

MODÈLES DE SUBSTITUTION POUR L'OPTIMISATION GLOBALE DE FORME EN AÉRODYNAMIQUE ET MÉTHODE LOCALE SANS PARAMÉTRISATION

Manuel Bompard

▶ **To cite this version:**

Manuel Bompard. MODÈLES DE SUBSTITUTION POUR L'OPTIMISATION GLOBALE DE FORME EN AÉRODYNAMIQUE ET MÉTHODE LOCALE SANS PARAMÉTRISATION. Analyse numérique [math.NA]. Université Nice Sophia Antipolis, 2011. Français. NNT : . tel-00771799

HAL Id: tel-00771799

https://theses.hal.science/tel-00771799

Submitted on 9 Jan 2013

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 - UFR Sciences
Ecole doctorale de Sciences Fondamentales et Appliquées

THÈSE

pour obtenir le titre de
Docteur en Sciences
de l'UNIVERSITÉ de Nice-Sophia Antipolis :
Spécialité : Mathématiques Appliquées

présentée et soutenue par
Manuel BOMPARD

MODÈLES DE SUBSTITUTION POUR L'OPTIMISATION GLOBALE DE FORME EN AÉRODYNAMIQUE ET MÉTHODE LOCALE SANS PARAMÉTRISATION

Thèse dirigée par **Jean-Antoine DÉSIDÉRI** et **Jacques PETER**
soutenue le 6 décembre 2011

JURY :
M. Gilbert ROGÉ
Ingénieur de recherche, DASSAULT Aviation, Rapporteur
M. Patrick SIARRY
Professeur, Université Paris-Est Créteil Val-de-Marne, Rapporteur
Mme Paola CINNELLA
Professeur, Arts et Métiers ParisTech, Présidente
Mme Ingrid LEPOT
Responsable du groupe Méthodes Numériques et Optimisation, CENAERO, Examinatrice
M. Jacques PETER
Ingénieur de recherche, ONERA, Examinateur
M. Jean-Antoine DÉSIDÉRI
Directeur de recherche, INRIA, Examinateur

Remerciements

Ce travail de thèse a été effectué au sein du Département de Simulation Numérique des écoulements et Aéroacoustique de l'ONERA à Châtillon. Je tiens donc à remercier tout d'abord Monsieur Jean-Marie Le Gouez, directeur du DSNA, et Monsieur Vincent Couaillier, responsable de l'unité NUMF, de m'avoir accueilli dans leur équipe.

Je remercie Monsieur Jean-Antoine Désidéri, Directeur de recherche à l'INRIA, d'avoir accepté de diriger cette thèse et pour ses conseils éclairés qui m'ont permis de prendre de la hauteur sur mon travail au cours de ces trois années.

J'adresse mes plus vifs remerciements à Monsieur Jacques Peter, Ingénieur de recherche à l'ONERA, qui fut mon encadrant pour ce travail. Ses compétences, en particulier dans le domaine aéronautique, assez nouveau pour moi en raison de ma formation initiale, ont été d'un grand secours. Sa disponibilité permanente et son aide au quotidien m'ont permis de franchir les difficultés qui ont parsemé mon parcours.

Mes sincères remerciements vont à Monsieur Gilbert Rogé, Ingénieur de recherche à DASSAULT Aviation, et Monsieur Patrick Siarry, Professeur à l'Université Paris-Est Créteil Val-de-Marne, pour avoir jugé ce mémoire en tant que rapporteurs et pour l'intérêt qu'ils ont porté à ce travail.

Je remercie Madame Paola Cinnella, Professeur à l'école des Arts et Métiers ParisTech, de m'avoir fait l'honneur de présider mon jury et Madame Ingrid Lepot, Responsable du groupe Méthodes Numériques et Optimisation à CENAERO, d'avoir pris part au jury.

Mon travail de thèse n'aurait pas pu aboutir sans l'aide technique et scientifique de nombreux ingénieurs, techniciens, employés de l'ONERA, que je souhaite ici remercier dans leur globalité.

Ces années de thèse m'ont permis également de rencontrer et de cotoyer de nombreux salariés, doctorants et stagiaires, de l'ensemble des départements de l'Onera. Je les remercie toutes et tous pour les moments agréables passés en leur compagnie.

Enfin, car la thèse est une longue aventure qui se vit aussi en dehors du monde professionnel, je remercie ma famille et mes amis pour tout leur soutien.

Table des matières

Liste des symboles **1**

Introduction **3**

1 Optimisation de forme en aérodynamique **11**

1.1 Paramétrisation géométrique d'une forme solide 12

1.1.1 Bosses de Hicks-Henne 12

1.1.2 Paramétrisations sur une base géométrique 12

1.1.3 Méthode PARSEC 14

1.1.4 *Free-form deformation* 14

1.1.5 Paramétrisation par les variables de CAO 15

1.2 Remaillage et méthodes de déformation volumique 15

1.3 Résolution des équations de la mécanique des fluides 16

1.3.1 Équations de Navier-Stokes 17

1.3.2 Équations d'Euler 18

1.3.3 Résolution des équations de Navier-Stokes 19

1.3.4 Méthodes numériques du projet *elsA* 23

1.4 Extraction des coefficients aérodynamiques 29

1.5 Résolution du problème d'optimisation 29

1.6 Calcul du gradient des fonctions d'intérêt par rapport aux paramètres de forme . 31

1.6.1 Calcul des gradients par différences finies 32

1.6.2 Méthode de l'équation linéarisée discrète 32

1.6.3 Méthode de l'équation adjointe discrète 33

1.6.4 Comparaison des méthodes adjointes discrètes et continues. 33

1.6.5 Linéarisation du modèle de turbulence pour les écoulements visqueux turbulents 34

2 Modèles de substitution **39**

2.1 Krigeage 41

2.1.1 Description mathématique 42

2.1.2 Modélisation de la structure de dépendance 43

2.1.3 Les types de krigeage 43

2.1.4 Krigeage ordinaire 44

2.1.5 Implémentation efficace . . . 45
2.2 Réseau de fonctions à base radiale . . . 45
2.2.1 Description mathématique . . . 45
2.2.2 Choix des centres . . . 46
2.2.3 Réglage des paramètres . . . 46
2.3 Les Machines à Vecteur de Support . . . 48
2.3.1 Présentation de la méthode pour la régression linéaire . . . 48
2.3.2 Extension aux approximations non linéaires . . . 50
2.3.3 Résolution du problème quadratique . . . 51
2.3.4 Choix de la fonction noyau . . . 53
2.3.5 Réglage des paramètres . . . 53
2.4 Optimisation des paramètres des modèles . . . 54
2.4.1 Estimation de la qualité du modèle . . . 55
2.4.2 Formulation du problème . . . 56
2.4.3 Intervalle d'optimisation . . . 56
2.4.4 Algorithme d'optimisation . . . 56
2.4.5 Évaluation efficace de l'erreur *leave-one-out* . . . 57
2.4.6 Validation de la méthode de réglage des paramètres . . . 60
2.5 Utilisation de l'information gradient dans les modèles . . . 63
2.5.1 Le co-krigeage . . . 63
2.5.2 L'utilisation de l'information gradient dans la méthode RBF . . . 67
2.5.3 L'utilisation de l'information gradient dans la méthode SVR . . . 68
2.5.4 Optimisation des paramètres des modèles avec gradients . . . 73
2.5.5 Première comparaison des modèles . . . 77

3 Optimisation globale avec modèles de substitution 81
3.1 Méthodes d'échantillonnage de l'espace des paramètres . . . 84
3.1.1 Plans construits à partir de critères statistiques . . . 84
3.1.2 Plans construits à partir de critères déterministes . . . 85
3.1.3 Méthode de remplissage de l'espace (*Space filling design*) . . . 85
3.2 Enrichissement de l'échantillon pour l'optimisation . . . 88
3.2.1 Enrichissement par l'optimum . . . 89
3.2.2 Enrichissement par des méthodes statistiques . . . 90
3.3 Applications à des fonctions mathématiques . . . 96
3.3.1 Présentation des cas tests . . . 96
3.3.2 Précision de l'approximation . . . 96
3.3.3 Optimisation . . . 98
3.4 Applications à l'optimisation de forme aérodynamique . . . 101
3.4.1 Écoulements de fluide parfait autour du profil NACA64A212 . . . 101
3.4.2 Écoulements de fluide visqueux autour d'un profil RAE2822 . . . 109

4 Stratégie d'optimisation locale sans paramétrisation **119**
4.1 Calcul des dérivées par rapport aux points de la surface . . . 120
4.1.1 Méthode adjointe totale . . . 120
4.1.2 Méthode de déformation volumique algébrique [MCV04] . . . 121
4.2 Chaîne d'optimisation sans paramètre de forme . . . 122
4.2.1 Méthode numérique d'optimisation . . . 122
4.2.2 Lissage des sensibilités . . . 122
4.2.3 Contraintes géométriques . . . 125
4.3 Chaîne d'optimisation avec placement a priori des paramètres de forme . . . 127
4.4 Chaîne d'optimisation avec placement optimal des paramètres de forme . . . 128
4.5 Application à l'optimisation d'un profil soumis à un écoulement de fluide visqueux 129
4.5.1 Présentation des cas tests . . . 130
4.5.2 Optimisation basée sur $\frac{d\tilde{J}}{dS}$ (S_1) . . . 130
4.5.3 Optimisation avec paramètres de forme répartis a priori (S_2) . . . 134
4.5.4 Optimisation avec paramètres de forme placés à partir de $\frac{d\tilde{J}}{dS}$ (S_3) . . . 134
4.5.5 Comparaison et commentaires des résultats . . . 135

Conclusion **137**

A Méthode de points intérieurs [Van94] **141**
A.1 Formes primale et duale du problème d'optimisation . . . 141
A.2 Conditions KKT d'optimalité . . . 142
A.3 Chemin central . . . 142
A.4 Recherche d'un point initial . . . 143
A.5 Évolution sur le chemin central . . . 143
A.6 Estimation de μ . . . 145
A.7 Résolution du système linéaire . . . 145
A.8 Calcul de la longueur des pas . . . 146
A.9 Conditions d'arrêt . . . 146
A.10 L'algorithme . . . 146

B Calcul efficace de l'erreur *leave-one-out* **[Rip99]** **147**
B.1 *Radial Basis Function* . . . 147
B.2 Krigeage . . . 149
B.3 Réseau RBF avec gradients . . . 150
B.4 Co-krigeage . . . 153

Références bibliographiques **167**

Liste des symboles

Lettres grecques

α	Vecteur des paramètres de forme du problème d'optimisation
γ	Rapport des chaleurs spécifiques du fluide
λ	Vecteur adjoint de la fonction objectif J
λ_k	Vecteur adjoint de la contrainte G_k
μ	Coefficient de viscosité cinématique
ρ	Masse volumique du fluide
σ_k	Écart-type de l'approximation par les modèles de substitution
τ	Tenseur des contraintes visqueuses
ζ	Paramètre de la fonction gaussienne

Lettres latines

$\mathcal{D}$	Domaine de définition des paramètres de forme (parallélépipède de $\mathcal{R}^{n_f}$)
E	Énergie totale
e	Énergie interne
G_k	Contraintes du problème d'optimisation (dépendance en α)
g_k	Contraintes du problème d'optimisation (dépendance en W et X)
H	Enthalpie totale
J	Fonction objectif du problème d'optimisation (dépendance en α)
j	Fonction objectif du problème d'optimisation (dépendance en W et X)
K_T	Conductivité thermique
n_a	Dimension du champ aérodynamique ($n_a = n_x n_e$)
n_c	Nombre de contraintes du problème d'optimisation
n_e	Taille des équations discrétisées de la mécanique des fluides
n_f	Nombre de paramètres de forme du problème d'optimisation

n_s	Nombre de points du maillage de surface
n_x	Nombre de points du maillage de volume
p	Pression statique
R	Constante spécifique d'un gaz
S	Maillage de surface
s	Flux de chaleur
T	Température
V	Vitesse
W	Champ aérodynamique
X	Maillage de volume

Introduction

L'augmentation du prix du kérosène et les exigences des pouvoirs publics en termes de nuisances environnementales et sonores ont imposé de nouveaux objectifs de conception aux avionneurs. Ainsi, une consommation réduite en carburant, un faible niveau d'émissions de gaz à effet de serre et un fonctionnement discret sont dorénavant des priorités pour concevoir un avion. A cet effet, l'optimisation des formes, aussi bien en aérodynamique externe (recherche de la forme d'un aéronef qui minimise la traînée sous contraintes de portance, de volume interne, de moment de tangage, ...) qu'en aérodynamique interne (recherche de la forme des aubes d'un compresseur ou d'une turbine qui maximise le rendement isentropique) est devenue un maillon incontournable de la chaîne de conception. Longtemps dominée par le savoir-faire des ingénieurs de conception, elle est de plus en plus réalisée numériquement et a connu des développements majeurs ces 30 dernières années.

Historique

Premières expériences en souffleries

L'optimisation aérodynamique de forme peut être envisagée à la fois d'un point de vue expérimental et d'un point de vue numérique. Avant les années 60, l'utilisation de méthodes numériques pour l'analyse de configurations aérodynamiques était très peu répandue. La conception des formes était donc établie à partir d'expériences [LPZ01]. Par exemple, la recherche d'une bonne forme aérodynamique s'effectuait en vol ou en soufflerie : des modifications de la forme étaient appliquées successivement, jusqu'à ce que les performances mesurées soient proches des objectifs fixés. Ces performances étaient évaluées à partir de caractéristiques aérodynamiques globales (traînée, portance, moments). Mais, en raison d'un coût prohibitif, un faible nombre d'expériences pouvait avoir lieu et la forme était bien loin d'une forme optimale. De nos jours, on teste encore occasionnellement un concept aérodynamique de cette manière (par exemple, des essais en vol de *Winglet* ont été effectués dans le cadre du projet AVIATOR [AWI02]). Les limitations de ces procédures ont conduit au développement des méthodes numériques, et dans un premier temps, de méthodes inverses appliquées à des écoulements simplifiés.

Méthodes de conception inverses

Après les découvertes de Prandtl sur la théorie des couches limites, il devient évident que les performances d'un profil sont directement liées à la distribution de vitesse le long de la surface.

Cette observation conduisit au développement des méthodes inverses : il s'agit de rechercher la forme du profil correspondant à une distribution de vitesse ou de pression statique donnée. Les fluides sont considérés comme incompressibles, stationnaires, non visqueux, et sans action extérieure (théorie des écoulements à potentiel de vitesse). Sous ces hypothèses, la distribution de vitesse à la surface peut être calculée de manière exacte sur un cercle unitaire. En utilisant un changement de coordonnées (méthode de transformation conforme) pour se ramener à cette géométrie simplifiée, Mangler [Man38] et Lighthill [Lig45] proposent de calculer le profil correspondant à une distribution de vitesse donnée. C'est la première méthode inverse. Depuis, les méthodes de conception inverse ont été étendues aux fluides compressibles [McF79] et utilisées régulièrement. Elles ont permis notamment de concevoir les profils de Liebeck [Lie73]. Elles possèdent en outre l'avantage d'être relativement peu coûteuses : Giles et Drela [GD97] ont développé une méthode de conception inverse dont le coût de calcul est équivalent à celui de la résolution du problème d'analyse. Ces dernières années, des configurations tridimensionnelles ont d'ailleurs été conçues en utilisant cette méthode [FJW$^+$01]. Néanmoins, la conception optimale de la forme nécessite une définition à priori de la distribution optimale de pression ou de vitesse, ce qui requiert une bonne compréhension du comportement du fluide. Les méthodes de conception inverse ne peuvent donc pas être appliquées de manière générique. Mais elles restent particulièrement intéressantes pour des applications tout à fait spécifiques et pour lesquelles les phénomènes physiques entrant en jeu sont bien compris.

Les méthodes d'optimisation de forme basées sur l'utilisation des méthodes numériques d'optimisation peuvent au contraire être appliquées sans connaissance particulière de la distribution de pression sur le profil optimal. Elles englobent de plus les méthodes de conception inverse, puisque la solution d'un problème inverse peut être calculée comme celle d'un problème de minimisation de la norme de la différence entre la distribution courante de vitesse et celle désirée.

Méthodes numériques d'optimisation de forme

Alors que le développement des premières méthodes inverses a plus de 80 ans, l'utilisation de méthodes numériques d'optimisation de forme est plus récente. Les premiers travaux de ce type ont été effectués par Pironneau [Pir73] [Pir74], puis Glowinski et Pironneau [GP75] au milieu des années 70. Dans ces études, les auteurs calculent théoriquement, par résolution d'un problème de contrôle optimal, la forme minimisant la traînée pour un corps plongé dans un volume et qui se déplace à vitesse constante dans un fluide régi par les équations de Stokes. Leur résultat est que la vorticité de l'écoulement doit être égale en chaque point de la surface solide, ce qui conduit à une forme ovale dont l'écrasement dépend du nombre de Reynolds.

Dans le même temps, avec l'avènement de codes robustes pour la simulation numérique d'écoulements compressibles bidimensionnels, les performances d'un profil peuvent être calculées de manière systématique. Il devient alors envisageable de déterminer numériquement la forme optimale pour une application donnée. A cet effet, la surface du corps au contact du fluide est caractérisée par des paramètres géométriques. Résoudre le problème d'optimisation consiste alors à sélectionner, parmi l'ensemble des formes paramétrées, celle qui minimise une fonction objectif et respecte un ensemble de contraintes, qui peuvent être à la fois géométriques (épaisseur d'un profil, rayon de bord d'attaque) ou aérodynamiques (traînée, portance, moments). La résolution

du problème s'effectue par application des méthodes mathématiques d'optimisation. Hicks et al. [HMV74] appliquent pour la première fois cette méthodologie à la minimisation de la traînée d'un profil. Depuis cet article fondateur, ces méthodes ont été utilisées massivement pour la conception optimale de configurations aéronautiques. On peut par exemple citer les travaux de Aidala et al. [ADM83], Reneaux et Thibert [RT85] ou encore Jameson [Jam88].

Avec ce formalisme, une méthode numérique d'optimisation est donc couplée au solveur des équations de la mécanique des fluides. Dans les premières applications citées ci-dessus, on utilise des méthodes locales. Elles assurent une convergence rapide vers un optimum situé dans le voisinage de la forme initiale. Parmi ces méthodes, on peut distinguer des méthodes directes, comme l'algorithme du simplexe [NM65], utilisé dans un cadre d'optimisation de forme par Duvigneau et Visonneau [DV01]. Si ces méthodes sont plutôt robustes, leur convergence est assez lente et le nombre d'évaluations de la fonction d'intérêt nécessaires à la détermination de l'optimum est important. Dans la majeure partie des cas, on utilise plutôt des méthodes de descente : à partir d'un point initial, l'algorithme se déplace de manière itérative dans l'espace des paramètres en recherchant à chaque itération un nouvel itéré réduisant la fonction d'intérêt. Différentes stratégies de descente ont été proposées dans la littérature. La plus simple d'entre elles est la méthode de plus forte pente (*Steepest Descent*). La recherche d'un nouvel itéré est effectuée par une minimisation unidimensionnelle dans la direction opposée au gradient au point courant. Mais cette méthode est connue pour avancer en zigzag dans l'espace d'optimisation, et par conséquent sa vitesse de convergence est réduite. En raison de sa simplicité d'implémentation, elle a pourtant été utilisée assez largement dans un contexte aéronautique (voir [KAJ00] par exemple). D'autres auteurs proposent d'utiliser une méthode de gradient conjugué [BP10], qui améliore la méthode de plus forte pente, en calculant la direction de descente à partir des dérivées calculées aux itérations courantes et précédentes. On peut également s'orienter vers des méthodes de Newton [BP10] qui nécessitent non seulement le gradient de la fonction objectif, mais également son hessien, ou de quasi-Newton [BP10], dans lesquelles le hessien est approché durant l'optimisation. Toutes ces méthodes d'optimisation sont parfaitement efficaces pour la résolution de problèmes sans contrainte. Si des contraintes sont ajoutées au problème d'optimisation, il est possible de les prendre en compte en pénalisant la fonction objectif et d'effectuer l'optimisation à l'aide d'une méthode citée ci-dessus [CH86] [AV97]. Mais l'approche la plus populaire consiste à utiliser directement des algorithmes d'optimisation avec contraintes. On peut citer notamment la méthode des directions admissibles, utilisée par Hicks et al. dans leur article de 1974 [HMV74], ou la méthode SQP [GMW81], assez largement reconnue comme l'une des méthodes les plus performantes pour l'optimisation non linéaire sous contraintes. Quelle que soit la méthode numérique utilisée, les méthodes de descente nécessitent le calcul du gradient des fonctions d'intérêt. Ces fonctions n'étant pas définies de manière explicite, mais calculées à partir des résultats de simulation d'un écoulement autour de la forme, elles ne sont bien évidemment pas dérivables de manière analytique. Leurs dérivées ont d'abord été calculées par les méthodes de différences finies [HMV74] [HH78]. Le calcul de points décalés d'un pas dans chaque direction de l'espace de dessin permet d'estimer la valeur de la dérivée par rapport à chacune de ces directions par une approximation de la tangente. Mais le coût de cette méthode croît linéairement avec la dimension, et le choix du pas optimal pour les calculs décalés nécessite

une étude paramétrique complexe [Haf85]. Elle reste donc un bon outil de vérification, mais est de moins en moins utilisée en pratique. En effet, depuis l'introduction des méthodes adjointes [Lio71], les gradients des fonctions aérodynamiques peuvent être obtenus de manière précise et pour un coût indépendant de la dimension de l'espace de paramétrisation. La méthode adjointe, appliquée aux écoulements régis par les équations d'Euler [Jam88], a permis des optimisations de profil d'aile et de l'ensemble de la structure d'un aéronef [JR94]. Elle a depuis été généralisée aux écoulements visqueux [JPM97]. Elle a permis une utilisation massive des méthodes locales depuis le début des années 90 et des applications à de nombreuses configurations bidimensionnelles ou tridimensionnelles.

Mais les méthodes locales ne peuvent prétendre atteindre l'optimum de la fonction d'intérêt sur l'ensemble de l'espace de définition des paramètres de forme. En effet, il est connu que les problèmes d'optimisation de forme en aérodynamique conduisent à des fonctions d'intérêt extrêmement multimodales, y compris sur des problèmes simplifiés [OT96]. Si l'on souhaite se rapprocher du minimum global, il est donc nécessaire d'utiliser des méthodes d'optimisation globale. En raison de leur coût de calcul important, celles-ci n'ont été utilisées qu'assez récemment pour des applications en aéronautique. On peut citer comme étude de référence les travaux de Mosetti et Poloni [MP93] ou de Quagliarella et Della Cioppa [QDC94], qui concernent l'application de la méthode génétique. Le principe fondamental de cette méthode est basé sur une analogie avec la sélection naturelle. Depuis, d'autres méthodes d'optimisation globale ont été utilisées, comme la méthode PSO [PD07] ou la méthode des colonies de fourmis [ZAG$^+$01]. Ces méthodes sont particulièrement adaptées aux fonctions objectifs multimodales, mais nécessitent de nombreuses itérations et l'évaluation de la fonction objectif sur un nombre de configurations géométriques très important. A titre d'exemple, certains auteurs ont présenté des optimisations pour lesquelles plus de dix mille évaluations de la fonction d'intérêt sont nécessaires [TC00] [HP01]. L'approche globale ne peut donc être utilisée actuellement de manière intensive que pour des configurations bidimensionnelles et des paramétrisations réduites.

Pour permettre l'utilisation de ces méthodes sur des configurations plus complexes, certains auteurs proposent d'utiliser un modèle physique simplifié [FBK06] ou un modèle mathématique [Gia02] qui simule le comportement de la fonction objectif à partir d'un échantillon de valeurs exactes. L'optimisation est alors effectuée en utilisant à la fois des évaluations exactes et approchées. Ces approches ont permis de réduire sensiblement le nombre d'évaluations de la fonction exacte pour des optimisations globales, notamment en aérodynamique [Gia02] [LM08]. La construction des modèles nécessite bien évidemment des ressources de calcul supplémentaires, mais celles-ci sont en pratique négligeables par rapport au coût des évaluations de la fonction objectif.

Enfin, notons de manière non exhaustive quelques sujets de recherche en cours dans le domaine de l'optimisation aérodynamique de forme :

– le concept d'optimisation de forme isogéométrique, pour lequel l'ensemble de la chaîne d'optimisation (de la paramétrisation au calcul d'analyse, en passant par le calcul de gradient) est basé sur les paramètres géométriques de la forme [HDB11]. Cette approche permet notamment d'éliminer les erreurs dues aux approximations géométriques. Plus simplement, il est possible d'utiliser les paramètres de la CAO comme paramètres de

forme d'une optimisation, mais de conserver une approche en maillage pour la résolution des équations de la mécanique des fluides ;
- l'optimisation multidisciplinaire, et en particulier l'optimisation aéroélastique, qui consiste à prendre en compte les interactions fluide-structure dans le processus d'optimisation [GHK$^+$90] [GSS98] [Mar08] ;
- l'optimisation robuste [Duv07], dont l'objectif est la résolution de problème d'optimisation de forme en aérodynamique incluant des conditions de fonctionnement incertaines, et qui se développe parallèlement au domaine du calcul d'incertitudes.

État de l'art

Les méthodes d'optimisation aérodynamique de forme sont donc aujourd'hui fortement utilisées. Des travaux sur des configurations bidimensionnelles ou tridimensionnelles, soumises à des écoulements turbulents ou de fluide parfait, ont été effectués. Presque toutes les études publiées ces dernières années sont basées sur l'utilisation de méthodes classiques d'optimisation, qu'il est possible de classer en deux grandes catégories.

Les méthodes locales sont les plus répandues aujourd'hui en raison de leur capacité à améliorer rapidement une forme donnée. De nombreuses variantes ont été proposées : elles concernent le choix des paramètres géométriques utilisés pour modéliser la forme solide, la méthode de déformation du maillage volumique, l'algorithme d'optimisation locale ou encore la formulation de la fonction objectif. Elles furent longtemps réservées à des configurations simples, avec un faible niveau de description de la géométrie des objets solides, en raison des méthodes coûteuses tout d'abord disponibles pour le calcul de gradient.

L'avènement de la méthode adjointe a permis de faire un saut quantitatif dans le développement de ces méthodes. Le gradient d'une fonction aérodynamique est dorénavant disponible pour un temps de calcul limité, et surtout indépendamment du nombre de paramètres. Dès lors, des optimisations basées sur des descriptions précises des formes géométriques ont été effectuées. La mise en place d'algorithmes locaux couplés à des codes adjoints pour le calcul des sensibilités s'est généralisée dans toutes les équipes de recherche et ces algorithmes ont été appliqués à de nombreuses configurations (profils d'ailes, ailes, géométrie complète d'un aéronef). Elles ont permis d'améliorer sensiblement les performances des aéronefs. Dans ce domaine, les travaux de recherche récents se focalisent dorénavant notamment sur le lien entre la paramétrisation et les caractéristiques des problèmes d'optimisation. On peut citer par exemple la contribution de Chernukhin et Zingg [CZ11] quant à l'étude de la multimodalité d'un problème d'optimisation de forme et celle de Vassberg et al. [VHRJ11] sur l'impact de la richesse de description géométrique d'une forme lors d'une optimisation.

Mais ces méthodes restent confinées à l'amélioration de formes existantes. Si l'on souhaite déterminer la meilleure forme, au sens d'un critère donné, sur l'ensemble de l'espace des paramètres, il est nécessaire de faire appel aux méthodes d'optimisation globale. Leur utilisation est aujourd'hui assez réduite dans un contexte industriel, en raison de leur forte exigence en ressources de calcul. Afin de contourner ces difficultés, de nouvelles stratégies d'optimisation ont été développées ces dernières années. Les méthodes globales et locales peuvent par exemple

être couplées pour profiter de la capacité de recherche globale des premières, et de la vitesse de convergence rapide des secondes [Duv11]. De plus, des modèles de substitution peuvent être utilisés pour reconstruire les fonctions d'intérêt et réduire le nombre d'évaluations du modèle physique nécessaires aux optimisations [Gia02].

Présentation de la démarche

C'est dans ce cadre que s'inscrit ce travail. Il consiste à développer un algorithme d'optimisation globale utilisant des modèles de substitution, puis à proposer une méthode d'optimisation locale sans paramétrisation. Pour le premier objectif, il s'agit d'explorer en détail le domaine des modèles de substitution et notamment leur utilisation dans le cadre de l'optimisation de forme. Une fois ce travail effectué, il est proposé d'utiliser les informations peu coûteuses fournies par la méthode adjointe pour améliorer la précision des modèles, et donc l'efficacité des algorithmes d'optimisation mis en place. C'est également le développement de la méthode adjointe qui est à l'origine de la deuxième partie de cette thèse. Elle rend en effet accessible le calcul de la dérivée d'une fonction objectif dans un espace de paramètres de forme de dimension importante (par exemple, la position des nœuds de la paroi solide du profil). Il s'agit ici de profiter de cette ressource pour évaluer les performances d'un algorithme d'optimisation dans lequel la forme géométrique n'est pas paramétrée (les points du maillage surfacique discret sont utilisés directement comme paramètres de l'optimisation).

Ce travail est donc articulé en quatre chapitres.

Dans un premier chapitre, nous présenterons la formulation mathématique d'un problème d'optimisation aérodynamique de forme, ainsi que les différents composants d'un algorithme de résolution. Nous donnerons notamment un aperçu des méthodes de paramétrisation géométrique et de déformation volumique. Les équations de la mécanique des fluides considérées pour ce travail seront décrites dans une seconde section, ainsi que les méthodes numériques utilisées pour leur résolution. Enfin, la méthode adjointe discrète sera présentée. La difficile question de son extension aux équations RaNS sera abordée en détail.

Dans un second chapitre, une étude bibliographique des différents types de modèles de substitution a été effectuée. Trois modèles ont été sélectionnés pour leur coût de construction réduit et leur précision. Ils ont été implémentés dans un outil développé en langage Fortran. De premières expériences mathématiques ont démontré la grande sensibilité des performances de ces modèles au choix de leurs paramètres internes. Par conséquent, une méthode de réglage automatique des modèles a été proposée et sera détaillée. Chacun des trois modèles sélectionnés a également été modifié afin d'utiliser pour sa construction à la fois les valeurs des fonctions d'intérêt, mais également les valeurs de leurs dérivées. Enfin, des applications à la reconstruction de fonctions mathématiques ont permis de vérifier la précision et la robustesse des modèles ainsi développés.

Le chapitre suivant est consacré à la mise en place d'une stratégie d'optimisation globale combinant des évaluations sur le modèle physique, et des approximations évaluées sur un modèle de substitution. Dans un premier temps, le choix de la méthode d'échantillonnage utilisée pour construire l'échantillon de départ est fait en comparant différents échantillonnages disponibles dans la littérature. Puis, la stratégie d'enrichissement et le choix de l'algorithme d'optimisation

globale utilisé sont discutés. La procédure ainsi obtenue est alors testée, tout d'abord pour la recherche du minimum de fonctions mathématiques classiques, puis pour l'optimisation de forme en aérodynamique.

Enfin, dans un dernier chapitre, un algorithme d'optimisation locale dans lequel les paramètres de l'optimisation sont directement les points du maillage recouvrant l'objet solide est proposé. Après une étude bibliographique, les différentes étapes de cet algorithme sont décrites. Une méthode de lissage utilisée pour éviter l'apparition de formes solides non lisses est présentée en détail. De même, les contraintes géométriques utilisées pour s'assurer de la validité géométrique de la forme finale sont abordées. Enfin, des applications pour l'optimisation d'un profil d'aile pour des écoulements de fluide visqueux sont présentées.

Une conclusion générale dresse un bilan de cette thèse et introduit de nouvelles perspectives de travail. De plus, deux annexes complètent ce manuscrit : tout d'abord, dans une première partie, un algorithme de résolution des problèmes quadratiques convexes par une méthode de points intérieurs est présenté en détail. Enfin, la démonstration précise des formules de calcul efficace de l'erreur *leave-one-out* de quatre modèles de substitution est donnée dans une seconde partie.

Cette thèse a été effectuée au sein du département DSNA de l'Onera sous la direction de Jacques Peter, ingénieur de recherche à l'Onera et de Jean-Antoine Désidéri, directeur de recherche à l'INRIA Sophia-Antipolis. Elle s'inscrit dans un ensemble de thèses effectués ces dernières années autour de la méthode adjointe, et notamment :

- la thèse de Frédérique Drullion [Dru04] concernant l'utilisation du calcul adjoint pour l'optimisation locale par méthode de gradient ;
- la thèse de Chi-Tuan Pham [Pha06] consacrée à la linéarisation de modèles de turbulence pour la méthode adjointe appliquée aux équations RaNS ;
- la thèse de Meryem Marcelet [Mar08] concernant l'extension de la méthode adjointe pour des problèmes d'optimisation aéroélastique et des premières études sur les modèles de substitution [PMBP07].

Chapitre 1

Optimisation de forme en aérodynamique

Introduction

Dans le cadre d'un problème d'optimisation de forme, on cherche à trouver la meilleure forme solide, au sens d'un critère donné. Cette recherche peut être effectuée d'un point de vue expérimental ou d'un point de vue numérique. Nous nous restreignons ici au domaine de la simulation numérique.

Il s'agit alors de sélectionner une forme solide discrète, définie par un maillage de surface S, et qui minimise une fonction objectif (ou fonction d'intérêt) dépendant de l'application. Ainsi, dans le cadre de l'optimisation de forme en aérodynamique, cette fonction doit évaluer la qualité aérodynamique de la forme. Les applications classiques en aérodynamique externe sont la recherche de la meilleure forme, au sens de la réduction de la traînée sous différentes contraintes.

L'évaluation de la qualité aérodynamique d'une forme, définie par un jeu de n_f paramètres $\alpha \in \mathcal{D}$, nécessite donc l'enchainement des quatre étapes suivantes :

1. d'abord, il est nécessaire de calculer le maillage de surface S correspondant à ce jeu de paramètres. Cette étape dépend du type de paramétrisation de la forme solide choisie pour la définition du problème. Plusieurs approches seront présentées dans la première section ;
2. à partir de ce maillage de surface, un nouveau maillage volumique X doit être généré. En pratique, ce maillage est le plus souvent calculé en propageant les déformations du maillage surfacique sur le maillage volumique initial. Différentes méthodes de déformation volumique ont été introduites dans la littérature et seront détaillées dans la seconde section ;
3. le champ aérodynamique correspondant W est ensuite calculé par résolution d'équations discrètes de mécanique des fluides (ici notées $R(W, X) = 0$) sur le volume défini par le maillage volumique X. Cette résolution est effectuée à l'aide du logiciel *elsA*, dont la présentation sera l'objet de la troisième section ;
4. enfin, les coefficients aérodynamiques sont déduits du champ aérodynamique W et de la géométrie X. A cet effet, des outils d'extraction de traînée sont nécessaires. Ce sera l'objet

de la quatrième section.

Les quatre premières sections de ce chapitre sont donc dédiées à la présentation de chacun des blocs ci-dessus. Ensuite, le problème d'optimisation de forme sera formulé mathématiquement et les méthodes de résolution de ce type de problème seront discutées. Enfin, la dernière section de ce chapitre sera consacrée à la question délicate du calcul de gradient des fonctions d'intérêt par rapport aux paramètres de forme, qui est un élément indispensable pour la résolution des problèmes d'optimisation de forme par des méthodes de descente.

1.1 Paramétrisation géométrique d'une forme solide

La plupart du temps, les points du maillage de surface décrivant la forme solide ne sont pas directement déplacés durant l'optimisation. On procède en effet à une paramétrisation qui permet de réduire la dimension de l'espace dans lequel s'effectue la recherche de la forme optimale. De nombreuses techniques de paramétrisation géométrique ont été introduites dans la littérature. On présente brièvement ci-dessous quelques techniques très répandues.

1.1.1 Bosses de Hicks-Henne

Dans leurs travaux sur l'optimisation numérique en aérodynamique, Hicks et Henne [HH78] ont proposé de paramétrer des profils d'ailes ou des ailes en ajoutant à une géométrie de base une combinaison linéaire de fonctions bosses. Des fonctions, maintenant connues sous le nom de bosses de Hicks-Henne, ont été introduites à cet effet. Elles sont définies, à une abscisse x prise sur la corde normalisée, par la formule suivante :

$$h(x) = \left[\sin\left(\pi x^{\frac{\log 0.5}{\log t_1}}\right)\right]^{t_2}, \quad 0 \leq x \leq 1, \tag{1.1}$$

où les paramètres A, t_1 et t_2 contrôlent respectivement l'amplitude, la position du maximum et la largeur de la bosse considérée.

Selon Jameson [JR94], cette forme de paramétrisation a l'avantage de permettre un contrôle local de la forme. De plus, elle permet d'assurer une régularité des formes obtenues. Par contre, ces fonctions ne forment pas une base complète de l'espace des fonctions nulles en dehors de l'intervalle $[0, 1]$. Cela signifie qu'elles ne peuvent prétendre représenter l'ensemble des formes possibles. Ainsi, en considérant le cas particulier où la solution exacte du problème d'optimisation est connue, elle n'est pas nécessairement décrite par les bosses de Hicks-Henne.

Néanmoins, ces fonctions se sont révélées particulièrement adaptées pour des calculs d'optimisation de forme avec un nombre assez réduit de paramètres. Ainsi, Reuther et al. [RJF+96] ont utilisé cette paramétrisation avec succès pour l'optimisation de la forme d'une aile d'un avion (écoulement transsonique).

1.1.2 Paramétrisations sur une base géométrique

Il est également possible de décrire une forme par une combinaison linéaire de fonctions de base géométrique. Classiquement, ces fonctions de bases sont des polynômes ou des dérivées de

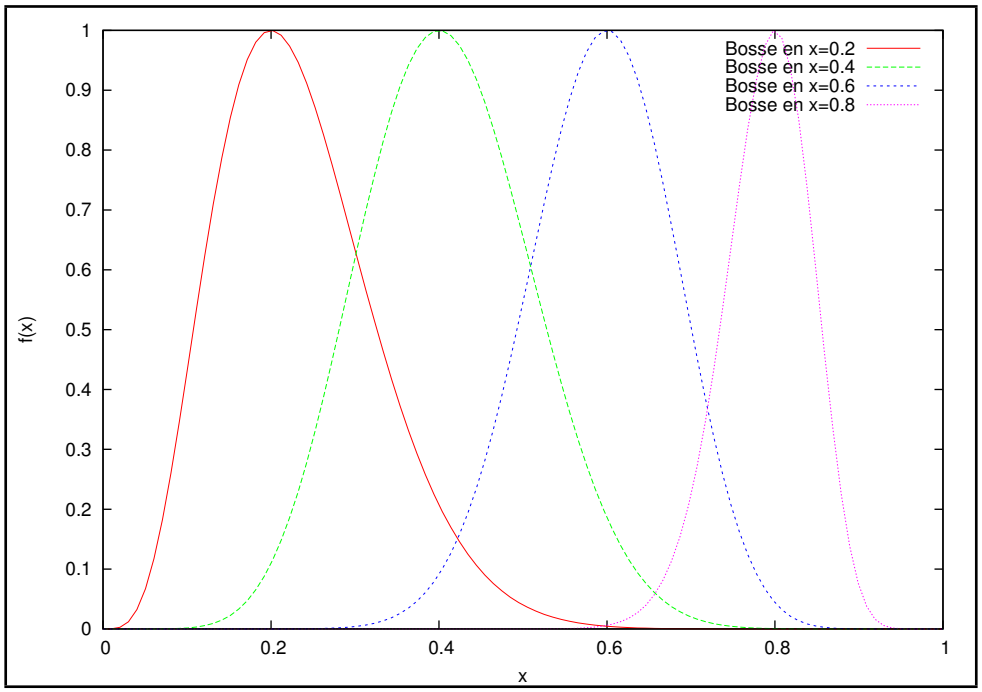


FIG. 1.1 – Bibliothèques de 4 fonctions de Hicks-Henne avec $A = 1$ et $t_2 = 10$

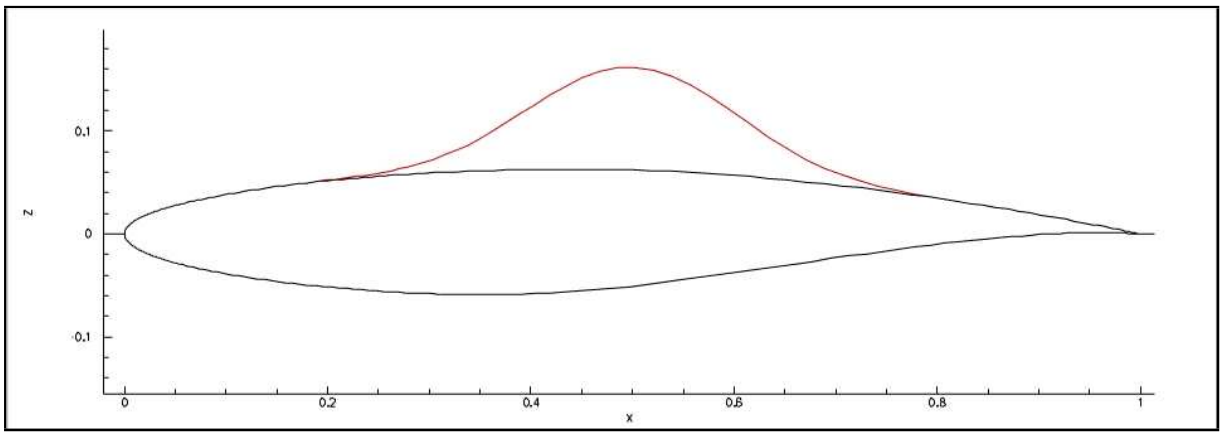


FIG. 1.2 – Déformation de profil par une bosse de Hicks-Henne

polynômes. On trouve notamment :

- des courbes et surfaces de Bézier. Elles sont définies à partir des polynômes de Bernstein, qui possèdent des propriétés particulières (définition récursive et minimisation des erreurs d'arrondi). Elles ont notamment été utilisées par Burgreen et Baysal [BB96] pour l'optimisation d'une aile (écoulement transsonique). Malheureusement, elles décrivent imparfaitement des formes complexes. En effet, l'augmentation du degré des polynômes est alors nécessaire, ce qui conduit à des erreurs numériques importantes (car les polynômes sont définis récursivement) et à un coût de calcul non négligeable ;
- des courbes splines. En plus de reprendre la majeure partie des avantages des courbes de Bézier, elles pallient à leurs inconvénients. Ainsi, la prise en compte d'une forme complexe

peut être effectuée par l'ajout d'un point de contrôle, plutôt que par l'augmentation du degré du polynôme. La paramétrisation géométrique de profils sur une base de fonctions splines (B-spline) a été appliquée avec succès pour des optimisations multiobjectifs [NZP04] [DZ07] ;

– enfin, des NURBS (*NonUniform Rational B-Splines*) qui sont une généralisation des fonctions splines construites sous forme de fractions rationnelles. Ce type de description constitue la représentation géométrique de base dans les outils de CAO. Les NURBS ont été utilisées par Lépine et al. [LTP00] [LGTP01] dans le cadre de l'optimisation de forme en aérodynamique.

1.1.3 Méthode PARSEC

Dans la méthode PARSEC [Sob98], la géométrie d'un profil d'aile est définie à partir de 11 paramètres caractéristiques. Ces paramètres ont été choisis en raison de leur influence sur les performances aérodynamiques des profils. On trouve, par exemple, le rayon de bord d'attaque, la position et la courbure pour les points extrêmes du profil (point le plus bas de l'intrados et point le plus haut de l'extrados). Cette méthode est limitée à la définition de profil et a connu quelques modifications pour prendre en compte des profils avec des bords de fuite divergents. Son implémentation sera présentée plus en détail dans le chapitre 3.

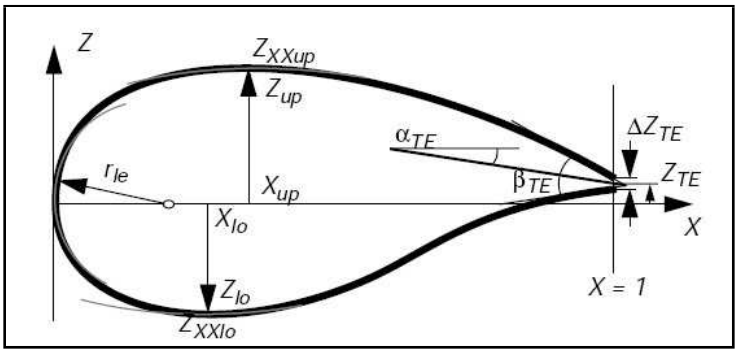


Fig. 1.3 – Paramètres de la méthode PARSEC (extrait de [Sob98])

1.1.4 *Free-form deformation*

Les algorithmes *free-form deformation* (FFD) ont été développés par Sederberg et Parry [SP86]. Il s'agit d'englober un objet solide complexe dans une boîte dont la géométrie sera simple à décrire et de définir les déformations de l'objet solide en fonction des déformations de ce volume de contrôle (voir figure 1.4). A cet effet, les points de l'objet solide sont liés aux points du volume de contrôle à l'aide de polynômes de Bernstein trivariés. Les paramètres de forme sont alors les coordonnées du volume de contrôle. Cette méthode est très performante, car elle permet de déformer facilement des objets à géométrie complexe. Elle est également capable de décrire des déformations importantes à l'aide d'un nombre réduit de paramètres. A l'inverse, elle rencontre des difficultés pour capter de faibles déformations très localisées. Néanmoins, les

approches FFD sont un choix de paramétrisation attractif pour l'optimisation de forme. Elles ont notamment été utilisées dans les travaux de Perry et al. [PBL98] et Samareh [Sam01].

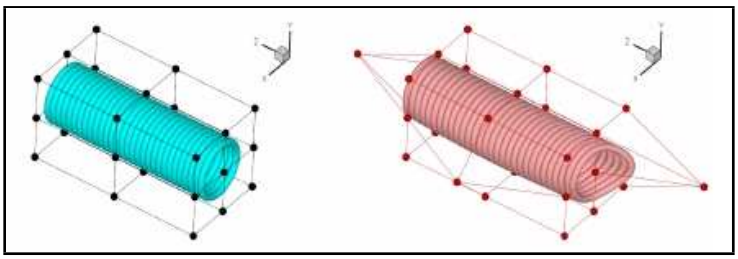


FIG. 1.4 – Déformation par méthode FFD pour une forme cylindrique (extrait de [YMC96])

Comme d'autres méthodes, la paramétrisation par méthode FFD est basée sur une approche par points de contrôle. Ainsi, c'est la modification des coordonnées des points de contrôle qui permet de piloter la surface. Un inconvénient majeur de cette méthode est qu'il n'existe pas de relation intuitive entre la déformation des points de contrôle et l'allure de la courbe ou de la surface résultante. Le pilotage précis de la courbe est alors un exercice complexe. C'est la raison pour laquelle des méthodes de manipulation directe ont été proposées : les déformations de chaque point de la surface sont données en entrée et le déplacement des points de contrôle est alors effectué automatiquement. Cette approche a été appliquée à la méthode FFD dans les travaux de Yamazaki et al. [YMC96].

1.1.5 Paramétrisation par les variables de CAO

Cette approche consiste à utiliser les variables des logiciels de conception assistée par ordinateur (CAO) comme paramètres de l'optimisation. Elle permet de relier directement le processus d'optimisation et les logiciels de CAO. C'est donc un avantage non négligeable dans une perspective industrielle. Néanmoins, elle nécessite des développements techniques supplémentaires, car les dérivées partielles de la forme par rapport aux variables CAO ne sont en général pas fournies par le modeleur géométrique. La paramétrisation basée sur les variables de CAO a été utilisée notamment par Yu et al. [YMJC11]. Dans leurs travaux, les dérivées de la surface, définie par des NURBS, ont été calculées par différentiation automatique.

1.2 Remaillage et méthodes de déformation volumique

Durant le processus d'optimisation de forme, une fois que les paramètres de forme ont été mis à jour et que le solide a été modifié, le maillage volumique doit être adapté à la nouvelle forme, afin de pouvoir résoudre à nouveau les équations de la mécanique des fluides. De manière générale, deux situations distinctes se présentent :

– les maillages de surface et de volume sont construits de manière séquentielle par le même outil de CAO et les paramètres de forme sont reliés aux paramètres de cet outil. Dans ce cas, il est possible de calculer, à chaque nouvelle itération, à la fois les maillages de surface et de volume correspondant au jeu de paramètres courant ;

– le maillage de surface est engendré directement à partir des paramètres de forme, mais le maillage de volume ne peut pas être construit automatiquement à partir de celui-ci. Il doit alors être construit par déformation du maillage volumique initial, en propageant les déformations appliquées au maillage de surface.

Dans le second cas, il est nécessaire de déterminer comment déformer le maillage de volume en fonction des déformations de la surface. A cet effet, plusieurs méthodes ont été introduites dans la littérature :

– les méthodes algébriques ou intégrales : ce sont des méthodes qui donnent explicitement le déplacement de chaque point de maillage à partir du déplacement des points de la paroi. Elles font intervenir soit un calcul algébrique, soit un calcul intégral. Elles ne font pas intervenir de résolution de systèmes linéaires, si bien que leur mise en œuvre est peu coûteuse. On peut citer par exemple la méthode proposée par Meaux et al. [MCV04] (figure 1.5) que nous utiliserons pour les applications de cette thèse, et qui sera présentée plus en détail par la suite ;
– les méthodes de déformation par modèle de substitution : la fonction de déformation définie en chaque point du maillage volumique est approchée par un modèle de substitution. Ce modèle est entraîné sur les points pour lesquels la déformation est connue, c'est-à-dire pour chaque point du maillage surfacique. Cette méthode a notamment été proposée avec des réseaux RBF par Jakobsson et Amoignon [JA07] et De Boer et al. [DBVDSB07] ;
– les méthodes basées sur la résolution d'une équation aux dérivées partielles : ces méthodes consistent à assimiler le problème de déformation volumique à un problème de physique classique et à utiliser une méthode numérique connue pour sa résolution. On trouve par exemple, dans cette catégorie, des méthodes basées sur une analogie entre le maillage et un treillis de ressorts, comme la méthode des ressorts [Dug98] ou la méthode de Batina [Bat89] qui en est une simplification moins coûteuse. Dans la méthode développée par Jacquotte et al. [JMG89], le maillage est vu comme un solide élastique et la résolution s'effectue à l'aide des outils développés pour l'élasticité linéaire des milieux continus : la matrice de rigidité du maillage est calculée et le déplacement des nœuds est relié au bilan des efforts en chaque nœud. Enfin, on peut citer la méthode des singularités [Can02] qui est fondée sur une analogie avec le calcul du champ électromagnétique produit par une source ponctuelle.

1.3 Résolution des équations de la mécanique des fluides

La troisième étape nécessaire à l'évaluation de la qualité aérodynamique d'une forme solide est la résolution des équations de la mécanique des fluides, qui régissent le comportement du fluide autour de cette forme. Dans le cadre de nos applications, ces équations sont les équations moyennées de Navier-Stokes (approche RaNS) ou leur simplification dans le cas d'un fluide parfait (équations d'Euler). Dans cette section, nous présenterons donc ces équations, ainsi que les méthodes numériques utilisées pour leur résolution.

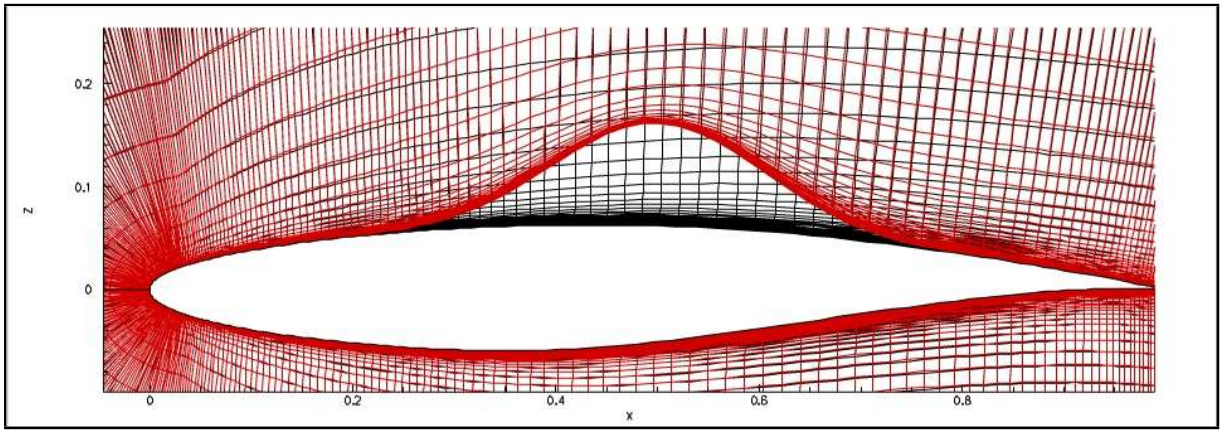


FIG. 1.5 – Déformation de maillage volumique par une méthode algébrique à la suite d'une déformation de la surface par une bosse de Hicks-Henne

1.3.1 Équations de Navier-Stokes

L'écoulement d'un fluide visqueux, turbulent, newtonien et compressible est régi par les équations de Navier-Stokes. La forme locale de ces équations découle du bilan de masse, de quantité de mouvement et d'énergie totale d'un volume particulaire.

Lois de conservation

Ces équations sont présentées sous la forme d'équations aux dérivées partielles permettant de déterminer l'évolution, à partir de conditions initiales données, des champs de la masse volumique ρ, de la quantité de mouvement par unité de volume $\rho V = [\rho u, \rho v, \rho w]^T$ et de l'énergie totale par unité de volume ρE (l'énergie totale E étant égale à la somme de l'énergie interne e et de l'énergie cinétique $\frac{1}{2}V^2$, toutes ces grandeurs étant homogènes à des énergies par unité de masse). Les équations de Navier-Stokes s'écrivent :

– équation de bilan de masse :

$$\frac{\partial \rho}{\partial t} + div(\rho V) = 0; \tag{1.2}$$

– équation de bilan de quantité de mouvement :

$$\frac{\partial \rho V}{\partial t} + div(\rho V \otimes V + pI - \tau) = 0; \tag{1.3}$$

– équation de bilan d'énergie :

$$\frac{\partial \rho E}{\partial t} + div(\rho E V + pV - \tau \cdot V + s) = 0. \tag{1.4}$$

Dans ces équations, V désigne le champ de la vitesse absolue ; p, τ et s représentent respectivement la pression statique, le tenseur des contraintes dues à la viscosité et le vecteur de flux de chaleur dû à la conductivité thermique. Pour fermer le système, il faut donc préciser l'expression du scalaire p, du vecteur s et du tenseur τ.

Lois de comportement

Pour un fluide newtonien, l'hypothèse de Newton permet d'exprimer le tenseur des contraintes τ à partir du champ de vitesse :

$$\tau = \lambda(divV)I + 2\mu(\nabla V + \nabla V^T), \tag{1.5}$$

où λ et μ sont les paramètres de Lamé. En supposant valide l'hypothèse de Stokes - $3\lambda + 2\mu = 0$, le coefficient de viscosité volumique λ s'exprime en fonction de μ. Alors on peut exprimer τ en fonction de V et de μ :

$$\tau = -\frac{2}{3}\mu(divV)I + \mu(\nabla V + \nabla V^T). \tag{1.6}$$

Le vecteur de flux de chaleur s est donné par la loi de Fourier :

$$s = -K_T \nabla T, \tag{1.7}$$

où K_T désigne le coefficient de conductivité thermique et T la température absolue.

On introduit ici le nombre de Prandtl, nombre adimensionné qui ne dépend pas de la température en pratique et qui permet de relier le coefficient de conductivité thermique K_T et le coefficient de viscosité moléculaire μ :

$$Pr = \frac{c_p \mu}{K_T}. \tag{1.8}$$

Dans le cadre de nos applications (écoulement d'air), le nombre de Prandtl vaut $0,72$.

Lois d'état

Ces lois permettent d'exprimer l'énergie interne e, la pression p, la conductivité thermique K_T et le coefficient de viscosité μ en fonction de la température T.

La loi des gaz parfaits introduit une relation entre la pression, la température et l'énergie interne :

$$p = \rho R_s T = (\gamma - 1)\rho e, \tag{1.9}$$

où R_s représente la constante spécifique du gaz considéré, c'est-à-dire le rapport de la constante universelle des gaz parfaits R et de sa masse molaire ($287 Jkg^{-1}K^{-1}$ dans le cas de l'air sec).

Enfin, le second principe de la thermodynamique affirme que le coefficient de viscosité moléculaire μ n'est fonction que de la température. Il est donné ici par la loi de Sutherland :

$$\mu = \mu_0 (\frac{T}{T_0})^{\frac{3}{2}} \frac{T_0 + C_0}{T + C_0}, \tag{1.10}$$

μ_0 désigne la valeur de la viscosité pour la température T_0 et $C_0 = 110.4K$.

1.3.2 Équations d'Euler

Les équations d'Euler sont une simplification des équations de Navier-Stokes dans le cas d'un fluide parfait, c'est-à-dire d'un fluide non visqueux, et non conducteur de chaleur. Ces caractéristiques particulières permettent d'effectuer les approximations suivantes :

– le tenseur des contraintes visqueuses s'annule : $\tau = 0$;
– le terme de flux de chaleur est également nul : $s = 0$.

Les équations d'Euler se formulent alors par simplification des équations de Navier-Stokes :

– équation de bilan de masse :

$$\frac{\partial \rho}{\partial t} + div(\rho V) = 0; \tag{1.11}$$

– équation de bilan de quantité de mouvement :

$$\frac{\partial \rho V}{\partial t} + div(\rho V \otimes V + pI) = 0; \tag{1.12}$$

– équation de bilan d'énergie :

$$\frac{\partial \rho E}{\partial t} + div(\rho E V + pV) = 0. \tag{1.13}$$

Le système d'équations ainsi formulé fournit cinq équations pour six inconnues. Par conséquent, celui-ci doit être fermé en utilisant une loi d'état, le plus souvent la loi des gaz parfait déjà évoquée ci-dessus.

Les équations d'Euler permettent d'effectuer un certain nombre de simulations, mais ne permettent pas la description des phénomènes impliquant la viscosité du fluide, notamment les couches limites. Il faut alors résoudre le système non simplifié des équations de Navier-Stokes.

1.3.3 Résolution des équations de Navier-Stokes

Écoulements laminaires et écoulements turbulents

Le nombre de Reynolds *Re* permet de caractériser le rapport entre les forces d'inertie et les forces visqueuses. On distingue trois principaux régimes, suivant les valeurs du Reynolds et le type d'écoulement :

– les faibles valeurs du nombre de Reynolds (inférieures à 2000 pour l'écoulement de Poiseuille) conduisent à un écoulement laminaire et réversible (l'inertie est négligée). On parle d'écoulement de Stokes ;
– aux valeurs intermédiaires du nombre de Reynolds (entre 2000 et 3000 pour ce même cas), les forces d'inertie sont prépondérantes, l'écoulement reste laminaire mais n'est plus réversible ;
– enfin, pour les fortes valeurs du nombre de Reynolds, les forces d'inertie sont si importantes que l'écoulement devient turbulent.

Simulation des écoulements turbulents

A grand nombre de Reynolds, la turbulence rend les grandeurs physiques locales instationnaires et les mesures non reproductibles dans une partie du domaine fluide. La très faible taille des phénomènes observés requiert des maillages extrêmement fins. Différentes méthodes correspondant à différents niveaux d'approximation ont été développées :

– la simulation directe (*Direct Numerical Simulation* - DNS) consiste à prendre en compte

toutes les échelles caractéristiques de l'écoulement. Elle nécessite des maillages suffisamment fins et donc des ressources de calcul très importantes ;

– la simulation des grandes échelles (*Large Eddy Simulation* - LES) propose de simuler directement les plus grandes échelles de l'écoulement, et d'approcher les plus petites échelles par l'introduction de modèles dits de *sous-maille*. L'approximation obtenue est alors moins fiable, mais les coûts de calcul sont plus faibles ;

– enfin, la méthode des équations de Navier-Stokes moyennées (*Reynolds-averaged Navier-Stokes* - RaNS) est une approche statistique. Le système des équations de Navier-Stokes est alors remplacé par un système d'équations formulé sur la valeur moyenne du champ (au sens de la répétition de l'écoulement). Cette méthode est la moins précise, mais son coût informatique réduit en fait la méthode la plus utilisée actuellement pour les simulations industrielles.

Équations de Navier-Stokes moyennées

La modélisation de la turbulence dans la méthode RaNS est effectuée par une approche statistique. Le champ est séparé en une partie moyenne (écoulement moyen), obtenue par une moyenne d'ensemble sur un grand nombre de réalisations, et une partie fluctuante (écoulement turbulent) de moyenne nulle :

$$q(x,t) = \bar{q} + q', \tag{1.14}$$

où $\bar{q}$ est la moyenne statistique, dite moyenne de Reynolds :

$$\bar{q}(x,t) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} q_i(x,t). \tag{1.15}$$

Cette moyenne est très bien adaptée au cas des écoulements incompressibles, mais entraîne l'apparition de nombreux termes supplémentaires dans les équations différentielles dans le cas des écoulements compressibles. Il est alors plus indiqué d'utiliser une moyenne pondérée par la masse volumique, dite moyenne de Favre. Cette moyenne permet de retrouver formellement un système d'équations proche de celui obtenu pour un écoulement compressible laminaire. En outre, elle permet de réduire le nombre de corrélations dans les équations RaNS. Par définition, la moyenne de Favre d'une quantité q est donnée par :

$$\tilde{\bar{q}} = \frac{\overline{\rho q}}{\bar{\rho}}. \tag{1.16}$$

En moyennant les équations de Navier-Stokes, et en utilisant les propriétés de la moyenne de Favre, on obtient les équations RaNS :

$$\frac{\partial \bar{\rho}}{\partial t} + div(\bar{\rho}\tilde{\bar{V}}) = 0, \tag{1.17}$$

$$\frac{\partial \bar{\rho}\tilde{\bar{V}}}{\partial t} + div(\bar{\rho}\tilde{\bar{V}} \otimes \tilde{\bar{V}} + \bar{p}I) = div(\tau + \tau_R), \tag{1.18}$$

$$\frac{\partial \bar{\rho}\tilde{\bar{E}}}{\partial t} + div(\bar{\rho}\tilde{\bar{E}}\tilde{\bar{V}} + \bar{p}\tilde{\bar{V}}) = div((\tau + \tau_R)\tilde{\bar{V}} + \tilde{\bar{s}} + \tilde{\bar{s}}_t. \tag{1.19}$$

On a fait apparaître :

– le tenseur des contraintes turbulentes, ou tenseur de Reynolds, τ_R :

$$\tau_R = -\overline{\rho V'' \otimes V''}; \tag{1.20}$$

– le tenseur des contraintes moyennes $\bar{\tau}$:

$$\bar{\tau} = -\frac{2}{3}\mu div(\bar{\bar{V}})I + \mu(\nabla\bar{\bar{V}} + \nabla\bar{\bar{V}}^T); \tag{1.21}$$

– le flux de chaleurs turbulent :

$$\bar{\bar{s}}_t = -\overline{pV''} - \overline{pe''V''}; \tag{1.22}$$

– et enfin le flux de chaleur laminaire :

$$\bar{\bar{s}} = -\lambda\nabla\bar{\bar{T}}. \tag{1.23}$$

Dans un espace tridimensionnel, l'opération qui consiste à moyenner les équations a fait apparaître six nouvelles inconnues provenant du tenseur de Reynolds τ_R (symétrique, de dimension 3×3), ainsi que trois nouvelles inconnues provenant du flux de chaleur turbulent. Il est nécessaire, pour que la résolution du système soit possible, de le fermer, c'est-à-dire de réintroduire des informations sous forme d'hypothèses physiques. C'est l'étape de modélisation de la turbulence.

Modélisation de la turbulence

La fermeture à l'ordre 0 est basée sur l'hypothèse de Boussinesq : le tenseur de Reynolds et le flux de chaleur turbulent sont définis par analogie au tenseur des contraintes visqueuses et au flux de chaleur :

$$\tau_R = -\frac{2}{3}(\bar{\rho}k + \mu_t div\bar{\bar{V}})I + \mu_t(\nabla\bar{\bar{V}} + \nabla\bar{\bar{V}}^T), \tag{1.24}$$

$$s_t = -\frac{C_p\mu_t}{P_{r_t}}\nabla\bar{\bar{T}}. \tag{1.25}$$

Dans ces deux expressions, P_{r_t} désigne le nombre de Prandtl turbulent, constante indépendante de la température prise en pratique égale à $0,9$.

L'hypothèse de Boussinesq permet donc de réduire la modélisation de la turbulence à deux termes :

– le coefficient de viscosité turbulente μ_t qui relie le tenseur de Reynolds et le champ moyen ;

– l'énergie cinétique moyenne du champ turbulent $k = \frac{1}{2}\bar{\bar{V''^2}}$.

Ces termes peuvent être obtenus soit par un modèle algébrique, soit par des modèles à équations de transport. On pourra se reporter à un document de Cambier [Cam90] pour plus de détails. Dans le cadre de notre travail, deux modèles de turbulence à équations de transport ont été utilisés et sont présentés par la suite :

– le modèle à une équation de transport de Spalart-Allmaras [SA92], qui propose de résoudre une quantité proportionnelle à μ_t à l'aide d'une équation de transport formulée de façon empirique ;

– le modèle à deux équations de transport $k-\omega$ de Wilcox [Wil88] [Wil93], qui fait intervenir une équation pour l'énergie cinétique de la turbulence k, et une autre équation pour le taux spécifique de dissipation ω.

Modèle de Spalart-Allmaras [SA92] Le modèle de Spalart-Allmaras utilise une équation de transport pour la quantité $\tilde{\nu}$ qui, loin de la paroi, est semblable à $\nu_t = \frac{\mu_t}{\rho}$. La viscosité turbulente s'exprime en fonction de $\tilde{\nu}$ par les relations suivantes :

$$\mu_t = \rho\tilde{\nu} f_{v1}, \tag{1.26}$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \tag{1.27}$$

$$\chi = \min\left[\frac{\max(\rho\tilde{\nu}, cv1)}{\max(\mu, its)}, \mu_{RM}\right], \tag{1.28}$$

où le paramètre μ_{RM} permet de limiter la valeur du rapport $\frac{\mu_t}{\mu}$ et où $cv1$ et $its = 10^{-30}$ représentent des valeurs minimales imposées à la variable turbulente et au coefficient de viscosité moléculaire.

L'équation propageant la quantité $\rho\tilde{\nu}$ s'écrit :

$$\begin{aligned}\frac{\partial\rho\tilde{\nu}}{\partial t} + \frac{\partial\rho\tilde{\nu}u_j}{\partial x_j} = C_{b1}\tilde{S}\rho\tilde{\nu} - C_{w1}f_w\frac{|\rho\tilde{\nu}|}{\rho}\frac{\rho\tilde{\nu}}{\eta^2} + \\ \frac{1}{\sigma_{SA}}\left[\frac{\partial}{\partial x_j}\left((\mu+\rho\tilde{\nu})\frac{\partial\tilde{\nu}}{\partial x_j}\right) + \min\left(C_{b2}\rho\frac{\partial\tilde{\nu}}{\partial x_j}\frac{\partial\tilde{\nu}}{\partial x_j}, 20\rho\tilde{\nu}\max\left(0, C_{b1}\tilde{S}\right)\right)\right],\end{aligned} \tag{1.29}$$

avec :

$$\tilde{S} = \bar{\omega} + \frac{|\rho\tilde{\nu}|}{\rho}\frac{f_{v2}}{\kappa_{SA}^2\eta^2}, \tag{1.30}$$

$$f_{v2} = 1 - \frac{\chi}{1+\chi f_{v1}}, \tag{1.31}$$

$$f_w = g\left(\frac{1+C_{w3}^6}{g^6+C_{w3}^6}\right)^{\frac{1}{6}}, \tag{1.32}$$

$$g = \tilde{r} + C_{w2}(\tilde{r}^6 - \tilde{r}), \tag{1.33}$$

$$\tilde{r} = \max\left[0, \min\left(10, \frac{1}{\max\left(\tilde{S}, cutoff\right)}\frac{|\rho\tilde{\nu}|}{\rho}\frac{1}{\kappa_{SA}^2\eta^2}\right)\right], \tag{1.34}$$

où η représente la distance par rapport à la paroi et $\bar{\omega} = |\nabla \wedge U|$ désigne le module de la vorticité. Enfin, les constantes du modèle sont fixées aux valeurs données dans le tableau 1.1.

Modèle $k-\omega$ de Wilcox [Wil88] Les deux quantités transportées par le modèle sont l'énergie cinétique de la turbulence k, et le taux spécifique de dissipation ω. A partir de ces deux données, on peut calculer la viscosité turbulente à l'aide de l'équation suivante :

$$\mu_t = \frac{\rho k}{\omega}. \tag{1.35}$$

C_{b1}	$0,1355$
C_{b2}	$0,622$
σ_{SA}	$\frac{2}{3}$
κ_{SA}	$0,41$
C_{w1}	$\frac{C_{b1}}{\kappa_{SA}^2} + \frac{1+C_{b2}}{\sigma_{SA}}$
C_{w2}	$0,3$
C_{w3}	2
C_{v1}	$7,1$

TAB. 1.1 – Constantes du modèle de Spalart-Allmaras

Les équations du modèle de turbulence sont :

$$\frac{\partial \rho k}{\partial t} + \frac{\partial \rho u_j k}{\partial x_j} = P - \beta^*_{KW} \rho \omega k + \frac{\partial}{\partial x_j}\left[\left(\mu + \sigma_{KWk}\frac{\rho k}{\omega}\right)\frac{\partial k}{\partial x_j}\right], \tag{1.36}$$

$$\frac{\partial \rho \omega}{\partial t} + \frac{\partial \rho u_j \omega}{\partial x_j} = \frac{\gamma_{KW}\omega}{k} P - \beta_{KW}\rho\omega^2 + \frac{\partial}{\partial x_j}\left[\left(\mu + \sigma_{KW\omega}\frac{\rho k}{\omega}\right)\frac{\partial \omega}{\partial x_j}\right], \tag{1.37}$$

où :

$$P = \left[\mu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) - \frac{2}{3}\rho k \delta_{ij}\right]\frac{\partial u_i}{\partial x_j}. \tag{1.38}$$

Enfin, les constantes du modèle sont fixées aux valeurs données dans le tableau 1.2.

β^*_{KW}	$0,09$
β_{KW}	$0,075$
σ^*_{KW}	$0,5$
σ_{KW}	$0,5$
γ_{KW}	$\frac{\beta_{KW}}{\beta^*_{KW}} - \sigma_{KW}\frac{\kappa^2_{KW}}{\sqrt{\beta^*_{KW}}}$
κ_{KW}	$0,41$
σ_{KWk}	$\frac{1}{\sigma^*_{KW}}$
$\sigma_{KW\omega}$	$\frac{1}{\sigma_{KW}}$

TAB. 1.2 – Constantes du modèle $k-\omega$ de Wilcox

1.3.4 Méthodes numériques du projet *elsA*

Le projet *elsA*[1] [CV08], lancé en 1997, est un projet de regroupement de l'ensemble des logiciels développés à l'Onera pour la simulation numérique en aérodynamique. C'est un logiciel multiapplication (aérodynamique externe avec volumes fixes ou tournants et aérodynamique interne).

Il s'appuie sur les langages *C++* et *Fortran* pour les méthodes de calcul scientifique, ainsi que le langage *Python* pour l'interface utilisateur. Il permet de simuler des écoulements compressibles stationnaires ou instationnaires, de fluide parfait ou de fluide visqueux, en repère fixe ou mobile, dans des maillages rigides ou déformables.

[1] ensemble logiciel pour la simulation en Aérodynamique

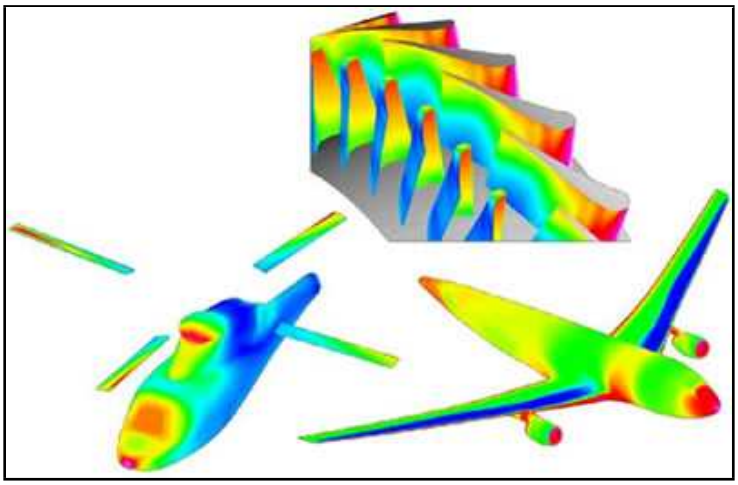


FIG. 1.6 – Résultat de simulations effectuées avec *elsA*

Méthodes des volumes finis

On s'intéresse ici à la discrétisation des équations RaNS pour un problème stationnaire. La méthode de discrétisation considérée et implémentée dans le logiciel *elsA* est la méthode des volumes finis, dans laquelle les inconnues sont des valeurs moyennes dans les cellules de discrétisation affectées au centre de celles-ci (approche *cell-centered*).

La discrétisation de l'espace est réalisée avec des maillages structurés multiblocs, avec une grande flexibilité d'assemblage de ces blocs, afin de faciliter la prise en compte des géométries complexes. Les indices de maillage sont notés (i, j, k) dans le cas tridimensionnel. L'application aux écoulements bidimensionnels est traitée en trois dimensions entre deux plans parallèles.

Le domaine de calcul est donc constitué d'un ensemble de cellules hexaèdriques élémentaires. Chaque cellule de discrétisation Ω est caractérisée par son volume $V_\Omega = \int_\Omega d\Omega$, sa surface fermée $\delta\Omega = \bigcup_{l=1}^{6} \Sigma_l$, où Σ_l représente la face l de la cellule, dont la normale extérieure unitaire est notée N.

On rappelle la formulation locale des équations de Navier-Stokes exprimées dans le repère absolu, telles que définies à la section 1.3.1 :

$$\frac{\partial \rho}{\partial t} + div(\rho V) = 0, \tag{1.39}$$

$$\frac{\partial \rho V}{\partial t} + div(\rho V \otimes V + pI) - div(\tau) = 0, \tag{1.40}$$

$$\frac{\partial \rho E}{\partial t} + div(\rho E V + pV) - div(\tau \cdot V - s) = 0. \tag{1.41}$$

On introduit les notations suivantes :

- le vecteur du champ aérodynamique $W = \left[\rho, \rho V^T, \rho E\right]^T$;
- le vecteur des termes convectifs $Fc = \left[\rho V, (\rho V \otimes V + pI)^T, \rho E V + pV\right]^T$;
- et enfin le vecteur des termes diffusifs $Fd = \left[0, (-\tau)^T, -\tau \cdot V - s\right]^T$.

Alors, en intégrant les équations de Navier-Stokes sur une cellule Ω, on obtient un système

de lois de bilan sous forme intégrale :

$$\frac{d}{dt}\int_{\Omega} W d\Omega + \oint_{\delta\Omega} \left[Fc\left(W\right) + Fd\left(W, \nabla W\right)\right] \cdot N d\Sigma = 0. \tag{1.42}$$

On définit :

– la valeur moyenne du vecteur des variables conservatives W dans la cellule Ω :

$$\overline{W_{\Omega}} = \frac{1}{V_{\Omega}} \int_{\Omega} W d\Omega; \tag{1.43}$$

– la valeur du vecteur flux à travers Σ_l :

$$\left(\overline{F_n}\right)_{\Sigma_l} = \int_{\Sigma_l} \left[Fc\left(W\right) + Fd\left(W, \nabla W\right)\right] \cdot N d\Sigma. \tag{1.44}$$

Alors on peut écrire l'équation 1.42 sous la forme d'un système d'équations différentielles ordinaires :

$$\frac{d}{dt}\left(V_{\Omega}\overline{W_{\Omega}}\right) = -\left[\sum_{l=1}^{6}\left(\overline{F_n}\right)_{\Sigma_l}\right]. \tag{1.45}$$

On recherche alors une fonction de flux numérique comme une approximation numérique du flux exact $\left(\overline{F_n}\right)_{\Sigma_l}$, satisfaisant la propriété usuelle de consistance. On considère une méthode numérique semi-discrétisée en espace :

$$\frac{d}{dt}\left(V_{\Omega} W_{\Omega}\right) = -\left[\sum_{l=1}^{6} F\left(W_{\Omega}, W_{\Omega_l}\right) \cdot N_{\Sigma_l}\right], \tag{1.46}$$

où :

– W_{Ω} est une approximation numérique de la valeur moyenne $\overline{W_{\Omega}}$. Pour un schéma numérique d'ordre au plus 2 en espace, on peut confondre la valeur au centre et la moyenne volumique ;
– $F\left(W_{\Omega}, W_{\Omega_l}\right)$ est le flux numérique, fonction de W_{Ω} et des approximations numériques W_{Ω_l} calculées dans un certain nombre de cellules voisines à chacune des faces Σ_l ;
– N_{Σ_l} est un vecteur normal à l'interface Σ_l non unitaire égal à $\int_{\Sigma_l} N d\Sigma$.

Lorsque le maillage multibloc ne dépend pas de la variable temps, comme c'est le cas ici, on peut considérer la méthode semi-discrétisée en espace sous la forme suivante :

$$\frac{d}{dt} W_{\Omega} = -\frac{1}{V_{\Omega}}\left[\sum_{l=1}^{6} F\left(W_{\Omega}, W_{\Omega_l}\right) \cdot N_{\Sigma_l}\right] = -R_{\Omega}, \tag{1.47}$$

où R_{Ω} est le résidu numérique explicite de modélisation.

Traitement des flux convectifs Le flux convectif peut être discrétisé à l'aide de formules centrées ou décentrées. Le choix de l'un ou l'autre de ces types de formules dépend des applications envisagées. Ainsi, dans le cadre de l'utilisation des méthodes de calcul de gradients discrets, il est nécessaire d'inverser la matrice jacobienne $\frac{\partial R}{\partial W}$. Or, il est connu qu'une discrétisation utilisant des flux décentrés assure un meilleur conditionnement de cette matrice [Cor00]. Dans la

suite de notre travail, nous utiliserons donc le schéma décentré de Roe [Roe81].

Le schéma de Roe Dans le cas d'un système à cinq équations, le flux physique dans la direction $\overrightarrow{n} = (n_x, n_y, n_z)^T$ s'écrit :

$$Fc = \left\langle \overrightarrow{Fc}, \overrightarrow{n} \right\rangle = \begin{bmatrix} \rho u \\ \rho u^2+p \\ \rho uv \\ \rho uw \\ \rho uH \end{bmatrix} n_x + \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2+p \\ \rho vw \\ \rho vH \end{bmatrix} n_y + \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2+p \\ \rho wH \end{bmatrix} n_z, \tag{1.48}$$

où l'enthalpie totale est définie par $H = E + \frac{p}{\rho}$. Si l'on note W_g et W_d deux états de l'espace des états, le flux numérique de Roe est alors calculé par :

$$F_{Roe}(W_g, W_d) = \frac{Fc(W_g) + Fc(W_d)}{2} - \frac{1}{2}|A|(W_d - W_g), \tag{1.49}$$

où $|A| = Mdiag(\lambda_A) M^{-1}$ avec $diag(\lambda_A)$ la matrice diagonale des valeurs absolues des valeurs propres de A et $A = A(W_g, W_d)$ est la matrice de Roe satisfaisant les trois conditions suivantes :

- A est diagonalisable avec des valeurs propres réelles ;
- $A(W_d - W_g) = Fc(W_d) - Fc(W_g)$;
- A est la différentielle de Fc par rapport au champ W.

Dans le cas des équations d'Euler, la matrice A est exactement la matrice jacobienne de Fc en l'état moyen de Roe $\tilde{W}$, où :

$$\tilde{W} = \begin{bmatrix} \sqrt{\rho_g \rho_d} \\ \frac{\sqrt{\rho_g} u_g + \sqrt{\rho_d} u_d}{\sqrt{\rho_g} + \sqrt{\rho_d}} \\ \frac{\sqrt{\rho_g} v_g + \sqrt{\rho_d} v_d}{\sqrt{\rho_g} + \sqrt{\rho_d}} \\ \frac{\sqrt{\rho_g} w_g + \sqrt{\rho_d} w_d}{\sqrt{\rho_g} + \sqrt{\rho_d}} \\ \frac{\sqrt{\rho_g} H_g + \sqrt{\rho_d} H_d}{\sqrt{\rho_g} + \sqrt{\rho_d}} \end{bmatrix}. \tag{1.50}$$

Correction de Harten Le flux numérique de Roe 1.49 peut conduire à des solutions non entropiques [HH83]. Pour approcher la solution physique du problème, il est nécessaire d'assurer la compatibilité du schéma de Roe avec la condition d'entropie. Harten [HH83] a montré que la solution vérifie cette condition si le flux numérique de Roe est modifié de la manière suivante :

$$F_{Roe}(W_g, W_d) = \frac{Fc(W_g) + Fc(W_d)}{2} - \frac{1}{2}\Psi(|A|)(W_d - W_g), \tag{1.51}$$

où $\Psi(|A|) = Mdiag(\Psi(|\lambda_A|)) M^{-1}$. La fonction Ψ est définie par :

$$\Psi = \begin{cases} |z| & \text{si} \quad |z| \geq \sigma, \\ \frac{z^2+\sigma^2}{2\sigma} & \text{si} \quad |z| < \sigma. \end{cases} \tag{1.52}$$

Le paramètre σ est le paramètre de correction entropique qui est proportionnel à un majorant de la vitesse maximale des ondes :

$$\sigma = \epsilon_\sigma (|\bar{u}| + |\bar{v}| + |\bar{w}| + \bar{a}) \tag{1.53}$$

Avec cette formulation, le flux de Roe reste toujours dissipatif aux points où une vitesse caractéristique s'annule. L'inconvénient de cette correction est qu'elle introduit un paramètre ϵ_σ

(qui doit être ajusté en fonction des conditions d'écoulement).

Extension au second ordre d'approximation en espace La procédure permettant une extension au second ordre d'approximation en espace est la méthode MUSCL de van Leer [VL79] qui repose sur une reconstruction linéaire par morceaux de la solution approchée. Cette méthode est indépendante du schéma décentré mis en œuvre et est appliquée en maillage structuré successivement dans les trois directions de maillage. L'extrapolation est réalisée sur les variables primitives P en bijection avec les variables conservatives W :

$$P = (\rho, u, v, w, p)^T . \tag{1.54}$$

On écrit les formules dans le cas particulier d'un maillage structuré, en particularisant la direction curviligne décrite par l'indice de maillage i, les indices j et k étant fixés. On repère une cellule de maillage par les indices (i, j, k). Deux états, respectivement notés $P^g_{i+\frac{1}{2},j,k}$ et $P^d_{i+\frac{1}{2},j,k}$, sont définis pour chaque facette $\left(i+\frac{1}{2}, j, k\right)$, de la manière suivante :

$$P^g_{i+\frac{1}{2},j,k} = P_{i,j,k} + \frac{1}{2} slopi(i, j, k), \tag{1.55}$$

$$P^d_{i+\frac{1}{2},j,k} = P_{i,j,k} - \frac{1}{2} slopi(i+1, j, k), \tag{1.56}$$

où *slopi* désigne la pente calculée dans la direction de maillage i de la manière suivante :

$$slopi(i, j, k) = \phi\left(P_{i,j,k} - P_{i-1,j,k}, P_{i+1,j,k} - P_{i,j,k}\right). \tag{1.57}$$

La fonction ϕ désigne un limiteur de pente. Le limiteur de van Albada [VAVLR82] est le seul limiteur différentiable implémenté dans le logiciel *elsA*. C'est celui que nous utiliserons dans la suite. Son expression est donnée par :

$$\phi_{va}(a, b) = \frac{(b^2+\epsilon)a + (a^2+\epsilon)b}{a^2 + b^2 + 2\epsilon}, \tag{1.58}$$

avec ϵ *très petit* fixé. Les états $W^g_{i+\frac{1}{2},j,k}$ et $W^d_{i+\frac{1}{2},j,k}$ intervenant dans les formules de flux sont déduits des valeurs de $P^g_{i+\frac{1}{2},j,k}$ et $P^d_{i+\frac{1}{2},j,k}$.

Traitement des flux diffusifs La matrice Fd du flux diffusif fait intervenir le gradient de la vitesse et de la température, ainsi que des quantités turbulentes. Pour calculer ces gradients, on utilise la formule de Green :

$$\int_\Omega W, x d\Omega = \frac{1}{|\Omega|} \int_{\delta\Omega} W n_X d\epsilon.$$

Il est donc nécessaire d'établir un schéma de discrétisation de cette formule. Il est alors possible d'effectuer la discrétisation aux centres des cellules ou bien aux centres des interfaces. Cette dernière possibilité a l'inconvénient de requérir une plus force capacité en mémoire. En effet, il est alors nécessaire de stocker trois fois plus de champs pour un problème tridimensionnel (deux dans un cas bidimensionnel). Une formulation au centre des cellules est donc utilisée dans

la suite de nos travaux.

Dans ce type d'approche, la discrétisation la plus naturelle du flux visqueux consiste à évaluer la densité de flux visqueux au centre de chaque cellule, à moyenner ces densités pour obtenir une évaluation aux interfaces et à intégrer cette moyenne sur les interfaces. On utilise ici une correction, qui consiste à modifier la moyenne des gradients à l'interface, dans la formule du tenseur évalué aux interfaces, dans une direction privilégiée. Cette approche est désignée comme une approche "gradient au centre corrigé aux interfaces".

Considérons un problème bidimensionnel, sur maillage cartésien régulier orienté selon les axes du repère et supposons le champ de viscosité constant. Le flux visqueux laminaire dans la direction x pour les équations de conservation de la quantité de mouvement est donné par :

$$Fd, x = \mu \left(\begin{bmatrix} \frac{4}{3}u,x - \frac{2}{3}v,y \\ u,y + v,x \end{bmatrix} \right). \tag{1.59}$$

Lorsque l'on n'utilise pas la correction exposée ci-avant, la discrétisation est :

$$Fd, x_{i+\frac{1}{2}} = \mu \left(\begin{bmatrix} \frac{4}{3}\frac{u_{i+2}+u_{i+1}-u_i-u_{i-1}}{4\Delta x} - \frac{2}{3}\frac{v_{i,j+1}-v_{i,j-1}+v_{i+1,j+1}-v_{i+1,j-1}}{4\Delta y} \\ \frac{u_{i,j+1}-u_{i,j-1}+u_{i+1,j+1}-u_{i+1,j-1}}{4\Delta y} + \frac{v_{i+1,j}-v_{i-1,j}+v_{i+2,j}-v_{i,j}}{4\Delta x} \end{bmatrix} \right). \tag{1.60}$$

Si l'on utilise la correction, la discrétisation devient :

$$Fd, x_{i+\frac{1}{2}} = \mu \left(\begin{bmatrix} \frac{4}{3}\frac{u_{i+1}-u_i}{\Delta x} - \frac{2}{3}\frac{v_{i,j+1}-v_{i,j-1}+v_{i+1,j+1}-v_{i+1,j-1}}{4\Delta y} \\ \frac{u_{i,j+1}-u_{i,j-1}+u_{i+1,j+1}-u_{i+1,j-1}}{4\Delta y} + \frac{v_{i+1}-v_i}{\Delta x} \end{bmatrix} \right). \tag{1.61}$$

Intégration temporelle L'efficacité de l'utilisation de schémas implicites pour la résolution des équations RaNS compressibles a été largement démontrée. Parmi ces méthodes, le schéma linéarisé *backward-Euler*, introduit par Beam et Warming [BW76], est couramment utilisé :

$$(I + \frac{\Delta t}{Vol}\frac{dR^{APP}}{dW})(W^{n+1} - W^n) = -\frac{\Delta t}{Vol}R(W^n), \tag{1.62}$$

où $\frac{dR^{APP}}{dW}$ est une approximation de $\frac{dR}{dW}$.

Dans les applications de cette thèse, le système linéaire correspondant est résolu de manière approchée par une méthode de relaxation LU [Pet96].

Discrétisation des équations de la turbulence La discrétisation des équations de la turbulence est donnée ci-dessous dans le cas du modèle de Spalart-Allmaras. Les termes convectifs sont discrétisés à l'aide d'un schéma de Roe d'ordre 1, étendu à l'ordre 2 par une approche de type VTD :

$$\begin{aligned} f_{i+\frac{1}{2}}^{RoeTVD2} = \left\langle \frac{1}{2}\left[\rho\tilde{\nu}_i V_i + \rho\tilde{\nu}_{i+1}V_{i+1}\right], S_{i+\frac{1}{2}} \right\rangle - \frac{1}{2}\Psi\left\langle \left(\tilde{\nu}_{i+\frac{1}{2}}, S_{i+\frac{1}{2}}\right)\right\rangle \left[(\rho\tilde{\nu}_{i+1} - \rho\tilde{\nu}_i) - \right. \\ \left. minmod\left(\rho\tilde{\nu}_{i+2} - \rho\tilde{\nu}_{i+1}, \rho\tilde{\nu}_{i+1} - \rho\tilde{\nu}_i, \rho\tilde{\nu}_i - \rho\tilde{\nu}_{i-1}\right)\right], \end{aligned} \tag{1.63}$$

où *minmod* vaut 0 si deux de ses arguments ont un signe différent, et l'argument de plus faible valeur absolue dans le cas contraire.

En ce qui concerne les termes visqueux, ils sont traités par une approche de type "gradient au

centre corrigé aux interfaces" déjà présentée ci-dessus. Enfin, les termes sources, et les gradients qui en font partie, sont calculés aux centres des cellules.

1.4 Extraction des coefficients aérodynamiques

La résolution des équations de la mécanique des fluides sur le maillage volumique autour de la forme solide déformée permet d'obtenir le champ aérodynamique correspondant à cette géométrie. A partir de ces informations, il est possible de calculer la valeur des coefficients aérodynamiques. On distingue deux approches de calculs :

- le calcul des efforts "champ proche" (traînée de pression L_p, portance de pression D_p et traînée de frottement D_f) s'effectue par une intégration sur le contour de la forme solide. C'est la méthode la plus simple à mettre en œuvre, mais la traînée est alors très affectée par la dissipation numérique ;
- on préfèrera donc utiliser pour ce terme une approche champ lointain. Elle consiste à analyser la traînée source par source (traînée de choc D_w, traînée induite D_i et traînée visqueuse D_v) et permet de réduire les effets de la dissipation numérique [Des03].

Une fois les efforts obtenus, on peut calculer les coefficients aérodynamiques adimensionnés. On distingue souvent le coefficient de portance CL_p et les coefficients de traînée champ proche CD_{nf} et champ lointain CD_{ff} calculés à partir des formules suivantes :

$$CL_p = \frac{L_p}{\frac{1}{2}\rho_\infty U_\infty^2 S_{ref}}, \tag{1.64}$$

$$CD_{nf} = \frac{D_p + D_f}{\frac{1}{2}\rho_\infty U_\infty^2 S_{ref}}, \tag{1.65}$$

$$CD_{ff} = \frac{D_v + D_w + D_i}{\frac{1}{2}\rho_\infty U_\infty^2 S_{ref}}, \tag{1.66}$$

où ρ_∞ désigne la masse volumique à l'infini, U_∞ le module du vecteur vitesse à l'infini et S_{ref} la surface de référence de la configuration.

Les fonctions objectifs des optimisations de forme que nous mènerons par la suite seront définies à partir de l'un ou plusieurs de ces coefficients. Dans le cadre de notre travail, le calcul des coefficients aérodynamiques, ainsi que de leurs dérivées partielles par rapport aux points du maillage volumique et au champ aérodynamique, est effectué par les logiciels *FFD41* et *FFD72* [Des03] [EMVGB06] développés au Département d'Aérodynamique APpliquée (DAAP) de l'Onera.

1.5 Résolution du problème d'optimisation

Dans les sections précédentes, nous avons décrit les différentes étapes de l'évaluation de la qualité aérodynamique d'une forme solide discrète et paramétrée. On introduit maintenant les notations utilisées par la suite.

La forme solide est donc caractérisée par n_f paramètres de forme α, définis sur un domaine de conception D_α. Le maillage surfacique correspondant est noté $S(\alpha)$ et ses déformations par

rapport à la forme initiale sont propagées pour définir un nouveau maillage de volume $X(\alpha)$ autour du solide. On note W, vecteur de dimension n_a, le champ aérodynamique. Les équations discrétisées de la mécanique des fluides sont notées $R(X, W) = 0$. Les fonctions $S : \alpha \to S(\alpha)$, $X : \alpha \to X(\alpha)$, et $R : (X, W) \to R(W, X)$ sont supposées de classe C^1. Si l'on suppose de plus la géométrie X fixée à $X_i = X(\alpha_i)$, alors les équations $R(X_i, W)$ constituent un système non linéaire en W de n_a équations et n_a inconnues. Considérons les vecteurs X_i et W_i tels que :

$$R(W_i, X_i) = 0 \quad \text{et} \quad det\left[\frac{\partial R}{\partial W}(W_i, X_i)\right] \neq 0. \tag{1.67}$$

Alors le théorème des fonctions implicites permet de définir W comme une fonction C^1 de X sur un voisinage de X_i, puis par régularité de X_i, comme une fonction C^1 de α sur un voisinage de α_i. Cela signifie qu'il y a localement unicité de l'écoulement associé au maillage X_i. Bien qu'il soit connu que certaines configurations particulières peuvent conduire à des solutions aérodynamiques multiples[2], on supposera désormais cette propriété vraie sur tout le domaine de conception D_α.

On peut alors noter $W(\alpha)$ et réécrire les équations de la mécanique des fluides :

$$R\left(W(\alpha), X(\alpha)\right) = 0. \tag{1.68}$$

Notons J la fonction objectif que l'on souhaite minimiser :

$$J(\alpha) = j(W(\alpha), X(\alpha)).$$

Dans un cadre général, il est également possible de définir des contraintes du problème d'optimisation, dépendant également du champ aérodynamique et de la géométrie. Ainsi, par exemple, la portance pour un problème de minimisation de traînée sous contrainte de portance. On notera G_k les n_c contraintes que l'on définira comme précédemment :

$$G_k(\alpha) = g_k(W(\alpha), X(\alpha)) \leq 0 \qquad \forall k \in \{1, n_c\}.$$

Le problème d'optimisation de forme s'écrit alors :

$$\text{Trouver } \alpha^* \in D_\alpha \text{ qui minimise } J(\alpha) \tag{1.69}$$

$$\text{sous les contraintes } g_k(W(\alpha), X(\alpha)) \leq 0 \qquad \forall k \in \{1, n_c\}. \tag{1.70}$$

Il s'agit d'un problème de minimisation non linéaire sous contraintes, pour lequel la forme explicite des fonctions objectifs et contraintes n'est pas connue. Par conséquent, l'optimum doit être calculé par des méthodes itératives d'optimisation numérique. Comme le minimum est recherché sur l'ensemble de l'espace de définition des paramètres, une méthode d'optimisation globale doit être utilisée : on peut citer notamment les algorithmes génétiques [Hol75], les méthodes à essaims de particules [KE95] ou encore les méthodes d'évolution différentielle [SP95].

Ces méthodes sont rarement utilisables en pratique, en raison de besoins en ressources de

[2]par exemple, des tuyères surdétendues avec décollement

calcul considérables. On restreint donc souvent la recherche à un voisinage V_{α^0} du point initial. Le problème 1.69 devient :

$$\text{Trouver } \alpha^* \in V_{\alpha^0} \text{ qui minimise } J(\alpha) \tag{1.71}$$

$$\text{sous les contraintes } g_k(W(\alpha), X(\alpha)) \leq 0 \qquad \forall k \in \{1, n_c\}. \tag{1.72}$$

On peut alors utiliser une méthode d'optimisation locale, comme par exemple une méthode de descente, qui progresse par itérations en utilisant le gradient des fonctions objectifs et contraintes. Ce type d'approche est extrêmement utilisé en raison de sa vitesse rapide de convergence et, par conséquent, de ses exigences limitées en ressources de calcul. Un grand nombre d'algorithmes existent et il serait inapproprié de les décrire ici. On pourra par exemple se reporter à [BP10] pour la présentation de certains d'entre eux. Le calcul des gradients des fonctions aérodynamiques devient alors nécessaire. C'est un exercice complexe qui est l'objet de la prochaine section.

1.6 Calcul du gradient des fonctions d'intérêt par rapport aux paramètres de forme

Il s'agit ici de préciser le calcul du gradient des $(1+n_c)$ fonctions objectifs et contraintes, $J(\alpha)$ et $G_k(\alpha)$, par rapport aux n_f paramètres de forme, composantes du vecteur α. On présente trois des quatre méthodes classiquement utilisées :

- différences finies ;
- méthode de l'équation linéarisée discrète ;
- méthode de l'équation adjointe discrète.

Pour la quatrième méthode (méthode de l'équation adjointe continue), on invite le lecteur à se reporter au papier fondateur de Jameson [Jam88], qui présente les équations de la méthode adjointe continue, avec transformation de coordonnées, pour les équations d'Euler.

Ce papier est souvent cité comme première contribution au domaine du calcul de gradient pour l'optimisation de forme en aéronautique. Cependant, ce sujet se rattache à l'optimisation sur un espace de paramètres d'une fonction faisant intervenir la solution d'une équation différentielle (le paramètre étant susceptible de figurer et dans l'équation différentielle et dans la fonction à optimiser). Il est donc légitime de citer le livre fondateur du contrôle optimal [Lio71], mais aussi les travaux de Murat-Simon [MS76] et Pironneau [Pir84] sur l'optimisation de forme et l'article de Pironneau sur l'optimisation de forme en mécanique des fluides [Pir73].

Pour les approches discrètes, nous avons rappelé précédemment que, sous une hypothèse (signifiant qu'il y a localement unicité de l'écoulement associé à un maillage déterminé) qui permet l'application du théorème des fonctions implicites, le champ aérodynamique W peut être considéré comme une fonction de α. Le problème posé est donc :

$$\text{Différentier } J(\alpha) \text{ et } G_k(\alpha) \qquad \forall k \in \{1, n_c\} \tag{1.73}$$

$$\text{sous la contrainte } R(X(\alpha), W(\alpha)) = 0. \tag{1.74}$$

Les fonctions J et G_k étant des fonctions explicites de W et X, leurs dérivées par rapport au champ aérodynamique et au maillage sont accessibles. De même, la quantité $\frac{dX}{d\alpha}$ est calculée à partir des outils de maillage de surface et de volume présentés dans les sections 1 et 2 de ce chapitre ou par différences finies.

1.6.1 Calcul des gradients par différences finies

Cette méthode a été utilisée jusque durant les années 1990. Elle a tendance à disparaître au bénéfice des deux méthodes présentées ensuite (méthode de l'équation linéarisée et méthode de l'équation adjointe). Elle a en effet l'inconvénient, souvent rédhibitoire, de nécessiter n_f calculs d'écoulement (autant qu'il y a de paramètres de forme) pour une évaluation à l'ordre un des dérivées, soit la résolution de n_f problèmes non linéaires de taille n_a. Par opposition, les méthodes suivantes ne nécessiteront que la résolution de systèmes linéaires.

Soit $\delta\alpha_i$ l'incrément choisi pour les différences finies pour le paramètre de forme α_i. Les écoulements $W\left(\alpha + [0, 0, ..., \delta\alpha_i, ..., 0]\right)$ correspondant aux maillages $X\left(\alpha + [0, 0, ..., \delta\alpha_i, ..., 0]\right)$ sont calculés et les dérivées de fonctions sont évaluées par :

$$\frac{dJ}{d\alpha} \simeq \frac{j(W(\alpha + \delta\alpha_i), X(\alpha + \delta\alpha_i)) - j(W(\alpha), X(\alpha))}{\delta\alpha_i}. \tag{1.75}$$

Ceci s'étend facilement à une formule à l'ordre deux en $\delta\alpha_i$. Le nombre d'écoulements à calculer est alors multiplié par un facteur égal à l'ordre de précision. Pour certaines applications, le résultat est très sensible à l'amplitude de l'incrément, ce qui rend la méthode inapplicable en pratique. Cette question est discutée notamment par les auteurs de [Haf85]. Cette méthode est aussi ancienne que la simulation numérique en mécanique des fluides. On pourra se référer à [DR93] pour des exemples d'application et une bibliographie appropriée.

1.6.2 Méthode de l'équation linéarisée discrète

Cette méthode, ainsi que la suivante, ont été tout d'abord étudiées, dans le contexte de l'aérodynamique, par Shubin et Frank au début des années 1990 [SF91] [Shu91]. Sous les hypothèses faites précédemment, le résidu discret des équations de la mécanique est continûment dérivable. En dérivant les équations discrétisées de la mécanique, on obtient :

$$\frac{\partial R}{\partial W}\frac{dW}{d\alpha_i} = -(\frac{\partial R}{\partial X}\frac{dX}{d\alpha_i}) \qquad \forall i \in \{1, n_f\}. \tag{1.76}$$

Cette équation est appelée, dans le domaine de l'optimisation de formes, *équation linéarisée*. En la résolvant pour toutes les composantes de α, on accède à la dérivée partielle de l'écoulement par rapport aux variations de forme $\frac{dW}{d\alpha}$. On peut alors calculer les dérivées totales de l'objectif et des contraintes sous la forme directe :

$$\nabla_\alpha J(\alpha) = \frac{\partial j}{\partial X}\frac{dX}{d\alpha} + \frac{\partial j}{\partial W}\frac{dW}{d\alpha}, \tag{1.77}$$

$$\nabla_\alpha G_k(\alpha) = \frac{\partial g_k}{\partial X}\frac{dX}{d\alpha} + \frac{\partial g_k}{\partial W}\frac{dW}{d\alpha} \qquad \forall k \in \{1, n_c\}. \tag{1.78}$$

La méthode de l'équation linéarisée permet donc d'accéder aux $(n_c+1)\times n_f$ dérivées des fonctions par rapport aux paramètres de forme en résolvant n_f systèmes linéaires.

1.6.3 Méthode de l'équation adjointe discrète

Les équations du calcul des gradients par la méthode adjointe peuvent être obtenues par différents formalismes et différentes manipulations algébriques. La méthode la plus classique consiste à introduire un lagrangien associé à la fonction j :

$$L(W, X, \lambda) = j(W, X) + \lambda^T R(W, X). \tag{1.79}$$

L'adjonction d'un terme correspondant aux équations discrétisées de la mécanique permet de considérer W et X comme deux variables indépendantes de la fonction L. L'équation adjointe du problème de calcul de gradient est alors obtenue en imposant $\frac{dL}{dW} = 0$ soit :

$$\frac{\partial j}{\partial W} + \lambda^T \frac{\partial R}{\partial W} = 0. \tag{1.80}$$

La contrainte du problème $R(W, X) = 0$ équivaut à $\frac{dL}{d\lambda} = 0$. Enfin la valeur de $\frac{dJ}{d\alpha}$ est égale à celle de $\frac{dL}{d\alpha}$ (W et X dépendant de α, pas λ), lorsque les deux égalités précédentes sont vérifiées, soit :

$$R(W, X) = 0, \tag{1.81}$$

$$\frac{\partial j}{\partial W} + \lambda^T \frac{\partial R}{\partial W} = 0, \tag{1.82}$$

$$\nabla_\alpha j(\alpha) = \frac{\partial j}{\partial X}\frac{dX}{d\alpha} + \lambda^T(\frac{\partial R}{\partial X}\frac{dX}{d\alpha}). \tag{1.83}$$

La résolution de l'équation adjointe permet d'obtenir le vecteur adjoint λ. Le calcul de gradient s'effectue alors par la formule 1.83. La méthode de l'équation adjointe permet donc de calculer les $(n_c+1)\times n_f$ dérivées des fonctions par rapport aux paramètres de forme en résolvant autant d'équations adjointes qu'il y a de fonctions à dériver $(n_c + 1)$.

1.6.4 Comparaison des méthodes adjointes discrètes et continues.

La méthode adjointe continue a été introduite par Jameson [Jam88] et développée dans de très nombreuses publications. L'article de Giles et Pierce [GP00] constitue sans doute l'introduction la plus pédagogique à cette méthode. La difficulté mathématique de celle-ci est très supérieure à celle des deux précédentes. En effet, il s'agit de trouver tout d'abord les équations adjointes continues du problème d'optimisation continu, c'est-à-dire de trouver un problème aux dérivées partielles correspondant au problème aux dérivées partielles initial, tandis que, dans les paragraphes précédents, nous pouvions considérer directement le problème discret et manipuler des systèmes linéaires et non linéaires de taille n_a. C'est dans un deuxième temps seulement que le problème adjoint continu est discrétisé.

Le logiciel *elsA* implémente la méthode adjointe discrète. Nous ne présenterons donc pas en détail la méthode adjointe continue, mais donnons ci-dessous quelques éléments de comparaison

entre les deux méthodes :

- la méthode discrète garantit la cohérence entre une évaluation par différences finies et la valeur calculée en résolvant les équations adjointes discrètes. Cette cohérence n'est pas assurée par la méthode adjointe continue, même si un schéma de discrétisation identique est utilisée pour le calcul d'analyse et le calcul adjoint ;
- le coût de la phase explicite du calcul adjoint continu est en pratique inférieur à celui de la phase explicite de la méthode adjointe discrète. Il est difficile de discuter ce type de proposition en toute généralité. En pratique, pour les termes convectifs, le codage de la dérivée d'un flux convectif décentré classique comprend plusieurs fois plus d'opérations que l'application de ce flux au terme convectif de l'équation adjointe continue ;
- la méthode adjointe continue pose différents problèmes mathématiques que ne pose pas la méthode discrète :

  1. théoriquement, des termes de saut apparaissent dans les intégrations par parties si le domaine de calcul contient des lignes de discontinuité. Si l'on peut supposer que le résidu discret est une fonction C^1 du champ aérodynamique, la méthode discrète ne pose aucun problème de régularité : la variation avec les paramètres de forme de la représentation discrète des discontinuités est régulière ;
  2. pour certaines conditions aux limites, la définition des équations adjointes continues est délicate, voire impossible (il est impossible d'éliminer la dérivée de l'écoulement dans les calculs définissant le vecteur adjoint) ;

- le schéma de la méthode adjointe discrète issue de la linéarisation du schéma initial n'a pas forcément le caractère dissipatif de la discrétisation des équations continues ;
- la méthode adjointe continue ne permet pas de calculer le gradient d'une fonction intégrale posée sur une courbe située - même pour partie - à l'intérieur du domaine de calcul. C'est un problème pour l'analyse classique de la traînée, puisque la traînée de choc est une intégrale sur un contour situé dans le domaine de calcul autour du choc.

1.6.5 Linéarisation du modèle de turbulence pour les écoulements visqueux turbulents

Jusqu'alors, nous avons présenté de manière générale les méthodes adjointes et linéarisées discrètes en écrivant de manière formelle le résidu des équations pour la mécanique des fluides $R(W(\alpha), X(\alpha)) = 0$. Nous avons eu à préciser des propriétés de régularité des fonctions X, W et R, mais nous n'avons pas eu à préciser la modélisation retenue pour la mécanique des fluides.

Si la linéarisation des équations discrètes est réalisée sans approximation particulière lorsqu'une modélisation de fluide parfait est envisagée, il n'en va pas de même lorsque des écoulements turbulents de fluide visqueux sont considérés. La dérivation des méthodes discrètes est alors extrêmement ardue et diverses simplifications ont été envisagées dans la littérature. A l'opposé, quelques auteurs ont dérivé aussi précisément que possible les équations de Navier-Stokes moyennées associées à celle du modèle de turbulence. Ces auteurs discutent le bénéfice de la linéarisation exacte et l'effet des approximations. On pourra notamment se reporter à l'introduction de l'article de Lemoigne et al. [LMQ02], aux discussions des articles de Kim et al. [KKRL99]

[KKR01] [KKR02], à l'introduction de l'article de Nemec et al. [NZ01]. Trois principaux niveaux d'approximation ont été envisagés :

1. dans un très grand nombre d'articles, le coefficient de viscosité turbulente et le coefficient du flux de chaleur turbulent sont considérés constants lors des linéarisations. Ces approximations sont si fréquemment adoptées, qu'elles peuvent être considérées comme le traitement par défaut pour le calcul de gradient avec les équations RaNS. On pourra se reporter notamment aux études de Jameson [JPM97], Soemarwoto [Soe97] et à la thèse de Valentin [Val02]. Pour fixer les idées, supposons qu'un modèle à deux équations de transport soit utilisé et indiçons ensuite les vecteurs associés à une cellule par leur taille. Le système d'équations pour la mécanique des fluides est un système non linéaire à 7 champs, posé sur les cellules de calcul,

$$R_7(W_7(\alpha), X(\alpha)) = 0, \tag{1.84}$$

   dont les équations linéarisées et adjointes discrètes exactes s'écrivent :

$$\frac{\partial R_7}{\partial W_7}\frac{dW_7}{d\alpha_i} = -(\frac{\partial R_7}{\partial X}\frac{dX}{d\alpha_i}) \qquad (\frac{\partial R_7}{\partial W_7})^T\lambda_7 = -(\frac{\partial J}{\partial W_7})^T. \tag{1.85}$$

   Si l'on utilise l'approximation classique décrite ci-dessus, ces équations sont remplacées par des systèmes linéaires à 5 champs (et non plus 7) posés sur les cellules de calcul, obtenus en figeant μ_t et k_t :

$$\frac{\partial R_5}{\partial W_5}\frac{dW_5}{d\alpha_i} = -(\frac{\partial R_5}{\partial X}\frac{dX}{d\alpha_i}) \qquad (\frac{\partial R_5}{\partial W_5})^T\lambda_5 = -(\frac{\partial J}{\partial W_5})^T. \tag{1.86}$$

   Cela suppose que la fonction à dériver dépende exclusivement du champ moyen, ce qui est le cas en aérodynamique externe, où toutes les composantes de traînée, même la traînée de frottement, ne dépendent que du champ moyen. Pour la méthode linéarisée et la méthode adjointe discrète, les vecteurs à 5 champs $\frac{dW_5}{d\alpha_i}$ et λ_5 ne sont évidemment pas égaux aux 5 premiers champs des solutions des systèmes complets $\frac{dW_7}{d\alpha_i}$ et λ_7 ;

2. certains auteurs ont ajouté une hypothèse de couche mince à celle qui vient d'être décrite. Cette hypothèse consiste à ne prendre en compte dans la linéarisation du flux visqueux suivant une direction $\vec{n}$ que les dérivées suivant la direction $\vec{n}$. On pourra se reporter par exemple aux articles [PDP04] et [MCV04] ;

3. un nombre limité d'études présentent des calculs de gradients obtenus avec une linéarisation complète des équations de la mécanique (champ moyen et modèle de turbulence). On présente en détail ces publications ci-dessous. A notre connaissance, trois types de modèle de turbulence ont été linéarisés :
   - le modèle algébrique de Baldwin-Lomax. Il a été différencié par Le Moigne et al. [LMQ02] et par Kim et al. [KKRL99]. Le Moigne et al. présentent l'application de leur méthode à l'optimisation d'un profil d'aile. Kim et al. présentent des calculs de gradients de fonctions aérodynamiques (C_l, C_d, C_m) par rapport à un paramètre de forme (amplitude d'une bosse de Hicks-Henne). Les cas d'applications sont des écoulements transsoniques

et subsoniques autour du profil NACA0012 ;

- le modèle de Spalart-Allmaras. Il a été différencié par Giles et al. [GDM01] ainsi que par l'équipe d'Anderson, Nielsen et Bonhaus [AB97] [AB99] [NA99] [NA02]. Les articles [NA98] et [NA99] discutent précisément de la validité de l'approximation μ_t figé, mais aussi de l'effet d'une linéarisation approchée au premier ordre d'un flux convectif du second ordre, et des approximations possibles pour le calcul de la dérivée partielle du résidu par rapport à la géométrie. L'article [AB99] mentionne ce problème de précision des gradients, néanmoins il est centré sur la description de la méthode adjointe discrète, la paramétrisation géométrique, le calcul de la dérivée partielle du résidu par rapport à la géométrie et la présentation de cas tests (profil, profil volet). Il a enfin été linéarisé par Nemec et al. dont l'article [NZ01] présente des gradients très précis et des optimisations complètes pour des profils d'aile. Les méthodes adjointes discrète et linéarisée discrète sont considérées ;
- les modèles à deux équations de transport. Les modèles $k-\epsilon$ et $k-\omega$ SST et $k-\omega$ standard de Wilcox ont été différenciés, sans recours à un outil de différenciation automatique par Kim et al. [KKR01] [KKR02]. Dans la référence [KKR01], les auteurs comparent les gradients de fonction obtenus par différences finies et par méthode de gradient (adjointe et linéarisée discrète), pour des profils (RAE 2822 et NACA4412), avec et sans l'hypothèse de viscosité turbulente figée, pour les modèles $k-\epsilon$ et $k-\omega$ et $k-\omega$ SST. Les écarts les plus importants sont obtenus pour une configuration transsonique avec un choc fort. Dans [KKR02], les auteurs présentent des calculs de gradient et des optimisations de forme réalisées, avec les mêmes modèles de turbulence, sur le profil RAE 2822 (écoulement transsonique et écoulement subsonique) et la configuration NLR 173 (profil + volet, écoulement à faible Mach). Comme dans l'article [KKR01], ils observent de fortes erreurs sur les gradients pour la configuration transsonique avec choc fort. Pour les optimisations complètes, ils remarquent que les erreurs sur les gradients dues à l'hypothèse μ_t figée ralentissent (plus ou moins suivant les cas) le processus global d'optimisation de forme.

Signalons aussi les articles de Hou et al. [HMT$^+$] et Mohammadi [Moh97], qui s'intéressent principalement à la mise en œuvre d'outils de différenciation automatique (pour la méthode linéarisée discrète dans la première publication et pour la méthode adjointe discrète dans la seconde). Ils ont appliqué cette technique à toutes les équations d'un problème visqueux turbulent (modèle de Baldwin-Lomax dans [HMT$^+$] et modèle $k-\epsilon$ dans [Moh97]), mais ne discutent pas de l'utilité de linéariser l'ensemble des équations sans approximation. Mentionnons enfin l'article de Lund et al. [LMJ02], qui s'intéressent au calcul de gradient par la méthode discrète directe, pour un problème fluide-structure (écoulement incompressible bidimensionnel/élasticité). Ils ont procédé à la linéarisation des équations du modèle de turbulence (Baldwin-Lomax et $k-\omega$ de Wilcox) et donnent des indications sur la résolution des équations linéarisées du problème.

Conclusion

Dans ce premier chapitre, les éléments de base des méthodes d'optimisation de forme ont été présentés. Plusieurs méthodes de paramétrisation et de déformation de maillage volumique ont été décrites brièvement. Par ailleurs, les outils numériques utilisés pour notre travail (solveur *elsA* et outils d'extraction de traînée *FFD72*/*FFD41*) ont été détaillés. Enfin, après avoir évoqué rapidement les méthodes de résolution des problèmes d'optimisation de forme, nous avons consacré la dernière section à la description des méthodes de calcul de gradient et à leurs spécificités pour l'utilisation en aéronautique.

Le prochain chapitre est dédié à l'étude et à l'implémentation de modèles de substitution robustes et efficaces. Ces modèles seront utilisés par la suite pour la résolution des problèmes d'optimisation de forme posés dans ce chapitre.

Chapitre 2

Modèles de substitution

Introduction

La meilleure manière d'approcher une fonction continue à partir d'un échantillon discret de points d'évaluation est une question complexe qui occupe de nombreux chercheurs depuis plus d'une décennie. Selon [Ras91], elle intervient dans les recherches de mathématiciens arabes dès le dixième siècle, et est utilisée pour le calcul de tables trigonométriques ou astronomiques. A l'origine, on trouve bien sûr les méthodes d'interpolation linéaire, puis quadratique. Ces méthodes ont ensuite été étendues à l'interpolation polynomiale, au sens large, puis formalisées notamment avec les travaux de Weierstrass. L'interpolation polynomiale permet d'approcher rapidement des fonctions simples. Elle est à l'origine de la méthodologie des surfaces de réponse (*Response Surface Methodology* - RSM) qui a été proposée par Box et Wilson [BW51] dès le début des années 1950. Elle a également été utilisée pour des méthodes de régression : la fonction recherchée ne passe alors pas exactement par les points de l'échantillon. Mais l'augmentation du degré du polynôme, nécessaire pour approcher des fonctions plus complexes, génère des oscillations qui rendent rapidement cette méthode inopérante pour des applications pratiques. Les mathématiciens ont donc cherché par la suite des méthodes capables de mieux approcher des fonctions non linéaires quelconques.

Dès la fin des années 50, les neurologues McCulloch et Pitts publient un article posant les bases des réseaux de neurones [LMMP59]. Ils démontrent notamment que les réseaux de neurones peuvent approcher des fonctions logiques, algébriques ou symboliques complexes. Ces modèles franchissent un nouveau palier avec l'introduction du perceptron multi-couches, proposé simultanément par Le Cun [LC85] et par Rumelhart et Mc Clelland [RMC86]. Depuis, les réseaux de neurones ont connu un essor considérable et sont utilisés dans de nombreux domaines : on relève par exemple des applications en aéronautique [SRB94] [LB00], en météorologie, en traitement d'images ou encore en informatique, pour la reconnaissance optique de caractères. Si ces méthodes permettent d'approcher tout type de fonctions, elles ont l'inconvénient d'être coûteuses : l'apprentissage (c'est-à-dire la détermination des coefficients synaptiques de chaque neurone) est souvent effectué par une méthode de descente, qui peut être très longue à converger lorsque le nombre de neurones est important, ce qui s'avère parfois nécessaire pour l'approximation de fonctions complexes.

Depuis le début des années 80, trois classes de méthodes plus simples à construire ont été proposées et massivement utilisées : le krigeage (*Kriging*), initialement développé en géostatistique [Kri51] puis étendu à l'interpolation de fonctions mathématiques [SWMW89], la régression multivariée par splines adaptatives (*Multivariate adaptive regression splines* - MARS) proposée par Friedman [Fri91] et considérée comme une extension des modèles linéaires, et enfin les réseaux de fonctions à base radiale (*Radial Basis Function* - RBF), qui sont une forme particulière de réseaux de neurones pouvant être entraînés par la résolution d'un système linéaire [BL88].

Le nombre important de méthodes disponibles pour l'approximation de fonctions a conduit à mener des travaux de comparaison. Ainsi, Simpson et al. [SMKM98] et Giunta et al. [GWK] ont comparé le krigeage à des méthodes d'interpolation polynomiale dans le cadre d'applications aéronautiques. Jin et al. [JCS00] ont appliqué la régression polynomiale, le krigeage, la méthode MARS et la méthode RBF à une série de modèles mathématiques et à un problème d'optimisation de forme en automobile. Varadarajan et al. [VCP00] ont effectué le même type de travail avec les réseaux de neurones et la régression polynomiale pour des applications en thermodynamique. Enfin, Peter et al. [PMBP07] ont également évalué la qualité respective des modèles sur des problèmes d'optimisation d'aube de turbomachine en dimension deux. Les conclusions de ces études ne sont pas évidentes. Les échantillons de fonctions sont différents, les critères de comparaison difficiles à établir, et aucun modèle ne surpasse tous les autres, quel que soit le problème considéré. Néanmoins, il semble que le krigeage et les réseaux de fonctions à base radiale soient en général les méthodes les plus efficaces. Ce sont ces deux méthodes que nous avons retenues dans notre étude. Elles seront présentées plus en détails dans ce chapitre.

Au milieu des années 90, une nouvelle méthode a été proposée ; elle est absente des comparaisons présentées ci-dessus. En raison d'un cadre théorique puissant, les Machines à Vecteur de Support [Vap95] (*Support Vector Machine* - SVM) semblent très prometteuses. De nombreux efforts de recherche ont été consacrés au développement de cette méthode ces dernières années, mais les applications à des problèmes physiques semblent peu nombreuses. En effet, les méthodes SVM sont complexes et encore coûteuses à mettre en place, car elles requièrent la résolution d'un problème d'optimisation quadratique. Nous avons donc décidé de nous intéresser également à cette technique de régression.

Dans ce chapitre, les trois méthodes retenues, suite à l'étude bibliographique sont présentées dans les trois premières sections. On parlera de modèles de substitution (ou métamodèles) car ces méthodes sont souvent utilisées pour remplacer un modèle numérique complexe à évaluer. Dans la quatrième section, la mise en place d'une méthode de réglage automatique des paramètres internes des modèles sera explicitée. On verra que ces paramètres influencent fortement les performances des méthodes. Notre approche est ensuite validée par des applications à des fonctions mathématiques.

La précision d'un modèle dépend bien évidemment du nombre de points disponibles pour sa construction. Plus l'échantillon d'apprentissage est fourni, plus le modèle dispose d'informations sur la fonction à approcher et plus il est capable de la simuler précisément. Mais, pour les applications de cette thèse, les évaluations exactes de la fonction d'intérêt sont coûteuses à effectuer, puisqu'elles nécessitent la résolution d'un système d'équations non linéaires discrètes issu de la mécanique des fluides. Par conséquent, on peut chercher à fournir plus d'informations

en chaque point de l'échantillon. Ainsi, on peut notamment envisager de construire le modèle à partir des valeurs de la fonction complétées par des valeurs de sa dérivée. Cette approche est particulièrement intéressante pour les applications où la dérivée est disponible pour un temps de calcul modéré, ce qui est le cas en aérodynamique depuis le développement de la méthode adjointe. Dans la section 2.5 de ce chapitre, nous proposons donc de modifier chaque modèle afin de prendre en compte les dérivées en chaque point de l'échantillon. Enfin, en fin de chapitre, nous comparons les modèles sans dérivée et les modèles avec dérivées pour la reconstruction d'une fonction mathématique.

Introduisons quelques notations utilisées pour la présentation des méthodes. Les modèles de substitution ont donc pour objectif d'approcher une fonction quelconque f de $\mathcal{D} \subset \mathbb{R}^d$ dans $\mathbb{R}$ en un point x^* en utilisant un échantillon d'apprentissage, c'est-à-dire un ensemble d'observations E de N couples $\{(x^i, f(x^i)) = (x^i, y^i) \quad i \in \{1, N\}\}$ et un ensemble de paramètres P. Le schéma 2.1 résume ces notations.

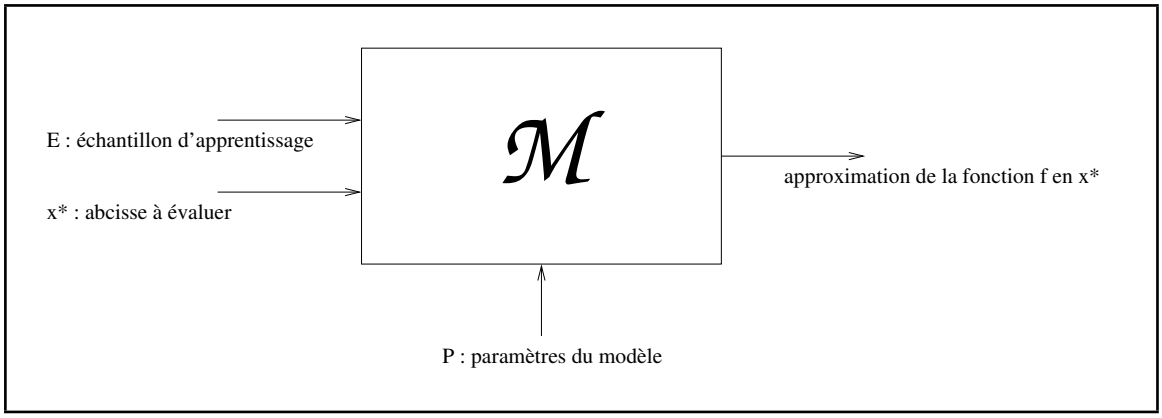


FIG. 2.1 – Représentation schématique d'un modèle

2.1 Krigeage

Le krigeage (*Kriging*) a été introduit par l'ingénieur des mines sud-africain Daniel G. Krige lors de son travail d'estimation des gisements miniers dans le massif de Witwatersrand en Afrique du Sud. Dans les années 50, il a ainsi développé une série de méthodes statistiques empiriques afin de déterminer la distribution spatiale de minerais à partir d'un ensemble de forages [Kri51]. Son travail a ensuite été formalisé mathématiquement par Matheron à l'école des Mines de Paris [Mat62], qui fonda le domaine de la géostatistique. Parallèlement, des idées similaires ont été développées par Gandin, en URSS, dans le domaine de la météorologie [Gan63] - le krigeage est alors appelé *méthode d'interpolation optimale*. Il est aussi connu sous le nom de *méthode d'interpolation de Gauss-Markov* en océanographie.

C'est à la fin des années 80 que Sacks et al. [SWMW89] proposent d'utiliser cette méthode pour l'approximation de fonctions issues de problèmes physiques. Depuis, l'utilisation de ce modèle s'est fortement développée et de nombreuses variantes ont été proposées par Currin et al [CMMY91]. La différenciation du modèle a été proposée par Koehler [Koe98].

2.1.1 Description mathématique

Le krigeage est basé sur l'hypothèse issue de la géostatistique que le champ à interpoler se déploie dans l'espace ou dans le temps avec une apparence de continuité. Il prend donc en chaque point de l'espace une valeur définie, mais, du fait de la complexité de ses propriétés, ne peut, en général, pas être formulé mathématiquement. La variable ainsi considérée est dite **régionalisée**. C'est Matheron qui introduit cette notion : *"Plus généralement, nous dirons d'un phénomène qui se déploie dans l'espace et y manifeste une certaine structure, qu'il est régionalisé. Les sciences de la terre, entre autres, en fournissent de nombreux exemples. Si $f(x)$ désigne la valeur au point x d'une caractéristique f de ce phénomène, nous dirons que $f(x)$ est une variable régionalisée ..."* [Mat70]. Plus loin, il dote la variable généralisée de *"deux aspects contradictoires"*, un aspect aléatoire et un aspect structuré, que nous qualifierons de déterministe.

Définition 2.1.1.1 (Variable régionalisée)**.** *On peut ainsi qualifier une variable régionalisée $f(x)$ par la somme d'une quantité déterministe $\mu(x)$ et d'une quantité aléatoire $\epsilon(x)$ d'espérance nulle et de structure de dépendance connue :*

$$f(x) = \mu(x) + \epsilon(x).$$

La méthode repose sur le principe suivant : en général, la valeur d'une variable en un point est corrélée aux valeurs des points voisins. On peut donc estimer la valeur de f en un point inconnu x^* par une combinaison linéaire des valeurs observées.

$$\hat{f}(x^*) = \sum_{i=1}^{N} \lambda^i(x^*) f(x^i).$$

On cherche alors à formuler le meilleur prédicteur linéaire non biaisé (*Best Linear Unbiased Predictor* - BLUP).

Définition 2.1.1.2 (Best Linear Unbiased Predictor)**.** *Un prédicteur $\hat{f}(x)$ d'une variable aléatoire $f(x)$ est le meilleur prédicteur linéaire non biaisé si :*

- *il est sans biais :*
$$E(f(x) - \hat{f}(x)) = 0;$$
- *il minimise l'erreur quadratique moyenne $E(f(x) - \hat{f}(x)^2) = Var(f(x) - \hat{f}(x))$.*

Il s'agit donc de trouver l'estimateur qui minimise

$$\begin{aligned}
\sigma_k^2(x^*) &= Var(\hat{f}(x^*) - f(x^*)) \\
&= Var(\hat{f}(x^*)) + Var(f(x^*)) - 2Cov(\hat{f}(x^*), f(x^*)) \\
&= Var(\sum_{i=1}^{N} \lambda^i(x^*) y^i) + Var(f(x^*)) - 2Cov(\sum_{i=1}^{N} \lambda^i(x^*) y^i, f(x^*)) \\
&= \sum_{i=1}^{N} \sum_{j=1}^{N} Cov(\lambda^i(x^*) y^i, \lambda^j(x^*) y^j) + Var(f(x^*)) - 2 \sum_{i=1}^{N} \lambda^i(x^*) Cov(y^i, f(x^*)).
\end{aligned} \tag{2.1}$$

et qui est sans biais :

$$E[\hat{f}(x^*) - f(x^*)] = \sum_{i=1}^{N} \lambda^i(x^*)\mu(x^i) - \mu(x^*) = 0. \tag{2.2}$$

2.1.2 Modélisation de la structure de dépendance

En pratique, on ne connaît presque jamais la structure de dépendance de $f(x)$. On doit donc la modéliser en utilisant une fonction de covariance. Plusieurs fonctions ont été proposées pour cette modélisation dans la littérature [Kle09]. Mais, dans la plupart des études sur le krigeage, et à la suite des travaux de Sacks et al. [SWMW89], c'est une fonction gaussienne qui est utilisée :

$$\phi(x, x^{'}) = Cov(f(x), f(x^{'})) = \exp\left[-\frac{1}{2}\sum_{k=1}^{d}(\frac{x_k - x_k^{'}}{\zeta_k})^2\right],$$

où $\zeta = (\zeta_k)_{k=1,d}$ représente le rayon de la fonction. Une fois que la fonction de covariance est fixée, la méthode peut être appliquée à des fonctions déterministes.

Il reste à calculer la valeur optimale de ζ. A cet effet, la méthode la plus utilisée semble être la méthode du maximum de vraisemblance [CMMY91] : elle consiste à utiliser le cadre statistique du krigeage pour déterminer la valeur de ζ qui maximise la fonction de vraisemblance. Cette fonction est fortement multimodale et possède de longues plages de valeurs égales [WR87]. Elle est donc complexe à optimiser, ce qui rend coûteux le calcul des paramètres du krigeage par cette méthode. Cette optimisation est souvent réalisée par une méthode de recuit simulé [KGV83] (voir par exemple [SKMM01]) mais d'autres méthodes numériques (*Quasi-Newton*, *Pattern Search*) ont également été envisagées [GRMS05].

Une alternative au maximum de vraisemblance est d'utiliser la validation croisée [LM68] pour estimer la qualité du modèle. Cette méthode permet d'approcher l'erreur de généralisation à partir de l'échantillon d'apprentissage. Les deux approches ont été comparées par Wahba [Wah85] et Stein [Ste90] : la méthode de validation croisée semble la plus appropriée pour un nombre relativement restreint de points d'observation ($N = 64$ dans les travaux de Wahba), alors que la maximisation de la vraisemblance semble surpasser la validation croisée, lorsque l'échantillon atteint des dimensions importantes ($N = 128$). En outre, la méthode de validation croisée possède l'avantage de ne pas dépendre du cadre statistique du krigeage, et notamment de l'hypothèse d'une distribution gaussienne des données.

Le nombre d'observations qui seront utilisées par la suite pour la construction de nos modèles peut aller de quelques dizaines à quelques centaines. Par conséquent, aucune des deux méthodes ne s'impose a priori. Une approche basée sur la validation croisée a finalement été utilisée ici afin d'éviter une forte dépendance du modèle au cadre statistique. Elle sera présentée à la section 2.4.

2.1.3 Les types de krigeage

Différentes implémentations du krigeage ont été proposées par Matheron. Elles se différencient par la manière de modéliser le terme déterministe $\mu(x)$. Ainsi, on peut distinguer :

– le krigeage simple, pour lequel il est supposé constant et connu a priori ;
– le krigeage ordinaire, pour lequel il est supposé constant mais inconnu ;
– le krigeage universel, pour lequel il est supposé polynomial.

Depuis, de nouvelles formes de krigeage ont été proposées, comme le krigeage "aveugle" [JHS08] (*Blind kriging*) qui est une variante du krigeage universel. Les comparaisons qui ont été effectuées entre ces différentes approches ont montré que le surcoût de complexité imposé par le krigeage universel ne s'accompagnait pas d'une amélioration significative des performances [Kle09]. Par conséquent, le krigeage ordinaire est de loin le plus utilisé dans la littérature. C'est celui qui a été implémenté dans ce travail.

2.1.4 Krigeage ordinaire

On suppose donc ici l'espérance de f inconnue mais constante :

$$\mu(x) = \mu.$$

Alors la condition d'approximation sans biais devient :

$$\sum_{i=1}^{N} \lambda^i(x^*)\mu - \mu = 0 \Leftrightarrow \sum_{i=1}^{N} \lambda^i(x^*) = 1.$$

Par l'introduction d'un multiplicateur de Lagrange ν, le problème à résoudre (équations 2.1 et 2.2) se ramène à la minimisation du lagrangien :

$$\mathcal{L}_{x^*}(\lambda,\nu) = \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda^i(x^*)\lambda^j(x^*)\phi(x^i,x^j) + Var(f(x^*)) - 2\sum_{i=1}^{N} \lambda^i(x^*)\phi(x^i,x^*) + 2\nu(\sum_{i=1}^{N} \lambda^i(x^*) - 1).$$

En annulant le gradient par rapport à chaque composante λ^j, on obtient le système suivant :

$$\begin{aligned} 2\sum_{i=1}^{N} \lambda^i(x^*)\phi(x^i,x^j) - 2\phi(x^j,x^*) + 2\nu &= 0 \qquad \forall j \in \{1,N\} \\ \sum_{i=1}^{N} \lambda^i(x^*) &= 1. \end{aligned}$$

Pour une estimation de la fonction à une abscisse quelconque x^*, les poids λ^i s'obtiennent donc par résolution du système

$$\underbrace{\begin{bmatrix} \phi(x^1,x^1) & \cdots & \phi(x^1,x^N) & 1 \\ \vdots & & \vdots & \vdots \\ \phi(x^N,x^1) & \cdots & \phi(x^N,x^N) & 1 \\ 1 & \cdots & 1 & 0 \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} \lambda^1 \\ \vdots \\ \lambda^N \\ \nu \end{bmatrix}}_{\Gamma} = \underbrace{\begin{bmatrix} \phi(x^1,x^*) \\ \vdots \\ \phi(x^N,x^*) \\ 1 \end{bmatrix}}_{\phi_{x^*}}.$$

Et la valeur de l'estimation s'obtient trivialement par :

$$\hat{f}(x^*) = \sum_{i=1}^{N} \lambda^i(x^*) y^i.$$

2.1.5 Implémentation efficace

Pour un calcul sur un nombre important d'abscisses x^*, il est plus efficace de n'effectuer qu'une seule inversion de système linéaire, et donc de calculer des poids différents $\hat{\lambda}^i$, correspondant à la décomposition dans la base des $\phi(x^i, x^*)$.

Ainsi, le calcul de ces poids ne dépend plus du point auquel on effectue l'évaluation, mais seulement de l'échantillon d'apprentissage. Il requiert l'inversion du système linéaire suivant :

$$\underbrace{\begin{bmatrix} \phi(x^1,x^1) & \cdots & \phi(x^1,x^N) & 1 \\ \vdots & & \vdots & \vdots \\ \phi(x^N,x^1) & \cdots & \phi(x^N,x^N) & 1 \\ 1 & \cdots & 1 & 0 \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} \hat{\lambda}^1 \\ \vdots \\ \hat{\lambda}^N \\ \hat{\lambda}^{N+1} \end{bmatrix}}_{\Lambda} = \underbrace{\begin{bmatrix} y^1 \\ \vdots \\ y^N \\ 0 \end{bmatrix}}_{Y}.$$

Alors la valeur du métamodèle en tout point x^* s'obtient par :

$$\hat{f}(x^*) = \sum_{i=1}^{N} \hat{\lambda}^i \phi(x^i, x^*) + \hat{\lambda}^{N+1} = \phi_{x^*}^T \Lambda = \phi_{x^*}^T \Phi^{-1} Y. \tag{2.3}$$

2.2 Réseau de fonctions à base radiale

Les fonctions à base radiale sont une technique d'approximation des fonctions à plusieurs variables par une combinaison linéaire de fonctions de base à une seule variable [ABR64]. Ces fonctions de base sont "radiales" (elles prennent en entrée la distance entre un point central et le point courant) et peuvent donc être utilisées quelle que soit la dimension de l'espace. Elles ont été introduites par Hardy [Har71] (fonctions multiquadriques), Powell [Pow81] (fonctions gaussiennes) ou encore Duchon [Duc76] (fonctions TPS) pour l'interpolation de fonctions de plusieurs variables.

Broomhead et Lowe ont proposé d'utiliser ce type de fonction dans des réseaux de neurones [BL88]. Depuis, et notamment avec les contributions de Moody et Darken [MD89], les réseaux de fonctions à base radiale (*Radial Basis Function Network* - RBF) se sont généralisés. Ils sont devenus des alternatives peu coûteuses aux réseaux de neurones classiques, pour lesquels le coût d'entraînement du modèle est souvent rédhibitoire. Une présentation très détaillée de ces modèles et des problématiques connexes a été effectuée par Orr [Orr99]. On pourra se reporter à cette publication pour plus de précisions.

2.2.1 Description mathématique

Les réseaux RBF sont des réseaux de neurones à deux couches : une couche cachée de N_c cellules qui appliquent une fonction d'activation radiale (la fonction d'activation de la cellule i sera notée ϕ^i par la suite) et une couche de sortie, qui est composée de neurones linéaires. La sortie du réseau se calcule donc par :

$$\hat{f}(x) = \sum_{i=1}^{N_c} w^i \phi^i(x, c^i).$$

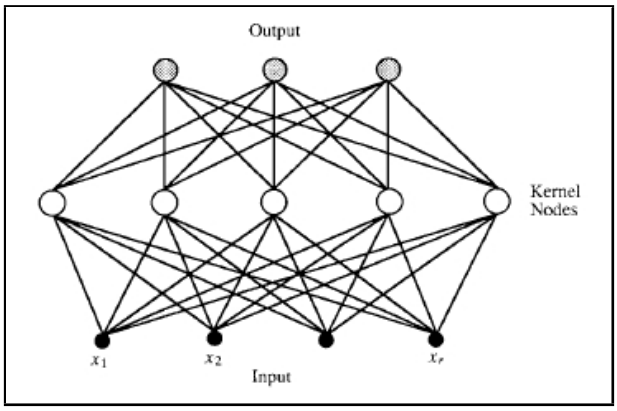


FIG. 2.2 – Réseaux de fonctions à base radiale (extrait de [PS91])

En utilisant les valeurs de la fonction en chacun des N points de l'échantillon d'apprentissage, on obtient

$$\forall j \in \{1, N\}, \quad f(x^j) = \sum_{i=1}^{N_c} w^i \phi^i(x^j, c^i).$$

2.2.2 Choix des centres

Le choix du nombre de neurones de la couche cachée et de l'emplacement du centre c^i de chaque fonction d'activation ϕ^i est un premier facteur de qualité du modèle. Il s'agit d'un problème non linéaire complexe qui n'a pas de solution optimale [CI01]. Si la détermination a priori du nombre de centres et de leurs emplacements est l'approche la plus courante, on trouve également dans la littérature des études dans lesquelles les centres sont déterminés dynamiquement par des méthodes de descente et des critères statistiques. On pourra se référer à [CI01] pour une bibliographie des différentes méthodes.

Dans le cadre de notre travail, et pour des raisons de simplicité et de rapidité de construction du modèle, on choisit d'utiliser autant de neurones dans la couche cachée qu'il y a de points dans l'échantillon d'apprentissage ($N_c = N$). De plus, on fait coïncider les centres c^i des fonctions d'activation avec les points de l'échantillon. Alors l'ensemble des coefficients w^i s'obtient par résolution du système linéaire carré suivant :

$$\underbrace{\begin{bmatrix} \phi^1(x^1,x^1) & \cdots & \phi^N(x^1,x^N) \\ \vdots & & \vdots \\ \phi^1(x^N,x^1) & \cdots & \phi^N(x^N,x^N) \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} w^1 \\ \vdots \\ w^N \end{bmatrix}}_{\Lambda} = \underbrace{\begin{bmatrix} y^1 \\ \vdots \\ y^N \end{bmatrix}}_{Y}.$$

Une fois les coefficients w_i calculés, la valeur de l'estimation s'obtient trivialement :

$$\hat{f}(x^*) = \sum_{i=1}^{N} w^i \phi^i(x^*, x^i) = \phi_{x^*}^T \Lambda = \phi_{x^*}^T \Phi^{-1} Y.$$

2.2.3 Réglage des paramètres

Plusieurs fonctions radiales peuvent être utilisées pour la construction des réseaux RBF : les fonctions gaussiennes et les fonctions multiquadriques semblent les plus utilisées en pratique.

Suivant les recommandations de Orr [Orr99], on utilisera par la suite la fonction gaussienne :

$$\phi^s(x, x^{'}) = \exp\left[-\frac{1}{2}\sum_{k=1}^{d}(\frac{x_k - x_k^{'}}{\zeta_k^s})^2\right].$$

Reste à déterminer les Nd paramètres ζ_k^s. Différentes approches sont décrites dans les publications.

Si l'on considère les rayons de chaque fonction d'activation indépendants, on peut appliquer des règles empiriques de la littérature. On peut notamment citer la contribution de Moody et Darken [MD89] qui calculent le rayon d'une fonction d'activation à partir de la distance entre le centre correspondant et les n plus proches centres ou encore l'article de Verleysen et Hlavackova [VH96] qui utilisent une approche statistique.

On peut également décider de fixer égaux les rayons de chacune des fonctions d'activation [PS91] :

$$\zeta_k^s = \zeta_k \quad \forall s \in \{1, N\} \quad \forall k \in \{1, d\}.$$

Alors, dans le cas de l'approximation de fonctions d'une variable, il ne reste qu'à choisir un seul paramètre. A nouveau, on peut se tourner vers des méthodes empiriques. Par exemple, Haykin [Hay99] propose de fixer $\zeta = \frac{d_{max}}{\sqrt{2N}}$, où d_{max} représente la distance maximale entre deux centres. Beaucoup d'auteurs ont espéré qu'une valeur indépendante de la taille de l'échantillon ou de la nature du problème considéré puisse être déterminée. Rippa [Rip99] semble avoir mis fin à ces espoirs en concluant que la valeur optimale du rayon dépend à la fois de l'échantillon et du problème. Or les heuristiques présentées ci-dessus, si elles tiennent bien compte du nombre et de la distribution des points de l'échantillon, sont indépendantes de la fonction à approcher. Par conséquent, on propose ici de se tourner vers des méthodes plus complexes d'optimisation du modèle par rapport à la valeur de ses paramètres. Dans ce domaine, les travaux de Orr [Orr98] qui propose de parcourir un intervalle donné et d'évaluer pour chaque valeur de ζ un critère quantifiant la qualité du modèle, puis de Rippa [Rip99] qui préfère une méthode de Brent [Bre73] à un parcours exhaustif semblent faire référence. Comme la méthode de Brent nécessite un intervalle de départ, Praveen et Duvigneau [PD07] proposent l'utilisation d'une méthode à essaims particulaires (PSO [KE95]).

Ces méthodes ont été proposées pour des approximations de fonctions d'une variable réelle. Elles peuvent être généralisées facilement en fixant les rayons des fonctions d'activation dans le cas multidimensionnel comme isotrope. Chaque composante du rayon a alors la même valeur que l'on détermine par les méthodes ci-dessus : $\zeta = \zeta_k^s \quad \forall k \in \{1, d\}$. En pratique, on peut montrer que ce choix appauvrit la qualité du modèle, car il lui impose un même comportement dans chacune des directions de l'espace (voir par exemple les expériences de la section 2.4.6). Un choix anisotrope semble plus approprié. On doit alors déterminer autant de paramètres indépendants qu'il y a de dimensions d'espace. Dans ce cas, le parcours exhaustif ou la méthode de Brent deviennent inutilisables. Il faut donc se tourner vers des méthodes d'optimisation plus complexes. C'est une approche de ce type que nous avons choisie ici et qui sera présentée dans la section 2.4.

2.3 Les Machines à Vecteur de Support

Les Machines à Vecteur de Support (*Support Vector Machine* - SVM) ont été développées dans les années 90 par Vapnik [Vap95] à partir de la théorie de l'apprentissage de Vapnik et Chervonenkis [VC74]. Elles ont initialement été utilisées pour résoudre des problèmes de classification de données, puis ont été étendues pour l'estimation de fonction à partir d'un échantillon d'apprentissage. On parle alors de Régression à Vecteur de Support (*Support Vector Regression* - SVR).

Ces méthodes récentes ont fait l'objet d'intenses recherches ces quinze dernières années. Pour beaucoup d'observateurs, elles sont aujourd'hui les méthodes de régression les plus efficaces [MH99]. Des présentations très complètes en ont été faites par Smola [SS98] et par Gunn [Gun98].

2.3.1 Présentation de la méthode pour la régression linéaire

Formulation du problème

Afin d'en faciliter la compréhension, on présentera tout d'abord cette méthode pour la régression linéaire, avant de l'étendre au cas non linéaire. La méthode, initialement proposée pour la classification de données, est basée sur la recherche d'un hyperplan de séparation optimale. Appliquée à l'approximation d'une fonction, il s'agit de trouver la fonction linéaire $f(x) = \langle w, x\rangle + b$ la plus plate (c'est-à-dire celle qui minimise $\|w\|^2$ [Gun98]) qui approche l'ensemble des points de l'échantillon avec une précision ϵ donnée. Il s'agit donc de résoudre le problème d'optimisation suivant (L_2 SVR) :

$$\begin{aligned} \min_{w,b,\xi,\xi^*} \quad & \frac{1}{2}\|w\|^2 + C_0 \sum_{i=1}^{N} (\xi^i + \xi^{i*}) \\ \text{sous les contraintes (slc) :} \quad & \left\langle w, x^i \right\rangle + b - y^i \leq \epsilon_0 + \xi^i \quad \forall i \in \{1, N\} \\ & y^i - \left\langle w, x^i \right\rangle - b \leq \epsilon_0 + \xi^{i*} \quad \forall i \in \{1, N\} \\ & \xi^i, \xi^{i*} \geq 0 \quad \forall i \in \{1, N\}\,. \end{aligned} \tag{2.4}$$

La constante $C_0 > 0$ détermine un compromis entre le caractère lisse de la fonction d'approximation et sa faculté à respecter la tolérance maximale de ϵ_0 en chaque point de l'échantillon d'apprentissage. La formulation 2.4 est dite *ϵ-insensitive* car l'erreur sur les points de l'échantillon est prise en compte par une fonction de coût qui ne pénalise pas les erreurs inférieures à ϵ_0. Ainsi, comme cela est représenté sur la figure 2.3, seuls les points au delà de ϵ_0 contribuent, de manière linéaire, à la fonction de coût. Il faut noter qu'il est également possible de formuler la méthode SVR pour d'autres fonctions de coût (on pourra se reporter à [SS98] pour une revue des fonctions d'erreurs utilisées dans la littérature), mais cette approche semble la plus répandue.

Écriture du problème dual

Le principe de dualité de Lagrange peut être appliqué au problème 2.4 pour formuler un problème dual qu'il sera plus facile de résoudre. De plus, c'est ce problème dual qui permettra l'extension de la méthode à l'approximation de fonctions non linéaires. Dans un premier temps,

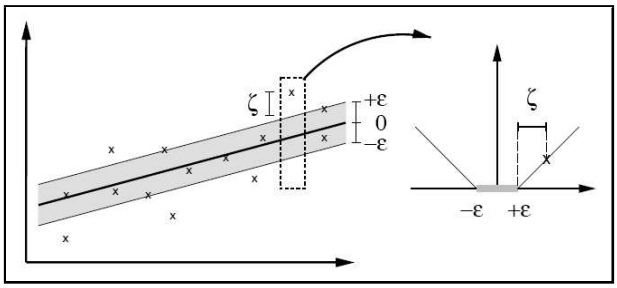


FIG. 2.3 – Fonction de coût ϵ-*insensitive* (extrait de [SS98])

on écrit donc le lagrangien correspondant à la formulation primale :

$$L(w,b,\xi,\xi^*,\alpha,\alpha^*) = \frac{1}{2}\|w\|^2 + C_0\sum_{i=1}^{N}(\xi^i+\xi^{i*}) - \sum_{i=1}^{N}(\mu^i\xi^i+\mu^{i*}\xi^{i*})$$
$$-\sum_{i=1}^{N}\alpha^i\left[\epsilon_0+\xi^i-\left\langle w,x^i\right\rangle-b+y^i\right]-\sum_{i=1}^{N}\alpha^{i*}\left[\epsilon_0+\xi^{i*}+\left\langle w,x^i\right\rangle+b-y^i\right].$$

Puis, on formule les conditions d'optimalité de Karush-Kuhn-Tucker (KKT) :

$$\begin{aligned}
\frac{\partial L}{\partial w}(w,b,\xi,\xi^*,\alpha,\alpha^*) &= w+\sum_{i=1}^{N}(\alpha^i-\alpha^{i*})x^i=0,\\
\frac{\partial L}{\partial b}(w,b,\xi,\xi^*,\alpha,\alpha^*) &= \sum_{i=1}^{N}(\alpha^i-\alpha^{i*})=0,\\
\frac{\partial L}{\partial \xi^i}(w,b,\xi,\xi^*,\alpha,\alpha^*) &= C_0-\alpha^i-\mu^i=0 \quad \forall i\in\{1,N\},\\
\frac{\partial L}{\partial \xi^{i*}}(w,b,\xi,\xi^*,\alpha,\alpha^*) &= C_0-\alpha^{i*}-\mu^{i*}=0 \quad \forall i\in\{1,N\}.
\end{aligned}$$

Alors :

$$\begin{aligned}
L(w,b,\xi,\xi^*,\alpha,\alpha^*) &= \frac{1}{2}\|w\|^2+\sum_{i=1}^{N}(\alpha^i+\mu^i)\xi^i+\sum_{i=1}^{N}(\alpha^{i*}+\mu^{i*})\xi^{i*}-\sum_{i=1}^{N}(\mu^i\xi^i+\mu^{i*}\xi^{i*})\\
&\quad-\sum_{i=1}^{N}\alpha^i\left[\epsilon_0+\xi^i-\left\langle w,x^i\right\rangle-b+y^i\right]-\sum_{i=1}^{N}\alpha^{i*}\left[\epsilon_0+\xi^{i*}+\left\langle w,x^i\right\rangle+b-y^i\right]\\
&= \frac{1}{2}\|w\|^2-\epsilon_0\sum_{i=1}^{N}(\alpha^i+\alpha^{i*})-\sum_{i=1}^{N}(\alpha^i-\alpha^{i*})y^i-b\sum_{i=1}^{N}(\alpha^i-\alpha^{i*})\\
&\quad-\sum_{i=1}^{N}(\alpha^{i*}-\alpha^i)\left\langle w,x^i\right\rangle.
\end{aligned}$$

Or :

$$-\sum_{i=1}^{N}(\alpha^{i*}-\alpha^i)\left\langle w,x^i\right\rangle = -\langle w,w\rangle = -\|w\|^2.$$

Le problème dual s'écrit alors :

$$\begin{aligned} \max_{\alpha,\alpha^*} \quad & -\frac{1}{2}\|w\|^2 - \epsilon_0 \sum_{i=1}^{N}(\alpha^i + \alpha^{i*}) - \sum_{i=1}^{N}(\alpha^i - \alpha^{i*})y^i \\ \text{slc} \quad : \quad & \sum_{i=1}^{N}(\alpha^i - \alpha^{i*}) = 0 \\ & \alpha^i, \alpha^{i*} \in [0, C_0] \quad \forall i \in \{1, N\}, \end{aligned}$$

soit :

$$\begin{aligned} \min_{\alpha,\alpha^*} \quad & \frac{1}{2}\left[\sum_{i,j=1}^{N}(\alpha^{i*} - \alpha^i)(\alpha^{j*} - \alpha^j)\left\langle x^i, x^j\right\rangle\right] + \epsilon_0 \sum_{i=1}^{N}(\alpha^i + \alpha^{i*}) + \sum_{i=1}^{N}(\alpha^i - \alpha^{i*})y^i \\ \text{slc} : \quad & \sum_{i=1}^{N}(\alpha^i - \alpha^{i*}) = 0 \\ & \alpha^i, \alpha^{i*} \in [0, C_0] \quad \forall i \in \{1, N\}. \end{aligned} \tag{2.5}$$

Si l'on définit la matrice de Gram,

$$\Phi = \begin{bmatrix} \langle x^1, x^1\rangle & \dots & \langle x^1, x^N\rangle \\ \vdots & & \vdots \\ \langle x^N, x^1\rangle & \dots & \langle x^N, x^N\rangle \end{bmatrix},$$

le problème 2.5 s'écrit alors :

$$\begin{aligned} \min_{\alpha,\alpha^*} \quad & \frac{1}{2}\begin{bmatrix}\alpha \\ \alpha^*\end{bmatrix}^T \begin{bmatrix}\Phi & -\Phi \\ -\Phi & \Phi\end{bmatrix}\begin{bmatrix}\alpha \\ \alpha^*\end{bmatrix} + \begin{bmatrix}\epsilon_0 + Y \\ \epsilon_0 - Y\end{bmatrix}^T \begin{bmatrix}\alpha \\ \alpha^*\end{bmatrix} \\ \text{slc} : \quad & \begin{bmatrix}\mathbb{1} \\ -\mathbb{1}\end{bmatrix}^T \begin{bmatrix}\alpha \\ \alpha^*\end{bmatrix} = 0 \\ & \alpha^i, \alpha^{i*} \in [0, C_0] \quad \forall i \in \{1, N\}. \end{aligned} \tag{2.6}$$

La résolution de ce problème quadratique en α^i, α^{i*} s'effectue à l'aide d'un algorithme de résolution des problèmes d'optimisation quadratique (QP). Ce sujet est discuté dans le paragraphe 2.3.3.

Valeur du modèle en un point quelconque

Une fois les vecteurs de coefficients α et α^* calculés, la valeur du modèle en x^* est obtenue par la formule :

$$\hat{f}(x^*) = \langle w, x^*\rangle + b = \sum_{i=1}^{N}(\alpha^{i*} - \alpha^i)\left\langle x^i, x^*\right\rangle + b, \tag{2.7}$$

où b, variable duale associée à la contrainte d'égalité, est fournie par le solveur quadratique.

2.3.2 Extension aux approximations non linéaires

Il s'agit maintenant d'étendre notre modèle à l'approximation d'une fonction non linéaire. L'approche classique est de projeter l'espace d'entrée $\mathcal{I}$ dans un espace de représentation $\mathcal{F}$ de

dimension beaucoup plus importante à l'aide d'une fonction de projection $\mathcal{P} : \mathcal{I} \to \mathcal{F}$. Une régression linéaire est alors effectuée dans ce nouvel espace en résolvant le problème suivant :

$$\begin{aligned} \min_{\alpha,\alpha^*} \quad & \frac{1}{2}\left[\sum_{i,j=1}^{N}(\alpha^{i*}-\alpha^i)(\alpha^{j*}-\alpha^j)\left\langle \mathcal{P}(x^i),\mathcal{P}(x^j)\right\rangle\right] + \epsilon_0\sum_{i=1}^{N}(\alpha^i+\alpha^{i*}) + \sum_{i=1}^{N}(\alpha^i-\alpha^{i*})y^i \\ \text{slc} : \quad & \sum_{i=1}^{N}(\alpha^i-\alpha^{i*}) = 0 \\ & \alpha^i,\alpha^{i*} \in [0,C_0] \quad \forall i \in \{1,N\}. \end{aligned} \tag{2.8}$$

Bien qu'il existe des situations pour lesquelles la fonction $\mathcal{P}$ peut être formulée de façon explicite, ce n'est pas possible dans le cas général. Mais on peut remarquer que la fonction de projection n'intervient que par des produits scalaires : on introduit donc une fonction noyau $\phi(x^i,x^j) = \langle \mathcal{P}(x^i),\mathcal{P}(x^j)\rangle$ qui simule le produit scalaire dans l'espace de représentation $\mathcal{F}$. On obtient alors le problème suivant :

$$\begin{aligned} \min_{\alpha,\alpha^*} \quad & \frac{1}{2}\left[\sum_{i,j=1}^{N}(\alpha^{i*}-\alpha^i)(\alpha^{j*}-\alpha^j)\phi(x^i,x^j)\right] + \epsilon_0\sum_{i=1}^{N}(\alpha^i+\alpha^{i*}) + \sum_{i=1}^{N}(\alpha^i-\alpha^{i*})y^i \\ \text{slc} : \quad & \sum_{i=1}^{N}(\alpha^i-\alpha^{i*}) = 0 \\ & \alpha^i,\alpha^{i*} \in [0,C_0] \quad \forall i \in \{1,N\}. \end{aligned} \tag{2.9}$$

Si, de plus, on définit à nouveau la matrice de Gram,

$$\Phi = \begin{bmatrix} \phi(x^1,x^1) & \dots & \phi(x^1,x^N) \\ \vdots & & \vdots \\ \phi(x^N,x^1) & \dots & \phi(x^N,x^N) \end{bmatrix},$$

alors, le problème à résoudre s'écrit

$$\begin{aligned} \min_{\alpha,\alpha^*} \quad & \frac{1}{2}\begin{bmatrix}\alpha \\ \alpha^*\end{bmatrix}^T\begin{bmatrix}\Phi & -\Phi \\ -\Phi & \Phi\end{bmatrix}\begin{bmatrix}\alpha \\ \alpha^*\end{bmatrix} + \begin{bmatrix}\epsilon_0+Y \\ \epsilon_0-Y\end{bmatrix}^T\begin{bmatrix}\alpha \\ \alpha^*\end{bmatrix} \\ \text{slc} : \quad & \begin{bmatrix}\mathbb{1} \\ -\mathbb{1}\end{bmatrix}^T\begin{bmatrix}\alpha \\ \alpha^*\end{bmatrix} = 0 \\ & \alpha^i,\alpha^{i*} \in [0,C_0] \quad \forall i \in \{1,N\}. \end{aligned} \tag{2.10}$$

Et la formule de régression en x^* devient :

$$\hat{f}(x^*) = \langle w,\mathcal{P}(x^*)\rangle + b = \sum_{i=1}^{N}(\alpha^{i*}-\alpha^i)\phi(x^i,x^*) + b. \tag{2.11}$$

2.3.3 Résolution du problème quadratique

Le problème 2.5 issu de la transformation du problème initial est un problème dit de programmation quadratique (*Quadratic Programming* - QP).

Définition 2.3.3.1 (Programmation quadratique)**.** *Un problème de programmation quadratique est un problème d'optimisation dont la fonction objectif est quadratique et les contraintes sont*

linéaires. On définit la forme nominale d'un tel problème sous la forme suivante (les notations sont propres à cette définition) :

$$\begin{aligned} \min_x \quad & f + c^T x + \frac{1}{2} x^T H x \\ \textit{avec les contraintes :} \quad & b \leq Ax \leq b + r \\ \textit{et :} \quad & l \leq x \leq u, \end{aligned}$$

où $H \in \mathbb{R}^{n \times n}$, $A \in \mathbb{R}^{m \times n}$, $x, c, l, u \in \mathbb{R}^n$, $b, r \in \mathbb{R}^m$ *et* $f \in \mathbb{R}$.

De plus, ce problème est convexe. En effet, comme la matrice Φ est définie positive par définition des noyaux [SS98], la matrice $\begin{bmatrix} \Phi & -\Phi \\ -\Phi & \Phi \end{bmatrix}$ est également définie positive. On cherche donc à résoudre un problème de programmation quadratique convexe. La méthode de référence pour résoudre ce type de problème a longtemps été la méthode de points intérieurs [Kar84] (*Interior Point Method* - IPM).

Les méthodes de points intérieurs ont été introduites dans le milieu des années 80 par un papier fondateur de Karmarkar [Kar84]. Elles sont alors utilisées pour la résolution de problèmes de programmation linéaire, c'est-à-dire de problèmes où les fonctions objectifs et contraintes sont linéaires. De nombreuses variantes de cet algorithme ont depuis été développées autour d'une même idée : la résolution simultanée de la forme primale et de la forme duale du problème jusqu'à la satisfaction des conditions d'optimalité (KKT). L'algorithme cherche donc un point de départ qui est à la fois faisable pour le problème primal et pour sa forme duale. A partir de ce point, appelé base, l'algorithme évolue de base en base en augmentant à chaque fois la satisfaction aux conditions complémentaires.

L'inconvénient de cette méthode est qu'elle manipule des vecteurs et des matrices de la dimension du problème à résoudre (deux fois le nombre de points de l'échantillon dans le cas des machines à vecteurs de support). Pour des problèmes de programmation quadratique convexe de très forte dimension, cette méthode devient inutilisable et il est alors indispensable de se tourner vers d'autres catégories de méthodes, qui transforment le problème en une succession de sous-problèmes de dimensions plus faibles. La plus connue de ces méthodes est la méthode d'optimisation minimale séquentielle [Pla98] (*Sequential Minimal Optimization* - SMO) introduite par Platt en 1998. D'autres algorithmes ont également été utilisés pour résoudre les problèmes QP : on peut notamment citer la librairie *MINOS*, développée à Stanford [MS83], basée sur une combinaison d'un algorithme de quasi-Newton et d'un algorithme de Gradient Réduit.

Dans le cadre de notre travail, la dimension des problèmes quadratiques à résoudre est deux fois le nombre de points qui seront utilisés pour construire les modèles. Comme l'évaluation d'un point de l'échantillon nécessite la résolution des équations de la mécanique des fluides, cette dimension ne sera jamais supérieure à quelques centaines. Par conséquent, la méthode qui va être utilisée pour la résolution du problème 2.5 est une méthode IPM qui est considérée comme la plus efficace pour des problèmes de dimension relativement faible [Van94]. La méthode LOQO, développée par Vanderbei [Van94], a donc été implémentée. Les détails de cet algorithme sont présentés en annexe A.

Utilisation spécifique pour la méthode SVR

En comparant la formulation duale utilisée pour l'application de la méthode des points intérieurs (Annexe A) et le problème initial de la méthode SVR, on peut obtenir directement à partir de la valeur des variables duales la valeur du paramètre b de la formule 2.11. Ainsi, on a :

$$b = y_{IPM}.$$

On note également pour la suite qu'il est possible de calculer le terme $\sum_{i=1}^{N}(\xi^i + \xi^{i*})$ de la méthode SVR à partir de la variable duale s de la méthode des points intérieurs :

$$\sum_{i=1}^{N}(\xi^i + \xi^{i*}) = s_{IPM}.$$

2.3.4 Choix de la fonction noyau

Une fonction est considérée comme un noyau admissible si elle satisfait la condition de Mercer (voir [SS98]). A partir de cette condition, plusieurs noyaux ont été présentés dans la littérature. On peut notamment citer les noyaux de polynômes, les noyaux RBF ou encore les noyaux de fonctions B-Spline. On trouvera une liste assez complète dans [SS98]. Une propriété intéressante des fonctions noyaux est qu'une combinaison linéaire de noyaux est un noyau [SS98]. Cette propriété a conduit par exemple Qiu et Lane [QL05] à utiliser une combinaison linéaire des noyaux classiques et à intégrer les coefficients comme paramètres du problème d'optimisation. Mais, dans la plupart des applications de la méthode SVR, le noyau gaussien est utilisé. On fera de même dans ce travail :

$$\phi(x, x') = \exp\left[-\frac{1}{2}\sum_{k=1}^{d}(\frac{x_k - x'_k}{\zeta_k})^2\right].$$

2.3.5 Réglage des paramètres

Facteur de pénalité C_0

Le paramètre C_0 détermine un compromis entre la complexité du modèle et les erreurs supérieures à ϵ_0 tolérées sur l'échantillon d'apprentissage. Il a été introduit pour permettre à la méthode SVR de fournir un résultat acceptable dans le cas où l'approximation précise de l'échantillon est impossible. Cherkassky et Ma [CM02], tout comme Mattera et Haykin [MH99], proposent de choisir C_0 égal à la taille de l'intervalle des y^i, tout en précisant qu'avec un bon choix du paramètre ϵ_0, la valeur de C_0 n'a que peu d'influence sur les performances de généralisation, tant qu'elle est suffisamment grande. Par ailleurs, Shawe-Taylor et Christianini [STC04] ont choisi de faire varier ce paramètre dans un intervalle donné et de déterminer le meilleur choix par validation croisée.

Largeur de la zone insensible ϵ_0

Le paramètre ϵ_0 détermine la largeur de l'intervalle autour des points de l'échantillon dans lequel l'approximation est considérée comme exacte. Ce paramètre rend la fonction produite par le modèle plus ou moins lisse, et influe sur la complexité du modèle construit. Un ϵ_0 fort nuit à la précision du modèle mais permet d'obtenir un modèle plus lisse. A contrario, réduire la largeur de la zone conduit à un modèle complexe, mais qui approche précisément la fonction sur l'échantillon d'apprentissage.

Pour de nombreux auteurs ([CM02], [SMSKR98], [Kwo01]), le choix de ce paramètre est lié au niveau de bruit en entrée. Ils proposent alors une méthode pour déterminer la valeur de ϵ_0 dans le cas où ce niveau est connu, en reliant linéairement ϵ_0 et la variance σ_k du bruit gaussien. Mattera et Haykin [MH99] proposent de choisir ce paramètre de telle sorte que le nombre de vecteurs de support (SV) du modèle alors produit soit d'environ 50% de la taille de l'échantillon d'apprentissage.

Le paramètre ζ du noyau gaussien

Le choix du paramètre ζ du noyau gaussien est important. Si ce paramètre est choisi trop petit, alors le modèle produit sera trop local : il n'aura aucun mal à coller à l'échantillon d'apprentissage, mais donnera de mauvais résultats pour les points qui en sont éloignés. Si, par contre, il est choisi trop grand, alors il y a un risque de surdéterminer le modèle, alors incapable d'approcher précisément l'échantillon d'apprentissage. Cristianini et al. [CCST99] proposent une méthode pour adapter dynamiquement la valeur de ce paramètre aux données par observation d'une borne sur l'erreur de généralisation. Wang et al. [WXLZ03] donnent une méthode de calcul du ζ optimal pour les problèmes de classification et démontrent l'existence d'un intervalle à l'intérieur duquel les performances de généralisation sont stables. Enfin, Yang et al. [YLL06] présentent une méthode de descente basée sur le principe de minimisation du risque pour le choix optimal du paramètre ζ. Tout comme pour les deux autres modèles, on propose ici de déterminer la valeur optimale de ζ par optimisation. C'est l'objet de la prochaine section.

2.4 Optimisation des paramètres des modèles

Le choix des paramètres des différents modèles est donc un problème complexe. Quelle que soit la méthode, il n'existe pas dans la littérature de règle empirique qui permettrait de choisir les bons paramètres indépendamment du problème considéré. A l'inverse, il a été montré qu'un mauvais choix de paramètres peut détériorer très fortement les performances d'un modèle (par exemple, pour les réseaux RBF, on peut se reporter aux expériences numériques de Praveen et Duvigneau [PD07] ou aux résultats de la section 2.4.6).

Pour chaque modèle, on a vu que plusieurs auteurs ont finalement proposé de déterminer la valeur optimale des paramètres par des algorithmes d'optimisation. C'est une approche de ce type que nous allons maintenant présenter. Elle nécessite de formuler un critère qui permet de quantifier, pour une valeur des paramètres, la qualité d'approximation du modèle.

2.4.1 Estimation de la qualité du modèle

La précision globale du modèle est donnée par l'erreur quadratique moyenne sur le domaine de travail.

Définition 2.4.1.1 (Erreur quadratique moyenne (MSE))**.** *L'erreur quadratique moyenne (Mean Square Error, MSE) entre la fonction exacte f et le modèle $\hat{f}$ sur le domaine de travail $\mathcal{D}$, notée $E(\hat{f})$, est définie par*

$$E(\hat{f}) = \frac{1}{|\mathcal{D}|} \int_{\mathcal{D}} \left[\hat{f}(x) - f(x)\right]^2 dx.$$

En pratique, la valeur de la fonction exacte n'étant connue qu'en un nombre fini de points, on s'intéressera plutôt à l'erreur de généralisation, qui est une évaluation de l'erreur quadratique moyenne sur un ensemble de points $\mathcal{E}_V$ n'ayant pas servi à la construction du modèle.

Définition 2.4.1.2 (Erreur de généralisation)**.** *L'erreur de généralisation entre la fonction exacte f et le modèle $\hat{f}$ sur l'ensemble de N_V points $\mathcal{E}_V$, notée $E(\hat{f})^{\mathcal{E}_V}$, est définie par*

$$E(\hat{f})^{\mathcal{E}_V} = \frac{1}{N_V} \sum_{i=1}^{N_V} (\hat{f}(x^i) - y^i)^2.$$

Pour les applications de cette thèse, le calcul d'une évaluation exacte y^i est très coûteux et nous n'avons bien évidemment pas intérêt à utiliser une base de validation distincte de la base d'apprentissage. Toutes les informations disponibles doivent être consacrées à la construction du modèle. Comment alors quantifier ses performances ?

Proposée par Lachenbruch et Mickey [LM68], la méthode de validation croisée, ou méthode *leave-k-out*, permet d'approcher l'erreur de généralisation du modèle uniquement à partir de la base d'apprentissage. Pour un entier k fixé, elle consiste à retirer k points au hasard de la base d'apprentissage, et à calculer l'erreur de généralisation sur ces points du modèle construit à partir de l'échantillon réduit. Afin de diminuer le caractère aléatoire de la méthode, on effectue N tirages successifs des points à retirer et on construit N modèles $\hat{f}_j$. L'erreur *leave-k-out* est alors définie comme la moyenne des N erreurs de généralisation ainsi calculées.

Définition 2.4.1.3 (Erreur leave-k-out)**.** *L'erreur leave-k-out du modèle $\hat{f}$ construit sur un échantillon d'apprentissage $\mathcal{E}$, notée $E(\hat{f})_{lko}$, est définie par*

$$E(\hat{f})_{lko} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{k} \sum_{j=1}^{k} \left[\hat{f}_i(x^j) - y^j\right]^2.$$

Barton et al. [BBMS02] ont montré que cette méthode permet d'obtenir des estimations correctes de l'erreur de généralisation, pour peu que le paramètre k soit bien choisi. Leur étude montre que la valeur $k = 1$ donne une bonne estimation pour les réseaux RBF, mais a tendance à sous-estimer l'erreur de généralisation dans le cas du krigeage. Pour autant, ce qui nous intéresse ici n'est pas d'obtenir la meilleure approximation de l'erreur de généralisation, mais bien une estimation qui permet de déterminer le choix optimal des paramètres. Dans la suite de notre étude, pour toutes les méthode étudiées, c'est donc l'erreur *leave-one-out* qui sera utilisée.

Définition 2.4.1.4 (Erreur leave-one-out)**.** *L'erreur leave-one-out du modèle $\hat{f}$ construit sur un échantillon d'apprentissage $\mathcal{E}$, notée $E(\hat{f})_{loo}$, est définie par*

$$E(\hat{f})_{loo} = \frac{1}{N} \sum_{i=1}^{N} \left[\hat{f}_i(x^i) - y^i \right]^2 .$$

L'erreur *leave-one-out* se calcule donc par la construction de N modèles sur des bases d'apprentissage de taille $N - 1$.

2.4.2 Formulation du problème

On note p le vecteur des paramètres à optimiser et p_{min} (respectivement p_{max}) la borne inférieure (resp. supérieure) de l'optimisation. On cherche donc à résoudre le problème suivant :

$$\begin{aligned} &\min_{p \in \mathcal{D}_p} && e_{loo}(p) \\ &\text{où} \quad : \quad \mathcal{D}_p = \bigcap_{k=1}^{d} \left[p_k^-, p_k^+ \right] . \end{aligned} \tag{2.12}$$

2.4.3 Intervalle d'optimisation

Il est nécessaire de déterminer un intervalle de recherche pour les paramètres optimaux. Afin de construire des modèles pouvant s'adapter à des types de problème très différents, nous avons choisi un intervalle large pour la valeur des paramètres. En généralisant les conseils de Büche et al. [BSK04] pour le krigeage, chaque composante du rayon des fonctions ϕ est donc recherchée dans l'intervalle $[p_k^- = 0.01, p_k^+ = 10] \quad \forall k \in \{1, d\}$.

2.4.4 Algorithme d'optimisation

La résolution du problème 2.12 n'est pas un exercice facile : en effet, comme on pourra le voir par la suite (voir la figure 2.14 de la section 2.4.6), la fonction $e_{loo}(p)$ est multimodale dans l'espace de définition des paramètres. Par conséquent, les méthodes d'optimisation locale ne peuvent déterminer les valeurs optimales des paramètres.

Il est donc nécessaire d'utiliser des algorithmes d'optimisation globale. Ces méthodes ont l'inconvénient de requérir de nombreuses évaluations de la fonction objectif afin d'identifier son minimum. C'est la raison pour laquelle de nombreuses études ont été menées ces dernières années pour améliorer les méthodes existantes ou en proposer de nouvelles qui soient à la fois robustes et rapides. A l'heure actuelle, on peut distinguer au moins trois algorithmes performants :

- les méthodes génétiques [Hol75] (*Genetic Algorithm [Hol75]* - GA) ;
- les méthodes à essaims de particules [KE95] (*Particle Swarm Optimization* - PSO) ;
- les méthodes d'évolution différentielle [SP95] (*Differential Evolution* - DE).

Ces trois méthodes ont été comparées par Vesterstrom et Thomsen [VT04] sur un ensemble de 34 problèmes mathématiques. Les résultats de leur étude semblent montrer que les méthodes d'évolution différentielle sont en moyenne les plus efficaces. Par la suite, on utilisera donc des méthodes de ce type, dès lors qu'un algorithme d'optimisation globale sera nécessaire. Bien

que cet algorithme dispose de bonnes performances, il reste vrai que l'optimisation globale du problème 2.12 nécessite un nombre conséquent d'évaluations de l'erreur *leave-one-out*. Or l'évaluation de cette quantité peut nécessiter des coûts de calcul importants, lorsque les modèles de substitution sont utilisés pour des problèmes de grandes dimensions. Il est donc nécessaire de mettre en place des méthodes efficaces pour l'évaluation de l'erreur *leave-one-out*. C'est l'objet de la prochaine section.

2.4.5 Évaluation efficace de l'erreur *leave-one-out*

Réseaux de fonctions à base radiale et krigeage

Rippa [Rip99] a proposé, dans le cadre de la méthode RBF, une méthode efficace de calcul de l'erreur *leave-one-out* ne nécessitant que la résolution d'un seul système linéaire de taille $[N \times N]$, au lieu de N systèmes linéaires de taille $[(N-1) \times (N-1)]$. L'erreur en un point i du domaine lorsque celui-ci est ôté de l'échantillon, se calcule alors par la formule suivante :

$$e_i = \frac{(\Phi^{-1} y)_i}{(\Phi^{-1})_{ii}},$$

d'où la formule d'évaluation de l'erreur *leave-one-out* :

$$e_{loo} = \frac{e^T e}{N}. \tag{2.13}$$

Cette méthode s'étend sans difficulté au krigeage. On trouvera en annexe B la démonstration de cette formule pour les réseaux RBF et pour le krigeage.

Machine à vecteurs de support

La question est plus complexe en ce qui concerne les machines à vecteurs de support. Ainsi, le modèle est toujours linéaire, mais le calcul des coefficients fait intervenir la résolution d'un problème d'optimisation quadratique, ce qui empêche d'appliquer à nouveau la méthode présentée par Rippa. A notre connaissance, aucune approche ne permet aujourd'hui de calculer analytiquement l'erreur *leave-one-out* d'un modèle SVR.

Il reste à choisir une autre méthode pour l'approximation de l'erreur de généralisation. En 1998, Vapnik [Vap98] a introduit plusieurs formulations de bornes de l'erreur de généralisation dans le cadre de l'utilisation d'une méthode SVR pour des problèmes de classification. En 2000, Vapnik et Chapelle [VC00] ont proposé une borne basée sur le concept de *Span*, qui semble plus précise que les autres bornes proposées précédemment. Chapelle et al. [CVBM02] proposent alors de calculer les paramètres optimaux par une minimisation par méthode de descente de ces bornes. De leur côté, et toujours à partir de ces estimations, Wu et al. [WTL09] présentent une métaoptimisation d'un modèle SVR par une méthode génétique. C'est l'article de Chang et Lin [CL05] qui dérive ces bornes pour la régression. On en donne ici deux, qui semblent les plus utilisées en pratique.

Borne *Rayon-Marge* **[Vap98]** La borne *Rayon-Marge* (*Radius-Margin bound*) de l'erreur *leave-one-out* $E(\hat{f})_{loo}$ du modèle $\hat{f}$ construit sur un échantillon d'apprentissage $\mathcal{E}$ est définie par :

$$E(\hat{f})_{loo} \leq \frac{1}{N}(4R^2 \sum_{i=1}^{N}(\alpha^i + \alpha^{i*}) + \sum_{i=1}^{N}(\xi^i + \xi^{i*})) + \epsilon,$$

où R^2 est la solution du problème quadratique suivant :

$$\begin{aligned} R^2 = \min_{\beta \in \mathbb{R}^N} \quad & \frac{1}{2}\beta^T \Phi \beta + \sum_{i=1}^{N} \beta^i \phi(x^i, x^i) \\ \text{slc} \quad : \quad & \sum_{i=1}^{N} \beta^i = 1 \\ & \beta^i \in [0, C_0] \quad \forall i \in \{1, N\}. \end{aligned}$$

Le calcul de cette borne requiert donc la résolution de deux problèmes de minimisation quadratique : le premier, de taille $(2N \times 2N)$, afin de déterminer la valeur des vecteurs de coefficients (α, α^*) et le second, de dimension $(N \times N)$, pour calculer la valeur du rayon R. On a montré que le terme $S_{\xi,\xi^*} = \sum_{i=1}^{N}(\xi^i + \xi^{i*})$ peut se calculer à partir d'une variable duale de la méthode de points intérieurs utilisée pour la résolution du problème quadratique (voir section 2.3.3).

Borne *Span* **[VC00]** La borne *Span* (*Span bound*) de l'erreur *leave-one-out* $E(\hat{f})_{loo}$ du modèle $\hat{f}$ construit sur un échantillon d'apprentissage $\mathcal{E}$ est définie par :

$$E(\hat{f})_{loo} \leq \frac{1}{N}(\sum_{t=1}^{N}(S_t^2(\alpha^t + \alpha^{t*})) + \sum_{t=1}^{N}(\xi^t + \xi^{t*})) + \epsilon_0,$$

où S_t^2 est la solution du problème quadratique suivant :

$$\begin{aligned} S_t^2 = \min_{\lambda \in \mathbb{R}^{\text{Card}(\mathcal{F}\setminus\{t\})}} \quad & \sum_{i,j \in \mathcal{F}\setminus\{t\}} \lambda^i \lambda^j \phi(x^i, x^i) - 2 \sum_{i \in \mathcal{F}\setminus\{t\}} \lambda^i \phi(x^t, x^i) + \phi(x^t, x^t) \\ \text{slc} \quad : \quad & \sum_{i \in \mathcal{F}\setminus\{t\}} \lambda^i = 1, \end{aligned}$$

avec $\mathcal{F} = \{i | 0 < \alpha^i + \alpha^{i*} < C_0\}$ est l'ensemble des indices des vecteurs de support non bornés (*Unbounded Support Vector* - USV).

Sous cette forme, le calcul de cette borne nécessite la résolution d'autant de problèmes de minimisation quadratique qu'il y a de vecteurs de support (c'est-à-dire, pour tous les t tels que $\alpha^t + \alpha^{t*} > 0$). En pratique, comme le problème ci-dessus possède uniquement une contrainte d'égalité, on peut formuler la solution de manière analytique [CVBM02].

En ajoutant un lagrangien μ et en utilisant la notion de dualité lagrangienne, S_t^2 peut se

calculer par résolution du problème suivant :

$$S_t^2 = \min_{\lambda} \max_{\mu} \left[\phi(x_t, x_t) - 2 \sum_{i \in \mathcal{F} \setminus \{t\}} \lambda_i \phi(x_i, x_t) + \sum_{i,j \in \mathcal{F} \setminus \{t\}} \lambda_i \lambda_j \phi(x_i, x_j) \right] + 2\mu \left[\sum_{i \in \mathcal{F} \setminus \{t\}} \lambda_i - 1 \right].$$

On note Φ_{USV} la matrice des $[\phi(x_i, x_j)]_{i,j \in \mathcal{F} \setminus \{t\}}$ et λ et ϕ_{USV} les vecteurs dont les composantes sont respectivement $[\lambda_i]_{i \in \mathcal{F} \setminus \{t\}}$ et $[\phi(x_i, x_t)]_{i \in \mathcal{F} \setminus \{t\}}$. On peut alors écrire le problème sous forme matricielle :

$$S_t^2 = \min_{\lambda} \max_{\mu} \left[\phi(x_t, x_t) - 2\lambda^T v + \lambda^T M \lambda \right] + 2\mu \left[\lambda^T e - 1 \right],$$

où $M = \begin{cases} \Phi_{USV} & \text{si} \quad t \notin \mathcal{F} \\ \Phi_{USV}^t & \text{si} \quad t \in \mathcal{F} \end{cases}$ avec Φ_{USV}^t qui désigne la matrice Φ_{USV} privée de sa t-ième ligne et de sa t-ième colonne, et $v = \begin{cases} \phi_{USV} & \text{si} \quad t \notin \mathcal{F} \\ \phi_{USV}^t & \text{si} \quad t \in \mathcal{F} \end{cases}$ avec ϕ_{USV}^t qui désigne le vecteur ϕ_{USV} privé de sa t-ième ligne.

Si on note $\tilde{\lambda} = \begin{pmatrix} \lambda \\ \mu \end{pmatrix}$, $\tilde{v} = \begin{pmatrix} v \\ 1 \end{pmatrix}$, $\tilde{M} = \begin{pmatrix} M & 1 \\ 1 & 0 \end{pmatrix}$, alors

$$S_t^2 = \min_{\lambda} \max_{\mu} \left[\phi(x_t, x_t) - 2\tilde{\lambda}^T \tilde{v} + \tilde{\lambda}^T \tilde{M} \tilde{\lambda} \right].$$

On peut chercher la valeur optimale de $\tilde{\lambda}$ en annulant la dérivée, ce qui conduit à $\tilde{\lambda^*} = \tilde{M}^{-1} \tilde{v}$ et

$$S_t^2 = \phi(x_t, x_t) - \tilde{v}^T \tilde{M} \tilde{v}.$$

Si $t \in \mathcal{F}$, on peut simplifier la formulation ci-dessus en utilisant la formule de Woodbury [Lut96]

$$\begin{pmatrix} A_1 & A^T \\ A & A_2 \end{pmatrix}^{-1} = \begin{pmatrix} B_1 & B^T \\ B & B_2 \end{pmatrix},$$

avec $B_1 = \left(A_1 - A A_2 A^T \right)^{-1}$. En appliquant cette formule avec $A_1 = \phi(x_t, x_t)$, $A = \tilde{v}$ et $A_2 = \tilde{\Phi_{USV}^t}$, et en notant u l'indice correspondant au point t dans la liste des vecteurs de support non bornés, on peut écrire :

$$S_t^2 = \frac{1}{\left(\tilde{\Phi_{USV}}^{-1} \right)_{uu}}.$$

La borne *Span* peut donc se calculer en résolvant un problème quadratique pour déterminer la liste et la valeur des coefficients des vecteurs supports, puis en inversant la matrice K_{USV}. Cette borne est donc moins coûteuse à évaluer que la borne *Rayon Marge*. Par ailleurs, elle semble approcher avec plus de précision la valeur de l'erreur *leave-one-out* [VC00]. Par conséquent, la détermination des paramètres pour le modèle SVR sera effectuée par la suite à l'aide de cette borne.

2.4.6 Validation de la méthode de réglage des paramètres

L'objet de cette section est de valider le processus d'ajustement des paramètres internes des modèles présentés ci-dessus. Pour cela, on présentera d'abord quelques résultats pour une fonction mathématique dans un espace unidimensionnel, puis bidimensionnel.

Dans un cas unidimensionnel

On utilise la fonction f_1 définie par :

$$f_1(x) = x(1-x)\sin 2\pi x.$$

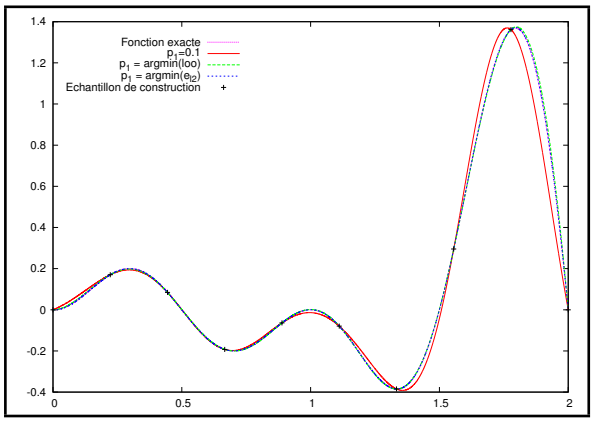


FIG. 2.4 – Approximation de f_1 par krigeage pour différents choix de paramètres

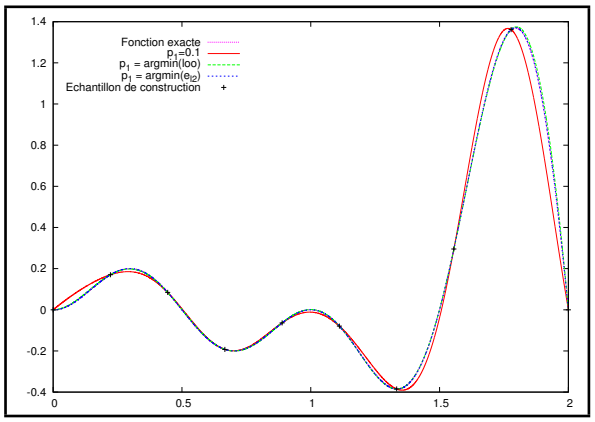


FIG. 2.5 – Approximation de f_1 par réseau RBF pour différents choix de paramètres

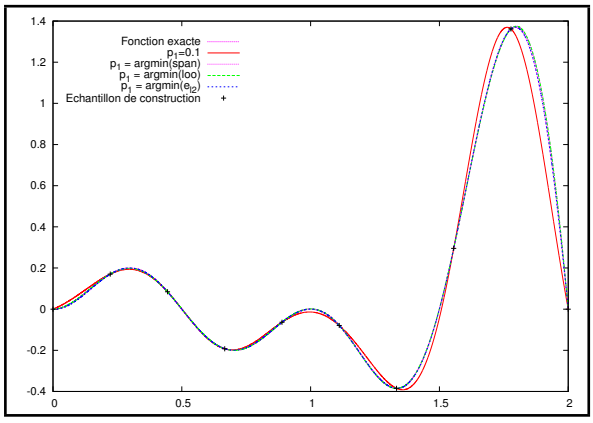


FIG. 2.6 – Approximation par la méthode SVR de f_1 pour différents choix de paramètres

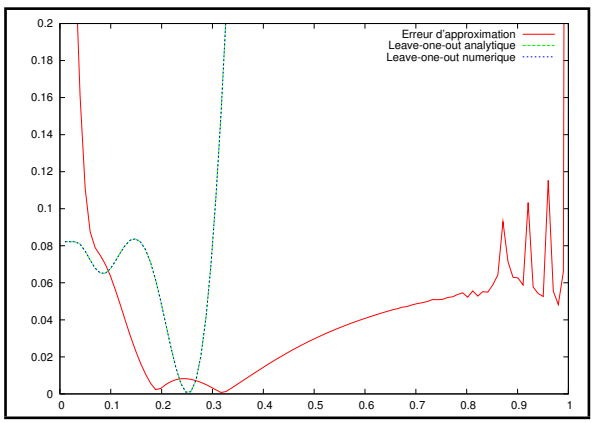


FIG. 2.7 – $E(\hat{f}_1)_{loo}$ et $E(\hat{f}_1)^{\mathcal{E}_V}$ par rapport à ζ_1 (krigeage)

L'échantillon utilisé pour la construction, composé de 10 points, est choisi selon une méthode de Hammersley, qui sera présentée dans le chapitre suivant. On cherche d'abord à mettre en évidence la nécessité de choisir soigneusement les paramètres d'un modèle. On approche donc la fonction f_1 par chacun des trois métamodèles présentés ci-dessus, et pour trois choix du

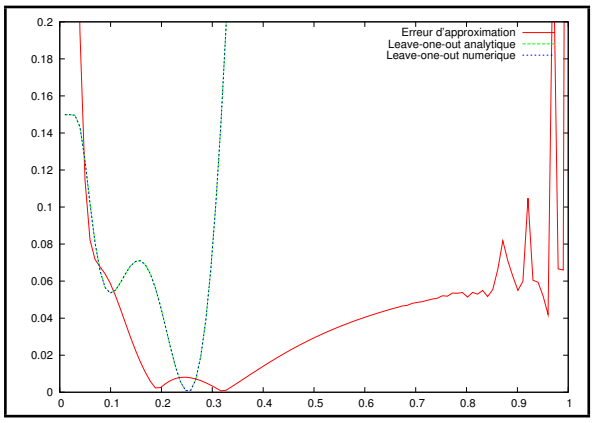


FIG. 2.8 – $E(\hat{f}_1)_{loo}$ et $E(\hat{f}_1)^{\mathcal{E}_V}$ par rapport à ζ_1 (RBF)

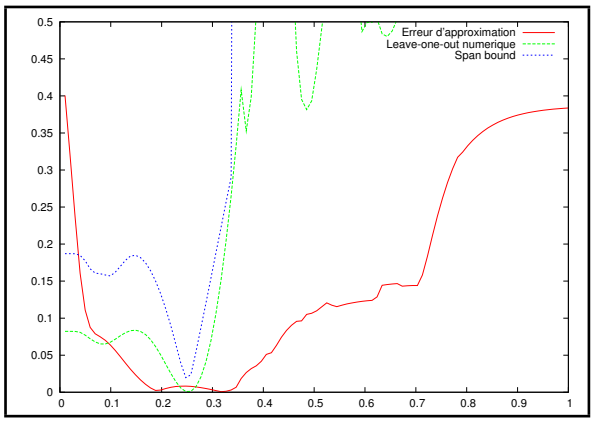


FIG. 2.9 – $E(\hat{f}_1)_{loo}$, $E(\hat{f}_1)^{\mathcal{E}_V}$ et borne *Span* par rapport à ζ_1 (SVR)

paramètre ζ : choix manuel du paramètre (on divise l'intervalle normalisé $[0, 1]$ par le nombre de points de l'échantillon), choix par minimisation de l'erreur *leave-one-out* ($E(\hat{f}_1)_{loo}$) et de l'erreur de généralisation ($E(\hat{f}_1)^{\mathcal{E}_V}$), calculée sur une discrétisation fine de l'espace de dessin à partir de la formulation analytique de la fonction. Les fonctions approchées sont présentées dans les figures 2.4, 2.5 et 2.6.

On souhaite montrer que la détermination de la valeur des paramètres par minimisation de l'erreur *leave-one-out* est un bon choix pour la précision des modèles. On représente donc, pour des valeurs du paramètre ζ comprises dans l'intervalle $[0, 1]$, cette erreur et l'erreur de généralisation de la fonction. Dans cet intervalle, l'erreur *leave-one-out* est une bonne estimation de l'erreur de généralisation (krigeage(2.7), réseau RBF(2.8), SVR(2.9)). De plus, la borne *Span* proposée pour l'optimisation du modèle SVR semble avoir un comportement similaire à l'erreur *leave-one-out*. Une minimisation de la borne donne, sur cet exemple, une valeur de paramètre équivalente à la minimisation de l'erreur *leave-one-out* (2.9).

Pour chacun de nos 3 métamodèles, les résultats de cette étude sont synthétisés dans les tableaux 2.1, 2.2 et 2.3.

TAB. 2.1 – Approximation de f_1 par krigeage pour différents choix de paramètres

Méthode de sélection de ζ_1	ζ_1	$E(\hat{f}_1)_{loo}$	$E(\hat{f}_1)^{\mathcal{E}_V}$
Sélection manuelle	$0,1$	$0,06785$	$0,06317$
Minimisation de $E(\hat{f}_1)_{loo}$	$0,2523$	$0,0003625$	$0,008130$
Minimisation de $E(\hat{f}_1)^{\mathcal{E}_V}$	$0,3169$		$0,0007223$

TAB. 2.2 – Approximation de f_1 par réseau RBF pour différents choix de paramètres

Méthode de sélection de ζ_1	ζ_1	$E(\hat{f}_1)_{loo}$	$E(\hat{f}_1)^{\mathcal{E}_V}$
Sélection manuelle	$0,1$	$0,05350$	$0,05834$
Minimisation de $E(\hat{f}_1)_{loo}$	$0,2476$	$0,0008689$	$0,008108$
Minimisation de $E(\hat{f}_1)^{\mathcal{E}_V}$	$0,3209$		$0,0003579$

TAB. 2.3 – Approximation de f_1 par modèle SVR pour différents choix de paramètres

Méthode de sélection de ζ_1	ζ_1	$E(\hat{f}_1)_{loo}$	$E(\hat{f}_1)^{\mathcal{E}_V}$
Sélection manuelle	0.1	0.06785	0.06317
Minimisation de la borne *Span*	0.2515	0.0003821	0.008160
Minimisation de $E(\hat{f}_1)_{loo}$	0.2523	0.0003630	0.008132
Minimisation de $E(\hat{f}_1)^{\mathcal{E}_V}$	0.3199		0.0005252

Dans un cas bidimensionnel

On considère la fonction de Rosenbrock, qui est une fonction de test classique pour les méthodes d'optimisation. Elle est définie, dans un espace de dimension d, par la formule suivante :

$$f_2(x) = \sum_{k=1}^{d-1} \left[(1 - x_k)^2 + 100(x_{k+1} - x_k^2)^2 \right].$$

Les figures 2.11, 2.12 et 2.13 représentent le modèle construit par krigeage à partir d'un échantillon de 10 points choisis selon la méthode de Hammersley, pour trois choix de paramètres : choix manuel, optimisation de l'erreur *leave-one-out* dans un espace de dimension un, et optimisation complète de chacune des composantes de ζ.

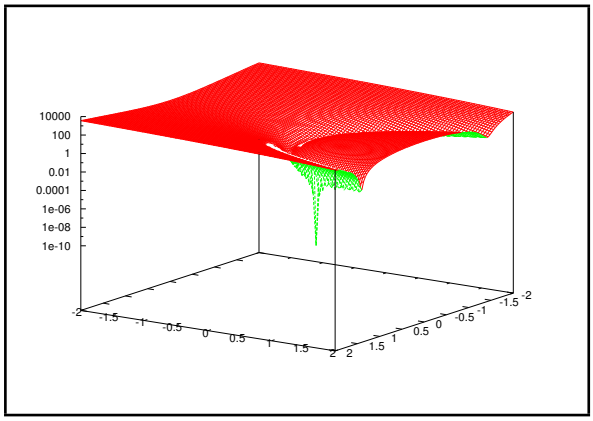


FIG. 2.10 – Fonction de Rosenbrock 2D sur $[-2, 2] \times [-2, 2]$

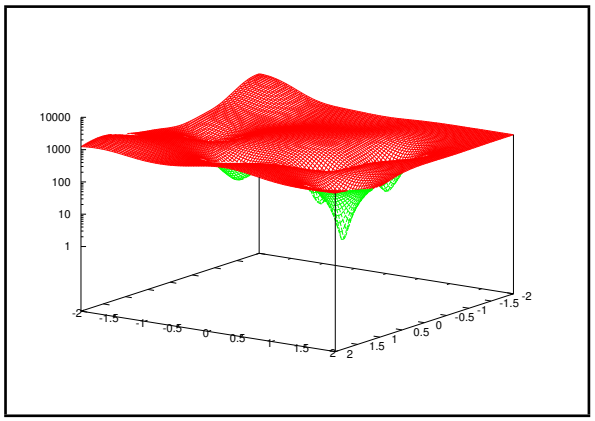


FIG. 2.11 – Approximation de f_2 par krigeage avec $\zeta_1 = \zeta_2 = 0.1$

Pour les deux premiers choix, le modèle de substitution est incapable de capter la vallée qui traverse la fonction au centre de l'espace de dessin. Seule l'optimisation complète du paramètre ζ permet d'approcher correctement la fonction. Les erreurs de généralisation calculées sur un échantillon de 100 points par dimension d'espace sont récapitulées dans le tableau 2.4 pour chaque choix de paramètres.

Cette étape est très coûteuse, puisqu'elle nécessite l'utilisation d'un algorithme d'optimisation globale. Ce type de méthode est indispensable, puisque l'erreur *leave-one-out* dans l'espace des paramètres (ζ_1, ζ_2) possède plusieurs minima locaux, comme on peut le voir sur la figure 2.14.

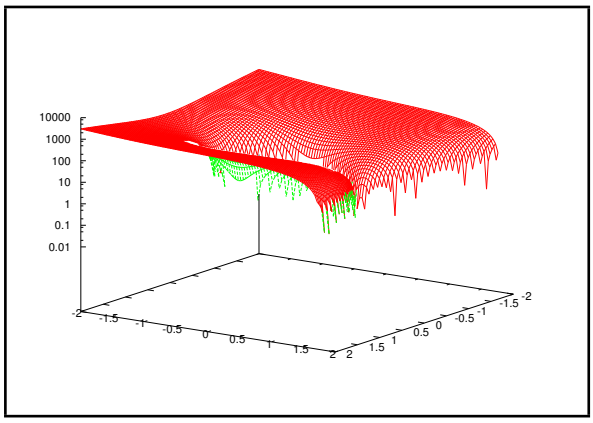


FIG. 2.12 – Approximation de f_2 par krigeage avec $(\zeta_1 = \zeta_2)$ déterminé par optimisation

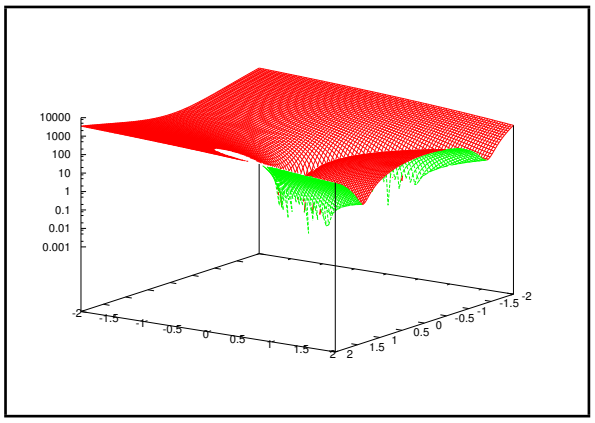


FIG. 2.13 – Approximation de f_2 par krigeage avec $(\zeta_1 \neq \zeta_2)$ déterminés par optimisation

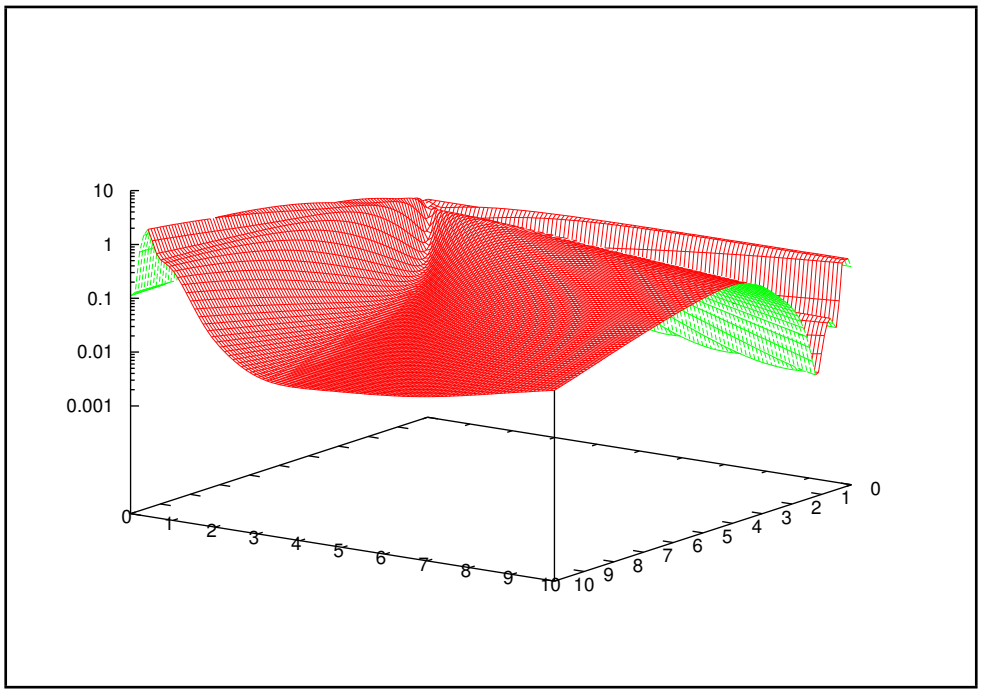


FIG. 2.14 – Erreur *leave-one-out* du krigeage dans l'espace des paramètres (ζ_1, ζ_2)

TAB. 2.4 – Approximation de f_2 par krigeage pour différents choix de paramètres

Méthode de sélection de ζ	(ζ_1, ζ_2)	$E(\hat{f}_2)_{loo}$	$E(\hat{f}_2)^{\mathcal{E}_V}$
Sélection manuelle	$(0, 1, 0, 1)$	0,1116	467,2
Optimisation isotropique	$(10, 10)$	0,03485	261,2
Optimisation anisotropique	$(1, 1655, 10)$	0,001797	13,66

2.5 Utilisation de l'information gradient dans les modèles

2.5.1 Le co-krigeage

L'estimation par méthode de krigeage n'utilise que les valeurs de la fonction aux points de l'échantillon. Cependant, il est possible que les gradients de la fonction soient également

disponibles. Il peut alors être intéressant de les utiliser pour renforcer la précision de l'estimation. La première apparition de cette approche dans la littérature semble être l'article de Morris et al. [MMY93] de 1993. Depuis, plusieurs termes qui recoupent des approches similaires sont utilisés pour désigner cette méthode :

- krigeage d'ordre 1 (*First Order Kriging*) dans les travaux de Lewis [Lew98], qui compare l'approximation de fonctions issues de problèmes d'optimisation de forme avec cette méthode et avec le krigeage classique et note un gain important dans la précision de l'approximation lorsque le gradient est utilisé en complément de la valeur des fonctions ;
- co-krigeage (*co-Kriging*) dans les articles de Chung et Alonso [CA02b] [CA02a], Laurenceau et Sagaut [LS08] puis Laurenceau et Meaux [LM08]. Dans les premiers cités, les modèles de krigeage et de co-krigeage sont utilisés pour construire un modèle de substitution, qui est ensuite exploré par un algorithme d'optimisation locale. Laurenceau et Sagaut [LS08] comparent quant à eux la précision des métamodèles construits par chacune des deux méthodes pour approcher la traînée d'onde d'un profil RAE2822 dans un espace de paramétrisation décrit par des bosses de Hicks-Henne. Enfin, l'article de Laurenceau et Meaux détaille une méthodologie d'optimisation globale avec enrichissement progressif de l'échantillon semblable à ce que nous présenterons dans le chapitre suivant ;
- enfin le krigeage amélioré par gradient (*Gradient-enhanced Kriging* - GeK), dans les contributions de Liu et Batill [LB02] et de Han et al. [HGZ09].

Dans les travaux cités ci-dessus [CA02b] [CA02a] [LB02] [LS08], on distingue en fait deux types de co-krigeage :

- le co-krigeage indirect, qui consiste à appliquer une méthode de krigeage classique sur un échantillon enrichi par approximation linéaire. En chaque point, le gradient est utilisé pour calculer des points décalés d'un pas ϵ dans chaque direction de l'espace. La bonne manière de choisir le pas ϵ est alors une question complexe. Liu et Batill [LB02] proposent d'ajouter le pas ϵ comme paramètre du problème du maximum de vraisemblance ;
- le co-krigeage direct, qui prend en compte les dérivées directement dans la construction du modèle.

Dans les applications où ces méthodes ont été comparées, elles donnent des approximations de précision proche [CA02b]. Mais la méthode du co-krigeage indirect nécessite l'inversion d'une matrice moins bien conditionnée que la méthode directe, en raison de la présence dans l'échantillon de points proches les uns des autres. Par conséquent, elle est soumise plus facilement à des erreurs numériques qui peuvent nuire à la qualité de l'approximation [CA02b]. A l'inverse, le co-krigeage direct semble plus stable, bien que son implémentation soit plus complexe. C'est cette approche qui a été retenue dans le cadre de cette étude.

L'estimateur recherché est toujours un estimateur linéaire, mais de l'ensemble des données d'entrée :

$$\hat{f}(x) = \sum_{i=1}^{N} \lambda^i(x) y^i + \sum_{k=1}^{d} \sum_{i=1}^{N} \lambda_k^i(x) y_{x_k}^i.$$

Il s'agit encore de trouver le meilleur estimateur, donc celui qui minimise la variance,

$$\begin{aligned}
\sigma_k^2(x) &= Var(\hat{f}(x) - f(x)) \\
&= Var(\hat{f}(x)) + Var(f(x)) - 2Cov(\hat{f}(x), f(x)) \\
&= Cov(\sum_{i=1}^{N} \lambda^i(x) y^i, \sum_{j=1}^{N} \lambda_j(x) y^j) + Cov(\sum_{i=1}^{N} \lambda^i(x) y^i, \sum_{l=1}^{d} \sum_{j=1}^{N} \lambda_l^j(x) y_{x_l}^j) \\
&+ Cov(\sum_{k=1}^{d} \sum_{i=1}^{N} \lambda_k^i(x) y_{x_k}^i, \sum_{j=1}^{N} \lambda^j(x) y^j) + Cov(\sum_{k=1}^{d} \sum_{i=1}^{N} \lambda_k^i(x) y_{x_k}^i, \sum_{l=1}^{d} \sum_{j=1}^{N} \lambda_l^j(x) y_{x_l}^j) \\
&+ Var(f(x)) - 2Cov(\sum_{i=1}^{N} \lambda^i(x) y^i, f(x)) - 2Cov(\sum_{k=1}^{d} \sum_{i=1}^{N} \lambda_k^i(x) y_{x_k}^i, f(x)) \\
&= \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda^i(x) \lambda_j(x) \phi(x^i, x^j) + \sum_{i=1}^{N} \sum_{l=1}^{d} \sum_{j=1}^{N} \lambda^i(x) \lambda_l^j(x) \phi_{0l}(x^i, x^j) \\
&+ \sum_{k=1}^{d} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_k^i(x) \lambda^j(x) \phi_{k0}(x^i, x^j) + \sum_{k=1}^{d} \sum_{i=1}^{N} \sum_{l=1}^{d} \sum_{j=1}^{N} \lambda_k^i(x) \lambda_l^j(x) \phi_{kl}(x^i, x^j) \\
&+ Var(f(x)) - 2 \sum_{i=1}^{N} \lambda^i(x) \phi(x^i, x) - \sum_{k=1}^{d} \sum_{i=1}^{N} \lambda_k^i(x) \phi_{k0}(x^i, x),
\end{aligned}$$

où

$$\begin{aligned}
\phi(x, x') &= Cov(f(x), f(x')), \\
\phi_{k0}(x, x') &= Cov(\frac{\partial f}{\partial x_k}(x), f(x')) \quad \forall k \in \{1, d\}, \\
\phi_{0l}(x, x') &= Cov(f(x), \frac{\partial f}{\partial x'_l}(x')) \quad \forall l \in \{1, d\}, \\
\phi_{kl}(x, x') &= Cov(\frac{\partial f}{\partial x_k}(x), \frac{\partial f}{\partial x'_l}(x')) \quad \forall (k, l) \in \{1, d\}^2.
\end{aligned}$$

Dans le cas d'une fonction de covariance gaussienne, on peut écrire [MMY93] :

$$\begin{aligned}
\phi(x, x') &= \exp\left[-\frac{1}{2} \sum_{q=1}^{d} (\frac{x_q - x'_q}{\zeta_q})^2\right], \\
\phi_{k0}(x, x') &= -\frac{1}{\zeta_k^2}(x_k - x'_k)\phi(x, x') \quad \forall k \in \{1, d\}, \\
\phi_{0l}(x, x') &= \frac{1}{\zeta_l^2}(x_l - x'_l)\phi(x, x') \quad \forall l \in \{1, d\}, \\
\phi_{kl}(x, x') &= -\frac{1}{\zeta_k^2 \zeta_l^2}(x_k - x'_k)(x_l - x'_l)\phi(x, x') \quad \forall (k, l) \in \{1, d\}^2 \quad k \neq l, \\
\phi_{kk}(x, x') &= \frac{1}{\zeta_k^2}\left[1 - \frac{1}{\zeta_k^2}(x_k - x'_k)^2\right]\phi(x, x') \quad \forall k \in \{1, d\}.
\end{aligned}$$

On recherche à nouveau une estimation sans biais de la fonction :

$$E[\hat{f}(x) - f(x)] = \sum_{i=1}^{N} \lambda^i \mu(x^i) + \sum_{k=1}^{d} \sum_{i=1}^{N} \lambda_k^i \mu_k(x^i) - \mu(x) = 0.$$

L'approche retenue ici est une approche basée sur le krigeage ordinaire. Donc on fait l'hypothèse que la moyenne du processus stochastique est constante, mais inconnue. On pose donc $\mu = \mu(x)$, et on en déduit alors que : $\mu_k = 0 \quad \forall k \in \{1, d\}$ [MMY93]. D'où :

$$\sum_{i=1}^{N} \lambda^i = 1.$$

Par l'introduction d'un multiplicateur de Lagrange ν, on formule alors le lagrangien dont on annule le gradient par rapport à chaque composante λ^j et λ_k^j et par rapport à μ :

$$\begin{aligned}
2\sum_{i=1}^{N} \lambda^i \phi(x^j, x^i) + 2\sum_{l=1}^{d} \sum_{i=1}^{N} \lambda_l^i \phi_{0l}(x^i, x^j) - 2\phi(x^j, x) + 2\nu &= 0 \quad \forall j \in \{1, N\} \\
2\sum_{i=1}^{N} \lambda^i \phi_{k0}(x^i, x^j) + 2\sum_{l=1}^{d} \sum_{i=1}^{N} \lambda_l^i \phi_{kl}(x^i, x^j) - 2\phi_{k0}(x^j, x) + 2\nu &= 0 \quad \forall k \in \{1, d\}, \forall j \in \{1, N\} \\
\sum_{i=1}^{N} \lambda^i &= 1.
\end{aligned}$$

En fait, comme souligné dans [HGZ09], utiliser le terme de co-krigeage est ici un abus de langage. En effet, le co-krigeage est à l'origine une extension du krigeage dans laquelle l'estimation d'une variable primaire est rendue plus précise par l'utilisation de variables secondaires. Si la méthode développée ici pourrait être considérée comme un co-krigeage à deux variables, ce n'est pas exactement le cas. En effet :

- les conditions de biais sont différentes. Dans le co-krigeage classique, il faut ajouter autant de conditions de biais que de variables secondaires. Dans l'approche présentée ici, il suffit d'assurer que $\sum_{i=1}^{N} \lambda^i = 1$;
- avec la condition isotropique du co-krigeage classique, l'estimation n'est pas améliorée si les observations de la variable secondaire sont effectuées aux mêmes points que celles de la variable primaire. Or ce résultat n'est pas conservé pour l'approche avec gradients, comme on pourra le voir par la suite.

Pour une estimation de la fonction à une abscisse quelconque x^*, les poids λ^i et λ_k^j s'obtiennent donc par résolution du système :

$$\underbrace{\begin{bmatrix} \phi(x^1,x^1) & \cdots & \phi(x^1,x^N) & \phi_{01}(x^1,x^1) & \cdots & \phi_{01}(x^1,x^N) & \cdots & \phi_{0d}(x^1,x^1) & \cdots & \phi_{0d}(x^1,x^N) & 1 \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots & \vdots \\ \phi(x^N,x^1) & \cdots & \phi(x^N,x^N) & \phi_{01}(x^N,x^1) & \cdots & \phi_{01}(x^N,x^N) & \cdots & \phi_{0d}(x^N,x^1) & \cdots & \phi_{0d}(x^N,x^N) & 1 \\ \phi_{10}(x^1,x^1) & \cdots & \phi_{10}(x^1,x^N) & \phi_{11}(x^1,x^1) & \cdots & \phi_{11}(x^1,x^N) & \cdots & \phi_{1d}(x^1,x^1) & \cdots & \phi_{0d}(x^1,x^N) & 0 \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots & \vdots \\ \phi_{10}(x^N,x^1) & \cdots & \phi_{10}(x^N,x^N) & \phi_{11}(x^N,x^1) & \cdots & \phi_{11}(x^N,x^N) & \cdots & \phi_{1d}(x^N,x^1) & \cdots & \phi_{0d}(x^N,x^N) & 0 \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots & \vdots \\ \phi_{d0}(x^1,x^1) & \cdots & \phi_{d0}(x^1,x^N) & \phi_{d1}(x^1,x^1) & \cdots & \phi_{d1}(x^1,x^N) & \cdots & \phi_{dd}(x^1,x^1) & \cdots & \phi_{dd}(x^1,x^N) & 0 \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots & \vdots \\ \phi_{d0}(x^N,x^1) & \cdots & \phi_{d0}(x^N,x^N) & \phi_{d1}(x^N,x^1) & \cdots & \phi_{d1}(x^N,x^N) & \cdots & \phi_{dd}(x^N,x^1) & \cdots & \phi_{dd}(x^N,x^N) & 0 \\ 1 & \cdots & 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & 0 \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} \lambda^1 \\ \vdots \\ \lambda^N \\ \lambda_1^1 \\ \vdots \\ \lambda_1^N \\ \vdots \\ \lambda_d^1 \\ \vdots \\ \lambda_d^N \\ \nu \end{bmatrix}}_{\Gamma} = \underbrace{\begin{bmatrix} \phi(x^1,x^*) \\ \vdots \\ \phi(x^N,x^*) \\ \phi_{10}(x^1,x^*) \\ \vdots \\ \phi_{10}(x^N,x^*) \\ \vdots \\ \phi_{d0}(x^1,x^*) \\ \vdots \\ \phi_{d0}(x^N,x^*) \\ 1 \end{bmatrix}}_{\phi_{x^*}}.$$

Et la valeur de l'estimation s'obtient alors trivialement :

$$\hat{f}(x^*) = \sum_{i=1}^{N} \lambda^i y^i + \sum_{k=1}^{d} \sum_{i=1}^{N} \lambda_k^i y_{x_k}^i = \Gamma^T Y. \tag{2.14}$$

Implémentation efficace

De la même façon que pour le krigeage classique, pour des raisons d'efficacité, il est plus intéressant de résoudre, dans un premier temps, le système $\Phi\Lambda = Y$, puis de calculer la valeur en sortie du métamodèle par le produit scalaire $\phi_{x^*}^T \Lambda = \phi_{x^*}^T \Phi^{-1} Y$.

2.5.2 L'utilisation de l'information gradient dans la méthode RBF

L'utilisation de l'information gradient dans les réseaux de fonctions à base radiale a été proposée par Giannakoglou et al. [GPK06], qui ont utilisés les gradients pour des applications en optimisation aérodynamique de forme. On en présente ci-dessous le principe et on note GA-RBF cette méthode.

On rappelle tout d'abord la formule d'approximation par un réseau RBF :

$$\hat{f}(x) = \sum_{i=1}^{N_c} w^i \phi^i(x, c^i).$$

L'information gradient donne accès, en chaque point de l'échantillon d'apprentissage, à d informations supplémentaires, portant à $[(d+1)N]$ le nombre total d'équations. Pour prendre en compte ces informations dans la méthode RBF, Giannakoglou et al. [GPK06] proposent d'ajouter de nouvelles fonctions de base à l'approximation. Ici, on ajoute les dérivées des fonctions RBF comme fonctions de base supplémentaires. La formule d'estimation du modèle en un point x quelconque devient alors :

$$\hat{f}(x) = \sum_{i=1}^{N_c} b^i \phi^i(x, c^i) + \sum_{i=1}^{N_c} \sum_{k=1}^{d} a_k^i \frac{\partial \phi^i}{\partial x_k'}(x, c^i).$$

Utilisant les N valeurs de la fonction en chaque point de l'échantillon d'apprentissage, on obtient donc N équations :

$$\forall j \in \{1, N\}, \quad f(x^j) = \sum_{i=1}^{N} b^i \phi^i(x^j, c^i) + \sum_{i=1}^{N} \sum_{k=1}^{d} a_k^i \phi_{0k}^i(x^j, c^i).$$

L'information gradient nous donne (dN) équations supplémentaires :

$$\begin{aligned} \forall l \in \{1, d\}, \forall j \in \{1, N\}, \frac{\partial f}{\partial x_l}(x^j) &= \frac{\partial \left[\sum_{i=1}^{N} b^i \phi^i(\|x^j - c^i\|) + \sum_{k=1}^{d} a_k^i \phi_{0k}^i(\|x^j - c^i\|)\right]}{\partial x_l} \\ &= \sum_{i=1}^{N} \left[b^i \phi_{l0}^i(x^j, c^i) + \sum_{k=1}^{d} a_k^i \phi_{lk}^i(x^j, c^i) \right]. \end{aligned}$$

Reste donc à fixer les N fonctions de base qui sont choisies de la même façon que pour la méthode RBF classique. Les coefficients se calculent alors par résolution d'un système de taille $[(d+1)N \times (d+1)N]$:

$$\underbrace{\begin{bmatrix} \phi(x^1,x^1) & \cdots & \phi(x^1,x^N) & \phi_{01}(x^1,x^1) & \cdots & \phi_{01}(x^1,x^N) & \cdots & \phi_{0d}(x^1,x^1) & \cdots & \phi_{0d}(x^1,x^N) \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ \phi(x^N,x^1) & \cdots & \phi(x^N,x^N) & \phi_{01}(x^N,x^1) & \cdots & \phi_{01}(x^N,x^N) & \cdots & \phi_{0d}(x^N,x^1) & \cdots & \phi_{0d}(x^N,x^N) \\ \phi_{10}(x^1,x^1) & \cdots & \phi_{10}(x^1,x^N) & \phi_{11}(x^1,x^1) & \cdots & \phi_{11}(x^1,x^N) & \cdots & \phi_{1d}(x^1,x^1) & \cdots & \phi_{0d}(x^1,x^N) \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ \phi_{10}(x^N,x^1) & \cdots & \phi_{10}(x^N,x^N) & \phi_{11}(x^N,x^1) & \cdots & \phi_{11}(x^N,x^N) & \cdots & \phi_{1d}(x^N,x^1) & \cdots & \phi_{0d}(x^N,x^N) \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ \phi_{d0}(x^1,x^1) & \cdots & \phi_{d0}(x^1,x^N) & \phi_{d1}(x^1,x^1) & \cdots & \phi_{d1}(x^1,x^N) & \cdots & \phi_{d^2}(x^1,x^1) & \cdots & \phi_{d^2}(x^1,x^N) \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ \phi_{d0}(x^N,x^1) & \cdots & \phi_{d0}(x^N,x^N) & \phi_{d1}(x^N,x^1) & \cdots & \phi_{d1}(x^N,x^N) & \cdots & \phi_{d^2}(x^N,x^1) & \cdots & \phi_{d^2}(x^N,x^N) \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} b^1 \\ \vdots \\ b^N \\ a_1^1 \\ \vdots \\ a_1^N \\ \vdots \\ a_d^N \\ \vdots \\ a_d^N \end{bmatrix}}_{\Lambda} = \underbrace{\begin{bmatrix} y^1 \\ \vdots \\ y^N \\ y_{x_1}^1 \\ \vdots \\ y_{x_1}^N \\ \vdots \\ y_{x_d}^1 \\ \vdots \\ y_{x_d}^N \end{bmatrix}}_{Y}.$$

Alors l'approximation en un point x^* s'obtient par la formule suivante :

$$\hat{f}(x^*) = \sum_{i=1}^{N} b^i \phi(x^*, x^i) + \sum_{i=1}^{N} \sum_{k=1}^{d} a_k^i \phi_{k0}(x^*, x^i) = \phi_{x^*}^T \Lambda = \phi_{x^*}^T \Phi^{-1} Y. \tag{2.15}$$

2.5.3 L'utilisation de l'information gradient dans la méthode SVR

L'utilisation de l'information gradient par la méthode SVR a été proposée récemment par Lazaro et al. [LSPCAR05] pour la formulation classique de la méthode SVR. Une variante de cette méthode, basée sur une simplification de la méthode SVR (*Least-Square SVM*) a été proposée par Jayadeva et al. [JKC06]. Le modèle a alors l'avantage d'être moins coûteux à construire. Mais dans nos essais numériques, cette méthode n'a pas donné de résultats satisfaisants. Par conséquent, c'est la première approche, notée GA-SVR, qui sera présentée dans ce paragraphe :

Formulation du problème

On reprend donc l'approche classique de la méthode SVR, présentée précédemment, et on ajoute des contraintes sur les valeurs des dérivées partielles dans chacune des directions de

l'espace d'entrée. L'apprentissage du modèle correspond alors à la résolution du problème d'optimisation suivant :

$$
\begin{aligned}
\min_{w,b,\xi,\xi^*,\tau,\tau^*} \quad & \frac{1}{2}\|w\|^2 + C_0 \sum_{i=1}^{N}(\xi^i + \xi^{i*}) + \sum_{k=1}^{d} C_k \sum_{i=1}^{N}(\tau_k^i + \tau_k^{i*}) \\
\text{slc :} \quad & \left\langle w, \mathcal{P}(x^i)\right\rangle + b - y^i \leq \epsilon_0 + \xi^i \quad \forall i \in \{1,N\} \\
& y^i - \left\langle w, \mathcal{P}(x^i)\right\rangle - b \leq \epsilon_0 + \xi^{i*} \quad \forall i \in \{1,N\} \\
& \left\langle w, \frac{\partial \mathcal{P}(x^i)}{\partial x_k}\right\rangle - y^i_{x_k} \leq \epsilon_k + \tau_k^i \quad \forall i \in \{1,N\}, \quad \forall k \in \{1,d\} \\
& y^i_{x_k} - \left\langle w, \frac{\partial \mathcal{P}(x^i)}{\partial x_k}\right\rangle \leq \epsilon_k + \tau_k^{i*} \quad \forall i \in \{1,N\}, \quad \forall k \in \{1,d\} \\
& \xi^i, \xi^{i*}, \tau_k^i, \tau_k^{i*} \geq 0 \quad \forall i \in \{1,N\}, \quad \forall k \in \{1,d\}.
\end{aligned}
\tag{2.16}
$$

Écriture du problème dual

On cherche maintenant à formuler le problème dual. Dans un premier temps, on calcule le lagrangien :

$$
\begin{aligned}
L(w,b,\xi,\xi^*,\tau,\tau^*,\alpha,\alpha^*,\lambda,\lambda^*) = {} & \frac{1}{2}\|w\|^2 + C_0 \sum_{i=1}^{N}(\xi^i + \xi^{i*}) + \sum_{k=1}^{d} C_k \sum_{i=1}^{N}(\tau_k^i + \tau_k^{i*}) \\
& - \sum_{i=1}^{N} \alpha^i \left[\epsilon_0 + \xi^i - \left\langle w, \mathcal{P}(x^i)\right\rangle - b + y^i\right] - \sum_{i=1}^{N} \alpha^{i*} \left[\epsilon_0 + \xi^{i*} + \left\langle w, \mathcal{P}(x^i)\right\rangle + b - y^i\right] \\
& - \sum_{k=1}^{d}\sum_{i=1}^{N} \lambda_k^i \left[\epsilon_k + \tau_k^i - \left\langle w, \frac{\partial \mathcal{P}(x^i)}{\partial x_k}\right\rangle + y^i_{x_k}\right] - \sum_{k=1}^{d}\sum_{i=1}^{N} \lambda_k^{i*} \left[\epsilon_k + \tau_k^{i*} + \left\langle w, \frac{\partial \mathcal{P}(x^i)}{\partial x_k}\right\rangle - y^i_{x_k}\right] \\
& - \sum_{i=1}^{N}(\mu^i \xi^i + \mu^{i*}\xi^{i*}) - \sum_{k=1}^{d}\sum_{i=1}^{N}(\theta_k^i \tau_k^i + \theta_k^{i*}\tau_k^{i*}).
\end{aligned}
$$

On écrit ensuite les conditions KKT d'optimalité :

$$
\begin{aligned}
\frac{\partial L}{\partial w}(w,b,\xi,\xi^*,\tau,\tau^*,\alpha,\alpha^*,\lambda,\lambda^*) &= w + \sum_{i=1}^{N}(\alpha^i - \alpha^{i*})\mathcal{P}(x^i) + \sum_{k=1}^{d}\sum_{i=1}^{N}(\lambda_k^i - \lambda_k^{i*})\frac{\partial \mathcal{P}(x^i)}{\partial x_k} = 0 \\
\frac{\partial L}{\partial b}(w,b,\xi,\xi^*,\tau,\tau^*,\alpha,\alpha^*,\lambda,\lambda^*) &= \sum_{i=1}^{N}(\alpha^i - \alpha^{i*}) = 0 \\
\frac{\partial L}{\partial \xi^i}(w,b,\xi,\xi^*,\tau,\tau^*,\alpha,\alpha^*,\lambda,\lambda^*) &= C_0 - \alpha^i - \mu^i = 0 \quad \forall i \in \{1,N\} \\
\frac{\partial L}{\partial \xi^{i*}}(w,b,\xi,\xi^*,\tau,\tau^*,\alpha,\alpha^*,\lambda,\lambda^*) &= C_0 - \alpha^{i*} - \mu^{i*} = 0 \quad \forall i \in \{1,N\} \\
\frac{\partial L}{\partial \tau_k^i}(w,b,\xi,\xi^*,\tau,\tau^*,\alpha,\alpha^*,\lambda,\lambda^*) &= C_k - \lambda_k^i - \theta_k^i = 0 \quad \forall i \in \{1,N\}, \quad \forall \in k\{1,d\} \\
\frac{\partial L}{\partial \tau_k^{i*}}(w,b,\xi,\xi^*,\tau,\tau^*,\alpha,\alpha^*,\lambda,\lambda^*) &= C_k - \lambda_k^{i*} - \theta_k^{i*} = 0 \quad \forall i \in \{1,N\}, \quad \forall k \in \{1,d\}.
\end{aligned}
$$

Alors :

$$\begin{aligned}
L(w,b,\xi,\xi^*,\tau,\tau^*,\alpha,\alpha^*,\lambda,\lambda^*) &= \frac{1}{2}\|w\|^2 + \sum_{i=1}^{N}(\alpha^i+\mu^i)\xi^i + \sum_{i=1}^{N}(\alpha^{i*}+\mu^{i*})\xi^{i*} \\
&+ \sum_{k=1}^{d}(\sum_{i=1}^{N}(\lambda_k^i+\theta_k^i)\tau_k^i + \sum_{i=1}^{N}(\lambda_k^{i*}+\theta_k^{i*})\tau_k^{i*}) \\
&- \sum_{i=1}^{N}\alpha^i\left[\epsilon_0+\xi^i-\left\langle w,\mathcal{P}(x^i)\right\rangle - b + y^i\right] \\
&- \sum_{i=1}^{N}\alpha^{i*}\left[\epsilon_0+\xi^{i*}+\left\langle w,\mathcal{P}(x^i)\right\rangle + b - y^i\right] \\
&- \sum_{k=1}^{d}\sum_{i=1}^{N}\lambda_k^i\left[\epsilon_k+\tau_k^i-\left\langle w,\frac{\partial\mathcal{P}(x^i)}{\partial x_k}\right\rangle + y_{x_k}^i\right] \\
&- \sum_{k=1}^{d}\sum_{i=1}^{N}\lambda_k^{i*}\left[\epsilon_k+\tau_k^{i*}+\left\langle w,\frac{\partial\mathcal{P}(x^i)}{\partial x_k}\right\rangle - y_{x_k}^i\right] \\
&- \sum_{i=1}^{N}(\mu^i\xi^i+\mu^{i*}\xi^{i*}) - \sum_{k=1}^{d}\sum_{i=1}^{N}(\theta_k^i\tau_k^i+\theta_k^{i*}\tau_k^{i*}) \\
&= \frac{1}{2}\|w\|^2 - \epsilon_0\sum_{i=1}^{N}(\alpha^i+\alpha^{i*}) - \sum_{i=1}^{N}(\alpha^i-\alpha^{i*})y^i - b\sum_{i=1}^{N}(\alpha^i-\alpha^{i*}) \\
&- \sum_{i=1}^{N}(\alpha^{i*}-\alpha^i)\left\langle w,\mathcal{P}(x^i)\right\rangle - \sum_{k=1}^{d}\epsilon_k\sum_{i=1}^{N}(\lambda_k^i+\lambda_k^{i*}) \\
&- \sum_{k=1}^{d}\sum_{i=1}^{N}(\lambda_k^i-\lambda_k^{i*})y_{x_k}^i - \sum_{k=1}^{d}\sum_{i=1}^{N}(\lambda_k^{i*}-\lambda_k^i)\left\langle w,\frac{\partial\mathcal{P}(x^i)}{\partial x_k}\right\rangle.
\end{aligned}$$

Or :

$$-\sum_{i=1}^{N}(\alpha^{i*}-\alpha^i)\left\langle w,\mathcal{P}(x^i)\right\rangle - \sum_{k=1}^{d}\sum_{i=1}^{N}(\lambda_k^{i*}-\lambda_k^i)\left\langle w,\frac{\partial\mathcal{P}(x^i)}{\partial x_k}\right\rangle = -\langle w,w\rangle = -\|w\|^2.$$

Le problème dual s'écrit alors :

$$\begin{aligned}
\max_{\alpha,\alpha^*,\lambda,\lambda^*} \quad & -\frac{1}{2}\|w\|^2 - \epsilon_0\sum_{i=1}^{N}(\alpha^i+\alpha^{i*}) - \sum_{k=1}^{d}\epsilon_k\sum_{i=1}^{N}(\lambda_k^i+\lambda_k^{i*}) \\
& - \sum_{i=1}^{N}(\alpha^i-\alpha^{i*})y^i - \sum_{k=1}^{d}\sum_{i=1}^{N}(\lambda_k^i-\lambda_k^{i*})y_{x_k}^i \\
\text{slc} \quad : \quad & \sum_{i=1}^{N}(\alpha^i-\alpha^{i*}) = 0 \\
& \alpha^i,\alpha^{i*}\in[0,C_0] \quad \forall i\in\{1,N\} \\
& \lambda_k^i,\lambda_k^{i*}\in[0,C_k] \quad \forall i\in\{1,N\}, \quad \forall k\in\{1,d\},
\end{aligned}$$

soit :

$$
\begin{aligned}
\min_{\alpha,\alpha^*,\lambda,\lambda^*} \quad & \frac{1}{2}\left[\sum_{i,j=1}^{N}(\alpha^{i*}-\alpha^i)(\alpha^{j*}-\alpha^j)\left\langle \mathcal{P}(x^i),\mathcal{P}(x^j)\right\rangle\right] \\
& +\frac{1}{2}\left[\sum_{k=1}^{d}\sum_{i,j=1}^{N}(\lambda_k^{i*}-\lambda_k^i)(\alpha^{j*}-\alpha^j)\left\langle \frac{\partial\mathcal{P}(x^i)}{\partial x_k},\mathcal{P}(x^j)\right\rangle\right] \\
& +\frac{1}{2}\left[\sum_{k=1}^{d}\sum_{i,j=1}^{N}(\alpha^{i*}-\alpha^i)(\lambda_k^{j*}-\lambda_k^j)\left\langle \mathcal{P}(x^i),\frac{\partial\mathcal{P}(x^j)}{\partial x_k}\right\rangle\right] \\
& +\frac{1}{2}\left[\sum_{k,l=1}^{d}\sum_{i,j=1}^{N}(\lambda_k^{i*}-\lambda_k^i)(\lambda_k^{j*}-\lambda_k^j)\left\langle \frac{\mathcal{P}(x^i)}{\partial x_k},\frac{\mathcal{P}(x^j}{\partial x_k}\right\rangle\right] \\
& +\epsilon_0\sum_{i=1}^{N}(\alpha^i+\alpha^{i*})+\sum_{k=1}^{d}\epsilon_k\sum_{i=1}^{N}(\lambda_k^i+\lambda_k^{i*}) \\
& +\sum_{i=1}^{N}(\alpha^i-\alpha^{i*})y^i+\sum_{k=1}^{d}\sum_{i=1}^{N}(\lambda_k^i-\lambda_k^{i*})y_{x_k}^i \\
\text{slc} \quad : \quad & \sum_{i=1}^{N}(\alpha^i-\alpha^{i*})=0 \\
& \alpha^i,\alpha^{i*}\in[0,C_0] \quad \forall i\in\{1,N\} \\
& \lambda_k^i,\lambda_k^{i*}\in[0,C_k] \quad \forall i\in\{1,N\},\quad \forall k\in\{1,d\}.
\end{aligned}
$$

Introduction des fonctions noyaux

En introduisant les fonctions noyaux, et en utilisant leurs dérivées,

$$
\begin{aligned}
\left\langle \frac{\partial\mathcal{P}}{\partial x_k}(x),\mathcal{P}(x')\right\rangle &= \frac{\partial\phi}{\partial x_k}(x,x')=\phi_{k0}, \\
\left\langle \mathcal{P}(x),\frac{\partial\mathcal{P}}{\partial x_l'}(x')\right\rangle &= \frac{\partial\phi}{\partial x_l'}(x,x')=\phi_{0l}, \\
\left\langle \frac{\partial\mathcal{P}}{\partial x_k}(x),\frac{\partial\mathcal{P}}{\partial x_l'}(x')\right\rangle &= \frac{\partial^2\phi}{\partial x_k\partial x_l'}(x,x')=\phi_{kl},
\end{aligned}
$$

on obtient le problème suivant :

$$
\begin{aligned}
\min_{\alpha,\alpha^*,\lambda,\lambda^*} \quad & \frac{1}{2}\left[\sum_{i,j=1}^{N}(\alpha^{i*}-\alpha^i)(\alpha^{j*}-\alpha^j)\phi(x^i,x^j)\right] \\
& +\frac{1}{2}\left[\sum_{k=1}^{d}\sum_{i,j=1}^{N}(\lambda_k^{i*}-\lambda_k^i)(\alpha^{j*}-\alpha^j)\phi_{k0}(x^i,x^j)\right] \\
& +\frac{1}{2}\left[\sum_{k=1}^{d}\sum_{i,j=1}^{N}(\alpha^{i*}-\alpha^i)(\lambda_k^{j*}-\lambda_k^j)\phi_{0k}(x^i,x^j)\right] \\
& +\frac{1}{2}\left[\sum_{k,l=1}^{d}\sum_{i,j=1}^{N}(\lambda_k^{i*}-\lambda_k^i)(\lambda_k^{j*}-\lambda_k^j)\phi_{kl}(x^i,x^j)\right] \\
& +\epsilon_0\sum_{i=1}^{N}(\alpha^i+\alpha^{i*})+\sum_{k=1}^{d}\epsilon_k\sum_{i=1}^{N}(\lambda_k^i+\lambda_k^{i*}) \\
& +\sum_{i=1}^{N}(\alpha^i-\alpha^{i*})y^i+\sum_{k=1}^{d}\sum_{i=1}^{N}(\lambda_k^i-\lambda_k^{i*})y_{x_k}^i \\
\text{slc} : \quad & \sum_{i=1}^{N}(\alpha^i-\alpha^{i*})=0 \\
& \alpha^i,\alpha^{i*}\in[0,C_0] \quad \forall i\in\{1,N\} \\
& \lambda_k^i,\lambda_k^{i*}\in[0,C_k] \quad \forall i\in\{1,N\}, \quad \forall k\in\{1,d\}.
\end{aligned}
\tag{2.17}
$$

Pour simplifier l'écriture, on peut introduire les matrices de Gram dérivées :

$$
\begin{aligned}
\Phi_{00} &= \left[\phi(x^i,x^j)\right]_{i,j\in\{1,N\}}, \\
\Phi_{k0} &= \left[\phi_{k0}(x^i,x^j)\right]_{i,j\in\{1,N\}}, \\
\Phi_{0l} &= \left[\phi_{0l}(x^i,x^j)\right]_{i,j\in\{1,N\}}, \\
\Phi_{kl} &= \left[\phi_{kl}(x^i,x^j)\right]_{i,j\in\{1,N\}}.
\end{aligned}
$$

On obtient alors l'écriture matricielle du problème à résoudre :

$$\min_{\alpha,\alpha^*,\lambda,\lambda^*} \quad \frac{1}{2}\begin{bmatrix}\alpha\\ \alpha^*\\ \Lambda_1\\ \Lambda_1^*\\ \vdots\\ \Lambda_d\\ \Lambda_d^*\end{bmatrix}^T \begin{bmatrix} \Phi_{00} & -\Phi_{00} & \Phi_{01} & -\Phi_{01} & \dots & \Phi_{0d} & -\Phi_{0d} \\ -\Phi_{00} & \Phi_{00} & -\Phi_{01} & \Phi_{01} & \dots & -\Phi_{0d} & \Phi_{0d} \\ \Phi_{10} & -\Phi_{10} & \Phi_{11} & -\Phi_{11} & \dots & \Phi_{1d} & -\Phi_{1d} \\ -\Phi_{10} & \Phi_{10} & -\Phi_{11} & \Phi_{11} & \dots & -\Phi_{1d} & \Phi_{1d} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ \Phi_{d0} & -\Phi_{d0} & \Phi_{d1} & -\Phi_{d1} & \dots & \Phi_{dd} & -\Phi_{dd} \\ -\Phi_{d0} & \Phi_{d0} & -\Phi_{d1} & \Phi_{d1} & \dots & -\Phi_{dd} & \Phi_{dd} \end{bmatrix} \begin{bmatrix}\alpha\\ \alpha^*\\ \Lambda_1\\ \Lambda_1^*\\ \vdots\\ \Lambda_d\\ \Lambda_d^*\end{bmatrix} + \begin{bmatrix}\epsilon_0+Y\\ \epsilon_0-Y\\ \epsilon_1+Y_1\\ \epsilon_1-Y_1\\ \vdots\\ \epsilon_d+Y_d\\ \epsilon_d-Y_d\end{bmatrix}^T \begin{bmatrix}\alpha\\ \alpha^*\\ \Lambda_1\\ \Lambda_1^*\\ \vdots\\ \Lambda_d\\ \Lambda_d^*\end{bmatrix}$$

$$\text{slc} : \quad \begin{bmatrix}1\\ -1\\ 0\\ 0\\ \vdots\\ 0\\ 0\end{bmatrix}^T \begin{bmatrix}\alpha\\ \alpha^*\\ \Lambda_1\\ \Lambda_1^*\\ \vdots\\ \Lambda_d\\ \Lambda_d^*\end{bmatrix} = 0 \tag{2.18}$$

$$\alpha^i, \alpha^{i*} \in [0, C_0] \quad \forall i \in \{1, N\}$$

$$\lambda_k^i, \lambda_k^{i*} \in [0, C_k] \quad \forall i \in \{1, N\}, \quad \forall k \in \{1, d\},$$

avec $Y_k = \left[y^i_{x_k}\right]_{i\in\{1,N\}}$.

Ce problème est toujours un problème quadratique de dimension plus importante que celui de la méthode initiale. La résolution s'effectue donc également par un solveur basé sur une méthode de points intérieurs, comme celui présenté plus haut dans ce document.

Valeur du modèle en un point quelconque

Une fois les coefficients $\alpha^i, \alpha^{i*}, \lambda_k^i, \lambda_k^{i*}$ calculés, la valeur du modèle en x est fournie par la formule :

$$f(x^*) = \langle w, \mathcal{P}(x^*)\rangle + b = \sum_{i=1}^{N}(\alpha^{i*} - \alpha^i)\phi(x^i, x^*) + \sum_{k=1}^{d}\sum_{i=1}^{N}(\lambda_k^{i*} - \lambda_k^i)\phi_{k0}(x^i, x^*) + b, \tag{2.19}$$

où b est calculé par le solveur quadratique. On note pour la suite :

$$\phi_{x^*} = \left[\phi(x^1, x^*), \dots, \phi(x^N, x^*), \phi_{10}(x^1, x^*), \dots, \phi_{d0}(x^N, x^*)\right],$$

$$\Phi = \begin{bmatrix}\Phi_{00} & \dots & \Phi_{0d}\\ \vdots & & \vdots\\ \Phi_{d0} & \dots & \Phi_{dd}\end{bmatrix}.$$

2.5.4 Optimisation des paramètres des modèles avec gradients

La méthode présentée précédemment pour l'optimisation des paramètres des modèles sans gradient est ici étendue pour les modèles avec gradients. Pour cela, l'erreur *leave-one-out* est redéfinie, de telle sorte qu'elle prenne en compte à la fois des erreurs sur la valeur de la fonction, mais également sur ses dérivées dans chacune des directions :

Définition 2.5.4.1 (Erreur *leave-one-out* des méthodes avec gradients)**.** *L'erreur leave-one-out du modèle $\hat{f}$ construit sur un échantillon d'apprentissage $\mathcal{E}$ enrichi par les gradients de la*

fonction, notée $E(\hat{f})_{loo}$, est définie par :

$$E(\hat{f})_{loo} = \frac{1}{N(1+d)} \sum_{i=1}^{N} \left[(\hat{f}_i(x^i) - y^i)^2 + \sum_{k=1}^{d} (\frac{\partial \hat{f}_i^k}{\partial x_k}(x^i) - y^i_{x_k})^2) \right],$$

où $\hat{f}_i^k$ est le modèle construit sur l'échantillon $\mathcal{E}$ diminué de la dérivée au point i dans la direction k.

On cherche donc à minimiser cette erreur en utilisant les algorithmes proposés dans la première partie de ce document. Il est donc important de pouvoir l'évaluer de manière simple.

Évaluation efficace de l'erreur *leave-one-out*

Méthode RBF avec gradients La méthode de Rippa se généralise pour la méthode RBF avec gradients. L'erreur s'obtient alors par la résolution d'un seul système linéaire de taille $[N(d+1)]$ au lieu de $N(d+1)$ systèmes linéaires de taille $[N(d+1)-1]$.

La formule d'évaluation est la suivante :

$$e_{iN+n} = \frac{(\Phi^{-1} y)_{(iN+n)}}{(\Phi^{-1})_{(iN+n)(iN+n)}}, \quad i \in \{0, d\}, \quad n \in \{1, N\}, \tag{2.20}$$

d'où la formule d'évaluation de l'erreur *leave-one-out* :

$$e_{loo} = \frac{e^T e}{(d+1)N}. \tag{2.21}$$

On trouvera la démonstration de cette formule en annexe B.3.

Co-krigeage Cette formule de calcul de l'erreur *leave-one-out* s'étend sans difficulté au co-krigeage. On en trouvera la démonstration en annexe B.4.

SVR avec gradients On étend la borne *Span* à la régression à vecteurs de support avec gradients :

$$E(\hat{f})_{loo} \leq \frac{1}{N(d+1)} \left[\sum_{t=1}^{N} \left(S_t^2 (\alpha^t + \alpha^{t*}) + \sum_{k=1}^{d} S_{t+kN}^2 (\lambda_k^t + \lambda_k^{t*}) \right) + \sum_{t=1}^{N} \left((\xi^t + \xi^{t*}) + \sum_{k=1}^{d} (\tau_k^t + \tau_k^{t*}) \right) \right] + \epsilon_0 + \sum_{k=1}^{d} \epsilon_k.$$

On note $\mathcal{F}$ l'ensemble des vecteurs de support non bornés et Φ_{USV} la matrice Φ où seules les lignes et colonnes qui correspondent à des vecteurs de $\mathcal{F}$ sont conservées. Si l'indice $t+kN$ correspond également à un vecteur de $\mathcal{F}$ (on notera u son indice dans $\mathcal{F}$), alors on peut écrire :

$$S_{t+kN}^2 = \frac{1}{\left(\tilde{\Phi_{USV}}^{-1} \right)_{uu}}.$$

Dans le cas contraire, S^2_{t+kN} se calcule par (voir 2.4.5) :

$$S^2_{t+kN} = (\Phi)_{tt} - \tilde{\phi_{USV}}^T \Phi_{\tilde{USV}} \tilde{\phi_{USV}},$$

où $\Phi_{\tilde{USV}} = \begin{pmatrix} \Phi_{USV} & 1 \\ 1 & 0 \end{pmatrix}$ et $\tilde{\phi_{USV}} = \begin{pmatrix} \phi_{USV} \\ 1 \end{pmatrix}$, ϕ_{USV} désignant la $(t+kN)$-ème colonne de la matrice Φ, dans laquelle seule les lignes qui correspondent à des vecteurs de $\mathcal{F}$ sont conservées.

Validation de la méthode

On cherche ici à valider la méthode de réglage des paramètres dans le cas des modèles de substitution avec gradients. On utilise à nouveau la fonction $f_1 = x(1-x)\sin 2\pi x$. L'échantillon de validation est cette fois composé de 6 points, toujours choisis selon la méthode de Hammersley. On effectue les mêmes expériences que pour la validation des modèles sans gradient. Dans un premier temps, le modèle est construit pour trois choix du paramètre ζ et les résultats sont donnés sur les figures 2.15, 2.16 et 2.17.

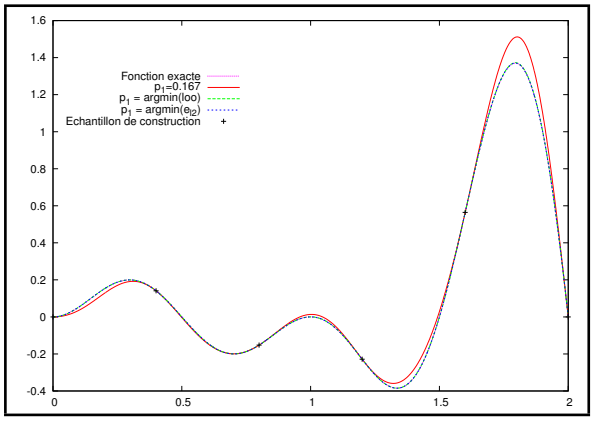


FIG. 2.15 – Approximation par co-krigeage de f_1 pour différents choix de paramètres

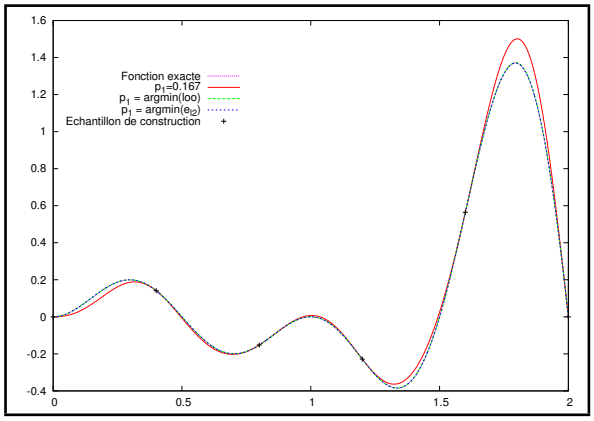


FIG. 2.16 – Approximation par GA-RBF de f_1 pour différents choix de paramètres

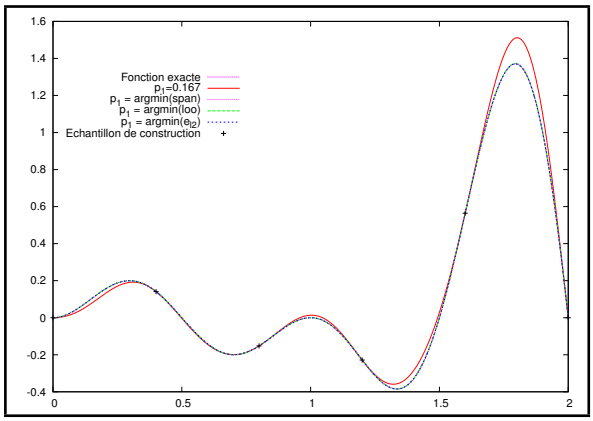


FIG. 2.17 – Approximation par GA-SVR de f_1 pour différents choix de paramètres

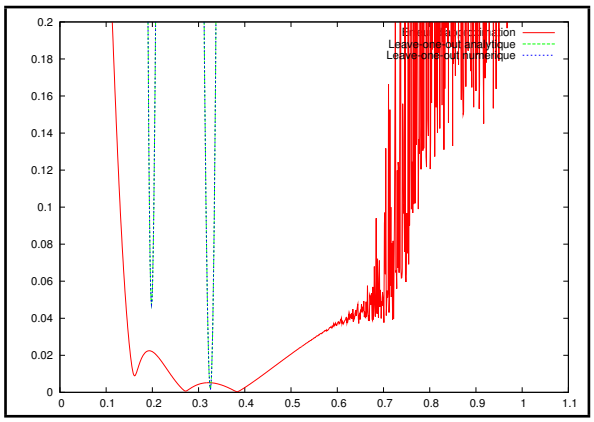


FIG. 2.18 – $E(\hat{f}_1)_{loo}$ et $E(\hat{f})^{\mathcal{E}_V}$ par rapport à ζ_1 (co-krigeage)

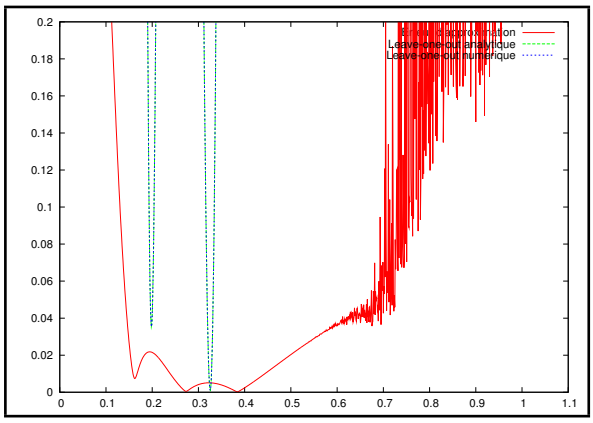


FIG. 2.19 – $E(\hat{f}_1)_{loo}$ et $E(\hat{f})^{\mathcal{E}_V}$ par rapport à ζ_1 (GA-RBF)

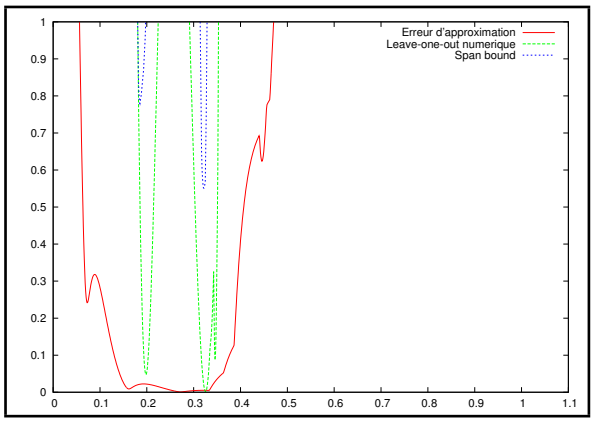


FIG. 2.20 – $E(\hat{f}_1)_{loo}$, $E(\hat{f})^{\mathcal{E}_V}$ et borne *Span* par rapport à ζ_1 (GA-SVR)

De même, l'erreur *leave-one-out* et l'erreur de généralisation de la fonction en fonction de la valeur de ζ sont représentées sur les figures 2.7, 2.8 et 2.9, chaque figure correspondant à un type de modèle. En outre, la borne *Span* est représentée pour le modèle SVR avec gradients. On relève une nouvelle fois qu'elle a un comportement similaire à l'erreur *leave-one-out* : l'optimiser semble donc être un bon choix de réglage des paramètres.

Pour chacun de nos 3 métamodèles, les résultats de cette étude sont synthétisés dans les tableaux 2.5, 2.6 et 2.7.

TAB. 2.5 – Approximation de f_1 par co-krigeage pour différents choix de paramètres

Méthode de sélection de ζ_1	ζ_1	$E(\hat{f}_1)_{loo}$	$E(\hat{f}_1)^{\mathcal{E}_V}$
Sélection manuelle	$0,167$	$0,2828$	$0,04312$
Minimisation de $E(\hat{f}_1)_{loo}$	$0,3254$	$0,001641$	$0,0007439$
Minimisation de $E(\hat{f}_1)^{\mathcal{E}_V}$	$0,3832$		$0,00003961$

TAB. 2.6 – Approximation de f_1 par GA-RBF pour différents choix de paramètres

Méthode de sélection de ζ_1	ζ_1	$E(\hat{f}_1)_{loo}$	$E(\hat{f}_1)^{\mathcal{E}_V}$
Sélection manuelle	$0,167$	$0,2657$	$0,04042$
Minimisation de $E(\hat{f}_1)_{loo}$	$0,3255$	$0,0009193$	$0,0007288$
Minimisation de $E(\hat{f}_1)^{\mathcal{E}_V}$	$0,3842$		$0,00002647$

TAB. 2.7 – Approximation de f_1 par GA-SVR pour différents choix de paramètres

Méthode de sélection de ζ_1	ζ_1	$E(\hat{f}_1)_{loo}$	$E(\hat{f}_1)^{\mathcal{E}_V}$
Sélection manuelle	$0,167$	$2,551$	$0,04312$
Minimisation de la borne *Span*	$0,3212$	$0,02248$	$0,0007484$
Minimisation de $E(\hat{f}_1)_{loo}$	$0,3254$	$0,001644$	$0,0007439$
Minimisation de $E(\hat{f}_1)^{\mathcal{E}_V}$	$0,2723$		$0,00009178$

2.5.5 Première comparaison des modèles

Afin de mettre en évidence l'intérêt de l'utilisation des gradients dans la construction des modèles, on présente dans cette section les résultats de la reconstruction de la fonction de Rosenbrock à partir d'un échantillon de 10 points. Les modèles avec ou sans gradient sont comparés dans les figures 2.22 à 2.27.

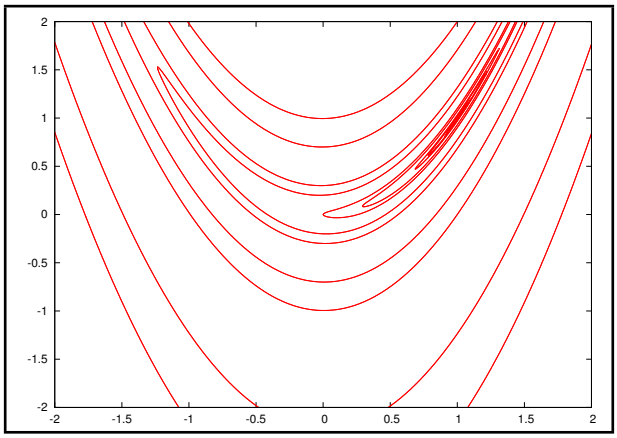


FIG. 2.21 – Fonction de Rosenbrock 2D

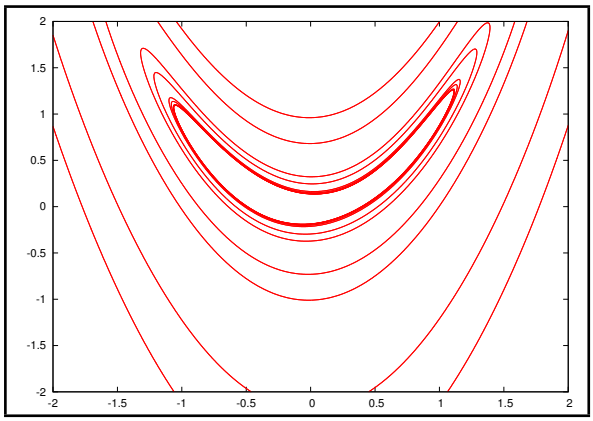


FIG. 2.22 – Reconstruction de la fonction de Rosenbrock en 2D par krigeage

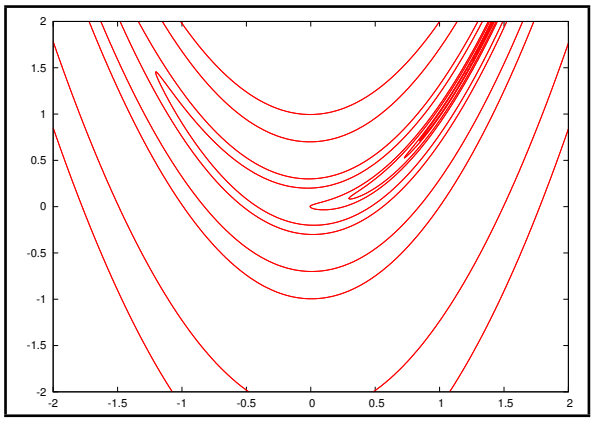


FIG. 2.23 – Reconstruction de la fonction de Rosenbrock en 2D par co-krigeage

L'utilisation des gradients pour la construction du modèle permet donc d'améliorer sensiblement la qualité du métamodèle pour un même nombre de points d'échantillonnage. Par ailleurs, la reconstruction de la fonction de Rosenbrock semble donner des résultats similaires avec le krigeage, le réseau RBF ou la régression SVR. Il est tout de même important de noter que le calcul du modèle SVR est extrêmement coûteux, en raison de la résolution d'un problème quadratique lors de la construction, mais surtout à chaque évaluation de l'erreur *leave-one-out* durant le processus de réglage des paramètres. Dans la suite de notre étude, lorsque la taille des échantillons d'apprentissage va augmenter, cet inconvénient va devenir rédhibitoire. Par conséquent, pour le chapitre suivant, le modèle SVR (et respectivement le modèle SVR avec gradients) sera construit avec le paramètre ζ optimal d'un réseau RBF (et respectivement d'un réseau RBF

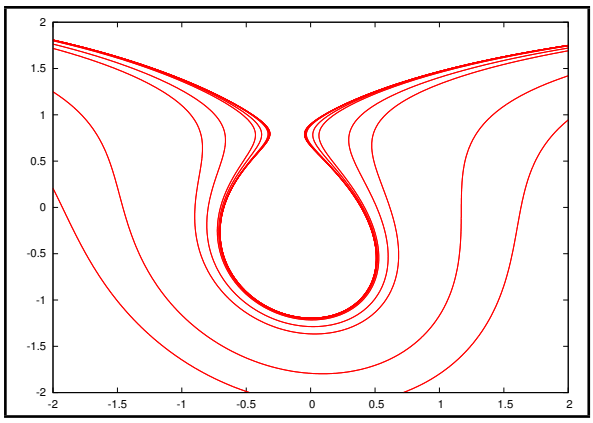


Fig. 2.24 – Reconstruction de la fonction de Rosenbrock en 2D par RBF

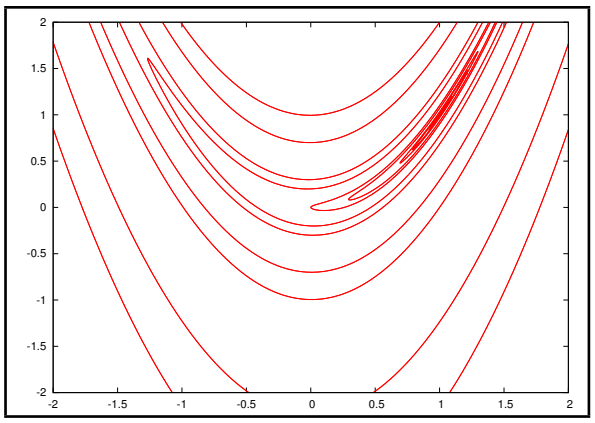


Fig. 2.25 – Reconstruction de la fonction de Rosenbrock en 2D par RBF avec gradients

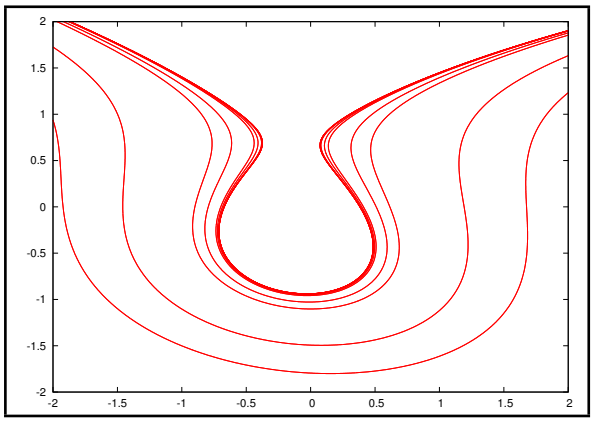


Fig. 2.26 – Reconstruction de la fonction de Rosenbrock en 2D par SVR

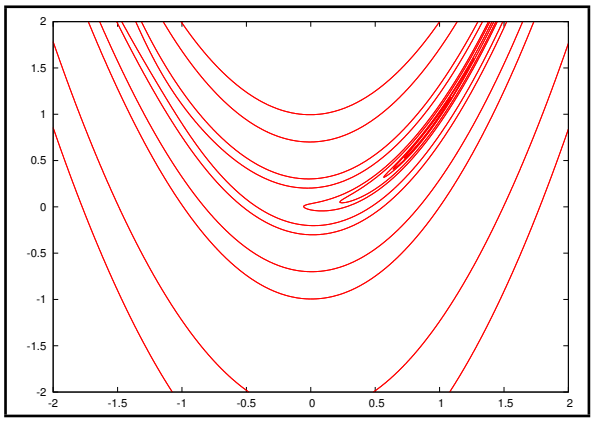


Fig. 2.27 – Reconstruction de la fonction de Rosenbrock en 2D par SVR avec gradients

avec gradients). En effet, dans nos tests numériques, l'erreur *leave-one-out* semble évoluer de manière similaire en fonction de ζ quel que soit le modèle considéré (voir notamment les figures 2.8 et 2.9 pour la comparaison RBF/SVR et les figures 2.19 et 2.20 lorsque les gradients sont utilisés).

Conclusion

Nous avons présenté dans ce chapitre les fondements mathématiques de trois modèles de substitution parmi les plus utilisés dans la littérature : le krigeage, les réseaux de fonctions RBF puis la méthode de régression par vecteurs de support. A l'issue d'une étude bibliographique, nous avons mis en place une procédure d'ajustement automatique des paramètres internes aux modèles, par minimisation de l'erreur *leave-one-out*, que nous avons validée sur des fonctions mathématiques. Enfin, dans une dernière partie, chacun des métamodèles sélectionnés pour cette étude a été étendu pour prendre en compte les gradients des fonctions aux points de l'échantillon. Dans le prochain chapitre, nous allons utiliser ces modèles pour l'optimisation, et notamment pour l'optimisation de forme en aérodynamique.

Chapitre 3

Optimisation globale avec modèles de substitution

Introduction

Les origines de l'utilisation de modèles de substitution dans le cadre de problèmes physiques remontent au milieu des années 70 où ils sont utilisés pour la conception de structures [STBV08]. Il s'agit alors de représenter, de manière peu coûteuse, l'évolution d'une fonction d'intérêt issue d'un modèle complexe en fonction d'un ou de plusieurs paramètres. Contrairement aux fonctions calculées à partir de mesures expérimentales, qui ont un comportement statistique en raison des approximations dues aux appareils expérimentaux, on parle ici de fonctions déterministes : pour un jeu de paramètres donné, le modèle physique ne fournira qu'un seul résultat. La simulation de ce type de fonction est connue sous le nom DACE (*Design and Analysis of Computer Experiments* [SWMW89]) en opposition aux approches de type DoE (*Design of Experiments*) qui ont pour principe de chercher à séparer la variation aléatoire due aux mesures des effets principaux de la fonction.

Si le nom de DACE n'a été introduit qu'en 89 dans l'article de Sacks et al. [SWMW89], des premières expériences d'utilisation de modèles de substitution dans le cadre de simulations de phénomènes physiques ont été effectuées dès la fin des années 70. La méthodologie des surfaces de réponses (*Surface Response Methodology* - RSM) qui consiste à utiliser un polynôme d'ordre deux [BW51] ou les réseaux de neurones sont alors exclusivement utilisés [STBV08] et il s'agit d'approcher la forme d'une réponse en fonction de plusieurs paramètres d'entrée.

Avec la formalisation de l'approche DACE, les chercheurs découvrent que les modèles de substitution sont des outils robustes et précis pour simuler le comportement d'une fonction d'intérêt issue de modèles physiques. Ils se tournent alors vers les problèmes de conception optimale de formes. En effet, à cette époque, l'optimisation de forme par des algorithmes numériques est un domaine en plein développement. L'extension de la méthode adjointe a permis la généralisation des optimisations de forme. Mais les algorithmes sont encore limités à des espaces de faible dimension, et la recherche de l'optimum est limitée au voisinage de la forme initiale. Au début des années 90, le développement des outils de calculs et le perfectionnement des modèles physiques permettent d'envisager des optimisations globales. On a déjà cité dans l'introduction

de ce mémoire les calculs de Mosetti et Poloni [MP93] ainsi que de Quagliarella et Della Cioppa [QDC94] qui concernent l'utilisation de la méthode génétique pour la résolution de problèmes d'optimisation de forme en aéronautique. Mais certains de ces calculs exigent jusqu'à dix mille évaluations de la fonction d'intérêt [TC00] [HP01]. Pour limiter ce nombre, on peut chercher à remplacer le plus grand nombre possible d'évaluations par des évaluations moins coûteuses.

Pour cela, l'utilisation de modèles physiques économiques (basse fidélité) en effectuant par exemple des hypothèses de simplification de la physique considérée, ou de modèles mathématiques construits à partir d'un échantillon d'évaluations est nécessaire. Différents modèles basse fidélité ont été introduits dans la littérature : Han et al. [HGZ09] ont utilisé les équations d'Euler pour construire un modèle de krigeage dans le cadre d'une optimisation de forme basée sur la résolution des équations RaNS ; dans l'article de Rajnarayan et Haas [DHK08], c'est la méthode *panel* (basse fidélité) qui est utilisé avec les équations d'Euler (haute fidélité) ; d'autres auteurs proposent de jouer sur les niveaux de convergence [FBK06] ou encore sur la densité du maillage [ALG+01].

Dans le cadre de ce travail, on ne souhaite pas utiliser deux modèles physiques de précisions différentes. C'est donc à l'utilisation des modèles de substitution pour l'optimisation que nous allons nous intéresser ici. Les premières optimisations de ce type apparaissent dans le domaine de la conception optimale de structures avec les travaux de Vanderplaats [Van89] ou Schoofs [Sch87]. Il s'agit alors de choisir un ensemble de points d'observation à évaluer sur le modèle physique, et de les utiliser pour construire le modèle mathématique sur lequel est effectué l'optimisation. Dans [Sch87], le modèle mathématique est construit par approximation polynomiale et une application pour un problème avec 10 paramètres de forme est présentée. Dans ces premières expérimentations, il n'y a pas d'interaction dynamique entre les modèles physiques et mathématiques : le metamodèle est construit une fois pour toute. Il est donc indispensable qu'il soit extrêmement précis afin que son optimum corresponde à celui du modèle physique. Mais comme on ne connaît pas a priori les zones de l'espace où peut se trouver cet optimum, la précision est nécessairement importante sur l'ensemble de l'espace de définition, ce qui s'avère très couteux et revient à gaspiller des ressources de calcul dans les zones peu intéressantes de l'espace de dessin.

A partir de ces observations, des études se multiplient sur la meilleure manière de combiner les modèles physiques et mathématiques. Dennis et Torczon [DT95] proposent un algorithme de type *pattern search* qui fait intervenir le krigeage pour accélérer la convergence. L'usage est alors limité aux algorithmes locaux.

A partir de la fin des années 90, plusieurs études proposent l'utilisation des modèles dans le cadre d'algorithmes évolutionnaires [RK98] [Rat98] [EBK98]. Généralement, le krigeage est utilisé. Un échantillonnage initial de l'espace de dessin est réalisé et un premier modèle est construit. Puis ce modèle est utilisé en lieu et place du modèle physique pendant un nombre fixé d'itérations. Une fois ce nombre atteint, des points sélectionnés parmi la population de l'algorithme génétique sont évalués sur le modèle physique et ajoutés à l'échantillon (tous ou seulement certains d'entre eux, remplaçant ou non d'autres points en fonction des auteurs). Le modèle est alors mis à jour et la procédure peut recommencer jusqu'à ce qu'un critère d'arrêt soit atteint, ou que le nombre d'évaluations sur le modèle physique ait dépassé un maximum autorisé. Dans [RH00], Rasheed et Hirsch proposent une autre manière de combiner les algorithmes évo-

lutionnaires et les modèles de substitution : à chaque itération, les opérateurs de mutation et de croisement sont utilisés pour générer la population candidate de la prochaine itération mais tous les candidats ne sont pas évalués ensuite sur le modèle physique. En effet, ils sont triés à partir de la valeur de leurs réponses sur le modèle mathématique, et seul le meilleur d'entre eux est calculé exactement et ajouté pour la construction du metamodèle. Enfin, Giannakoglou [Gia02] a présenté une variante appelée GA-IPE (*Genetic Algorithm - Inexact Pre Evaluation*) dans la littérature. A chaque itération, hormis la première, la génération de la nouvelle population se fait en deux phases : tout d'abord, chaque individu est évalué sur un modèle de substitution local construit à partir des voisins les plus proches dans l'espace. Ensuite, les individus avec les meilleures évaluations sont calculés par le modèle physique. Giannakoglou donne des applications performantes de cet algorithme à l'optimisation de profils d'aile. Cette approche est également utilisée dans le même contexte par Praveen et Duvigneau [PD09] mais l'algorithme génétique est alors remplacé par un algorithme à essaims particulaires (PSO). Des modèles de substitution locaux sont également utilisés dans les stratégies d'optimisation présentées par Ong et al. [ONW04].

Mais à l'heure actuelle, il semble que l'article de Jones, Schonlau et Welch [JSW98] constitue la référence sur le sujet. Il s'agit de construire un modèle de substitution à partir d'un échantillon initial et de maximiser un critère d'enrichissement défini sur ce modèle et visant à identifier le point le plus prometteur. Ce point est alors évalué exactement et ajouté à l'échantillon de construction du modèle. En itérant ce processus, on obtient un algorithme connu sous le nom d'EGO (*Efficient Global Optimization*). Depuis le milieu des années 2000, cette méthode est quasi exclusivement utilisée pour combiner des modèles de substitution et des algorithmes d'optimisation globale. Elle a été mise en place avec les modèles de krigeage [Jon01], avec les réseaux de fonctions à base radiale [Gut01] et enfin avec les méthodes de co-krigeage [LBK04a] [YM11]. Elle a été étendue aux problèmes d'optimisation avec contraintes, notamment dans la thèse de Sasena [Sas02]. On trouve des applications de cette méthode en aéronautique dans les travaux de Jeong et al. [JMY05], Rajnarayan et al. [DHK08] ou encore de Yamakazi et al. [YM11].

Dans cet algorithme, on distingue donc trois blocs importants :

- la génération de l'échantillon initial ;
- la construction et le réglage des paramètres du modèle ;
- le choix du critère d'enrichissement.

Chacune de ces parties a notamment été développée par Queipo et al. [QHS+05], puis par Forrester et al. [FK09].

Une comparaison des différents modèles de substitution et une étude sur la bonne manière de sélectionner les paramètres influents de la construction de chacun d'entre eux a d'ores et déjà été effectuée dans le chapitre précédent. Dans ce chapitre, nous nous intéresserons donc au problème de l'échantillonnage (section 1), puis nous effectuerons une étude des différents critères d'enrichissement présents dans la littérature. Enfin, nous présenterons des applications de l'algorithme développé ici, avec chacun des six modèles du chapitre 2, à des fonctions mathématiques puis à l'optimisation de profils d'aile pour les équations d'Euler, puis RaNS.

3.1 Méthodes d'échantillonnage de l'espace des paramètres

La première étape de l'utilisation des modèles de substitution est donc le choix de l'échantillon d'apprentissage. Il n'est pas évident de déterminer cet échantillon puisque la fonction que l'on souhaite représenter est considérée comme une fonction "boîte noire" dont on ne connaît pas les caractéristiques. Les premières méthodes d'échantillonnage ont été introduites pour la compilation, puis l'analyse de données expérimentales. On parle alors de plans d'expérience (*Design of Experiments*). Comme déjà évoqué, en raison de l'influence des dispositifs expérimentaux, un même jeu de facteurs peut alors conduire à des réponses différentes. Les échantillons utilisés dans ce cadre possèdent donc des spécificités comme l'utilisation de points multiples (la réponse correspondant à un jeu de paramètres est calculée plusieurs fois) ou le placement des points afin de minimiser l'influence des perturbations aléatoires. On ne présentera pas ici ce type de méthodes mais on pourra se référer à un ouvrage classique dans ce domaine [Mon97].

Dans le cas de l'analyse des réponses d'un code de simulation (approche DACE), les données sont déterministes. En raison de la complexité des modèles utilisées, les fonctions d'intérêt sont souvent irrégulières et cette irrégularité peut émerger en raison de l'absence d'erreur expérimentale. Les plans d'expérience classiques ne sont donc pas adaptés à ce type de fonction : ils sont en effet dédiés à la modélisation par des polynômes de premier ou second degré, et concentrent les points sur les bords du domaine expérimental. Il est donc nécessaire de formuler de nouveaux objectifs d'échantillonnage pour la simulation de données issues d'un code de calcul. Dans l'article classique sur les approches DACE, Sacks et al. [SWMW89] modélisent les réponses du code de calcul comme la réalisation d'un processus stochastique. Ils introduisent alors un critère statistique pour déterminer le meilleur échantillonnage de l'espace de définition. Depuis cet article fondateur, d'autres critères du même type ont été introduits et sont résumés dans un article de référence sur le sujet [KO96].

3.1.1 Plans construits à partir de critères statistiques

Ces méthodes ont été développées exclusivement pour le krigeage. On peut citer notamment :

- le critère d'entropie [BH67], basé sur la notion d'entropie de Shannon, qui permet de mesurer la quantité d'information fournie par une expérimentation à un point donné ;
- l'erreur quadratique moyenne intégrée (*Integrated Mean Square Error* - IMSE) [BD59] qui consiste à sélectionner le plan qui minimise l'écart entre la réponse du simulateur et sa prédiction.

Ces approches ont été très utilisées au début des années 90. Elles souffrent cependant de deux inconvénients majeurs. Tout d'abord, leur construction numérique est très exigeante en ressources de calcul. Il faut en effet résoudre un problème d'optimisation de dimension Nd pour placer N points dans un espace de dimension d. De plus, les plans obtenus ne sont optimaux que pour un choix de la structure de covariance du krigeage. Ils doivent donc être calculés à nouveau dès que la valeur du paramètre ζ est modifiée. Par conséquent, d'autres plans d'expérience ont été proposés depuis. Ils sont construits de sorte à remplir correctement l'espace de définition : on parle alors de *space filling design* que l'on peut chercher à calculer soit à partir de critères déterministes, soit à partir de critères statistiques.

3.1.2 Plans construits à partir de critères déterministes

Dans ce type de méthode, la faculté de l'échantillon à bien se répartir dans l'espace est évaluée à partir d'un critère de distance. Les critères les plus utilisés dans la littérature sont les critères *maximin* et *minimax* [JMY90] :

- un plan *maximin* est un plan qui maximise la distance minimum entre deux points de l'échantillon :
$$\max_{\mathcal{E}\subset\mathcal{D}} \min_{x^i,x^j\in\mathcal{E}} d(x^i, x^j);$$
- un plan *minimax* minimise la distance maximale entre un point du domaine et un point de l'échantillon :
$$\min_{x\in\mathcal{D}} \max_{x^i\in\mathcal{E}} d(x^i, x).$$

Le plan *maximin* a tendance à concentrer les points sur les bords du domaine [JMY90]. Le plan *minimax* n'a pas cet inconvénient mais est plus complexe à générer car la distance doit en théorie être évaluée sur tous les points d'un maillage dense du domaine.

3.1.3 Méthode de remplissage de l'espace (*Space filling design*)

On s'inspire ici des méthodes d'intégration d'une fonction numérique. Pour calculer une intégrale, on peut en effet utiliser la méthode de Monte Carlo en tirant un ensemble de points X uniformément distribués dans l'espace de définition. Ainsi, le calcul de l'intégrale de la fonction $h(t)$ se ramène au calcul de l'espérance de la distribution $h(X)$. Il est alors possible de formuler un intervalle de confiance sur la valeur de l'intégrale qui dépend à la fois du nombre de tirage N et de la variance de l'estimateur. Malheureusement, on sait que les méthodes de Monte Carlo converge en $\frac{1}{\sqrt{N}}$.

Pour rendre plus rapide la convergence, on peut se tourner vers des méthodes de réduction de variance. L'échantillonnage par hypercube latin (*Latin Hypercube Sampling* - LHS) [MBC79] est l'une de ces méthodes.

Échantillonnage par hypercube latin

Cette méthode fait partie de la catégorie des méthodes d'échantillonnage stratifié qui consistent à découper l'espace des variables d'entrée en sous espaces disjoints, puis à échantillonner chacun de ces sous espaces. Dans la méthode LHS, on assure que le découpage a défini des sous espaces équiprobables, et que chaque sous espace est échantillonné de manière uniforme. Plus précisément, si l'on souhaite générer un hypercube latin de N points dans $[0,1]^d$, on se donne d permutations aléatoires uniformes indépendantes de $\{1, N\}$ dans $\{1, N\}$ que l'on notera $\Pi_j(1), \ldots, \Pi_j(N)$ avec $j \in \{1, d\}$. Les points sont alors définis par :

$$x_k^{(n)} = \frac{\Pi_k(n) - U_n^k}{N} \quad \forall k \in \{1 \ldots d\}\, \forall n \in \{1 \ldots N\},$$

où U_n^k est un ensemble de Nd variables aléatoires indépendantes et identiquement distribuées (i.i.d.) de la loi uniforme sur $[0,1[^d$. On peut montrer que la variance de l'estimateur de Monte Carlo est réduite par ce choix de l'échantillon [Ste87]. L'échantillonnage par hypercube latin

est sans doute la méthode d'échantillonnage la plus utilisée ces dernières années. Elle a connu plusieurs améliorations, consistant notamment à choisir parmi tous les hypercubes latins disponibles, celui qui minimise l'un des critères statistiques présentés ci-dessus (par exemple le critère d'entropie pour Morris et Mitchell [MM95] ou le critère IMSE pour Park [Par94]).

Les hypercubes latins permettent de garantir que les projections suivant chacun des paramètres sont uniformes. Pour généraliser cette propriété à tous les ensembles de t paramètres, on doit s'intéresser aux tableaux orthogonaux [Owe92] (*Orthogonal arrays*). Un tableau orthogonal de force t garantit que les projections sur un ensemble de t paramètres ou moins seront bien uniformes. Cette classe de méthodes inclut donc les hypercubes latins qui sont des tableaux orthogonaux de force 1. Cependant, ces tableaux sont extrêmement complexes à construire.

Ce type de plans d'expérience a donc la propriété de garder une bonne qualité lors de la diminution de la dimension de l'espace. Néanmoins, ils ne sont pas forcément satisfaisants en terme de remplissage de l'espace. Ainsi, malgré la réduction de variance, la convergence de l'erreur d'estimation est toujours en $\frac{1}{\sqrt{N}}$. Pour des méthodes à convergence plus rapide, il faut alors se tourner vers les méthodes de quasi-Monte Carlo (QMC).

Méthodes de quasi-Monte Carlo

Les méthodes de quasi-Monte Carlo consistent à remplacer le tirage aléatoire des points de l'échantillon par des suites déterministes réparties uniformément dans l'espace. Ces suites sont définies à partir d'un critère appelé discrépance, qui permet de quantifier l'écart entre celles-ci et la répartition uniforme. On peut montrer que l'estimation d'erreur de la méthode de quasi-Monte Carlo est meilleure que celle de la méthode de Monte Carlo, et converge plus rapidement en utilisant la borne de Koksma-Hlawka [Nie92]. L'erreur d'intégration dépend alors de la variation de la fonction et de la discrépance de la suite utilisée. Parmi les suites à discrépance faible, on trouve notamment les suites de Hammersley et de Halton qui ont été utilisées pour l'approximation de fonction.

Échantillonnage de Hammersley

La méthode d'échantillonnage de Hammersley a été présentée dans un article de Kalagnanam et Diwekar [KD97]. Il s'agit de choisir N points dans un espace de dimension d. Chacun de ces points est calculé par la formule suivante :

$$x^{(n)} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} - \begin{bmatrix} \frac{n}{N} \\ \Phi_{R_1}(n) \\ \Phi_{R_2}(n) \\ \vdots \\ \Phi_{R_{d-1}}(n) \end{bmatrix} \quad \forall n \in \{1 \dots N\}$$

où $R_1, \dots, R_{d-1}$ sont les $d-1$ premiers nombres premiers et $\Phi_R(n) = n_0 R^{-1} + n_1 R^{-2} + \dots + n_m R^{-m-1}$ avec n_p les composantes de l'écriture de n en base R ($n = n_0 + n_1 R + n_2 R^2 + \dots + n_m R^m$, où m désigne la puissance maximale requise pour cette écriture). On peut montrer que la discrépance de la suite de Hammersley diminue en $O\left(\frac{\log N^d}{N}\right)$ [DR84].

Échantillonnage de Halton

L'un des inconvénients de la méthode de Hammersley est que la modification du nombre de points de l'échantillon nécessite de recalculer la nouvelle position de l'ensemble des points. Pour parer à ce problème, Rafajlowicz et Schwabe [RS06] proposent l'échantillonnage de Halton, défini de la manière suivante :

$$x^{(n)} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} \Phi_{R_1}(n) \\ \Phi_{R_2}(n) \\ \vdots \\ \Phi_{R_{d-1}}(n) \\ \Phi_{R_d}(n) \end{bmatrix} \quad \forall n \in \{1 \dots N\}$$

où $R_1, \dots, R_d$ sont les d premiers nombres premiers et $\Phi_R(n) = n_0 R^{-1} + n_1 R^{-2} + \dots + n_m R^{-m-1}$ avec n_p les composantes de l'écriture de n en base R ($n = n_0 + n_1 R + n_2 R^2 + \dots + n_m R^m$, où m désigne la puissance maximale requise pour cette écriture). La discrépance de la suite de Halton évolue en $O\left(\frac{\log N^{d-1}}{N}\right)$ [DR84].

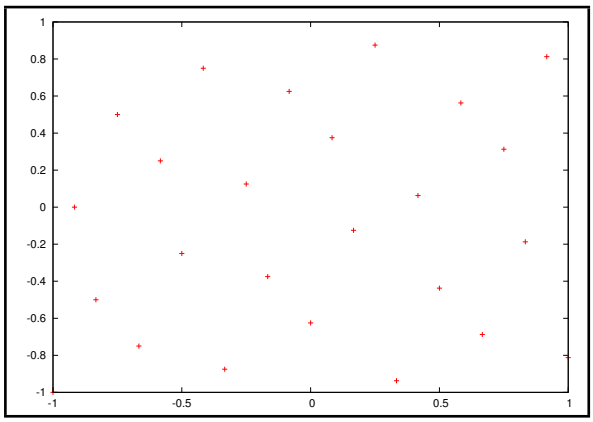


FIG. 3.1 – Échantillonnage de Hammersley (25 points) de $[-1, 1]^2$

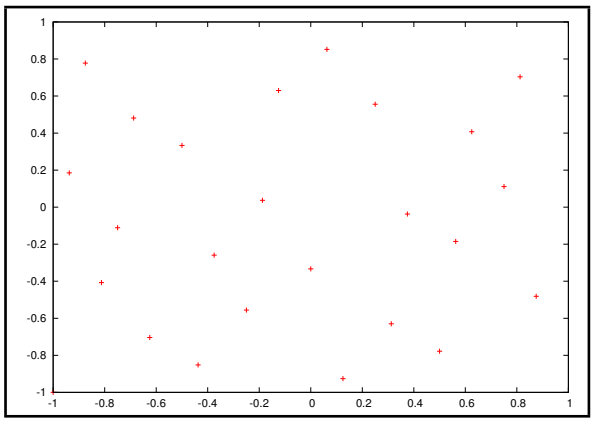


FIG. 3.2 – Échantillonnage de Halton (25 points) de $[-1, 1]^2$

Uniform Design

Au lieu d'utiliser des suites à discrépance faible, on peut chercher à les générer par un algorithme d'optimisation. Ainsi, pour positionner N points dans un espace de dimension d, on pourra chercher à minimiser une fonction de discrépance. Cette approche est extrêmement coûteuse et requiert la résolution d'un problème d'optimisation de dimension Nd. En pratique, on préfèrera effectuer cette minimisation sur un ensemble discret de plans. Par exemple, on pourra chercher parmi les hypercubes latins, le plan qui dispose de la plus faible discrépance. La question du type de discrépance à utiliser n'est pas une question facile, et de nombreux travaux de recherche ont été menés sur ce thème ces derniers années. On pourra se reporter à [FLWZ00] pour plus d'informations.

Comparaison des méthodes d'échantillonnage

Plusieurs études ont été menées pour comparer les méthodes d'échantillonnage de la littérature. Par exemple, Simpson et al. [SLC01] ont effectué des expérimentations sur des problèmes d'ingénierie en dimension 3 et 14. Ils concluent que les méthodes de Hammersley et les méthodes de type *Uniform Design* semblent surpasser les hypercubes latins, les méthodes *Orthogonal Array* et la méthode de tirage aléatoire. En raison de la difficulté de construction des *Uniform Design*, on utilisera donc par la suite les échantillonnages de Hammersley et de Halton.

3.2 Enrichissement de l'échantillon pour l'optimisation

Une fois déterminée la méthode de construction de l'échantillon initial, un premier modèle de substitution peut être construit. Pour optimiser la fonction, il est indispensable que l'approximation de la fonction autour des optima de la fonction exacte soit la plus précise possible. L'étape suivante est donc d'identifier les zones intéressantes afin d'y enrichir l'échantillon. Une fois le point le plus prometteur identifié, il sera évalué par le modèle physique (évaluation exacte) et ajouté à l'échantillon d'apprentissage. Le modèle de substitution sera alors mis à jour en fonction de ces nouvelles informations et le processus sera réitéré jusqu'à satisfaction d'un critère d'arrêt. On peut donc résumer cette approche par le schéma de la figure 3.3.

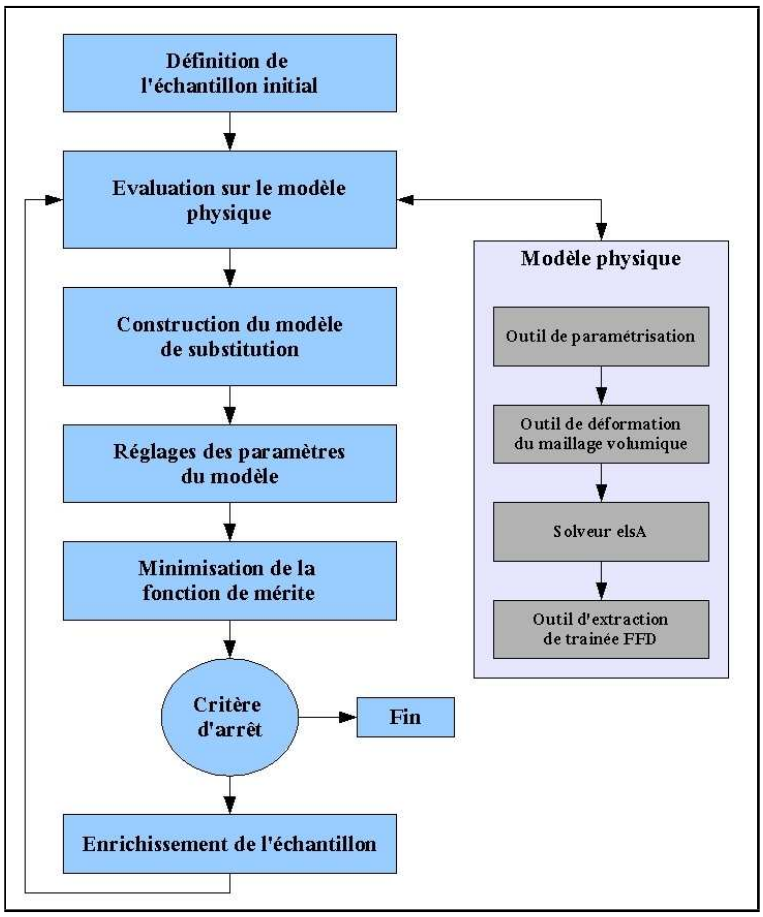


Fig. 3.3 – Schéma d'un algorithme d'optimisation basé sur des modèles de substitution

Il reste à déterminer le critère utilisé pour l'enrichissement. Le choix de celui-ci est crucial, puisque chaque nouveau point ajouté à l'échantillon nécessite un calcul exact de la fonction à optimiser. Il n'est donc pas souhaitable d'ajouter des points qui n'apportent pas d'informations significatives sur la fonction d'intérêt, ce qui conduirait à gâcher des ressources de calcul. Plusieurs critères ont été proposés et comparés dans la littérature. On pourra notamment se reporter aux articles de Jones et al. [JSW98] et Jones [Jon01] qui semblent constituer la référence dans le domaine.

Dans ces articles, Jones distingue deux types de critère d'enrichissement :

1. approche deux étapes (*Two-stage*) : le modèle de substitution est construit à partir de l'échantillon courant, puis le critère est formulé à partir du modèle ;
2. approche une étape (*One-stage*) : le critère est calculé à partir des fondements mathématiques du modèle de substitution sans qu'il soit nécessaire de le construire, ce qui permet de réduire le temps de calcul nécessaire pour localiser les points prometteurs.

Dans notre étude, le coût de construction du modèle est négligeable par rapport au calcul d'un seul point de l'échantillon. Par conséquent, nous avons choisi de nous pencher ici sur les approches *Two-stage*. Reste à déterminer le critère d'enrichissement.

3.2.1 Enrichissement par l'optimum

L'idée la plus intuitive consiste à choisir l'optimum du modèle de substitution comme point à ajouter à l'échantillon. Cela revient à choisir le modèle lui-même comme critère et à l'optimiser par exemple par des méthodes numériques d'optimisation. Cette approche a notamment été utilisée dans l'étude de Peter et al. [PMBP07] (krigeage, réseaux RBF, réseaux de neurones, approximation polynomiale) ou encore dans celle de Leary et al. [LBK04b] (krigeage, réseaux RBF, co-krigeage et réseaux RBF avec gradients). Si cette méthode a l'avantage de ne pas nécessiter de développements supplémentaires, elle est en fait assez mal adaptée à l'optimisation de fonctions complexes car rien ne permet d'indiquer que le minimum du modèle de substitution correspond bien à un minimum de la fonction d'intérêt. On pourra notamment trouver un contre exemple simple sur la figure 3.4.

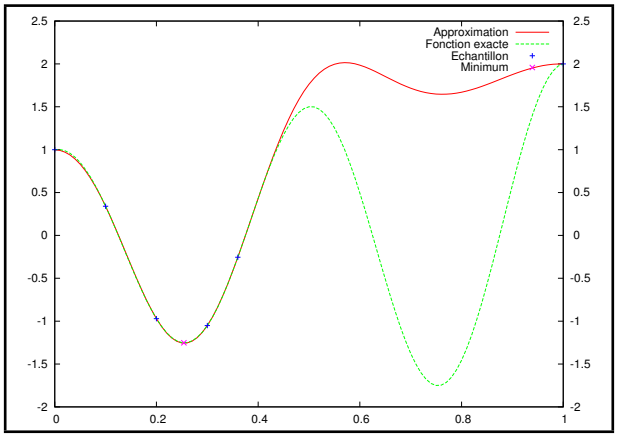


FIG. 3.4 – Échec de l'enrichissement par le minimum de la fonction objectif

Il est donc parfois inadapté d'utiliser l'optimum de la fonction approchée pour enrichir l'échantillon. Il devient alors nécessaire de se tourner vers des approches statistiques. Celles-ci sont basées sur le fait que le krigeage (et par approximation tous les modèles présentés précédemment) fournit en plus de l'approximation de la fonction objectif (la moyenne statistique) une estimation statistique de l'erreur d'approximation (l'écart type). C'est cette estimation que l'on peut utiliser pour l'enrichissement, tout en gardant à l'esprit que celle-ci n'est qu'une estimation statistique d'une quantité déterministe.

3.2.2 Enrichissement par des méthodes statistiques

Ces méthodes d'enrichissement ont tout d'abord été introduites dans le cadre du krigeage par les travaux de Schonlau [Sch97], Cox et John [CJ97], Jones [JSW98] [Jon01] ou encore Sobester et al. [SLK05]. Elles sont basées sur l'écart type dont le calcul est évident dans le cadre du krigeage, et a été étendu aux autres modèles de substitution. Pour chaque modèle, ce calcul est présenté ci-dessous et l'écart type est représenté sur les figures 3.5 à 3.10 pour la fonction $f(x) = \cos 4\pi x$ sur $[0, 1]$ (l'axe des ordonnées à gauche représente la valeur de la fonction exacte et de son approximation tandis que celui à droite donne la valeur de l'écart type).

Calcul de l'écart type des modèles de substitution

Krigeage On peut calculer facilement la variance de l'approximation faite par le krigeage :

$$\begin{aligned} \sigma_k^2(x) &= Var(\hat{f}(x) - f(x)) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda^i \lambda^j \phi(x^i, x^j) + \phi(x, x) - 2\sum_{i=1}^{N} \lambda^i \phi(x^i, x). \end{aligned}$$

Alors on peut écrire :

$$\sigma_k^2(x) = \Lambda^T \Phi \Lambda + \phi(x, x) - 2\Lambda^T \phi_x - 2\nu + 2\nu.$$

Or $\Lambda = \Phi^{-1}\phi_x$ donc :

$$\sigma_k^2(x) = 1 - \phi_x^T \Phi^{-1} \phi_x.$$

Réseaux de fonctions à base radiale Sobester et al. [SLK05] proposent une formule d'estimation de l'écart type de l'erreur statistique de la méthode RBF en faisant l'hypothèse que les valeurs déterministes y^i utilisées pour la construction du modèle sont en fait la réalisation d'un processus stochastique. Gibbs [Gib97] donne alors la formule suivante :

$$\sigma_k^2(x) = 1 - \phi_x^T \Phi^{-1} \phi_x.$$

Régression à vecteurs de support Gao et al. [GGH02] ont introduit une formule d'estimation de l'écart type de l'erreur en modélisant la régression par une méthode SVR comme un processus gaussien multidimensionnel. La variance de l'erreur est alors la somme de deux

termes :

$$\sigma_k^2(x) = \sigma_s^2(x) + \sigma_n^2(x).$$

Le terme $\sigma_s^2(x)$ modélise la variance de l'erreur due à l'approximation fonctionnelle de la méthode SVR :

$$\sigma_s^2(x) = \phi(x,x) - \phi_{USV}(x)^T \Phi_{USV}^{-1} \phi_{USV}(x).$$

où Φ_{USV} et ϕ_{USV} sont respectivement la matrice Φ de la méthode SVR et le vecteur ϕ_x (défini par $\phi_x = \left[\phi(x^i, x)\right]_{i\in\{1,N\}}$) où seules les lignes et les colonnes correspondant à des vecteurs de support non bornés (c'est à dire les $x^t, \quad t \in \{1 \dots N\}$ tels que $0 < \alpha^t + \alpha^{t*} < C_0$) sont conservées.

Enfin, le terme $\sigma_n^2(x)$ est due à la variabilité intrinsèque des données :

$$\sigma_n^2(x) = \epsilon_0^2 \left[\frac{2}{(\epsilon_0 C_0)^2} + \frac{\epsilon_0 C_0 + 3}{3(\epsilon_0 C_0 + 1)} \right].$$

Co-krigeage On cherche à calculer la variance de l'approximation faite par le Co-krigeage :

$$\begin{aligned}
\sigma_k^2(x) &= Var(\hat{f}(x) - f(x)) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda^i \lambda^j \phi(x^i, x^j) + \sum_{k=1}^{d}\sum_{l=1}^{d}\sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_k^i \lambda_l^j \phi_{kl}(x^i, x^j) + Var(f(x)) - 2\sum_{i=1}^{N} \lambda^i \phi(x^i, x) \\
&\quad -2\sum_{k=1}^{d}\sum_{i=1}^{N} \lambda_k^i \phi_{k0}(x^i, x).
\end{aligned}$$

Alors on peut écrire :

$$\sigma_k^2(x) = \Lambda^T \Phi \Lambda + \phi(x,x) - 2\Lambda^T \phi_x.$$

Or $\Lambda = \Phi^{-1}\phi_x$ donc :

$$\sigma_k^2(x) = 1 - \phi_x^T \Phi^{-1} \phi_x.$$

Réseaux RBF avec gradients On étend sans difficulté la formule d'estimation de l'écart type aux réseaux RBF avec gradients, puisque la prise en compte des dérivées revient seulement à ajouter de nouvelles fonctions de bases et de nouvelles données. Par conséquent :

$$\sigma_k^2(x) = 1 - \phi_x^T \Phi^{-1} \phi_x.$$

SVR avec gradients De la même manière, la formule d'estimation de σ_k^2 dans le cas de la régression SVR se généralise à la régression SVR avec gradients :

$$\sigma_k^2(x) = \sigma_s^2(x) + \sigma_n^2(x).$$

$$\sigma_s^2(x) = \phi(x,x) - \phi_{USV}(x)^T \Phi_{USV}^{-1} \phi_{USV}(x).$$

où Φ_{USV} et ϕ_{USV} sont respectivement la matrice Φ et le vecteur ϕ_x de la régression SVR avec gradients où seules les lignes et les colonnes correspondant à des vecteurs de support non bornés

sont conservées.

Enfin, le terme $\sigma_n^2(x)$ est due à la variabilité intrinsèque des données :

$$\sigma_n^2(x) = \epsilon_0^2 \left[\frac{2}{(\epsilon_0 C_0)^2} + \frac{\epsilon_0 C_0 + 3}{3(\epsilon_0 C_0 + 1)} \right].$$

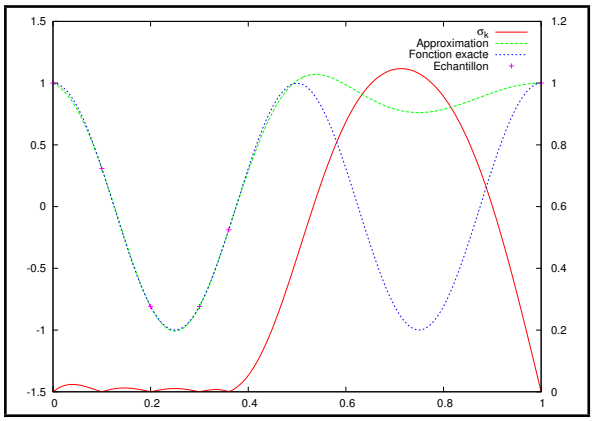


Fig. 3.5 – Écart type de l'erreur statistique du krigeage ($f(x) = \cos 4\pi x$)

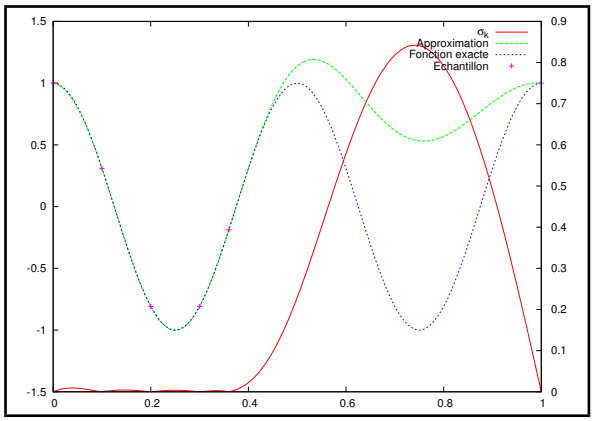


Fig. 3.6 – Écart type de l'erreur statistique du réseau RBF ($f(x) = \cos 4\pi x$)

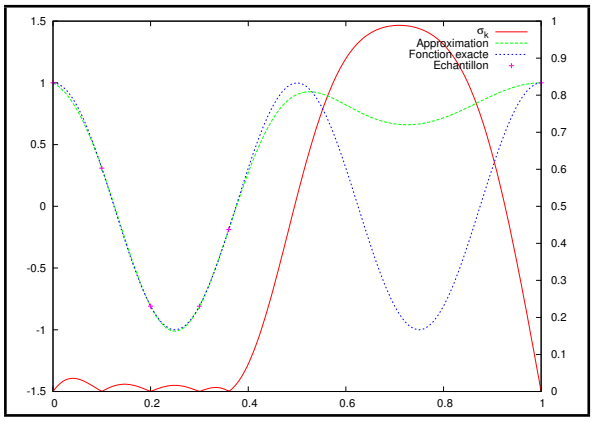


Fig. 3.7 – Écart type de l'erreur statistique de la régression SVR ($f(x) = \cos 4\pi x$)

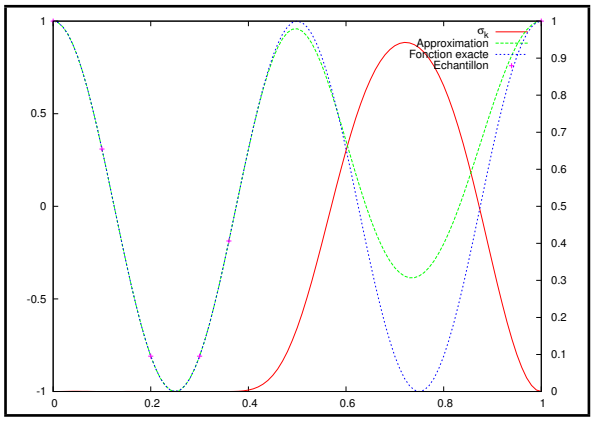


Fig. 3.8 – Écart type de l'erreur statistique du co-krigeage ($f(x) = \cos 4\pi x$)

Exploration et exploitation

Des limites de l'enrichissement par l'optimum, on peut déduire qu'un bon critère doit satisfaire deux objectifs contradictoires :

- il doit apporter des informations dans les optima d'ores et déjà identifiés par le modèle de substitution. On parlera dans ce cas d'une capacité d'exploitation du critère ;
- il doit également explorer des zones de l'espace peu connues dans lesquelles pourrait se trouver un minimum de la fonction exacte non capté par le modèle de substitution en raison d'un échantillon inadapté. Il s'agit de la capacité d'exploration du modèle.

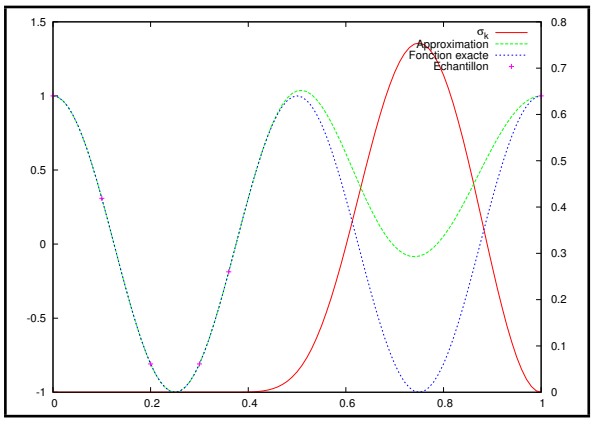


FIG. 3.9 – Écart type de l'erreur statistique de la méthode GA-RBF ($f(x) = \cos 4\pi x$)

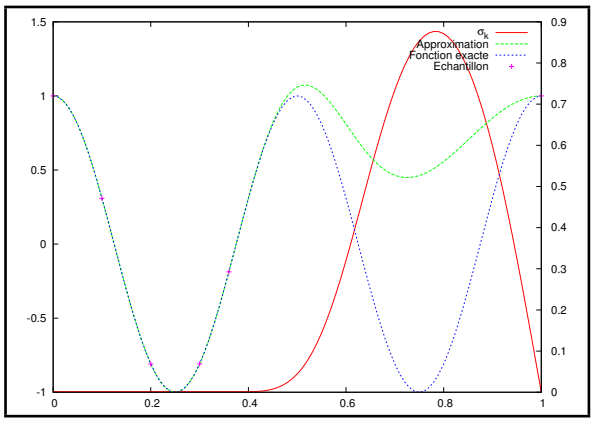


FIG. 3.10 – Écart type de l'erreur statistique de la méthode GA-SVR ($f(x) = \cos 4\pi x$)

Un bon critère doit donc combiner ces deux capacités et donc être formulé à la fois à partir de la fonction donnée par le modèle (qui donne des informations sur les zones à exploiter) et à partir de l'écart type calculé par les formules ci-dessus (qui donne des informations sur les zones à explorer). Plusieurs formulations ont été proposées dans la littérature.

Minimisation d'une borne inférieure de confiance (*Lower Confidence Bound* - LCB)

Ce critère a été introduit par Cox et John [CJ97] et consiste à minimiser la borne inférieure de confiance définie par :

$$LCB(x) = \hat{f}(x) - \kappa\sigma_k(x). \tag{3.1}$$

La valeur de κ permet d'arbitrer entre la capacité d'exploration et la capacité d'exploitation. Ainsi, des valeurs faibles de κ conduisent à privilégier la fonction au détriment de l'écart type et donc à exploiter en priorité les zones intéressantes déjà identifiées. Au contraire, de fortes valeurs de κ conduisent à ajouter davantage de points dans les zones de l'espace peu explorées.

Maximisation de la probabilité d'amélioration (*Probability of Improvement* - PI)

Une seconde possibilité est d'utiliser la notion de probabilité d'amélioration introduite par Kushner dès 1964 [Kus64] pour rechercher l'optimum d'une fonction unidimensionnelle en présence de bruit. Il s'agit de considérer à nouveau la valeur de l'estimation comme la réalisation d'une variable aléatoire de moyenne $\hat{f}(x)$ et d'écart type $\sigma_k(x)$. La probabilité d'amélioration est alors la probabilité que la valeur de l'estimation soit inférieure à une valeur objectif donnée T. Cette valeur sera choisie inférieure au minimum calculé sur l'échantillon d'apprentissage actuel (notons le $y_{min} = \min(y^i, i \in \mathcal{E})$). Si par exemple, on choisit $T = y_{min} - 0.25|y_{min}|$, alors on recherche la probabilité d'amélioration de 25%. En considérant que le variable aléatoire suit la loi normale, on peut alors formuler la probabilité d'amélioration par :

$$PI(x) = \Psi\left(\frac{T - \hat{f}(x)}{\sigma_k(x)}\right), \tag{3.2}$$

où $\Psi(u) = \frac{1}{2}\left[1 + err(u)\right]$ est la fonction de répartition de la loi normale centrée réduite. Selon Jones [Jon01], la difficulté de cette méthode provient du fait qu'elle est extrêmement sensible au choix de l'objectif T. Si T est trop loin de y_{min}, la recherche sera en priorité locale et l'exploration d'autres zones de l'espace ne sera faite qu'à partir du moment où le voisinage du minimum courant aura été exploré en détails. A l'inverse, si T est grand, l'algorithme aura tendance à sauter de bassins en bassins sans explorer suffisamment chacune des potentialités d'amélioration. Jones propose alors deux pistes pour améliorer cette méthode. La première consiste à utiliser cette méthode plusieurs fois, c'est à dire à enrichir l'échantillon de plusieurs points correspondant à l'optimum du critère pour plusieurs valeurs de T. La seconde consiste à se tourner vers les critères d'amélioration espérée qui font l'objet de la prochaine section.

Maximisation de l'amélioration espérée (*Expected Improvement* - EI)

Cette méthode a été introduite par Schonlau [Sch97]. Il s'agit de quantifier l'amélioration qu'un point pourrait apporter. On considère à nouveau que la valeur de l'estimation est la réalisation d'une variable aléatoire $Y(x)$ normalement distribuée, de moyenne $\hat{f}(x)$ et d'écart type $\sigma_k(x)$. Une amélioration de valeur I sera obtenue si $Y(x) = y_{min} - I$ où y_{min} identifie le minimum calculé sur l'échantillon courant. Alors, la probabilité de réaliser cette amélioration est donnée par la fonction de masse (densité de probabilité) de la loi normale :

$$P(Y(x) = f_{min} - I) = \frac{1}{\sqrt{2\pi}\sigma_k(x)} \exp\left[-\frac{(y_{min} - I - \hat{f}(x))^2}{2\sigma_k(x)^2}\right].$$

En intégrant par partie cette probabilité pour toutes les valeurs d'amélioration (donc seulement les valeurs positives de I), on obtient l'amélioration espérée :

$$EI(x) = \begin{cases} (y_{min} - \hat{f}(x))\Psi(\frac{y_{min} - \hat{f}(x)}{\sigma_k(x)}) + \sigma_k(x)\psi(\frac{y_{min} - \hat{f}(x)}{\sigma_k(x)}) & si \quad \sigma_k(x) > 0 \\ 0 & si \quad \sigma_k(x) = 0 \end{cases} \tag{3.3}$$

où $\Psi(u) = \frac{1}{2}\left[1 + err(u)\right]$ est la fonction de répartition de la loi normale centrée réduite et $\psi(u) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}u^2\right]$ est donc la densité de probabilité. Ce critère a l'avantage de ne pas nécessiter que soit fixé un niveau d'amélioration attendu comme la méthode précédente. De plus, Locateli [Loc97] a démontré la convergence de l'optimisation globale basée sur ce critère. Cette convergence peut néanmoins être extrêmement lente si l'échantillon de départ conduit à la construction d'un modèle avec un écart type très faible. C'est pour cette raison que Sobester et al. [SLK05] ont introduit l'amélioration espérée pondérée. Il s'agit de distinguer dans l'équation 3.3 le terme d'exploration et le terme d'exploitation et d'ajouter une pondération pour privilégier l'une ou l'autre de ces propriétés :

$$WEI(x) = \begin{cases} w(y_{min} - \hat{f}(x))\Psi(\frac{y_{min} - \hat{f}(x)}{\sigma_k(x)}) + (1-w)\sigma_k(x)\psi(\frac{y_{min} - \hat{f}(x)}{\sigma_k(x)}) & si \quad \sigma_k(x) > 0 \\ 0 & si \quad \sigma_k(x) = 0 \end{cases} \tag{3.4}$$

On remarque que choisir $w = 0,5$ conduit à $WEI(x) = 0,5EI(x)$ et que les valeurs $w = 0$ et $w = 1$ conduisent respectivement à des critères d'exploration pure et d'exploitation pure (voir

les représentations des figures 3.11 à 3.16, toujours sur la fonction $f(x) = \cos 4\pi x$, avec l'axe des ordonnées à gauche représentant la valeur de la fonction exacte et de l'approximation, et celui de droite la valeur du critère pour $w = 0$ et $w = 1$). Dans leur étude sur le sujet, Sobester et al. [SLK05] ont effectué de nombreuses expérimentations en faisant varier la valeur du coefficient de pondération w. Finalement, ils recommandent d'utiliser une approche cyclique, c'est à dire de prendre successivement les valeurs de w parmi $[0, 1, 0, 3, 0, 5, 0, 7, 0, 9]$ au cours des itérations. C'est cette approche qui a été retenue dans cette étude et qui sera utilisée par la suite.

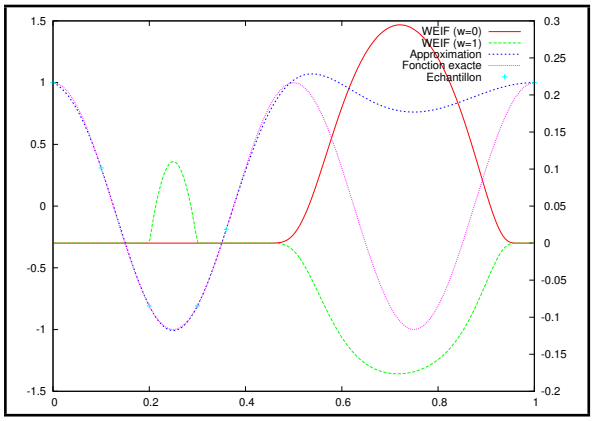


FIG. 3.11 – Critère WEI pour le krigeage ($f(x) = \cos 4\pi x$)

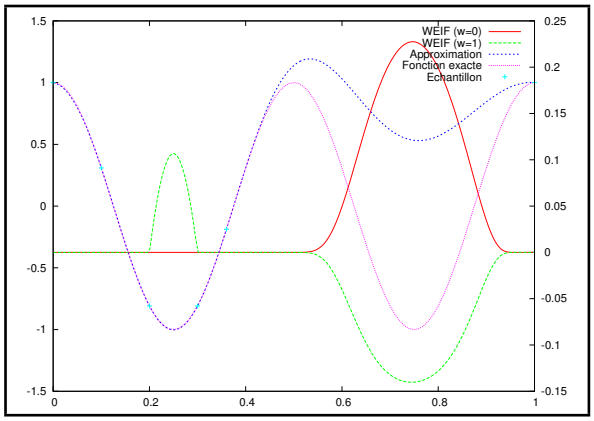


FIG. 3.12 – Critère WEI pour le réseau RBF ($f(x) = \cos 4\pi x$)

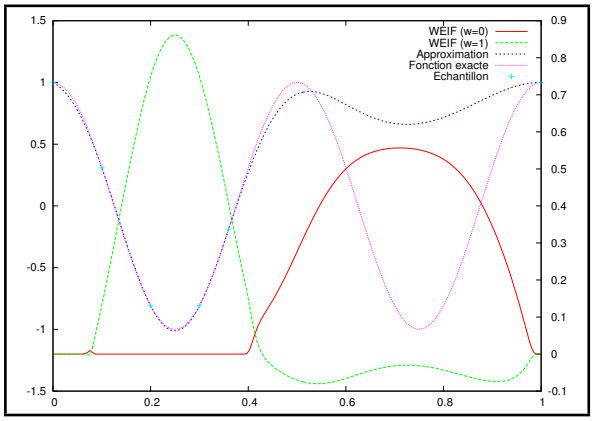


FIG. 3.13 – Critère WEI pour la régression SVR ($f(x) = \cos 4\pi x$)

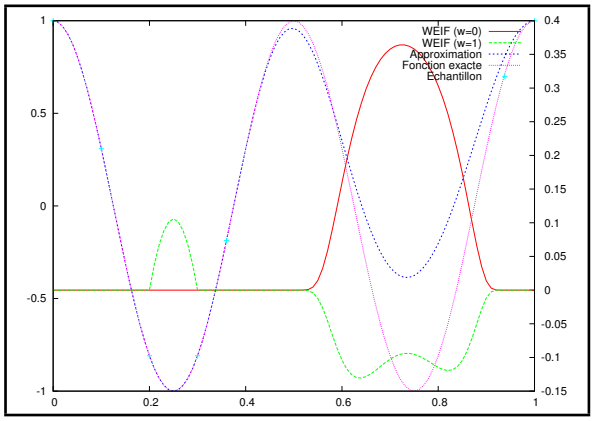


FIG. 3.14 – Critère WEI pour le co-krigeage ($f(x) = \cos 4\pi x$)

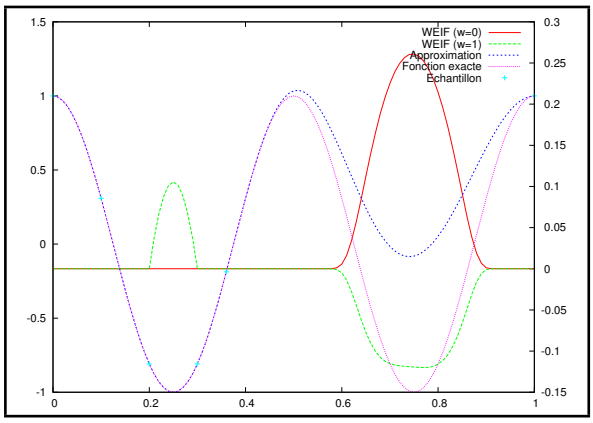


FIG. 3.15 – Critère WEI pour le réseau RBF avec gradients ($f(x) = \cos 4\pi x$)

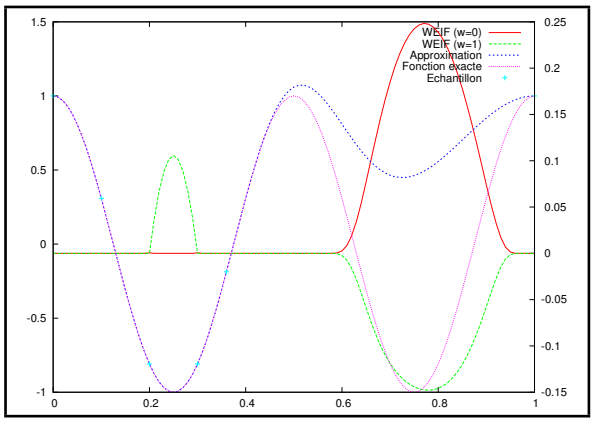


FIG. 3.16 – Critère WEI pour la régression SVR avec gradients ($f(x) = \cos 4\pi x$)

3.3 Applications à des fonctions mathématiques

3.3.1 Présentation des cas tests

Les applications à des fonctions mathématiques seront réalisées sur les fonctions suivantes :

– la fonction de Rosenbrock, déjà utilisée précédemment :

$$f(x) = \sum_{k=1}^{d-1} \left[(1 - x_k)^2 + 100(x_{k+1} - x_k^2)^2 \right] ;$$

– la fonction de Rastrigin (figure 3.17), autre fonction test classique en optimisation globale :

$$f(x) = \sum_{k=1}^{d} \left[x_k^2 - \cos 2\pi x_k + 10 \right] ;$$

– enfin, la fonction de Freudenstein et Roth (figure 3.18) :

$$f(x) = \left[-13 + x_1 + ((5 - x_2)x_2 - 2)x_2 \right]^2 + \left[29 + x_1 + ((x_2 + 1)x_2 - 14)x_2 \right]^2 .$$

3.3.2 Précision de l'approximation

On propose d'évaluer la précision des modèles de substitution pour l'approximation de fonctions mathématiques utilisées classiquement pour l'optimisation globale. L'erreur d'approximation est calculée sur un échantillon de validation $\mathcal{E}_\mathcal{V}$ par :

$$E(\hat{f})^{\mathcal{E}_\mathcal{V}} = \sqrt{\frac{1}{N_V} \sum_{i=1}^{N_V} (\hat{f}(x^i) - y^i)^2}.$$

Dans la littérature, cette quantité est connue sous le nom d'erreur RMSE (*Root Mean Square Error*). Ici, elle est calculée sur une grille $\mathcal{E}_\mathcal{V}$ dont la taille dépend de la dimension de l'espace (101×101 points en dimension 2 et $11 \times 11 \times 11 \times 11$ points en dimension 4) pour un nombre de

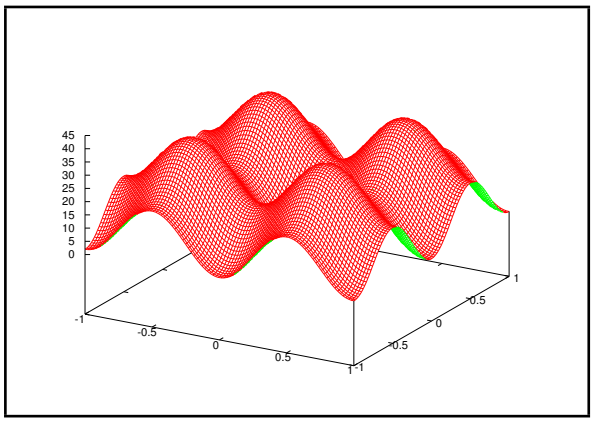


FIG. 3.17 – Fonction de Rastrigin en 2D

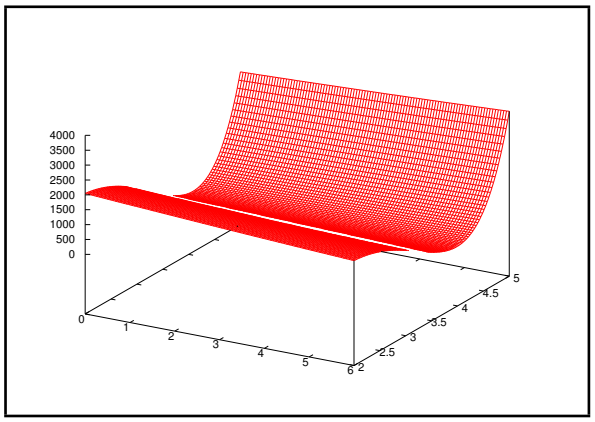


FIG. 3.18 – Fonction de Freudenstein et Roth

points d'apprentissage (l'échantillon sera noté $\mathcal{E}_{\mathcal{A}}$) compris entre 3 et N_A points, où la valeur de N_A dépend de la dimension de l'espace et de la fonction test considérée. Afin de ne pas mettre à jour l'ensemble des points lorsque la dimension de l'échantillon d'apprentissage augmente, celui-ci est générée à partir des points de Halton définis ci-dessus. Les résultats de l'approximation des fonctions de Rastrigin en dimension 2 ($N_A = 40$), puis de Rosenbrock en dimension 4 ($N_A = 80$) sont donnés respectivement sur les figures 3.19 et 3.20.

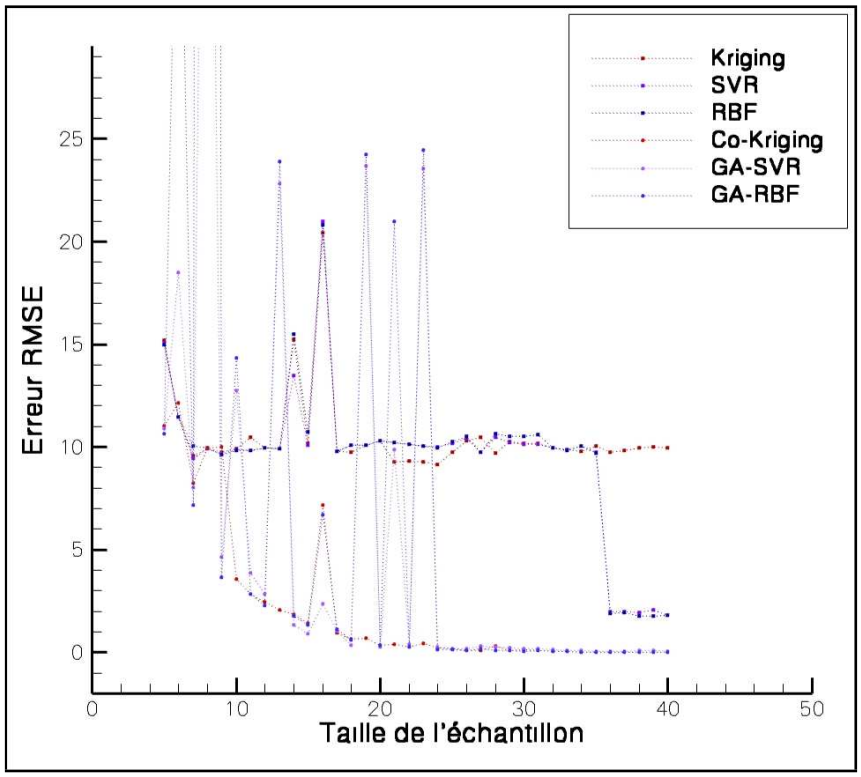


FIG. 3.19 – Approximation de la fonction de Rastrigin en 2D

On remarque que l'évolution de l'erreur RMSE est assez chaotique. L'ajout d'un point à un échantillon d'apprentissage ne garantit pas forcément la diminution de l'erreur RMSE. Cela est notamment dû au fait qu'à chaque modification de $\mathcal{E}_{\mathcal{A}}$, toutes les étapes de construction du

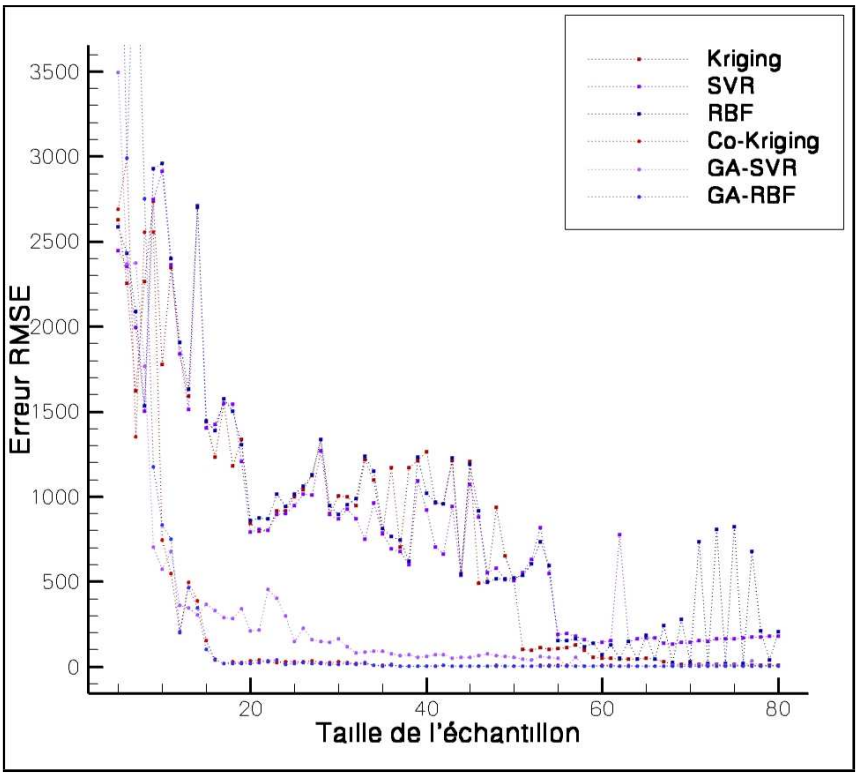


Fig. 3.20 – Approximation de la fonction de Rosenbrock en 4D

modèle, et en particulier l'optimisation des paramètres internes, sont effectuées à nouveau. Il est donc possible d'obtenir des modèles très différents lorsqu'on ajoute un point à l'échantillon d'apprentissage. Néanmoins, on peut faire trois remarques :

- tout d'abord, l'erreur RMSE est globalement décroissante : plus l'échantillon d'apprentissage est fourni et plus le modèle sera précis ;
- par ailleurs, les trois modèles présentés dans le chapitre précédent semblent donner des résultats similaires en terme de précision ;
- enfin, les méthodes avec gradients approchent plus précisément la fonction et requièrent donc un nombre de points d'apprentissage plus faible pour obtenir une précision similaire aux méthodes sans gradient.

3.3.3 Optimisation

L'algorithme d'optimisation retenu est donc composé de la méthode d'échantillonnage de Hammersley, d'un modèle de substitution et d'une procédure d'enrichissement basée sur la maximisation de l'amélioration espérée pondérée (WEI). On applique cet algorithme pour le krigeage et le co-krigeage, les réseaux de fonctions RBF avec ou sans gradient et la régression SVR avec ou sans gradient. Pour chaque cas test, on compare les résultats de l'algorithme avec ceux d'une méthode classique d'optimisation globale (méthode d'évolution différentielle) avec les mêmes paramètres numériques. L'échantillon initial est composé de $10(d-1)$ points pour les méthodes sans gradient et de $5(d-1)$ points pour les méthodes avec gradients. L'optimisation des paramètres internes du modèle est effectuée à chaque itération de l'algorithme d'optimisa-

tion par minimisation de l'erreur *leave-one-out* par une méthode d'évolution différentielle : le nombre de particules est fixé à 6 fois le nombre de paramètres, et 2000 itérations de l'algorithme sont effectuées. Ce même algorithme est utilisé pour la maximisation de la fonction de mérite (critère WEI) avec $10d$ particules et 2000 itérations. Enfin, la valeur de la pondération du critère WEI parcours un cycle sur 5 itérations conformément à ce qui a été présenté ci-dessus. Cette méthode a été appliquée aux fonctions de Rosenbrock en dimension 2 et 5, à la fonction de Rastrigin en dimension 2 et à la fonction de Freudenstein et Roth. L'évolution de l'optimum (pour chacune des fonctions, le minimum exact est zéro) en fonction du nombre d'évaluations exactes est donnée respectivement sur les figures 3.21, 3.22, 3.23 et 3.24. Le critère d'arrêt est atteint lorsque le pourcentage d'erreur sur le minimum déterminé par la méthode est inférieur à un pour mille pour les fonctions en dimension 2 et un pour cent pour les dimensions supérieures.

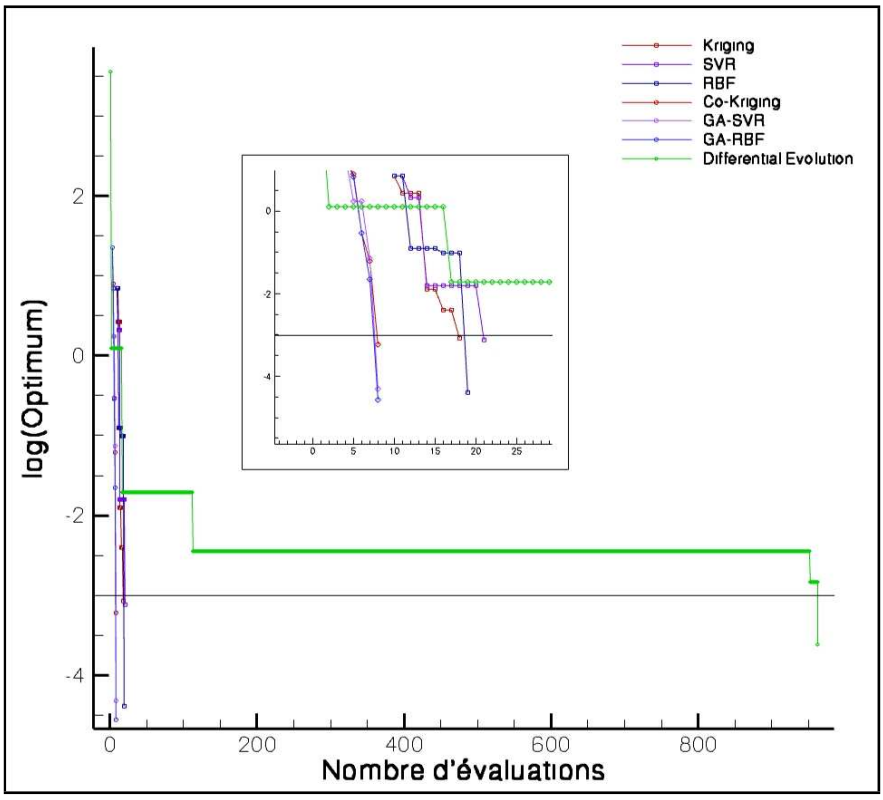


FIG. 3.21 – Optimisation de la fonction de Rosenbrock en 2D

Comme on peut le constater, l'utilisation des modèles de substitution permet de réduire sensiblement le nombre d'évaluations de la fonction objectif pour atteindre le minimum avec la même précision. De plus, les modèles avec gradients accélèrent encore sensiblement la procédure d'optimisation et l'intérêt de ceux-ci augmentent avec la dimension du problème. Ces optimisations ont été effectuées pour plusieurs initialisations différentes du générateur de nombres aléatoires, et à chaque fois les résultats sont d'un ordre de grandeur similaire. On remarque par ailleurs qu'il est difficile de dire qu'un type de modèle est le plus précis, car les performances sont dépendantes du problème considéré.

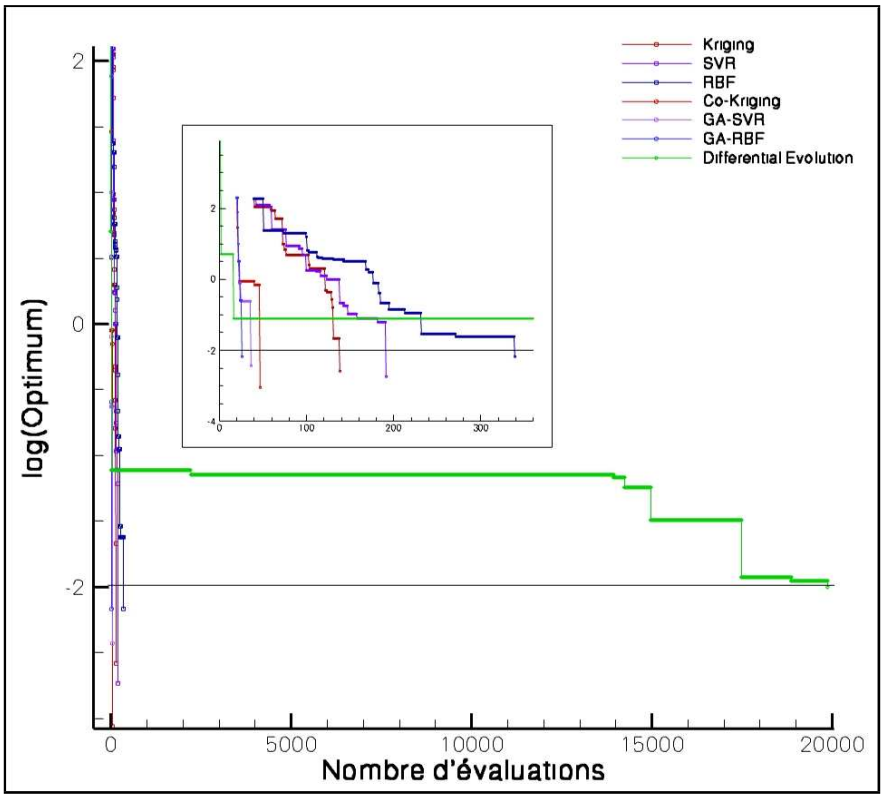


Fig. 3.22 – Optimisation de la fonction de Rosenbrock en 5D

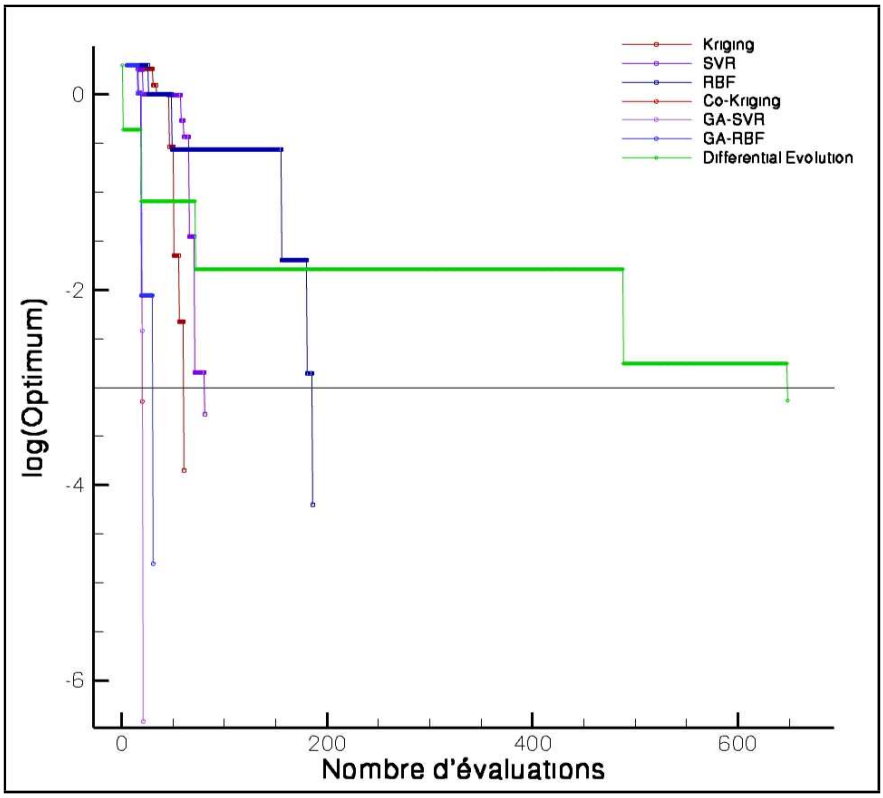


Fig. 3.23 – Optimisation de la fonction de Rastrigin en 2D

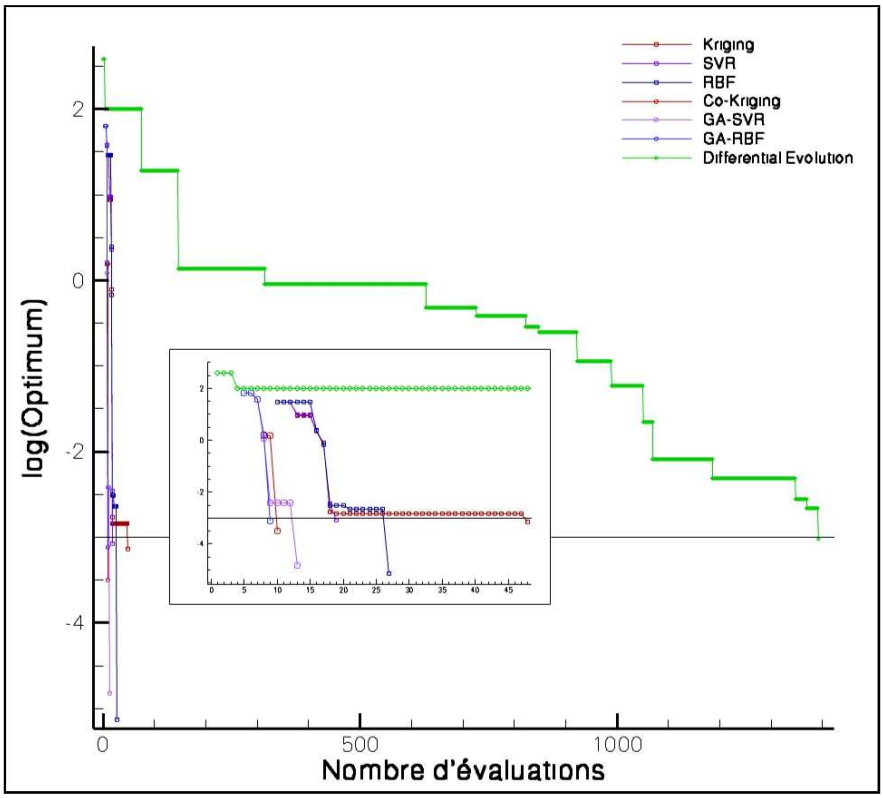


FIG. 3.24 – Optimisation de la fonction de Freudenstein et Roth

3.4 Applications à l'optimisation de forme aérodynamique

3.4.1 Écoulements de fluide parfait autour du profil NACA64A212

Les premières applications en aérodynamique sont effectuées autour d'un profil NACA64A212, avec un nombre de Mach à l'infini $M_\infty = 0.73$ et un angle d'attaque initial de $AoA = 2.5$ degrés. Le maillage, structuré, est composé de 513×65 nœuds (voir la figure 3.25).

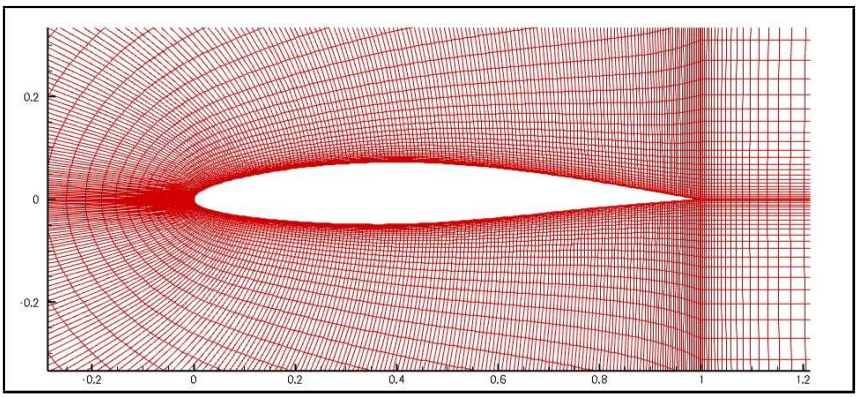


FIG. 3.25 – Maillage du profil NACA64A212

La paramétrisation de la forme est réalisée à l'aide d'un outil développé au département d'aérodynamique appliquée de l'Onera qui définit des profils déformés par combinaison linéaire

de plusieurs profils (représentés sur la figure 3.26) :

$$p(P) = p_0 + P_1(p_1 - p_0) + P_2(p_2 - p_0) + P_3(p_3 - p_0) + P_4(p_4 - p_0). \tag{3.5}$$

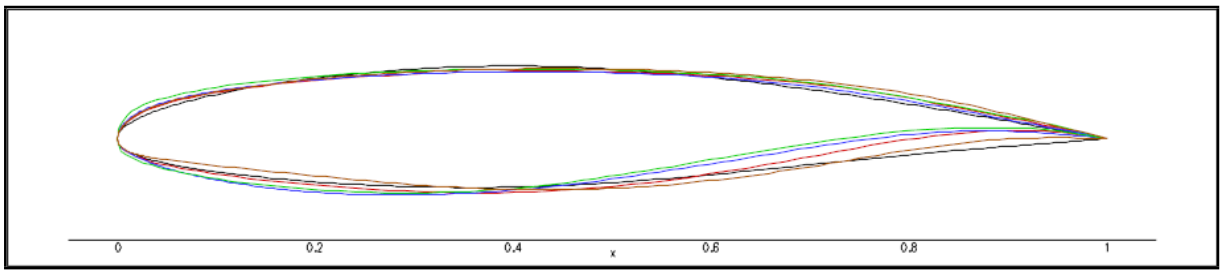


Fig. 3.26 – Bibliothèques de profils (p_i superposés)

Le maillage structuré d'une forme déformée est calculé par déformation du maillage initial. Les sensibilités du maillage surfacique par rapport aux paramètres de forme sont calculées facilement par dérivation de la formule 3.5 (la sensibilité par rapport à P_i vaut p_i). L'écoulement sur le profil initial, caractérisé par la distribution de pression sur la surface du profil et du nombre de Mach dans le maillage volumique, est donné sur les figures 3.27 et 3.28.

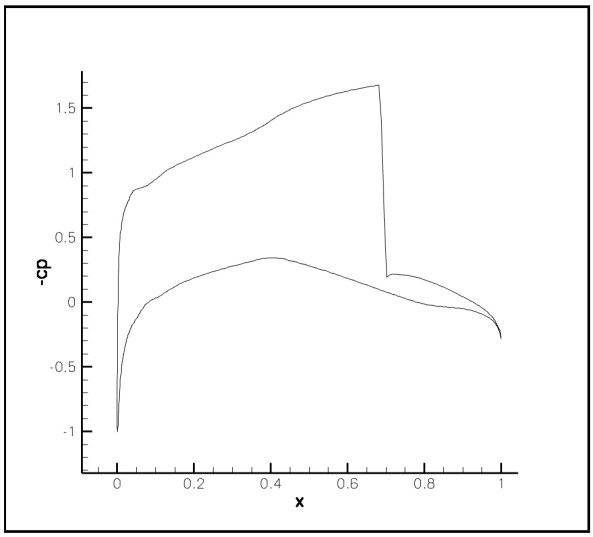


Fig. 3.27 – Distribution de pression statique à la surface

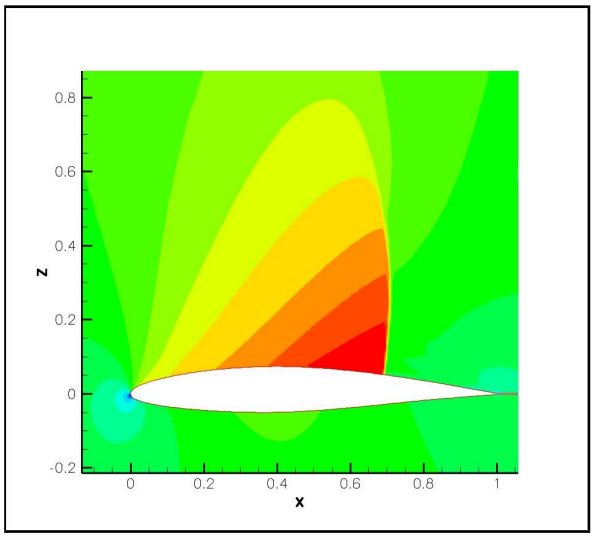


Fig. 3.28 – Nombre de Mach sur le profil initial

Calcul des fonctions aérodynamiques

Les équations d'Euler sont utilisées, le fluide étant considéré comme un fluide parfait. Le système de cinq équations non linéaires régissant le comportement du fluide autour d'une forme donnée est résolu par le code *elsA*, présenté dans le premier chapitre de cette thèse. De même, les coefficients aérodynamiques des écoulements sont calculés par le logiciel *FFD72*.

Les gradients des fonctions aérodynamiques sont donnés par le code *elsA* et leur calcul est basé sur une approche de type adjointe discrète. Étant donné que les équations d'Euler sont

considérées ici, la difficulté relative à la linéarisation d'un modèle de turbulence n'apparaît pas. Par conséquent, aucune approximation n'est nécessaire et les gradients obtenus par la méthode adjointe sont très proches de ceux obtenus par une approche de différences finies. Cette comparaison a été effectuée sur le profil initial. Elle est donnée figure 3.4.1 et 3.4.1 pour, respectivement, les fonctions champ proche (CD_p et CL_p) qui sont les mieux adaptées pour les calculs par différences finies. En effet, pour les fonctions de type champ lointain, il est possible qu'une très faible modification du paramètre de forme modifie le contour d'intégration de la fonction, créant ainsi des discontinuités. Pour chacun des paramètres de forme (de P_1 à P_4, ainsi que l'angle d'attaque), on compare le gradient obtenu par des différences finies avec trois valeurs du pas (données apparaissant sous forme de points) et le gradient de l'approche adjointe discrète (données apparaissant sous forme de ligne). Les résultats obtenus donnent une bonne adéquation, avec une erreur d'approximation inférieure à cinq pour cent.

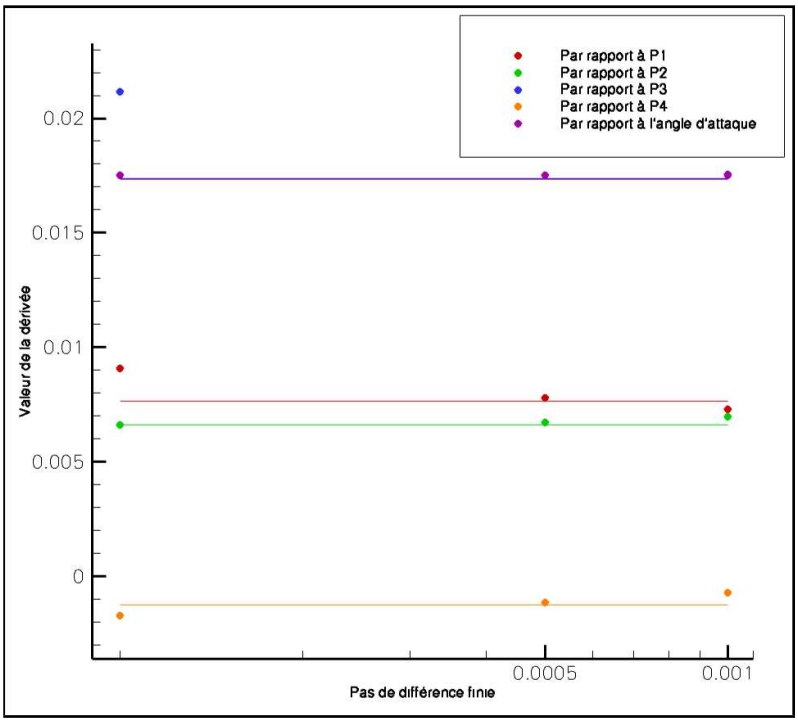


FIG. 3.29 – Gradient de CD_p calculé par la méthode adjointe (ligne) et par différences finies (points)

Problème inverse avec deux paramètres de forme

On propose d'appliquer notre algorithme d'optimisation globale à la résolution d'un problème inverse dans un espace de deux paramètres de forme $\alpha = [\alpha_1, \alpha_2] = [P_1, P_2]$. On définit la fonction objectif J_1 par :

$$J_1(\alpha) = \left(CL_p(\alpha) - CL_p^0\right)^2 + 100\left(CD_{ff}(\alpha) - CD_{ff}^0\right)^2,$$

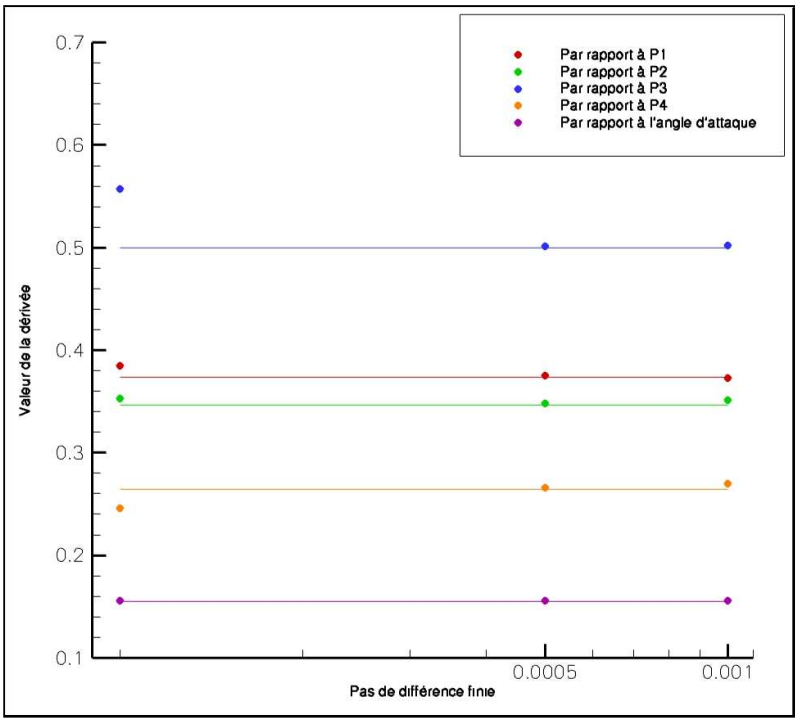


FIG. 3.30 – Gradient de CL_p calculé par la méthode adjointe (ligne) et par différences finies (points)

où CL_p^0 et CD_{ff}^0 représentent respectivement la valeur de la portance champ proche et de la traînée champ lointain sur le profil NACA64A212 qui est choisi comme cible du problème inverse. On connaît donc la solution exacte du problème considéré, de valeur 0, et positionnée en $(\alpha_1, \alpha_2) = (0, 0)$. L'intervalle de variation des paramètres α_1 et α_2 est $[-1, 1]$, de telle sorte que la solution se trouve au centre de l'espace de dessin. Les modèles de substitution avec gradients sont initialisés avec un échantillon de 5 points choisis par la méthode de Hammersley. Dans le cas des modèles sans gradient, 10 points sont utilisés. L'algorithme est stoppé quand la fonction objectif devient inférieure à 5×10^{-6}, ce qui permet d'assurer une localisation de l'optimum à cinq pour cent près. L'évolution de la fonction objectif par rapport au nombre d'évaluations de la fonction exacte est donnée sur la figure 3.31 et les résultats sont synthétisés dans le tableau 3.1.

TAB. 3.1 – Résultats de la résolution du problème inverse

Type de modèle	**Valeur de l'optimum**	**Position de l'optimum**
Krigeage	1.26×10^{-6}	$(2.78 \times 10^{-2}, -2.11 \times 10^{-2})$
Réseau RBF	2.36×10^{-6}	$(-2.52 \times 10^{-2}, 1.47 \times 10^{-2})$
Régression SVR	2.10×10^{-6}	$(-2.64 \times 10^{-3}, -2.10 \times 10^{-3})$
Co-krigeage	3.85×10^{-6}	$(1.23 \times 10^{-2}, -1.41 \times 10^{-2})$
Réseau RBF avec gradients	3.45×10^{-6}	$(-1.64 \times 10^{-2}, 6.82 \times 10^{-3})$
Régression SVR avec gradients	1.18×10^{-6}	$(-2.65 \times 10^{-2}, 1.80 \times 10^{-2})$

Comme on peut le constater sur la figure 3.32, la distribution de pression statique à la surface des profils obtenus par minimisation de la fonction J_1 est bien identique à celle obtenue sur le

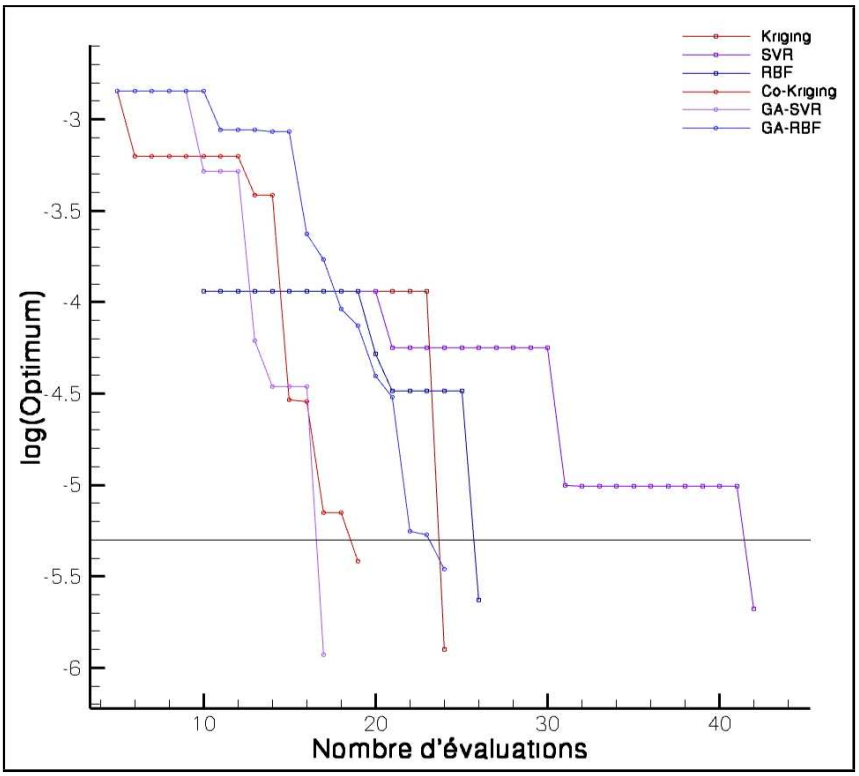


FIG. 3.31 – Évolution de la fonction objectif pour la résolution du problème inverse

profil NACA64A212. De plus, la répartition du nombre de Mach dans le maillage volumique fait apparaître un choc identique à environ trois quart de la corde (figures 3.33 à 3.37).

Notons d'abord que, malgré sa faible dimension, ce problème d'optimisation est complexe. En effet, la fonction objectif J_1 décrit dans l'espace des paramètres de forme une longue vallée plate difficile à optimiser. Comme on peut le voir sur la figure 3.40, l'enrichissement progressif de l'échantillon a bien conduit à ajouter des points dans la vallée pour mieux la modéliser. L'algorithme permet donc d'identifier la solution exacte quel que soit le modèle utilisé. De plus, l'utilisation des gradients dans la construction des modèles permet de réduire le nombre d'évaluations exactes nécessaires pour déterminer le minimum avec une même précision. Néanmoins, on constate que l'apport du gradient semble moins fort que lors de l'application à des fonctions mathématiques. Cela s'explique par le fait que le gradient calculé par la méthode adjointe n'est pas exact. Bien qu'il soit assez précis (comme on l'a montré dans le paragraphe précédent), cela conduit à fournir des informations biaisées pour la construction du modèle. Par conséquent, les modèles avec gradients sont relaxés, c'est à dire qu'une petite quantité est ajoutée sur la diagonale de la matrice du modèle afin de ne pas coller exactement aux dérivées.

Minimisation de traînée sous contrainte de portance avec cinq paramètres de forme

On propose d'appliquer notre algorithme d'optimisation globale à la résolution d'un problème de minimisation de traînée sous contrainte de portance dans l'espace des cinq paramètres de

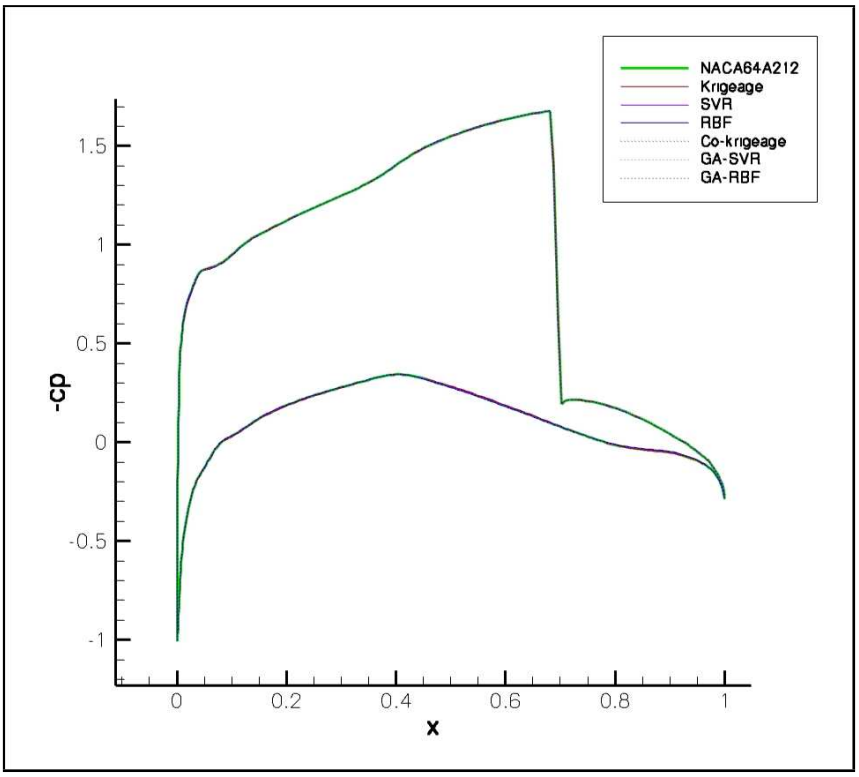


FIG. 3.32 – Distribution de pression statique à la surface (J_1)

forme ($\alpha = [P_1, P_2, P_3, P_4, AoA]$). On définit la fonction objectif J_2 par :

$$J_2(\alpha) = \frac{CD_{ff}(\alpha)}{CD^0_{ff}} + 100 \max\left(0, 1 - \frac{CL_p(\alpha)}{CL^0_p}\right),$$

où $CL^0_p = 0.818$ et $CD^0_{ff} = 0.0357$ représentent respectivement la valeur de la portance champ proche et de la traînée champ lointain sur le profil NACA64A212 qui est choisi comme référence. On ne connaît pas la solution exacte de ce problème. Par conséquent, l'algorithme s'arrêtera lorsque 50 itérations successives n'auront pas entraîné d'amélioration de la fonction objectif. L'intervalle de variation des paramètres P_i est $[-0.2, 0.2]$ afin de rester dans des formes géométriques acceptables. L'intervalle de variation de l'angle d'attaque est $[1.4, 3.6]$. Les modèles de substitution sont initialisés avec un échantillon de 20 points lorsque le gradient est utilisé et de 40 points sinon. L'évolution de la fonction objectif par rapport au nombre d'évaluations de la fonction exacte est donnée sur la figure 3.41 et les résultats sont synthétisés dans le tableau 3.2.

TAB. 3.2 – Résultats de la minimisation de traînée sous contrainte de portance

Type de modèle	**Optimum**	**Position de l'optimum**	CD_{ff}	CL_p
Krigeage	0.629	$(0.200, -0.00151, 0.119, 0.200, 1.450)$	0.0224	0.862
Réseau RBF	0.613	$(0.200, -0.200, 0.200, 0.200, 1.400)$	0.0219	0.820
Régression SVR	0.675	$(0.200, 0.105, 0.200, 0.200, 1.400)$	0.0241	0.944
Co-krigeage	0.629	$(0.123, 0.121, 0.160, 0.200, 1.400)$	0.0225	0.893
Réseau GA-RBF	0.620	$(0.199, -0.200, 0.191, 0.200, 1.430)$	0.0223	0.853
Régression GA-SVR	0.626	$(0.200, -0.120, 0.200, 0.200, 1.400)$	0.0221	0.819

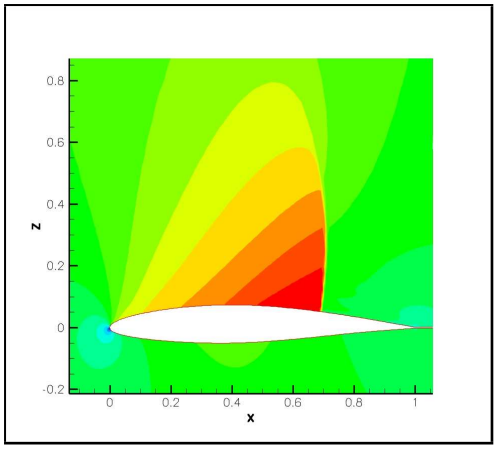


FIG. 3.33 – Nombre de Mach sur le profil optimisé (J_1) avec krigeage

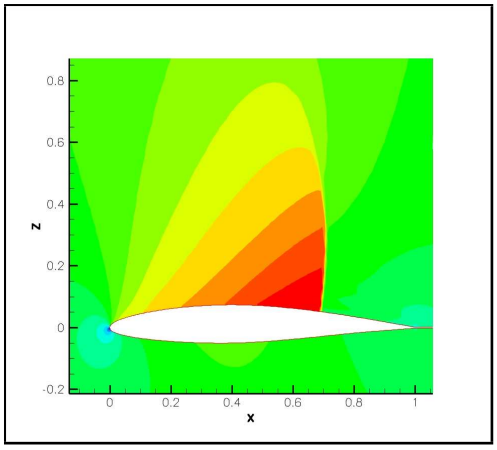


FIG. 3.34 – Nombre de Mach sur le profil optimisé (J_1) avec SVR

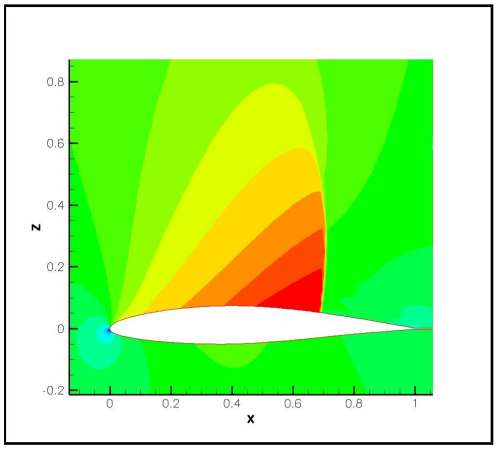


FIG. 3.35 – Nombre de Mach sur le profil optimisé (J_1) avec RBF

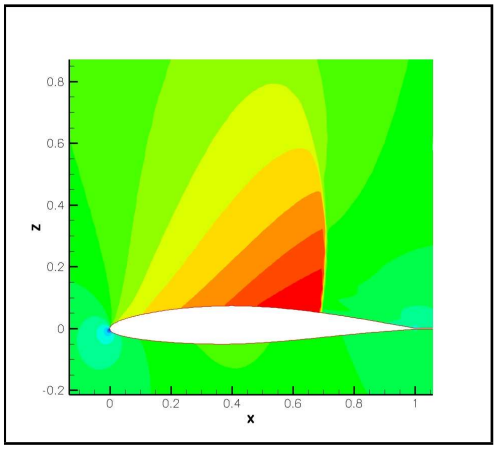


FIG. 3.36 – Nombre de Mach sur le profil optimisé (J_1) avec co-krigeage

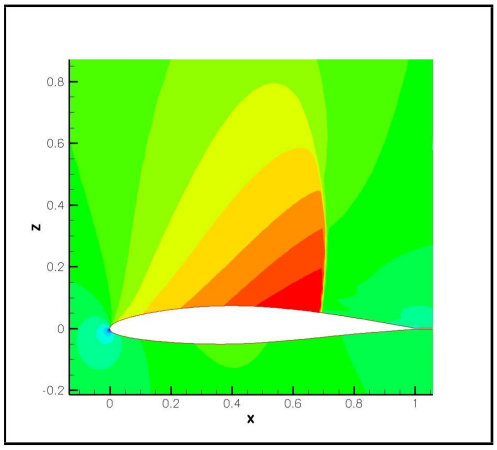


FIG. 3.37 – Nombre de Mach sur le profil optimisé (J_1) avec GA-SVR

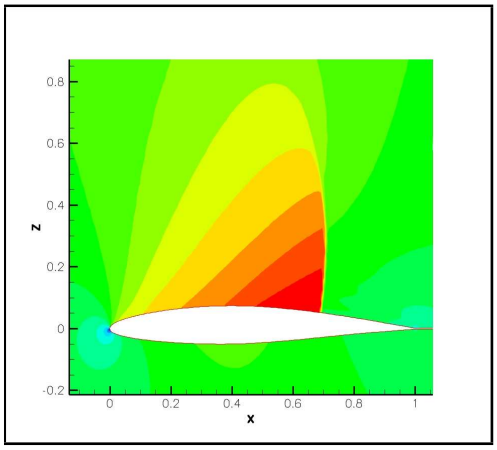


FIG. 3.38 – Nombre de Mach sur le profil optimisé (J_1) avec GA-RBF

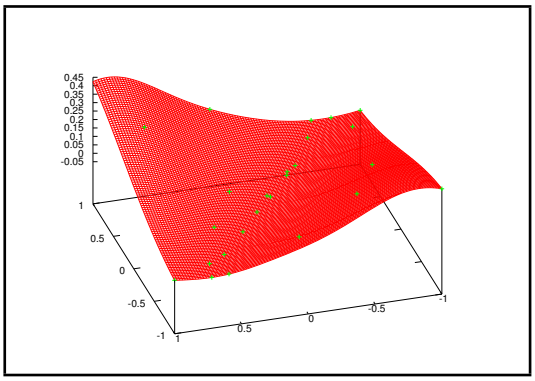


Fig. 3.39 – Fonction objectif J_1

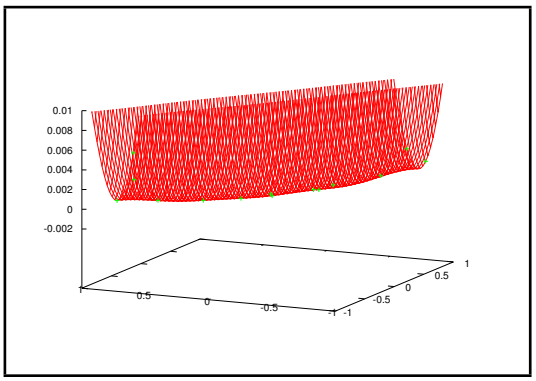


Fig. 3.40 – Zoom sur la vallée de J_1

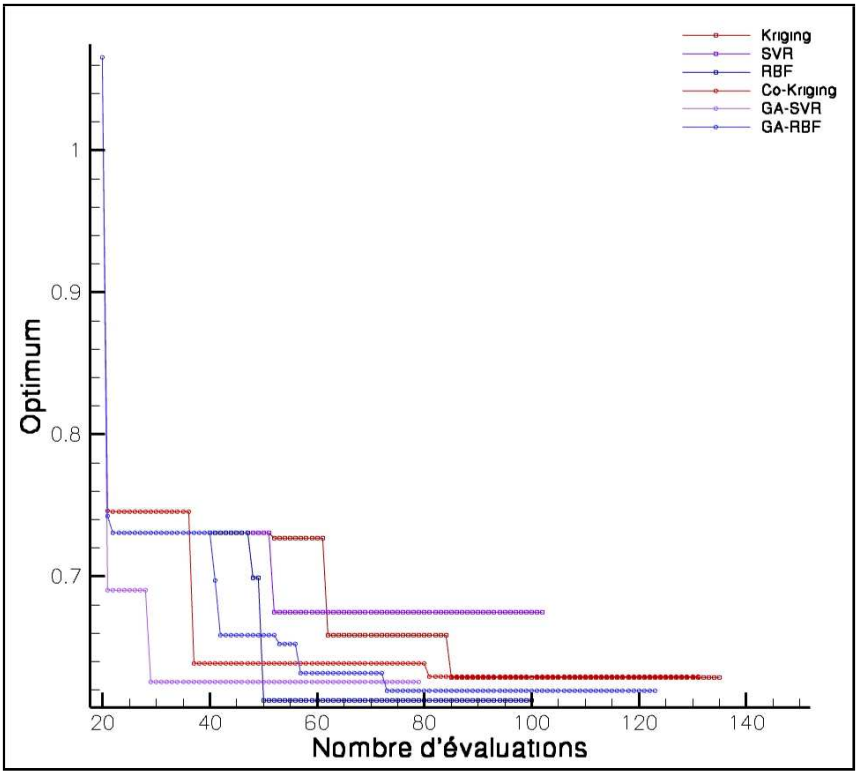


Fig. 3.41 – Évolution de la fonction objectif pour la minimisation de traînée sous contrainte de portance

Comme précédemment, la distribution de pression statique à la surface des profils est donnée sur la figure 3.42. Le choc a été légèrement décalé vers le bord de fuite (voir également la répartition du nombre de Mach sur les figures 3.43 à 3.48).

Une nouvelle fois, l'utilisation des modèles avec gradients permet l'identification plus rapide d'un optimum. Néanmoins, bien que l'optimisation soit effectuée dans un espace de cinq paramètres, cet intérêt est moins fort que lors de nos applications à des fonctions mathématiques, à nouveau en raison des approximations sur la valeur des dérivées. De plus, toutes les méthodes ne se terminent pas sur le même optimum : il serait sans doute nécessaire de poursuivre plus loin l'optimisation ou d'effectuer une analyse plus approfondie des caractéristiques de la fonction pour identifier l'optimum global, et ainsi mieux quantifier les performances de chacune des

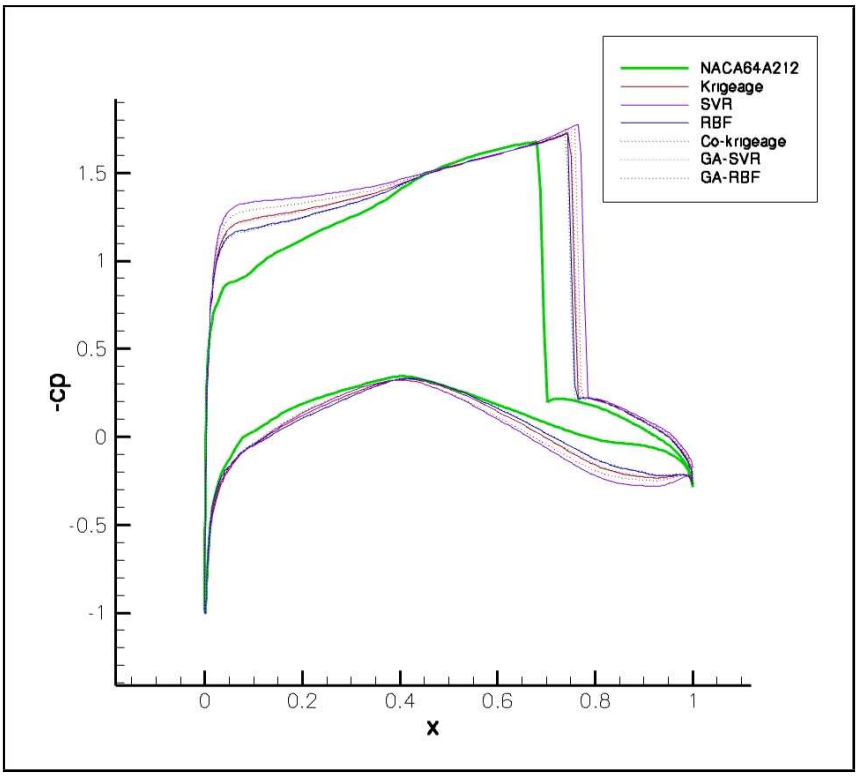


FIG. 3.42 – Distribution de pression statique à la surface (J_2)

méthodes.

3.4.2 Écoulements de fluide visqueux autour d'un profil RAE2822

On cherche ici à vérifier la validité de notre approche lorsque les écoulements considérés sont régis par les équations RaNS et que les dérivées sont calculées par une méthode adjointe discrète pour laquelle le modèle de turbulence est linéarisé. Le profil utilisé comme configuration de référence est un profil RAE2822, très utilisé pour la validation des modèles de turbulence. Les conditions de vol sont celles correspondant à la neuvième expérience de la série d'essais présentée par Cook et al [CMF79]. Les caractéristiques de l'écoulement sont donc un nombre de Mach infini $M_\infty = 0.730$, un angle d'attaque $AoA = 3.19$, et un nombre de Reynolds $Re = 6.5 \times 10^6$ basé sur la masse volumique, la vitesse et la viscosité turbulente à l'infini, et sur la corde du profil. Nous retenons ici la correction classique du projet EUROVAL [HBE$^+$93] qui suggère de corriger l'influence de la soufflerie sur les résultats en modifiant l'angle d'attaque ($AoA = 2.79$ plutôt que $AoA = 3.19$).

Le maillage est composé de deux domaines et de 32832 points. Un domaine forme l'ensemble du volume situé en dessous du profil (en rouge), le second l'ensemble du volume situé au dessus (en bleu). Le maillage est représenté sur les figures 3.49 et 3.50.

Paramétrisation de la forme par la méthode PARSEC

L'approche PARSEC [Sob98] est utilisée pour la paramétrisation du profil. Elle consiste à définir des paramètres de forme qui ont une signification aérodynamique et donc une influence

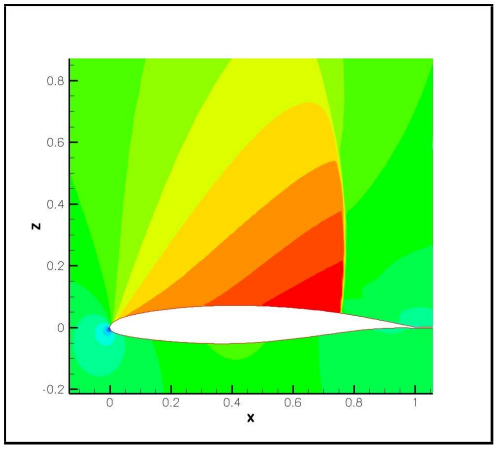


FIG. 3.43 – Nombre de Mach sur le profil optimisé (J_2) avec krigeage

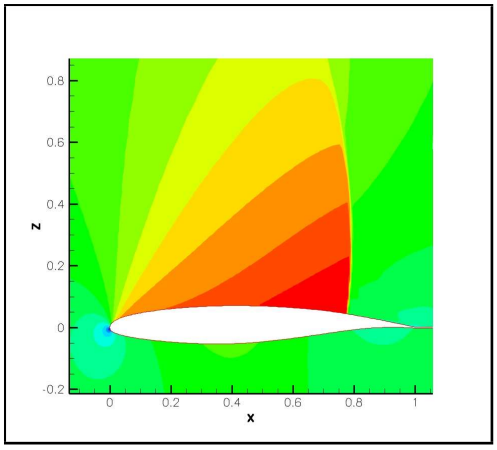


FIG. 3.44 – Nombre de Mach sur le profil optimisé (J_2) avec SVR

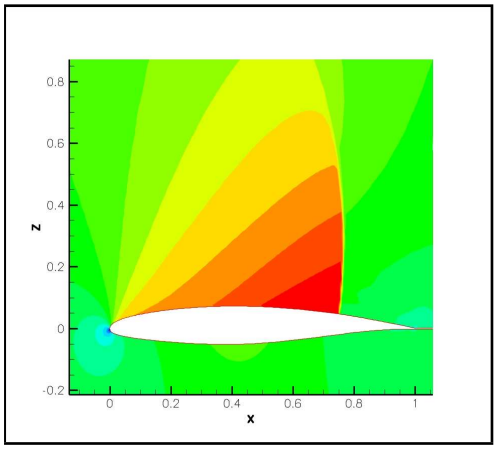


FIG. 3.45 – Nombre de Mach sur le profil optimisé (J_2) avec RBF

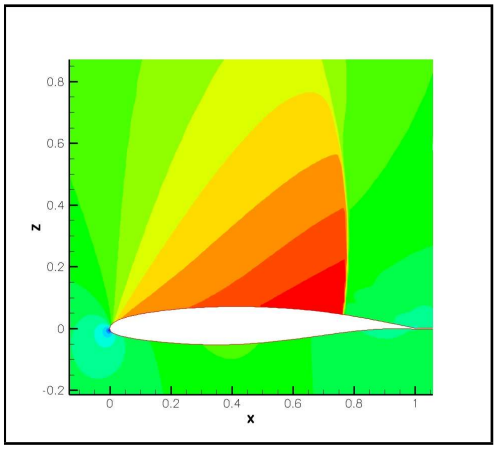


FIG. 3.46 – Nombre de Mach sur le profil optimisé (J_2) avec co-krigeage

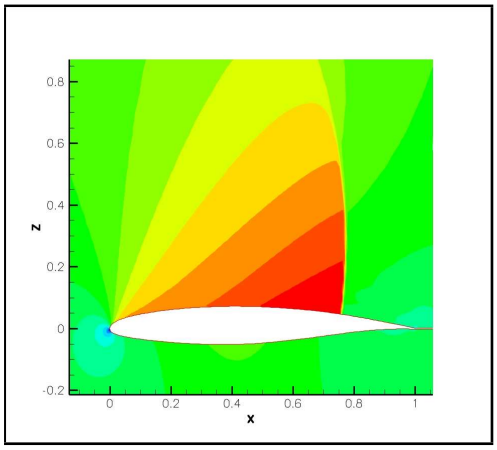


FIG. 3.47 – Nombre de Mach sur le profil optimisé (J_2) avec GA-SVR

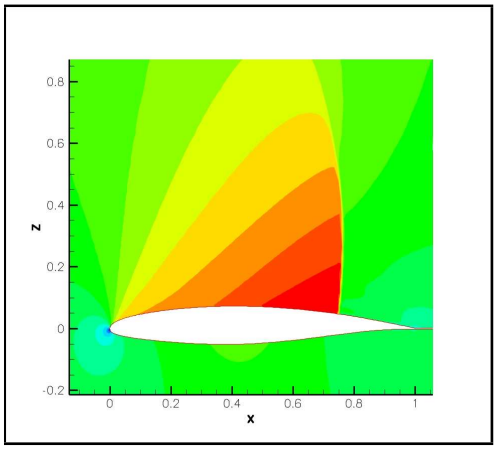


FIG. 3.48 – Nombre de Mach sur le profil optimisé (J_2) avec GA-RBF

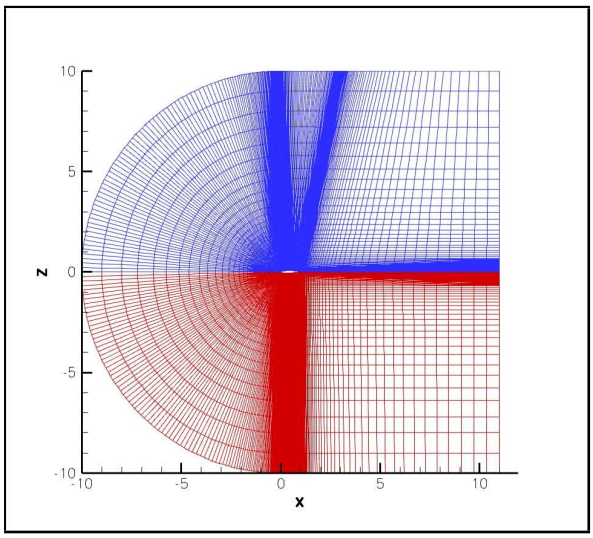


FIG. 3.49 – Maillage du profil RAE2822

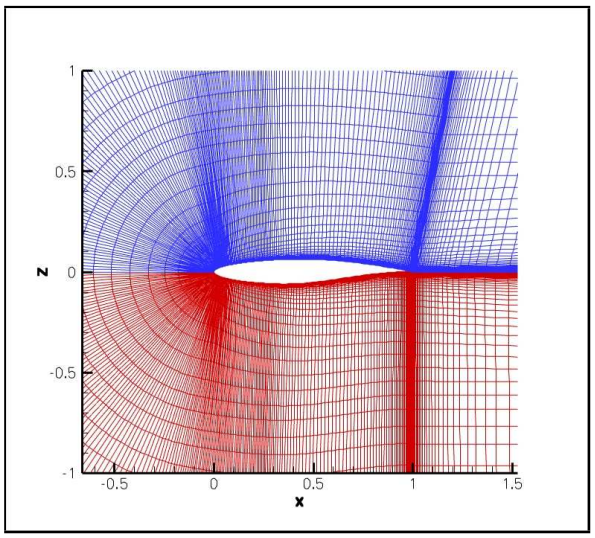


FIG. 3.50 – Zoom sur le profil RAE2822

importante sur les écoulements bidimensionnels. Le profil est défini par deux polynômes :

$$Z^1 = \sum_{n=1}^{6} a_n^1 (X^1)^{n-\frac{1}{2}},$$
$$Z^2 = \sum_{n=1}^{6} a_n^2 (X^2)^{n-\frac{1}{2}},$$

où (X^1, Z^1) et (X^2, Z^2) sont respectivement les coordonnées du profil à l'extrados et à l'intrados. Les 11 paramètres sont présentés dans la figure 3.51.

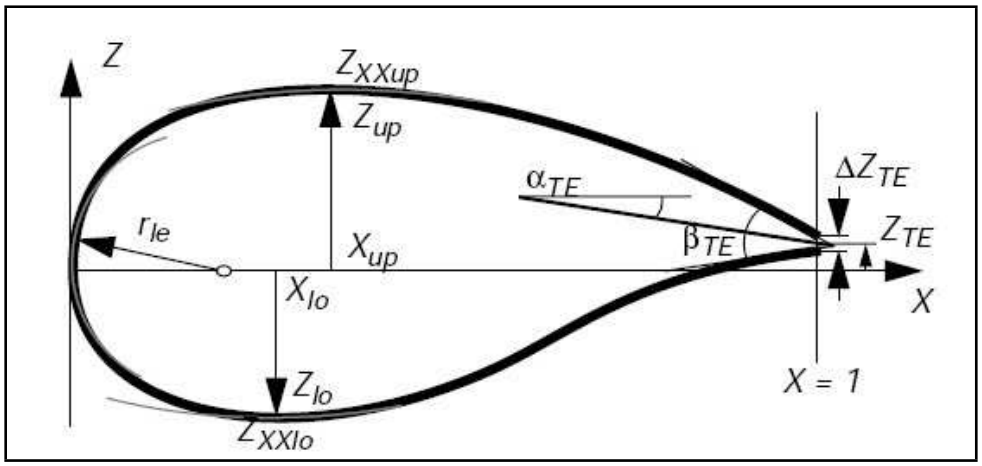


FIG. 3.51 – Paramètres de la méthode PARSEC (extrait de [Sob98])

En introduisant ces paramètres dans les équations des polynômes, on obtient le système

linéaire suivant :

$$\begin{aligned}
\forall k \in [1,2], & \\
a_1^k &= (-1)^{(k+1)}\sqrt{2r_{le}}, \\
\sum_{n=1}^{6} a_n^k (X_M^k)^{n-\frac{1}{2}} &= Z_M^k, \\
\sum_{n=1}^{6} a_n^k &= Z_{TE}^k + \frac{(-1)^{(k+1)}}{2}\Delta Z_{TE}, \\
\sum_{n=1}^{6} \left(n-\frac{1}{2}\right) a_n^k (X_M^k)^{n-\frac{3}{2}} &= 0, \\
\sum_{n=1}^{6} \left(n-\frac{3}{2}\right)\left(n-\frac{1}{2}\right) a_n^k (X_M^k)^{n-\frac{5}{2}} &= Z_{XX}^k, \\
\sum_{n=1}^{6} \left(n-\frac{1}{2}\right) a_n^k &= \tan\left(\alpha_{TE} + (-1)^k \beta_{TE}\right),
\end{aligned}$$

où X_M^1, Z_M^1 et Z_{XX}^1 (respectivement X_M^2, Z_M^2 et Z_{XX}^2) représentent sur la figure 1.3 les quantités X_{up}, Z_{up} et Z_{XXup} (respectivement X_{lo}, Z_{lo} et Z_{XXlo}). Le coefficient a_1^k s'obtient donc directement :

$$\forall k \in [1,2], a_1^k = (-1)^{(k+1)}\sqrt{2r_{le}}.$$

Puis les autres coefficients se calculent par résolution de deux systèmes linéaires de taille 5×5 :

$$\forall k \in [1,2], \begin{bmatrix}
(X_M^k)^{\frac{3}{2}} & (X_M^k)^{\frac{5}{2}} & (X_M^k)^{\frac{7}{2}} & (X_M^k)^{\frac{9}{2}} & (X_M^k)^{\frac{11}{2}} \\
1 & 1 & 1 & 1 & 1 \\
\frac{3}{2}(X_M^k)^{\frac{1}{2}} & \frac{5}{2}(X_M^k)^{\frac{3}{2}} & \frac{7}{2}(X_M^k)^{\frac{5}{2}} & \frac{9}{2}(X_M^k)^{\frac{7}{2}} & \frac{11}{2}(X_M^k)^{\frac{9}{2}} \\
\frac{3}{4}(X_M^k)^{-\frac{1}{2}} & \frac{15}{4}(X_M^k)^{\frac{1}{2}} & \frac{35}{4}(X_M^k)^{\frac{3}{2}} & \frac{63}{4}(X_M^k)^{\frac{5}{2}} & \frac{99}{4}(X_M^k)^{\frac{7}{2}} \\
\frac{3}{2} & \frac{5}{2} & \frac{7}{2} & \frac{9}{2} & \frac{11}{2}
\end{bmatrix}
\begin{bmatrix} a_2^k \\ a_3^k \\ a_4^k \\ a_5^k \\ a_6^k \end{bmatrix}
=
\begin{bmatrix}
Z_M^k - a_1^k (X_M^k)^{\frac{1}{2}} \\
Z_{TE} + \frac{1}{2}\Delta Z_{TE} - a_1^k \\
-\frac{a_1^k}{2(X_M^k)^{\frac{1}{2}}} \\
Z_{XX}^k + \frac{a_1^k}{4(X_M^k)^{\frac{3}{2}}} \\
\tan\left(\alpha_{TE} + (-1)^k \beta_{TE}\right) - \frac{1}{2} a_1^k
\end{bmatrix}.$$

La méthode PARSEC classique permet de contrôler à l'aide d'un nombre réduit de paramètres la forme générale du profil. En pratique, elle souffre d'une limitation importante puisqu'elle ne permet pas de représenter des profils à bord de fuite divergent qui sont utilisés abondamment en raison de leurs bonnes propriétés aérodynamiques. Sobieczky lui-même propose une méthode pour ajouter trois nouveaux paramètres régissant la forme du bord de fuite [Sob98]. Shahrokhi et Jahangirian [SJ07] ont montré que cette modification peut conduire à représenter des profils dans lesquels l'intrados vient traverser l'extrados dans la zone du bord de fuite. Ils proposent alors une autre méthode, dans laquelle un meilleur contrôle du bord de fuite est obtenu

par l'ajout d'un nouveau paramètre. Un champ ΔZ est ajouté à l'intrados et à l'extrados :

$$Z^1 = \sum_{n=1}^{6} a_n^1 (X^1)^{n-\frac{1}{2}} + \Delta Z, \tag{3.6}$$

$$Z^2 = \sum_{n=1}^{6} a_n^2 (X^2)^{n-\frac{1}{2}} + \Delta Z. \tag{3.7}$$

Ce champ est défini à partir d'un paramètre $\Delta\alpha$ qui contrôle la cambrure ajoutée au bord de fuite (voir la figure 3.52) :

$$\Delta Z = \frac{\tan \Delta\alpha}{\mu n} \left[1 + \nu (X^k)^n - (1 - (X^k)^n)^\mu \right], \tag{3.8}$$

où $\nu = 0.8$, $n = 6$ et $\mu = 1.3$.

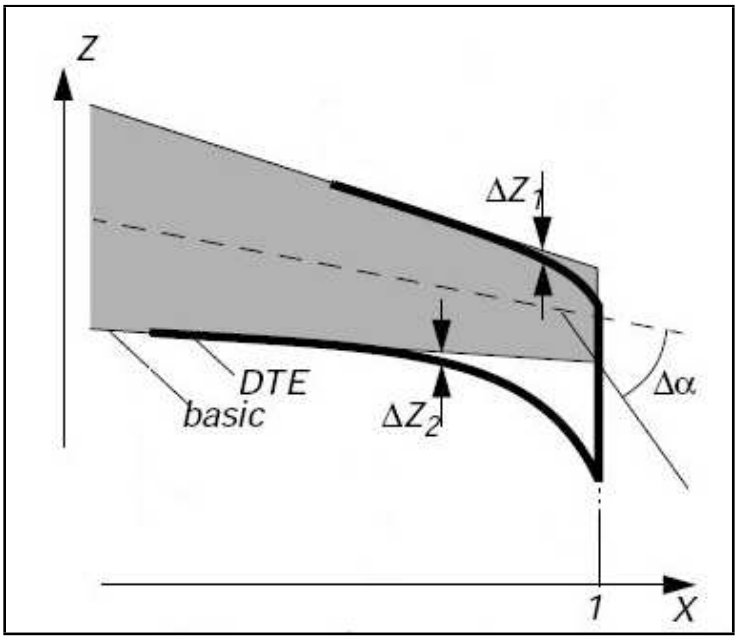


FIG. 3.52 – Ajout d'un paramètre $\Delta\alpha$ au bord de fuite (extrait de [Sob98])

Afin de calculer la sensibilité de la fonction objectif par rapport aux paramètres de forme, il est nécessaire de déterminer la sensibilité du maillage de surface par rapport à ces mêmes paramètres. On cherche donc à calculer $\frac{\partial Z^k}{\partial p_j} \quad \forall k \in [1,2]$ où p_j représente un des paramètres de la méthode PARSEC pour un j donné ($p_1 = r_{le}$, $p_2 = X_{lo}$, ..., $p_{12} = \Delta\alpha$). On peut écrire :

$$\forall k \in [1,2],$$

$$\frac{\partial Z^k}{\partial p_j} = \sum_{n=1}^{6} \frac{\partial a_n^k}{\partial p_j} (X^k)^{n-\frac{1}{2}} \quad \forall j \in [1,11],$$

$$\frac{\partial Z^k}{\partial p_j} = \frac{\partial \Delta Z}{\partial p_{12}}.$$

On peut calculer $\frac{\partial a_1^k}{\partial p_j}$ et $\frac{\partial \Delta Z}{\partial p_{12}}$ directement en dérivant les équations qui définissent a_1^k et ΔZ :

$$\begin{aligned} \forall k \in [1,2], \\ \frac{\partial a_1^k}{\partial p_1} &= \frac{(-1)^{k+1}}{\sqrt{r_{le}}}, \\ \frac{\partial a_1^k}{\partial p_j} &= 0 \quad \forall j \in [2,12], \\ \frac{\partial \Delta Z}{\partial p_{12}} &= \left[1 + \tan \Delta\alpha^2\right] \frac{\left[1 + \nu (X^k)^n - (1 - (X^k)^n)^\mu\right]}{\mu n}. \end{aligned}$$

Pour calculer la sensibilité des autres a_n^k, il est nécessaire de résoudre le système linéarisé des équations de la paramétrisation. Pour cela, on dérive le système d'équations linéaires qui permet d'obtenir les a_n^k par rapport à chacun des paramètres, et on résout un système linéaire par paramètres.

Configuration initiale et validation du solveur

On souhaite donc effectuer une optimisation de forme autour du profil RAE2822. Mais rien ne permet d'assurer que ce profil appartient à l'espace des formes atteignables par PARSEC. Pour cette étude, il est donc proposé de rechercher le profil défini par PARSEC le plus proche du RAE2822. Pour cela, on minimise dans l'espace des paramètres une fonction d'erreur prise comme l'écart en norme L_2 entre le profil cible (RAE2822) et le profil courant. Cette optimisation conduit à un jeu de paramètres, qui définit le profil représenté figure 3.53, qui est visuellement très proche du RAE2822.

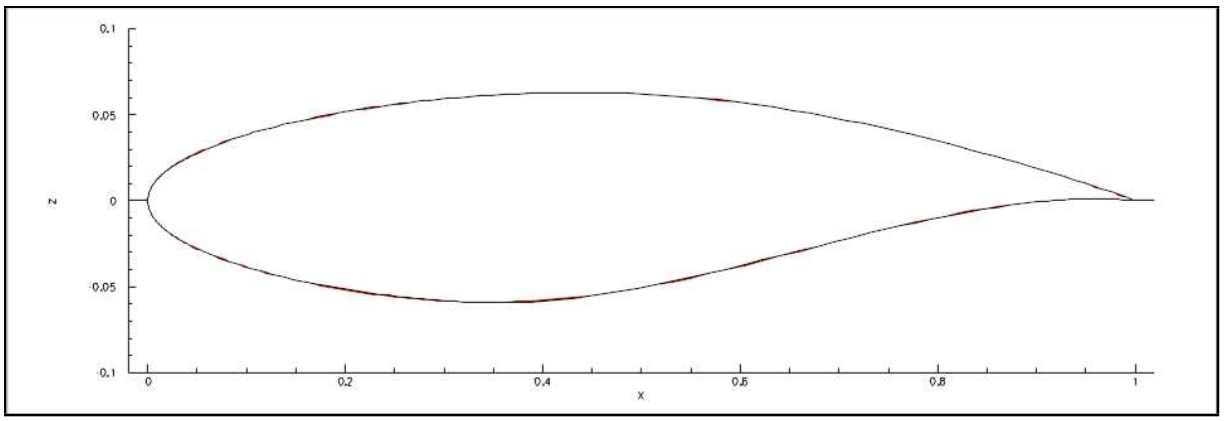


FIG. 3.53 – Profil RAE2822 (noir) et meilleure approximation par la méthode PARSEC (rouge)

Afin de valider le calcul d'analyse sur cette configuration, la distribution de pression à la surface du profil est comparée aux données expérimentales de Cook et al. sur la figure 3.54. On donne également la distribution de pression du profil RAE2822. La correspondance entre les données expérimentales et les résultats numériques correspond à ce qui est observé dans la littérature. On pourra par exemple comparer nos résultats à ceux donnés par Jouhaud et al. [JSL06] : la différence de répartition de pression statique est jugée acceptable pour cet exercice.

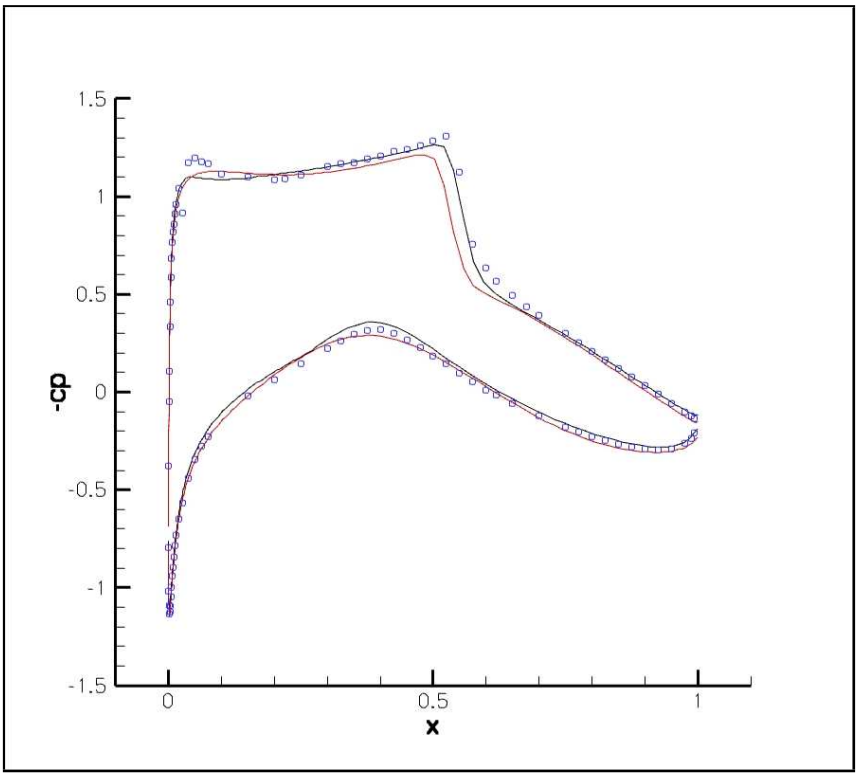


FIG. 3.54 – Comparaison de la répartition de pression sur le profil RAE2822 (noir) et sur la meilleure approximation par la méthode PARSEC (rouge) et des données expérimentales (bleu)

Minimisation du ratio traînée sur portance en fonction de r_{le}, X_{up} et $\Delta\alpha$

L'optimisation effectuée est la minimisation du ratio traînée sur portance en fonction des paramètres $\alpha = [r_{le}, X_{up}, \Delta\alpha]$. On définit donc la fonction objectif suivante :

$$J_3(\alpha) = \frac{CD_{ff}(\alpha)}{CL_p(\alpha)},$$

Les bornes de chacun des paramètres sont choisies de telle sorte que les profils restent acceptables d'un point de vue géométrique. Ainsi, le domaine d'optimisation est $r_{le} \in [0.002, 0.02]$,$X_{up} \in [0.05, 0.08]$ et $\Delta\alpha \in [0, 0.262]$ (l'angle est en radian). Pour chaque calcul d'analyse, une contrainte de portance minimale ($CL \geq CL_0$) est satisfaite par rotation du profil. La taille de l'échantillon initial, toujours calculé par la méthode de Hammersley, est de 10 points pour l'ensemble des modèles. Comme précédemment, afin de prendre en compte l'inexactitude sur les gradients, les modèles avec gradients sont relaxés par l'ajout d'un terme sur la diagonale et l'optimisation s'arrête lorsque le minimum calculé cesse de diminuer pendant un nombre donné d'itérations. L'évolution de l'optimum est donné sur la figure 3.55 ci-dessous et les résultats sont synthétisés dans le tableau 3.3. La comparaison des distributions de pression statique à la surface des profils est donnée sur la figure 3.56.

Comme on peut le voir, l'optimisation a tendance à réduire l'intensité du choc à l'extrados. Il semble que les profils obtenus ne sont pas encore tout à fait optimaux, puisque la distribution

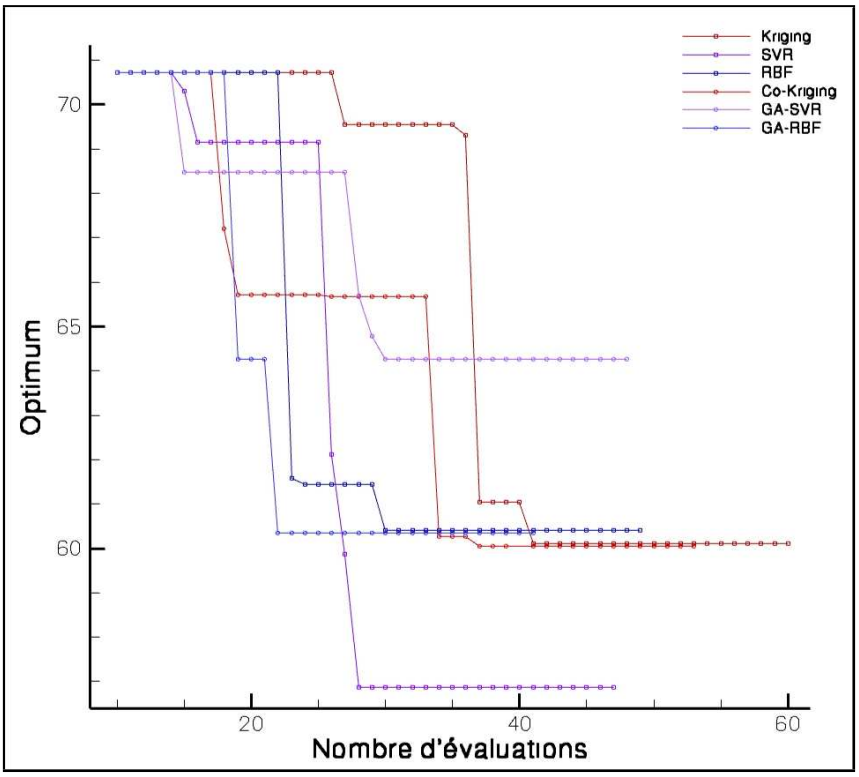


Fig. 3.55 – Évolution de la fonction objectif pour la minimisation du ratio traînée sur portance autour du profil RAE2822

Tab. 3.3 – Résultats de la minimisation du ratio traînée sur portance autour du profil RAE2822

Type de modèle	**Valeur de l'optimum**	**Position de l'optimum**
Krigeage	60.11	$(0.00989, 0.0582, 0.126)$
Réseau RBF	60.41	$(0.0112, 0.0604, 0.134)$
Régression SVR	56.86	$(0.00935, 0.0587, 0.0979)$
Co-krigeage	60.05	$(0.0109, 0.0607, 0.118)$
Réseau RBF avec gradients	60.34	$(0.00314, 0.0519, 0.0166)$
Régression SVR avec gradients	64.27	$(0.00822, 0.0553, 0.0931)$

de pression dans la zone du choc est encore fortement irrégulière : une optimisation plus longue aurait sans doute pu réduire davantage la valeur objectif. Par ailleurs, à nouveau, l'intérêt de l'utilisation des gradients au cours du processus d'optimisation est réel, mais limité : la relaxation due à la mauvaise qualité des gradients explique sans doute cette observation. Ainsi, dans le cadre des travaux de validation de la méthode adjointe du logiciel *elsA*, on observe une erreur d'environ cinq pour cent malgré la linéarisation du modèle de Spalart-Allmaras.

Conclusion

Dans ce chapitre, les modèles de substitution présentés précédemment ont été utilisés pour l'optimisation globale de fonctions mathématiques, puis pour la résolution de problèmes d'optimisation de forme en aérodynamique. Les résultats de cette approche sont très convaincants pour les fonctions mathématiques : ainsi, le nombre d'évaluations de la fonction nécessaire pour

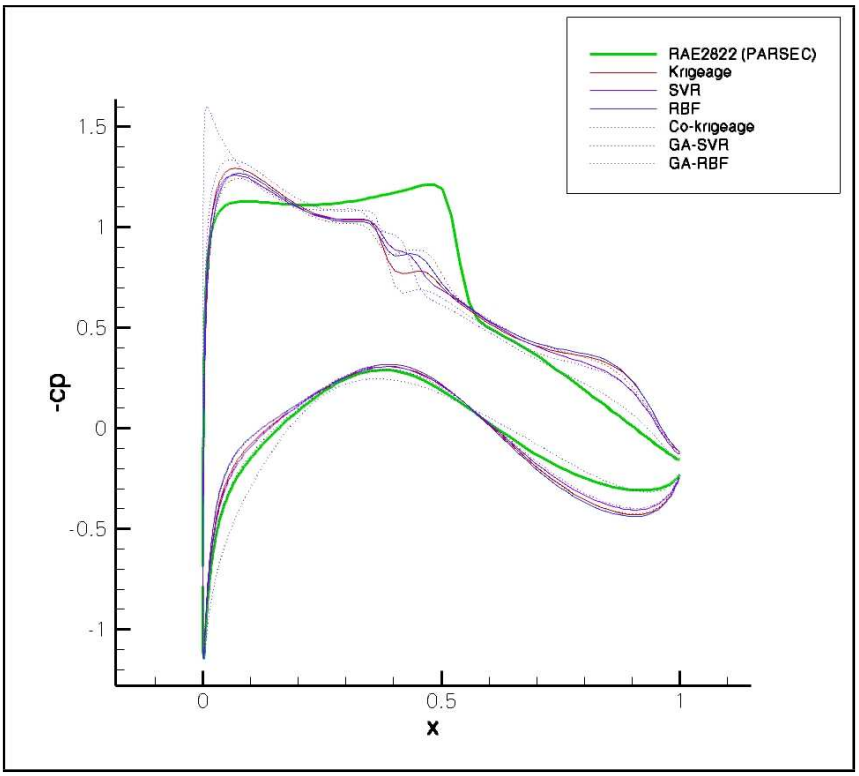


FIG. 3.56 – Distribution de pression statique à la surface (J_3)

déterminer l'optimum avec une même précision est fortement réduit lorsque des metamodèles sont utilisés. De plus, l'utilisation des gradients permet de construire des modèles plus précis, et donc de réduire encore le coût de l'optimisation.

Cette méthode peut résoudre également des problèmes de recherche de formes aérodynamiques optimales. Néanmoins, le bruit qui affecte les gradients des fonctions aérodynamiques calculés par la méthode adjointe discrète réduit l'intérêt de leur utilisation pour des dimensions faibles. L'introduction des méthodes de différentiation automatique, ou l'amélioration de la précision de la méthode adjointe, pourraient permettre d'utiliser des gradients plus précis et donc de diminuer le niveau de relaxation nécessaire. Par ailleurs, à la suite des applications sur des fonctions tests mathématiques réalisées au cours de cette thèse, on a pu remarquer que l'apport des gradients dépendait fortement de la complexité de la fonction objectif : ainsi, plus la fonction objectif a un comportement complexe et plus les informations fournies par les dérivées sont importantes. A l'inverse, pour approcher une fonction affine, l'intérêt des gradients est inexistant. Par conséquent, il conviendrait d'appliquer la méthode d'optimisation proposée ici à de nouveaux cas tests aérodynamiques en effectuant en préalable une étude de la fonction objectif optimisée.

De plus, l'extension de cette approche à des espaces d'optimisation de forte dimension est rendue difficile par les ressources de calcul alors nécessaires pour entraîner un modèle de substitution précis. Dans notre étude, au delà de 200 points d'échantillon et de 5 paramètres de forme, le coût de construction du modèle devient non négligeable par rapport au cout d'une simulation. L'amélioration de la procédure de sélection des paramètres du modèle, ou l'étude de modèles

spécifiquement développés pour des échantillons de grande taille [Ant05], nous semble donc des pistes importantes pour généraliser le travail présenté dans ce chapitre.

Enfin, nous avons décidé dans le cadre de notre travail de prendre en compte les contraintes des problèmes d'optimisation en pénalisant la fonction objectif. Mais plusieurs études ont proposé de généraliser la méthode EGO à des problèmes d'optimisation avec contraintes. On pourra se tourner par exemple vers [Sas02].

Chapitre 4

Stratégie d'optimisation locale sans paramétrisation

Introduction

Le grand avantage de la méthode adjointe est de permettre le calcul de la dérivée pour un coût indépendant de la dimension de l'espace des paramètres de forme. Elle permet donc de réaliser des optimisations avec des paramétrisations extrêmement riches de la surface discrète que l'on souhaite optimiser. Or, plusieurs études ont montré que la richesse de l'espace des formes atteignables dans lequel s'effectue l'optimisation a un impact considérable sur ses performances. Par exemple, Li et Padula [LP04] ont démontré qu'un trop faible nombre de paramètres peut altérer les performances d'un algorithme d'optimisation de forme. Dans un papier récent, Vassberg et al. [VHRJ11] ont comparé les résultats d'une minimisation de traînée pour des paramétrisation par courbes de Bézier allant de 3 à 36 points de contrôle. Enfin, Castonguay et Nadarajah [CN07] ont effectué ce même type de comparaison pour des B-Splines avec 11 à 128 points de contrôle. Sans surprise, les paramétrisations les plus riches conduisent au meilleur résultat.

Dans ce chapitre, on propose donc de ne pas effectuer de paramétrisation de la forme géométrique. Les nœuds décrivant la surface discrète seront utilisés directement comme paramètres de l'optimisation. Le calcul du gradient des fonctions d'intérêt par rapport aux coordonnées de ces points s'effectue facilement par la méthode adjointe. Il sera présenté dans la première section de ce chapitre. Ces gradients seront ensuite utilisés comme direction de descente d'un algorithme d'optimisation locale. On parle alors en anglais d'approche *CAD-free* [MP01]. Dans un papier déjà cité [CN07], cette approche semble donner de très bons résultats. Elle est décrite dans la seconde section de ce chapitre. Une méthode intermédiaire peut également être considérée. Il s'agit d'utiliser le gradient de la fonction d'intérêt en chaque point de la surface pour déterminer le placement optimal des points de contrôle d'un type de paramétrisation donné. Ce sera le sujet de notre troisième section. Enfin, dans un dernier paragraphe, nous présenterons et comparerons les résultats de ces approches à une méthode classique d'optimisation de forme avec une paramétrisation B-Spline.

4.1 Calcul des dérivées par rapport aux points de la surface

4.1.1 Méthode adjointe totale

Dans la méthode adjointe, la dérivée de la fonction objectif par rapport à chaque nœud du maillage volumique (en reprenant les notations du premier chapitre, on notera cette quantité $\nabla_X J(\alpha) = \frac{dJ}{dX}$) est calculée avec un temps de calcul en théorie équivalent à celui nécessaire à la résolution du problème d'analyse :

$$\nabla_X J(\alpha) = \frac{\partial J}{\partial X} + \lambda^T \frac{\partial R}{\partial X}.$$

Dans une approche classique, la dérivée de la fonction objectif par rapport à un paramètre de forme α_i est alors obtenue en multipliant $\nabla_X J(\alpha)$ à la sensibilité du maillage volumique par rapport à α_i :

$$\frac{dJ}{d\alpha_i}(\alpha) = \nabla_X J(\alpha) \frac{dX}{d\alpha_i}(\alpha).$$

Les sensibilités du maillage par rapport aux paramètres de forme ont longtemps été calculées par différences finies. Elles sont alors déduites d'une déformation du maillage volumique d'une petite quantité ϵ dans la direction du paramètre de forme α_i :

$$\frac{dX}{d\alpha_i}(\alpha) = \frac{X(\alpha_i + \epsilon) - X(\alpha_i)}{\epsilon}.$$

Comme le coût de déformation du maillage volumique est généralement faible par rapport à celui de la résolution directe et adjointe des équations du fluide considéré, ce calcul est globalement beaucoup plus bref que les appels au solveur CFD.

Cependant, avec l'augmentation de la complexité géométrique des problèmes d'optimisation, le nombre de paramètres de forme n_f va croissant. Or si l'on considère que le coût de la résolution des équations de la mécanique des fluides est n fois plus important que celui de la déformation du maillage volumique, alors cette dernière étape va dominer le coût total du calcul de gradient à partir d'un nombre de paramètres de forme $n_f = n$. De plus, les approches sous forme de différences finies possèdent l'inconvénient de nécessiter une étude paramétrique en amont afin de choisir proprement l'incrément utilisé. Une solution à cette difficulté a été proposée par Nielsen et Park [NP05]. Il s'agit de construire l'adjoint de la méthode de déformation en plus de celui des équations de la mécanique des fluides. Ainsi, les sensibilités du maillage volumique par rapport au maillage de peau de l'objet solide, noté S, sont éliminées de l'expression du gradient, dont l'évaluation devient cette fois totalement indépendante du nombre de paramètres de forme n_f. Le calcul de n_f déformations du maillage volumique est remplacé par celui d'un vecteur adjoint [YM07] [JA07] [HZ10].

Si l'on s'inscrit toujours dans une approche avec paramétrisation, il reste alors à évaluer les sensibilités du maillage surfacique par rapport aux paramètres de forme. On peut le faire directement si $S(\alpha)$ est dérivable ou utiliser à nouveau une méthode de différence finies et

l'appliquer cette fois à la déformation du maillage surfacique :

$$\frac{dS}{d\alpha_i}(\alpha) = \frac{S(\alpha_i + \epsilon) - S(\alpha_i)}{\epsilon}.$$

La dépendance par rapport au nombre de paramètres de forme apparait donc à nouveau. Mais il est clair que la déformation du maillage surfacique est bien moins coûteuse que celle du maillage volumique. Cette méthode est donc préférable lorsque la géométrie à optimiser est complexe : elle est par exemple appliquée pour l'optimisation d'une aile ONERA M6 dans une contribution récente de Widhalm et al. [WBIL10].

Mais dans ce travail, on souhaite utiliser directement les points de la surface comme paramètres de forme. Il est donc nécessaire de dériver la méthode de déformation volumique. L'ajout d'une équation adjointe n'est pas nécessaire ici car la méthode de déformation volumique considérée est une méthode algébrique [MCV04], qui se formule et se dérive de manière explicite. Elle est décrite ci-dessous.

4.1.2 Méthode de déformation volumique algébrique [MCV04]

Notons δS le déplacement des nœuds du maillage surfacique ($\delta S = S - S^0$ si S^0 désigne la surface initiale). Alors, le déplacement δX d'un point p_i du maillage volumique est donné explicitement par la formule suivante.

$$\delta X(p_i) = \nu(p_i) \frac{\sum_{p_j \in S} \delta S(p_j) \left(\left\langle \frac{\overrightarrow{p_j\ p_i}}{||p_j\ p_i||}, \overrightarrow{n_j} \right\rangle + 1. \right) da_j}{\sum_{p_j \in S} \left(\left\langle \frac{\overrightarrow{p_j\ p_i}}{||p_j\ p_i||}, \overrightarrow{n_j} \right\rangle + 1. \right) da_j}, \tag{4.1}$$

où da_j et n_j désignent l'aire et le vecteur normal sortant associés au nœud p_j de la surface solide. La fonction ν est définie de telle sorte que la déformation est progressivement réduite lorsque la distance avec l'objet solide augmente. Elle est choisie pour valoir 1 pour les points à une distance de l'objet inférieure à quatre fois la corde et est nulle pour les points plus loin que 8 fois la corde. Le maillage déformé est donc calculé par modification du maillage initial :

$$X = X^0 + \delta X.$$

L'intérêt d'utiliser une méthode de déformation algébrique, outre ses bonnes propriétés, est le fait que la formule de calcul est explicite et peut donc être dérivée. Si, de plus, on considère que les déformations sur le maillage sont faibles durant le processus d'optimisation, on peut calculer toutes les distances sur le maillage initial. La méthode de déformation devient alors linéaire et son jacobien peut être calculé facilement :

$$\frac{d\ \delta X(p_i)}{d\ \delta S(p_l)} = \nu(p_i) \frac{\left(\left\langle \frac{\overrightarrow{p_l\ p_i}}{||p_l\ p_i||}, \overrightarrow{n_l} \right\rangle + 1. \right) da_l}{\sum_{p_j \in S} \left(\left\langle \frac{\overrightarrow{p_j\ p_i}}{||p_j\ p_i||}, \overrightarrow{n_j} \right\rangle + 1. \right) da_j}, \tag{4.2}$$

Comme les déformations sont ajoutées au maillage initial pour former le maillage déformé, la dérivée par rapport à la déformation est aussi la dérivée par rapport au maillage déformé,

c'est à dire :

$$\frac{d\ \delta X(p_i)}{d\ \delta S(p_l)} = \frac{dX(p_i)}{d\ \delta S(p_l)}.$$

On peut alors déduire la sensibilité de la fonction objectif par rapport aux nœuds de la surface solide :

$$\frac{dJ}{dS} = \frac{dJ}{dX}\frac{dX}{d\ \delta S}$$

4.2 Chaîne d'optimisation sans paramètre de forme

A partir des sensibilités calculées à la surface, on souhaite donc mettre en place une méthode d'optimisation locale dans laquelle les paramètres d'optimisation définissent la position de chaque point du maillage surfacique. Pour préserver la qualité du maillage durant l'optimisation, celui-ci n'est déformé que dans la direction normale en chacun de ses points [LBT10]. Ainsi, la sensibilité d'un point de la surface est projetée sur le vecteur normal et utilisée pour calculer la direction de descente. La méthode numérique retenue pour l'optimisation est la méthode CONMIN, développée par Vanderplaats [Van73].

4.2.1 Méthode numérique d'optimisation

La méthode CONMIN a été largement utilisée pour l'optimisation de forme en aérodynamique. Elle est aujourd'hui considérée comme une méthode robuste de résolution de problèmes d'optimisations linéaires ou non linéaires, soumis ou non à des contraintes. Elle est basée sur la combinaison de deux méthodes classiques d'optimisation. Dans le cas d'une optimisation sans contrainte, ou si les contraintes du problème d'optimisation ne sont pas actives, la direction de descente est calculée à partir de la méthode du gradient conjuguée de Fletcher et Reeves [FR64]. Si, au contraire, au moins une contrainte du problème d'optimisation est active, alors la méthode des directions admissibles de Zoutendijk [Zou60] est utilisée. Une fois que la direction de descente est déterminée, le pas de descente est calculé par une minimisation unidimensionnelle effectuée par interpolation polynomiale.

4.2.2 Lissage des sensibilités

Il est évident que le fait de considérer chaque point du maillage de surface comme un paramètre de forme du problème d'optimisation permet de rechercher la forme optimale dans un espace de formes atteignables extrêmement large. Cependant, il est connu qu'une résolution dense de l'espace de paramétrisation a tendance à générer du bruit à haute fréquence durant le processus d'optimisation [RJA+99]. La convergence est alors fortement ralentie et les formes finales peuvent être non lisses. Or on sait que les formes efficaces d'un point de vue aérodynamique sur un large domaine de conditions à l'infini sont des formes lisses [JV01] pour lesquelles la distribution de pression statique à la surface du profil est régulière. Il existe notamment un lien direct entre la régularité de la distribution de pression statique à la surface et la courbure de celle-ci [Bar73] (c'est sur ce principe que sont construites les méthodes inverses).

Par conséquent, une pratique courante est de lisser les formes obtenues durant le processus d'optimisation. Par exemple, dans le processus CDISC [SC90] qui implémente un outil de

conception par méthodes inverses, les formes sont lissées par interpolation polynomiale afin de réduire les modifications brusques de courbure générées par les déformations successives. Cependant, ce type d'approche peut conduire à supprimer le bénéfice issu de l'optimisation. Une seconde approche consiste à lisser la direction de descente pendant le processus d'optimisation, c'est à dire à lisser les modifications de forme et non les formes elles-mêmes. Dans ce cadre, l'opérateur classique de lissage est l'opérateur IRS (*Implicit Residual Smoothing*) introduit par Jameson [Jam94] dans le but d'accélérer la convergence d'une méthode d'optimisation (on parle aussi de gradient Sobolev). Il s'agit d'appliquer un schéma de résolution implicite à une équation de la chaleur, soit en notant respectivement $\mathcal{G}$ le gradient non lissé et $\overline{\mathcal{G}}$ le gradient lissé :

$$\overline{\mathcal{G}} - \frac{\partial}{\partial \xi}\epsilon\frac{\partial}{\partial \xi}\overline{\mathcal{G}} = \mathcal{G}.$$

Alors :

$$\overline{\mathcal{G}} = A\mathcal{G},$$

où A est une matrice tridiagonale dont les termes sont $[-\epsilon, 1+2\epsilon, -\epsilon]$.

Le choix du paramètre ϵ est crucial pour la qualité du lissage. Dans les travaux initiaux de Jameson [Jam94], ϵ est choisi à une valeur constante (dans l'espace) déterminée après plusieurs essais. Dans la contribution de Schmidt et al. [SIGS08], il est proposé de lier la valeur de ϵ à la composante tangentielle du gradient. Stück et Rung [SR11] ont utilisé une méthode de lissage par convolution avec un noyau gaussien, dont ils démontrent qu'elle est équivalente à la méthode IRS. Une optimisation *CAD-free* dans le cadre de l'optimisation hydrodynamique a été effectuée dans ce même article. Avec ce type de lissage, on peut montrer qu'une direction de descente reste une direction de descente après lissage [JV01]. Cette méthode est donc presque exclusivement utilisée dans la littérature.

Néanmoins, appliquée dans le cadre de ce travail, elle a posé de nombreuses difficultés. Ainsi, le paramètre de lissage n'est pas déterminé en fonction de la géométrie. Dans le cas d'un profil, cela revient à lisser de la même manière le bord d'attaque, le bord de fuite et les zones au centre du profil. Si le lissage est trop fort, les variations du champ de déformation s'annulent. Dans le cas contraire, la forme obtenue a de fortes variations dans la zone du bord d'attaque, ce qui conduit à des profils inacceptables d'un point de vue aérodynamique. Une autre stratégie est donc utilisée ici : au lieu de lier l'intensité du lissage à la valeur du champ, il est proposé de déterminer la quantité de lissage nécessaire localement en fonction de la régularité géométrique du profil ainsi obtenu. Le lissage IRS est donc remplacé par une méthode de lissage par splines proposée par Dierckx [Die80]. Celle-ci recherche de manière itérative la courbe spline qui interpole une courbe donnée avec une tolérance ϵ_s tout en réduisant les variations de sa dérivée seconde. Le champ lissé conduit alors à des profils déformés sur lesquels les variations de courbure sont raisonnables. Cette méthode a d'ailleurs été utilisée pour le lissage de la courbure d'un profil par Li et Krist [LK05]. Elle sera présentée dans la section suivante.

Lissage par splines minimisant les variations de la dérivée seconde

Le principe du lissage par spline provient des travaux de Whittaker dans les années 20 [Whi23]. Il s'agit de fixer une mesure du caractère lisse d'une courbe et de rechercher la courbe qui offre un compromis entre l'approximation de la courbe initiale et la minimisation de cette mesure. Dans la méthode de Dierckx, cette mesure est calculée à partir des discontinuités de la dérivée de la courbe. Dit autrement, une courbe est lisse si les discontinuités de sa dérivée sont minimales. Dans ce travail, nous avons utilisé le code source FITPACK développé par Dierckx et disponible librement [Die93]. En entrée, on donne une tolérance ϵ_s et une courbe initiale. L'algorithme choisit alors automatiquement des points de contrôle, et détermine la courbe spline qui minimise les discontinuités de la dérivée tout en respectant la précision d'approximation spécifiée par ϵ_s.

Par la suite, cette méthode sera utilisée au sein d'un processus itératif jusqu'à satisfaction d'un critère d'arrêt, basé sur l'évaluation de la courbure d'un profil déformé. A chaque nouvelle itération, la valeur de ϵ_s est incrémentée de sorte à augmenter l'intensité du lissage. Ce processus itératif est décrit ci-dessous.

Calcul de la courbure d'un profil

Le calcul de la courbure d'une surface nécessite de déterminer la valeur des dérivées première et seconde du profil. Pour une courbe définie par $y = f(x)$, la courbure se calcule par :

$$\kappa_S(x) = \frac{f''(x)}{(1 + f'(x)^2)^{\frac{3}{2}}}. \tag{4.3}$$

Or la surface n'est décrite ici que de manière discrète. Un calcul des dérivées par l'approximation d'ordre 1 conduit à des erreurs numériques importantes, notamment dans les zones où la discrétisation est moins précise. Par conséquent, on propose d'interpoler le champ discret par une courbe spline et de dériver explicitement cette courbe. Afin d'éviter des oscillations dues à l'interpolation, les splines d'Akima [Aki70] sont utilisées dans ce travail.

Algorithme de lissage

En préparation de l'algorithme itératif, on stocke l'opposé de la valeur normale du gradient à la surface :

$$d^1 = -\left\langle \frac{dJ}{dS}, n_S \right\rangle.$$

De plus, le paramètre de lissage ϵ_s qui fixe le compromis entre l'écart aux points et l'intensité du lissage est fixé à 0. Dans une première itération, le champ d^1 sera donc interpolé exactement.

A une étape l, les différentes étapes de notre algorithme de lissage sont alors les suivantes :

1. Le champ d^l est interpolé avec le paramètre de lissage pris à sa valeur initiale. L'interpolant est stocké dans le champ s^l.
2. Un pas de descente estimé est calculé à partir de la valeur actuelle de la fonction objectif et de la direction de descente lissée s^l. Il est choisi comme le pas qui réduirait la fonction

objectif d'un pourcentage donné β_{obj} si sa variation était linéaire :

$$t^l = \beta_{obj} \frac{J}{\langle d^l, d^l \rangle}.$$

3. Un profil estimé S^l est alors calculé à partir de la direction de descente actuelle d^l et du pas estimé t^l

$$S^l = S_0 + t^l d^l n_S.$$

4. La courbure du profil estimé κ_{S^l} est calculée par la méthode présentée ci-dessus.
   - Le processus de lissage s'arrête si la variation totale de la courbure est inférieure à une constante positive γ :
   $$\int_{x \in S^l} |\kappa_{S^l}(x)| \leq \gamma.$$
   - Sinon, le paramètre d'interpolation ϵ_s est augmenté, l'indice l est incrémenté et l'algorithme reprend à l'étape 1.

4.2.3 Contraintes géométriques

L'utilisation des coordonnées des points composant le maillage surfacique comme paramètres d'optimisation a l'avantage de ne pas sélectionner par avance le type de formes que l'algorithme génère. Néanmoins, cette liberté de déformation peut conduire à des formes non acceptables d'un point de vue géométrique.

On sait par exemple que pour augmenter la portance d'un profil, il est nécessaire de cambrer la forme du profil au niveau du bord de fuite. Dans les applications présentées dans la section suivante, cette recherche de cambrure conduisait à des profils pour lesquels l'intrados venait traverser l'extrados. Il est donc nécessaire d'ajouter à notre problème d'optimisation des contraintes qui assurent la validité géométrique de la forme au cours du processus. En chaque point proche du bord de fuite, l'épaisseur du profil est donc calculée et contrainte de rester supérieur à une valeur minimale. Cette dernière est choisie comme pourcentage de l'épaisseur initiale.

Considérons un point i de l'extrados, proche du bord de fuite, de coordonnées spatiales (x_i, y_i). Alors l'épaisseur t_i se formule de la manière suivante :

$$t_i = y_i - py_i,$$

où py_i est l'ordonnée de l'intrados à l'abscisse x_i. On peut la calculer en utilisant la formule suivante :

$$py_i = y_j + \frac{x_i - x_j}{x_{j+1} - x_j}(y_{j+1} - y_j),$$

où le point de coordonnées (x_j, y_j) (respectivement (x_{j+1}, y_{j+1})) est le point de l'intrados dont l'abscisse est inférieure (respectivement supérieure) et la plus proche de x_i.

De la même manière, l'épaisseur en un point j de l'intrados se calcule par :

$$g_j = py_j - y_j,$$

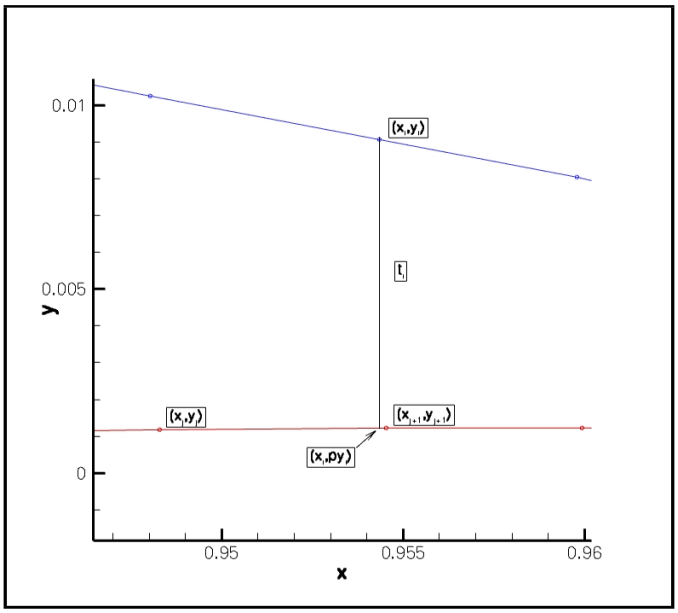


FIG. 4.1 – Contraintes sur l'épaisseur du profil en x_i

où py_j est l'ordonnée de l'extrados à l'abscisse x_j.

Les contraintes géométriques sont prises en compte comme contraintes du problèmes d'optimisation dans le logiciel CONMIN. Leurs gradients sont calculés de manière analytique. Par exemple, en considérant un point i de l'extrados, on peut calculer les composantes du gradient par les formules suivantes :

$$\begin{aligned}
\frac{\delta t_i}{\delta x_i} &= -\frac{y_{j+1}-y_j}{x_{j+1}-x_j}, \\
\frac{\delta t_i}{\delta y_i} &= 1, \\
\frac{\delta t_i}{\delta x_j} &= -\frac{(x_i-x_{j+1})(y_j-y_{j+1})}{(x_{j+1}-x_j)^2}, \\
\frac{\delta t_i}{\delta y_j} &= \frac{x_i-x_j}{x_{j+1}-x_j}-1, \\
\frac{\delta t_i}{\delta x_{j+1}} &= \frac{(x_i-x_j)(y_j-y_{j+1})}{(x_{j+1}-x_j)^2}, \\
\frac{\delta t_i}{\delta y_{j+1}} &= -\frac{x_i-x_j}{x_{j+1}-x_j}.
\end{aligned}$$

De même, si l'on considère un point j de l'extrados, les formules deviennent :

$$\begin{aligned}
\frac{\delta t_j}{\delta x_j} &= \frac{y_{i+1}-y_i}{x_{i+1}-x_i}, \\
\frac{\delta t_j}{\delta y_j} &= -1, \\
\frac{\delta t_j}{\delta x_i} &= \frac{(x_j-x_{i+1})(y_{i+1}-y_i)}{(x_{i+1}-x_i)^2}, \\
\frac{\delta t_j}{\delta y_i} &= 1-\frac{x_j-x_i}{x_{i+1}-x_i}, \\
\frac{\delta t_j}{\delta x_{i+1}} &= -\frac{(x_j-x_{i+1})(y_{i+1}-y_i)}{(x_{i+1}-x_i)^2}, \\
\frac{\delta t_j}{\delta y_{i+1}} &= \frac{x_j-x_i}{x_{i+1}-x_i}.
\end{aligned}$$

Une fois l'épaisseur calculée, on peut formuler les contraintes pour préserver une épaisseur du profil raisonnable. Dans la suite, on proposera de formuler en chaque point i dont l'abscisse est comprise dans les cinq derniers pour cent de la corde une contrainte g_i :

$$g_i = \left(0.8t_i^0 - t_i\right),$$

où t_i^0 est l'épaisseur au point i sur le profil initial. Autrement dit, une réduction de l'épaisseur du profil est tolérée tant qu'elle reste supérieure à 80% de celle du profil initial.

4.3 Chaîne d'optimisation avec placement a priori des paramètres de forme

La paramétrisation est basée sur une perturbation de l'intrados et de l'extrados à partir de B-Splines. Deux courbes, pilotées par 20 points de contrôle, décrivent respectivement les modifications de l'intrados et de l'extrados. Les premier et derniers points de contrôle correspondent respectivement au bord d'attaque et au bord de fuite ($x/c = 0$ et $x/c = 1$, respectivement). Les 18 points de contrôle restant sont répartis sur la surface en respectant une distribution en tangente hyperbolique, comme illustré sur la figure 4.2. Les points de contrôle de l'intrados et de l'extrados situés aux bords sont identiques. Enfin, un point supplémentaire est positionné au bord d'attaque sur chaque côté du profil. Le nombre total de degré de liberté est $2 \times 18 + 2 = 38$.

Un paramètre de forme est utilisé pour contrôler indépendamment le déplacement vertical de chaque point de contrôle, à l'exception des points de contrôle supplémentaires positionnés au bord d'attaque, qui sont liés par un unique paramètre et se déplacent en même temps dans deux directions opposées pour contrôler efficacement le rayon de bord d'attaque.

L'outil de paramétrisation et de déformation volumique *SeAnDef* (*Sequential Analytical Deformation*, développé au département DAAP de l'Onera) est utilisé pour déformer à la fois le profil et la maillage volumique correspondant. L'optimisation est conduite en utilisant l'optimiseur *CONMIN*, déjà présenté ci-dessus.

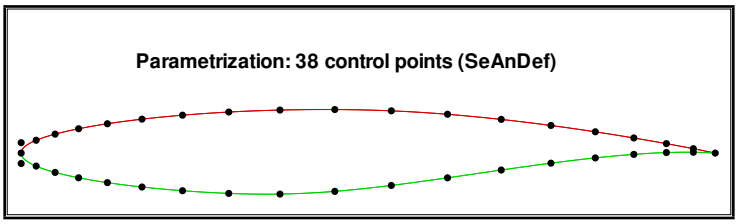


FIG. 4.2 – Points de contrôle de la paramétrisation B-Spline de la géométrie RAE2822

4.4 Chaîne d'optimisation avec placement optimal des paramètres de forme

Il est également possible d'envisager une solution intermédiaire aux deux chaînes de calcul présentées ci-dessus. En effet, le gradient à la surface procure des informations importantes sur la sensibilité de la fonction objectif aux différentes zones du profil. Il permet d'identifier notamment les zones dans lesquelles les déformations auront un impact considérable. Dans cette section, on souhaite donc utiliser ces informations pour placer de manière dynamique les points de contrôle de la paramétrisation B-Spline décrite ci-dessus.

Pour cela, on propose de calculer la variation totale de la composante normale du gradient ($ndJ = -\left\langle \frac{dJ}{dS}, n_S \right\rangle$) pondérée par la taille de l'élément de surface correspondant :

$$\begin{aligned} F_d(1) &= 0, \\ F_d(n) &= F_d(n-1) + |ndJ_n - ndJ_{n-1}| \, \|p_n - p_{n-1}\| , \end{aligned}$$

où n est l'indice des points de la surface ($p_n \forall n \in \{1, N\}$). On répartit alors les M points de contrôle que l'on souhaite placer de telle sorte qu'ils coupent en partie égale la fonction F_d (voir la figure 4.3).

En pratique, la fonction F_d a tendance à concentrer les points de contrôle uniquement dans les zones de fortes variations. Afin d'atténuer cet effet par un effet de répartition uniforme, on ajoute une puissance υ à la fonction F_d :

$$\begin{aligned} F_d^{\upsilon}(1) &= 0, \\ F_d^{\upsilon}(n) &= F_d^{\upsilon}(n-1) + |ndJ_n - ndJ_{n-1}|^{\upsilon} \, \|p_n - p_{n-1}\| . \end{aligned}$$

Par exemple, les positions de 16 points de contrôle qui correspondent à $\upsilon = 0.4$ et $\upsilon = 0.7$ sont données respectivement sur les figures 4.4 et 4.5. La valeur de 0.7 a finalement été retenue et les calculs qui seront présentés par la suite ont été effectué avec la distribution correspondante, à laquelle a été ajoutée les deux points de contrôle placés sur le bord d'attaque et le bord de fuite afin de maintenir ces points fixes au cours de l'optimisation.

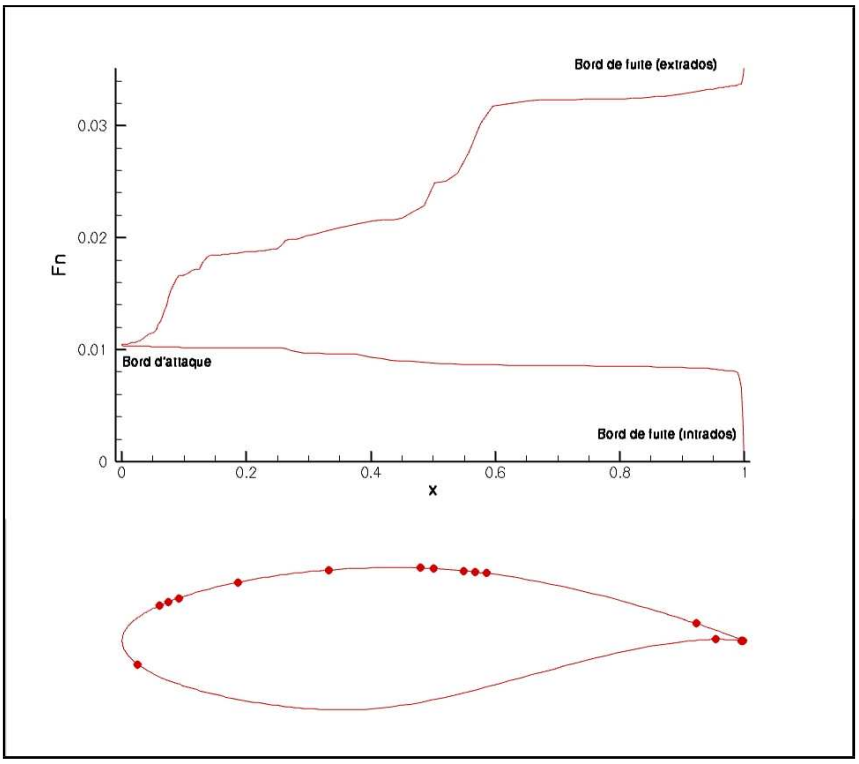


FIG. 4.3 – Placement de 16 points de contrôle sur la fonction F_d

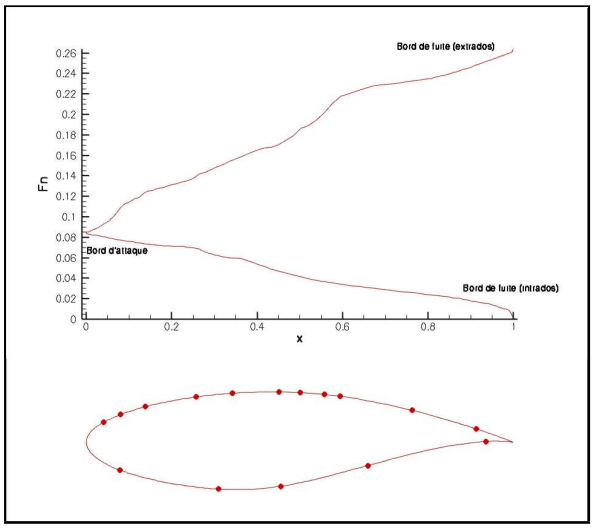


FIG. 4.4 – Placement de 16 points de contrôle sur la fonction $F_d^{0.4}$

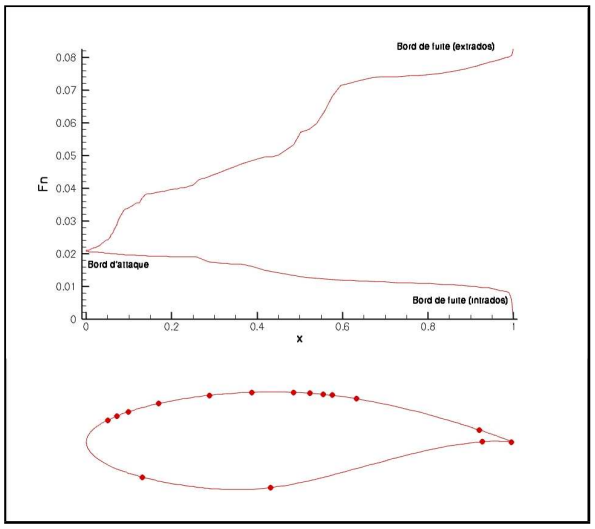


FIG. 4.5 – Placement de 16 points de contrôle sur la fonction $F_d^{0.7}$

4.5 Application à l'optimisation d'un profil soumis à un écoulement de fluide visqueux

Le profil de départ de l'optimisation est le profil RAE2822, déjà utilisé à la fin du troisième chapitre. Le maillage (deux domaines, 32832 points) et les conditions de vol de l'écoulement ($M_\infty = 0.730$, $AoA = 2.79$ et $Re = 6.5 \times 10^6$) sont identiques.

4.5.1 Présentation des cas tests

Dans chacun des calculs détaillés ci-dessous, la chaîne est initialisée avec la géométrie du profil RAE2822 aux conditions de vol présentées ci-dessus. Quatre optimisations, similaires à celles introduites par Kim et al. [KAJ00], sont considérées :

1. minimisation de la traînée à angle d'attaque fixé $AoA = 2.79$;
2. maximisation de la portance à angle d'attaque fixé $AoA = 2.79$;
3. minimisation de la traînée à coefficient de portance fixé $CL_0 = 0.8$;
4. maximisation de la portance à coefficient de traînée fixé $CD_0 = 0.0110$.

Dans l'étude de Kim et al., la surface du profil était paramétrée par 50 bosses de Hicks-Henne réparties uniformément sur l'intrados et l'extrados. Dans notre travail, les mêmes calculs sont effectués par les trois chaînes de calculs détaillées dans ce chapitre. On notera respectivement S_1, S_2 et S_3 les stratégies d'optimisation sans paramètre de forme, avec une paramétrisation déterminée a priori et avec des paramètres de forme positionnés à partir de $\frac{d\tilde{J}}{dS}$.

4.5.2 Optimisation basée sur $\frac{d\tilde{J}}{dS}$ (S_1)

Le processus d'optimisation est effectué à deux reprises :
- d'abord en utilisant le gradient issu de la méthode adjointe avec une hypothèse de viscosité turbulente figé ;
- dans un second temps, en linéarisant le modèle de turbulence dans la méthode adjointe.

Il est interrompu lorsque la fonction objectif cesse d'évoluer. Les contraintes géométriques, décrites ci-dessus, sont définies afin de contraindre l'épaisseur des profils à ne pas diminuer au delà de 80 pour cent de l'épaisseur du profil initial.

Minimisation de la traînée à angle d'attaque fixé

La somme de la traînée d'onde (CDw) et de la traînée de pression visqueuse (CDvp) calculée par l'outil FFD72 [EMVGB06] est minimisée. Le calcul est effectué sans modifié l'angle d'attaque ($AoA = 2.79$). On ne garantit donc pas que le coefficient de traînée reste à un niveau minimal. Pour chaque configuration, la traînée champ lointain (CDff) est également calculée.

La composante normale du gradient en chaque nœud de la surface du profil est représentée sur les figures 4.6 et 4.7 pour les deux types de méthodes adjointes mises en œuvre (μ_t figé et linéarisation du modèle de turbulence). Comme on peut le voir, le champ est fortement perturbé, et de manière très irrégulière. La procédure de lissage est donc indispensable pour conserver des profils lisses durant le processus. De plus, on remarque que la composante normale du gradient est forte au bord d'attaque et au bord de fuite, alors qu'on souhaite préserver la position de ces points. Par conséquent, le gradient est fixé à zéro aux bords.

Le champ lissé est donné sur les figures 4.8 et 4.9. Les variations de basse fréquence du gradient normal sont conservées alors que les hautes fréquences sont éliminées par le processus de lissage mais des oscillations apparaissent en raison de l'utilisation de fonctions splines. Au bord d'attaque et au bord de fuite, le champ est réduit progressivement à une valeur nulle.

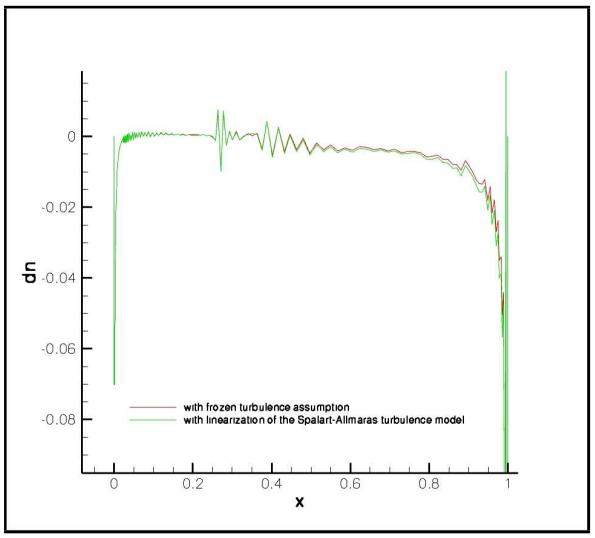


FIG. 4.6 – Composante normale du gradient de la traînée à l'intrados avec hypothèse μ_t figé (rouge) et linéarisation du modèle de Spalart-Allmaras (vert)

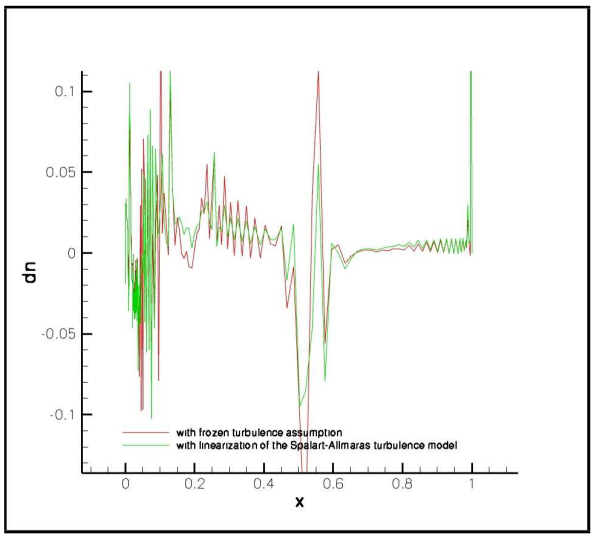


FIG. 4.7 – Composante normale du gradient de la traînée à l'extrados avec hypothèse μ_t figé (rouge) et linéarisation du modèle de Spalart-Allmaras (vert)

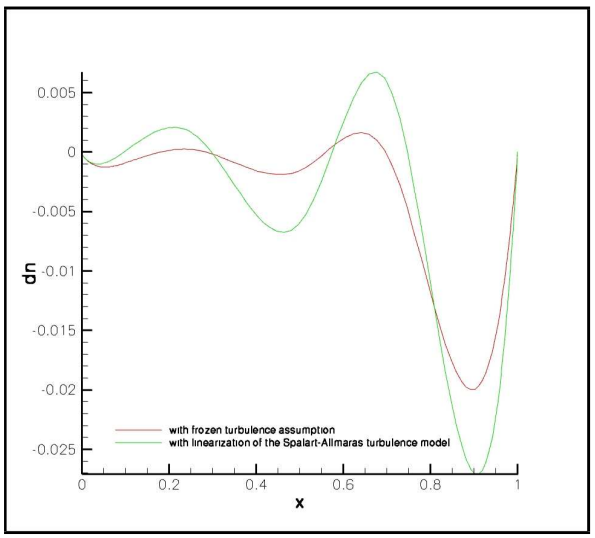


FIG. 4.8 – Résultat du lissage de la composante normale du gradient de la traînée à l'intrados avec hypothèse μ_t figé (rouge) et linéarisation du modèle de Spalart-Allmaras (vert)

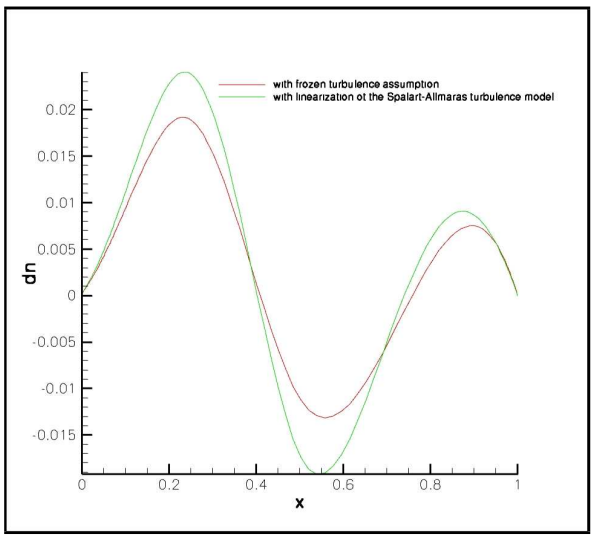


FIG. 4.9 – Résultat du lissage de la composante normale du gradient de la traînée à l'extrados avec hypothèse μ_t figé (rouge) et linéarisation du modèle de Spalart-Allmaras (vert)

Enfin, les profils optimisés et la distribution de pression à la surface de ceux-ci sont donnés sur les figures 4.10 et 4.11 pour les deux méthodes de calcul de gradient.

Au cours de l'optimisation, la traînée champ lointain a été réduite de 143 points à 98 points lorsque l'hypothèse μ_t figé est utilisé (respectivement 102 points lorsque le modèle de Spalart-Allmaras est linéarisé). Dans le même temps, le coefficient de portance est diminué de 0.757 à 0.225 (respectivement 0.258). La réduction de la traînée s'explique essentiellement par l'élimination de la traînée d'onde qui passe de 27 points à 2 points (resp. 0 point). Enfin, on peut noter

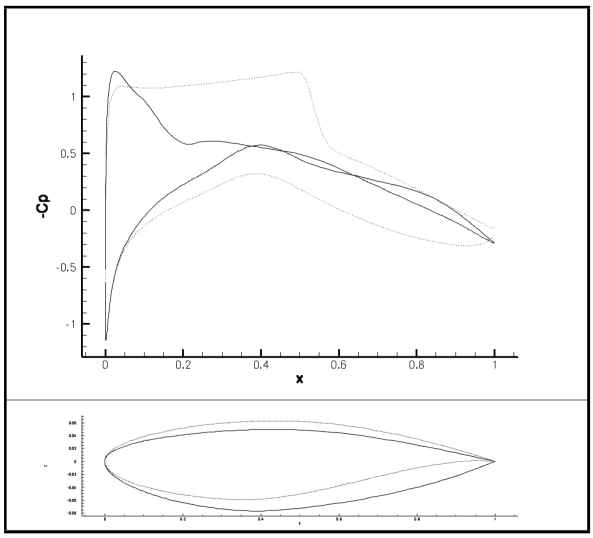


FIG. 4.10 – Distribution de pression à la surface du profil initial (pointillé) et du profil optimisé par S_1 (trait plein) par minimisation de traînée à angle d'attaque fixé avec hypothèse μ_t figé

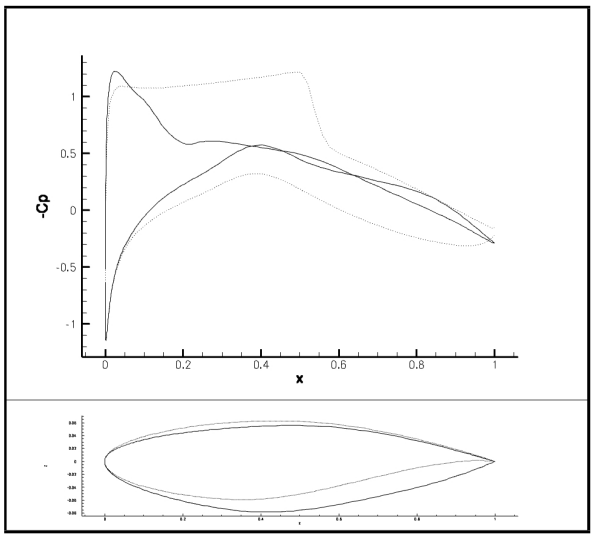


FIG. 4.11 – Distribution de pression à la surface du profil initial (pointillé) et du profil optimisé par S_1 (trait plein) par minimisation de traînée à angle d'attaque fixé avec linéarisation du modèle de Spalart-Allmaras

que la composante de traînée de frottement qui n'est pas considérée dans la fonction objectif de l'optimisation (car le logiciel *elsA* ne permet pas le calcul de sa dérivée) augmente de 5 points pendant les deux optimisations.

Maximisation de la portance à angle d'attaque fixé

On cherche maintenant à maximiser le coefficient de portance du profil RAE2822 en modifiant sa forme sans changer l'angle d'attaque. La fonction objectif est calculé par FFD72 comme la somme de la composante de pression (CLp), qui est prédominante, et de la composante de frottement (CLf).

Les figures 4.12 et 4.13 montrent le résultat de l'optimisation.

La portance augmente de 0.757 à 0.901 (respectivement 0.916), alors que la traînée passe de 143 points à 146 (respectivement 163).

Minimisation de la traînée sous contrainte de portance

La minimisation de la somme de la traînée d'onde (CDw) et de la traînée de pression visqueuse (CDvp) est effectuée avec une contrainte de portance minimum $CL_0 = 0.8$. Par conséquent, l'angle d'attaque est ajouté aux paramètres de l'optimisation et sa valeur initiale est fixée à $AoA = 3.09$. La figure 4.14 montre le résultat du calcul avec l'hypothèse μ_t figé. La traînée est réduite de 172 à 120 points, alors que dans le même temps, l'angle d'attaque est modifiée de 3.09 à 3.068.

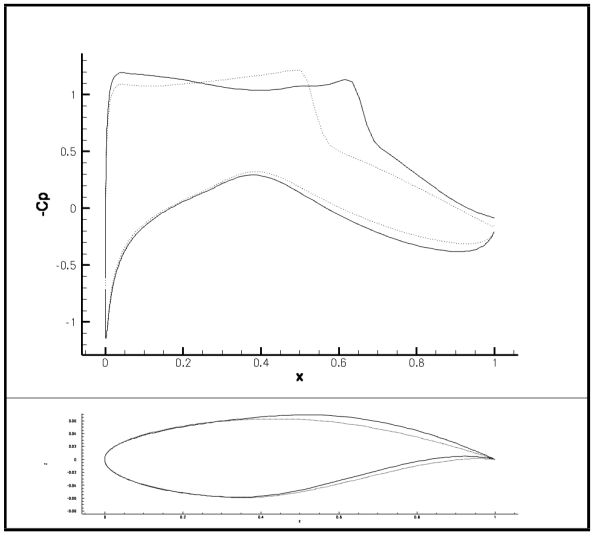


FIG. 4.12 – Distribution de pression à la surface du profil initial (pointillé) et du profil optimisé par S_1 (trait plein) par maximisation de portance à angle d'attaque fixé avec hypothèse μ_t figé

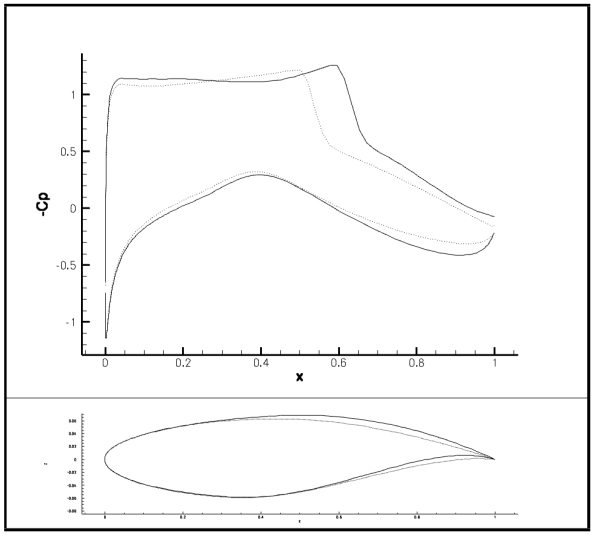


FIG. 4.13 – Distribution de pression à la surface du profil initial (pointillé) et du profil optimisé par S_1 (trait plein) par maximisation de portance à angle d'attaque fixé avec linéarisation du modèle de Spalart-Allmaras

Maximisation de la portance sous contrainte de traînée

Enfin, le coefficient de portance (CL) est maximisé avec une contrainte de traînée maximum $CD_0 = 0.0110$. A nouveau, l'angle d'attaque est considéré comme paramètres de l'optimisation. Le résultat du calcul est donné sur la figure 4.15. Le coefficient de portance augmente de 0.800 à 0.998 et l'angle d'attaque de la configuration optimale est $AoA = 3.088$.

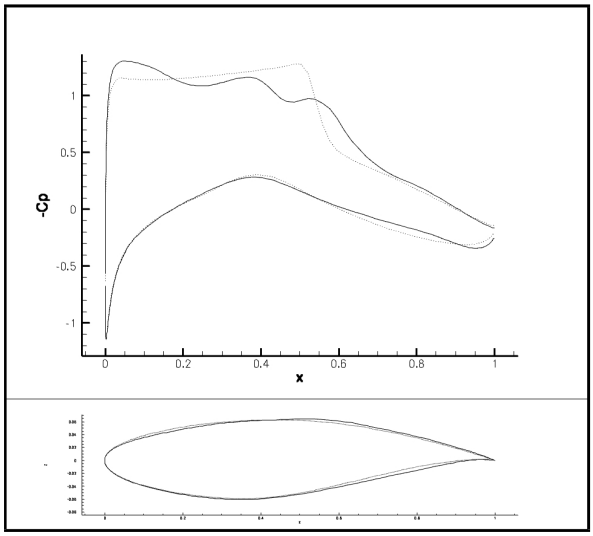


FIG. 4.14 – Distribution de pression à la surface du profil initial (pointillé) et du profil optimisé par S_1 (trait plein) par minimisation de traînée avec contrainte de portance $CL \geq = 0.8$ avec hypothèse μ_t figé

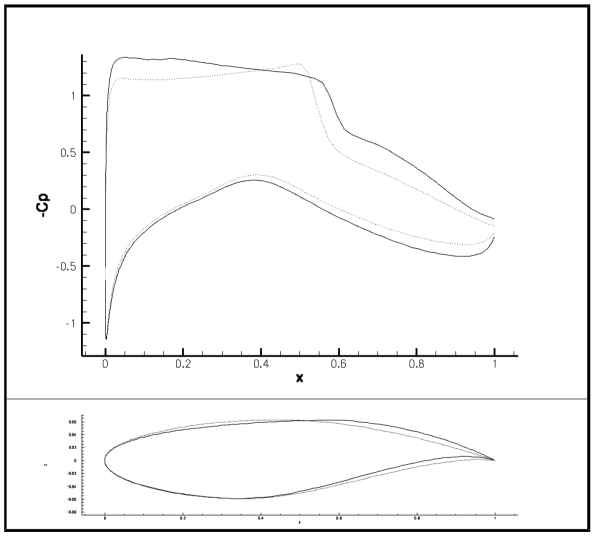


FIG. 4.15 – Distribution de pression à la surface du profil initial (pointillé) et du profil optimisé par S_1 (trait plein) par maximisation de portance avec contrainte de traînée $CD \leq = 0.0110$ avec hypothèse μ_t figé

4.5.3 Optimisation avec paramètres de forme répartis a priori (S_2)

Enfin, les mêmes calculs que précédemment sont effectués sur la chaîne d'optimisation avec paramètres de forme répartis a priori. Comme ci-dessus, les calculs ne sont effectués que dans le cas de la méthode adjointe avec hypothèse μ_t figé.

Minimisation de la traînée à angle d'attaque fixé

Au cours de l'optimisation, le coefficient de traînée a diminué de 143 à 104 points. Le profil initial et le profil optimisé sont donné sur la figure 4.16. A nouveau, c'est la disparition de la traînée d'onde qui explique la diminution de la traînée.

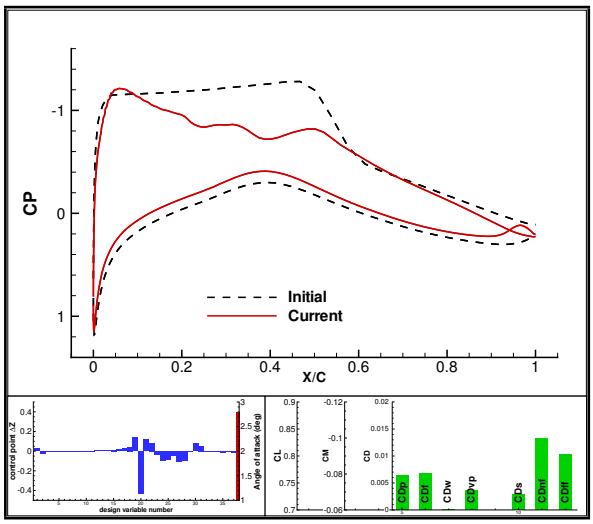


FIG. 4.16 – Distribution de pression à la surface du profil initial (pointillé) et du profil optimisé par S_2 (trait plein) par minimisation de traînée à angle d'attaque fixé avec hypothèse μ_t figé

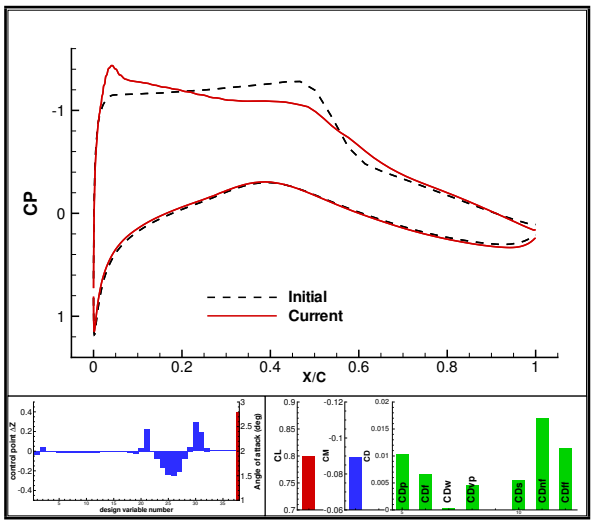


FIG. 4.17 – Distribution de pression à la surface du profil initial (pointillé) et du profil optimisé par S_2 (trait plein) par minimisation de traînée avec contrainte de portance $CL \geq = 0.8$ avec hypothèse μ_t figé

Minimisation de la traînée sous contrainte de portance

Lorsque l'on considère une contrainte sur la portance, le coefficient de traînée champ lointain est réduit de 172 à 115 points. Le profil optimisé est très proche de la contrainte ($CL = 0.798$). Les résultats sont représentés sur la figure 4.17.

4.5.4 Optimisation avec paramètres de forme placés à partir de $\frac{d\tilde{J}}{dS}$ (S_3)

Enfin, on effectue un calcul de minimisation de traînée sous contrainte de portance avec une paramétrisation dynamique de la surface. Afin de comparer les résultats, il est ici proposé d'utiliser la répartition des 16 points de contrôle donnée ci-dessus (voir la figure 4.5) auxquels on ajoute un point au bord d'attaque et au bord de fuite. Le calcul est donc ici effectué avec moitié moins de paramètres de forme que ceux du paragraphe précédent.

Au cours de l'optimisation, le coefficient de traînée passe de 172 à 111 points, alors que la la portance a une valeur très proche de la contrainte ($CL = 0.799$). Le profil initial et le profil optimisé, ainsi que les distributions de pression à la surface, sont donnés sur la figure 4.18.

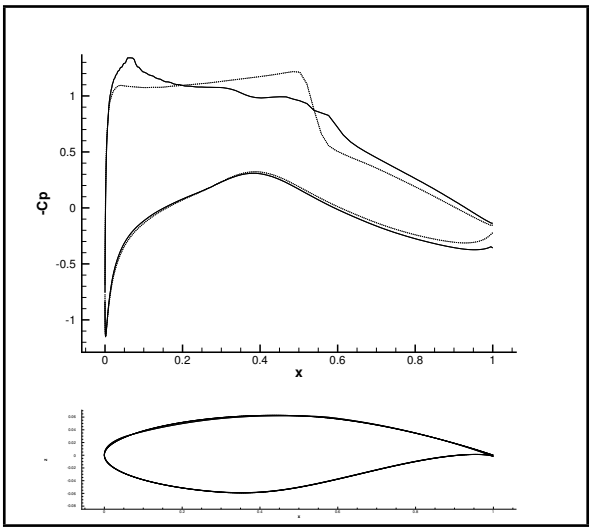


FIG. 4.18 – Distribution de pression à la surface du profil initial (pointillé) et du profil optimisé par S_3 (trait plein) par minimisation de traînée à angle d'attaque fixé avec hypothèse μ_t figé

4.5.5 Comparaison et commentaires des résultats

Les résultats des calculs sont résumés dans le tableau 4.1, et sont comparés aux résultats donnés par Kim et al. [KAJ00] (50 bosses de Hicks-Henne) sur des cas test similaires. La comparaison exacte des performances est délicate puisque les outils numériques utilisés ne sont pas similaires. Par conséquent, même les performances sur le profil initial ne sont pas les mêmes.

Sur chacun des cas, la méthode sans paramètre a produit des résultats comparables aux approches classiques avec paramètres de forme. En particulier, les cas de maximisation de traînée semble profiter de la flexibilité de déformation que procure la déformation de chaque point du maillage. A l'inverse, les optimisations sous contraintes rencontrent des difficultés. Cela s'explique sans doute par le fait que la procédure de lissage, qui tend à uniformiser le champ de déformation, doit être appliquée deux fois : une première fois pour lisser indépendamment les fonctions objectif et contraintes, et une seconde pour lisser la direction de descente obtenue à partir de ces fonctions dans la méthode des directions admissibles.

Conclusion

Dans ce chapitre, nous avons cherché à utiliser au mieux le bénéfice de la méthode adjointe. Pour cela, nous avons exploré deux idées principales. La première est que les dérivées à la surface, dont le calcul a été rendu possible par la méthode adjointe, peuvent être exploitées directement dans le cadre d'une méthode d'optimisation où les nœuds du maillage surfacique sont utilisés comme paramètres de l'optimisation. Il a alors été nécessaire de mettre au point une méthode

	min CD	max CL	min CD sous $CL \geq CL_0$	max CL sous $CD \leq CD_0$
S_1 μ_t figé $CD_0 = 0.0110$ $CL_0 = 0.8$	$0.0143 \Rightarrow 0.0098$	$0.756 \Rightarrow 0.901$	$0.0172 \Rightarrow 0.0120$	$0.8 \Rightarrow 0.998$
S_1 lin. S.A. $CD_0 = 0.0110$ $CL_0 = 0.8$	$0.0143 \Rightarrow 0.0102$	$0.756 \Rightarrow 0.916$	$0.0172 \Rightarrow 0.0118$	$0.8 \Rightarrow 0.818$
S_2 μ_t figé $CL_0 = 0.8$	$0.0143 \Rightarrow 0.0104$		$0.0172 \Rightarrow 0.0115$	
S_3 μ_t figé $CL_0 = 0.8$			$0.0172 \Rightarrow 0.0111$	
Kim et al. [KAJ00] lin. S.A. $CD_0 = 0.0143$ $CL_0 = 0.83$	$0.0152 \Rightarrow 0.0100$	$0.799 \Rightarrow 0.857$	$0.0167 \Rightarrow 0.0109$	$0.799 \Rightarrow 0.976$

Tab. 4.1 – Résumé des résultats des différentes approches sur les quatre cas tests

de lissage afin de conserver au cours du processus des formes lisses qui sont indispensables en aérodynamique. Nous avons proposé ici une nouvelle méthode de lissage, directement reliée à la courbure d'un profil déformé plutôt que basée sur le champ de déformation. La chaîne d'optimisation ainsi définie a permis d'obtenir de bons résultats dans le cadre d'optimisations locales sans contrainte, et a rencontré des difficultés lorsque des contraintes aérodynamiques sont ajoutées au problème. Néanmoins, elle reste encore à tester sur d'autres configurations.

Dans un second temps, nous avons proposé une approche intermédiaire qui consiste à utiliser les dérivées à la surface pour positionner les points de contrôle d'une paramétrisation. Dans ce travail, c'est une paramétrisation par B-Spline qui a été utilisée. Cette approche augmente la densité de points de contrôle dans les zones de fortes sensibilités de la fonction objectif et semble donner de meilleurs résultats que le placement à priori. Mais il est nécessaire d'effectuer une étude plus systématique de l'impact de la paramétrisation et d'étendre cette approche à d'autres cas tests pour conclure quant à sa pertinence.

Conclusion

Dans ce travail, nous avons proposé des extensions techniques originales pour l'optimisation de forme en aérodynamique. Un état de l'art du domaine et la présentation des méthodes numériques utilisées à l'Onera ont fait l'objet de notre premier chapitre.

La première partie de cette thèse a concerné les méthodes d'optimisation globale, dont l'utilisation sur des problèmes industriels est aujourd'hui limitée par un fort besoin en ressources de calcul. Dans une étude bibliographique importante, nous avons cherché à faire le point sur les développements technologiques effectués ces dernières années dans le but de généraliser ce type de méthodes. Nous nous sommes par conséquent tournés vers l'utilisation des modèles de substitution. Ainsi, le second chapitre de ce manuscrit est consacré à l'étude de trois types de modèle (Krigeage, réseaux de fonctions à bases radiales, régression à vecteur de support) qui ont été implémentés dans un code de calcul en Fortran 90. Afin de permettre une utilisation systématique de ces modèles, une procédure d'ajustement automatique des paramètres internes a été mise en place. Elle est basée sur la minimisation de l'erreur *leave-one-out* par un algorithme d'optimisation globale. Des développements mathématiques ont permis le calcul (ou l'approximation) efficace de cette erreur. Par ailleurs, chaque modèle a été étendu afin de prendre en compte le gradient de la fonction à approcher, souvent disponible en aérodynamique, depuis le développement de la méthode adjointe.

Dans le troisième chapitre, ces modèles ont été utilisés dans le cadre d'un algorithme d'optimisation de fonctions dont l'évaluation requiert des ressources de calcul importantes. Les différents éléments de cet algorithme - méthode d'échantillonnage et critère d'enrichissement - ont ainsi été présentés. Puis l'algorithme a été appliqué à l'optimisation d'une batterie de trois fonctions tests, pour lesquels les résultats ont été convaincants. Ainsi :

- l'utilisation des modèles de substitution permet de réduire sensiblement le nombre d'évaluations de la fonction à optimiser ;
- la prise en compte des gradients dans la construction des modèles permet de construire des modèles plus précis, et donc de réduire à nouveau le nombre d'évaluations nécessaires.

L'algorithme a ensuite été appliqué à des problèmes d'optimisation de forme aérodynamique. Il a permis de résoudre avec succès trois problèmes d'optimisation : un problème inverse dans un espace de deux paramètres de forme et une optimisation de traînée, sous contrainte de portance, avec cinq variables, dans le cas d'écoulements de fluide parfait ; une optimisation du ratio traînée sur portance avec trois paramètres de forme dans le cas de fluide visqueux.

Néanmoins, on peut noter deux limitations. Tout d'abord, l'utilisation des gradients ne permet d'accélérer que légèrement la méthode, en raison de la faible précision des gradients. En

effet, ceux-ci sont entachés d'erreurs numériques dues à la résolution approchée du système linéaire de la méthode adjointe. L'amélioration de la précision du calcul de gradient est donc une première piste de travail intéressante. A cet effet, on peut, par exemple, se tourner vers les outils de différenciation automatique, dont l'usage s'est généralisé ces dernières années en utilisant par exemple l'outil TAPENADE [HP04]. Une seconde difficulté de l'utilisation des modèles est que leur complexité croît rapidement avec la dimension. Ainsi, pour le krigeage et les réseaux RBF, la taille du système linéaire à résoudre augmente linéairement en fonction du nombre de points de l'échantillon de construction. Cette dépendance linéaire pénalise également les méthodes SVR, pour lesquelles le problème à résoudre est alors quadratique. Elle est encore plus problématique pour le cas des modèles avec gradients, puisque le coût de calcul dépend alors du produit du nombre de points par la dimension de l'espace. Comme par ailleurs le système linéaire n'est pas résolu une seule fois, mais un grand nombre de fois (en raison de la procédure de réglage des paramètres internes), cela constitue un handicap majeur de ce type de méthode. On peut envisager deux pistes de travail pour contourner cette difficulté :

- la mise en place d'une méthode de réglage des paramètres internes plus efficace, c'est-à-dire nécessitant moins de résolutions du système linéaire du modèle : on pourrait réfléchir à ne pas régler les paramètres internes du modèle à chaque enrichissement, mais par exemple seulement toutes les 10 itérations, ou repartir des valeurs optimales de l'itération précédente, pour débuter l'optimisation des paramètres internes ;
- l'utilisation de modèles de substitution spécialement développés pour l'approximation de problèmes de grande dimension [Ant05].

Par ailleurs, il conviendrait d'appliquer notre algorithme à la résolution d'autres problèmes d'optimisation de forme, car le type de fonction objectif et la paramétrisation géométrique impactent fortement la forme de la fonction mathématique, et donc les performances des approches basées sur des modèles de substitution. La méthode de prise en compte des contraintes influe également : dans ce travail, les contraintes sont traitées par pénalisation de la fonction d'intérêt mais des études [Sas02] ont été menées pour modifier le critère d'enrichissement en fonction des contraintes, et donc les traiter spécifiquement.

Dans la deuxième partie de notre travail, nous avons mis au point un algorithme d'optimisation locale sans paramètres de forme. A cet effet, nous avons utilisé la méthode adjointe pour calculer le gradient de la fonction objectif et des contraintes par rapport à chaque point du maillage de peau de la forme solide. Une fois lissée, cette quantité est utilisée comme direction de descente d'un algorithme classique de plus forte pente. Si les méthodes de lissage *Implicit Residual Smoothing* sont classiquement mises en œuvre, nous avons rencontré des difficultés dans l'utilisation de cet opérateur, notamment pour la déformation des profils d'aile au niveau du bord d'attaque. Par conséquent, nous avons mis au point un algorithme de lissage basé sur la régularité de la courbure du profil déformé. Cela présente l'avantage d'assurer que le profil reste géométriquement acceptable durant le processus d'optimisation. Appliquée à l'optimisation d'un profil RAE2822, notre approche semble donner de bons résultats pour la résolution des problèmes sans contraintes : en particulier, lorsque la portance est maximisée, la grande flexibilité de déformation offerte par cette méthode est un atout non négligeable. Néanmoins, lorsque l'on ajoute des contraintes au problème d'optimisation, les résultats sont moins convaincants,

ce qui s'explique sans doute par la nécessité d'appliquer plusieurs fois la procédure de lissage. Les procédures d'optimisation sans paramétrisation nous semblent donc une piste de travail à approfondir. En particulier, le développement d'un opérateur de lissage performant reste un sujet de recherche assez peu exploré.

Par ailleurs, dans le même cadre, nous avons proposé une approche hybride entre optimisation sans paramètres et approche plus classique. A partir du gradient calculé par rapport aux points de la surface, il est en effet possible de positionner les paramètres de forme dans les zones à fort gradient. Sur le cas test que nous avons présenté, pour le même nombre de paramètres de forme, cela conduit à de meilleurs résultats. Néanmoins, afin de ne pas tirer de conclusions hâtives, il nous semble indispensable d'approfondir ce travail en étudiant l'impact de la richesse de la description géométrique de la forme sur les performances de l'optimisation.

L'ensemble de nos applications aérodynamiques, pour des raisons de coût de calcul, ont été limitées à l'optimisation de profil. La question de l'extension à des configurations tridimensionnelles mérite d'être évoquée. Elle ne pose pas de difficultés majeures pour l'optimisation globale avec modèles de substitution, hormis, bien sûr, la durée des optimisations, qui peut atteindre plusieurs semaines, si le calcul d'une configuration prend plusieurs heures. Par contre, cette extension requiert des développements supplémentaires pour l'optimisation sans paramètres de forme. Ainsi, le lissage doit alors être effectué dans un espace de deux dimensions. Si l'on souhaite étendre notre opérateur de lissage, il est nécessaire de redéfinir le critère de conformité pour une forme tridimensionnelle, et donc de calculer la courbure d'une surface, au lieu de celle d'une courbe simple. Cela permettrait de définir un opérateur de lissage particluièrement adapté à des configurations tridimensionnelles. C'est donc une piste de recherche intéressante pour les prochaines années.

Annexe A

Méthode de points intérieurs [Van94]

Dans cette première annexe, on présente la méthode de points intérieurs pour la résolution des problèmes quadratiques convexes. Cette méthode est utilisée dans le cadre d'un modèle à régression par vecteurs de supports détaillé dans le second chapitre de ce mémoire.

A.1 Formes primale et duale du problème d'optimisation

La forme primale du problème à résoudre est la formulation canonique d'un problème de programmation quadratique :

$$\begin{aligned} \min_x \quad & f + c^T x + \frac{1}{2} x^T H x, \\ \text{avec les contraintes :} \quad & b \leq Ax \leq b + r, \\ \text{et :} \quad & l \leq x \leq u, \end{aligned} \tag{A.1}$$

où $H \in \mathbb{R}^{n \times n}$, $A \in \mathbb{R}^{m \times n}$, $x, c, l, u \in \mathbb{R}^n$, $b, r \in \mathbb{R}^m$ et $f \in \mathbb{R}$.

Dans cette partie, les notations utilisées seront donc les notations ci-dessus. Afin de supprimer les contraintes d'inégalité, on ajoute des variables d'écarts à ce problème pour obtenir la formulation suivante :

$$\begin{aligned} \min_x \quad & f + c^T x + \frac{1}{2} x^T H x, \\ \text{avec les contraintes :} \quad & Ax - w = b, \\ & Ax - w = b, \\ & x - g = l, \\ & x + t = u, \\ & w + p = r, \\ & g, w, t, p \geq 0. \end{aligned} \tag{A.2}$$

On peut alors établir la forme dual de ce problème :

$$\begin{aligned} \max_{y} \quad & f + b^T y - \frac{1}{2} x^T H x + l^T z - u^T s - r^T q, \\ \text{avec les contraintes :} \quad & A^T y + z - s - Hx = c, \\ & y + q - v = 0, \\ & z, v, s, q \geq 0. \end{aligned} \tag{A.3}$$

A.2 Conditions KKT d'optimalité

On peut alors formuler les conditions KKT d'optimalité :

$$\begin{aligned} &\text{faisabilité primal :} \\ Ax - w &= 0, \\ x - g &= l, \\ x + t &= u, \\ w + p &= r; \\ &\text{faisabilité dual :} \\ A^T y + z - s - Hx &= c, \\ y + q - v &= 0; \\ &\text{conditions de complémentarité :} \\ GZ &= 0, \\ VW &= 0, \\ ST &= 0, \\ PQ &= 0, \end{aligned}$$

où les termes en majuscules G, Z, V, W, S, T, P, Q représentent les matrices diagonales dont les termes sont ceux du vecteur désigné par la même lettre minuscule.

A.3 Chemin central

L'idée central de la méthode est donc de relaxer les conditions supplémentaires afin de délimiter un chemin paramétrée des points faisables à la fois pour la forme primale et pour la forme duale du problème résolue. Un paramètre $\mu > 0$ est donc introduit et les conditions KKT

sont reformulée de la manière suivante :

$$\begin{aligned} Ax - w &= 0, \\ x - g &= l, \\ x + t &= u, \\ w + p &= r, \\ A^T y + z - s - Hx &= c, \\ y + q - v &= 0, \\ GZe &= \mu e, \\ VWe &= \mu e, \\ STe &= \mu e, \\ PQe &= \mu e. \end{aligned}$$

Ce chemin central converge vers la solution des conditions KKT lorsque μ tend vers zéro. A partir d'un point situé sur le chemin central, l'algorithme va donc chercher à s'approcher de manière itérative de cette solution.

A.4 Recherche d'un point initial

La première étape d'un algorithme de point intérieur est donc de se positionner sur le chemin central. Pour cela, il faut trouver un point qui satisfait l'ensemble des contraintes. A cet effet, on peut par exemple choisir x et y comme solution du système linéaire suivant :

$$\begin{pmatrix} -(H+I) & A^T \\ A & I \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ b \end{pmatrix}. \tag{A.4}$$

A.5 Évolution sur le chemin central

A partir d'un point initial, la méthode de points intérieurs cherche donc à cheminer sur le chemin central paramétré par μ. Supposons donc qu'à l'étape k, l'algorithme est positionné sur un point du chemin central x_k correspondant à la valeur μ_k et cherche à évoluer vers un point correspondant à une nouvelle valeur $\mu_{k+1} < \mu_k$. Pour cela, on propose d'écrire chacune des variables à l'étape $k+1$ comme la somme de la même variable à l'étape k et d'un terme de mise

à jour, puis de calculer cet incrément en résolvant le système des conditions KKT :

$$\begin{aligned}
x_{k+1} &= x_k + \frac{\delta x}{\alpha},\\
g_{k+1} &= g_k + \frac{\delta g}{\alpha},\\
w_{k+1} &= w_k + \frac{\delta w}{\alpha},\\
t_{k+1} &= t_k + \frac{\delta t}{\alpha},\\
p_{k+1} &= p_k + \frac{\delta p}{\alpha},\\
y_{k+1} &= y_k + \frac{\delta y}{\alpha},\\
z_{k+1} &= z_k + \frac{\delta z}{\alpha},\\
v_{k+1} &= v_k + \frac{\delta v}{\alpha},\\
s_{k+1} &= s_k + \frac{\delta s}{\alpha},\\
q_{k+1} &= q_k + \frac{\delta q}{\alpha}.
\end{aligned}$$

Les pas α_p et α_d sont introduits de telle sorte de pouvoir raccourcir l'itération dans le cas où celle-ci conduirait les variables à violer les contraintes les encadrant. Le système KKT peut alors s'écrire (dans la suite, pour des raisons de lisibilité, on omettra l'indice k des variables ; ainsi, x_k sera noté x) :

$$\begin{aligned}
A\Delta x - \Delta w &= b - Ax + w(=\rho),\\
\Delta x - \Delta g &= l - x - g(=\nu),\\
\Delta x + \Delta t &= u - x - t(=\tau),\\
\Delta w + \Delta p &= r - w - p(=\alpha),\\
A^T\Delta y + \Delta z - \Delta s - H\Delta x &= c - A^T y - z + s + Hx(=\sigma),\\
-\Delta y - \Delta q + \Delta v &= y + q - v(=\beta),\\
ZG^{-1}\Delta g + \Delta z &= G^{-1}\mu - z - G^{-1}\Delta g\Delta z(=\gamma_z),\\
WV^{-1}\Delta v + \Delta w &= V^{-1}\mu - w - V^{-1}\Delta v\Delta w(=\gamma_w),\\
ST^{-1}\Delta t + \Delta s &= T^{-1}\mu - s - T^{-1}\Delta t\Delta s(=\gamma_s),\\
QP^{-1}\Delta p + \Delta q &= P^{-1}\mu - q - P^{-1}\Delta p\Delta q(=\gamma_q).
\end{aligned} \tag{A.5}$$

Le système ci-dessus est un système linéaire à l'exception des quatre dernières équations qui possèdent des non linéarités que nous avons regroupées dans les termes γ. Il ne peut donc pas être résolu directement par un solveur linéaire. Une méthode dite prédicteur-correcteur introduite par Mehrota [Meh92] est utilisée ici.

Dans cette approche, le système A.5 est résolu une première fois sans les non linéarités (donc en posant $\gamma_X = -X$) : c'est l'étape de prédiction. Les nouvelles valeurs des paramètres ainsi obtenues sont utilisées pour estimer une valeur approximative de μ et pour réintroduire les

termes non linéaires. Le système est résolu une seconde fois : il s'agit de l'étape de correction.

A.6 Estimation de μ

Il reste à estimer une valeur objectif de μ pour l'étape de correction. On sait que l'algorithme converge vers la solution du problème KKT lorsque μ tend vers 0, donc on cherche à réduire la valeur de μ. Mais en même temps, une réduction trop forte de μ conduirait à une différence importante entre l'étape de prédiction et l'étape de correction, ce qui risque de limiter les performances de l'algorithme. Par conséquent, Vanderbei propose un compromis illustré par la formule suivante :

$$\mu = \frac{z^T g + v^T w + s^T t + p^T q}{2n + 2m} \left(\frac{\alpha - 1}{\alpha + 10} \right)^2 . \tag{A.6}$$

La première partie de la formule permet d'estimer la valeur de μ à partir du point courant situé sur le chemin central. Le second terme pondère cette valeur en fonction de la taille du pas de l'itération précédente. Ainsi, si l'algorithme a fait un pas important lors de l'itération précédente, c'est à dire si l'on se trouve loin de la valeur objectif, alors on peut se permettre de réduire μ plus rapidement.

A.7 Résolution du système linéaire

Le système complet A.5 peut se réduire puisque certaines variables dépendent direcrement l'une de l'autre :

$$\begin{aligned}
\Delta t &= TS^{-1}(\gamma_s - \Delta s), \\
\Delta g &= GZ^{-1}(\gamma_z - \Delta z), \\
\Delta p &= PQ^{-1}(\gamma_q - \Delta q), \\
\Delta v &= VW^{-1}(\gamma_w - \Delta w), \\
\Delta q &= QP^{-1}(\Delta w - \alpha + PQ^{-1}\gamma_q), \\
\Delta z &= ZG^{-1}(\nu + GZ^{-1}\gamma_z - \Delta x), \\
\Delta s &= ST^{-1}(\Delta x - \tau + TS^{-1}\gamma_s), \\
\Delta w &= -\left(VW^{-1} + QP^{-1}\right)^{-1} \left[\beta - VW^{-1}\gamma_w - QP^{-1}\left(\alpha - PQ^{-1}\gamma_q\right) + \Delta y\right],
\end{aligned}$$

ce qui conduit au système KKT réduit :

$$\begin{pmatrix} -(H+D) & A^T \\ A & E \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} \hat{\sigma} \\ \hat{\rho} \end{pmatrix}, \tag{A.7}$$

avec :

$$E = \left(VW^{-1} + QP^{-1}\right)^{-1},$$
$$D = \left(ZG^{-1} + ST^{-1}\right),$$
$$\hat{\sigma} = \sigma - ZG^{-1}\left(\nu + GZ^{-1}\gamma_z\right) - ST^{-1}\left(\tau - TS^{-1}\gamma_s\right),$$
$$\hat{\rho} = \rho - E\left[\beta - VW^{-1}\gamma_w - QP^{-1}\left(\alpha - PQ^{-1}\gamma_q\right)\right].$$

Dans le cadre de cette étude, on résout ce système, symétrique, par les routines fournies par la librairie LAPACK [ABB+99].

A.8 Calcul de la longueur des pas

Une fois les incréments calculés, les variables sont mises à jour à partir de ces incréments et d'un poids permettant d'éviter la violation des contraintes. Dans le cadre de notre étude, suivant les conseils de Vanderbei [Van98], on pose :

$$\alpha = \max_{i\in\{1,n\},j\in\{1,m\}} \left[0.95, -\frac{\Delta g_i}{g_i}, -\frac{\Delta w_j}{w_j}, -\frac{\Delta t_i}{t_i}, -\frac{\Delta p_j}{p_j}, -\frac{\Delta z_i}{z_i}, -\frac{\Delta v_j}{v_j}, -\frac{\Delta s_i}{s_i}, -\frac{\Delta q_j}{q_j}\right]. \quad \text{(A.8)}$$

A.9 Conditions d'arrêt

Il reste maintenant à arrêter l'algorithme une fois atteint un point satisfaisant les conditions KKT avec une certaine précision. Pour cela, suivant toujours les conseils de Vanderbei [Van98], on propose de stopper l'algorithme si un point est compatible avec les contraintes des formes primales et duales du problème, et si les valeurs objectifs du primal et du dual sont suffisamment proches.

$$\begin{aligned} \frac{\sqrt{\rho^T\rho + \tau^T\tau + \alpha^T\alpha + \nu^T\nu}}{b^Tb + 1} &< \epsilon \\ \frac{\sqrt{\sigma^T\sigma + \beta^T\beta}}{c^Tc + 1} &< \epsilon \\ \max\left[-\log\frac{|p_obj - d_obj|}{|p_obj| + 1}, 0\right] & \end{aligned} \quad \text{(A.9)}$$

A.10 L'algorithme

Algorithm 1 Méthode de points intérieurs

Calculer un point initial en résolvant le système A.4
while Conditions d'arrêt A.9 ne sont pas atteintes **do**
  Résoudre le système linéaire A.7 correspondant à l'étape de prédiction
  Estimer la valeur objectif de μ par la formule A.6
  Résoudre le système linéaire A.7 correspondant à l'étape de correction
  Calculer la valeur du pas α par la formule A.8
  Mettre à jour les variables (A.5)
end while

Annexe B

Calcul efficace de l'erreur *leave-one-out* [Rip99]

B.1 *Radial Basis Function*

On va démontrer ici les formules données au paragraphe 2.4.5 que l'on rappelle ci-dessous. Elles permettent le calcul efficace de l'erreur *leave-one-out* d'un réseau RBF construit à partir d'un échantillon E.

$$e_n = \frac{\Phi^{-1} y}{\Phi_{nn}^{-1}}, \tag{B.1}$$

$$e_{loo} = \frac{E^T E}{N} = \frac{Y^T \Phi^{-2} Y}{N diag(\Phi^{-2})}. \tag{B.2}$$

Démonstration. On considère le modèle RBF, défini au second chapitre et construit sur un échantillon $E = \Big\{(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}, \quad i \in \{1, n\}\Big\}$, défini par un vecteur de poids w appliqué à une base de fonctions RBF $\phi(x, y)$. On note Y le vecteur des y_i et Φ la matrice, de taille $(N \times N)$, composée des $\phi(x_i, x_j)$.

Soit $\hat{f}^{(n)}(x)$ l'évaluation en un point x quelconque fournie par le modèle construit à partir de l'échantillon E diminué du point d'indice n (on appellera l'échantillon diminué $E^{(n)}$) :

$$\hat{f}^{(n)}(x) = \sum_{m=1; m \neq n}^{N} w_m^{(n)} \phi(x, x_m).$$

La propriété d'interpolation du réseau RBF fournit le système linéaire suivant, de taille $[(N-1) \times (N-1)]$:

$$\Phi^{(n)} w^{(n)} = Y^{(n)},$$

où $\Phi^{(n)}$ est la matrice Φ diminué de ses n-èmes lignes et colonnes et $w^{(n)}$ et $Y^{(n)}$ les vecteurs w et Y diminués de leurs n-ème ligne.

Remarquons que si un vecteur quelconque $y \in \mathbb{R}^N$ est tel que $y_n = 0$, alors :

$$\Phi y = z \Rightarrow \Phi^{(n)} y^{(n)} = z^n, \tag{B.3}$$

où les vecteurs $y^{(n)}$ et $z^{(n)}$ sont les vecteurs y et z diminués de leurs n-ème ligne.

Soit $u^{(n)} \in \mathbb{R}^N$ la solution du système :

$$\Phi u^{(n)} = \mathbb{1}^{(n)}, \tag{B.4}$$

où $\mathbb{1}^{(n)} \in \mathbb{R}^N$ est le vecteur dont seule la composante n, de valeur 1, est non nulle.

On peut vérifier que $u_n^{(n)} \neq 0$. Sinon, par (B.3) :

$$u_n^{(n)} = 0 \Rightarrow \Phi^{(n)} \left[u_1^{(n)}, \ldots, u_{n-1}^{(n)}, u_{n+1}^{(n)}, \ldots, u_N^{(n)}\right]^T = \mathbf{0}.$$

Alors, comme Φ est régulière, $u^{(n)} = \mathbf{0}$ ce qui n'est pas possible puisque $u^{(n)}$ est solution de (B.4).

Comme $u_n^{(n)} \neq 0$, on peut alors considérer le vecteur suivant :

$$v^{(n)} = w - \frac{w_n}{u_n^{(n)}} u^{(n)}.$$

Alors :

$$\Phi v^{(n)} = \Phi w - \Phi \frac{w_n}{u_n^{(n)}} u^{(n)} = Y - \frac{w_n}{u_n^{(n)}} \mathbb{1}^{(n)}.$$

Comme $v_n^{(n)} = 0$, on peut utiliser (B.3) pour écrire :

$$w^{(n)} = \left[v_1^{(n)}, \ldots, v_{n-1}^{(n)}, v_{n+1}^{(n)}, \ldots, v_N^{(n)}\right]^T.$$

Alors on obtient la formule suivante pour l'erreur au point n :

$$\begin{aligned}
e_n = \left|\hat{f}^{(n)}(x_n) - y_n\right| &= \left|\sum_{m=1, m\neq n}^{N} w_m^{(n)} \phi(x_n, x_m) - y_n\right| \\
&= \left|\sum_{m=1; m\neq n}^{N} v_m^{(n)} \phi(x_n, x_m) - y_n\right| \\
&= \left|\sum_{m=1}^{N} v_m^{(n)} \phi(x_n, x_m) - y_n\right| \\
&= \left|(\Phi v^{(n)})_n - y_n\right| \\
&= \left|\frac{w_n}{u_n^{(n)}}\right| \\
&= \left|\frac{(\Phi^{-1} y)_n}{(\Phi^{-1})_{nn}}\right|,
\end{aligned}$$

et, par définition de l'erreur *leave-one-out* :

$$e_{loo} = \frac{E^T E}{N} = \frac{Y^T \Phi^{-2} Y}{N diag(\Phi^{-2})}.$$

□

B.2 Krigeage

On va montrer ici que la formule de Rippa, présenté au paragraphe 2.4.5 pour la méthode RBF, s'applique également pour le calcul de l'erreur *leave-one-out* d'une interpolation par krigeage construite à partir d'un échantillon E. On rappelle ci-dessous la formule :

$$e_n = \frac{\Phi^{-1} y}{\Phi_{nn}^{-1}}, \tag{B.5}$$

$$e_{loo} = \frac{E^T E}{N} = \frac{Y^T \Phi^{-2} Y}{N diag(\Phi^{-2})}. \tag{B.6}$$

Démonstration. On considère une approximation par krigeage, construite à partir d'un échantillon $E = \left\{(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}, \quad i \in \{1, N\}\right\}$, défini par un vecteur de poids w appliqué à une base de fonctions gaussiennes $\phi(x, y)$. On note Y le vecteur des y_i et $\hat{\Phi}$ la matrice, de taille $(N \times N)$, composée des $\phi(x_i, x_j)$, c'est à dire la matrice Φ privée de sa dernière ligne et de sa dernière colonne.

Soit $\hat{f}^{(n)}(x)$ l'évaluation en un point x quelconque fournie par le modèle construit à partir de l'échantillon E diminué du point d'indice n (on appellera l'échantillon diminué $E^{(n)}$) :

$$\hat{f}^{(n)}(x) = \sum_{m=1; m \neq n}^{N} w_m^{(n)} \phi(x, x_m) + \mu^{(n)}.$$

Les poids w sont obtenus par résolution du système linéaire suivant, de taille $(N \times N)$:

$$\underbrace{\begin{bmatrix} \hat{\Phi}^{(n)} & \mathbb{1} \\ \mathbb{1}^T & 0 \end{bmatrix}}_{\Phi^{(n)}} \begin{bmatrix} w^{(n)} \\ \mu^{(n)} \end{bmatrix} = \begin{bmatrix} Y^{(n)} \\ 0 \end{bmatrix},$$

où $\hat{\Phi}^{(n)}$ est la matrice $\hat{\Phi}$ diminué de ses n-èmes lignes et colonnes et $w^{(n)}$ et $Y^{(n)}$ les vecteurs w et Y diminués de leurs n-ème ligne.

Remarquons que si un vecteur quelconque $y \in \mathbb{R}^{N+1}$ est tel que $y_n = 0$, alors :

$$\Phi y = z \Rightarrow \Phi^{(n)} y^{(n)} = z^{(n)}, \tag{B.7}$$

où les vecteurs $y^{(n)}$ et $z^{(n)}$ sont les vecteurs y et z diminués de leurs n-ème lignes.

Soit $\left[u^{(n)}, \nu_u^{(n)}\right]^T \in \mathbb{R}^{N+1}$ la solution du système :

$$\Phi \begin{bmatrix} u^{(n)} \\ \nu_u^{(n)} \end{bmatrix} = \begin{bmatrix} \mathbb{1}^{(n)} \\ 0 \end{bmatrix}, \tag{B.8}$$

où $\mathbb{1}^{(n)} \in \mathbb{R}^N$ est le vecteur dont seule la composante n, de valeur 1, est non nulle.

On peut vérifier que $u_n^{(n)} \neq 0$. Sinon, par (B.7) :

$$u_n^{(n)} = 0 \Rightarrow \Phi^{(n)} \left[u_1^{(n)} \dots u_{n-1}^{(n)} u_{n+1}^{(n)} \dots u_N^{(n)} \nu_u^{(n)}\right]^T = \mathbf{0}.$$

Alors, comme Φ est régulière, $\left[u^{(n)}, \nu_u^{(n)}\right]^T = \mathbf{0}$ ce qui n'est pas possible puisque $\left[u^{(n)}, \nu_u^{(n)}\right]^T$ est

solution de (B.8).

Comme $u_n^{(n)} \neq 0$, on peut alors considérer le vecteur suivant :

$$\begin{bmatrix} v^{(n)} \\ \nu_v^{(n)} \end{bmatrix} = \begin{bmatrix} w \\ \mu \end{bmatrix} - \frac{w_n}{u_n^{(n)}} \begin{bmatrix} u^{(n)} \\ \nu_u^{(n)} \end{bmatrix}.$$

Alors :

$$\Phi \begin{bmatrix} v^{(n)} \\ \nu_v^{(n)} \end{bmatrix} = \Phi \begin{bmatrix} w \\ \mu \end{bmatrix} - \hat{\Phi} \frac{w_n}{u_n^{(n)}} \begin{bmatrix} u^{(n)} \\ \nu_u^{(n)} \end{bmatrix} = \begin{bmatrix} Y \\ 0 \end{bmatrix} - \frac{w_n}{u_n^{(n)}} \begin{bmatrix} \mathbb{1}^{(n)} \\ 0 \end{bmatrix}.$$

Comme $v_n^{(n)} = 0$, on peut utiliser (B.7) pour écrire :

$$\left[w^{(n)}, \mu^{(n)}\right]^T = \left[v_1^{(n)}, \ldots, v_{n-1}^{(n)}, v_{n+1}^{(n)}, \ldots, v_N^{(n)}, \nu_v^{(n)}\right]^T.$$

Alors on obtient la formule suivante pour l'erreur au point n :

$$\begin{aligned}
e_n &= \left|\hat{f}^{(n)}(x_n) - y_n\right| \\
&= \left|\sum_{m=1, m\neq n}^{N} w_m^{(n)} \phi(x_n, x_m) + \mu^{(n)} - y_n\right| \\
&= \left|\sum_{m=1; m\neq n}^{N} v_m^{(n)} \phi(x_n, x_m) + \nu_v^{(n)} - y_n\right| \\
&= \left|\sum_{m=1}^{N} v_m^{(n)} \phi(x_n, x_m) + \nu_v^{(n)} - y_n\right| \\
&= \left|(\Phi \begin{bmatrix} v^{(n)} \\ \nu_v^{(n)} \end{bmatrix})_n - y_n\right| \\
&= \left|\frac{w_n}{u_n^{(n)}}\right| \\
&= \left|\frac{(\Phi^{-1} y)_n}{(\Phi^{-1})_{nn}}\right|,
\end{aligned}$$

et, par définition de l'erreur *leave-one-out* :

$$e_{loo} = \frac{E^T E}{N} = \frac{Y^T \Phi^{-2} Y}{N diag(\Phi^{-2})}.$$

□

B.3 Réseau RBF avec gradients

On va démontrer ici que les formules données au paragraphe 2.4.5 pour le calcul de l'erreur *leave-one-out* d'un réseau RBF, s'appliquent et se généralisent dans le cadre d'un réseau RBF avec gradients. L'échantillon E de construction est ainsi enrichi, en chaque point, par l'ensemble des dérivées de la fonction à approximer.

Dans ce cas, on peut montrer que l'erreur *leave-one-out* telle que définie, pour les méthodes avec gradients, par l'équation (2.5.4.1), se calcule par les formules suivantes :

$$e_{iN+n} = \frac{\Phi^{-1} y}{\Phi^{-1}_{(iN+n)(iN+n)}}, \quad i \in \{0, d\}, \quad n \in \{1, N\}, \tag{B.9}$$

$$e_{loo} = \frac{E^T E}{(d+1)N} = \frac{Y^T \Phi^{-2} Y}{(d+1)N diag(\Phi^{-2})}. \tag{B.10}$$

Démonstration. On considère un réseau RBF avec gradients, construit sur un échantillon $E = \left\{(x_n, y_n, y_n^1, \ldots, y_n^d) \in \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d, \quad n \in \{1, N\}\right\}$. On note :

- Y le vecteur $[y_1, \ldots, y_N, y_1^1, \ldots y_N^1, \ldots, y_1^d, \ldots, y_N^d]$;
- w le vecteur $[a_1, \ldots, a_N, b_1^1, \ldots b_N^1, \ldots, b_1^d, \ldots, b_N^d]$;
- Φ la matrice, de taille $[(d+1)N \times (d+1)N]$ du système linéaire à résoudre pour le calcul des coefficients.

Soit $\hat{f}^{(iN+n)}(x)$ l'évaluation en un point x quelconque fournie par le modèle construit à partir de l'échantillon E diminué du point d'indice n si $i = 0$, ou de la dérivée i-ème en ce point sinon (on appellera l'échantillon diminué $E^{(iN+n)}$). Alors $\hat{f}^{(iN+n)}(x)$ vaut :

$$\begin{cases} \sum_{m=1;m\neq n}^{N} a_m^{(n)} \phi(x, x_m) + \sum_{m=1}^{N} \sum_{k=1}^{d} b_m^{k(n)} \phi_{0k}(x, x_m) & \text{si} \quad i = 0 \\ \sum_{m=1}^{N} a_m^{(n)} \phi_{i0}(x, x_m) + \sum_{m=1}^{N} \sum_{k=1;k\neq i}^{d} b_m^{k(n)} \phi_{ik}(x, x_m) + \sum_{m=1;m\neq n}^{N} b_m^{i(n)} \phi_{0i}(x, x_m) & \text{si} \quad i > 0 \end{cases}.$$

La propriété d'interpolation du réseau RBF avec gradients fournit le système linéaire suivant, de taille $[(N(d+1)-1) \times (N(d+1)-1)]$:

$$\Phi^{(iN+n)} w^{(iN+n)} = Y^{(iN+n)},$$

où $\Phi^{(iN+n)}$ est la matrice Φ diminué de ses $(iN+n)$-èmes lignes et colonnes et $w^{(iN+n)}$ et $Y^{(iN+n)}$ les vecteurs w et Y diminués de leurs $(iN+n)$-ème ligne.

Remarquons que si un vecteur quelconque $y \in \mathbb{R}^{N(d+1)}$ est tel que $y_{iN+n} = 0$, alors :

$$\Phi y = z \Rightarrow \Phi^{(iN+n)} y^{(iN+n)} = z^{(iN+n)}, \tag{B.11}$$

où les vecteurs $y^{(iN+n)}$ et $z^{(iN+n)}$ sont les vecteurs y et z diminués de leur $(iN+n)$-ème ligne.

Soit $u^{(iN+n)} \in \mathbb{R}^{N(d+1)}$ la solution du système :

$$\Phi u^{(iN+n)} = \mathbb{1}^{(iN+n)}, \tag{B.12}$$

où $\mathbb{1}^{(iN+n)} \in \mathbb{R}^{N(d+1)}$ est le vecteur dont seule la composante $(iN+n)$, de valeur 1, est non nulle.

On peut vérifier que $u_{iN+n}^{(iN+n)} \neq 0$. Sinon, par (B.11) :

$$u_{iN+n}^{(iN+n)} = 0 \Rightarrow \Phi^{(iN+n)} \left[u_1^{(iN+n)}, \ldots, u_{iN+n-1}^{(iN+n)}, u_{iN+n+1}^{(iN+n)}, \ldots, u_{(d+1)N}^{(iN+n)}\right]^T = \mathbf{0}.$$

Alors, comme Φ est régulière, $u^{(iN+n)} = \mathbf{0}$ ce qui n'est pas possible puisque $u^{(iN+n)}$ est solution de (B.12).

Comme $u_{iN+n}^{(iN+n)} \neq 0$, on peut alors considérer le vecteur suivant :

$$v^{(iN+n)} = w - \frac{w_{iN+n}}{u_{iN+n}^{(iN+n)}} u^{(iN+n)}.$$

Alors :

$$\Phi v^{(iN+n)} = \Phi w - \Phi \frac{w_{iN+n}}{u_{iN+n}^{(iN+n)}} u^{(iN+n)} = Y - \frac{w_{iN+n}}{u_{iN+n}^{(iN+n)}} \mathbb{1}^{(iN+n)}.$$

Comme $v_{iN+n}^{(iN+n)} = 0$, on peut utiliser (B.11) pour écrire :

$$w^{(iN+n)} = \left[v_1^{(iN+n)}, \ldots, v_{iN+n-1}^{(iN+n)}, v_{iN+n+1}^{(iN+n)}, \ldots, v_{(d+1)N}^{(iN+n)} \right]^T.$$

Alors on obtient la formule suivante pour l'erreur d'approximation au point n :

$$\begin{aligned}
e_n = \left| \hat{f}^{(n)}(x_n) - y_n \right| &= \left| \sum_{m=1, m\neq n}^{N} a_m^{(n)} \phi(x_n, x_m) + \sum_{m=1}^{N} \sum_{k=1}^{d} b_m^{k(n)} \phi_{0k}(x_n, x_m) - y_n \right| \\
&= \left| \sum_{m=1; m\neq n}^{N} v_m^{(n)} \phi(x_n, x_m) + \sum_{m=1}^{N} \sum_{k=1}^{d} v_{mN+k}^{(n)} \phi_{0k}(x_n, x_m) - y_n \right| \\
&= \left| \sum_{m=1}^{N} v_m^{(n)} \phi(x_n, x_m) + \sum_{m=1}^{N} \sum_{k=1}^{d} v_{mN+k}^{(n)} \phi_{0k}(x_n, x_m) - y_n \right| \\
&= \left| (\Phi v^{(n)})_n - y_n \right| \\
&= \left| \frac{w_n}{u_n^{(n)}} \right| \\
&= \left| \frac{(\Phi^{-1} y)_n}{\Phi_{nn}^{-1}} \right|,
\end{aligned}$$

ainsi que la formule suivante pour l'erreur d'approximation sur la dérivée i-ème au point n :

$$\begin{aligned}
e_{iN+n} &= \left| \frac{\partial \hat{f}^{(iN+n)}}{\partial x^i}(x_n) - y_n^i \right| \\
&= \left| \sum_{m=1}^{N} a_m^{(n)} \phi_{i0}(x_n, x_m) + \sum_{m=1}^{N} \sum_{k=1; k\neq i}^{d} b_m^{k(i)} \phi_{ik}(x_n, x_m) + \sum_{m=1}^{N} b_m^{k(i)} \phi_{ii}(x_n, x_m) - y_n^i \right| \\
&= \left| \sum_{m=1}^{N} v_{iN+m}^{(iN+n)} \phi_{i0}(x_n, x_m) + \sum_{m=1}^{N} \left[\sum_{k=1; k\neq i}^{d} v_{kN+m}^{(iN+n)} \phi_{ik}(x_n, x_m) + v_{iN+m}^{(iN+n)} \phi_{ii}(x_n, x_m) \right] - y_n^i \right| \\
&= \left| \sum_{m=1}^{N} v_{iN+m}^{(iN+n)} \phi_{i0}(x_n, x_m) + \sum_{m=1}^{N} \sum_{k=1}^{d} v_{kN+m}^{(iN+n)} \phi_{ik}(x_n, x_m) - y_n^i \right| \\
&= \left| (\Phi v^{(n)})_{iN+n} - y_n^i \right| \\
&= \left| \frac{w_{iN+n}}{u_{iN+n}^{(iN+n)}} \right| \\
&= \left| \frac{(\Phi^{-1} y)_{iN+n}}{(\Phi^{-1})_{(iN+n)(iN+n)}} \right|,
\end{aligned}$$

et, par définition de l'erreur *leave-one-out* :

$$e_{loo} = \frac{E^T E}{N(d+1)} = \frac{Y^T \Phi^{-2} Y}{N(d+1) diag(\Phi^{-2})}.$$

□

B.4 Co-krigeage

On souhaite démontrer ici que les formules données au paragraphe 2.4.5 pour le calcul de l'erreur *leave-one-out* d'un réseau RBF, s'appliquent et se généralisent dans le cadre de la méthode de co-krigeage où les données secondaires correspondent aux gradients de la fonction d'intérêt. L'échantillon E de construction est ainsi enrichi, en chaque point, par l'ensemble des dérivées de la fonction à approximer.

Dans ce cas là, on peut montrer que l'erreur *leave-one-out* telle que défini, pour les méthodes avec gradients, par l'équation (2.5.4.1), peut être approchée par les formules suivantes :

$$e_{iN+n} = \frac{\Phi^{-1} y}{\Phi^{-1}_{(iN+n)(iN+n)}}, \quad i \in \{0, d\}, \quad n \in \{1, N\}, \tag{B.13}$$

$$e_{loo} = \frac{E^T E}{(d+1)N} = \frac{Y^T \Phi^{-2} Y}{(d+1) N diag(\Phi^{-2})}. \tag{B.14}$$

Démonstration. On considère un modèle construit par la méthode de co-krigeage sur un échantillon $E = \left\{(x_n, y_n, y_n^1, \ldots, y_n^d) \in \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d, \quad n \in \{1, N\}\right\}$. On note :

- Y le vecteur $[y_1, \ldots, y_N, y_1^1, \ldots y_N^1, \ldots, y_1^d, \ldots, y_N^d]$;
- w le vecteur $[a_1, \ldots, a_N, b_1^1, \ldots b_N^1, \ldots, b_1^d, \ldots, b_N^d]$;
- $\hat{\Phi}$ la matrice, de taille $[(d+1)N \times (d+1)N]$, correspondante à la matrice Φ privée de sa dernière ligne et de sa dernière colonne.

Soit $\hat{f}^{(iN+n)}(x)$ l'évaluation en un point x quelconque fournie par le modèle construit à partir de l'échantillon E diminué du point d'indice n si $i = 0$, ou de la dérivée i-ème en ce point sinon (on appellera l'échantillon diminué $E^{(iN+n)}$). Alors $\hat{f}^{(iN+n)}(x)$ vaut :

$$\begin{cases} \sum_{m=1;m\neq n}^{N} a_m^{(n)} \phi(x, x_m) + \sum_{m=1}^{N} \sum_{k=1}^{d} b_m^{k(n)} \phi_{0k}(x, x_m) + \mu^{(iN+n)} & \text{si} \quad i = 0 \\ \sum_{m=1}^{N} a_m^{(n)} \phi_{i0}(x, x_m) + \sum_{m=1}^{N} \sum_{k=1;k\neq i}^{d} b_m^{k(n)} \phi_{ik}(x, x_m) + \sum_{m=1;m\neq n}^{N} b_m^{i(n)} \phi_{0i}(x, x_m) & \text{si} \quad i > 0 \end{cases}.$$

La propriété d'interpolation du co-krigeage fournit le système linéaire suivant, de taille $[N(d+1)]$:

$$\underbrace{\begin{bmatrix} \hat{\Phi}^{(iN+n)} & \gamma \\ \gamma^T & 0 \end{bmatrix}}_{\Phi^{(iN+n)}} \begin{bmatrix} w^{(iN+n)} \\ \mu^{(n)} \end{bmatrix} = \begin{bmatrix} Y^{(iN+n)} \\ 0 \end{bmatrix},$$

où $\hat{\Phi}^{(iN+n)}$ est la matrice $\hat{\Phi}$ diminué de ses $(iN+n)$-èmes lignes et colonnes, $w^{(iN+n)}$ et $Y^{(iN+n)}$ sont les vecteurs w et Y diminués de leurs $(iN+n)$-ème ligne et λ représente un vecteur avec les N premières composantes fixées à 1, et les Nd autres à 0.

Remarquons que si un vecteur quelconque $y \in \mathbb{R}^{N(d+1)+1}$ est tel que $y_{iN+n} = 0$, alors :

$$\Phi y = z \Rightarrow \Phi^{(iN+n)} y^{(iN+n)} = z^{(iN+n)}, \tag{B.15}$$

où les vecteurs $y^{(iN+n)}$ et $z^{(iN+n)}$ sont les vecteurs y et z diminués de leur $(iN+n)$-ème ligne.

Soit $\left[u^{(iN+n)}, \nu_u^{(iN+n)}\right]^T \in \mathbb{R}^{N(d+1)+1}$ la solution du système :

$$\Phi \begin{bmatrix} u^{(iN+n)} \\ \nu_u^{(iN+n)} \end{bmatrix} = \begin{bmatrix} \mathbb{1}^{(iN+n)} \\ 0 \end{bmatrix}, \tag{B.16}$$

où $\mathbb{1}^{(iN+n)} \in \mathbb{R}^{N(d+1)}$ est le vecteur dont seule la composante $(iN+n)$, de valeur 1, est non nulle.

On peut vérifier que $u_{iN+n}^{(iN+n)} \neq 0$: sinon, par (B.15) :

$$u_{iN+n}^{(iN+n)} = 0 \Rightarrow \Phi^{(iN+n)} \left[u_1^{(iN+n)}, \ldots, u_{iN+n-1}^{(iN+n)}, u_{iN+n+1}^{(iN+n)}, \ldots, u_{(d+1)N}^{(iN+n)}, \nu_u^{(iN+n)}\right]^T = \mathbf{0}.$$

Comme Φ est régulière, $\left[u^{(iN+n)}, \nu_u^{(iN+n)}\right] = \mathbf{0}$ ce qui n'est pas possible puisque $\left[u^{(iN+n)}, \nu_u^{(iN+n)}\right]^T$ est solution de (B.16).

Comme $u_{iN+n}^{(iN+n)} \neq 0$, on peut alors considérer le vecteur suivant :

$$\begin{bmatrix} v^{(iN+n)} \\ \nu_v^{(iN+n)} \end{bmatrix} = \begin{bmatrix} w \\ \mu \end{bmatrix} - \frac{w_{iN+n}}{u_{iN+n}^{(iN+n)}} \begin{bmatrix} u^{(iN+n)} \\ \nu_u^{(iN+n)} \end{bmatrix}.$$

Alors :

$$\Phi \begin{bmatrix} v^{(iN+n)} \\ \nu_v^{(iN+n)} \end{bmatrix} = \Phi \begin{bmatrix} w \\ \mu \end{bmatrix} - \Phi \frac{w_{iN+n}}{u_{iN+n}^{(iN+n)}} \begin{bmatrix} u^{(iN+n)} \\ \nu_u^{(iN+n)} \end{bmatrix} = \begin{bmatrix} Y \\ 0 \end{bmatrix} - \frac{w_{iN+n}}{u_{iN+n}^{(iN+n)}} \begin{bmatrix} \mathbb{1}^{(iN+n)} \\ 0 \end{bmatrix}.$$

Comme $v_{iN+n}^{(iN+n)} = 0$, on peut utiliser (B.15) pour écrire :

$$\left[w^{(iN+n)} \mu^{(iN+n)}\right]^T = \left[v_1^{(iN+n)}, \ldots, v_{iN+n-1}^{(iN+n)}, v_{iN+n+1}^{(iN+n)}, \ldots, v_{(d+1)N}^{(iN+n)}, \nu_v^{(iN+n)}\right]^T.$$

Alors on obtient la formule suivante pour l'erreur d'approximation au point n :

$$\begin{aligned}
e_n &= \left|\hat{f}^{(n)}(x_n) - y_n\right| \\
&= \left|\sum_{m=1,m\neq n}^{N} a_m^{(n)}\phi(x_n,x_m) + \sum_{m=1}^{N}\sum_{k=1}^{d} b_m^{k(n)}\phi_{0k}(x_n,x_m) + \mu^{(n)} - y_n\right| \\
&= \left|\sum_{m=1;m\neq n}^{N} v_m^{(n)}\phi(x_n,x_m) + \sum_{m=1}^{N}\sum_{k=1}^{d} v_{mN+k}^{(n)}\phi_{0k}(x_n,x_m) + \nu_v^{(n)} - y_n\right| \\
&= \left|\sum_{m=1}^{N} v_m^{(n)}\phi(x_n,x_m) + \sum_{m=1}^{N}\sum_{k=1}^{d} v_{mN+k}^{(n)}\phi_{0k}(x_n,x_m) + \nu_v^{(n)} - y_n\right| \\
&= \left|(\Phi\begin{bmatrix} v^{(n)} \\ \nu_v^{(n)}\end{bmatrix})_n - y_n\right| \\
&= \left|\frac{w_n}{u_n^{(n)}}\right| \\
&= \left|\frac{(\Phi^{-1}y)_n}{\Phi_{nn}^{-1}}\right|,
\end{aligned}$$

ainsi que la formule suivante pour l'erreur d'approximation sur la dérivée i-ème au point n :

$$\begin{aligned}
e_{iN+n} &= \left|\frac{\partial\hat{f}^{(iN+n)}}{\partial x^i}(x_n) - y_n^i\right| \\
&= \left|\sum_{m=1}^{N} a_m^{(n)}\phi_{i0}(x_n,x_m) + \sum_{m=1}^{N}\sum_{k=1;k\neq i}^{d} b_m^{k(i)}\phi_{ik}(x_n,x_m) + \sum_{m=1}^{N} b_m^{k(i)}\phi_{ii}(x_n,x_m) - y_n^i\right| \\
&= \left|\sum_{m=1}^{N} v_{iN+m}^{(iN+n)}\phi_{i0}(x_n,x_m) + \sum_{m=1}^{N}\left[\sum_{k=1;k\neq i}^{d} v_{kN+m}^{k(iN+n)}\phi_{ik}(x_n,x_m) + v_{iN+m}^{k(iN+n)}\phi_{ii}(x_n,x_m)\right] - y_n^i\right| \\
&= \left|\sum_{m=1}^{N} v_{iN+m}^{(iN+n)}\phi_{i0}(x_n,x_m) + \sum_{m=1}^{N}\sum_{k=1}^{d} v_{kN+m}^{k(iN+n)}\phi_{ik}(x_n,x_m) - y_n^i\right| \\
&= \left|(\Phi\begin{bmatrix} v^{(iN+n)} \\ \mu^{(iN+n)}\end{bmatrix})_{iN+n} - y_n^i\right| \\
&= \left|\frac{w_{iN+n}}{u_{iN+n}^{(iN+n)}}\right| \\
&= \left|\frac{(\Phi^{-1}y)_{iN+n}}{(\Phi^{-1})_{(iN+n)(iN+n)}}\right|,
\end{aligned}$$

et, par définition de l'erreur *leave-one-out* :

$$e_{loo} = \frac{E^TE}{N(d+1)} = \frac{Y^T\Phi^{-2}Y}{N(d+1)diag(\Phi^{-2})}.$$

□

Bibliographie

[AB97] W. K. Anderson and D. L. Bonhaus. Aerodynamic design on unstructured grids for turbulent flows. Technical Report TM 112867, NASA, 1997.

[AB99] W. K. Anderson and D. L. Bonhaus. Airfoil design optimization on unstructured grids for turbulent flows. *AIAA Journal*, 37(2) :185–191, 1999.

[ABB+99] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphie, third edition, 1999.

[ABR64] M. A. Aizerman, F. M. Braverman, and L. Rozoner. The probability problem in pattern recognition learning and the method of potential functions. *Automation and Remote Control*, 25(9), 1964.

[ADM83] P. V. Aidala, W. H. Davis, and W. H. Mason. Smart aerodynamic optimization. In *Proceedings of AIAA Applied Aerodynamics Conference, Danvers, Etats-Unis*, number 1863, Juillet 1983.

[Aki70] H. Akima. A new method of interpolation and smooth curve fitting based on local procedures. *Journal of the Association for Computing Machinery*, 17(4) :589–602, 1970.

[ALG+01] N. M. Alexandrov, R. M. Lewis, C. R. Gumbert, L. L. Green, and P. A. Newman. Approximation and model management in aerodynamic optimization with variable-fidelity models. *Journal of Aircraft*, 38(6) :1093–1101, 2001.

[Ant05] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. Society for Industrial and Applied Mathematics, Philadelphie, 2005.

[AV97] W. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. In *Proceedings of 35th Aerospace Sciences Meeting and Exhibit, Reno, Etats-Unis*, number 0643, Janvier 1997.

[AWI02] Aircraft wing with advanced technology operation (awiator). `http://www.transport-research.info/web/projects/project_details.cfm?id=14385`, 2002.

[Bar73] R. L. Barger. Streamline curvature design procedure for subsonic and transonic ducts. Technical Report TN D-7368, NASA, 1973.

[Bat89] J. T. Batina. Unsteady euler airfoil solutions using unstructured dynamic meshes. In *Proceedings of 27th AIAA Aerospace Sciences Meeting, Reno, Etats-Unis*, number 0150, Janvier 1989.

[BB96] G. W. Burgreen and O. Baysal. Three-dimensional aerodynamic shape optimization using discrete sensitivity analysis. *AIAA Journal*, 34 :1761–1770, 1996.

[BBMS02] R. R. Barton, A. J. Booker, M. Meckesheimer, and T. W. Simpson. Computanionally inexpensive metamodel assessment strategies. *AIAA Journal*, 40(10) :2053–2060, 2002.

[BD59] G. E. P. Box and N. R. Draper. A basis for the selection of a response surface design. *Journal of the American Statistical Association*, 54 :622–654, 1959.

[BH67] G. E. P. Box and W. J. Hill. Discrimination among mechanistic models. *Technometrics*, 9(1), 1967.

[BL88] D. Broomhead and D. Lowe. Multivariable functional interpolation and adaptative networks. *Complex Systems*, 2 :321–355, 1988.

[BP10] M. Bompard and J. Peter. Local search methods for design in aeronautics. In *Strategies for Optimization and Automated Design*, pages 1–46. Janvier 2010.

[Bre73] R. P. Brent. *Algorithms for Minimisation without Derivatives*. Prentice-Hall, 1973.

[BSK04] D. Büche, N. N. Schraudolph, and P. Koumoutsakos. Accelerating evolutionary algorithms with gaussian process fitness function models. *IEEE Transactions on Systems, Man and Cybernetics*, 35 :183–194, 2004.

[BW51] G. E. P. Box and K. B. Wilson. On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society*, 13(1) :1–45, 1951.

[BW76] R. R. Beam and R. F. Warming. An implicit factored finite difference algorithm for hyperbolic systems in conservation law form. *Journal of Computational Physics*, 22 :87–109, 1976.

[CA02a] H. S. Chung and J. J. Alonso. Design of a low-boom supersonic business jet using cokriging approximation models. In *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Etats-Unis*, number 5598, Septembre 2002.

[CA02b] H. S. Chung and J. J. Alonso. Using gradients to construct cokriging approximation models for high-dimensional design optimization problems. In *Proceedings of the 40th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Etats-Unis*, number 0317, Janvier 2002.

[Cam90] L. Cambier. Quelques notes sur la modélisation de la turbulence pour les équations de navier-stokes compressibles, 1990. Cours de DEA Mécanique Théorique.

[Can02] B. Cantaloube. Technique de calcul en maillage déformable (ale) pour les écoulements instationnaires. Technical Report RTS 71/1285 DSNSA, Onera, 2002.

[CCST99] N. Cristianini, C. Campbell, and J. Shawe-Taylor. Dynamically adapting kernels in support vector machines. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 204–210. MIT Press, Cambridge, 1999.

[CH86] G. B. Cosentino and T. L. Holst. Numerical optimization design of advanced transonic wing configurations. *NASA Technical Memorandum*, 23 :192–199, 1986.

[CI01] S. Cohen and N. Intrator. Global optimization of rbf networks, 2001.

[CJ97] D. D. Cox and S. John. *Multidisciplinary Design Optimization : State of the Art*, chapter SDO : A statistical method for global optimization, pages 315–329. Society for Industrial and Applied Mathematics, Philadelphie, 1997.

[CL05] M.-W. Chang and C.-J. Lin. Leave-one-out bounds for support vector regression model selection. *Neural Computation*, 17(5) :1188–1222, 2005.

[CM02] V. Cherkassky and Y. Ma. Selection of meta-parameters for support vector regression. In *Proceedings of the International Conference on Artificial Neural Networks, Londres, Angleterre*, pages 687–693. Springer Verlag, New York, Août 2002.

[CMF79] P. H. Cook, M. A. McDonald, and M. C. P. Firmin. Aerofoil rae 2822 - pressure distributions, and boundary layer and wake measurements. In *Experimental Data Base for Computer Program Assessment AR 138*. 1979.

[CMMY91] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker. Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, pages 953–963, 1991.

[CN07] P. Castonguay and S. K. Nadarajah. Effect of shape parameterization on aerodynamic shape optimization. In *Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Etats-Unis*, Janvier 2007.

[Cor00] M. Cormery. *De l'Optimisation Aérodynamique vers l'Optimisation Multidisciplinaire dans un Contexte Industriel*. PhD thesis, Université Paul Sabatier, Toulouse, 2000.

[CV08] L. Cambier and J.-P. Veuillot. Status of the elsa cfd software for flow simulation and multidisciplinary applications. In *Proceedings of the 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada*, Janvier 2008.

[CVBM02] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3) :131–159, 2002.

[CZ11] O. Chernukhin and D. W. Zingg. An investigation of multi-modality in aerodynamic shape optimization. 2011.

[DBVDSB07] A. De Boer, M. S. Van Der Schoot, and H. Bijl. Mesh deformation based on radial basis function interpolation. *Computers Structures*, 85 :784–795, 2007.

[Des03] D. Destarac. Far field/near field drag balance and applications of drag extraction in cfd. In *VKI Lecture Series, CFD-based Aircraft Drag Prediction and Reduction*. 2003.

[DHK08] D. Dajnarayan, A. Haas, and I. Kroo. A multifidelity, gradient-free optimization method and application to aerodynamic design. In *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, Etats-Unis*, number 6020, Septembre 2008.

[Die80] P. Dierckx. An algorithm for surface-fitting with spline functions. *Journal of Numerical Analysis*, 1 :267–283, 1980.

[Die93] P. Dierckx. Fitpack. `http://www.netlib.org/dierckx/`, 1993.

[DR84] P. J. Davis and P. Rabinowitz. *Methods of numerical integration*. Academic Press, 1984.

[DR93] D. Destarac and J. Reneaux. Application de l'optimisation numérique à l'aérodynamique des avions de transport. *La Recherche Aérospatiale*, (2), 1993.

[Dru04] F. Drullion. *Définition et étude de systèmes linéaires pour la simulation d'écoulements et l'optimisation de formes aérodynamiques par méthode de gradient. De l'Optimisation Aérodynamique vers l'Optimisation Multidisciplinaire dans un Contexte Industriel.* PhD thesis, Université Bordeaux 1, Bordeaux, 2004.

[DT95] J. E. Dennis and V. Torczon. Managing approximation models in optimization. In *Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization, Hampton, Etats-Unis*, pages 330–347, Mars 1995.

[Duc76] J. Duchon. *Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces*, volume 571, pages 85–100. Schempp & Zeller, 1976.

[Dug98] A. Dugeai. Méthode de mouvement de maillage pour l'aéroelasticité. Technical Report RTS 101/3064 DDSS, Onera, 1998.

[Duv07] R. Duvigneau. Aerodynamic shape optimization with uncertain operating conditions using metamodels. Technical Report RR 6143, INRIA, 2007.

[Duv11] R. Duvigneau. Coupling local and global shape optimization in aerodynamic design. Technical Report 7684, INRIA, 2011.

[DV01] R. Duvigneau and M. Visonneau. Shape optimization of incompressible and turbulent flows using the simplex method. In *Proceedings of 19th AIAA Applied Aerodynamics Conference, Anaheim, Etats-Unis*, number 2533, Juin 2001.

[DZ07] J. Driver and D. W. Zingg. Numerical aerodynamic optimization incorporating laminar-turbulent transition prediction. *AIAA Journal*, 45 :1810–1818, 2007.

[EBK98] M. A. El-Beltagy and A. J. Keane. Optimization for multilevel problems : A comparison of various algorithms. In *Proceedings of 3rd International Conference on Adaptive Computing in Design and Manufacture, Devon, Angleterre*, pages 111–120. Springer Verlag, New York, Avril 1998.

[EMVGB06] S. Esquieu, M. Meheut, H. F. Von Geyr, and O. Brodersen. Far-field drag extraction from numerical solutions and wake surveys. In *Proceedings of the 7th ONERA-DLR Aerospace Symposium, Toulouse, France*, Octobre 2006.

[FBK06] A. I. J. Forrester, N. W. Bressloff, and A. J. Keane. Optimization using surrogate models and partially converged computational fluid dynamics simulations. *Proceedings of the Royal Society*, 462(2071) :2177–2204, 2006.

[FJW+01] I. Fejtek, D. Jones, G. Waller, E. Hansen, and S. Obayashi. A transonic wing inverse design capability for complete aircraft configurations. In *Proceedings of 19th AIAA Applied Aerodynamics Conference, Anaheim, Etats-Unis*, number 2443, Juin 2001.

[FK09] A. Forrester and A. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45 :50–79, Janvier 2009.

[FLWZ00] K.-T. Fang, S. K. J. Lin, P. Winker, and Y. Zhang. Uniform design : Theory and application. *Technometrics*, 42(3) :237–248, 2000.

[FR64] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *British Computer Journal*, 7(2), 1964.

[Fri91] J. H. Friedman. Multivariate adaptive regression splines. *Annals of Statistics*, 19(1) :1–67, 1991.

[Gan63] L. S. Gandin. *Objective Analysis of Meteorological Fields.* GIMIZ, 1963.

[GD97] M. B. Giles and M. Drela. Two-dimensional transonic aerodynamic design method. *AIAA Journal*, 25 :1199–1206, 1997.

[GDM01] M. B. Giles, M. C. Duta, and J. D. Muller. Adjoint code developpments using the exact discrete approach. In *Proceedings of the 15th AIAA Computational Fluid Dynamics Conference, Anaheim, Etats-Unis*, number 2596, Juin 2001.

[GGH02] J. B. Gao, S. R. Gunn, and C. J. Harris. A probalistic framework for svm regression and error bar estimation. *Machine Learning*, 46 :71–89, 2002.

[GHK+90] B. Grossman, R. Haftka, P. Kao, D. Polen, M. Rais-Rohani, and J. Sobieszczanski-Sobieski. Integrated aerodynamic-structural design of a transport wing. *Journal of Aircraft*, 27(12) :1050–1056, 1990.

[Gia02] K. C. Giannakoglou. Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Progress in Aerospace Sciences*, 38(1) :43–76, 2002.

[Gib97] M. N. Gibbs. *Bayesian Gaussian Processes for Regression and Classification.* PhD thesis, University of Cambridge, 1997.

[GMW81] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, Toronto, 1981.

[GP75] R. Glowinski and O. Pironneau. On the numerical computation of the minimum-drag profile in laminar flow. *Journal of Fluid Mechanics*, 72 :385–389, 1975.

[GP00] M. B. Giles and A. N. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65 :393–415, 2000.

[GPK06] K. C. Giannakoglou, D. I. Papadimitriou, and I. C. Kampolis. Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels. *Computer Methods in Applied Mechanics and Engineering*, 195(44-47) :6312–6329, 2006.

[GRMS05] S. E. Gano, J. E. Renaud, J. D. Martin, and T. W. Simpson. Update strategies for kriging models for use in variable fidelity optimization. In *Proceedings of 46th AIAA/ASME/ASCE/AHS/ASC Structures Structural Dynamics and Materials Conference, Austin, Etats-Unis*, Avril 2005.

[GSS98] A. Giunta and J. Sobieszczanski-Sobieski. Progress toward using sensitivity derivatives in a high-fidelity aeroelastic analysis of a supersonic transport. 1998.

[Gun98] S. R. Gunn. Support vector machines for classification and regression. Technical report, University of Southampton, Mai 1998.

[Gut01] H.-M. Gutmann. A radial basis function method for global optimization. *Journal of Global Optimization*, 19 :201–227, 2001.

[GWK] A. Giunta, L. T. Watson, and J. Koelher. A comparison of approximation modeling techniques : Polynomial versus interpolating models. In *Proceedings of 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis & Optimization, Saint-Louis, Etats-Unis*, volume 1, pages 392–404, Septembre.

[Haf85] R. T. Haftka. Sensitivity calculations for iteratively solved problems. *International Journal of Numerical Methods in Engineering*, 21 :1535–1546, 1985.

[Har71] R. L. Hardy. Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysic Ressources*, 2 :321–355, 1971.

[Hay99] S. Haykin. *Neural Networks : A Comprehensive Foundation*. Prentice Hall PTR, 2nd edition, 1999.

[HBE+93] W. Haase, F. Bradsma, E. Elsholz, M. Leschziner, and D. Schwamborn. *Notes on numerical fluid mechanics*, volume 42, chapter EUROVAL - an european initiative on validation of CFD codes. Vieweg Verlag, Wiesbaden, 1993.

[HDB11] C. Heinrich, R. Duvigneau, and L. Blanchard. Isogeometric shape optimization in fluid-structure interaction. Technical Report RR 7639, INRIA, 2011.

[HGZ09] Z. H. Han, S. Görtz, and R. Zimmermann. On improving efficiency and accuracy of variable-fidelity surrogate modeling in aero-data for loads context. In *Proceedings of CEAS 2009 European Air and Space Conference, Manchester, Angleterre*, Octobre 2009.

[HH78] R. M. Hicks and P. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15 :407–412, 1978.

[HH83] A. Harten and J. M. Hyman. Self-adjusting grid methods for one-dimensional hyperbolic conservation laws. *Journal of Computational Physics*, 50 :235–269, 1983.

[HMT+] G. J. Hou, V. Maroju, A. C. Taylor, V. M. Korivi, and P. A. Newman. Transonic turbulent airfoil design optimization with automatic differentiation in incremental iterative form. In *Proceedings of the 12th AIAA Computational Fluid Dynamics Conference, San Diego, Etats-Unis*, Juin.

[HMV74] R. M. Hicks, E. M. Murman, and G. N. Vanderplaats. An assessment of airfoil design by numerical optimization. Technical Report TM X-3092, NASA, 1974.

[Hol75] J. H. Holland. *Adaptation in Natural and Artifical Systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[HP01] T. L. Holst and T. H. Pulliam. Aerodynamic shape optimization using a real-number-encoded genetic algorithm. In *Proceedings of the 19th AIAA Applied Aerodynamics Conference, Anaheim, Etats-Unis*, number 2473, Juin 2001.

[HP04] L. Hascoët and V. Pascual. Tapenade 2.1 user's guide. Technical Report 0300, INRIA, 2004.

[HZ10] J. E. Hicken and D. W. Zingg. Aerodynamic optimization algorithm with integrated geometry parameterization and mesh movement. *AIAA Journal*, 48(2) :400–413, 2010.

[JA07] S. Jakobsson and O. Amolgnon. Mesh deformation using radial basis function for gradient-based aerodynamic shape optimization. *Computer and Fluids*, 36(6) :1119–1136, 2007.

[Jam88] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3 :233–260, 1988.

[Jam94] A. Jameson. Optimum aerodynamic design via boundary control. In *VKI Lecture Series, Optimum Design Methods in Aeronautics*. Von Karman Institute for Fluid Dynamics, 1994.

[JCS00] R. Jin, W. Chen, and T. W. Simpson. Comparative studies of metamodeling techniques under multiple modeling criteria. In *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, Etats-Unis*, number 4801, Septembre 2000.

[JHS08] V. R. Joseph, Y. Hung, and A. Sudjianto. Blind kriging : A new method for developing metamodels. *Journal of mechanical design*, 130(3), 2008.

[JKC06] Jayadeva, Reshma Khemchandani, and Suresh Chandra. Regularized least squares twin svr for the simultaneous learning of a function and its derivative. In *Proceedings of International Joint Conference on Neural Networks, Vancouver, Canada*, pages 1192–1197, Juillet 2006.

[JMG89] O.-P. Jacquotte, F. Montigny, and G. Goussement. Generation, optimization, and adaptation of multiblock structured grids for complex configurations. *Survey on Mathematics for Industry*, 4 :267–277, 1989.

[JMY90] M. E. Johnson, L. M. Moore, and D. Ylvisaker. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26 :131–148, 1990.

[JMY05] S. Jeong, M. Murayama, and K. Yamamoto. Efficient optimization design method using kriging model. *Journal of Aircraft*, 42(2) :413–420, 2005.

[Jon01] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4) :345–383, 2001.

[JPM97] A. Jameson, N. A. Pierce, and L. Martinelli. Optimum aerodynamic design using the navier-stokes equations. In *Proceedings of the 35th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Etats-Unis*, number 0101, Janvier 1997.

[JR94] A. Jameson and J. Reuther. Control theory based airfoil design using euler equations. In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City Beach, Etats-Unis*, pages 206–222, Septembre 1994.

[JSL06] J.-C. Jouhaud, P. Sagaut, and B. Labeyrie. A kriging approach for cfd/wind-tunnel data comparison. *Journal of Fluids Engineering*, 128 :847–855, 2006.

[JSW98] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4) :455–492, 1998.

[JV01] A. Jameson and J. Vassberg. Computational fluid dynamics for aerodynamic design : It's current and future impact. In *Proceedings of the 39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Etats-Unis*, number 0538, Janvier 2001.

[KAJ00] S. K. Kim, J. J. Alonso, and A. Jameson. Two-dimensional high-lift aerodynamic optimization using the continuous adjoint method. In *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, Etats-Unis*, number 4741, Septembre 2000.

[Kar84] N. K. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4 :373–395, 1984.

[KD97] J. Kalagnanam and U. Diwekar. An efficient sampling technique for off-line quality control. *Technometrics*, 39(3) :308–319, 1997.

[KE95] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of the 1995 IEEE International Conference on Neural Networks, Perth, Australie*, volume 4, pages 1942–1948. IEEE Press, Novembre 1995.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4589) :671–680, 1983.

[KKR01] C. S. Kim, C. Kim, and O. H. Rho. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA Journal*, 39(5) :838–845, Mai 2001.

[KKR02] C. S. Kim, C. Kim, and O. H. Rho. Effects of constant eddy viscosity assumption on gradient-based design optimization. In *Proceedings of the 40th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Etats-Unis*, number 0262, Janvier 2002.

[KKRL99] C. S. Kim, C. Kim, O. H. Rho, and S. Lee. Aerodynamic sensitivity analysis for navier-stokes equations. In *Proceedings of the 37th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Etats-Unis*, number 0402, Janvier 1999.

[Kle09] J. P. C. Kleijnen. Kriging metamodeling in simulation : A rewiew. *European Journal of Operational Research*, 192(3) :707–716, 2009.

[KO96] J. R. Koehler and A. B. Owen. *Handbook of Statistics*, chapter Computer experiments, pages 261–308. 1996.

[Koe98] J. R. Koehler. Estimating the response, derivatives, and transmitted variance using computer experiments. *Computing science and statistics*, pages 25–32, 1998.

[Kri51] D. G. Krige. A statistical approach to some mine valuations and allied problems at the witwatersrand. Master's thesis, University of Witwatersrand, 1951.

[Kus64] H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86 :97–106, 1964.

[Kwo01] J. T. Kwok. Linear dependency between epsilon and the input noise in epsilon-support vector regression. In *Proceedings of the International Conference on Artificial Neural Networks, Vienne, Autriche*, pages 405–410. Springer Verlag, New York, Août 2001.

[LB00] W. Liu and S. M. Batill. Gradient-enhanced neural network response surface approximations. Number 4923, 2000.

[LB02] W. Liu and S. M. Batill. Gradient-enhanced response surface approximations using kriging models. In *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Etats-Unis*, number 5456, Septembre 2002.

[LBK04a] S. J. Leary, A. Bhaskar, and A. J. Keane. A derivative based surrogate model for approximating and optimizing the output of an expensive computer simulation. *Journal of Global Optimization*, 30 :39–58, 2004.

[LBK04b] S. J. Leary, A. Bhaskar, and A. J. Keane. Global approximation and optimization using adjoint computational fluid dynamics codes. *AIAA Journal*, 42(3) :631–641, 2004.

[LBT10] C. Le, T. Bruns, and D. Tortorelli. A gradient-based, parameter-free approach to shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 200(9-12) :985 – 996, 2010.

[LC85] Y. Le Cun. Une procédure d'apprentissage pour réseau à seuil asymétrique. In *Proceedings of Cognitiva 85, Paris, France*, pages 599–604, 1985.

[Lew98] R. M. Lewis. Using sensitivity information in the construction of kriging models for design optimization. In *Proceedings of 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis & Optimization, Saint-Louis, Etats-Unis*, number 4799, Juin 1998.

[LGTP01] J. Lépine, F. Guilbault, J.-Y. Trépanier, and F. Pépin. Optimized nonuniform rational b-spline geometrical representation for aerodynamic design of wings. *AIAA Journal*, 39 :2033–2041, 2001.

[Lie73] R. H. Liebeck. A class of airfoils designed for high lift in incompressible flow. *Journal of Aircraft*, 10 :610–617, 1973.

[Lig45] M. J. Lighthill. A new method of two-dimensional aerodynamic design. Technical Report R & M 2112, Aeronautical Research Council, 1945.

[Lio71] J.-L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer Verlag, New York, 1971.

[LK05] W. Li and S. Krist. Spline-based airfoil curvature smoothing and its applications. *Journal of Aircraft*, 42(4), 2005.

[LM68] P. A. Lachenbruch and M. R. Mickey. Estimation of error rates in discriminant analysis. *Technometrics*, 10(1) :1–11, 1968.

[LM08] J. Laurenceau and M. Meaux. Comparison of gradient and response surface based optimization frameworks using adjoint method. In *Proceedings of the 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Schaumburg, Etats-Unis*, number 1889, Avril 2008.

[LMJ02] E. Lund, H. Moller, and L. A. Jakobsen. Shape optimization of fluid-structure interaction problems using two-equation turbulence models. In *Proceedings of the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Denver, Etats-Unis*, number 1478, Avril 2002.

[LMMP59] J. Y. Lettvin, H. R. Maturana, W. S. McCulloch, and W. H. Pitts. What the frog's eye tells the frog's brain. In *Proceedings of the Institute of Radic Engineers*, volume 47, pages 1940–1959, 1959.

[LMQ02] A. Le Moigne and N. Qin. A discrete adjoint method for aerodynamic sensitivities for navier-stokes. In *Proceedings of CEAS Aerodynamics Research Conference, Cambridge, Etats-Unis*, Juin 2002.

[Loc97] M. Locatelli. Bayesian algorithms for one-dimensional global optimization. *Journal of Global Optimization*, 10 :57–76, 1997.

[LP04] W. Li and S. Padula. Using high resolution design spaces for aerodynamic shape optimization under uncertainty. Technical Report TP 2004-213004, NASA, 2004.

[LPZ01] H. Lomax, T. H. Pulliam, and D. W. Zingg. *Fundamentals of Computational Fluid Dynamics*. Springer Verlag, New York, 2001.

[LS08] J. Laurenceau and P. Sagaut. Building efficient response surfaces of aerodynamic functions with kriging and cokriging. *AIAA Journal*, 46(2) :498–507, 2008.

[LSPCAR05] M. Lazaro, I. Santamaría, F. Perez-Cruz, and A. Artés-Rodríguez. Support vector regression for the simultaneous learning of a multivariate function and its derivatives. *Neurocomputing*, 69(1-3) :42–61, 2005.

[LTP00] J. Lépine, J.-Y. Trépanier, and F. Pépin. Wing aerodynamic design using an optimized nurbs geometrical representation. 2000.

[Lut96] H. Lutkepohl. *Handbook of Matrices*. Wiley, New York, 1996.

[Man38] W. Mangler. Die berechnung eines tragflügelprofiles mit vorgeschriebener druckverteilung. *Jahrbuch der Deutschen Luftfahrtforschung*, 1 :46–53, 1938.

[Mar08] M. Marcelet. *Etude et mise en oeuvre d'une méthode d'optimisation de forme couplant simulation numérique en aérodynamique et en calcul de structure*. PhD thesis, Ecole Nationale Supérieure d'Arts et Métiers, 2008.

[Mat62] G. Matheron. *Traité de Géostatistique appliquée*. Paris : Editions Technip, 1962.

[Mat70] G. Matheron. *La théorie des variables régionalisées, et ses applications*. Centre de Morphologie Mathématique de Fontainebleau, 1970.

[MBC79] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2), 1979.

[McF79] G. B. McFadden. An artificial viscosity method for the design of supercritical airfoils. Technical Report C00-3077-158, New York University, 1979.

[MCV04] M. Meaux, M. Cormery, and G. Voizard. Viscous aerodynamic shape optimization based on the discrete adjoint state for 3d industrial configurations. In *Proceedings of ECCOMAS 2004, Jyväskylä, Finlande*, Juillet 2004.

[MD89] J. Moody and C. J. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1 :289–303, 1989.

[Meh92] S. Mehrotra. On the implementation of a primal–dual interior point method. *SIAM Journal on Optimization*, 2(4) :575–601, 1992.

[MH99] D. Mattera and S. Haykin. Support vector machines for dynamic reconstruction of a chaotic system. In *Advances in kernel methods : support vector learning*, pages 211–241. MIT Press, Cambridge, 1999.

[MM95] M. Morris and T. J. Mitchell. Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, 43(3) :381–402, 1995.

[MMY93] M. D. Morris, T. J. Mitchell, and D. Ylvisaker. Bayesian design and analysis of computer experiments : Use of derivatives in surface prediction. *Technometrics*, 35(3) :243–255, 1993.

[Moh97] B. Mohammadi. A new optimal shape design procedure for inviscid and viscous flows. *International Journal For Numerical Methods In Fluids*, 25, 1997.

[Mon97] D. C. Montgomery. *Design and Analysis of Experiments*. Wiley, New York, 1997.

[MP93] G. Mosetti and C. Poloni. Aerodynamic shape optimisation by means of a genetic algorithm. In *Proceedings of the 5th International Symposium of Computational Fluid Dynamics, Sendai, Japon*, pages 279–284, Septembre 1993.

[MP01] B. Mohammadi and O. Pironneau. *Applied Shape Design for Fluids*. Oxford Univ. Press, 2001.

[MS76] F. Murat and J. Simon. Etude de problèmes d'optimal design. *Lecture Notes in Computer Sciences*, 41 :54–62, 1976.

[MS83] B. A. Murtagh and M. A. Saunders. Minos 5.1 user's guide. Technical Report SOL 83-20R, Stanford University, 1983.

[NA98] E. J. Nielsen and W. K. Anderson. Aerodynamic design optimization on structured meshes using the navier-stokes equations. In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Saint-Louis, Etats-Unis*, number 4809, Septembre 1998.

[NA99] E. J. Nielsen and W. K. Anderson. Aerodynamic design optimization on unstructured meshes using the navier-stokes equations. *AIAA Journal*, 37(11) :1411–1419, 1999.

[NA02] E. J. Nielsen and W. K. Anderson. Recent improvments in aerodynamic design optimization on unstructured meshes. *AIAA Journal*, 40(6) :1155–1163, 2002.

[Nie92] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Siam, 1992.

[NM65] J. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7 :308–313, 1965.

[NP05] E. J. Nielsen and M. A. Park. Using an adjoint approach to eliminate mesh sensitivities. In *Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Etats-Unis*, number 0491, Janvier 2005.

[NZ01] N. Nemec and D. W. Zingg. Towards efficient aerodynamic shape optimization based on the navier-stokes equations. In *Proceedings of the 15th AIAA Computational Fluid Dynamics Conference, Anaheim, Etats-Unis*, number 2532, Juin 2001.

[NZP04] M. Nemec, D. W. Zingg, and T. H. Pulliam. Multipoint and multi-objective aerodynamic shape optimization. *AIAA Journal*, 42 :1057–1065, 2004.

[ONW04] Y. S. Ong, P. B. Nair, and K. W. Wong. *Knowledge Incorporation in Evolutionary Computation*, chapter Surrogate-Assisted Evolutionary Optimization Frameworks for High-Fidelity Engineering Design Problems, pages 307–332. Springer Verlag, New York, 2004.

[Orr98] M. Orr. Optimising the widths of radial basis functions. In *Proceedings of 5th Brazilian Symposium on Neural Networks, Belo Horizonte, Brésil*, pages 26–29, Décembre 1998.

[Orr99] M. Orr. An introduction to radial basis function networks. Technical report, Center for Cognitive Science, University of Edimbourg, Ecosse, Juin 1999.

[OT96] S. Obayashi and T. Tsukahara. Comparison of optimisation algorithms for aerodynamic shape design. In *Proceedings of 14th AIAA Applied Aerodynamics Conference, La Nouvelle Orléans, Etats-Unis*, number 2394, Juin 1996.

[Owe92] A. B. Owen. Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica*, 2 :439–452, 1992.

[Par94] J.-S. Park. Optimal latin-hypercube designs for computer experiments. *Journal of Statistical Planning and Inference*, 39(1) :95–111, 1994.

[PBL98] E. Perry, R. Balling, and M. Landon. A new morphing method for shape optimization. In *Proceedings of 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis & Optimization, Saint-Louis, Etats-Unis*, number 2896, Septembre 1998.

[PD07] C. Praveen and R. Duvigneau. Radial basis functions and kriging metamodels for aerodynamic optimization. Technical Report RR 6151, INRIA, 2007.

[PD09] C. Praveen and R. Duvigneau. Study of some strategies for global optimization using gaussian process models with application to aerodynamic design. Technical Report 6964, INRIA, 2009.

[PDP04] J. Peter, F. Drullion, and C.-T. Pham. Contribution to discrete implicit gradient and discrete adjoint method for aerodynamic shape optimization. In *Proceedings of ECCOMAS 2004, Jyväskylä, Finlande*, Juillet 2004.

[Pet96] J. Peter. *Etude de schémas numériques pour la résolution des équations de Navier–Stokes instationnaires. Application à des écoulements en turbomachines*. PhD thesis, Ecole Nationale Supérieure d'Arts et Métiers, Paris, 1996.

[Pha06] C.-T. Pham. *Linéarisation du flux visqueux des équations de Navier-Stokes et de modèles de turbulence pour l'optimisation aérodynamique en turbomachines*. PhD thesis, Arts et Métiers ParisTech, 2006.

[Pir73] O. Pironneau. On optimum design in stokes flow. *Journal of Fluid Mechanics*, 59 :117–128, 1973.

[Pir74] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64 :97–110, 1974.

[Pir84] O. Pironneau. *Optimal Shape Design for Elliptic Systems*. Springer Verlag, New York, 1984.

[Pla98] J. Platt. A fast algorithm for training support vector machines. Technical Report Technical Report MSR-TR-98-14, Microsoft Research, 1998.

[PMBP07] J. Peter, M. Marcelet, S. Burguburu, and V. Pediroda. Comparison of surrogate models for the actual global optimization of a 2d turbomachinery flow. In *Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization, Pekin, Chine*, Septembre 2007.

[Pow81] M. J. D. Powell. *Approximation Theory and Methods*. Cambridge University Press, 1981.

[PS91] J. Park and I. W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3 :246–257, 1991.

[QDC94] D. Quagliarella and A. Della Cioppa. Genetic algorithms applied to the aerodynamic design of transonic airfoils. In *Proceedings of the 12th AIAA Applied Aerodynamics Conference, Colorado Springs, Etats-Unis*, pages 686–693, Juin 1994.

[QHS+05] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1) :1–28, Janvier 2005.

[QL05] S. Qiu and T. Lane. Multiple kernel learning for support vector regression. Technical Report 42, Computer Science Departement, University of New Mexico, 2005.

[Ras91] R. Rashed. Al-samaw'al, al-biruni et brahmagupta : les méthodes d'interpolation. *Arabic Sciences and Philosophy*, 1 :101–160, 1991.

[Rat98] A. Ratle. Accelerating the convergence of evolutionary algorithms by fitness landscape approximation. *Parallel Problem Solving from Nature*, 1498 :87–96, 1998.

[RH00] K. Rasheed and H. Hirsh. Informed operators : Speeding up genetic-algorithm-based design optimization using reduced models. In *Proceedings of the Genetic and Evolutionary Computation Conference, Las Vegas, Etats-Unis*, pages 628–635. Morgan Kaufmann, San Francisco, Juillet 2000.

[Rip99] S. Rippa. An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Advances in Computational Mathematics*, 11(2-3) :193–210, 1999.

[RJA+99] J. Reuther, A. Jameson, J. Alonso, M. Rimlinger, and D. Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers. *Journal of Aircraft*, 36 :51–74, 1999.

[RJF+96] J. Reuther, A. Jameson, J. Farmer, L. Martinelli, and D. Saunders. Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. In *Proceedings of the 34th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Etats-Unis*, number 0094, Janvier 1996.

[RK98] G. M. Robinson and A. J. Keane. A case for multi-level optimisation in aeronautical design. In *Proceedings of the RAeS Conference on Multidisciplinary Design and Optimisation*, pages 9.1–9.6, 1998.

[RMC86] D. E. Rumelhart and J. L. Mc Clelland. *Parallel Distributed Processing : Exploration in the MicroStructure of Cognition*. MIT Press, 1986.

[Roe81] P. L. Roe. Approximate riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics*, 43 :357–372, 1981.

[RS06] E. Rafajlowicz and R. Schwabe. Halton and hammersley sequences in multivariate nonparametric regression. *Statistics & Probability Letters*, 76(8) :803–812, 2006.

[RT85] J. Reneaux and J.-J. Thibert. The use of numerical optimization for airfoil design. In *Proceedings of 3rd AIAA Applied Aerodynamics Conference, New York, Etats-Unis*, number 5026, Octobre 1985.

[SA92] P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. 1992.

[Sam01] J. A. Samareh. Novel multidisciplinary shape parameterization approach. *Journal of Aircraft*, 38 :1015–1024, 2001.

[Sas02] M. J. Sasena. *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*. PhD thesis, University of Michigan, 2002.

[SC90] L. A. Smith and R. L. Campbell. A method for the design of transonic flexible wings. Technical Report TP 3045, NASA, 1990.

[Sch87] A. J. Schoofs. *Experimental design and structural optimization*. Technische Universiteit Eindhoven, 1987.

[Sch97] M. Schonlau. *Computer Experiments and Global Optimization*. PhD thesis, University of Waterloo, 1997.

[SF91] G. R. Shubin and P. D. Frank. A comparison of implicit gradient approach and the variational approach to aerodynamic design optimization. Technical Report AMS-TR-163, Boeing computer service, Juin 1991.

[Shu91] G. R. Shubin. Obtaining "cheap" optimization gradients from computational aerodynamics codes. Technical Report AMS-TR-164, Boeing computer service, Juin 1991.

[SIGS08] S. Schmidt, C. Ilic, N.R. Gauger, and V. Schulz. Shape gradients and their smoothness for practical aerodynamic design optimization. Technical Report SPP1253, DFG, 2008.

[SJ07] A. Shahrokhi and A. Jahangirian. Airfoil shape parameterization for optimum navier-stokes design with genetic algorithm. *Aerospace Science and Technology*, 11 :443–450, 2007.

[SKMM01] T. W. Simpson, J. J. Korte, T. M. Mauery, and F. Mistree. Kriging models for global approximation in simulation-based multidisciplinary design optimization. *AIAA Journal*, 39(12) :2233–2241, 2001.

[SLC01] T. Simpson, D. Lin, and W. Chen. Sampling strategies for computer experiments : design and analysis. *International Journal of Reliability and Safety (IJRS)*, 2(3) :209–240, 2001.

[SLK05] A. Sóbester, S. J. Leary, and A. J. Keane. On the design of optimization strategies based on global response surface approximation models. *Journal of Global Optimization*, 33(1) :31–59, 2005.

[SMKM98] T. W. Simpson, T. M. Mauery, J. J. Korte, and F. Mistree. Comparison of response surface and kriging models for multidisciplinary design. In *Proceedings of 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis & Optimization, Saint-Louis, Etats-Unis*, volume 1, pages 381–391, Septembre 1998.

[SMSKR98] A. J. Smola, N. Murata, B. Schölkopf, and Müller K.-R. Asymptotically optimal choice of epsilon-loss for support vector machines. In *Proceedings of the 8th International Conference on Artificial Neural Networks, Skövde, Suède*, pages 105–110. Springer Verlag, New York, Septembre 1998.

[Sob98] H. Sobieczky. Parametric airfoils and wings. *Notes on Numerical Fluid Mechanics*, 68 :71–88, 1998.

[Soe97] B. I. Soemarwoto. The variational method for aerodynamic optimization using the navier-stokes equations. Technical Report ICASE Report 97-71, NASA Langley, Décembre 1997.

[SP86] T. W. Sederberg and S. R. Parry. Free-form deformation of polygonal data. In *Proceedings of the International Electronic Image Week, Nice, France*, pages 633–639, Avril 1986.

[SP95] R. Storm and K. Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute, 1995.

[SR11] A. Stück and T. Rung. Adjoint-rans with filtered shape derivatives for hydrodynamic optimisation. *Computers & Fluids*, 47 :22–32, 2011.

[SRB94] R. S Sellar, J. E. Renaud, and S. M. Batill. Optimization of mixed discrete/continuous design variable system using neural networks. In *Proceedings of the 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City, Etats-Unis*, number 4348, Septembre 1994.

[SS98] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. Technical Report 30, NeuroCOLT2, Octobre 1998.

[STBV08] T. W. Simpson, V. Toropov, V. Balabanov, and F. A. C. Viana. Design and analysis of computer experiments in multidisciplinary design optimization : A review of how far we have come–or not. In *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, Etats-Unis*, number 5802, Septembre 2008.

[STC04] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, 2004.

[Ste87] M. Stein. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2) :143–151, 1987.

[Ste90] N. L. Stein. A comparison of generalized cross validation and modified maximum likelihood for estimating the parameters of a stochastic process. *The Annals of Statistics*, 18(3) :1139–1157, 1990.

[SWMW89] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. *Statistical science*, 4(4) :409–423, 1989.

[TC00] D. Tse and L. Chan. Transonic airfoil design optimization using soft computing methods. *Canadian Aeronautics and Space Journal*, 46 :65–73, 2000.

[Val02] V. Valentin. *Optimisation Aérodynamique 3D des Aubages dans les Turbomachines Axiales Multi-Etages*. PhD thesis, Université Paris 6, Paris, 2002.

[Van73] G. N. Vanderplaats. Conmin - a fortran program for constrained function minimization - user's manual. Technical Report TM X-62 282, NASA, 1973.

[Van89] G. N. Vanderplaats. Effective use of numerical optimization in structural design. *Finite Elements in Analysis and Design*, 6 :97–112, 1989.

[Van94] R. J. Vanderbei. Loqo : An interior point code for quadratic progamming. Technical Report 15, Statistics and Operations Research, Princeton University, 1994.

[Van98] R. J. Vanderbei. Loqo : An interior point code for quadratic progamming (revised). Technical Report 15, Statistics and Operations Research, Princeton University, 1998.

[Vap95] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995.

[Vap98] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[VAVLR82] G. D. Van Albada, B. Van Leer, and W. W. Roberts. A comparative study of computational methods in cosmic gas dynamics. *Astronomy and Astrophysics*, 108 :76–84, 1982.

[VC74] V. Vapnik and A. Chervonenkis. *Theory of pattern recognition*. Nauka, Moscow, 1974.

[VC00] V. Vapnik and O. Chapelle. Bounds on error expectation for support vector machines. *Neural Computation*, 12(9) :2013–2036, 2000.

[VCP00] S. Varadarajan, W. Chen, and C. Pelka. The robust concept exploration method with enhanced model approximation capabilities. *Engineering Optimization*, 32(3) :309–334, 2000.

[VH96] M. Verleysen and K. Hlavácková. Learning in rbf networks. In *Proceedings of International Conference on Neural Networks, Washington, Etats-Unis*, pages 199–204, Juin 1996.

[VHRJ11] J. C. Vassberg, N. A. Harrisson, D. L. Roman, and A. Jameson. A systematic study on the impact of dimensionality for a two-dimensional aerodynamic optimization model problem. 2011.

[VL79] B. Van Leer. Towards the ultimate conservative difference scheme. v - a second order sequel to godunov's method. *Journal of Computational Physics*, 32 :101–136, 1979.

[VT04] J. Vesterstrom and R. Thomsen. A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. volume 2, pages 1980–1987. Proceedings of Congress on Evolutionary Computation 2004, Juin 2004.

[Wah85] G. Wahba. A comparison of gcv and gml for choosing the smoothing parameter in the generalized spline smoothing problem. *The Annals of Statistics*, 13(4) :1378–1402, 1985.

[WBIL10] M. Widhalm, J. Brezillon, C. Ilic, and T. Leicht. Investigation on adjoint based gradient computations for realistic 3d aero-optimization. In *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, Etats-Unis*, Septembre 2010.

[Whi23] E. Whittaker. On a new method of graduation. *Proceedings of the Edinburgh Mathematical Society*, 41 :63–75, 1923.

[Wil88] D. C. Wilcox. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA Journal*, 26 :1299–1310, 1988.

[Wil93] D. C. Wilcox. A two-equation turbulence model for wall-bounded and free-shear flows. 1993.

[WR87] J. J. Warnes and D. D. Ripley. Problems with likelihood estimation of covariance function of spatial gaussian processes. *Biometrika*, 74(3) :640–2, 1987.

[WTL09] C.-H. Wu, G.-H. Tzeng, and R.-H. Lin. A novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression. *Expert System with Applications*, 36(3) :4725–4735, 2009.

[WXLZ03] W. Wang, Z. Xu, W. Lu, and X. Zhang. Determination of the spread parameter in the gaussian kernel for classification and regression. *Neurocomputing*, 55(3) :643–663, Octobre 2003.

[YLL06] C.-C. Yang, W.-J. Lee, and S.-J. Lee. Learning of kernel functions in support vector machines. In *Proceedings of 2006 International Joint Conference on Neural Networks, Vancouver, Canada*, pages 1150–1155, Juillet 2006.

[YM07] Z. Yang and D. Mavriplis. A mesh deformation strategy optimized by the adjoint method on unstructured meshes. In *Proceedings of the 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Etats-Unis*, number 0557, Janvier 2007.

[YM11] W. Yamazaki and D. Mavriplis. Derivative-enhanced variable fidelity surrogate modeling for aerodynamic functions. In *Proceedings of the 49th AIAA Aerospace Sciences Meeting and Exhibit, Orlando, Etats-Unis*, number 1172, Janvier 2011.

[YMC96] W. Yamazaki, S. Mouton, and G. Carrier. Efficient design optimization by physics-based direct manipulation free-form deformation. 1996.

[YMJC11] G. Yu, J.-D. Müller, D. Jones, and F. Christakopoulos. Cad-based shape optimisation using adjoint sensitivities. *Computers and Fluids*, 46(1) :512–516, 2011.

[ZAG+01] T. Zervogiannis, V. Assouti, K. Gagas, A. Kaounis, and K. C. Giannakoglou. Inverse design of aerodynamic shapes using ant colony optimization. In *Proceedings of the 6th National Congress of Mechanics, Thessalonique, Grèce*, Juillet 2001.

[Zou60] G. Zoutendijk. *Methods of Feasible Directions*. Elsevier, 1960.