Université d'Orsay – Paris XI

Département de Mathématiques **École doctorale de Mathématiques de la région Sud de Paris**

Optimisation Différentiable en Mécanique des Fluides Numérique

THÈSE

présentée et soutenue publiquement le 26 Novembre 2003

pour l'obtention du

Doctorat de l'université Paris XI – Orsay

(spécialité E.D.P. et Calcul Scientifique)

par

Francois Courty

Composition du jury

Président :	François Alouges
Rapporteurs :	Olivier Pironneau Andreas Griewank
Examinateurs :	Mike Giles Frédéric Pascal Alain Dervieux

Institut National de la Recherche en Informatique et en Automatique — U.R. de Sophia-Antipolis

Mis en page avec la classe thloria.

Remerciements

Les travaux présentés dans ce mémoire de thèse ont été réalisés aux cours de ces trois années passées à l'INRIA Sophia-Antipolis dans le projet TROPICS dirigé par Laurent Hascoet.

Je tiens tout d'abord à exprimer toute ma gratitude envers Alain Dervieux qui a dirigé mes recherches tout au long de cette thèse. J'ai pu, à son contact, profiter de son expérience dans le domaine de la mécanique des fluides numérique ainsi que de son enthousiasme pour la recherche.

Je tiens également à associer à ces remerciements Laurent Hascoet qui m'a initialement accueilli dans le projet TROPICS et avec qui j'ai plus particulièrement collaboré sur les travaux qui ont abouti à la première partie de cette thèse.

Je remercie très vivement Andreas Griewank et Olivier Pironneau qui ont bien voulu consacrer leur temps à étudier cette thèse. Ma reconnaissance va également à François Alouges qui a bien voulu présider le jury de soutenance. Je n'oublie pas Frédéric Pascal, Mike Giles et Alain Dervieux pour leur participation au jury de thèse.

Je remercie aussi toutes les personnes de l'INRIA Sophia-Antipolis qui ont contribué à ce que mon travail se déroule dans les meilleures conditions possibles (équipe du projet TROPICS, centre de calcul, centre de documentation,..)

Enfin je remercie Valérie Pascual pour son idée d'avoir installé le nouveau thésard dans le bureau d'une ravissante thésarde qui s'appelait Antonia. On connait la suite ...

Nos prétendus raisonnements consistent à chercher des arguments pour continuer de croire ce que nous croyons déjà.
James Harvey Robinson

Table des matières

Introduction générale

0.1 Le projet de l'avion d'affaire supersonique

Un projet industriel comme l'avion d'affaire supersonique SSBJ de Dassault-Aviation est une entreprise prenant en compte un très grand nombre de contraintes, avec de nos jours un plus grand souci de maitriser, notamment par une réglementation stricte, l'impact du vol de l'aéronef sur l'environnement. Le prédécesseur Concorde ne pouvait en effet voler en régime supersonique qu'au-desssus de l'océan. Les performances du nouvel avion seront le produit de diverses innovations et améliorations, ainsi que d'une Conception Assistée par Ordinateur s'appuyant sur des phases d'Optimisation Assistée par Ordinateur.

Cette optimisation remplace progressivement la simple Simulation Aérodynamique, appelée aussi souvent "Soufflerie Numérique", qui permet d'analyser une forme aérodynamique, d'en prédire les performances, de comparer différentes formes et d'aider, par des visualisations comme celles de la figure 1, à comprendre pourquoi telle forme essayée est meilleure.

En Optimisation de forme sur ordinateur, le calcul doit donner des informations supplémentaires. Supposons que l'ingénieur ait muni son outil d'analyse d'un système d'évaluation, que l'on appellera le coût correspondant à la forme, prenant en compte le maximum de critères tels que, pour le SSBJ, la portance, la trainée, et la puissance du bang sonique. Alors le nouvel outil devra donner des informations sur la sensibilité du coût par rapport à la forme. Ces informations peuvent elles-aussi se présenter sous la forme de niveaux portés par la géométrie :

La figure 2 montre le **gradient** du coût par rapport à la forme. Ce gradient montre à l'ingénieur quelles déformations (enfoncement en rouge, enflement en bleu) il faut appliquer en priorité pour diminuer le coût d'une forme, c'est à dire l'améliorer.

La mise en place d'un outil d'optimisation de forme aérodynamique évolué comme celui mis au point par le projet Tropics à l'INRIA est une aventure **transdisciplinaire** faisant coopérer :

- les informaticiens de l'Environnement de Programmation, spécialistes de la Différentiation de Programme. Il s'agit du sujet principal de l'équipe Tropics dans laquelle cette thèse a été préparée. Tropics comporte un volet applicatif reposant sur l'application des techniques d'état adjoint à l'Aérodynamique.

- les spécialistes des Méthodes d'Optimisation Différentiable,

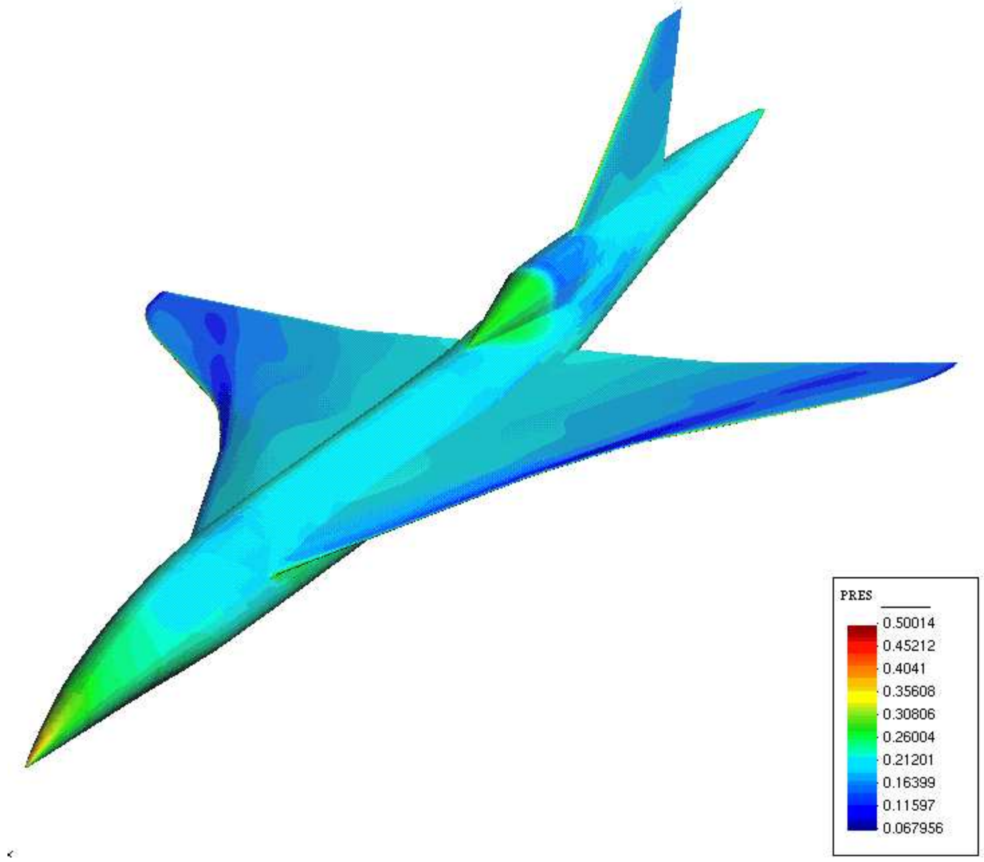

FIG. 1 – *Représentation de la pression sur un Falcon en régime supersonique*

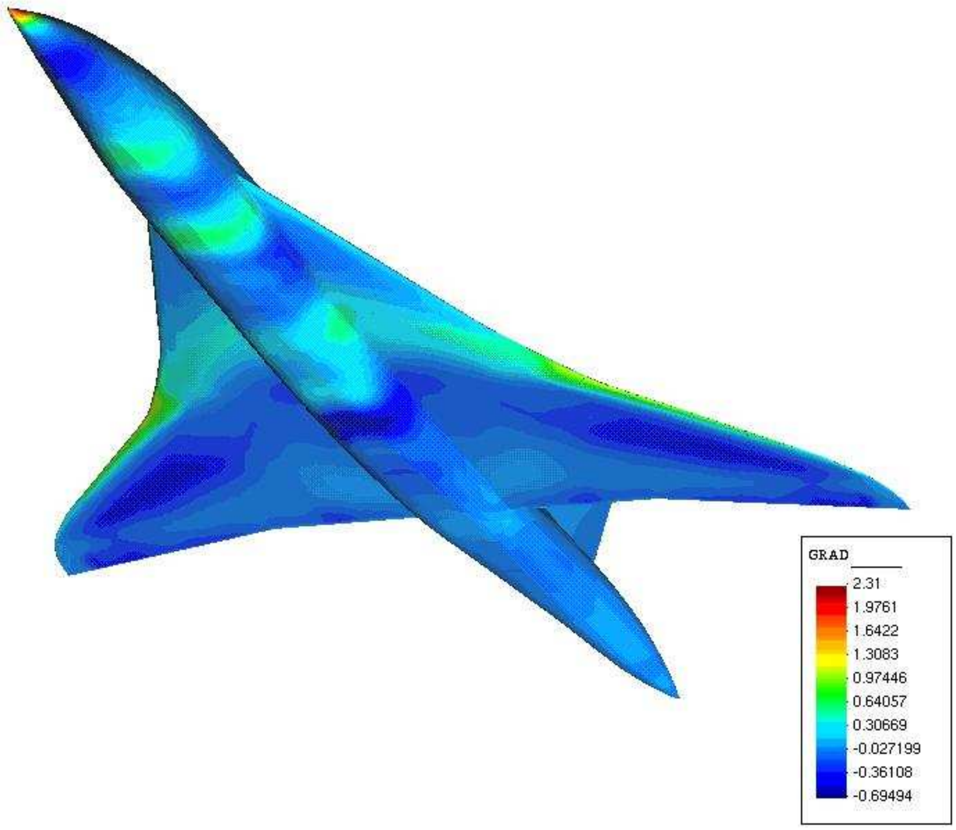

FIG. 2 – *Gradient de la fonction coût sur un Falcon en régime supersonique*

- les spécialistes de l'optimisation de forme,

- les spécialistes de la Mécanique des Fluides Numérique, qui sont à la fois des Mécaniciens et des Numériciens. A ce titre, ils peuvent s'intéresser à l'amélioration des outils de simulation qui forment le cœur de l'optimisation, et en particulier à l'adaptation de maillage. En effet, les états adjoints de Tropics trouvent aussi en adaptation de maillage une intéressante application.

Cette thèse présente donc des contributions dans les trois domaines suivants :
Partie 1 : Différentiation Automatique de Programme.
Partie 2 : Optimisation de forme pour de grands systèmes.
Partie 3 : Adaptation de maillage.

0.2 Différentiation Automatique de Programme

Dans les problèmes d'optimisation de formes en aérodynamique, un problème s'impose très rapidement : comment calculer le gradient de la fonction "coût" ? Afin de calculer le gradient, il est essentiel de pouvoir calculer l'état adjoint du problème. Deux méthodes pour calculer le système adjoint sont à l'heure actuelle possibles :
- soit le calculer à la main à partir de l'EDP initiale, puis le discrétiser et enfin le résoudre dans un programme [Jameson, 1988a]. Cette méthode est celle de l'adjoint continu.
- soit le *mode adjoint* de la *Différentiation Automatique (DA)* peut générer un nouveau programme qui résoud le système adjoint, en partant du programme qui résoud l'EDP originale. Cette méthode est celle de l'adjoint discret.
Ces deux approches sont assez similaires et ont des complexités voisines. Dans cette thèse, nous avons choisi d'utiliser la seconde option. Cette option nous permet d'obtenir le gradient exact de la fonctionnelle discrète qui est calculée.

La différentiation automatique et le logiciel TAPENADEdéveloppé par le projet Tropics de l'INRIA ont été utilisés dans toutes les applications décrites dans ce mémoire, c'est-à-dire :
- Dans le démonstrateur d'optimisation de forme aérodynamique bidimensionnel utilisé pour présenter les méthodes introduites en Différentiation Automatique pour le Contrôle Optimal (Partie I).
- Dans le démonstrateur pré-industriel 3D pour la rédustion du bang sonique (Partie II).
- Dans le démonstrateur d'optimisation de maillage présenté en Partie III.

0.3 Optimisation de forme pour de grands systèmes

Diverses études théoriques ont été entreprises notamment pour le calcul de la variation de la solution d'une équation aux dérivées partielles par rapport à la frontière. Ainsi au début du siècle, J. Hadamard fut l'un des premiers à se pencher sur ces problèmes (voir [Hadamard, 1968]) ; par la suite, M. Schiffer s'intéressa à son tour au calcul des variations dans [Schiffer, 1946], ses travaux étant prolongés par P. Garabedian pour des problèmes de frontières libres (voir [Garabedian, 1964]). Dans les années 70-80, l'évolution des ordinateurs et les progrès accomplis dans le domaine du calcul scientifique ont suscité un regain d'intérêt pour ces problèmes de conception optimales de formes. On peut citer, par exemple, les recherches entreprises par J. Céa et son équipe ([Cea, 1981]), J.-P. Zolésio dans [Zolesio, 1979], B. Rousselet,...), par O.Pironneau dans [Pironneau, 1976], F. Murat et J. Simon dans [Murat and Simon, 1974] ; puis les travaux du potentiel en éléments finis de F.Angrand et al. dans [Angrand *et al.*, 1980] poursuivis en volumes finis par A. Jameson dans [Jameson, 1988b].

Les méthodes de résolution des problèmes d'optimisation de formes sont très diverses. Dans l'ensemble, elles consistent toutes à minimiser un ou plusieurs critères, avec ou sans contraintes. Nous considérons uniquement le cas mono-critère, que nous noterons j et que nous appellerons fonction coût. On distingue principalement deux grands types de méthodes : celles utilisant le gradient de j, et celle ne l'utilisant pas, c'est-à-dire essentiellement les méthodes évolutionnaires (algorithmes génétiques,...) ou recuit simulé. Nous n'étudierons que le premier type de méthodes, ce qui suppose que la fonction j est suffisamment régulière, ou pour le moins différentiable. De plus nous exigerons la même régularité pour les contraintes de conception. Notre choix est essentiellement motivé par le fait que les fonctions de coût classiques en aérodynamique nécessitent un temps de calcul important, ce qui rend quasi-impossible l'utilisation de méthodes stochastiques. Ces méthodes sont plus à réserver à une première approche lorsque l'on ne connait rien de l'optimum et que l'on désire trouver le minimum global du problème. Quitte ensuite à utiliser une méthode de gradient pour connaitre avec plus de précision le minimum.

Les méthodes de gradient requièrent généralement des **préconditionneurs**. Les expériences montrent que c'est le cas en Optimisation de Formes. Nous proposons une explication de ce problème, liée au manque de régularité de la dérivée de Hadamard de la fonction par rapport au domaine de forme. Nous proposons une famille de préconditionneurs pour laquelle le gain de régularité est un paramètre qui peut être spécifié selon le problème à résoudre. Pour cela, nous adaptons le préconditionneur multiniveau additif de Bramble-Pasciak-Xu (BPX)

0.4 Adaptation de maillage

La relation qui existe entre adapter et optimiser n'a échappé à personne depuis que l'on se pose la question de l'adaptation des maillages utilisés pour résoudre approximativement les Equations aux Dérivées Partielles. Cependant, pendant très longtemps, il a fallu déjà trouver des méthodes pour "améliorer" le maillage.

Pour améliorer le maillage, il faut d'abord identifier ses insuffisances. Cela s'est d'abord fait en se basant sur la théorie de la convergence et ses estimations *a priori*, c'est à dire avant le calcul, et en supposant connue la solution exacte. En Éléments Finis, le Lemme de Céa identifie le mécanisme de la convergence dans H^1, celui d'Aubin-Nitsche traite de la convergence L^2. Il a fallu attendre une dizaine d'années pour que Babuska et ses collaborateurs proposent une analyse *a posteriori*, c'est à dire en fonction d'un premier calcul.
L'analyse *a posteriori* s'est depuis bien développée notamment en s'inspirant du Contrôle Optimal, avec l'introduction d'**états adjoints** (Giles, Becker, Rannacher, etc.).

Pour améliorer le maillage il faut en construire un nouveau. Dans cette direction, et se basant sur les estimations *a priori*, l'équipe Gamma de l'INRIA a démontré l'efficacité des méthodes de régénération de maillage anisotropes en se basant sur des métriques issues du hessien de la solution.

Dans cette thèse, nous montrons que les métriques hessiennes s'inscrivent dans des démarches d'optimisation, lorsque l'on envisage des maillages adaptés pour une interpolation, et même dans une démarche de Contrôle Optimal, proposant par là une forme de synthèse entre la régénération sur métrique hessienne et l'adaptation via un état adjoint.

0.5 Plan de la thèse

Cette thèse est composée de trois parties traitant des trois domaines suivants : la différentiation Automatique de Programme, l'optimisation de forme pour de grands systèmes et l'adaptation de maillage.

Dans la partie 1, nous exposons notre contribution concernant l'amélioration du mode adjoint de la Différatiation automatique et nous présentons notre modèle d'application, tirée de la Mécanique des Fluides Numérique.

Dans le chapitre 1 de la partie 2, nous présentons une nouvelle génération d'algorithmes proche de la classe SQP qui incorporent quelques unes des propriétés modernes de SQP tout en conservant l'efficacité de l'approche one-shot.

Dans le chapitre 2 de la partie 2, nous proposons de pousser plus loin l'idée d'Arian et de Ta'asan en utilisant une famille de préconditionneurs pour laquelle le gain de régularité est un paramètre qui peut être spécifé selon le problème à résoudre. Pour cela, nous adaptons le préconditionneur multiniveau additif de Bramble-Pasciak-Xu (BPX) (voir [Bramble *et al.*, 1990; Xu, 1997; Cohen, 2000]).

Dans le chapitre 1 de la partie 3, nous étudions le problème du meilleur maillage adapté pour de l'interpolation pure. Nous spécifions le maillage par une métrique et nous modélisons l'erreur par le premier terme de la série de Taylor de l'erreur d'interpolation. La résolution du système d'optimalité nous permet alors de déterminer une expression complètement explicite de la métrique optimale en fonction de la fonction à adapter.

Dans les chapitre 2 et 3 de la partie 3, nous étendons la méthode proposée dans le chapitre 1 de la partie 3 au problème de l'adaptation de maillage pour EDP. Notre méthode repose sur une analyse *a priori* rigoureuse puis sur une modélisation. Il en résulte une formulation en contexte continu de la recherche du meilleur maillage, minimisant une certaine fonctionnelle. Cette nouvelle application des méthodes de l'optimisation avec adjoint (et DA) est en cours de mise en œuvre et sera appliquée à une stratégie combinant adaptation anisotrope avec adjoint, d'une part, et optimisation de forme, d'autre part, pour l'optimisation de l'avion supersonique.

Cette thèse a donné lieu aux publications et aux communications suivantes :

• • • Prix et Distinctions • • •

2003 Lauréat du **Prix Editeur** pour ma communication orale lors des Onzièmes Journées du groupe MODE de la SMAI, à Pau, du 27 au 29 Mars 2003,
Application d'un préconditionnement multiniveau en optimisation de formes

• • • Publications • • •

Publications accessibles sur :
`http ://www-sop.inria.fr/tropics/Francois.Courty/Publications.html`

Article accepté dans des journaux internationaux.

F. Courty, A. Dervieux, B. Koobus, L. Hascoet,
Reverse Automatic Differentiation for Optimum Design : from Adjoint State assembly to Gradient Computation
Optimization Methods and Software (OMS), Volume 18, Number 5 (October 2003), pp.615-627

Articles soumis à des journaux internationaux.

F. Courty, A. Dervieux
Multilevel Functional Preconditioning for Shape Optimization
soumis (Juin 2003) au journal Numerische Mathematik.

F. Courty, A. Dervieux
A SQP-like One Shot algorithm to optimal control problems
soumis (Avril 2003) au journal Mathematical Programming

F. Courty, D. Leservoisier, P.-L. Georges, A. Dervieux
Continuous metrics and mesh optimization
soumis (Janvier 2003) au journal Applied Numerical Mathematics.

Conférences internationales avec publication des actes.

A. Dervieux, F. Courty, M. Vázquez, B. Koobus,
Additive multilevel optimization and its application to sonic boom reduction,
Proceedings of Conference JP60, Jyvaskyla, Finlande, 12-15 Juin 2002, à paraître. Variational Problems and Applications, 2002.

Rapports de recherche INRIA.

F. Courty, A. Dervieux,
Trust Region and One Shot approach to optimal control problems,
Février 2003.

F. Courty, A. Dervieux, B. Koobus, L. Hascoet,
Reverse Automatic Differentiation for Optimum Design : from Adjoint State assembly to Gradient Computation, no 4363, Janvier 2002

• • • Communications • • •

Séminaires invités.

Séminaire E.D.P et Analyse Numérique au Laboratoire Dieudonné de l'université de Nice, 16 Octobre 2003.

Séminaire au Laboratoire de Mathématiques Jean Leray à l'université de Nantes, 26 Juin 2003.
Titre : *Quelques applications de l'optimisation : maillages optimaux, formes optimales.*

Séminaire d'Analyse Non-Linéaire et d'Optimisation au laboratoire d'Arithmétique, de Calcul formel et d'Optimisation (LACO) de l'université de Limoges, le 7 Mars 2003.
Titre : *Analyse et résolution d'un système d'optimalité en mécanique des fluides numérique.*

Séminaire au laboratoire de Mathématiques pour l'Industrie et la Physique (MIP) de l'université Paul Sabatier de Toulouse, 22 Janvier 2003.
Titre : *Optimisation différentiable en mécanique des fluides numérique.*

Conférences, Colloques, Workshop.

Conférence à la 21$^{\text{ème}}$ conférence IFIP TC7 on System Modeling and Optimization, à Sophia-Antipolis, France, du 21 au 25 Juillet 2003,
Trust Region and One Shot approach in optimal shape design

Séminaire sur la Simulation en Aérodynamique et en Espace dans le cadre du Séminaires croisé INRIA, entre les projets TROPICS et CAIMAN, à Sophia-Antipolis, le 1er Juillet 2003,
Quelques applications de l'optimisation : maillages optimaux, formes optimales.

Conférence aux Onzièmes Journées du groupe MODE de la SMAI, à Pau, du 27 au 29 Mars 2003,
Application d'un préconditionnement multiniveau en optimisation de formes

Conférence à la conférence sur les Mathématiques Appliquées et les Applications des Mathématiques„ AMAM 2003, Nice, du 10 au 13 Février 2003,
Analysis and solution of the optimality system in aerodynamical shape design.

Présentation à la réunion technique du projet européen Aeroshape, du 23 au 24 Janvier 2003, à l'Aérospatiale, à Toulouse.

Conférence au Workshop on Optimization in Partial Differential Equations and Applications, OPA 2002, Heidelberg, Allemagne, du 7 au 9 Octobre 2002,
Application of multilevel preconditioning to SQP algorithms.

Conférence à la conférence Franco-Germano-Polonaise sur l'optimisation, à Cottbus, Allemagne, du 9 au 13 Septembre 2002, *Trust region and one shot approach to optimal control problems.*

Présentation à la réunion technique du projet européen Aeroshape, les 19 et 20 Juin 2002,à l'ONERA à Chatillon.

Conférence aux Dixièmes Journées du groupe MODE de la SMAI, à Montpellier, en Mars 2002,
Application de la méthode one-shot aux algorithmes SQP en optimisation de formes.

Présentations aux réunions techniques du projet européen Aeroshape, du 29 au 31 Janvier 2002, au CIRA, Capua, Italie.

Première partie

Contribution à la Différentiation automatique

1

Application de la Différentiation Automatique en mode adjoint à l'optimum design

Sommaire

Le contenu de ce chapitre a été accepté pour publication dans la revue *Optimization Methods and Software*.

1.1 Introduction

La méthode de gradient est l'une des techniques clé en optimisation en particulier pour l'optimisation de systèmes gouvernés par des *Équations aux Dérivées Partielles (EDP)*. Dans ce contexte, les systèmes adjoints sont en train de devenir de plus en plus populaires dans le calcul des gradients.

Les systèmes adjoints peuvent être *(i)* calculés à la main à partir de l'EDP initiale, puis discrétisés et enfin résolus dans un programme [Jameson, 1988a]. Alternativement *(ii)*, le *mode adjoint* de la *Différentiation Automatique (DA)* peut générer un nouveau programme qui résoud le système adjoint, en partant du programme qui résoud l'EDP originale. Ces deux approches sont assez similaires et ont des complexités voisines. Elles peuvent être comparées comme suit :

(i) Un adjoint écrit à la main possède une base mathématique. Comme ce procédé revient à l'EDP, il peut être mieux compris et plus fiable que *(ii)*. Cette méthode implique une nouvelle phase d'implémentation séparée pour résoudre le système adjoint. Durant la phase manuelle, la connaissance mathématique du problème peut être traduite en de nombreux raffinements codés à la main. Mais cette méthode peut prendre un temps fastidieux. Excepté pour certaines stratégies (voir [Giles, 2001]), cette approche ne produit pas un gradient exact de la fonctionnelle discrétisée, et ceci peut être source de problèmes si l'on utilise des méthodes d'optimisation basées sur des directions de descente.

(ii) Un adjoint obtenu intégralement par la DA est l'adjoint de la fonctionnelle discrétisée calculé par le programme, qui est différentiable par morceaux. Il produit des dérivées exactes presque partout. Des résultats théoriques [Gilbert, 1992] garantissent la convergence de ces dérivées lorsque la fonctionnelle converge.
Cette stratégie rend possible l'utilisation de directions de descente dans le noyau d'optimisation, mais le pas de descente peut être petit à cause des discontinuités. La validation est plus facile (e.g. par différences divisées) parce qu'on dispose de l'adjoint du programme original lui-même. Mais surtout, l'adjoint DA est *généré* par un outil. Ceci économise un temps énorme de développement et de déboguage. Cependant cette approche systématique repose sur une utilisation massive du stockage mémoire, requérant une transformation du code à la main pour réduire l'utilisation de mémoire. Le travail de Mohammadi [Hovland *et al.*, 1997] [Mohammadi, 1997] illustre les avantages et les inconvénients de cette approche.

Nous proposons une nouvelle stratégie pour résoudre le système adjoint discret qui est un compromis entre les deux approches précédentes. Similairement à *(i)*, nous écrivons le système adjoint, mais directement dans le cas discret. Ces équations doivent être assemblées puis résolues. Dans ce but, nous identifions d'abord certaines parties de l'équation d'état discrète et de la fonctionnelle de coût dont les dérivées devront apparaître dans le système adjoint. Ces dérivées sont évaluées par des sous-programmes générés par le mode adjoint de la DA, comme dans l'approche *(ii)*. Puis nous utilisons ces sous-programmes pour assembler le résidu du système adjoint discret. Il faut noter que l'étape de résolution, nous devons construire un algorithme *matrix-free*, car seuls les résidus du système adjoint sont disponibles.

Examinons les bénéfices de notre stratégie, comparés aux travaux précédemment cités.

La classe des problèmes que nous considérons, aussi bien que notre exemple d'application, sont deux caractéristiques dont nous tirons profit : ce sont des *modèles stationnaires*, et leur implémentation utilise de nombreuses boucles avec des *itérations indépendantes* (*II*-boucles en raccourci), telles que les boucles d'assemblage. Ces caractéristiques sont indépendantes, et peuvent être exploitées séparément :

- **modèles stationnaires :** Dans le cas général des modèles non-stationnaires, l'adjoint, qu'il ait été obtenu soit par codage à la main, soit par DA, requiert de stocker toutes les données de la trajectoire du calcul parce que la DA utilise ces données dans l'ordre inverse duquel elles sont calculées par le programme. Ce stockage est souvent très important à moins qu'une stratégie de recalcul partiel ne soit appliquée, *voir* [Griewank, 2000]. Similairement, dans le cas des **modèles stationnaires**, la DA en mode adjoint apppliquée de manière directe nécessiterait la même quantité énorme de mémoire. Pouvons-nous éviter ceci en utilisant la stationarité du modèle ? C'est possible, comme cela a été montré dans [Griewank and Faure, 2002], où le calcul de l'état adjoint utilise les états itérés dans l'ordre direct. Alternativement, la plupart des chercheurs (voir par exemple [Hovland *et al.*, 1997]) utilise seulement l'état pleinement convergé pour calculer l'adjoint. Cette stratégie est usuellement implémentée par des modifications manuelles du code généré par la DA. Mais ce procédé est délicat et source d'erreur. Notre stratégie hybride utilise aussi l'état convergé. La DA est appliquée aux boucles d'assemblage, sauvant ainsi un gros effort de discrétisation et d'implémentation. D'un autre coté, un algorithme de résolution spécifique, spécialement adapté à l'état adjoint, nous semble la bonne option. Cette démarche ne requiert aucune modification à la main du code généré par la DA.
- ***II*-boucles :** À l'intérieur de la partie du code qui est différentiée automatiquement, nous trouvons de nombreuses boucles avec des itérations indépendantes. Notre contribution est un mode adjoint de DA spécifique pour ces boucles, qui demande beaucoup moins de mémoire. À notre connaissance, cette technique a été utilisée occasionnellement [Sevin, 1999], à la main, sans réelle assurance de son exactitude. Nous avons donné une description formelle de la technique dans un article précédent [Hascoet *et al.*, 2001] puis une preuve de son exactitude dans [Hascoet, 2001]. Nous proposons ici un traitement systématique de ces boucles à l'intérieur de notre outil de DA TAPENADE. L'outil TAMC de DA [Giering, 1997] utilise aussi une stratégie spéciale pour les boucles parallèles. Mais TAMC implémente une stratégie de *"tout-recalcul"* dans le mode adjoint, et se passe de stockage-mémoire au prix d'une duplication de l'éxécution des instructions du programme original. Par conséquent le problème de la mémoire est moins crucial, et une comparaison avec notre approche, basée sur le *"tout-stockage"* est difficile. Néanmoins elles sont similaires dans le sens que toutes deux reposent sur l'observation que les adjoints des instructions indépendantes sont indépendants.

Ce chapitre est organisée comme suit. Le paragraphe 1.2 présente notre modèle d'application, tirée de la Mécanique des Fluides Numérique. Les EDP sont discrétisées et résolues par un programme. Puis nous construisons l'EDP adjointe discrète, à l'intérieur de laquelle nous identifions les sous-expressions qui seront évaluées par la DA. Le paragraphe 1.3 décrit l'utilisation actuelle de notre outil de DA TAPENADE, en insistant sur le traitement spécifique des boucles-*II*. La section 1.4 décrit l'assemblage final et la résolution du gradient, en utilisant les sous-programmes générés par DA. Le paragraphe 1.5 donne des résultats numériques.

1.2 Un problème modèle en optimisation de formes

L'intérêt de la méthode proposée dans ce chapitre sera montré sur un problème certes académique mais cependant représentatif : l'optimisation de la forme d'une tuyère 2D, dans laquelle l'écoulement est modélisé par les équations d'Euler.

1.2.1 Contexte continu

La figure 1.1 décrit la géométrie et son maillage en triangles. Elle représente une tuyère dont l'entrée est le bord vertical gauche et la sortie est le bord vertical droit. La partie médiane de la tuyère peut varier sur son bord supérieur, et le maillage varie en conséquence. Ce problème a été étudié dans le cadre d'un projet européen BRITE-ECARP [Beux, 1994][Marco and Dervieux, 1995].

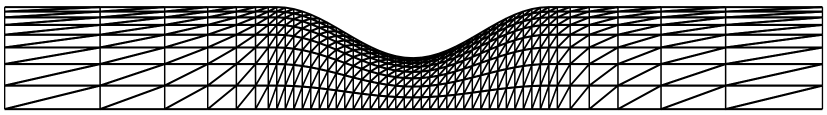

FIG. 1.1 – *Maillage de la tuyère comportant 423 noeuds*

Le paramètre de contrôle est une fonction γ de l'abscisse qui définit la forme du bord supérieur. Pour chaque fonction γ dans un ensemble admissible Γ_{ad}, l'écoulement stationnaire $W(\gamma)$ d'un gaz parfait dans la tuyère est défini par les équations d'Euler

$$\Psi(\gamma, W(\gamma)) = 0$$

Cette équation prend en compte les flux internes et les conditions aux bords :

$$\Psi(\gamma, W) \quad = \quad \frac{\partial F(W)}{\partial x} + \frac{\partial G(W)}{\partial y} + \text{ conditions aux bords}$$

$$\text{avec } W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix}, \quad F(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(e+p) \end{pmatrix}, \quad G(W) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(e+p) \end{pmatrix}$$

dans lesquelles ρ est la masse volumique, (u, v) la vitesse suivant (x, y), e l'énergie totale, et p la pression définie par

$$p = P(W) = (\kappa - 1)[e - \frac{1}{2}\rho(u^2 + v^2)] \quad \text{with} \quad \kappa = 1.4$$

Les calculs présentés auront deux types de conditions aux bords :

La condition de glissement qui exprime que le flux ne traverse pas la paroi, elle s'écrit :

$$(u, v).\vec{\eta} \ = \ 0$$

où $\vec{\eta}$ est la normale extérieure au domaine.

Les conditions à l'infini. Nous faisons l'hypothèse d'un écoulement uniforme à l'infini :

$$W \ = \ W_\infty \ = \ \begin{pmatrix} \rho_\infty \\ \rho_\infty u_\infty \\ \rho_\infty v_\infty \\ e_\infty \end{pmatrix}$$

avec $\rho_\infty \ = \ 1$, $(u_\infty, v_\infty) \ = \ (cos\alpha \ , \ sin\alpha)$ et $p_\infty \ = \ \frac{\rho_\infty}{\kappa M_\infty^2}$. α est l'angle d'incidence et M_∞ est le nombre de Mach à l'infini.

On définit une fonctionnelle J, qui mesure la déviation de la pression $P(W)$ avec la pression prescrite P^{cible} sur le bord supérieur

$$J(\gamma, W) = \int_\gamma (P(W) - P^{cible})^2 d\sigma$$

et on définit la fonction coût $j(\gamma) = J(\gamma, W(\gamma))$.

Le problème d'optimisation consiste à minimiser j par rapport à γ dans l'ensemble Γ_{ad}.

Dans tout ce qui suit, on admet qu'il existe un minimum pour ce problème d'optimisation de formes (voir [Mohammadi and Pironneau, 2001] pour la théorie).
On admet d'autre part que ce minimum est atteint par une méthode de gradient.

1.2.2 Contexte discret

Pour $\gamma = 1$, le maillage est une triangulation orthogonale d'un domaine rectangulaire. Pour un γ arbitraire, donné par le vecteur $\gamma_i = \gamma(x_i)$, for $i = 1, m$, le maillage est déformé en bougeant les nœuds sur les lignes verticales. Les coordonnées X des nœuds du maillage sont des fonctions explicites de la variable γ_i.

$$\forall i \in \{1, \ldots, m\}, \ y^\gamma_{im'} = \gamma_i \ ; \ \ \forall j \in \{1, \ldots, m'\}, \ y^\gamma_{ij} = \theta_j \ y^\gamma_{im'} \ + \ (1 - \theta_j) \ y^\gamma_{i1} \tag{1.1}$$

La fonction représentant les équations d'Euler, Ψ est discrétisée en utilisant une formulation en volumes finis. Le flux élémentaire $\Phi_{j,k}$ entre les volumes ou cellules j et k est évalué à l'aide de la décomposition de Roe. Cela donne Ψ_1 au premier ordre :

$$\Psi_1(X, W)_j \ = \sum_{k \ voisin \ de \ j} \Phi(W_j, W_k, \vec{\eta}_{jk}(X)) + \Phi_B(W_j, \vec{\eta}_j(X)) \ . \tag{1.2}$$

Les discrétisations dépendent aussi des coordonnées X du maillage à travers $\overrightarrow{\eta}_{jk}(X)$ et $\overrightarrow{\eta}_j(X)$.

Au second ordre, nous avons Ψ_2 :

$$\Psi_2(X,W)_j = \sum_{k\ voisin\ de\ j} \Phi(W_{jk}, W_{kj}, \overrightarrow{\eta}_{jk}(X)) + \Phi_B(W_j, \overrightarrow{\eta}_j(X)) \tag{1.3}$$

où le flux élémentaire inter-cellule Φ dépend de la variable d'état W au premier ordre à travers W_j et W_k, valeurs de W aux nœuds j et k, et au second ordre suivant W_{jk} et W_{kj}, les interpolations linéaires sur le bord j-k suivant les idées "MUSCL" de van Leer, étendues aux triangulations comme dans [Francescatto and Dervieux, 1998].

Les valeurs d'intégration W_{jk} et W_{kj} sont calculées à partir des valeurs moyennes dans les cellules et d'évaluation des gradients moyens sur les nœuds :

$$\begin{aligned}\vec{\nabla} W_i &= \frac{\iint_{C_i} \vec{\nabla} W\,dxdy}{\iint_{C_i} dxdy} \\ &= \frac{1}{aire\,(C_i)} \sum_{T, i\in T} \frac{aire\,(T)}{3} \sum_{k=1}^{3} W_{ik} \vec{\nabla}\phi_{ik}\,(T)\ .\end{aligned}$$

Introduisons :

$$G_i = \vec{\nabla} W_i . \vec{ij}$$

$$G_j = \vec{\nabla} W_j . \vec{ij} .$$

Les valeurs limitées à droite et à gauche s'écrivent :

$$W_{ij} = W_i + \frac{1}{2} Lim(2G_i - (W_j - W_i), W_j - W_i)\ ,$$

$$W_{ji} = W_j - \frac{1}{2} Lim(2G_j - (W_j - W_i), W_j - W_i)\ .$$

Deux options seront considérées plus loin, d'une part le cas où nous appliquons l'*ordre deux sans limiteurs* :

$$Lim(a,b) = \frac{a+b}{2}$$

et d'autre part le cas ou nous appliquons le *limiteur de Van Albada-Van Leer* [Steve, 1988; Van Leer, 1982] qui s'écrit :

$$Lim(a,b) = \frac{((a^2+\varepsilon)b + (b^2+\varepsilon)a)}{(a^2+b^2+2\varepsilon)} \quad \text{si ab} > 0$$

$$Lim(a,b) = 0\ , \quad \text{sinon}$$

où ε est un nombre positif petit, évitant la division par zéro.

La composante Φ_B concerne les flux au bord.

Le flux élémentaire Φ est décentré grâce à la décomposition de flux de Van Leer, qui est différentiable. Dans un souci de lisibilité, nous remplaçons, à partir de maintenant, les coordonnées X par le paramètrage γ de la frontière supérieure dans les arguments de Ψ.

Nous restaurons itérativement l'état avec un schéme implicite linéarisé avançant en temps :

$$\left(\frac{M}{\Delta t^n} + \frac{\partial \Psi_2}{\partial W}(\gamma, W^n)\right)(W^{n+1} - W^n) = -\Psi_2(\gamma, W^n)$$

où M est la matrice de masse diagonale formée des aires des volumes finis. Lorsque le pas de temps Δt^n croit indéfiniment, ce schéma devient une itération de Newton afin de résoudre $\Psi(\gamma, W) = 0$.

En pratique le jacobien $\frac{\partial \Psi_2}{\partial W}(\gamma, W^n)$ utilise trop d'espace mémoire. A la place, nous utilisons dans le code de calcul une matrice jacobienne simplifiée "spatialement au premier ordre" [Stoufflet, 1984]

$$A_1 = \frac{\partial \Psi_1}{\partial W}(\gamma, W^n).$$

L'opérateur A_1 est compact au sens suivant : les termes advectifs sont discrétisés en un nœud i, en utilisant seulement les voisins direct j de i :
$j \in V(i) \Leftrightarrow \exists\, T$ élément tel que $i \in T$, $j \in T$.
En moyenne un noeud possède 14 voisins. Au contraire, Ψ_2 utilise les voisins au second degré et est beaucoup plus grande.

Le schéma implicite linéarisé avançant en temps s'écrit alors :

$$W^{n+1} = W^n - \left(\frac{M}{\Delta t^n} + A_1\right)^{-1} \Psi_2(\gamma, W^n). \tag{1.4}$$

L'opérateur A_1 agit comme un préconditionneur. Nous nous référons à [Francescatto and Dervieux, 1998] pour une récente analyse détaillée de A_1. Grâce à ses propriétes de quasi-diagonale dominance, le système :

$$\left(\frac{M}{\Delta t^n} + A_1\right) \delta W = RHS \tag{1.5}$$

est aisément résolu avec des balayages de la méthode de Jacobi. Il en résulte une stratégie de résolution qui utilise un stockage mémoire raisonnable et qui est beaucoup plus efficace qu'une résolution explicite.

Le **problème d'optimisation discrétisé** consiste à minimiser

$$j(\gamma) = J(\gamma, W(\gamma)),$$

mais cette fois avec $W(\gamma)$ solution du **système d'état discrétisé** :

$$\Psi_\alpha(\gamma, W(\gamma)) = 0$$

défini par (1.2), (1.3) avec $\alpha = 1$ ou 2. On omet désormais l'indice α.
Nous pouvons maintenant écrire la condition d'optimalité discrète. Appliquant la dérivation composée dans la différentiation de j, et introduisant l'état adjoint Π, nous obtenons le **système d'optimalité** suivant

$$\left\{\begin{array}{rcll} \Psi(\gamma, W) & = & 0 & \textit{(équation d'état)} \\ \frac{\partial J}{\partial W}(\gamma, W) - (\frac{\partial \Psi}{\partial W}(\gamma, W))^t \,.\, \Pi & = & 0 & \textit{(équation adjointe)} \\ j'(\gamma) = \frac{\partial J}{\partial \gamma}(\gamma, W) - (\frac{\partial \Psi}{\partial \gamma}(\gamma, W))^t \,.\, \Pi & = & 0 & \textit{(condition d'optimalité)} \end{array}\right. \tag{1.6}$$

Dans ce système, nous mettons en grisé les quatre expressions que nous avons identifiées comme des différentielles d'expressions issues du problème original discrétisé (J et Ψ). Ces différentielles seront évaluées par des routines générées par le mode adjoint de la différentiation automatique, comme décrit dans la section 1.3.

1.3 Utilisation et amélioration de la DA en mode adjoint

La Différentiation Automatique ou Algorithmique (DA) différencie des *programmes*. Nous entendons, par cela, qu'un logiciel de DA prend en entrée un programme source qui, à partir d'un argument donné x, calcule une fonction $f(x)$. Le logiciel de différentiation automatique génère un nouveau programme source qui, à partir d'un argument donné x, calcule des dérivées de f au point x. Nous renvoyons le lecteur intéressé par la DA au recueil d'articles [Corliss *et al.*, 2001] et à la récente monographie [Griewank, 2000]. Dans le présent travail, nous avons besoin du mode *adjoint* de la DA , que nous allons présenter brièvement. Nous allons montrer dans quelle mesure le mode adjoint peut être largement amélioré dans un cas spécifique, et comment cette amélioration peut être automatisée.

Fondamentalement, la DA identifie les programmes avec des compositions de fonctions mathématiques. Plus précisément, tout programme P composé d'une suite d'instructions $I_k, k \in [1..p]$ et qui implémente une fonction f, $f : I\!R^m \to I\!R^n$, est tel que

$$f = f_p \circ f_{p-1} \circ \ldots \circ f_1$$

où chaque f_k est la fonction élémentaire implémentée par l'instruction I_k. Utilisant la dérivation composée et écrivant f' comme la dérivée (matrice jacobienne) de f, nous obtenons :

$$\begin{array}{rcl} f'(x) & = & (f'_p \circ f_{p-1} \circ f_{p-2} \circ \ldots \circ f_1(x)) \\ & & .\ (f'_{p-1} \circ f_{p-2} \circ \ldots \circ f_1(x)) \\ & & .\ \ldots \\ & & .\ (f'_1(x)). \end{array} \tag{1.7}$$

Le *mode inverse* de la DA produit un *gradient*, i.e. un programme qui, étant donnés x et un vecteur $\overline{y} \in I\!R^n$, calcule le produit $f'^t(x).\overline{y}$. On peut voir $\overline{y}$ comme un vecteur poids sur y, le résultat de f, qui définit un résultat scalaire composé, à partir duquel nous calculons le gradient. Ce mode est aussi appelé *mode adjoint*. Partant de l'équation (1.7) et après transposition, nous obtenons :

$$\begin{aligned} f'^t(x).\overline{y} \quad = \quad & (f_1'^t(x)) \\ & .\ (f_2'^t \circ f_1(x)) \\ & .\ \ldots \\ & .\ (f_p'^t \circ f_{p-1} \circ f_{p-2} \circ \ldots \circ f_1(x))\ .\ \overline{y}\ . \end{aligned}$$

Afin de calculer $f'^t(x).\overline{y}$, et parce que les produits matrice$\times$vecteur sont beaucoup moins coûteux en opérations que les produits matrice$\times$matrice, le mode adjoint commence par calculer

$$\overline{y}_{p-1} = (f_p'^t \circ f_{p-1} \circ f_{p-2} \circ \ldots \circ f_1(x)).\overline{y}$$

puis il calcule

$$\overline{y}_{p-2} = (f_{p-1}'^t \circ f_{p-2} \circ \ldots \circ f_1(x)).\overline{y}_{p-1}$$

et ainsi de suite, jusqu'à d'obtenir

$$f'^t(x).\overline{y} = \overline{y}_0 = (f_1'^t(x)).\overline{y}_1$$

Nous observons que ce processus nécessite de connaître les valeurs intermédiaires de l'éxécution de P dans l'*ordre inverse* de leur calcul par P. Un moyen de gérer cela est de recalculer chaque valeur intermédiaire lorsque nécessaire. Cela implique des calculs répétés de chaque fonction f_k, ce qui est très coûteux en calcul. A l'opposé, un autre moyen est de stocker tous les résultats intermédiaires lorsqu'ils sont calculés, afin de les retrouver lorsqu'ils sont demandés par les instructions inverses. Il est coûteux en espace mémoire. Il existe un compromis à trouver, mais dans tous les cas, c'est le principal inconvénient du mode adjoint de la DA. Concentrons nous sur la manière dont notre logiciel de DA TAPENADE [Tropics, 2001] met en œuvre ce mode adjoint. Notre choix est de *stocker* les valeurs intermédiaires, puis de les retrouver lorsque nécessaire. Pour commencer, supposons que le programme à différentier est un unique sous-programme. Appelons x_k les ensembles successifs de valeurs intermédiaires :

$$\begin{aligned} x_0 &= x \\ x_k &= f_k(x_{k-1}) \qquad \textit{pour } k = 1 \textit{ à } p \end{aligned}$$

Le programme différentié *inverse* est composé de deux parties successives. La première partie, appelée *trajectoire directe*, calcule et stocke les x_k successifs. Par conséquent la trajectoire directe est simplement une copie du programme original, entrecoupée par des instructions de stockage. La seconde partie, appelée *trajectoire rétrograde* ou bien *trajectoire adjointe*, calcule les différentielles, i.e. pour $k = p$ jusqu'à 1. Cette partie récupère l'ensemble des valeurs intermédiaires x_{k-1} et immédiatement après calcul :

$$\overline{y}_{k-1} = (f_k'^t(x_{k-1})).\overline{y}_k.$$

1.3.1 Amélioration de la DA pour des boucles à itérations indépendantes

Dans le cas général, il existe de nombreuses sous-programmes, rangés dans un *Graphe d'appels*. Le compromis entre stockage et recalcul est mis en évidence dans la figure 1.2. Lorsqu'un sous-programme A appelle un sous-programme B, la trajectoire directe appelle simplement le sous-programme original B. Donc aucun stockage n'est fait à l'intérieur de B. Durant la trajectoire adjointe de A, il ne suffit pas d'appeler la trajectoire adjointe de B, car les valeurs intermédiaires ont été perdues. On doit de nouveau exécuter B, cette fois comme une trajectoire directe, puis exécuter la trajectoire adjointe de B. Certaines valeurs doivent bien sûr être stockées pour permettre de dupliquer l'exécution de B, mais ce stockage est négligeable comparé à l'encombrement mémoire de la totalité des valeurs intermédiaires.

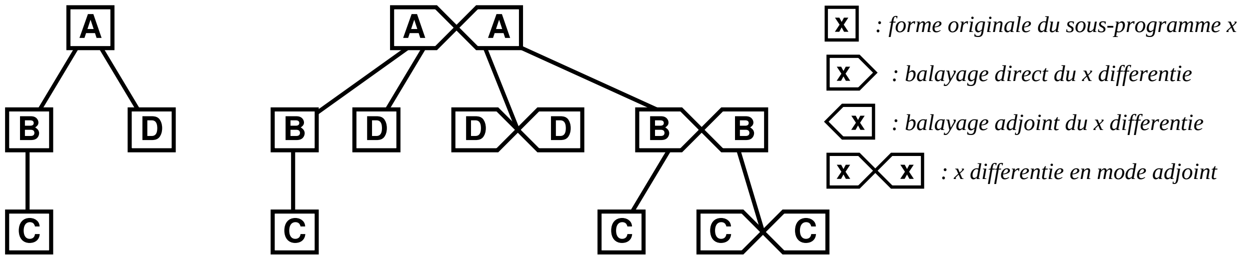

FIG. 1.2 – *Compromis stockage/recalcul sur un graphe d'appels*

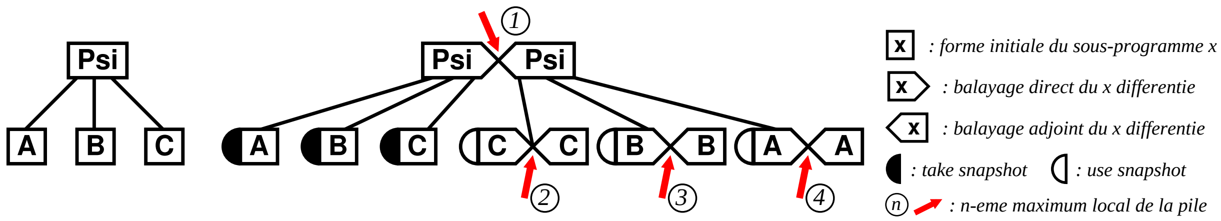

FIG. 1.3 – Graphes d'appels de la routine `Psi` originale et différentiée en mode adjoint

Nous discutons ici d'une amélioration du mode adjoint de la DA, qui réduit fortement l'occupation de la mémoire, pour le cas très fréquent de boucles dont les itérations sont à données indépendantes. Ce type d'itérations intervient très souvent dans les algorithmes de résolution d'E.D.P. notamment dans les boucles d'assemblage.

Notons qu'il existe une large classe de méthodes pour réduire la mémoire, liées essentiellement à la notion de *"checkpointing"* [Griewank, 2000]. Cette technique autorise certains recalculs afin d'économiser le stockage mémoire.

Pour les boucles d'assemblage, des techniques ad hoc ont été proposées, par exemple par Hovland, Mohammadi, and Bischof [Hovland *et al.*, 1997], consistant à faire du *checkpointing* sur chaque corps de boucle.

Notre stratégie consiste à d'abord observer que ces boucles d'assemblage (*gather-scatter*) sont essentiellement parallèles. Plus précisément, il n'y a pas de dépendance des données entre des itérations différentes de la boucle. En d'autres termes, les itérations peuvent être lancées dans n'importe quel ordre sans changer le résultat final. Nous appelons de telles boucles des boucles-*II*, pour *Itérations Indépendantes*. Dans [Hascoet *et al.*, 2001], nous avons introduit une nouvelle technique pour la différentiation adjointe des *II*-loops. Cette technique suppose que la différentiation adjointe standard d'une *II*-loop est équivalente à la forme améliorée de la figure 1.4.

Classique :		*Améliorée :*

`do i= 1,N`
$\quad$ *body(i)*
`end`
`do i= N,1`
$\quad \overline{body(i)}$
`end`
	$\Longleftrightarrow$	`do i= 1,N`
		$\quad$ *body(i)*
		$\quad \overline{body(i)}$
		`end`

FIG. 1.4 – *Transformation Équivalente d'une II-Boucle différentiée en mode adjoint*

La notation $\overline{body}$ représente le balayage adjoint correspondant au balayage direct *body*. Sur la partie droite de la figure 1.4, le balayage direct de chaque itération est immédiatement suivi par le balayage adjoint $\overline{body(i)}$, et par conséquent les valeurs intermédiaires stockées sont utilisées immédiatement après. Seule la mémoire correspondant à une itération est occupée. En conséquence, cette manipulation réduit drastiquement la taille de l'espace mémoire utilisé (d'un facteur N, taille de la boucle).

Dans [Hascoet *et al.*, 2001], nous avons donné la preuve que les deux algorithmes sont équivalents. Nous utilisons la propriété, démontrée dans [Hascoet, 2001] que le graphe de dépendance d'un algorithme adjoint est isomorphe au graphe de dépendance de l'algorithme direct. Par conséquent, puisque le balayage direct est une boucle à itérations indépendantes, les itérations du balayage adjoint possède la même propriété. Nous pouvons donc changer l'ordre des itérations de la boucle adjointe de `1` jusqu'à `N`. Remarquant enfin qu'il n'y a pas, à part le stockage/récupération des valeurs intermédiaires, de dépendance des données entre le balayage direct et le balayage adjoint, nous pouvons fusionner les boucles directes et adjointes en une unique boucle.

Améliorations visant à réduire la taille de la pile

Nous avons identifié deux améliorations pour réduire la pile. Leur automatisation à l'intérieur de TAPENADE est en progrès. On utilise des informations *in-out* pour réduire la taille des snapshots, prenant seulement des variables qui non seulement sont utilisées dans le "checkpoint", mais peuvent *aussi* être modifiées avant la deuxième éxécution du "checkpoint". On observe aussi que de nombreuses boucles dans les sous-programmes `A`, `B`, et `C`, sont des boucles d'assemblage opérant sur des éléments de maillage. Ces boucles ont de *itérations indépendantes*. Dans ce cas, grâce aux considérations sur le flot de données décrites dans [Hascoet *et al.*, 2001], nous pouvons améliorer le programme différentié en mode adjoint, voir Fig. 1.4, si bien que le *balayage direct* d'une itération de la boucle *body* est immédiatement suivie par le *balayage adjoint* correspondant $\overline{body}$. Ceci réduit la taille de la pile par un facteur N, taille de la boucle et du maillage.

La table 1.1 montre les quatre maxima locaux de la taille de la pile aux endroits définis dans la Fig. 1.3, pour différentes combinaisons de deux améliorations précédentes. On peut remarquer que lorsque les *deux* améliorations doivent être appliquées, le maximum global reste trop élevé. Ce test a été fait sur un maillage réduit de 2200 noeuds. En proportion, chaque nœud dans le maillage consomme 58 `REAL*8`s dans la taille de la pile, ce qui semble acceptable.

Maximum local de la pile #	①	②	③	④
Aucune modification :	12.40	12.37	13.60	9.66
Réduction seule du snapshot :	1.02	0.85	9.70	9.33
Amélioration des boucles seules :	12.38	7.98	4.10	0.02
Deux améliorations :	1.02	0.61	0.22	0.02

TAB. 1.1 – Influence des améliorations sur la taille maximale de la pile (Mbytes)

1.3.2 Utilisation d'un logiciel de différentiation automatique

Nous sommes maintenant prêts à utiliser le mode adjoint de la DA afin d'obtenir les programmes qui évaluent les quatre expressions différentielles en grisé des équations (1.6). Nous partons de l'implémentation du Ψ_2 discrétisé, que nous prenons dans le solveur initial de l'EDP.
Nous disposons aussi d'implémentation de la fonction de coût J. Le coût $J(\gamma, W)$ est implémenté par une fonction appelée `JCOST`, possédant deux tableaux comme arguments, `XMESH` and `WFLO`, représentant γ et W :

Procédure `JCOST` : `XMESH`, `WFLO` $\longmapsto$ `JCOST`.

Puisque J est à valeurs scalaires, les jacobiennes $\frac{\partial J}{\partial W}$ et $\frac{\partial J}{\partial \gamma}$ sont des matrices-lignes. La fonction différentiée en mode adjoint calcule $f'^t(x).\overline{y}$ pour tout $\overline{y}$ donné, qui est ici un vecteur avec une seule coordonnée. Par conséquent, dans ce cas particulier, imposer $\overline{y} = 1.0$ dans la fonction adjointe différentiée donne directement les jacobiens désirés. Nous appliquons donc à la fonction `JCOST` le mode adjoint, en différentiant la sortie `JCOST` par rapport à la variable en entrée `XMESH`. Nous obtenons un sous-programme qui, à partir de `XMESH`, `WFLO`, et `JCOSTB` (représentant $\overline{y}$, initialisé à 1.0), calcule le tableau `XMESHB` qui contient $\frac{\partial J}{\partial \gamma}$. De même, en remplaçant `WFLO` par `XMESH` dans ce qui précède, nous obtenons un sous-programme qui calcule $\frac{\partial J}{\partial W}$.
Le flux d'ordre deux, Ψ_2, est implémenté par un sous-programme appelée `FLUX`, dont les entrées sont de nouveau `XMESH` and `WFLO`, dont la variable en sortie `PSIFLUX` contient $\Psi_2(\gamma, W)$.

Procédure `FLUX` : `XMESH`, `WFLO` $\longmapsto$ `PSIFLUX`.

Cette fois les jacobiens $\frac{\partial \Psi_2}{\partial W}^t$ et $\frac{\partial \Psi_2}{\partial \gamma}^t$ sont des matrices trop grandes pour être calculées explicitement. Pour résoudre en Π l'équation adjointe du système (1.6), nous allons utiliser des techniques *matrix-free*, telles que `GMRES` [Saad and Schultz, 1986] , nécessitant uniquement le programme qui calcule, pour tout Π donné, le produit matrice-vecteur :

$$(\frac{\partial \Psi_2}{\partial W}(\gamma, W))^t \;.\; \Pi.$$

Définition 1.3.1
On appelle méthode matrix-free une méthode de résolution de système linéaire qui permet de ne pas avoir à stocker la matrice du système.

Ce sous-programme est précisément celui que nous construisons par la différentiation adjointe du programme `FLUX`, avec en sortie `PSIFLUX` par rapport à `WFLO` en entrée. De même en remplaçant `WFLO` par `XMESH` dans ce qui précède, nous obtenons un sous-programme qui calcule

$$(\frac{\partial\Psi_2}{\partial\gamma}(\gamma, W))^t.\Pi \text{ pour tout } \Pi \text{ donné,}$$

comme requis dans la condition d'optimalité du système (1.6).

Ces programmes différentiés sont particulièrement efficaces grâce au traitement des *II*-loops. Maintenant nous allons combiner ces programmes différenciés pour résoudre efficacement les équations (1.6), ce qui conduira au gradient $j'(\gamma)$. Ceci est décrit dans la section 1.4.

1.4 Assemblage du gradient en utilisant les programmes différentiés

Le gradient $j'(\gamma)$ est calculé en deux étapes. Nous résolvons d'abord l'équation adjointe afin d'obtenir l'état adjoint Π. Puis nous introduisons Π dans la condition d'optimalité afin de calculer $j'(\gamma)$. Durant ce processus, un soin particulier est apporté à la manipulation des grandes matrices. L'utilisation de techniques *matrix-free* nous permet de ne pas avoir à stocker les matrices jacobiennes de Ψ_2. En fait, la seule matrice que l'on stocke est le préconditionneur A_1^t.

1.4.1 Résolution par Defect-Correction du système adjoint

Nous avons besoin de résoudre l'équation adjointe par un algorithme *matrix-free*. Il est pratique d'exploiter l'algorithme de résolution existant pour l'équation d'état Ψ elle-même. Remarquons d'abord que l'équation adjointe est suffisamment proche de l'équation d'Euler stationnaire ou sa version correspondante linéarisée pour que sa discrétisation puisse être résolue d'une manière similaire. Une expérience considérable a été accumulée dans le domaine de la résolution des équations de type Euler tant par des approches explicites qu'implicites. L'itération implicite implique généralement la résolution d'un système linéarisé approché qui préconditionne le schéma en temps conduisant à une convergence plus rapide. Considérons les deux approches pour la résolution *matrix-free* de l'équation adjointe.

– Nous utilisons un schéma pseudo-instationnaire explicite : on introduit un pas de temps artificiel τ. Le caractère explicite correspond au fait qu'aucun grand système linéaire n'est résolu pour avancer d'un pas de pseudo temps.

$$D_{\Delta\tau}\Pi \;+\; (\frac{\partial\Psi_2}{\partial W}(\gamma, W))^t\Pi \;-\; (\frac{\partial J}{\partial W}(\gamma, W)) \;=\; 0$$

Dans ce cas, nous utilisons un pas de temps déduit de l'analyse de stabilité linéaire qui s'appliquerait à un schéma similaire avançant en temps pour l'équation d'état.

Supposons que le programme généré par Différentiation Automatique qui calcule

$$(\frac{\partial \Psi_2}{\partial W}(\gamma, W))^t.\Pi$$

est plus lente d'un petit facteur k de quelques unités que le programme qui calcule $\Psi_2(\gamma, W)$. Puisque ce calcul domine le temps total de calcul, ce facteur k sera approximativement le rapport entre les coûts de l'évaluation de l'état direct W et de l'état adjoint Π. Ce rapport peut varier légèrement, avec des gains par exemple sur l'évaluation du pas de temps pour l'adjoint, une meilleure convergence de l'itération de l'adjoint due à sa linéarité, et quelques pertes telles que la norme du résidu potentiellement plus élevé. Notons que cet algorithme d'avancement en temps artificiel peut être remplacé par un algorithme GMRES linéaire qui est aussi *matrix-free* (si l'on utilise pas de préconditionneur). Ceci résulterait en une complexité légèrement plus grande à chaque itération mais aussi en une meilleure convergence.

- Dans le cas implicite, on peut utiliser un algorithme de point fixe préconditionné dans lequel le préconditionneur peut être l'adjoint du préconditionneur A_1 utilisé en 1.4 dans la résolution de l'équation d'état

$$A_1^t(\Pi^{iter+1} - \Pi^{iter}) \;=\; -(\frac{\partial \Psi_2}{\partial W}(\gamma, W))^t \Pi^{iter} \;+\; (\frac{\partial J}{\partial W}(\gamma, W)).$$

Une approche très naturelle conduit à l'algorithme :

$$\begin{cases} A_1^t\ \Pi^0 & = \dfrac{\partial J}{\partial W}(\gamma, W) \\ \\ A_1^t\ (\Pi^{n+1} - \Pi^n) & = -(\dfrac{\partial \Psi_2}{\partial W})^t\ \Pi^n + \dfrac{\partial J}{\partial W}(\gamma, W) \end{cases}$$

L'algorithme applique une itération de Defect Correction (**DeC**) à une approximation précise au second degré d'un système de type Friedrichs au premier ordre préconditionné avec une approximation au premier ordre. Desideri et Hemker [Desideri and Hemker, 1995] ont étudié en détail ce type d'itération pour les équations d'advection et d'Euler et ont prouvé que le taux de convergence peut être indépendant du maillage et aussi petit que 0.5.

Remarque 1.4.1
Dans ce qui précède, nous n'avons pas considéré l'addition du terme de la matrice de masse $\frac{M}{\Delta t^n}$. Il pourrait être utile de le réintroduire lorsque la résolution du système linéaire avec A_1 seule se révèle trop difficile.

1.4.2 Calcul du gradient

Il ne reste plus de difficulté notable pour terminer le calcul du gradient :

$$j'(\gamma) = \frac{\partial J}{\partial \gamma}(\gamma, W) - (\frac{\partial \Psi_2}{\partial \gamma}(\gamma, W))^t.\Pi.$$

Nous utilisons le programme fourni par TAPENADE qui calcule le vecteur $(\frac{\partial \Psi_2}{\partial \gamma}(\gamma, W))^t.\Pi$. Nous donnons en entrée le Π calculé précédemment. De nouveau, la matrice jacobienne $(\frac{\partial \Psi_2}{\partial \gamma})^t$ n'est jamais calculée explicitement ni stockée. De même, nous avons vu qu'un autre programme fourni par TAPENADE calcule un deuxième vecteur $\frac{\partial J}{\partial \gamma}$. Afin d'obtenir $j'(\gamma)$, il nous suffit de soustraire ces deux vecteurs.

1.5 Applications numériques

Dans cette partie, nous nous concentrons sur la meilleure façon de calculer le gradient de la fonctionnelle coût. Nous discutons l'efficacité et la précision de la stratégie que nous proposons sur un problème modèle introduit dans la section 1.2. Nous ne discutons pas de la manière d'utiliser le gradient afin d'optimiser la fonctionnelle, qui est étudiée en Partie 2 de ce mémoire.

1.5.1 Conditions des expériences numériques

Nous considérons la *forme cible* suivante :

$$\gamma_1 \ : \ y(x) = \frac{1}{2} - \frac{1}{35} + \frac{1}{35} sin(\pi(x + \frac{1}{2})), \qquad 0 \ \leq \ x \ \leq \ 2.$$

La forme γ_{cible} est utilisée pour calculer la *pression cible* puis pour spécifier complétement la fonction coût dans les contextes continus et discrets.

Nous considérons aussi la *forme initiale* suivante :

$$\gamma_{init} \ : \ y(x) = \frac{1}{2} - \frac{1}{40} + \frac{1}{40} sin(\pi(x + \frac{1}{2})), \qquad 0 \ \leq \ x \ \leq \ 2.$$

La fonction γ_{init} spécifie le domaine initial Ω_{init} de la boucle continue d'optimisation.

Pour tout maillage de Ω_{init}, les coordonnées $\gamma_{init}(x_i)$ de γ_{init} des nœuds du haut spécifient la *forme initiale discrète*, qui sont les points où nous voulons calculer la fonctionnelle coût discrète.

Les conditions de l'écoulement à l'étude sont définis comme suit : le nombre de Mach à l'infini est 0.74.

Des maillages variés de respectivement 400, 1240, 2760, et 4880 noeuds sont utilisés dans nos expériences. Dans cette étude, nous ne considérons pas les difficultés possibles pouvant apparaitre de maillages vraiement non-structurés, mais au lieu de cela nous considérons des maillages de type I-J.

1.5.2 Résolution directe de l'équation d'état

Afin d'aider à l'analyse qui va suivre, nous décrivons la variable d'état pour la forme initiale. L'écoulement est un écoulement classique dans une tuyère. Le nombre de Mach entrant est 0.74. L'écoulement stationnaire présentes une poche supersonique dans la tuyère, limité en haut (à gauche) par une ligne sonique ($x \ = \ 0.2$) et en bas (à droite) par un choc (le Mach est alors égal à 1.4), sa position est approximativement égale à $x = 1.8$. On peut voir les contours du nombre de Mach sur la Fig. 1.5

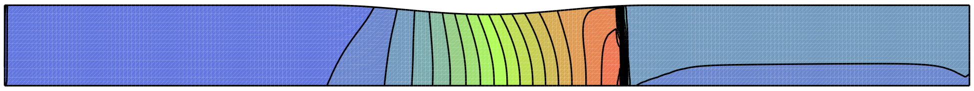

FIG. 1.5 – *Solution de l'équation d'état, nombre de Mach (sans limiteurs TVD, maillage de 4880 noeuds)*

1.5.3 Résolution efficace du système adjoint

Nous appliquons une boucle DeC avec à chaque itération une résolution complète du système d'ordre un (en pratique nous avons utilisé de nombreux balayages de Jacobi). Le nombre d'itérations DeC n'est pas aussi petit que celui prédit par Désidéri et Hemker ([Desideri and Hemker, 1995]). En effet pour la plupart des maillages, il a fallu plus de 200 itérations DeC pour diminuer le résidu de l'équation de 12 ordres de grandeur.

En pratique, cependant, nous n'avons pas besoin de converger complètement la boucle intérieure d'ordre un, mais seulement de 150 balayages de Jacobi pour avoir une bonne convergence de la DeC en moins de 300 itérations. Ce petit effort dépend fortement de la finesse du maillage mais le solveur de l'adjoint reste d'efficacité comparable à l'algorithme de résolution de l'équation d'état. En particulier, le coût, plus élevé de l'évaluation du résidu de l'adjoint apparait ne pas être une pénalisation lorsque il est fait au plus trois cent fois.

Certains contours de l'état adjoint sont présentés dans les Figs. 1.6 et 1.7 avec deux finesses différentes de maillage. Les deux calculs sont menés sans limiteurs TVD. L'influence des limiteurs est discutée dans la section 1.5.5.

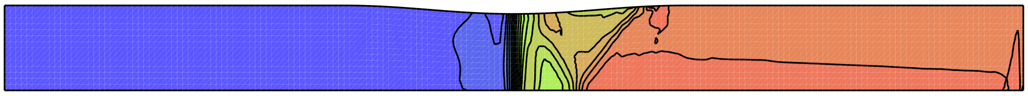

FIG. 1.6 – *État adjoint, deuxième composante, maillage moyen de 2760 noeuds*

L'interprétation du système adjoint continu des Équations aux Dérivées Partielles est une question délicate déjà abordée par plusieurs auteurs, [Cliff and Shenoy,], [Iollo and Salas,],[Iollo *et al.*, 1993]. On peut remarquer que pour les équations d'Euler, le système adjoint est linéaire, hyperbolique, et d'ordre un mais non conservatif. Il présente des lignes

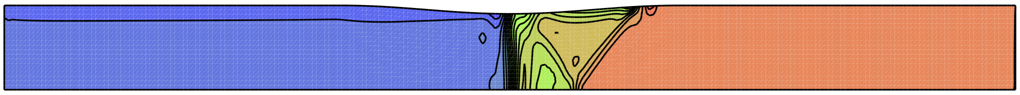

FIG. 1.7 – *État adjoint, deuxième composante, maillage fin de 4880 noeuds*

caractéristiques divergentes au voisinage des chocs de l'état et des lignes caractéristiques convergentes près des lignes soniques de la variable d'état.

Concernant les discontinuités, puisque les solutions de l'équations d'Euler satisfaisant la condition d'entropie sont les limites des solutions visqueuses, les adjoints sont aussi les limites des solutions des équations adjointes visqueuses. Ceci indique qu'un raccord continu pourrait résoudre l'apparente ambiguité de la divergence des caractéristiques.

Par conséquent, nous observons un comportement plutôt régulier de l'adjoint à l'endroit où l'état montre un choc fort.
A contrario, les lignes soniques de l'état direct ont pour effet de produire sur l'adjoint une région de fort gradient ressemblant à un choc.

Nous avons examiné le temps d'éxécution nécessaire pour résoudre le système adjoint en Π, comparé au temps nécessaire pour résoudre l'équation d'état en W. Nos évaluations montrent que résoudre en Π est au pire 4 fois plus long que résoudre en W.

Les deux opérations ont la même structure générale. Elles itèrent toutes les deux jusqu'à convergence de W_n [*resp.* Π_n] Chaque pas consiste en deux parties principales : un assemblage du résidu

$$r_n = -\Psi_2(\gamma, W^n)$$

resp.

$$-(\frac{\partial \Psi_2}{\partial W}(\gamma, W))^t \Pi^n \;+\; (\frac{\partial J}{\partial W}(\gamma, W))$$

suivi par un certain nombre d'itérations de Jacobi pour calculer l'incrément δW [*resp.* $\delta\Pi$], défini par le système linéaire

$$(\frac{M}{\Delta t^n} + A_1)\delta W = r_n$$

resp.

$$A_1^t \delta\Pi = r_n \ .$$

Nous avons d'abord étudié le compromis entre le degré de précision des itérations de Jacobi et le nombre global d'itérations pour obtenir la convergence. Lorsque l'on résoud complètement l'équation définissant l'incrément avec quelques centaines d'itérations de Jacobi, le nombre total d'itérations n'est pas aussi petit que celui qui est prédit par Desideri et Hemker [Desideri and Hemker, 1995], mais pour la plupart des maillages, 200 itérations sont suffisantes pour diviser le résidu par 10^{12}. D'un autre côté, si nous résolvons moins précisément l'incrément avec, disons, 10 fois moins d'itérations de Jacobi, alors la convergence est obtenue en moins de 300 itérations au total. Cependant, cette seconde approche est donc plus efficace.

Avec cette seconde approche, l'assemblage du résidu $-\Psi_2(\gamma, W^n)$ prend à peu près autant de temps que la résolution de l'incrément δW, et aussi autant de temps que la résolution en $\delta\Pi$. D'un autre côté, l'assemblage du résidu adjoint prend à peu près 7 fois plus longtemps que l'assemblage du résidu d'état. Ce rapport de sept est plus élevé que celui prévu par la théorie de la Différentiation Automatique. Deux raisons sont possibles : l'une est la présence de petits sous-programmes, pour lesquels le compromis stockage/recalcul de TAPENADE (figure 1.2) génère des duplications inutiles d'éxécutions. L'autre raison est la présence de très longues expressions, qui génère de nombreuses expressions dupliquées dans le code différentié. Nous pensons qu'une analyse améliorée du code dans TAPENADE peut mener à un rapport meilleur que sept. Finalement, en comptant 1 pour la durée de l'assemblage du résidu d'état, nous obtenons un total de 2 pour une itération globale de l'état, et 8 pour une itération globale de l'adjoint. Dans les deux cas, la convergence requiert 300 pas. Ceci explique le rapport de quatre environ que nous avons observé entre la résolution en Π et la résolution en W.

1.5.4 Validation du gradient

Cette stratégie de la différentiation analytique exacte peut être validée par comparaison avec des différences divisées. Nous insistons sur le fait que cette méthode n'est pas possible lorsque l'on utilise la discrétisation directe des équations EDP adjointes.

Nous espérons que les gradients obtenus par DA sont au moins aussi précis que ceux fournis par les différences divisées.

Nous présentons dans la Fig. 1.8 une comparaison entre les valeurs du gradient obtenues avec la méthode d'adjoint analytique présentée et les valeurs obtenues en applicant des différences divisées sur chaque composante du paramètre γ.

Bien que le maillage soit un maillage grossier de 400 nœuds, nous devons prêter attention à l'évaluation des gradients numériques. Des perturbations aussi petites que 10^{-7} sont appliquées aux composantes du contrôle, l'équation d'état est résolue au zéro machine (10^{-14}), et des différences divisées d'ordre deux sont appliquées. Ceci ne produit pas la meilleure précision pour toutes les composantes, mais pour chacune de ces composantes, rechercher un choix adéquat de la taille de la perturbation permet d'obtenir au moins 6 chiffres identiques à ceux des résultats de la DA. Le coût d'une telle évaluation est beaucoup plus grand que $2n$ fois la résolution du système d'état pour n paramètres.

Nous soulignons qu'une validation plus précise peut aussi être réalisée en générant une analyse de sensibilité par un mode direct avec TAPENADE [Tropics, 2001].

Puisque l'approche adjointe peut être construite dans un contexte continu (i.e. non discrétisé), il est raisonnable d'étudier la convergence du gradient discret vers le gradient continu. Cette étude est décrite dans la Fig.1.9. Nous observons que la plupart des paramètres sensibles correspondent aux positions de la ligne sonique et des chocs.

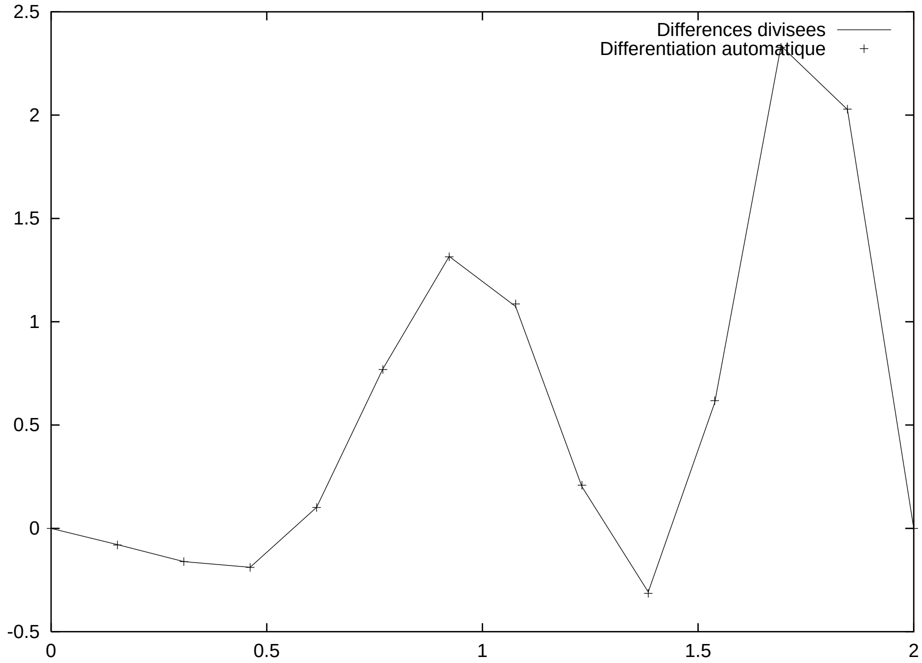

FIG. 1.8 – *Représentation du gradient de la fonctionnelle objectif comme une fonction d'une variable horizontale indépendante. La ligne continue est calculée avec la méthode d'adjoint exacte proposée. Les 14 croix sont obtenus en appliquant successivement des différences divisées aux 14 inconnues de la fonctionnelle objectif. Les calculs sont menés pour un maillage de 400 nœuds.*

Cette validation est faite pour un écoulement subsonique et pour un écoulement supersonique, afin de faire une étude plus complète du code différentié. Nous validons le gradient $j'(\gamma)$, où γ est défini par 14 points de contrôle $\gamma_i = \gamma(x_i)$ avec $i = 1, 14$. Les gradients sont triviaux pour $i = 1$ et $i = 14$, et ces valeurs ne sont pas présentés dans nos graphiques. Dans chaque cas et pour chaque point de contrôle, la table 1.2 montre l'erreur relative entre les valeurs de $j'(\gamma)$ calculées par notre algorithme et les valeurs correspondantes évaluées par Différences Finies (DF). La correspondance est bonne compte tenu de l'imprécision habituelle des Différences Finies. Notons que ces valeurs des DF sont plus coûteuses à calculer, essentiellement pour deux raisons :

- Deux évaluations de la fonction coût j sont nécessaires pour chaque paramètre de contrôle. Ceci fait un total de 28 évaluations.
- Chacune de ces 28 évaluations requiert de nombreuses éxécutions de j afin de sélectionner un ϵ convenable.

1.5.5 Sensibilité aux limiteurs TVD

Les résultats précédents furent obtenus sans utiliser les limiteurs TVD du noyau CFD. Nous terminons notre étude du gradient par une étude de l'impact de l'utilisation de ces limiteurs. Nous restreignons nos expériences à un maillage, avec 1240 noeuds. À ce niveau

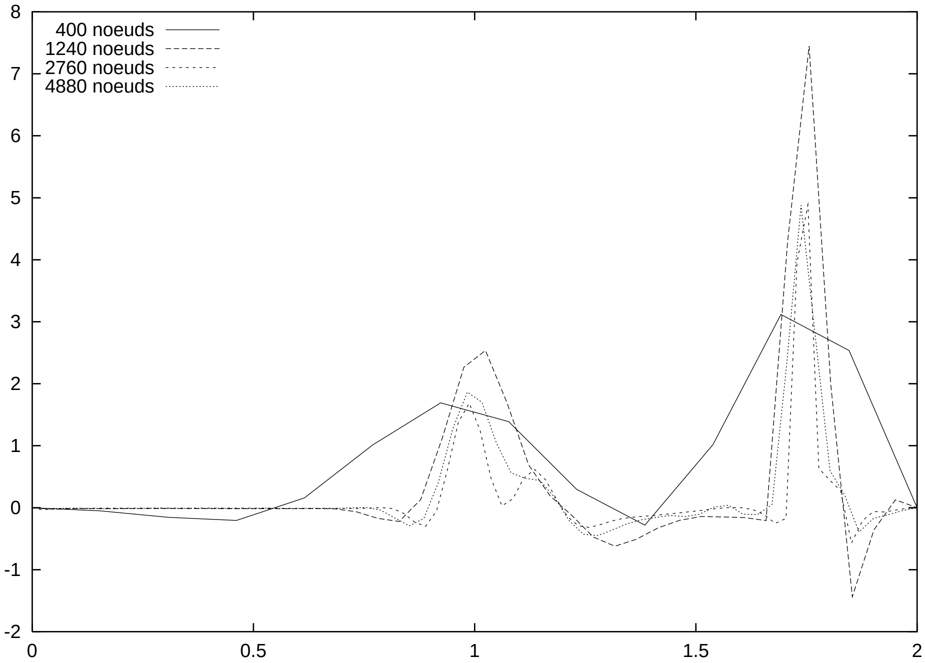

FIG. 1.9 – *Convergence du gradient vers la limite continue pour les quatre maillages de 400, 1240, 2760 et 4880 noeuds*

de finesse, les limiteurs ont une influence moyenne sur les variables d'état : les valeurs en entrée et en sortie sont modifiées légèrement, le choc bouge un peu en bas et les petits dépassements disparaissent, voir la coupe horizontale du nombre de Mach sur la Fig.1.10.

La résolution de l'état adjoint est plus perturbée par ce choix, puisque des valeurs quasi-constantes montrent de grandes différences, voir la coupe horizontale de la Fig. 1.11.

x_i :	0.154	0.308	0.461	0.615	0.769	0.923
cas subsonique :	1.0e-7	1.7e-7	2.3e-7	3.2e-7	1.4e-7	5.4e-9
cas supersonique :	4.0e-6	1.3e-5	1.4e-5	9.7e-8	1.1e-7	1.6e-7
x_i :	1.077	1.231	1.385	1.538	1.692	1.846
cas subsonique :	1.3e-8	1.5e-8	2.0e-5	8.2e-9	1.4e-7	4.0e-8
cas supersonique :	2.4e-7	3.0e-8	8.1e-8	6.3e-8	3.7e-8	1.1e-8

TAB. 1.2 – *Erreurs relatives entre les gradients calculés par Différentiation Automatique et les gradients calculés par Différences Divisées ($\epsilon = 10^{-8}$).*

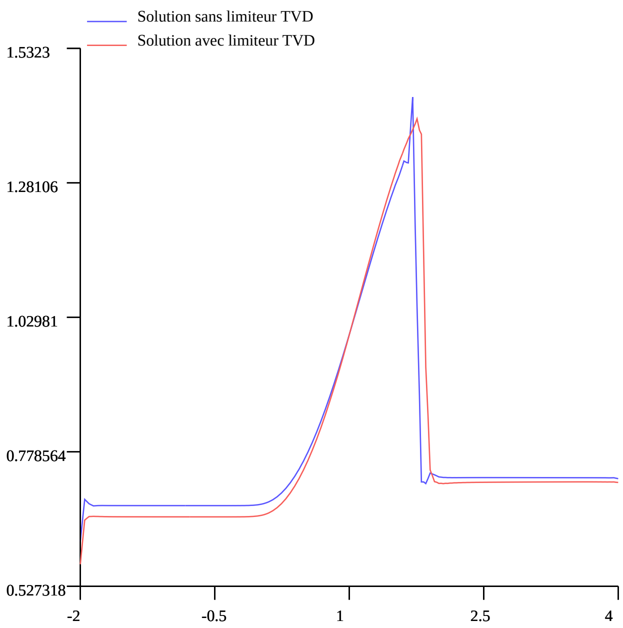

FIG. 1.10 – *Solution de l'équation de l'état, coupe horizontale du nombre de Mach, avec et sans limiteurs TVD (maillage de 1240 noeuds)*

Dans la Fig.1.12, nous présentons l'impact final des limiteurs sur les composantes du gradient. Nous observons que le comportement global est lissé. Au niveau du choc direct de l'état (abscisse voisine de 1.7), le Dirac du gradient ne bouge pas, mais est lissé et a un pic moins élevé. Au point sonique (abscisse voisine de 1.), nous avons une sorte de choc pour l'état adjoint. Les limiteurs apportent de nouveau une sorte de lissage du gradient plus difficile à interpréter, puisque les limiteurs ne réagissent pas dans cette région. Sur les autres régions, les valeurs sont inchangées tandis que l'adjoint montre des valeurs constantes mais différentes. Ceci peut être expliqué par la seule implication des dérivées spatiales des états dans les formulations du gradient. Par conséquent des variations en moyenne de l'état et de l'adjoint n'ont pas de conséquence sur le gradient.

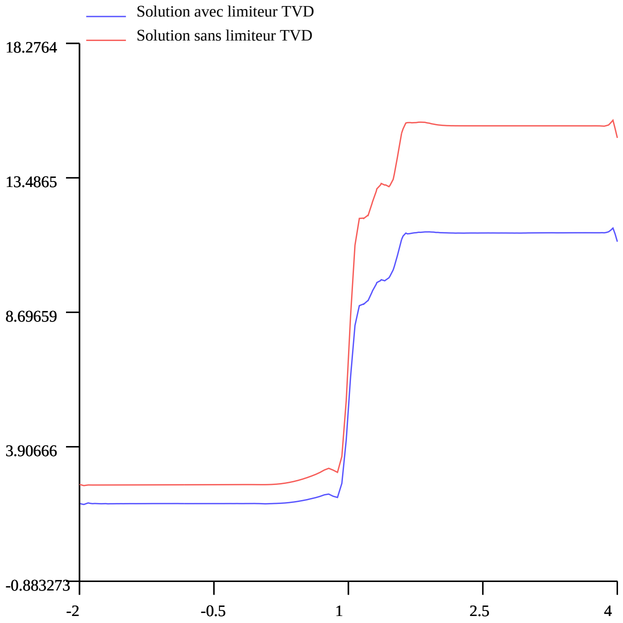

FIG. 1.11 – *Solution de l'équation adjointe, coupe horizontale de la deuxième composante, avec limiteurs TVD (courbe du bas) et sans limiteurs TVD (maillage de 1240 noeuds)*

1.5.6 Adjoint discret contre adjoint continu

Comme nous l'avons signalé dans la section 1.1, une approche alternative utilise un adjoint continu (*i.e.* non discrétisé). Cette approche suppose qu'une discrétisation du gradient continu est assez proche du gradient discret. Il est alors intéressant d'étudier la convergence de nos gradients discrets vers le gradient continu. Les résultats sont montrés dans la figure 1.9.

Nous observons que les gradients discrets varient substantiellement, plus précisément aux endroits qui correspondent aux lignes soniques et aux chocs dans l'état W. *A priori*, pour optimiser une fonctionnelle discrète correspondant à une discrétisation donnée, il est plus sûr d'utiliser un gradient discret basé sur la même discrétisation. Strictement parlant, utiliser un gradient continu mène seulement à une approximation du gradient discret, et la figure 1.9 montre que cette approximation peut être très grossière. Par conséquent l'utilisation d'adjoint continu est dangereuse notamment si la finesse du maillage est insuffisante et peut dégrader les résultats de l'optimisation.

1.6 Conclusion

Les méthodes avec état adjoint donnent un nouvel essort aux techniques d'optimisation de formes. Nous décrivons dans ce chapitre une méthode de calcul des gradients basés sur des adjoints pour un problème classique d'optimisation de formes. Cette méthode a essentiellement deux niveaux.

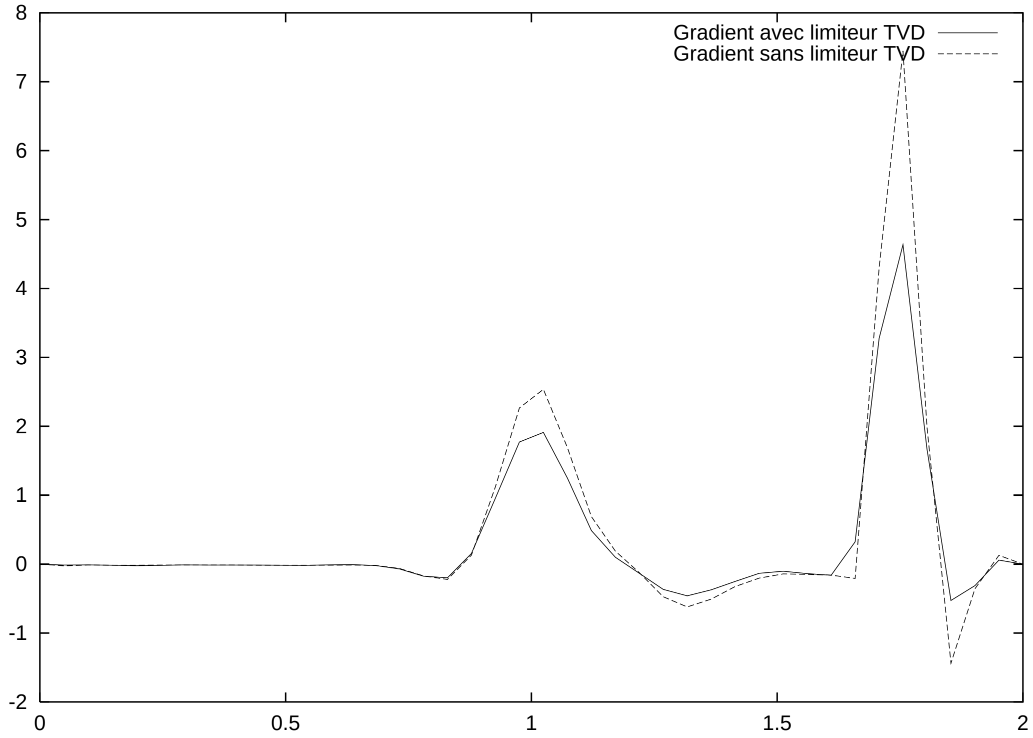

FIG. 1.12 – *Gradient, avec et sans limiteurs TVD (maillage de 1240 noeuds)*

Au niveau supérieur, nous expliquons comment déduire un gradient exact basé sur un état adjoint sans stocker le Jacobien explicitement. Elle est valide même pour des systèmes aussi complexes que ceux apparaissant en Mécanique des Fluides compressibles. Pour ces systèmes, les matrices jacobiennes exactes sont souvent trop grandes pour pouvoir être stockées. Ici, nous n'utilisons pas le jacobien (tranposé), mais seulement le résidu du système adjoint.

Au niveau inférieur, nous expliquons comment obtenir le résidu adjoint en utilisant le mode adjoint de la Différentiation Automatique. Dans le cas des systèmes stationnaires comme ceux utilisés en Mécanique Continue complexe, le programme original contient un grand nombre de boucles à itérations indépendantes, telles que les fréquentes boucles d'assemblage. Pour ces boucles, nous proposons un mode adjoint de la DA amélioré, qui utilise beaucoup moins d'espace mémoire. Nous décrivons le processus complet menant à l'assemblage du résidu de l'adjoint, et à l'assemblage des autres dérivées nécessitées par le niveau supérieur.

Le programme différentié résultant calcule complètement le gradient, avec des performances satisfaisante. Les gradients obtenus sont validés par comparaison avec des différences divisées.

Dans notre stratégie, la Différentiation Automatique est appliquée seulement aux programmes qui traitent de l'assemblage des équations et de la fonction coût et jamais aux algorithmes de résolution. En pratique, l'utilisateur doit encore sélectionner et utiliser un solveur approprié pour l'équation adjointe. Dans ce travail, nous avons choisi un algorithme de pur point fixe avec Defect Correction. Dans le futur, pour résoudre le système adjoint, nous pensons que de meilleurs algorithmes peuvent être construits. Ils pourraient être par exemple Quasi-Newton, et permettraient des extensions efficaces de décomposition de domaines. Pour résoudre le système d'optimalité, nous pensons à deux directions intéressantes. L'une est l'adaptation des algorithmes d'optimisation SQP aux systèmes CFD de grande taille. L'autre est l'application des itérations simultanées [Dadone and Grossman, 2000] ou one-shot [Ta'asan *et al.*, 1992] au système d'optimalité. Dans les deux cas, nous avons besoin du résidu adjoint, que nous calculons avec la méthode développée ici.

Remerciements

Ce travail a été partiellement financé par le projet européen Aeroshape.

Deuxième partie

Méthodes d'optimisation adaptées au contrôle optimal en aérodynamique

1

Un problème d'optimisation de formes

Sommaire

1.1 Un problème modèle : réduction du bang sonique

1.1.1 Introduction

Le but de ce premier chapitre est d'introduire un exemple concret de problème d'Optimum Design. Les ingrédients d'un tel problème sont :

- un **ensemble de paramètres**, les paramètres de *design*, qui seront, en terminologie du Contrôle Optimal, les paramètres de contrôle. Ils seront, en terme d'optimisation, la variable d'optimisation réduite, c'est à dire celle obtenue après élimination d'une partie des variables grâce à la prise en compte des contraintes égalité.

- un **système paramétré par ce contrôle**, c'est à dire défini de manière unique en fonction du contrôle grâce au système des équations d'*état*. Pour fixer les idées, ce système sera un système d'équations aux dérivées partielles, et, pour simplifier un système indépendant du temps ("stationnaire"). En effet, le contrôle envisagé ici est à distinguer du Contrôle Optimal historique, qui traite de commandes et de systèmes instationnaires. Dans le langage de l'optimisation, la variable d'état, inconnue de l'équation d'état, fera partie des variables d'optimisation avant "réduction" et l'équation d'état sera une contrainte "égalité".

- une **fonctionnelle "coût"** dépendant des contrôles et états.

Dans ce mémoire, nous n'aborderons pas les problèmes liés aux contraintes sur le contrôle ou sur l'état.

Ce contexte est bien adapté à la conception optimale d'un objet manufacturé dans la mesure où on pourra évaluer la qualité de l'objet à travers la résolution d'une EDP.

Dans l'application que nous allons considérer dans cette seconde partie, le contrôle sera la forme géométrique 3D extérieure d'un avion, l'équation d'état sera un modèle de l'aérodynamique, et la fonctionnelle coût mesurera les performances aérodynamiques ainsi qu'une partie de la nuisance acoustique de l'avion.

Il s'agit en effet de rechercher une forme "optimale" pour la réduction du bang sonique provoqué par un avion supersonique durant sa croisière à nombre de Mach élevé.

1.1.2 Modèlisation du bang sonique

Le bang sonique pose un problème de modélisation. En effet, les différentes ondes de choc émises par l'avion se propagent jusq'au sol, au niveau duquel on cherche à réduire leur intensité. Cf. la figure 1.1

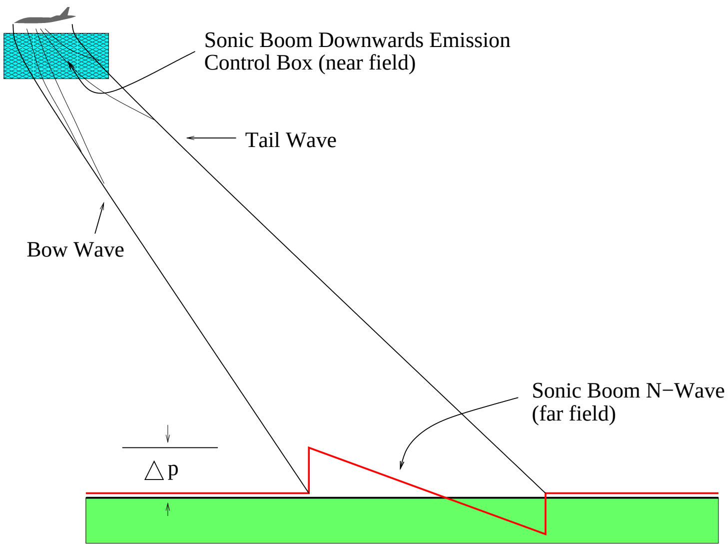

FIG. 1.1 – *Bang sonique : propagation de la perturbation jusqu'au sol. Boite de contrôle* Ω^B.

Cette propagation est *grosso modo* régie par les équations d'Euler 3D standard de la Mécanique des fluides. Cependant, la résolution **jusqu'au sol** de ces équations est hors d'atteinte des calculateurs actuels. Il faut appliquer un modèle moins coûteux.

Cette question est abordée dans [Maglieri and Plotkin, 1991] à partir des travaux pionniers de Witham et dans [Seebas and Argrow, 1998]. L'idée centrale de Witham est de propager la perturbation de manière bidimensionnelle dans un plan vertical de symétrie

de l'avion.
Ceci suppose un raccord avec l'écoulement tridimensionnel proche de l'avion et en particulier que celui-ci soit calculé avec précision et dans un domaine assez grand pour que la structure de choc au niveau du raccord soit quasi-axisymétrique.
Alors le calcul bidimensionnel "jusqu'au sol" de la "signature du bang sonique" est un calcul séparé de l'évaluation CFD. Cette stratégie est suivie par divers auteurs et nous citerons deux travaux représentatifs. Dans [Farhat *et al.*, 2002], il est proposé un schéma d'optimisation paramétrique pour le bang sonique. La géométrie est décrite par les positions de plusieurs élements de portance (canards,..) et de la pointe du nez de l'appareil. La signature est évalué à l'aide d'un modèle 2D du type précédent et un système adjoint est résolu pour calculer le gradient de la fonctionnelle coût. Celle-ci met en œuvre le critère de la minimisation de la première augmentation de pression au sol (*initial shock pressure rise, ISPR)*.
L'autre voie est de paramétrer finement la géométrie, typiquement en déplaçant le nœuds du maillage de peau. Elle est utilisée dans [Nadarajah *et al.*, 2002; Alonso *et al.*, 2002]. Dans [Mohammadi and Pironneau, 2001], Mohammadi désigne cette approche par l'expression **"CAD-free"**, sans CAO, par opposition à l'approche **"CAD-based"** utilisée notamment dans [Farhat *et al.*, 2002].
Dans [Mohammadi, 2002], le bang est réduit de manière indirecte par l'introduction d'une fonction de trainée spéciale. Cette fonction permet l'usage d'une approche de type "gradient simplifié", c'est à dire sans calcul d'adjoint.

Dans notre travail, nous adoptons une modélisation simplifiée assez différente, introduite par l'équipe Tropics dans [Vazquez *et al.*, to appear]. Nous en rappellons maintenant les principales caractéristiques.

L'idée de base est que la signature au sol prend sa source et son énergie dans la perturbation émise par l'avion, l'**émission vers le sol du bang sonique, ESBS**. C'est cette émission que nous allons minimiser. Cette approche néglige certains effets de focalisation du bang survenant dans la propagation. De nombreux calculs montrent néanmoins sa pertinence. Nous en présentons un en figure 1.2. L'utilisation de cette approche peut aussi être validée après la phase d'optimisation.

L'ESBS peut être évaluée comme la norme du gradient de pression dans une "boite d'observation" (figure 1.1) située sous l'aéronef. La fonctionnelle coût prenant aussi en compte les performances aérodynamques à travers des termes supplémentaires.

1.1.3 Fonctionnelle coût

Les différentes géométries sont paramétrées par le déplacement de la forme de l'avion qui en tout point de la forme initiale est glissée d'une longueur algébrique γ le long du vecteur normal à la forme.

Une fois la nouvelle forme définie, l'écoulement correspondant est évalué dans le domaine de calcul initial grâce à une technique de **transpiration**, qui simule le déplacement de la frontière par une condition au bord spéciale. Nous renvoyons à [Vazquez *et al.*, to

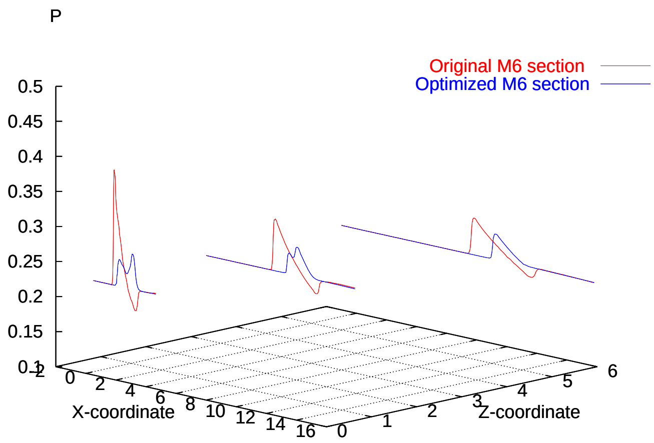

FIG. 1.2 – Transformation du profil de pression depuis la zone sous l'avion jusqu'à une distance de 6 fois la longueur de l'avion.

appear] pour une description détaillée de cette méthode.

La fonctionnelle va donc contenir un terme mesurant l'ESBS comme l'intégrale du gradient de pression sur B (cf. figure 1.1). On inclut deux autres termes relatifs aux performances aérodynamiques comme suit :

$$j(\gamma) = \alpha_1 (C_D(\gamma) - C_D^{cible})^2 + \alpha_2 (C_L(\gamma) - C_L^{cible})^2 + \alpha_3 \int_{\Omega^B} |\nabla P(\gamma)|^2 \ \ dV \qquad (1.1)$$

où α_1, α_2 et α_3 sont des constantes pondérant les trois critères constituant la fonctionnelle $j(\gamma)$. Les quantités $C_D(\gamma)$, $C_L(\gamma)$, $P(\gamma)$ sont déduites de l'écoulement.

Les différentes géométries sont paramétrées par le déplacement de la forme de l'avion qui en tout point de la forme initiale est glissée d'une longueur algébrique γ le long du vecteur normal à la forme.

Une fois la nouvelle forme définie, l'écoulement correspondant est évalué dans le domaine de calcul initial grâce à une technique de **transpiration**, qui simule le déplacement de la frontière par une condition au bord spéciale. Nous renvoyons à [Vazquez *et al.*, to appear] pour une description détaillée de cette méthode.

1.1.4 Équation d'état discrète

Ces différentes quantités dépendent de la forme γ à travers la résolution des équations d'Euler.

La résolution des équations d'Euler reposent sur les techniques suivantes :
- a : une formulation centrée-sommet,
- b : la méthode MUSCL,
- c : l'utilisation d'une formulation particulière "par arête" avec des éléments amonts.
Le schéma MUSCL a été introduit par van Leer dans une série de papiers assez populaires,

cf. notamment [Van Leer, 1975].

Les formulations mixtes éléments-volumes finis ([Fezoui and Stoufflet, 1989]) sont des formulations avec nœuds aux sommets des tétraèdres qui combinent éléments finis P_1-Galerkin et avec des volumes finis sur les cellules duales construites autour des sommets. Pour tout sommet interne i, la cellule duale C_i est construite autour de i et limitée par des facettes triangulaires. Chaque facette a pour sommet un milieu d'arête, un centre de gravité de face, et un centre de gravité de tétraèdre.

Les flux convectifs sont discrétisés par volumes finis appliqués à ces cellules, c'est à dire à partir de flux évalués sur toute frontière commune à deux cellules :

$$\sum_{j \in V(i)} \int_{\partial C_{ij}} \mathcal{F}(W, \vec{n})\, d\sigma \quad , \tag{1.2}$$

où $V(i)$ est l'ensemble des sommets voisins du sommet i, ∂C_{ij} est la frontière commune entre C_i et C_j, et $\vec{n}$ la normale extérieure à C_i. Dans la phase volumes-finis les inconnues sont représentées par des fonctions P_1 dans chaque cellules, discontinues aux frontières des cellules. L'intégration le long de ces discontinuités est réalisée à l'aide des valeurs intermédiaires fournies par des solveurs de Riemann. En pratique, nous utiliserons la décomposition de flux de Roe :

$$\int_{\partial C_{ij}} \mathcal{F}(W, \vec{n})\, d\sigma \simeq \Phi^R(W_i,\ W_j,\ \vec{\nu_{ij}}) \quad , \tag{1.3}$$

où $\vec{\nu}_{ij}$ est défini par :

$$\nu_{ij} = \int_{\partial C_{ij}} \vec{n} d\sigma \quad . \tag{1.4}$$

Ces flux de Roe s'écrivent comme suit :

$$\Phi^R(W_i,\ W_j,\ \vec{\nu_{ij}}) = \frac{\mathcal{F}(W_i,\ \vec{\nu_{ij}}) + \mathcal{F}(W_j,\ \vec{\nu_{ij}})}{2} - d^R(W_i,\ W_j,\ \vec{\nu_{ij}}) \quad , \tag{1.5}$$

$$d^R(W_i, W_j, \vec{\nu_{ij}}) = |\mathcal{R}(W_i, W_j, \nu_{ij})| \ \frac{W_j - W_i}{2} \quad , \tag{1.6}$$

où W_i(resp.W_j) est le vecteur écoulement à droite (resp. à gauche) et $\mathcal{R}$ la matrice de Roe :

$$\mathcal{R}(W_i, W_j, \nu_{ij}) = \frac{\partial \mathcal{F}}{\partial W}\left(\widehat{W}, \nu_{ij}\right) \quad . \tag{1.7}$$

Le terme $\hat{W}$ est la moyenne de Roe des vecteurs à gauche et à droite W_i et W_j. Si les vecteurs à gauche et à droite sont les valeurs aux sommets des inconnues, le schéma est seulement précis à l'ordre un. Pour transformer ce schéma en un schéma précis à l'ordre deux, on transpose la méthode MUSCL à ce contexte tétraédrique. Le solveur de Roe est appliqué non plus à des valeurs nodales mais à des extrapolations W_{ij} et W_{ji} sur l'interface entre C_i et C_j :

$$\int_{\partial C_{ij}} \mathcal{F}(W, \vec{n})\, d\sigma \simeq \Phi^R (W_{ij},\ W_{ji},\ \vec{\nu_{ij}}) \quad , \tag{1.8}$$

$$W_{ij} = W_i + \frac{1}{2} \left(\vec{\nabla} W\right)_{ij} \cdot \vec{ij} \tag{1.9}$$

$$W_{ji} = W_j - \frac{1}{2} \left(\vec{\nabla} W\right)_{ji} \cdot \vec{ij} \tag{1.10}$$

Les gradients approchés $\vec{\nabla} W_{ij}$ et $\vec{\nabla} W_{ji}$ sont obtenus en utilisant les gradients P_1-Galerkin sur trois tétraèdres différents.

Pour tout tétraèdre T^{ij} ayant ij pour arête, on calcule $\left(\vec{\nabla} W\right)_{ij}^{C} = \vec{\nabla} W|_{T^{ij}}$ ("C" au sens de "centré") qui vérifie $\left(\vec{\nabla} W\right)_{ij}^{C} \cdot \vec{ij} = W_j - W_i$.

On appelle tétraèdre amont (resp. aval) le tétraèdre (T_{ji}) (resp. (T_{ij}) associé à l'arète $\vec{ij}$ qui contient le prolongement du segment ij (resp. ji) (fig. 1.3).

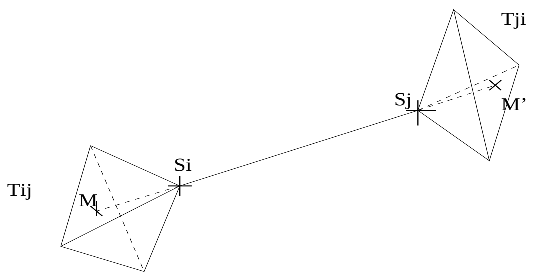

FIG. 1.3 – Tétrahèdres amont et aval autour d'une arête S_iS_j.

Les gradients amont $\left(\vec{\nabla} W\right)_{ij}^{U}$ et aval $\left(\vec{\nabla} W\right)_{ij}^{D}$ sont évalués respectivement sur les éléments amont (T_{ji}) et aval (T_{ij}).

En particulier, $\left(\vec{\nabla} W\right)_{ij}^{U} = \vec{\nabla} W|_{T_{ji}}$ et $\left(\vec{\nabla} W\right)_{ij}^{D} = \vec{\nabla} W|_{T_{ij}}$ où $\vec{\nabla} W|_T = \sum_{k \in T} W_k \vec{\nabla} \Phi_k|_T$ est le gradient en interpolation Galerkin sur le tétraèdre T.

Dans le cas d'un maillage cartésien (en tétraèdre, parfaitement périodique) et d'un modèle d'advection linéaire, l'option $\beta = 1/3$ permet d'avoir une précision à l'ordre trois.

1.1.5 Algorithme de résolution

Le système précédent est résolu par une méthode pseudo-instationnaire, avançant en temps suivant l'équation :

$$W_t + \Psi(W) = 0 \tag{1.11}$$

L'avancement en temps est réalisé avec une formulation implicite linéarisée préconditionnée par un Jacobien A_1 à l'ordre un spatial :

$$(M + \Delta t A_1(W^n))(W^{n+1} - W^n) + \Delta t \Psi(W^n) = 0 \tag{1.12}$$

la matrice M est une matrice de masse en volumes finis, c'est à dire diagonale et constituée des aires des cellules.

La résolution du système linéaire (1.12) permettant de calculer la différence $(W^{n+1} - W^n)$ est réalisée par une méthode de Jacobi par bloc (5×5).

2

Proposition d'une méthode One-Shot en Programmation Quadratique Successive

Sommaire

Le contenu de ce chapitre a été soumis à la revue *Mathematical Programming*.

2.1 Introduction

Les outils applicables à l'optimisation différentiable ont été significativement perfectionnés durant les deux dernières décennies, qui ont vu le remplacement des méthodes de gradient par la Programmation Quadratique Successive (SQP).
La cible initiale des méthodes SQP était la minimisation d'une fonctionnelle sous des contraintes égalités. Un cas particulier d'intérêt est celui ou les contraintes permettent d'éliminer une partie des inconnues. C'est le cas des problèmes de Contrôle Optimal comprenant une équation d'état. Dans ce cas les méthodes SQP sont bien adaptées à la résolution des équation de Karush-Kuhn-Tucker (KKT), en l'occurence l'équation d'état, l'équation de l'état adjoint, et la condition d'optimalité. En même temps, l'approche SQP conserve l'idée de faire décroître une fonctionnelle (proche de la fonctionnelle originale). Un SQP typique applique sucessivement deux étapes. Une étape de restauration fournit une nouvelle valeur pour la variable d'état, satisfaisant mieux la contrainte égalité. Une étape de minimisation met à jour le contrôle de manière à faire décroître la fonctionnelle à minimiser. Ces deux étapes, en accord avec la philosophie "successivement quadratique" sont de type Newton, avec un calcul de Hessien, ou plus fréquemment de type quasi-Newton, basé, la plupart du temps sur une formule BFGS, voir [Nocedal and Wright, 1999; Bonnans *et al.*, 2002].

La convergence d'une méthode de Newton est conditionnée par un démarrage dans le bassin d'attraction. Un important progrès en robustesse fut apporté aux méthodes SQP avec l'adjonction de mécanisme de contrôle tels que les régions de confiance.

L'algorithme de Byrd et Omojokun est un élément représentatif combinant ces méthodes, cf. [Nocedal and Wright, 1999; Bonnans *et al.*, 2002]. Il comprend les phases suivantes ainsi qu'une méthode de région de confiance permettant de résoudre progressivement équation d'état et condition d'optimalité.

La suite de notre introduction focalise sur cet algorithme. Le but est de montrer pourquoi on devrait le transformer pour l'appliquer plus efficacement à une classe particulière de problèmes. Les problèmes montrés du doigt existent aussi dans les autres algorithmes SQP.

Un avantage de SQP est que la résolution complète de l'équation d'état n'est pas nécessaire à chaque itération d'optimisation. Au contraire, l'état peut être avance progressivement vers une limite ne satisfaisant l'équation d'état qu'à convergence de l'optimiseur. Si cette équation d'état est très coûteuse, l'avantage peut être important par rapport à la génération précédente des méthodes de gradient.
Cependant, l'itération usuellement appliquée dans les méthodes SQP pour avancer l'état est un pas Newton, incluant donc la résolution complète d'un système linéarisé.
Plus grave, la convergence du processus dépend de la bonne résolution du système adjoint, linéaire, et de même taille.
Un dernier point et non des moindres, la dernière phase d'assemblage du pas de minimisation comprend la résolution d'un troisième système linéaire.
Au total, une méthode SQP standard devra résoudre à chaque itération principale trois systèmes linéarisés issus de l'équation d'état. Cette méthode n'est pas adaptée au cas de systèmes d'état de très grande dimension comme on en rencontre dans les applications industrielles de l'optimum design, typiquement en aéronautique.

Nous en résumerons les raisons comme suit :

- les méthodes de Newton ne sont pas les melleures méthodes pour résoudre les grands systèmes nonlinéaires raides comme ceux issus de la Mécanique des Fluides. Dans ce dernier cas les solveurs pseudo-instationnaires semblent actuellement les plus satisfaisants (efficacité et robustesse) et sont de plus disponibles dans les codes d'analyse.

- les systèmes linéarisés obtenus par différentiation exacte ne sont pas toujours aisés à résoudre. En CFD, on leur préfère des systèmes moins coûteux à assembler, à diagonale rendue dominante via divers moyens, dont le décentrage et l'adjonction d'une dérivée en temps. On pourrait toutefois imaginer d'utiliser un système jacobien simplifié en lieu et place des trois systèmes identifiés plus haut. Chaque itération d'optimisation resterait encore trop coûteuse avec de plus un risque de dégradation de la convergence globale de l'algorithme

- Le cas du système adjoint pose de plus quelques difficultés supplémentaires. Si la matrice ne peut être stockée, le coût de son assemblage répété devient dominant. Le linéarisé peut être plus complexe que le nonlinéaire dont il est déduit par différentiation. Signalons à nouveau le cas de la CFD dont les discrétisations sophistiquées (limiteurs TVD, solveurs de Riemann) sont particulièrement non-linéaires. Ce bilan peut encore s'aggraver quand la différentiation n'est pas optimalement implémentée, comme cela peut arriver quand le code dérivé a été généré par un outil de Différentiation Automatique.

Ces remarques tendent à montrer que dans des applications très intensives comme celles que nous envisageons, la résolution des trois systèmes linéaires précédents est aussi coûteuse que celle de l'équation d'état.

L'impact de ces résolutions sur le coût global de l'algorithme peut être évalué à l'aide du modèle suivant : en supposant (de manière très optimiste) que la complexité de la résolution du système linéaire est le nombre d'inconnues N, et que le nombre d'itérations globales est de l'ordre du nombre n d'inconnues de contrôle. Alors la complexité est le produit $N \times n$, c'est à dire quadratique. Elle devient notamment trop grande quand N est grand et n pas assez petit. Il est alors naturel de chercher à remplacer cet algorithme par un algorithme qui résoudra vraiment **simultanément** les trois équations KKT.

Shlomo Ta'asan a introduit ce type d'algorithme, baptisé "one-shot" dans un travail pionnier ([Ta'asan, 1991]). Un algorithme de comportement similaire a été introduit dans [Attouch *et al.*, 2000]. Divers chercheurs traitant d'applications particulières ont proposé des idées voisines, voir par exemple [Dadone *et al.*, 1999], [Dadone and Grossman, 2000], [Held and Dervieux, 2002].

Ces travaux ont mis en évidence d'importants gains en efficacité mais doivent être revisités sous l'angle de la théorie de l'optimisation. En effet, en pratique, ces méthodes sont insuffisamment robustes. Elles convergent seulement après règlage de certains paramètres. Il est donc nécessaire d'introduire dans ces méthodes des heuristiques assurant une bonne robustesse et si possible des mécanismes assurant une convergence superlinéaire. Les ingrédients nécessaires à ces améliorations sont disponibles dans la panoplie des méthodes et théories actuelles de l'optimisation.

Le but de ce chapitre est de contribuer à une nouvelle génération d'algorithmes proche de la classe SQP qui incorporent quelques unes des propriétés modernes de SQP tout en conservant l'efficacité de l'approche one-shot.

Dans cette perspective, nous considérerons d'abord l'algorithm SQP de Byrd-Omojokun.

Il sera utile :
- d'une part pour des comparaisons en efficacité et en robustesse,
- d'autre part pour présenter le nouvel algorithme comme le résultat de modifications du premier.

Pour appliquer le nouvel algorithme, nous aurons besoin d'une **hypothèse de base**, celle d'avoir à notre disposition un solveur itératif **progressif** pour les systèmes d'état et d'adjoint. Par progressif, nous entendons qu'un pas de ce solveur est beaucoup moins coûteux qu'une résolution linéaire, mais plutôt de l'ordre d'une évaluation de résidu de l'équation à résoudre. De tels solveurs sont disponibles dans de nombreuses applications et particulièrement en Aérodynamique.

Comme **principe de base** pour contrôler la convergence, nous vérifierons la décroissance d'une fonctionnelle. Une deuxième idée dominante sera de considérer le système adjoint comme une deuxième contrainte égalité, au même titre que l'équation d'état. Les deux solutions seront avancées lors d'une première phase de restauration qui, en contraste avec SQP, ne pourra pas être invalidée, de sorte que ces système avanceront continûment vers leur convergence itérative. Une conséquence de ceci sera la satisfaction d'une condition de type Wolfe, garantissant à son tour la qualité de la direction de descente.

L'efficacité du nouvel algorthme sera verifiée et comparée avec celle de l'algorithme de Byrd-Omojokun algorithm sur quelques exemples. Une première famille d'exemples est inspirée d'un probléme d'identification par moindres carrés. On discutera ensuite d'une application de la Mécanique des fluides plus difficile, la solution d'un problème de conception optimale de forme.

Le plan est le suivant :
- Section 2 : Description des caractéristiques de l'algorithme de Byrd-Omojokun,
- Section 3 : Algorithm one-shot SQP
- Section 4 : Application à un problème modèle
- Section 5 : Application à la réduction du bang sonique
- Section 6 : Conclusions directions futures.

2.2 Une version de l'algorithme de Byrd-Omojokun

L'algorithme proposé par Byrd [Byrd, 1987] et Omojokun[Omojokun, 1991] est un bon représentant des méthodes SQP avec Région de Confiance. Cet algorithme peut être considéré comme la suite de ceux proposés par Vardi [Vardi, 1985], Celis, Dennis and Tapia [Celis *et al.*, 1985], Gilbert [Gilbert, 1997], Byrd, Gilbert et Nocedal [Byrd *et al.*, 2000].

2.2.1 Principes généraux

Nous allons utiliser la version proposée dans [Lalee *et al.*, 1998] réécrite dans un contexte **Hessien-Réduit**. Nous considérons donc le problème de minimisation :

$$\begin{aligned} &\min \quad J(u,Y) \\ \text{sous la contrainte} \quad &\Psi(u,Y) = 0 \ . \end{aligned} \tag{2.1}$$

Le minimum est recherché par rapport à la variable x :

$$x = (u, Y) \ .$$

Nous supposons pour un u donné que la contrainte égalité : $\Psi(u, Y) = 0$ possède une seule solution Y. Dans le même ordre d'idée, nous supposons aussi que la matrice jacobienne

$$A = \frac{\partial \Psi}{\partial Y} \tag{2.2}$$

est toujours inversible.
Partant de l'itéré $x_k = (u_k, Y_k)$, nous définissons

$$\Psi_k \ = \ \Psi(x_k) \text{ et } A_k \ = \ \frac{\partial \Psi}{\partial Y}(x_k).$$

Le multiplicateur de Lagrange, Π_k, est une solution de

$$\Pi_k \ = \ \left(\frac{\partial \Psi}{\partial Y}(x_k)\right)^{-T} \frac{\partial J}{\partial Y}(x_k) \ . \tag{2.3}$$

Nous notons par $g_k \ = \ g(x_k, \Pi_k)$:

$$g(x_k, \Pi_k) \ = \ \frac{\partial J}{\partial u}(x_k) - \ < \Pi_k \ , \ \frac{\partial \Psi}{\partial u}(x_k) > \tag{2.4}$$

Méthode de région de confiance

La méthode de Région de Confiance est expliquée en détail dans ([Nocedal and Wright, 1999], 18.9) et [Bonnans *et al.*, 2002]. Nous rappelons ici quelques informations utiles à la compréhension de ce travail.

Les méthodes de recherche linéaire et les méthode de région de confiance génèrent toutes deux des pas à l'aide d'un modèle quadratique de la fonction objectif, elles utilisent cependant ce modèle de différentes façons. Les méthodes de recherche linéaire l'utilisent pour générer une direction de recherche, puis s'attachent à trouver une longueur de pas α le long de cette direction. Les méthode de région de confiance définissent une région autour de l'itéré courant à l'intérieur de laquelle elles considèrent que le modèle est une représentation adéquate de la fonction objectif, puis choisissent la direction et la longueur du pas simultanément. Si le pas n'est pas acceptable, elles réduisent la taille de la région puis cherche un nouveau minimiseur. En général la direction du pas change chaque fois que la taille de la région est altérée.

Phase de restauration de l'état

Etant donné une Région de Confiance de rayon Δ_k et un paramètre de relaxation $\xi \in (0, 1)$, nous calculons un pas v_k qui résout le sous-problème vertical ou normal :

$$\min_{v \in I\!R^n} \ \|A_k v + \Psi_k\| \tag{2.5}$$
$$\text{sous la condition} \quad \|v\| \leq \xi \Delta_k$$

Ce problème est généralement résolu à l'aide d'un pas de Newton ($v^n = -A_k^{-1}\Psi_k$) corrigé afin de satisfaire la contrainte de la Région de Confiance. Cette correction peut être faite avec une méthode *"dogleg"* , on pourra consulter sur ce point [Nocedal and Wright, 1999; Bonnans *et al.*, 2002]. Cette combinaison de Région de Confiance et de méthodes *"dogleg"* est un moyen de contrôler la convergence de l'itération de Newton pour l'équation d'état. Le problème vertical linéaire peut avoir plusieurs solutions, mais Lalee, Nocedal, Platenga ont montré dans [Lalee *et al.*, 1998] que ce problème a toujours une unique solution dans l'espace image de A_k. Ceci permet une séparation complète entre ce sous-problème et le second sous-problème.

Phase de minimisation

Nous calculons d'abord le pas δu :

$$\begin{aligned} &\min_{\delta u} \; {g_k}^T \delta u + \frac{1}{2}\delta u^T W_k \delta u \\ \textit{sous la contrainte} \quad &\|\delta u\| \leq \sqrt{\Delta_k^2 - \|v_k\|^2}. \end{aligned} \tag{2.6}$$

où W_K est une approximation B.F.G.S. (voir [Nocedal and Wright, 1999; Bonnans *et al.*, 2002]) du hessien réduit calculée à partir des itérés successifs g_k et δu_k.

Introduisant

$$h_k \; = \; (-\, A_k^{-1} \frac{\partial \Psi}{\partial u}(x_k)\delta u \; , \; \delta u) \tag{2.7}$$

le pas final est défini par

$$d_k \; = \; (v_k, 0) \; + \; h_k. \tag{2.8}$$

L'algorithme s'écrit donc comme suit

Initialisation des constantes $\varepsilon > 0$ et $\eta \in (0,1)$
Initialisation de x_0 et $\delta_0 > 0$.
Pour k=0,...
  Calculer $J_k, \Psi_k, A_k, \Pi_k, g_k$ à partir de (2.2),(2.3) and (2.4)
  Si $\|g_k\|_\infty < \varepsilon$ **et** $\|\Psi_k\|_\infty < \varepsilon$ **alors arrêt** (Le problème est résolu).
  Phase de Restoration
  Calculer v_k en résolvant le sous-problème vertical (2.5)
  Phase de Minimisation
  Calculer W_k en utilisant une approximation BFGS.
  Calculer h_k en résolvant le sous-problème horizontal (2.6).
  Soit $d_k = v_k \; + \; h_k$.
  Mise à jour de Δ_k avec une heuristique de Région de Confiance.
Fin de la boucle : k suivant.

2.2.2 Heuristique de la Région de Confiance.

La décroissance simultanée du résidu de l'état et de la fonctionnelle coût est contrôlée à l'aide d'une fonction de mérite de Han :

$$\Phi(x) = J(x) + \mu \|\Psi(x)\|_2 .\text{où } \mu > 0 \text{ est le coeficient de pénalité.}$$

Cet algorithme s'écrit :

Calculer le nouveau coefficient de pénalité μ^+. μ est le coefficient de pénalité à l'itération précédente (Voir [Lalee *et al.*, 1998]).

Calculer la **réduction prédite**

$$\text{p\_red} \;=\; -d_k^T g_k - \frac{1}{2} d_k^T W_k d_k + \mu(\|\Psi(x_k)\| - \|A_k^T v_k + \Psi(x_k)\|). \tag{2.9}$$

Calculer la **réduction effective** :

$$\text{a\_red} \;=\; J(x_k) - J(x_k + d_k) + \mu^+(\|\Psi(x_k)\| - \|\Psi(x_k + d_k)\|). \tag{2.10}$$

Si $\dfrac{\text{a\_red}}{\text{p\_red}} \geq \eta$

Alors $x_{k+1} = x_k + d_k$, $\mu_{k+1} = \mu^+$, et on augmente le rayon de confiance.

Sinon $x_{k+1} = x_k$, $\mu_{k+1} = \mu_k$, et on réduit le rayon de confiance.

Fin de la boucle : k **suivant.**

2.2.3 Quelques remarques

La phase de minimisation contient une correction u de la variable d'état calculée à partir du troisième problème linéaire (2.7). Cette correction utilise une formule de Taylor d'ordre un qui, pour de petits pas, assure que le résidu de l'équation d'état n'augmente pas beaucoup.

Le contrôle de l'algorithme repose sur l'utilisation de deux principaux paramètres, le rayon ε de la Région de Confiance, et le coefficient de pénalisation μ^+.

Le rayon de la Région de Confiance ε contrôle la correspondance locale entre le modèle quadratique et la fonctionnelle à minimiser. Plus précisément, cette correspondance est équivalente à la classique *condition de Wolfe* : $(0 < \omega_1 < \omega_2 < 1)$

$$\left\{ \begin{array}{lcl} h(\varepsilon) & \leq & h(0) + \omega_1 \varepsilon h'(0) \\ h'(\varepsilon) & \geq & \omega_2 h'(0). \end{array} \right. \tag{2.11}$$

où $h(\varepsilon) = \Phi(x + \varepsilon . d)$. Afin de la satisfaire, il suffit de réduire le rayon ε puisque le gradient usuel satisfait (2.11).

Le paramètre de pénalisation μ^+, via la fonction de mérite, contrôle la convergence simultanée de la fonctionnelle (vers le minimum) et des contraintes (vers zéro).

2.3 Algorithme One-Shot

2.3.1 Application à de grands systèmes

L'un des grands progrès des algorithmes SQP est de remplacer la résolution de sytèmes *a priori* non linéaires par des résolutions de problèmes quadratiques (Problèmes verticaux et de minimisation).
Dans l'esprit des méthodes SQP, ces problèmes sont relativement faciles à résoudre.
Cependant lorsque l'on applique ces algorithmes SQP à des problèmes de grande taille (nombre de variables d'état et nombre de paramètres de formes grands), les problèmes quadratiques deviennent eux-mêmes difficiles à résoudre. En effet le temps CPU nécessaire pour de telles résolution devient vite prohibitif surtout quand il doit être répété à chaque itération d'optimisation. Si l'on considère l'algorithme de Byrd-Omojokun présenté dans le paragraphe 2.2.1, on dénombre trois systèmes, potentiellement grands, à résoudre : un dans la résolution de l'équation adjointe 2.3, un autre dans la résolution du problème vertical 2.5 et enfin un dans la résolution du problème de minimisation 2.7.
L'algorithme One-Shot-Trust-Region que nous présentons dans ce paragraphe permet d'éviter les résolutions complètes de tels systèmes.

2.3.2 Principale hypothèse : l'itération pseudo-Newton

Le pas de restauration (2.5) de l'équation d'état est remplacé par un pas de *pseudo-Newton* :

$$v = -S\ \Psi(x_k)$$

reposant sur la particularité de l'opérateur S d'être relativement peu coûteux à évaluer et d'approcher l'inverse du Jacobien $\dfrac{\partial\Psi}{\partial Y}$. Typiquement, il peut être le résultat de quelques balayages de Gauss-Seidel ou d'itérations de Gradient Conjugué préconditionné. Nous utilisons cette solution simplifiée à la fois dans la phase de restauration et dans le calcul du pas final h_k lors de la phase de minimisation (2.7).
Nous utilisons la même démarche pour l'équation adjointe. Dans l'algorithme proposé, l'équation adjointe est considérée comme une contrainte d'égalité additionnelle. Au lieu de résoudre

$$\Pi = (\frac{\partial\Psi}{\partial Y})^{-T}\frac{\partial J}{\partial Y},$$

sa restauration consistera en la résolution d'un problème plus simple

$$p = S^T\left((\frac{\partial\Psi}{\partial Y})^T\Pi - \frac{\partial J}{\partial Y}\right).$$

Il est alors naturel de demander au nouvel algorithme de restaurer approximativement à la fois la contrainte sur l'état et la contrainte sur l'adjoint lors de deux phases de restauration avant de passer à la phase de minimisation. Par comparaison avec un algorithme SQP classique, l'utilisation de résolutions incomplètes doit décroitre considérablement le coût d'une itération globale.
Les hypothèses principales concernant l'itération précédente pseudo-Newton devraient comprendre :

- des propriétés sur la convergence de l'itération d'état,
- des propriétés sur la convergence de l'itération adjointe,
- des hypothèses sur la complexité algorithmique des itérations de Pseudo-Newton pour les deux équations,
- des hypothèses permettant la restauration de la propriété de descente.

Cela conduit à formuler les hypothèses suivantes :

• **Hypothèse 1 :** Pour $u \in I\!R^{N_u}$ et pour $p \in I\!R^{N_Y}$

$$\|\Psi(Y - S\Psi(Y,u),u)\| \ \leq \ 0.99 \ \|\Psi(Y,u)\| \qquad\qquad (a)$$

et

$$\| \left(\frac{\partial \Psi}{\partial Y}\right)^* (\Pi + p) - \frac{\partial J}{\partial Y}\| \ \leq \ 0.99 \ \| \left(\frac{\partial \Psi}{\partial Y}\right)^* \Pi - \frac{\partial J}{\partial Y}\| \qquad\qquad (b)$$

• **Hypothèse 2 :**

Les pas de pseudo-Newton (a) et (b) ne comptent qu'un petit nombre d'assemblages (<10).

• **Hypothèse 3 :** $\forall \ u_0$, après k_1 étapes de résolutions partielles, $g(Y_k, u_0, \Pi_k)$ est suffisamment proche du gradient pour être une "bonne" direction de descente :

$$\exists \ k_1, \exists \ \rho_{max} \ / \ \forall \ \rho, \ \ 0 < \ \rho \ \leq \ \rho_{max}, \ \forall \ k \geq k_1 \ \ ,$$

$$J(Y_k, u_0 - \rho W_k^{-1} g(Y_k, u_0, \Pi_k)) - J(Y_k, u_0) \ \leq \ -0.95\rho \ < g(Y_k, u_0, \Pi_k) \ , \ W_k^{-1} g(Y_k, u_0, \Pi_k) >$$

où $\Pi = (\frac{\partial \Psi}{\partial Y})^{-T} \frac{\partial J}{\partial Y}$ et $j'(u) \ = \ \frac{\partial J}{\partial u}(x) \ - \ < \Pi \ , \ \frac{\partial \Psi}{\partial u}(x) >$.

L'**hypothèse 1** est une hypothèse de convergence concernant l'itération de pseudo-Newton. Concernant l' **hypothèse 3** nous observons que pour $k \longrightarrow +\infty$, Y_k converge vers la solution de l'équation d'état, Π_k converge vers la solution de l'équation adjointe et alors g est le gradient de la fonctionnelle j, $j \, : \, u \ \rightarrow \ J(Y(u), u)$. L'**hypothèse 3** est donc une hypothèse de régularité. On peut considérer l'**hypothèse 3** comme une conséquence de l'**hypothèse 1** si la fonctionnelle J et l'opérateur Y sont suffisamment réguliers. Les valeurs des constantes positives .99 et .95 ne sont pas importantes aussi longtemps qu'elles restent inférieures et proches de l'unité.

2.3.3 Post-restauration de la variable d'état

La phase de minimisation de l'algorithme BO Hessien-réduit de la Section 2 requiert, dans (2.7), une résolution exacte de l'équation d'état linéarisée. Comme noté précédemment, cette mise-à-jour de l'état linéarisé assure que le pas de minimisation SQP maintient le résidu de l'équation d'état au niveau (précédent) atteint dans la phase de restauration.

Dans le nouvel algorithme, une bonne mise-à-jour de la variable d'état sera aussi nécessaire, en particulier afin de pouvoir rechercher (par une *recherche linéaire*) une correction qui satisfait à la condition de Wolfe.

Pour des raisons d'efficacité, nous ne pouvons appliquer une résolution linéaire complète pour mettre à jour l'état. A la place, nous utilisons l'opérateur S pseudo-Newton. Mais cet opérateur ne permettra pas une bonne évaluation de δY. Il est plus raisonnable de tenter de mettre à jour Y en appliquant une "phase de post-restauration" similaire à la phase de restauration précédente. Cette phase est plus efficace ou plus stable car elle évite de linéariser Ψ par rapport à u comme cela est fait dans (2.7). L'équation d'état est donc avancée itérativement à l'aide de l'opérateur S. Comme dans SQP, un certain niveau de convergence doit être satisfait. **Nous imposons de réitérer l'étape Pseudo-Newton tant que le résidu final de la post-restauration ne devient pas inférieur en module au résidu obtenu lors de la pré-restauration.**

Ce critère est appliqué à chaque restauration dans la recherche linéaire (voir 2.3.5, Phase de minimisation, Étape (1)).

Finalement, afin d'avoir des convergences itératives équilibrées de l'état et de l'adjoint vers leurs limites, nous mettons aussi à jour l'état adjoint. Deux possibilités existent pour évaluer Π, soit de mettre à jour la variable d'état pour le calcul de la variable adjointe, soit de ne pas le faire. Nous choisissons la première option qui est *a priori* plus stable et qui converge en pratique plus rapidement.

2.3.4 Étapes de contrôles de l'algorithme

Lors de la transformation de l'algorithme SQP en son extension one shot, la prise en compte des contraintes égalités (équation d'état) doit être profondément reconsidérée.

- Dans SQP, un pas complet peut être refusé (ou fortement réduit en longueur) en fin de boucle par l'heuristique de la Région de confiance (essentiellement parce que la condition de Wolfe n'est pas vérifiée). Dans ce cas, la variable d'état est remise à sa valeur précédente, et l'effort CPU passé à la restaurer est perdu).

- Dans one-shot, pour des raisons d'efficacité, l'effort CPU pour avancer les variables d'état ne doit pas être perdu. Lorsque l'itération d'état n'est pas suffisante pour satisfaire la condition de Wolfe, nous proposons de supprimer seulement la mise à jour de la variable de minimisation et de garder le bénéfice de la pré-restauration. Comme on revient au début de l'algorithme, la restauration sera ré-itérée, éventuellement plusieurs fois jusqu'à ce qu'un pas de minimisation satisfaisant la condition de Wolfe soit obtenu et appliqué sur la variable u.

Nous avons aussi besoin de pouvoir contrôler le niveau de résolution de l'équation d'état lors de la phase de minimisation. Dans SQP standard, la correction n'augmente pas le résidu de l'équation d'état aussi longtemps que le pas global de minimisation reste petit, à cause de la résolution du troisième système. Cette propriété est contrôlée par le mécanisme de la fonction de mérite. Dans notre nouvel algorithme, cette heuristique de pénalisation a été remplacée par un contrôle direct de la norme du résidu comme expliqué dans le paragraphe 2.3.3.

2.3.5 Présentation globale de l'algorithme

Notations : on définit de l'équation d'état par $Res_Y = \|\Psi\|$ et le résidu de l'équation adjointe par $Res_\Pi = \|\dfrac{\partial J}{\partial Y} - \left(\dfrac{\partial \Psi}{\partial Y}\right)^T \Pi\|$

Initialisation

Initialisation de x_0, Δ_0 et μ_0. ($nstep = 1$)

Pour k=0,...

Soit $\|G_k\|_\infty = Max(\|\dfrac{\partial J}{\partial Y} - \left(\dfrac{\partial \Psi}{\partial Y}\right)^T \Pi\| \;,\; \|\dfrac{\partial J}{\partial u} - \left(\dfrac{\partial \Psi}{\partial u}\right)^T \Pi\| \;,\; \|\Psi(x)\|)$
Si $\|G_k\|_\infty < \varepsilon$ alors arrêt.
Calculer la fonction coût J.
• Phase de Restauration de l'état .
Calculer un pas Pseudo-Newton $v_Y = -S(Y,u)\Psi(Y,u)$
Mise à jour de l'état Y : $Y^+ = Y + v_Y$
• Phase de Restauration de l'état adjoint.
Calculer un pas Pseudo-Newton $v_\Pi = -S(Y^+,u)^T \left(\left(\dfrac{\partial \Psi}{\partial Y}(Y^+,u)\right)^T \Pi - \dfrac{\partial J}{\partial Y}(Y^+,u) \right)$.
de telle sorte que $Res_Y < Res_\Pi$.
Mise à jour de l'état adjoint Π : $\Pi^+ = \Pi + v_\Pi$
Soit $u^+ = u$.
Calculer le gradient $g^+ \equiv g(x^+, \Pi^+) = g(Y^+, u^+, \Pi^+)$
Utiliser une accélération BFGS ou Gradient Conjugué
Phase de minimisation

(1) Calculer le pas δu en utilisant une recherche linéaire puis résoudre l'équation d'état au même niveau de résidu Res_Y que lors de la pré-restauration
(2) Si la condition de Wolfe 2.11 est satisfaite pour le pas,
Mise à jour du contrôle $u' = u + \delta u$,
Mise à jour de l'état Y : $Y' = Y^+ + \delta Y$
Sinon réduire le pas de la recherche linéaire puis retour en (2)
(4) Si la condition de Wolfe n'est satisfaite pour aucun pas,
l'étape de minimisation est invalidée, aller au début de la première restauration,

• Phase de post-restauration de l'adjoint post (sans région de confiance).

$\delta\Pi = -S(Y',u')^T \left(\left(\dfrac{\partial \Psi}{\partial Y}(Y',u')\right)^T \Pi - \dfrac{\partial J}{\partial Y}(Y',u') \right)$. avec $Res_\Pi < Res_Y$

Mise à jour de l'adjoint Π : $\Pi^+ = \Pi + v_\Pi$

2.3.6 Quelques propriétés de l'algorithme

Nous n'avons pas la démonstration de la convergence de cet algorithme. Par construction, cet algorithme fournit une suite (u_n, Y_n, Π_n) telle que $J(u_n, Y_n, \Pi_n)$ soit décroissante.

Quelques autres propriétés simples peuvent être énoncées. On fait l'hypothèse que dans un certain contexte , la suite (u_n, Y_n, Π_n) converge vers une limite, c'est-à-dire que

$$u_n \to u$$

$$Y_n \to Y$$

$$\Pi_n \to \Pi$$

alors les propriétés suivantes sont vérifiées :

<u>Lemme</u> 1
(Y, u, Π) *sont solutions des équations d'état et d'adjoint pour le contrôle* u*.*

En effet, puisque le contrôle tend vers une valeur fixe, l'état et l'adjoint sont en conséquence progressivement résolus dans l'itération proposée.

<u>Lemme</u> 2
(Y, U, Π) *n'est pas un point où le gradient n'est pas une direction de descente.*

Sinon, l'étape de minimisation ne serait pas valide et la convergence de l'état et de l'adjoint serait poussée plus loin, alors par l'hypothèse (2) le gradient deviendrait une direction de descente.

<u>Lemme</u> 3
(Y, u, Π) *n'est pas un point où la longueur de la direction de descente de la recherche linéaire tend vers zéro,*

Ceci entrerait en contradiction avec l'hypothèse puisque le pseudo-gradient satisfait la condition de Wolfe.

Corollaire 2.3.1
Le seul scénario alors possible est que Y_n *et* Π_n *convergent vers l'état et l'adjoint solutions et que* u_n *converge vers une solution de* $j' = 0$*.*

2.3.7 Complexité de l'algorithme one shot

Le coût par itération d'optimisation est beaucoup plus faible que celui de l'algorithme SQP standard. En effet à chaque itération d'optimisation, nous assemblons les équations et appliquons seulement les pas pseudo-Newton, qui sont de complexité similaire à l'assemblage (Hypothèse 2).

Pour avancer dans cette analyse, supposons que :
-les recherches linéaires aboutissent en un nombre d'étapes fini inférieur à un nombre k.
-les niveaux de résidus deviennent faibles seulement lorsque les variations du contrôle sont elles-mêmes faibles. Dans un contexte où cette seconde hypothèse ne serait pas vérifée l'algorithme se transformerait de lui-même en un algorithme SQP standard.

Si ces deux hypothèses sont vérifiées, la complexité serait alors de l'ordre de $N_u+k.N_Y$ où N_u est le nombre d'inconnues de contrôle et N_Y est le nombre d'inconnues d'état. Cette estimation est en mettre en regard avec le cas SQP standard $N_u \times N_Y$.

2.4 Application à un problème modèle

Il est intéressant de mesurer l'impact de la technologie one-shot sur un problème traité facilement par l'algorithme de Byrd-Omojokun. On choisit donc un problème quadratique de contrôle optimal distribué.

2.4.1 Problème théorique

Nous notons Ω l'intervalle unité, $\Omega = [0,1]$ et $J(Y,u)$ la fonctionnelle suivante :

$$J(Y,u) = \frac{1}{2}\int_{\Omega}(Y - Y_d)^2 dx \text{ indép. de } u \ . \tag{2.12}$$

Nous voulons résoudre le *problème de minimisation* suivant

$$\min_{v \in \mathcal{U}} \quad J(Y(v),v) \tag{2.13}$$

où $Y(v)$ est la solution du problème :

$$\begin{cases} -\Delta Y(v) &= f + v \\ Y(v) &\in H_0^1(\Omega) \end{cases} \tag{2.14}$$

Nous considérons une discrétisation en éléments finis P_1-Galerkin du problème continu (2.14). Pour des raisons de simplicité, nous gardons les mêmes notations. Nous notons par A la matrice de rigidité de l'équation d'état

$$AY = f + v \ . \tag{2.15}$$

Le système adjoint s'écrit

$$A^*\Pi = Y(v) - Y_d \ . \tag{2.16}$$

Le gradient s'écrit

$$g(Y, v) = \Pi(v) \ . \tag{2.17}$$

2.4.2 Calcul grossier

Nous appliquons d'abord l'algorithme de Byrd-Omojokun pour N = 50 (50 paramètres de contrôle et 50 variables d'état). L'équation d'état linéarisée est résolue deux fois à chaque itération d'optimisation, avec à chaque fois un résidu final inférieur à 10^{-12}. Ceci est obtenu en environ $30 - 50$ itérations de gradient conjugué. Nous présentons dans la figure 2.1 le comportement de la convergence de la condition d'optimalité, avec pour mémoire les résidus des équations d'état et d'état adjoint. Par construction les deux résidus d'état restent petits à chaque itération. Avec ce petit nombre d'inconnues, la convergence de l'algorithme de Byrd-Omojokun est plutôt rapide mais le zéro machine n'est obtenu qu'en 50 itérations d'optimisation. La convergence superlinéaire apparait après 30 itérations, comme on peut s'y attendre avec ce type d'algorithme Quasi-Newton sur un problème symétrique de 50 inconnues.

Nous passons maintenant au nouvel algorithme. Comme pour l'algorithme de Byrd-Omojokun, l'option BFGS est permise par le relativement petit nombre de paramètres. Les équations d'état et d'état adjoint sont résolues avec un algorithme "pseudo-Newton" consistant en 3 itérations seulement de gradient conjugué. Les résidus de l'état commencent avec des valeurs plutôt élevées puis décroissent d'une façon similaire à l'équation d'optimalité. Chaque itération d'optimisation est trois ou quatre fois moins coûteuse que pour le cas précédent. La convergence est, de manière surprenante, plutôt rapide, en 55 itérations. La convergence superlinéaire apparait en environ 35 itérations.

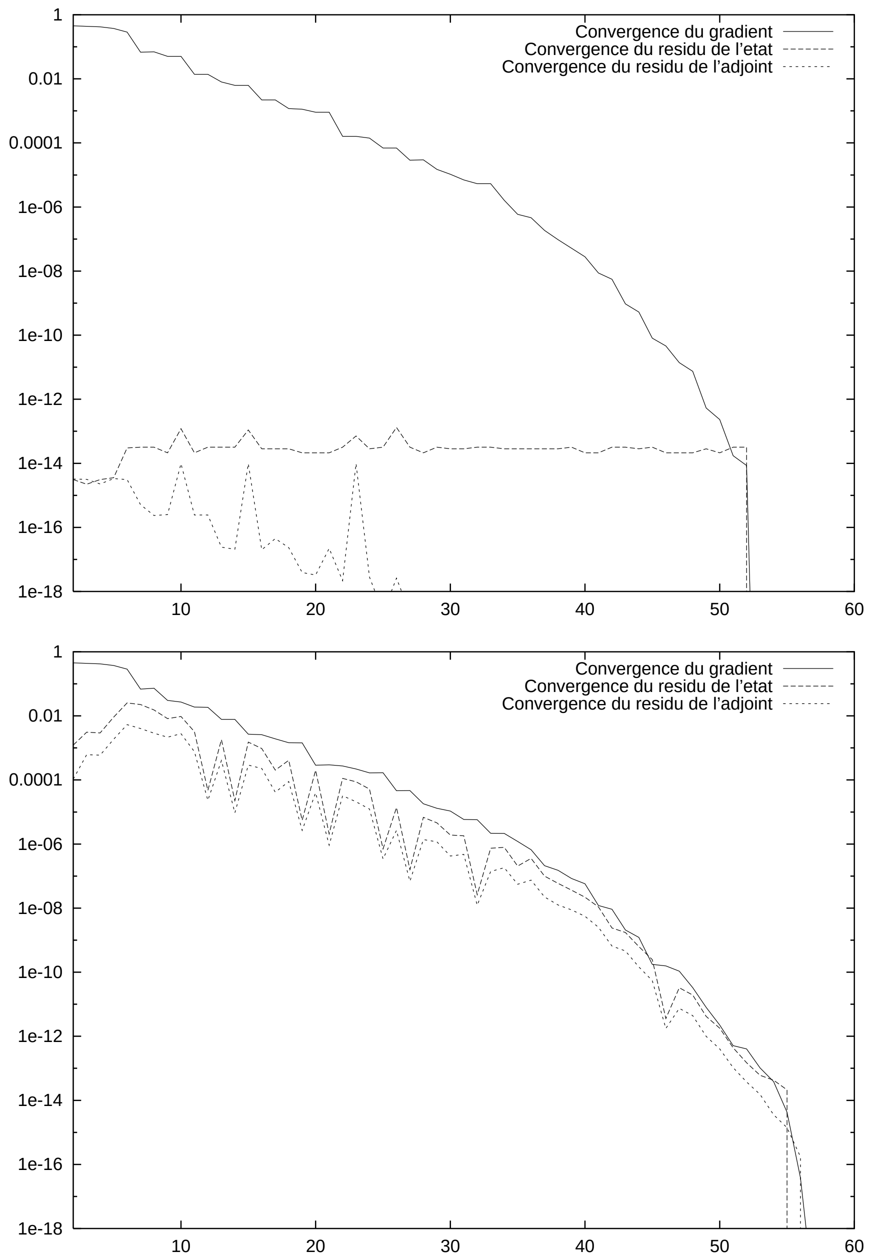

FIG. 2.1 – *Problème modèle de contrôle optimal (maillage grossier). Haut : convergence avec Byrd-Omojokun, bas : convergence one shot*

Tab. 2.1 – *Efficacité des différents calculs mesurés avec le nombre total d'itérations de gradient conjugué pour une descente jusqu'à* 10^{-14} *de tous les résidus.*

Cas	Itérations d'Opt.	Itérations de Grad. Conj.
Byrd-Omojokun, N=50	50	2000
One shot, N=50	55	660
Byrd-Omojokun, N=500	80	16000
One shot, N=500	85	1020

2.4.3 Calcul sur un maillage plus fin

Nous considérons maintenant le cas d'un problème plus grand avec 500 paramètres de contrôle (et 500 variables d'état). A chaque résolution de l'équation d'état, des centaines de balayages sont nécessaires pour amener le résidu de l'état au zéro machine. L'application de l'algorithme Byrd-Omojokun est de nouveau plutôt rapide, avec une convergence en 70 itérations d'optimisation. A cause du grand nombre d'inconnues, la convergence superlinéaire n'apparait pas avant d'atteindre le zéro machine.

Le nouvel algorithme est appliqué aussi avec l'option BFGS. L'itération pseudo-Newton consiste en 3 itérations de GC. En dépit de cela, la convergence est rapide, 85 itérations. Les pics dans les résidus de l'état et de l'adjoint correspondent à la réitération des pré-restaurations, montrant que les 3 itérations sont souvent insuffisantes. Mais le calcul est robuste et s'adapte bien à ce handicap, et garde un excellent facteur d'efficacité, puisque le gain par rapport à l'algorithme BO supérieur à 10. Les comparaisons précédentes sont résumées dans le Tableau 2.1.

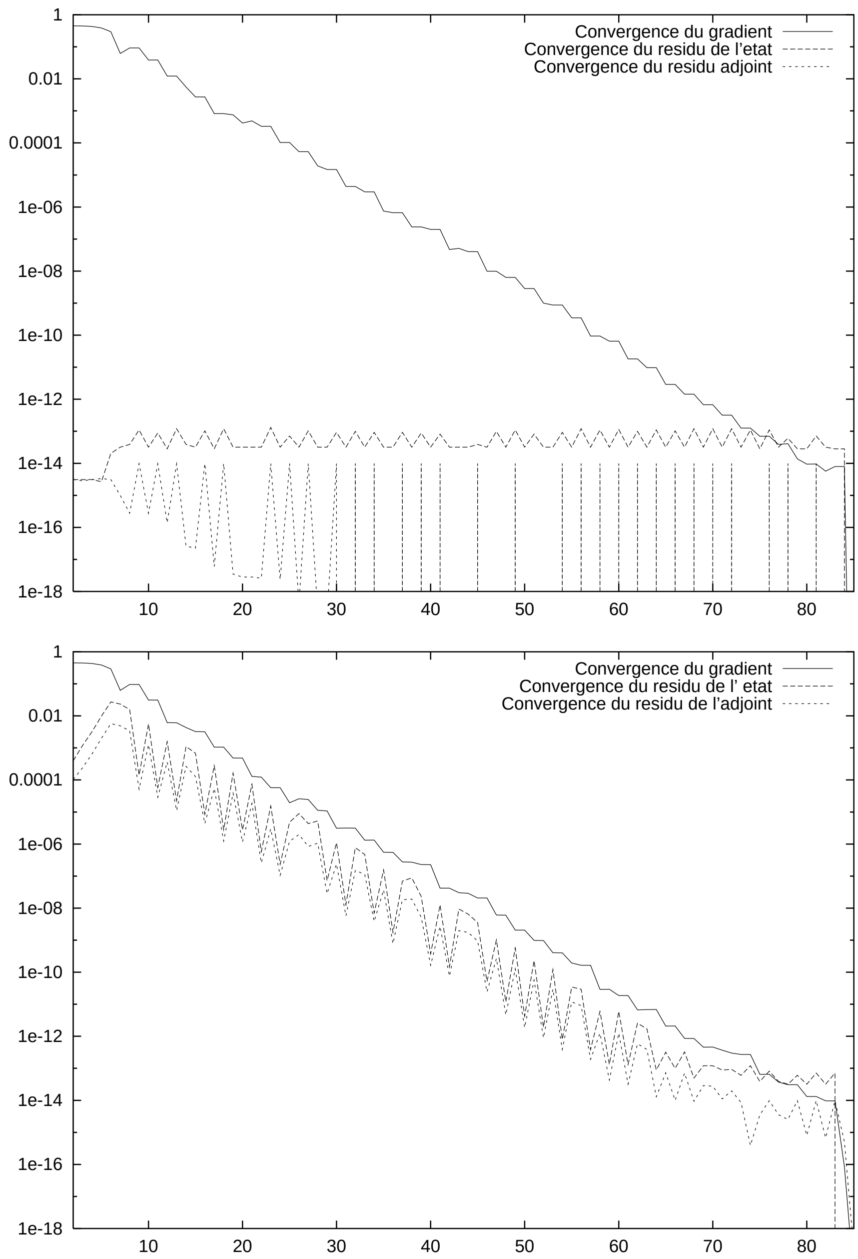

FIG. 2.2 – *Problème modèle de contrôle optimal (maillage fin). Haut : convergence avec Byrd-Omojokun, bas : convergence one shot*

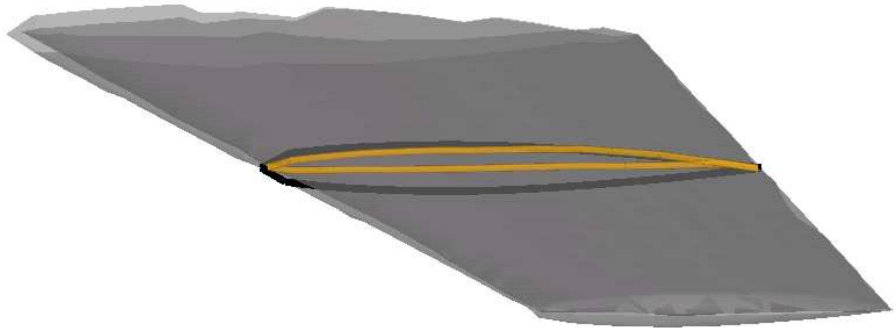

FIG. 2.3 – *Optimisation de l'aile ONERA M6. Aile initiale(noire) et optimisée (grisée).*

2.5 Application à la réduction du bang sonique

L'optimisation de formes en aérodynamique est un domaine très gourmand en temps de calcul. Il y a une vingtaine d'années, seules les approches par problèmes inverses étaient abordables. Elles furent progressivement remplacées par la théorie du contrôle basée sur l'optimisation à la suite des travaux pionniers de Pironneau puis de Jameson (voir [Jameson, 1988b; Jameson, 1991]). Des algorithmes d'optimisation pseudo-instationnaires émulant l'approche inverse avec une efficacité comparable aux méthodes one-shot ont été proposés dans [Iollo *et al.*, 1995].

2.5.1 Le problème d'optimisation de formes

Dans l'expérience présentée ici, nous utilisons la version du schéma spatiale précise au premier ordre. Dans cette version, les résolutions de l'état et de l'adjoint sont en effet beaucoup moins coûteuses en cpu (avec un facteur 4-5).
Dans le code initial, un gradient conjugué était utilisé pour l'optimisation. Nous nous concentrons sur le cas d'une optimisation débutant avec une aile ONERA-M6, et avec une géométrie grossière de 780 paramètres de contrôle (11.000 variables fluide). Dans ce cas la fonctionnelle de coût a été diminuée de 0.06 jusqu'à 0.01. L'aile optimisée est sensiblement différente de l'aile initiale, voir Fig.2.3, et la réduction de l'intensité du choc est démontrée par une coupe du champ de pression sous l'aile, voir Fig.2.5.
Le protocole pour nos expériences est de garder exactement la spécification du cas test, en remplaçant l'optimiseur initial avec gradient conjugué par (1) la version de l'algorithme SQP Byrd-Omojokun décrite dans ce chapitre, et (2) l'extension one-shot proposée dans ce chapitre.

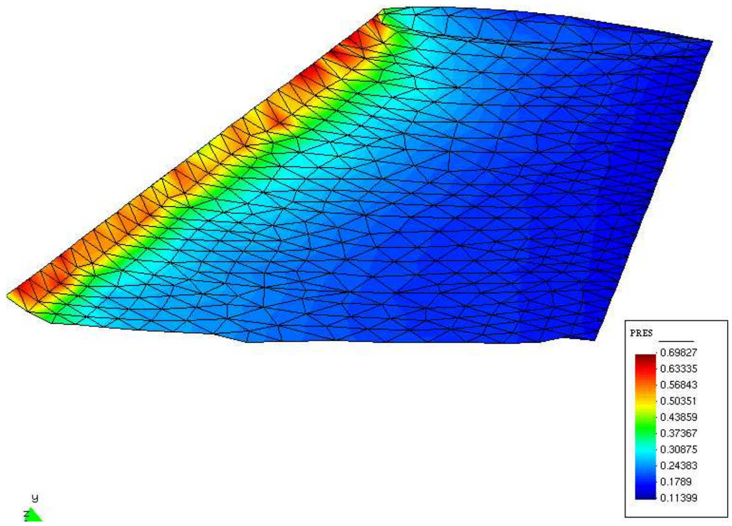

FIG. 2.4 – Aile ONERA M6 (Mach 0.84) isopression sur la peau de la voilure

2.5.2 Résultats obtenus avec un maillage grossier

Nous présentons un premier test avec un maillage grossier de 11.000 sommets pour l'équation d'Euler et de 780 paramètres de forme. Des expériences plus réalistes mèneraient à un nombre de paramètres beaucoup plus grand, rendant l'utilisation d'un BFGS classique impossible pour des raisons de stockage mémoire. Nous avons donc doté les algorithmes SQP d'un gradient conjugué nonlinéaire de type Polak-Ribière, comme utilisé dans la boucle de gradient non-SQP originelle de [Vazquez *et al.*, 2002].
Tous les résultats que nous présentons sont donc obtenus avec le modèle de gradient conjugué nonlinéaire à l'intérieur de l'algorithme SQP.
L'application de SQP de Byrd-Omojokun avec résolution complète non linéaire (jusqu'au zéro machine) de l'équation d'état est un cauchemar du point de vue du temps de calcul. Le temps CPU total pour 200 itérations est de 14 heures sur une station de travail de 1Ghz. Le coût CPU se décompose comme suit : une résolution de l'équation d'état consiste en 40 pas de temps, chacun incluant 40 balayages de Jacobi. Ceci résulte en un temps CPU de environ 100 secondes. La résolution de l'état adjoint comprend seulement 100-200 balayages de Jacobi et $5-10$ secondes car nous avons fixé l'ordre de précision du modèle numérique CFD global au premier ordre pour réduire le coût du test. Avec 200 itérations, voir Fig.2.6, le gradient de la fonctionnelle a diminué de deux ordres de grandeur. Du point de vue de la fonctionnelle de coût, il apparaît que les 100 dernières itérations sont utiles à décroitre encore la fonctionnelle d'un ordre de grandeur mais ne changent pas la forme obtenue. Pour des applications pratiques, seulement quelques itérations, typiquement moins de 50 seraient nécessaires.
Pour l'option one shot, la résolution partielle de l'équation d'état implique seulement un pas de temps avec donc une seule évaluation de matrice et 20 balayages de Jacobi. La

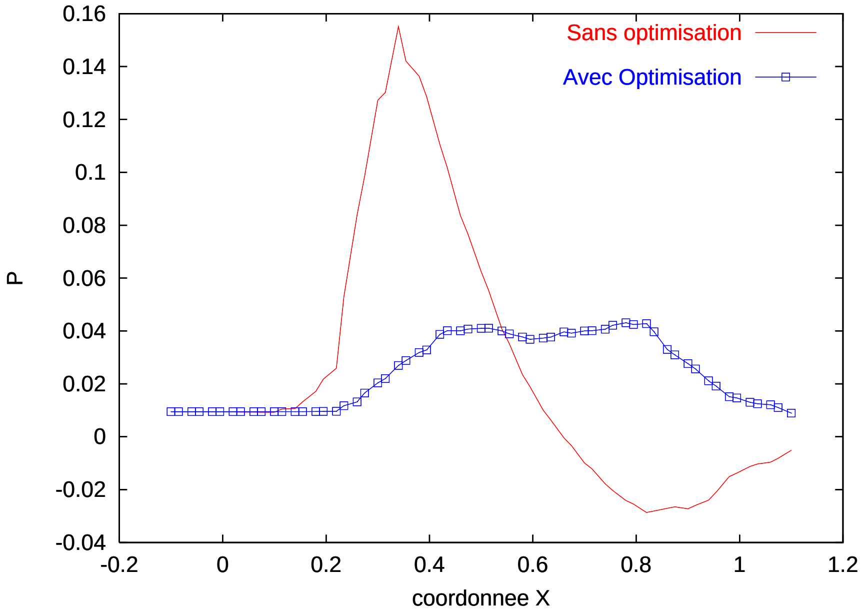

FIG. 2.5 – *Optimisation de l'aile ONERA M6. Pression sur une ligne sous l'aile.*

résolution de l'équation adjointe implique seulement 20 balayages de Jacobi. Nous présentons le résultat de 200 itérations, pour un temps CPU total de 40 minutes. Le gradient diminue d'un même facteur que dans le cas du SQP (Fig.2.7). En pratique, 50-100 pas ($10-20$ minutes) seront suffisants pour décroître significativement le coût.

Puisque une résolution complète de l'équation d'état prend 1.5 minutes, la solution optimale est obtenue par one-shot pour un coût d'environ 15 résolutions état plus adjoint, c'est-à-dire 40 fois plus petit que pour l'option SQP.

2.5.3 Résultats obtenus avec un maillage plus fin

La même méthode est appliquée à une géométrie plus fine avec 3222 paramètres de contrôle et 77,315 variables d'état. Dans ce cas, une seule résolution de l'écoulement prend 23 minutes. Une centaine d'itérations d'optimisation avec la méthode SQP prennent 44 heures de temps CPU.

Avec l'algorithme one-shot, les taux de convergence (Fig. 2.9) sont remarquablement proches de ceux obtenus à la fois pour le calcul SQP standard en maillage fin et pour ceux en maillage grossier. En pratique, 100 pas (6 heures) sont suffisants pour décroître significativement le coût. La solution optimale est de nouveau obtenue pour un coût d'environ 15 résolutions état plus adjoint.
Les comparaisons précédentes sont résumées dans les tableaux 2.2 et 2.3.

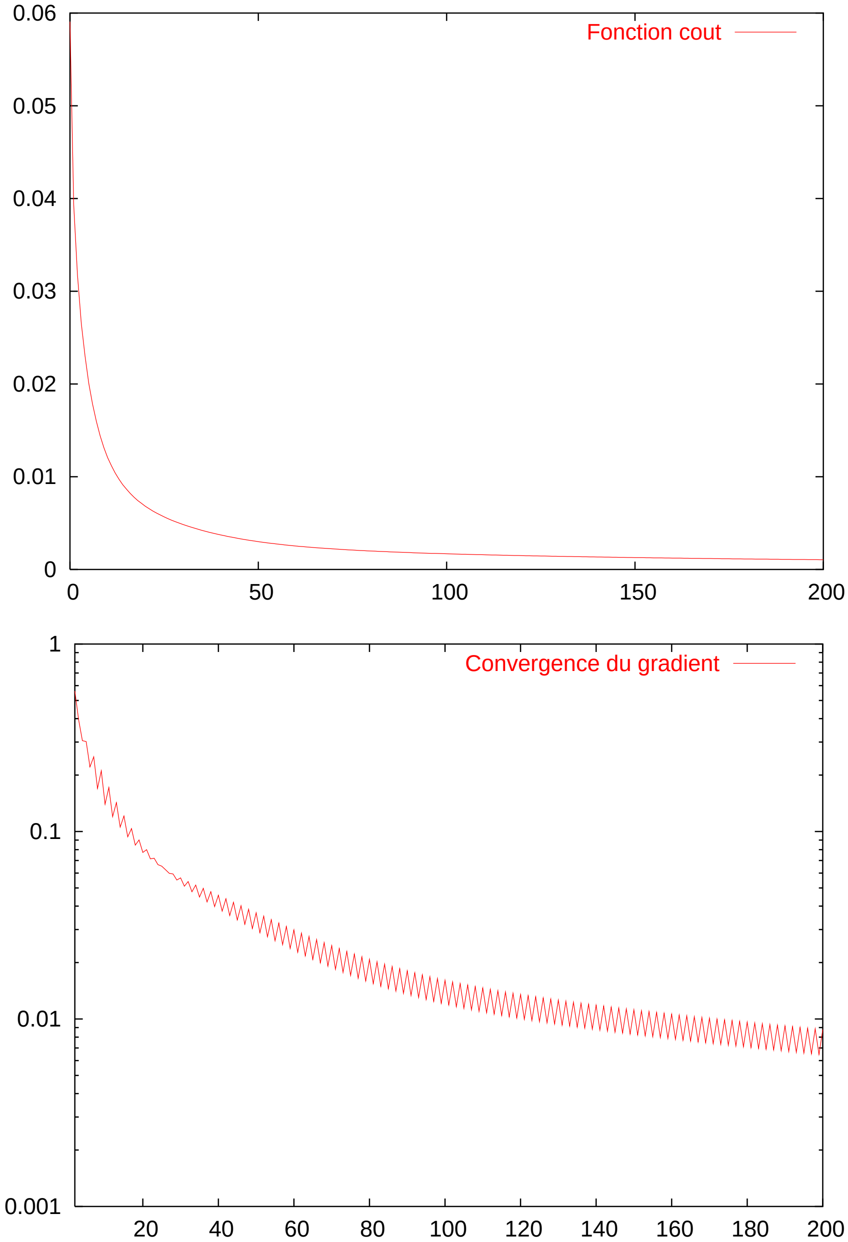

FIG. 2.6 – *Application à un problème d'optimisation de formes (780 paramètres, 11.000 variables d'état). La figure montre la convergence d'une méthode SQP traditionnelle, avec résolution parfaite de tous les systèmes linéaires*

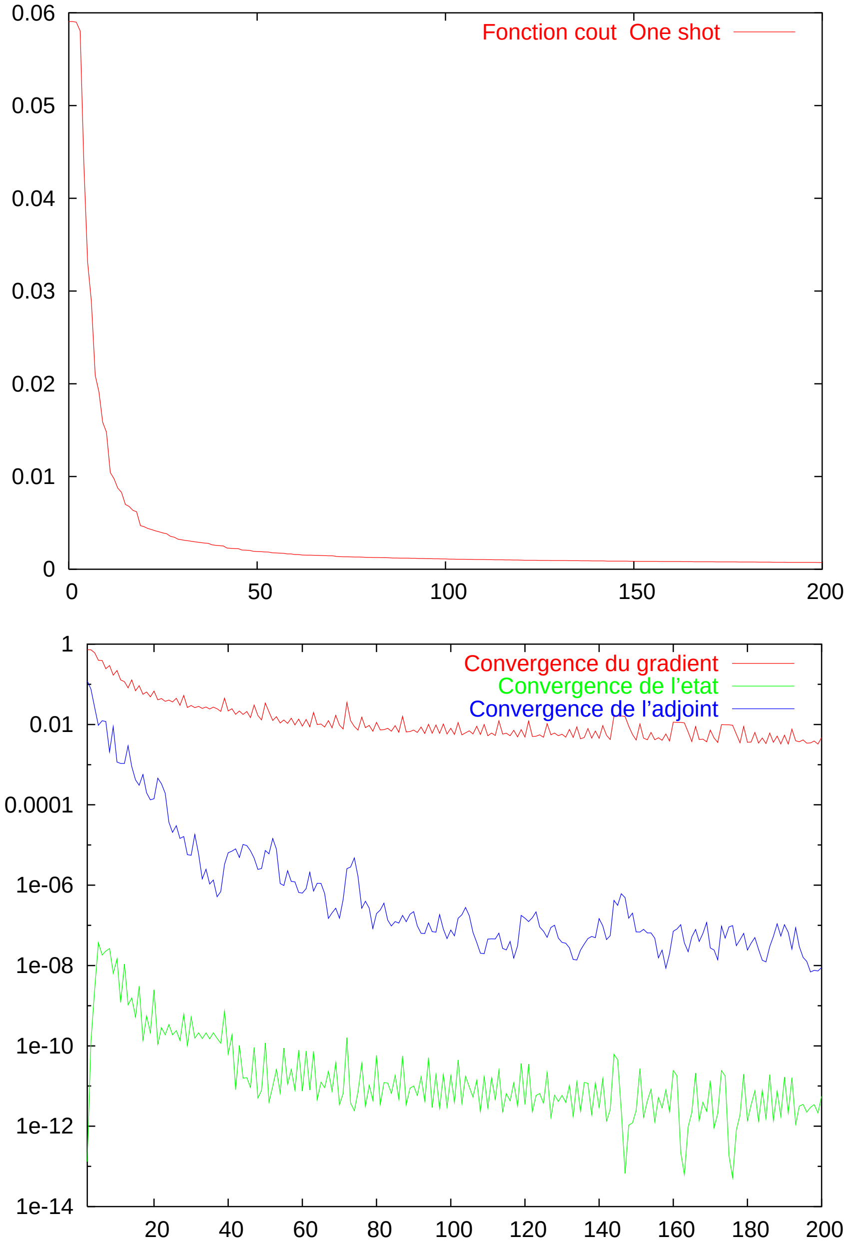

FIG. 2.7 – *Application à un problème d'optimisation de formes (780 paramètres, 11.000 variables d'état). Optimiseur One-shot. La figure du haut montre la convergence de la fonctionnelle de coût. La figure du bas montre la convergence du : gradient de la fonctionnelle (courbe du haut), résidu de l'état (courbe du bas), et au milieu, du résidu de l'adjoint.*

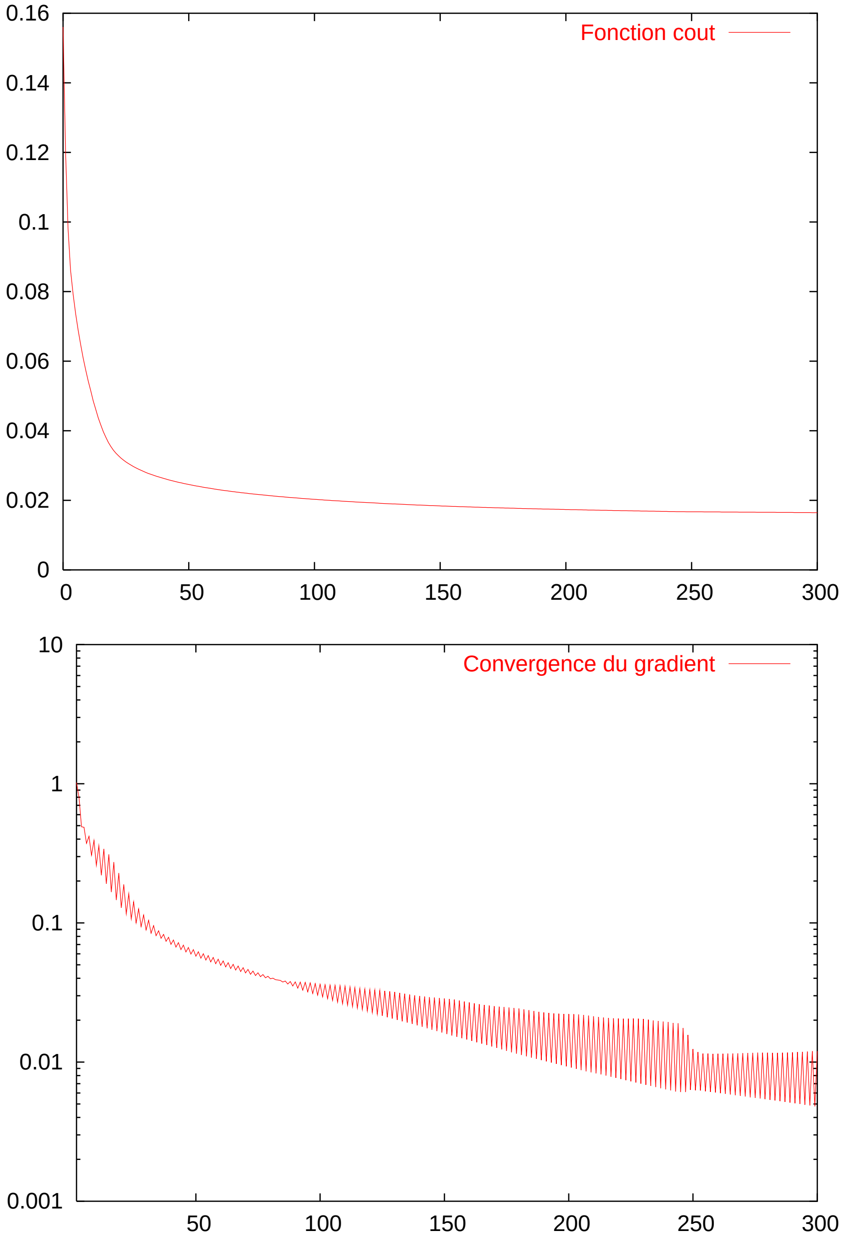

FIG. 2.8 – *Application à un problème d'optimisation de formes (3222 paramètres, 77,315 variables d'état). Optimiseur SQP. Convergence de la fonctionnelle de coût (haut) et du gradient de la fonctionnelle.*

TAB. 2.2 – *Minimisation du bang sonique : convergence*

Cas	Itérations d'opt.	|gradient |	temps CPU
SQP, N=780, M=11,000	100	$11.\ 10^{-2}$	420 mn
One shot, N=780, M=11,000	100	$.8.\ 10^{-2}$	20 mn
SQP, N=3222, M=77,000	100	2.10^{-2}	2640 mn
One shot, N=3222, M=77,000	100	$2.\ 10^{-2}$	360 mn

TAB. 2.3 – *Minimisation du bang sonique : cout en nombre équivalent de résolution complète des états (NERC) et rapport SPQ/one-shot*

Cas	NERC	rapport
SQP, N=780, M=11,000	315	
One shot, N=780, M=11,000	15	21
SQP, N=3222, M=77,000	110	
One shot, N=3222, M=77,000	15	7.3

2.6 Conclusion et extensions futures

Nous proposons une méthode pour résoudre une classe de problèmes d'optimisation avec contrainte égalité. Nous considérons le cas particulier où les systèmes linéarisés état et adjoint sont coûteux à résoudre. L'hypothèse clé est qu'une itération pseudo-Newton, peu coûteuse est disponible pour résoudre l'état et l'adjoint. Celle-ci permet une résolution simultanée des équations du système KKT. Le nouvel algorithme étend les idées de SQP, en utilisant en particulier l'accélération BFGS. Il contient un contrôle direct de la condition de Wolfe.

La méthode proposée travaille essentiellement comme les autres méthodes one shot, mais avec robustesse. Celle-ci est gagnée grâce aux heuristiques nécessitant certes des calculs supplémentaires mais qui ne sont pas chers.

Dans les exemples que nous avons testés, la convergence globale de l'algorithme de minimisation est seulement légèrement dégradée alors que chaque itération d'optimisation est beaucoup moins coûteuse. Encore plus important, ceci reste vrai pour des nombres d'inconnues variés.

Nous pensons que ce type d'algorithme (et ses améliorations futures) est qualifié pour résoudre de manière directe et robuste de nouveaux problèmes sans risque de divergence et sans nécessité de réglage préliminaire des paramètres.

Remerciements : Nous remercions Jean-Charles Gilbert pour nos discussions fructueuses et Stephen Wornom pour avoir relu ce chapitre.

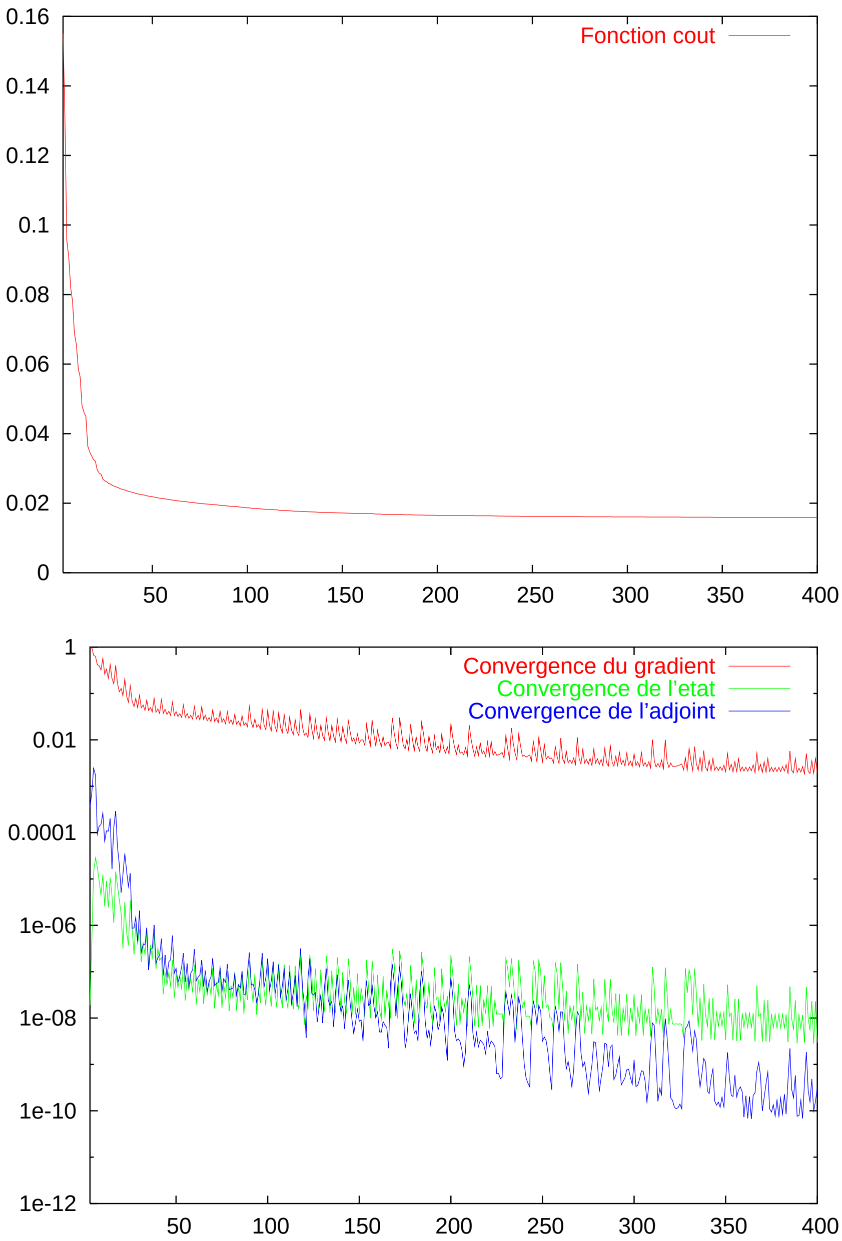

FIG. 2.9 – *Application à un problème d'optimisation de formes (3222 paramètres, 77,315 variables d'état). Optimiseur One-shot. La figure du haut montre la convergence de la fonctionnelle de coût. La figure du bas montre la convergence du : gradient de la fonctionnelle (courbe du haut), résidu de l'état (courbe du bas, pour les itérations de 1 à 100), et entre les deux, du résidu de l'état adjoint.*

3

Préconditionnement multiniveau appliqué à l'optimum design

Sommaire

Le contenu de ce chapitre a eté soumis à la revue *Numerische Mathematik.*

3.1 Introduction

Durant les deux dernières décennies, les chercheurs ont cherchés des algorithmes de résolution efficaces pour les problèmes de Mécanique des Fluides Numérique. Maintenant que de tels algorithmes sont disponibles, beaucoup de chercheurs s'intéressent à la résolution des problèmes d'optimisation de formes. La grande majorité utilisent des algorithmes de gradient, qui ont gagné en efficacité grâce à l'utilisation des états adjoints. Les états adjoints sont plus simples à développer grâce aux progrès des outils de Différentiation Automatique, voir par exemple [Tropics, 2001]. Mais les méthodes de gradient requiert souvent des préconditionneurs. Pour illustrer cela, considérons l'application d'une itération de gradient dans un espace de fonctions. L'analyse fonctionnelle nous dit que l'itération continue $u^{n+1} = u^n - \rho g^n$ est moins régulière que u^n lorsque le gradient g^n contient des dérivées k-ème de u^n. En d'autres termes l'opérateur d'itération n'est pas borné, avec une *perte de régularité* égale à k. Par conséquent l'itération diverge.

Dans le cas discret, ceci a pour conséquence une amplification des hautes fréquences, sauf pour de très petites longueurs du pas ρ, qui dépendent du maillage.

Les expériences montrent que ce problème est très fréquent en Optimisation de Formes. Dans ce chapitre, nous proposons une explication de ce problème, liée au manque de régularité de la dérivée de Hadamard de la fonction par rapport au domaine.

Comme conséquence pratique, les gradients discrets sont oscillatoires. Afin de résoudre ce problème, Jameson a proposé dans [Reuther and Jameson, 1995] une méthode de correction par lissage qui résoud un système de Laplace-Beltrami sur le bord. Cette méthode est aussi utilisé dans [Mohammadi and Pironneau, 2001]. Dans [Arian and Ta'asan, 1999], Arian et Ta'asan utilisent le symbole de Fourier du hessien fonctionnel de type Hadamard. Selon le degré de ce hessien, ces auteurs proposent d'utiliser soit un opérateur de Laplace-Beltrami lorsque la perte de régularité est deux, soit un opérateur pseudo-différentiel Dirichlet Neumann lorsque la perte de régularité est un. Cette dernière méthode est plus coûteuse puisqu'on doit résoudre des systèmes 3D.

Dans ce chapitre, nous proposons de pousser plus loin l'idée d'Arian et de Ta'asan en utilisant une famille de préconditionneurs pour laquelle le gain de régularité est un paramètre qui peut être spécifié selon le problème à résoudre. Pour cela, nous adaptons le préconditionneur multiniveau additif de Bramble-Pasciak-Xu (BPX) (voir [Bramble *et al.*, 1990; Xu, 1997; Cohen, 2000]).

Afin de relier sa forme avec la perte de régularité, nous le présentons directement sous un point de vue fonctionnel. Quelques propriétés intéressantes de sa version fonctionnelle peuvent aisément être déduites à partir de résultats existant.
La variante discrète est adapté aux maillages non-structurés en appliquant un principe d'agglomération [Lallemand *et al.*, 1992].
Il reste à étudier comment ce préconditionneur peut être utilisé pour résoudre des problèmes d'optimisation. Partant d'exemples 1D, nous examinons d'abord l'introduction de cette sorte de préconditionneurs dans une classe populaire des méthodes d'optimisation quasi Newton, celle des méthodes SQP reposant sur les techniques BFGS.
Puis nous considérons l'introduction de notre méthode dans une boucle d'optimisation de forme pré-industrielle, introduite dans [Vazquez *et al.*, 2002]. A l'aide de cette boucle d'optimisation, nous résolvons des problèmes d'optimisation de formes d'un avion supersonique afin de réduire le bang sonique émis par la voilure.

3.2 Préconditionnement fonctionnel

Les problèmes de grande taille issus des Équations aux Dérivées Partielles ont généralement un mauvais conditionnement qui, de plus, se dégrade lorsque le nombre de degrés de liberté augmente. Pour expliquer et résoudre ce problème, nous pouvons soit analyser directement le comportement des valeurs propres discrètes lorsque le nombre d'inconnues augmente ou bien analyser le problème continu -fonctionnel- et la version continue de l'algorithme à appliquer. Nous nous concentrons sur la seconde méthode.

3.2.1 Optimisation d'un problème modèle

Considérons la minimisation de la fonctionnelle suivante :

$$j \ : \quad H_0^1(0,1) \to I\!R \tag{3.1}$$

$$j(u) \ = \frac{1}{2}\int_0^1 (\ k|grad\ u|^2 \ - \ f\ u)\ dx \tag{3.2}$$

où k est une application positive régulière sur $[0,1]$. Nous observons d'abord que l'itération *formelle de gradient*

$$u^{n+1} \ = \ u^n \ - \ \rho(div\ k\ grad\ u^n \ - \ f) \tag{3.3}$$

produit un nouvel itéré u^{n+1} qui n'est pas assez régulier pour calculer $div\ k\ grad\ u^{n+1}$ lors de l'itération suivante puisque $div\ k\ grad\ u^n$ n'est généralement pas dans $H_0^1(0,1)$. Ce sont de mauvaises nouvelles pour la version discrète :

$$u_h^{n+1} \ = \ u_h^n \ - \ \rho((div\ k\ grad)_h\ u_h^n \ - \ f_h)\ . \tag{3.4}$$

L'itération (3.4) est essentiellement une **itération de Jacobi**. Son pas ρ et la convergence dépendent fortement de la finesse du maillage :

Révenons à (3.3), nous pouvons introduire l'isomorphisme canonique associé à la norme usuelle H^1 :

$$\Lambda \ : \ (H_0^1(0,1))' \longrightarrow H_0^1(0,1)) \tag{3.5}$$

défini comme suit :

$$(g,u)_{(H_0^1(0,1)))' \times H_0^1(0,1))} \ = \ <\Lambda g, u>_{H_0^1} \ = \ \int (grad\ \Lambda g).(grad\ u)\ dx \tag{3.6}$$

et calculé explicitement en résolvant l'*équation de Poisson* :

$$div\ grad\ \Lambda g \ = \ g \ \ on \ \ \Omega, \ \ \Lambda g \ = \ 0 \ \ on \ \ \partial\Omega. \tag{3.7}$$

Alors la méthode de *gradient fonctionnel* s'écrit :

$$u^{n+1} \ = \ u^n \ - \ \rho\Lambda j'(u^n), \tag{3.8}$$

ce qui donne :

$$u^{n+1} \ = \ u^n \ - \ \rho\Lambda(div\ k\ grad\ u \ - \ f). \tag{3.9}$$

L'itération (3.9) va de H_0^1 dans H_0^1 c'est un algorithme de Moindres Carrés Fonctionnels selon la terminologie [Bristeau *et al.*, 1979].

La motivation principale pour faire cela est que dès qu'il converge, l'algorithme continu a un taux de convergence qui ne dépend évidemment pas de la taille du maillage. Nous pouvons alors essayer de construire une discrétisation consistante de cet algorithme qui convergera avec un taux pas si différent du taux continu.

Ceci signifie que des taux *essentiellement maillage-indépendants* pourraient être obtenus. Une autre manière de comprendre ce point est de se rappeler que dans le cas linéaire périodique, l'analyse de Fourier montre que les opérateurs contenant des dérivées spatiales auront de grandes valeurs propres de l'ordre de l'inverse de la taille de maille à la puissance le degré de la différentiation spatiale. Le plus petit sera ce degré sera, le meilleur sera le conditionnement. Il est clair que les deux degrés de différentiation spatiale perdus par (3.3) sont compensés dans (3.9) par l'introduction de Λ.
En d'autres termes, nous devons modifier l'itération de sorte, qu'**après conditionnement, l'ordre résultant de la différentiation spatiale soit égal à zéro**.

3.2.2 Un exemple : La formule de Hadamard en elliptique

Nous voulons maintenant minimiser une fonctionnelle $j(\gamma)$ dépendant d'un "paramètre de forme du domaine" γ à travers la solution d'une E.D.P. calculée sur le domaine. Plus précisément, dans un problème d'optimisation de formes, une géométrie initiale Ω_0 possède une direction V normale à son bord. Une famille de domaines Ω_γ de R^d est paramétrisée par un déplacement $\gamma \in \mathcal{C}^{l+\alpha}(\partial\Omega_0)$ du bord dans la direction normale V.

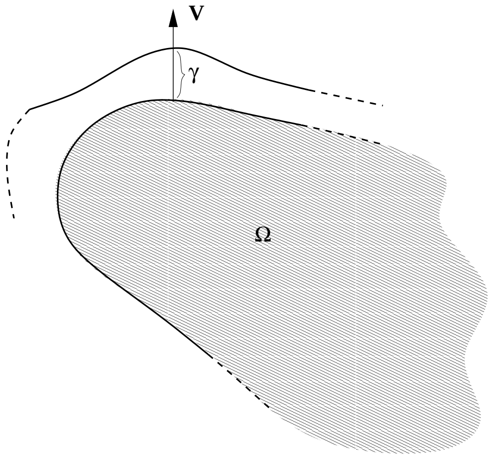

FIG. 3.1 – Paramétrisation du bord.

L'équation d'état est le problème de Poisson :

$$-div\ grad\ z(\gamma) \ = \ f \ \ ; \ \ z(\gamma) \ = \ 0 \quad on \quad \partial\Omega_\gamma\ .$$

Soit D un sous-domaine de Ω_γ (inclus dans Ω_γ pour tout γ admissible). La fonctionnelle j à minimiser est définie par :

$$j(\gamma) \;=\; \frac{1}{2}||z(\gamma) - z_{cible}||_D^2 \;.$$

Une difficulté classique pour calculer le gradient de j vient du fait que le domaine est variable dans l'équation d'état. Nous choisissons une famille de difféomorphismes $(T_\gamma)_\gamma$ tels que

$$T_\gamma \;\; envoie \;\; \Omega_0 \;\; sur \;\; \Omega_\gamma,$$
$$T_\gamma \;\; est\ l'identit\acute{e} \;\; sur \;\; D \;.$$

Il est alors possible d'adapter certains résultats bien connus reliés à la méthode des “variations intérieures” de Garabedian (voir [Garabedian, 1964], [Murat and Simon, 1974], [Pironneau, 1984], [Dervieux, 1981], [Dervieux and Palmerio, 1975]) et de montrer, sous des hypothèses de régularité sur f et $\partial\Omega_0$, que l'application :

$$\gamma \longmapsto z(\gamma)|_D$$

est continument différentiable de $\mathcal{C}^{l+\alpha}(\partial\Omega_0)$ dans $\mathcal{C}^{l-1+\alpha}(D)$.
En fait le gradient de cette application est la solution d'un problème de Dirichlet avec une condition de Dirichlet non-homogène exprimée comme fonction de la dérivée normale de $z(\gamma)$ et de la perturbation du bord. Nous reconnaissons dans l'application :

$$\gamma \longmapsto \frac{\partial z}{\partial n}(\gamma)|_{\partial\Omega_\gamma}$$

un opérateur pseudo-différentiel proche du classique Dirichlet-Neumann. Par la formule de dérivation composée, j est aussi différentiable et son gradient est exprimé comme suit :

$$j'(\gamma).\delta\gamma \;\;=\;\; \int_{\partial\Omega_\gamma} \frac{\partial z(\gamma)}{\partial n_\gamma} \; \frac{\partial p(\gamma)}{\partial n_\gamma} \;\; < n_\gamma, n_0 > \delta\gamma \;\; d\partial\Omega_\gamma$$

où n_γ est la normale à $\partial\Omega_\gamma$, et $p(\gamma)$ l'état adjoint suivant :

$$-div\ grad\ p(\gamma) \;=\; z_{cible} - z(\gamma) \;\;;\;\; p(\gamma) \;=\; 0 \quad on \quad \partial\Omega_\gamma \;.$$

Considérer L^2 comme espace pivot pour notre méthode de gradient produit la direction de descente suivante :

$$g_{L^2}(\gamma) \;\;=\;\; \frac{\partial z(\gamma)}{\partial n_\gamma} \; \frac{\partial p(\gamma)}{\partial n_\gamma} \;\; < n_\gamma, n_0 > \;\; .$$

Nous observons que le paramètre γ, appartient à $\mathcal{C}^{l+\alpha}(\partial\Omega_0)$, cette correction est seulement de régularité $\mathcal{C}^{l-1+\alpha}(\partial\Omega_0)$. Ce n'est pas assez régulier si nous voulons itérer dans $\mathcal{C}^{l+\alpha}(\partial\Omega_0)$. Pour ce choix d'espaces de Banach fonctionnels, notre analyse montre donc que la perte de régularité est de 1.

Remarque 1 : Pour appliquer une itération de gradient, nous pourrions préfèrer un contexte d'espaces de Hilbert. Nous pouvons travailler par exemple avec des espaces de Sobolev. Dans ce cas, nous n'avons pas d'analyse optimale. A partir du théorème d'injection de Sobolev, nous avons une application différentiable :

$$j : H^{l+\frac{d-1}{2}+\varepsilon+\alpha}(\partial\Omega_0) \to I\!R$$

où d est la dimension de l'espace et ε un nombre positif petit, et :

$$g_{L^2} \;\in\; H^{l-1+\alpha}(\partial\Omega_0).$$

Par conséquent, dans ce contexte de Hilbert, la perte de régularité entre γ et sa correction $g_{L^2}(\gamma)$ est au plus $1 + \frac{d-1}{2} + \varepsilon.\square$

L'exemple de cette section montre que dans les problèmes d'optimisation complexes, et similairement à l'exemple elliptique précédent, l'application directe sans isomorphisme canonique d'une méthode de gradient peut produire des itérations continues non-régulières, et donc peut induire dans la version discrète des comportements analogues aux solveurs elliptiques Jacobi. Comme dans la section 1, nous avons besoin d'un préconditionneur pour compenser la perte de régularité.

3.2.3 Formule de Hadamard et Euler

La différentiation rigoureuse par rapport au domaine géométrique dans le cas du modèle compressible des équations d'Euler est aujourd'hui hors de portée. Cependant, dans plusieurs travaux et en particulier dans [Beux and Dervieux, 1992], [Beux, 1993], une différentiation de Hadamard *formelle* est proposée. L'objet de cette partie est de rappeler brièvement la conclusion de ces études afin de l'appliquer à l'identification de la perte de régularité dans le contexte continu.

Nous considérons un problème d'optimisation de formes dans lequel le domaine Ω_γ de R^d est paramétrisé comme dans le paragraphe précédent.

Soit de nouveau D un sous-domaine de Ω_γ (inclus dans Ω_γ pour tout γ admissible). Nous considérons la minimisation de la fonctionnelle suivante :

$$j(\gamma) \;=\; \frac{1}{2}||W(\gamma) - W_{cible}||_D^2 \;.$$

L'équation d'état est maintenant le système des équations d'Euler stationnaires avec les conditions au bord appropriées. Elle est représentée dans sa forme variationnelle comme suit : pour tout $\phi = (\phi_1, \phi_2, \phi_3, \phi_4, \phi_5)$ appartenant à l'espace approprié,

$$\begin{aligned}(\Psi(\gamma, W), \phi) \;&=\; -\int_{\Omega_\gamma} (F(W)\frac{\partial \phi}{\partial x} + G(W)\frac{\partial \phi}{\partial y} + H(W)\frac{\partial \phi}{\partial z})\; d\Omega_\gamma \\ &+ \int_{\partial\Omega_\gamma} p\;(n_x^\gamma\;\phi_2 \;+\; n_y^\gamma\;\phi_3 \;+\; n_z^\gamma\;\phi_4)\; d\partial\Omega_\gamma = 0, \qquad (3.10)\end{aligned}$$

où les flux d'Euler sont $F(W)$, $G(W)$ et $H(W)$, correspondant respectivement à chacune des directions de l'espace. La normale extérieure $\vec{n}^\gamma$ à $\partial\Omega_\gamma$ a pour composantes $(n_x^\gamma, n_y^\gamma, n_z^\gamma)$.

On calcule **formellement** dans [Beux and Dervieux, 1992] la dérivée de Gâteaux de j en γ_0 dans dans la direction $\delta\gamma$:

$$\begin{aligned} j'(\gamma_0,\delta\gamma) &= -\int_{\partial\Omega_{\gamma_0}} (F(W)\frac{\partial\Pi}{\partial x} + G(W)\frac{\partial\Pi}{\partial y} + H(W)\frac{\partial\Pi}{\partial z})(\vec{n}^{\gamma_0}\cdot\vec{V})\ \delta\gamma d\partial\Omega_{\gamma_0} \\ &+ \int_{\partial\Omega_{\gamma_0}} (\nabla p\ \Pi\ +\ p\ \nabla\Pi)(\vec{n}^{\gamma_0}\cdot\vec{V})\ \delta\gamma d\partial\Omega_{\gamma_0} \end{aligned}$$

avec les notations suivantes,

$$\begin{aligned} \nabla p\ \Pi &= \frac{\partial p}{\partial x}\,\Pi_2\ +\ \frac{\partial p}{\partial y}\,\Pi_3\ +\ \frac{\partial p}{\partial z}\,\Pi_4, \\ p\ \nabla\Pi &= \frac{\partial \Pi_2}{\partial x}\,p\ +\ \frac{\partial \Pi_3}{\partial y}\,p\ +\ \frac{\partial \Pi_4}{\partial z}\,p. \end{aligned}$$

L'état adjoint Π est solution du système :

$$\left(\frac{\partial F}{\partial W}\right)^*\frac{\partial\Pi}{\partial x} + \left(\frac{\partial G}{\partial W}\right)^*\frac{\partial\Pi}{\partial y} + \left(\frac{\partial H}{\partial W}\right)^*\frac{\partial\Pi}{\partial z} \quad = \quad -\ (W(\gamma_0) - W_{cible})\chi_{\mathrm{D}} \quad \text{on} \quad \Omega_{\gamma_0},$$

où χ_{D} est la fonction caractéristique de D et avec les conditions au bord :

$$\Pi_2 n_x^{\gamma_0} + \Pi_3 n_y^{\gamma_0} + \Pi_4 n_z^{\gamma_0} \quad = \quad 0 \quad \text{on} \quad \partial\Omega_{\gamma_0}.$$

Considérer L^2 comme espace pivot pour notre méthode de gradient produit la correction suivante pour le paramètre de formes γ :

$$\gamma = \gamma_o\ -\ \rho g_{L^2}(\gamma_o, W, \Pi)$$

où :

$$\begin{aligned} g_{L^2}(\gamma_o, W, \Pi) = &\ -\ \ (F(W)\frac{\partial\Pi}{\partial x} + G(W)\frac{\partial\Pi}{\partial y} + H(W)\frac{\partial\Pi}{\partial z})\ (\vec{n}^{\gamma_0}\cdot\vec{V}) \\ &+\ \ (\nabla p\ \Pi\ +\ p\ \nabla\Pi)\ (\vec{n}^{\gamma_0}\cdot\vec{V})\ . \end{aligned} \tag{3.11}$$

Nous observons que cette correction est généralement beaucoup moins régulière que le paramètre de formes γ_0. En effet, nous inspirant de la régularité elliptique, nous pouvons estimer que les variables d'état sont au plus aussi régulières que le bord, mais la correction précédente contient des dérivées de l'état adjoint . De plus, le vecteur normal $\vec{n}^{\gamma_0}$ est aussi une dérivée du bord.

Le cas des conditions de transpiration est analysé dans [Vazquez *et al.*, to appear]. A nouveau les auteurs constatent une perte de régularité de degré un.

Ces exemples montrent que dans des problèmes complexes, l'application directe dans R^N d'une méthode de gradient, ce qui revient systématiquement à choisir l'espace L^2 comme espace pivot pour l'itération de gradient peut avoir pour conséquence des itérations fonctionnelles non régulières. De plus il apparait que la **perte de régularité est formellement 1**. Si aucun préconditionneur n'a été introduit afin de compenser la perte de régularité, la convergence montrera un comportement analogue à celui des solveurs Jacobi ou des solveurs elliptiques non-preconditionnés de type Jacobi, c'est-à-dire une convergence très lente et dépendante du maillage.

3.3 Préconditionneurs multiniveaux additifs

Les préconditionneurs multiniveaux additifs ont été initialement introduits dans un contexte discret afin de résoudre les équations aux dérivées partielles elliptiques qui sont typiquement d'ordre deux ou bien d'ordre impair. Une littérature extrêmement riche existe sur ce sujet. La méthode des bases hiérarchiques fut d'abord analysée par Yserentant dans son papier initiateur [Yserentant, 1986]. Le travail d'Yserentant fut apparemment motivé par le célèbre rapport technique non publié de Bank et Dupont [Bank and Dupont, 1980]. Une théorie plus complète fut proposée par Bramble, Pasciak and Xu [Bramble *et al.*, 1990], [J.Xu, 1989]. Voir aussi l'extension aux ondelettes, par exemple dans [Cohen, 2000]. Une théorie étendue sur les préconditionneurs multiniveaux peut être trouvée dans [Kunoth, 1994; Xu, 1997].
Le but de cette section est de rappeler les notations de base de [Xu, 1997] et de reformuler dans une version plus adaptée le résultat principal que nous voulons utiliser par la suite. Pour des raisons de simplicité, nous l'établissons dans le cas de conditions au bord de type Dirichlet.
Soit $V = H_0^1(\Omega)$ inclus dans $H = L^2(\Omega)$, le deuxième espace étant muni du produit scalaire et de la norme :

$$(u,v) \;=\; \int_\Omega u\ v\ \ dx; \quad \|u\| = \ (u,u)^{1/2}$$

Soit $(V_k)_{k=0,1,\ldots}$ une suite de sous-espaces de V :

$$V_0 \subset \cdots \subset V_k \subset \cdots \subset V$$

La suite d'espaces est supposée satisfaire certaines propriétés exprimant qu'ils sont de bons espaces d'approximation pour V, avec un rapport de maille de 2 entre V_k et V_{k-1}, voir [Xu, 1997]. La taille de maille uniforme dans V_k est supposée être :

$$h_k \;=\; 2^{-k}$$

Pour tout k, nous introduisons les opérateurs $Q_k : V \longrightarrow V_k$ définis pour tout $u \in V, v \in V_k$ par

$$(Q_k u, v) = (u, v) \ ; \ Q_{-1} = 0 \ . \tag{3.12}$$

Notation : Suivant les notations de [Xu, 1997], $x \ \approxeq \ y$ signifie que $cx \ \leq \ y \ \leq \ Cx$ pour des constantes C et c sont indépendantes de l'indice du niveau k.

Theorem 3.3.1
[Xu, 1997] :
Soit $s \ \in \ I\!R, \ \ -3/2 \ \leq \ s \ \leq \ 3/2.$

Pour tout $v \in H_0^s(\Omega)$, nous avons

$$\|v\|^2_{H^s(\Omega)} \approxeq \sum_{k=0}^{+\infty} \|(Q_k - Q_{k-1})v\|^2_{H^s(\Omega)} \approxeq \sum_{k=0}^{+\infty} h_k^{-2s} \|(Q_k - Q_{k-1})v\|^2.$$

□

Soit $a \in [0, 3/2]$ et B défini par :

$$B = \sum_{k=0}^{+\infty} \left(\frac{1}{2^a}\right)^k (Q_k - Q_{k-1}) \tag{3.13}$$

Soit u un élément de $H_0^s(\Omega)$ et s tel que $-3/2 \leq s \leq 3/2$.
Utilisant le théorème précédent, nous déduisons que

$$\begin{aligned}
\|u\|^2_{H^s(\Omega)} &\approxeq \sum_{k=0}^{+\infty} \|(Q_k - Q_{k-1})u\|^2_{H^s(\Omega)} \approxeq \sum_{k=0}^{+\infty} h_k^{-2s} \|(Q_k - Q_{k-1})u\|^2, \\
\|u\|^2_{H^s(\Omega)} &\approxeq \sum_{k=0}^{+\infty} h_k^{-2(s+a)} (h_k^a)^2 \|(Q_k - Q_{k-1})u\|^2, \\
\|u\|^2_{H^s(\Omega)} &\approxeq \sum_{k=0}^{+\infty} h_k^{-2(s+a)} \|(Q_k - Q_{k-1})Bu\|^2.
\end{aligned}$$

Utilisant de nouveau le théorème précédent, nous obtenons le résultat suivant :

Corollaire 3.3.2
Soit s tel que $-3/2 \leq s \leq s + a \leq 3/2$.
Alors l'opérateur B est borné de $H_0^s(\Omega)$ dans $H_0^{s+a}(\Omega)$ et pour tout $u \in H_0^s(\Omega)$, nous avons

$$\|Bu\|_{H^{s+a}(\Omega)} \approxeq \|u\|_{H^s(\Omega)} .$$

□

Cet énoncé montre que l'opérateur B a des propriétés de régularisation (de lissage) qui peuvent être quantifées comme un gain en régularité. Ce gain est exactement a. Il peut être prescrit par l'utilisateur afin de préconditionner un opérateur ayant a comme perte de régularité. Grace à ce point de vue continu, la seule adaptation que nous souhaitons appliquer au préconditionneur multiniveau pour le rendre adéquat à un opérateur donné est de choisir un exposant a qui compense la perte de régularité dans cet opérateur. Nous n'avons pas encore défini précisément les V_k. Nous introduisons dans les sections suivantes une construction non classique de ces espaces.

3.4 Préconditionneur 1D par agglomération des noeuds

La méthode d'agglomération pour construire des préconditionneurs multiniveaux additifs s'appliquant aux équations est introduite dans [N. Marco and Dervieux, 1997]. Le but de ce paragraphe est de considérer son adaptation en 1D aux problèmes d'optimisation.

3.4.1 Construction de l'opérateur

Considérons une suite de discrétisations de $[0,1]$ obtenues en divisant l'intervalle en 2^k sous-intervalles, avec $1 \leq k \leq N$.
Soit $h = \frac{1}{2^k}$ et $\mathcal{M}_h$. Nous nous concentrons sur la construction de l'opérateur de transfert entre $\mathcal{M}_h$ et $\mathcal{M}_{2h}$.

En premier lieu nous définissons une suite d'**espaces non-lisses**. Chaque fonction sur $[0,1]$ que nous considérons est définie par ses valeurs sur les *noeuds "fin"* du niveau N et une interpolation linéaire continue sur le maillage fin du niveau N. Soit j une cellule grossière, elle contient deux cellules fines, $2j$ et $2j-1$. Nous définissons $\overline{P}$ l'opérateur de prolongement linéaire de $E_{2h} = V_{k+1}$ dans $E_h = V_k$ comme suit :

$$\forall u_{2h} \in E_{2h} \quad (\overline{P}u_{2h})_{2j} \;=\; (\overline{P}u_{2h})_{2j-1} \;=\; (u_{2h})_j \tag{3.14}$$

Le transfert ci-dessus définit des espaces non-lisses générés par des **fonctions en escalier**. L'opérateur adjoint, $\overline{P}^*$, est l'opérateur de restriction de E_h dans E_{2h}. Si deux cellules fines, $2j$ et $2j-1$, sont incluses dans la cellule grossière j, nous posons :

$$(\overline{P}^* u_h)_j = \frac{1}{2}\left[(u_h)_{2j-1} \;+\; (u_h)_{2j}\right] \;. \tag{3.15}$$

Nous nous restreignons au cas des opérateurs différentiels (ou pseudo-différentiels) de degré compris entre zéro et deux. Par contraste avec la suite standard de discrétisations internes emboitées, la suite précédente d'espaces générés avec les transferts $\bar{P}$ et $\bar{P}^*$ ne vérifient pas les propriétés nécessaires de régularité pour construire un préconditionneur multiniveau efficace pour les opérateurs d'ordre deux. Ce point est discuté en détails dans [N. Marco and Dervieux, 1997]. Ceci peut être lié au fait que les ordres de précision des transferts ont une somme égale à $1+1=2$, non strictement plus grande que l'ordre 2 de l'opérateur, voir [Hackbusch, 1985].
Pour compenser ce défaut, on définit des **opérateurs de lissage**. Il est proposé dans [N. Marco and Dervieux, 1997], de recupérer la régularité nécessaire en introduisant un "opérateur de moyenne" L et son adjoint L^* par rapport au produit scalaire discret L^2.

$$(Lu)_j \;=\; \frac{1}{4}\,u_{j-1} \;+\; \frac{1}{2}\,u_j \;+\; \frac{1}{4}\,u_{j+1} \tag{3.16}$$

et en introduisant l'espace :

$$V_k \;=\; L_k\,\bar{P}_k\,(V_{k+1}) \tag{3.17}$$

Ce qui revient à définir l'opérateur de (3.12) comme suit :

$$Q_m \;=\; \Pi_{k=0}^{m-1} L_k\,\bar{P}_k\,\Pi_{k=m-1}^{k=0}\,\bar{P}_k^*\,L_k^*. \tag{3.18}$$

Le préconditionneur s'écrit :

$$B \quad = \quad \sum_{k=0}^{N} \left(\left(\frac{1}{2^a} \right)^k \quad - \quad \left(\frac{1}{2^a} \right)^{k-1} \right) Q_k \ . \tag{3.19}$$

3.4.2 Quelques expériences

Nous voulons insister sur le fait que ce type de préconditionneur peut être utilisé pour résoudre un système linéaire à l'intérieur d'un gradient conjugué aussi bien que pour trouver un minimum en combinaison avec un algorithme de minimisation quasi-Newton. Dans le premier cas, nous utilisons une seule fois par itération le **préconditionneur d'ordre deux** ($a = 2.$) à l'intérieur d'un gradient conjugué Polak-Ribière pour résoudre l'équation de Laplace. La figure 3.2 montre que la valeur optimale du paramètre a est bien 2, et que pour cette valeur optimale nous obtenons un facteur de convergence indépendant du maillage.

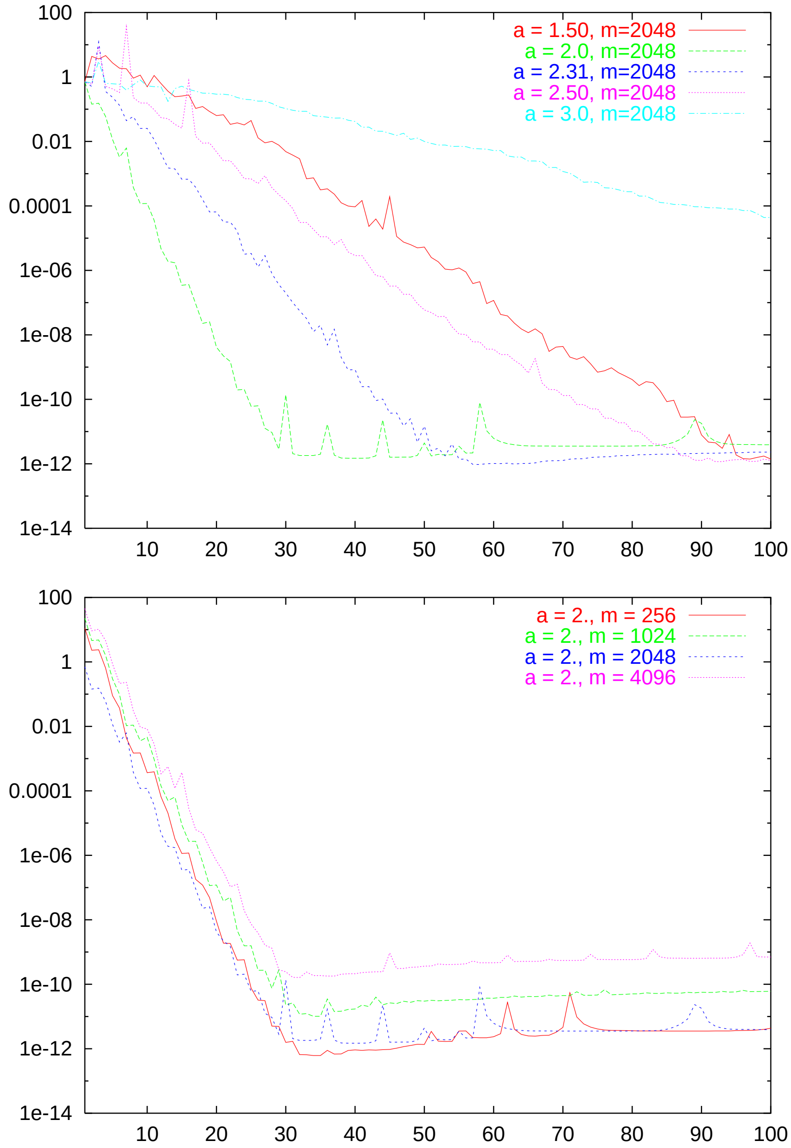

FIG. 3.2 – *Résolution du problème de Poisson avec un GC Polak-Ribiere préconditionné : L'optimalité de a = 2. est mise en évidence(haut) avec un convergence indépendante du maillage pour ce paramètre (bas)*

Dans le second cas -optimisation-, nous appliquons un optimiseur quasi-Newton SQP-BFGS (voir par exemple dans [Nocedal and Wright, 1999]) sur le même problème. La nouveauté est que, pour mettre à jour à la fois le hessien approché et son inverse dans la fomule BFGS, nous avons besoin d'utiliser des changements de variables. Ils consistent en un **préconditioneur d'ordre un** et son inverse, construits respectivement avec a égal à 1 et -1. Sur la figure 3.3 quatre expériences sont décrites. Le nombre d'inconnues est d'abord fixé à $m = 32$. Comme prévisible le SQP-BFGS standard trouve l'optimum en à peu près $5 + m/2$ itérations, c'est-à-dire en 21 itérations. La version préconditionnée a une convergence plus progressive, mais finalement pas plus rapide. En revanche pour un nombre d'inconnus plus important, $m = 256$, le SQP-BFGS non-préconditionné présente une convergence très lente. Par contraste, la convergence du SQP-BFGS préconditionné est peu près la même que celle obtenue avec le maillage grossier.

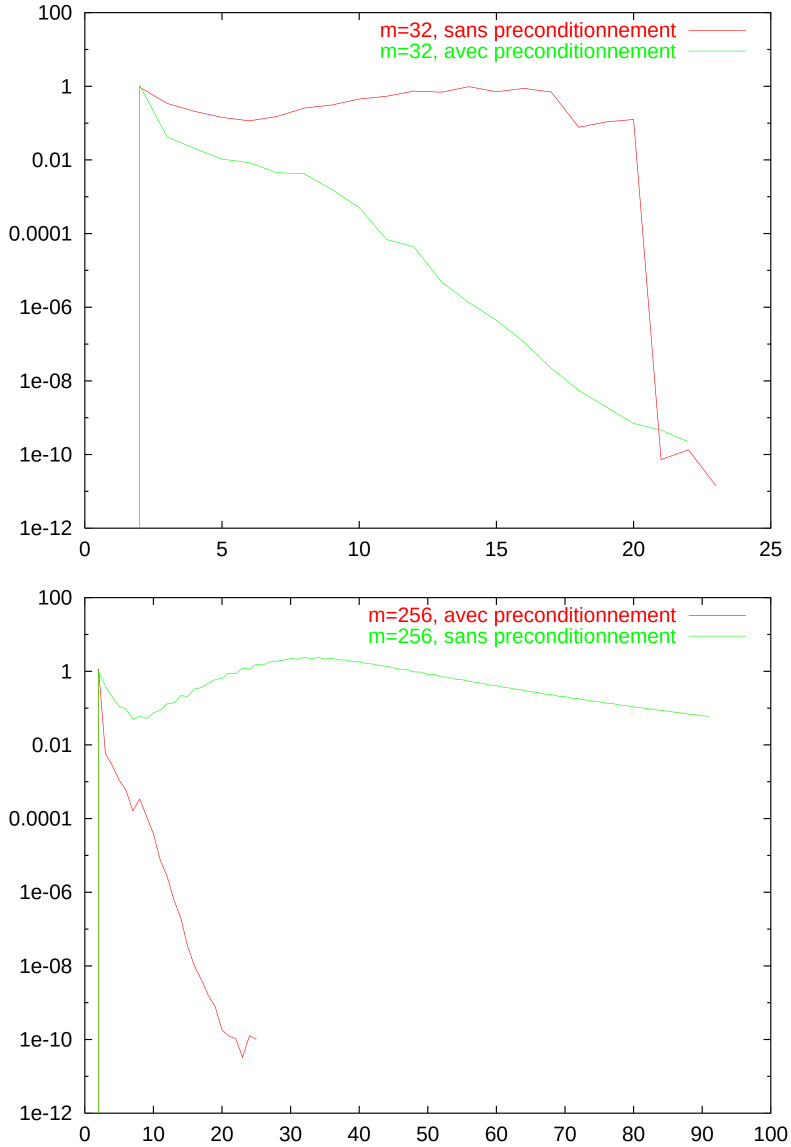

FIG. 3.3 – *Résolution d'un problème d'optimisation d'ordre deux avec SQP-BFGS préconditionné, comportement de différents résidus en fonction du nombre d'itérations, pour des problèmes de taille* $m = 32$ *and* $m = 256$*.*

3.5 Maillages non-structurés

Nous donnons maintenant la définition du préconditionneur multiniveau additif basé sur l'agglomération pour des maillages multidimensionnels non-structurés. Le premier paragraphe décrit le cas d'un maillage 2D ou 3D. Le deuxième paragraphe traite le cas particulier d'une inconnue définie sur une surface non-plane discrétisée par des triangles en 3D.

3.5.1 Agglomération multidimensionnelle

Nous rappelons brièvement les définitions des opérateurs de transfert $\overline{P}$ et $\overline{P}^*$ entre deux niveaux, et les définitions de L et L^*. Nous débutons avec une triangulation fine ou une tétrahédrisation. Les nœuds sont placés aux sommets. Le niveau fin discret est le sous-espace des combinaisons linéaires de fonctions chapeau P_1 : $E_h = span\{\varphi_i,\ i = 1 \cdots n_h\}$. E_h est muni du produit scalaire diagonalisé suivant, i.e. qui est pondéré par les mesures des cellules duales $\mathcal{M}es\ (i)$:

$$\forall\ u_h\ et\ v_h \in E_h \qquad (u_h, v_h)_h = \sum_i^{n_h} (u_h)_i\ (v_h)_i\ \mathcal{M}es\ (i)$$

Le processus d'agglomération repose sur une partition de l'ensemble $I^f = \{1, \cdots, i, \cdots, n_h\}$ des indices fins i.

$$I^f = I_1 \cup \cdots \cup I_J \cup \cdots \cup I_{n_{2h}} \qquad (n_{2h} \ll n_h) \tag{3.20}$$

où tout I_J contient les indices de quelques noeuds voisins. Un algorithme pour construire une telle partition a été proposé dans [Lallemand *et al.*, 1992].

Pour tout I_J, une fonction de base plus grossière est définie par : $\Phi_J = \sum_{i \in I_J} \varphi_i$ et l'espace plus grossier est donné par

$$E_{2h} = Span\{\Phi_J,\ J = 1 \cdots n_{2h}\}$$

L'opérateur linéaire de prolongement, $\overline{P}$, de E_{2h} dans E_h est défini par

$$\forall u_{2h} \in E_{2h} \quad \overline{P} u_{2h} = u_{2h}\ \in\ E_h\ . \tag{3.21}$$

Son adjoint $\overline{P}^*$ est l'opérateur de restriction de E_h dans E_{2h}, et il est défini via $\overline{P}$ comme son adjoint pour le produit scalaire précédent par :

$$(\overline{P}^* u_h)_J = \frac{\displaystyle\sum_{j_m \subset J}^{n_h} (u_h)_{j_m} \mathcal{M}es(j_m)}{\mathcal{M}es(J)} \tag{3.22}$$

où J est une cellule grossière, j_m sont des cellules fines incluses dans J. $\mathcal{M}es(J)$ représente la mesure de la cellule grossière J . Similairement au cas 1D, nous avons besoin d'introduire

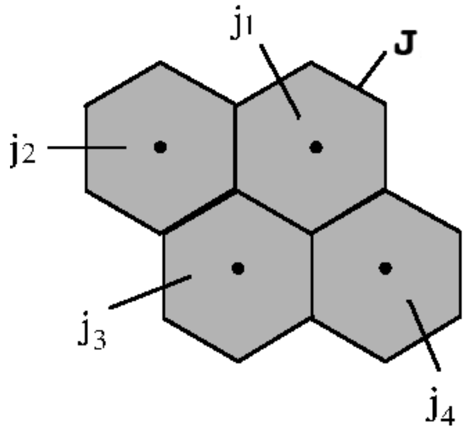

FIG. 3.4 – *Cellule grossière obtenue par agglomération de quatre cellules fines*

un opérateur de lissage. Cet opérateur de "lissage" est une moyenne entre un nœud et ses voisins. Il s'écrit :

$$(Lu)_i = (1-\theta)u_i + \theta \frac{\displaystyle\sum_{j\in\mathcal{N}(i)\cup\{i\}} \mathcal{M}es(j)\ u_j}{\displaystyle\sum_{j\in\mathcal{N}(i)\cup\{i\}} \mathcal{M}es(j)} \tag{3.23}$$

où $\mathcal{N}(i)$ représente l'ensemble des voisins de la cellule i et θ est un paramètre de lissage. L'adjoint L^* de L est défini par :

$$(L^*u)_i = (1-\theta)u_i + \theta \sum_{j\in\mathcal{N}(i)\cup\{i\}} \frac{u_j\ \mathcal{M}es(j)}{\displaystyle\sum_{k\in\mathcal{N}(j)\cup\{j\}} \mathcal{M}es(k)}\ . \tag{3.24}$$

Le reste de la construction du préconditionneur suit les règles (3.18), (3.19).

3.5.2 Agglomération pour une surface en 3D

Nous retournons aux notations de la Section 2.2.

Soit Σ_0 la surface 3D initiale, composée de triangles. La surface discrète générique Σ_γ, est définie par la translation de longueur γ, des nœuds de Σ_0 suivant un vecteur unité approché $\vec{n}$ normal à Σ_0 défini aux nœuds.

$$\vec{x}_i^\gamma \text{ est un nœud de } \Sigma_\gamma \quad \Leftrightarrow \quad \vec{x}_i^\gamma = \vec{x}_i^o + \gamma(i)\ \vec{n}$$

où i est l'indice du nœud, $\vec{x}_i^o$ est la position physique du noeud de Σ_0 avec le même indice i. Afin de préconditionner une correction de γ, nous construisons une suite d'espaces et d'opérateurs suivant le même processus que dans (3.20, 3.21, 3.22), mais restreinte à la surface et avec l'aire des cellules surfaciques $\mathcal{A}rea(j)$ au lieu des mesures des cellules. Afin d'adapter nos opérateurs aux surfaces irrégulières, l'opérateur de lissage L, est pondéré par un produit scalaire des normales à la surface :

$$(L\ \vec{x})_i = (1-\theta)\vec{x}_i + \theta\ \frac{\displaystyle\sum_{j\in\mathcal{N}(i)\cup\{i\}} w_{ij}\vec{x}_j}{\displaystyle\sum_{j\in\mathcal{N}(i)\cup\{i\}} w_{ij}}$$

où les w_{ij} sont les poids définis par :

$$w_{ij} = \max\ (\mathcal{A}rea(i)\vec{n}_i \cdot \mathcal{A}rea(j)\vec{n}_j\ ,\ 0) \quad \|\vec{n}_i\| = 1 \ \ \forall\ i\ .$$

Avec cette formule, le lissage est annulé sur les dièdres de la géométrie (e.g. sur les bords de fuite de l'aile). Le reste de la définition du préconditionneur est la même que dans la section précédente. Le paramètre de lissage θ sera pris en pratique égal à $\frac{1}{2}$. Puisque les formes paramétrisées à considérer sont des courbes fermées, nous n'avons besoin d'aucune condition au bord.

3.6 Application à un problème d'optimisation de formes en aérodynamique

Nous décrivons maintenant l'adaptation de notre préconditionneur multiniveau à une boucle d'optimisation de formes en aérodynamique. Cette boucle a été présentée et motivée du point de vue de la Mécanique des Fluides Numériques dans le chapitre 1 de cette partie.

Nous avons utilisé le préconditionneur pour une optimisation de formes en partant de l'aile ONERA M6 (Fig. 3.5).

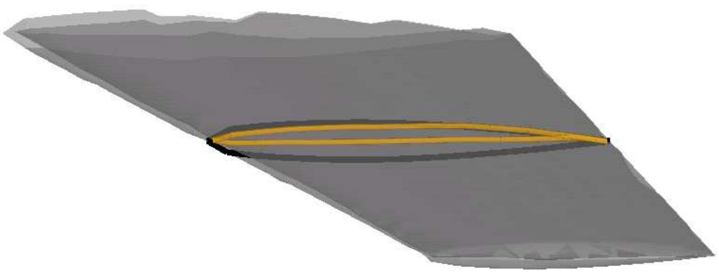

FIG. 3.5 – *Optimisation de la forme d'une aile ONERA M6 : esquisse de la géométrie initiale et de la forme finale.*

C'est l'occasion de vérifier si le paramètre optimal a est bien prédit par la théorie. Nous étudions d'abord cette optimalité pour une géométrie grossière, comprenant 2203 nœuds pour un maillage 3D, mais comprenant cependant 780 paramètres de forme.

Deux algorithmes d'optimisation sont utilisés, une méthode de gradient (Fig.3.6) et une méthode de gradient conjugué (Fig.3.7).

La convergence de l'itération de gradient peut être évaluée avec la norme du gradient. Le cas sans préconditionnement est le cas $a = 0$. La convergence est alors la plus lente des différentes options testées. Des valeurs comme $a = 0.5$, $a = 1.5$, $a = 1.$, $a = 2.$ donnent de bons taux de convergence, en particulier pour les 10 premières itérations. Ce point est important pour les boucles d'optimisation de formes qui, en pratique, sont coûteuses en temps CPU pour permettre plus d'itérations d'optimisation. Mais la valeur théorique $a = 1.$ apparait comme numériquement optimale, pour les deux cas, celui de l'optimiseur avec un pure gradient aussi bien que celui avec un gradient conjugué.
Vérifions que cette efficacité du préconditionneur est bonne pour des discrétisations plus fines. Nous considérons un deuxième maillage avec 15463 noeuds. Le nombre de paramètres de formes est 3222. Voir Fig.3.8 et Fig.3.9.
La valeur $a = 1.$ apparait de nouveau comme étant numériquement optimale. Le facteur de convergence est de nouveau bon (le gain CPU est de 4) et dans le cas du gradient conjugué, une bonne forme est obtenue en environ 12 itérations.
L'examen du cas $a = 2.$ appelle quelques commentaires. En effet, il est équivalent au lissage de Laplace-Beltrami. Tous les résultats précédents montrent que cette option n'est pas mauvaise, mais n'est clairement pas aussi bonne que l'option optimale $a = 1.$.
Les comparaisons précédentes sont résumées dans les tableaux 3.1 et 3.2.
Dans [Vazquez *et al.*, to appear] le préconditionneur proposé ici est appliqué avec succès à des géométries plus complexes.

TAB. 3.1 – *Minimisation du bang sonique : gradient conjugué, 10 itérations*

Cas	Itérations d'opt.	fonct	|gradient|
GC, N=780, M=11,000	10	0.02	0.3
GC préconditionné, N=780, M=11,000	10	0.04	0.04
GC, N=3222, M=77,000	10	0.06	0.3
GC préconditionné, N=3222, M=77,000	10	0.03	0.1

TAB. 3.2 – *Minimisation du bang sonique : division du coût par 8*

Cas	Nombre d'itérations d'opt.	|gradient|	Gain
GC, N=780, M=11,000	15	0.1	
GC préconditionné, N=780, M=11,000	3	0.1	5
GC, N=3222, M=77,000	100	0.03	
GC préconditionné, N=3222, M=77,000	20	0.04	5

3.7 Conclusion

Ce chapitre étudie une nouvelle stratégie de préconditionnement pour l'optimisation de formes. On construit un préconditionneur multiniveau additif à partir de (a) le principle classique de Bramble-Pasciak-Xu et (b) le principe d'agglomération. Des considérations d'analyse fonctionnelle montrent le rôle central de la perte de régularité dans l'itération de gradient. Celle-ci doit être compensée par le gain de régularité du préconditionneur. Dans la méthode proposée, ce gain de régularité est aisément spécifié par un seul paramètre réel.

Nous montrons que cette caractéristique aide à combiner le préconditionneur avec une accélération BFGS.

Dans le cas de l'optimisation de formes, nous montrons un exemple simplifié dans lequel la perte de dérivation peut être évaluée rigoureusement et est égale à 1.

L'application finale qui est décrite est une application pré-industrielle déjà résolue par différentes méthodes [Courty and Dervieux, 2003]. Les résultats (i) confirment l'analyse *a priori* de la perte de régularité et (ii) prouvent que la méthode proposée améliore notablement l'efficacité de l'outil d'optimisation de formes.

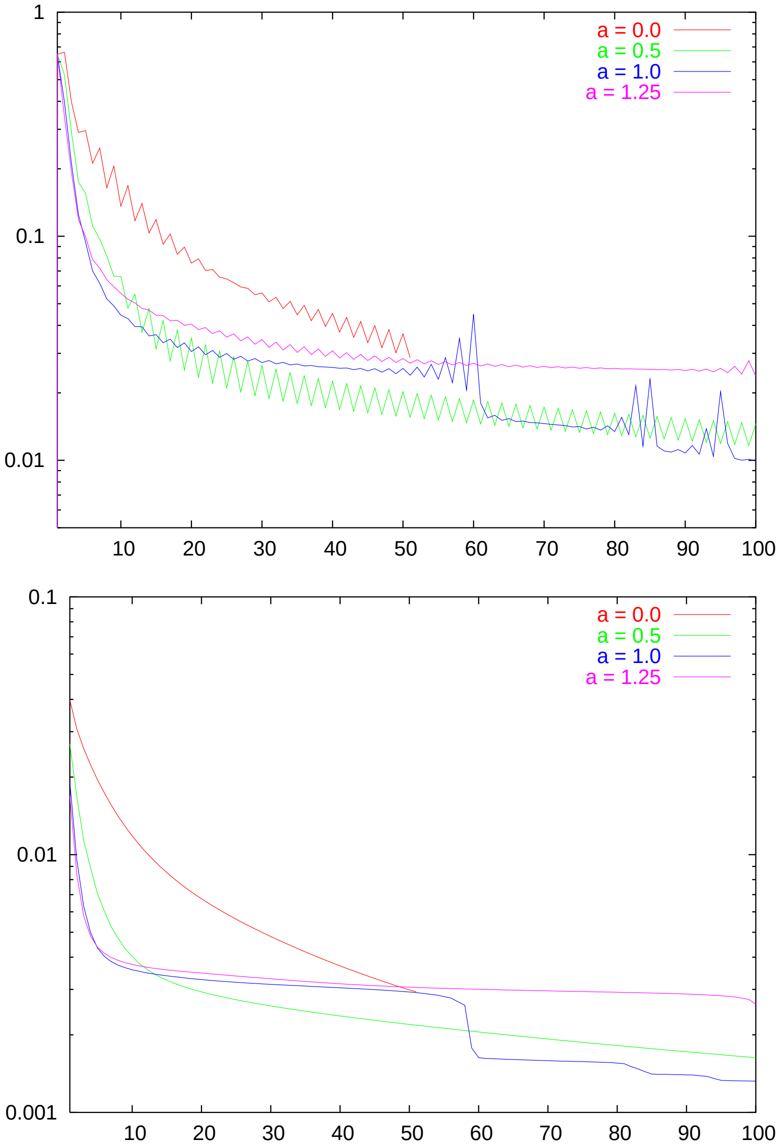

FIG. 3.6 – *Optimisation de la forme d'une aile ONERA M6 avec une méthode de gradient : Convergence du gradient (Haut), Fonction de coût (Bas), ns = 2203*

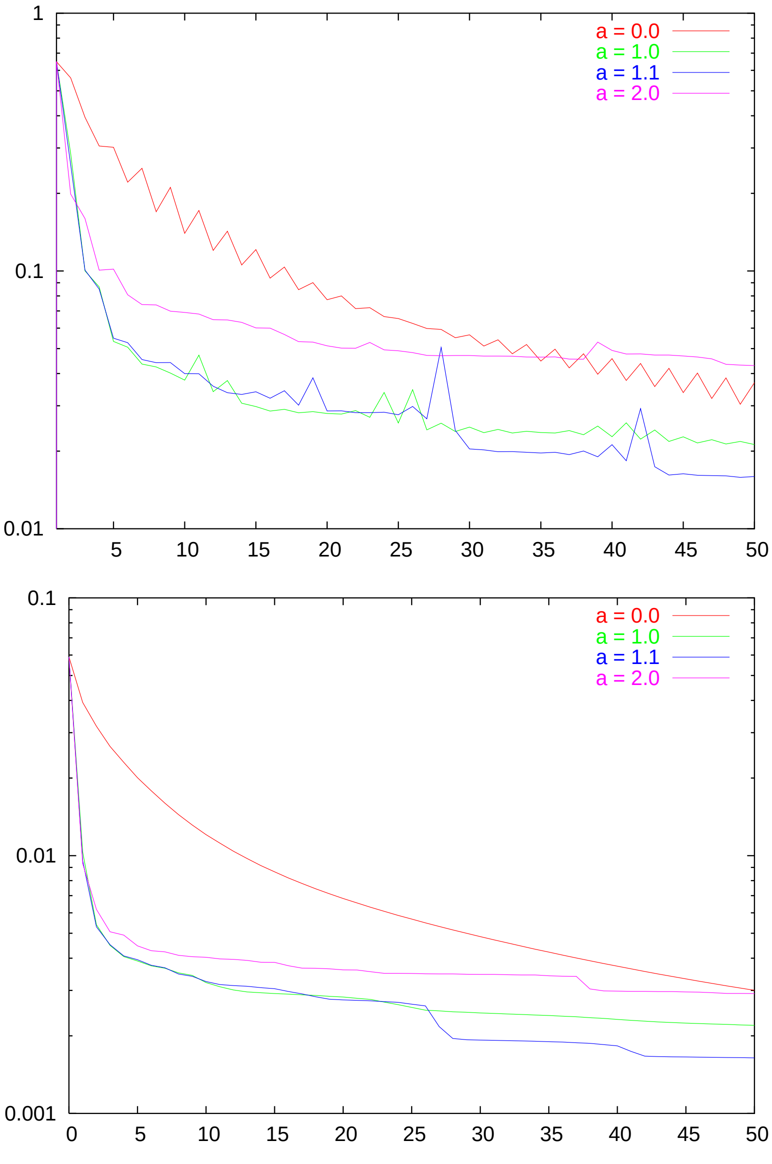

FIG. 3.7 – *Optimisation de la forme d'une aile ONERA M6 avec une méthode de gradient conjugué : Convergence du gradient (Haut), Fonction de coût (Bas), ns = 2203*

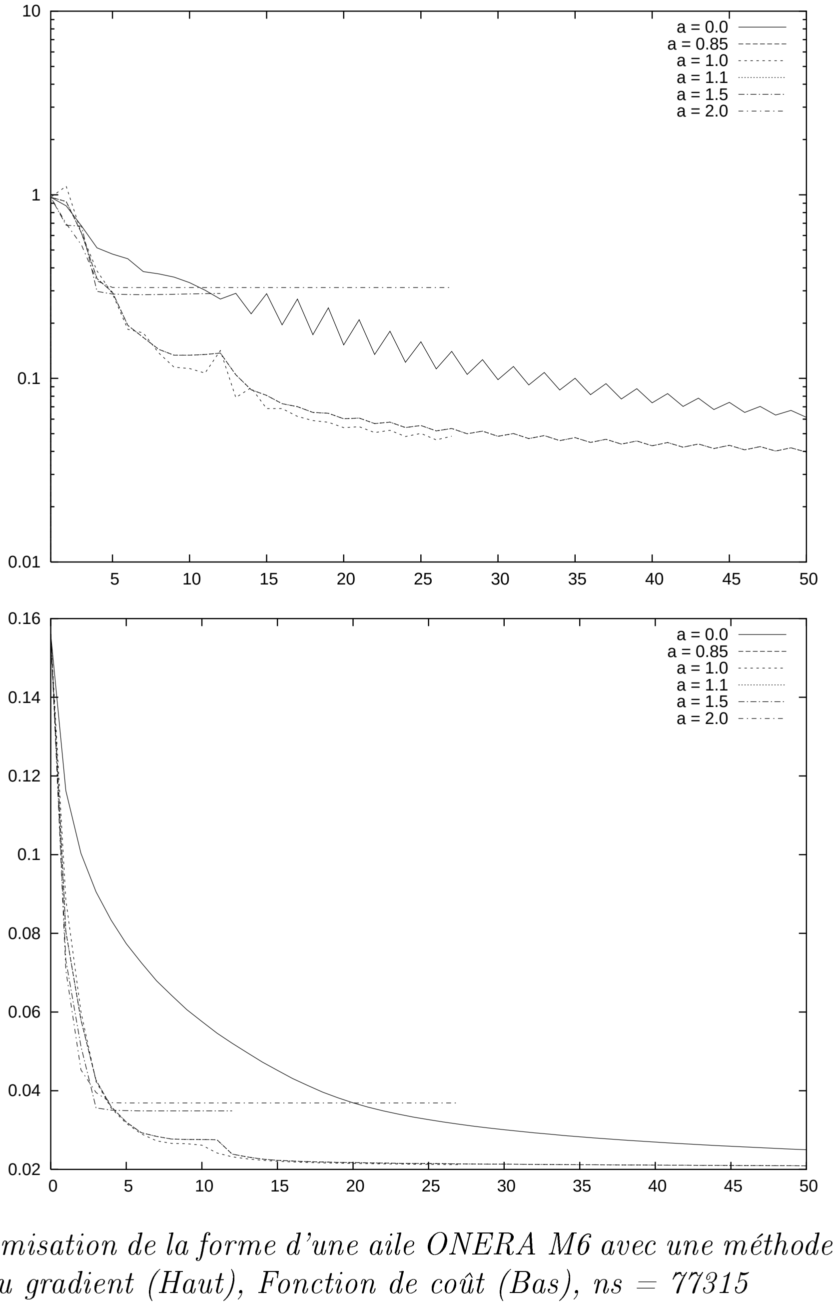

FIG. 3.8 – *Optimisation de la forme d'une aile ONERA M6 avec une méthode de gradient : Convergence du gradient (Haut), Fonction de coût (Bas), ns = 77315*

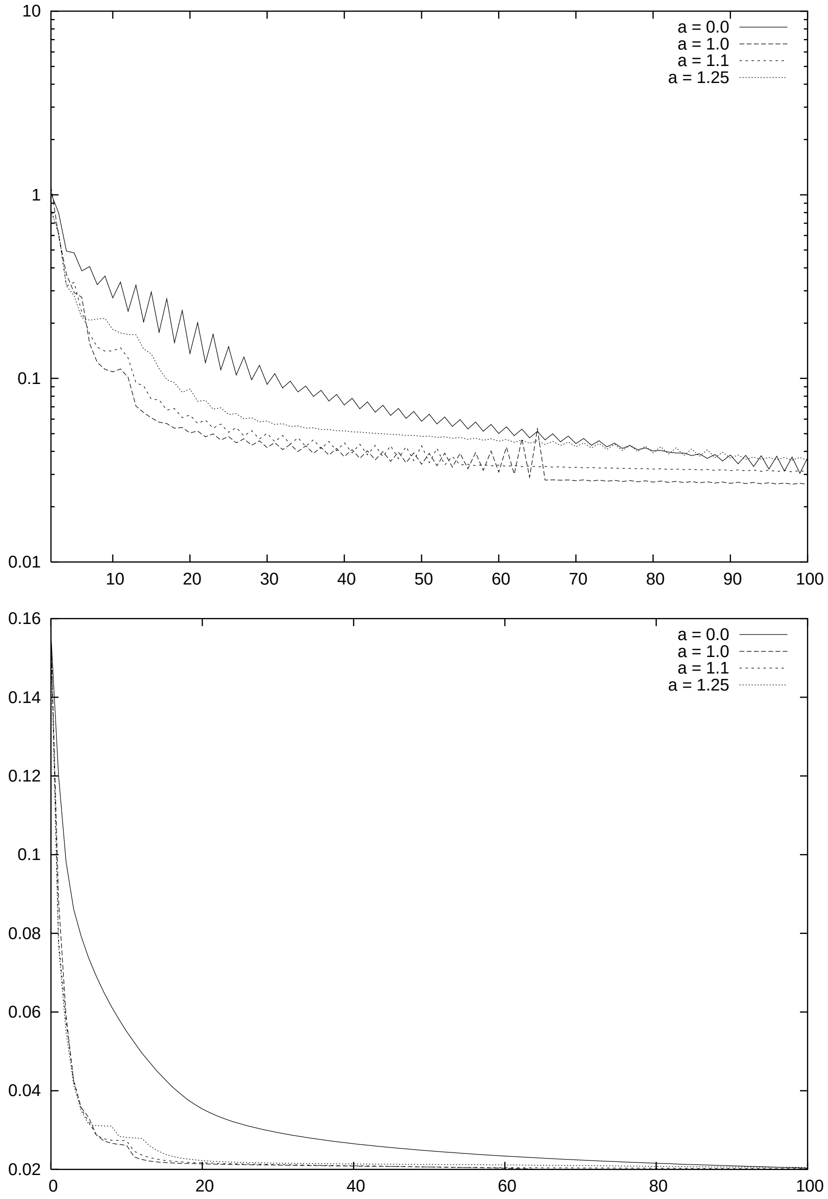

FIG. 3.9 – *Optimisation de la forme d'une aile ONERA M6 avec une méthode de gradient conjugué : Convergence du gradient (Haut), Fonction de coût (Bas), ns = 77315*

Troisième partie

Adaptation de maillages

1

Interpolation adaptative par optimisation fonctionnelle

Sommaire

Le contenu de ce chapitre a été soumis dans la revue *Applied Numerical Mathematics*.

1.1 Introduction

Depuis des dizaines d'années les chercheurs en Mécanique des Fluides Numérique construisent des schémas "précis à l'ordre deux" de plus en plus sophistiqués. Cependant, quand ces schémas sont appliqués à des cas industriels, la convergence n'est pas souvent à l'ordre deux. L'explication des théoriciens est que le maillage n'est pas encore assez fin, compte tenu du fait que les écoulement cherchés contiennent de très forts gradients localisés. Il peut même arriver que la convergence à l'ordre deux, qui est seulement vraie asymptotiquement, ne puisse être observée que pour des finesses de maillage inabordables en pratique.

En fait, pour répondre à l'attente de nos ingénieurs, une nouvelle théorie de l'approximation est en train de se construire, et elle prend en compte l'adaptation du maillage.

De nouveaux développements théoriques spécifient progressivement comment les maillages doivent être adaptés de façon que l'erreur d'approximation soit plus petite qu'un seuil spécifié par l'utilisateur.

La mise au point d'erreurs *a posteriori* est une direction d'investigation importante dans cet ordre d'idées, et nous référons aux travaux de [Babuska and Rheinboldt, 1978], [Becker *et al.*, 1999],[Fortin (Ed.), 2000].

L'ordre numérique de convergence vers la solution continue, c'est à dire exacte, est aussi très fréquemment à partir de la variation de l'erreur en fonction du nombre de nœuds (voir par exemple [Tie and Aubry, 1999]).
Il en ressort que le maillage devient une partie des inconnues du système à résoudre.

Dans le cas ou le maillage est recherché dans un ensemble de déformations d'un maillage de référence, de nombreux travaux dans la littérature proposent de prendre la déformation du maillage ou encore les coordonnées de celui-ci comme inconnues d'un système couplé avec la discrétisation de l'équation aux dérivées partielles. Voir par exemple [Cabello *et al.*, 1991],[Palmerio, 1996]. Dans ce cas, la topologie du maillage est par construction prescrite une fois pour tout par l'utilisateur et peut de pas être propice à une bonne adaptation.

Si au contraire l'utilisateur ne veut pas fixer la topologie, mais demande à l'algorithme de trouver la meilleure topologie, alors il se heurte à de nombreuses difficultés pour définir un système dont la solution serait cette topologie.

D'abord, deux maillages peuvent avoir des topologies très différentes toute en fournissant des solutions montrant les mêmes niveaux d'erreur aux mêmes endroits du domaine de calcul. Ensuite, il semble très difficile d'investiguer dans un ensemble abstrait d'entiers et de booleiens définissant les connectivités.

Ces remarques ont motivé les chercheurs à représenter les maillages par des fonctions continues. Il s'agira par exemple de fonctions définissant la taille de maille locale sur le domaine. Alors on peut en déduire une sorte de majorant de l'erreur de troncature locale. Mais ce majorant de donne pas une idée tout à fait précise de cette erreur car les étirements de maillage ne sont pas pris en compte. Dans plusieurs publications récentes, voir par exemple [Borouchaki and George, 1996a],[Borouchaki and George, 1996b], [Frey and George, 2000], and [Habashi *et al.*, 2000], l'étirement local est modélisé par la définition d'une métrique, champ non-scalaire défini d'après l'équation discrète.

Alors on peut appliquer une stratégie d'**équidistribution d'erreur** pour spécifier un nouveau maillage sur lequel le système d'EDP sera à nouveau résolu.

L'objet du présent travail est d'explorer les retombées d'une analyse dans laquelle tout serait **non-discrétisé**.

Dans la première phase présentée dans ce chapitre, nous nous concentrerons sur le problème plus facile de l'adaptation d'un maillage de manière à obtenir la meilleure **interpolation** $\mathcal{P}_1$ d'une fonction analytique donnée. Le problème de la meilleure métrique peut être mis sous la forme d'un problème d'optimisation abstrait et on en tire analytiquement la métrique optimale. Ce modèle mathématique permet d'investiguer l'ordre de convergence et d'intuiter celui du modèle discret correspondant.

Nous considérerons d'abord le cas 1D et nous rappellerons comment modèliser maillage et erreur d'interpolation. Le cas 1D permet notamment un tour d'horizon sur les diverses implications de ce type de modèle.
Puis nous proposons une extension 2D. L'erreur d'interpolation est définie à partir d'une estimation plus fine que celles disponibles jusqu'à présent. On obtient une métrique optimale, y compris dans le cas anisotropique.

L'établissement de ces modèles et de leurs propriétés est complété par une série d'expériences numériques de façon à montrer leur pertinence pratique. On rappelle enfin que le problème de la "meilleur interpolation" est aussi un problème de compression d'image.

1.2 Métrique continue dans un intervalle

Dans la cas de la dimension un, un maillage est spécifié de manière unique par une métrique. Après quelques définitions concernant la métrique, nous rappelons une estimation de l'erreur d'interpolation puis nous montrons qu'on peut en déduire une métrique optimale.

1.2.1 Définitions

Une métrique sur un espace euclidien permet de définir une distance entre deux éléments arbitraires de celui-ci. Nous appelons métrique sur un intervalle $[a, b]$ une fonction continue (strictement) positive $\mathcal{M} : x \to \mathcal{M}(x)$ définie sur $[a, b]$. Elle spécifie, pour tout c et d de cet intervalle la longueur du segment cd comme suit

$$L_{\mathcal{M}}(cd) = \int_c^d \sqrt{\mathcal{M}}(s) \quad ds \tag{1.1}$$

Considérons le maillage d'un intervalle $[a, b]$ avec N nœuds. C'est une subdivision $x_0 = a < x_1, ..., x_i < x_{i+1}, ..., x_{N-1} < x_N = b$ de cet intervalle. Une conséquence de la définition précédente est qu'une métrique peut spécifier une classe particulière de maillages. En effet, nous dirons qu'un maillage est conforme à une métrique $\mathcal{M}$ si est seulement si la relation suivante est vérifiée :

pour tout élément $[x_i, x_{i+1}]$, *nous avons* $\int_{x_i}^{x_{i+1}} \sqrt{\mathcal{M}} dx = 1$.

Si nous introduisons la taille de maille locale continue $m_{\mathcal{M}} = \mathcal{M}^{-1/2}$ nous avons :
pour tout élément $[x_i, x_{i+1}]$, $\int_{x_i}^{x_{i+1}} \frac{1}{m_{\mathcal{M}}}\, dx = 1$,
Une autre façon de comprendre ceci est d'introduire la densité de nœuds locale continue $d = 1/m$:

pour tout intervalle $[x_i, x_{i+1}]$, *nous avons* $\int_{x_i}^{x_{i+1}} d\; dx = 1$.

On peut vérifier que le nombre de nœuds (ou manière équivalente d'intervalles) du maillage est spécifié par la métrique. Il est donné par :

$$C(\mathcal{M}) = \int_a^b \sqrt{\mathcal{M}} dx = \int_a^b 1/m dx = \int_a^b d\; dx. \tag{1.2}$$

Si $C(\mathcal{M})$ est un nombre entier positif, un seul maillage est décrit par $\mathcal{M}$, si $C(\mathcal{M})$ n'est pas un entier, aucun maillage n'est décrit exactement par $\mathcal{M}$.

1.2.2 Erreur d'interpolation

Nous proposons maintenant de modéliser l'erreur numérique d'interpolation liée à une métrique. Puisque la métrique détermine la taille locale de maille, elle détermine aussi l'erreur d'interpolation résultant de l'application d'un interpolateur donné sur le maillage. Nous modélisons maintenant cette erreur par une fonction continue de la métrique.

Nous considérons :

- un maillage uniforme, i.e. : $x_0 < x_1 < ... < x_N$ avec $x_i = x_0 + \frac{i}{N-1}(x_N - x_0)$,
- une fonction u, suffisamment régulière, définie sur un segment $[a, b]$, $a = x_i, b = x_{i+1}$,
- h non nécessairement petit,
- $\Pi_h u$ l'interpolation $\mathcal{P}_1$ continue de u sur tout intervalle $[a, b]$ du maillage.
- supposant que $\Pi_h u(a) = u(a)$ et que $\Pi_h u(b) = u(b)$, l'erreur d'approximation est définie par :

$$e = u - \Pi_h u. \tag{1.3}$$

Pour tout x dans $[a, b]$, il existe t_1 dans $[0, 1]$ tel que :

$$\begin{aligned} e(a) &= (u - \Pi_h u)(a) \qquad (1.4) \\ &= (u - \Pi_h u)(x) + (a - x)(u - \Pi_h u)'(x) + \frac{(a-x)^2}{2} u''(x + t_1(a - x)). \end{aligned}$$

Nous notons $e = (u - \Pi_h u)$, pour x entre a et b, il existe un moyen de choisir t_2 dans $[0, 1]$:

$$\begin{aligned} e(b) &= (u - \Pi_h u)(b) \qquad (1.5) \\ &= (u - \Pi_h u)(x) + (b - x)(u - \Pi_h u)'(x) + \frac{(b-x)^2}{2} u''(x + t_2(b - x)). \end{aligned}$$

Trouver un majorant de $e = (u - \Pi_h u)$ amène à chercher un point x tel que :

$$e'(x) = (u - \Pi_h u)'(x) = 0. \tag{1.6}$$

Utilisant (1.5) et (1.6), lorsque $e(a) = e(b) = 0$, nous avons :

$$0 = (u - \Pi_h u)(x) + \frac{(a-x)^2}{2} u''(x + t_1(a-x)), \tag{1.7}$$

$$0 = (u - \Pi_h u)(x) + \frac{(b-x)^2}{2} u''(x + t_2(b-x)). \tag{1.8}$$

Ici, t_1 et t_2 dépendent de x. Par addition nous obtenons :

$$\begin{aligned} 0 &= 2(u - \Pi_h u)(x) + \frac{(a-x)^2}{2} u''(x + t_1(a-x)) \\ &+ \frac{(b-x)^2}{2} u''(x + t_2(b-x)). \end{aligned} \tag{1.9}$$

Donc :

$$\begin{aligned} 2(u - \Pi_h u)(x) &= -\frac{(a-x)^2}{2} u''(x + t_1(a-x)) \\ &- \frac{(b-x)^2}{2} u''(x + t_2(b-x)), \end{aligned} \tag{1.10}$$

et

$$|(u - \Pi_h u)(x)| \leq \frac{1}{2}(|\frac{(a-x)^2}{2}| + |\frac{(b-x)^2}{2}|)M, \tag{1.11}$$

où M est un majorant de $|u''|$ sur $[a, b]$. Alors :

$$|(u - \Pi_h u)(x)| \leq \frac{1}{2}(\frac{(a-x)^2}{2} + \frac{(b-x)^2}{2})M. \tag{1.12}$$

Prenant le maximum, nous avons :

$$|(u - \Pi_h u)(x)| \leq \frac{1}{2} \max_{\xi \in I}(\frac{(a-\xi)^2}{2} + \frac{(b-\xi)^2}{2}))M. \tag{1.13}$$

Le maximum est atteint en : $\xi_0 = \frac{(a+b)}{2}$, ce qui implique que, $\forall \xi \in I$:

$$|e(\xi)| = |(u - \Pi_h u)(\xi)| \leq \frac{(b-a)^2}{8} M. \tag{1.14}$$

Cette estimation, après avoir été modélisée en termes de fonctions continues, contribuera à l'énoncé du problème continu.

1.2.3 Métrique optimale

Condition d'optimalité pour la norme L^α

Nous pouvons maintenant *modéliser* l'erreur résultant de l'utilisation d'une interpolation $\mathcal{P}_1$ sur un maillage respectant une métrique donnée $\mathcal{M}$. L'analyse de la section précédente suggère (négligeant les termes d'ordre élevé) de prendre le modèle suivant :

$$|e_{\mathcal{M}}(x)| = (d_{\mathcal{M}}(x))^{-2}|u''(x)|. \tag{1.15}$$

où $d_{\mathcal{M}}(x)$ est la densité de nœuds du maillage, ou de manière équivalente l'inverse de la taille de maille locale, i.e. l'inverse de $m_{\mathcal{M}}(x)$. Nous voulons maintenant trouver le minimum par rapport à la métrique $\mathcal{M}$ de la norme L^α $(0 < \alpha < \infty)$ de l'erreur $e_{\mathcal{M}}$:

$$\min_{\mathcal{M}} \left(|e_{\mathcal{M}}(x)|\right)^\alpha_{L^\alpha} = \min_{\mathcal{M}\in K} \frac{1}{2}\int_a^b (d_{\mathcal{M}}(x)^{-2}|u''(x)|)^\alpha dx. \tag{1.16}$$

Pour éviter d'obtenir une solution triviale correspondant à un maillage infiniment fin, nous spécifions le nombre de nœuds comme suit :

$$C(\mathcal{M}) = \bar{C}(d) = \int_a^b d(x)dx \ = N. \tag{1.17}$$

ce qui donne une contrainte linéaire sur d.
Afin d'obtenir (au moins formellement) une condition d'optimalité, nous dérivons la fonctionnelle de (1.16) par rapport à d :

$$-2\alpha \int_a^b d^{-2\alpha-1}\left(|u''|\right)^\alpha \delta d \ ds \geq 0 \, , \, \forall \delta d : \int_a^b \delta d = 0. \tag{1.18}$$

Donc :

$$d_{opt}(x) = Cte.|u''(x)|^{\frac{\alpha}{2\alpha+1}} \tag{1.19}$$

et prenant en compte la contrainte $\bar{C}(d) = N$, nous obtenons :

$$d_{opt}(x) = \frac{N}{\int |u''|^{\frac{\alpha}{2\alpha+1}} ds} \ |u''(x)|^{\frac{\alpha}{2\alpha+1}}, \tag{1.20}$$

ou bien en termes de taille de maille locale :

$$m_{opt}(x) = \frac{\int |u''|^{\frac{\alpha}{2\alpha+1}} ds}{N} \ |u''(x)|^{\frac{-\alpha}{2\alpha+1}}. \tag{1.21}$$

Remarque 1.2.1
Afin d'identifier la métrique optimale solution de (1.21) comme un minimum du problème initial, plusieurs questions restent à clarifier.
D'abord, l'existence d'un tel minimum n'est pas sure. Il n'est probablement pas raisonnable de le chercher sans hypothèses de régularité supplémentaires. Il semble par exemple

utile de supposer que la métrique est bornée dans un espace de Sobolev. Ceci correspondrait à une hypothèse sur la régularité des maillages : aucun élément dont la mesure s'annule, pas de variation brusque de le taille ou de la forme des éléments.
Deuxièmement, dans le cas où nous faisons une hypothèse menant à une compacité suffisante de l'ensemble admissible, la question de l'unicité doit aussi être posée.
Cependant une analyse locale est possible. Le hessien de la fonctionnelle est positif, ce qui montre que l'optimum formel est au moins un optimum local dans un sous-ensemble de régularité suffisante de l'ensemble des contraintes. $\square$

Remarque 1.2.2
La taille de maille locale, naturellement proportionnelle à l'inverse du nombre de nœuds, est définie par (1.21) seulement si la dérivée seconde u'' ne s'annule jamais. En pratique, nous remplaçons $|u''|$ par $\max(\varepsilon, |u''|)$, avec un ε petit et strictement positif.$\square$

Le minimum de la fonctionnelle s'écrit :

$$(\mathcal{E}_\alpha^{opt})^\alpha \;=\; 1/2\left(\frac{\int |u''|^{\frac{\alpha}{2\alpha+1}} ds}{N}\right)^{2\alpha} \int_a^b |u''(s)|^{\frac{\alpha}{2\alpha+1}} ds. \tag{1.22}$$

Exemples :

- Dans le cas de la norme L^1 : cette norme est plutôt classique en traitement d'image,

$$m_{opt}(x) \;=\; \frac{\int |u''|^{1/3} ds}{N}\; |u''(x)|^{-1/3}. \tag{1.23}$$

- Dans le cas de la norme L^2 : c'est l'option naturelle pour les EDP,

$$m_{opt}(x) \;=\; \frac{\int |u''|^{2/5} ds}{N} |u''(x)|^{-2/5}\;. \tag{1.24}$$

- Dans le cas de la norme L^∞ : nous pouvons faire tendre l'exposant vers l'infini et obtenir *formellement* :

$$m_{opt}(x) \;=\; \frac{\int |u''|^{1/2} ds}{N} |u''(x)|^{-1/2}\;. \tag{1.25}$$

Remarque 1.2.3
Dans le dernier cas, introduisant d_{opt} :

$$d_{opt}(x) = \frac{N}{\int |u''|^{\frac{1}{2}} ds}\; |u''(x)|^{\frac{1}{2}}. \tag{1.26}$$

dans (1.15) donne une erreur uniforme,

$$|e_{\mathcal{M}}(x)| = (d(x))^{-2}|u''(x)| = \frac{(\int |u''|^{\frac{1}{2}} ds)^2}{N^2}, \tag{1.27}$$

ce qui n'est rien d'autre que la formule usuelle de répartition de l'erreur. Nous remarquons aussi que la taille de l'erreur optimale est déterminée par la norme $\mathcal{L}^{1/2}$ de la derivée seconde.

Si nous prenons la racine α-ème de l'expression (1.22), nous obtenons :

$$\mathcal{E}_\alpha^{opt} \;=\; 1/2\left(\frac{\int |u''|^{\frac{\alpha}{2\alpha+1}}ds}{N}\right)^2 \;(\int_0^1 |u''(s)|^{\frac{\alpha}{2\alpha+1}}ds)^{1/\alpha}, \tag{1.28}$$

puis en passant à la limite :

$$\mathcal{E}_\infty^{opt} \;=\; \frac{1}{N^2}\left(\int |u''|^{\frac{1}{2}}ds\right)^2. \tag{1.29}$$

Remarque 1.2.4
La puissance $1/2$ de la dérivée seconde $|u''|$ est reliée à l'ordre de précision de l'interpolation. La même analyse peut être développée avec une interpolation d'ordre plus élevée, ce qui est typiquement le cas avec un modèle d'erreur de puissance κ-ème.

$$|e_{\mathcal{M}}(x)| \;=\; (d_{\mathcal{M}}(x))^\kappa |u^{(\kappa)}| \;; \tag{1.30}$$

dans ce cas la puissance optimale est $\frac{\alpha}{\kappa\alpha+1}$ et elle est toujours plus petite que $1/\kappa$.

1.2.4 Ordre de convergence du modèle de métrique continue

Examinons comment chercher une métrique optimale dans le *cas d'une fonction u discontinue.* Une fois que l'interpolation $\mathcal{P}_1$ est choisie, nous modélisons l'erreur par :

$$\int_0^1 |e_{\mathcal{M}}(x)|^\alpha ds = \int_0^1 (m^2|\delta^{-2}(u(x+\delta) - 2u(x) + u(x-\delta))|)^\alpha ds. \tag{1.31}$$

où δ est plus petit que m. Nous observons que le quotient différentiel :

$\delta^{-2}(u(x+\delta) - 2u(x) + u(x-\delta))$:

- est proche de $\frac{\partial^2 u}{\partial x^2}$ où u est régulier,

- ou de l'ordre de δ^{-2} au voisinage des singularités de u.

De plus, puisque u est borné,

$$||\delta^{-2}(u(x+\delta) - 2u(x) + u(x-\delta))||_{L^{1/2}} \;\text{ est borné indépendamment de }\; \delta. \tag{1.32}$$

Le calcul des variations donne maintenant :

$$m_N(x) = Cte.|(|\delta^{-2}(u(x+\delta) - 2u(x) + u(x-\delta))|(x))|^{\frac{-2}{5}}, \tag{1.33}$$

et l'erreur optimale résultante dans L^2 s'écrit :

$$\text{Erreur} = \tfrac{2}{N^2}\ \left(\int |\delta^{-2}(u(x+\delta)-2u(x)+u(x-\delta))|^{\frac{2}{5}}\right)^{\frac{5}{2}} < \tfrac{K}{N^2}$$

où K est une constante bornée, compte tenu de à (1.32). Nous en déduisons que la stratégie d'adaptation est formellement précise à l'ordre 2.

1.3 Le cas 2D

Nous proposons un modèle étendu pour l'erreur d'interpolation puis nous appliquons de nouveau un calcul des variations.

1.3.1 Définition

Principales notations

Soit u une fonction deux fois continûement différentiable d'un ensemble Ω de $\mathbb{R}^2$ dans $\mathbb{R}$.
À chaque triangulation $\mathcal{T}_h$ de Ω correspond une interpolation $\mathcal{P}^1$ de u que nous notons par $\Pi_h u$.

À partir de l'analyse locale d'erreur :
- Nous considérons $K=[a,b,c]$, un triangle de diamètre h_{max} qui ne tend pas vers zéro.
- Nous considérons u, une fonction de $\mathbb{R}^2$ dans $\mathbb{R}$ qui est $\mathcal{C}^2$,
- Nous notons par $\Pi_h u$, l'interpolé linéaire de u sur K,
- Nous supposons que u et $\Pi_h u$ coincident en a, b et c.

Nous voulons trouver un bon majorant de l'erreur $e = u - \Pi_h u$ sur $K=[a,b,c]$.

Matrice hessienne $\mathcal{H}$

Le hessien de u est noté par :

$$\mathcal{H} = \begin{pmatrix} \frac{\partial^2 u}{\partial x^2} & \frac{\partial^2 u}{\partial x.\partial y} \\ \frac{\partial^2 u}{\partial x.\partial y} & \frac{\partial^2 u}{\partial y^2} \end{pmatrix} \tag{1.34}$$

$\mathcal{H}$ est diagonalisable et peut être écrit :

$$\mathcal{H} = \mathcal{R} * \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} * \mathcal{R}^{-1} \tag{1.35}$$

où nous introduisons la matrice de rotation $\mathcal{R}$ qui permet de passer du système usuel de coordonnées (x,y) au système (ξ,η) :

$$\mathcal{H} \equiv \mathcal{R}\widehat{\mathcal{H}}\mathcal{R}^{-1}. \tag{1.36}$$

Enfin :

$$\lambda_1 = \frac{\partial^2 u}{\partial \xi^2} \tag{1.37}$$

et

$$\lambda_2 = \frac{\partial^2 u}{\partial \eta^2}. \tag{1.38}$$

1.3.2 Métrique $\mathcal{M}$

La famille de métriques que nous considérons contient un tenseur dépendant de (x, y) et défini comme suit à partir des deux rotations $\mathcal{S}$ et $\mathcal{S}^{-1}$:

$$\mathcal{M}(x,y) = \mathcal{S}^{-1} \begin{pmatrix} \frac{1}{m_\xi}^2 & 0 \\ 0 & \frac{1}{m_\eta}^2 \end{pmatrix} \mathcal{S}, \tag{1.39}$$

où $\mathcal{S}, m_\xi$, et m_η dépendent de x et de y. Les coefficients m_ξ, et m_η sont les tailles locales de maille dans chacune des deux directions ξ et η définies par la rotation $\mathcal{R}$. Similairement au cas 1D, la longueur $L_{\vec{u}}$ du vecteur $\vec{u}$ dans la métrique $\mathcal{M}$ est définie comme suit :

$$L_{\vec{u}} = \int_0^1 \sqrt{\vec{u}.\mathcal{M}.\vec{u}} \;\; ds. \tag{1.40}$$

Les quantités $\frac{1}{m_\xi}$ et $\frac{1}{m_\eta}$ représentent le nombre d'éléments du maillage par unité de longueur respectivement suivant les axes ξ et η.

1.3.3 Complexité $C(\mathcal{M})$

D'un façon similaire au cas 1D, nous associons à la métrique $\mathcal{M}$, la complexité, ou nombre total de nœuds, calculée à partir de l'intégrale de la densité du maillage, i.e. de ε, m_ξ et m_η, sur une aire de ε^2 unité d'aire.

$$C(\mathcal{M}) = \int_\Omega \mathcal{M}^{\frac{1}{2}} ds = \int_\Omega \frac{1}{m_\xi} \frac{1}{m_\eta} \; dxdy. \tag{1.41}$$

Soit $d(\xi, \eta)$ la densité locale de nœuds pour la métrique $\mathcal{M}$. Elle est égale à $\frac{1}{m_\xi} . \frac{1}{m_\eta}$.

<u>Lemme</u> 4
Si le maillage tensoriel de $N + 1$ nœuds dans chaque direction satisfait idéalement la métrique $\mathcal{M}$ alors :

$$\int_\Omega m_\xi^{-1}.m_\eta^{-1}\,(x,y)\,dxdy = \sum_i \int_{x_i}^{x_{i+1}} \int_{y_i}^{y_{i+1}} d\,(x,y)\,dxdy = \sum_{i=1}^{N} (1) = N. \tag{1.42}$$

1.3.4 Une majoration brutale

Écrivons $(u - \Pi_h u)$ au voisinage de a :

$$\begin{aligned} (u-\Pi_h u)(a) &= (u-\Pi_h u)(x) + < \vec{xa}, \nabla(u-\Pi_h u)(x) > \\ &+ \frac{1}{2} < \vec{ax}, H_u(x+t_1\vec{xa})\vec{ax} >, \end{aligned} \tag{1.43}$$

où t_1 est entre 0 et 1 et dépend de x et de a (nous notons par $< \vec{v}, H(.)\vec{v} >$ le produit scalaire lié associé à $H(.)$). Similairement, pour b et c, nous obtenons :

$$\begin{aligned} (u-\Pi_h u)(b) &= (u-\Pi_h u)(x) + < \vec{xb}, \nabla(u-\Pi_h u)(x) > \\ &+ \frac{1}{2} < \vec{bx}, H_u(x+t_2\vec{xb})\vec{bx} >, \end{aligned} \tag{1.44}$$

$$\begin{aligned} (u-\Pi_h u)(c) &= (u-\Pi_h u)(x) + < \vec{xc}, \nabla(u-\Pi_h u)(x) > \\ &+ \frac{1}{2} < \vec{cx}, H_u(x+t_3\vec{xc})\vec{cx} > . \end{aligned} \tag{1.45}$$

Afin d'obtenir un majorant de $e = (u - \Pi_h u)$, nous cherchons un point x où l'extremum est atteint. Si x est dans K alors :

$$\nabla(u-\Pi_h u)(x) = 0, \tag{1.46}$$

ou :

$$< v\vec{e}c, \nabla(u-\Pi_h u)(x) >= 0, \tag{1.47}$$

pour tout $v\vec{e}c$ dans R^2 ou dans K.

Utilisant les trois dévloppements précédents et remarquant que $e(a) = e(b) = e(c) = 0$, nous obtenons :

$$\begin{aligned} 0 &= (u-\Pi_h u)(x) + \frac{1}{2} < \vec{ax}, H_u(x+t_1\vec{xa})\vec{ax} >, \\ 0 &= (u-\Pi_h u)(x) + \frac{1}{2} < \vec{bx}, H_u(x+t_2\vec{xb})\vec{bx} >, \\ 0 &= (u-\Pi_h u)(x) + \frac{1}{2} < \vec{cx}, H_u(x+t_3\vec{xc})\vec{cx} > \end{aligned} \tag{1.48}$$

puis par addition :

$$\begin{aligned} 0 = \; & 3(u-\Pi_h u)(x) \\ & + \frac{1}{2} < \vec{ax}, H_u(x+t_1\vec{xa})\vec{ax} > \\ & + \frac{1}{2} < \vec{bx}, H_u(x+t_2\vec{xb})\vec{bx} > \\ & + \frac{1}{2} < \vec{cx}, H_u(x+t_3\vec{xc})\vec{cx} > . \end{aligned} \tag{1.49}$$

Soit M un nombre réel tel que :

$$M = \max_{x \in K} \left(\max_{v\vec{e}c \in R^2} \frac{|v\vec{e}c, H_u(x) v\vec{e}c|}{||v\vec{e}c||^2} \right), \tag{1.50}$$

Alors :

$$|(u - \Pi_h u)(x)| \leq \frac{1}{6} \left(||\vec{ax}||^2 + ||\vec{bx}||^2 + ||\vec{cx}||^2 \right) M. \tag{1.51}$$

Par définition,

$$x = \lambda_a a + \lambda_b b + \lambda_c c, \tag{1.52}$$

avec :

$$\lambda_a + \lambda_b + \lambda_c = 1. \tag{1.53}$$

Donc :

$$\begin{aligned} \vec{ax} &= \lambda_b \vec{ab} + \lambda_c \vec{ac}, \\ \vec{bx} &= \lambda_c \vec{bc} + \lambda_a \vec{ba}, \\ \vec{cx} &= \lambda_a \vec{ca} + \lambda_b \vec{cb}. \end{aligned} \tag{1.54}$$

Nous déduisons que :

$$\begin{aligned} ||\vec{ax}||^2 + ||\vec{bx}||^2 + ||\vec{cx}||^2 &\leq (\lambda_a^2 + \lambda_b^2)||\vec{ab}||^2 \\ &+ (\lambda_a^2 + \lambda_c^2)||\vec{ac}||^2 \\ &+ (\lambda_b^2 + \lambda_c^2)||\vec{bc}||^2 \\ &+ 2(\lambda_a \lambda_b)| < \vec{ca}, \vec{cb} > | \\ &+ 2(\lambda_a \lambda_c)| < \vec{ba}, \vec{bc} > | \\ &+ 2(\lambda_b \lambda_c)| < \vec{ab}, \vec{ac} > |. \end{aligned} \tag{1.55}$$

Si nous notons par L la longueur de la plus grande arête, alors :

$$||\vec{ax}||^2 + ||\vec{bx}||^2 + ||\vec{cx}||^2 \leq 2(\lambda_a^2 + \lambda_b^2 + \lambda_c^2 + \lambda_a \lambda_b + \lambda_a \lambda_c + \lambda_b \lambda_c) L^2. \tag{1.56}$$

On vérifie aisément que l'extremum est atteint pour :

$$\lambda_a = \lambda_b = \lambda_c = \frac{1}{3} \tag{1.57}$$

et donc la majoration s'écrit :

$$|(u - \Pi_h u)(x)| \leq \frac{2}{9} L^2 M. \tag{1.58}$$

Ce résultat suggère la forme de la majoration à considérer dans la cas d'une dimension arbitraire d :

$$|(u - \Pi_h u)(x)| \leq \frac{1}{2}\frac{1}{1+d}\left(\frac{d(d+1)}{(d+1)^2} + 2\frac{d(d-1)}{2}\frac{d+1}{d+1}^2\right) L^2 M \tag{1.59}$$
$$= \frac{1}{2}\left(\frac{d}{1+d}\right)^2 L^2 M.$$

Nous retournons au cas où l'extremum n'est pas atteint dans K. Alors il correspond à une arête, disons l'arête ab. Le gradient s'annule sur ab et il s'ensuit que :

$$\begin{aligned} 0 = 2(u - \Pi_h u)(x) \quad &+ \quad \frac{1}{2} < \vec{ax}, H_u(x + t_1\vec{xa})\vec{ax} > \\ &+ \quad \frac{1}{2} < \vec{bx}, H_u(x + t_2\vec{xb})\vec{bx} > . \end{aligned} \tag{1.60}$$

Soit M tel que :

$$M = \max_{x \in \vec{ab}} \left(\max_{v\vec{e}c \in ab} \frac{| < v\vec{e}c, H_u(x) v\vec{e}c > |}{||v\vec{e}c||^2} \right), \tag{1.61}$$

alors :

$$|(u - \Pi_h u)(x)| \leq \frac{1}{4}\left(||\vec{ax}||^2 + ||\vec{bx}||^2\right) M. \tag{1.62}$$

Puisque $x = \lambda_a a + \lambda_b b$, nous retrouvons la majoration établie en dimension un :

$$|(u - \Pi_h u)(x)| \leq \frac{1}{8}\left(||\vec{ab}||^2\right) M. \tag{1.63}$$

Puis :

$$|(u - \Pi_h u)(x)| \leq \frac{1}{8} L^2 M. \tag{1.64}$$

Ce résultat est meilleur que (1.58) mais ne fournit aucune information concernant la possible anisotropie de la fonction et par conséquent ne peut pas être utilisé afin de spécifier un étirement le maillage.

1.3.5 Une majoration anisotropique

Les majorations anisotropiques sont le sujet de nombreuses recherches actuelles, on peut citer par exemple [Formaggia and Perotto, 2002]. Nous présentons ici un résultat adapté à nos besoins. Supposons que le point x où le maximum est atteint est plus près de a que de b ou c. Nous supposons aussi que x est dans K (pas sur une arête). Nous notons par a' le point d'intersection entre ax et l'arête opposée à a dans K, i.e. bc. Nous développons l'expression de e en a :

$$\begin{aligned} e(a) &= (u - \Pi_h u)(a) \\ &= (u - \Pi_h u)(x) + < \vec{xa}, \nabla(u - \Pi_h u)(x) > \\ &+ \int_0^1 (1-t) < \vec{xa}, H_u(x + t\vec{xa})\vec{ax} > dt. \end{aligned} \tag{1.65}$$

Puisque x est plus proche de a, le nombre λ, tel que $\vec{ax} = \lambda\vec{aa'}$, est plus petit que $\frac{2}{3}$:

$$\begin{aligned}
|e(x)| &= |\int_0^1 (1-t)\lambda^2 < \vec{aa'}, H_u(a+t\vec{xa})\vec{aa'} > dt|, && (1.66)\\
|e(x)| &\leq \frac{4}{9}|\int_0^1 (1-t) < \vec{aa'}, H_u(a+t\vec{xa})\vec{aa'} > dt|, \\
|e(x)| &\leq \frac{4}{9}|\int_0^1 (1-t)dt| \max_{t\in[0,1]} | < \vec{aa'}, H_u(a+t\vec{xa})\vec{aa'} > |,
\end{aligned}$$

Alors :

$$|e(x)| \leq \frac{2}{9}| \max_{y\in\vec{aa'}} |\vec{aa'}, H_u(y)\vec{aa'} > |, \qquad (1.67)$$

ou encore :

$$|e(x)| \leq \frac{2}{9}| \max_{y\in K} |\vec{aa'}, H_u(y)\vec{aa'} > |. \qquad (1.68)$$

La formule de récurrence précédente réapparait alors avec la constante suivante :

$$\lambda^2|\int_0^1 (1-t)dt = \frac{1}{2}\left(\frac{d}{d+1}\right)^2. \qquad (1.69)$$

Le cas où x est situé sur une arête, disons l'arête ab, mène éventuellement à la même majoration :

$$|e(x)| \leq \frac{1}{8}\max_{y\in ab} | < \vec{ab}, H_u(y)\vec{ab} > |, \qquad (1.70)$$

ce qui peut formellement être écrit comme ci-dessus (avec $a' = b$ et remarquant que $\frac{1}{8} < \frac{2}{9}$).

1.3.6 Minimisation de l'erreur d'interpolation (I)

Nous considérons d'abord la métrique isotrope. La taille locale de maille est définie par un champ scalaire unique, $m(x,y)$ ou de manière équivalente par la densité de nœuds par unité d'aire $d(x,y) = 1/m^2(x,y)$. Le nombre total de nœuds est donné par :

$$C(\mathcal{M}) = \int_\Omega d(x,y)\ dxdy. \qquad (1.71)$$

Pour la modélisation de l'erreur, nous nous inspirons de l'estimation grossière précédente :

$$e_{\mathcal{M}}(x,y) = m^2(x,y)M(x,y) = d^{-1}(x,y)M(x,y)\ . \qquad (1.72)$$

où M, défini au-dessus est pris égal à la plus grande des valeurs propres du hessien de u. Nous minimisons la norme $\mathcal{L}^\alpha$ de cette erreur sous la contrainte que le nombre de nœuds soit égal à N :

$$\min_{\mathcal{M}} \frac{1}{2}\int_\Omega M^\alpha\ d^{-\alpha}\ dxdy \qquad (1.73)$$

sous la contrainte $C(\mathcal{M}) = N$.

Les conditions d'optimalité sont :

$$-\alpha \int_\Omega M^\alpha \ d^{-\alpha-1} \ \delta d \ dxdy \ \leq \ 0 \tag{1.74}$$

$$\text{pour tout } \delta d \text{ tel que} \int_\Omega \delta d \ dxdy = 0 \ ,$$

ou prenant en compte la contrainte :

$$d_{opt}(x) = \frac{N}{\int_\Omega M^{\frac{\alpha}{-\alpha-1}} dxdy} \ M(x,y)^{\frac{\alpha}{\alpha+1}}. \tag{1.75}$$

Remarque 1.3.1
De nouveau le cas $\alpha = +\infty$ donne $d = M$, qui est l'isorépartition de l'erreur.

En termes de taille de maille locale m cela donne :

$$m_{opt}(x) = \frac{(\int_\Omega M^{\frac{\alpha}{-\alpha-1}} ds)^{1/2}}{N} \ M(x,y)^{\frac{-\alpha}{2\alpha+2}}. \tag{1.76}$$

En particulier pour $\alpha = 2$, l'exposant dans (1.76) est 1/3.

1.3.7 Minimisation de l'erreur d'interpolation (II)

Considérant une famille de métriques anisotropiques, nous retournons aux notations générales des Sections 3.1, 3.2,et 3.3.

Problème d'optimisation

Nous supposons que la fonction u possède des dérivées secondes bornées $\frac{\partial^2 u}{\partial x^2}$, $\frac{\partial^2 u}{\partial x \partial y}$, $\frac{\partial^2 u}{\partial y^2}$. Nous admettons que la meilleure métrique doit être choisie parmi les métriques qui sont alignées avec le hessien de u. Nous considérons alors la métrique $\mathcal{M}$:

$$\mathcal{M}_{x,y} = \mathcal{R}_\mathcal{M}^{-1} \begin{pmatrix} (m_\xi)^{-2} & 0 \\ 0 & (m_\eta)^{-2} \end{pmatrix} \mathcal{R}_\mathcal{M} \tag{1.77}$$

telle que :

$$\mathcal{R}_\mathcal{M} \quad = \quad \mathcal{R}_u \ , \tag{1.78}$$

où $\mathcal{R}_u$ est la rotation qui diagonalise le hessien de u :

$$\mathcal{H}_u = \begin{pmatrix} \frac{\partial^2 u}{\partial x^2} & \frac{\partial^2 u}{\partial x . \partial y} \\ \frac{\partial^2 u}{\partial x . \partial y} & \frac{\partial^2 u}{\partial y^2} \end{pmatrix} = \mathcal{R} * \begin{pmatrix} \frac{\partial^2 u}{\partial \xi^2} & 0 \\ 0 & \frac{\partial^2 u}{\partial \eta^2} \end{pmatrix} * \mathcal{R}^{-1} \tag{1.79}$$

Comme précédemment, par simplicité, les dérivées secondes $\frac{\partial^2 u}{\partial \xi^2}$ et $\frac{\partial^2 u}{\partial \eta^2}$ sont supposées avoir des valeurs absolues strictement positives.

L'estimation d'erreur précédente amène à considérer le *modèle continu d'erreur anisotropique* suivant :

$$\mathcal{E}_\alpha = \int \left(|\frac{\partial^2 u}{\partial \xi^2}|.m_\xi^2 + |\frac{\partial^2 u}{\partial \eta^2}|.m_\eta^2 \right)^\alpha \; dxdy \; . \tag{1.80}$$

La métrique optimale minimise la fonctionnelle $\mathcal{E}_\alpha$ sous la contrainte $C(\mathcal{M}) = N$:

$$\min_{\mathcal{M}} \int \left(|\frac{\partial^2 u}{\partial \xi^2}|.m_\xi^2 + |\frac{\partial^2 u}{\partial \eta^2}|.m_\eta^2 \right)^\alpha \; dxdy \tag{1.81}$$

$$\text{sous la contrainte} \int {m_\xi}^{-1} {m_\eta}^{-1} \; dxdy = N.$$

Le système d'optimalité s'écrit :

$$\begin{aligned} \mathcal{E}'_\alpha(\mathcal{M})\delta\mathcal{M} &= 0, \\ \forall \delta\mathcal{M}, \quad C'(\mathcal{M}_{opt}).\delta\mathcal{M} &= 0. \end{aligned} \tag{1.82}$$

La seconde équation peut être utilisée pour écrire une relation entre $\mathcal{M}$ et $\mathcal{C}$:

$$\begin{aligned} C'(\mathcal{M}_{opt}).\delta\mathcal{M} &= 0, \\ &\Updownarrow \\ \int \frac{-1}{m_\xi}.\frac{\delta m_\eta}{m_\eta^2} + \frac{-1}{m_\eta}.\frac{\delta m_\xi}{m_\xi^2} &= 0, \\ &\Updownarrow \\ \int \frac{1}{m_\xi}.\delta m_\eta + \frac{1}{m_\eta}.\delta m_\xi &= 0. \end{aligned} \tag{1.83}$$

On peut écrire :

$$\begin{pmatrix} \delta m_\xi \\ \delta m_\eta \end{pmatrix} = \zeta \begin{pmatrix} -m_\xi \\ m_\eta \end{pmatrix}. \tag{1.84}$$

L'équation (1.82) est vérifiée pour tout couple $(\delta m_\xi, \delta m_\eta)$ tel que (1.84) soit vraie au moins pour une fonction scalaire ζ de (x, y).

Nous développons maintenant l'équation (1.82) :

$$\int \left(|\frac{\partial^2 u}{\partial \xi^2}|m_\xi^2 + |\frac{\partial^2 u}{\partial \eta^2}|.m_\eta^2 \right)^{\alpha-1} \left(|\frac{\partial^2 u}{\partial \xi^2}|m_\xi \delta m_\xi + |\frac{\partial^2 u}{\partial \eta^2}|m_\eta \delta m_\eta \right) dxdy = 0. \tag{1.85}$$

Suite à l'énoncé (1.84) nous pouvons remplacer δm_ξ et δm_η :

$$\int \left(\left| \, |\frac{\partial^2 u}{\partial \xi^2}|m_\xi^2 + |\frac{\partial^2 u}{\partial \eta^2}|.m_\eta^2 \right|^{\alpha-1} \right) \zeta \left(-|\frac{\partial^2 u}{\partial \xi^2}|m_\xi m_\xi + |\frac{\partial^2 u}{\partial \eta^2}|m_\eta m_\eta \right) dxdy = 0\,. \tag{1.86}$$

Puisque m_η, m_ξ et les dérivées secondes de u ne s'annulent pas, cette équation est égale à zéro pour toute fonction ζ si :

$$|\frac{\partial^2 u}{\partial \xi^2}|.m_\xi^2 = |\frac{\partial^2 u}{\partial \eta^2}|.m_\eta^2. \tag{1.87}$$

De cette expression nous pouvons déduire la rapport entre m_ξ et m_η

$$\frac{m_\xi}{m_\eta} = \sqrt{\frac{|\left(\frac{\partial^2 u}{\partial \eta^2}\right)|}{|\left(\frac{\partial^2 u}{\partial \xi^2}\right)|}} \tag{1.88}$$

Pour la suite, il sera plus simple d'exprimer la métrique $\mathcal{M}$ en termes de la densité de nœuds d, nombre de nœuds par unité d'aire, et du rapport local de forme μ :

$$\mathcal{M} = \frac{1}{d}\,\mathcal{R}_\mathcal{M}^{-1}\begin{pmatrix} \mu & 0 \\ 0 & \frac{1}{\mu} \end{pmatrix}\mathcal{R}_\mathcal{M}\ , \tag{1.89}$$

Plus précisément nous posons : $m_\xi = \sqrt{\frac{\mu}{d}}$ et $m_\eta = \sqrt{\frac{1}{\mu d}}$. La constrainte (1.83) devient :

$$\int \delta d = 0. \tag{1.90}$$

La condition (1.82) s'écrit maintenant :

$$\begin{gathered}\int \left(|\frac{\partial^2 u}{\partial \xi^2}|\frac{\mu}{d} + |\frac{\partial^2 u}{\partial \eta^2}|\frac{1}{\mu d}\right)^{\alpha-1}. \\ \left(|\frac{\partial^2 u}{\partial \xi^2}|(\frac{\delta\mu}{d} - \frac{\mu\delta d}{d^2}) - |\frac{\partial^2 u}{\partial \eta^2}|\frac{d\delta\mu + \mu\delta d}{\mu^2 d^2}\right) = 0, \\ \forall \delta d \quad \text{tel que} \int \delta d = 0 \,\text{et}\, \forall \delta\mu. \end{gathered} \tag{1.91}$$

Nous développons cette expression en fonction de $\delta\mu$ et de δd. Pour $\delta\mu$, nous obtenons :

$$\int (*)^{\alpha-1}.\left(|\frac{\partial^2 u}{\partial \xi^2}|\frac{1}{d} - |\frac{\partial^2 u}{\partial \eta^2}|\frac{1}{\mu^2 d}\right)\delta\mu \;=\; 0 \;\; \forall \delta\mu\ , \tag{1.92}$$

où $(*)$ représente $|\frac{\partial^2 u}{\partial \xi^2}|\frac{\mu}{d} + |\frac{\partial^2 u}{\partial \eta^2}|\frac{1}{\mu d}$ qui, par hypothèse, ne s'annule jamais. Nous en déduisons que :

$$|\frac{\partial^2 u}{\partial \xi^2}|\frac{1}{d} - |\frac{\partial^2 u}{\partial \eta^2}|\frac{1}{\mu^2 d} = 0. \tag{1.93}$$

A partir de cela, nous obtenons :

$$\mu = \left(\frac{|\frac{\partial^2 u}{\partial \eta^2}|}{|\frac{\partial^2 u}{\partial \xi^2}|}\right)^{1/2} \tag{1.94}$$

qui est (1.88).

Pour δd :

$$\int (*)^{\alpha-1} . \left(|\frac{\partial^2 u}{\partial \xi^2}| \frac{-\mu}{d^2} + |\frac{\partial^2 u}{\partial \eta^2}| \frac{-1}{\mu d^2} \right) \delta d \; = \; 0 \; . \tag{1.95}$$

Nous obtenons alors :

$$(*)^{\alpha-1} \frac{1}{d^2} \left(|\frac{\partial^2 u}{\partial \xi^2}| (\mu) + |\frac{\partial^2 u}{\partial \eta^2}| \frac{1}{\mu} \right) = Cte. \tag{1.96}$$

ou, en d'autres termes :

$$\frac{1}{d^{\alpha+1}} \left(|\frac{\partial^2 u}{\partial \xi^2}| (\mu) + |\frac{\partial^2 u}{\partial \eta^2}| \frac{1}{\mu} \right)^{\alpha} = Cte. \tag{1.97}$$

Nous remplaçons μ par sa valeur :

$$d^{\alpha+1} = Cte \; \left(|\frac{\partial^2 u}{\partial \xi^2}| . |\frac{\partial^2 u}{\partial \eta^2}| \right)^{\frac{\alpha}{2}} . \tag{1.98}$$

Nous obtenons donc :

$$d = C_\alpha \; \left(|\frac{\partial^2 u}{\partial \xi^2}| . |\frac{\partial^2 u}{\partial \eta^2}| \right)^{\frac{\alpha}{2\alpha+2}} . \tag{1.99}$$

où la constante C_α est donnée par :

$$C_\alpha \; = \; \left(\int \left(|\frac{\partial^2 u}{\partial \xi^2}| . |\frac{\partial^2 u}{\partial \eta^2}| \right)^{\frac{\alpha}{2\alpha+2}} \; dxdy \right)^{-1} N \; . \tag{1.100}$$

Finalement les tailles de locale maille élevées au carré sont données par :

$$m_\xi^2 = C_\alpha^{-1} |\frac{\partial^2 u}{\partial \xi^2}|^{\frac{-2\alpha-1}{2(\alpha+1)}} |\frac{\partial^2 u}{\partial \eta^2}|^{\frac{1}{2(\alpha+1)}} \quad ; \quad m_\eta^2 = C_\alpha^{-1} |\frac{\partial^2 u}{\partial \xi^2}|^{\frac{1}{2(\alpha+1)}} |\frac{\partial^2 u}{\partial \eta^2}|^{\frac{-2\alpha-1}{2(\alpha+1)}} \tag{1.101}$$

ce qui signifie que la métrique $\mathcal{M}_{opt}$ est définie par :

$$\mathcal{M}_{opt} = C_\alpha^{-1} \; \left(|\frac{\partial^2 u}{\partial \xi^2}| . |\frac{\partial^2 u}{\partial \eta^2}| \right)^{\frac{-\alpha}{2\alpha+2}} \; \mathcal{R}^{-1} \; \begin{pmatrix} \left(\frac{|\frac{\partial^2 u}{\partial \eta^2}|}{|\frac{\partial^2 u}{\partial \xi^2}|} \right)^{1/2} & 0 \\ 0 & \left(\frac{|\frac{\partial^2 u}{\partial \xi^2}|}{|\frac{\partial^2 u}{\partial \eta^2}|} \right)^{1/2} \end{pmatrix} \; \mathcal{R} \; . \tag{1.102}$$

Dans le cas de la norme $\mathcal{L}^2$, ceci devient :

$$\mathcal{M}_{opt,2} = C_2^{-1} \; \mathcal{R}^{-1} \; \begin{pmatrix} |\frac{\partial^2 u}{\partial \xi^2}|^{-5/6} |\frac{\partial^2 u}{\partial \eta^2}|^{1/6} & 0 \\ 0 & |\frac{\partial^2 u}{\partial \eta^2}|^{-5/6} |\frac{\partial^2 u}{\partial \xi^2}|^{1/6} \end{pmatrix} \; \mathcal{R} \; . \tag{1.103}$$

Le cas de la norme $\mathcal{L}^\infty$ peut *formellement* être déduite en passant à la limite :

$$\mathcal{M}_{opt,\infty} = C_\infty^{-1} \; \mathcal{R}^{-1} \; \begin{pmatrix} |\frac{\partial^2 u}{\partial \xi^2}|^{-1} & 0 \\ 0 & |\frac{\partial^2 u}{\partial \eta^2}|^{-1} \end{pmatrix} \; \mathcal{R} \; . \tag{1.104}$$

Nous obtenons de nouveau une condition d'isorépartition de l'intégrande de (1.82).

1.3.8 Ordre de précision

Il est possible d'étendre l'analyse d'erreur de la Section 2.4. Dans le cas d'une adaptation isotrope, la valeur obtenue à l'optimum est :

$$\mathcal{E} = \frac{C}{N} \int_{\Omega} M^{\frac{\alpha}{\alpha+1}} dxdy \tag{1.105}$$

La précision d'ordre deux sera alors obtenue si l'intégrale précédente est bornée. Dans le cas de la fonction de Heavyside en 2D qui est égale à un si x est positif, à zéro partout ailleurs, la plus grande valeur propre est nécessairement celle correspondant à la direction x. Similairement au cas 1D, la dérivée seconde est intégrable au mieux à la puissance un demi. Pour $\alpha = 2$, la puissance dans l'erreur optimale est $2/3$. La précision d'ordre deux n'est donc pas obtenue.

Dans le cas anisotrope, un calcul analogue montre que pour obtenir la précision d'ordre deux avec la fonction de Heavyside, il suffit que la dérivée seconde en x soit intégrable à la puissance $1/4$, ce qui est vrai puisque elle appartient à $L^{1/2}$.

On peut alors remarquer que le modèle proposé suggère que l'adaptation de maillage optimal isotrope ne produit pas une méthode précise à l'ordre deux dans L^2 alors que l'adaptation de maillage optimal anisotrope permet d'obtenir une telle précision. Ces prévisions sont en concordance avec les résultats de [Coudiere *et al.*, 2002],[Dervieux *et al.*, 2001].

1.4 Quelques expériences numériques

Plusieurs points dans notre théorie nécessitent des illustrations numériques pour convaincre le lecteur de leur pertinence et de leur impact.

D'abord, les conclusions de notre analyse sont valides seulement si nos modèles sont suffisamment proches du contexte discret. Il est crucial de vérifier que ceci n'est pas réservé au cas des maillages extrêmement fins qui ne peuvent être utilisés en pratique.

Deuxièment, les maillages idéaux qui sont décrits comme optimaux dans le cas d'une norme d'erreur L^∞ sont ceux généralement proposés dans la littérature pour les résolutions de problèmes elliptiques. Maintenant les estimations d'erreur usuelle pour les problèmes elliptiques reposent sur la théorie hilbertienne. Nous avons montré que dans le cas L^2, une famille différente de maillages optimaux a déjà été trouvée dans le cas de l'erreur d'interpolation. Il est alors important de valider notre assertion selon laquelle la seconde famille est en quelque sorte optimale, et d'étudier quelles différences qualitatives apparaissent lorsque nous remplaçons L^∞ par L^2.

Afin de faire ceci, nous devons passer au contexte discret. Pour nous, le problème discret est juste une discrétisation du système d'optimalité continu, ce qui exprime la métrique optimale comme une fonction du hessien de la fonction continue à interpoler.

Les étapes pour la construire sont les suivantes :

- construire le hessien, soit par différentiation exacte, ou, de manière préférable pour la suite, en utilisant un maillage d'arrière-plan, qui soit suffisamment fin,

- en déduire notre métrique (continue sur le maillage d'arrière-plan).

À partir de cela, nous obtenons une solution discrète (la métrique sur le maillage d'arrière-plan) du problème continu ("Trouver la métrique"). L'erreur de discrétisation est commise en calculant la métrique et est liée à la qualité du maillage d'arrière-plan.

Une fois la métrique discrète obtenue, construire le maillage adapté revient à un post-traitement. Bien sûr, l'intérêt principal de la démarche réside dans l'obtention du maillage adapté et dans l'interpolation sur celui-ci. Nous nous focalisons sur la vérification de ses propriétés optimales. Avant cela, nous décrivons comment le maillage adapté est construit à partir de la métrique.

1.4.1 Outil d'adaptation de maillage

Nous avons réalisé toutes les expériences présentées ici avec le logiciel BAMG [Castro-Diaz *et al.*, 1996]. Étant donné un maillage d'arrière-plan et une fonction analytique, BAMG évalue sur le maillage le hessien de la fonction par une formule de dérivation discrète.
Nous avons modifié BAMG afin que la métrique soit calculée à partir du hessien en utilisant la formule précédente. Une fois la métrique obtenue (sur les nœuds du maillage .....), elle est utilisée dans le regénérateur de maillages afin de reconstruire un nouveau maillage respectant la métrique. Le regénérateur de maillages repose sur une reconnection Delaunay dans un espace équipé de la métrique et sur l'addition de nœuds, de nouveau afin de respecter la métrique. Le nombre de nœuds est ajusté lorsque nécessaire par modification d'un coefficient multiplicatif de la métrique. De nombreuses expériences avec BAMG peuvent être trouvées dans [Leservoisier, 2001].

1.4.2 Une certaine optimalité

Le but de cette section est de présenter des expériences numériques montrant un certaine optimalité des solutions proposées.

Nous considérons l'interpolation sur le disque unité du plan de la fonction suivante :

$$f_\epsilon(x,y) \quad = \quad 10x^3 \;+\; y^3 \quad + \quad atan(\frac{\epsilon}{sin(5y) - 2x}) \tag{1.106}$$

Nous considérons une série de maillage, indexée par un nombre β positif, telle que tous les maillages aient tous environ 2100 nœuds et soient adaptés selon la formule suivante :

$$\mathcal{M}_{opt} = \left(|\frac{\partial^2 u}{\partial \xi^2}|.|\frac{\partial^2 u}{\partial \eta^2}| \right)^{\beta} \mathcal{R}^{-1} \begin{pmatrix} \left(\frac{|\frac{\partial^2 u}{\partial \xi^2}|}{|\frac{\partial^2 u}{\partial \eta^2}|} \right)^{1/2} & 0 \\ 0 & \left(\frac{|\frac{\partial^2 u}{\partial \eta^2}|}{|\frac{\partial^2 u}{\partial \xi^2}|} \right)^{1/2} \end{pmatrix} \mathcal{R} \; . \tag{1.107}$$

La famille de métriques contient la métrique usuelle d'isorépartition, pour $\beta = 1$ et la métrique optimale obtenue en théorie pour la norme L^2, pour $\beta = 5/6$ et $\epsilon = 0.0001$. Nous calculons alors l'erreur L^2. Les résultats sont décrits dans la Fig.1.1. La norme de l'erreur associée à $\beta = 5/6$ est la plus petite, et est environ trois fois plus petite que celles résultant de $\beta = 0.7$ ou de $\beta = 1$.

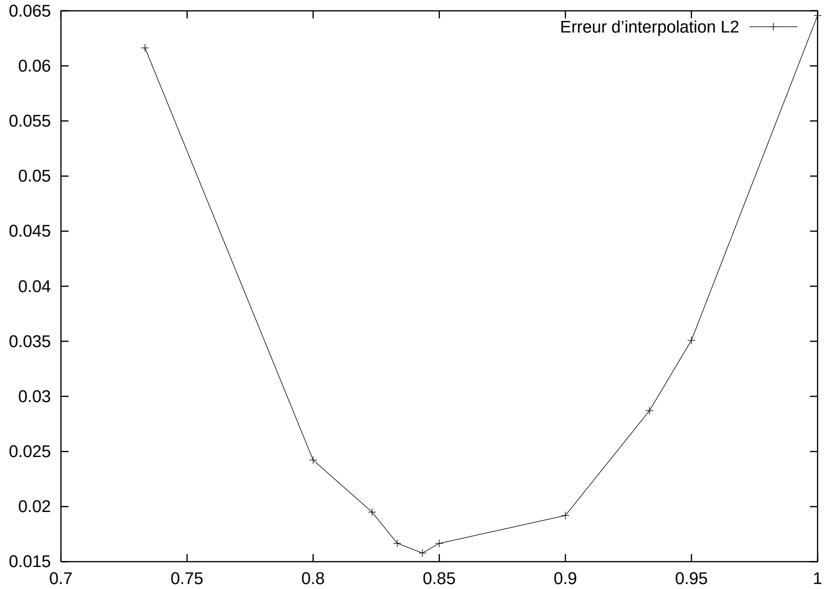

FIG. 1.1 – *Optimalité de la métrique proposée : les valeurs du paramètre β dans le critère d'adaptation sont en abscisse, l'erreur d'interpolation L^2 résultante est en ordonnée.*

1.4.3 Précision d'ordre deux

Dans [Palmerio and Dervieux, 1996], on affirme que pour une fonction régulière avec des *gradients raides*, le raffinement de maillage uniforme montre une convergence d'ordre deux seulement pour des maillages très fins, et que au contraire les méthodes d'adaptation de maillage montre une propriété de "capture précoce des détails", selon laquelle la convergence numérique d'ordre deux est observée avec un nombre de points beaucoup plus petit. Le but de cette section est de montrer des exemples pour lesquels la capture précoce des détails apparait.
Afin d'évaluer ce phénomène, nous avons considéré l'interpolation de trois fonctions, f_1, f_2, f_3, f_4, de type arctangent (1.106), avec quatre coefficients de "raideur" différents :

$\epsilon = 1.0,\ 0.1\ , 0.01\ , 0.001$. La convergence de maillage est d'abord mesurée avec des raffinements uniformes, Fig. 1.2 Lorsque la fonction n'est pas raide, la convergence d'ordre deux est obtenue facilement. A contrario, nous ne l'observons pas pour la fonction la plus raide, f_4, même avec des maillages d'environ 100.000 nœuds. Nous nous concentrons maintenant sur l'étude de f_4 et adaptons les maillages en appliquant la métrique proposée. L'effet est une convergence beaucoup plus rapide, essentiellement d'ordre deux, observable pour des maillages aussi grossier qu'une centaine de nœuds, Fig.1.3.

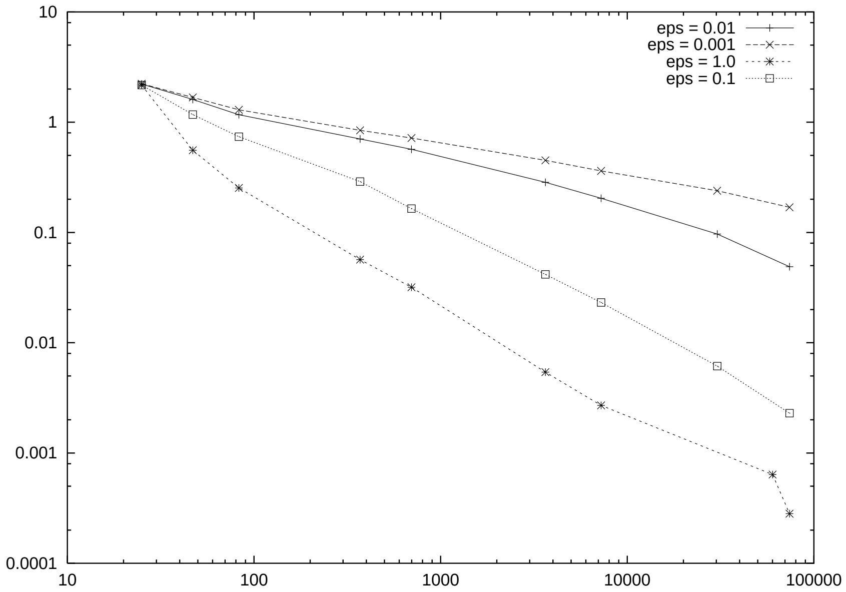

FIG. 1.2 – *Convergence de l'interpolée $\Pi_h f_\epsilon$ vers la fonction exacte f_ϵ pour $\beta = 0.33$ et des valeurs différentes de ϵ, les nombres de nœuds des maillages utilisés sont en abscisse, les valeurs de l'erreur d'interpolation L^2 résultante sont en ordonnée.*

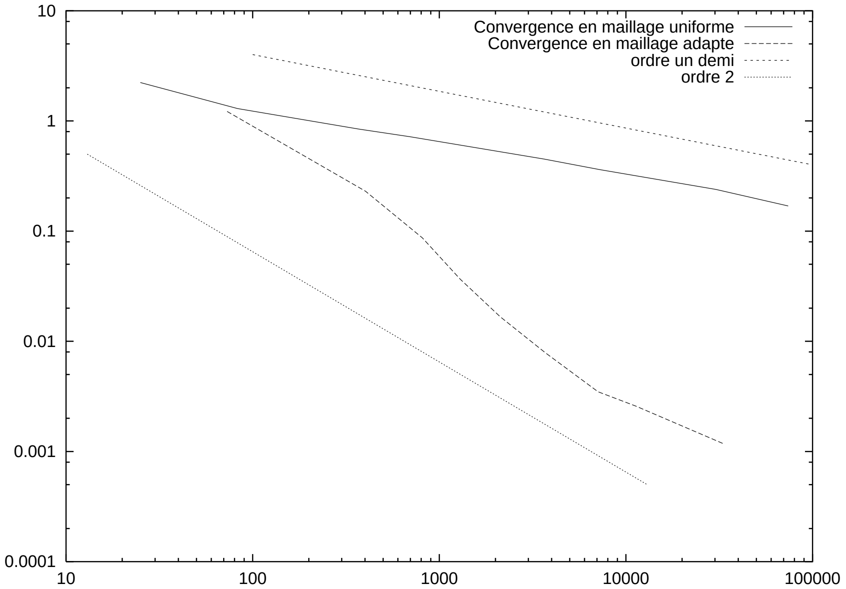

FIG. 1.3 – *Convergence de l'interpolée* $\Pi_h f_4$ *vers la fonction exacte* f_4 *pour* $\beta = 0.66$ *et* $\epsilon = 0.001$*, les nombres de nœuds des maillages utilisés sont en abscisse, les valeurs de l'erreur d'interpolation* L^2 *résultante sont en ordonnée.*

1.4.4 Influence du choix de la norme

Comme mentionné précdemment, l'analyse variationnelle proposée prend en compte les espaces fonctionnels L^α dans lesquels nous minimisons l'erreur d'interpolation.

Nous allons maintenant étudier l'influence de la norme fonctionnelle en relation avec l'application de la méthode à un problème de compression d'image.

En effet, étant donnée une fonction définie sur un maillage fin (uniforme ou non), une compression pourrait consister à la stocker sur un maillage plus petit, acceptant en quelques sortes une dégradation dans de la précision de sa définition. L'interpolation de maillages adaptative est une réponse à ce problème, déjà utilisée en traitement d'image [Marquant *et al.*, 2000].

Le premier exemple va illustrer la meilleur capacité de l'option L_p avec un petit p pour adapter le maillage aux détails de faible amplitude de la fonction. Nous commençons avec la somme d'une fonction arctangente d'amplitude 1., avec une fonction sinus d'une amplitude dix fois plus petite.

$$f(x,y) \quad = \quad 0.1 * sin(50x) \quad + \quad atan(\frac{0.001}{sin(5y) - 2x}) \tag{1.108}$$

Nous comparons les maillages adaptées avec environ 2000 nœuds chacun. Le premier est adapté en suivant le principe d'isorépartition, en d'autres termes, en minimisant l'erreur fonctionnelle L^∞. Le deuxième est adapté en minimisant la norme L^2 de l'erreur. Nous observons que l'option L^2 restitue la faible amplitude des oscillation du sinus alors que le L^∞ ne le montre pas du tout.

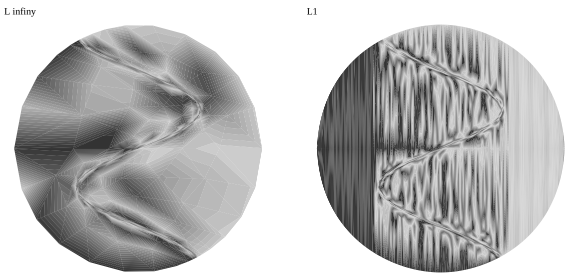

FIG. 1.4 – *Représentation d'une function avec les deux options, L^∞ (à gauche) et L^2 : contours.*

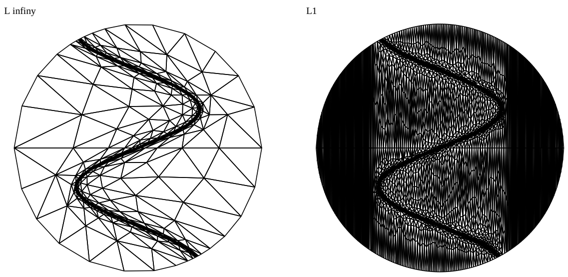

FIG. 1.5 – *Représentation d'une function avec les deux options, L^∞ (à gauche) et L^2 : maillages correspondants.*

La compression d'image basée sur des maillages est particulièrement utile pour stocker des images produites par des calculs numériques avec des discrétisation en éléments finis. Comme illustration nous considérons la compression des contours du Mach de l'analyse d'un écoulement. Les conditions de calcul de l'écoulement ne sont pas importantes pour notre sujet. Dans Fig.1.6, nous comparons les maillages adaptés L^∞ et L^2 de même nombre total de nœuds. Les chocs de type chocs de rampe partant du milieu de l'aile correspondent à une amplitude des variations beaucoup plus petite que le choc avant vertical de la partie gauche. Nous observons qu'ils sont presqu'ignorés par l'option L^∞ tandis qu'ils sont bien captés par l'option L^1.

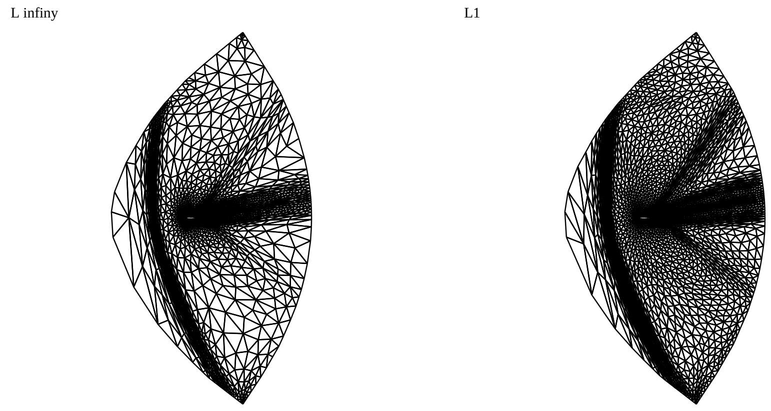

FIG. 1.6 – *Compression des contours du Mach basée un maillage : vue des maillages pour des compressions L^∞ et L^2.*

Un dernier exemple est la compression de maillage d'un portrait noir et blanc de Mona Lisa. L'image initiale utilisée est décrite en bas à droite de la Fig.1.7. Elle a été décrite par un maillage fin adapté de 60.000 nœuds résultant d'un traitement d'image présenté dans [Frey, 1993]. Le jeu consiste à le compresser jusqu'à 5000 nœuds seulement avec l'algorithme présenté. Nous avons vérifié que le taux de compression sur les fichiers postscript est en effet de 12.. Dans ce cas, l'identification de la meilleure approche n'est pas claire. Des contrastes sont importants pour la vision et certaines parties de l'image (les yeux) sont plus importantes que d'autres. Nous notons cependant que certaines régions avec un faible contrastes telles que la manche et la main, Fig.1.8, sont mieux reproduites avec l'option L^2.

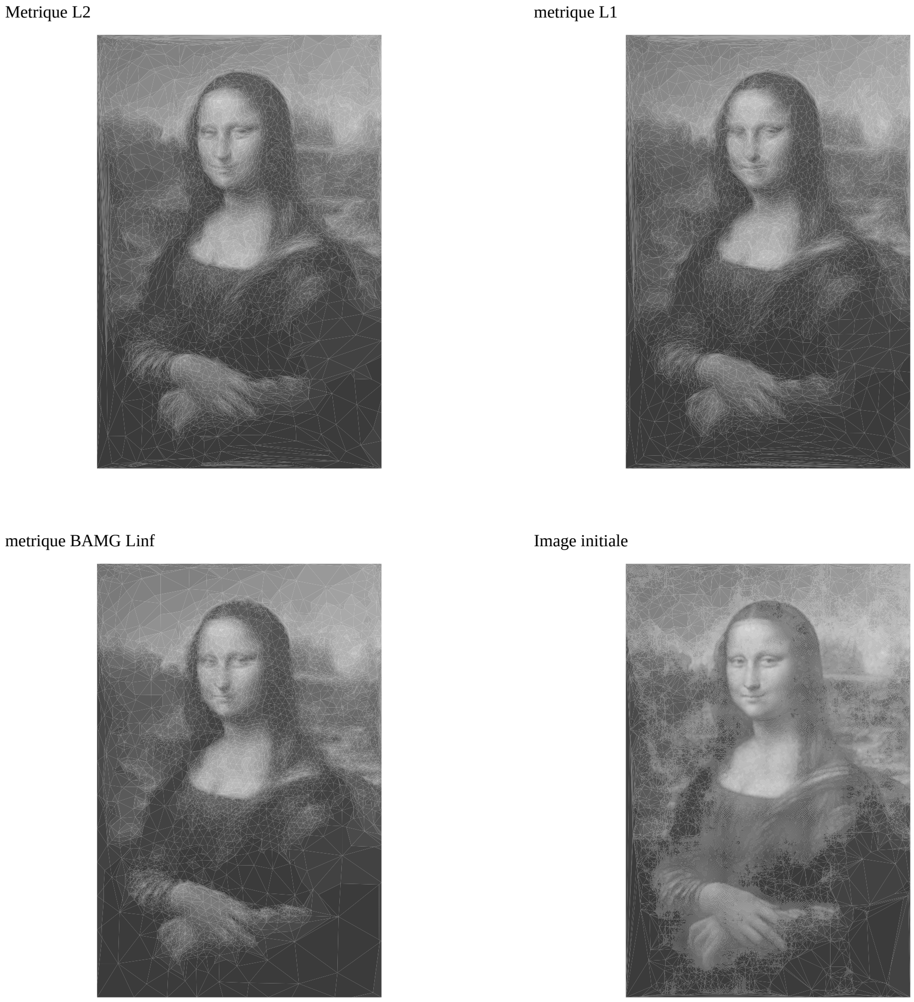

FIG. 1.7 – *Compression d'image basée sur un maillage, vues globales : image initiale (en bas à droite), adaptation L^2 (en haut, à gauche), adaptation L^1 (en haut, à droite), adaptation L^∞ (en bas à gauche).*

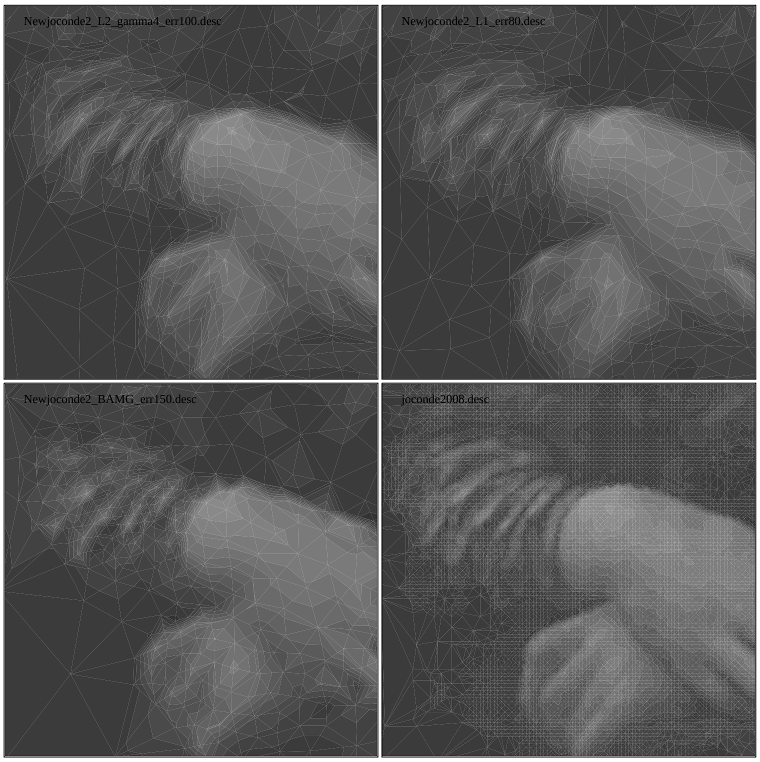

FIG. 1.8 – *Compression d'image basée sur un maillage, zoom : initial picture at bottom, image initiale (en bas à droite), adaptation L^2 (en haut, à gauche), adaptation L^1 (en haut, à droite), adaptation L^∞ (en bas à gauche).*

1.5 Conclusions

Ce travail explore certaines capacités de la théorie continue pour l'adaptation de maillages.
Dans cette première étude, nous nous sommes restreints au problème du meilleur maillage adapté pour de l'interpolation pure.

Le maillage est spécifié par une métrique, le nombre total de nœuds est une intégrale continue de la métrique, l'erreur est modélisée par le premier terme de la série de Taylor de l'erreur d'interpolation.

Une fois que ces modèles sont choisis, avec la norme de l'erreur que nous voulons être la plus petite, l'approche continue fournit, par un argument d'optimisation, une réponse au problème du meilleur maillage possible.

Nous obtenons une expression complètement explicite de la métrique optimale en fonction de la fonction à adapter.

Cette méthode est aussi un modèle pour l'évaluation de l'ordre de précision de la méthode d'adaptation et peut aider à spécifier les conditions d'obtention de l'ordre deux.

La transposition au contexte discret est montrée par quelques expériences numériques donnant une certaine (encore partielle bien sûr) confirmation que cette approche est efficace et montre les comportements prédits par la théorie.

Ce type d'analyse a une utilité potentielle pour plusieurs applications :

- *en compression d'image* : la méthode de métrique continue définit une méthode compression optimale sur un maillage isotropique.

- *en calcul scientifique* nous étendrons au Chapitre 2 de cette partie la stratégie proposée ici afin d'adapter la méthode à la recherche d'un maillage optimale pour la résolution d'une EDP.

1.6 Remerciements

Merci à Pascal Frey pour avoir fourni les données liées à Mona Lisa.

2

Contrôle optimal d'un maillage en Différences Finies

Sommaire

2.1 Introduction

Une difficulté importante de l'adaptation de maillage réside dans le choix des quantités qui vont spécifier la finesse du maillage adapté.

Un moyen très fréquent de simplifier ce problème consiste à choisir une fonction particulière de la variable dépendante, le **senseur**. Ses dérivées sont considérées comme un bon indicateur de l'erreur de troncature locale. Cette option est motivée par une analyse d'erreur *a priori*, qui produit une estimation dépendant de la solution exacte.

Cette approche ne prend pas en compte le fait que l'erreur d'approximation est actuellement une fonction non-locale de la finesse du maillage. Afin de prendre en compte ce problème, nous avons besoin d'identifier l'erreur avec la solution d'un système linéaire de l'erreur continu ou discret qui doit dépendre seulement de la qualité locale du maillage

Au contraire, dans l'analyse *a posteriori*, l'erreur est exprimée en fonction de l'effet de la solution discrète sur le résidu continu de l'équation.

Les deux analyses *a priori* et *a posteriori* montrent l'estimation de l'erreur d'approximation comme la solution d'une EDP linéarisée avec comme second membre un modèle de l'erreur locale de troncature. Une première approche naturelle a consisté à essayer de forcer l'équirépartition du second membre.

La stratégie que nous avons choisie consiste à rechercher le maillage qui minimisera une fonctionnelle scalaire dépendant de l'erreur d'approximation. Comme cette erreur d'approximation est solution d'une EDP dépendant du paramètre “maillage”, il en résultera une formulation de type Contrôle Optimal avec variable de Contrôle (le maillage), équation d'état (l'équation de l'erreur) et fonctionnelle dépendant à la fois de l'état et du contrôle.

Il reste à paramétrer le contrôle, c'est à dire le maillage. Une description précise du maillage, incluant à la fois les coordonnées des sommets et la connectivité complète est quelque chose que nous ne souhaitons pas dans l'espace de paramétrisation. En effet ce type d'option ne favoriserait pas la variation continue d'un maillage à un autre, et au contraire produirait inutilement de nombreux contextes de non-convexité pour la fonctionnelle. En particulier, deux maillages proches au point de produire à peu près la même précision de solution peuvent avoir des paramétrages très différents

Il est donc naturel de s'intéresser à une description continue du maillage. Nous voulons dire par là que comme dans le chapitre précédent, le maillage sera paramétrisé par un ensemble de fonctions continues définies sur le domaine de calcul. Il s'agit donc d'une extension de la “méthode de métrique continue” à l'adaptation de maillage pour les EDP.

Puisque le principe de base est de modéliser le problème du maillage optimal sous la forme d'un problème purement continu, notre modèle d'erreur sera inspiré par des estimations *a priori*.

Nous allons nous concentrer dans cette première approche sur le modèle elliptique standard. L'approximation choisie sera de type Eléments Finis usuelle continue linéaire par morceaux. Mais l'analyse sera simplifiée à une analyse de type Différences Finies. Nous nous intéresserons à l'existence d'un contrôle optimal et présenterons quelques résultats numériques.

2.2 Modèle en Différences Finies

2.2.1 Hypothèses simplificatrices

L'erreur de troncature locale dépend fortement de la régularité locale du maillage, et par exemple de la variation de la taille de maille dans une certaine direction, du nombre de voisins d'un nœud, de la régularité des éléments.
Dans le cas d'une estimation *a posteriori*, il est naturel de tenter d'avoir une information exacte concernant la contribution à l'erreur des différentes caractéristiques du maillage courant.
En revanche, dans notre démarche, nous essayons de spécifier le meilleur maillage possible pour un certain objectif. Il n'est pas abusif de supposer que le maillage recherché devra posséder certaines propriétés de régularité, dans la mesure où les quantités le décrivant, la métrique en l'occurence, seront aussi suffisamment régulières. Nous admettons dans ce chapitre que la métrique sera régulière dans la mesure où les données de l'EDP seront régulières. Le cas singulier pourrait être abordé en étendant les idées présentées dans le chapitre précédent.

2.2.2 Adaptation isotrope

Le modèle que nous appellerons de type Différence Finies va reposer sur les hypothèses de régularité de maillage suivantes :
- toute molécule d'approximation autour d'un nœud est proche d'une molécule symétrique et régulière,
- en particulier dans le cas isotrope, toute molécule est régulièrement répartie sur un cercle de centre le nœud considéré,
- dans le cas anisotrope, la molécule est régulièrement répartie sur une ellipse dont les axes principaux sont alignés avec ceux du hessiens de la solution exacte u de l'EDP.

La discrétisation par Eléments Finis de l'équation de Laplace peut être assimilée à des différences finies cartésiennes à cinq points. Transposons l'erreur de troncature résultante :

$$T_{et}(u,x,y) \ = \ h(x,y)^2 \ |u^{(4)}|(x,y)$$

où $|u^{(4)}|(x)$ désigne la plus grande valeur propre en module de la dérivée quatrième de u. $h(x,y)$ est la taille de maille locale dans n'importe quelle direction en vertu de l'hypothèse d'isotropie.
Notons :

$$T(u,x,y) \ = \ h^2 \ m(u) \ \ ; \ \ m(u) \quad \text{fonction de} \quad u \ .$$

La densité du maillage, c'est à dire le nombre local de nœuds par unité de surface en 2D est :

$$d(x,y) \ = \ h(x,y)^{-2} \ .$$

Le problème de la meilleure précision peut se mettre sous la forme du problème continu suivant :

$$\text{Trouver} \quad d \;=\; Argmin \;\; \|\mathcal{A}^{-1}\; d^{-1}\; m(u)\|_{L^2}^2$$

avec la contrainte d'un nombre de nœuds donné, qui s'écrit :

$$\int d \;=\; N\,.$$

2.2.3 Un paradoxe

Le problème de minimisation précédent peut s'écrire comme suit :

$$\begin{aligned} \text{Trouver} \quad d \;&=\; Argmin \;\; \|\mathcal{A}^{-1}\; d^{-1}\; m(u)\|_{L^2}^2 \\ \text{sous la contrainte} \int d \;&=\; N \end{aligned} \tag{2.1}$$

Soit $\bar{d} \;=\; d/N$. On en tire :

$$\begin{aligned} \text{Trouver} \quad \bar{d} \;&=\; Argmin \;\; N^{-2}\|\mathcal{A}^{-1}\; \bar{d}^{-1}\; m(u)\|_{L^2}^2 \\ \text{sous la contrainte} \int \bar{d} \;&=\; 1 \end{aligned} \tag{2.2}$$

Si $\bar{d}$ est l'optimum, les conditions d'optimalité du problème s'écrivent :

$$\begin{aligned} < \mathcal{A}^{-1}\bar{d}^{-1}\; m(u)\;,\; -\mathcal{A}^{-1}\frac{\delta\bar{d}}{\bar{d}^2}m(u) > \;&=\; 0 \quad \forall\delta\bar{d}, \int \delta\bar{d} \;=\; 0 \\ \int \frac{m(u)}{\bar{d}^2}\delta\bar{d}\;\; \mathcal{A}^{-*}\mathcal{A}^{-1}\bar{d}^{-1}\; m(u)\; dx \;&=\; 0 \quad \forall\delta\bar{d}, \int \delta\bar{d} \;=\; 0 \end{aligned}$$

Comme $\int \delta\bar{d} \;=\; 0$, nous tirons :

$$\frac{m(u)}{\bar{d}^2}\; \mathcal{A}^{-*}\mathcal{A}^{-1}\bar{d}^{-1}\; m(u) \;=\; C$$

où C est indépendant de N.
Puisque $\mathcal{A}^{-*}\phi$, pour tout ϕ, est nul sur la frontière de Ω, la seule valeur posible pour C est zéro. mais le facteur $\frac{m(u)}{\bar{d}^2}$ n'est jamais nul et nous obtenons :

$$\mathcal{A}^{-*}\mathcal{A}^{-1}\bar{d}^{-1}\; m(u) \;=\; 0$$

ce qui implique :

$$\bar{d}^{-1}\; m(u) \;=\; 0$$

mais ce nombre est lui aussi strictement positif et nous aboutissons à une contradiction. Il en résulte que le système d'optimalité n'a aucune solution régulière.

Remarque 2.2.1
Le point critique dans le calcul suivant est lié au fait que, indépendamment du maillage, la solution approchée est exacte sur la frontière du domaine.□.

En fait, nous n'avons pas de théorème d'existence d'un optimum faute par exemple d'une propriété de compacité dans notre formulation.
Cherchons donc à introduire un problème d'optimisation mieux posé.

2.2.4 Problème d'optimisation régularisé

Nous allons voir qu'une régularisation dans H^1 suffit si nous ajoutons un terme de pénalisation pour l'inverse de la densité d. Pour toute fonction strictement positive d de $H^1(\Omega)$, posons :

$$J(d) = J_1(d) \ + \ J_2(d) \ + \ J_3(d^{-1})$$

avec

$$\begin{aligned} J_1(d) &= |\mathcal{A}^{-1}\ d^{-1}\ m(u)|^2_{L^2} \\ J_2(d) &= \varepsilon |d|^2_{H^1} \\ J_3(d^{-1}) &= \eta |d^{-1}|^2_{L^2} \end{aligned}$$

à minimiser sous la constrainte :

$$\int d \ = \ N \ ,$$

Lemme :*Le problème précédent admet au moins une solution.*

Soit (d_n) une suite minimisante de J sous les contraintes précédentes :

$$J(d_n) \to Inf J$$

Puisque $J(d_n)$ est bornée, cette suite reste dans une boule (bornée) de H^1, la suite d_n^{-1} reste dans une boule de L^2, et nous pouvons extraire une sous-suite (encore notée (u_n)) qui minimise J et converge faiblement :

$$d_n \to d^* \ faiblement\ dans\ H^1$$

$$d_n \to d^* \ fortement\ dans\ L^2$$

$$d_n^{-1} \to k^* \ faiblement\ dans\ L^2 \ .$$

Pour toute fonction ϕ assez régulière,

$$(\phi d_n, d_n^{-1}) = \ (\phi(d_n \ - \ d^*), d_n^{-1}) \ + \ (\phi d^*, d_n^{-1})$$

Appliquant l'inégalité de Cauchy-Schwarz, on tire :

$$(\phi(d_n \ - \ d^*), d_n^{-1}) \to 0 \ .$$

Utilisant la convergence faible de d_n^{-1}, on obtient

$$(\phi d^*, d_n^{-1}) \to (\phi d^*, k^*) \ .$$

On déduit que

$$(\phi d_n, d_n^{-1}) \to (\phi d^*, k^*).$$

Mais :

$$(\phi d_n, d_n^{-1}) \ = \ \int \phi$$

et :

$$(\phi d^*, k^*) \ = \ \int \phi d^* k^* \ .$$

Nous concluons que $d^* k^* = 1$ $a.e.$ i.e. $k^* \ = \ (d^*)^{-1}$ $a.e.$.

Il reste à passer à la limite :

$$J_1(d^*) \leq lim \ Inf \, J_1(d_n)$$

$$J_2(d^*) \leq lim \ Inf \, J_2(d_n)$$

$$J_3((d^{-1})^*) \leq lim \ Inf \, J_3((d_n)^{-1})$$

où la "lim" est prise pour un n' tendant vers ∞, et le "Inf" pour tout n plus grand que n'. Nous rappellant que ces nombres sont positifs ou nuls, nous pouvons sommer ces inégalités :

$$J_1(d^*) \ + \ J_2(d^*) \ + \ J_3((d^{-1})^*) \ \leq \ lim \ Inf \ (J_1(d_n) + J_2(d_n) + J_3((d_n)^{-1}) \ ) \ = \ Inf \ J.$$

et d^* est bien un optimum.$\square$

2.2.5 Conditions d'optimalité

Les conditions d'optimalité du problème régularisé expriment que pour tout accroissement δd tel $\int \delta d \;=\; 0$, on a :

$$< \mathcal{A}^{-1}d^{-1}\; m(u)\;, -\mathcal{A}^{-1}\frac{\delta d}{d^2}m(u) > +\varepsilon((d,\delta d))_{H^1} - \eta < d^{-1}, d^{-2}\delta d >_{L^2} \;=\; 0$$

ou :

$$\int \frac{m(u)}{d^2}\delta d\;\; \mathcal{A}^{-*}\mathcal{A}^{-1}d^{-1}\; m(u) \;+\; \varepsilon((d,\delta d))_{H^1} \;-\; \eta < d^{-1}, d^{-2}\delta d >_{L^2} \;=\; 0\;.$$

Comme $\int \delta d \;=\; 0$, nous déduisons (formellement) que :

$$-\;\varepsilon\Delta d \;+\; \frac{m(u)}{d^2}\;\mathcal{A}^{-*}\mathcal{A}^{-1}d^{-1}\; m(u) \;-\; \eta\; d^{-3} \;\;=\;\; C$$

où C est indépendant de N, et

$$\frac{\partial d}{\partial n} \;\;=\;\; 0. \tag{2.3}$$

Nous observons que les deux nouveaux paramètres ε et η contribuent à résoudre le paradoxe précédent.

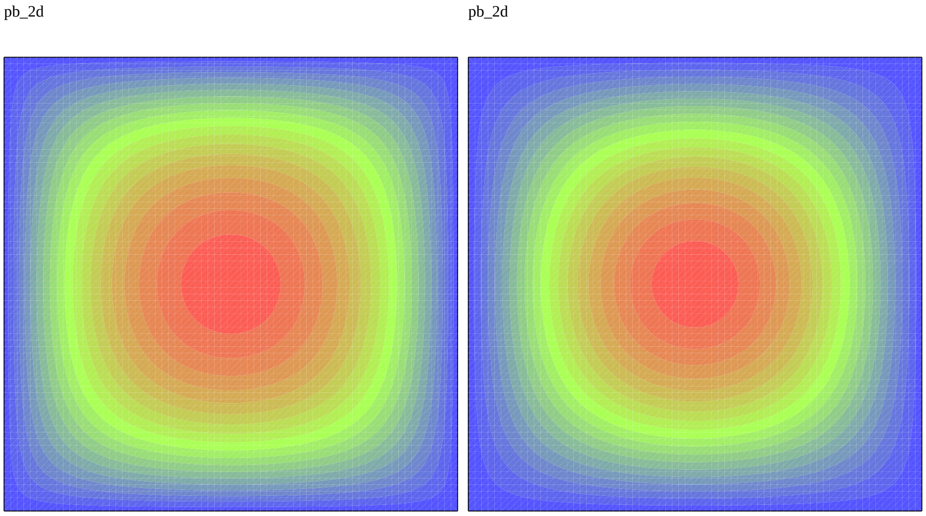

FIG. 2.1 – Comparaison entre erreur exacte et modèle de l'erreur pour la fonction u_1

2.3 Expériences numériques

Le modèle est appliqué au cas de la fonction u_1.

$$u(x,y) \;=\; u_1(x,y) \;=\; (x^2-x)(y^2-y)$$

Le modèle (3.16) représente fidèlement l'erreur comme l'illustre la comparaison des isovaleurs de l'erreur véritable et du modèle en figure 2.1. pour un maillage uniforme de 4900 nœuds. Ces erreurs sont nulles au bord et présentes un maximum au centre du domaine.

La fonctionnelle choisie est une norme L^2 du modèle d'erreur. Les termes de pénalisation et régularisation H^1 de la densité de points sont mis à des niveaux de l'ordre de l'unité. Nous avons lancé une minimisation de cette fonctionnelle par rapport à la densité de point, faisant donc d'emblée une hypothèse d'isotropie de la métrique continue. La condition initiale est une densité de points uniforme, qui rend aussi uniforme le modèle d'erreur de troncature. Cette densité initiale est donc déjà optimale vis à vis de ce critère local et rend les termes de pénalisation minimaux. Cependant la fonctionnelle diminue encore de 50%. L'effet de cette minimisation est observable à travers quelques coupes médianes du domaine de calcul :
- la densité de points croît au centre, diminue au bord (Fig.2.2),
- l'erreur de troncature évolue en sens inverse (Fig.2.3), diminue de 40% au centre et prend de grandes valeurs sur le bord,
- le modèle d'erreur d'approximation décroît partout (Fig.2.4).

Nous recommençons l'expérience en prenant pour fonctionnelle la norme L^2 du modèle d'erreur intégrée seulement sur la moitié droite du domaine de calcul. La réponse de l'approche locale serait d'avoir un maillage très grossier sur la partie gauche. Dans la configuration optimale obtenue, le maximum de la densité de points est très légèrement décalé, d'abscisse 0.6, et l'erreur légèrement décalée vers la gauche. On touche du doigt le caractère peu local de l'opérateur de Poisson inverse.

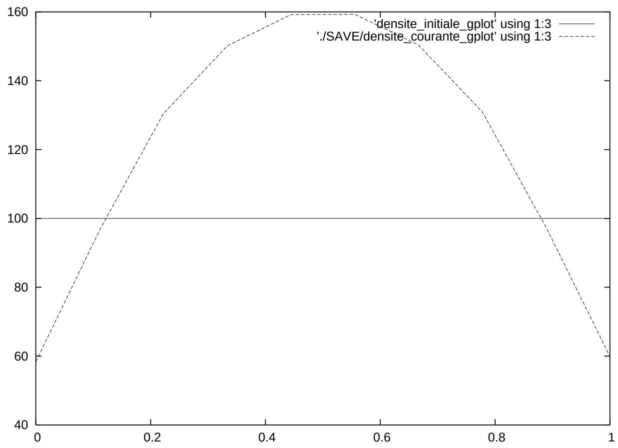

FIG. 2.2 – Densités de point initiales et finales

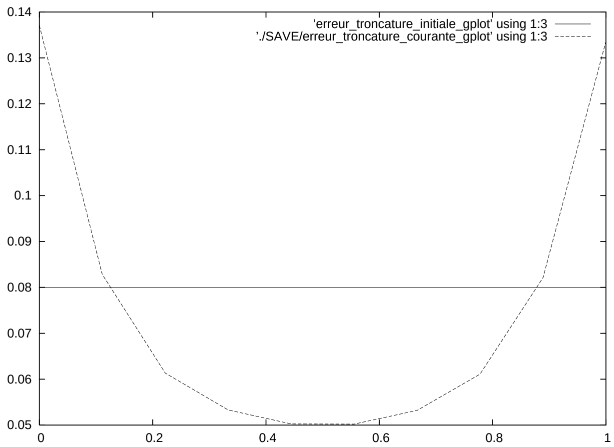

FIG. 2.3 – Erreur de troncature initiale et finale.

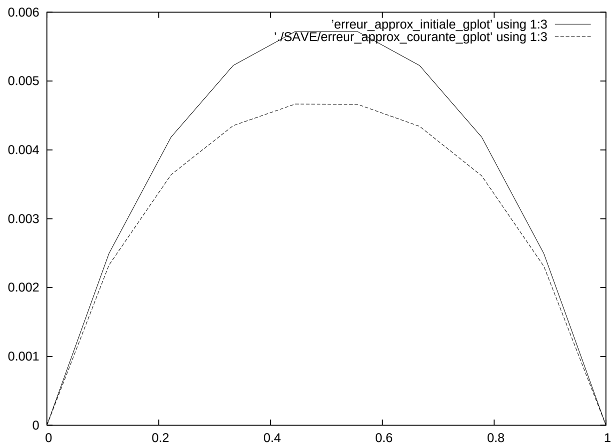

FIG. 2.4 – Erreurs d'approximation initiales et finales

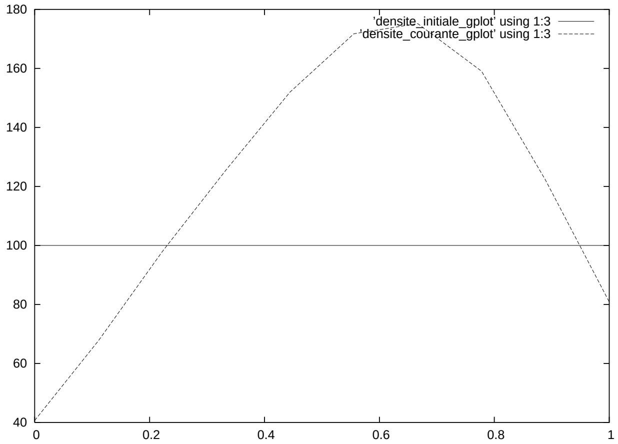

FIG. 2.5 – Observation à droite. densités de point initiales et finales

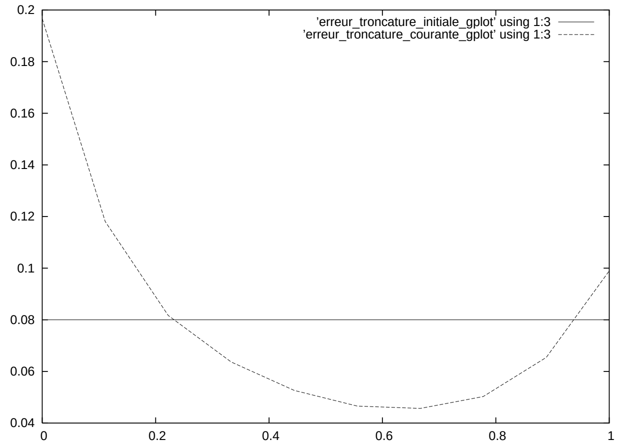

FIG. 2.6 – Observation à droite. Erreur de troncature

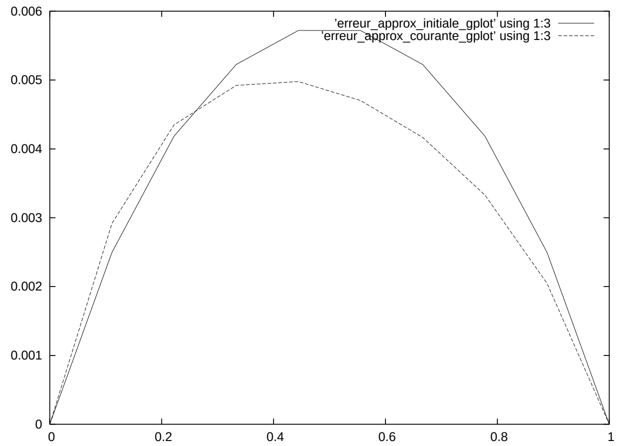

FIG. 2.7 – Observation à droite. modèle de l'erreur d'approximation

2.4 Conclusions

Le but de ce chapitre était de proposer une nouvelle approche pour l'adaptation du maillage d'une EDP. Cette approche est par certains aspects intermédiaire entre d'une part la voie simplifiée et assez empirique de l'adaptation à un senseur, et, d'autre part, les méthodes basées sur les estimations *a posteriori*.

Dans cette méthode, il y a une première phase d'analyse *a priori* rigoureuse, suivie d'une seconde phase plus empirique de modèlisation.

Il résulte de notre approche une formulation en contexte continu de la recherche du meilleur maillage, minimisant une certaine fonctionnelle.

Alors se pose la question de l'existence d'un tel minimum, que nous proposons de résoudre en exigeant un certain niveau de régularité pour les maillages recherchés.
L'intérêt de l'approche est illustrée par quelques calculs simples de comparaison modèle-approximation et plusieurs résultats d'optimisation de maillage.

3

Contrôle Optimal d'un maillage en Éléments Finis

Sommaire

3.1 Introduction

Le but de ce chapitre est de ré-examiner l'approche de métrique optimale pour une EDP dans un contexte Eléments Finis plus général.

En Eléments Finis, l'analyse d'erreur a l'avantage d'être réductible à l'erreur d'interpolation pour la solution exacte en vertu du lemme de Céa :

$$||u - u_h||_{H^1} \quad \leq \quad ||u - \Pi_h u||_{H^1} \ .$$

Dans ce cas, le terme d'erreur est indépendant du système à résoudre, et ne dépend pas non plus, jusqu'à une certaine limite, de l'approximation choisie pour ce système (voir par exemple [Castro-Diaz *et al.*, 1996]). Il suffirait donc, d'après cette remarque, d'appliquer l'algorithme d'interpolation adaptative en l'adaptant à la norme H^1.
Cependant cette estimation purement locale néglige complètement l'influence d'une erreur de troncature ponctuelle sur tout le domaine de calcul, telle que nous l'avons mise en évidence dans le chapitre précédent.

Cette utilisation directe de l'analyse *a priori* nous semble donc insuffisante.

La théorie de l'analyse d'erreur en Éléments Finis est continuellement enrichie dans une direction duale, l'analyse d'erreur *a posteriori*, initiée par Babuska et Rheinbolt ([Babuska and Rheinboldt, 1978], voir [Verfurth, 1994] pour un état de l'art dans ce domaine). Une propriété intéressante de cette analyse est la prise en compte de la qualité du maillage courant.
Dans les travaux les plus récents, apparaissent des spécifications plus précises du but de l'adaptation. Il s'agit typiquement de minimiser une norme de l'erreur ou de calculer précisément une fonctionnelle à valeur réelle.
Sous l'impulsion de Giles ([Giles and Suli, 2002]), plusieurs travaux ont permis la mise au point de méthodes visant à une meilleure évaluation d'un résultat scalaire déduit de la solution d'une EDP. Par exemple on réalise un calcul d'écoulement afin de donner une bonne prédiction du coefficient scalaire de portance. Le résultat est considéré comme une **fonctionnelle** d'après la théorie standard du Contrôle Optimal. Une analyse de l'erreur d'approximation de cette fonctionnelle est alors plus facile grâce à l'introduction d'un **état adjoint**. Dans [Giles, 2001] une méthode est proposée afin d'évaluer cette erreur (scalaire) et d'utiliser un correcteur pour obtenir un résultat "superconvergent".
Se basant aussi sur une fonctionnelle, Becker et Rannacher ([Becker and Rannacher, 1996]) utilisent la formulation de l'erreur en termes d'état et d'adjoint afin de construire des algorithmes tendant à équidistribuer l'intégrande de cette erreur. Les deux approches précédentes sont en quelque sorte complémentaires et sont combinées dans [Venditi and Darmofal, 2003]. Par ailleurs le traitement des maillages anisotropes par des estimations *a posteriori* reste un problème essentiellement ouvert malgré quelques avancées récentes [Formaggia and Perotto, 2002; Kunert, 2000].

Il est maintenant intéressant d'introduire quelque contraste entre, d'un côté, la recherche d'une bonne approximation d'une fonctionnelle particulière (dépendant de la solution d'une EDP), et d'un autre côté, la minimisation d'une norme particulière de l'erreur d'approximation. Les deux sont nécessaires dans les études numériques. Ce chapitre repose, comme le précédent, sur la seconde problématique. Un de ses avantages est que nous pouvons pousser plus loin les idées des précédents auteurs dans la direction d'une analogie avec le contrôle optimal, et essayer de trouver le maillage qui minimise la fonctionnelle d'erreur.

À la différence du chapitre précédent, ce chapitre utilise une analyse d'erreur de type Éléments Finis. Il explore plusieurs modèles continus, permettant notamment d'aborder une stratégie d'adaptation anisotrope.

3.2 Estimation d'erreur pour la FEM

Soit Ω un sous-domaine polyhédrique borné du plan ou de l'espace 3D.

Nous considérons le problème usuel de Poisson sur le domaine Ω :

$$-\Delta u \;=\; f \;\; sur \;\; \Omega \;\; ; \;\; u = 0 \;\; sur \;\; \partial\Omega \;, \tag{3.1}$$

dont la forme variationnelle s'écrit

$$a(u,v) \;=\; \int \nabla u.\nabla v \; dx \;=\; < \; f \; , \; v > \qquad \forall \; v \in V \;, \tag{3.2}$$

où V est $H^1_0(\Omega)$. L'opérateur correspondant $A = -\Delta$ va de V dans son dual topologique V'.

Pour tout h dans un ensemble $\mathcal{H}$ de nombres petits positifs ayant zéro comme point d'accumulation, nous supposons avoir un maillage τ_h, fait de tétraèdres, du domaine Ω.

$$\forall h \in \mathcal{H}, \;\; \Omega \;=\; \cup_{\bar{T}_h \;\in\; \tau_h} \bar{T}_h$$

L'indice h de τ_h est aussi supposé être le plus grand diamètre des élément de τ_h.

Soit V_h le sous-espace de V des fonctions continues qui sont $\mathcal{P}_1$ sur chaque élément du maillage τ_h.

$$V_h \;=\; \{\phi \in H^1_0(\Omega), \; \phi \text{ continue}, \; \phi|_K \text{ est } P_1, \text{ pour tout} K \in \tau_h\}.$$

Le problème variationnel discret dans V_h s'écrit

$$a(u_h, v_h) \;=\; (\; f \; , \; v_h) \qquad \forall \; v_h \in V_h \tag{3.3}$$

Les estimations d'erreur *a posteriori* ont été déduites de cette approximation par beaucoup d'auteurs (voir e.g. [Verfurth, 1994]). Ces analyses proposent des formes différentes du terme principal de l'erreur. Les estimations d'erreur *a priori* ont été déduites beaucoup plus tôt dans $H^1(\Omega)$ ("propriété de projection"), et dans $L^2(\Omega)$ (analyse d'Aubin-Nitsche).

Afin de pousser plus loin l'analyse *a priori* dans $H^1(\Omega)$, nous devons introduire une famille de projecteurs de Clément [Clément, 1975]. Ces projecteurs s'utilisent sur l'espace H^1 et possèdent des propriétés d'approximation proches de celles de l'interpolation usuelle. L'opérateur de Clément standard repose sur des valeurs aux nœuds estimées sur le support des fonctions de base. Nous pouvons restreindre le domaine de cette moyenne en prenant une petite boule, de rayon spécifié ε, autour du sommet tout en ayant un opérateur $\bar{\Pi}^\varepsilon_h$ sur H^1 :

$$\bar{\Pi}^\varepsilon_h : \;\; H^1(\Omega) \;\to\; V_h$$

La barre et l'indice supérieur ε seront omis pour le moment.

Le système discret s'écrit :

$$a(u_h, v_h) \;\; = \;\; (\ f, v_h)\ \forall\ v_h \in V_h \tag{3.4}$$

Pour tout v dans V sa projection $\Pi_h v$ est dans V_h, donc

$$a(u_h, \Pi_h v) \;\; = \;\; (\ f, \Pi_h v\ )\ \forall\ v \in V$$

ou bien

$$(Au_h, \Pi_h v)_{V',V} \;\; = \;\; (\ f, \Pi_h v\ )\ \forall\ v \in V$$

nous introduisons l'adjoint $\Pi_h^* : V' \to V'$ du projecteur :

$$(\ \Pi_h^* w\ ,\ v\ )_{V',V} \;\; = \;\; (\ w\ ,\ \Pi_h v\ )_{V',V} \;\; \forall\ w \in V'\ ,\ \forall\ v \in V\ . \tag{3.5}$$

Alors :

$$(\ \Pi_h^* A u_h\ ,\ v\ )_{V',V} \;\; = \;\; (\ \Pi_h^* f\ ,\ v\ )\ \forall\ v \in V \tag{3.6}$$

<u>Lemme</u> 5
L'opérateur $\widetilde{\Pi_h^* A}$ est un isomorphisme :

$$\begin{array}{ll} \widetilde{\Pi_h^* A} : & V_h \to \Pi_h^* A(V) \\ \widetilde{\Pi_h^* A} : & v_h \mapsto \Pi_h^* A v_h\ . \end{array}$$

où les espaces précédents sont respectivement munis de normes de Hilbert sur V et sur V'.$\square$

Pour prouver ce résultat, étudions d'abord $\Pi_h^* A :\ V\ \to\ \Pi_h^* A(V)$. Soit w un élément de V et b l'élément de V' défini par :

$$b \;\; = \;\; \Pi_h^* A\ w\ ; \tag{3.7}$$

nous considérons le problème :

$$\begin{array}{rcl} & Trouver\quad q \in V \quad tel\quad que & \\ \Pi_h^* A q & = & b\ . \end{array} \tag{3.8}$$

L'équation (3.8) a un nombre infini de solutions dans V. Le noyau de $\Pi_h^* A$ est défini comme l'ensemble des $\bar{q}$ tel que pour tout v dans V,

$$(\Pi_h^* A \bar{q}, v)_{V',V} \;\; = \;\; (A\bar{q}, \Pi_h v)_{V',V} \;\; = \;\; 0$$

alors, puisque $\Pi_h(V) = V_h$, tout élément $\bar{q}$ de $Ker\Pi_h^* A$ satisfait :

$$\forall v_h \in V_h, \quad (A\bar{q}, v_h)_{V',V} \ = \ 0$$

Considérons un élément $\bar{q}_h$ de l'intersection de $Ker\ \Pi_h^* A$ avec V_h. Il satisfait :

$$\bar{q}_h \in V_h, \quad et \quad \forall v_h \in V_h, (A\bar{q}_h, v_h)_{V',V} \ = \ a(\bar{q}_h, v_h) \ = \ 0 \ ,$$

si bien qu'en utilisant l'unicité de notre problème discret :

$$Ker\ \Pi_h^* A \cap V_h \quad = \quad 0.$$

Donc $\widetilde{\Pi_h^* A}$ est injectif de V_h dans $\Pi_h^* A(V)$.

Nous observons aussi que pour tout v appartenant à V, la solution q_h de :

$$q_h \in V_h \quad and \quad \forall v_h \in V_h,$$

$$(Aq_h, v_h) \ = \ (Aw, v_h) \ .$$

Donc

$$(Aq_h, \Pi_h v) \ = \ (Aw, \Pi_h v) \ .$$

D'où

$$(\Pi_h^* Aq_h, v) \ = \ (b, v)$$

et donc q_h est solution de (3.8) c'est-à-dire l'antécédent de b. Ce qui prouve la surjectivité de l'opérateur $\widetilde{\Pi_h^* A}$.

Ce qui prouve le Lemme 5.□

Retournant à l'estimation d'erreur, nous concluons comme suit :

$$\Pi_h^* Au_h \quad = \quad \Pi_h^* f \quad dans \quad V' \ .$$

Ceci permet l'analyse d'erreur suivante :

$$\begin{aligned} \Pi_h^* Au_h \ - \ \Pi_h^* A\Pi_h u \quad &= \quad \Pi_h^* f \ - \ \Pi_h^* A\Pi_h u \quad dans \quad V' \\ \Pi_h^* A(u_h \ - \ \Pi_h u) \quad &= \quad \Pi_h^* A(u \ - \ \Pi_h u) \quad dans \quad V' \ . \end{aligned} \tag{3.9}$$

Cette estimation peut être transformée comme suit :

$$\left(\widetilde{\Pi_h^* A}\bar{Y}_h, \phi\right) \ = \ (\nabla(\Pi_h \phi), \nabla(u \ - \ \Pi_h u)) \ .$$

Ceci donne aussi :

$$\left(A\bar{Y}_h, \Pi_h \phi\right) \ = \ (\nabla(\Pi_h \phi), \nabla(u \ - \ \Pi_h u)) \ .$$

Puisque Π_h s'applique sur V, nous pouvons prendre tout ϕ_h dans V_h :

$$\left(A\bar{Y}_h, \phi_h\right) \ = \ (\nabla(\phi_h), \nabla(u \ - \ \Pi_h u)) \ .$$

***Lemme* 6**
On a :

$$u_h \; - \; \Pi_h u \;\; = \;\; \left(\widetilde{\Pi_h^* A}\right)^{-1} \;\; \Pi_h^* A(u \; - \; \Pi_h u) \; . \tag{3.10}$$

ou de manière équivalente :

$$\left(A\bar{Y}_h, \phi_h\right) \;\; = \;\; (\nabla(\phi_h), \nabla(u \; - \; \Pi_h u)) \; . \, \square \tag{3.11}$$

Cette analyse est une analyse *a priori* puisque nous exprimons le terme d'erreur $u_h - \Pi_h u$ comme une fonction de l'inconnue u. Cette partie de l'erreur est un élément de l'espace discret d'approximation et l'équation mise en évidence est discrète. L'erreur d'approximation totale est séparée en deux composantes comme suit :

$$u - u_h \; = \; (u - \Pi_h u) \; + \; (\Pi_h u - u_h) \tag{3.12}$$

Dans la suite, nous distinguons la **composante d'erreur implicite** $\Pi_h u - u_h$, qui est solution du système discret, de l'**erreur d'interpolation** $u - \Pi_h u$.

Remarque 3.2.1
La composante d'erreur implicite précédente est constituée de deux facteurs :
D'abord, le **facteur local résiduel** *:*

$$\Pi_h^* A(u \; - \; \Pi_h u)$$

puis le **facteur nonlocal** *:*

$$\left(\widetilde{\Pi_h^* A}\right)^{-1} \; .\square$$

L'influence de l'erreur locale résiduelle sur le reste du maillage est de la complète responsabilité du facteur nonlocal. Afin d'évaluer précisément l'impact de l'erreur local sur l'erreur d'approximation, il est nécessaire de prendre en compte ce facteur nonlocal. Ceci a pour conséquence l'introduction d'un système adjoint.

Le terme local résiduel donne la taille de l'erreur implicite. Par opposition à une analyse *a posteriori*, nous avons besoin de mettre en évidence la partie principale de l'erreur de troncature.

L'estimation précédente reste vraie lorsque l'on remplace le projecteur de Clément $\bar{\Pi}_h$ par l'interpolateur P_1 standard Π_h .

En effet la seule difficulté liée à l'interpolateur Π_h est qu'il n'est pas défini sur l'espace H^1 tout entier. Mais l'estimation précédente reste vraie pour tout maillage arbitraire τ_h et pour tout projecteur de type Clément $\bar{\Pi}_h^\varepsilon$. Puisque u est supposé appartenir à H^2, nous avons :

$$\bar{\Pi}_h^\varepsilon u_h \;\; - \;\; \Pi_h u \;\; \to \;\; 0 \;\; \text{dans} \;\; H^1(\Omega) \text{ lorsque } \varepsilon \;\; \to \;\; 0. \tag{3.13}$$

Il est donc possible de passer à la limite et nous obtenons ainsi la proposition suivante.

Propriété 1
Sous les hypothèses précédentes et en supposant que u appartient à $H^2(\Omega)$, on a

$$(A(u - \Pi_h u), \phi_h) = (\nabla(\phi_h), \nabla(u - \Pi_h u)) \tag{3.14}$$

où Π_h est l'interpolateur usuel sur $H^2(\Omega)$. $\square$

À partir de maintenant et pour tout le reste de ce chapitre, Π_h sera l'interpolateur usuel.

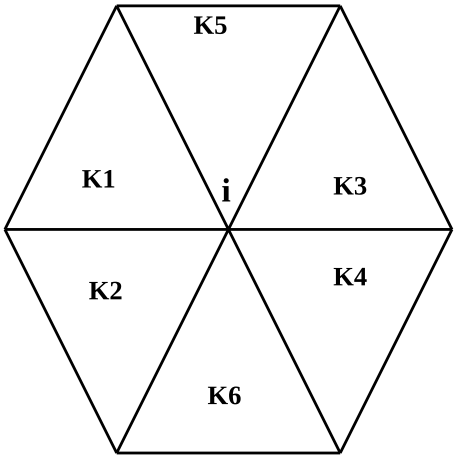

FIG. 3.1 – Support de la fonction de base ϕ_i

3.3 Analyse du second membre

Dans ce paragraphe, nous nous proposons d'étudier la modélisation (dans le cas 2D pour simplifier les notations) du second membre de l'estimation qui sera du type :

$$(A(u - \Pi_h u), \phi_h) \approxeq (\phi_h, U_h)$$

Nous allons étudier successivement la modélisation sur un maillage uniforme isotrope puis étiré. Nous notons par i l'index d'un sommet du maillage et par $Supp(\phi_i)$ le support de la fonction de base ϕ_i (Fig. 3.1).

On a

$$(A(u - \Pi_h u), \phi_h) = (\nabla(\phi_i), \nabla(u - \Pi_h u))$$

$\nabla(\phi_i)$ est constant sur chacun des triangles composant son support.

Ainsi on a :

- $\nabla(\phi_i) = (\frac{1}{\Delta x}, -\frac{1}{2\Delta y})$ sur le triangle K1,
- $\nabla(\phi_i) = (\frac{1}{\Delta x}, \frac{1}{2\Delta y})$ sur le triangle K2,
- $\nabla(\phi_i) = (-\frac{1}{\Delta x}, -\frac{1}{2\Delta y})$ sur le triangle K3,
- $\nabla(\phi_i) = (\frac{1}{\Delta x}, \frac{1}{2\Delta y})$ sur le triangle K4,
- $\nabla(\phi_i) = (0, -\frac{1}{\Delta y})$ sur le triangle K5 et
- $\nabla(\phi_i) = (0, \frac{1}{\Delta y})$ sur le triangle K6.

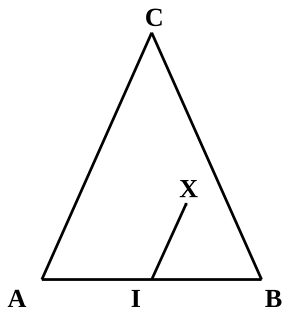

FIG. 3.2 – Triangle générique K

Nous considérons un triangle générique ABC (voir Fig.3.3) tel que le côté AB soit parallèle à l'axe Ox et nous désignons par I le milieu de [A,B].

***Lemme** 7*
On a

$$\begin{aligned}\int_K \frac{\partial}{\partial x}u \; - \; \frac{\partial}{\partial x}\Pi_h u \; dK \quad &= \quad u_{xX}(I)\int_K \vec{IX}\; dK \; - \; \frac{1}{24}u_{xxx}(I)\int_K \Delta x^2 \; dK \\ &+ \frac{1}{2}u_{xXX}(I)\int_K \vec{IX}^2 \; dK \; + \; ....\end{aligned}$$

***Lemme** 8*
Dans le cas d'un maillage uniforme, le terme $\int_{K_1\cup K_2} \vec{IX}\; dK$ *est égal à* $(0,0)$.
Dans le cas d'un maillage non uniforme, le terme $\int_{K_1\cup K_2} \vec{IX}\; dK$ *est égal à* $(0\;,\;\frac{1}{6}(m_x m_y^2 \frac{\partial}{\partial y}(m_y)))$.

***Lemme** 9*
On a

$$\int_{K_1\cup K_2} \frac{\partial}{\partial x}u - \frac{\partial}{\partial x}\Pi_h u \, dK \;=\; -\frac{1}{24}u_{xxx}(I)\int_{K_1\cup K_2} \Delta x^2 \, dK + \frac{1}{2}u_{xXX}(I)\int_{K_1\cup K_2} \vec{IX}^2 \, dK + ... \tag{3.15}$$

On en déduit le résultat suivant

***Lemme** 10*

$$\begin{aligned}\int_{Supp\;\;\Phi_i} \frac{\partial}{\partial x}\phi_i\left(\frac{\partial}{\partial x}u \; - \; \frac{\partial}{\partial x}\Pi_h u\right) \; dK \quad &= \quad mes(Supp\;\Phi_i).(\frac{1}{24}u_{xxxx}(i)\Delta x^2 \\ &+ \quad \frac{1}{12}u_{xxyy}(i)\Delta y^2) \; + \; ..... \\ \int_{Supp\;\;\Phi_i} \frac{\partial}{\partial y}\phi_i\left(\frac{\partial}{\partial x}u \; - \; \frac{\partial}{\partial y}\Pi_h u\right) \; dK \quad &= \quad mes(Supp\;\Phi_i).(\frac{1}{12}u_{xxyy}(i)\Delta x^2) \\ &+ \quad \frac{1}{24}u_{yyyy}(i)\Delta y^2) \; + \; .....\end{aligned}$$

3.4 Modèle en éléments finis

3.4.1 Transformations à partir du discret

On se restreint à la partie implicite $\bar{Y}_h = u_h - \Pi_h u$.

Supposons que le modèle continu $\hat{Y}_h$ que nous proposons pour l'erreur d'approximation s'écrive :

$$a(\hat{Y}_h, \phi) \;=\; (\phi, \hat{U}_h). \tag{3.16}$$

On définira $\hat{U}_h$ plus loin. Ce modèle peut être introduit dans la formulation d'un problème d'optimisation d'une fonctionnelle **continue** par rapport à la solution approchée u_h pour la norme H^1.

Soit :

$$J(h, Y) \;=\; \frac{1}{2}|\nabla Y|^2_{L^2}$$

et

$$j(h) \;=\; J(h, \hat{Y}_h) \;=\; \frac{1}{2}|\nabla \hat{Y}_h|^2_{L^2}$$

où $\hat{Y}_h$ est la solution de :

$$(\nabla \hat{Y}_h, \nabla \phi) \;=\; (\phi, \hat{U}_h) \quad \forall \phi \;.$$

Le problème de minimisation s'écrit :

$$\begin{aligned} \text{Trouver } h \;&=\; \text{Argmin } J(h, \hat{Y}_h) \quad \text{avec} \\ (\nabla \hat{Y}_h, \nabla \phi) \;&=\; (\phi, \hat{U}_h) \quad \forall \phi \;. \end{aligned} \tag{3.17}$$

La seconde équation de (3.17) est son équation d'état, permettant de calculer la variable d'état $\hat{Y}_h$ à partir du maillage h. Le vecteur $\hat{U}_h$ dépend explicitement du maillage h. Le sustème d'optimalité pour la minimisation de j peut formellement s'écrire comme suit :

$$\begin{aligned} (\nabla \hat{Y}_h, \nabla \phi) \;&=\; (\phi, \hat{U}_h) \quad \forall \phi \\ (\nabla \phi, \nabla P) \;&=\; \frac{\partial J}{\partial Y}.\phi \quad \forall \phi \\ \frac{\partial J}{\partial h}.\delta h \;-\; (P, \frac{\partial \hat{U}_h}{\partial h}.\delta h) \;&=\; 0 \;. \end{aligned}$$

Avec :

$$\begin{aligned} \frac{\partial J}{\partial Y}.\phi \;&=\; (\nabla \hat{Y}_h, \nabla \phi) \\ \frac{\partial J}{\partial h}.\delta h \;&=\; 0 \;. \end{aligned}$$

P est un état adjoint. La synthèse des deux systèmes donne :

$$(\nabla \hat{Y}_h, \nabla \phi) \;=\; (\phi, \hat{U}_h) \quad \forall \phi \tag{3.18}$$

$$P \;=\; \hat{Y}_h \tag{3.19}$$

$$(P, \frac{\partial \hat{U}_h}{\partial h}.\delta h) \;=\; 0 \quad \forall \delta h. \tag{3.20}$$

3.4.2 Cas isotrope

Dans le cas isotrope, nous considérons que tout élément est quasi-équilatéral. Une manière d'écrire cela dans le cas de simplexes est de supposer que :

$$\Delta x \;=\; \Delta y \quad \text{partout} ,$$

et puisque nous ne voulons pas prendre en compte une quelconque direction dans l'espace dans notre modèle, nous posons :

$$\hat{U}_h \;=\; (u_{xxxx} \;+\; 4u_{xxyy} \;+\; u_{yyyy})(\Delta x)^2$$

On définit la densité de nœuds d et le terme de troncature $m(u)$:

$$d \;=\; (\Delta x)^{-2} \quad \text{et} \quad M(u) \;=\; (|\lambda_u|, |\lambda_u|)^T \;. \tag{3.21}$$

À partir de ce contexte nous pouvons différentier comme suit :

$$\frac{\partial U_d}{\partial d}\,.\delta d \;=\; = \;\left(-(u_{xxxx} \;+\; 4u_{xxyy} \;+\; u_{yyyy})\frac{1}{d^2}\delta d\right)^T \;.$$

Alors les conditions d'optimalité peuvent s'écrire :

$$(\nabla\hat{Y}_d, \nabla\phi) \;=\; (\phi, \hat{U}_d) \quad \forall\phi \tag{3.22}$$

$$P \;=\; \hat{Y}_d \tag{3.23}$$

$$(P, \frac{\partial \hat{U}_d}{\partial d}.\delta d) \;=\; 0 \quad \forall\delta h. \tag{3.24}$$

3.4.3 Cas anisotrope

En tout point du domaine nous avons besoin de trouver les caractéristiques du maillage sous la forme d'une métrique, ou, de manière équivalente, par une (2D) ou deux (3D) directions d'étirement, le (les) rapport(s) d'étirement(s) et la densité de maille.
Notre stratégie se décompose en deux étapes :

- En tout point du domaine Ω, nous minimisons l'erreur en supposant comme au chapitre précédent que tout étirement raisonnable est aligné avec le hessien $H(u)$ de u. En termes de métrique $\mathcal{M}$, cela s'écrit :

$$\forall\ X \in \Omega, \quad \mathcal{R}_{\mathcal{M}}(X) \quad = \quad \mathcal{R}_H(X)$$

- Alors nous sommes ramenés à une formulation simplifiée dans laquelle seuls les rapports d'étirement et la densité de maille sont inconnus et à chercher globalement dans tout le domaine.

Nous présentons la suite dans le cas 2D pour plus de simplicité.
Posons :

$$\hat{U}_h^a \quad = \quad \mathcal{R}_H\ (u_{\xi\xi\xi\xi}(\Delta\xi)^2\ +\ 2u_{\xi\xi\eta\eta}((\Delta\xi)^2 + (\Delta\eta)^2)\ +\ u_{\eta\eta\eta\eta}(\Delta\eta)^2)\mathcal{R}_H^T$$

3.4.4 Illustrations numériques dans H^1

Nous présentons maintenant quelques calculs sur l'analyse H^1. Nous avons comparé l'erreur implicite avec son modèle continu. La fonction u_1 est d'abord considérée :

$$u(x,y) \;=\; u_1(x,y) \;=\; (x^2-x)(y^2-y)$$

On constate un très bon accord entre ces deux quantités (voir Fig.2.8).

On a ensuite considéré la fonction u_2 suivante :

$$u_2(x,y) \;=\; (1-exp(-\alpha x)-(1-exp(-\alpha))x)*4*y*(1-y) \text{ avec } \alpha \;=\; 10.0$$

Pour cette fonction, qui comporte une couche limite sur le côté vertical gauche du domaine, l'accord est un peu moins bon, le modèle étant plus concentré au voisinage de la couche limite (Fig.2.9). Nous constatons que le maillage adapté possède un raffinement plus important au voisinage de la couche limite.

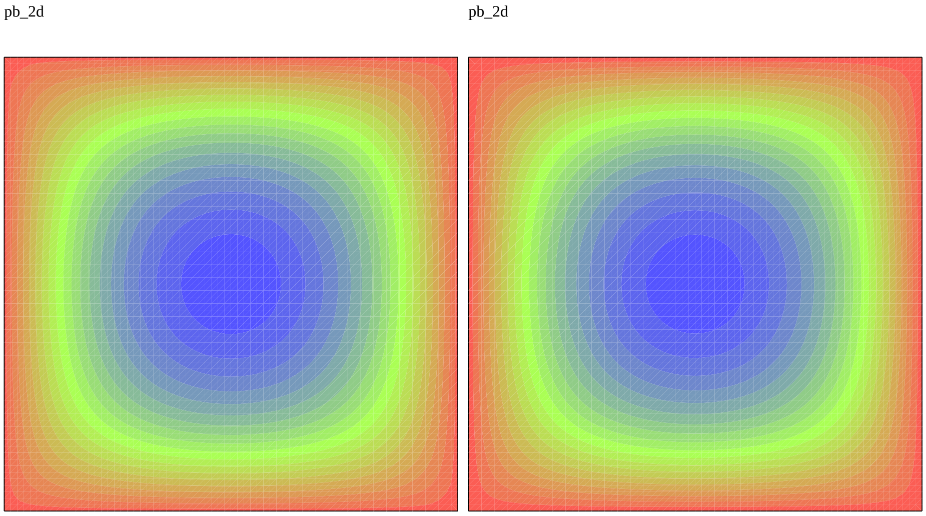

FIG. 3.3 – Comparaison entre erreur exacte et modèle d'erreur, fonction u_1

3.5 Conclusions

Le but de ce chapitre est de proposer une nouvelle approche basée sur l'analyse en Éléments Finis pour l'adaptation du maillage d'une EDP. Cette approche est par certains aspects intermédiaire entre d'une part la voie simplifiée et assez empirique de l'adaptation à une erreur d'interpolation, et, d'autre part, les méthodes basées sur les estimations *a posteriori*.

Il y a une première phase d'analyse *a priori* rigoureuse, suivie d'une seconde phase plus empirique de modélisation.

L'analyse *a priori* a pour but d'identifier les parties principales de l'erreur. Nous l'avons réalisée pour la semi-norme H^1.
Nous avons aussi proposé une analyse pour la norme L^2.

Nous proposons ensuite plusieurs modélisations pour le cas H^1.

Il résulte de notre approche une formulation en contexte continu de la recherche du meilleur maillage, minimisant une certaine fonctionnelle.

Nous construisons enfin des algorithmes d'optimisation pour trouver ces minima aussi bien pour des maillages isotropes qu'anisotropes.

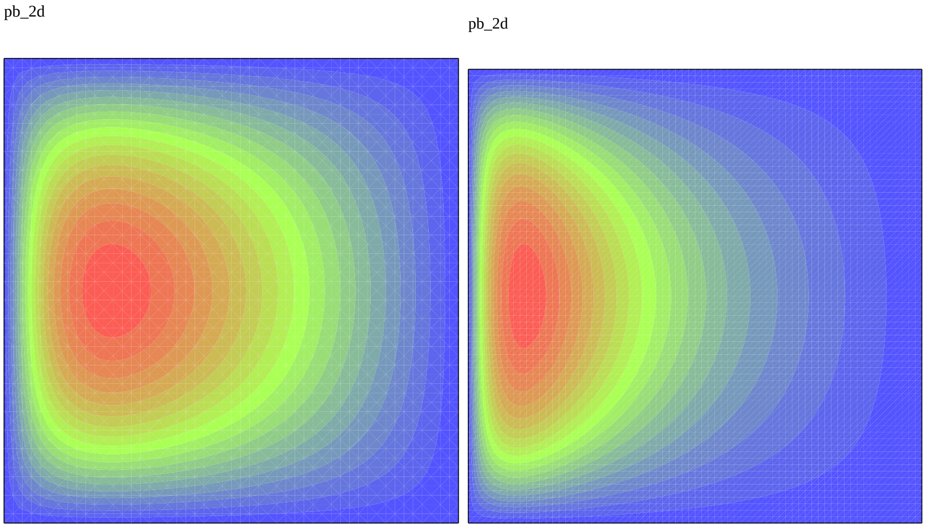

FIG. 3.4 – Comparaison entre erreur exacte et modèle d'erreur, fonction u_2

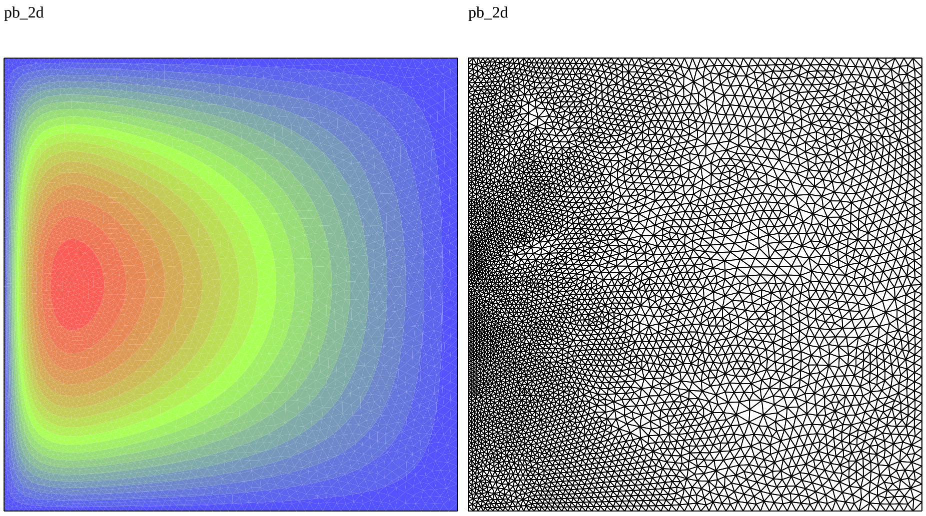

FIG. 3.5 – Nouvelle solution et nouveau maillage , fonction u_2

Conclusion générale

Dans cette thèse, nous avons présenté notre contribution dans les trois domaines complémentaires suivants :

- La différentiation automatique de programmes,
- L'optimisation de formes pour de grands systèmes,
- L'adaptation de maillages.

Dans ces trois directions, le principe fondamental est la mise sous forme d'une problème de Contrôle Optimal, l'introduction d'un état adjoint, l'application d'un algorithme de type gradient.

Dans le chapitre 1 de la partie 1, nous avons exposé une méthode de calcul de gradients par Différentiation Automatique pour un problème classique d'optimisation de formes. Nous avons expliqué comment déduire un gradient exact basé sur un état adjoint sans stocker explicitement le Jacobien. Cette méthode se révèle particulièrement intéressante pour le cas des systèmes issus de la Mécanique des Fluides Numériques, où les matrices jacobiennes exactes sont souvent trop grandes pour être stockées. Dans notre méthode nous n'utilisons pas le jacobien transposé mais seulement le résidu du système adjoint. Notre autre contribution dans le domaine de la différentiation automatique de programme est que nous proposons un mode adjoint de la DA amélioré, qui utilise beaucoup moins d'espace mémoire.

Dans ce contexte nous avons validé à la fois l'application de la DA et le nouvel outil de l'équipe, TAPENADE.

Dans le chapitre 2 (partie 2), nous avons proposé une méthode destinée à résoudre une classe de problèmes d'optimisation avec contraintes égalités. Cette méthode repose sur l'utilisation d'une itération peu coûteuse pour résoudre l'état et l'adjoint. Le nouvel algorithme permet une résolution quasi-simultanée du système KKT. Il étend les idées de SQP, en utilisant en particulier l'accélération BFGS. La méthode proposée travaille essentiellement comme les autres méthodes one shot, mais avec robustesse. Nous pensons que ce type d'algorithme est qualifié pour résoudre de manière robuste de nouveaux problèmes sans risque de divergence et sans nécessité de réglages préliminaires des paramètres.

Dans le chapitre 3 (partie 2), nous étudions une nouvelle stratégie de préconditionnement pour l'optimisation de formes. Nous construisons un préconditionnement multiniveau additif à partir de (a) le principe classique de Bramble-Pasciak-Xu et (b) le principe d'agglomération. On peut montrer via des résultats d'analyse fonctionnelle que l'itération de gradient entraine une perte de régularité de l'itéré. Cette perte doit être compensé par un gain de régularité du préconditionneur. Nous spécifions aisément le gain en régularité de notre préconditionneur avec un seul paramètre réel ce qui aide à l'utiliser avec une méthode BFGS.

Ces deux travaux concourent à démontrer que l'optimisation de forme en Aérodynamique Appliquée est de plus en plus abordable sur des micro-ordinateurs pour le modèle Euler. En Navier-Stokes, les méthodes proposées s'appliquent aussi et devraient contribuer à permettre l'optimisation sur des calculateurs parallèles de taille moyenne.

Dans la partie 3, nous avons exposé notre contribution dans le domaine de l'adaptation de maillage.

Dans le chapitre 1 (partie 3), nous avons étudié le problème du meilleur maillage adapté pour de l'interpolation pure. Nous spécifions le maillage par une métrique et nous modélisons l'erreur par le premier terme de la série de Taylor de l'erreur d'interpolation. La résolution du système d'optimalité nous permet alors de déterminer une expression complètement explicite de la métrique optimale en fonction de la fonction à adapter.

Dans les chapitres 2 et 3 (partie 3), nous étendons la méthode proposée dans le chapitre 1 (partie 3) au problème de l'adaptation de maillage pour EDP. Notre méthode repose sur une analyse *a priori* rigoureuse puis sur une modélisation. Il en résulte une formulation en contexte continu de la recherche du meilleur maillage, minimisant une certaine fonctionnelle. Cette nouvelle application des méthodes de l'optimisation avec adjoint (et DA) est en cours de mise en œuvre et sera appliquée à une stratégie combinant adaptation anisotrope avec adjoint, d'une part, et optimisation de forme, d'autre part, pour l'optimisation de l'avion supersonique.

Les idées proposées se prêtent à de nombreuses extensions non nécessairement triviales :

- La mise au point d'un outil totalement automatique pour la prise en compte des boucles parallèles reste un problème ouvert.
- La méthode one-shot devrait pouvoir s'étendre à l'optimisation avec contraintes inégalités, par exemple par l'introduction de techniques de point intérieurs telles que les fonctions barrières.
- L'extension de la méthode multiniveau à de plus hauts degrés de régularisation est une question pas plus facile à aborder.
- La méthodologie de la métrique continue a démontré qu'elle ouvrait pas mal de questions, notamment sur l'ordre de convergence, abordé dans ce mémoire seulement en 1D.
- L'extension de la théorie du dernier chapitre à d'autres modèles plus réalistes est aussi une direction d'investigation tout à fait stimulante.

A

Analyse du second membre

Dans ce paragraphe, nous allons développer les calculs nécessaires à la modélisation (dans le cas 2D pour simplifier les notations) du second membre de l'estimation :

$$(A(u - \Pi_h u), \phi_h) \approxeq (\nabla(\phi_h), U_h)$$

Nous notons par i l'index d'un sommet du maillage et par $Supp(\phi_i)$ le support de la fonction de base ϕ_i. (Fig. A.1)

On a

$$(\nabla(\phi_i), U_h) = (\nabla(\phi_i), \nabla(u - \Pi_h u))$$

$\nabla(\phi_i)$ est constant sur chacun des triangles composant son support.

Ainsi on a

- $\nabla(\phi_i) = (\frac{1}{\Delta x}, -\frac{1}{2\Delta y})$ sur le triangle K1,
- $\nabla(\phi_i) = (\frac{1}{\Delta x}, \frac{1}{2\Delta y})$ sur le triangle K2,
- $\nabla(\phi_i) = (-\frac{1}{\Delta x}, -\frac{1}{2\Delta y})$ sur le triangle K3,
- $\nabla(\phi_i) = (\frac{1}{\Delta x}, \frac{1}{2\Delta y})$ sur le triangle K4,
- $\nabla(\phi_i) = (0, -\frac{1}{\Delta y})$ sur le triangle K5 et
- $\nabla(\phi_i) = (0, \frac{1}{\Delta y})$ sur le triangle K6.

Nous considérons un triangle générique ABC (voir Fig.A) tel que le côté AB soit parallèle à l'axe Ox et nous désignons par I le milieu de [A,B].

Nous désignons par I le milieu du segment [A,B].

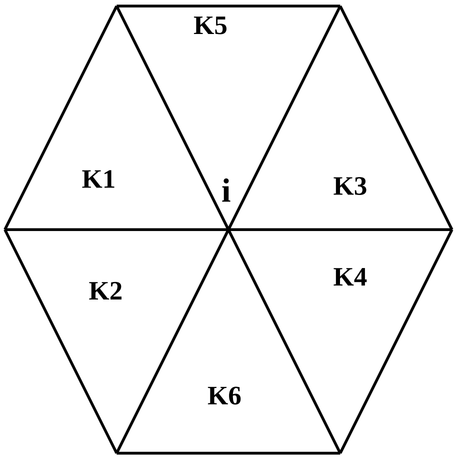

FIG. A.1 – Support de la fonction de base ϕ_i

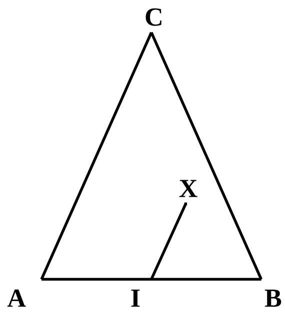

FIG. A.2 – Triangle générique K

$$
\begin{aligned}
u(B) &= u(I) + u_X(I)\vec{IB} + \frac{1}{2}u_{XX}(I)\vec{IB}.\vec{IB} + \frac{1}{6}u_{XXX}(I)\vec{IB}.\vec{IB}.\vec{IB} + \qquad \text{(A.1)}\\
u(A) &= u(I) - u_X(I)\vec{IB} + \frac{1}{2}u_{XX}(I)\vec{IB}.\vec{IB} - \frac{1}{6}u_{XXX}(I)\vec{IB}.\vec{IB}.\vec{IB} + ...
\end{aligned}
$$

Donc, par soustraction, nous obtenons

$$
\begin{aligned}
u(B) - u(A) &= 2.u_X(I)\vec{IB} + \frac{1}{3}u_{XXX}(I)\vec{IB}.\vec{IB}.\vec{IB} + ...\\
&= u_X(I)\vec{AB} + \frac{1}{3}u_{XXX}(I)\frac{\vec{AB}}{2}.\frac{\vec{AB}}{2}.\frac{\vec{IB}}{2} + ...\\
&= u_X(I)\vec{AB} + \frac{1}{24}u_{XXX}(I)\vec{AB}.\vec{AB}.\vec{AB} + ...
\end{aligned}
$$

On en déduit que

$$
\begin{aligned}
\frac{\partial}{\partial x}\Pi_h u &= u_x(I) + \frac{1}{24}u_{xxx}(I)\Delta x^2 + ... \qquad \text{(A.2)}\\
\frac{\partial}{\partial x}u &= u_x(I) + u_{xX}(I)\vec{IX} + \frac{1}{2}u_{xXX}(I)\vec{IX}^2 + \frac{1}{6}u_{xXXX}(I)\vec{IX}^3 + ...
\end{aligned}
$$

En soustrayant ces deux développements, nous obtenons :

$$
\frac{\partial}{\partial x}u - \frac{\partial}{\partial x}\Pi_h u = u_{xX}(I)\vec{IX} - \frac{1}{24}u_{xxx}(I)\Delta x^2 + \frac{1}{2}u_{xXX}\vec{IX}^2 + ... \qquad \text{(A.3)}
$$

Puis en prenant l'intégrale de ce terme sur les deux triangles K_1 et K_2, on obtient

$$
\begin{aligned}
\int_{K_1\cup K_2} \frac{\partial}{\partial x}u - \frac{\partial}{\partial x}\Pi_h u \, dK &= u_{xX}(I)\int_{K_1\cup K_2} \vec{IX}\, dK - \frac{1}{24}u_{xxx}(I)\int_{K_1\cup K_2} \Delta x^2\, dK\\
&\quad + \frac{1}{2}u_{xXX}(I)\int_{K_1\cup K_2} \vec{IX}^2\, dK + ...
\end{aligned}
$$

En maillage uniforme, le terme $\int_{K_1\cup K_2} \vec{IX}\, dK$ est égal à 0.

D'où

$$
\begin{aligned}
\int_{K_1\cup K_2} \frac{\partial}{\partial x}u - \frac{\partial}{\partial x}\Pi_h u \, dK &= -\frac{1}{24}u_{xxx}(I)\int_{K_1\cup K_2} \Delta x^2\, dK\\
&\quad + \frac{1}{2}u_{xXX}(I)\int_{K_1\cup K_2} \vec{IX}^2\, dK + ...
\end{aligned}
$$

On pose $w = u_x$ et $\vec{IX} = (a, b)$.

On a

$$\begin{aligned}
\frac{1}{2}u_{xXX}(I)\int_{K_1\cup K_2} I\vec{X}^2\ dK &= \frac{1}{2}\int_{K_1\cup K_2}\frac{\partial^2 w}{\partial x^2}(I)a^2 \ +\ 2\frac{\partial^2 w}{\partial x\partial y}(I)ab \ +\ \frac{\partial^2 w}{\partial y^2}(I)b^2 \qquad (A.4)\\
&= \frac{1}{2}\left(\frac{\partial^2 w}{\partial x^2}(I)\int_{K_1\cup K_2} a^2 \ +\ 2\frac{\partial^2 w}{\partial x\partial y}(I)\int_{K_1\cup K_2} ab \ +\ \frac{\partial^2 w}{\partial y^2}(I)\int_{K_1\cup K_2} b^2\right)\\
&= \frac{1}{2}\left(\frac{\partial^2 w}{\partial x^2}(I)\int_{K_1\cup K_2} a^2 \ +\ 2\frac{\partial^2 w}{\partial x\partial y}(I).0 \ +\ \frac{\partial^2 w}{\partial y^2}(I)\int_{K_1\cup K_2} b^2\right)\\
&= \frac{1}{2}\left(\frac{\partial^2 w}{\partial x^2}(I)\int_{K_1\cup K_2} a^2 \ +\ \frac{\partial^2 w}{\partial y^2}(I)\int_{K_1\cup K_2} b^2\right)\\
&= \frac{1}{2}\left(\frac{\partial^2 w}{\partial x^2}(I)\int_{K_1\cup K_2} a^2 \ +\ \frac{\partial^2 w}{\partial y^2}(I)\int_{K_1\cup K_2} b^2\right)
\end{aligned}$$

(A.5)

On a

$$\begin{aligned}
\int_{K_1\cup K_2} a^2 &= \frac{1}{3}((\frac{\Delta x}{2})^2+(\frac{\Delta x}{2})^2+0).mes(K_1)\\
&+ \frac{1}{3}((\frac{\Delta x}{2})^2+(\frac{\Delta x}{2})^2+0).mes(K_2)\\
&= \frac{1}{6}(\Delta x)^2.mes(K_1) \ +\ \frac{1}{6}(\Delta x)^2.mes(K_2)\\
&= \frac{1}{6}(\Delta x)^2.mes(K_1\cup K_2) \qquad (A.6)
\end{aligned}$$

$$\begin{aligned}
\int_{K_1\cup K_2} b^2 &= \frac{1}{3}((\frac{\Delta y}{2})^2+(\frac{\Delta y}{2})^2+0).mes(K_1)\\
&+ \frac{1}{3}((\frac{-\Delta y}{2})^2+(\frac{-\Delta y}{2})^2+0).mes(K_2)\\
&= \frac{1}{6}(\Delta y)^2.mes(K_1) \ +\ \frac{1}{6}(\Delta y)^2.mes(K_2)\\
&= \frac{1}{6}(\Delta y)^2.mes(K_1\cup K_2) \qquad (A.7)
\end{aligned}$$

$$\begin{aligned}
\int_{K_1\cup K_2}\frac{\partial}{\partial x}(u\ -\ \Pi_h u)\ dK &= -\ mes(K_1\cup K_2)\frac{1}{24}u_{xxx}(I)\Delta x^2 \qquad (A.8)\\
&+ \frac{1}{2}(mes(K_1\cup K_2)\frac{1}{6}u_{xxx}(I)\Delta x^2\\
&+ mes(K_1\cup K_2)\frac{1}{6}u_{xyy}(I)\Delta y^2)\\
&= mes(K_1\cup K_2)\left(\frac{1}{24}u_{xxx}(I)\Delta x^2 \ +\ \frac{1}{12}u_{xyy}(I)\Delta y^2\right)
\end{aligned}$$

On en conclut

$$\int_{Supp\Phi_i} \frac{\partial}{\partial x}\phi_i \left(\frac{\partial}{\partial x}u \; - \; \frac{\partial}{\partial x}\Pi_h u\right) \, dK \quad = \quad mes(Supp\Phi_i).(\frac{1}{24}u_{xxxx}(\frac{I_1+I_2}{2})\Delta x^2 + \frac{1}{12}u_{xxyy}(\frac{I_1+I_2}{2})\Delta y^2) + ...$$

Cas non uniforme :

Soient G_1 le centre de gravité du triangle K_1 et G_2 le centre de gravité du triangle K_2.

on a

$$\begin{aligned} \int_{K_1\cup K_2} I\vec{X}dX &= \int_{K_1} I\vec{G}_1 dX \; + \; \int_{K_2} I\vec{G}_2 dX \\ &= \int_{K_1} (0 \; , \; \frac{\Delta y^+}{3}) \; - \; (0 \; , \; \frac{\Delta y^-}{3})) \end{aligned} \tag{A.9}$$

Donc

$$\begin{aligned} \int_{K_1\cup K_2} I\vec{X}dX &= (0 \; , \; \frac{\Delta x^- \Delta y^+}{2} \times \frac{\Delta y^+}{3} \; - \; \frac{\Delta x^- \Delta y^-}{2} \times \frac{\Delta y^-}{3}) \\ &= (0 \; , \; \frac{\Delta x^-}{6}(\Delta y^+ \; - \; \Delta y^-)(\Delta y^+ \; + \; \Delta y^-)) \\ &= (0 \; , \; \frac{\Delta x^-}{6}\frac{(\Delta y^+ \; - \; \Delta y^-)}{\frac{1}{2}(\Delta y^+ \; + \; \Delta y^-)}\frac{1}{2}(\Delta y^+ \; + \; \Delta y^-)(\Delta y^+ \; + \; \Delta y^-)) \\ &\approxeq (0 \; , \; \frac{1}{6}(m_x m_y^2 \frac{\partial}{\partial y}(m_y))) \end{aligned} \tag{A.10}$$

On déduit de la relation (A.4) que :

$$\int_{K_1\cup K_2} \frac{\partial}{\partial x}u \; - \; \frac{\partial}{\partial x}\Pi_h u \; dK \quad \approxeq \quad \frac{1}{3}(u_{xy}(I) m_x m_y^2 \frac{\partial}{\partial y}(m_y))) \tag{A.11}$$

$$+ \; mes(Supp\Phi_i).(\frac{1}{24}u_{xxxx}(\frac{I_1+I_2}{2})\Delta x^2 \; + \; \frac{1}{12}u_{xxyy}(\frac{I_1+I_2}{2})\Delta y^2) + ...$$

Dérivation en y

On désigne par A,B, et C les trois sommets d'un triangle dont le côté AB est parallèle à l'axe (Ox) (voir figure 3.3).
On a

$$\frac{\partial}{\partial y}\Pi_h u \quad = \quad \frac{2 * u(C) \; - \; u(A) \; - \; u(B)}{2 * \Delta y} \tag{A.12}$$

On désigne par i le centre de la cellule.

$$u(C) \;=\; u(i) \;+\; u_X(i)\vec{iC} \;+\; \frac{1}{2}u_{XX}(i)\vec{iC}^2 \;+\; \frac{1}{6}u_{XXX}(i)\vec{iC}^3 + ... \qquad \text{(A.13)}$$

$$\begin{aligned} 2*u(C) \;-\; u(A) \;-\; u(B) \quad &= \quad 0 \;+\; u_X(i).(2\vec{iC} \;-\; \vec{iA} \;-\; \vec{iB}) \\ &+ \quad \frac{1}{2}u_{XX}(i).(2\vec{iC}^2 \;-\; \vec{iA}^2 \;-\; \vec{iB}^2) \\ &+ \quad \frac{1}{6}u_{XXX}(i).(2\vec{iC}^3 \;-\; \vec{iA}^3 \;-\; \vec{iB}^3) \;+\; .... \end{aligned} \qquad \text{(A.14)}$$

$$\frac{\partial}{\partial y}u(X) \;=\; \frac{\partial}{\partial y}u(i) \;+\; u_{yX}(i)\vec{iX} \;+\; \frac{1}{2}u_{yXX}(i).\vec{iX}^2 \;+\; \frac{1}{6}u_{yXXX}(i).\vec{iX}^3 \;+\; .... \text{(A.15)}$$

$$\begin{aligned} \int_K \frac{\partial}{\partial y}u(X) \;-\; \frac{\partial}{\partial y}\Pi_h u \quad &= \quad \int_K \frac{\partial}{\partial y}u(X) \;-\; \frac{2*u(C) \;-\; u(A) \;-\; u(B)}{2*\Delta y}dK \\ &= \quad u_{XX}(i).e_2\int_K(\;\vec{iX} \;-\; \frac{1}{2}.\frac{2\vec{iC}^2 \;+\; \vec{iA}^2 \;+\; \vec{iB}^2}{2\Delta y})dK \\ &+ \quad u_{XXX}(i).e_2\int_K(\;\frac{1}{2}\vec{iX}^2 \;-\; \frac{1}{6}.\frac{(2\vec{iC}^3 \;+\; \vec{iA}^3 \;+\; \vec{iB}^3)}{2\Delta y})dK \end{aligned} \qquad \text{(A.16)}$$

avec $e_2 \;=\; (0,1)$.

Cas 1) Triangle K_1

$$\int_K \vec{iX} \;=\; mes(K).\vec{iG} \;=\; \frac{mes(K)}{3}(-\frac{3\Delta x}{2}\;,\;\Delta y) \qquad \text{(A.17)}$$

On a $\vec{iA} \;=\; (-\Delta x,\; 0)$, $\;\vec{iB} \;=\; (0,\; 0)$, et $\;\vec{iC} \;=\; (-\frac{\Delta x}{2},\; \Delta y)$.

Donc

$$u_{XX}(i).e_2\int_K(\;\vec{iX} \;-\; \frac{2\vec{iC}^2 \;+\; \vec{iA}^2 \;+\; \vec{iB}^2}{4\Delta y}) \qquad \text{(A.18)}$$

$$= \quad mes(K).(u_{xx}(i)(\frac{2.\frac{(-\Delta x)^2}{4} \;+\; (\Delta x)^2}{4.\Delta y} \;-\; \frac{\Delta x}{2}) \;+\; u_{xy}(i).0 \;+\; u_{yy}(i).(\frac{\Delta y}{3} \;-\; \frac{2.(\Delta y)^2}{4.\Delta y}))$$

$$= \quad mes(K).(u_{xx}(i)(\frac{3}{8}.\frac{(\Delta x)^2}{\Delta y} \;-\; \frac{\Delta x}{2}) \;+\; u_{yy}(i).(-\frac{1}{6}\;.\Delta y))$$

Cas 2) Triangle K_2.

$$\int_K i\vec{X} \; = \; mes(K).i\vec{G} \; = \; \frac{mes(K)}{3}(\frac{3\Delta x}{2} \; , \; -\Delta y) \tag{A.19}$$

On a $i\vec{A} \; = \; (0 \; , \; 0)$, $\; i\vec{B} \; = \; (\Delta x, \; 0)$, et $\; i\vec{C} \; = \; (\frac{\Delta x}{2}, \; -\Delta y)$.

Donc

$$\begin{aligned}
& u_{XX}(i).e_2 \int_K (\; i\vec{X} \; - \; \frac{1}{2}.\frac{2i\vec{C}^2 \; + \; i\vec{A}^2 \; + \; i\vec{B}^2}{2\Delta y})dK \\
= \quad & mes(K).(u_{xx}(i)(\frac{2.\frac{(-\Delta x)^2}{4} \; + \; (\Delta x)^2}{4.\Delta y} \; + \; \frac{\Delta x}{2}) \; + \; u_{xy}(i).0 \; + \; u_{yy}(i).(\frac{\Delta y}{3} \; - \; \frac{2.(\Delta y)^2}{4.\Delta y})) \\
= \quad & mes(K).(u_{xx}(i)(-\frac{3}{8}.\frac{(\Delta x)^2}{\Delta y} \; + \; \frac{\Delta x}{2}) \; + \; u_{yy}(i).(\frac{1}{6} \; .\Delta y))
\end{aligned}$$

Ce terme se détruit avec le terme symétrique obtenu pour le triangle K_1 et donc on en déduit que

$$\begin{aligned}
u_{XX}(i).e_2 \int_{K_1 \cup K_2} (\; i\vec{X} \; - \; \frac{1}{2}.\frac{2i\vec{C}^2 \; + \; i\vec{A}^2 \; + \; i\vec{B}^2}{2\Delta y})dK \quad &= \quad 0 \\
u_{XX}(i).e_2 \int_{K_3 \cup K_4} (\; i\vec{X} \; - \; \frac{1}{2}.\frac{2i\vec{C}^2 \; + \; i\vec{A}^2 \; + \; i\vec{B}^2}{2\Delta y})dK \quad &= \quad 0
\end{aligned}$$

Cas d'un maillage non uniforme :

$$u_{XX}(i).e_2 \int_{K_1 \cup K_2} (\; i\vec{X} \; - \; \frac{1}{2}.\frac{2i\vec{C}^2 \; + \; i\vec{A}^2 \; + \; i\vec{B}^2}{2\Delta y})dK \tag{A.20}$$

$$= \quad mes(K_1).(u_{xx}(i)(-\frac{5}{16}.\frac{(\Delta x^-)^2}{\Delta y^+}) \; + \; u_{yy}(i).(\frac{1}{6} \; .\Delta y^+)) \tag{A.21}$$

$$- \quad mes(K_2).(u_{xx}(i)(-\frac{5}{16}.\frac{(\Delta x^-)^2}{\Delta y^-}) \; + \; u_{yy}(i).(\frac{1}{6} \; .\Delta y^-))$$

$$= \quad \frac{\Delta x^- \Delta y^+}{2}.(u_{xx}(i)(-\frac{5}{16}.\frac{(\Delta x^-)^2}{\Delta y^+}) \; + \; u_{yy}(i).(\frac{1}{6} \; .\Delta y^+)) \tag{A.22}$$

$$- \quad \frac{\Delta x^- \Delta y^-}{2}..(u_{xx}(i)(-\frac{5}{16}.\frac{(\Delta x^-)^2}{\Delta y^-}) \; + \; u_{yy}(i).(\frac{1}{6} \; .\Delta y^-))$$

$$= \quad \frac{1}{2}.(u_{xx}(i)(-\frac{5}{16}.(\Delta x^-)^3) \; + \; u_{yy}(i).(\frac{1}{6} \; .\Delta x^-(\Delta y^+)^2)) \tag{A.23}$$

$$- \quad \frac{1}{2}.(u_{xx}(i)(-\frac{5}{16}.(\Delta x^-)^3) \; + \; u_{yy}(i).(\frac{1}{6} \; .\Delta x^-(\Delta y^-)^2))$$

$$= \quad u_{yy}(i).\frac{1}{6} \; . \left(\Delta x^-(\Delta y^+)^2 \; - \; .\Delta x^-(\Delta y^-)^2\right) \tag{A.24}$$

$$= \quad u_{yy}(i).\frac{1}{6} \; .\Delta x^-(\Delta y^+ \; - \; \Delta y^-)(\Delta y^+ \; + \; \Delta y^-) \tag{A.25}$$

On en déduit d'apres la formule (A.10) que :

$$\begin{aligned} u_{XX}(i).e_2 \int_{K_1 \cup K_2} (\; i\vec{X} \; - \; \frac{1}{2}.\frac{2i\vec{C}^2 \; + \; i\vec{A}^2 \; + \; i\vec{B}^2}{2\Delta y})dK \\ \approxeq \; (0 \; , \; u_{yy}(i)\frac{1}{6}(m_x m_y^2 \frac{\partial}{\partial y}(m_y))) \end{aligned} \tag{A.26}$$

On a

$$\begin{aligned} & u_{XXX}(i).e_2 \int_{K_1 \cup K_2} (\; \frac{1}{2} i\vec{X}^2) \; dK \\ = \; & u_{yxx}(i) \int_{K_1 \cup K_2} (\; \frac{1}{2}x^2) \; dK \; + \; u_{yxy}(i) \int_{K_1 \cup K_2} \frac{1}{2} xy dK \; + \; u_{yyy}(i) \int_{K_1 \cup K_2} \frac{1}{2} y^2 dK \\ = \; & u_{yxx}(i) mes(K_1 \cup K_2) \frac{1}{12}(\Delta x)^2 \; + \; 0 \; + \; u_{yyy}(i) mes(K_1 \cup K_2) \frac{1}{12}(\Delta y)^2 \end{aligned} \tag{A.27}$$

D'autre part

$$\begin{aligned} & u_{XXX}(i).(2i\vec{C}^3 \; - \; i\vec{A}^3 \; - \; i\vec{B}^3) \\ = \; & u_{xxx}(i).(\frac{\Delta x^3}{8} - \frac{\Delta x^3}{8}) \; + \; u_{xxy}(i).(0) \; + \; u_{xyy}(i).(0) \; + \; u_{yyy}(i).(\frac{1}{2}.\Delta y^3) \\ = \; & u_{yyy}(i).(\frac{1}{2}.\Delta y^3) \end{aligned} \tag{A.28}$$

Donc

$$\frac{1}{6}.\frac{(2i\vec{C}^3 \; + \; i\vec{A}^3 \; + \; i\vec{B}^3)}{2\Delta y} \; = \; u_{yyy}(i).(\frac{1}{24}.\Delta y^2) \tag{A.29}$$

On en déduit que

$$\begin{aligned} & u_{XXX}(i).e_2 \int_{K_1 \cup K_2} (\; \frac{1}{2} i\vec{X}^2 \; - \; \frac{1}{6}.\frac{(2i\vec{C}^3 \; + \; i\vec{A}^3 \; + \; i\vec{B}^3)}{2\Delta y}) \; dK \\ = \; & mes(K_1 \cup K_2) \left(u_{yxx}(i)\frac{1}{12}(\Delta x)^2 \; + \; u_{yyy}(i)\frac{1}{12}(\Delta y)^2 \; - \; u_{yyy}(i).(\frac{1}{24}.\Delta y^2) \right) \\ = \; & mes(K_1 \cup K_2) \left(u_{yxx}(i)\frac{1}{12}(\Delta x)^2 \; + \; u_{yyy}(i)\frac{1}{24}(\Delta y)^2 \right) \end{aligned} \tag{A.30}$$

Calcul de ces termes sur les triangles centraux
Cas du triangle K_5(haut)

$$\int_K i\vec{X} \; = \; mes(K).i\vec{G} \; = \; mes(K)(0 \; , \; \frac{2\Delta y}{3}) \tag{A.31}$$

On a $i\vec{A} = (\frac{\Delta x}{2},\ \Delta y)$, $i\vec{B} = (-\frac{\Delta x}{2}\ ,\ \Delta y)$, , $i\vec{C} = (\ 0,\ 0)$.

$$
\begin{aligned}
& u_{XX}(i).e_2\int_K (\ i\vec{X} \ - \ \frac{2i\vec{C}^2 \ + \ i\vec{A}^2 \ + \ i\vec{B}^2}{2\Delta y}) && \text{(A.32)}\\
= \quad & mes(K).(u_{xx}(i)(\frac{\frac{(-\Delta x)^2}{4} \ + \ \frac{(\Delta x)^2}{4}}{4.\Delta y}) \ + \ u_{xy}(i).0 && \text{(A.33)}\\
+ \quad & u_{yy}(i).(\frac{2\Delta y}{3} \ - \ \frac{2.(\Delta y)^2}{2.\Delta y}))\\
= \quad & mes(K).(u_{xx}(i)(\frac{1}{8}.\frac{(\Delta x)^2}{\Delta y} \ - \ \frac{1}{3}\ .\Delta y))
\end{aligned}
$$

Cas du triangle K_6(bas)

$$\int_K i\vec{X} \ = \ mes(K).i\vec{G} \ = \ mes(K)(0\ ,\ -\frac{2\Delta y}{3}) \tag{A.34}$$

On a $i\vec{A} = (-\frac{\Delta x}{2},\ -\Delta y)$, $i\vec{B} = (\frac{\Delta x}{2}\ ,\ -\Delta y)$, , $i\vec{C} = (\ 0,\ 0)$.

$$
\begin{aligned}
& u_{XX}(i).e_2\int_K (\ i\vec{X} \ - \ \frac{2i\vec{C}^2 \ + \ i\vec{A}^2 \ + \ i\vec{B}^2}{2\Delta y}) && \text{(A.35)}\\
= \quad & mes(K).(u_{xx}(i)(\frac{\frac{(-\Delta x)^2}{4} \ + \ \frac{(\Delta x)^2}{4}}{4.\Delta y}) \ + \ u_{xy}(i).0 && \text{(A.36)}\\
+ \quad & u_{yy}(i).(-\frac{2\Delta y}{3} \ + \ \frac{2.(\Delta y)^2}{2.\Delta y}))\\
= \quad & mes(K).(u_{xx}(i)(-\frac{1}{8}.\frac{(\Delta x)^2}{\Delta y} \ + \ \frac{1}{3}\ .\Delta y))
\end{aligned}
$$

On en déduit que

$$u_{XX}(i).e_2\int_{K_5\cup K_6} (i\vec{X} \ - \ \frac{2i\vec{C}^2 \ + \ i\vec{A}^2 \ + \ i\vec{B}^2}{4\Delta y}) \quad = \quad (0,0) \tag{A.37}$$

Donc

$$\int_{Supp\Phi_i} \frac{\partial}{\partial y}u(X) \ - \ \frac{\partial}{\partial y}\Pi_h u \quad = \quad u_{XXX}(i).e_2\int_{Supp\Phi_i} (\ \frac{1}{2}i\vec{X}^2 \ - \ \frac{1}{6}.\frac{(2i\vec{C}^3 \ + \ i\vec{A}^3 \ + \ i\vec{B}^3)}{2\Delta y})dK$$

On a

$$u_{XXX}(i).e_2\int_{K_5\cup K_6} (\ \frac{1}{2}i\vec{X}^2)\ dK \tag{A.38}$$

$$\begin{aligned}
&= u_{yxx}(i)\int_{K_5\cup K_6}(\;\frac{1}{2}x^2)\;dK\;+\;u_{yxy}(i)\int_{K_5\cup K_6}\frac{1}{2}xydK\;+\;u_{yyy}(i)\int_{K_5\cup K_6}\frac{1}{2}y^2dK\\
&= u_{yxx}(i)mes(K_5\cup K_6)\frac{1}{12}(\Delta x)^2\;+\;0\;+\;u_{yyy}(i)mes(K_5\cup K_6)\frac{1}{12}(\Delta y)^2
\end{aligned}$$

D'autre part

$$\begin{aligned}
&u_{XXX}(i).(2i\vec{C}^3\;-\;i\vec{A}^3\;-\;i\vec{B}^3) \qquad (A.39)\\
&= u_{xxx}(i).(\frac{\Delta x^3}{8}-\frac{\Delta x^3}{8})\;+\;u_{xxy}(i).(0)\;+\;u_{xyy}(i).(0)\;+\;u_{yyy}(i).(\frac{1}{2}.\Delta y^3)\\
&= u_{yyy}(i).(\frac{1}{2}.\Delta y^3)
\end{aligned}$$

Donc

$$\frac{1}{6}.\frac{(2i\vec{C}^3\;+\;i\vec{A}^3\;+\;i\vec{B}^3)}{2\Delta y}\;=\;u_{yyy}(i).(\frac{1}{24}.\Delta y^2) \qquad (A.40)$$

On en déduit que

$$\begin{aligned}
&u_{XXX}(i).e_2\int_{K_5\cup K_6}(\;\frac{1}{2}i\vec{X}^2\;-\;\frac{1}{6}.\frac{(2i\vec{C}^3\;+\;i\vec{A}^3\;+\;i\vec{B}^3)}{2\Delta y})\;dK \qquad (A.41)\\
&= mes(K_5\cup K_6)\left(u_{yxx}(i)\frac{1}{12}(\Delta x)^2\;+\;u_{yyy}(i)\frac{1}{12}(\Delta y)^2\;-\;u_{yyy}(i).(\frac{1}{24}.\Delta y^2)\right)\\
&= mes(K_5\cup K_6)\left(u_{yxx}(i)\frac{1}{12}(\Delta x)^2\;+\;u_{yyy}(i)\frac{1}{24}(\Delta y)^2\right)
\end{aligned}$$

Conclusion
On en déduit que

$$\begin{aligned}
&\int_{Supp\Phi_i}\frac{\partial}{\partial y}u(X)\;-\;\frac{\partial}{\partial y}\Pi_h u \qquad (A.42)\\
&= mes(Supp\Phi_i)\left(u_{yxx}(i)\frac{1}{12}(\Delta x)^2\;+\;u_{yyy}(i)\frac{1}{24}(\Delta y)^2\right)
\end{aligned}$$

On en conclut que

$$\int_{cellule}\frac{\partial}{\partial y}\phi_i\left(\frac{\partial}{\partial y}u\;-\;\frac{\partial}{\partial y}\Pi_h u\right)\;dK \quad = \quad mes(Supp\Phi_i)\left(u_{yyxx}(i)\frac{1}{12}(\Delta x)^2\;+\;u_{yyyy}(i)\frac{1}{24}(\Delta y)^2\right)$$

Bibliographie

[Alonso *et al.*, 2002] J. Alonso, I. Kroo, and A. Jameson. Advanced algorithms for design and optimization of quiet supersonic platforms. *AIAA Paper*, 2002-0144 :1–13, 2002.

[Angrand *et al.*, 1980] F. Angrand, R Glowinski, J. Periaux, P. Perrier, O. Pironneau, and G. Poirier. Optimum design for potential flow. In *Finite elements in flows problems*, Calgary, 1980.

[Arian and Ta'asan, 1999] E. Arian and S. Ta'asan. Analysis of the Hessian for aerodynamic optimization : Inviscid flow. *Comput. Fluids*, 28(7) :853–877, 1999.

[Attouch *et al.*, 2000] H. Attouch, X. Goudou, and P. Redont. The heavy ball with friction method. I : The continuous dynamical system : Global exploration of the local minima of a real-valued function by asymptotic analysis of a dissipative dynamical system. *Commun. Contemp. Math.*, 2(1) :1–34, 2000.

[Babuska and Rheinboldt, 1978] I. Babuska and W.C. Rheinboldt. A posteriori error estimates for finite element method. *Int. J. Numer. Meth. Eng.*, (12) :1597–1615, 1978.

[Bank and Dupont, 1980] R.E. Bank and T. Dupont. Analysis of a two-level scheme for solving finite element equations. Technical report, Technical Report CNA-159,Center for Numerical Analysis,University of Texas at Austin, 1980.

[Becker and Rannacher, 1996] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods : basic analysis and examples. *East-West J. Numer. Math.*, 4 :237–264, 1996.

[Becker *et al.*, 1999] R. Becker, M. Braack, and R. Rannacher. Numerical simulation of laminar flames at low mach number with adaptative finite elements. *Combustion Theory and Modelling*, 3 :503–534, 1999.

[Beux and Dervieux, 1992] F. Beux and A. Dervieux. Exact-gradient shape optimization of a 2d Euler flow. *Finite Elements in Analysis and Design*, 12 :281–302, 1992.

[Beux, 1993] F. Beux. *Conception optimale de formes aeérodynamiques et méthodes d'approximations décentrées pour des écoulements incompressibles.* PhD thesis, Université de Nice, France, 1993. (In French).

[Beux, 1994] F. Beux. Shape optimization of an Euler flow in a nozzle. *Notes on numerical fluid mechanics*, 55 :115–131, 1994.

[Bonnans *et al.*, 2002] J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Numerical Optimization – Theoretical and Practical Aspects.* Springer Verlag, Berlin, 2002.

[Borouchaki and George, 1996a] H. Borouchaki and P.L. George. Maillage de surfaces paramétriques. partie I : Aspects théoriques. Technical Report 2928, INRIA Rocquencourt, 1996.

[Borouchaki and George, 1996b] H. Borouchaki and P.L. George. Maillage de surfaces paramétriques. partie II : Exemples d'applications. Technical Report 2944, INRIA Rocquencourt, 1996.

[Bramble *et al.*, 1990] J.H. Bramble, J.E. Pasciak, and J. Xu. Parallel multilevel preconditionners. *Math. Comp.*, 55 :1–22, 1990.

[Bristeau *et al.*, 1979] M.-O. Bristeau, R. Glowinski, J. Periaux, P. Perrier, and O. Pironneau. On the numerical solution of nonlinear problems in fluid dynamics by least squares and finit element methods (I). least squares formulation and conjugate gradient solutions of the continuous problems. *Comput. Meths. Appl. Mech. Engrg.*, (17/18) :619–657, 1979.

[Byrd *et al.*, 2000] R.H. Byrd, J.Ch. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89 :149–185, 2000.

[Byrd, 1987] R.H. Byrd. Robust trust region methods for constrained optimization. Third SIAM Conference on Optimization, Houston, TX, mai 1987.

[Cabello *et al.*, 1991] J. Cabello, R. Lohner, and O.-P. Jacquotte. A variational method for the optimization of directionally stretched elements generated by the advancing front method (afm). In A.S. Arcilla, J. Hauser, P.R. Eiseman, and J.F. Thompson, editors, *Numerical Grid Generation in Computational Field Simulation and Related Fields, Proceedings of the 3rd International Grid Conference*, page 521. North Holland, 1991.

[Castro-Diaz *et al.*, 1996] M.J. Castro-Diaz, F. Hecht, B. Mohammadi, and P.-L. George. *Anisotropic adaptative mesh generation in two dimensions for CFD*, pages 181–192. Computationals Fluids Dynamic '96, 1996.

[Cea, 1981] J. Cea. Numerical methods of shape optimal design. In *Optimization of Distributed Parameter Structure*, 1981.

[Celis *et al.*, 1985] M.R. Celis, J.E. Dennis, and R.A. Tapia. A trust region strategy for nonlinear equality constrained optimization. In P.T. Boggs, R.H. Byrd, and R.B. Schnabel, editors, *Numerical Optimization 1984*, pages 71–82. SIAM Publication, Philadelphia, 1985.

[Clément, 1975] P. Clément. Approximation by finite element functions using local regularization. *RAIRO Analyse Numèr.*, 9 :77–84, 1975.

[Cliff and Shenoy,] E. Cliff and A. Shenoy. On the optimality system for the 1-d euler flow problem.

[Cohen, 2000] A. Cohen. Wavelet methods in numerical analysis. In P.G. Ciarlet and J.L. Lions, editors, *Handbook of Numerical Analysis*, volume VII, pages 417–713. Elsevier Science, 2000.

[Corliss *et al.*, 2001] G. Corliss, C. Faure, A. Griewank, L. Hascoet, and U. Naumann(editors). *Automatic Differentiation of Algorithms, from Simulation to Optimization*. Springer, 2001. Selected proceedings of AD2000, Nice, France.

[Coudiere *et al.*, 2002] Y. Coudiere, B. Palmerio, A. Dervieux, and D. Leservoisier. Accuracy barriers in mesh adaptation. Technical Report 4528, INRIA Sophia-Antipolis, 2002.

[Courty and Dervieux, 2003] F. Courty and A. Dervieux. A SQP-like one-shot algorithm to optimal control problems. *submitted to Mathematical Programming*, 2003.

[Dadone and Grossman, 2000] A. Dadone and B. Grossman. Progressive optimization of inverse fluid dynamic design problems. *Computer and Fluids*, 29 :1–32, 2000.

[Dadone *et al.*, 1999] A. Dadone, B. Mohammadi, and N. Petruzzelli. Incomplete sensitivities and bfgs methods for 3d aerodynamic shape design. Technical Report 3633, INRIA, 1999.

[Dervieux and Palmerio, 1975] A. Dervieux and B. Palmerio. Une formule de Hadamard dans des problèmes d'identification de domaines ; exemples. *C. R. Acad. Sc. Paris*, 280, Serie A :1761–1764, 1975. (in French).

[Dervieux *et al.*, 2001] A. Dervieux, D. Leservoisier, P.-L. George, and Y. Coudière. About theoretical and practical impact of mesh adaptions on approximation of functions and of solution of pde. In *Conférence invitée à ECCOMAS-Swansea.* to appear in I.J.M.N.F., 2001.

[Dervieux, 1981] A. Dervieux. Résolution de problèmes à frontière libre. *Thèse, université de Paris VI*, 1981.

[Desideri and Hemker, 1995] J.-A. Desideri and P.W. Hemker. Convergence analysis of iterative implicit and defect-correction algorithms for hyperbolic problems. *SIAM J. Sci. Comput.*, pages 88–118, 1995.

[Farhat *et al.*, 2002] C. Farhat, K. Maute, B. Argrow, and M. Nikbay. A shape optimization methodology for reducing the sonic boom initial pressure rise. *AIAA Paper*, 2002-0145 :1–11, 2002.

[Fezoui and Stoufflet, 1989] L. Fezoui and B. Stoufflet. A class of implicit upwind schemes for euler simulations with unstructured meshes. *J. Comput. Phys.*, 84(1) :174–206, 1989.

[Formaggia and Perotto, 2002] L. Formaggia and S. Perotto. Anisotropic error estimates for elliptic problems. *Numer. Math*, 2002.

[Fortin (Ed.), 2000] M. Fortin (Ed.). Estimations a posteriori et adaptation de maillages (partly in french). *Special issue of the Revue Europeenne des Elements Finis*, 2000.

[Francescatto and Dervieux, 1998] J. Francescatto and A. Dervieux. A semi-coarsening strategy for unstructured multigrid based on agglomeration. *International Journal for Numerical Methods in Fluids*, 26 :927–957, 1998.

[Frey and George, 2000] P-J Frey and P-L. George. *Mesh generation.* Hermes, 2000.

[Frey, 1993] P. Frey. Génération de maillages 3d dans des ensembles discrets. application biomédicale aux méthodes d'éléments finis. Master's thesis, University of Strasbourg I, 1993.

[Garabedian, 1964] P.R. Garabedian. *Partial Differential Equations.* Wiley, 1964.

[Giering, 1997] R. Giering. Tangent linear and adjoint model compiler, users manual. Technical report, 1997. http ://www.autodiff.com/tamc.

[Gilbert, 1992] J.C. Gilbert. Automatic differentiation and iterative processes. *Optimization Methods and Software*, 1 :13–21, 1992.

[Gilbert, 1997] J.Ch. Gilbert. On the realization of the Wolfe conditions in reduced quasi-Newton methods for equality constrained optimization. *SIAM Journal on Optimization*, 7 :780–813, 1997.

[Giles and Suli, 2002] M. Giles and E. Suli. Adjoint methods for PDEs : a posteriori error analysis and postprocessing by duality. *Acta Numerica*, pages 145–236, 2002.

[Giles, 2001] M.-B. Giles. Adjoint methods for aeronautical design. In *Proceedings of the ECCOMAS CFD Conference*, 2001.

[Griewank and Faure, 2002] A. Griewank and Ch. Faure. Reduced Gradients and Hessians from Fixed Point Iteration for State Equations. *Numerical Algorithms*, 30(2) :113–139, 2002.

[Griewank, 2000] A. Griewank. *Evaluating Derivatives : Principles and Techniques of Algorithmic Differentiation.* SIAM, Frontiers in Applied Mathematics, 2000.

[Habashi *et al.*, 2000] W.G. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, and M.-G. Vallet. Anisotropic mesh adaptation : Towards user-independent, mesh-independent and solver-independent cfd solutions : Part I : General principles. *International Journal for Numerical Methods in Fluids*, 32 :725–744, 2000.

[Hackbusch, 1985] W. Hackbusch. *Multi-Grid Methods and Applications.* Springer-Verlag, Berlin-Heidelberg-New-York, 1985.

[Hadamard, 1968] J. Hadamard. *Mémoires sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées (1908).* CNRS, 1968.

[Hascoet *et al.*, 2001] L. Hascoet, S. Fidanova, and C. Held. Adjoining independent computations. *in [Corliss* et al.*, 2001]*, pages 185–190, 2001.

[Hascoet, 2001] L. Hascoet. The data-dependence graph of adjoint programs. research report 4167, INRIA, 2001.

[Held and Dervieux, 2002] C. Held and A. Dervieux. One shot airfoil optimisation without adjoint. *Computer and Fluids*, 31 :1015–1049, 2002.

[Hovland *et al.*, 1997] P. Hovland, B. Mohammadi, and C. Bischof. Automatic differentiation of navier-stokes computations. Technical Report MCS-P687-0997, Argonne National Laboratory, 1997.

[Iollo and Salas,] A. Iollo and M.D. Salas. Entropy jump across an inviscid shock wave. Technical report.

[Iollo *et al.*, 1993] A. Iollo, M.D. Salas, and S. Ta'asan. Shape optimization governed by the euler equations using an adjoint method. Technical report, ICASE, 1993. report 93-98.

[Iollo *et al.*, 1995] A. Iollo, G. Kuruvila, and S. Ta'asan. Pseudo-time method for optimal shape design using Euler equations. Report 95-59, NASA contractor report 198205, ICASE, 1995.

[Jameson, 1988a] A. Jameson. Aerodynamic design via control theory. Report 1824 MAE, Princeton University, New Jersey, 1988.

[Jameson, 1988b] A. Jameson. Aerodynamic design via control theory. *SIAM Journal on Scientific Computing*, 3(3) :233–261, 1988.

[Jameson, 1991] A. Jameson. Aerodynamic design via control theory. Report 91-2, NASA contractor report 187497, ICASE, 1991.

[J.Xu, 1989] J.Xu. *Theory of multilevel methods.* PhD thesis, Cornell university, 1989.

[Kunert, 2000] G. Kunert. An a posteriori residual error estimator for the finite element method on anisotropic meshes. *Numerische Mathematik*, 2000.

[Kunoth, 1994] A. Kunoth. *Multilevel Preconditioning.* PhD thesis, University of Berlin, 1994.

[Lalee *et al.*, 1998] M. Lalee, J. Nocedal, and T. Plantenga. On the implementation of an algorithm for large-scale equality constrained optimization. *SIAM Journal on Optimization*, 8 :682–706, 1998.

[Lallemand *et al.*, 1992] M.-H. Lallemand, H. Steve, and A. Dervieux. Unstructured multiggridding by volume agglomeration : current status. *Computer and Fluids*, 21(3) :397–433, 1992.

[Leservoisier, 2001] D. Leservoisier. *Strategies d'adaptation et de raffinement de maillages en Mecanique des fluides.* PhD thesis, Université Pierre et Marie Curie, 2001.

[Maglieri and Plotkin, 1991] D.J. Maglieri and K.J. Plotkin. *Aeroacoustics of flight vehicles : theory and practice.* Acoustical Society of America, Publications, 1991.

[Marco and Dervieux, 1995] N. Marco and A. Dervieux. Numerical optimizers for aerodynamic design using transonic finite-element solvers. Final report, BRITE-ECARP, 1995.

[Marquant *et al.*, 2000] G. Marquant, S Pateux, and C. Labit. Mesh-based scalable image coding with rate-distortion. In *European Signal Processing Conference EUSIPCO 2000*, 2000.

[Mohammadi and Pironneau, 2001] B. Mohammadi and O. Pironneau. *Applied shape optimization for fluids.* Clarendon Press - Oxford, 2001.

[Mohammadi, 1997] B. Mohammadi. Practical application to fluid flows of automatic differentiation for design problems. *Von Karman Lecture Series*, 1997.

[Mohammadi, 2002] B. Mohammadi. Optimization of aerodynamic and acoustic performances of supersonic civil transports. In *Center for Turbulence Research. Proceedings of the Summer Program 2002*, 2002.

[Murat and Simon, 1974] F. Murat and J. Simon. Quelques résultats sur le contrôle par un domaine géométrique. *Publications du Laboratoire d'Analyse Numérique, university of Paris VI*, 1974.

[N. Marco and Dervieux, 1997] B. Koobus N. Marco and A. Dervieux. An additive multilevel preconditioning method. *Journal of Scientific Computing*, 12(3) :233–251, 1997.

[Nadarajah *et al.*, 2002] S. Nadarajah, A. Jameson, and J. Alonso. An adjoint method for the calculation of remote sensitivities in supersonic flow. *AIAA Paper*, 2002-0261 :1–15, 2002.

[Nocedal and Wright, 1999] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, 1999.

[Omojokun, 1991] E.O. Omojokun. *Trust region algorithms for optimization with nonlinear equality and inequality constraints*. PhD thesis, Department of Computer Science, University of Colorado, Boulder, Colorado 80309, 1991.

[Palmerio and Dervieux, 1996] B. Palmerio and A. Dervieux. Multimesh and multiresolution analysis for mesh adaptive interpolation. *Applied Numerical Mathematics*, 22 :477–493, 1996.

[Palmerio, 1996] B. Palmerio. Coupling mesh and flow in viscous fluid calculations on unstructured triangular finite element. *Computational Fluid Dynamics*, 6 :275–290, 1996.

[Pironneau, 1976] O. Pironneau. *Sur les problèmes d'optimisation de structure en mécanique des fluides*. PhD thesis, Université de Paris 6, 1976.

[Pironneau, 1984] O. Pironneau. *Optimal shape design for elliptic systems*. Springer-Verlag, 1984.

[Reuther and Jameson, 1995] J. Reuther and A. Jameson. Aerodynamic Shape Optimization of Wing and Wing-Body Configurations Using Control Theory. AIAA Paper 95-0123, 1995. 33rd Aerospace Sciences Meeting and Exhibit.

[Saad and Schultz, 1986] Y. Saad and M.H. Schultz. Gmres : A generalized minimal residual algorithm for solving nonsymmetric linear systems. *J. Sci. Comput.*, 7 :856–869, 1986.

[Schiffer, 1946] M. Schiffer. Hadamard's formula and variation of domain functions. *American Journal of Mathematics*, LXVIII :417–448, 1946.

[Seebas and Argrow, 1998] R. Seebas and B. Argrow. Sonic boom minimization revisited. *AIAA Paper*, 98-2956 :1–13, 1998.

[Sevin, 1999] C. Sevin. *Optimisation de formes en mécanique des fluides numérique*. PhD thesis, Université Pierre et Marie Curie, 1999.

[Steve, 1988] H. Steve. *Schémas implicites linéaires décentrés pour la résolution des équations d'Euler en plusieurs dimensions*. PhD thesis, Université de Provence Aix-Marseille 1, 1988.

[Stoufflet, 1984] B. Stoufflet. *Résolution numérique des équations d'Euler des fluides parfaits compressibles par des schémas implicites en éléments finis*. PhD thesis, Paris 6, 1984.

[Ta'asan *et al.*, 1992] S. Ta'asan, G. Kuruvila, and M.D. Salas. Aerodynamic design and optimization in one shot. In *30th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, AIAA Paper 91-0025*, 1992.

[Ta'asan, 1991] S. Ta'asan. One shot methods for optimal control of distributed parameter systems i : finite dimensional control. ICASE, Technical Report 91-2, Nasa contractor report, 1991.

[Tie and Aubry, 1999] B. Tie and D. Aubry. Adaptative FE strategy for non-linear and coupled structural computation. In *Numerical Methods in Engineering '96, Proceeding of the Second ECCOMAS Conference on Numerical Methods in Engineering, 9-13 September 1996*, pages 516–522. John Wiley, 1999.

[Tropics, 2001] INRIA Tropics. On-line documentation of the tapenade ad tool. Technical report, 2001. http ://www.inria.fr/tropics.

[Van Leer, 1975] B. Van Leer. Towards the ultimate conservative difference scheme v : a second-order sequel to godunov's method. *Journal of Computational Physics*, 14 :159–179, 1975.

[Van Leer, 1982] B. Van Leer. Flux vector splitting for the euler equations. *Lecture Notes in Physics*, 170 :405–512, 1982.

[Vardi, 1985] A. Vardi. A trust region algorithm for equality constrained minimization : convergence properties and implementation. *SIAM Journal on Numerical Analysis*, 22 :575–591, 1985.

[Vazquez *et al.*, 2002] M. Vazquez, A. Dervieux, and B. Koobus. Aerodynamical and sonic boom optimization of a supersonic aircraft. INRIA Report 4520, 2002.

[Vazquez *et al.*, to appear] M. Vazquez, A. Dervieux, and B. Koobus. Multilevel optimization of a supersonic aircraft. to appear.

[Venditi and Darmofal, 2003] D.A. Venditi and D.L. Darmofal. Anisotropic grid adaptation for functional outputs : application to two-dimensional viscous flows. *J. Comput. Phys.*, 187 :22–46, 2003.

[Verfurth, 1994] R. Verfurth. A posteriori error estimates for nonlinear problems. finite element discretization of elliptic problems. *Math. of Comp.*, 62(206) :445–475, 1994.

[Xu, 1997] J. Xu. An introduction to multilevel methods. In M. (ed.) et al. Ainsworth, editor, *Wavelets, multilevel methods and elliptic PDEs. 7th EPSRC numerical analysis summer school, University of Leicester, Leicester, GB, July 8–19, 1996.*, pages 213–302. Oxford : Clarendon Press. Numerical Mathematics and Scientific Computation., 1997.

[Yserentant, 1986] H. Yserentant. On the multi-level splitting of finite element spaces. *Numer. Math.*, 49 :379–412, 1986.

[Zolesio, 1979] J.-P. Zolesio. *Identification de domaines par déformations.* PhD thesis, Université de Nice, 1979.

Résumé

Notre contribution concerne les trois domaines complémentaires suivants : la différentiation automatique de programmes, l'optimisation de formes pour de grands systèmes, l'adaptation de maillages. Dans le chapitre 1 de la partie 1, nous exposons une méthode de calcul de gradients par Différentiation Automatique pour un problème classique d'optimisation de formes. Nous expliquons comment déduire un gradient exact basé sur un état adjoint sans stocker explicitement le jacobien. Le mode adjoint de la DA que nous proposons utilise beaucoup moins d'espace mémoire. Dans le chapitre 2 de la partie 2, nous proposons une méthode de type SQP pour résoudre une classe de problèmes d'optimisation avec contraintes égalités. Le nouvel algorithme permet une résolution simultanée du système d'optimalité. Cette méthode one shot combine efficacité et robustesse. Dans le chapitre 3 de la partie 2, nous étudions une nouvelle stratégie de préconditionnement pour l'optimisation de formes. Nous construisons un préconditionnement multiniveau additif à partir du principe classique de Bramble-Pasciak-Xu et du principe d'agglomération. Nous spécifions aisément le gain en régularité de notre préconditionneur avec un seul paramètre réel. Dans le chapitre 1 de la partie 3, nous étudions le problème du meilleur maillage adapté pour de l'interpolation pure. La résolution du système d'optimalité donne une expression complètement explicite de la métrique optimale en fonction de la fonction à adapter. Dans le chapitre 2 de la partie 3, nous étendons la méthode du chapitre précédent au problème de l'adaptation de maillage pour EDP. Notre méthode repose sur une analyse *a priori* rigoureuse puis sur une modélisation.

Mots-clés: optimisation de formes, différentiation automatique, one shot, Quasi-Newton, multiniveau, aérodynamique, adaptation de maillages, métrique

Abstract

Our contribution concerns the following three complementary domains : Automatic Differentiation, optimal shape design for large systems, mesh adaption. In the chapter 1 of the part 1, we expose a method to compute gradients using Automatic Differentiation for a classical optimal shape design problem. We explain how to deduce an exact gradient based on an adjoint state without storing explicitly the Jacobian matrix. In the chapter 2 of the part 2, we propose a SQP-like method to solve a class of optimization problems with equality constraints. The new algorithm enables to solve simultaneously the optimality system. In the chapter 3 of the part 2, we study a new preconditioning strategy for optimal shape design. We build an additive multilevel preconditioning starting from the classical Bramble-Pasciak-Xu principle and from the agglomeration principle. In the chapter 1 of the part 3, we study the problem of the best adapted mesh for a pure interpolation problem. The optimality system solution gives a completely explicite expression of the optimal metric as a function of the function to adapt. In the chapter 2 of the part 3, we extend the method of the previous chapter to the problem of mesh adaption for P.D.E. We obtain an optimal control formulation with an adjoint state.

Keywords: optimal shape design, automatic differentiation, one shot, Quasi-Newton, multilevel, aerodynamic, mesh adaption, metric