



HAL
open science

Paramétrisation de Bézier adaptative pour l'optimisation de forme en Aérodynamique

Farid Bélahcène, Jean-Antoine Desideri

► **To cite this version:**

Farid Bélahcène, Jean-Antoine Desideri. Paramétrisation de Bézier adaptative pour l'optimisation de forme en Aérodynamique. [Rapport de recherche] RR-4943, INRIA. 2003. inria-00071636

HAL Id: inria-00071636

<https://inria.hal.science/inria-00071636v1>

Submitted on 23 May 2006

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.

*Paramétrisation de Bézier adaptative pour
l'optimisation de forme en Aérodynamique*

Farid Bélahcène — Jean-Antoine Désidéri

N° 4943

Septembre 2003

THÈME 4



*Rapport
de recherche*

Paramétrisation de Bézier adaptative pour l'optimisation de forme en Aérodynamique

Farid Bélahcène , Jean-Antoine Désidéri

Thème 4 — Simulation et optimisation
de systèmes complexes
Projet Opale

Rapport de recherche n° 4943 — Septembre 2003 — 42 pages

Résumé : Ce rapport fait suite aux rapports [22] et [16], dans lesquels on avait montré (en discret) qu'une optimisation de forme en aérodynamique gagnait significativement en conditionnement par adaptation, à intervalles réguliers, de la paramétrisation de Bézier sur laquelle s'appuie la représentation de la forme. Ici, à partir de la connaissance d'une forme optimale approchée, on considère l'ensemble des paramétrisations de Bézier de même degré qui approchent cette forme au sens des moindres carrés. On choisit celle pour laquelle le polygone de contrôle est le plus régulier (minimisation de la variation totale). On fournit tous les détails de cette formulation, très voisine de celle de nos travaux initiaux mais reposant sur un moins grand nombre de quadratures numériques qui sont pour la plupart réalisées au 4^e ordre. On propose un cas-test d'optimisation de forme issu du calcul des variations. Cet ersatz d'"optimisation physique" nous permet à faible coût de mettre en évidence la convergence du couplage entre l'optimisation pseudo-physique et l'auto-adaptation de la paramétrisation.

Mots-clés : Aérodynamique, éléments finis, optimisation de formes, paramétrisation de Bézier, élévation du degré, algorithmes hiérarchiques

Adaptive Bézier parameterization for shape-optimization in Aerodynamics

Abstract: This report is a sequel of [22] and [16], in which it was shown (with discrete formulations) that a shape optimization in aerodynamics could benefit greatly in conditioning by adaption, at regular intervals, of the parameterization supporting the shape representation. Here, from the knowledge of an approximate optimal shape, one considers the set of all Bézier parameterizations of same degree that approximate the shape in the sense of least squares. One elects the parameterization associated with the most regular control polygon (least total variation). We provide all the details of the formulation, that is very close to our initial works but relies on a lesser number of numerical quadratures most of which are realized by fourth-order formulas. We propose a shape-optimization test-case originating from calculus of variations. This surrogate “physical optimization” allows us, at moderate cost, to demonstrate the convergence of the algorithm coupling the pseudo-physical optimization with the Bézier parameterization adaption.

Key-words: Aerodynamics, finite elements, shape optimization, Bézier parameterization, degree elevation, hierarchical algorithms

Table des matières

1	Introduction	4
2	Paramétrisation de Bézier de courbes planes	6
3	Auto-adaptation de la paramétrisation	7
4	Calcul du critère et de son gradient, validation numérique	13
4.1	Développement formel des calculs.	14
4.2	Validation numérique du calcul formel du critère et de son gradient	17
5	Méthodes numériques pour identifier la paramétrisation adaptée	19
5.1	Méthode du simplexe	20
5.2	Méthode par grilles emboîtées.	20
5.3	Méthode mixte	21
5.4	Conclusion	21
6	Exemple d'optimisation auto-adaptée	23
6.1	Problème test de calcul des variations	23
6.2	Optimisation auto-adaptative	25
7	Conclusion	32
8	Perspectives	33
8.1	Implantation de procédures d'optimisation robustes et efficaces	33
8.2	Précision des intégrations numériques	33
8.3	Transposition des procédures aux <i>B-splines</i>	33
8.4	Examen de cas tests voisins	34
8.5	Généralisation au 2D	34
8.6	Test d'autres procédures d'adaptation	34
8.7	Adaptation et degré de la paramétrisation	35
9	Annexe 1:Programmation.	37
10	Annexe 2:Validation de l'intégration numérique intervenant dans le calcul du périmètre	38

1 Introduction

Cette étude s'insère dans le cadre de l'optimisation de forme en aérodynamique. On ne traite pas ici du problème d'optimisation lui-même, mais pour fixer les idées, on pense à une méthode numérique de type éléments finis pour calculer un écoulement dans un domaine dont l'une des frontières, le corps, est à optimiser de manière à minimiser une fonctionnelle coût liée à un critère de performance aérodynamique. Le contexte de nos méthodes est largement illustré par de nombreux exemples de telles optimisations par algorithmes génétiques [4] [9] [5] [6] [7] [10] [15] [14] [23] [22] [16] [20] [16] [21] [26] [25] [31] [30] [17] [28] [32] [29], qui traitent d'optimisation aérodynamique monocritère (réduction de traînée en transsonique ou maximisation de portance en régime subsonique de décollage ou d'atterrissage, sous contraintes) ou multicritère (équilibre de Pareto, stratégies de jeux dynamiques) de géométries 2D ou 3D (profils d'ailes, voilures ou dispositifs multicorps hypersustentateurs), ou méthodes de gradient [2] [3] [8] [12] [24] ou hybrides, dans des applications analogues.

Lorsque l'outil d'optimisation est une méthode de type gradient, l'algorithme numérique fournit directement des approximations des dérivées partielles de la fonctionnelle coût par rapport aux coordonnées des nœuds frontières. Dans ce contexte, on peut concevoir des algorithmes itératifs qui opèrent explicitement sur ces coordonnées. Une représentation naturelle de la frontière optimisée est alors fournie par la trace du maillage éléments finis. Cependant, le nombre souhaitable de variables d'optimisation, ou paramètres de *design*, dans une optimisation pratique est généralement un ou deux ordres de grandeur inférieur au nombre de nœuds frontières, en raison de la raideur numérique qui en est une fonction rapidement croissante. Il convient donc, d'une manière explicite ou implicite, d'adopter une procédure de *réduction* de la dimension. En outre, ces dérivées partielles qui sont dans un espace de régularité moindre que le champ et *a fortiori* les géométries elles-mêmes (qui sont lisses), sont généralement numériquement bruitées. Il convient donc que la réduction de dimension s'accompagne d'une forme de *filtrage* ou de *régularisation*. Enfin, certains travaux de Dervieux, Marco *et al* [12] ont montré l'intérêt de construire un filtre *hiérarchique*, inspiré de la méthode multigrille par agglomération des volumes. Même si ces auteurs n'emploient pas toujours explicitement le terme de paramétrisation, le filtre hiérarchique auquel ils ont abouti est une forme de paramétrisation lisse de la forme pour laquelle les variables sont des paramètres géométriques associés aux volumes de contrôle les plus grossiers, obtenus après plusieurs agglomérations.

Dans le contexte un peu différent où l'outil d'optimisation est un algorithme évolutionnaire, par exemple génétique, il est indispensable d'opter pour une représentation explicite, lisse et de dimension réduite de la forme, car l'opérateur de croisement ne peut s'appliquer directement à la géométrie sans que cela génère des formes non admissibles. En outre, pour des raisons d'efficacité, il est souhaitable dans ce contexte également, que la paramétrisation se prête à la construction d'algorithmes hiérarchiques, multiniveaux.

Guidés par ces considérations générales, on a opté, dans les applications aux écoulements bidimensionnels autour de profils d'aile, d'approcher ces profils par des représentations de Bézier de courbes planes qui répondent à la nécessité de réduction de dimension (par le faible nombre de points de contrôle nécessaires), de régularisation (par le fait que les courbes planes générées sont lisses), et de construction hiérarchique (sur la base de la propriété d'élévation du degré explorée dans [17] et rappelée brièvement plus loin).

Tout comme en éléments finis, au cours de la résolution itérative d'une équation aux dérivées partielles (E.D.P.), se pose le problème de l'adaptation du maillage, en optimisation de

forme, lorsqu'une paramétrisation particulière est adoptée, se pose la question de son adaptation au cours du processus itératif.

Dans la situation "réelle" d'un problème d'aérodynamique, l'optimisation "physique" par voie numérique est très coûteuse car chaque évaluation de fonctionnelle nécessite le calcul de l'écoulement c'est-à-dire la résolution d'un jeu d'E.D.P., équations d'Euler et parfois même de Navier-Stokes en fluide compressible, dans un nouveau domaine. Quelquefois, pour réduire ce coût on n'utilise pas systématiquement le modèle réel, mais un ersatz construit par une technique de réduction de modèle telle que P.O.D. (*Proper Orthogonal Decomposition*, [11]), ou un réseau de neurones [13] [19].

En outre, la vitesse de convergence des optimiseurs courants (méthode de descente basée sur une approximation du gradient, ou algorithmes évolutionnaires), décroît nonlinéairement, et le plus souvent malheureusement plus que quadratiquement, lorsqu'on augmente le nombre de paramètres.

Sans donner tous les détails de notre stratégie d'adaptation, qui sont fournis à la Section 3, on comprend qu'en partitionnant l'ensemble des paramètres d'optimisation en deux sous-ensembles, ξ (lié aux abscisses de contrôle) et η (lié aux ordonnées de contrôle), on peut réduire substantiellement le coût d'une optimisation physique en la réalisant par optimisation du vecteur η seulement, le vecteur ξ étant fixé (étape 1). Ceci est particulièrement naturel dans le cas d'un profil d'aile qui est plus long qu'épais.

Cette réduction de l'espace de recherche, pour ne pas être trop pénalisante, doit s'accompagner d'une optimisation symétrique du vecteur ξ , le vecteur η étant fixé (étape 2), ce qui correspond à intervertir les rôles de x et y . On peut conduire ces deux étapes d'optimisation en minimisant le même critère physique. Mais ceci n'est pas toujours aussi simple qu'il ne paraît du point de vue de la programmation. En effet, certains codes (en aérodynamique) tiennent compte dans la représentation de la forme du partitionnement extrados/intrados (haut/bas) du profil pour lequel la formulation $y = y(x)$ est plus naturelle que la formulation $x = x(y)$ qu'un partitionnement avant/arrière exigerait. Lorsque les perturbations sont faibles (en fin de convergence), l'interversion est souvent viable; mais dans le cas inverse de grandes perturbations, elle peut mettre en défaut la représentation posée *a priori*.

Pour ces raisons, on a choisi de conduire la deuxième étape par optimisation d'un critère purement géométrique de régularisation de la paramétrisation. Dans ce cas, on qualifie cette étape d'"adaptation de la paramétrisation" à l'approximation courante de la forme optimisée vis-à-vis du critère physique. Cette adaptation géométrique a un coût marginal.

On peut alors voir le couplage optimisation physique-adaptation de la paramétrisation comme un jeu de Stackelberg [1], dont les étapes 1 et 2 correspondent respectivement aux stratégies du meneur (*leader*) et du suiveur (*follower*). Ce processus est tout à fait semblable à l'adaptation statique d'un maillage au cours de la résolution itérative d'une E.D.P..

Nos principaux travaux en matière d'adaptation de la paramétrisation ont été publiés dans les rapports [22] et [16]. Bien que nous ne disposions pas encore d'argument théorique totalement satisfaisant, de nombreuses expériences numériques ont montré que d'importants gains en efficacité d'optimisation pouvaient être atteints en procédant, au cours de la convergence, à quelques réadaptations de la paramétrisation de Bézier consistant à régulariser le polygone de contrôle par réduction d'un critère de variation totale.

Dans ce rapport, on fournit les détails de formulation de la procédure d'auto-adaptation et on donne un exemple d'optimisation adaptée dans le cas d'un problème très simple issu du calcul des variations, ersatz d'une véritable optimisation physique. Au préalable, quelques rappels sont faits des propriétés élémentaires des représentations de Bézier.

2 Paramétrisation de Bézier de courbes planes

Dans cette section, on rappelle la définition et deux propriétés importantes des paramétrisations de Bézier. On renvoie à [36] pour une plus ample présentation de ces notions.

Définition. Le point courant, $P(t)$, d'un arc paramétré de Bézier de degré n se définit de la manière suivante:

$$P(t) = \sum_{k=0}^n B_n^k(t) P_k \quad (1)$$

où le paramètre t varie de 0 à 1,

$$B_n^k(t) = C_n^k t^k (1-t)^{n-k} \quad (2)$$

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

$$P_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix} \quad (k = 0, 1, \dots, n) \quad (3)$$

est le "point de contrôle" générique.

Dans cette expression les $(n+1)$ points de contrôles P_k n'appartiennent généralement pas à la courbe de Bézier, à l'exception des points extrêmes $P_0 = P(0)$ et $P_n = P(1)$.

Formule des dérivées successives. La dérivée d'ordre α ($\alpha \leq n$) de $P(t)$ s'exprime comme suit:

$$P^{(\alpha)}(t) = n(n-1)\dots(n-\alpha+1) \sum_{k=0}^{n-\alpha} B_{n-\alpha}^k(t) \Delta^\alpha P_k \quad (4)$$

où Δ est l'opérateur de différence avancée ($\Delta P_k = P_{k+1} - P_k$), et Δ^α la puissance α de cet opérateur ($\Delta^2 P_k = P_{k+2} - 2P_{k+1} + P_k$, ...). En particulier:

$$x'(t) = n \sum_{k=0}^{n-1} B_{n-1}^k(t) (x_{k+1} - x_k) \quad (5)$$

Cette équation montre en particulier qu'à une suite $\{x_k\}$ ($k = 0, \dots, n$) croissante correspond une fonction $x(t)$ croissante (quand t croît). Nous imposons cette condition dans nos applications aux profils d'aile.

On constate également que la spécification de la dérivée $P^{(\alpha)}(0)$ (resp. $P^{(\alpha)}(1)$) conduit à une condition linéaire locale portant seulement sur les points de contrôle $P_0, P_1, \dots, P_\alpha$ (resp. $P_n, P_{n-1}, \dots, P_{n-\alpha}$). Par conséquent, la spécification de $P(0), P'(0), \dots, P^{(\alpha)}(0)$ (resp. $P(1), P'(1), \dots, P^{(\alpha)}(1)$) peut se faire par résolution d'un système linéaire local triangulaire.

Élévation du degré. En multipliant $P(t)$ par $1 = (1 - t) + t$, il vient:

$$\begin{aligned}
P(t) &= \sum_{k=0}^n C_n^k t^k (1-t)^{n+1-k} P_k + \sum_{k=0}^n C_n^k t^{k+1} (1-t)^{n-k} P_k \\
&= \sum_{k=0}^n C_n^k t^k (1-t)^{n+1-k} P_k + \sum_{k=1}^{n+1} C_n^{k-1} t^k (1-t)^{n+1-k} P_{k-1} \\
&= (1-t)^{n+1} P_0 + t^{n+1} P_n + \sum_{k=1}^n [C_n^k P_k + C_n^{k-1} P_{k-1}] t^k (1-t)^{n+1-k} \\
&= \sum_{k=0}^{n+1} \underbrace{C_{n+1}^k t^k (1-t)^{n+1-k}}_{B_{n+1}^k(t)} P'_k
\end{aligned}$$

à condition de définir la nouvelle suite de points de contrôle $\{P'_k\}$ ($k = 0, \dots, n+1$) de la manière suivante:

$$\begin{cases} P'_0 = P_0 \\ P'_k = \frac{k}{n+1} P_{k-1} + (1 - \frac{k}{n+1}) P_k & (k = 1, \dots, n) \\ P'_{n+1} = P_n \end{cases} \quad (6)$$

Par conséquent, par ce processus, sans changer l'arc paramétré, ni sa description lorsque t varie, on peut augmenter indéfiniment le degré de la paramétrisation de Bézier, par construction de nouveaux points de contrôle obtenus par combinaisons convexes des précédents. Le processus a donc pour effet de produire des polygones de contrôle de plus en plus lisses. On démontre par passage à la limite ($n \rightarrow \infty$), que le polygone de contrôle converge vers l'arc lui-même. En particulier, si dans cette suite, un polygone est convexe, tous les suivants le sont également, et l'arc paramétré l'est aussi.

Un arc de courbe de Bézier est donc la limite de polygones de contrôle obtenus par élévations successives du degré, c'est-à-dire par itérations d'un processus particulier de lissage. A l'inverse, un polygone de contrôle très irrégulier est la marque d'un stade peu avancé dans cette convergence. Cette observation nous a conduit à l'idée intuitive suivante: pour accélérer la convergence de la représentation d'une forme optimisée, il convient sans trop changer la forme d'en régulariser le polygone de contrôle. Notre procédure d'auto-adaptation de la paramétrisation met en œuvre ce principe intuitif.

Par ailleurs, la technique d'élévation du degré permet de construire des espaces de représentation emboîtés et des algorithmes d'optimisation de forme multiniveaux associés inspirés des stratégies multigrilles dont l'efficacité est optimale [17].

3 Auto-adaptation de la paramétrisation

On fait l'hypothèse qu'on optimise une forme représentée par la paramétrisation de Bézier précédente, en fixant *a priori* les abscisses $\{x_k\}$ ($k = 0, \dots, n$) des points de contrôle et en itérant sur les ordonnées correspondantes $\{y_k\}$ ($k = 0, \dots, n$) de manière à minimiser une

fonctionnelle coût ayant une certaine pertinence vis-à-vis du problème physique traité¹. A un certain stade de ce processus d'optimisation, on dispose d'une approximation de la forme optimale que l'on souhaite utiliser pour redéfinir une suite "mieux adaptée" d'abscisses $\{x_k\}$ et redémarrer le processus d'optimisation.

Pour mettre en œuvre ce principe général, on considère l'approximation courante de la forme comme une courbe cible (ou plusieurs arcs formant une cible continue et lisse par morceaux). On considère l'ensemble des paramétrisations de Bézier de même degré n , chacune associée à une suite particulière $\{x_k\}$ (monotone croissante et satisfaisant certaines contraintes géométriques) et approchant la cible au sens des moindres carrés; la suite $\{y_k\}$ étant solution des équations normales est une fonction implicite de la suite $\{x_k\}$. Dans cet ensemble de représentations disponibles, on souhaite identifier la plus régulière, c'est-à-dire, d'après le principe intuitif exposé à la section précédente, celle dont le polygone de contrôle est associé à la plus petite valeur du critère suivant de variation totale:

$$TV(\{y_k\}) = \sum_{k=1}^n |y_k - y_{k-1}| \quad (7)$$

Pour expliciter les calculs, on traite désormais le cas d'une forme cible constituée par une seule courbe plane d'équation:

$$y = y_T(x) \quad (0 \leq x \leq 1) \quad (8)$$

bien qu'en pratique pour l'optimisation aérodynamique d'un profil, on aura deux telles courbes, tangentes à l'origine (bord d'attaque) à l'axe des y et se rejoignant au bord de fuite de coordonnées $(1,0)$. Pour initialiser l'algorithme numérique (avant toute optimisation physique), on choisit comme forme cible initiale le profil d'aile RAE2822, documenté et tabulé dans le cadre du Projet Européen ECARP [37] et connu pour sa faible traînée en régime transsonique (voir Figure 1).

On introduit les notations vectorielles suivantes dans \mathbb{R}^{n+1} :

$$X = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad Y = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad B_n(t) = \begin{pmatrix} B_n^0(t) \\ B_n^1(t) \\ \vdots \\ B_n^n(t) \end{pmatrix}, \quad (9)$$

de sorte que:

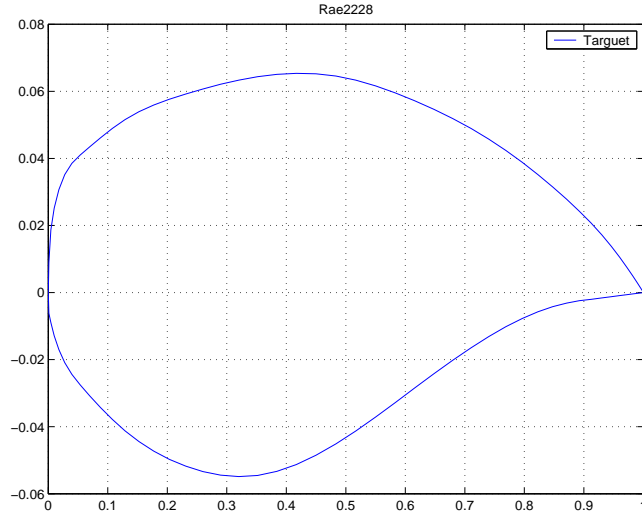
$$\begin{cases} x(t) = B_n(t)^T X \\ y(t) = B_n(t)^T Y \end{cases} \quad (10)$$

où l'indice supérieur T indique une transposition.

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

$$x_0 = y_0 = 0 \quad x_{n-1} = y_{n-1} = 0 \quad (11)$$

1. Dans le cas de l'optimisation d'un profil d'aile, c'est-à-dire d'un contour fermé, on utilise deux paramétrisations de Bézier indépendantes pour l'extrados et l'intrados séparément.


 FIG. 1 – *Forme cible initiale (profil d'aile RAE2222)*

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

$$x_1 = x_0 (= 0) \quad (12)$$

qui implique que $x'(0) = 0$. Ces spécifications permettent de réduire le jeu (X, Y) de $2(n + 1)$ paramètres à un sous-ensemble (ξ, η) tel que:

$$\begin{cases} X = E \xi + e \\ Y = F \eta + f \end{cases} \quad (13)$$

où

$$\xi = \begin{pmatrix} x_2 \\ x_3 \\ \vdots \\ x_{n-1} \end{pmatrix}, \quad \eta = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{pmatrix} \quad (14)$$

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

$$e = \hat{e}_{n-2} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \quad f = 0 \quad (15)$$

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

$$E = \hat{E}_{n-2} = \begin{pmatrix} \dots & 0 & \dots \\ \dots & 0 & \dots \\ & I_{n-2} & \\ \dots & 0 & \dots \end{pmatrix}, \quad F = \hat{F}_{n-1} = \begin{pmatrix} \dots & 0 & \dots \\ & I_{n-1} & \\ \dots & 0 & \dots \end{pmatrix} \quad (16)$$

Les vecteurs sous les accolades indiquent la dépendance de chaque terme de X ou Y .

Compte tenu des contraintes géométriques, la fonctionnelle J est considérée comme une fonction des vecteurs colonnes réduits ξ et η .

On souhaite imposer la contrainte suivante: à X fixé (i.e. à ξ fixé), Y (i.e. η) minimise J , ce qui équivaut à la condition suivante:

$$\frac{\partial J}{\partial \eta} = \frac{\partial J}{\partial Y} F = 0 \quad (23)$$

Or,

$$\begin{aligned} \frac{\partial J}{\partial Y} &= \int_0^1 [y(t) - \bar{y}_T(t, X)] \frac{\partial y}{\partial Y} x'(t) dt \\ &= \int_0^1 \underbrace{[B_n(t)^T Y - \bar{y}_T(t, X)]}_{\text{scalaire}} B_n(t)^T \underbrace{nB_{n-1}(t)^T \Delta X}_{\text{scalaire}} dt \end{aligned} \quad (24)$$

où désormais Δ est la matrice $n \times (n+1)$:

$$\Delta = \begin{pmatrix} -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & -1 & 1 & & \\ & & & \ddots & \ddots & \\ & & & & -1 & 1 \end{pmatrix} \quad (25)$$

Il vient:

$$\left(\frac{\partial J}{\partial Y} \right)^T = \int_0^1 B_n(t) [B_n(t)^T Y - \bar{y}_T(t, X)] nB_{n-1}(t)^T \Delta X dt = A(X) Y - b(X) \quad (26)$$

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

$$A(X) = \int_0^1 \underbrace{B_n(t) B_n(t)^T}_{\substack{\text{matrice} \\ (n+1) \times (n+1)}} \underbrace{nB_{n-1}(t)^T \Delta X}_{\substack{\text{forme linéaire} \\ \text{en } X}} dt \quad (27)$$

et le vecteur

$$b(X) = \int_0^1 \underbrace{B_n(t)}_{\text{vect. col.}} \underbrace{\bar{y}_T(t, X) nB_{n-1}(t)^T \Delta X}_{\substack{\text{fonct. nonlinéaire} \\ \text{scalaire de } X}} dt \quad (28)$$

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 :

$$A(X) = \mathcal{A}.X \quad (29)$$

où \mathcal{A} est un tenseur d'ordre 3, et $.$ correspond à un certain produit contracté.

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

$$\left(\frac{\partial J}{\partial \eta}\right)^T = F^T \left(A(X) Y - b(X) \right) = 0 \quad (30)$$

Régularisation du polygone de contrôle. Les expériences d'optimisation de forme aérodynamique ont montré qu'il était avantageux de régulariser la paramétrisation en minimisant la variation totale des ordonnées des points de contrôle:

$$TV(Y) = \sum_{k=1}^n |y_k - y_{k-1}| = J_2(\xi) \quad (31)$$

sous les contraintes précédentes (contraintes géométriques et contrainte de meilleure approximation), où la dépendance implicite en ξ est le fait de ces contraintes.

Pour réaliser numériquement la minimisation de la fonctionnelle $J_2(Y)$, il peut être intéressant de disposer de son gradient, dont on précise maintenant le calcul.

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

$$\frac{\partial J_2}{\partial Y} = p(Y)^T \quad (32)$$

où

$$p_i(Y) = \begin{cases} +2 & \text{si } y_i \text{ est un maximum local de la suite } \{y_k\}, \\ -2 & \text{si } y_i \text{ est un minimum local de la suite } \{y_k\}, \\ 0 & \text{autrement.} \end{cases} \quad (33)$$

Il vient:

$$\frac{\partial J_2}{\partial \eta} = \frac{\partial J_2}{\partial Y} \frac{\partial Y}{\partial \eta} = p(Y)^T F \quad (34)$$

puis

$$\frac{\partial J_2}{\partial \xi} = \frac{\partial J_2}{\partial \eta} \frac{\partial \eta}{\partial \xi} = p(Y)^T F \frac{\partial \eta}{\partial \xi} \quad (35)$$

Le facteur

$$q = \frac{\partial \eta}{\partial \xi} \quad (36)$$

s'obtient en dérivant par rapport à ξ la contrainte de meilleure approximation (au sens des moindres carrés)

$$F^T \left[A'(X) \underbrace{Y}_{F \eta + f} - b'(X) + A(X) F \frac{\partial \eta}{\partial X} \right] E = 0 \quad (37)$$

où l'indice supérieur ' indique une dérivation par rapport au vecteur X , et $\frac{\partial \eta}{\partial X} E = \frac{\partial \eta}{\partial \xi} = q$ est le facteur cherché. Notant que

$$A'(X) Y = A(Y) \quad (38)$$

car la matrice A dépend linéairement du vecteur X , il vient:

$$F^T [A(Y) - b'(X)] E + F^T A(X) F q = 0 \quad (39)$$

ce qui définit le vecteur q par la résolution d'un système linéaire.

Ces équations générales sont reprises et détaillées à la section suivante, où on valide leur calcul numérique par comparaison à des formules de différences finies.

4 Calcul du critère et de son gradient, validation numérique

Reprenant les résultats formels de la section précédente, on résume d'abord l'ensemble des étapes pour calculer le critère J_2 et son gradient $\frac{\partial J_2}{\partial \xi}$, ensuite on développe les calculs de la matrice $A(X)$, du vecteur $b(X)$, puis ceux de la matrice $b'(X)$, et enfin on présente les applications numériques.

1. CALCUL DU CRITÈRE:

- Calculer le vecteur

$$X = E \xi + e$$

ainsi que la matrice $A(X)$ et le vecteur $b(X)$.

- Résoudre le système linéaire

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

- Calculer le vecteur

$$Y = F \eta + f$$

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

2. CALCUL DU GRADIENT DU CRITÈRE (Première méthode):

- Calculer le vecteur

$$p = p(Y)$$

et les matrices $A(Y)$ et $b'(X)$.

- Résoudre le système linéaire

$$F^T A(X) F q = F^T [b'(X) - A(Y)] E$$

- Calculer la dérivée

$$\frac{\partial J_2}{\partial \xi} = p^T F q.$$

Remarque: le calcul rigoureux de la dérivée $b'(X)$ inclut deux termes qui font intervenir respectivement la fonction $y_T(x)$ et sa dérivée :

$$b'(x) = \int_0^1 B_n(t) \bar{y}_T(t, X) n B_{n-1}(t)^T \Delta dt + \int_0^1 B_n(t) \bar{y}_T'(t, X) n B_{n-1}(t)^T \Delta X dt \quad (40)$$

4.1 Développement formel des calculs.

– Calcul de la matrice $A(X)$

$$A(X) = \int_0^1 \underbrace{B_n(t) B_n(t)^T}_{\substack{\text{matrice} \\ (n+1) \times (n+1)}} \underbrace{n B_{n-1}(t)^T \Delta X}_{\substack{\text{forme linéaire} \\ \text{en } X}} dt \quad (41)$$

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

$$\begin{aligned} A(X)(i, j) &= \int_0^1 B_n^i(t) B_n^j(t) n \sum_{k=0}^{n-1} B_{n-1}^k(t) (x_{k+1} - x_k) dt \\ &= n \sum_{k=0}^{n-1} (x_{k+1} - x_k) \int_0^1 B_n^i(t) B_n^j(t) B_{n-1}^k(t) dt \\ &= n \sum_{k=0}^{n-1} (x_{k+1} - x_k) C_n^i C_n^j C_{n-1}^k \int_0^1 t^{i+j+k} (1-t)^{3n-1-i-j-k} dt \\ &= n \sum_{k=0}^{n-1} (x_{k+1} - x_k) \frac{C_n^i C_n^j C_{n-1}^k}{3n C_{3n-1}^{i+j+k}} \end{aligned} \quad (42)$$

Finalement:

$$A(X)(i, j) = \frac{1}{3} \sum_{k=0}^{n-1} (x_{k+1} - x_k) \frac{C_n^i C_n^j C_{n-1}^k}{C_{3n-1}^{i+j+k}} \quad (43)$$

– Calcul du vecteur $b(X)$

$$b(X) = \int_0^1 \underbrace{B_n(t)}_{\text{vect. col.}} \underbrace{\bar{y}_T(t, X) n B_{n-1}(t)^T \Delta X}_{\substack{\text{fonct. nonlinéaire} \\ \text{scalaire de } X}} dt \quad (44)$$

Pour $0 \leq i \leq n$:

$$b(X)(i) = \int_0^1 B_n^i(t) \bar{y}_T(t, X) n \sum_{k=0}^{n-1} B_{n-1}^k(t) (x_{k+1} - x_k) dt \quad (45)$$

$$b(X)(i) = n \sum_{k=0}^{n-1} (x_{k+1} - x_k) C_n^i C_{n-1}^k \int_0^1 \bar{y}_T(t, X) t^{i+k} (1-t)^{2n-1-i-k} dt \quad (46)$$

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

$$b(X)(i) = n \sum_{k=0}^{n-1} (x_{k+1} - x_k) C_n^i C_{n-1}^k \int_0^1 f_{i,k}(t) dt \quad (47)$$

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:

$$b(X)(i) \simeq n \sum_{k=0}^{n-1} (x_{k+1} - x_k) C_n^i C_{n-1}^k \sum_{l=0}^{M-1} (t_{l+1} - t_l) \frac{(f_{i,k}(t_{l+1}) + f_{i,k}(t_l))}{2} \quad (48)$$

et enfin:

$$b(X)(i) \simeq n \sum_{k=0}^{n-1} \sum_{j=0}^{m-1} (x_{k+1} - x_k) C_n^i C_{n-1}^k (t_{j+1} - t_j) \frac{(f_{i,k}(t_{j+1}) + f_{i,k}(t_j))}{2} dt \quad (49)$$

Remarque: Ordre de précision. Classiquement, lorsque l'intégrande est suffisamment régulier (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 [35] s'appuie sur une interpolation hermitienne du 3^e 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^e ordre (en supposant l'intégrande C^4). Or la dérivée de la fonction $f_{i,k}(t)$ s'exprime comme suit:

$$f'_{i,k}(t) = \bar{y}_T(t, X)' t^{i+k} (1-t)^{2n-1-i-k} + \bar{y}_T(t, X) \frac{d}{dt} t^{i+k} (1-t)^{2n-1-i-k} \quad (50)$$

On a: $0 \leq i \leq n$ et $0 \leq k \leq n-1$. Par conséquent: $0 \leq i+k \leq 2n-1$. 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^e ordre, sauf peut-être pour $(i,k) = (0,0)$ et $(n,n-1)$.

- Calcul de la matrice $A(Y)$

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

$$A(Y)(i, j) = \frac{1}{3} \sum_{k=0}^{n-1} (y_{k+1} - y_k) \frac{C_n^i C_n^j C_{n-1}^k}{C_{3n-1}^{i+j+k}} \quad (51)$$

– Calcul de la matrice $b'(X)$

$$b'(x) = b'_1(X) + b'_2(X) \quad (52)$$

où :

$$\begin{cases} b'_1(X) = \int_0^1 B_n(t) \bar{y}_T(t, X) n B_{n-1}(t)^T \Delta dt \\ b'_2(X) = \int_0^1 B_n(t) \bar{y}_T'(t, X) \underbrace{n B_{n-1}(t)^T \Delta X}_{x'(t)} dt \end{cases} \quad (53)$$

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

On va d'abord calculer $b'_1(X)$ et $b'_2(X)$ séparément.

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

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

Le calcul de $b'_2(X)$ est plus complexe, car il fait intervenir la dérivée $\bar{y}_T'(t, X)$. Pour contourner cette difficulté, on calcule l'intégrale en $\bar{y}_T'(t, X)$ par application des règles de dérivation des fonctions composées et d'intégration par parties :

$$\begin{aligned} b'_2(X) &= \int_0^1 B_n(t) \frac{\partial}{\partial t} \bar{y}_T \frac{\partial t}{\partial x} \frac{\partial x}{\partial X} x'(t) dt \\ &= \int_0^1 B_n(t) \frac{\partial}{\partial t} \bar{y}_T(t, X) B_n(t)^T dt \end{aligned}$$

On intègre par partie, on obtient :

$$\begin{aligned} b'_2(X) &= \left[\bar{y}_T(t, X) B_n(t)^T B_n(t)^T \right]_0^1 - \int_0^1 \bar{y}_T(t, X) \frac{\partial}{\partial t} (B_n(t) B_n(t)^T) dt \\ &= - \int_0^1 \bar{y}_T(t, X) \frac{\partial}{\partial t} (B_n(t) B_n(t)^T) dt \quad (\text{car } \bar{y}_T(0) = \bar{y}_T(1) = 0) \end{aligned}$$

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

$$\begin{aligned}
 b'_2(X)(i,j) &= - \int_0^1 \bar{y}_T(t,X) \frac{\partial}{\partial t} \left(B_n^i(t) B_n^j(t) \right) dt \\
 &= C_n^i C_n^j \int_0^1 \bar{y}_T(t,X) t^{i+j-1} (1-t)^{2n-1-i-j} \left(-(i+j)(1-t) + (2n-i-j)t \right) dt \\
 &= C_n^i C_n^j \int_0^1 \bar{y}_T(t,X) t^{i+j-1} (1-t)^{2n-1-i-j} \left(-i-j+2nt \right) dt
 \end{aligned}$$

Après simplification de la somme de $b'_1(X) + b'_2(X)$, il vient :

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

$$b'(X)(i,j) = C_n^i C_n^j \int_0^1 (nt - i) \bar{y}_T(t,X) t^{i+j-1} (1-t)^{2n-1-i-j} dt \quad (54)$$

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

$$b'(X)(i,j) = C_n^i C_n^j \int_0^1 g_{i,j}(t) dt \quad (55)$$

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 :

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

Remarque: Ordre de précision. 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 :

$$g'_{i,j}(0) = g'_{i,j}(1) = 0 \quad (56)$$

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

4.2 Validation numérique du calcul formel du critère et de son gradient

La première expérience consiste à chercher une approximation de la courbe cible (sans optimisation de critère physique), par une courbe de Bézier de degré $n = 16$.

On se donne un vecteur $\xi(x_2, x_3)$ des abscisses de contrôle d'une paramétrisation de degré $p = 4$ (spécifiquement $x_2 = \frac{1}{3}$ et $x_3 = \frac{2}{3}$ dans notre cas). On applique 12 fois le processus d'élévation de degré, ce qui fournit le vecteur X abscisses des points de contrôle de la paramétrisation de degré $n = 16$. Puis on calcule le vecteur Y des ordonnées de contrôle réalisant la meilleure approximation au sens des moindres carrés. On trace alors la courbe de Bézier de degré 16 et on la compare à la courbe cible (voir Figure 2). On observe que les deux courbes

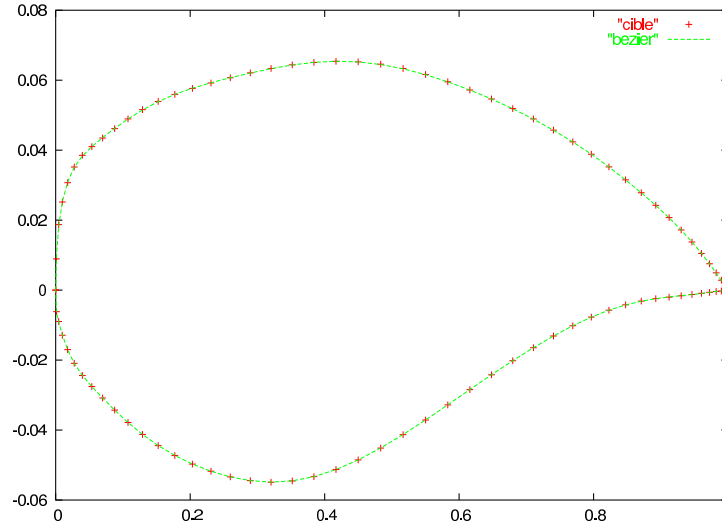


FIG. 2 – Courbe cible et son approximant de Bézier de degré 16.

sont quasiment confondues, ce qui est confirmé par la valeur de la norme L_2 (écart quadratique entre les deux courbes) qui est de l'ordre de 10^{-5} .

La Table 1 indique les valeurs obtenues de la variation totale TV , la norme L_2 de l'écart quadratique entre la courbe et la paramétrisation de Bézier, et les valeurs maximales et minimales des ordonnées de contrôle.

	x_2	x_3	norme L_2 ($\times 10^{-5}$)	max y_k	min y_k	TV
intrados	1/3	2/3	6.507	1.64	-1.33	15.293
extrados	1/3	2/3	2.016	0.096	-1.08	9.566

TAB. 1 – Meilleure approximation du profil RAE2822 de variation totale minimale

A présent, on souhaite valider le calcul formel (et sa programmation) du gradient de critère par comparaison à un calcul par différences finies:

$$\frac{\partial J_2}{\partial x_2} \simeq \frac{J_2(\xi + \epsilon \vec{e}_1) - J_2(\xi)}{\epsilon} \quad (57)$$

et

$$\frac{\partial J_2}{\partial x_3} \simeq \frac{J_2(\xi + \epsilon \vec{e}_2) - J_2(\xi)}{\epsilon} \quad (58)$$

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

Aux Figures 3 et 4, on a représenté les dérivées partielles du critère par rapport aux variables x_2 et x_3 respectivement, en comparant le résultat par évaluation de formule (à gauche) à celui par différence divisée (à droite). On remarque quelques différences, mais dans l'ensemble une très grande similitude, qui nous permet de conclure à la validation de la formule et de sa programmation.

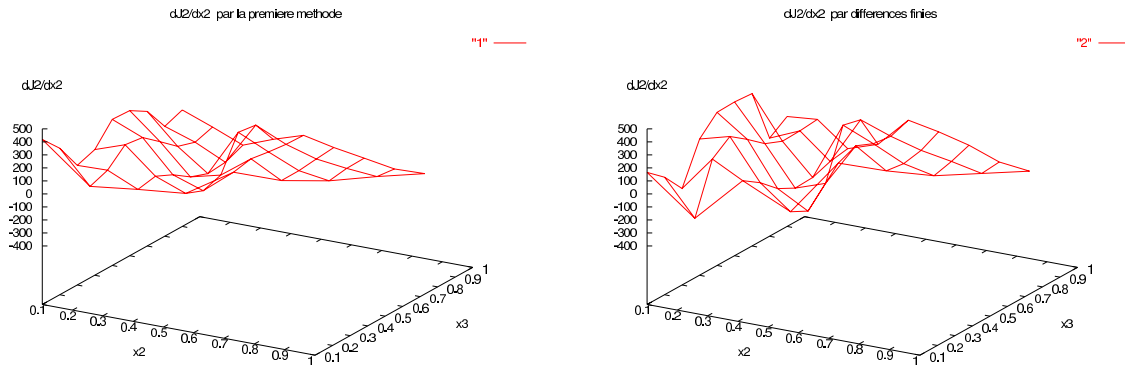


FIG. 3 – Dérivée partielle $\frac{\partial J_2}{\partial x_2}(x_2, x_3)$ calculée par évaluation de formule (à gauche), et par différences finies (à droite).

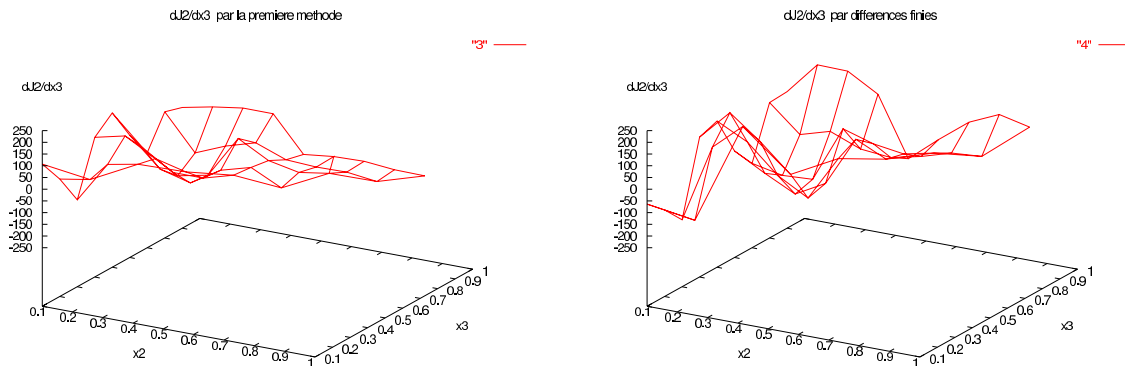


FIG. 4 – Dérivée partielle $\frac{\partial J_2}{\partial x_3}(x_2, x_3)$ calculée par évaluation de formule (à gauche), et par différences finies (à droite).

5 Méthodes numériques pour identifier la paramétrisation adaptée

Disposant désormais de procédures validées pour le calcul du critère (J_2) et de son gradient, on propose ici la construction de plusieurs méthodes numériques pour réaliser la minimisation du critère et on les compare entre elles.

On se place dans le cas où l'on optimise les deux abscisses x_2 et x_3 d'une paramétrisation (grossière) de degré 4 de manière à minimiser la variation totale J_2 de la paramétrisation (fine) de degré 16 obtenue par 12 élévations du degré et calcul de la meilleure approximation au sens des moindres carrés.

5.1 Méthode du simplexe

On a choisi de tester d'abord cette méthode classique connue pour sa robustesse et ne nécessitant pas l'évaluation du gradient.

Pour la réalisation de cette méthode, on a utilisé la procédure E04JBC de la bibliothèque scientifique NAG (minimisation sous les contraintes $0 \leq x_2, x_3 \leq 1$).

L'expérimentation de cette méthode s'est révélée un peu décevante. Tout d'abord, il est difficile de bien régler tous les paramètres optionnels de la procédure NAG; ces paramètres permettent à la procédure de réaliser divers contrôles systématiques de convergence en fonction de la précision arithmétique (machine) attendue, dont il est difficile d'apprécier la pertinence générale pour notre problème; en définitive, la procédure effectue plus de 800 appels à la fonction et environ 60 itérations, sans satisfaire tous les critères d'arrêt. La manipulation de cette procédure en "boîte noire" est donc un peu délicate. Plus gravement, pour certaines conditions initiales, on a atteint à convergence un optimum local non global.

Les résultats de convergence sont consignés à la Table 2 dont la première ligne correspond à l'optimum global et les deux autres à des cas d'optimums locaux.

x_2 initial	x_3 initial	x_2 final	x_3 final	norme- L_2 ($\times 10^{-5}$)	TV
0.1	0.4	0.078	0.414	5.6731	1.686
0.4	0.6	0.36	0.73	5.6625	2.398
0.5	0.8	0.348	0.758	5.62635	1.768

TAB. 2 – Méthode du simplexe

5.2 Méthode par grilles emboîtées.

L'expérience précédente ayant révélé que même la méthode robuste du simplexe pouvait converger vers un optimum local dans le cas du problème traité. On s'est donc dirigé vers une méthode encore plus robuste consistant à d'abord localiser l'optimum global en discrétisant assez finement la fonction sur tout le domaine, puis en raffinant l'identification en plusieurs iterations; à chaque itération, on discrétise plus finement un sous-domaine moins étendu.

Première étape : On discrétise le carré unité avec un pas uniforme et égal à $h_1 = \frac{1}{10}$. La fonction J_2 est évaluée en tous les nœuds de la partie triangulaire supérieure ($x_2 \leq x_3$); voir Figure 5.

On note $\xi^{(1)} = (x_2^{(1)}, x_3^{(1)})$ le point correspondant à la plus petite variation totale.

Deuxième étape : On pose $h_2 = \frac{h_1}{10}$. On discrétise avec le pas h_2 le carré centré en $\xi^{(1)}$ de côté $2h_1$.

On note $\xi^{(2)} = (x_2^{(2)}, x_3^{(2)})$ le nouveau meilleur point.

Troisième étape : On réitère l'opération précédente autour de $\xi^{(2)}$ avec le pas $h_3 = \frac{h_2}{10}$.

On note $\xi^{(3)} = (x_2^{(3)}, x_3^{(3)})$ le meilleur point à l'issue de cette étape.

La Table 3 indique les valeurs obtenues pour $\xi^{(1)}$, $\xi^{(2)}$, $\xi^{(3)}$ et les valeurs correspondantes de la variation totale. Le minimum est atteint pour $x_2 = 0.077$, et $x_3 = 0.409$, et produit $J_2 = 1.6643$.

	x_2	x_3	L_2 -norme ($\times 10^{-5}$)	TV
étape 1	0.1	0.4	5.57482	2.988
étape 2	0.08	0.41	5.75494	1.7351
étape 3	0.077	0.409	5.77881	1.6643

TAB. 3 – Méthode grilles emboîtées

Remarque: la fonction $J_2(x_2, x_3)$ admet plusieurs minimums locaux, il est donc nécessaire de faire un quadrillage assez fin dès la première étape pour éviter la convergence vers un minimum non global.

5.3 Méthode mixte

Malgré les difficultés observées à localiser un optimum global, on souhaite savoir si l'utilisation du gradient peut améliorer l'efficacité de la recherche. On considère donc une méthode mixte dont la première étape est identique à celle de la méthode précédente, mais dont les étapes suivantes de raffinement sont effectuées dans la direction opposée au gradient (*steepest descent*); voir Figure 6.

Les données relatives à la convergence sont consignées à la Table 4. On constate qu'en trois étapes, la convergence est insuffisante par cette méthode.

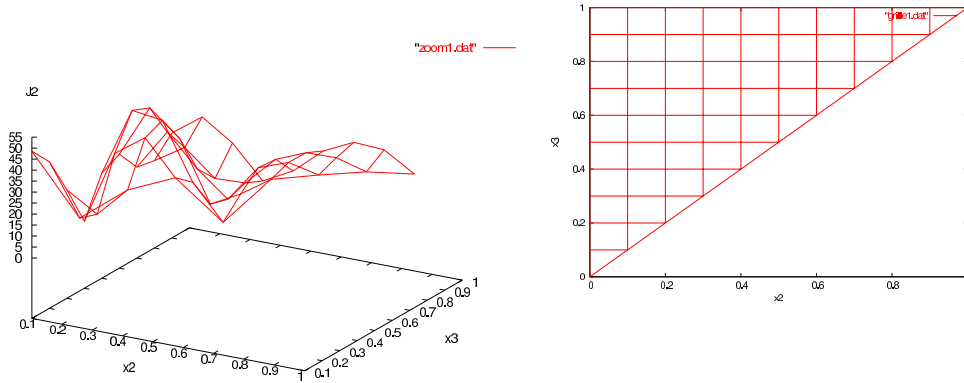
	x_2	x_3	norme- L_2 ($\times 10^{-5}$)	TV
étape 1	0.1	0.4	5.57482	2.988
étape 2	0.06	0.3667	5.756891	2.03215
étape 3	0.059	0.3669	5.68228	1.91686

TAB. 4 – Méthode mixte

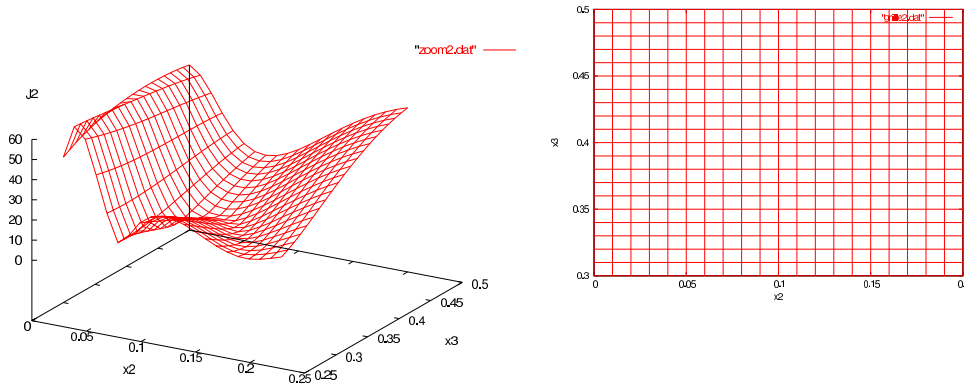
5.4 Conclusion

La méthode utilisant le gradient n'est pas assez robuste pour être démarrée sans étape préalable de localisation de l'optimum global. En conséquence, on ne profite pas du gain potentiel en efficacité que cette méthode réaliserait dans le cas d'une convergence asymptotique très poussée, car ici, il est inutile que le critère d'arrêt soit très rigoureux. On opte donc en faveur de la méthode par grilles emboîtées qui a le mérite de combiner simplicité et robustesse, sans que le coût soit excessif.

a) Première étape



b) Deuxième étape



c) Troisième étape

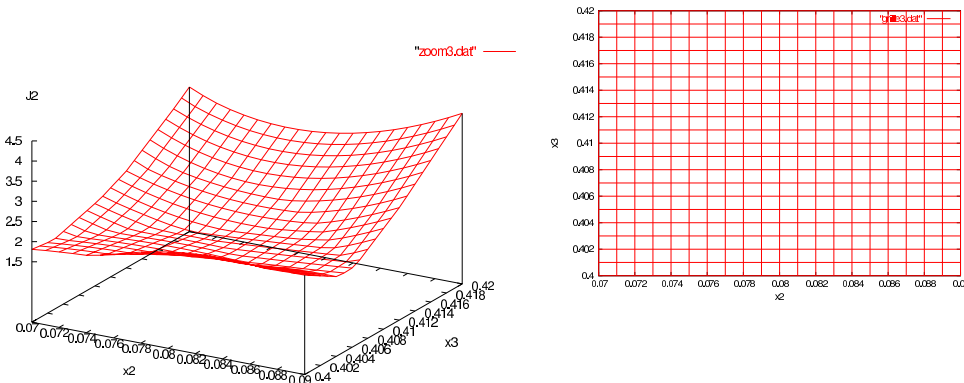


FIG. 5 – Graphes de la fonction $J_2(x_2, x_3)$ (à gauche) et grilles de discrétisation (à droite) successives

methode mixte :

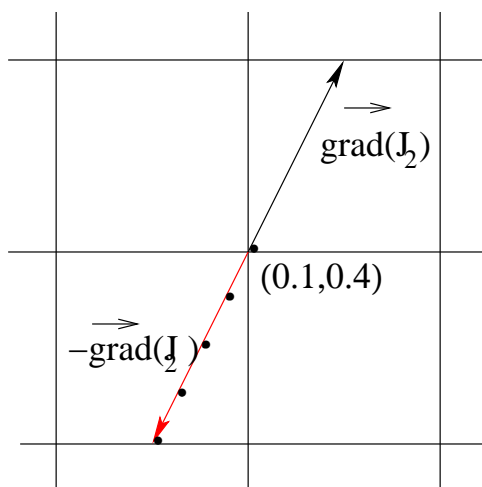


FIG. 6 – Méthode mixte: étape de descente dans la direction opposée au gradient

6 Exemple d'optimisation auto-adaptée

On souhaite tester l'efficacité de coupler une optimisation de forme avec la procédure précédente d'adaptation de la paramétrisation. Pour simplifier l'expérimentation, on traite un problème d'optimisation de forme issu du calcul des variations dont la solution est connue.

6.1 Problème test de calcul des variations

On considère un arc de courbe lisse reliant l'origine $(0,0)$ au point $(1,0)$ pouvant être représenté par une équation analytique de la forme $y = f(x)$, ou par la paramétrisation

$$\begin{cases} x = x(t) \\ y = y(t) \end{cases} \quad (59)$$

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

$$\begin{cases} x(0) = 0, x(1) = 1 \\ y(0) = y(1) = 0 \end{cases} \quad (60)$$

La longueur de l'arc p et l'aire A entre l'arc et l'axe des x sont donnés par:

$$\begin{cases} p = \int_0^1 \sqrt{x'^2(t) + y'^2(t)} dt \\ A = \int_0^1 y(t) x'(t) dt \end{cases} \quad (61)$$

Désormais, on considère la fonction $x(t)$ comme figée, et on cherche à optimiser la fonction $y(t)$ de manière à minimiser la fonctionnelle:

$$J(y) = \frac{p^2}{\mathcal{A}} \quad (62)$$

On identifie d'abord la solution exacte de ce problème. La fonctionnelle $J(y)$ est stationnaire ssi :

$$2 \frac{\delta p}{p} - \frac{\delta \mathcal{A}}{\mathcal{A}} = 0 \quad (63)$$

soit encore:

$$2 \mathcal{A} \delta p = p \delta \mathcal{A} \quad (64)$$

Or,

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

car $\delta y(0) = \delta y(1) = 0$. Après simplification, il vient:

$$\begin{cases} \delta p = \int_0^1 \frac{-y'' x'^2 + y' x' x''}{x'^2 + y'^2 \frac{3}{2}} \delta y dt \\ \delta \mathcal{A} = \int_0^1 x'(t) \delta y dt \end{cases} \quad (65)$$

La condition de stationarité s'écrit donc:

$$\forall \delta y, \quad \int_0^1 \left[2 \mathcal{A} \frac{-y'' x'^2 + y' x' x''}{x'^2 + y'^2 \frac{3}{2}} - p x' \right] \delta y dt = 0 \quad (66)$$

ce qui équivaut à l'équation différentielle suivante:

$$\forall t, \quad 2 \mathcal{A} \frac{-y'' x'^2 + y' x' x''}{(x'^2 + y'^2)^{3/2}} = p x' \quad (67)$$

soit enfin:

$$\forall t, \quad \frac{-y'' x' + y' x''}{(x'^2 + y'^2)^{3/2}} = \frac{p}{2 \mathcal{A}} (= \text{const}) \quad (68)$$

La quantité à gauche de cette équation représente l'inverse du rayon de courbure local r . L'équation équivaut donc à la condition

$$\frac{1}{r} = \frac{p}{2 \mathcal{A}} \quad (69)$$

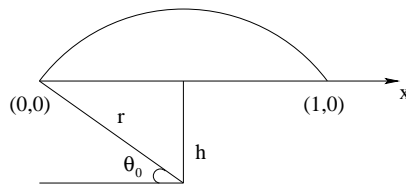


FIG. 7 – Représentation de l'arc de courbe.

qui caractérise un cercle de rayon r , dont le centre, compte tenu des conditions aux limites, a pour abscisse $x = \frac{1}{2}$; voir Figure 7. On a: $p = (\pi - 2\theta_0)r$ et $\mathcal{A} = (\pi - 2\theta_0)r^2/2 - \frac{1}{2}h \times 1$ de sorte que $h = r \sin \theta_0 = \theta_0 = 0$; par conséquent:

$$r = \frac{1}{2} \quad (70)$$

La solution du problème d'optimisation est donc fournie par le demi cercle de centre $(\frac{1}{2}, 0)$ et de rayon $\frac{1}{2}$ pour lequel:

$$J = \frac{p^2}{\mathcal{A}} = \frac{(\pi/2)^2}{\frac{1}{2} \pi (\frac{1}{2})^2} = 2 \pi \quad (71)$$

6.2 Optimisation auto-adaptative

On souhaite tester l'efficacité de la procédure d'adaptation de la paramétrisation à l'optimisation de forme correspondant au problème test précédent.

On se restreint aux paramétrisations de Bézier de degré 4 satisfaisant les contraintes géométriques suivantes:

$$x_0 = x_1 = 0, x_4 = 1, y_0 = y_4 = 0 \quad (72)$$

qui ont pour effet d'imposer les points extrêmes, ainsi qu'une tangente verticale à l'origine. Ces conditions sont réalistes pour un profil d'aile, mais pour la solution exacte du problème considéré, il serait naturel d'imposer également que la tangente à l'extrémité de l'arc soit verticale. Cependant, ce choix n'a pas été fait afin de conserver la structure de programmation des procédures, mais également pour permettre d'observer l'effet de cette non prise en compte sur la précision de la solution numérique.

On a donc au total 5 paramètres: x_2, x_3, y_1, y_2 et y_3 . On pourrait réaliser la minimisation du critère $J = p^2/\mathcal{A}$ globalement par rapport à ces cinq paramètres. Mais ce qui nous intéresse ici c'est de tester l'auto-adaptation proposée lorsqu'on distribue à $\xi = (x_2, x_3)$ le rôle d'adaptation géométrique et à $\eta = (y_1, y_2, y_3)$ celui de la minimisation de J , ersatz d'une "minimisation physique", qui dans le contexte d'un problème d'aérodynamique reviendrait à optimiser une performance dont l'évaluation dépendrait d'un calcul d'écoulement complexe.

On adopte donc un algorithme découplant ces tâches, schématisé à la Figure 8.

Après une initialisation de ξ , on effectue plusieurs itérations comprenant chacune 2 étapes:

1. Minimisation du pseudo-critère physique $J_\xi(\eta)$ par rapport à η à ξ fixé, ce qui conduit à identifier une approximation de la géométrie optimale prise comme géométrie cible γ_0 en vue de l'étape suivante.
2. Adaptation de la paramétrisation à la cible, et définition d'un nouveau vecteur ξ .

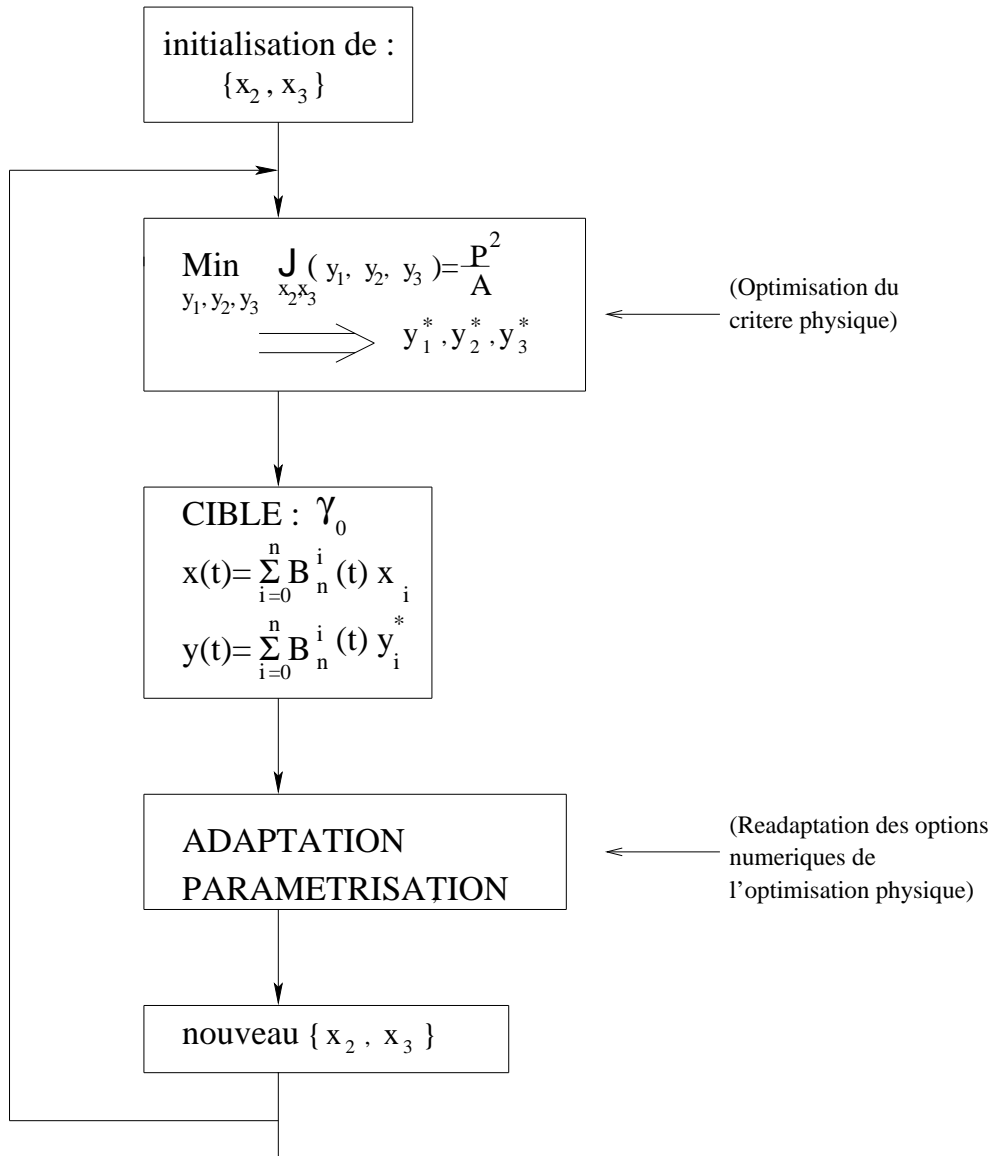


FIG. 8 – Simulation d'une optimisation couplée à une adaptation de la paramétrisation

Pour réaliser l'étape 1 d'optimisation de forme, on a discrétisé les trois composantes de η de 0 à 1 par pas de $\frac{1}{100}$, et on a identifié le meilleur des 10^6 points correspondants.

Pour réaliser l'étape 2 d'adaptation de la paramétrisation, on a appliqué la méthode des grilles emboîtées avec un pas h successivement égal à 10^{-2} , 10^{-3} et 10^{-4} . On n'a pas cherché à remédier à l'écart de précision des recherches en x et y afin de trivialisier l'expérimentation numérique.

Avant de passer aux résultats de convergence de cet algorithme, on donne quelques précisions sur le calcul numérique des paramètres \mathcal{A} et p .

Evaluation de l'aire \mathcal{A} et du périmètre p On rapelle que :

$$\begin{cases} x'(t) = \sum_{k=0}^{n-1} B_{n-1}^k(t) (x_{k+1} - x_k) \\ y(t) = \sum_{m=0}^n B_n^m(t) y_m \end{cases} \quad (73)$$

– Calcul de l'aire \mathcal{A} :

$$\mathcal{A} = \int_0^1 y(t) x'(t) dt = \sum_{m=0}^n \left(\underbrace{\sum_{k=0}^{n-1} \left(\underbrace{\int_0^1 n B_{n-1}^k(t) B_n^m(t) dt}_{\alpha_n^{k,m}} \right) (x_{k+1} - x_k)}_{\beta_n^m} \right) y_m \quad (74)$$

qui s'écrit:

$$\mathcal{A} = \sum_{m=0}^n \beta_n^m(X) y_m \quad (75)$$

où:

$$\beta_n^m(X) = \sum_{k=0}^{n-1} \alpha_n^{k,m} (x_{k+1} - x_k) \quad (76)$$

– Calcul des coefficients $\alpha_n^{k,m}$:

Pour $0 \leq k \leq n-1$ et $0 \leq m \leq n$:

$$\begin{aligned} \alpha_n^{k,m} &= \int_0^1 n B_{n-1}^k(t) B_n^m(t) dt \\ &= n C_{n-1}^k C_n^m \int_0^1 t^{k+m} (1-t)^{2n-1-(k+m)} dt \\ &= n C_{n-1}^k C_n^m \frac{\Gamma(k+m+1) \Gamma(2n-(k+m))}{\Gamma(2n+1)} \end{aligned} \quad (77)$$

où Γ est la fonction eulérienne ($\Gamma(p) = (p-1)!$ pour tout p entier strictement positif).

Après simplification, il vient:

$$\alpha_n^{k,m} = \frac{C_{n-1}^k C_n^m}{2 C_{2n-1}^{k+m}} \quad (78)$$

– Calcul du périmètre p :

$$p = \int_0^1 f(t) dt \quad (79)$$

où l'on a posé: $f(t) = \sqrt{x'^2(t) + y'^2(t)}$.

L'intégrale est approchée par la règle du trapèze corrigée [35] avec un pas constant $\frac{1}{M}$:

$$\int_0^1 f(t) dt \simeq \frac{1}{M} \sum_{i=0}^{M-1} \frac{1}{2} \left[f\left(\frac{i+1}{M}\right) + f\left(\frac{i}{M}\right) \right] + \frac{1}{12M^2} [f'(0) - f'(1)] \quad (80)$$

où

$$f'(t) = \frac{x'(t)x''(t) + y'(t)y''(t)}{\left(x'(t)^2 + y'(t)^2\right)^{\frac{3}{2}}} \quad (81)$$

Pour la paramétrisation considérée, on a:

$$\begin{cases} x'(0) = n(x_1 - x_0) = 0 \\ y'(0) = n(y_1 - y_0) = 4y_1 \end{cases} \quad \begin{cases} x'(1) = n(x_n - x_{n-1}) = 4(1 - x_3) \\ y'(1) = n(y_n - y_{n-1}) = -4y_3 \end{cases} \quad (82)$$

ainsi que:

$$\begin{cases} x''(0) = n(n-1)(x_2 - 2x_1 + x_0) = 12x_2 \\ y''(0) = n(n-1)(y_2 - 2y_1 + y_0) = 12(y_2 - 2y_1) \end{cases} \quad (83)$$

et

$$\begin{cases} x''(1) = n(n-1)(x_n - 2x_{n-1} + x_{n-2}) = 12(1 - 2x_3 + x_2) \\ y''(1) = n(n-1)(y_n - 2y_{n-1} + y_{n-2}) = 12(y_2 - 2y_3) \end{cases} \quad (84)$$

ce qui permet de calculer $f'(0)$ et $f'(1)$.

Résultats de convergence. On a réalisé deux expériences correspondants à des initialisations différentes données à la Table 5.

	x_2	x_3	remarque
1 ^{ère} Expe.	1/3	2/3	support uniforme
2 ^e Expe.	1/2	1	support symétrique

TAB. 5 – Initialisations correspondant aux deux expériences de couplage optimisation-adaptation

Les quadratures numériques ont été réalisées au 4^e ordre.

On fournit aux Figures 9, 10 et 11 les courbes de convergence, au cours des itérations de couplage optimisation-adaptation, du vecteur $\xi = (x_2, x_3)$, de la fonctionnelle J et du vecteur $\eta = (y_1, y_2, y_3)$ pour les deux expériences considérées. Les valeurs correspondantes sont également consignées à la Table 6. Enfin, on illustre à la Figure 12 diverses formes obtenues:

- forme circulaire optimale du problème continu (trait rouge);
- forme optimisée numériquement à partir du support non optimal de l'expérience 1, $x_2 = 1/3$, $x_3 = 2/3$ (trait vert);
- forme optimisée numériquement par 4 itérations de couplage optimisation-adaptation (trait bleu);
- forme optimisée numériquement à partir du support symétrique, $x_2 = 1/2$, $x_3 = 1$ (croix mauves).

(a) 1^{ière} expérience

itération	x_2	x_3	y_1^*	y_2^*	y_3^*	J^*
0	0.333333	0.666667	0.49	0.35	0.86	6.35086
1	0.6223	0.7812	0.61	0.46	0.66	6.3266
2	0.6001	0.844	0.57	0.54	0.59	6.30749
3	0.5999	0.8581	0.56	0.56	0.57	6.30394
4	0.6079	0.8598	0.57	0.56	0.57	6.30378
5	0.5686	0.8678	0.54	0.57	0.58	6.30079
6	0.5889	0.8678	0.55	0.57	0.57	6.30132
7	0.5978	0.8666	0.55	0.58	0.56	6.30187
8	0.6196	0.8581	0.57	0.56	0.56	6.30455
9	0.6061	0.8599	0.56	0.56	0.57	6.3037
10	0.6126	0.8618	0.56	0.57	0.56	6.30343
11	0.6139	0.8588	0.57	0.56	0.57	6.3042
12	0.6101	0.8624	0.56	0.57	0.56	6.30319
13	0.6147	0.859	0.57	0.56	0.57	6.30418
14	0.6101	0.8624	0.56	0.57	0.56	6.30319
15	0.6147	0.859	0.57	0.56	0.57	6.30418
16	0.6101	0.8624	0.56	0.57	0.56	6.30319
17	0.6147	0.859	0.57	0.56	0.57	6.30418
18	0.6101	0.8624	0.56	0.57	0.56	6.30319
19	0.6147	0.859	0.57	0.56	0.57	6.30418
20	0.6101	0.8624	0.56	0.57	0.56	6.30319

(b) 2^e expérience

itération	x_2	x_3	y_1^*	y_2^*	y_3^*	J^*
0	0.5	1	0.44	0.75	0.44	6.2832
1	0.6381	0.8977	0.56	0.61	0.5	6.29596
2	0.6128	0.871	0.56	0.58	0.55	6.30123
3	0.6201	0.8637	0.57	0.56	0.56	6.30319
4	0.6102	0.8661	0.56	0.57	0.56	6.30231
5	0.6131	0.8629	0.56	0.57	0.56	6.30317
6	0.6159	0.8597	0.57	0.56	0.56	6.30405
7	0.6041	0.8617	0.56	0.56	0.57	6.30321
8	0.6107	0.8636	0.56	0.57	0.56	6.30292
9	0.6154	0.8602	0.57	0.56	0.56	6.30392
10	0.6055	0.8626	0.56	0.57	0.56	6.30302
11	0.6104	0.8592	0.57	0.56	0.57	6.30399
12	0.6101	0.8635	0.56	0.57	0.56	6.30292
13	0.6148	0.8601	0.57	0.56	0.57	6.30392
14	0.5777	0.8679	0.54	0.58	0.57	6.30101
15	0.6221	0.8591	0.57	0.56	0.56	6.30436
16	0.6087	0.8609	0.56	0.57	0.56	6.30353
17	0.6099	0.8579	0.57	0.56	0.57	6.3043
18	0.6019	0.8617	0.56	0.56	0.57	6.30313
19	0.6085	0.8636	0.56	0.57	0.56	6.30285
20	0.6132	0.8602	0.57	0.56	0.57	6.30384

TAB. 6 – Résultats de convergence (1^{ière} et 2^e expériences)

On constate dans les deux expériences que le couplage converge en 4 ou 5 itérations et reste stable. En toute rigueur, on atteint un cycle limite lié à la précision arithmétique utilisée pour la recherche des points optimaux en x et y .

Les valeurs limites de x_2 et x_3 ne sont pas égales à $1/2$ et 1 comme on aurait espéré. Ceci n'est pas surprenant. Le couplage considéré correspond à un jeu de Stackelberg [1] dont l'optimisation est le processus meneur (*leader*), et l'adaptation le processus suiveur (*follower*). Le critère optimisé par le suiveur influence faiblement (0.3 % sur la valeur optimisée du critère) mais visiblement le point de convergence. En effet, la solution optimale (vis-à-vis du critère J) du problème continu n'a aucune raison de satisfaire le critère de régularité (minimisation du critère J_2) que nous imposons par adaptation.

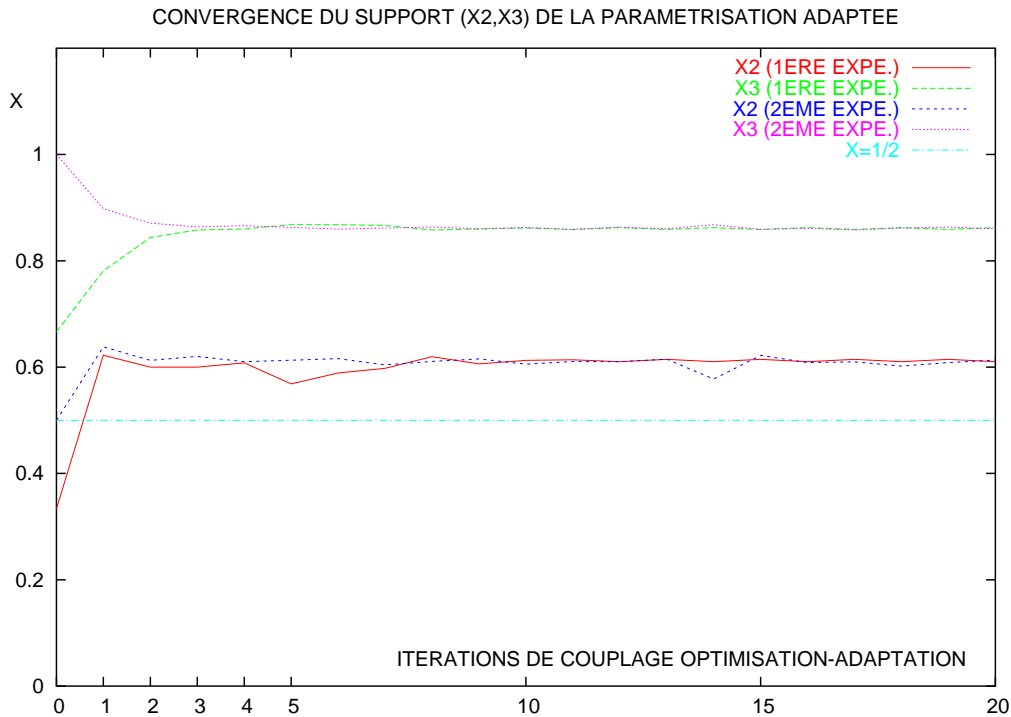


FIG. 9 – Convergence du vecteur $\xi = (x_2, x_3)$ représentant le support de la paramétrisation de Bézier de degré 4 ($x_0 = x_1 = 0$; $x_4 = 1$)

Les paramètres de forme y_1^* , y_2^* et y_3^* ont ici la même limite, et dans les deux expériences, cette limite est la même, comme attendu.

En examinant les formes auxquelles on aboutit, on constate qu'une paramétrisation de Bézier de degré 4 est suffisamment précise pour ce problème puisque la comparaison de la solution exacte (trait rouge) et de la solution optimisée sur le support de paramétrisation symétrique (croix mauves) est excellente. Ayant volontairement omis de prendre en compte la tangente verticale en $x = 1$, on observe clairement les erreurs d'approximation qui sont les plus grandes dans le voisinage de ce point. Avec le support de paramétrisation uniforme ($x_2 = 1/3$; $x_3 = 2/3$), l'approximation est assez grossière (1.1 % sur J); avec l'adaptation l'approximation s'améliore nettement (0.3 %), mais reste inférieure à l'optimum numérique global atteint avec le support de paramétrisation symétrique.

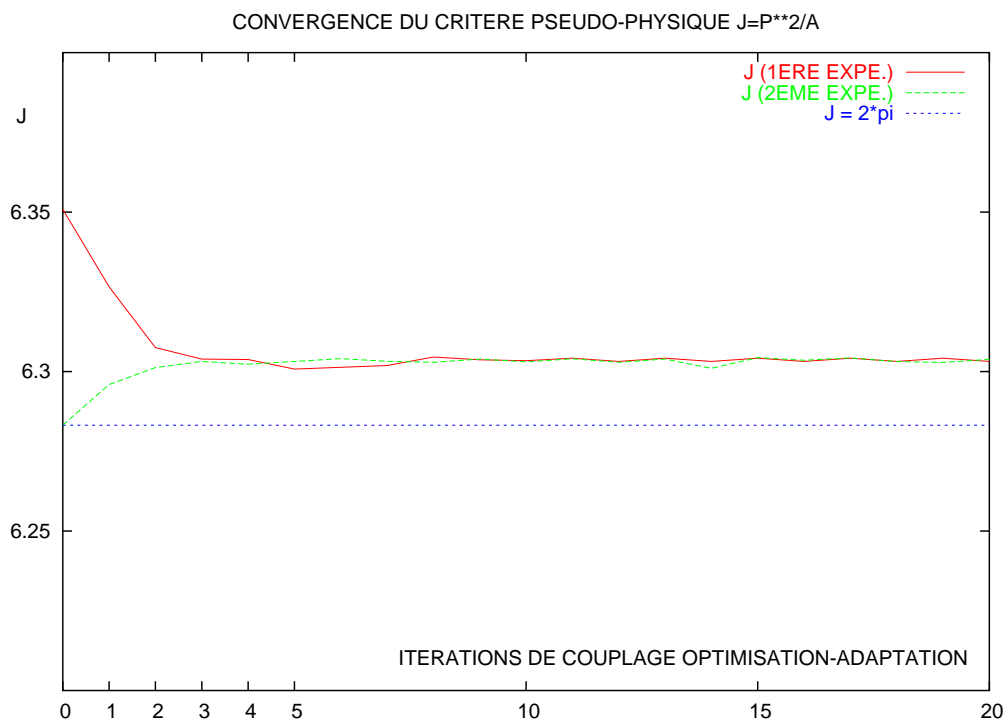


FIG. 10 – Convergence de la fonctionnelle J

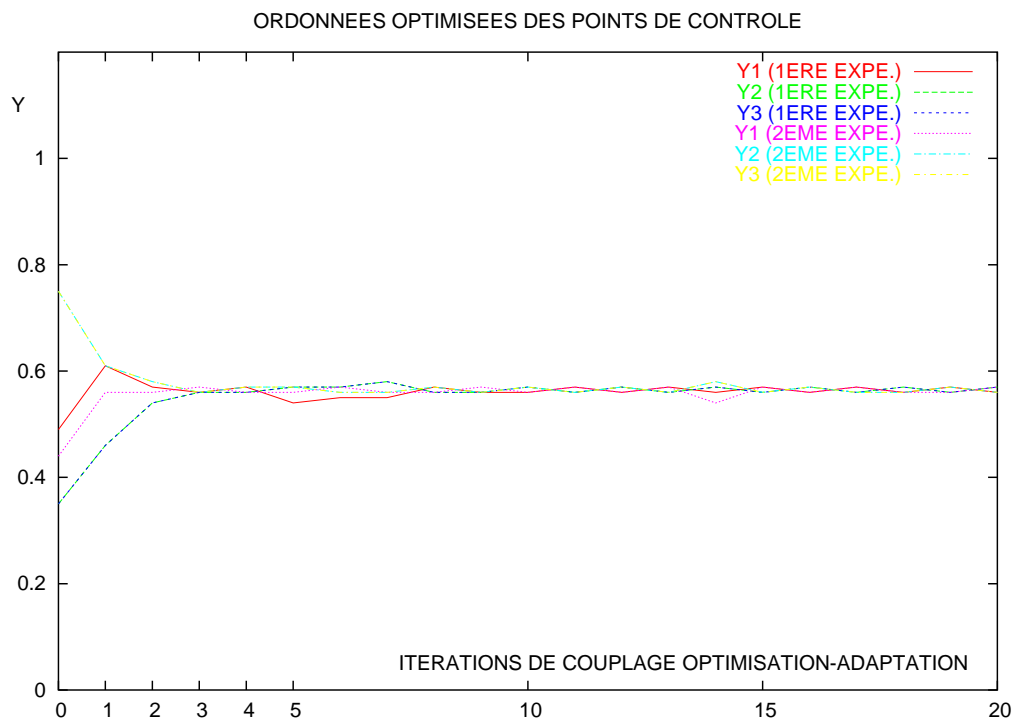


FIG. 11 – Convergence du vecteur $\eta = (y_1, y_2, y_3)$ des paramètres de forme de la paramétrisation de Bézier de degré 4 ($y_0 = y_4 = 0$)

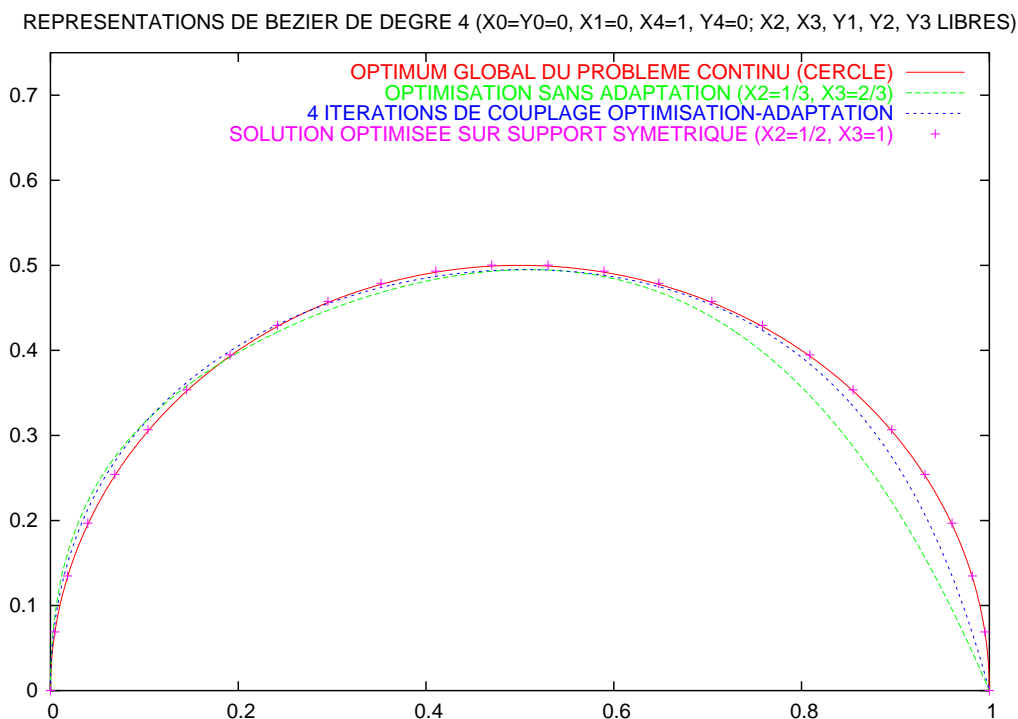


FIG. 12 – Formes optimale et optimisées par voie numérique

En conclusion de ce cas-test, le couplage entre l’optimisation pseudo-physique (minimisation de $J_\xi(\eta)$ à ξ fixé) avec l’adaptation de la paramétrisation (optimisation de ξ à cible fixée) converge et conduit à un équilibre de Stackelberg correspondant à une forme sous-optimale, mais nettement plus précise que le résultat sans adaptation.

7 Conclusion

Dans les rapports [22] et [16], on avait montré (en discret) qu’une optimisation de forme en aérodynamique gagnait significativement en conditionnement par adaptation à intervalles réguliers de la paramétrisation de Bézier sur laquelle s’appuie la représentation de la forme.

A partir de la connaissance d’une forme optimale approchée, on considère l’ensemble des paramétrisations de même degré qui approchent cette forme au sens des moindres carrés. On choisit celle pour laquelle le polygone de contrôle est le plus régulier (minimisation de la variation totale).

Dans ce rapport, on a fourni tous les détails de cette formulation, très voisine de celle de nos travaux initiaux mais reposant sur un moins grand nombre de quadratures numériques dont la plupart sont réalisées au 4^e ordre.

On a également proposé un cas-test d’optimisation de forme issu du calcul des variations. Cet ersatz d’“optimisation physique” nous a permis à faible coût de mettre en évidence la convergence du couplage entre l’optimisation pseudo-physique et l’auto-adaptation de la paramétrisation.

8 Perspectives

Les résultats encourageants de cette étude nous conduisent à chercher à la généraliser et à rendre les procédures qui en sont issues plus robustes et efficaces. On envisage donc de poursuivre ce travail par des développements techniques et des expériences numériques, dont on donne ici une liste provisoire.

8.1 Implantation de procédures d'optimisation robustes et efficaces

Pour augmenter l'efficacité algorithmique de la procédure d'adaptation de la paramétrisation, il conviendra, pour la recherche du vecteur ξ optimal (le vecteur η étant fixé), de remplacer l'algorithme des grilles emboîtées par un meilleur optimiseur. On pense à un algorithme hybride combinant une phase de recherche robuste, vraisemblablement par la méthode du simplex, à une phase de convergence efficace, sans doute par la méthode SQP (*Sequential Quadratic Programming*).

8.2 Précision des intégrations numériques

Les intégrales définies calculées numériquement ont la forme suivante

$$I_{\alpha,\beta} = \int_0^1 \bar{y}_T(t, X) t^\alpha (1-t)^\beta dt = \int_0^1 \Phi(t) w_{\alpha,\beta}(t) dt \quad (85)$$

où α et β sont des entiers naturels, $\Phi(t) = \bar{y}_T(t, X)$ est une fonction supposée régulière (grâce à l'adaptation) de sorte que $\Phi(0) = \Phi(1) = 0$, et

$$w_{\alpha,\beta}(t) = t^\alpha (1-t)^\beta \quad (86)$$

est un polynôme à valeurs positives sur l'intervalle d'intégration.

Cette forme particulière incite à envisager de remplacer la formule du trapèze corrigée, qui est du 4^e ordre, mais dont la précision effective dépend du nombre N d'évaluations de la fonction $\Phi(t)$ par le biais du pas $h = \frac{1}{N}$, par une quadrature gaussienne dont l'ordre de précision formelle est $(2N)$ [35], ce qui permettrait de réduire N et le coût du calcul à précision effective donnée.

Comme les valeurs de l'intégrande sont connues aux limites, ici égales à 0, on utilisera de préférence, une formule de Gauss-Lobatto (voir par exemple [34]) pour une efficacité maximale. En retenant précisément la fonction $w_{\alpha,\beta}(t)$ comme fonction poids, conduira à considérer une formule basée sur les polynômes de Jacobi (voir par exemple [33]).

8.3 Transposition des procédures aux *B-splines*

Les travaux récents de MM. B. Mourrain et M. Celikbas [27] du Projet Galaad fournissent des résultats comparables aux nôtres pour des paramétrisations de type *B-splines* qui sont plus générales et plus locales. Il est donc envisagé d'uniformiser nos approches.

8.4 Examen de cas tests voisins

En modifiant la définition des grandeurs p ou \mathcal{A} , on pense pouvoir construire des cas tests d'optimisation de forme dont la solution exacte est connue par le calcul des variations (Equation d'Euler) et présente un type différent de difficulté ou de simplification par rapport au cas-test étudié à la Section 6.

En particulier, en maintenant dans la solution exacte du problème continu la dérivée infinie en $x = 0$ mais en la supprimant en $x = 1$, on obtiendrait un problème plus représentatif encore de l'optimisation d'un profil d'aile.

En outre, en changeant la métrique, on pourra étudier l'influence du facteur de forme (épaisseur/longueur) sur le conditionnement du couplage optimisation-adaptation de la méthode numérique.

8.5 Généralisation au 2D

En 1D, le critère discret de variation totale est le discrétisé de la variation totale en continu; en effet, si $y(x)$ représente un interpolant régulier des points de contrôle, on a:

$$\begin{aligned}
 TV(\{y_k\}) &= \sum_{k=1}^n |y_k - y_{k-1}| \\
 &= \sum_{k=1}^n |y(x_k) - y(x_{k-1})| \\
 &= \sum_{k=1}^n |y'(x_{k-1} + \theta_k \Delta x_k) \Delta x_k + \dots| \quad (\Delta x_k = x_k - x_{k-1}) \\
 &= \sum_{k=1}^n |y'(x_{k-1} + \theta_k \Delta x_k)| \Delta x_k + \dots \quad (\text{car } \Delta x_k > 0) \\
 &= \int_0^1 |y'(x)| dx + \dots \tag{87}
 \end{aligned}$$

Une généralisation naturelle sera donc de minimiser une discrétisation (éventuellement non structurée) de l'intégrale double $\iint \|\nabla z\| dS$, où $z(x,y)$ représente une fonction régularisée donnant la cote du point de contrôle au point (x,y) ; cette fonction est connue en quelques points seulement.

8.6 Test d'autres procédures d'adaptation

La question reste ouverte quant à l'optimalité du critère de variation totale. Il sera intéressant d'étudier l'influence sur la robustesse des procédures autant que sur l'efficacité du couplage optimisation-adaptation du remplacement de ce critère, qui est différentiable seulement par morceaux, par un critère différentiable partout tel que $\sum_k (y_k - y_{k-1})^2 \omega_k$ où ω_k est un poids approprié (vraisemblablement $\omega_k = 1/\Delta x_k$ serait pertinent).

Peut-on définir les poids ω_k pour tenir compte avantagement dans l'adaptation de grandeurs locales telles que la courbure?

Est-ce qu'un réseau de neurones artificiel peut guider le choix du critère?

Une autre question portant sur le couplage est la suivante: dans le jeu de Stackelberg peut-on augmenter l'importance du meneur par rapport au suiveur, c'est-à-dire rendre l'optimisation physique davantage "maître du jeu" par rapport à l'adaptation? Peut-on le faire progressivement de sorte qu'à convergence seule l'optimisation physique compte?

8.7 Adaptation et degré de la paramétrisation

On a bien vu avec l'exemple de la Section 6 que l'adaptation permettait de rattraper une partie seulement de la perte de précision sur la forme optimisée due à la réduction du nombre de paramètres effectifs réalisée pour réduire le coût de l'optimisation. Il conviendra d'approfondir le lien entre adaptation et ordre de précision.

Remerciements

Cette étude a été partiellement soutenue par le Réseau de recherche et d'innovation technologique "Recherche Aéronautique sur le Supersonique" (Optimisation multicritère pour le supersonique – Méthodes, Stratégies et Applications), Dassault Aviation (*Self-Adaptive Parameterization for Shape-Optimum Design in Aerodynamics and Multi-Disciplinary Applications*), et l'Action de Coopération Locale "Formes Hiérarchiques pour l'Optimisation Multiéchelle". On remercie particulièrement MM. B. Mourrain et M. Celikbas du Projet Galaad de leurs conseils scientifiques et techniques.

9 Annexe 1: Programmation.

Le code a été écrit dans le langage de programmation C++ en s'appuyant, pour les manipulations d'algèbre linéaire, sur la bibliothèque SYNAPS, *S*ymbolic *N*umeric *A*pplication*S* [18] (voir: <http://www-sop.inria.fr/galaad/logiciels/synaps/>), développée par le Projet Galaad et ses collaborateurs.

10 Annexe 2: Validation de l'intégration numérique intervenant dans le calcul du périmètre

L'estimation numérique du coefficient d'Aitken (noté q) [35] permet de vérifier *a posteriori* l'ordre de précision d'une quadrature numérique. Afin de valider nos procédures, on a appliqué cette technique à l'approximation p_h de l'intégrale $p = \int_0^1 \Phi(t) dt$ ($\Phi(t) = \sqrt{x'^2(t) + y'^2(t)}$), dans le cas d'une quadrature par la méthode du trapèze, qui est d'ordre 2, et la méthode du trapèze corrigée, qui est d'ordre 4; cette dernière, lorsque le pas d'espace est constant, ne diffère de la formule standard que par l'ajout d'un terme de correction proportionnel à la variation de la dérivée première $\Phi'(1) - \Phi'(0)$ [35].

Rappel : si p_h est une approximation d'ordre α de l'intégrale p ,

$$p_h = p + C h^\alpha + o(h^\alpha) \quad (88)$$

Par conséquent, si la quadrature numérique est répétée avec des pas deux fois moindres, on aboutit à:

$$p_{h/2} = p + C \left(\frac{h}{2}\right)^\alpha + o(h^\alpha) \quad (89)$$

et

$$p_{h/4} = p + C \left(\frac{h}{4}\right)^\alpha + o(h^\alpha) \quad (90)$$

Par conséquent:

$$q = \frac{p_h - p_{h/2}}{p_{h/2} - p_{h/4}} = \frac{1 - \frac{1}{2^\alpha} + o(1)}{\frac{1}{2^\alpha} - \frac{1}{4^\alpha} + o(1)} = 2^\alpha + o(1) \quad (91)$$

En particulier, on doit vérifier *a posteriori* que $q \simeq 4$ pour une approximation d'ordre 2, et $q \simeq 16$ pour une approximation d'ordre 4.

Afin d'effectuer ce test, on a considéré la courbe de Bézier de degré 4 dont les points de contrôle ont les coordonnées précisées à la Table 7. Cette courbe approche l'arc de cercle considéré à la Section 6.

x_c	0	0	0.60785	0.864	1
y_c	0	0.56	0.57	0.56	0

TAB. 7 – Points de contrôle définissant les fonctions $x(t)$ et $y(t)$

Avec $h = \frac{1}{100}$, on a calculé p_h , $p_{h/2}$ et $p_{h/4}$ et le paramètre q par les deux méthodes. Les résultats sont consignés à la Table 8; ils confirment très bien le résultat théorique concernant l'ordre de précision de ces règles d'intégration.

En "extrapolant à la limite" [35] les deux derniers résultats (censément plus précis) de la méthode du trapèze standard, on obtient l'estimation suivante, précise au 3^e ordre:

$$p \simeq 1.5577 + \frac{4 \times 0.00006209 - 0.00008212}{3} \simeq 1.5577\ 5541 \quad (92)$$

ce qui est sensiblement plus proche du meilleur résultat précis au 4^e ordre, à savoir 1.5577 5545 44, dont la très grande précision ne peut être augmentée par la seule extrapolation des éléments de la table, par manque de décimales connues.

formule de quadrature	p_h	$p_{h/2}$	$p_{h/4}$	q
trapèze	1.55786321	1.55778212	1.55776209	4.0472376
trapèze corrigée	1.5577554539	1.5577554543	1.5577554544	16.3297555548

TAB. 8 – Paramètre d'Aitken pour l'estimation de l'ordre de précision de la règle d'intégration du trapèze (sans et avec terme de correction)

Références

- [1] P. Loridan and J. Morgan. A Theoretical Approximation Scheme for Stackelberg Games. *Optimization Theory and Applications*, 61(1):95–110, 1989.
- [2] N. Marco. Optimisation de formes aérodynamiques 2d et 3d par une méthode multi-niveau en maillages non-structurés. Thèse de doctorat, Université de Nice-Sophia Antipolis, 1995.
- [3] N. Marco and A. Dervieux. Multilevel parametrization for aerodynamical optimization of 3d shapes. *Finite Elements in Analysis and Design*, 26:259–277, 1997.
- [4] N. Marco, S. Lanteri, J.-A. Désidéri, B. Mantel, and J. Périaux. Parallel genetic algorithms applied to optimum shape design in aeronautics. In C. Lengauer, M. Griehl, and S. Gorlatch, editors, *Proceedings of the Third International Euro-Par Conference (Euro-Par '97)*, *Lecture Notes in Computer Science*, 1300, pages 856–863, Passau, 1997.
- [5] B. Mantel, J. Périaux, M. Sefrioui, B. Stoufflet, J.-A. Désidéri, S. Lanteri, and N. Marco. Evolutionary computational methods for complex design in aerodynamics. In *Proceedings of the AIAA Aerospace Sciences Conference*, Reno, Nevada (USA), 1998. AIAA 98-0222.
- [6] N. Marco-Blaszka and J.-A. Désidéri. Numerical solution of optimisation test-cases by genetic algorithms. Rapport de recherche 3622, INRIA, Février 1999.
- [7] J.-A. Désidéri, L. Fournier, S. Lanteri, N. Marco, B. Mantel, J. Périaux, and J.F. Wang. Parallel multigrid solution and optimization in compressible flow simulation and design. In D. Keyes, A. Ecer, N. Satofuka, P. Fox, and J. Périaux, editors, *Parallel Computational Fluid Dynamics, Towards Teraflops, Optimization and Novel Formulations*. North Holland, 2000. Proc. of PARCFD'99, Williamsburg, VA 24-27 April, 1999.
- [8] C. Held. Optimisation de formes en aérodynamique sans gradient par résolution simultanée. Rapport de Recherche 3864, INRIA, 2000.
- [9] N. Marco, S. Lanteri, J.-A. Désidéri, B. Mantel, and J. Périaux. A parallelized genetic algorithm for a 2-D shape optimum design problem. *Surveys on Mathematics for Industry*, 9:207–221, 2000.
- [10] M. Renversade. Optimisation d'un dispositif hyper-sustentateur par algorithmes génétiques. Rapport de Recherche 4029, INRIA, 2000.
- [11] G. Vigo. Méthodes de décomposition orthogonale aux valeurs propres appliquées aux écoulements instationnaires compressibles complexes. Thèse de doctorat, Université de Paris IX Dauphine, 2000.
- [12] A. Dervieux, N. Marco, C. Held, and B. Koobus. Hierarchical Principles and Preconditioning for Optimum Design and Identification. In J. Périaux, P. Joly, O. Pironneau, and E. Onate, editors, *Innovative Tools for Scientific Computation in Aeronautical Engineering*, *Handbooks on Theory and Engineering Applications of Computational Methods*, pages 272–280, Paris, June 2001. CIMNE, Barcelona.
- [13] K. C. Giannakoglou, A.P. Giotis, and M.K. Karakasis. Low-cost genetic optimization based on inexact pre-evaluations and the sensitivity analysis of design parameters. *Inverse Problems in Engineering*, 9:389–412, 2001.
- [14] H. Lièvre, J.-A. Désidéri, and A. Habbal. Optimisation numérique de profils d'aile par algorithmes génétiques et jeux de nash. Technical Report 4275, INRIA, 2001.
- [15] Z.L. Tang, J.-A. Désidéri, and J. Périaux. Multi-objective shape-design optimization and inverse problems using control theory and nash games. In K. Morgan and P. Weatherill, editors, *ECCOMAS CFD2001 Conference*, Swansea, South Wales, 4-7 September 2001.

- [16] A. Clarich and J.-A. Désidéri. Self-adaptive parameterisation for aerodynamic optimum-shape design. Rapport de Recherche 4428, INRIA, March 2002.
- [17] J.-A. Désidéri. Hierarchical optimum-shape algorithms using embedded Bézier parameterizations. In E. Heikkola and O. Pironneau, editors, *Finnish-French Seminar on Scientific Computing*, Jyväskylä, Finland, 2002. CIMNE, Barcelona, Spain. A scientific event honoring Jacques Périaux's 60th birthday.
- [18] G. Dos Reis, B. Mourrain, R. Rouillier, and Ph. Trébuchet. An environment for symbolic and numeric computation. In *Proc. of the International Conference on Mathematical Software*, World Scientific, pages 239–249, 2002.
- [19] K. C. Giannakoglou. Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *International Review Journal Progress in Aerospace Sciences*, 38:43–76, 2002.
- [20] M. Karakasis and J.-A. Désidéri. Model reduction and adaption of optimum-shape design in aerodynamics by neural networks. Rapport de Recherche 4503, INRIA, July 2002.
- [21] J. Morice, J.-A. Désidéri, and J.P. Zolésio. Fonctions géométriques principales et paramétrisation de forme adaptée à l'optimisation en aérodynamique. Rapport de Recherche 4643, INRIA, Octobre 2002.
- [22] Z.L. Tang and J.-A. Désidéri. Towards self-adaptive parameterization of bézier curves for airfoil aerodynamic design. Rapport de Recherche 4572, INRIA, September 2002.
- [23] Z.L. Tang, J.-A. Désidéri, and J. Périaux. Distributed optimization using virtual and real game strategies for aerodynamic design. Rapport de Recherche 4543, INRIA, September 2002.
- [24] M. Vazquez, A. Dervieux, and B. Koobus. Application of a multilevel method to the optimisation of a supersonic aircraft. Rapport de Recherche 4520, INRIA, 2002.
- [25] J. Wang and J. Périaux. Search space decomposition of nash/stackelberg games using gas for multi-point design optimization in aerodynamics. In N. Debit, M. Garbey, R. Hoppe, D. Keyes, U. Kuznetsov, and J. Périaux, editors, *Domain Decomposition Methods in Sciences and Engineering*, Theory and Engineering Applications of Computational Methods, Barcelona, 2002. CIMNE.
- [26] J. Wang, J. Périaux, and M. Sefrioui. Parallel evolutionary algorithms for optimization problems in aerospace engineering. *J. of Comput. and Appl. Math.*, 149:155–169, 2002.
- [27] M. Celikbas. B-splines pour l'optimisation géométrique. Rapport de recherche, INRIA, 2003. à paraître.
- [28] J.-A. Désidéri, J. Périaux, M. Sefrioui, K. Srinivas, J. Wang, and E. Whitney. Parallel hierarchical evolutionary algorithms and game strategies for solving cfd optimization problems in aeronautics. In *ICCFD02 Conference Proc.* Springer Verlag, 2003. to appear.
- [29] B. Galvan, D. Greiner, J. Périaux, M. Sefrioui, and G. Winter. Parallel evolutionary computation for solving complex cfd optimization problems: a review and some nozzle applications. In *Parallel CFD02 Conference Proc.*, Nara, Japan, 2003. Elsevier. to appear.
- [30] Z.L. Tang, J.-A. Désidéri, and J. Périaux. Distributed optimization using virtual and real game strategies for aerodynamic multi-objective design. In *West East High Speed Flow Fields – Aerospace applications from high subsonic to hypersonic regime*, 2003.
- [31] J. Wang, J. Périaux, and P. Thomas. Parallel hierarchical nash gas for lift optimization problems. In K. Giannakoglou, Tahalis, J. Périaux, K. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design and Control in Industry*, Theory and Applications of

- Computational Methods, Barcelona, 2003. CIMNE. EUROGEN01 Conf. Proc., Athens, 2002.
- [32] E. Whitney, L. Gonzalez, K. Srinivas, and J. Périaux. Multi-criteria aerodynamic shapes design problems in cfd using a modern evolutionary algorithm on distributed computers. In *ICCFD02 Conference Proc.* Springer Verlag, 2003. to appear.
- [33] Milton Abramovitz and Irene A. Stegun, editors. *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*. United States Department of Commerce, National Bureau of Standards, Applied Mathematics Series – 55. U.S. Government Printing Office, Washington, D.C., Tenth printing edition, December 1972.
- [34] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods in Fluid Dynamics*. Springer Series in Computational Physics. Springer-Verlag, New York Heidelberg, 1988. Subsection 2.2.3.
- [35] S.D. Conte and C. de Boor. *Elementary Numerical Analysis, an algorithmic approach*. McGraw-Hill Book Company, New York, second edition, 1972. (Règle corrigée du trapèze: p. 293; coefficient d’Aitken: pp. 307-315).
- [36] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design – A practical guide*, W. Rheinboldt and D. Siewiorek Eds. Computer Science and Scientific Computing. Academic Press, Boston, 2nd edition, 1990.
- [37] J. Périaux, G. Bugeđa, P. K. Chaviaropoulos, K. Giannakoglou, S. Lantéri, and B. Mantel, editors. *Optimum Aerodynamic Design & Parallel Navier-Stokes Computations, ECARP European Computational Aerodynamics Research Project*, volume 61 of *Notes on Numerical Fluid Mechanics*. Vieweg, Braunschweig/Wiesbaden, Germany, 1998.



Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399