, 0 \leq k \leq l}$  défini par :

$$P_{ijk} = \left( X_{min} + \frac{i}{n}(X_{max} - X_{min}), Y_{min} + \frac{j}{m}(Y_{max} - Y_{min}), Z_{min} + \frac{k}{l}(Z_{max} - Z_{min}) \right) \quad (5.96)$$

tel que :

$$Q(u, v, w) = \sum_{i=0}^n \sum_{j=0}^m \sum_{k=0}^l B_i^n(u) B_j^m(v) B_k^l(w) P_{ijk} \quad (5.97)$$

avec  $(u, v, w)$  sont les coordonnées locale de  $Q$  dans la boîte :

$$u = \frac{x - X_{min}}{X_{max} - X_{min}}, \quad v = \frac{y - Y_{min}}{Y_{max} - Y_{min}}, \quad w = \frac{z - Z_{min}}{Z_{max} - Z_{min}} \quad (5.98)$$

On utilisant cette proposition dans le contexte de l'optimisation de forme aérodynamique, on peut définir, après une phase d'optimisation, une carte FFD adéquate qui permet de repositionner les points de contrôle d'une façon régulière autour de la forme. En effet, soit  $C_0$  la carte FFD initiale, qui est composée par les coordonnées locales  $(u^0, v^0, w^0)$  des noeuds du maillage initiale  $\mathcal{M}_{C_0}^{(0)}$ , et soit  $\mathcal{M}_{C_0}^{(K)}$  le maillage à l'intérieur de la boîte FFD qui est obtenu après  $K$  itérations d'optimisation, et qui correspond au volume de contrôle  $P_{C_0}^{(K)} = (P_{ijk}^0)_{0 \leq i \leq n, 0 \leq j \leq m, 0 \leq k \leq l}$ .

Alors, d'après la proposition précédente, il existe, pour tout noeud  $Q$  du maillage courant (ou cible)  $\mathcal{M}_{C_0}^{(K)}$ , un nouveau polygone de contrôle  $(P_{ijk})_{0 \leq i \leq n, 0 \leq j \leq m, 0 \leq k \leq l}$  régulier et une nouvelle carte FFD, composée par les coordonnées  $(u, v, w)$  qui sont données par la formule (5.98), telle que :

$$\begin{aligned} Q &= \sum_{i=0}^n \sum_{j=0}^m \sum_{k=0}^l B_i^n(u^0) B_j^m(v^0) B_k^l(w^0) P_{ijk}^0 \\ &= \sum_{i=0}^n \sum_{j=0}^m \sum_{k=0}^l B_i^n(u) B_j^m(v) B_k^l(w) P_{ijk} \end{aligned} \quad (5.99)$$

On illustre dans la figure (5.13) ce processus d'adaptation de paramétrisation ; les deux polygones de contrôle sont clairement de régularités différentes, puisqu'ils sont associés à deux différentes cartes FFD, et correspondent à la même forme.

Cette approche, contrairement à la précédente, est simple et facile à implémenter. L'adaptation de paramétrisation au cours de l'optimisation se fait exactement sans perte d'informations, puisqu'on ne fait recours à aucune approximation. Cette propriété intéressante, formulée par la proposition précédente, propose une autre manière pour effectuer des transferts de paramétrisations d'un niveau à un autre, puisque le processus permet de construire des boîtes englobant la forme avec autant de points de contrôle qu'on veut pour un maillage donné. Le transfert de paramétrisation se fait donc par insertion avec changement de la carte FFD. Quant au processus d'élévation du degré, que nous avons utilisé dans nos stratégies multiniveaux (voir chapitre (3)), le transfert s'effectue indépendamment de la carte FFD en faisant une combinaison convexe des points de contrôle.

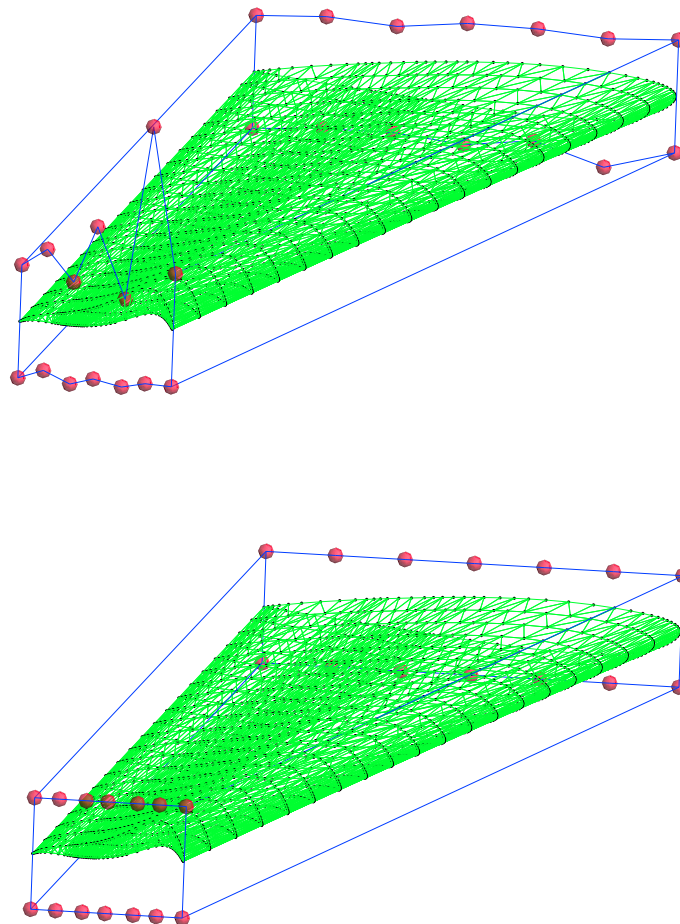


FIGURE 5.13 – Effet de la régularisation de la boîte FFD par adaptation : (en haut) avant adaptation, (en bas) après adaptation

### 5.3.2.2 Résultats numériques

On utilise ici le même cas test qu'avant sur l'optimisation d'une voilure d'avion d'affaire en régime transsonique qui a comme objectif la réduction de la traînée avec une contrainte



sur la portance.

On compare dans tous les tests qui suivent deux types de stratégies :

1. Une optimisation pure (sans adaptation) convergée complètement.
2. Une optimisation adaptative qui fait appel à la procédure d'adaptation dans la boucle d'optimisation une ou plusieurs fois.

On considère trois types de paramétrisations : une paramétrisation grossière de degré  $3 \times 1 \times 1$  qui correspond à 8 degrés de liberté (d.d.l), une paramétrisation moyenne de degré  $6 \times 1 \times 1$  qui correspond à 20 d.d.l, et une paramétrisation fine de degré  $9 \times 1 \times 1$  qui correspond à 32 d.d.l La méthode du simplexe de Nelder-Mead a été utilisée pour la minimisation de la fonctionnelle coût.

**Optimisation avec une paramétrisation grossière ( $3 \times 1 \times 1$ ) :** Dans ce cas test, on fait intervenir la procédure d'adaptation après chaque 50 itérations dans la stratégie d'optimisation adaptative. On montre dans la figure (5.14) les courbes de convergence pour chaque optimisation ; on constate que l'optimisation adaptative, avec 12 adaptations de paramétrisation, permet d'accélérer la convergence itérative significativement. On compare dans la table (5.2) les coefficients aérodynamiques ainsi que les valeurs de la fonctionnelle coût à l'optimum de chaque stratégie. Ces résultats sont illustrés dans la figure (5.15) qui donne le champ de Mach sur la surface de l'aile optimisée par les deux stratégies en comparaison avec la configurations initiale ; Une réduction de la zone supersonique peut être observée pour les deux stratégies avec un avantage pour l'optimisation adaptative. Quant aux figures (5.16) et (5.17), on montre respectivement les formes et les coefficients de pression, pour la configuration initiale et les stratégies avec et sans adaptation, à trois sections de l'aile : l'extrémité, 50% de l'envergure, et l'emplanture.

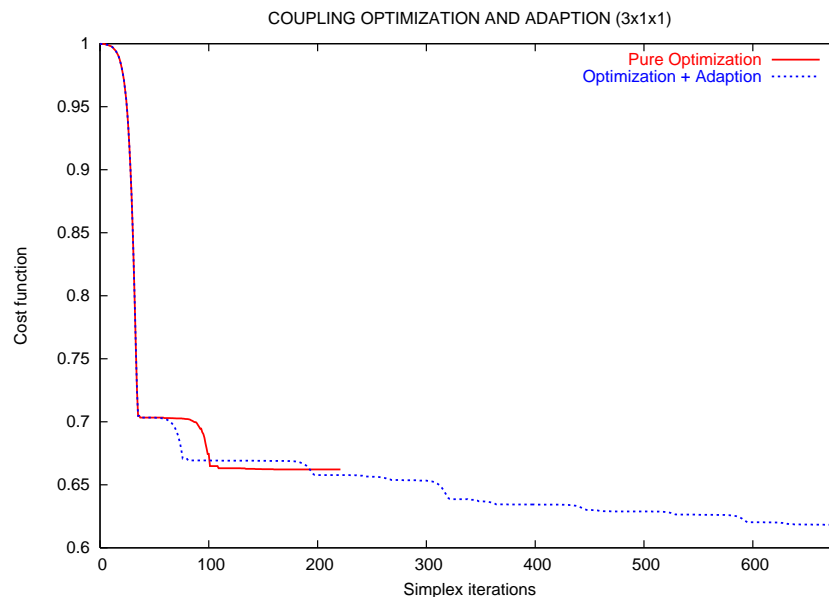


FIGURE 5.14 – Courbes de convergence : optimisation pure (jusqu'à convergence) et optimisation adaptative où l'adaptation est intervenue après chaque 50 itérations

	C_L	C_D	Fonction coût
Config. initiale	0.319192893	0.026352608	1.
Optimisation Pure	0.318874966	0.017450289	0.662184501
Optimisation + Adaptation	0.318999078	0.016299483	0.618515468

TABLE 5.2 – Comparaison entre les coefficients aérodynamiques obtenus par les deux stratégies d'optimisation

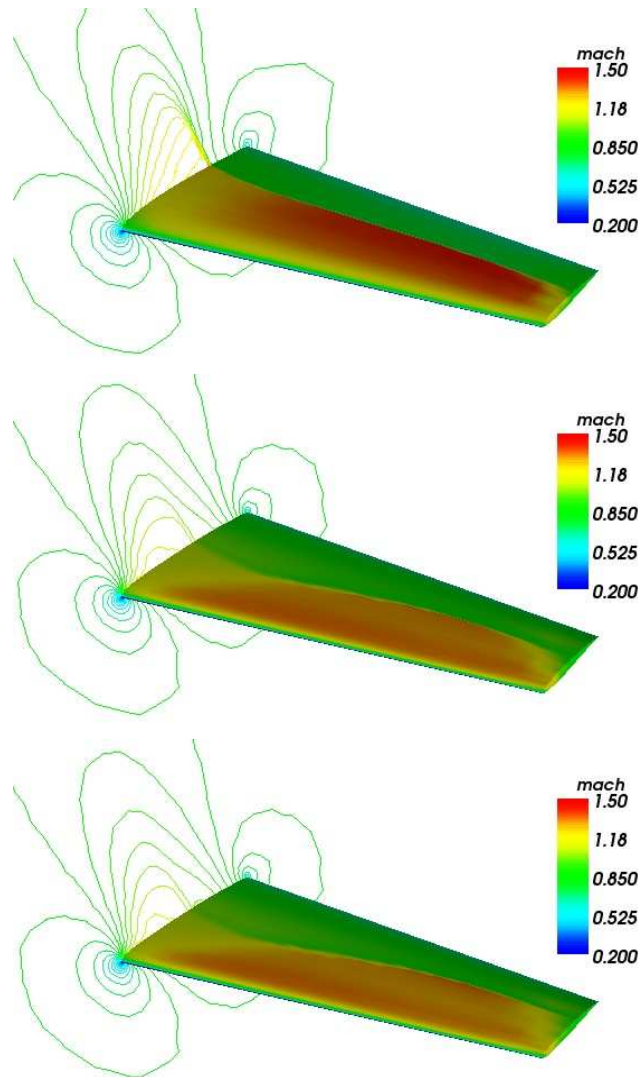


FIGURE 5.15 – Champ de Mach sur la surface de l'aile : configuration initiale (en haut), sans adaptation (au milieu), et avec adaptation (en bas)

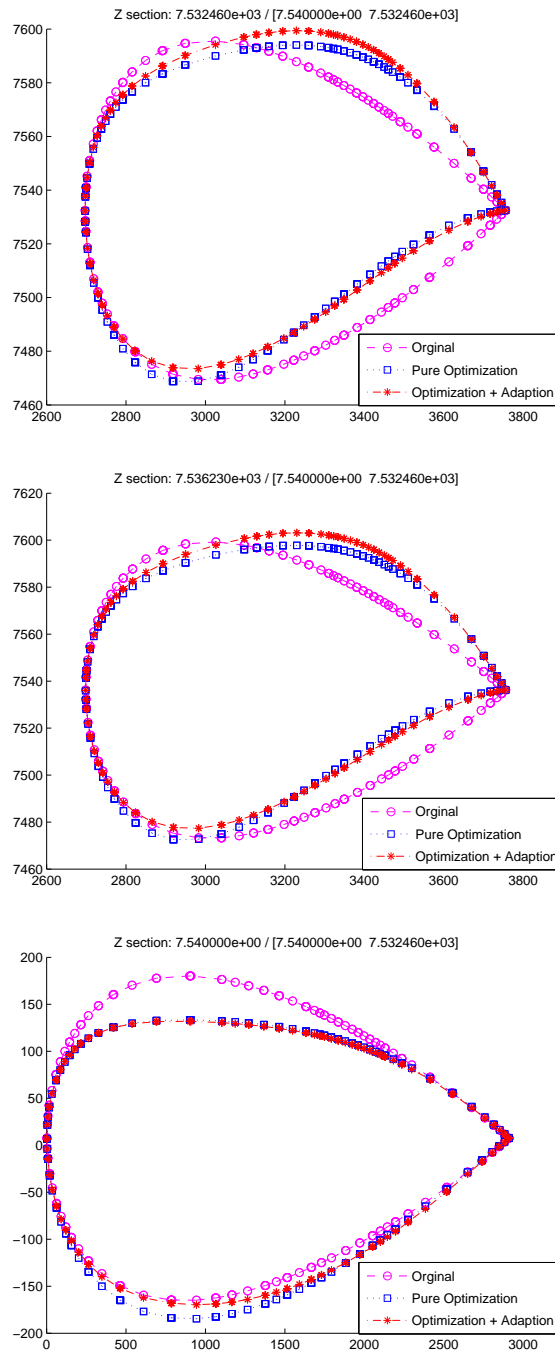


FIGURE 5.16 – Comparaison entre les formes, pour la configuration initiale et les deux stratégies d’optimisation, à trois sections de l’aile : l’extrémité (en haut), 50% de l’envergure (au milieu), et l’emplanture (en bas)

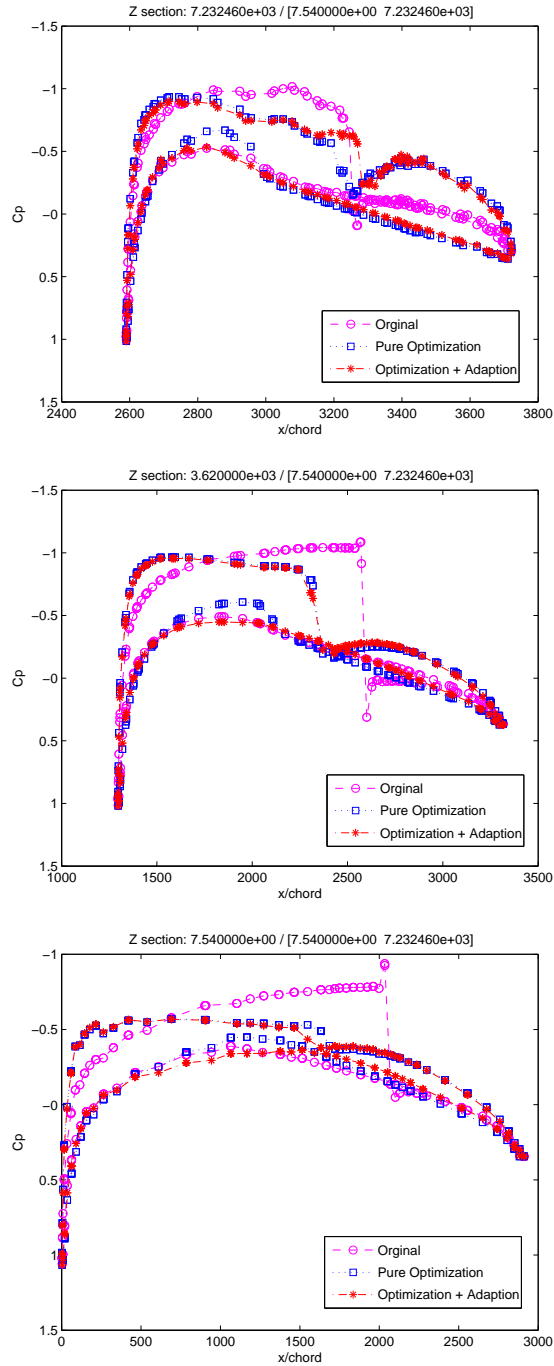


FIGURE 5.17 – Comparaison entre les coefficients de pression, pour la configuration initiale et les deux stratégies d’optimisation, à trois sections de l’aile : l’extrémité (en haut), 50% de l’envergure (au milieu), et l’emplanture (en bas)

**Optimisation avec une paramétrisation moyenne ( $6 \times 1 \times 1$ ) :** On reprend le test précédent avec une paramétrisation plus fine. Comme le montre la figure (5.18), la convergence de l'optimisation adaptative est moins rapide au début du processus, ce qui est normal, puisque l'irrégularité de la paramétrisation est encore faible. Elle devient ensuite plus performante par rapport à l'optimisation pure dont l'optimum atteint est à convergence complète. On résume dans la table (5.3) les valeurs optimales, obtenues par les deux stratégies d'optimisation, des coefficients aérodynamiques et de la fonctionnelle coût. On réalise avec l'optimisation adaptative un gain d'environ 6% en fonction objectif par rapport à l'optimisation pure. Ceci est illustré par la figure (5.19) qui donne une comparaison entre le champ de Mach sur la surface de l'aile pour la configuration initiale et les deux stratégies d'optimisation. On montre enfin dans les figures (5.20) et (5.21), respectivement, les formes et les coefficients de pression, pour la configuration initiale et les stratégies avec et sans adaptation, à trois sections de l'aile : l'extrémité, 50% de l'envergure, et l'emplanture.

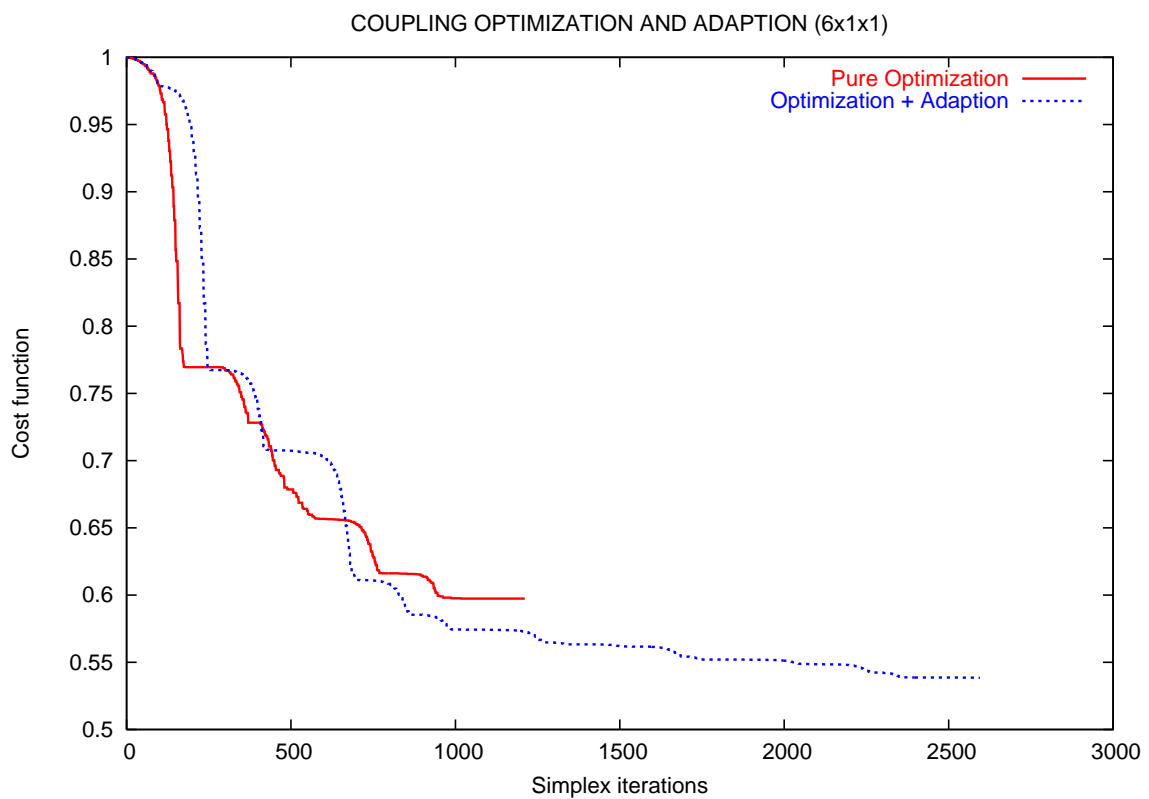


FIGURE 5.18 – Courbes de convergence : optimisation pure (jusqu'à convergence) et optimisation adaptative où l'adaptation est intervenue après chaque 50 itérations

	C_L	C_D	Fonction coût
Config. initiale	0.319192893	0.026352608	1.
Optimisation Pure	0.318873704	0.015741397	0.597337341
Optimisation + Adaptation	0.318885541	0.014196581	0.538446306

TABLE 5.3 – Comparaison entre les coefficients aérodynamiques obtenus par les deux stratégies d'optimisation

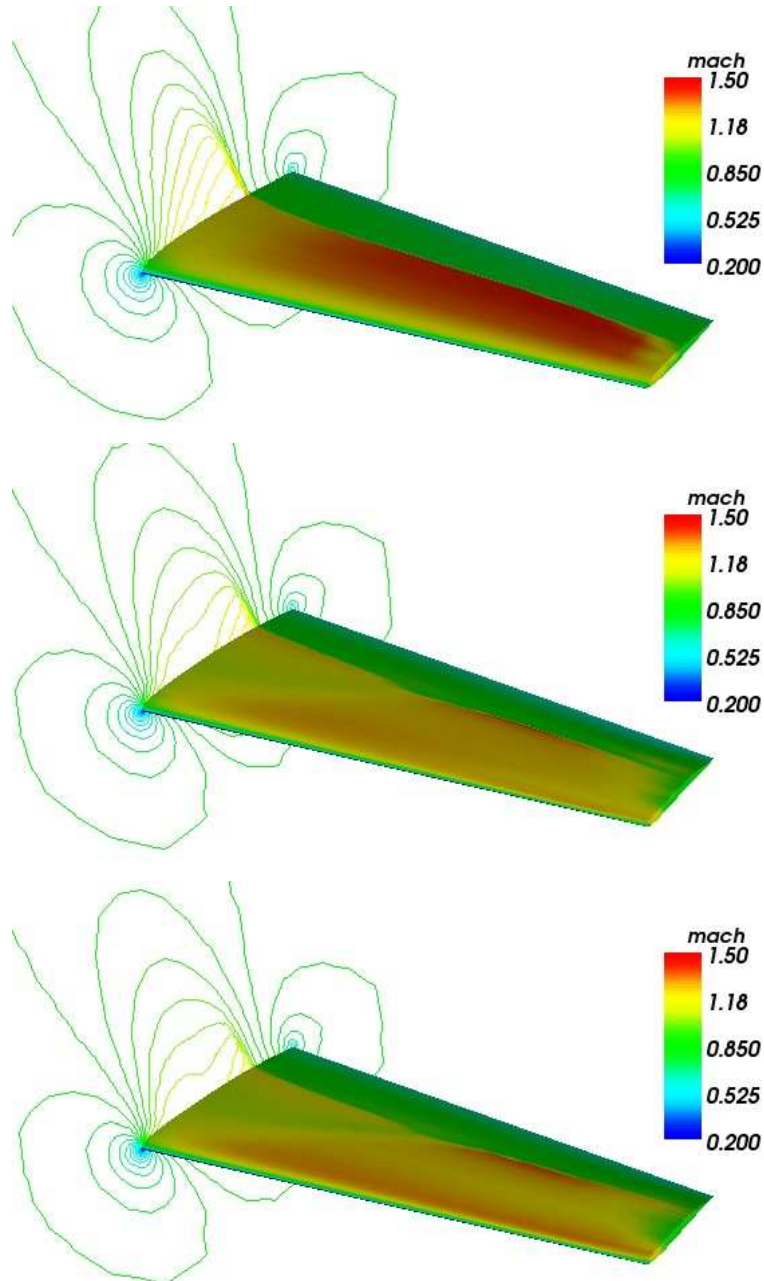


FIGURE 5.19 – Champ de Mach sur la surface de l'aile : configuration initiale (en haut), sans adaptation (au milieu), et avec adaptation (en bas)

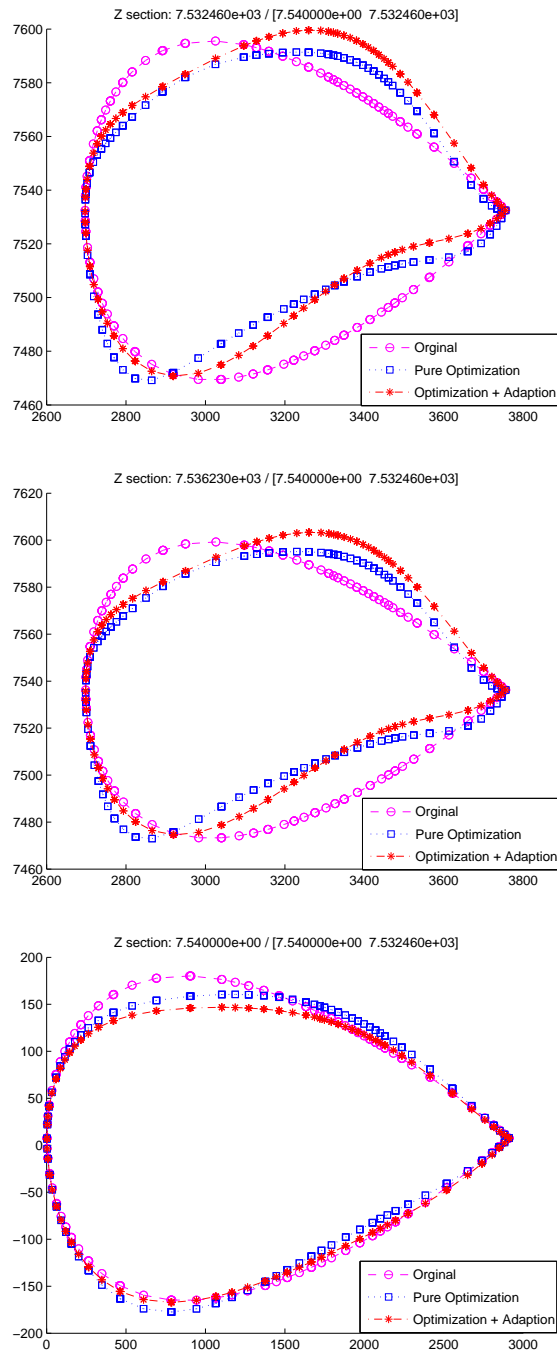


FIGURE 5.20 – Comparaison entre les formes, pour la configuration initiale et les deux stratégies d’optimisation, à trois sections de l’aile : l’extrémité (en haut), 50% de l’envergure (au milieu), et l’emplanture (en bas)

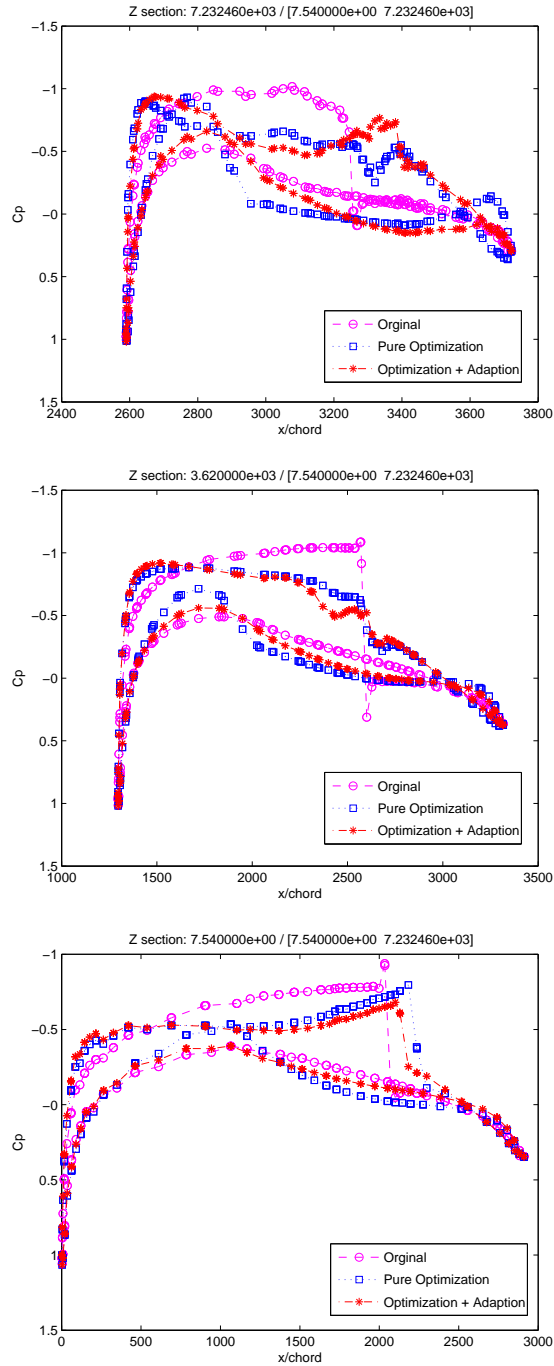


FIGURE 5.21 – Comparaison entre les coefficients de pression, pour la configuration initiale et les deux stratégies d’optimisation, à trois sections de l’aile : l’extrémité (en haut), 50% de l’envergure (au milieu), et l’emplanture (en bas)



**Optimisation avec une paramétrisation fine ( $9 \times 1 \times 1$ ) :** On utilise dans ce dernier test une paramétrisation encore plus fine. La stratégie d'optimisation adaptative fait appel à la procédure d'adaptation de paramétrisation après 1000 itérations. La figure (5.22) montre que l'utilisation de l'adaptation de paramétrisation dans la boucle d'optimisation permet d'obtenir un gain en efficacité et en précision. Les résultats obtenus par l'optimisation adaptative à 2000 itérations (n'est pas encore convergée) sont meilleurs que ceux réalisés par l'optimisation pure à convergence complète, avec plus de 3500 itérations. On donne dans la table (5.4) les coefficients aérodynamiques et la valeur de la fonction objectif associés à ces deux stratégies d'optimisation en comparaison avec la configuration initiale. On constate une réduction de la traînée avec l'optimisation adaptative qui correspond à environ 3% en fonctionnelle coût par rapport à l'optimisation pure. La figure (5.23) donne une comparaison entre le champ de Mach sur la surface de l'aile pour la configuration initiale et les deux stratégies d'optimisation. On montre enfin dans les figures (5.24) et (5.25) respectivement les formes et les coefficients de pression, pour la configuration initiale et les stratégies avec et sans adaptation, à trois sections de l'aile : l'extrémité, 50% de l'envergure, et l'emplanture.

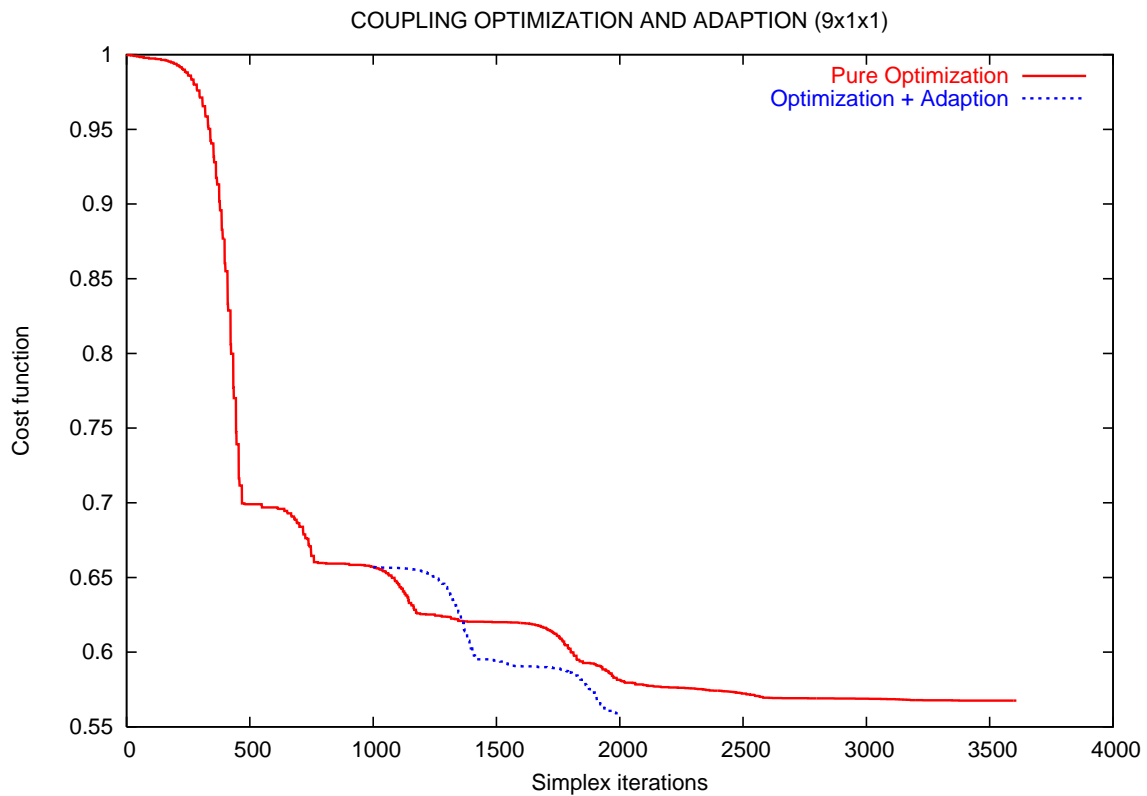


FIGURE 5.22 – Courbes de convergence : optimisation pure (jusqu'à convergence) et optimisation adaptative où l'adaptation est intervenue après 1000 itérations

	C_L	C_D	Fonction coût
Config. initiale	0.319192893	0.026352608	1.
Optimisation Pure	0.318888433	0.015321996	0.581422372
Optimisation + Adaptation	0.324819614	0.014732604	0.559057227

TABLE 5.4 – Comparaison entre les coefficients aérodynamiques obtenus par les deux stratégies d'optimisation à l'itération 2000

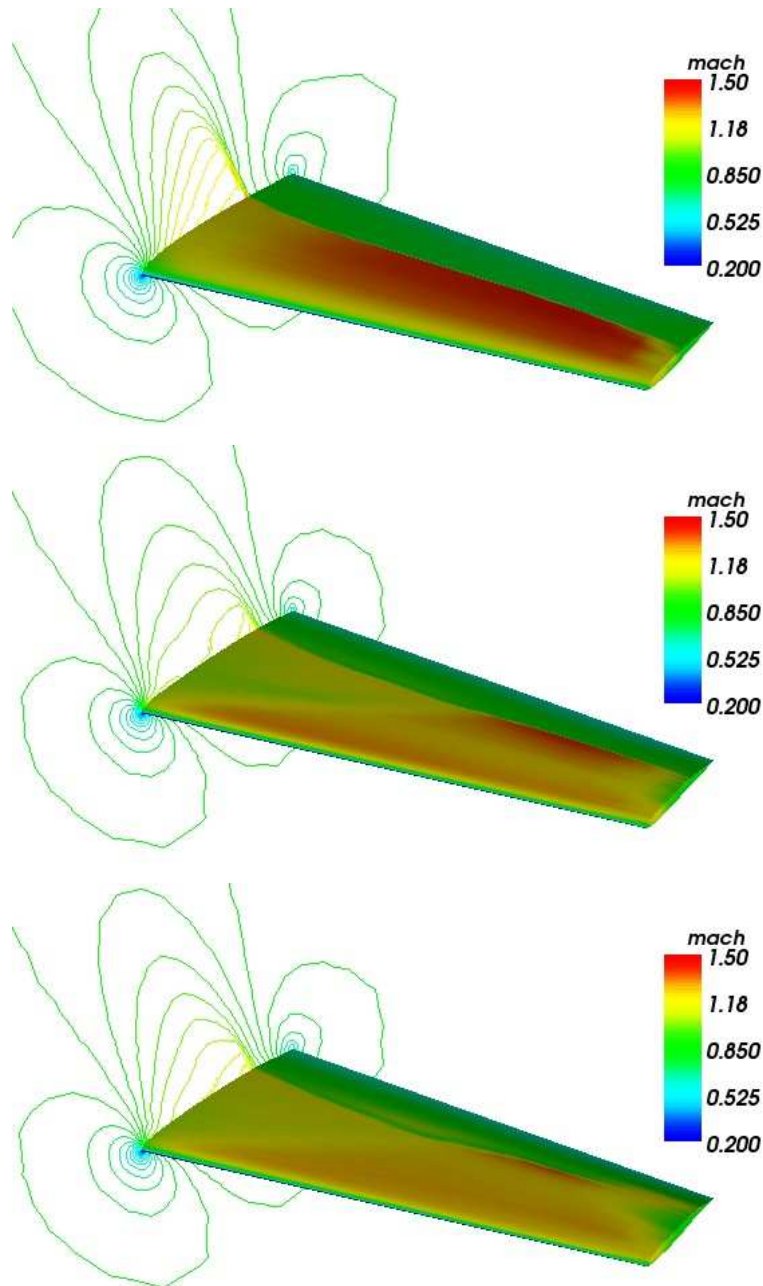


FIGURE 5.23 – Champ de Mach sur la surface de l'aile : configuration initiale (en haut), sans adaptation (au milieu), et avec adaptation (en bas)

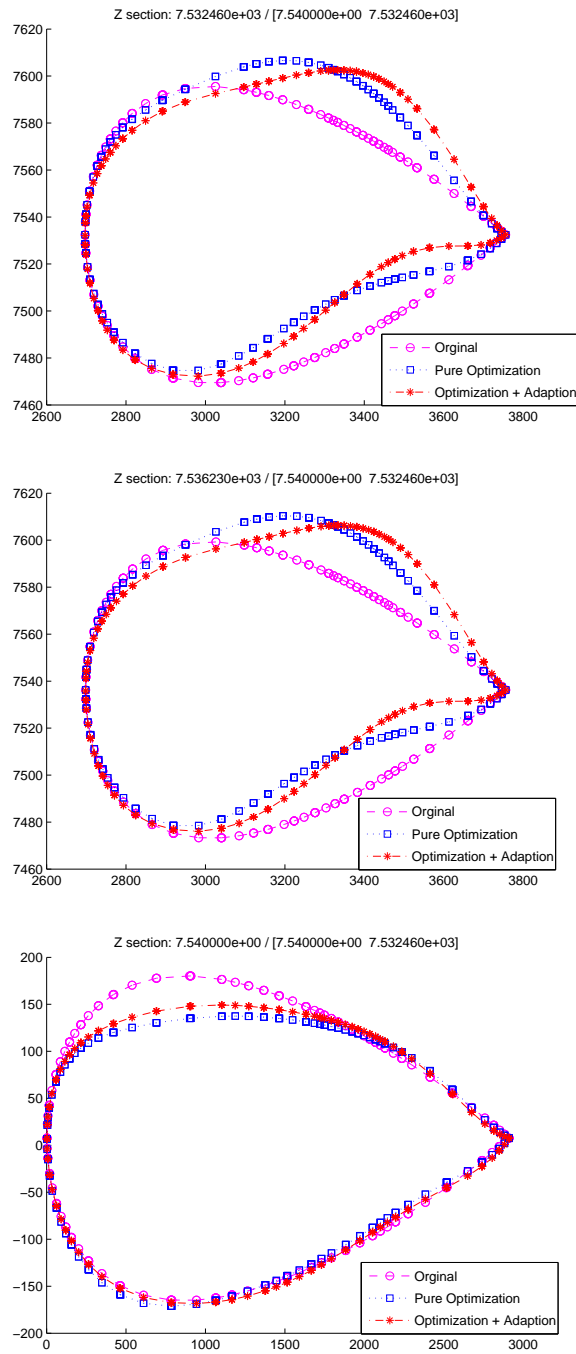


FIGURE 5.24 – Comparaison entre les formes, pour la configuration initiale et les deux stratégies d’optimisation, à trois sections de l’aile : l’extrémité (en haut), 50% de l’envergure (au milieu), et l’emplanture (en bas)

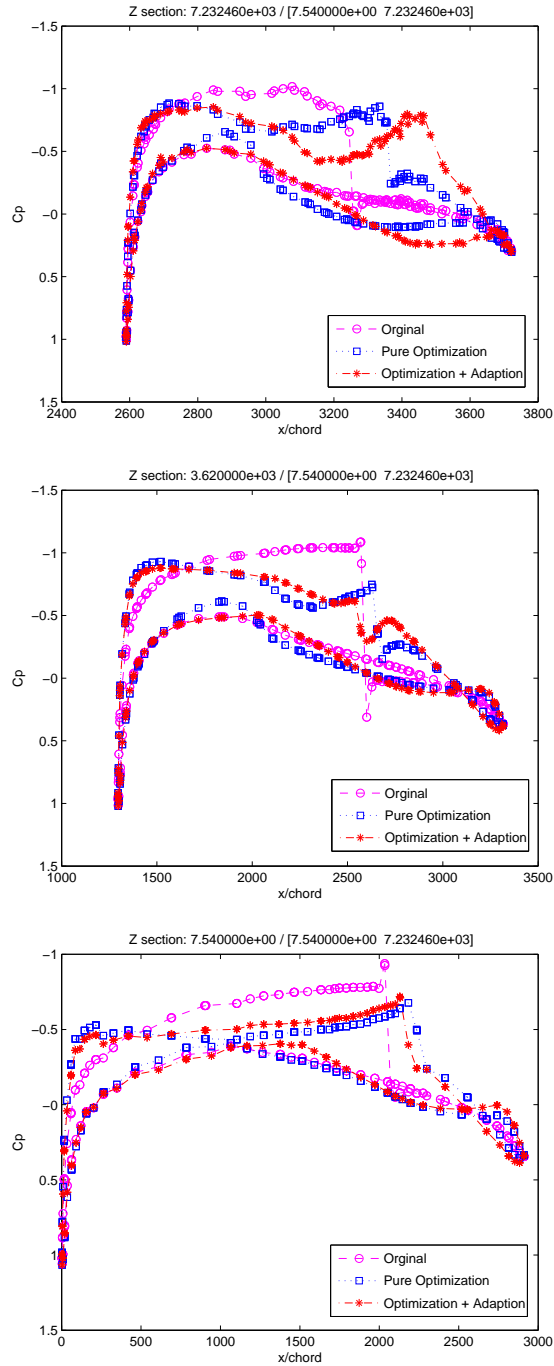


FIGURE 5.25 – Comparaison entre les coefficients de pression, pour la configuration initiale et les deux stratégies d’optimisation, à trois sections de l’aile : l’extrémité (en haut), 50% de l’envergure (au milieu), et l’emplanture (en bas)

## 5.4 Conclusion

On a présenté dans ce chapitre notre procédure d'adaptation de paramétrisation dans deux contextes différents. Dans la première section, on s'est limité seulement à une approche 1D dans laquelle la forme à optimiser (une courbe plane) est représentée directement par une paramétrisation de Bézier. Cette approche a été testée en utilisant un problème modèle de calcul des variations. Les résultats numériques affirment l'efficacité de l'adaptation de paramétrisation qui permet d'accélérer la convergence itérative ; On arrive avec une optimisation adaptative, avec deux paramètres, à obtenir la même précision qu'une optimisation pure avec 10 paramètres d'optimisation. On a étendu ensuite l'approche en 3D pour l'appliquer à un problème d'optimisation de forme en aérodynamique où la forme à optimiser est représentée en utilisant la méthode Free-Form deformation. La procédure d'adaptation, qui consiste à régulariser la boîte englobant la forme en changeant la carte FFD, a un coût marginal puisqu'il s'agit d'un processus purement géométrique. Cette approche a été testée sur un problème d'optimisation d'une voilure d'avion d'affaire en utilisant plusieurs types de paramétrisations. Les résultats numériques prouvent l'efficacité de la procédure d'adaptation qui permet, à moindre coût, d'accélérer la convergence itérative ou d'éviter les minima locaux.

On a étudié dans les chapitres précédents un problème d'optimisation qui fait intervenir une seule discipline. On s'intéresse désormais à un problème d'optimisation multidisciplinaire où plusieurs disciplines peuvent intervenir. Plus précisément, on considère deux disciplines : l'aérodynamique et la mécanique des structures.

## Chapitre 6

# Stratégies de jeux en optimisation multidisciplinaire : Aérodynamique versus Coques minces

### 6.1 Introduction

L'objectif majeur de l'industrie aéronautique est la conception d'avions non seulement performants, mais aussi avec un coût le moins élevé possible. Pour cela, plusieurs disciplines doivent intervenir dans la phase de conception ; par exemple : l'aérodynamique, la mécanique de structure, l'acoustique, etc. Ce processus multidisciplinaire devient de plus en plus complexe avec l'augmentation du nombre des disciplines puisqu'il nécessite l'interaction entre plusieurs équipes spécialisées afin de trouver un compromis entre différentes disciplines dont les objectifs peuvent être contradictoires. C'est pour cette raison que la méthodologie qui traite ce genre de processus est parfois appelée la conception collaborative [77] au lieu de l'optimisation multidisciplinaire (MDO). On se réfère aux articles [63][103] pour avoir un aperçu général sur les recherches qui ont été menées en optimisation multidisciplinaires dans le domaine aéronautique.

Il existe plusieurs approches qui permettent la résolution d'un problème d'optimisation multidisciplinaire ; pour un état de l'art, voir par exemple [2][4] [77][103]. Ces méthodologies peuvent être divisées en deux catégories : la première, qu'on désigne par mono-niveau, consiste à minimiser une seule fonctionnelle ; on fait appel seulement à un seul optimiseur. La méthode la plus naturelle, qui se nomme MDA (MultiDisciplinary Analysis) [3], consiste à résoudre (jusqu'à convergence) un système d'équations pour toutes les disciplines à chaque itération de la procédure d'optimisation. Cette méthode peut se formuler comme suit :

$$\begin{cases} \min f(x, u(x)) \\ g(x, u(x)) \geq 0, \end{cases} \quad (6.1)$$

où  $x$  est le vecteur de conception et  $u(x)$  est la solution du système des équations suivant,

$$\begin{cases} A_1(x, u_1(x), \dots, u_N(x)) = 0 \\ \vdots \\ A_N(x, u_1(x), \dots, u_N(x)) = 0 \end{cases} \quad (6.2)$$

Ce système est constitué des équations d'état des  $N$  disciplines considérées. On suppose, dans cette formulation, que chaque équation d'état contient les contraintes internes (ou privés) à la discipline.  $u_i(x)$  est la sortie de la discipline  $i$ . Par exemple, dans la première équation,  $u_k(x)$  ( $k = 2, \dots, N$ ) représentent la contribution des autres disciplines dans le calcul de  $u_1(x)$ .

Pour chaque vecteur  $x$  donné par la procédure d'optimisation, on calcul le vecteur d'état  $u(x)$  en résolvant le système (6.2). Ceci réduit la dimension du problème d'optimisation (6.1) en prenant seulement le vecteur  $x$  comme variable d'optimisation. Cependant, la procédure peut être coûteuse puisqu'elle nécessite la résolution de  $N$  équations différentielles. La formulation SAND (Simultaneous Analysis and Design) propose une alternative qui consiste à considérer les équations du système (6.2) comme des contraintes d'égalité. Donc, ces équations ne sont pas forcément satisfaites à chaque itération contrairement à la méthode MDA. Cette approche est connue par la désignation SAD ou SAND [51], et aussi par l'approche AAO (All-at-Once) [51]. La méthode DAO (Disciplinary Analysis Optimization) donne, contrairement à la méthode MDA, une certaine autonomie aux disciplines en introduisant des variables auxiliaires. Ces variables permettent de relaxer le vecteur de conception  $x$  afin que chaque discipline satisfasse ses propres contraintes. On rajoute des contraintes supplémentaires au problème d'optimisation pour que les variables auxiliaires soient identiques au vecteur  $x$  à la convergence. L'approche IDF (Individual Discipline Feasible) est une variante de DAO qui est apparue avant dans [2][26].

La deuxième catégorie, qu'on désigne par multi-niveau, décompose le problème en plusieurs sous-problèmes d'optimisation au nombre des disciplines; on peut faire appel par conséquent à plusieurs optimiseurs. La méthode CO (Collaborative Optimization) [18][116] est une approche multi-niveau qui consiste à décomposer la formulation DAO en  $N$  problèmes d'optimisation où chaque discipline fixe comme objectif, au sens des moindres carrés, que son équation d'état soit satisfaite et que ses variables auxiliaires coïncident avec le vecteur de conception  $x$ . On trouve dans [5] plusieurs exemples de ce type de méthode. Dans cette catégorie, on cite aussi la méthode CSSO (Concurrent SubSpace Optimization) qui consiste à partager les composantes du vecteur de conception  $x$  entre les disciplines. Chaque discipline optimise ses paramètres et communique l'état de ces variables aux autres disciplines. On se réfère à [102][110] pour plus de détails sur cette méthode.

Toutes ces méthodes, jusqu'à présent, traitent le problème multidisciplinaire en considérant une sorte de pondération "implicite" de tous les critères disciplinaires. Une autre idée consiste à affecter à chaque discipline son propre critère. Comme par exemple dans le cas de deux disciplines, la traînée (avec une contrainte sur la portance) pour l'aérodynamique, et la norme du tenseur des contraintes pour la mécanique des structures. Ce problème multicritère peut être résolu en attribuant à chaque critère un poids (un coefficient de substitution); on se ramène donc à un problème monocritère. Cette approche à deux principaux inconvénients : (1) le choix des poids à attribuer aux différents critères est arbitraire, ceci a une influence sur l'optimum atteint, (2) le problème de compensation ne permet pas de traiter l'un des critères comme plus ou moins important que les autres. Une autre alternative qu'on peut envisager pour résoudre les problèmes multicritères consiste à identifier les fronts de Pareto qui représentent l'ensemble des stratégies non-dominées, i.e. l'ensemble des compromis entre les critères. Cette approche est généralement coûteuse puisqu'elle demande un grand nombre d'évaluations de plusieurs critères. La deuxième difficulté est liée au choix du

meilleur point sur le front de Pareto. Pour remédier à toutes ces difficultés, la théorie des jeux définit un autre cadre pour traiter les problèmes d'optimisation multicritère. On considère  $N$  sous-problèmes d'optimisation (au nombre des disciplines) qui sont considérés comme des joueurs où chacun fixe comme objectif la minimisation d'un critère disciplinaire par rapport à un sous-ensemble du vecteur de conception. Les joueurs communiquent entre eux en étant coopératifs (jeu de Pareto), compétitifs (jeu de Nash) ou d'une manière hiérarchique (jeu de Stackelberg). Les joueurs passent par une situation de conflit en cherchant à converger vers un équilibre qui représente un compromis entre eux. Cette théorie a été étudiée par Périaux et al [89][94][117] comme un moyen efficace pour résoudre les problèmes d'optimisation multidisciplinaire. Dans la thèse de Wang [117], on trouve une étude qui combine les algorithmes génétiques et la théorie des jeux (jeu de Pareto, jeu de Nash et jeu de Stackelberg) pour la conception aérodynamique de systèmes hypersustentés en milieu pré-industriel. Il a aboutit à de nouveaux algorithmes évolutionnaires qui sont plus rapides et plus efficaces que les algorithmes génétiques classiques. Quant à Tang et al [109], ils ont utilisé une formulation d'un jeu de Nash avec une technique d'état adjoint, pour résoudre un problème d'optimisation multipoint en aérodynamique. L'une des stratégies testées consiste à optimiser une portion du bord d'attaque du profil par le premier joueur en maximisant la portance en régime subsonique, alors que la portion arrière du profil est optimisée par le deuxième joueur en minimisant la traînée en régime transsonique. Dans l'article [49], Habbal et al ont résolu un problème d'optimisation multidisciplinaire en utilisant un jeu non-coopératif (jeu de Nash). C'est un jeu à deux joueurs; le premier contrôle la structure et le second contrôle la distribution de température dans la structure. Les résultats numériques montrent la capacité de cette approche à résoudre les problèmes d'optimisation multidisciplinaire. On trouve aussi une comparaison entre la méthode par pondération avec l'approche jeu de Nash.

On s'intéresse, dans ce chapitre, à un cas d'étude en optimisation multidisciplinaire en utilisant la théorie des jeux. On traite un problème d'optimisation couplée entre l'aérodynamique et la mécanique des structures dans un régime où l'aérodynamique est le critère le plus important. Le couplage entre ces deux disciplines est très fort dans le sens que la première fournit le chargement nécessaire à la deuxième pour déterminer le champ des déplacements et le tenseur des contraintes. En revanche, la structure modifie le calcul de l'écoulement en raison de la déformation de forme. Ce problème sera traité en utilisant une idée qui a été développée dans [30], et qui propose une manière d'effectuer le partage de territoire dans le cas d'une optimisation concurrente afin que la dégradation du critère de la discipline principale soit la moindre possible. Le problème d'optimisation couplée est formulé comme un jeu à deux joueurs qui interagissent parallèlement dans un jeu de Nash et hiérarchiquement dans un jeu de Stackelberg par l'intermédiaire des paramètres de conception.

On présente brièvement, dans la section suivante, les deux problèmes d'optimisation de forme. On rappelle ensuite les notions de base du jeu de Nash et du jeu de Stackelberg ainsi que les algorithmes de couplage utilisés. On présente enfin quelques résultats numériques; on optimise la forme d'une voilure d'avion d'affaire, où le concepteur aérodynamique interagit avec le concepteur structural via des stratégies de jeu.



## 6.2 Deux problèmes d'optimisation de forme

### 6.2.1 Optimisation aérodynamique

On utilise le même problème d'optimisation de forme aérodynamique que celui qu'avant, dont les composantes essentielles ont été présentées dans le deuxième chapitre. On peut le formuler comme suit :

$$\min_Y J_A(Y) \quad (6.3)$$

où  $Y$  est le vecteur de conception, et  $J_A$  est la fonctionnelle coût pour l'aérodynamique.

Le problème d'optimisation consiste à minimiser la fonctionnelle coût  $J_A$ , pour un jeu de paramètres, vis-à-vis des contraintes aérodynamiques et géométriques. Chaque évaluation de la fonction coût se traduit par la résolution des équations d'Euler compressible par la méthode de volumes finis en maillage non-structuré.

### 6.2.2 Optimisation structurale

Dans cette étude, on traite un problème d'optimisation de forme à l'aide d'un modèle de coque mince élastique. Le but est d'améliorer la géométrie initiale pour qu'elle ait une meilleure tenue mécanique possible en minimisant une fonctionnelle coût pour la structure. Chaque évaluation de la fonction coût passe par la résolution des équations de coque mince linéaire par des éléments finis DKT en faisant appel à un code d'analyse par éléments finis de EDF qui se nomme Code Aster (voir Annexe C).

En résumé, le problème d'optimisation structurale se formule comme suit :

$$\min_Y J_S(Y) \quad (6.4)$$

où  $Y$  désigne le vecteur de conception, et  $J_S$  est la fonctionnelle coût pour la structure.

Dorénavant, on utilise le terme "structure" pour désigner le modèle de coques minces élastiques linéaires.

## 6.3 Deux approches d'optimisation multidisciplinaire

On traite dans cette étude un problème d'optimisation multidisciplinaire en faisant appel à la théorie des jeux, qui a été introduite dans les années 30 par Von Neumann [86]. Cette approche d'optimisation concurrente considère les différents critères à minimiser comme des joueurs qui interagissent entre eux d'une façon symétrique (Jeu de Nash) ou d'une façon hiérarchique (Jeu de Stackelberg). On se réfère aux ouvrages [8][9][88] pour une présentation détaillée de la théorie des jeux. On présente ci-dessous les deux approches utilisées pour résoudre le problème d'optimisation couplée entre deux disciplines : l'aérodynamique et la structure.

### 6.3.1 Jeu de Nash

Cette stratégie, qui prend le nom de son auteur John Nash [84], est un jeu non-coopératif entre plusieurs joueurs où chacun cherche à améliorer son propre critère simultanément et indépendamment des autres. L'état d'équilibre est atteint lorsqu'aucun joueur ne peut améliorer son propre critère unilatéralement, c'est ce qu'on nomme l'équilibre de Nash (EN). Prenons par exemple le cas d'un jeu à deux joueurs (l'extension à plusieurs joueurs est directe). On considère  $\bar{U}$  (resp.  $\bar{V}$ ) l'ensemble des stratégies admissibles du joueur (1) (resp. du joueur (2)).

Les deux joueurs agissent respectivement comme suit :

$$\text{Joueur (1) : } \min_{U \in \bar{U}} J_1(U, V) \quad (6.5)$$

$$\text{Joueur (2) : } \min_{V \in \bar{V}} J_2(U, V) \quad (6.6)$$

où  $J_1$  et  $J_2$  sont deux fonctionnelles coût définies sur  $\bar{U} \times \bar{V}$ .

Les deux joueurs cherchent à améliorer leurs propres critères parallèlement. Le but est de trouver une stratégie  $(U^*, V^*) \in \bar{U} \times \bar{V}$  qui définit l'état d'équilibre du jeu ; autrement dit :

$$J_1(U^*, V^*) = \min_{U \in \bar{U}} J_1(U, V^*) \quad (6.7)$$

$$J_2(U^*, V^*) = \min_{V \in \bar{V}} J_2(U^*, V) \quad (6.8)$$

La stratégie  $(U^*, V^*)$  désigne un point d'équilibre de Nash.

### 6.3.2 Jeu de Stackelberg

Le jeu de Stackelberg est une autre approche, issue de la théorie des jeux, pour traiter les problèmes d'optimisation multidisciplinaire. C'est un jeu non coopératif dont les joueurs n'ont pas un rôle symétrique, comme c'est le cas pour le jeu de Nash, mais interagissent d'une manière hiérarchique en désignant l'un d'eux comme meneur et les autres comme suiveurs.

On considère un jeu à deux joueurs dont le premier est un meneur et le second est un suiveur. Soit  $\bar{U}$  (resp.  $\bar{V}$ ) l'ensemble des stratégies admissibles du joueur (1) (resp. du joueur (2)). Soit  $J_k$  une fonction coût pour le joueur ( $k$ ) définie sur  $\bar{U} \times \bar{V}$ , où  $k=1,2$ . On définit les deux fonctions suivantes  $T^1 : \bar{V} \rightarrow \bar{U}$  et  $T^2 : \bar{U} \rightarrow \bar{V}$  comme suit :

$$T^1(V) = \arg \min_{U \in \bar{U}} J_1(U, V) \quad (6.9)$$

et

$$T^2(U) = \arg \min_{V \in \bar{V}} J_2(U, V) \quad (6.10)$$

Le jeu de Stackelberg consiste à résoudre le problème d'optimisation suivant :

$$\min_{U \in \bar{U}} J_1(U, T^2(U)) \quad (6.11)$$

La stratégie  $(U^*, V^*) \in \bar{U} \times \bar{V}$  est une solution du jeu de Stackelberg si  $U^*$  est une solution de (6.11) et  $V^* = T^2(U^*)$ .

## 6.4 Stratégies d'optimisation

### 6.4.1 Paramétrisation de forme

La géométrie est paramétrisée en utilisant l'approche FFD (Free-Form Deformation) présentée dans le deuxième chapitre. On rappelle que cette approche consiste à définir un volume de contrôle (ou une boîte FFD) englobant la forme ; la déformation de la boîte se répercute sur la forme en utilisant un tensoriel 3D de Bézier. Autrement dit, pour tout noeud  $q$  du maillage, on pose :

$$m_q = m_q^{init} + \sum_{i=0}^{n_i} \sum_{j=0}^{n_j} \sum_{k=0}^{n_k} B_i^{n_i}(s_q) B_j^{n_j}(t_q) B_k^{n_k}(u_q) \delta p_{ijk} \quad (6.12)$$

où  $m_q$  (resp.  $m_q^{init}$ ) représente les coordonnées de  $q$  dans le maillage courant (resp. dans la configuration initiale),  $t_q = (s_q, t_q, u_q)$  sont les coordonnées locales de  $q$  dans la boîte englobante,  $\delta p_{ijk}$  sont les paramètres de déformation, et  $B_i^{n_i}$ ,  $B_j^{n_j}$  et  $B_k^{n_k}$  sont les polynômes de Bernstein d'ordre  $n_i$ ,  $n_j$  et  $n_k$ .

On montre dans la figure (6.1) un exemple d'une paramétrisation de forme de degré  $6 \times 1 \times 1$  qui correspond à 28 points de contrôle (les points rouge de la figure). On fixe, durant le processus d'optimisation, le bord de fuite et le bord d'attaque, ce qui revient à fixer les sommets de la boîte englobante ( $i = 0, n_i$ ). On bouge les points de contrôle selon l'axe des ordonnés. On désigne par les paramètres de conception les degrés de liberté de la paramétrisation qui sont les composantes variables de  $\delta p_{ijk}$  selon l'axe des ordonnées :  $(\delta y_{ijk})_{(i,j,k) \in (1, \dots, n_i-1) \times (0, \dots, n_j) \times (0, \dots, n_k)}$ .

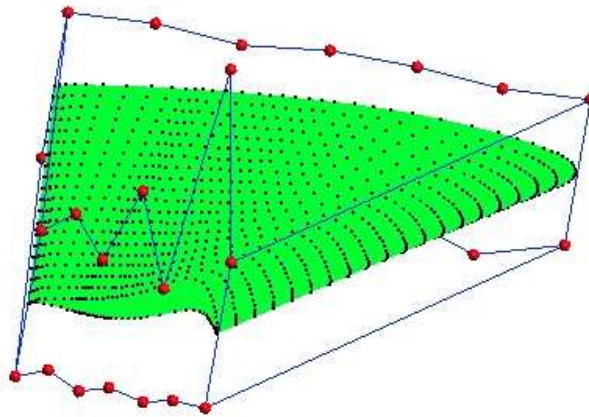


FIGURE 6.1 – Exemple d'une boîte englobant la forme dans le cas d'une représentation de Bézier de degré  $6 \times 1 \times 1$

## 6.4.2 Partage de territoires entre les disciplines

Pour conduire un problème d'optimisation concourante en considérant deux joueurs : l'aérodynamique et la structure, qu'on note respectivement par A et S, on doit assigner à chaque joueur un sous-ensemble de paramètres de conception, ce qui revient à partager l'espace paramétrique global en deux sous-espaces. Soit  $Y^* \in \mathbb{R}^{n+1}$  le vecteur de conception associé à la forme initiale. Soit  $U \in \mathbb{R}^{n+1-p}$  (resp.  $V \in \mathbb{R}^p$ ) la stratégie du joueur A (resp. du joueur S), avec  $1 \leq p \leq n$ . A chaque itération du processus d'optimisation, le vecteur de conception  $Y \in \mathbb{R}^{n+1}$  est tel que,

$$Y = Y(U, V) = Y^* + S \begin{pmatrix} U \\ V \end{pmatrix} \quad (6.13)$$

où  $S$  est une matrice inversible de dimension  $(n+1) \times (n+1)$ , qu'on nomme la matrice de partage (Splitting).

On peut choisir la matrice  $S$  de telle sorte que le partage s'effectue directement sur les points de contrôles qui définissent la boîte FFD. Quand on choisit par exemple  $S$  égale à l'identité, cela revient à attribuer les points de contrôles qui sont à l'extrémité de l'aile à l'aérodynamique et ceux qui sont à l'emplanture à la structure ; ceci correspond, pour une paramétrisation de degré  $6 \times 1 \times 1$  comme dans la figure (6.1), à  $n+1 = 20$  et  $p = 10$ . La question qui se pose est comment choisir la matrice  $S$  d'une façon optimale ? On note que dans ce processus d'optimisation concourante, l'aérodynamique est le plus important. On ne peut donc tolérer qu'une petite dégradation des performances aérodynamiques qui peut être causée par le critère structural. La réponse à la question a été abordé en détail dans [30]. L'idée de base est la suivante : on effectue une optimisation purement aérodynamique dans l'espace paramétrique global (de dimension  $n+1$ ) ; on prend, dans ce cas,  $Y^*$  comme étant le vecteur de conception associé à la forme optimisée, donc le choix se base sur les résultats de ce problème d'optimisation, et sur le fait que les perturbations infinitésimales des paramètres du second critère causent la moindre dégradation possible de la valeur de la fonctionnelle du premier critère.

On considère le développement limité de la fonctionnelle  $J_A$  (le critère principal) au point  $Y^*$  dans la direction du vecteur unitaire  $\omega \in \mathbb{R}^{n+1}$  :

$$J_A(Y^* + \epsilon\omega) = J_A(Y^*) + \epsilon G^* \cdot \omega + \frac{\epsilon^2}{2} \omega \cdot H^* \cdot \omega + O(\epsilon^3) \quad (6.14)$$

avec  $G^*$  et  $H^*$  sont respectivement le gradient et le hessien de  $J_A$  au point  $Y^*$ .

L'idée de cette deuxième stratégie consiste à trouver une base des vecteurs  $\{\omega^j\} (j = 1, \dots, n+1)$ , de telle sorte que  $|J_A(Y^* + \epsilon\omega^j) - J_A(Y^*)|$  soit le plus petit possible (pour un petit  $\epsilon$  fixé) pour  $n+2-p \leq j \leq n+1$ . On détaillera cette stratégie à la fin de ce chapitre.

## 6.4.3 Algorithmes de couplage

On présente ci-dessous les deux algorithmes utilisés pour conduire une optimisation concourante entre l'aérodynamique et la structure. On utilise les mêmes notations que la

section précédente. Soit  $\bar{U}$  et  $\bar{V}$  les deux espaces d'actions pour le joueur A et B respectivement. Les deux fonctionnelles coût  $J_A$  et  $J_S$ , respectivement pour les deux joueurs A et B, sont définies sur  $\bar{U} \times \bar{V}$ .

### Algorithme 1 : Jeu de Nash

1. Initialisation :  $k = 0$
2. Soit  $(U^{(k)}, V^{(k)})$  la paramétrisation initiale, et  $P^{(k)}$  la pression aérodynamique associée.
3. On fixe  $V^{(k)}$ , et on minimise  $J_A$  par rapport à  $U$ , i.e :

$$\min_{U \in \bar{U}} J_A(U, V^{(k)}) \rightarrow (U^{(k+1)}, V^{(k)}) \quad (6.15)$$

avec

$$J_A(U, V^{(k)}) = J_A(Y(U, V^{(k)})) \quad (6.16)$$

4. On fixe  $U^{(k)}$ , et on minimise  $J_S$  par rapport à  $V$  en appliquant  $P^{(k)}$  comme chargement par noeud, i.e :

$$\min_{V \in \bar{V}} J_S(U^{(k)}, V) \rightarrow (U^{(k)}, V^{(k+1)}) \quad (6.17)$$

avec

$$J_S(U^{(k)}, V) = J_S(Y(U^{(k)}, V)) \quad (6.18)$$

5.  $k = k + 1$  ; retour à (2)

Le point  $(U^*, V^*)$  est un point d'équilibre de Nash si,

$$J_A(U^*, V^*) = \min_{U \in \bar{U}} J_A(U, V^*) \quad (6.19)$$

$$J_S(U^*, V^*) = \min_{V \in \bar{V}} J_S(U^*, V) \quad (6.20)$$

### Algorithme 2 : Jeu de Stackelberg

Dans ce cas, le jeu est répété séquentiellement entre l'aérodynamique comme meneur et la structure comme suiveur. On définit la fonction  $T^F : \bar{U} \rightarrow \bar{V}$  telle que :

$$T^F(U) = \arg \min_{V \in \bar{V}} J_S(U, V) \quad (6.21)$$

On note aussi que :

$$J_L(U, V) = J_L(Y(U, V)), \quad L = A, S \quad (6.22)$$

1. Initialisation :  $k = 0$
2. Soit  $(U^{(k)}, V^{(k)})$  la paramétrisation initiale, et  $P^{(k)}$  la pression aérodynamique associée.
3. Le suiveur optimise sa stratégie en appliquant  $P^{(k)}$  comme chargement par noeud. Soit  $V^{(k+1)} = T^F(U^{(k)})$
4. On évalue la fonction coût  $J_A$  au point  $(U^{(k)}, V^{(k+1)})$
5.  $k = k + 1$ , retour à (2)

Le point  $(U^*, V^*)$  est un point d'équilibre de Stackelberg si,

$$V^* = T^F(U^*) \text{ et } U^* = \arg \min_{U \in \bar{U}} J_A(U, T^F(U)) \quad (6.23)$$

## 6.5 Expérimentation numérique

### 6.5.1 Données aérodynamiques

On utilise le même cas test décrit dans le chapitre (4). On rappelle ici les données basiques. On optimise une aile d'avion 3D en écoulement transsonique. Les conditions à l'infini sont les suivantes :

Nombre de Mach	0.83
Angle d'incidence	2°

Notre solveur utilise un maillage non-structuré composé de 31124 noeuds et 173445 éléments, et qui est adapté à un écoulement transsonique avec un raffinement à l'emplacement du choc (voir figure (6.2)).

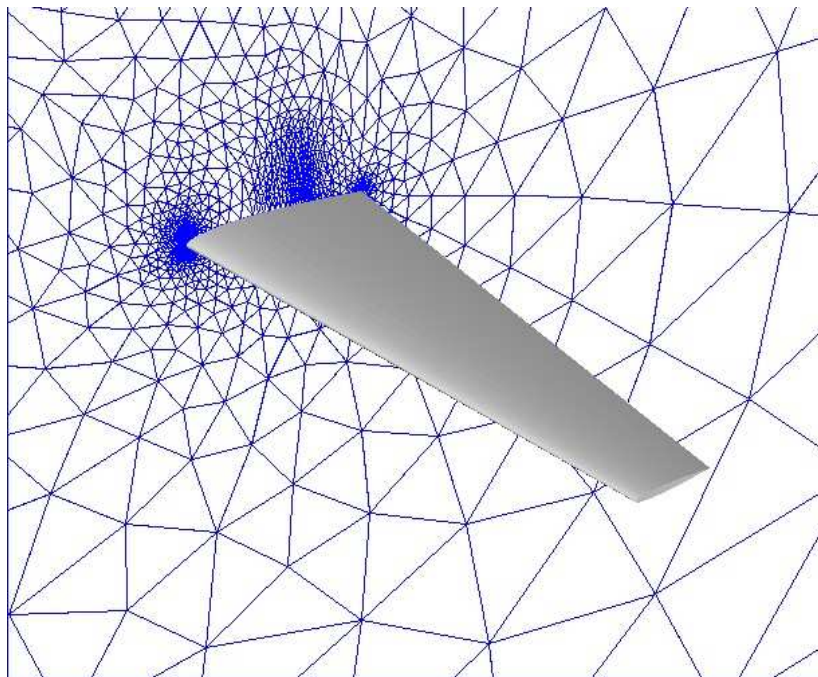


FIGURE 6.2 – Forme initiale et maillage dans le plan de symétrie

Le but de l'optimisation est de réduire le coefficient de la traînée  $C_D$  avec une contrainte sur la portance  $C_L$  qui ne doit pas diminuer de plus de 0.1%. Cette contrainte est imposée par une pénalisation :

$$J_A = \frac{C_D}{C_{D_0}} + 10^4 \cdot \max\left(0, 0.999 - \frac{C_L}{C_{L_0}}\right) \quad (6.24)$$

avec  $C_{D_0}$  et  $C_{L_0}$  sont respectivement la traînée et la portance de la configuration initiale.

### 6.5.2 Données structurales

L'étude concerne une coque mince d'épaisseur 5mm dont la surface moyenne correspond à la surface d'une aile d'avion. On impose un encastrement à l'emplanture (voir figure (6.3)). Le matériau utilisé pour la structure de l'aile est un alliage d'aluminium de module de Young

68948 MPa et de coefficient de Poisson 0.33.

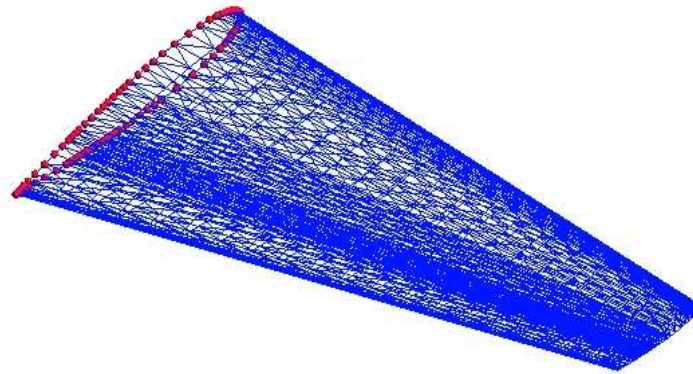


FIGURE 6.3 – Surface moyenne avec encastrement à l’emplanture

La fonctionnelle coût pour la structure est définie comme suit :

$$J_S = \int_S \|\sigma \cdot \vec{n}\| dS + 10^4 \cdot \max(0, -\Delta \text{Vol}) + 10^4 \cdot \max(0, \Delta \text{Surf}) \quad (6.25)$$

où  $\sigma$  est le tenseur des contraintes,  $\vec{n}$  est le vecteur normal à la surface de l’aile  $S$ , et  $\Delta X = (X - X_A)/X_A$  pour  $X = \text{Vol}, \text{Surf}$ , avec  $X_A$  est la valeur de  $X$  après chaque optimisation aérodynamique et Vol et Surf sont respectivement le volume interne et la surface de l’aile.

On impose deux contraintes géométriques sur le changement de volume et de surface par pénalisation afin de réduire l’amplitude de la déformation géométrique due à l’optimisation structurale. Le champ des déplacements et le tenseur des contraintes sont calculés en résolvant les équations de coque mince linéaire en faisant appel au Code Aster (voir Annexe C).

### 6.5.3 Résultats

Dans une optimisation concurrente, chaque joueur cherche à améliorer son propre critère. Donc, les joueurs passent par une situation de conflit qui évolue avant d’atteindre un certain équilibre qui représente une situation de compromis entre tous les joueurs. Les méthodes d’optimisation utilisées et la nature des disciplines considérées peuvent rendre ce processus très coûteux. Prenons par exemple le cas d’un jeu entre l’aérodynamique et la structure. Dans les deux processus d’optimisation, chaque évaluation de la fonction coût nécessite la résolution de deux équations d’état : les équations de coque mince et les équations d’Euler. Pour une méthode d’optimisation robuste, qui nécessite un nombre important d’évaluations de la fonctionnelle coût, l’équilibre ne peut être atteint que pour un coût CPU important. C’est pour cette raison que nous avons utilisé initialement un méta-modèle [22] pour le calcul de l’écoulement, qui se base sur des fonctions à base radiale (RBF) et le modèle de

krigeage (voir Annexe D). Ce choix a permis d'effectuer plusieurs tests numériques, avec un temps calcul faible, afin de tester l'influence de certains paramètres vis-à-vis de l'optimisation. Ceci nous a servi par la suite pour effectuer des tests d'optimisation avec un code CFD.

Dans les deux sections suivantes, on résout le problème d'optimisation concurrente en utilisant les deux stratégies de partage de territoire décrites précédemment. On initialise le processus par une forme initiale associée à une paramétrisation de degré  $3 \times 1 \times 1$ , et qui correspond à un vecteur de conception  $Y^* \in \mathbb{R}^8$ . Cette forme est soit le NACA0012 ou une solution d'une optimisation purement aérodynamique.

Le méta-modèle utilisé dans les expérimentations numériques a été construit pour approximer les valeurs du coefficient de traînée  $C_D$  et du coefficient de portance  $C_L$ . Ces approximations ont été calculées en utilisant une base de données aérodynamiques ( $C_D$ ,  $C_L$ ) qui sont calculées au voisinage du point optimale  $Y^*$ , qui résulte d'une optimisation purement aérodynamique dans l'espace paramétrique global, avec un code CFD de cette manière :  $Y = Y^* + Y_r$  où  $Y_r \in \mathbb{R}^8$  est un vecteur arbitraire dont les éléments appartiennent à l'intervalle  $[-50, 50]$ .

### 6.5.3.1 Première stratégie d'optimisation

On utilise une stratégie d'optimisation qui associe à chaque discipline un sous-ensemble de paramètres de contrôle. Ceci revient à prendre la matrice de partage comme étant la matrice identité à une permutation près des colonnes. Le problème qui se pose est alors de choisir la meilleure stratégie de partage qui réalise la moindre dégradation possible du critère important, qui est l'aérodynamique et une amélioration du critère structural. Pour cela, on a effectué trois optimisations mono-disciplinaire (pour la structure et pour l'aérodynamique) avec seulement 4 paramètres d'optimisation ; les autres paramètres sont maintenus fixe. Trois types de stratégies ont été considérés : (1) tous les points de contrôle sont à l'extrémité de l'aile, (2) tous les point de contrôle sont à l'emplanture de l'aile, et (3) les points de contrôles sont répartis entre l'extrémité et l'emplanture de l'aile. On illustre dans la figure (6.4) trois types de stratégies de partage de terittoires ; on utilise la lettre 'A' (resp. 'S') pour désigner les points de contrôle donnés à l'aérodynamique (resp. à la structure). La figure (6.5) montre une comparaison entre les courbes de convergence correspondant aux stratégies (1), (2) et (3) pour l'aérodynamique. On constate que la stratégie la plus influente sur l'optimisation aérodynamique est celle où les points de contrôle sont à l'extrémité. Quant à la stratégie (3), l'amélioration aérodynamique est très petite. Ceci est dû au nombre faible des paramètres à l'extrémité et à l'emplanture de l'aile. Il s'agit donc, pour que l'optimisation couplée soit moins violente entre l'aérodynamique et la structure, à attribuer les points de contrôle qui sont à l'emplanture de l'aile à la structure et ceux qui sont à l'extrémité de l'aile à l'aérodynamique, comme dans la figure (6.4)(milieu). On reprenant la même expérience pour la structure, cette dernière stratégie permet une amélioration du critère structurale, comme on peut le voir dans la figure (6.6).



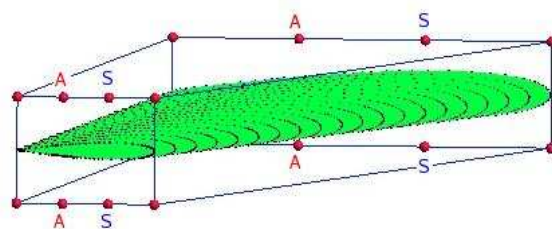
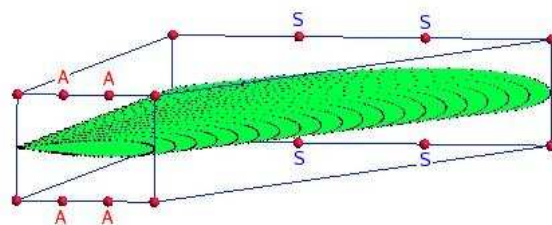
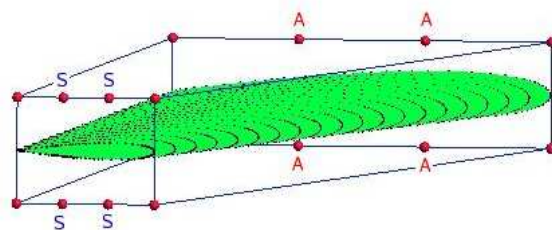


FIGURE 6.4 – Illustration de différentes stratégies de partage de territoires

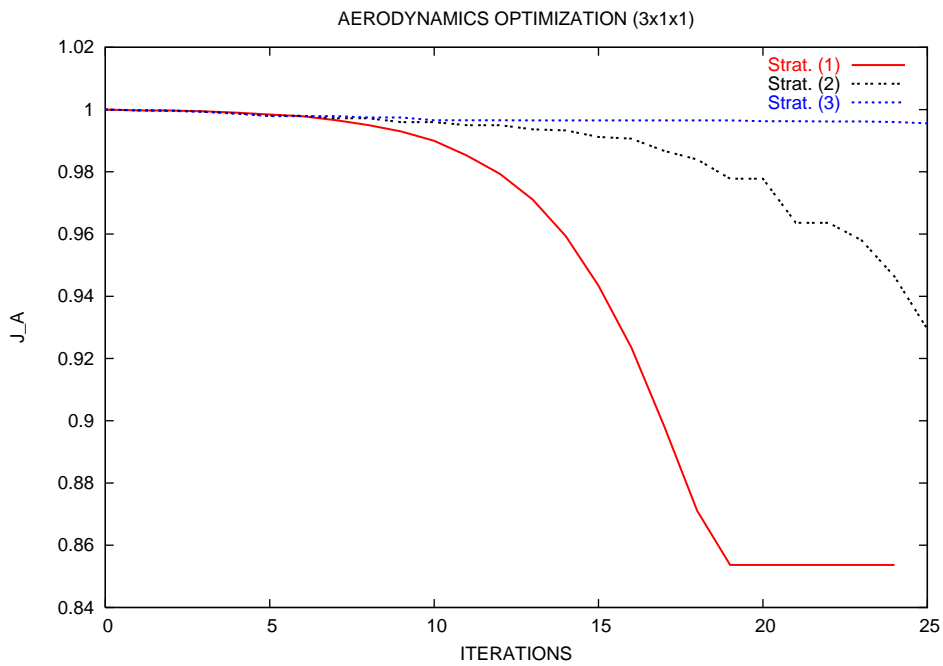


FIGURE 6.5 – Comparaison entre les courbes de convergence pour une optimisation purement aérodynamique correspondant à trois différentes stratégies

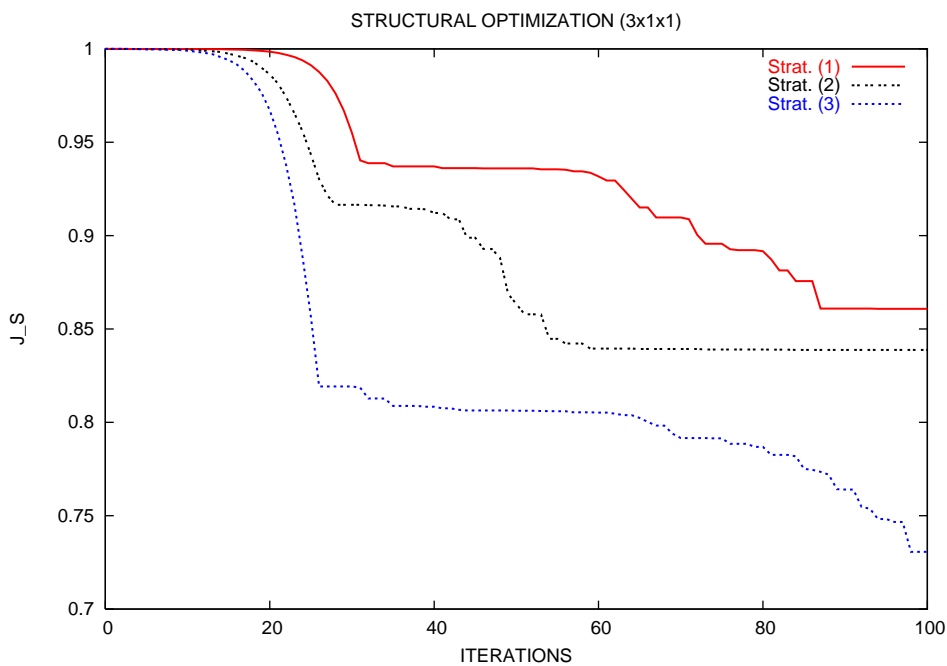


FIGURE 6.6 – Comparaison entre les courbes de convergence pour une optimisation purement structurale correspondant à trois différentes stratégies

**Test A :** On résout le problème d'optimisation concurrente en utilisant un jeu de Nash. On initialise le processus par une forme qui résulte d'une optimisation purement aérodynamique associée à une paramétrisation de degré  $3 \times 1 \times 1$ . À chaque itération de Nash, on effectue 15 et 10 itérations du simplexe (de Nelder-Mead) respectivement pour l'optimisation structurale et l'optimisation aérodynamique. L'écoulement est calculé en utilisant un méta-modèle.

On montre dans la figure (6.7) les courbes de convergence obtenues par les deux joueurs dans un jeu de Nash, après 40 itérations de Nash (en haut) et 100 itérations de Nash (en bas). On constate que 100 itérations ne sont pas suffisantes pour atteindre l'équilibre entre les deux disciplines. Néanmoins, une certaine stabilité est observée pendant les 20 dernières itérations de Nash, et la solution obtenue correspond à une dégradation inférieure à 10% du critère  $J_A$ , et une amélioration du critère structurale de 6%. Les sauts observés dans la courbe de convergence sont dus à des petites violations de la contrainte sur la portance sous l'effet de l'optimisation structurale.

On donne dans la table (6.1) les valeurs des coefficients aérodynamiques et de la fonction objectif associées à la forme obtenue après 40 et 100 itérations de Nash. On constate une dégradation de la traînée qui reste acceptable vu que la forme initiale résulte d'une optimisation aérodynamique. Cela se traduit par une légère dégradation de la zone supersonique sur l'extrados de l'aile comme le montre la figure (6.8), qui montre le champ de Mach sur la surface de l'aile optimisée après 40 et 100 itérations de Nash. Les figures (6.9) et (6.10) donnent les formes de trois sections de la forme optimisée à l'emplanture, à 50% de l'envergure et à l'extrémité. On observe une variation de la forme sur l'extrados et l'intrados à l'extrémité et à l'emplanture de l'aile.

	$C_L$	$C_D$	$J_A$	$J_S$
Configuration initiale	0.318952941	0.012764486	1.	1.
Après 40 itérations de Nash	0.318830971	0.013234922	1.03685506	0.976376671
Après 100 itérations de Nash	0.318634504	0.014047362	1.10050351	0.948236509

TABLE 6.1 – Valeurs des coefficients aérodynamiques et de la fonction objectif associées à la configuration initiale et optimale

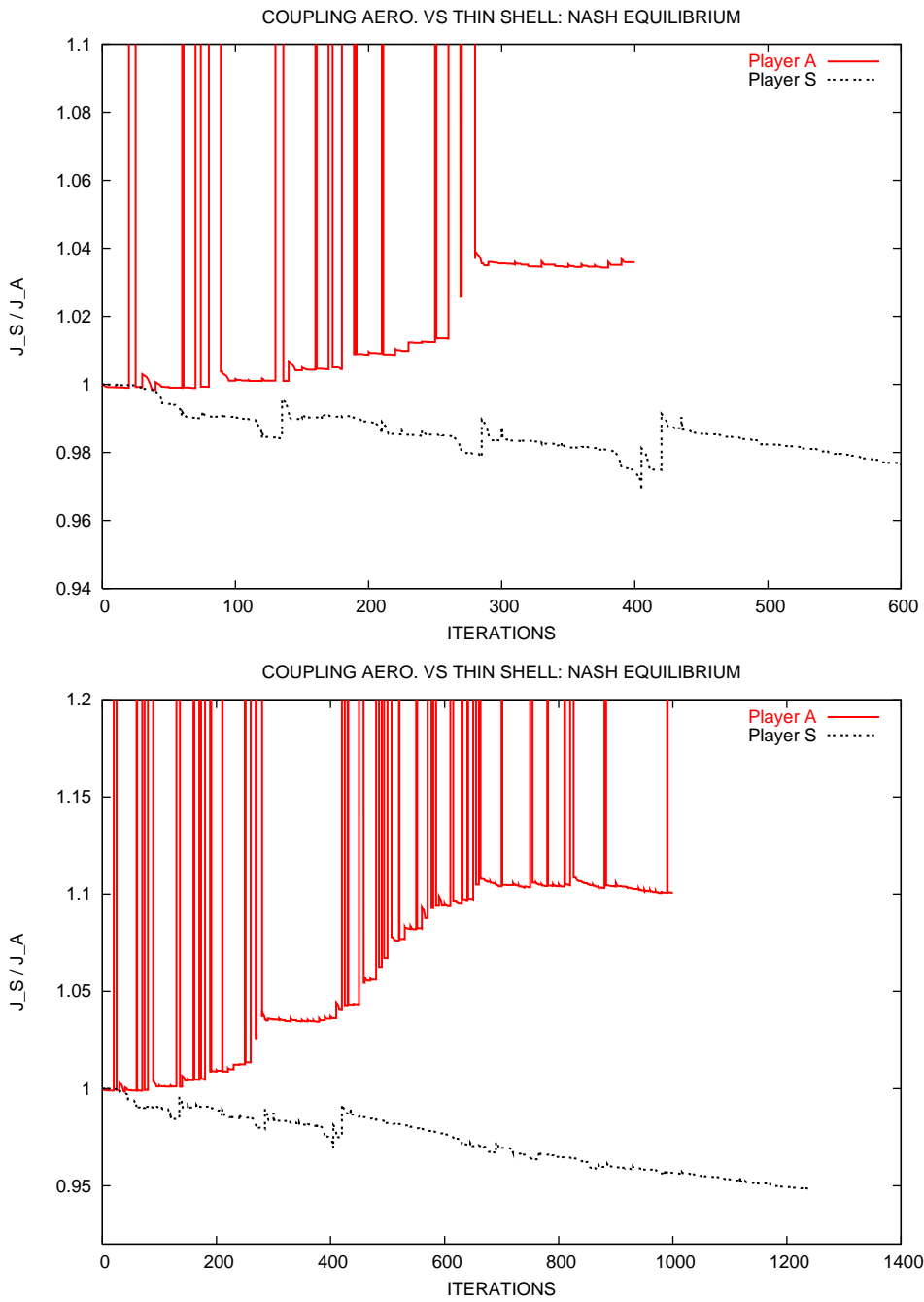


FIGURE 6.7 – Courbes de convergence du jeu de Nash après 40 itérations de Nash (en haut) et 100 itérations de Nash (en bas)

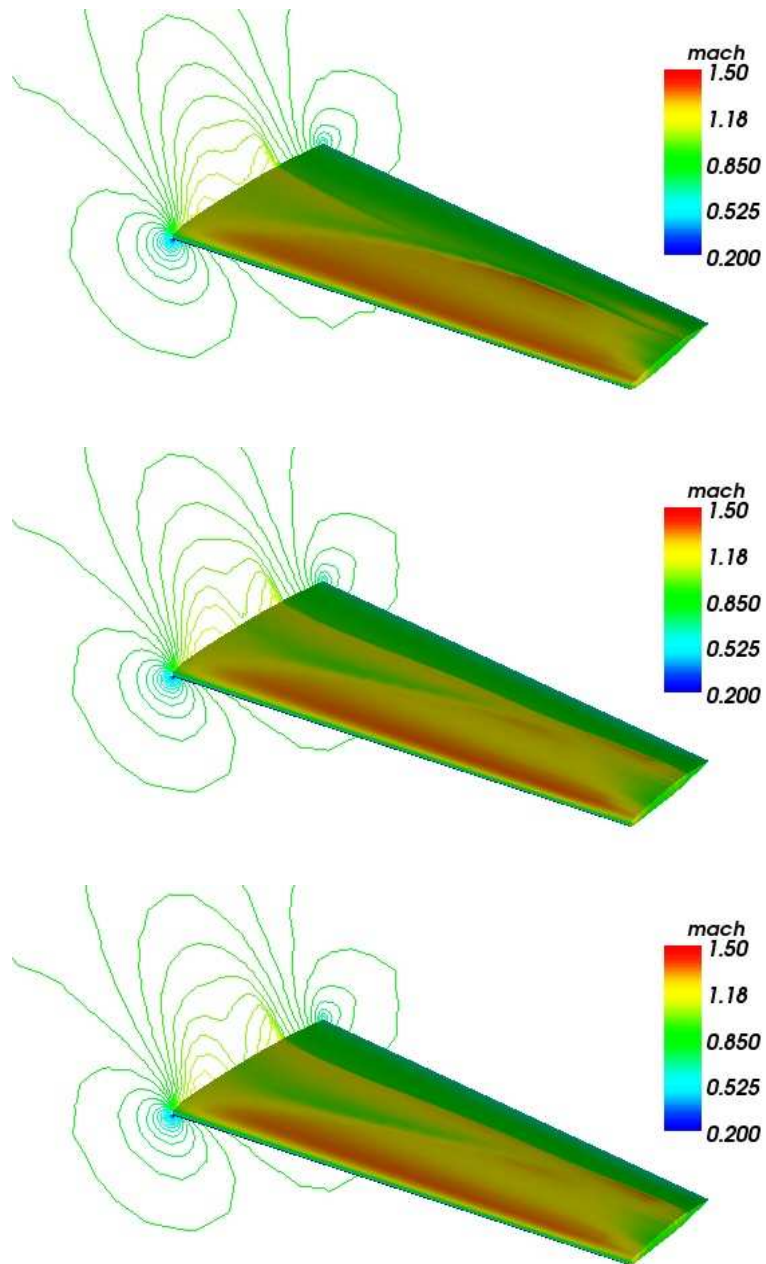


FIGURE 6.8 – Champ de Mach sur la surface de la forme obtenue après 100 itérations de Nash (en bas) et 40 itérations de Nash (au milieu) en comparaison avec la configuration initiale (en haut)

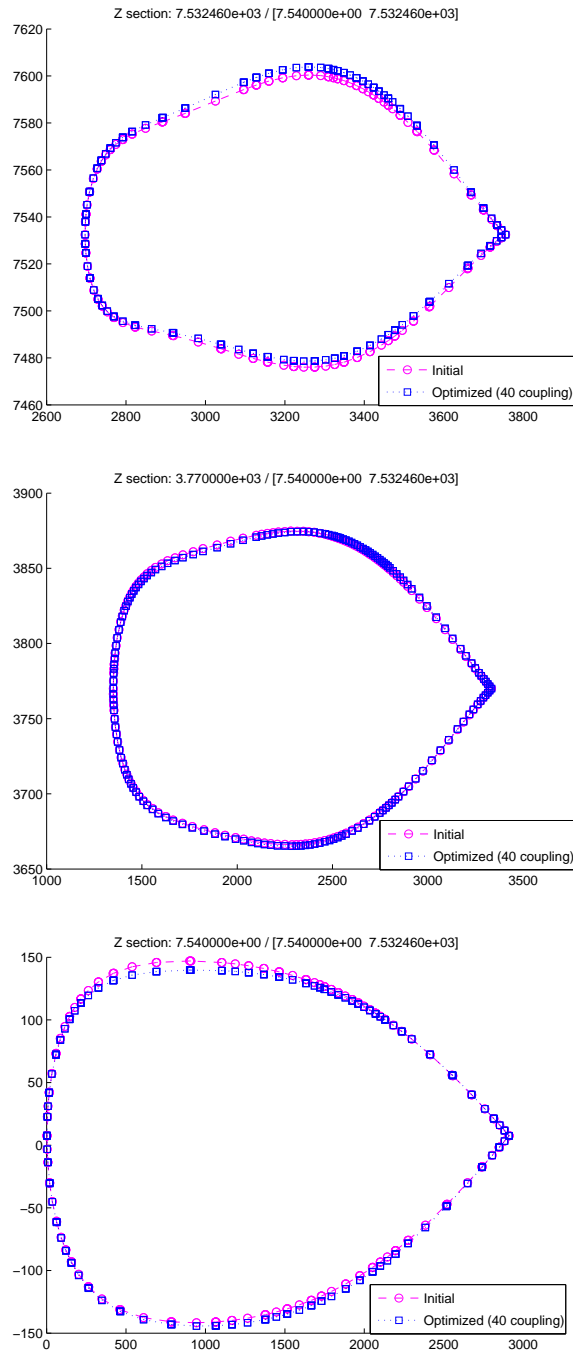


FIGURE 6.9 – Forme des sections de l’aile après 40 itérations de Nash : à l’emplacement (en bas), à 50% de l’envergure (au milieu) et à l’extrémité (en haut)

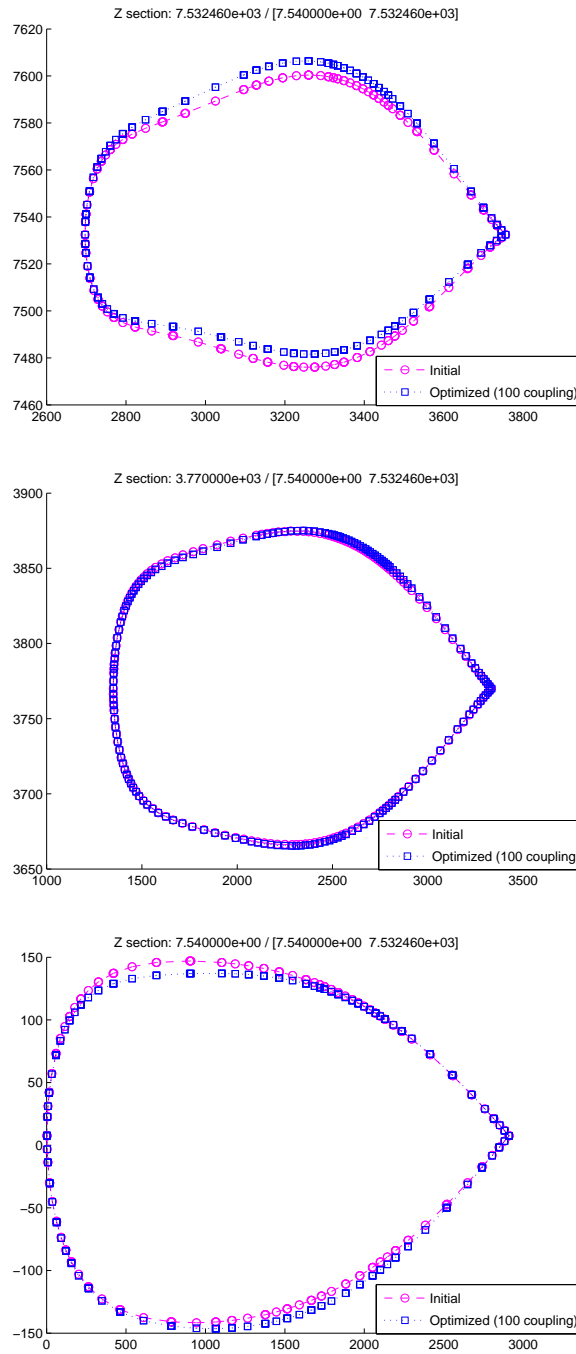


FIGURE 6.10 – Forme des sections de l’aile après 100 itérations de Nash : à l’emplacement (en bas), à 50% de l’envergure (au milieu) et à l’extrémité (en haut)

**Test B :** On utilise cette fois-ci un code CFD au lieu du meta-modèle pour le calcul de l'écoulement. Les paramètres de l'expérience sont identiques au test précédent.

On montre dans la figure (6.11) les résultats de convergence du jeu de Nash entre les deux disciplines. Après 26 itérations de Nash, le critère  $J_A$  s'est dégradé de moins de 5% et le critère  $J_S$  s'est amélioré d'environ 5%. On donne dans la table (6.2) les coefficients aérodynamiques et la valeur de la fonction objectif obtenus en comparaison avec la configuration initiale. On constate une augmentation de la traînée qui se traduit par une petite dégradation de l'intensité du choc sur l'extrados comme le montre la figure (6.12) qui donne le champ de Mach sur la surface de l'aile. La figure (6.13) montre trois sections de la forme initiale et optimisée à l'emplanture, à 50% de l'envergure et à l'extrémité. On observe une petite variation de la forme au voisinage de la zone du choc.

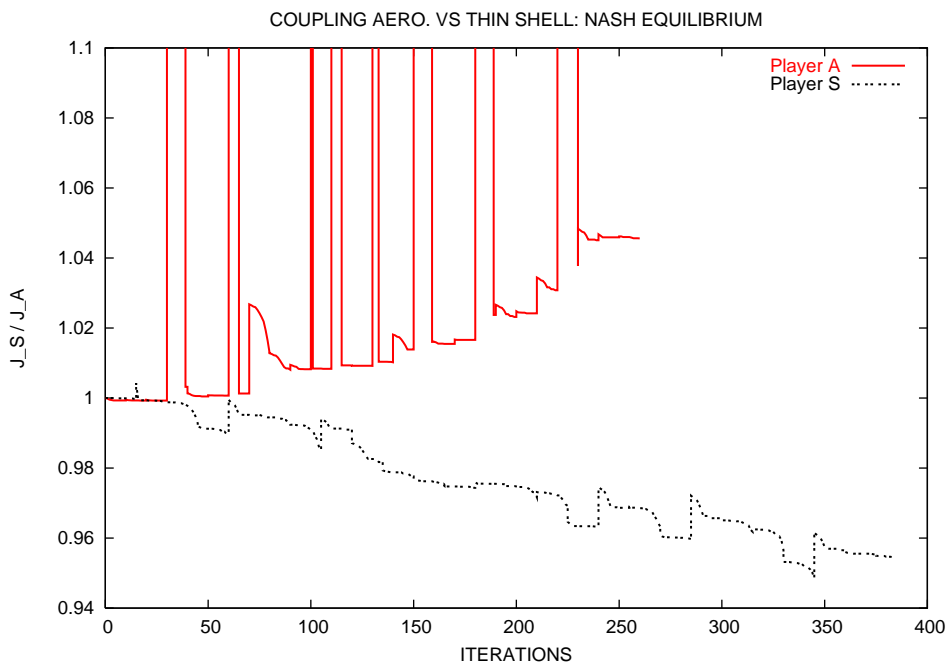


FIGURE 6.11 – Courbe de convergence obtenue par le jeu de Nash

	$C_L$	$C_D$	$J_A$	$J_S$
Configuration initiale	0.318952941	0.012764486	1.	1.
Après 26 itérations de Nash	0.319404488	0.013407172	1.045586660	0.954453732

TABLE 6.2 – Valeurs des coefficients aérodynamiques et de la fonction objectif associées à la configuration initiale et optimale



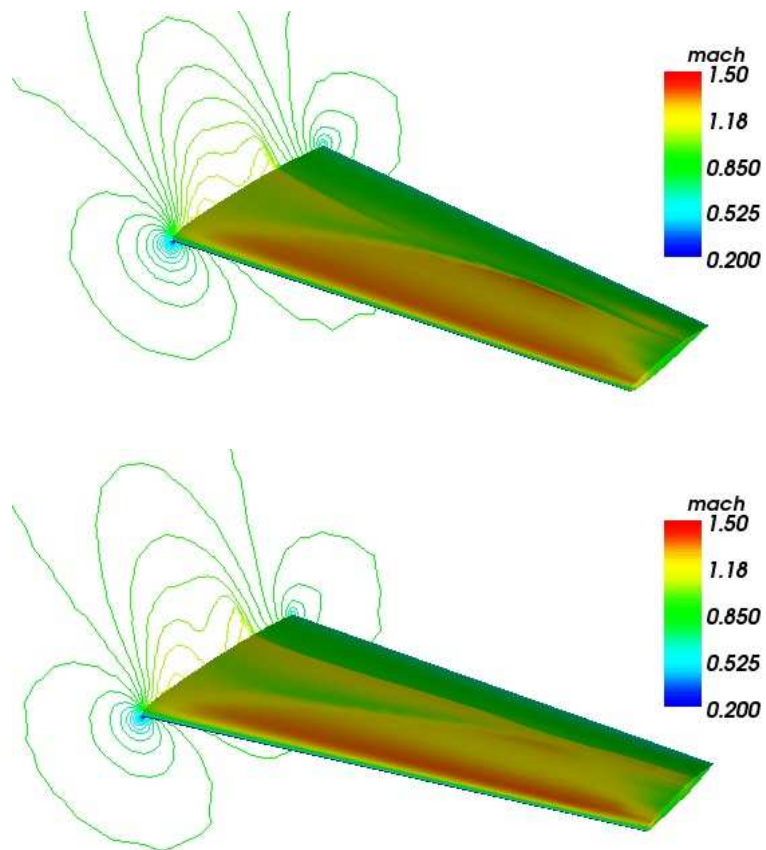


FIGURE 6.12 – Champ de Mach sur la surface de la forme obtenue après 26 itérations de Nash (en bas) en comparaison avec la configuration initiale (en haut)

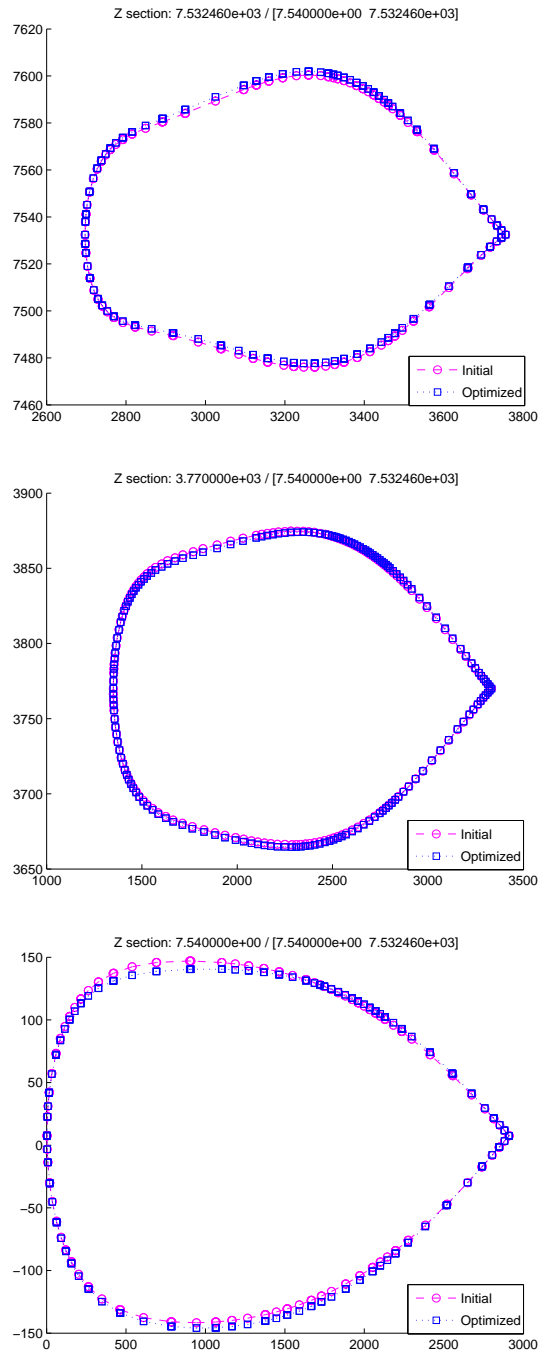


FIGURE 6.13 – Forme des sections de l’aile après 26 itérations de Nash : à l’emplanture (en bas), à 50% de l’envergure (au milieu) et à l’extrémité (en haut)

**Test C :** On initialise cette fois-ci le processus d'optimisation par un profil NACA0012 au lieu d'une forme optimisée. À chaque itération de Nash, on effectue 10 itérations du simplexe (de Nelder-Mead) pour les deux optimisations structurale et aérodynamique. L'écoulement est calculé en utilisant un code CFD.

On montre dans la figure (6.14) les courbes de convergence obtenues par le jeu de Nash. On constate que les deux joueurs ont pu améliorer leur propre critère. Donc après 100 itérations de Nash seulement, le critère  $J_A$  s'est amélioré d'environ 22% et le critère  $J_S$  de 16%. On donne dans la table (6.3) les coefficients  $C_L$  et  $C_D$  et la valeur de la fonctionnelle coût qui correspond à la configuration initiale et optimisée. Cette amélioration en fonction objectif se traduit par une réduction du choc sur l'extrados de l'aile comme le montre la figure (6.15) qui donne le champ de Mach sur la surface de l'aile associé à la forme initiale et optimisée. La figure (6.16) montre trois sections de la forme initiale et optimisée à l'emplanture, à 50% de l'envergure et à l'extrémité. On constate une variation de la forme assez sensible à l'emplanture de l'aile.

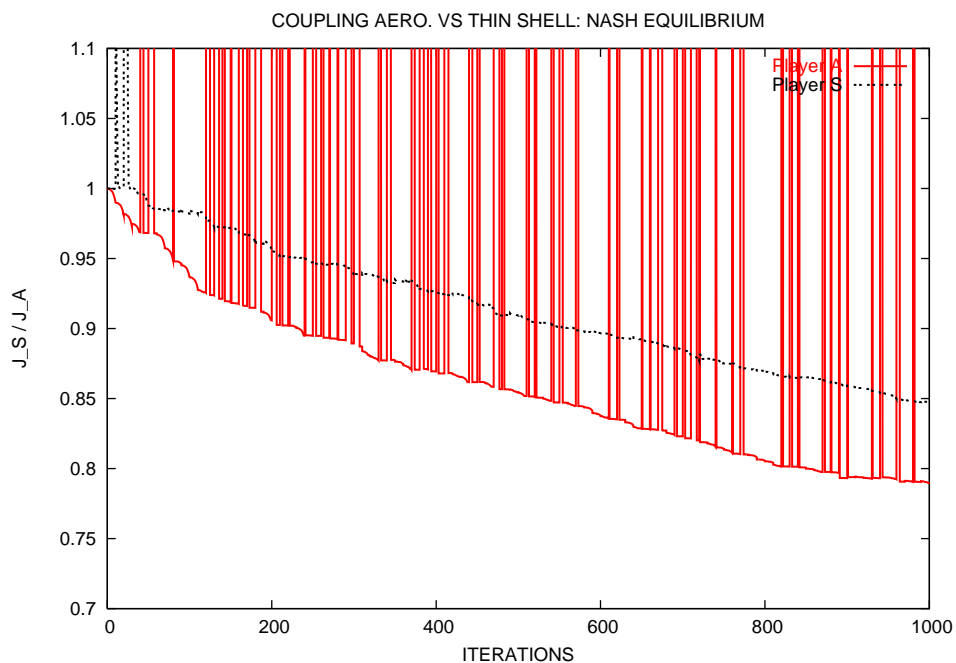


FIGURE 6.14 – Courbes de convergence obtenues par le jeu de Nash

	$C_L$	$C_D$	$J_A$	$J_S$
Configuration initiale	0.319751423	0.0263987218	1.	1.
Après 100 itérations de Nash	0.319554793	0.208816739	0.7890994595	0.8476323898

TABLE 6.3 – Valeurs des coefficients aérodynamiques et de la fonction objectif associées à la configuration initiale et optimale

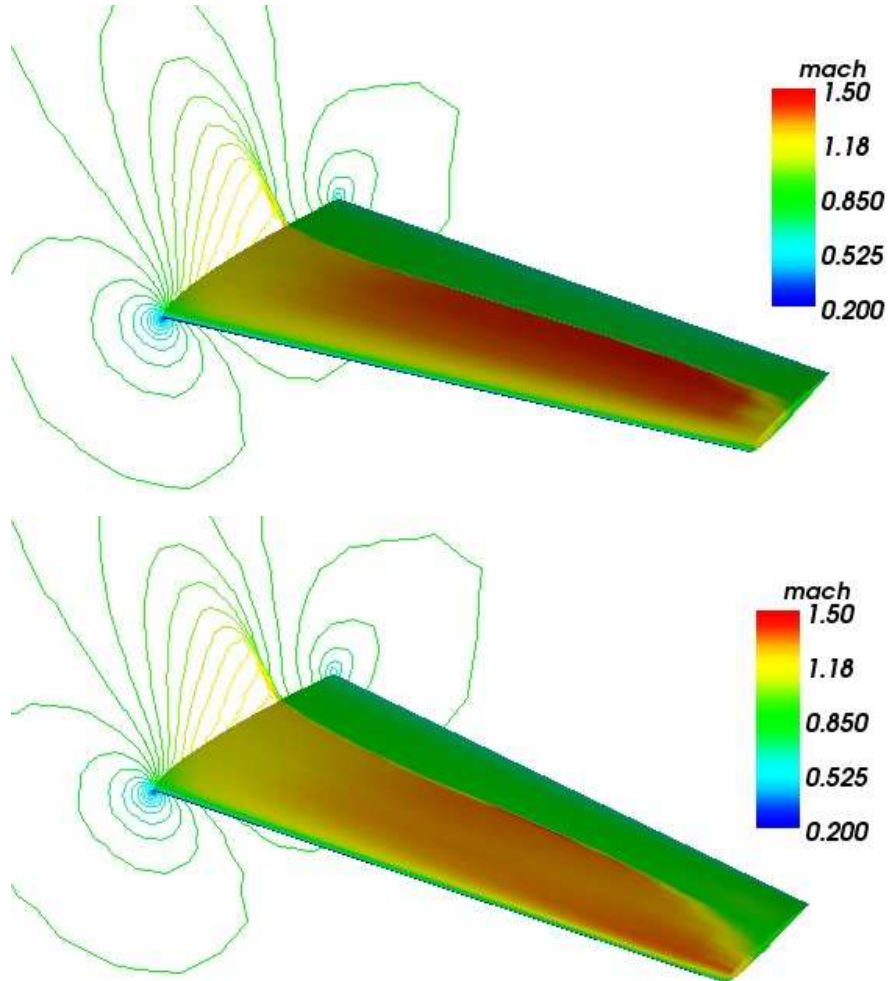


FIGURE 6.15 – Champ de Mach sur la surface de la forme initiale (en haut) et optimale (en bas)

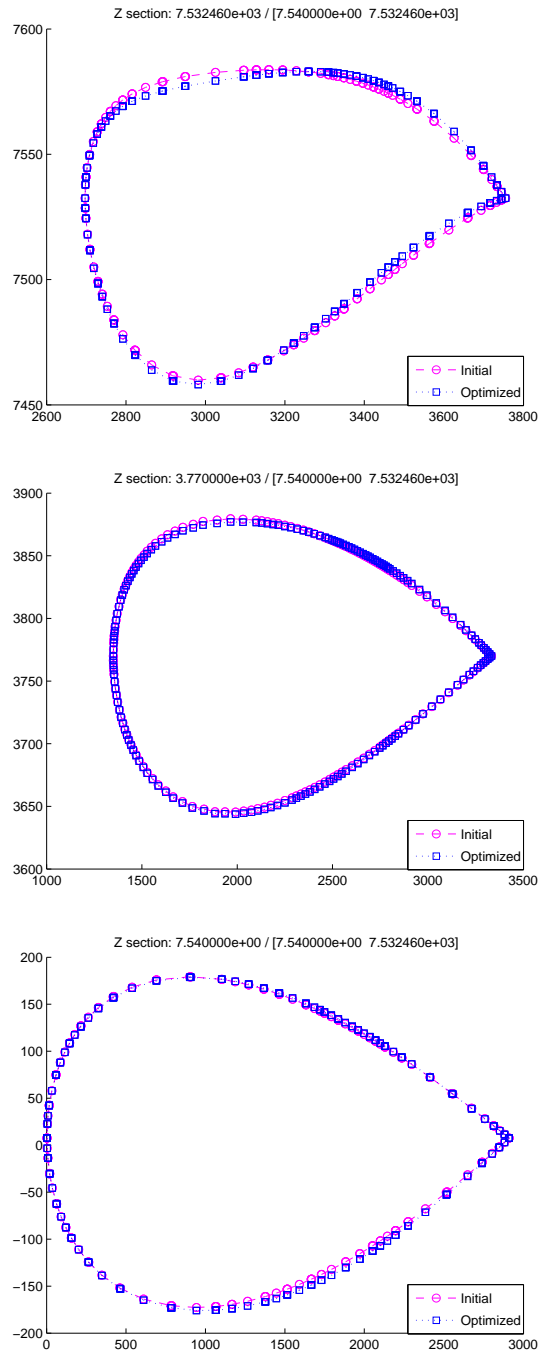


FIGURE 6.16 – Forme des sections de l’aile avant et après optimisation : à l’emplanture (en bas), à 50% de l’envergure (au milieu) et à l’extrémité (en haut)

**Test D :** On utilise dans ce cas test un jeu de Stackelberg pour résoudre le problème d'optimisation concourante. Pour chaque évaluation de la fonction coût  $J_A$ , on effectue une optimisation structurale avec 10 itérations du simplexe (de Nelder-Mead). Les autres paramètres de l'expérience sont identiques au test C.

On montre dans la figure (6.17) les courbes de convergence obtenues par le meneur et le suiveur du jeu de Stackelberg. On constate une amélioration des critères  $J_A$  et  $J_S$  respectivement de 10% et de 30%. C'est un algorithme coûteux puisque chaque évaluation de  $J_A$  nécessite la résolution d'un problème d'optimisation structurale. On donne dans la table (6.4) les coefficients aérodynamiques et la valeur de la fonctionnelle coût qui correspond à la configuration initiale et optimisée. Cette amélioration en fonction objectif se traduit par une réduction du choc sur l'extrados de l'aile comme le montre la figure (6.18) qui donne le champ de Mach sur la surface de l'aile associé à la forme initiale et optimisée. La figure (6.19) montre trois sections de la forme initiale et optimisée à l'emplanture, à 50% de l'envergure et à l'extrémité. On constate une petite variation de la forme surtout à l'emplanture et à l'extrémité.

	$C_L$	$C_D$	$J_A$	$J_S$
Configuration initiale	0.319751423	0.0263987218	1.	1.
Configuration Optimale	0.313169387	0.0235016738	0.9046297875	0.787030641

TABLE 6.4 – Valeurs des coefficients aérodynamiques et de la fonction objectif associées à la configuration initiale et optimale

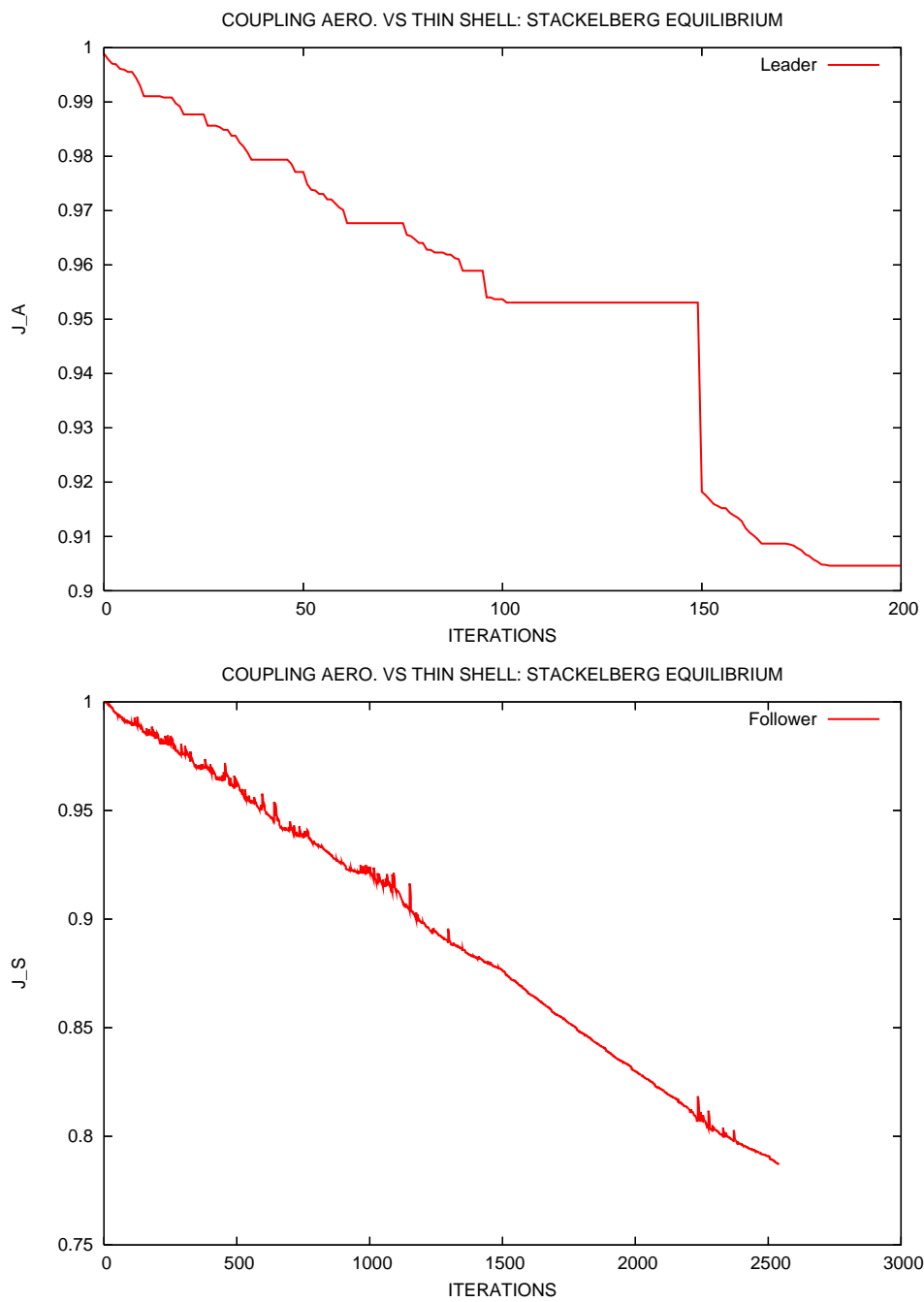


FIGURE 6.17 – Courbes de convergence obtenues par le jeu de Stackelberg : (en haut) pour le meneur, (en bas) pour le suiveur

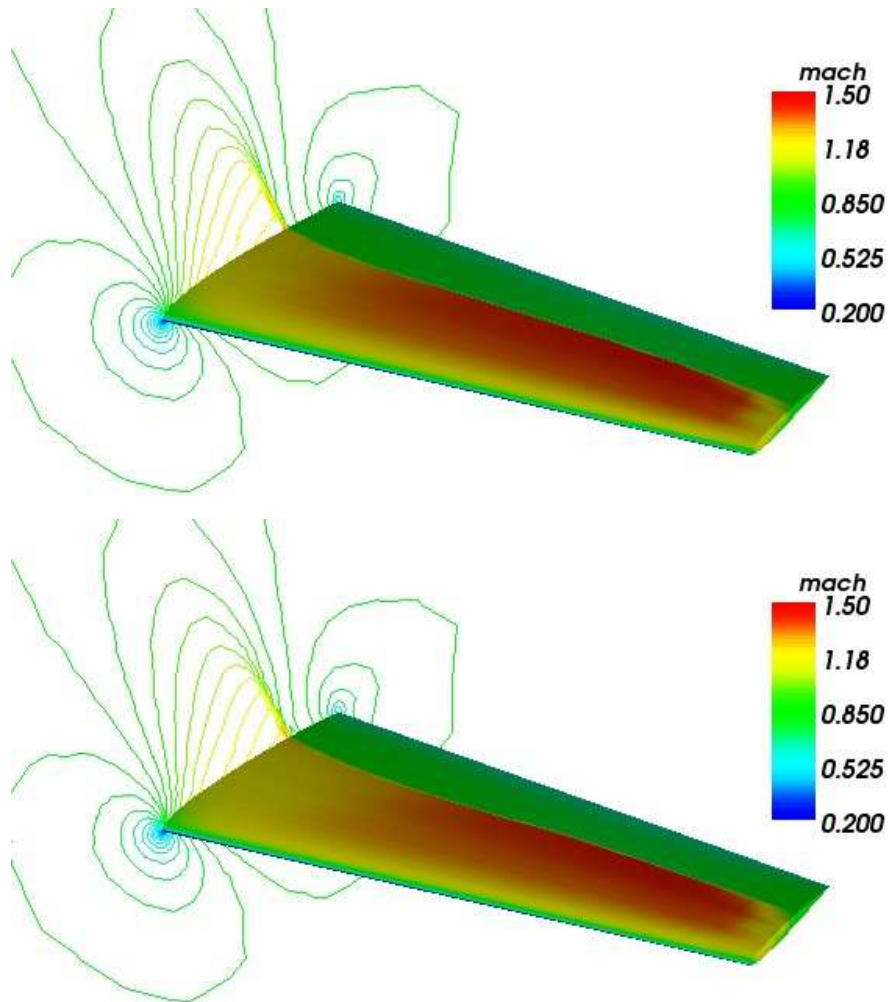


FIGURE 6.18 – Champ de Mach sur la surface de la forme initiale (en haut) et optimale (en bas)



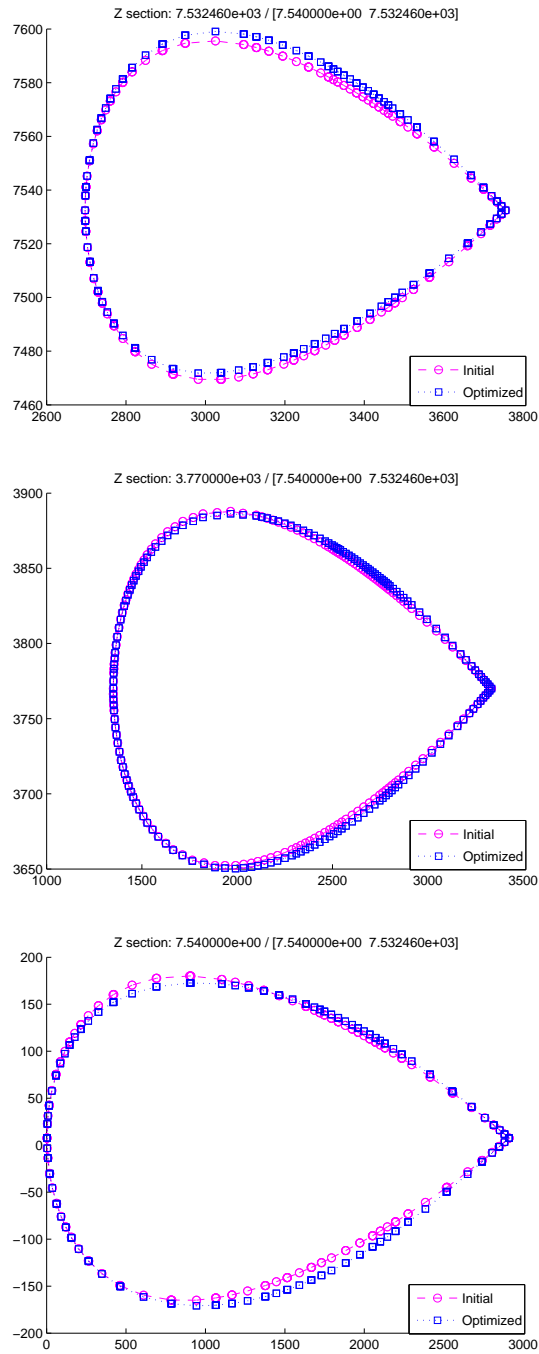


FIGURE 6.19 – Forme des sections de l’aile avant et après optimisation : à l’emplanture (en bas), à 50% de l’envergure (au milieu) et à l’extrémité (en haut)

Dans les cas test précédents, on a utilisé une stratégie de partage de territoires qui s'effectue directement sur les points de contrôle qui définissent la paramétrisation de forme. Quelques essais ont été effectués pour définir une meilleure stratégie afin que le couplage entre les deux disciplines considérées soit le moins violent possible. Il s'avère difficile avec cette procédure de définir la meilleure stratégie possible, et surtout pour un problème avec un nombre élevé de paramètres d'optimisation. On adopte dans la section suivante une nouvelle stratégie de partage plus pratique qui tient compte de l'importance du critère aérodynamique.

### 6.5.3.2 Seconde stratégie d'optimisation

On utilise maintenant une autre stratégie de partage de territoire, qui a été étudié en détail par Désidéri dans [30], et qui se base sur l'estimation du hessien du critère aérodynamique  $J_A$  et du gradient des contraintes au point de conception  $Y^*$ , qui résulte d'une optimisation purement aérodynamique dans l'espace paramétrique global. Ces estimations ont été calculées avec le même méta-modèle qu'on a utilisé dans les tests précédents. Cette stratégie est formulée par le théorème suivant qui donne une construction d'une matrice de partage adéquate de telle sorte que la dégradation du critère principal  $J_A$  soit la moindre possible.

#### Théorème 6.5.1 (de [30])

Soient  $n, p$  et  $K$  des entiers positifs tels que :

$$1 \leq p \leq n, \quad 0 \leq K < n + 1 - p \quad (6.26)$$

Soient  $J_A, J_B$  et, si  $K \geq 1$ ,  $\{g_k\}$  ( $1 \leq k \leq K$ )  $K + 2$  fonctions régulières à valeurs réelles de la variable vectorielle  $Y \in \mathbb{R}^{n+1}$ . Les fonctions  $J_A$  and  $J_B$  sont à valeurs positives, et on considère le problème d'optimisation principal suivant,

$$\min_{Y \in \mathbb{R}^{n+1}} J_A(Y) \quad (6.27)$$

soumis à aucune contrainte si  $K = 0$ , ou aux  $K$  contraintes d'égalité suivantes sinon :

$$g(Y) = (g_1, g_2, \dots, g_K)^T = 0 \quad (6.28)$$

On fait l'hypothèse que ce problème d'optimisation admet une solution locale ou globale au point  $Y_A^* \in \mathbb{R}^{n+1}$  et que  $J_A^* = J_A(Y_A^*) > 0$  et  $J_B^* = J_B(Y_A^*) > 0$ , et on note  $H_A^*$  la matrice hessienne du critère  $J_A$  en  $Y = Y_A^*$ .

Si  $K = 0$ , on pose  $P = I$ , matrice identité de dimension  $(n + 1) \times (n + 1)$ , et  $H'_A = H_A^*$ ; sinon, on suppose que les gradients de contraintes,  $\{\nabla g_k^*\}$  ( $1 \leq k \leq K$ ), sont linéairement indépendants, et on leur applique le processus d'orthogonalisation de Gram-Schmidt, et on note  $\{\omega^k\}$  ( $1 \leq k \leq K$ ) les vecteurs orthonormés qui en résultent. On définit alors la matrice  $P$  comme celle associée à l'opérateur de projection orthogonale sur le sous-espace de dimension  $K$  tangent aux hyper-surfaces  $g_k = 0$  ( $1 \leq k \leq K$ ) en  $Y = Y_A^*$ ,

$$P = I - \omega^1 \omega^{1T} - \omega^2 \omega^{2T} - \dots - \omega^K \omega^{KT} \quad (6.29)$$

et on considère la matrice réelle-symétrique suivante :

$$H'_A = P H_A^* P \quad (6.30)$$

Soit  $\Omega$  une matrice orthogonale dont les vecteurs colonnes sont des vecteurs propres normalisés de la matrice  $H'_A$  arrangés de telle sorte que les  $K$  premiers sont précisément  $\{\omega^k\}$  ( $1 \leq k \leq K$ ), et que les  $n+1-K$  suivants sont ordonnés par valeurs décroissantes de la valeur propre

$$h'_k = \omega^k \cdot H'_A \omega^k = \omega^k \cdot H_A^* \omega^k \quad (K+1 \leq k \leq n+1) \quad (6.31)$$

On considère le partage des paramètres défini comme suit :

$$Y = Y_A^* + \Omega \begin{pmatrix} U \\ V \end{pmatrix}, \quad U = \begin{pmatrix} u_1 \\ \vdots \\ u_{n+1-p} \end{pmatrix}, \quad V = \begin{pmatrix} v_p \\ \vdots \\ v_1 \end{pmatrix} \quad (6.32)$$

Soit  $\varepsilon$  un petit paramètre ( $0 \leq \varepsilon \leq 1$ ); on note  $Y_\varepsilon$  le point d'équilibre de Nash associé au problème d'optimisation concourante suivant :

$$\begin{cases} \min_{U \in \mathbb{R}^{n+1-p}} J_A \\ \text{Soumis à : } g = 0 \end{cases} \quad \text{et} \quad \begin{cases} \min_{V \in \mathbb{R}^p} J_{AB} \\ \text{Soumis à : aucune contrainte} \end{cases} \quad (6.33)$$

dans lequel à nouveau la contrainte  $g = 0$  est ignorée si  $K = 0$ , et

$$J_{AB} := \frac{J_A}{J_A^*} + \varepsilon \left( \theta \frac{J_B}{J_B^*} - \frac{J_A}{J_A^*} \right) \quad (6.34)$$

où  $\theta$  est un paramètre de relaxation strictement positif ( $\theta < 1$  : sous-relaxation;  $\theta > 1$  : sur-relaxation).

Alors :

- [Optimalité de la décomposition orthogonale] Si la matrice  $H'_A$  est semi-définie positive, ce qui est le cas en particulier si le problème d'optimisation principal est soumis à aucune contrainte ( $K = 0$ ), ou s'il est soumis à des contraintes d'égalité linéaires, ses valeurs propres ont la structure suivante :

$$h'_1 = h'_2 = \dots = h'_K = 0 \quad h'_{K+1} \geq h'_{K+2} \geq \dots \geq h'_{n+1} \geq 0 \quad (6.35)$$

et les derniers vecteurs propres associés  $\{\omega^k\}$  ( $K+1 \leq k \leq n+1$ ) admettent la caractérisation variationnelle suivante :

$$\begin{aligned} \omega^{n+1} &= \text{Argmin}_\omega |\omega \cdot H_A^* \omega| \quad \text{tq. } \|\omega\| = 1 \text{ et } \omega \perp \{\omega^1, \omega^2, \dots, \omega^K\} \\ \omega^n &= \text{Argmin}_\omega |\omega \cdot H_A^* \omega| \quad \text{tq. } \|\omega\| = 1 \text{ et } \omega \perp \{\omega^1, \omega^2, \dots, \omega^K, \omega^{n+1}\} \\ \omega^{n-1} &= \text{Argmin}_\omega |\omega \cdot H_A^* \omega| \quad \text{tq. } \|\omega\| = 1 \text{ et } \omega \perp \{\omega^1, \omega^2, \dots, \omega^K, \omega^{n+1}, \omega^n\} \\ &\vdots \end{aligned} \quad (6.36)$$

- [Préservation du point optimal par l'équilibre de Nash] Pour  $\varepsilon = 0$ , un équilibre de Nash existe, à savoir :

$$Y_0 = Y_A^* \quad (6.37)$$

- [Robustesse de la solution optimale d'origine] Si un point d'équilibre de Nash existe pour  $\varepsilon > 0$  et suffisamment petit, et s'il est une fonction régulière de ce paramètre, les fonctions :

$$j_A(\varepsilon) = J_A(Y_\varepsilon), \quad j_{AB}(\varepsilon) = J_{AB}(\bar{Y}_\varepsilon) \quad (6.38)$$

sont telles que :

$$j'_A(0) = 0 \quad (6.39)$$

$$j'_{AB}(0) = \theta - 1 \quad (6.40)$$

et

$$j_A(\varepsilon) = J_A^* + O(\varepsilon^2) \quad (6.41)$$

$$j_{AB}(\varepsilon) = 1 + (\theta - 1)\varepsilon + O(\varepsilon^2) \quad (6.42)$$

– [Représentation du continuum dans le cas de contraintes linéaires] Dans le cas de contraintes d'égalité linéaires, le point d'équilibre de Nash satisfait identiquement :

$$u_k(\varepsilon) = 0 \quad (1 \leq k \leq K) \quad (6.43)$$

$$\bar{Y}_\varepsilon = Y_A^* + \sum_{k=K+1}^{n+1-p} u_k(\varepsilon) \omega^k + \sum_{j=1}^p v_j(\varepsilon) \omega^{n+2-j} \quad (6.44)$$

– [Lien avec la Pareto-optimalité] Pour  $K = 1$  et  $p = n$ , le point d'équilibre de Nash  $\bar{Y}_\varepsilon$  est Pareto optimal.

On se place désormais dans les conditions particulière :  $\varepsilon = 1$  et  $\theta = 1$ . On prend  $J_A$  et  $J_{AB}$  comme étant respectivement le critère aérodynamique  $J_A$  et le critère structural  $J_S$ . On déduit de ce théorème que si, dans un jeu dynamique, le second critère  $J_S$  est minimisé dans un sous-espace engendré par  $\{\omega^k\}(K+1 \leq k \leq n+1)$ , alors la dégradation du premier critère  $J_A$  est du second ordre par rapport à sa valeur à l'optimum. Pour cela, Il suffit de prendre comme matrice de partage  $S$  (dans la formule (6.13)) la matrice dont les vecteurs colonnes sont les  $\{\omega^k\}(1 \leq k \leq n+1)$ .

**Test E :** On résout le problème d'optimisation concourante formulé comme un jeu de Nash en utilisant un méta-modèle pour le calcul d'écoulement. Le processus est initialisé par une forme qui résulte d'une optimisation purement aérodynamique, et qui correspond au vecteur de conception  $Y^*$ . À chaque itération de Nash, on effectue 5 et 15 itérations du simplexe respectivement pour l'optimisation aérodynamique et l'optimisation structurale.

On montre dans la figure (6.20) Les courbes de convergence des joueurs. Après une phase de conflit entre les deux joueurs, le jeu s'est stabilisé ensuite pour atteindre l'équilibre. On note que les sauts observés sur la courbe de convergence sont dûs à la violation de la contrainte sur la portance. À l'équilibre, le critère aérodynamique s'est dégradé de moins de 3% avec une amélioration d'environ 7% du critère structural. On donne dans la table (6.5) les valeurs des coefficients aérodynamiques et de la fonction objectif obtenues à l'équilibre en comparaison avec la configuration initiale. On constate une augmentation de la traînée qui reste acceptable comme on peut le voir dans la figure (6.21) qui donne le champ de Mach sur la surface de l'aile optimisée. La figure (6.22) montre trois sections de l'aile avant et après optimisation, à l'emplanture, à 50% de l'envergure et à l'extrémité.

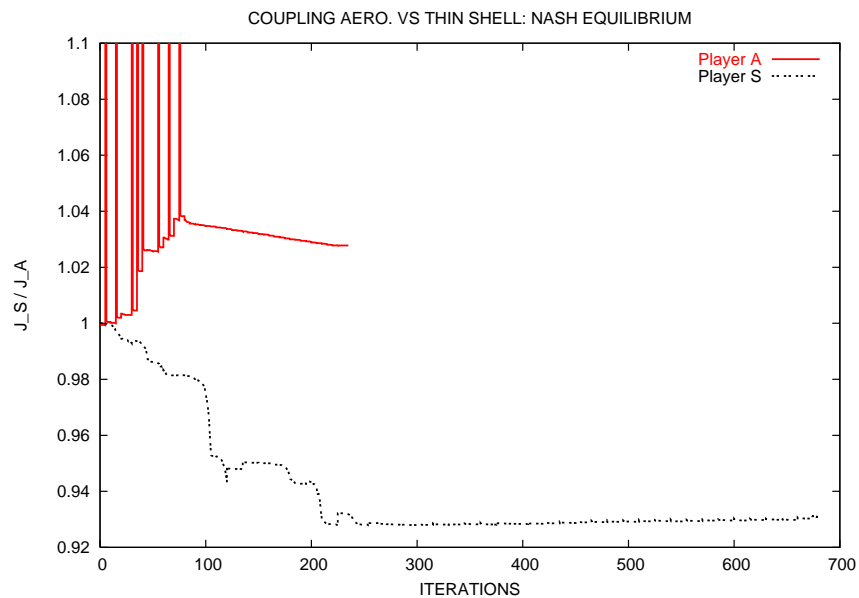


FIGURE 6.20 – Courbes de convergence du jeu de Nash

	$C_L$	$C_D$	$J_A$	$J_S$
Configuration initiale	0.318952941	0.012764486	1.	1.
À l'équilibre	0.319745070	0.013194608	1.027767848	0.930748139

TABLE 6.5 – Valeurs des coefficients aérodynamiques et de la fonction objectif associées à configuration initiale et optimale

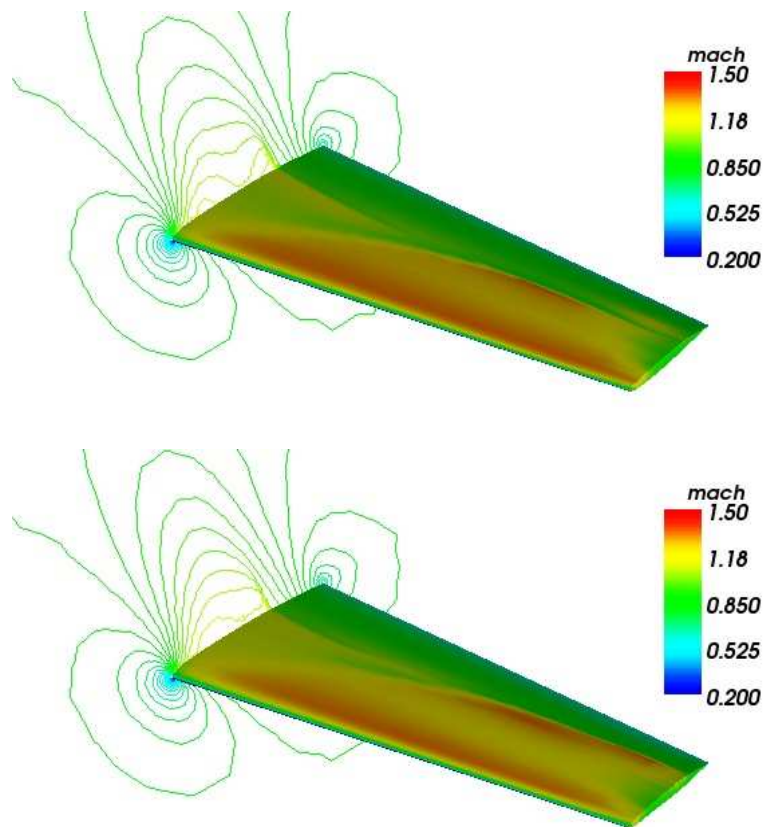


FIGURE 6.21 – Champ de Mach sur la surface de la forme obtenue à l'équilibre (en bas) en comparaison avec la configuration initiale (en haut)

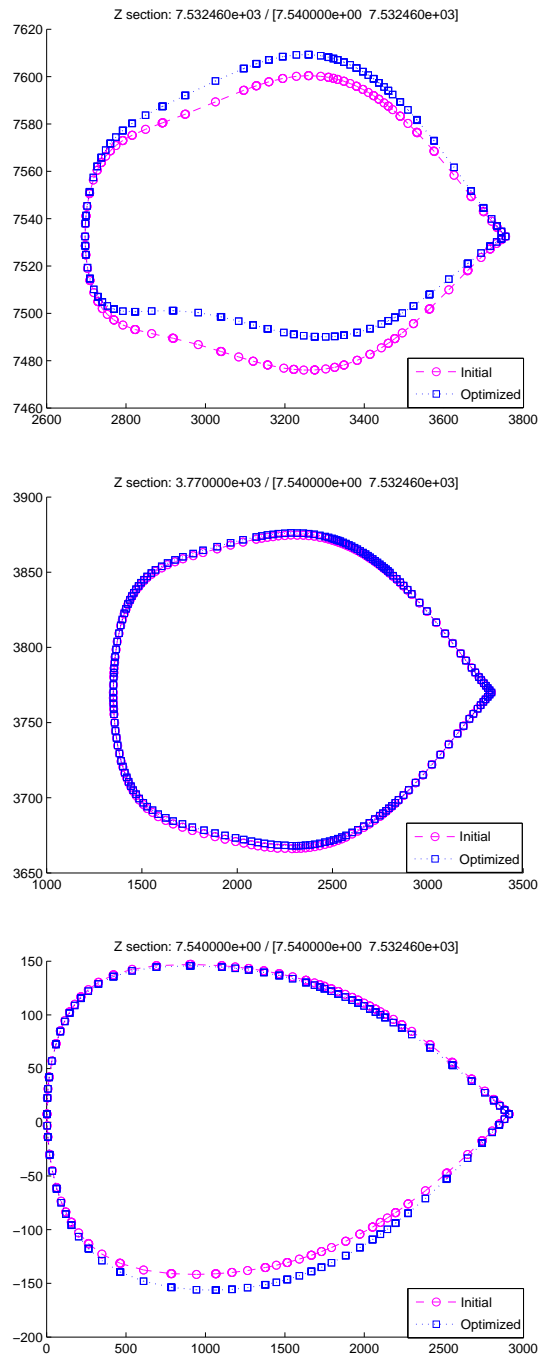


FIGURE 6.22 – Forme des sections de l’aile avant et après optimisation : à l’emplacement (en bas), à 50% de l’envergure (au milieu) et à l’extrémité (en haut)

**Test F :** On utilisant cette fois-ci le code CFD au lieu du méta-modèle pour le calcul de l'écoulement. les paramètres de l'expérience sont identiques au test E. On s'est limité dans ce cas test à 23 itérations de Nash seulement pour des raisons de temps CPU.

On montre dans la figure (6.23) les courbes de convergence obtenues par les deux joueurs. On constate une dégradation du critère  $J_A$  de moins de 3% et une amélioration du critère  $J_S$  d'environ 9%. Il est important de noter que les sauts observés ne sont pas dû à nature violente du couplage mais plutôt à la violation de la contrainte sur la portance qui se rétablit en quelques itérations. On note que 23 couplages n'est pas suffisant pour atteindre l'équilibre, mais la solution obtenue est acceptable comme on peut le voir dans la table (6.6) qui donne les valeurs des coefficients aérodynamiques et de la fonction objectif obtenues après 23 itérations de Nash en comparaison avec la configuration initiale. L'augmentation de la traînée est illustrée dans la figure (6.21) qui donne le champ de Mach sur la surface de l'aile optimisée. Quant à la figure (6.22), on montre trois sections de l'aile avant et après optimisation, à l'emplanture, à 50% de l'envergure et à l'extrémité. On remarque que les formes obtenues sont de même nature que celles du test précédent.

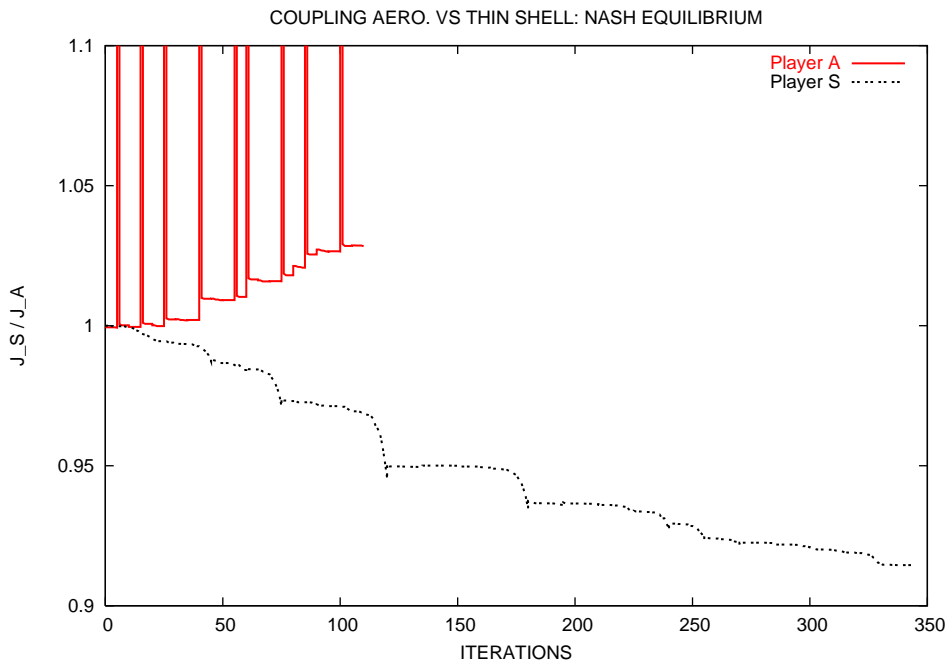


FIGURE 6.23 – Courbe de convergence du jeu de Nash après 50 itérations de Nash

	$C_L$	$C_D$	$J_A$	$J_S$
Config. initiale	0.318952941	0.012764486	1.	1.
Après 23 itérations de Nash	0.318884626	0.013129073	1.028363155	0.914534516

TABLE 6.6 – Valeurs des coefficients aérodynamiques et de la fonction objectif associées à la configuration initiale et optimale



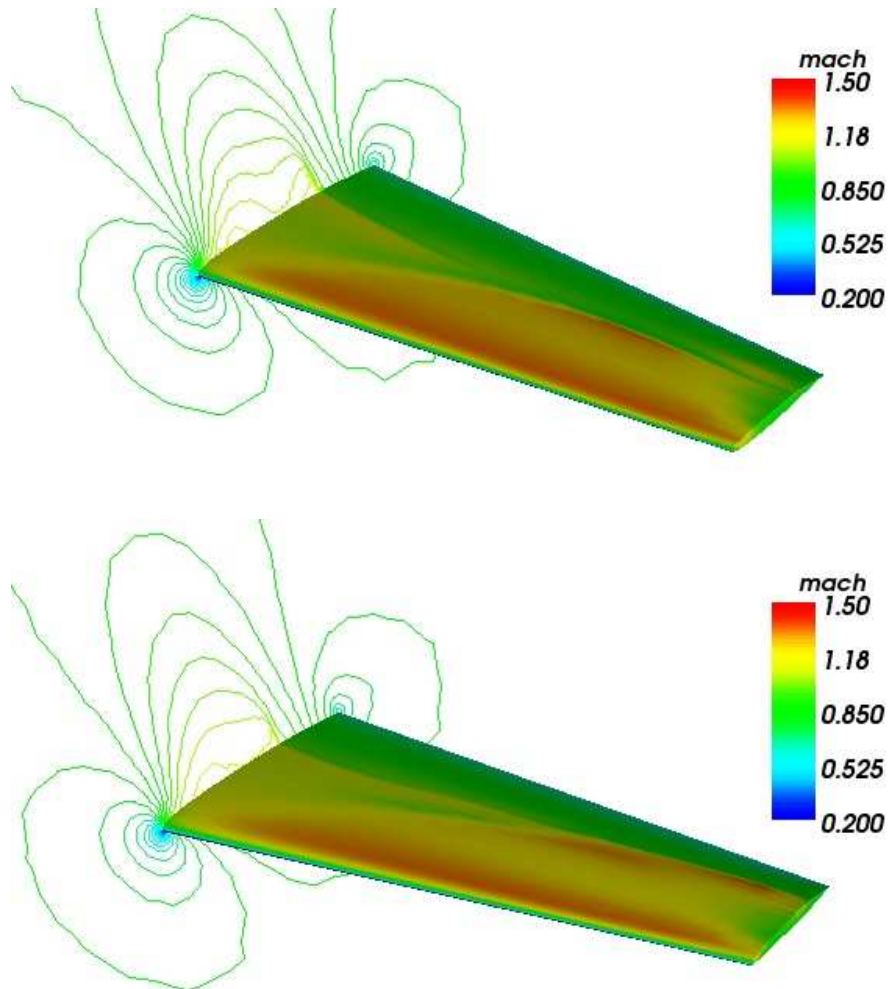


FIGURE 6.24 – Champ de Mach sur la surface de la forme obtenue après 23 itérations de Nash (en bas) en comparaison avec la configuration initiale (en haut)

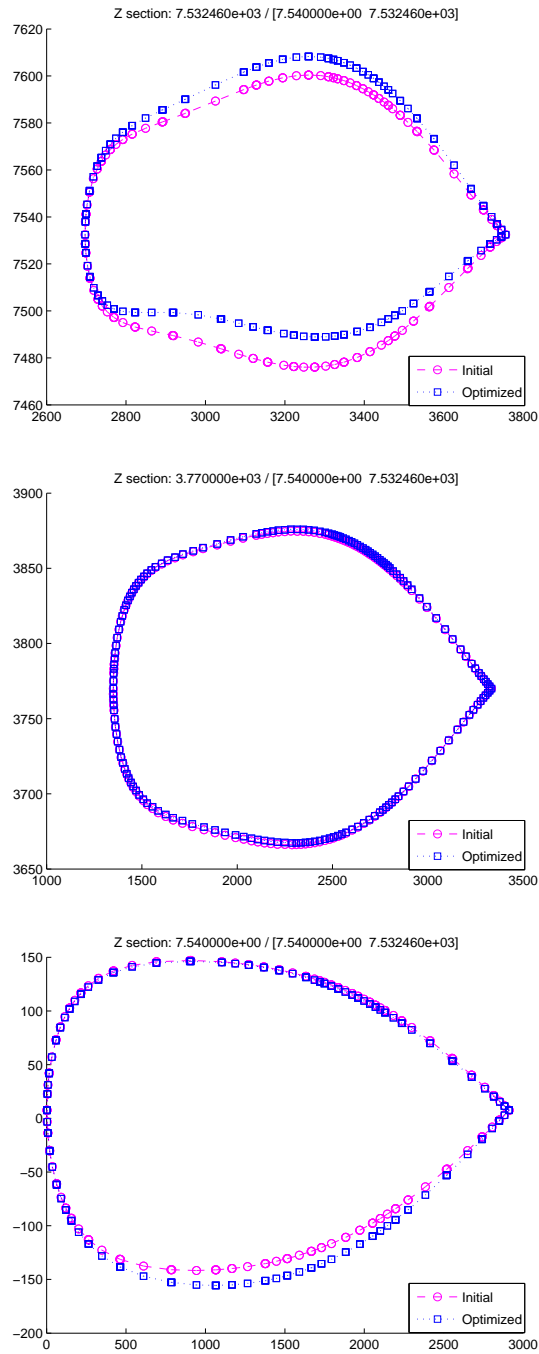


FIGURE 6.25 – Forme des sections de l’aile après 23 itérations de Nash : à l’implanture (en bas), à 50% de l’envergure (au milieu) et à l’extrémité (en haut)

## 6.6 Conclusion

On a étudié dans ce chapitre un problème d'optimisation concourante entre deux disciplines : l'aérodynamique et la mécanique de la structure. Ce problème d'optimisation multidisciplinaire a été résolu en utilisant la théorie des jeux dans un régime où l'aérodynamique est le plus important. On a adopté deux types de stratégie de partage de territoires entre les deux disciplines :

- La première s'effectue directement sur les points de contrôle qui définissent la boîte englobant la forme. Il s'avère difficile, avec cette stratégie, de définir la bonne répartition des points pour tenir compte de la nature du problème physique à résoudre où le critère aérodynamique est primordial. Pour cela, on a effectué plusieurs essais afin de trouver une meilleure stratégie pour un tel problème. Les résultats numériques montrent que le couplage est moins violent et que la solution obtenue après un certain nombre de couplages est aérodynamiquement satisfaisante. Cependant, cette stratégie de partage n'est pas pratique en général, puisqu'elle nécessite plusieurs essais préliminaires, surtout pour des problèmes avec un nombre important de variables d'optimisation.
- La seconde se base sur la construction d'une base de modes propres, liés aux résultats d'une optimisation purement aérodynamique dans l'espace paramétrique global, de telle sorte que la dégradation du critère principal soit du second ordre par rapport à sa valeur à l'optimum. Les résultats numériques montrent l'efficacité de cette approche qui est plus stable et plus systématique que la précédente. On atteint l'équilibre de Nash dans l'un des cas tests avec seulement 50 couplages.

L'étude théorique de la question d'existence de l'équilibre de Nash est délicate. Elle fait appel parfois à la résolution d'un algorithme itératif, comme dans [61][112], où plusieurs hypothèses de régularité sont nécessaires pour garantir la convergence vers un équilibre acceptable. Ces hypothèses ne sont pas satisfaites en général pour des problèmes pratiques comme le nôtre. Cependant, l'équilibre de Nash peut être obtenu en utilisant une certaine formulation d'optimisation concourante avec un choix spécifique de la stratégie de partage de territoires, du nombre d'itérations à accorder à chaque discipline, etc.

## Chapitre 7

# Conclusions et perspectives

Dans la première partie de la thèse, on s'est fixé comme objectif le développement des méthodologies numériques en optimisation de forme qui permettent non seulement d'accélérer la convergence itérative, mais aussi d'augmenter la précision de la solution optimale pour des applications où chaque évaluation de la fonction objectif est coûteuse. Comme par exemple en aérodynamique où la résolution d'un jeu d'EDP est nécessaire (ex. les équations d'Euler) pour calculer les coefficients aérodynamiques de la configuration à optimiser. Cet objectif a été atteint en utilisant deux concepts différents :

- Le premier se base sur une construction hiérarchique à l'instar des méthodes multigrilles. On représente la forme par une succession de paramétrisations, de Bézier ou de type Bézier-tensoriel (FFD), qui sont rigoureusement emboîtées selon plusieurs schémas. Le schéma le plus simple se base sur un enrichissement progressif de paramétrisation en considérant plusieurs niveaux de degrés différents. Ce schéma utilise initialement une paramétrisation grossière pour accélérer la convergence puisque le processus est moins raide avec un petit nombre de variables de conception. La solution obtenue est reconstruite exactement au niveau suivant en utilisant le processus d'élévation du degré et ainsi de suite. Un autre schéma plus complet consiste à incorporer à chaque niveau un V-cycle de paramétrisation qui utilise une correction au niveau grossier pour accélérer davantage la convergence. Les différentes expérimentations numériques sur des cas tests variés ont démontré que les stratégies multiniveaux permettent d'atténuer la raideur numérique en optimisation de forme, qui s'accroît rapidement avec le nombre de degrés de liberté.
- Le second se base sur des techniques d'adaptation de paramétrisation. Ce concept a été présenté et expérimenté dans un contexte monodimensionnel où la forme est représentée directement par une paramétrisation de Bézier classique, et en 3D où la forme est représentée en utilisant la technique Free-Form Deformation (FFD). L'idée de base consiste à réajuster le support de paramétrisation (en 1D) ou la carte FFD (en 3D) afin de régulariser la représentation de forme. En couplant cette procédure d'adaptation avec l'algorithme d'optimisation, on améliore considérablement la précision de la solution optimale. On a atteint avec une paramétrisation grossière des performances équivalentes à celles obtenues par une paramétrisation plus fine.

D'un point de vue théorique, une analyse spectrale d'un algorithme biparamétrisation idéal a été proposée dans [31] en utilisant un problème inverse qui consiste à minimiser une

fonctionnelle quadratique dont le gradient dépend linéairement du vecteur de conception. Cette analyse, sur un problème d'optimisation linéaire, a révélé que les modes propres de la matrice du système linéaire ressemblent à des modes de Fourier, mais la relation fréquence et valeur propre varient en sens inverse par rapport au cas classique d'une discrétisation d'EDP. Par conséquent, une nouvelle formulation d'un algorithme biparamétrisation idéal a été proposé en inversant cette relation. Il en résulte que la phase de relaxation, qui ressemble à une itération de Jacobi, soit à interpréter comme un opérateur de lissage. Cette approche a été testé avec succès dans [120].

On fixe donc comme perspective d'étendre cette analyse à un problème d'optimisation de forme non-linéaire en s'inspirant de la méthode FAS (Full Approximation Scheme) dû à Brandt [17]. Dans ce cas, l'algorithme multiniveau peut être formulé d'une façon hybride en utilisant une méthode du gradient au niveau fin où quelques itérations sont suffisantes pour atténuer les composantes hautes fréquences de l'erreur. Quant à la correction au niveau grossier où il faut beaucoup d'itérations pour atténuer les composantes basses fréquences, on peut faire appel par exemple aux algorithmes génétique ou de type PSO.

Dans la deuxième partie de la thèse, nous avons étudié un problème d'optimisation multidisciplinaire entre l'aérodynamique et la mécanique de la structure. La première tâche qu'on a accomplie était d'effectuer une analyse structurale sur une aile modélisée comme une structure de coque mince, dont la surface moyenne correspond au maillage surfacique d'une voilure d'avion d'affaire, en utilisant le Code Aster. On s'est focalisé ensuite sur la définition et l'expérimentation de ce problème d'optimisation couplée en utilisant la théorie des jeux. Les deux critères à minimiser sont considérés comme deux joueurs qui interagissent entre eux parallèlement (jeu de Nash) ou hiérarchiquement (jeu de Stackelberg). La nature antagoniste du couplage a été résolue en utilisant une stratégie intelligente du partage de territoires. Cette dernière minimise l'amplitude des conflits entre les deux critères, et c'est ce qui permet d'atteindre l'équilibre avec un petit nombre de couplages.

On envisage comme perspective de traiter ce problème d'optimisation couplée fluide structure dans un cas plus réaliste où l'aile serait modélisée comme une structure mince de *coque rigide* par l'adjonction d'éléments structuraux internes (ESI) : barres, poutres, plaques, renforts et raidisseurs par exemple. La coque est rigide et transmet aux ESI, qui sont déformables, les sollicitations mécaniques auxquelles elle est soumise. L'optimisation structurale a pour but de définir une nouvelle géométrie de l'aile (peau et éléments structuraux internes) satisfaisante d'un point de vue mécanique et la plus fidèle possible au résultat de l'optimisation aérodynamique.

## **Chapitre 8**

# **Bibliographie**



# Bibliographie

- [1] Optimum aerodynamic design and parallel navier-stokes computations. In P. K. Chaviaropoulos S. Lanteri J.Périaux, G. Bugeda and b. Mantel, editors, *ECARP European Computational Aerodynamic Research Project*, Notes on Numerical Fluid Mechanics. Vieweg, Braunschweig/Wiesbaden, Germany, 1998.
- [2] N. M. Alexandrov and M. Y. Hussaini. *multidisciplinary design optimization : state of the art*. Philadelphia, Society for Industrial and Applied Mathematics, 1997.
- [3] N. M. Alexandrov and R. M. Lewis. Comparative properties of collaborative optimization and other approaches to mdo. Icase report no. 99-24, NASA Langley Research Center, 1999.
- [4] N. M. Alexandrov and R. M. Lewis. algorithmic perspectives on problem formulations in mdo. Aiaa 2000-4719, NASA Langley Research Center, 2000.
- [5] N.M. Alexandrov and R.M. Lewis. Analytical and computational aspects of collaborative optimization. Nasa/tm/226210104/2262000, National Aeronautics and Space Administration, April 2000.
- [6] M. Andreoli, A. Janka, and J.-A. Désidéri. Free-form deformation parametrization for multilevel 3d shape optimization in aerodynamics. Research report No. 5019, INRIA, 2003.
- [7] E. Arian and S. Ta'asan. Multigrid one-shot methods for optimal design problems : Infinite dimensional control. Technical Report No. 52-91, ICASE, 1994.
- [8] J.P. Aubin. *Mathematical methods of game and economic theory*. New York, North-Holland Publishing Co. - Amsterdam.
- [9] J.P. Aubin. *Dynamic Noncooperative Game Theory*. Second Edition, Academic Press, London and San Diego, 1995.
- [10] R.H. Bartels, C.J. Beatty, and B.A. Barsky. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann, 1987.
- [11] F. Béhlacène and J-A. Désidéri. Paramétrisation de bézier adaptative pour l'optimisation de forme en aérodynamique. Research report 4943, INRIA, 2003.
- [12] F. Béhlacène and J.-A. Désidéri. Paramétrisation de bézier adaptative pour l'optimisation de forme en aérodynamique. Research report no. 4943, INRIA, Sep. 2003.
- [13] F. Beux and A. Dervieux. Exact-gradient shape optimization of a 2d euler flow. *Finite Elements in Analysis and Design*, Vol. 12 :281–302, 1992.
- [14] F. Beux and A. Dervieux. A hierarchical approach for shape optimization. Research report no. 1868, INRIA, 1993.



- [15] H. Borouchari, M. Castro-Diaz, P.L. George, F. Hecht, and B. Mohammadi. Anisotropic adaptive mesh generation in two dimension for cfd. In *5th int. conf numerical grid generation in computational field simulations*, Mississippi State University, 1996.
- [16] A. Brandt. Multi-level adaptive technique for fast numerical solution to boundary-value problems. In H. Cabannes and R. Temam, editors, *Proc. of the Third International Conference on Numerical Methods in Fluid Mechanics*, 1972. Lecture Notes in Physics 18, Springer-Verlag, Berlin, 1973.
- [17] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, Vol. 31 :333–390, 1977.
- [18] R.D. Braun. Collaborative optimization : An architecture for large-scale distributed design. Phd thesis, stanford university, Department of Aeronautics and Astronautics, May 1997.
- [19] W.-L. Briggs. A multigrid tutorial. *SIAM*, 1987.
- [20] M.D. Buhmann. Radial basis functions. *Acta Numerica*, 9 :1–38, 2000.
- [21] M. Castro-Diaz, F. Hecht, and B. Mohammadi. Anisotropic grid adaptation for inviscid and viscous flows simulations. *Int. J. Numer. Meth. Fluids*, Vol. 25 :475491, 1995.
- [22] P. Chandrashekarappa and R. Duvigneau. Radial basis functions and kriging meta-models for aerodynamic optimization. Research report no. 6151, INRIA, 2007.
- [23] S.D. Conte and C. de Boor. *Elementary Numerical Analysis, an algorithmic approach*. McGraw-Hill Book Company, New York, 1972.
- [24] S. Coquillart. Extended free form deformation : a sculptural tool for 3d geometric modelling. *SIGGRAPH*, pages 187–196, 1990.
- [25] F. Courty, A. Dervieux, B. Koobus, and L. Hascoet. Reverse automatic differentiation for optimum design : from adjoint state assembly to gradient computation. *Journal of Optimization Methods and Software*, 18(5) :615–627, 2003.
- [26] E.J. Cramer, J.E. Dennis, P.D. Frank, R.M. Lewis, and G. R. Shubin. On alternative problem formulations for multidisciplinary design optimization. In proceedings of the fourth aiaa/usaf/nasa/oai symposium on multidisciplinary analysis and optimization, 1992.
- [27] J.E. Dennis and V. Torczon. Direct search methods on parallel machines. *SIAM Journal of Optimization*, 1(4) :448–474, 1991.
- [28] J.-A. Désidéri. *Modèles Discrets et Schémas Itératifs, Applications aux Algorithmes Multigrilles et Multidomaines*. Editions Hermes, Paris, 1998.
- [29] J.-A. Désidéri. Hierarchical optimum-shape algorithms using embedded bézier parameterization. In Y. Kuznetsov *et al*, editor, *Numerical Methods for Scientific Computing, Variational Problems and Applications*, CIMNE, Barcelona, 2003.
- [30] J.-A. Désidéri. Split of territories in concurrent optimization. Research report no. 6108, INRIA, Feb. 2007.
- [31] J.-A. Désidéri. Two-level ideal algorithm for parametric shape optimization. In *Proc. of Int. Advances in Numerical Mathematics*, Moscow, September 16-17, 2005.
- [32] J.-A. Désidéri and A. Janka. Parameterization for aerodynamic optimization – application to drag and noise reduction of transonic/supersonic business jet. In E. Heikkola *et*

- al, editor, *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004)*, Jyväskylä, 24-28 July 2004., 2004.
- [33] J.-A. Désidéri, B. Abou El Majd, and A. Janka. Nested and self-adaptive bézier paramétrization for shape optimization. *J. Comput. Phys.*, 2007, to appear.
- [34] J.-A. Désidéri and J.P. Zolésio. Inverse shape optimization problems and applications to airfoils. *Control and Cybernetics*, Vol. 34(No. 1), 2005.
- [35] N. Dicesare, O. Pironneau, and E. Polak. Consistent approximations and optimal shape design. Report lan-up6-98001, 1998.
- [36] V. Dolean and S. Lanteri. A domain decomposition approach to finite volume solution of the euler equations on unstructured triangular meshes. *Int. J. Numer. Meth. Fluids*, Vol. 37-6 :625656, 2001.
- [37] R. Duvigneau. Contribution à l'optimisation de forme pour des écoulements à forts nombres de reynolds autour de géométries complexes. Thèse de doctorat, Ecole Centrale de Nante, 17 Octobre 2002.
- [38] R. Duvigneau. Adaptive parameterization using free-form deformation for aerodynamic shape optimization. Research report 5949, INRIA, 2006.
- [39] J. Elliott and J. Peraire. Aerodynamique design using unstructured meshes. *AIAA*, pages 96–1941, 1996.
- [40] C. Farhat and C. Degand. A tree-dimensional torsional spring analogy method for unstructured dynamic mesh. *Computer and Structures*, 80 :305–316, 2002.
- [41] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne. Torsional springs for two dimensional dynamic unstructured fluid meshes. *Computational Methods in Applied Mechanics and Engineering*, pages 231–245, 1998.
- [42] G. Farin. *Curves and Surfaces for Computer-Aided Geometric Design - A Practical Guide*. Academic Press, Boston, 1990.
- [43] R.P. Fedorenko. A relaxation method for solving elliptic difference equations. *USSR Comp. Math. and Math. Phys.*, 1(1092), 1962.
- [44] R.P. Fedorenko. The rate of convergence of an iterative process. *USSR Comp. Math. and Math. Phys.*, 4(227), 1964.
- [45] L. Fezoui and B. Stoufflet. A class of implicit upwind schemes for euler simulations with unstructured meshes. *J. of Comp. Phys.*, Vol. 84 :174–206, 1989.
- [46] C.F. Fisher and A.S. Arena. On the transpiration method for efficient aeroelastic analysis using an euler sover. Technical report, AIAA 96-3436, 1996.
- [47] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, 1987.
- [48] E. Oñate G. Bugada. Optimum aerodynamic shape design for fluid flow problems including mesh adaptivity. *Int. J. Numer. Meth. Fluids*, Vol. 30 :161178, 1999.
- [49] A. Habbal, J. Petersson, and M. Thellner. Multidisciplinary topology optimization solved as a nash game. *Int. J. Numer. Meth. Engng*, 61(7) :949–963, 2004.
- [50] W. HackBusch. *Multigrid Methods and Applications*. Springer Verlag, Heidelberg, 1985.
- [51] R. T. Haftka, Z. Gurdal, and M. P. Kamat. *Elements of structural optimization*. Philadelphia, Academic Publishers, Dordrecht, 1990.

- [52] R.T. Haftka. Simultaneous analysis and design. *AIAA J.*, Vol. 23(No. 7) :1099–1103, July 1985.
- [53] L. Hascoet, M. Vázquez, and A. Dervieux. Automatic differentiation for optimum design, applied to sonic boom reduction. pages 85–94, Montreal, Canada, 2003. LNCS 2668, Springer Publish.
- [54] R.M. Hicks and P.A. Henne. Wing design by numerical airfoil optimization. *AIAA Paper 77-1247*, 1979.
- [55] R.M. Hicks, E. M. Murman, and G.N. Vanderplaats. An assesement of airfoils design by numerical optimization. Technical report, NASA TM-3092, July 1974.
- [56] W. Huffman, R. Melvin, D. Young, J. Johnson, J. Bussoletti, M. Bieterman, and C. Hilmes. Practical design and optimization in computational fluid dynamics. Technical report, AIAA 93-3111, 1993.
- [57] A. Iollo, A. Kuruvila, and S. Ta’asan. Pseudo-time method for optimal shape design using the euler equations. Technical Report No. 59-95, ICASE, Aug. 1995.
- [58] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3) :233–260, 1988.
- [59] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Comp.*, Vol. 3 :233–260, 1988.
- [60] E. Kansa. Motivation of using radial basis function to solve pdes. available at <http://rbf-pde.uah.edu/kansaweb.pdf>.
- [61] J.B. Krawczyk and S. Uryasev. Relaxation algorithms to find nash equilibria with economic applications. *Environnement Modeling and Assessment*, 5 :63–73, 2000.
- [62] D.G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *J. of Chem., Metal. and Mining Soc. of South Africa*, 52 :119–139, 1951.
- [63] I. Kroo, S. Altus, R. Braun, P. Gage, , and I. Sobieski. Multidisciplinary optimisation methods for aircraft preliminary design. *AIAA Paper 94-4325*, 1994.
- [64] G. Kuruvila, S. Ta’asan, and M.D. Salas. *Airfoil Optimization by One-Shot Method, Optimum Design Methods in Aerodynamics*. AGARD-FDP-VKI Special Course, April 25-29, 1994.
- [65] J. Lagarias, J. Reeds, M. Wright, and W. P.E. Convergence properties of the nelder-mead simplex algorithm in low dimensions. *SIAM Journal of Optimization*, 9 :112–147, 1998.
- [66] H. Lamousin and W. Waggenspack. Nurbs-based free form deformation. *IEEE Computer Graphics and Applications*, pages 95–108, 1994.
- [67] E. Larsson and B. Fornberg. A numerical study of some radial basis function based solution methods for elliptic pdes. *Computers and Mathematics with Applications*, 46(5-6) :891–902, 2003.
- [68] B. Van Leer. Towards the ultimate conservative differences scheme v : A second-order sequel to godunov’s method. *J. of Comp. Phys.*, Vol. 32 :361–370, 1979.
- [69] J. Lépine, F. Guibault, J-Y. Trépanier, and P. Pépin. Optimized nonuniform rational b-splines geometrical representation for aerodynamic design of wings. *AIAA Journal*, Vol. 39(No. 11), 2001.

- [70] M.E. Lores, P.R. Smith, and R.M. Hicks. Supercritical wing design using numerical optimization and comparison with experiment. *AIAA Paper 79-0065*, 1979.
- [71] D.J.C. MacKay. Gaussian processes - a replacement for supervised neural networks. *Mathematical Geology*, 1997. Tutorial lecture notes for NIPS 1997.
- [72] D.J.C. MacKay. Introduction to gaussian processes. volume 168 of *Neural Networks and Machine Learning*, pages 133–165. Springer, Berlin, 1998.
- [73] B. Abou El Majd, J-A. Désidéri, T.T. Do, L. Fourment, A. Habbal, and A. Janka. Multilevel strategies and hybrid methods for shape optimization and application to aerodynamics and metal forming. In *Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems Conference (EUROGEN 2005)*, Munich, Germany, September 12-14, 2005. FLM, Munich, 2005.
- [74] N. Marco and F. Beux. Multilevel optimization : Application to shape optimum design with a one shot method. Research report no. 2068, INRIA, 1993.
- [75] N. Marco and A. Dervieux. Multilevel parametrization for aerodynamic optimization of 3d shapes. Research report, INRIA, July 1996.
- [76] A. Marocco. Simulations numériques dans la fabrication des circuits à semiconducteurs. Research report 0305, INRIA.
- [77] M. Masmoudi, D. Auroux, and Y. Parte. The state of the art in collaborative design.
- [78] D. Matheron. Principales of geostatistics. *Economic Geol.*, 58 :1246–1268, 1963.
- [79] K. Mckinnon. Convergence of the nelder-mead simplex method to a nonstationary point. *SIAM Journal of Optimization*, 9 :148–158, 1999.
- [80] M. Minoux. *Programmation mathématique : théorie et algorithmes*. Dunod, Collection Technique et Scientifique des Télécommunications, 1989.
- [81] B. Mohammadi. A new optimal shape design procedure for inviscid and viscous turbulent flows. *International Journal for Numerical Methods in Fluids*, 25 :183–203, 1997.
- [82] B. Mohammadi and O. Pironneau. Mesh adaption and automatic differentiation in a cad-free framework for optimal shape design. *Int. J. Numer. Meth. Fluids*, Vol. 30 :127136, 1999.
- [83] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for fluids*. Oxford, Clarendon Press, 2001.
- [84] J.F. Nash. Non-cooperative games. *Annals of Mathematics*, 54 :289, 1951.
- [85] J.A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7 :308 :313, 1965.
- [86] J. Von Neumann. Zur theorie der gesellschaftsspiele. *Math. Ann.*, pages 100 :295–320, 1928.
- [87] H. Omre. Beysian kriging - merging observations and qualified guesses in kriging. *Mathematical Geology*, 19(1) :25–39, 1987.
- [88] M.J. Osborne and A. Rubinstein. *A course in game theory*. Massachusetts Institute of Technology, 1997.
- [89] J. Périaux. *Genetic algorithms and evolution strategy in engineering and computer science : recent advances and industrial applications*. John Wiley & Son Ltd, 1998.

- [90] W.D. Pilkey. *Formulas for stress, Strain and Structural Matrices*. New York, Wiley and Cons, 1994.
- [91] O. Pironneau and E. Polak. Consistent approximations and approximate functions and gradients in optimal control. *SIAM J. Control Optim.*, Vol. 41(2) :487510, 2002.
- [92] E. Polak. *Optimization : algorithms and consistent approximations*. Springer, New York, 1997.
- [93] P. Raj and B. Harris. Using surface transpiration with an euler method for cost-effective aerodynamic analysis. Technical report, AIAA 93-3506, 1993.
- [94] R.G. Ramos and J. Périaux. Nash equilibria for the multiobjective control of linear partial differential equations. *Journal of Optimization Theory and Applications*, 112(3) :457–498, 2002.
- [95] J. Reuther, M.J. Rimlinger, J.J. Alonso, and A. Jameson. Rapid cycle aerodynamic shape optimization of complex configuration via an adjoint formulation and parallel computing. NASA Computational Aerosciences Workshop, 1996.
- [96] P.L. Roe. Approximate riemann solvers, parameter vectors and differences scheme. *J. of Comp. Phys.*, Vol. 43 :357–371, 1981.
- [97] J.A. Samareh. A novel shape parametrization approach. Technical report, NASA TM-209116, 1999.
- [98] J.A. Samareh. Survey of shape parametrization technique for high-fidelity multidisciplinary shape optimization. *AIAA Journal*, Vol. 39(No. 5) :877–884, 2001.
- [99] T. Sederberg and S. Parry. Free form deformation of solid geometric models. pages 151–160, 1986.
- [100] T.W. Sederberg. *Computer Aided Geometric Design*. <http://tom.cs.byu.edu/tom/>.
- [101] T.W. Sederberg and S.R. Parry. Free-form deformation of solid geometric models. *Computer Graphics*, Vol. 20(No. 4) :151–160, 1986.
- [102] J. Shankar, C. Ribbens, and R. Haftka. Computational study of nonhierarchical decomposition algorithm. *Computational Optimization and Applications*, 1993.
- [103] J. Sobieszczanski-Sobieski and R.T. Haftka. Multidisciplinary aerospace design optimization : Survey of recent developments. *Structural Optimization*, 14 :1–23, 1997.
- [104] W. Song and A.J. Keane. A study of shape parametrization methods for airfoils optimisation. Technical report, AIAA.
- [105] W. Spendley, G.R. Hext, and F.R. Himsworth. Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics*, 7 :441–461, 1962.
- [106] H. Stève. Schémas implicites linéarisés décentrés pour la résolution des équations d’euler en plusieurs dimensions. Thèse de doctorat, Université de Provence Aix-Marseille I, 1988.
- [107] S. Ta’asan. One-shot methods for optimal control of distributed parameter systems : Finite dimensional control. Technical Report No. 2-91, ICASE, 1991.
- [108] Z.L. Tang and J-A. Désidéri. Towards self-adaptive parameterization of bézier curves for airfoil aerodynamic design. Research report no. 4572, INRIA, Sep. 2002.

- [109] Z.L. Tang, J.-A. Désidéri, and J. Périaux. Multicriterion aerodynamic shape design optimization and inverse problems using control theory and nash games. *Journal of Optimization, Theory and Applications (JOTA)*, 135(1), Oct. 2007.
- [110] R. Tappeta, S. Nagendra, J.E. Renaud, and K. Badhrinath. Concurrent sub-space optimization (csso) code usage in isight. N 97crd188, in Technical Information Series, GE Research and Developpement Center, January 1998.
- [111] V. Torczon. Multi-directional search : A direct search algorithm for parallel machines. Phd thesis, Houston, TX, USA, 1989.
- [112] S. Uryasev and R.Y. Rubinstein. On relaxation algorithms in computation of noncooperative equilibria. *IEEE Transactions on Automatic Control*.
- [113] G.N. Vanderplaats. Approximation concepts for numerical airfoil optimization. Technical report, NASA TP-1370, 1979.
- [114] G.N. Vanderplaats and R.M. Hicks. Numerical airfoil optimization using reduced number of design coordinates. Technical report, NASA TM-73151, July 1974.
- [115] M. Vázquez, A. Dervieux, and B. Koobus. Aerodynamical and sonic boom optimization of a supersonic aircraft. Research report 4520, INRIA, 2002.
- [116] J.L. Walsh, K.C. Young, J.I. Pritchard, H.M. Adelman, and W.R. Mantay. Integrated aerodynamic/ dynamic/structural optimization of helicopter rotor blades using multi-level decomposition. Technical report nasa tp 3465, NASA Langley Research Center, January 1995.
- [117] J.F. Wang. Optimisation distribuée multicritère par algorithme génétiques et théorie des jeux. application à la simulation numérique de problèmes d'hypersustentation en aérodynamique. Phd thesis, Université Paris VI, 2001.
- [118] P. Wesseling. *An Introduction to Multigrid Methods*. John Wiley & Sons, 1991.
- [119] H.Y. Wu, S.C. Yang, F. Lui, and H.M. Tsai. Comparaison of three geometric representation of airfoils for aerodynamic optimization. In *16th AIAA Computational Fluid Dynamics Conference*, Orlando, FL, June 23-26. AIAA 2003-4095.
- [120] J. ZHAO, J.-A. Désidéri, and B. Abou El Majd. Two level correction algorithms for model problems. Research report, to appear, INRIA, 2007.
- [121] J.L. Zhou, A.L. Tits, and C.T. Lawrence. User's guide for ffsqp version 3.7 : A fortran code for solving optimization programs, possibly minimax, with general inequality constraints and linear equality constraints, generating feasible iterates. Technical report.
- [122] C. Zillober. Software manual for scpip 2.3. Technical report, tr02-1, informatik, Universität Bayreuth, 2002. <http://www.uni-bayreuth.de/departements/math/czillober/abstracts/tr02-1>.



## Annexe A

# Continuité des transferts du hessien dans une optimisation de forme multiniveau

### A.1 Introduction

Comme exposé précédemment, l'une des clés de réussite de notre approche d'optimisation hiérarchique réside dans l'efficacité des transferts entre les espaces de paramétrisation. Cette question se résout en grande partie par l'utilisation de l'opérateur d'élévation du degré qui permet de construire des paramétrisations rigoureusement emboîtées. Pour satisfaire au mieux la qualité des transferts, il faut régler un certain nombre de paramètres qui dépendent généralement de l'optimiseur. Dans cette étude, on utilise comme optimiseur la méthode de Quasi-Newton, qui utilise une approximation du hessien par la méthode BFGS pour chaque itération. Cette méthode a principalement deux mérites : elle évite le calcul du hessien, qui peut être très coûteux pour des applications complexes, et la direction de recherche vérifie la propriété de descente puisque le hessien est par construction défini positif. En revanche, une telle méthode affecte considérablement la qualité des transferts dans une optimisation de forme multiniveau, ceci est dû à l'initialisation du hessien par la matrice identité pour chaque nouvel appel à l'optimiseur.

Le but de ce travail consiste à étudier cette question de continuité des transferts du gradient et du hessien en optimisation de forme hiérarchique et en auto-adaptation de la paramétrisation. On présente quelques résultats numériques avec le problème modèle pour illustrer ce concept.

### A.2 Opérateurs de transfert

On introduit dans le théorème suivant une construction de deux opérateurs qui permettent un transfert de paramétrisation d'un niveau à un autre. Le premier permet de construire des paramétrisations rigoureusement emboîtées dans le sens croissant, on le nomme par l'opérateur d'élévation du degré. Le second est l'opérateur de réduction du degré qui permet un transfert de paramétrisation approché dans le sens décroissant.



**Théorème A.2.1**

Soit  $X' = \{x'_k\}$  ( $k = 0, \dots, n'$ ) le support d'une paramétrisation de Bézier de degré  $n' = n + 1$ .

1. **Élévation du degré :** Il existe un support  $X = \{x_k\}$  ( $k = 0, \dots, n$ ) de paramétrisation de Bézier de degré  $n$  donnant  $X'$  par élévation du degré ssi la relation suivante est vraie :

$$\sum_{k=0}^{n+1} (-1)^k C_{n+1}^k x'_k = 0 \quad (\text{A.1})$$

Par ailleurs, si  $t_i$  ( $i = 0, \dots, n'$ ) est une partition de  $[0,1]$  (les  $t_i$  sont 2 à 2 distincts), alors on peut construire une matrice de Van der Monde généralisée inversible  $B_{n'}$  de dimension  $(n' + 1) \times (n' + 1)$  et une matrice rectangulaire  $B_{n',n}$  de dimension  $(n' + 1) \times (n + 1)$  telles que :

$$X' = (B_{n'})^{-1} B_{n',n} X \quad (\text{A.2})$$

avec

$$B_{n'} = (B_{n'}^j(t_i)) \quad (i = 0 \dots, n'; j = 0 \dots, n')$$

et

$$B_{n',n} = (B_n^j(t_i)) \quad (i = 0 \dots, n'; j = 0 \dots, n)$$

où

$B_n^k(t) = C_n^k t^k (1-t)^{n-k} \forall t \in [0,1]$  est le polynôme de Bernstein.

2. **Réduction du degré :** Dans le cas où (A.1) n'est pas satisfaite les fonctions  $x(t) = B_n(t)^T X$  et  $x'(t) = B_{n'}(t)^T X'$  ne sont pas identiques. Néanmoins, la relation suivante (moins forte) est vraie :

$$X = (\bar{B}_n)^{-1} \bar{B}_{n,n'} X' \quad (\text{A.3})$$

où les matrices  $\bar{B}_n$  et  $\bar{B}_{n,n'}$  ont les définitions suivantes :  
si  $q_i$  ( $i = 0, \dots, n$ ) est une partition de  $[0,1]$

$$\bar{B}_n = (B_n^j(q_i)) \quad (i = 0 \dots, n; j = 0 \dots, n)$$

et

$$\bar{B}_{n,n'} = (B_{n'}^j(q_i)) \quad (i = 0 \dots, n; j = 0 \dots, n')$$

En effet,

Soit  $X = \{x_k\}$  ( $k = 0, \dots, n$ ) le support d'une paramétrisation de Bézier de degré  $n$ .

Il existe un support  $X' = \{x'_k\}$  ( $k' = 0, \dots, n' = n + 1$ ) d'une paramétrisation de Bézier de degré  $n'$  obtenu par élévation du degré à partir du support  $X$ , autrement dit :

$$B_n(t)^T X = B_{n'}(t)^T X', \forall t \in [0,1] \quad (\text{A.4})$$

Soit  $\{t_i\}$  ( $i = 0, \dots, n'$ ) une partition de  $[0,1]$  et  $\{e_i\}$  ( $i = 0, \dots, n'$ ) est la base canonique de  $\mathbb{R}^{n'}$

Donc,

$$e_k B_n(t_k)^T X = e_k B_{n'}(t_k)^T X', \forall t_k (k = 0, \dots, n') \quad (\text{A.5})$$

Si on somme toutes les équations (A.5) pour  $k = 1, \dots, n'$ , on obtient la formule suivante :

$$B_{n'} X' = B_{n',n} X$$

avec

$$B_{n'} = (B_{n'}^j(t_i)) \quad (i = 0 \dots, n'; j = 0 \dots, n')$$

et

$$B_{n',n} = (B_n^j(t_i)) \quad (i = 0 \dots, n'; j = 0 \dots, n)$$

Or  $B_{n'}$  est une matrice de Vandermonde généralisée, donc inversible. d'où la relation (A.2).

Avec un raisonnement analogue au précédent, en inversant les rôles de  $n$  et  $n'$ , on aboutit à la formule (A.3).

On illustre dans les figures (A.1) et (A.2) les deux opérateurs d'élévation et de réduction du degré qui sont construits par le théorème précédent. On constate que les formes obtenues avant et après réduction (ou élévation) du degré sont superposées.

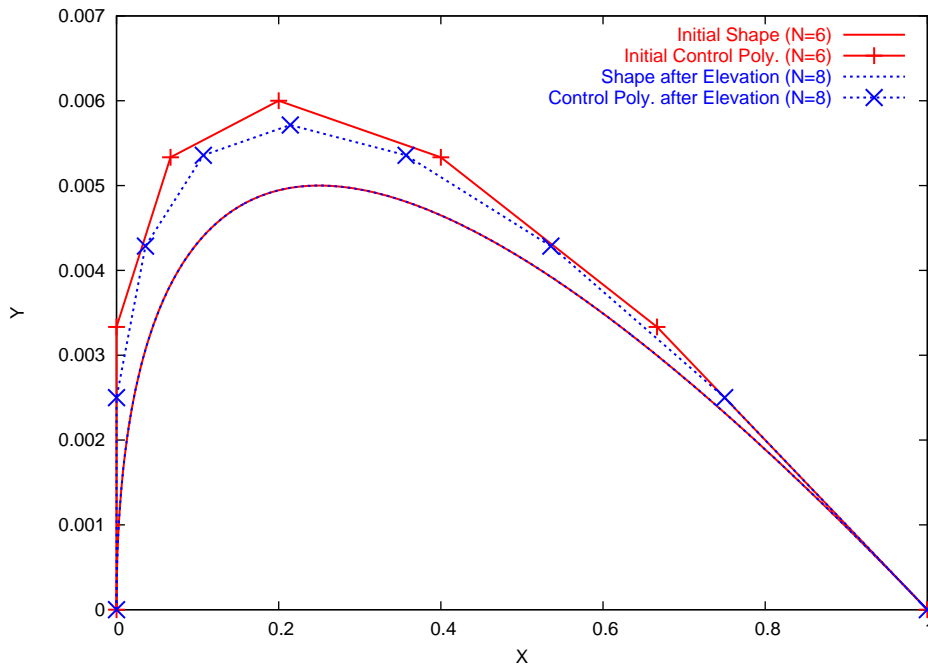


FIGURE A.1 – Illustration de l'opérateur d'élévation du degré :  $6 \rightarrow 8$

### A.3 Optimisation à support fixe

Soit  $(X^0, Y^0)$  un polygone de contrôle de degré  $n$ , associé à la courbe de Bézier  $\Gamma_0$ , qui résulte d'une optimisation de forme à support fixe  $X^0$ . Le but est d'exprimer les relations

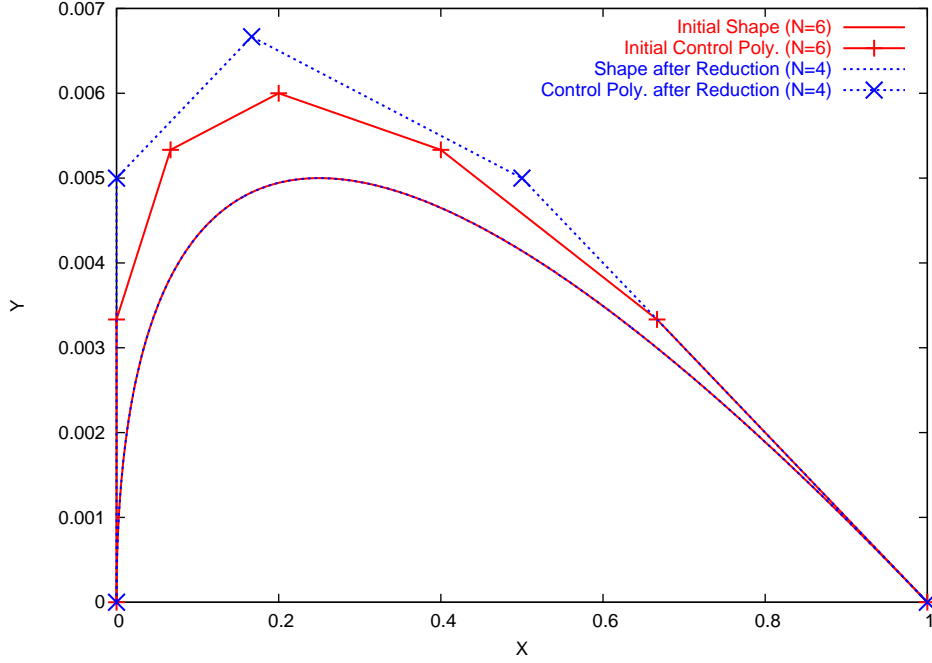


FIGURE A.2 – Illustration de l'opérateur de réduction du degré : 6  $\rightarrow$  4

de continuité du gradient et du hessien par passage à un niveau élevé afin de garantir un bon transfert de paramétrisation dans l'algorithme hiérarchique. Soit  $(X, Y)$  le polygone de contrôle de degré  $N$  ( $N > n$ ), associé à la même forme  $\Gamma$ , obtenu par  $N - n$  élévations du degré à partir de  $(X^0, Y^0)$ , qu'on note par  $E_n^N$ . Donc,

$$\begin{cases} B_n(t)^T X^0 = B_N(t)^T X \\ B_n(t)^T Y^0 = B_N(t)^T Y \end{cases} \quad (\text{A.6})$$

qui s'exprime d'une façon équivalente par,

$$\begin{cases} E_n^N X^0 = X \\ E_n^N Y^0 = Y \end{cases} \quad (\text{A.7})$$

alors, les deux courbes de Bézier  $\Gamma_0$  et  $\Gamma$  sont identiques. Il vient,

$$j_N(Y = E_n^N Y^0) = j_n(Y^0) \quad (\text{A.8})$$

avec  $j_n$  (resp.  $j_N$ ) est la fonctionnelle coût à minimiser dans  $\mathbb{R}^n$  (resp.  $\mathbb{R}^N$ ).

Au départ, on dérive l'équation (A.8) par rapport à  $Y^0$ , il vient,

$$\begin{aligned} \frac{dj_n(Y^0)}{dY^0} &= \frac{dj_N(Y)}{dY} \frac{dY}{dY^0} \\ &= \frac{dj_N(Y)}{dY} E_n^N \end{aligned} \quad (\text{A.9})$$

et, si on dérive une deuxième fois par rapport à  $Y^0$ , on obtient,

$$\frac{d^2 j_n(Y^0)}{d^2 Y^0} = (E_n^N)^T \frac{d^2 j_N(Y)}{d^2 Y} (E_n^N) \quad (\text{A.10})$$

en dérivant la relation (A.7) par rapport à  $Y$ , on a,

$$E_n^N \frac{dY^0}{dY} = I_N \quad (\text{A.11})$$

Notons  $R_N^n = \frac{dY^0}{dY}$ . Alors,

$$\frac{d j_N(Y)}{dY} = \frac{d j_n(Y^0)}{dY^0} R_N^n \quad (\text{A.12})$$

et

$$\frac{d^2 j_N(Y)}{d^2 Y} = (R_N^n)^T \frac{d^2 j_n(Y^0)}{d^2 Y^0} (R_N^n) \quad (\text{A.13})$$

et si la matrice  $\frac{d^2 j_n(Y^0)}{d^2 Y^0}$  est inversible, alors,

$$\left( \frac{d^2 j_N(Y)}{d^2 Y} \right)^{-1} = (E_n^N) \left( \frac{d^2 j_n(Y^0)}{d^2 Y^0} \right)^{-1} (E_n^N)^T \quad (\text{A.14})$$

Les équations (A.13) et (A.14) traduisent respectivement les relations de continuité du hessien et de son inverse dans une optimisation multiniveau à support fixe. Tout dépend de la méthode Quasi-Newton utilisée avec une mise à jour BFGS soit du hessien soit de son inverse. En pratique,  $E_n^N$  et  $R_N^n$  sont donnés par le théorème précédent, et qui correspondent respectivement à l'opérateur d'élévation et de réduction du degré. Or, dans une telle méthode, le hessien est par construction défini positif, ceci exige que le hessien obtenu par (A.13) ou (A.14) soit définie positive. Soit,

$$B_n = \left( \frac{d^2 j_n(Y^0)}{d^2 Y^0} \right)^{-1} \text{ et } B_N = \left( \frac{d^2 j_N(Y)}{d^2 Y} \right)^{-1}$$

La matrice  $B_n$  est par hypothèse définie positive, puisqu'elle correspond à une mise à jour BFGS, c'est aussi le hessien inverse associé à l'optimisation initiale correspondant à une paramétrisation de degré  $n$ .

alors, il existe  $W_n \in \mathcal{M}_{n,n}$  telle que,

$$B_n = W_n^T W_n \quad (\text{A.15})$$

donc,

$$\begin{aligned} B_N &= (E_n^N) B_n (E_n^N)^T \\ &= (E_n^N) W_n^T W_n (E_n^N)^T \\ &= \widetilde{W}_n^T \widetilde{W}_n \end{aligned} \quad (\text{A.16})$$

où  $\widetilde{W}_n = W_n (E_n^N)^T$

Soit  $\lambda$  une valeur propre de  $B_N$

donc, il existe  $x \in \mathcal{R}^N, x \neq 0$ , telle que  $B_N x = \lambda x$

ceci implique que,

$$x^T \widetilde{W}_n^T \widetilde{W}_n x = \lambda x^T x$$

i.e

$$y^T y = \lambda x^T x$$

où  $y = \widetilde{W}_n x$

Donc,  $\lambda \geq 0$

Par conséquent, la matrice  $B_N$  est positive mais pas forcément définie.

Or, la matrice  $B_N + \epsilon I$  est définie positive pour tout  $\epsilon > 0$ . On remplace donc, dans les tests numériques, la matrice  $B_N$  par la matrice  $B_N + \epsilon I$  pour un petit  $\epsilon > 0$ .

## A.4 Auto-adaptation de la paramétrisation

Cette approche a été présentée précédemment, elle consiste à régulariser le polygone de contrôle, en minimisant un critère de variation totale, afin d'entamer une nouvelle phase d'optimisation avec une paramétrisation régulière. L'adaptation de la paramétrisation permet de réduire la raideur du système mais soulève la question de la continuité du gradient et du hessien après chaque adaptation de la paramétrisation.

Soit  $(X^0, Y^0)$  le polygone de contrôle, associé à la courbe de Bézier  $\Gamma_0$ , obtenu après une phase d'optimisation, et  $(X^1, Y^1)$  est le polygone de contrôle, associé à la courbe de Bézier  $\Gamma_1$ , qui résulte d'une procédure d'adaptation à partir du polygone  $(X^0, Y^0)$ . Alors,

$$j_n^{X^0}(Y^0) = j_n^{X^1}(Y^1) \quad (\text{A.17})$$

On rappelle ici la contrainte de meilleure approximation utilisée dans la procédure d'adaptation. L'écart quadratique entre l'arc de courbe de Bézier  $\Gamma$ , associé au polygone de contrôle  $(X, Y)$  et l'arc cible  $\Gamma_1$ , qui est aussi une courbe de Bézier, s'exprime comme suit :

$$J = \int_0^1 \frac{1}{2} [B_n(t)^T Y - \bar{y}_T(t, X)]^2 B_{n-1}(t)^T \Delta X dt \quad (\text{A.18})$$

avec :

$$\bar{y}_T(t, X) = y^0(\tau) = B_n(\tau)^T Y^0 \quad (\text{A.19})$$

où la valeur du paramètre  $\tau = \tau(t, X)$  est liée au changement de support  $X^0 \rightarrow X$  par la condition :

$$B_n(\tau)^T X^0 = B_n(t)^T X \quad (\text{A.20})$$

dans laquelle on suppose, pour garantir l'unicité, que la suite  $\{x_k^0\}$  ( $k = 0, 1, \dots, n$ ), des composantes du vecteur  $X^0$ , est monotone croissante.

On souhaite imposer la contrainte suivante : à  $X$  fixé,  $Y$  minimise  $J$ , ce qui équivaut à la condition suivante :

$$\frac{\partial J}{\partial Y} = 0 \quad (\text{A.21})$$

Or,

$$\left(\frac{\partial J}{\partial Y}\right)^T = \int_0^1 B_n(t) \left[ B_n(t)^T Y - \bar{y}_T(t, X) \right] nB_{n-1}(t)^T \Delta X dt = A(X) Y - b(X) \quad (\text{A.22})$$

où la matrice  $A(X)$  est de dimension  $(n+1) \times (n+1)$  et admet la définition suivante :

$$A(X) = \int_0^1 B_n(t) B_n(t)^T nB_{n-1}(t)^T \Delta X dt \quad (\text{A.23})$$

et le vecteur

$$b(X) = \int_0^1 B_n(t) \bar{y}_T(t, X) nB_{n-1}(t)^T \Delta X dt \quad (\text{A.24})$$

dont la nonlinéarité en  $X$  provient de la dépendance de  $X$  du facteur  $\bar{y}_T(t, X)$ . On note que la matrice  $A(X)$  est symétrique et dépend linéairement de  $X$  :

$$A(X) = \mathcal{A} \otimes X \quad (\text{A.25})$$

où  $\mathcal{A} = A'(X)$  est un tenseur d'ordre 3, indépendant de  $X$ , et  $\otimes$  correspond à un certain produit contracté.

Si on dérive l'équation (A.17) par rapport à  $Y^0$ , on obtient,

$$\frac{dj_n^{X^0}(Y^0)}{dY^0} = \frac{dj_n^{X^1}(Y^1)}{dY^1} \frac{dY^1}{dY^0} \quad (\text{A.26})$$

et

$$\frac{d^2 j_n^{X^0}(Y^0)}{d^2 Y^0} = \left(\frac{dY^1}{dY^0}\right)^T \frac{d^2 j_n^{X^1}(Y^1)}{d^2 Y^1} \left(\frac{dY^1}{dY^0}\right) + \frac{dj_n^{X^1}(Y^1)}{dY^1} \frac{d^2 Y^1}{d^2 Y^0} \quad (\text{A.27})$$

Le terme  $\frac{\partial Y^1}{\partial Y^0}$  s'obtient en dérivant par rapport à  $Y^0$  la contrainte due à l'approximation moindre carrée :

$$A(X^1) Y^1 = b(X^1) \quad (\text{A.28})$$

Donc :

$$A(X^1) \frac{dY^1}{dY^0} = \frac{db(X^1)}{dY^0} \quad (\text{A.29})$$

Avec,

$$\frac{db(X^1)}{dY^0} = \int_0^1 B_n(t) B_n(\tau)^T nB_{n-1}(t)^T \Delta X^1 dt \quad (\text{A.30})$$

Le second terme  $\frac{d^2 Y^1}{d^2 Y^0}$  est tel que,

$$A(X^1) \frac{d^2 Y^1}{d^2 Y^0} = 0 \quad (\text{A.31})$$

#### Proposition A.4.1

Soit  $X \in \mathcal{R}^n$  telle que la suite de ses composantes  $\{X_k\}$  ( $k = 0, \dots, n$ ) est monotone croissante

Alors les matrices  $\frac{db(X^1)}{dY^0}$  et  $A(X)$  sont définies positives.

En effet,  
 Soit  $Y \in \mathcal{R}^n$ ,  $Y \neq 0$   
 Alors,

$$Y^T \frac{db(X^1)}{dY^0} Y = \int_0^1 Y^T B_n(t) B_n(\tau)^T Y n B_{n-1}(t)^T \Delta X dt \quad (\text{A.32})$$

Posons  $Z = B_n(t) B_n(\tau)^T$ .

Puisque la suite  $\{X_k\}$  ( $k = 0, \dots, n$ ) est monotone croissante, alors  $\Delta X > 0$ .

Donc, pour que la matrice  $Y^T A(X) Y$  soit strictement positive, il suffit que  $Z$  le soit également.

On a,

$$\begin{aligned} Y^T B_n(t) B_n(\tau)^T Y = 0 &\implies \left( \sum_{i=0}^n B_n^i(t) Y_i \right) \left( \sum_{j=0}^n B_n^j(\tau) Y_j \right) = 0 \\ &\implies \sum_{i=0}^n B_n^i(t) Y_i = 0 \text{ ou } \sum_{j=0}^n B_n^j(\tau) Y_j = 0 \\ &\implies Y_k = 0 \text{ (} k = 0, \dots, n \text{)} \end{aligned} \quad (\text{A.33})$$

car  $\{B_n^k\}$  ( $k = 0, \dots, n$ ) forme une base de  $\mathcal{R}^n$ .

Ceci contredit l'hypothèse initiale, d'où le résultat.

Pour montrer que  $A(X)$  est définie positive, il suffit de prendre  $\tau = t$  dans le raisonnement précédent.

On déduit que  $A(X^1)$  et  $\frac{db(X^1)}{dY^0}$  sont inversibles, puisqu'elles sont symétriques et le vecteur  $X^1$  satisfait les conditions de la proposition précédente.

Donc, d'après (A.29),  $\frac{dY^1}{dY^0}$  est inversible, et d'après (A.31) on déduit que,

$$\frac{d^2 Y^1}{d^2 Y^0} = 0$$

Il en résulte que,

$$\frac{d^2 j_n^{X^0}(Y^0)}{d^2 Y^0} = \left( \frac{dY^1}{dY^0} \right)^T \frac{d^2 j_n^{X^1}(Y^1)}{d^2 Y^1} \left( \frac{dY^1}{dY^0} \right) \quad (\text{A.34})$$

ce qui est équivalent à,

$$\left( \frac{d^2 j_n^{X^1}(Y^1)}{d^2 Y^1} \right)^{-1} = \left( \frac{dY^1}{dY^0} \right) \left( \frac{d^2 j_n^{X^0}(Y^0)}{d^2 Y^0} \right)^{-1} \left( \frac{dY^0}{dY^1} \right)^T \quad (\text{A.35})$$

Les équations (A.34) et (A.35) définissent respectivement les relations de continuité du hessien et de son inverse dans une optimisation adaptative.

## A.5 Algorithme d'optimisation

On utilise l'algorithme UCMINF d'optimisation non linéaire sans contrainte [?]. Cet algorithme est de type quasi-Newton avec une approximation du hessien inverse par la méthode BFGS (Broyden-Fletcher-Goldfarb-Shanno). Le principe de l'algorithme est le suivant :

- Initialisation :  $x_0 \in \mathbb{R}^n$ ,  $\epsilon > 0$ ,  $B_0 = H_0^{-1} := I$  et  $k := 0$
- On calcul la direction de descente  $d_k$

$$d_k = -B_k g_k, \quad g_k := \nabla f(x_k)$$

- On approche itéré suivant par

$$x_{k+1} := x_k + d_k$$

- si  $\|g_{k+1}\| < \epsilon$ , STOP ; autrement :
- On adapte le hessien inverse approché par

$$B_{k+1} := B_k - \frac{d_k (y_k)^T B_k + B_k y_k (d_k)^T}{(y_k)^T d_k} + \left( \frac{1 + (y_k)^T B_k y_k}{(y_k)^T d_k} \right) \frac{d_k (d_k)^T}{(d_k)^T y_k}, \quad (\text{A.36})$$

où  $y_k := g_{k+1} - g_k$

Dans la littérature, il y a plusieurs formules qui calculent  $B_{k+1}$  à partir de  $B_k$  ; la formule de la mise à jour de type BFGS (A.36) est la plus utilisée.

L'algorithme ci-dessus permet d'éviter le calcul du hessien, qui est coûteux surtout pour des applications complexes, et le vecteur  $d_k$  est toujours une direction descente puisque le hessien est par construction défini positif, ce qui n'est pas le cas pour une méthode de Newton classique.

## A.6 Expérimentations numériques

On utilise le problème modèle, décrit au troisième chapitre, pour tester notre approche de continuité des transferts du hessien quand on utilise la méthode Quasi-Newton de type BFGS. Une telle méthode initialise le hessien inverse à l'identité ; ceci a un impact sur la précision des transferts en optimisation multiniveau et en auto-adaptation de paramétrisation.

Dans un premier temps, on compare trois types d'expériences qui ont le même degré de paramétrisation  $N = 12$  : une optimisation jusqu'à convergence et une optimisation avec et sans transfert du hessien après 20 itérations. On constate dans la figure (A.3)(en haut), la perte en convergence itérative quand on réinitialise le hessien par l'identité. On montre dans la figure (A.3)(en bas) les formes obtenues qui sont superposées.

On teste maintenant un algorithme d'enrichissement progressif de paramétrisation à deux niveaux :  $n = 4$  et  $N = 12$ . On utilise l'opérateur d'élévation du degré pour construire deux paramétrisations rigoureusement emboîtées. On constate dans la figure (A.4) qu'un bon transfert du hessien permet de gagner un facteur de deux en convergence itérative par rapport à l'algorithme basique. Par contre, on perd complètement le potentiel de notre algorithme multiniveau quand on initialise le hessien par l'identité par passage du niveau grossier ( $n = 4$ ) au niveau fin ( $N = 12$ ).

Quant à la figure (A.5), on montre les résultats de l'auto-adaptation de la paramétrisation en comparaison avec la méthode basique pour  $n = 8$ . Cette expérience confirme l'efficacité de la procédure d'adaptation de paramétrisation pour résoudre le problème de la raideur



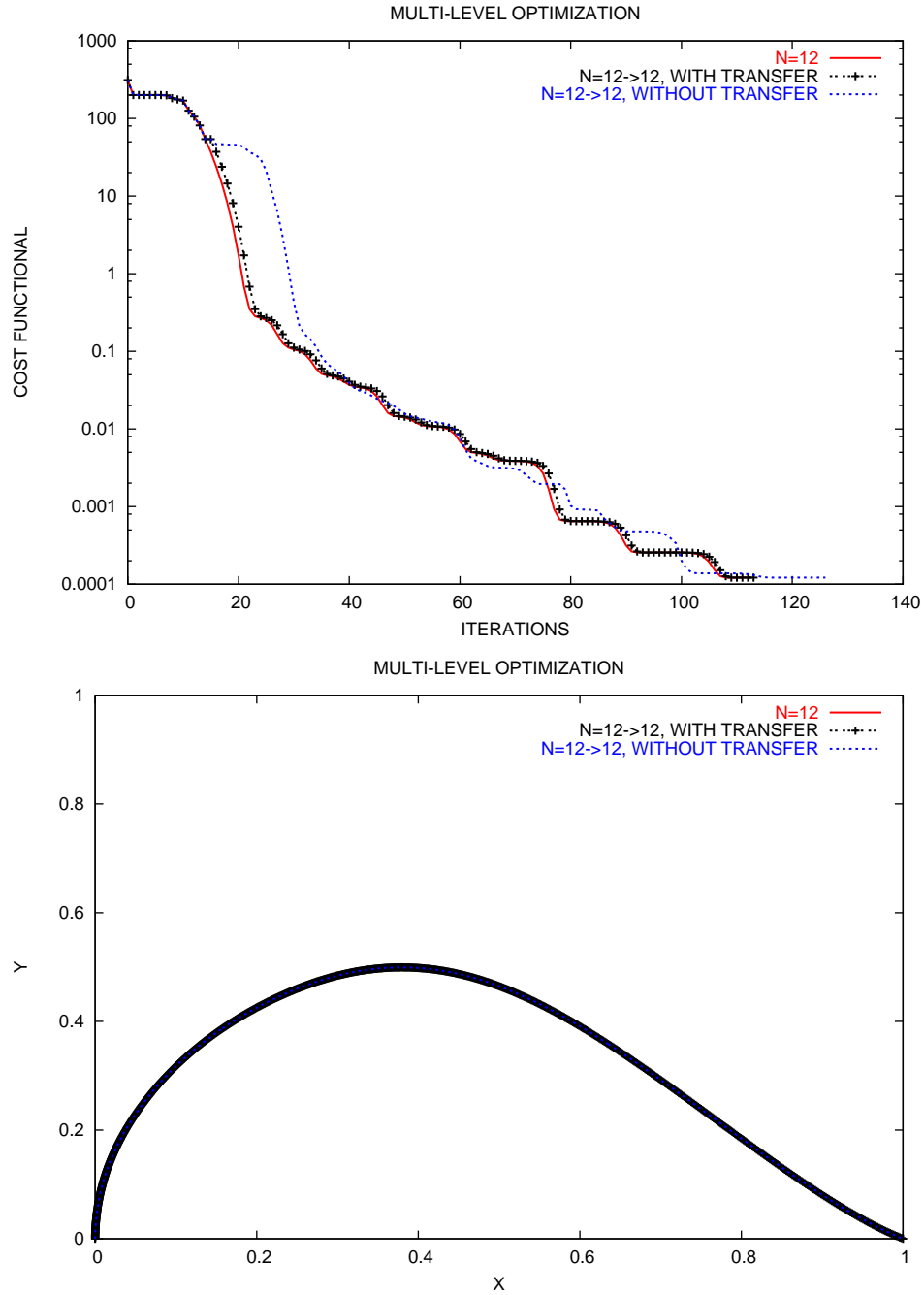


FIGURE A.3 – Algorithme basique pour  $N = 12$  : (1) à convergence complète, (2) avec et sans transfert du hessien

numérique du système. En particulier, la bonne initialisation du hessien après la phase d'adaptation permet d'accélérer au mieux la convergence.

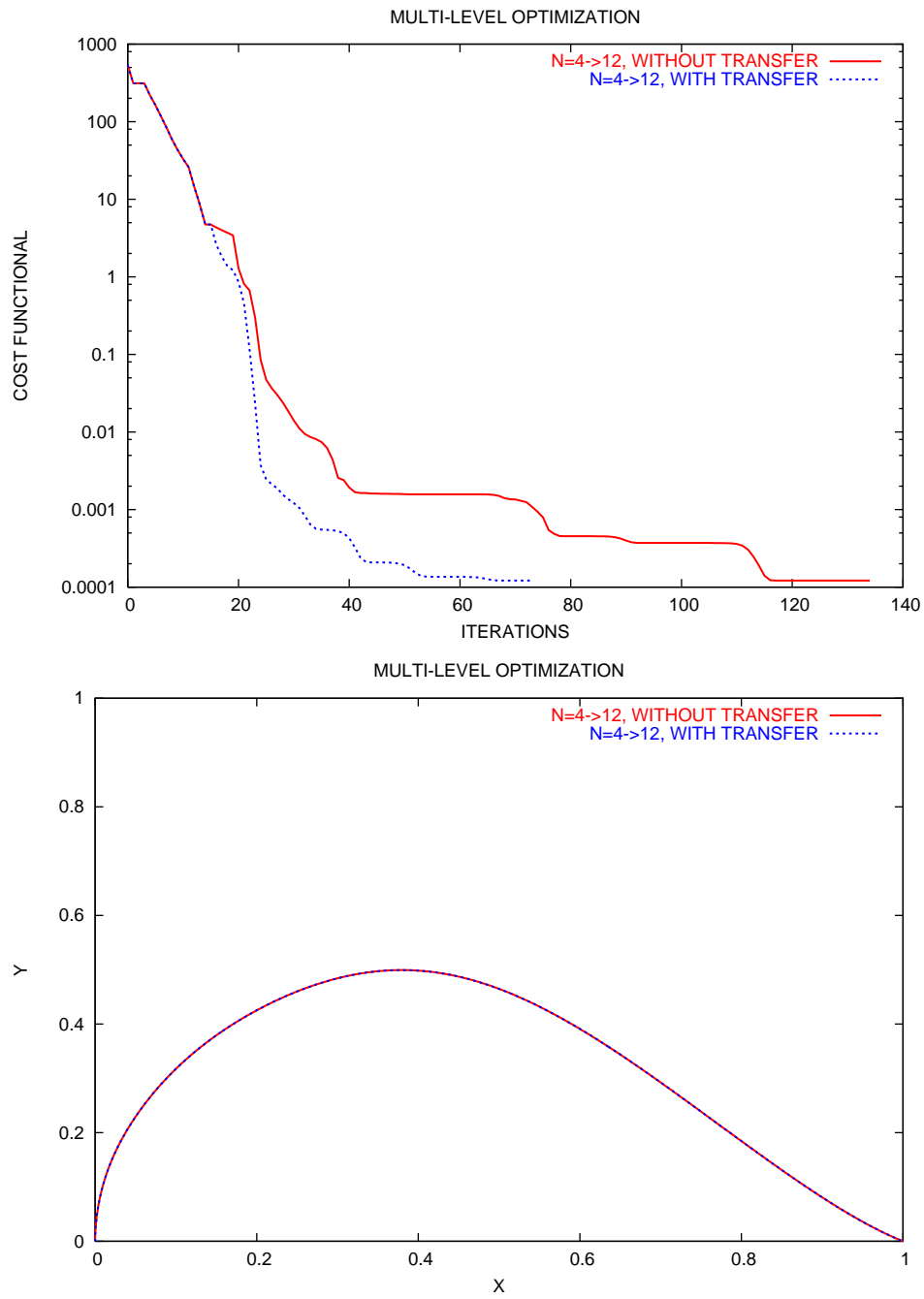


FIGURE A.4 – Enrichissement progressif de paramétrisation avec et sans transfert du hessien( $n = 4$  et  $N = 12$ )

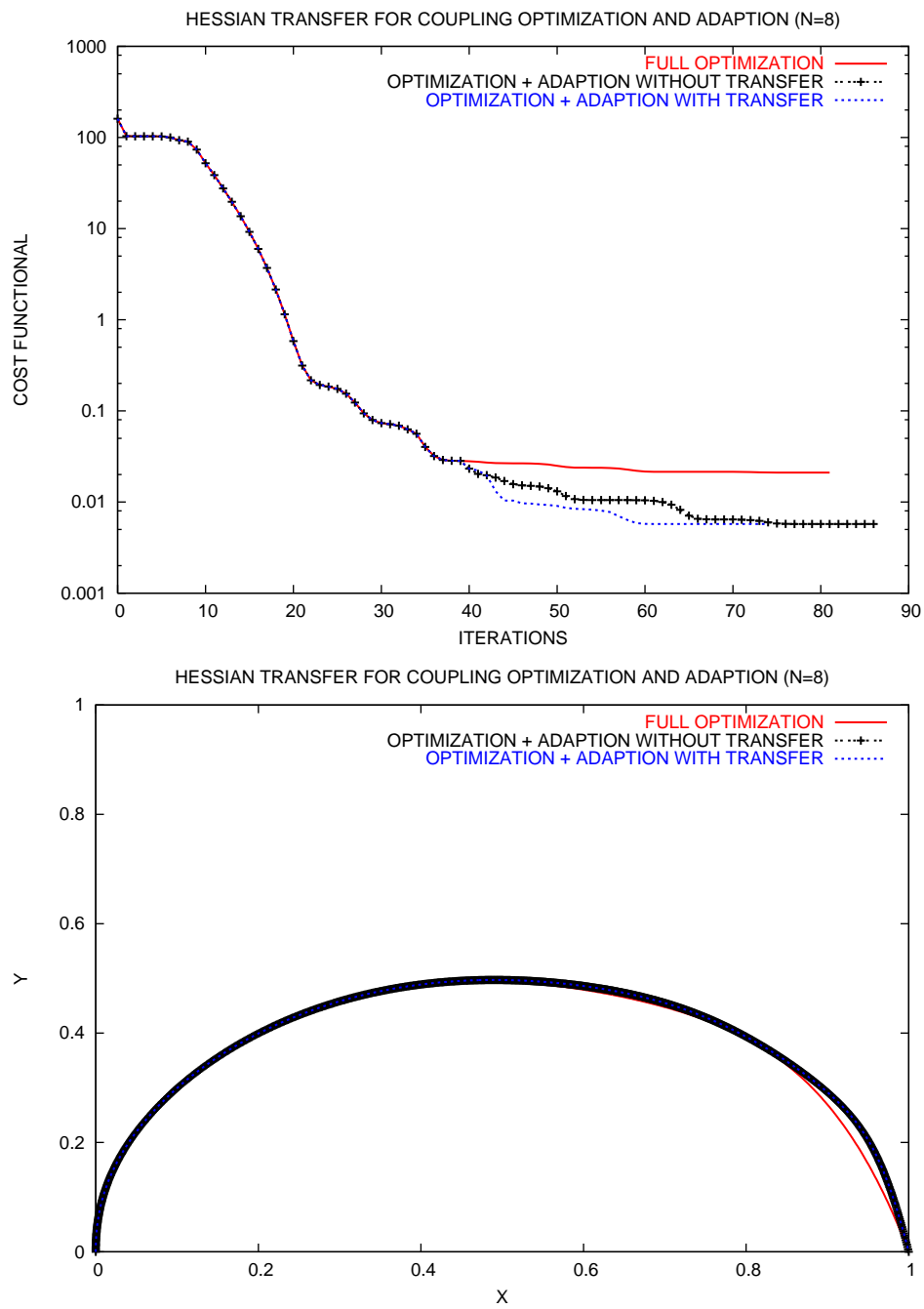


FIGURE A.5 – Auto-adaptation de paramétrisation avec et sans transfert du hessien pour  $n = 8$

## A.7 Conclusion

On a étendu dans cette étude le spectre des méthodes d'optimisation qui peuvent être utilisées dans un algorithme d'optimisation multiniveau ou en auto-adaptation de paramétrisation. Il s'agit de la méthode de Quasi-Newton avec une mise à jour du hessien inverse par BFGS. Cette dernière a plusieurs mérites rappelés précédemment mais soulève la question de continuité des transferts puisque le hessien inverse est initialisé à la matrice identité pour chaque appel à la méthode. Les résultats numériques de la section précédente montrent combien cette question de continuité des transferts est importante pour gagner en convergence itérative et en précision.



## Annexe B

# Paramétrisations non emboîtées

À l'tant donnée  $(x_k^{(n_1)}, y_k^{(n_1)})_{0 \leq k \leq n_1}$  les points de contrôle de la paramétrisation de départ de degré  $n_1$ , et  $(x_k^{(n_2)}, y_k^{(n_2)})_{0 \leq k \leq n_2}$  le support de la paramétrisation d'arrivée de degré  $n_2$  tel que  $x^{(n_2)'}(t) > 0$ . On cherche à trouver le vecteur  $(y_k^{(n_2)})_{0 \leq k \leq n_2}$  qui vérifie les équations normales suivantes :

$$\int_0^1 y^{(n_2)}(x) \varphi^m(x) dx = \int_0^1 y^{(n_1)}(x) \varphi^m(x) dx \quad \text{où } \varphi^m \in \Phi \quad (\text{B.1})$$

avec  $\Phi$  est un ensemble dense dans l'ensemble des courbes de Bézier de degrés max  $\{n_1, n_2\}$ .

On rappelle que la courbe de Bézier associée au polygone de contrôle  $(x_k^{(n_j)}, y_k^{(n_j)})_{0 \leq k \leq n_j}$  ( $j = 1, 2$ ) est de la forme :

$$\begin{cases} x^{(n_j)}(t) = \sum_{k=0}^{n_j} B_{n_j}^k(t) x_k^{(n_j)} \\ y^{(n_j)}(t) = \sum_{k=0}^{n_j} B_{n_j}^k(t) y_k^{(n_j)} \end{cases} \quad (\text{B.2})$$

avec  $B_{n_j}^k$  est le polynôme de Bernstein.

Si on effectue le changement de variable  $x = x^{(n_1)}(t)$  (resp.  $x = x^{(n_2)}(t)$ ) dans le membre de droite de (B.1) (resp. dans le membre de gauche de (B.1)) avec  $t \in [0, 1]$ , on obtient :

$$\int_0^1 y^{(n_2)}(t) \varphi^m(x^{(n_2)}(t)) x^{(n_2)'}(t) dt = \int_0^1 y^{(n_1)}(t) \varphi^m(x^{(n_1)}(t)) x^{(n_1)'}(t) dt \quad (\text{B.3})$$

Posons  $I_{1m} = \int_0^1 y^{(n_2)}(t) \varphi^m(x^{(n_2)}(t)) x^{(n_2)'}(t) dt$  et  $I_{2m} = \int_0^1 y^{(n_1)}(t) \varphi^m(x^{(n_1)}(t)) x^{(n_1)'}(t) dt$

On choisit les fonctions  $\varphi^m$  telles que :  $\varphi^m(x^{(n_2)}(t)) = B_{n_2}^m(t)$ ,  $m \in \{0, \dots, n_2\}$ . Alors pour tout  $x$  tel que  $x^{(n_2)}(\tau) = x$ ,  $\tau \in [0, 1]$ ,  $\varphi^m(x) = B_{n_2}^m(\tau)$ . En particulier, pour  $x = x^{(n_1)}(t)$ ,  $\varphi(x^{(n_1)}(t)) = B_{n_2}^m(\tau)$  où  $x^{(n_2)}(\tau) = x^{(n_1)}(t)$ . Il en résulte que

$$I_{2m} = \int_0^1 y^{(n_1)}(t) B_{n_2}^m(\tau) x^{(n_1)'}(t) dt$$

Pour tout  $t_i \in [0, 1]$ , l'évaluation de l'intégrande de  $I_{2m}$  au point  $t_i$  passe par la résolution de l'équation  $x^{(n_2)}(\tau(t_i)) = x^{(n_1)}(t_i)$ , que l'on résout par la méthode de Newton, pour déduire  $\tau(t_i)$ . On note que l'équation  $x^{(n_2)}(\tau(t_i)) = x^{(n_1)}(t_i)$  admet une solution unique vu l'hypothèse

$x^{(n_2)'}(t) > 0 \forall t \in [0, 1]$ . Numériquement, le calcul de l'intégrale  $I_{2m}$  se fait par la méthode de Gauss-Legendre en faisant appel à une sous-routine de NAG (Numerical Algorithms Group).

Quant au membre de gauche  $I_{1m}$  on a ( $\forall m \in \{0, \dots, n_2\}$ ) :

$$\begin{aligned}
 I_{1m} &= \int_0^1 y^{(n_2)}(t) \varphi^m(x^{(n_2)}(t)) x^{(n_2)'}(t) dt \\
 &= \int_0^1 y^{(n_2)}(t) B_{n_2}^m(t) x^{(n_2)'}(t) dt \\
 &= \int_0^1 \left( \sum_{k=0}^{n_2} B_{n_2}^k(t) y_k^{(n_2)} \right) \left( n_2 \sum_{l=0}^{n_2-1} B_{n_2-1}^l(t) (x_{l+1}^{(n_2)} - x_l^{(n_2)}) \right) B_{n_2}^m(t) dt \\
 &= n_2 \sum_{k=0}^{n_2} \sum_{l=0}^{n_2-1} \left( \int_0^1 B_{n_2}^k(t) B_{n_2-1}^l(t) B_{n_2}^m(t) dt \right) (x_{l+1}^{(n_2)} - x_l^{(n_2)}) y_k^{(n_2)}
 \end{aligned}$$

Posons  $\beta_n^k = \int_0^1 B_n^k(t) dt$

On a :

$$\begin{aligned}
 \beta_n^k &= \int_0^1 C_n^k t^k (1-t)^{n-k} dt \\
 &= C_n^k \int_0^1 t^k (1-t)^{n-k} dt \\
 &= C_n^k \underbrace{\left[ \frac{t^{k+1}}{k+1} (1-t)^{n-k} \right]_0^1}_{=0} + C_n^k \int_0^1 \frac{t^{k+1}}{k+1} (n-k)(1-t)^{n-k-1} dt \quad \text{par intégration par parties} \\
 &= C_n^k \int_0^1 \frac{t^{k+1}}{k+1} (n-k)(1-t)^{n-k-1} dt \\
 &= \frac{C_n^k}{k+1} (n-k) \int_0^1 t^{k+1} (1-t)^{n-k-1} dt \\
 &= \frac{C_n^k}{k+1} (n-k) \frac{1}{C_n^{k+1}} \beta_n^{k+1} \\
 &= \frac{n!}{k!(n-k)!} \frac{n-k}{k+1} \frac{(k+1)!(n-k-1)!}{n!} \beta_n^{k+1} \\
 &= \frac{k!(n-k-1)!}{k!(n-k-1)!} \beta_n^{k+1} \\
 &= \beta_n^{k+1}
 \end{aligned}$$

Donc,

$$\beta_n^k = \beta_n^{k+1} = \dots = \beta_n^n = \int_0^1 t^n dt = \frac{1}{n+1}$$

Posons  $\gamma_{n_2}^{k,l,m} = n_2 \left( \int_0^1 B_{n_2}^k(t) B_{n_2-1}^l(t) B_{n_2}^m(t) dt \right)$  On a :

$$\begin{aligned} \gamma_{n_2}^{k,l,m} &= n_2 C_{n_2}^k C_{n_2-1}^l C_{n_2}^m \int_0^1 t^{k+l+m} (1-t)^{3n_2-1-k-l-m} dt \\ &= n_2 C_{n_2}^k C_{n_2-1}^l C_{n_2}^m \frac{\beta_{3n_2-1}^{k+l+m}}{C_{3n_2-1}^{k+l+m}} \\ &= n_2 \frac{C_{n_2}^k C_{n_2-1}^l C_{n_2}^m}{3n_2 C_{3n_2-1}^{k+l+m}} \\ &= \frac{C_{n_2}^k C_{n_2-1}^l C_{n_2}^m}{C_{3n_2-1}^{k+l+m}} \end{aligned}$$

Par conséquent, la condition d'optimalité (B.3) s'écrit donc :

$$I_{1m} = \sum_{k=0}^{n_2} \sum_{l=0}^{n_2-1} \gamma_{n_2}^{k,l,m} (x_{l+1}^{(n_2)} - x_l^{(n_2)}) y_k^{(n_2)} = \int_0^1 y^{(n_1)}(t) \varphi^m(x^{(n_1)}(t)) x^{(n_1)'}(t) dt \quad \text{pour } m \in \{0, \dots, n_2\}$$

Soit  $A$  la matrice de dimension  $(n_2 + 1) \times (n_2 + 1)$  telle que :

$$A_{mk} = \sum_{l=0}^{n_2-1} \gamma_{n_2}^{k,l,m} (x_{l+1}^{(n_2)} - x_l^{(n_2)})$$

Alors l'équation (B.3) est équivalente au système linéaire suivant :

$$AY^{(n_2)} = b \tag{B.4}$$

$$\text{où } Y^{(n_2)} = \begin{pmatrix} y_0^{(n_2)} \\ y_1^{(n_2)} \\ \vdots \\ y_{n_2}^{(n_2)} \end{pmatrix} \text{ et } b_i = \int_0^1 y^{(n_1)}(t) B_{n_2}^m(\tau) x^{(n_1)'}(t) dt$$

On note que la matrice  $A$  est symétrique, car  $\gamma_{n_2}^{k,l,m}$  l'est aussi, mais n'est pas forcément définie positive.

**Proposition B.0.1**

*Le système  $AY^{(n_2)} = b$  admet une solution unique*

En effet,

Il suffit de montrer l'injectivité de la matrice  $A$  puisqu'on se place dans un espace de dimension finie.

Supposons que

$$\sum_{k=0}^{n_2} A_{mk} y_k = 0 \quad \forall m, k \in \{0, \dots, n_2\}$$



ce qui implique

$$\int_0^1 y^{(n_2)}(t) B_{n_2}^m(t) x^{(n_2)'}(t) dt = 0 \quad , \quad \forall m \in \{0, \dots, n_2\}$$

alors

$$\begin{aligned} \int_0^1 (y^{(n_2)}(t))^2 x^{(n_2)'}(t) dt &= \int_0^1 y^{(n_2)}(t) y^{(n_2)}(t) x^{(n_2)'}(t) dt \\ &= \int_0^1 y^{(n_2)}(t) \sum_{k=0}^{n_2} B_{n_2}^k y_k^{(n_2)} x^{(n_2)'}(t) dt \\ &= \sum_{k=0}^{n_2} y_k^{(n_2)} \underbrace{\int_0^1 y^{(n_2)}(t) B_{n_2}^k x^{(n_2)'}(t) dt}_{=0} \end{aligned}$$

Donc  $\int_0^1 (y^{(n_2)}(t))^2 x^{(n_2)'}(t) dt = 0 \Rightarrow y^{(n_2)}(t) = 0$  car  $x^{(n_2)'}(t) > 0$ .

Ce qui implique que  $y_k^{(n_2)} = 0$  pour tout  $k \in \{0, \dots, n_2\}$ .

**Remarque :** Si  $(x_k^{(n_2)})_{k=0, \dots, n_2}$  est le résultat de  $n_2 - n_1$  élévations du degré à partir de  $(x_k^{(n_1)})_{k=0, \dots, n_1}$  alors  $x^{(n_2)}(t) = x^{(n_1)}(t)$  pour le même  $t$ . Donc, les équations normales s'écrivent :

$$\int_0^1 y^{(n_2)}(t) B_{n_2}^m(t) x'(t) dt = \int_0^1 y^{(n_1)}(t) B_{n_2}^m(t) x'(t) dt \quad \text{pour tout } m \in \{0, \dots, n_2\}$$

Ces équations admettent comme solution particulière  $y^{(n_2)}(t) = y^{(n_1)}(t)$ . Par conséquent, l'opérateur d'interpolation n'est autre que l'opérateur d'élévation du degré. On illustre cet opérateur dans la figure suivante pour  $n_1 = 3$  et  $n_2 = 4$ .

On note qu'on peut reprendre la construction précédente avec l'hypothèse  $n_2 \leq n_1$ , on aboutit donc à un opérateur de réduction de degré. On illustre cet opérateur dans la figure suivante pour  $n_1 = 3$  et  $n_2 = 2$ . L'écart entre les deux courbes obtenues est dû à une perte de précision.

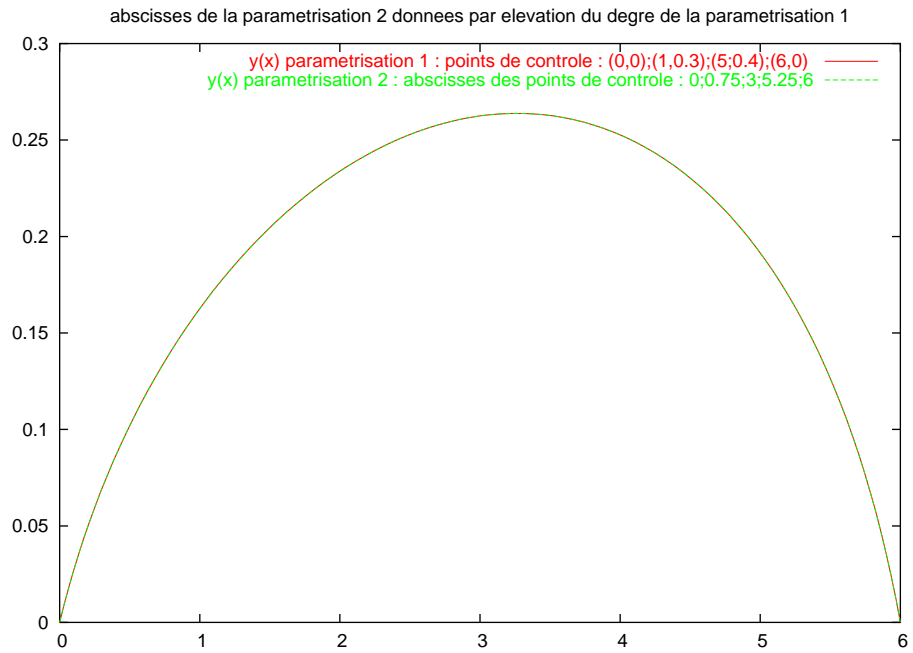


FIGURE B.1 – Illustration de l’opérateur d’interpolation :  $n_1 = 3$  et  $n_2 = 4$

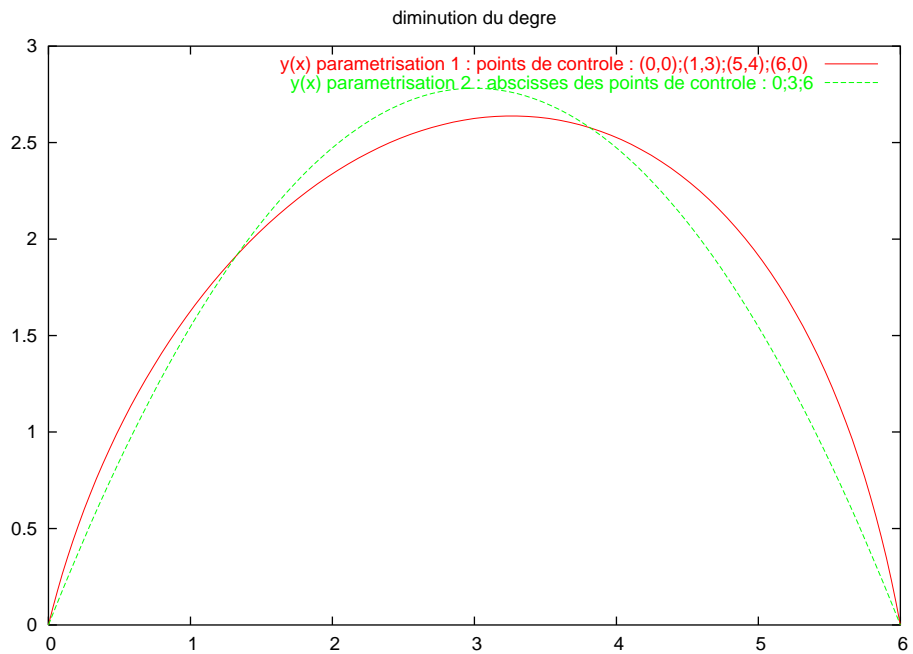


FIGURE B.2 – Illustration de l’opérateur d’interpolation :  $n_1 = 3$  et  $n_2 = 2$



## Annexe C

# Analyse structurale

On utilise le Code Aster, développé par EDF, comme un outil d'analyse. C'est un code général d'analyse par éléments finis qui étudie les comportements mécaniques des structures. On se reportera aux différentes documentations (voir <http://www.code-aster.org>) pour avoir tous les détails sur les principes de fonctionnement et les règles d'utilisation.

On présente ici un test de validation du calcul de structures avec le Code Aster, et les étapes de réalisation d'une analyse structurale sur une géométrie d'aile d'avion.

### C.1 Test de validation

On utilise un test de validation extrait du "Guide de validation des progiciels de calcul de structures" publié par l'ANFOR (Association Française de Normalisation) et qui a été élaboré par la société française des mécaniciens (SFM). Le but est de valider le calcul des déplacements et des contraintes par comparaison avec une solution de référence analytique. Deux types de modélisation éléments finis sont utilisés : *DKT* avec des quadrangles à 4 noeuds (*QUAD4*) et *COQUE\_3D* avec des triangles à 7 noeuds (*TRIA7*).

#### C.1.1 Géométrie

La géométrie considérée est un cylindre de rayon moyen  $R = 5.7 \text{ m}$ , d'épaisseur  $e = 0.04 \text{ m}$  et de hauteur  $L = 16 \text{ m}$ , simplement appuyée sur sa base ( $z = 0$ ). Le matériau utilisée est élastique isotrope de module de Young  $E = 2.1E + 11 \text{ Pa}$  et de coefficient de Poisson  $\nu = 0.3$ .

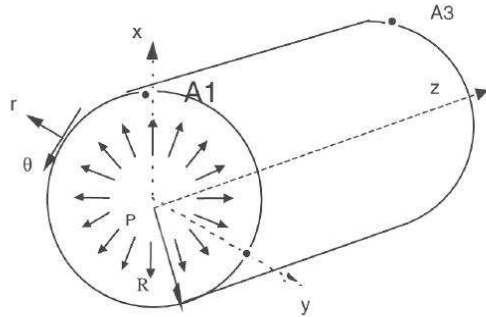


FIGURE C.1 – La géométrie initiale

### C.1.2 Résultats

On soumet le cylindre à une pression hydrostatique interne variant linéairement suivant  $z$  :  $p(z) = P_0 * (1 - z/L)$ , avec  $P_0 = 15000 Pa$  (voir figure C.1). Le champ des déplacements obtenu est donné par la figure [?] en utilisant une modélisation *DKT*.

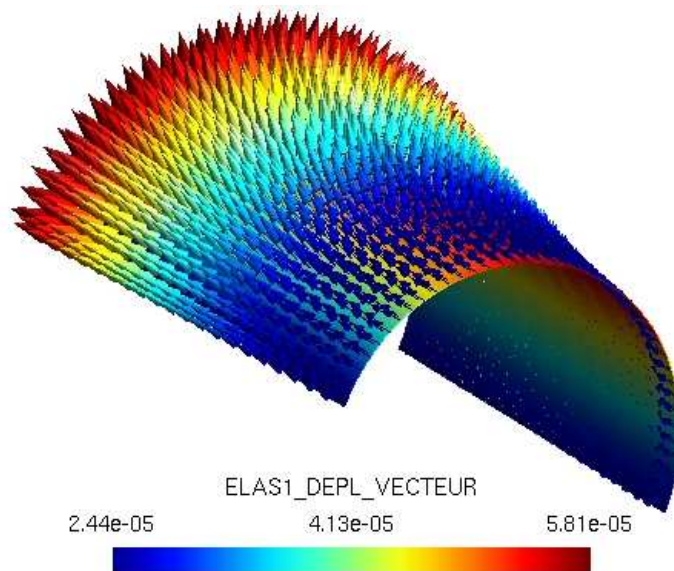


FIGURE C.2 – Champ des déplacements

A partir de la solution analytique donnée par [90]), on déduit les résultats de référence suivants :

- Déplacement radial au noeud A1 : 5.8017857142857E-05
- Déplacement vertical au noeud A3 : 2.4428571428571E-05
- Contraintes circonférentielle au noeud A5 : 2.1375000000000E+06

On compare dans les tables (C.1.2) et (C.1.2) les résultats obtenus par le Code Aster en comparaison avec les résultats de référence en utilisant deux types de modélisations éléments finis : *DKT* avec des éléments *QUAD4* et *COQUE\_3D* avec des éléments *TRIA7*. On

considère seulement la moitié du cylindre pour des raisons de symétrie. On effectue les calculs aux noeuds suivants (voir figure C.1) :  $A1(R, 0, 0)$ ,  $A2(-R, 0, 0)$ ,  $A3(R, 0, L)$ ,  $A4(-R, 0, L)$  et  $A5(0, R, 0)$ .

	Référence	Aster	%
DX(A1)	5.8017857142857E-05	5.7932308604598E-05	-0.147
DX(A2)	5.8017857142857E-05	-5.8056386541030E-05	0.066
DY(A5)	5.8017857142857E-05	5.7994337500786E-05	-0.041
DZ(A3)	-2.4428571428571E-05	-2.4427564238751E-05	-0.004
DZ(A4)	-2.4428571428571E-05	-2.4428086935531E-05	-0.002
SIYY(A1)	2.1375000000000E+06	2.1376776987378E+06	0.008

TABLE C.1 – Table de validation pour une modélisation *COQUE\_3D* avec des éléments *TRIA7*

	Référence	Aster	%
DX(A1)	5.8017857142857E-05	5.7915866202620E-05	-0.176
DX(A2)	5.8017857142857E-05	-5.7915866231395E-05	-0.176
DY(A5)	5.8017857142857E-05	5.7915866217974E-05	-0.176
DZ(A3)	-2.4428571428571E-05	-2.4420200371490E-05	-0.034
DZ(A4)	-2.4428571428571E-05	-2.4420200366422E-05	-0.002
SIYY(A1)	2.1375000000000E+06	2.1371182520661E+06	-0.018

TABLE C.2 – Table de validation pour une modélisation *DKT* avec des éléments *QUAD4*

## C.2 Étapes d'analyse avec le Code Aster

L'étude concerne une géométrie d'aile d'avion donnée initialement par un maillage triangulaire non-structuré. Le but est d'effectuer une analyse structurale avec le Code Aster en imposant un encastrement à l'emplanture de l'aile, et en appliquant un chargement uniforme. Pour cela, on effectue deux étapes :

1. On construit un maillage, sous un format spécifique au Code Aster (GMSH, IDEAS,...), qui contient les coordonnées des noeuds du maillage initial, des mailles sur lesquelles on affecte des propriétés physiques, des éléments finis, des conditions aux limites ou des chargements, et des groupes de noeuds ou de mailles pour faciliter les opérations de modélisations. On montre dans la figure (C.3) le maillage GMSH obtenu.
2. On définit un fichier de commandes, contenant un ensemble de commandes spécifiques au Code Aster, qui permet de lire le maillage, d'affecter les données de la modélisation sur les entités du maillage, de résoudre le problème de mécanique statique pour déduire la réponse structurale (déplacements, tenseur des contraintes,...), et d'éditer les résultats sous un format spécifique. On présente ci-dessous le fichier de commandes utilisé :

✓ Début du fichier.

*DEBUT()*;

✓ On spécifie le format du maillage utilisée (GMSH).

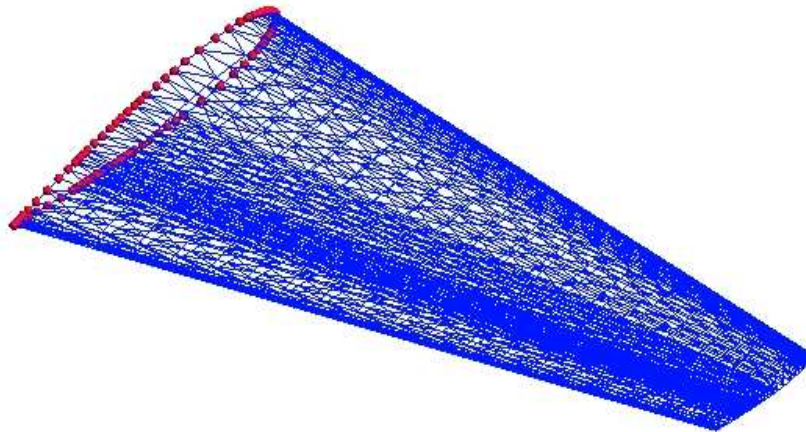


FIGURE C.3 – Maillage initial au format GMSH

*PRE\_GMSH*();

✓ On produit une structure de donnée de type maillage.

*MAIL = LIRE\_MAILLAGE*();

✓ On définit deux groupes de mailles à partir de ceux créés par le maillage IDEAS : GM1 désigne la totalité des mailles, et GM2 désigne les mailles de la zone encastrée.

```
MAIL = DEFI_GROUP(  reuse = MAIL,
                    MAILLAGE = MAIL,
                    CREA_GROUP_MA = (
                        _F(NOM = ' WING', GROUP_MA = ' GM1'),
                        _F(NOM = ' ENCAST', GROUP_MA = ' GM2')));
```

✓ On associe à toutes les mailles le type des éléments finis. On a utilisé deux types de modélisation : *DKT*, qui correspond à des éléments triangulaires à trois noeuds (*TRIA3*), et *COQUE\_3D*, qui correspond à des éléments triangulaires à 7 noeuds (*TRIA7*).

```
MODMECA = AFFE_MODELE(  MAILLAGE = MAIL,
                        AFFE = _F(TOUT = ' OUI',
                                    PHENOMENE = ' MECANIQUE',
                                    MODELISATION = ' DKT/COQUE_3D'));
```

✓ On définit les éléments du modèle comme étant de type coque, le maillage du départ représente la surface moyenne de la coque, et on fournit en plus l'épaisseur et une direction de référence dans le plan tangent.

```
CARA_COQ = AFFE_CARA_ELEM(  MODELE = MODMECA,
                            COQUE = _F(GROUP_MA = (' WING', ' ENCAST'),
                                        EPAIS = 0.005,
                                        ANGL_REP = (45.0, 45.0)));
```

✓ On définit les caractéristiques du matériau : module de Young et coefficient de Poisson.

```
ACIER = DEFI_MATERIAU( ELAS = _F(E = 68948000000.0,  
NU = 0.33));
```

✓ On affecte le matériau sur le maillage (on considère le matériau est le même sur tous le maillage).

```
CHMAT = AFFE_MATERIAU( MAILLAGE = MAIL,  
AFFE = _F(TOUT = ' OUI',  
MATER = ACIER));
```

✓ On définit les conditions aux limites. Ici les éléments du groupe de mailles 'ENCAST' sont à déplacement et à rotation nulle dans toutes les directions.

```
BLOQUAGE = AFFE_CHAR_MECA( MODELE = MODMECA,  
DDL_IMPO = _F(GROUP_MA = ' ENCAST',  
DX = 0.0,DY = 0.0,DZ = 0.0,  
DRX = 0.0,DRY = 0.0,DRZ = 0.0));
```

✓ On applique un chargement de pression uniforme sur toutes les mailles. Dans nos applications, on utilise plutôt un chargement de pression non-uniforme qui résulte d'un calcul aérodynamique.

```
CHG = AFFE_CHAR_MECA( MODELE = MODMECA,  
FORCE_COQUE = _F(GROUP_MA = ' WING',  
PRES = 300000.));
```

✓ On résout un problème de mécanique statique linéaire qui correspond à une coque définie par les modèles précédents. Le champ des déplacements obtenu est contenu dans un concept résultat "RESU1". Pour éviter les problèmes de stockage, On choisit une méthode de résolution des systèmes linéaires de type multi-frontale, qui utilise une représentation creuse de type "Morse", avec une numérotation "MDA" économique à calculer (conseillée pour les gros modèles).

```
RESU1 = MECA_STATIQUE( MODELE = MODMECA,  
CHAM_MATER = CHMAT,  
CARA_ELEM = CARA_COQ,  
EXCIT = (_F(CHARGE = BLOQUAGE),  
_F(CHARGE = CHG)),  
SOLVEUR = _F(METHODE = ' MULT_FRONT',  
RENUM = ' MDA'));
```



✓ A partir du champ des déplacements, on calcule les contraintes indiquées par le champ "OPTION". Ici, on calcule les contraintes aux noeuds de chaque élément.

```
RESU1 = CALC_ELEM( reuse = RESU1,  
MODELE = MODMECA,  
CHAM_MATER = CHMAT,  
CARA_ELEM = CARA_COQ,  
RESULTAT = RESU1,  
OPTION = ' SIGM_ELNO_DEPL', );
```

✓ On imprime les résultats (le champ des déplacements et des contraintes) dans un fichier sous un format spécifique au Code Aster (GMSH par exemple).

```
IMPR_RESU( MODELE = MODMECA,  
FORMAT = ' GMSH',  
UNITE = 55,  
RESU = _F(RESULTAT = RESU1,  
NOM_CHAM = ('DEPL', ' SIGM_ELNO_DEPL')));
```

✓ Fin du fichier  
FIN();

On montre dans les figure (C.4) et (C.5) la réponse structurale (champ des déplacements et des contraintes) obtenue par application respectivement d'un chargement de pression uniforme et d'un champ de force dans la direction verticale.

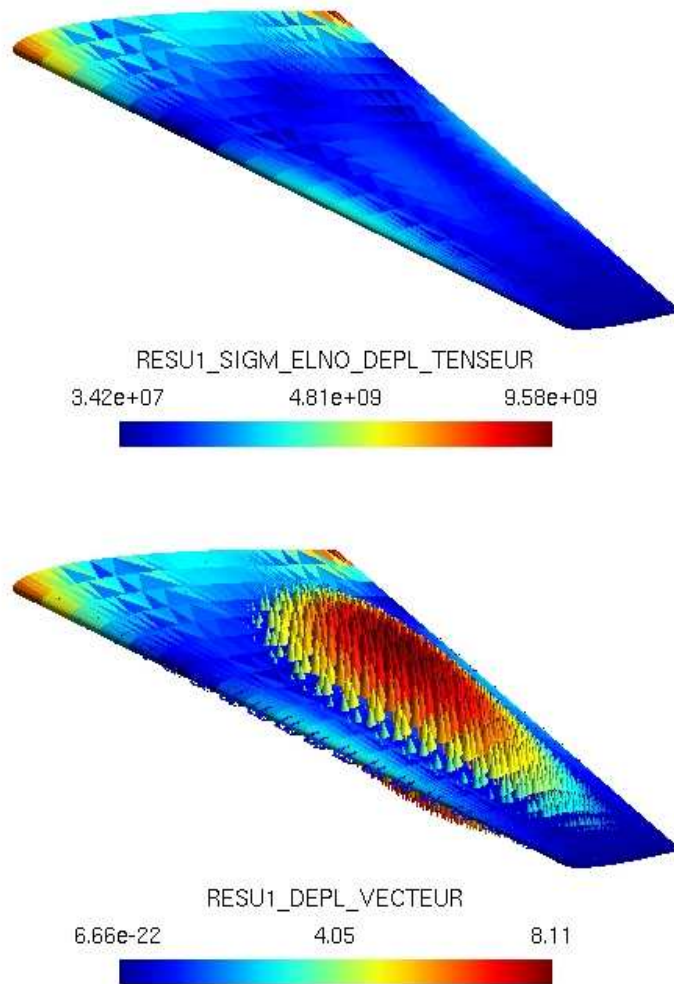


FIGURE C.4 – Champ des contraintes (bas) et champ des déplacements (haut) obtenus par un chargement de pression uniforme

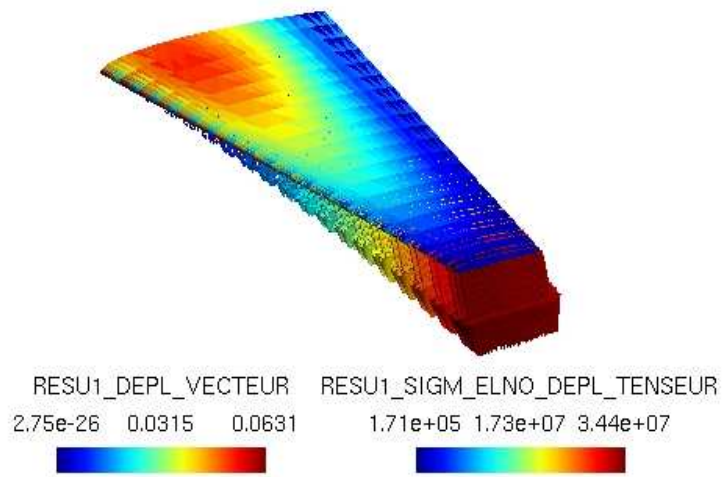


FIGURE C.5 – Champ des déplacements et des contraintes obtenus par un chargement de force uniforme dans la direction verticale

## Annexe D

# Fonctions radiale de base et Krigeage

L'utilisation des méthodes d'optimisation robuste, comme par exemple les algorithmes génétiques et les algorithmes par essaim de particules (PSO), nécessitent un nombre important d'évaluations de la fonction coût où chaque évaluation est coûteuse. Ces évaluations peuvent être approchées en utilisant des modèles d'approximation ou métamodèles. On présente ici brièvement les deux modèles d'approximation utilisés [22] : la méthode des fonctions radiale de base et le modèle de krigeage.

### D.1 Problème d'interpolation

On considère un ensemble de  $N$  points distincts,

$$X_N = \{x_1, x_2, \dots, x_N\} \quad (\text{D.1})$$

et les valeurs de la fonction coût  $f$  en ces points

$$F_N = [f_1, f_2, \dots, f_N]^T \quad (\text{D.2})$$

avec  $f_i = f(x_i)$ ,  $i = 1, \dots, N$

On cherche à approximer la valeur  $f_{N+1}$  de  $f$  en un nouveau point  $x_{N+1}$ .

### D.2 Méthode des fonctions radiale de base

Les fonctions radiale de base (RBFs) ont été utilisées depuis les années 70 pour différents types de problèmes d'interpolation [20]. Cette méthode a été utilisée aussi pour approximer les solutions des équations aux dérivées partielles [60][67] ; ceci est dû principalement à trois raisons :

- La méthode ne fait appel à aucun maillage, ce qui signifie qu'elle est flexible à des géométries complexes.
- La méthode est efficace pour des problèmes de dimension grande.
- Pour des fonctions régulières, les approximations ont une convergence spectrale.

Une fonction radiale de base (RBF) est une fonction à valeur dans  $\mathbf{R}$  où sa valeur en un point dépend de la distance de ce point à un autre point, telle que  $\Phi(x) = \phi(\|x - a\|)$  avec  $a$  se nomme le centre de RBF. On présente ci-dessous quelques exemples de fonctions radiale de base :

– Gaussienne :

$$\phi(r) = \exp(-\beta r^2), \beta > 0 \quad (\text{D.3})$$

– Multiquadratique :

$$\phi(r) = \sqrt{r^2 + \beta^2}, \beta > 0 \quad (\text{D.4})$$

– Spline plaque mince :

$$\phi(r) = r^2 \log(r) \quad (\text{D.5})$$

Les fonctions radiale de base sont utilisées pour construire une fonction d'approximation  $\bar{f}$  de la forme :

$$\bar{f}(x) = \sum_{i=1}^N \omega_i \Phi(x - x_i) \quad (\text{D.6})$$

où  $\Phi(x) = \phi(\|x\|)$  est une fonction radiale de base (RBF), les positions  $x_i, i = 1, \dots, N$  sont les centres de RBF, et les coefficients  $\omega = [\omega_1, \omega_2, \dots, \omega_N]^T$  sont obtenues à partir des conditions d'interpolation :

$$\bar{f}(x_i) = f_i, i = 1, \dots, N \quad (\text{D.7})$$

Donc,  $\omega$  est solution du système linéaire suivant,

$$A\omega = F_N \quad (\text{D.8})$$

avec  $A$  est une matrice de dimension  $(N + 1) \times (N + 1)$  qui a comme éléments  $A_{ij} = \Phi(x_i - x_j)$ .

### D.3 Modèle de Krigeage

La méthode connue sous le nom de "Krigeage" est une méthode stochastique d'interpolation spatiale. Le krigeage porte le nom de son précurseur, l'ingénieur minier sud-africain D.G. Krige [62]. C'est cependant le français Matheron qui a formalisé l'approche en donnant le nom "Krigeage" à la méthode [78]. Plusieurs types de krigeage ont été développés dans la littérature : le krigeage simple, le krigeage ordinaire, le krigeage robuste, etc. On présente seulement La méthode de krigeage bayésien [71][72], proposée par [87], qui consiste à prendre les paramètres de la fonction de variation comme des variables aléatoires qui sont associées à des lois a priori.

La densité de probabilité de  $f_{N+1}$  (la valeur de  $f$  au point  $x_{N+1}$ ), étant donné  $X_N$  et  $F_N$ , est comme suit

$$p(f_{N+1}|X_{N+1}, F_N) = \frac{p(F_{N+1}|X_{N+1})}{p(F_N|X_N)} \quad (\text{D.9})$$

Ceci en utilisant la relation suivante :

$$p(A|B) = \frac{p(A, B)}{p(B)} \quad (\text{D.10})$$

Le vecteur  $F_N$  est considéré comme une réalisation d'un processus Gaussien à  $N$  variables avec la densité de probabilité

$$p(F_N|X_N) = \frac{\exp(-\frac{1}{2}F_N^T C_N^T F_N)}{\sqrt{(2\pi)^N \det(C_N)}} \quad (\text{D.11})$$

où  $C_N$  est une matrice de covariance de dimension  $N \times N$ . Les éléments  $C_{ij}$  de la matrice de covariance sont tels que,  $C_{ij} = c(x_i, x_j)$ , avec  $c$  est une fonction de covariance.

Quand on rajoute un nouveau point  $x_{N+1}$ , le vecteur  $F_{N+1}$  est considéré comme une réalisation d'un processus Gaussien à  $N + 1$  variables avec la densité de probabilité

$$p(F_{N+1}|X_{N+1}) = \frac{\exp(-\frac{1}{2}F_{N+1}^T C_{N+1}^T F_{N+1})}{\sqrt{(2\pi)^{N+1} \det(C_{N+1})}} \quad (\text{D.12})$$

On note que la matrice de covariance  $C_{N+1}$  peut s'écrire comme suit :

$$C_{N+1} = \begin{bmatrix} C_N & \Lambda \\ \Lambda^T & \kappa \end{bmatrix} \quad (\text{D.13})$$

où

$$\Lambda = [c(x_1, x_{N+1}), c(x_2, x_{N+1}), \dots, c(x_N, x_{N+1})]^T \quad (\text{D.14})$$

et

$$\kappa = c(x_{N+1}, x_{N+1}) \quad (\text{D.15})$$

On déduit que

$$C_{N+1}^{-1} = \begin{bmatrix} M & \Delta \\ \Delta^T & \mu \end{bmatrix} \quad (\text{D.16})$$

où

$$M = C_N^{-1} + \frac{1}{\mu} \Delta \Delta^T, \quad \Delta = -\mu C_N^{-1} \Lambda, \quad \mu(\kappa - \Lambda^T C_N^{-1} \Lambda)^{-1} \quad (\text{D.17})$$

par conséquent, la densité de probabilité au point  $f_{N+1}$  est

$$p(f_{N+1}|X_{N+1}, F_N) = \exp\left(-\frac{(f_{N+1} - \bar{f}_{N+1})^2}{2\sigma_{f_{N+1}}^2}\right) \quad (\text{D.18})$$

où

$$\bar{f}_{N+1} = \Lambda C_N^{-1} F_N, \quad \sigma_{f_{N+1}}^2 = \kappa - \Lambda^T C_N^{-1} \Lambda \quad (\text{D.19})$$

Les estimations produites par le processus Gaussien dépend entièrement de la matrice de covariance. Plusieurs types de fonction de covariance peuvent être utilisés ; une forme possible est

$$c(x, y) = \theta_1 \exp\left(-\frac{1}{2} \sum_{i=1}^m \frac{(x_i - y_i)^2}{r_i^2}\right) + \theta_2 \quad (\text{D.20})$$

où  $(\theta_1, \theta_2, r_1, r_2, \dots, r_m)$  sont des paramètres à déterminer.



## **Algorithmes hiérarchiques et stratégies de jeux pour l'optimisation multidisciplinaire Application à l'optimisation de la voilure d'un avion d'affaires**

Cette thèse a pour objectif le développement de méthodes numériques innovantes pour la conception optimale de forme en aérodynamique et plus généralement pour les problèmes d'optimisation multicritère ou multidisciplinaire dans un contexte aéronautique. La première partie est consacrée à l'amélioration de l'efficacité des algorithmes d'optimisation de forme en matière de convergence. Dans un premier volet, on a développé des algorithmes d'optimisation multiniveaux qui, à l'instar des méthodes multigrilles particulièrement performantes en convergence itérative, s'appuient sur une hiérarchie de paramétrisations emboîtées. Dans un deuxième volet, on a proposé des techniques d'adaptation automatique de la paramétrisation par régularisation. Par des simulations d'écoulements tridimensionnels autour de géométries de voilures d'avions, on a résolu des problèmes de réduction de traînée en transsonique et de réduction de critère de bruit en supersonique et montré que les algorithmes multiniveaux auto-adaptatifs permettaient de réduire le coût du calcul d'environ un ordre de grandeur. La deuxième partie est consacrée au traitement d'un problème d'optimisation concurrente où le concepteur aérodynamique interagit avec le concepteur structural, parallèlement dans un jeu symétrique de Nash, ou hiérarchiquement dans un jeu de Stackelberg. On a proposé et expérimenté avec succès des algorithmes de calcul d'équilibre pour cette optimisation couplée aéro-structurale dans une situation où le critère aérodynamique est prépondérant.

**Mots-clés :** Optimisation de forme, Algorithme hiérarchique, Aérodynamiques, Voilure d'avion d'affaires, Auto-adaptation de la paramétrisation, Optimisation concurrente, Coque mince, Théorie des jeux.

## **Hierarchical algorithms and game strategies for multidisciplinary optimization Application to the optimization of a business aircraft wing**

This thesis aims at the development of innovating methods for optimum design in aerodynamics and more generally for multicriterion or multidisciplinary optimization problems in the aeronautical context. The first part is devoted to the improvement of efficiency of shape-optimization algorithms in terms of convergence. In a first section, multilevel optimization algorithms inspired from multigrid methods, well-known to be particularly efficient in iterative convergence, have been developed on the basis of a hierarchy of nested parameterizations. In a second section, self-adaptive parameterization procedures by regularization have been proposed. By means of three-dimensional flow simulations about geometries of aircraft wings, we have been able to solve problems of drag reduction in a transonic regime, and noise-criterion reduction in a supersonic regime, and to demonstrate that self-adaptive multilevel algorithms permitted to reduce the cost of an optimization by about one order of magnitude. The second part is devoted to the treatment of a problem of concurrent optimization in which the aerodynamicist interacts with the structural designer, in a parallel way in a symmetric Nash game, or hierarchically in a Stackelberg game. Algorithms for the calculation of the equilibrium point have been proposed and successfully tested for this coupled aero-structural shape optimization in a situation where the aerodynamical criterion is preponderant.

**Key-words :** Shape Optimization, Hierarchical algorithm, Aerodynamics, Business aircraft wing, Self-adaptive parameterization, Concurrent optimization, Thin shell, Game theory.