HAL
open science

Algorithmes multigrilles adaptatifs et scalables

Gautier Brethes

To cite this version:

Gautier Brethes. Algorithmes multigrilles adaptatifs et scalables. Equations aux dérivées partielles [math.AP]. Université Nice Sophia Antipolis, 2015. Français. NNT : . tel-01256185v1

HAL Id: tel-01256185

https://inria.hal.science/tel-01256185v1

Submitted on 14 Jan 2016 (v1), last revised 8 Mar 2016 (v2)

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

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

UNIVERSITÉ DE NICE - SOPHIA ANTIPOLIS - UFR Sciences

ÉCOLE DOCTORALE SFA
SCIENCES FONDAMENTALES ET APPLICATIONS

T H È S E

pour obtenir le titre de

Docteur en Sciences

de l'Université Nice Sophia Antipolis
Discipline : Mathématiques

Présentée et soutenue par

Gautier Brèthes

Algorithmes multigrilles adaptatifs et scalables

Thèse dirigée par Alain Dervieux

préparée à l'INRIA Sophia Antipolis, Projet Ecuador

soutenue le 8 décembre 2015

Jury :

Paul-Louis George	Directeur de Recherche INRIA	Rapporteur
Bruno Koobus	Professeur à l'université de Montpellier 2	Rapporteur
Boniface Nkonga	Professeur à l'université de Nice	Examinateur
Thierry Coupez	Professeur à l'Ecole Centrale de Nantes	Examinateur
Elie Hachem	Professeur associé à l'Ecole des Mines de Paris	Examinateur
Alain Dervieux	Directeur de Recherche INRIA émérite	Directeur de thèse

Cette thèse a été soutenue par l'entreprise LEMMA. Les droits d'utilisation de son contenu à des fins commerciales sont donc réservés.

Remerciements

Avant tout, je remercie mon directeur de thèse Alain Dervieux à qui je dois d'avoir pu faire cette thèse et qui, durant trois ans, m'a encadré patiemment, me fournissant les outils qui me manquaient pour avancer lorsque les choses devenaient difficiles.

Je remercie les deux organismes financeurs de ma thèse c'est-à-dire la région Provence-Alpes-Côte-d'Azur et la société LEMMA dont je souhaite remercier tous les membres et plus particulièrement Charles Leca, Olivier Allain et Vincent Levasseur.

Je remercie mes deux rapporteurs de thèses, Paul-Louis George et Bruno Koobus pour avoir pris le temps de se pencher sur mes travaux.

Je remercie Boniface Nkonga, Thierry Coupez et Elie Hachem pour avoir accepté de faire partie de mon jury.

Je remercie chacun des membres de l'équipe ECUADOR d'INRIA qui ont supporté ma présence pour certains durant trois ans.

Je remercie tous ceux avec qui j'ai collaboré durant cette thèse, notamment Frédéric Alauzet et Adrien Loseille.

Enfin, je remercie mes parents et ma sœur qui m'ont soutenu de loin pendant que je vivais cette aventure.

"À la mémoire de Radii Petrovich Fedorenko, inventeur des multigrilles."

Contenu

Préface **1**

1 Représentation de maillage **9**
1.1 Introduction . . . 9
1.2 Métrique . . . 9
1.3 Maillage unité . . . 10
1.3.1 Elément unité . . . 11
1.3.2 Elément quasi-unité et maillage unité . . . 12
1.3.3 Nombre de sommets . . . 12
1.4 Erreurs d'interpolation P1 . . . 13
1.4.1 Modélisation de l'erreur d'interpolation . . . 13
1.4.2 Optimisation de l'erreur d'interpolation continue . . . 14
Conclusion . . . 15

2 Modèle numérique, estimateurs et correcteurs **17**
2.1 Introduction . . . 17
2.2 Approximation P1 du problème de Poisson . . . 17
2.3 Rappels: estimations d'interpolation . . . 18
2.4 Estimations *a priori* . . . 18
2.5 Estimateurs *a posteriori* de Becker-Rannacher . . . 19
2.6 Correcteurs . . . 21
2.6.1 Correcteurs pour des fonctionnelles . . . 21
2.6.2 Champ correcteur basé Defect-Correction . . . 22
Conclusion . . . 25

3 Méthode de résolution multigrille avec critère d'arrêt **27**
3.1 Introduction . . . 27
3.2 MG . . . 28
3.2.1 Algorithme bi-grille . . . 28
3.2.2 Algorithme multigrille . . . 30
3.2.3 Critère de convergence des méthodes multigrilles . . . 31
3.2.4 Propriété de lissage . . . 31
3.2.5 Propriété d'approximation . . . 31
3.2.6 Convergence . . . 31
3.2.7 Complexité des méthodes multigrilles . . . 32
3.3 FMG . . . 32
3.3.1 Description de la méthode Full-Multigrid . . . 32
3.3.2 Propriétés de la méthode Full-Multigrid . . . 34
3.3.3 Complexité de la méthode Full-Multigrid . . . 35
3.3.4 Synthèse sur la complexité FMG . . . 36

3.4 Maillages non-emboités . . . 36
3.5 Un nouveau critère d'arrêt . . . 37
3.5.1 Estimation de l'erreur d'approximation . . . 38
3.5.2 Erreur itérative contrôlée FMG . . . 41
3.5.3 Application de la méthode FMG proposée . . . 43
Conclusion . . . 44

4 FMGA basé-hessien 49
4.1 Avant-propos . . . 49
4.2 Introduction . . . 49
4.3 Mesh parametrization and mesh adaptation loop . . . 51
4.3.1 Continuous mesh model . . . 51
4.3.2 Fixed-point mesh adaptation . . . 52
4.4 The MG Anisotropic fixed-point . . . 53
4.4.1 Generating the different grids . . . 53
4.4.2 Global fixed point . . . 55
4.5 The FMG algorithm . . . 55
4.5.1 Global FMG convergence . . . 55
4.5.2 Approximation error estimate . . . 57
4.5.3 Iterative-Error-controlled FMG . . . 59
4.5.4 Application to the proposed FMG . . . 61
4.6 The FMG Anisotropicly adaptative algorithm . . . 62
4.7 Examples . . . 65
4.7.1 A smooth boundary layer test case . . . 65
4.7.2 A singular internal layer case . . . 67
4.7.3 Circular spot test case . . . 71
4.7.4 Thinner circular test case . . . 74
4.8 Concluding remarks . . . 74
4.9 Acknowledgements . . . 78

5 FMGA orienté-objectif 81
5.1 Introduction . . . 81
5.2 Théorie . . . 81
5.3 Cas tests . . . 84
5.3.1 Cas test circulaire discontinu . . . 84
5.3.2 Cas test circulaire continu avec interface fine . . . 85
5.3.3 Couche limite . . . 87
Conclusion . . . 88

6 FMGA orienté-norme 89
6.1 Avant propos . . . 89
6.2 Abstract . . . 89
6.3 Introduction . . . 89
6.4 Two correctors for the Poisson problem . . . 92

6.4.1 Notations . . . 92
6.4.2 A priori corrector for the PDE solution . . . 93
6.4.3 Finer-grid defect correction corrector for the PDE solution . . 94
6.5 Interpolation error optimization . . . 95
6.5.1 Mesh parametrization . . . 95
6.5.2 Interpolation-based optimal metric . . . 95
6.6 Implicit a priori error estimate . . . 97
6.7 Equation-based adaptation . . . 97
6.7.1 Scalar output "goal-oriented" analysis . . . 97
6.7.2 Norm-based functional . . . 99
6.8 Numerical examples . . . 101
6.8.1 Numerical features . . . 101
6.8.2 2D Boundary layer . . . 102
6.8.3 Bubble-like test case with thick interface . . . 105
6.8.4 Bubble-like test case with thin interface . . . 107
6.8.5 Poisson problem with discontinuous coefficient . . . 111
6.8.6 A 1D Boundary layer . . . 115
6.9 Conclusion . . . 117

7 FMGA orienté-norme tensoriel 119
7.1 Introduction . . . 119
7.2 Poisson problem approximation . . . 121
7.3 Continuous metric parametrization . . . 123
7.3.1 Mesh parametrization . . . 123
7.3.2 Optimal continuous metric . . . 124
7.4 Approximation of metric properties . . . 126
7.4.1 Generic mesh notations . . . 126
7.4.2 Discretizing an arbitrary continuous metric on a background mesh . . . 127
7.5 Second-order error of a metric on a background mesh . . . 128
7.5.1 First example: interpolation error . . . 128
7.5.2 Goal-oriented error . . . 129
7.5.3 Norm-oriented error . . . 132
7.6 Optimal metric . . . 134
7.6.1 Pointwise optimal metric . . . 134
7.6.2 Global optimal metric . . . 135
7.7 Numerical examples . . . 136
7.7.1 A 2D boundary layer test case . . . 136
7.7.2 Poisson problem with discontinuous coefficient [22] . . . 140
7.7.3 Bubble-like test case with thin interface . . . 140
7.7.4 Bubble-like test case with thick interface . . . 140
7.7.5 A 1D boundary layer test case . . . 142
7.8 Conclusion . . . 143

8 Application aux écoulements incompressibles **147**
8.1 Introduction 147
8.2 Modèle Navier-Stokes incompressible instationnaire 148
8.2.1 Équations bi-fluides via Level-Set 148
8.2.2 Approximation par éléments finis 149
8.3 Une idée globale du module adaptatif 151
8.4 Vers MG en instationnaire : FMV 151
8.4.1 Motivations 151
8.4.2 Description de la méthode 152
8.4.3 Exemple avec le cas test du cercle 152
8.5 MG en MPI 156
8.6 Un cas test de validation/évaluation en 3D 157
Conclusion 159

Bibliography **165**

Introduction

Dans toutes sortes de milieux industriels comme l'aéronautique, l'industrie spatiale, l'industrie pétrolière et tant d'autres, il est indispensable d'effectuer des calculs numériques pour simuler des phénomènes intervenant dans des systèmes naturels ou artificiels modélisables par la mécanique des milieux continus.

Parmi les calculs auxquels cette thèse est susceptible de contribuer, figure le calcul modèle illustré par les figures 1 et 2 et mis au point à Lemma durant la thèse de Damien Guègan, [47].

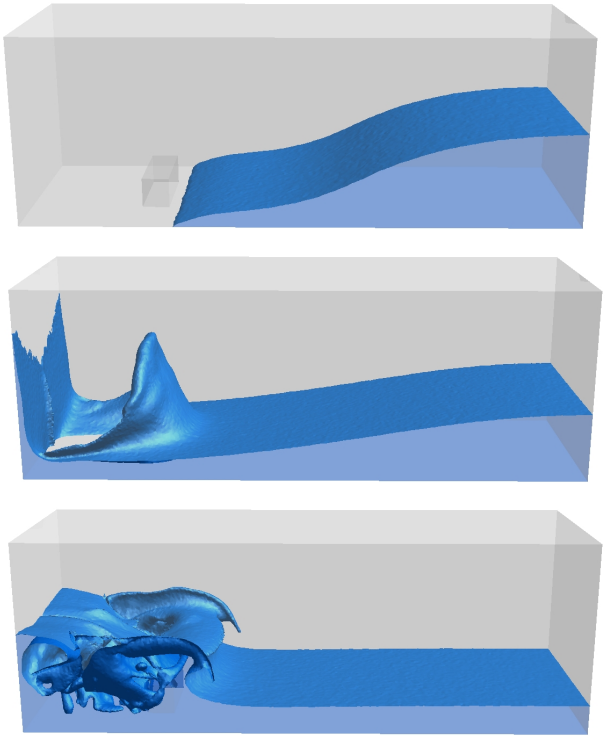

Figure 1: Calcul de l'impact d'une colonne d'eau sur un petit batiment (dans un bassin rectangulaire), visualisation de la surface de la masse d'eau pour trois instants successifs.

Ce calcul est réalisé à l'aide d'un logiciel numérique, ANANAS, développé en utilisant un nombre de plus en plus grand de méthodes mathématiques. Dans le calcul présenté, on utilise de manière assez usuelle un modèle mathématique représentatif de la physique, les équations de Navier-Stokes. Ces équations sont discrétisées par des méthodes d'approximation numérique (des méthodes d'éléments finis en l'occurence). Les équations discrètes sont résolues notamment par un algorithme de résolution linéaire du problème elliptique discret définissant le champ de pression. D'ordinaire, la discrétisation est supportée par un maillage fourni initialement

par l'utilisateur (obtenu avec d'autres outils). Le choix de ce maillage influe sur la qualité du résultat. L'originalité du calcul montré ici est que le logiciel de calcul ANANAS développé par Lemma régénère automatiquement une série de maillages de telle sorte que les maillages effectivement utilisés maximisent la précision du calcul, cf. la figure 2.

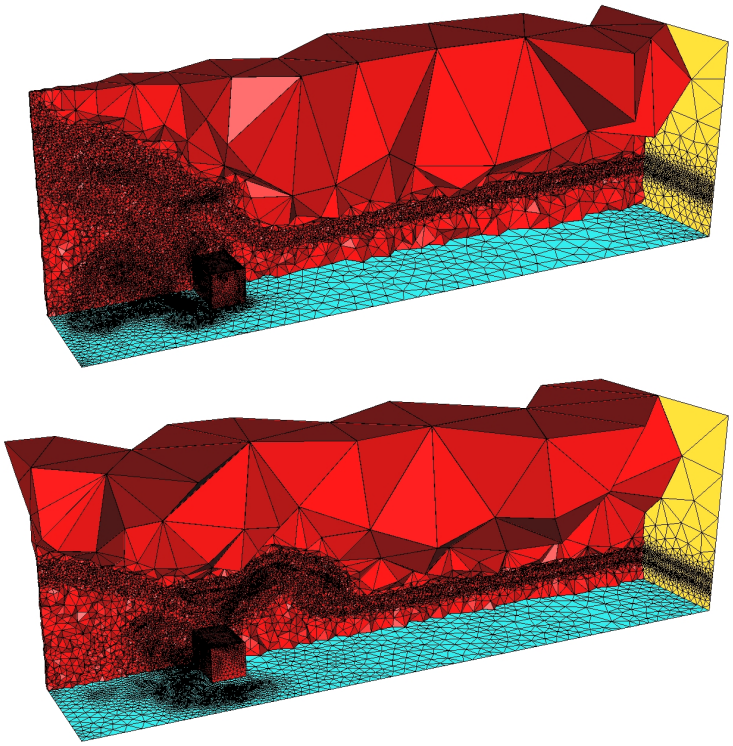

Figure 2: Deux maillages adaptés typiques issus de la collection des maillages engendrés et utilisés successivement par l'algorithme de calcul.

Les apports de la présente thèse vis-à-vis du logiciel ANANAS concerneront deux étapes différentes: d'une part la partie résolution linéaire recevra l'algorithme FMG sous une version parallèle MPI, d'autre part, le critère d'adaptation sera étendu à terme pour prendre en compte les nouveautés des formulations orientées norme. La première étape de transfert est en cours et décrite dans le dernier chapitre de ce mémoire.

La place des simulations dans les activités humaines est désormais si cruciale qu'un des problèmes technologiques essentiels de la décennie est le contrôle de la précision de ces simulations compte tenu des données et modèles employés, ou inversement de l'évaluation des incertitudes liées au processus de simulation. Ce problème est au centre du travail de consortiums tels le projet Européen UMRIDA (Uncertainty Management for Robust Industrial Design in Aeronautics, 21 industriels et instituts, dont l'INRIA). Les travaux de ce type de consortium portent sur des méthodes alimentées par des recherches en analyse numérique telles que

celles réalisées dans le cadre du projet ANR MAIDESC (Maillages adaptatifs pour les interfaces instationnaires avec déformations, étirements, courbures) coordonné par l'INRIA. Ces recherches sont aussi soutenues par des sociétés d'ingénierie et de recherche telles que la société Lemma. Cette thèse s'incrit dans ces deux projets, UMRIDA et MAIDESC.

Plus précisément, nous nous sommes intéressés à la question scientifique suivante:

Comment pour une simulation donnée et des moyens de calcul donnés obtenir la plus grande précision de prédiction?

Cette question relève de deux thèmes de l'analyse numérique. D'une part, pour une taille du vecteur inconnu donnée, nous voulons obtenir la solution avec le moins d'opérations possibles. Il s'agit d'un problème d'algorithmique numérique. D'autre part, pour une taille du vecteur inconnu donnée, nous voulons la meilleure précision. Il s'agit d'approximation numérique, et nous allons examiner des méthodes de maillage adaptatif. Cette thèse touche à ces deux domaines et son but aussi est de faire le lien entre deux techniques de simulation numérique : les méthodes multigrilles et les nouvelles méthodes adaptatives récemment développées.

Les *méthodes multigrilles (MG)* peuvent produire des algorithmes de résolution rapides et efficaces. Elles peuvent être appliquées à une large variété de modèles et d'approximations en calculs mécaniques.

Les multigrilles utilisent de façon intensive les propriétés d'approximation. Une simple itération locale est appliquée sur une grille donnée. La convergence itérative est ensuite accélérée au moyen d'un ensemble de corrections grille grossière. Typiquement, chaque correction est calculée sur un maillage deux fois plus grossier que le maillage donné et le maillage le plus grossier contient seulement quelques dizaines de points. Cette approche produit généralement une convergence itérative qui, exprimée en terme de (logarithme de) la norme du résidu en fonction des itérations, est de pente plus ou moins constante. De plus, cette pente ne dépend pas de la taille du maillage. En particulier, une bonne solution discrète avec N degrés de liberté est obtenue avec un nombre d'opérations délimité par $const.\ N\ Log(N)$, une complexité presque optimale. Enfin, les multigrilles peuvent être combinées avec le raffinement de maillage, produisant l'algorithme Full-Multigrid (FMG): FMG implique n phases, chaque phase j utilisant j maillages, du plus fin au plus grossier. La j-ième phase de FMG résout l'EDP discrétisée sur le j-ième maillage. Cette j-ième phase part d'une interpolation du résultat de la $j-1$-ième phase et fait un certain nombre k_j de cycles multigrilles avec les $j-1$ maillages grossiers disponibles, du premier au $j-1$-ième. Dans un cas idéal, le nombre k_j de cycles dans chaque phase est le même. Notons que FMG a une complexité optimale de $const.\ N$, prédite par la théorie et observée sur beaucoup d'exemples pratiques, voir [76]. On doit cependant noter quelques inconvénients. Tout d'abord, dans beaucoup de cas, la simple itération locale des multigrilles n'est pas suffisante pour traiter des problèmes singuliers ou raides comme des discontinuités ou des couches limites. Pour régler ce problème, on peut utiliser des maillages grossiers définis avec des méthodes plus so-

phistiquées comme les multigrilles algébriques [19][75] ou le déraffinement anisotrope de maillages [66], [63], [39]. On peut aussi utiliser des transferts inter-grilles plus adaptés.

En fait, dans certains cas, la correction de la grille immédiatement plus grossière n'est pas capable de corriger l'itération grille fine ou peut même fonctionner de façon défavorable. En conséquence, le meilleur ensemble de grilles à appliquer pour l'accélération multigrille n'est pas nécessairement le meilleur ensemble de grilles à appliquer pour le procédé Full-Multigrid.

Pour beaucoup d'applications complexes, il a été remarqué que le procédé FMG ne fonctionne pas car le critère d'arrêt habituel produit une solution finale discrète avec une précision détériorée par une résolution itérative insuffisante. Voir, par exemple, [24]. Les deux hypothèses de la théorie FMG sont (1) une convergence MG qui ne dépend pas de la finesse du maillage et (2) la convergence asymptotique d'ordre élevé de la solution discrète vers la solution continue sur les différents maillages de FMG, y compris les grossiers. Si on assume que la convergence du cycle MG est bonne, l'échec du FMG peut être expliqué par le manque de convergence asymptotique vers le continu, soit parce que les maillages sont encore trop grossiers, soit parce que la solution inclut des petits détails ou des singularités.

En conséquence, quand l'algorithme FMG basique, impliquant un nombre fixe de cycles par phase, est appliqué, la solution produite à la fin peut être mauvaise. Le fait d'ajuster le nombre de cycles, si ce n'est pas fait correctement, peut entraîner un coût de calcul beaucoup plus important que celui que la théorie FMG prédit et avoir pour conséquence la perte de la complexité optimale $const.N$.

Les *critères d'arrêt* pour les problèmes itératifs ont été le sujet de beaucoup de travaux publiés. Dans le cas de l'itération quasi-Newton, un travail typique est celui d'Arioli et de ses co-auteurs [61]. Il est communément admis que le meilleur critère consiste à arrêter l'itération quand l'erreur d'itération entre la solution approchée itérée et la solution approchée convergée est plus petite que l'erreur d'approximation entre la solution approchée convergée et la solution exacte [76],[61]. Hélas, le fait de calculer l'erreur d'approximation est coûteux en terme de temps calcul. Dans [61], cette évaluation est remplacée par une hypothèse concernant la convergence en $O(h^2)$. Une proposition de cette thèse est d'utiliser un résidu *a posteriori* comme référence pour l'arrêt des cycles MG.

Les méthodes MG et FMG ont été combinées avec l'adaptation de maillage dans de nombreux travaux. Nous pouvons citer un travail pionnier de R. Bank, [11], et quelques autres plus récents comme [73],[17],[67] ou [64]. Dans ces travaux, l'adaptation est, le plus souvent, basée sur le raffinement de maillage par division locale, produisant des maillages emboîtés. De façon plus générale, l'application des méthodes MG et FMG avec des maillages non-structurés non-emboîtés a été pénalisée pendant des années par la difficulté de construire et d'utiliser de multiples maillages non-structurés, grossiers et fins, en particulier pour les applicatons industrielles. Cette difficulté est plus facile à surmonter aujourd'hui avec les récents progrès dans la génération et l'adaptation de maillages, voir e.g. [42]. Grâce à ces progrès, de nouveaux algorithmes d'adaptation de maillages fortement anisotropes

sont désormais disponibles.

Les *méthodes d'adaptation de maillage anisotropes* autorisent la génération de maillages dont chaque maille est susceptible de présenter des étirements importants (triangles très plat par exemple) pour prendre en compte des différences d'échelles suivant les directions spatiales de forte variation qui auront pu être détectées dans le champ physique inconnu par l'algorithme d'adaptation.

La parfaite adaptation du maillage à la physique nécessite de traiter le maillage comme une des inconnues que l'algorithme mathématique doit calculer.

La définition d'un maillage repose sur une structure de donnée très composite, comprenant typiquement des dimensions de tableaux, un graphe complexe, des vecteurs de coordonnées. Prendre en compte intégralement cette structure dans un processus d'optimisation du maillage conduit à des formulations topologiquement complexes (comment dire si deux maillages sont proches?) et d'une grande complexité de résolution (pas de convexité, coût probablement non-polynomial vis-à-vis du nombre d'inconnues).

À l'heure actuelle, un grand nombre d'approches par maillage adaptatif anisotrope remplacent la recherche directe d'un maillage par la recherche d'une métrique Riemannienne. Pour nous, une métrique Riemannienne est une matrice symétrique définie positive, calculable en tout point du domaine de calcul, permettant de mesurer par intégration la longueur de tout chemin joignant deux points de ce domaine. Elle permet en pratique de mesurer les arêtes du maillage. Dans la plupart des méthodes adaptatives anisotrope, la métrique optimale apparaît comme le minimum d'une forme quadratique convexe, c'est à dire calculable de manière simple. Autrement dit la nouvelle formulation a *régularisé* le problème de la recherche du maillage optimal. Dans cette thèse, nous utiliserons cette formulation en construisant un contexte mathématique continu. Les méthodes de métrique continues ont été formalisées progressivement par différents auteurs. Nous adopterons la formulation particulièrement complète de *métrique Riemannienne continue* proposée dans [54, 55].

Les maillages adaptés anisotropes présentent deux avantages importants. D'une part, ils permettent que de nombreux calculs soient effectués dans de bien meilleures conditions qu'avec les méthodes traditionnelles, permettant même dans certains cas des calculs qui n'étaient simplement pas possibles sans adaptation anisotrope, comme la propagation d'un bang sonique depuis un avion jusqu'au sol [60]. D'autre part, les adaptateurs de maillages anisotropes fournissent une convergence en maillage à un ordre élevé pour des problèmes singuliers [53].

Cependant, un second ingrédient important doit être pris en compte dans le processus d'adaptation. Il s'agit du *critère d'adaptation*, qui constitue le lien entre maillage et erreur de discrétisation.

La première génération de méthodes basées-métriques est celle des approches basées-hessien. Elle minimise l'*erreur d'interpolation* du champ inconnu, indépendamment de l'EDP résolue. Son développement a représenté un grand progrès en adaptation car le problème d'adaptation a eu pour la première fois une formulation

mathématique rigoureuse dans le cadre de la théorie de l'optimisation.

Néanmoins, l'utilisateur de l'outil qui en résulte peut objecter que son véritable souci réside dans la précision de *ce qui sort du calcul*, avec prise en compte de l'équation résolue. Les méthodes *orientées fonctionnelles* reposent sur le choix (par le programmeur) d'une quantité scalaire, nommée fonctionnelle, sortant du calcul, par exemple, dans notre exemple, la pression exercée sur l'obstacle à un temps donné. Dans ces méthodes, l'*erreur d'approximation* sur cette fonctionnelle remplace l'erreur d'interpolation précédente grâce à l'évaluation d'un champ supplémentaire, l'*état adjoint*. L'adaptation anisotrope orientée fonctionnelle répond rigoureusement au problème du maillage rendant la plus petite possible l'erreur d'évaluation de la fonctionnelle.

Les approches orientées fonctionnelle montrent leur limites quand l'utilisateur ne souhaite pas se contenter du contrôle d'une seule sortie scalaire du calcul. Dans cette thèse, nous proposons une nouvelle formulation, *orientée norme* permettant de contrôler plusieurs sorties scalaires (pour le même coût) et plus généralement l'écart dans une norme choisie entre un champ approché et le champ exact.

Cette thèse propose donc une combinaison de chacun des trois critères adaptatifs, basé sur l'interpolation, orienté fonctionnelle, orienté norme, avec la méthode de résolution FMG et une évaluation des trois algorithmes FMG-adaptatifs résultants. Enfin, elle propose une synthèse entre ces approches en métriques continues et une approche discrète, l'approche des métriques tensorielles.

Ce mémoire comporte huit chapitres. En premier lieu, nous présentons un rappel des notions de métriques riemanniennes et de maillages unités ainsi qu'une description du calcul d'erreurs d'interpolation $P1$ (chapitre 1).
Il s'agit ensuite de présenter les modèles numériques d'estimateurs d'erreur et de correcteurs, avec les approches *a priori* et *a posteriori* (chapitre 2).
Puis, on passe aux descriptions des méthodes de résolution elles-mêmes: tout d'abord le Full-Multigrid (chapitre 3), puis le Full-Multigrid adaptatif basé hessien (chapitre 4), puis le Full-Multigrid adaptatif orienté fonctionnelle (chapitre 5), le Full-Multigrid adaptatif orienté norme (chapitre 6) et, enfin, le Full-Multigrid adaptatif orienté norme en tensoriel (chapitre 7). Les travaux effectués lors du cadre de la collaboration avec la société LEMMA sont également présentés (chapitre 8) .

Le chapitre 4 résulte d'une collaboration avec Olivier Allain et Alain Dervieux publiée dans un journal, le chapitre 6 résulte d'une collaboration avec Alain Dervieux et est soumise dans un autre journal et le chapitre 7, article en préparation, résulte d'une collaboration avec Thierry Coupez et Alain Dervieux. *Ces trois chapitres reprennent le format et la langue des manuscrits d'articles correspondants.* Le chapitre 8 résulte d'une collaboration avec Olivier Allain et Vincent Levasseur.

Cette thèse a donné lieu aux communications scientifiques suivantes:

G. Brèthes, O. Allain, A. Dervieux, "A Mesh-Adaptative Metric-Based Full-Multigrid for the Poisson problem", *International Journal for Numerical Methods*

in Fluids,79-1, 30 53, 2015

G. Brèthes, A. Dervieux, "Anisotropic Norm-Oriented Mesh Adaptation for a Poisson problem" (soumis)

G. Brèthes, O. Allain, A. Dervieux, "Main issues in anisotropic mesh adaptive FMG", 11th World Congress on Computational Mechanics (WCCM XI)/ 5th European Conference on Computational Mechanics (ECCM V)/ 6th European Conference on Computational Fluid Dynamics (ECFD VI), July 20-25, 2014, Barcelona, Spain

G. Brèthes, A. Loseille, F. Alauzet, A. Dervieux, Convergent error-controlled mesh adaptation, VII International Conference on Adaptive Modeling and Simulation, ADMOS 2015, F. Chinesta, L. Chamoin, P. Diez (Eds)

G. Brèthes, A. Loseille, F. Alauzet, A. Dervieux, Main issues in anisotropic mesh adaptive FMG, PANACM 2015, 1st. Pan-American Congress on Computational Mechanics Buenos Aires, 27-29 April 2015

"Les deux idées des méthodes multigrilles", Exposé au Séminaire INRIA CaféIn, 2014.

"Adaptive multigrid methods", Exposé à la conférence Young ECCOMAS 2013 à Bordeaux, 2013

"Algorithmes full-multigrid adaptatifs basés sur des métriques riemanniennes", poster au congrès SMAI 2013 à Seignosse-le-Penon.

Ainsi qu'aux communications vers le grand public suivantes:

- Démonstrations aux fêtes de la Science 2013, 2014 et 2015,

- Participation à un film "Adaptation automatique de maillage en mécanique numérique industrielle", 2015.

Chapter 1

Représentation de maillage

1.1 Introduction

La mécanique des milieux continu considère la matière à l'image de la droite des réels, comme formée de blocs infiniment denses (sans vide), et infiniment découpables sans que l'on isole un quelconque "grain élémentaire de matière". Cette infinité de composants est incompatible avec le traitement numérique. Le bloc de matière va être remplacé par un réseau de mailles porteuses de propriétés physiques, pas si loin du réseau d'atomes d'un solide, mais dimensionné par les capacités de l'ordinateur, c'est-à-dire beaucoup plus grossier (beaucoup moins de mailles que d'atomes). Ce procédé est la discrétisation. Il repose sur le choix et la définition d'un maillage sur lequel sera résolu un problème approché.

Avec les maillages non-structurés, l'ingénieur peut mieux s'approcher de la physique en maillant mieux la forme du domaine de calcul (définie par exemple par celle d'un objet manufacturé de géométrie complexe) et en maillant mieux les champs solutions de l'équation discrète via l'adaptation de maillage. Mais le maillage non-structuré est un objet mathématique complexe : un graphe et des coordonnées. Il est un *objet peu régulier* car on ne sait pas bien identifier que deux maillages sont proches pour les applications que nous en tirons: nous ne savons pas mesurer une distance utile entre deux maillages, autrement dit les maillages non-structurés ne sont pas dans un espace métrique. Or on aimerait bien les optimiser, et dans ce but, idéalement, les mettre dans un espace de Hilbert.

Dans ce premier chapitre, nous rappelons les différentes notions nécessaires à une représentation régulière du mailllage dans notre étude: les métriques riemanniennes, les maillages unités et une méthode d'estimation des erreurs d'interpolation $P1$. Une présentation et une justification en grand détails de ces notions peuvent être trouvées dans [54, 55].

1.2 Métrique

L'outil le plus essentiel à l'adaptation de maillages anisotropes est la métrique riemannienne.
Considérons l'espace $\mathbb{R}^{dim}$ où la dimension dim est 2 ou 3. En géométrie euclidienne, la norme L_2, correspondant à la longueur euclidienne classique, est définie par le produit scalaire usuel. Ainsi, la distance, c'est-à-dire la norme euclidienne, entre deux points $\mathbf{A}$ et $\mathbf{B}$ est définie de la façon suivante:

$$dist(\mathbf{A},\mathbf{B}) = \left|\left|\overrightarrow{\mathbf{AB}}\right|\right| = \left\langle \overrightarrow{\mathbf{AB}}, \overrightarrow{\mathbf{AB}} \right\rangle = \sqrt{\overrightarrow{\mathbf{AB}}^T \overrightarrow{\mathbf{AB}}}.$$

En géométrie riemannienne, les distances sont mesurées à l'aide d'un produit scalaire variant en fonction du point de l'ensemble considéré (en général une variété). Dans $\mathbb{R}^{dim}$, une métrique riemannienne $g_{\mathbf{x}}$ est une forme quadratique définie positive dépendant du point $\mathbf{x}$ du domaine Ω.

$$g_{\mathbf{x}} : \Omega \longrightarrow \mathbb{R}$$

$$g_{\mathbf{x}} : (\mathbf{u},\mathbf{v}) \longrightarrow g_{\mathbf{x}}(\mathbf{u},\mathbf{v})$$

On note $\mathcal{M}(\mathbf{x})$ la matrice associée à $g_{\mathbf{x}}$. On peut alors définir la distance entre deux points suivant le champ de métriques $\mathcal{M}$:

$$dist_{\mathcal{M}}(\mathbf{A},\mathbf{B}) = \int_0^1 \sqrt{\overrightarrow{\mathbf{AB}}^T \mathcal{M}(\mathbf{A} + \theta\overrightarrow{\mathbf{AB}})\overrightarrow{\mathbf{AB}}}\ \mathrm{d}\theta$$

La géométrie euclidienne précédente est simplement le cas particulier pour lequel $\mathcal{M}(\mathbf{x})$ est égale l'identité pour tout $\mathbf{x}$. Dans les travaux effectués ici, chaque $\mathcal{M}(\mathbf{x})$ est définie à partir d'une solution approchée du problème. $\mathcal{M}(\mathbf{x})$ est une matrice symétrique définie positive de taille $(dim \times dim)$ que l'on peut diagonaliser sous la forme suivante:

$$\mathcal{M}(\mathbf{x}) = \mathcal{R}(\mathbf{x})^T \begin{pmatrix} \lambda_1(\mathbf{x}) & 0 \\ 0 & \lambda_2(\mathbf{x}) \end{pmatrix} \mathcal{R}(\mathbf{x}), \tag{1.1}$$

en 2D ou

$$\mathcal{M}(\mathbf{x}) = \mathcal{R}(\mathbf{x})^T \begin{pmatrix} \lambda_1(\mathbf{x}) & 0 & 0 \\ 0 & \lambda_2(\mathbf{x}) & 0 \\ 0 & 0 & \lambda_3(\mathbf{x}) \end{pmatrix} \mathcal{R}(\mathbf{x}), \tag{1.2}$$

en 3D.

Les rotations $\mathcal{R}$ définissent les dim directions d'étirement locales, et les λ_k leur forces respectives.

1.3 Maillage unité

Étant donné un champ de métrique, on en déduit un maillage unité qui est le maillage paramétré par cette métrique. Dans ce qui suit, on considère seulement des maillages par simplexes, c'est-à-dire des triangles en 2D, des tétraèdres en 3D. On suppose de plus qu'ils sont des maillages en éléments finis valides, au sens suivant:

- le domaine de calcul est partitionné en un nombre fini d'éléments simplexes tous de mesure non-nulle,
- l'intersection entre deux éléments distincts est soit vide, soit un sommet, soit une arête, soit une face.

1.3.1 Elément unité

Considérons un élément K d'un maillage, triangle ou tétraèdre suivant qu'on travaille en 2D ou en 3D. K est un *élément unité suivant le champ de métriques* $\mathcal{M}$ si chacune de ses arêtes a une longueur suivant $\mathcal{M}$ égale à 1. Pour un élément K, on considère sa liste d'arêtes $(\mathbf{e_i})_{i=1,...,nbAretes}$ où $nbAretes$ vaut 3 en 2D et 6 en 3D.
La longueur $l_{\mathcal{M}}(\mathbf{e_i})$ d'une arête $\mathbf{e_i}$ suivant $\mathcal{M}$ est définie comme la distance suivant $\mathcal{M}$ entre ses deux extrémités.

Ainsi, si on appelle $\mathbf{A_i}$ et $\mathbf{B_i}$ les deux extrémités de $\mathbf{e_i}$,

$$l_{\mathcal{M}}(\mathbf{e_i}) = dist_{\mathcal{M}}(\mathbf{A_i}, \mathbf{B_i}) = \int_0^1 \sqrt{\overrightarrow{\mathbf{A_iB_i}}^T \mathcal{M}(\mathbf{A_i} + \theta\overrightarrow{\mathbf{A_iB_i}})\overrightarrow{\mathbf{A_iB_i}}}\, \mathrm{d}\theta$$

K est un élément unité suivant $\mathcal{M}$ si et seulement si:

$$\forall i = 1, ..., nbAretes, l_{\mathcal{M}}(\mathbf{e_i}) = 1.$$

En 2D, l'aire de K suivant $\mathcal{M}$ vaut $|K|_{\mathcal{M}} = \frac{\sqrt{3}}{4}$.
En 3D, le volume de K suivant $\mathcal{M}$ vaut $|K|_{\mathcal{M}} = \frac{\sqrt{2}}{12}$.

Theorem 1.1
[54, 55] Pour tout élément K tel que $|K|_{Id} \neq 0$, il existe un unique tenseur de métrique $\mathcal{M}_K$ pour lequel K est un élément unité suivant $\mathcal{M}_K$, voir figure 1.1.

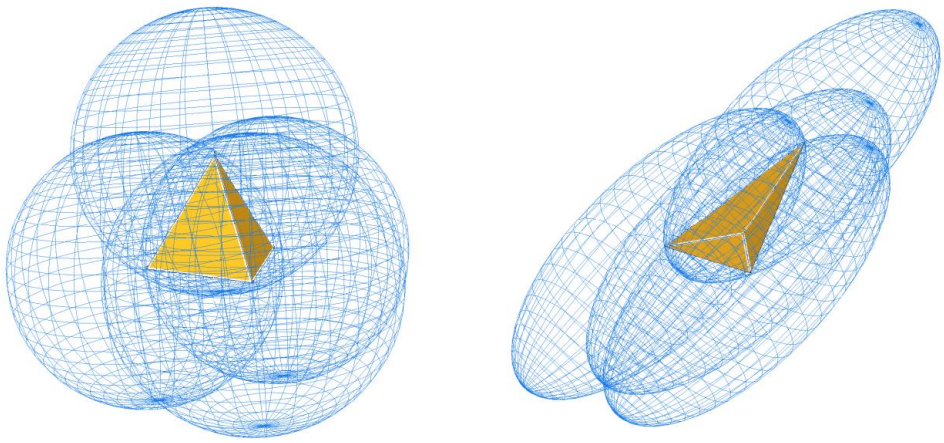

Figure 1.1: Exemples d'éléments unité 3D

Sur la figure 1.1, on peut observer chaque boule unité ayant pour centre un des sommets du tétraèdre. À gauche, le cas où $\mathcal{M}_K$ vaut l'identité et, à droite, le cas où $\mathcal{M}_K$ est quelconque. Sur les deux exemples, on observe que chaque boule unité,

centrée sur un sommet, a les trois autres sommets à sa surface. Dans les deux cas, K est un élément unité suivant $\mathcal{M}_K$.

1.3.2 Elément quasi-unité et maillage unité

On est tenté de définir la notion de maillage unité comme un maillage dont tous les éléments sont des éléments unités suivant un champ de métrique donné. Malheureusement, il n'est pas du tout garanti que, pour une métrique donnée, on puisse construire un maillage dont tous les éléments sont des éléments unité.
On introduit donc une notion d'*élément quasi-unité*. Au lieu de considérer des longeurs d'arêtes égales à 1, on se contente de longueurs comprises dans un intervalle proche de 1. Malheureusement, cette contrainte plus faible entraîne une incertitude sur la valeur de l'aire/volume qui peut dès lors valoir 0. Pour éviter cela, on pose une contrainte sur l'aire/volume en définissant la fonction qualité

$$\mathcal{Q}_{\mathcal{M}}(K) = \frac{36|K|_{\mathcal{M}}^{2/3}}{3^{1/3}\sum_{i=1}^{6} l_{\mathcal{M}}^2(\mathbf{e_i})}$$

Pour deux réels α et β tels que $\alpha > 0$ et $0 < \beta < 1$, un élément K est un *élément quasi-unité* suivant $\mathcal{M}$ si et seulement si

$$\forall i = 1, ..., nbAretes, l_{\mathcal{M}}(e_i) \in [\frac{1}{\alpha}, \alpha] \quad \text{et} \quad \mathcal{Q}_{\mathcal{M}}(K) \in [\beta, 1]$$

En pratique, on prend $\alpha = \sqrt{2}$ et $\beta = 0.8$.

Un maillage est un *maillage unité suivant* $\mathcal{M}$ si et seulement si tous ses éléments sont des éléments quasi-unité suivant $\mathcal{M}$

Le paramétrage qui fait correspondre en pratique une métrique et un maillage unité n'est donc certainement pas une correspondance biunivoque. En particulier à chaque métrique correspond une *classe* de maillages unité. L'important est qu'à cette classe on fasse (approximativement) correspondre un seul champ d'erreur d'approximation. Ce point sera abordé à partir du paragraphe suivant. Signalons enfin qu'en pratique, le représentant dans la classe sera instancié par l'algorithme discret de remaillage choisi.

1.3.3 Nombre de sommets

En algorithmique numérique il est coutumier de définir la complexité d'un algorithme de résolution d'EDP discrète en fonction du nombre d'inconnues, qui n'est autre que le nombre de sommets du maillage multiplié par un facteur faisant intervenir des quantités constantes (notamment nombre d'inconnues par nœuds). Or on sait que résoudre une EDP sur un maillage très étiré va coûter plus cher que la résolution sur un maillage non-étiré (isotrope) contenant le même nombre de nœuds. Cette différence compte cependant peu devant les avantages de la simplicité d'une mesure de la complexité du maillage basée sur le nombre de sommets. C'est l'option choisie dans la théorie de la métrique continue. Par ailleurs, plusieurs maillages unité d'une même métrique peuvent avoir des nombres de sommets légèrement

différents. Nous préférerons compter ce nombre de sommets directement à partir de la métrique. Définissons donc le nombre de sommets relatifs à une métrique donnée. Rappelons qu'un tenseur de métriques s'écrit sous la forme:

$$\mathcal{M}(\mathbf{x}) = \mathcal{R}(\mathbf{x})^T \begin{pmatrix} \lambda_1(\mathbf{x}) & 0 \\ 0 & \lambda_2(\mathbf{x}) \end{pmatrix} \mathcal{R}(\mathbf{x}), \tag{1.3}$$

en 2D ou

$$\mathcal{M}(\mathbf{x}) = \mathcal{R}(\mathbf{x})^T \begin{pmatrix} \lambda_1(\mathbf{x}) & 0 & 0 \\ 0 & \lambda_2(\mathbf{x}) & 0 \\ 0 & 0 & \lambda_3(\mathbf{x}) \end{pmatrix} \mathcal{R}(\mathbf{x}), \tag{1.4}$$

en 3D.

Si on considère un maillage unité suivant $\mathcal{M}$, on a $\lambda_i(\mathbf{x}) = \frac{1}{h_i^2(\mathbf{x})}$ où $h_i(\mathbf{x})$ est la taille de maille dans la i-ième direction caractéristique. On peut alors calculer la densité par sommet $d_{\mathcal{M}}(\mathbf{x})$.
En 2D,

$$d_{\mathcal{M}}(\mathbf{x}) = \frac{1}{h_1(\mathbf{x})h_2(\mathbf{x})} = \sqrt{\lambda_1(\mathbf{x})\lambda_2(\mathbf{x})} = \sqrt{det(\mathcal{M}(\mathbf{x}))}.$$

En 3D,

$$d_{\mathcal{M}}(\mathbf{x}) = \frac{1}{h_1(\mathbf{x})h_2(\mathbf{x})h_3(\mathbf{x})} = \sqrt{\lambda_1(\mathbf{x})\lambda_2(\mathbf{x})\lambda_3(\mathbf{x})} = \sqrt{det(\mathcal{M}(\mathbf{x}))}.$$

On en déduit l'expression du *nombre total de sommets, noté $\mathcal{C}(\mathcal{M})$ d'une métrique* et par extension, du nombre de sommets de tout maillage unité de cette métrique $\mathcal{M}$ comme donné par:

$$\mathcal{C}(\mathcal{M}) = \int_\Omega d_{\mathcal{M}}(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_\Omega \sqrt{det(\mathcal{M}(\mathbf{x}))}\,\mathrm{d}\mathbf{x}.$$

1.4 Erreurs d'interpolation P1

La première erreur de discrétisation à laquelle nous allons nous intéresser est l'erreur d'interpolation P_1.

1.4.1 Modélisation de l'erreur d'interpolation

On se donne un domaine de calcul Ω dans $\mathbb{R}^2$ ou $\mathbb{R}^3$ et sa discrétisation Ω_h. Pour simplifier nos notations, nous prendrons l'habitude de supposer que $\Omega = \Omega_h$. On note:

$$\mathcal{V} = H_0^1(\Omega) \cap H^2(\Omega).$$

où $H^1(\Omega)$ et $H^2(\Omega)$ sont les espaces de Sobolev usuels et $H_0^1(\Omega)$ le sous-esapce des fonctions de $H^1(\Omega)$ nulles au bord.

Le domaine $\Omega = \Omega_h$ est triangulé par une triangulation (2D) ou une tétraèdrisation (3D) notée $\mathcal{H}$. On note $V_{\mathcal{H}}$ l'espace des fonctions continues sur Ω_h, affines sur chaque élément K de $\mathcal{H}$. On note $\Pi_{\mathcal{H}}$ l'*interpolateur* P_1 sur le maillage $\mathcal{H}$:

$$\Pi_{\mathcal{H}} : \mathcal{V} \to V_{\mathcal{H}} \quad v \mapsto \Pi_{\mathcal{H}}(v) \text{ avec } \Pi_{\mathcal{H}}(v)(\mathbf{x}_i) = v(\mathbf{x}_i) \ \forall\, i, \text{ sommet de } \mathcal{H}.$$

Pour une fonction régulière u, on peut faire correspondre à tout maillage unité $\mathcal{H}$ d'une métrique $\mathcal{M}$ une erreur d'interpolation P_1 notée $u - \Pi_{\mathcal{H}} u$. Pour deux maillages unité d'une même métrique, la différence entre les erreurs d'interpolation n'est pas à notre sens significative ; on devrait pouvoir noter $\Pi_{\mathcal{M}}$ pour désigner $\Pi_{\mathcal{H}}$. Pour deux métriques très proches l'une de l'autre au sens par exemple de:

$$||\mathcal{M} - \mathcal{M}'||_{(\mathcal{C}^0(\Omega))^{dim^2}} \leq \varepsilon$$

nous considérerons que la différence des erreurs d'interpolation est elle aussi très petite. Ces idées sont bien exprimées par la construction d'un *modèle continu d'erreur d'interpolation.* On montre dans [54, 55], que l'erreur d'interpolation liée au maillage unité est bien représentée par une fonction de la métrique et du hessien de u.

L'erreur d'interpolation continue est définie comme suit:

$$|u - \pi_{\mathcal{M}} u|(\mathbf{x}) \quad = \quad \frac{1}{10} \text{trace}(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})\, |H_u(\mathbf{x})|\, \mathcal{M}^{-\frac{1}{2}}(\mathbf{x})) \tag{1.5}$$

où $|H_u|$ est déduit de H_u en prenant les valeurs absolues de ses valeurs propres. Le lecteur notera que ce modèle dépend analytiquement de la métrique dans l'ouvert des matrices symétriques à valeurs propres strictement positives dans l'espace $\mathbb{R}^{\frac{dim(dim+1)}{2}}$ des matrices symétriques.

1.4.2 Optimisation de l'erreur d'interpolation continue

Une première illustration de la puissance de cette modélisation est le calcul de la métrique optimale minimisant le second membre de (1.5) sous la contrainte d'un nombre total de sommets $\mathcal{C}(\mathcal{M})$ égal au paramètre N. Dans [54, 55], après avoir résolu analytiquement ce problème d'optimisation, on déduit l'unique métrique optimale $(\mathcal{M}^{\mathbf{L}^p}_{opt}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ comme:

$$\mathcal{M}^{\mathbf{L}^p}_{opt} = D_{\mathbf{L}^p} \left(\det |H_u|\right)^{\frac{-1}{2p+2}} |H_u| \text{ avec } D_{\mathbf{L}^p} = N \left(\int_\Omega (\det |H_u|)^{\frac{p}{2p+2}} \right)^{-1}, \tag{1.6}$$

où $D_{\mathbf{L}^p}$ est un terme de normalisation globale pour obtenir un maillage continu de complexité N. On notera que la partie matricielle $|H_u|$ est indépendante du choix de la norme $\mathbf{L}^p$ tandis que le champ scalaire $(\det |H_u|)^{\frac{-1}{2p+2}}$ est un terme de normalisation locale reflétant l'influence de cette norme. *Dans la suite, pour simplifier, nous nous restreignons au cas où* $p = 2$.

Conclusion du chapitre

La paramétrisation de maillage par métrique continue est une paramétrisation très régulière, dans un sous-ensemble de fonctions supposément régulières à valeurs dans $\mathbb{R}^{dim}$. Pour qu'elle donne accès à des applications, nous devons introduire des modèles décrivant l'effet du choix du maillage paramétré sur les différentes erreurs de discrétisation. Nous avons introduit un premier modèle d'erreur d'interpolation P_1. C'est une fonction régulière et simple de la métrique. Nous avons vu que cela permet une optimisation analytique de l'erreur d'interpolation par rapport à la métrique. Nous verrons au chapitre 4 comment ce schéma conduit à une méthode d'adaptation basée sur l'erreur d'interpolation. Le passage à des erreurs d'approximation d'EDP (abordé dans les chapitres 4 et suivants) est moins direct. Dans cette perspective, nous rappellerons dans le prochain chapitre (chapitre 3) quelques éléments d'estimation d'erreur et de construction d'un correcteur.

Chapter 2

Modèle numérique, estimateurs et correcteurs

2.1 Introduction

En simulation numérique le premier résultat produit est le champ solution approchée, disons u_h, approximation de la solution u, dite continue ou exacte, de l' EDP modélisant le modèle physique. Nous rappellerons rapidement sur le cas particulier qui nous intéresse dans ce mémoire, c'est à dire les éléments finis P_1 pour le problème de Poisson, comment est défini u_h. L'erreur à réduire qui nous intéresse maintenant est l'erreur d'approximation $u - u_h$. Elle pourra être mesurée suivant différentes normes, et en particulier en norme L^2:

$$\varepsilon = |u - u_h|_{L^2(\Omega)}.$$

Un but essentiel du numéricien est de contrôler le niveau de cette erreur. Mais bien sûr, il ne connait pas $u - u_h$ mais seulement u_h. Il s'agira donc de substituer à la connaissance de $u - u_h$ celle de termes remplaçants qui l'approchent de manière plus ou moins précise. Deux types de termes remplaçants seront considérés. D'une part des estimateurs qui seront typiquement des majorants de $|u - u_h|$, et d'autre part des correcteurs c_h rendant $u_h + c_h$ plus proche de u.

Ce chapitre couvre pour le problème de Poisson un rapide état de l'art concernant le cheminement qui conduit du maillage jusqu'à l'erreur et sa correction.

2.2 Approximation P1 du problème de Poisson

Les méthodes numériques étudiées dans cette thèse ont pour objectif la résolution des équations aux dérivées partielles (EDP). Nous les présentons pour un modèle elliptique simple utilisé dans toute cette thèse pour illustrer les méthodes de résolution et d'adaptation proposées.

On considère un espace Ω inclus dans $\mathbb{R}^{dim}$ et on pose $V = H_0^1(\Omega)$. Le modèle d'EDP de base qui sera considéré dans la plus grande partie de ce mémoire est le suivant:

$$u \in H_0^1(\Omega), \quad -div(\mu \nabla u) = f \tag{2.1}$$

où $f \in L^2(\Omega)$ est donnée, où $u \in V$ est notre inconnue et où μ est une variable connue pouvant éventuellement être discontinue. Notons que, dans le cas où μ est constante et égale à 1, on retombe sur le Laplacien.

Classiquement, il existe une unique solution à (2.1). Pour μ suffisamment régulier, cette solution est dans $H^2(\Omega)$ et dans $\mathcal{C}^0(\Omega)$. La continuité de u est vraie aussi pour des champs μ discontinus mais réguliers de part et d'autre d'une courbe de discontinuité assez régulière, comme on en rencontre dans les modèles multi-fluides.

La formulation variationnelle s'écrit:

$$u \in V, \quad \forall v \in V \quad \int_\Omega \mu(\mathbf{x}) \nabla u(\mathbf{x}).\nabla v(\mathbf{x}) \, \mathrm{d}\mathbf{x} = (f, v).$$

avec $(f, v) = \displaystyle\int_\Omega f(\mathbf{x}) v(\mathbf{x}) \, \mathrm{d}\mathbf{x}$

Considérons un maillage $\mathcal{H}^{\mathcal{M}}$, unité pour la métrique $\mathcal{M}$ et de N sommets. Nous utilisons une méthode d'éléments finis P_1 pour discrétiser notre équation sur ce maillage. Nous indexons directement le contexte discret par la métrique $\mathcal{M}$.

Pour tous les sommets internes du maillage on définit les fonctions $(\phi_i^{\mathcal{M}}) \in V$ telles que, pour chaque fonction $\phi_i^{\mathcal{M}}$ et pour chaque point $\mathbf{x}_j$ du maillage $\mathcal{H}^{\mathcal{M}}$, on ait:

$$\begin{cases} \phi_i^{\mathcal{M}}(\mathbf{x}_j) = 0 \ si \ j \neq i \\ \phi_i^{\mathcal{M}}(\mathbf{x}_j) = 1 \ si \ j = i \quad et \quad \mathbf{x}_j \notin \partial\Omega. \end{cases}$$

Nous notons $V_{\mathcal{M}}$ l'espace engendré par les fonctions $\phi_i^{\mathcal{M}}$. Le problème discret se formule comme suit:

$$u_{\mathcal{M}} \in V_{\mathcal{M}}, \quad \forall v \in V_{\mathcal{M}} \quad \int_\Omega \mu(\mathbf{x}) \nabla u_{\mathcal{M}}(\mathbf{x}).\nabla v(\mathbf{x}) \, \mathrm{d}X = (f, v).$$

2.3 Rappels: estimations d'interpolation

Étant donnée une fonction u que l'on interpole P_1, son interpolée $\Pi_h u$ approche la fonction u à l'ordre un dans H^1 et à l'ordre deux dans L^2:

Proposition: Pour $u \in H^2$, il existe $k_1 \geq 0$ et $k_2 \geq 0$ tels que:

$$|u - \Pi_h u|_{H^1} \quad \leq \quad k_1 h |u|_{H^2} \tag{2.2}$$

et

$$|u - \Pi_h u|_2 \quad \leq \quad k_2 h^2 |u|_2 \tag{2.3}$$

avec les notations: $|v|_2 = \sqrt{(vv)}$ et $|u|_{H^m} = \sqrt{\sum_{\alpha \leq m} (D^\alpha v, D^\alpha v)}$.

2.4 Estimations *a priori*

Quand on utilisera $u_{\mathcal{M}}$ au lieu de u, on commettra une erreur d'approximation $u - u_{\mathcal{M}}$. La théorie de l'approximation par éléments finis nous dit que si le maillage est pris assez fin, certaines normes de l'erreur seront petites. Les estimations *a priori* ne sont pas formulées en fonction de la solution discrète $u_{\mathcal{M}}$ mais seulement en

fonction de la solution continue u. Elles permettent notamment d'établir la convergence de l'une vers l'autre et l'ordre asymptotique de convergence. L'estimation de base pour notre approximation est le lemme de Céa (qui remonte à [26]):

$$||u - u_{\mathcal{M}}||_{H^1(\Omega)} \leq kh||u||_{H^2(\Omega)}. \tag{2.4}$$

Cette estimation découle de la propriété de projection que l'on peut écrire:

$$||u - u_{\mathcal{M}}||_{H^1(\Omega)} \leq k'||u - \Pi_{\mathcal{M}} u||_{H^1(\Omega)}.$$

Cette dernière estimation est souvent invoquée pour justifier une adaptation basée sur le hessien de u via (2.2) ou plus exactement un hessien approché de $u_{\mathcal{M}}$. Mais dans la pratique les méthodes adaptatives basées hessien ne mesurent que des normes L^2 de l'erreur d'interpolation au lieu d'une norme H^1. La justification n'est donc pas rigoureuse mais seulement heuristique. D'autre part, l'estimation (2.4) a l'avantage de concerner aussi les dérivées premières, mais n'est que d'ordre un vis-à-vis de la taille de maille.

Des analyses d'erreur *a priori* de l'*erreur d'approximation* d'ordre deux ont été obtenues indépendamment par Aubin [9] et Nitsche [68]. Il s'agit d'une majoration de la norme de l'erreur dans l'espace L^2. *Ces estimations à l'ordre deux dans L^2 seront au centre des critères d'adaptation développés dans cette thèse.* Nous en donnons maintenant une formulation.

Proposition: Si on suppose que la solution exacte u est dans H^2, alors il existe $k \geq 0$ tel que :

$$|u - u_h|_2 \quad \leq \quad kh^2|u|_{H^2} \tag{2.5}$$

où u_h est la solution discrète dans $\mathcal{V}_h$.

La démonstration repose sur l'introduction d'un état adjoint. L'estimation issue du lemme d'Aubin-Nitsche est sous sa forme actuelle très grossière et nous devrons faire dans les chapitres 5, 6, 7 des analyses plus fines pour trouver des critères d'adaptation. L'introduction d'un état adjoint suggère de rendre plus facile l'analyse d'erreur en introduisant une fonctionnelle, ce qu'ont proposé Becker et Rannacher.

2.5 Estimateurs *a posteriori* de Becker-Rannacher

L'idée est de s'intéresser à la précision obtenue sur un scalaire déduit de l'inconnue u:

$$j : V \to \mathbb{R} \qquad u \mapsto j(u).$$

On parlera d'analyse orientée fonctionnelle (*goal-oriented*). Considérons une forme bilinéaire a de V^2 dans $\mathbb{R}$ et une fonction donnée $g \in V^2$. On définit z comme l'unique solution du système adjoint:

$$\psi \in V, a(\psi, z) = (g, \psi),$$

On peut alors écrire:

$$j(u) - j(u_h) = (g, u - u_h) = a(u - u_h, z).$$

On note $\bar{\Pi}_h$ l'interpolateur de Clément introduit dans [28]:

$$\bar{\Pi}_h : V \to V_h \ , \ v \mapsto \bar{\Pi}_h v \quad ; \quad \bar{\Pi}_h(v)(\mathbf{x}_i) = \int_{Supp\phi_i} v \mathrm{d}\mathbf{x} / \int_{Supp\phi_i} \mathrm{d}\mathbf{x} \ \ \forall\, i, \ \text{sommet de } \mathcal{H},$$

où $Supp\phi_i$ désigne le support de la fonction de base ϕ_i. Alors on a :

$$a(u, z) - a(u_h, z) = a(u, z - \bar{\Pi}_h z) - a(u_h, z - \bar{\Pi}_h z) + a(u, \bar{\Pi}_h z) - a(u_h, \bar{\Pi}_h z) \ .$$

Et d'après les équations d'état continues et discrètes, les derniers termes sont nuls. En notant $\delta z = z - \bar{\Pi}_h z$, il vient:

$$a(u, \delta z) - a(u_h, \delta z) = (f, \delta z) - a(u_h, \delta z)$$

et en posant $\rho(u_h, \delta z) = (f, \delta z) - a(u_h, \delta z)$ il vient

$$j(u) - j(u_h) = a(u - u_h, \delta z) = \rho(u_h, z - \bar{\Pi}_h z) \ .$$

Ce résidu de poids δz se décompose sur les élements K et faces E du maillage:

$$\rho(u_h, \delta z) \ = \ \sum_K (f + \Delta u_h, \delta z)_K + \sum_E \left([\partial_n u_h] \, , \delta z \right)_E .$$

Commençons les majorations:

$$\rho(u_h, \delta z) \ \leq \ \Sigma_{K \in \tau_h} \rho_K(u_h, \delta Z_h)$$

$$\rho_K(u_h, \delta Z_h) \ = \ ||f + \Delta u_h||_K \ ||\delta z||_K \ + \ \frac{1}{2} \Sigma_{E \in \mathcal{E}(K)} || \, [\partial_n u_h] \, ||_E ||\delta z||_E .$$

Le terme $||\delta z|| = ||z - \bar{\Pi}_h z||$ doit être approché : deux niveaux d'interpolation de Z_h sont utilisés pour construire δZ_h. Cela peut reposer sur des différentes finesse de maillage (un grossier, et un deux fois plus fin) ou sur différentes degrés d'interpolation (ce qui suppose un minimum de convergence en maillage pour l'adjoint z).

Pour décider de la finesse du nouveau maillage, $\rho(u_h, \delta Z_h)_K$ sera comparé avec une tolérance "TOL". On verra plus loin que la comparaison avec une tolérance est rendue inutile par le choix d'un paramétrage par métrique. Le reste du développement plus haut est à la base des approches *orientées fonctionnelles*.

Becker et Rannacher proposent aussi une méthode *a posteriori* sans fonctionnelle dans [12] qui se rapproche de ce que nous appellerons dans un autre chapitre une méthode orientée-norme :

$$(u_h - u, g) = \rho(u_h, \delta z) \ \ \text{mais } g \text{ est inconnu} \ .$$

Dans ce cas ces auteurs choisissent comme estimation:

$$\rho(u_h, \delta z) \ \leq \ \Sigma_{K\in\tau_h}||f + \Delta u_h||_K \ ||\delta z||_K \ + \ \Sigma_{E\in\mathcal{E}}|| \left[\partial_n u_h\right] ||_E ||\delta z||_E$$

L'adjoint z est inconnu mais supposé régulier, d'où:

$$||u - u_h|| \leq const. \left(\sum_{K\in\tau_h} \eta_K^2 \right)^{1/2}$$

avec:

$$\eta_K \ = \ h_K \left(||f + \Delta u_h||_K \ | \ + \ \frac{1}{2}\Sigma_{E\in\mathcal{E}(K)}|| \left[\partial_n u_h right\right]||_E) \ .$$

La méthode de la tolérance est alors adaptée comme suit: l'indicateur local η_K est équilibré en raffinant ou de-raffinant les éléments K de τ_h suivant:

$$\eta_K \ = \ \frac{TOL}{\sqrt{N} const.} \ , \qquad N \ \text{ nombre d'éléments.}$$

La majoration de l'état adjoint par une constante rend cette méthode moins précise que la précédente.

2.6 Correcteurs

2.6.1 Correcteurs pour des fonctionnelles

L'intérêt pour la construction de correcteurs a cru de manière importante avec la proposition de correcteurs *a posteriori* pour les fonctionnelles formulée par Giles et Pierce, cf. [44, 72, 45, 43].

On considère à nouveau une fonctionnelle dépendant du champ inconnu u:

$$j : L^2(\Omega) \to I\!R, \quad j(u) = (u, g).$$

Et on admet que l'erreur $|R_h \mathbf{u}_h - u|$ est d'ordre α, alors:

$$j_h = (R_h \mathbf{u}_h, g) = j(u) + (R_h \mathbf{u}_h - u, g) = j(u) + O(h^\alpha).$$

On peut alors soit mieux évaluer cette erreur, soit essayer de la corriger :

$$j_h \ = \ j(u) + (A^{-1}(AR_h \mathbf{u}_h - f), g) \ = \ j(u) + (AR_h \mathbf{u}_h - f, z)$$

où on a introduit l'état adjoint continu noté z et son analogue discret Z_h:

$$z \ = \ A^{-*} g \ ; \ Z_h = R_h A_h^{-*} g.$$

Le terme d'erreur que nous avons mis en évidence peut être approximativement corrigé, de façon à obtenir la fonctionnelle corrigée :

$$\hat{j}_h \ = \ (R_h \mathbf{u}_h, g) - (AR_h \mathbf{u}_h - f, Z_h)$$

dont l'intérêt est démontré par l'ordre formellement élevé du nouveau terme d'erreur:

$$\hat{j}_h - j(u) = (AR_h\mathbf{u}_h - f, z - R_hZ_h).$$

On pourrait aussi supposer que le système adjoint converge à l'ordre α, $|z - R_hZ_h| = O(h^\alpha)$. Il en résulterait:

$$\hat{j}_h \quad = \quad j(u) + O(h^{2\alpha}) \ .$$

Dans le cas d'approximations en différences finies structurées avec des maillages lisses et des interpolations R_h lisses, des coefficients de trainée et de portance sont obtenus au quatrième ordre dans [72, 71].
Dans le cas général et notamment en non-structuré, le résidu continu $AR_h\mathbf{u}_h - f$ ne converge pas vers zéro dans $L^2(\Omega)$ à l'ordre α mais le correcteur peut néanmoins apporter un bon supplément de précision sur la fonctionnelle.

2.6.2 Champ correcteur basé Defect-Correction

Soit $u_\mathcal{M}$ un champ discret. Nous nous proposons de trouver une approximation *a posteriori* robuste du *champ d'erreur* $u - u_\mathcal{M}$ et non plus seulement d'une fonctionnelle scalaire. Pour cela, nous nous reposerons sur un principe de convergence en maillage. On considère un nouveau maillage, plus fin que le maillage courant. Pour le construire, on divise chaque élément du maillage courant en prenant le milieu de chaque arête. La figure 2.1 montre une subdivision d'un triangle et la figure 2.2 celle d'un tétraèdre. On obtient quatre sous-triangles en 2D et huit sous-tétraèdres en

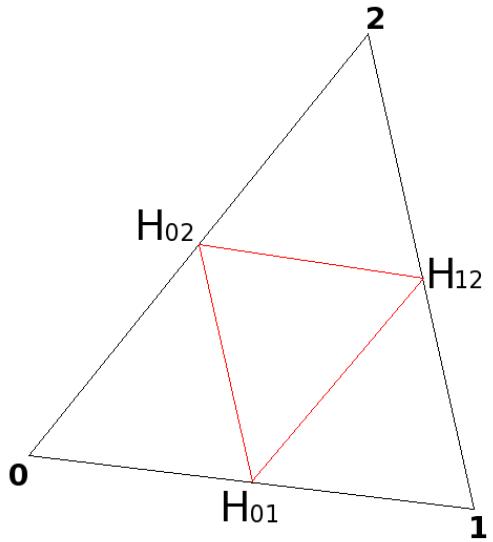

Figure 2.1: Subdivision d'un triangle

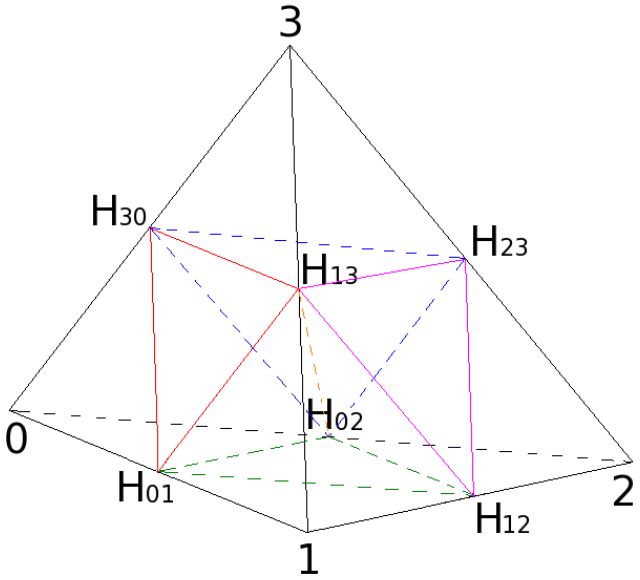

Figure 2.2: Subdivision d'un tétraèdre

3D. On considère les fonctions $(\tilde{\phi}_i^{\mathcal{M}/2})$ de base de l'espace $\tilde{V}_{\mathcal{M}/2}$ associé à ce nouveau maillage. Pour chaque point $\mathbf{x}_i$,

$$a(u - u_{\mathcal{M}}, \tilde{\phi}_i^{\mathcal{M}/2}) = a(u, \tilde{\phi}_i^{\mathcal{M}/2}) - a(u_{\mathcal{M}}, \tilde{\phi}_i^{\mathcal{M}/2})$$

$$a(u - u_{\mathcal{M}}, \tilde{\phi}_i^{\mathcal{M}/2}) = (f, \tilde{\phi}_i^{\mathcal{M}/2}) - a(u_{\mathcal{M}}, \tilde{\phi}_i^{\mathcal{M}/2})$$

On définit le résidu $res_{\mathcal{M}/2}^{post}$ de sorte que
$\forall \mathbf{x}_i \in \mathcal{M}/2$, $res_{\mathcal{M}/2}^{post}(\mathbf{x}_i) = (f, \tilde{\phi}_i^{\mathcal{M}/2}) - a(u_{\mathcal{M}}, \tilde{\phi}_i^{\mathcal{M}/2})$.

On définit $\mathcal{R}_{\mathcal{M}/2 \to \mathcal{M}}$ de sorte que $\forall \mathbf{x}_i \in \mathcal{M}$:

$$(\mathcal{R}_{\mathcal{M}/2 \to \mathcal{M}} res_{\mathcal{M}/2}^{post})(\mathbf{x}_i) = res_{\mathcal{M}/2}^{post}(\mathbf{x}_i) + \sum_{T \in \mathcal{M} | X_i \in T} (\frac{1}{2} res_{\mathcal{M}/2}^{post}(Hij_{|T}) + \frac{1}{2} res_{\mathcal{M}/2}^{post}(Hik_{|T}))$$

En faisant cela, on transfère $res_{\mathcal{M}/2}^{post}$ de $\mathcal{M}/2$ sur $\mathcal{M}$. Puis on définit $\bar{u}'_{post} \in \tilde{V}_{\mathcal{M}}$ tel que pour tout entier $i \in [0, N-1]$,

$$a(\bar{u}'_{post}, \tilde{\phi}_i^{\mathcal{M}}) = (\mathcal{R}_{\mathcal{M}/2 \to \mathcal{M}} res_{\mathcal{M}/2}^{post})(X_i)$$

et on suppose que $\bar{u}'_{post} \approx \Pi_{\mathcal{M}} u_{\mathcal{M}/2} - u_{\mathcal{M}}$. On utilise ensuite l'interpolation de Richardson sur $u - u_{\mathcal{M}}$ et sur $u - u_{\mathcal{M}/2}$:

$$u - u_{\mathcal{M}} = h^2 u_2 + O(h^3)$$

$$et$$

$$u - u_{\mathcal{M}/2} = u_2 \left(\frac{h}{2}\right)^2 + O\left(\left(\frac{h}{2}\right)^3\right) = \frac{1}{4}h^2 u_2 + O(h^3)$$

On en déduit

$$u_{\mathcal{M}/2} - u_{\mathcal{M}} = (u - u_{\mathcal{M}}) - (u - u_{\mathcal{M}/2}) = h^2 u_2 - \frac{1}{4}h^2 u_2 + O(h^3) = \frac{3}{4}h^2 u_2 + O(h^3)$$

Cela nous donne

$$u_{\mathcal{M}/2} - u_{\mathcal{M}} = \frac{3}{4}(u - u_{\mathcal{M}}) + O(h^3)$$

$$u - u_{\mathcal{M}} = \frac{4}{3}(u_{\mathcal{M}/2} - u_{\mathcal{M}}) + O(h^3)$$

donc

$$u - u_{\mathcal{M}} \approx \frac{4}{3}(u_{\mathcal{M}/2} - u_{\mathcal{M}})$$

et donc

$$\Pi_{\mathcal{M}} u - u_{\mathcal{M}} = \Pi_{\mathcal{M}}(u - u_{\mathcal{M}}) \approx \Pi_{\mathcal{M}}\left(\frac{4}{3}(u_{\mathcal{M}/2} - u_{\mathcal{M}})\right)$$

$$\approx \frac{4}{3}\left(\Pi_{\mathcal{M}} u_{\mathcal{M}/2} - u_{\mathcal{M}}\right) \approx \frac{4}{3}\bar{u}'_{post}.$$

On définit $[\mathbf{rhsPostApprox}] \in R^N$ tel que, pour tout entier $i \in [0; N-1]$,

$$[\mathbf{rhsPostApprox}]_i = \mathcal{R}_{\mathcal{M}/2 \to \mathcal{M}} res^{post}_{\mathcal{M}/2}(i)$$

et on écrit

$$[\tilde{\mathbf{A}}_{\mathcal{M}}][\bar{\mathbf{u}}'_{post}] \approx [\mathbf{rhsPostApprox}].$$

Définition : *Le correcteur Defect Correction est défini par:*

$$[\tilde{\boldsymbol{A}}_{\mathcal{M}}][\bar{\boldsymbol{u}}'_{post}] = \mathcal{R}_{\mathcal{M}/2 \to \mathcal{M}} res^{post}_{\mathcal{M}/2}(i).$$

Précisons comment ce champ est calculé en pratique. On résout l'équation exactement comme dans le cas *a priori* et on obtient $\bar{u}'_{post} \approx \frac{3}{4}\left(\Pi_{\mathcal{M}} u - u_{\mathcal{M}}\right)$. Faisons apparaître l'erreur d'interpolation:

$$u - u_{\mathcal{M}} = (\Pi_{\mathcal{M}} u - u_{\mathcal{M}}) - (\Pi_{\mathcal{M}} u - u)$$

$$u - u_{\mathcal{M}} \approx u'_{prio} = \frac{4}{3}\bar{u}'_{prio} - errQuad_{\mathcal{M},u}$$

Nous voulons calculer le second membre $[\mathbf{rhsPostApprox}]$. Pour tout entier $i \in [0, N-1]$,

$$[\mathbf{rhsPostApprox}]_i = res^{post}_{\mathcal{M}/2}(\mathbf{x}_i) + \sum_{T \in \mathcal{M} | X_i \in T} \left(\frac{1}{2} res^{post}_{\mathcal{M}/2}(Hij_{|T}) + \frac{1}{2} res^{post}_{\mathcal{M}/2}(Hik_{|T})\right)$$

avec $\forall \mathbf{x}_j \in \mathcal{M}/2$, $res^{post}_{\mathcal{M}/2}(\mathbf{x}_j) = (f, \tilde{\phi}_j^{\mathcal{M}/2}) - a(u_{\mathcal{M}}, \tilde{\phi}_j^{\mathcal{M}/2})$.

Chaque $(f, \tilde{\phi}_j^{\mathcal{M}/2})$ est calculé grâce à une quadrature de Gauss.

$$a(u_{\mathcal{M}}, \tilde{\phi}_j^{\mathcal{M}/2}) = \int_{\Omega} \mu(\mathbf{x}) \nabla u_{\mathcal{M}}(\mathbf{x}).\nabla \tilde{\phi}_j^{\mathcal{M}/2}(\mathbf{x}) \, \mathrm{d}\mathbf{x}$$

$$a(u_{\mathcal{M}}, \tilde{\phi}_j^{\mathcal{M}/2}) = \sum_{T \in \mathcal{M}}^{nbElementsFins} \int_{T} \mu(\mathbf{x}) \nabla u_{\mathcal{M}}(\mathbf{x}).\nabla \tilde{\phi}_i^{\mathcal{M}/2}(\mathbf{x}) \, \mathrm{d}\mathbf{x}$$

μ est considérée comme constante par élément et $u_{\mathcal{M}}$ et $\tilde{\phi}_i^{\mathcal{M}/2}$ sont linéaires par élement donc cette somme est facile à calculer. Par conséquent, on sait calculer chacun des $a(u_{\mathcal{M}}, \tilde{\phi}_j^{\mathcal{M}/2})$ et $(f, \tilde{\phi}_j^{\mathcal{M}/2})$ et, donc, [**rhsPostApprox**]. À partir de là, on peut calculer u'_{post}.

Au total, le calcul du correcteur Defect Correction proposé coûte l'assemblage sur un maillage plus fin d'un résidu, suivi d'une résolution du système linéarisé sur le maillage courant.

Conclusion du chapitre

Ce chapitre a planté le décor du modèle continu choisi et de sa discrétisation. Il a rappelé les détails clé des estimations d'erreur existantes. Tout le reste de la thèse fera référence à une analyse dans L^2 et à l'ordre deux prolongeant l'estimation de type Aubin-Nitsche. Nous avons aussi proposé d'aller plus loin que la correction de fonctionnelles en corrigeant le champ de solution approché par une méthode Defect-Correction. Cette dernière approche sera au centre des formulations d'adaptation orientées norme.

Chapter 3

Méthode de résolution multigrille avec critère d'arrêt

3.1 Introduction

La taille de l'espace de discrétisation détermine en grande partie le coût calcul d'une simulation. À taille égale, l'effort de calcul, consommé en pratique dans la phase de résolution du problème discret, est conditionné par l'efficacité de l'algorithme de résolution du ou des systèmes discrets. Les systèmes des modèles haute fidélité en mécanique des milieux continus sont des systèmes *de grande taille* et souvent non-linéaires. Il en résulte que leur traitement par ordinateur parallèle utilise la plupart du temps un algorithme itératif, jugé plus efficace pour des grands systèmes linéaires et occupant moins de mémoire. On analyse la convergence des algorithmes itératifs en examinant le comportement de l'*erreur itérative ou de résolution*, différence entre l'itéré et sa limite, point fixe de l'algorithme. Les méthodes de résolution itératives comme les méthodes de Jacobi ou de Gauss-Seidel permettent de faire disparaître en quelques itérations les composantes hautes fréquences de l'erreur. Malheureusement, ces méthodes sont très inefficaces pour ce qui concerne les composantes basses fréquences.
Les méthodes multigrilles (MG) permettent de régler ce problème. En effet, des composantes basses fréquences sur une grille fine deviennent des composantes hautes fréquences sur une grille plus grossière. On peut alors les faire disparaître en itérant sur cette grille. Ces idées ont inspiré les premiers travaux sur les multigrilles qui remontent à Fedorenko [37] et Bakhvalov [10].
Depuis, ces méthodes ont connus d'importants développements théoriques (voir les livres de Hackbusch [49] et de Wesseling [82]) ainsi que méthodologiques (voir [18] et le livre synthétique de Trottenberg, Oosterlee et Schuller [76].
Grâce â la combinaison des MG avec les raffinements successifs (*nested iteration*) les multigrilles FMG sont de complexité optimale, produisant N résultats en KN opérations.

Dans cette thèse, nous combinons de manière originale les méthodes FMG avec les nouvelles méthodes d'adaptation de maillage. Dans ce but nous introduisons une nouvelle méthode de critère d'arrêt pour FMG. Cette méthode est la contribution originale de ce chapitre.

La spécification d'un critère d'arrêt pour une méthode de résolution itérative est

une question assez sensible, un critère insuffisamment sévère pouvant donner une solution itérée trop différente de la solution discrète cherchée. Un critère trop sévère renchérit le coût calcul d'un facteur qui peut dégrader la complexité de l'algorithme. Le critère d'arrêt devrait atteindre la situation idéale où l'erreur de résolution, écart entre l'itéré et sa limite le point fixe, est plus petite que l'écart entre ce point fixe et la solution continue de l'EDP. Mais cette propriété est, elle aussi, coûteuse à contrôler. Dans le cas des méthodes FMG, une théorie est proposée, par exemple dans [49]. Elle ne va pas à notre sens au bout du chemin dans la mesure où elle suppose que la convergence des cycles MG est idéalement fixe. Notons par ailleurs des contributions focalisées sur des méthodes quasi-Newton de type gradient conjugué ou GMRES, cf. par exemple [61]. Ces méthodes supposent que l'approximation est dans sa phase asymptotique (erreur d'approximation quatre fois plus petite sur maillage deux fois plus fin), ce qui reste difficile ou coûteux à vérifier, et inadaptable en FMG si FMG débute avec des maillages plutôt grossiers.

Avant d'expliquer ce nouveau critère d'arrêt, nous nous intéressons au fonctionnement et aux propriétés des méthodes bigrille, multigrille et Full-Multigrid sur maillages emboités ainsi qu'aux méthodes à employer sur des maillages non-emboités.

3.2 MG

3.2.1 Algorithme bi-grille

Un algorithme multigrille repose sur une boucle récursive. Pour en comprendre le principe, on se propose d'étudier l'algorithme bi-grille, décrit par Peter Wesseling [82].

On reprend notre équation aux dérivées partielles discrétisée sur un maillage/métrique $\mathcal{M}$ de N nœuds et écrite sous forme matricielle:

$$\mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}} = \mathbf{f}_{\mathcal{M}}$$

On commence par la phase de pré-lissage, c'est-à-dire que l'on fait quelques itérations de la résolution de l'équation à l'aide d'une méthode itérative classique, par exemple celle de Jacobi.
On considère trois pas que l'on note $\frac{1}{3}$, $\frac{2}{3}$ et 1.
On part de $\mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}} = \mathbf{f}_{\mathcal{M}}$ et on pose:

$$\mathbf{v}_{\mathcal{M}} = \mathbf{S}_{\mathcal{M}}\mathbf{u}^0_{\mathcal{M}} + \mathbf{D}^{-1}_{\mathcal{M}}\mathbf{f}_{\mathcal{M}},$$

avec $\mathbf{S}_{\mathcal{M}} = \mathbf{D}^{-1}_{\mathcal{M}}\mathbf{N}_{\mathcal{M}}$ et $\mathbf{D}_{\mathcal{M}} - \mathbf{N}_{\mathcal{M}} = \mathbf{A}_{\mathcal{M}}$, et où $\mathbf{D}_{\mathcal{M}}$ est la diagonale de $\mathbf{A}_{\mathcal{M}}$.

On applique ce schéma ν_1 fois et on obtient le résultat suivant:

$$\mathbf{u}_{\mathcal{M}}^{\frac{1}{3}} = \mathbf{S}_{\mathcal{M}}^{\nu_1}\mathbf{u}_{\mathcal{M}}^{0} + \mathbf{T}_{\mathcal{M}}(\nu_1)\mathbf{f}_{\mathcal{M}}$$

avec $\mathbf{T}_{\mathcal{M}}(\nu_1) = (\mathbf{S}_{\mathcal{M}}^{\nu_1-1} + \mathbf{S}_{\mathcal{M}}^{\nu_1-2} + ... + \mathbf{I}_{\mathcal{M}})\mathbf{M}_{\mathcal{M}}^{-1}$.
On définit alors l'erreur et le résidu, donnés par les formules suivantes:

$$\mathbf{e}_{\mathcal{M}}^{\frac{1}{3}} = \mathbf{u}_{\mathcal{M}} - \mathbf{u}_{\mathcal{M}}^{\frac{1}{3}}$$

et

$$\mathbf{r}_{\mathcal{M}}^{\frac{1}{3}} = \mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{\frac{1}{3}} - \mathbf{f}_{\mathcal{M}}.$$

On connait $\mathbf{r}_{\mathcal{M}}^{\frac{1}{3}}$ et on veut calculer $\mathbf{e}_{\mathcal{M}}^{\frac{1}{3}}$. On remarque que l'on a:

$$\mathbf{A}_{\mathcal{M}}\mathbf{e}_{\mathcal{M}}^{\frac{1}{3}} = \mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}} - \mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{\frac{1}{3}} = \mathbf{f}_{\mathcal{M}} - \mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{\frac{1}{3}} = -\mathbf{r}_{\mathcal{M}}^{\frac{1}{3}}.$$

Pour résoudre cette équation d'inconnue $\mathbf{e}_{\mathcal{M}}^{\frac{1}{3}}$, on change de grille. Les données sont transférées sur une grille plus grossière notée $\bar{\mathcal{M}}$. Les transferts inter-grilles se font grâce à des opérateurs de prolongement et de restriction notés $P_{\bar{\mathcal{M}}\to\mathcal{M}}$ et $R_{\mathcal{M}\to\bar{\mathcal{M}}}$.
Il existe plusieurs possibilités pour définir $P_{\bar{\mathcal{M}}\to\mathcal{M}}$ et $R_{\mathcal{M}\to\bar{\mathcal{M}}}$. Voici un bon exemple appliqué au cas 1D.
Considérons une grille grossière $\bar{\mathcal{M}}$ de $\bar{N}$ sommets et une grille fine $\mathcal{M}$ de $N = 2\bar{N}-1$ sommets.
Pour un vecteur $\mathbf{u}$ sur la grille fine $\mathcal{M}$, on note $\bar{\mathbf{u}}$ sa restriction sur la grille grossière $\bar{\mathcal{M}}$ et on définit $P_{\bar{\mathcal{M}}\to\mathcal{M}}$ et $R_{\mathcal{M}\to\bar{\mathcal{M}}}$ par les formules suivantes:

$$(P_{\bar{\mathcal{M}}\to\mathcal{M}}\bar{\mathbf{u}})_{2j} = \bar{\mathbf{u}}_j$$
$$(P_{\bar{\mathcal{M}}\to\mathcal{M}}\bar{\mathbf{u}})_{2j+1} = \frac{1}{2}(\bar{\mathbf{u}}_j + \bar{\mathbf{u}}_{j+1})$$

où l'indice j varie de 0 à $\bar{N}-1$ et

$$(R_{\mathcal{M}\to\bar{\mathcal{M}}}\mathbf{u})_0 = \frac{1}{2}\mathbf{u}_0 + \frac{1}{4}\mathbf{u}_1$$
$$(R_{\mathcal{M}\to\bar{\mathcal{M}}}\mathbf{u})_{\bar{N}-1} = \frac{1}{4}\mathbf{u}_{2\bar{N}-3} + \frac{1}{2}\mathbf{u}_{2\bar{N}-2}$$
$$(R_{\mathcal{M}\to\bar{\mathcal{M}}}\mathbf{u})_j = \frac{1}{4}\mathbf{u}_{2j-1} + \frac{1}{2}\mathbf{u}_{2j} + \frac{1}{4}\mathbf{u}_{2j+1}$$

où l'indice j varie de 1 à $\bar{N}-1$.

Il est important de remarquer que $\bar{\mathbf{u}} = (R_{\mathcal{M}\to\bar{\mathcal{M}}}\mathbf{u})$ mais que $\mathbf{u} \neq (P_{\bar{\mathcal{M}}\to\mathcal{M}}\bar{\mathbf{u}})$.
La figure 3.1 montre un exemple de ces transferts.

On cherche à présent $\bar{\mathbf{e}}_{\bar{\mathcal{M}}}$ tel que $P_{\bar{\mathcal{M}}\to\mathcal{M}}\bar{\mathbf{e}}_{\bar{\mathcal{M}}} = \mathbf{e}_{\mathcal{M}}^{\frac{1}{3}}$. On a $\mathbf{A}_{\mathcal{M}}\mathbf{e}_{\mathcal{M}}^{\frac{1}{3}} = -\mathbf{r}_{\mathcal{M}}^{\frac{1}{3}}$ donc $\mathbf{A}_{\mathcal{M}}P_{\bar{\mathcal{M}}\to\mathcal{M}}\bar{\mathbf{e}}_{\bar{\mathcal{M}}} = -\mathbf{r}_{\mathcal{M}}^{\frac{1}{3}}$ donc

$$R_{\mathcal{M}\to\bar{\mathcal{M}}}\mathbf{A}_{\mathcal{M}}P_{\bar{\mathcal{M}}\to\mathcal{M}}\bar{\mathbf{e}}_{\bar{\mathcal{M}}} = -R_{\mathcal{M}\to\bar{\mathcal{M}}}\mathbf{r}_{\mathcal{M}}^{\frac{1}{3}}.$$

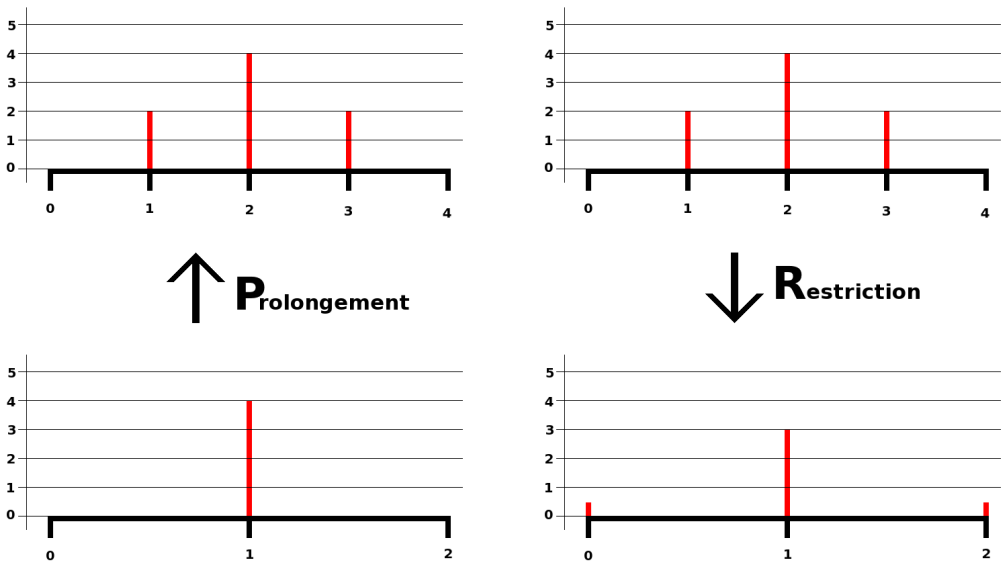

Figure 3.1: Exemples de prolongement et de restriction entre deux maillages de tailles 3 et 5

Ainsi, on peut écrire

$$\bar{\mathbf{A}}_{\bar{\mathcal{M}}}\bar{\mathbf{e}}_{\bar{\mathcal{M}}} = \bar{\mathbf{f}}_{\bar{\mathcal{M}}}$$

avec $\bar{\mathbf{A}}_{\bar{\mathcal{M}}} = R_{\mathcal{M}\to\bar{\mathcal{M}}}\mathbf{A}_{\mathcal{M}}P_{\bar{\mathcal{M}}\to\mathcal{M}}$ et $\bar{\mathbf{f}}_{\bar{\mathcal{M}}} = -R_{\mathcal{M}\to\bar{\mathcal{M}}}\mathbf{r}^{\frac{1}{3}}_{\mathcal{M}}$.
On suppose que l'on peut facilement résoudre cette équation et trouver une bonne approximation $\tilde{\mathbf{e}}_{\bar{\mathcal{M}}}$ de $\bar{\mathbf{e}}_{\bar{\mathcal{M}}}$. Dans le cas contraire, on utilise encore d'autres grilles et ce n'est donc plus le cas de la méthode bi-grille. On prolonge $\tilde{\mathbf{e}}_{\bar{\mathcal{M}}}$ sur la grille fine en utilisant $P_{\bar{\mathcal{M}}\to\mathcal{M}}$ et on trouve $P_{\bar{\mathcal{M}}\to\mathcal{M}}\tilde{\mathbf{e}}_{\bar{\mathcal{M}}}$ qui est une approximation de $\mathbf{e}^{\frac{1}{3}}_{\mathcal{M}}$ puisque $P_{\bar{\mathcal{M}}\to\mathcal{M}}\bar{\mathbf{e}}_{\bar{\mathcal{M}}} = \mathbf{e}^{\frac{1}{3}}_{\mathcal{M}}$. On peut ainsi écrire:

$$\mathbf{u}^{\frac{2}{3}}_{\mathcal{M}} = \mathbf{u}^{\frac{1}{3}}_{\mathcal{M}} + P_{\bar{\mathcal{M}}\to\mathcal{M}}\tilde{\mathbf{e}}_{\bar{\mathcal{M}}}.$$

On peut ensuite appliquer le post-lissage c'est-à-dire appliquer à nouveau la méthode itérative utilisée dans le pré-lissage mais en prenant $u^{\frac{2}{3}}$ comme valeur initiale. On fait cela un nombre donné de fois ν_2. On trouve ainsi:

$$\mathbf{u}^{1}_{\mathcal{M}} = \mathbf{S}^{\nu_2}_{\mathcal{M}}\mathbf{u}^{\frac{2}{3}}_{\mathcal{M}} + \mathbf{T}_{\mathcal{M}}(\nu_2)\mathbf{f}_{\mathcal{M}}.$$

On réitère ensuite ce procédé jusqu'à trouver une approximation de u suffisamment bonne.

3.2.2 Algorithme multigrille

L'algorithme multigrille s'appuie sur l'algorithme bi-grille en utilisant un principe de récursivité. Si on considère l'algorithme bi-grille, on constate qu'il y a une étape

où l'on doit résoudre l'équation $\bar{A}\bar{e} = \bar{f}$ sur la grille grossière. Pour cela, on peut appliquer encore une fois l'algorithme bi-grille en prenant en considération une autre grille encore plus grossière. Il faudra ensuite résoudre une autre équation similaire sur cette nouvelle grille, ce que l'on pourra faire en utilisant encore une autre grille et ainsi de suite jusqu'à ce que l'on ait une grille suffisamment grossière pour y résoudre facilement l'équation. Il ne reste plus ensuite qu'à transférer les résultats sur la grille fine tout en effectuant, éventuellement, des post-lissages.

3.2.3 Critère de convergence des méthodes multigrilles

On étudie le cas sans post-lissage c'est-à-dire le cas où ν_2 =0. On a alors

$$\mathbf{e}^1_{\mathcal{M}} = \mathbf{Q}_{\mathcal{M}}\mathbf{e}^0_{\mathcal{M}}$$

avec:

$$\mathbf{Q}_{\mathcal{M}} \quad = (\mathbf{A}^{-1}_{\mathcal{M}} - P_{\bar{\mathcal{M}}\to\mathcal{M}}\bar{\mathbf{A}}^{-1}_{\bar{\mathcal{M}}}R_{\mathcal{M}\to\bar{\mathcal{M}}})(\mathbf{A}_{\mathcal{M}}\mathbf{S}^{\nu_1}_{\mathcal{M}}). \tag{3.1}$$

3.2.4 Propriété de lissage

Définition: La propriété de lissage est respectée s'il existe une constante C_S et une fonction $\eta(\nu)$ indépendantes de la taille h des mailles de la grille fine telles que $||\mathbf{A}_{\mathcal{M}}\mathbf{S}^{\nu}_{\mathcal{M}}|| \leq C_S h^{-2m}\eta(\nu)$, avec $\eta(\nu) \underset{\nu\to+\infty}{\longrightarrow} 0$ où $2m$ est l'ordre de l'équation aux dérivées partielles à résoudre. Cela suppose que l'ordre est pair, le cas où il est impair n'étant pas étudié ici.
Notons que cette propriété implique que la méthode de lissage soit une méthode itérative convergente puisque cela donne:

$$||\mathbf{S}^{\nu}_{\mathcal{M}}|| \leq ||\mathbf{A}^{-1}_{\mathcal{M}}||||\mathbf{A}_{\mathcal{M}}\mathbf{S}^{\nu}_{\mathcal{M}}|| \leq ||\mathbf{A}^{-1}_{\mathcal{M}}||C_S h^{-2m}\eta(\nu).$$

3.2.5 Propriété d'approximation

Définition: La propriété d'approximation est respectée s'il existe une constante C_A indépendante de h telle que
$||\mathbf{A}^{-1}_{\mathcal{M}} - P_{\bar{\mathcal{M}}\to\mathcal{M}}\bar{\mathbf{A}}^{-1}_{\bar{\mathcal{M}}}R_{\mathcal{M}\to\bar{\mathcal{M}}}|| \leq C_A h^{2m}$.

3.2.6 Convergence

Proposition: Si les propriétés de lissage et d'approximation sont vérifiées, alors il existe un nombre $\bar{\nu}$ indépendant de h tel que

$$||\mathbf{Q}_{\mathcal{M}}|| \leq C_S C_A \eta(\nu) < 1, \forall \nu \geq \bar{\nu}$$

et la méthode MG est convergente. Le démonstration peut être trouvée dans [49].

3.2.7 Complexité des méthodes multigrilles

L'erreur itérative est donnée par la formule $|u_{\mathcal{M}}^{nbcy} - u_{\mathcal{M}}^{\infty}| < \eta^{nbcy}|u_{\mathcal{M}}^{0} - u_{\mathcal{M}}^{\infty}|$, η étant le facteur de réduction et $nbcy$ le nombre de cycles effectués. On veut obtenir l'erreur d'approximation $\varepsilon(N) = N^{\frac{-\alpha}{d}}$ c'est-à-dire avoir

$$\eta^{nbcy}|u_{\mathcal{M}}^{0} - u_{\mathcal{M}}^{\infty}| = N^{\frac{-\alpha}{d}}$$

ce qui s'écrit:

$$\exp(ln(|u_{\mathcal{M}}^{0} - u_{\mathcal{M}}^{\infty}|))\exp(nbcy \times ln(\eta)) = exp(\frac{-\alpha}{d}ln(N))$$

$$ln(|u_{\mathcal{M}}^{0} - u_{\mathcal{M}}^{\infty}|) + nbcy \times ln(\eta) = \frac{-\alpha}{d}ln(N)$$

$$nbcy = O(ln(N)).$$

Chaque cycle a une complexité de $N(1 + \frac{1}{2} + \frac{1}{4} + ...)$.

Or, $1 < (1 + \frac{1}{2} + \frac{1}{4} + ...) < 2$ donc $N(1 + \frac{1}{2} + \frac{1}{4} + ...) = O(N)$.

Ainsi, chaque cycle a une complexité de $O(N)$ et on a $nbcy = O(ln(N))$ donc l'ensemble de l'algorithme a une complexité de $O(N) \times O(ln(N)) = O(Nln(N))$.

3.3 FMG

3.3.1 Description de la méthode Full-Multigrid

Quand on utilise une méthode itérative, on est obligé de partir d'une solution initiale u_0 qui peut être plus ou moins éloignée de la solution exacte que l'on recherche. La méthode Full-Multigrid, ou multigrilles complètes, consiste à résoudre tout d'abord le système sur une grille grossière, puis à transférer la solution sur une grille plus fine par interpolation et à utiliser la solution ainsi obtenue comme solution initiale d'une méthode multigrille et ainsi de suite jusqu'à la grille la plus fine, cf. schéma de la Figure 3.2. En d'autres termes, au lieu de prendre une solution initiale quelconque, on prend une solution déjà calculée sur une grille plus grossière. Cette méthode est décrite et analysée dans le cas d'une approximation d'ordre deux par Eric Morano dans sa thèse [65].

On part de la grille la plus grossière l. On interpole la solution calculée sur l sur la grille $l-1$. On applique une phase multigrille sur plusieurs cycles en utilisant les grilles l et $l-1$. On trouve une solution sur $l-1$, on l'interpole sur $l-2$ et on lance une nouvelle phase multigrille en utilisant, l, $l-1$ et $l-2$. On continue à procéder de la sorte jusqu'à ce que l'on trouve la solution sur la grille la plus fine.

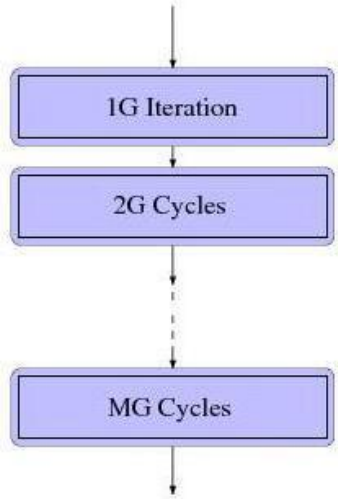

Figure 3.2: Chaque phase de numéro k_ϕ de l'algorithme Full Multi-Grid comprend quelques cycles de k_ϕ-grille

On a toujours l'équation $Au = f$. On la discrétise et on la résout sur la grille grossière $\mathcal{H}_l$. Cela nous donne:

$$\mathbf{A}_{\mathcal{H}_l}\mathbf{u}_{H_l} = \mathbf{f}_{\mathcal{H}_l}$$

et

$$\mathbf{u}_{\mathcal{H}_l} = \mathbf{A}_{\mathcal{H}_l}^{-1}\mathbf{f}_{\mathcal{H}_l}.$$

On transfère ce résultat sur la grille $\mathcal{H}_{l-1}$ et on le prend comme solution initiale:

$$\mathbf{u}^0_{\mathcal{H}_{l-1}} = \mathbf{P}_{\mathcal{H}_l \longrightarrow \mathcal{H}_{l-1}}\mathbf{u}_{\mathcal{H}_l}.$$

Pour cette condition initiale, on applique n fois un schéma multigrille à l'équation $\mathbf{A}_{\mathcal{H}_{l-1}}\mathbf{u}_{\mathcal{H}_{l-1}} = \mathbf{f}_{\mathcal{H}_{l-1}}$ avec $\mathcal{H}_{l-1}$ comme grille fine et $\mathcal{H}_l$ comme grille grossière. On calcule ainsi $\mathbf{u}^n_{\mathcal{H}_{l-1}}$.
On pose $\mathbf{u}^0_{\mathcal{H}_{l-2}} = \mathbf{P}_{\mathcal{H}_{l-1} \longrightarrow \mathcal{H}_{l-2}}\mathbf{u}^n_{\mathcal{H}_{l-1}}$ et on continue ainsi jusqu'à trouver $\mathbf{u}^n_{\mathcal{H}_0}$ où $\mathcal{H}_0$ est la grille la plus fine considérée.

À la phase k_{phase}, le nombre de nœuds du maillage le plus fin de la phase est noté $N_{k_{phase}}$. On a choisi des maillages de sorte qu'à chaque nouvelle phase, le nouveau maillage ait une taille de maille deux fois inférieure à celle du maillage précédent. En terme de nombre de points, cela implique:

$$N_{k_{phase}+1} = 2^{dim}\ N_{k_{phase}}. \tag{3.2}$$

A l'ordre 2, l'erreur d'approximation qui en résulte sera 4 fois plus petite. Si, au contraire, on choisit un nouveau maillage de sorte que $N_{k_{phase}+1} = 2\ N_{k_{phase}}$, l'erreur en 2D sera seulement 2 fois plus petite et le nombre de cycles nécessaire

sera aussi 2 fois plus petit. Par contre, deux phases seraient nécessaires pour la même précision, ce qui nécessiterait un effort CPU global similaire. *Par conséquent, l'efficacité de l'algorithme ne dépend pas sensiblement du choix de* $N_{k_{phase}+1}/N_{k_{phase}}$.

Algorithme Full-Multigrid:
- résoudre sur le maillage le plus fin avec une résolution 1-grille,
- interpoler la solution sur un deuxième maillage légèrement plus fin,
- résoudre sur le deuxième maillage avec une résolution 2-grille,
- interpoler la nouvelle solution sur un troisième maillage encore plus fin,
- résoudre sur le troisième maillage avec une résolution 3-grille,
- etc...

3.3.2 Propriétés de la méthode Full-Multigrid

Les propriétés de lissage et d'approximation déjà nécessaires pour les méthodes multigrilles ont pour conséquence que l'algorithme est une contraction pour l'erreur itérative.
On définit les opérateurs de prolongement et de restriction. Pour cela, considérons une grille $\mathcal{H}_l$ et une grille deux fois plus grossière $\mathcal{H}_{l+1}$. L'opérateur de prolongement $P_{\mathcal{H}_{l+1}\to\mathcal{H}_l}$ est l'interpolation $P1$ des fonctions affines par élément sur $\mathcal{H}_{l+1}$ vers les fonctions affines par éléments sur $\mathcal{H}_l$. In versement, l'opérateur de restriction $R_{\mathcal{H}_l\to\mathcal{H}_{l+1}}$ est l'accumulation des fonctions fonctions affines par éléments sur $\mathcal{H}_l$ vers les fonctions affines par éléments sur $\mathcal{H}_{l+1}$.

Hypothèse 1: On suppose que le facteur de contraction est unique pour tous les cycles et toutes les phases, et égal à S.

La méthode Full-Multigrid a besoin en plus de la propriété de convergence numérique suivante:

Hypothèse 2: $||u^{k_{phase}+1} - P_{\mathcal{H}_0^{k_{phase}}\to\mathcal{H}_0^{k_{phase}+1}} u^{k_{phase}}|| \leq C_1 h^{k_{phase}+1}$ pour chaque phase k_{phase}
avec $\mathcal{H}_0^{k_{phase}}$ et $\mathcal{H}_0^{k_{phase}+1}$ les maillages les plus fins respectivement de la phase k_{phase} et de la phase $k_{phase}+1$.

On prend $Au = f$ avec A elliptique. On considère $(\mathcal{H}, ||.||)$ l'espace de Hilbert engendré par A, avec $||.||$ le produit scalaire associé. On obtient alors le problème variationnel associé à $Au = f$:

$$(Au, v) = (f, v), \forall v \in \mathcal{H}.$$

On suppose l'existence de sous-espaces éléments finis de $\mathcal{H}$ notés $\mathcal{H}_1$, $\mathcal{H}_2$..., $\mathcal{H}_l$, chacun d'entre eux servant à définir une grille.

Soit r_k l'opérateur de restriction de $\mathcal{H}$ sur $\mathcal{H}_k$. On définit l'erreur de discrétisation: $r_k u - u_k$. On a alors $r_{k-1} = Rr_k$ et on définit l'erreur d'interpolation

$PRu_k - u_k$. Remarquons que R et P dépendent de k même si, traditionnellement, ils sont notés sans indice k.

$$\text{Si} \begin{cases} ||r_k u - u_k|| = O(h_k^\kappa), \text{ avec } \kappa \text{ l'ordre de consistance} \\ |P|| = O(1) \\ ||R|| = O(1) \\ ||PRu_k - u_k|| = O(h_k^\tau), \text{ avec } \tau \text{ l'ordre d'interpolation} \end{cases}$$

alors il existe une constante réelle C_1 telle que

$$||Pu_{k-1} - u_k|| \leq C_1 k_k^{min(\kappa,\tau)}.$$

Définissons maintenant le rapport de précision C_2 entre les solutions u_k et u_{k-1}:

$$C_2 = \max_{1 \leq k \leq 1} (h_{k-1}/h_k)^\kappa$$

où $r = \max_{1 \leq k \leq 1}(h_{k-1}/h_k)$ est le rapport de maille entre les deux maillages. Soit i le numéro de l'itération pour laquelle on obtient une solution $\tilde{u_k}$ quand on applique un schéma multigrille au niveau k. On a:

$$||\tilde{u}_k - u_k|| \leq C_3 C_1 h_k^\kappa$$

avec $C_3 = \frac{S^i}{1 - C_2 S^i}$.

Si on a $||\tilde{u}_k - u_k|| \leq C_3 C_1 h_k^\kappa$ et $C_2 S^i \leq \frac{1}{2}$, alors $||\tilde{u}_k - u_k|| \leq C_1 h_k^\kappa$.

On déduit de cela le nombre de cycles i nécessaire pour atteindre l'erreur de discrétisation spatiale:

$$i \geq -\frac{ln(2C_2)}{ln(S)}.$$

Notons que la condition $C_2 S^i \leq \frac{1}{2}$ exprime le fait qu'après i contractions MG de rapport S, on doit avoir réduit l'erreur d'un facteur $2\ C_2 = 2\ r^\kappa$.

Dans le cas d'approximations d'ordre deux qui nous intéresse, et avec un rapport de maillage classique de 2 entre deux niveaux successifs, cela signifie que le facteur de réduction à chaque cycle doit être d'au moins $2\ C_2 = 8$.

3.3.3 Complexité de la méthode Full-Multigrid

On commence l'algorithme sur la grille la plus grossière et, à chaque phase multigrille, on part d'une erreur de $(\frac{N}{2})^{\frac{-\alpha}{d}}$ et on veut la faire passer à $N^{\frac{-\alpha}{d}}$, c'est-à-dire la diviser par $2^{\frac{\alpha}{d}}$. On veut donc

$$\eta^{nbcy} = 2^{\frac{-\alpha}{d}}$$

$$nbcy \times ln(\eta) = \frac{-\alpha}{d} ln(2)$$

$$nbcy = \frac{-\alpha}{d} ln(2 - \eta) = O(1).$$

On fait cela pour chaque phase multigrille. La complexité vaut alors:
$nbcy \times 1 + nbcy \times 2(1 + \frac{1}{2}) + nbcy \times 4(1 + \frac{1}{2} + \frac{1}{4}) + ...$

$$= \sum_{i=0}^{log_2(N)} nbcy \times 2^i \sum_{j=0}^{i} (\frac{1}{2})^j$$

$$= \sum_{i=0}^{log_2(N)} 2^i nbcy \times \frac{1-(\frac{1}{2})^{i+1}}{1-\frac{1}{2}}$$

$$= \sum_{i=0}^{log_2(N)} 2^{i+1} nbcy \times (1 - (\frac{1}{2})^{i+1})$$

$$= nbcy \sum_{i=0}^{log_2(N)} (2^{i+1} - 1)$$

$$= nbcy(2 \sum_{i=0}^{log_2(N)} 2^i - \sum_{i=0}^{log_2(N)} 1)$$

$$= nbcy(2 \times \frac{2^{1+log_2(N)}-1}{2-1} - (1 + log_2(N)))$$
$$= nbcy(2(2 \times 2^{log_2(N)} - 1) - 1 - log_2(N)) = nbcy(4 \times 2^{log_2(N)} - 2 - 1 - log_2(N))$$
$$= nbcy(4N - 3 - log_2(N)) = O(1) \times O(N) = O(N).$$

Ce calcul montre, sous les hypothèses 1 et 2, la complexité optimale $O(N)$ du FMG tel que défini dans ce chapitre et utilisé dans les prochains chapitres.

3.3.4 Synthèse sur la complexité FMG

Nous avons rappelé l'état de l'art dans l'analyse de FMG. En pratique, on contrôle le nombre de cycles par phase en le forçant à une valeur fixe. Mais les deux hypothèses 1 et 2 ne sont pas systèmatiquement vérifiées en pratique:
- le facteur de contraction n'est pas toujours constant,
- la propriété de convergence n'est pas toujours vraie.
Il peut en résulter un échec de la stratégie FMG, caractérisé par un résultat final entâché d'une importante erreur itérative.

Une discussion des échecs de FMG et une tentative assez complexe d'y remédier à l'aide d'un critère d'erreur multi-niveau peuvent être trouvées dans [24]. Notre proposition de critère d'arrêt, assez différente, est décrite plus loin.

3.4 Maillages non-emboités

Jusqu'à présent, nous nous sommes intéressés aux maillages emboités, c'est-à-dire aux maillages tels que, à chaque niveau, les points de la grille grossière fassent partie des points de la grille fine. Il peut être intéressant d'étendre les méthodes multigrilles au cas des maillages non-emboités. Il faut donc définir les transferts entre les grilles. Pour cela, on utilise des opérateurs d'interpolation et d'accumulation basées sur des coordonnées barycentriques.

Pour changer de grille, pour commencer, on considère chaque point de la grille fine et on recherche le triangle de la grille grossière dans lequel il se trouve.
Pour passer de la grille grossière à la grille fine, on alloue, à chaque point de la grille fine, les valeurs aux trois sommets du triangle de la grille grossière correspondant, pondérées par les coordonnées barycentriques du point.
Pour passer de la grille fine à la grille grossière, on prend les valeurs à chaque point de la grille fine, on la divise par l'aire de la cellule du point puis on alloue sa valeur, pondérée par les coordonnées barycentriques, à chaque sommet du triangle de la grille grossière correspondant. Ensuite, pour chaque point de la grille grossière, on divise la valeur ainsi attribuée par l'aire de la cellule du point.
Les cellules en question sont délimitées par les segments formés par les milieux des arêtes associées au points et les centres de gravités des triangles voisins, cf. figure 3.3.

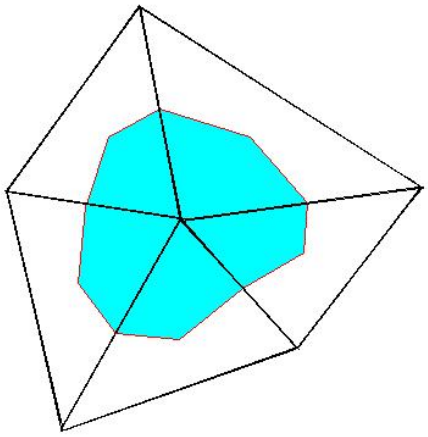

Figure 3.3: Cellule autour d'un point donné.

3.5 Un nouveau critère d'arrêt

On appelle *k_ϕ-ième phase FMG* la k_ϕ-ième phase d'itération utilisant le *cycle k_ϕ-grid*. Grâce à l'initialisation par la phase précédente, une convergence suffisante à chaque phase k_ϕ peut être obtenue par un petit nombre k_c de cycles k_ϕ-grid. Grâce à la propriété des multigrilles d'avoir un ordre de convergence quasi-indépendant du nombre de niveaux et de la taille de la grille, on peut obtenir un gain important. L'arrêt des itérations de la solution est généralement basé sur le critère suivant:

Critère d'arrêt: *L'erreur d'itération doit être plus petite que l'erreur d'approximation.*

Dans la théorie FMG présentée dans [49] et, pour des schémas d'approximation de second ordre, il existe une constante K telle que, pour toute phase k_ϕ, la solution

approchée u_{k_ϕ} calculée en fin de phase satisfasse:

$$||u - u_{k_\phi}|| \leq kh_{k_\phi}^2. \tag{3.3}$$

L'erreur pour u_{k_ϕ} est quatre fois plus grande que pour $u_{k_\phi - 1}$ ce qui suggère naïvement une réduction de résidu par quatre à chaque phase. L'analyse plus rigoureuse rappelée plus haut montre qu'une condition suffisante pour (3.3) est que chaque phase de cyclage FMG divise le résidu par un facteur plus grand que $2r^\kappa$, où $r = (N_{k_\phi+1}/N_{k_\phi})^{1/d}$ est le ratio (fixé) des tailles de maille entre deux phases, choisi à $r = 2$, d est la dimension spatiale et $\kappa = 2$ est l'ordre de précision du schéma. Cela donne $2r^\kappa = 8$. Notons $h = h_{k_\phi}$ la taille de maille à la phase k_ϕ et $u_h^{k_c}$ la solution itérée obtenue à cette phase après l'application de k_c cycles. En prenant une marge de sécurité, nous choisissons d'imposer à chaque phase une réduction d'un facteur 10:

$$||\mathbf{A}_h u_h^{k_c} - f_h|| \leq \frac{1}{10}||\mathbf{A}_h u_h^0 - f_h||. \tag{3.4}$$

La faiblesse de cette théorie réside dans l'hypothèse principale (3.3). Une première remarque évidente est que (3.3) est une hypothèse de taux convergence en maillage asymptotique, ce qui ne peut être établi que pour un maillage suffisamment fin. Cela est vrai pour des grilles ou niveaux ℓ où ℓ est suffisament grand. Au contraire, pour ℓ près de l'unité, le nombre de sommets est petit, le maillage est grossier, et l'estimation précédente est sans doute *fausse*. Une autre difficulté majeure survient quand (3.3) ne convient pas pour les dernières grilles fines. Dans le second cas, le dilemme du FMG est le suivant.

Si le test de convergence (3.4) est basé sur une décroissance fixe du résidu à chaque phase, alors, la convergence, n'étant pas suffisante, produira une mauvaise solution sur le maillage fin.

Si, au contraire, nous appliquons un critère d'arrêt trop sévère à chaque phase, alors nous aurons une bonne solution mais nous aurons augmenté le coût de calcul de façon imprévisible et probablement perdu la complexité optimale const.N du FMG.

Nous sommes aussi motivés par une seconde question en relation avec l'adaptation de maillage : si la boucle de convergence est appliquée, le cycle MG sera, à chaque nouvelle phase, initialisé par une solution obtenue avec un maillage de même finesse mais légèrement moins adapté. Cette solution peut être déjà très proche de la solution convergée. Ainsi, dans le but d'éviter un trop grand temps de calcul inutile, nous avons besoin de déterminer quand notre solution est assez bonne afin d'arrêter les calculs relativement tôt dans le cycle MG.

Les deux prochains paragraphes traitent de l'introduction d'une erreur d'approximation et du contrôle de l'erreur d'itération dans le FMG en vue d'améliorer la test d'arrêt du FMG et de consolider le précision du FMG.

3.5.1 Estimation de l'erreur d'approximation

Considérons $\mathcal{H}_\mathcal{M}$ le maillage courant sur lequel nous voulons résoudre itérativement le système discret $A_\mathcal{M}\mathbf{u}_\mathcal{M} = f_\mathcal{M}$.

Soit $u^k_{\mathcal{M}} = \sum \mathbf{u}^k_{\mathcal{M},i} N_i$ la fonction approchée à l'itération k (N_i étant la fonction de base éléments finis associée au nœud i). Pour obtenir une bonne approximation d'ordre 2, Arioli et collaborateurs proposent, dans [61], un critère d'arrêt basé sur

$$||\mathbf{A}_{\mathcal{M}} u^k_{\mathcal{M}} - f_{\mathcal{M}}||_{H^{-1}} \ \leq \ h^2 \ ||f_{\mathcal{M}}||_{H^{-1}}$$

où H^{-1} désigne l'espace dual de l'espace de Sobolev H^1_0.

En utilisant le facteur h^2, on suppose que la convergence numérique en maillage est proche de la convergence asymptotique, ce qui risque d'être souvent faux en pratique. On propose ici une méthode qui, *a priori*, ne suppose pas la convergence en maillage. Introduisons quelques notations. Soit $V = H^1_0(\Omega)$, Ω étant le domaine de calcul. Le système continu d'EDP s'écrit

$$Au = f \quad ou \quad u \in V \quad \forall\, \phi \in V \quad a(u, \phi) = (f, \phi)$$

avec

$$a(u, \phi) = \int_\Omega \frac{1}{\rho} \nabla u \cdot \nabla \phi \ \mathrm{d}x\mathrm{d}y$$

où ρ est un coefficient positif non uniforme qui peut être discontinu. Soit $V_{\mathcal{M}}$ l'espace usuel continu d'approximation éléments-finis P_1. La discrétisation éléments-finis s'écrit:

$$u_{\mathcal{M}} \in V_{\mathcal{M}} \quad \forall\, \phi_{\mathcal{M}} \in V_{\mathcal{M}} \quad a(u_{\mathcal{M}}, \phi_{\mathcal{M}}) = (f_{\mathcal{M}}, \phi_{\mathcal{M}}).$$

On introduit les composantes de $u_{\mathcal{M}}$ pour la base d'éléments finis (N_i) de la façon suivante avec $T : \mathbb{R}^n \to V$:

$$u_{\mathcal{M}} = \sum \mathbf{u}_{\mathcal{M},i} N_i \quad \Leftrightarrow \quad u_{\mathcal{M}} = T\mathbf{u}_{\mathcal{M}}.$$

Remarquons que T est un isomorphisme de $\mathbb{R}^N$ dans $V_{\mathcal{M}}$. Donc $\mathbf{u}_h = T^{-1} u_h$. Notons T^* l'adjoint de T pour le produit scalaire L^2:

$$T^* : V' \to \mathbb{R}^N \quad [T^* f]_i = (f, N_i).$$

Puisque la matrice de masse de ces éléments finis est inversible, T^* est un isomorphisme de $V_{\mathcal{M}}$ dans $\mathbb{R}^n$.

La *discrétisation variationnelle*:

$$a(\sum \mathbf{u}_{\mathcal{M},j} N_j, N_i) = (f_{\mathcal{M}}, N_i) \ \ \forall i \quad \Leftrightarrow \quad \sum a(N_j, N_i) \mathbf{u}_{\mathcal{M},j} = (f_{\mathcal{M}}, N_i) \ \ \forall i$$

donne, sous forme algégrique:

$$\mathbf{A}_h \mathbf{u}_h = \mathbf{f}_h, \quad \text{where} \quad [\mathbf{A}_h]_{ij} = a(N_j, N_i) \quad \text{and} \quad \mathbf{f}_h = T^* f_h. \tag{3.5}$$

L'estimation usuelle de l'erreur *a posteriori* s'écrit:

$$u - u_{\mathcal{M}} = A^{-1}(f - Au_{\mathcal{M}}).$$

Cette estimation comporte deux éléments que nous avons besoin de transformer:

- l'opérateur A est continu et ne peut pas être inversé de façon exacte,
- le résidu $f - Au_{\mathcal{M}}$ comporte des variations de Dirac, le rendant difficile à utiliser dans des codes numériques.

Il est naturel d'approcher A^{-1} par son analogue discret, ce qui s'écrit:

$$A^{-1} \approx T\mathbf{A}_{\mathcal{M}}^{-1}T^{*}.$$

Dans le but de rendre le résidu plus lisse, nous utiliserons le procédé *Defect Correction* introduit au chapitre précédent mais seulement pour une évaluation de résidu. On évalue ce résidu sur un maillage deux fois plus fin $\mathcal{H}_{\mathcal{M}/2}$ obtenu en divisant chaque élément de $\mathcal{H}_{\mathcal{M}}$ en quatre éléments égaux. (Figure 3.4). On appelle $(N_{\mathcal{M}/2,i})$

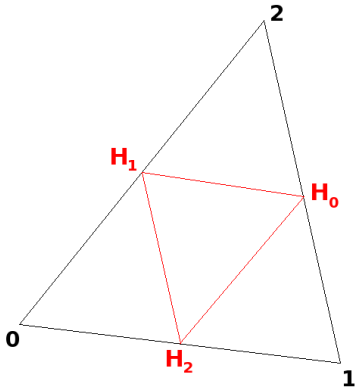

Figure 3.4: Points d'intégration pour l'évaluation du résidu sur un maillage plus fin $\mathcal{H}^{\mathcal{M}/2}$.

la base canonique éléments finis de $V_{\mathcal{M}/2}$ défini sur $\mathcal{H}^{\mathcal{M}/2}$. La fonction discrète $u_{\mathcal{M}}^{(k)}$ appartient à $V_{\mathcal{M}/2}$ mais ne satisfait pas, en général, le système discret associé. En d'autres termes

$$res_{\mathcal{M}/2} = A_{\mathcal{M}/2}T_{\mathcal{M}/2}u_{\mathcal{M}}^{(k)} - \mathbf{f}_{\mathcal{M}/2} \neq 0$$

où $res_{\mathcal{M}/2}$ peut être exprimé en termes de notation variationnelle:

$$[res_{\mathcal{M}/2}]_i = (f, N_{\mathcal{M}/2,i}) - a(u_{\mathcal{M}}^{(k)}, N_{\mathcal{M}/2,i}).$$

Ce résidu est transféré sur le maillage courant en utilisant un opérateur $\mathcal{R}_{\mathcal{M}/2\to\mathcal{M}}$ que l'on définit de la façon suivante:

$$\forall\ i \text{ sommet de } \mathcal{H}^{\mathcal{M}},$$
$$[\mathcal{R}_{h/2\to h}(res_{\mathcal{M}/2})]_i = [res_{\mathcal{M}/2}]_i + \sum_{T|i\in T} \frac{1}{2}([res_{\mathcal{M}/2}]_{(H_j)} + [res_{\mathcal{M}/2}]_{(H_l)})$$

où H_j (resp. H_l) est le milieu de l'arête faisant face au sommet j (resp. l). On propose d'appliquer l'approximation suivante:

$$T^{*}(f - Au_{\mathcal{M}}^{(k)}) \approx \mathcal{R}_{\mathcal{M}/2\to\mathcal{M}}(res_{\mathcal{M}/2}).$$

Finalement, on propose l'estimation *a posteriori*:

$$u - u_{\mathcal{M}}^{(k)} \approx (T\mathbf{A}_{\mathcal{M}}^{-1}T^{*})T^{-*}\mathcal{R}_{\mathcal{M}/2\to\mathcal{M}}(res_{\mathcal{M}/2}) \approx T\mathbf{A}_{\mathcal{M}}^{-1}\mathcal{R}_{\mathcal{M}/2\to\mathcal{M}}(res_{\mathcal{M}/2}).$$

3.5.2 Erreur itérative contrôlée FMG

Maintenant, le système algébrique (3.5) est résolu par un nombre k_c de cycles:

$$\mathbf{u}_{\mathcal{M}}^{0} = 0 \quad ; \quad \mathbf{u}_{\mathcal{M}}^{k_c+1} = Iterate(\mathbf{u}_h^{k_c}) \quad ; \quad \mathbf{u}_h^{\infty} = \mathbf{u}_h$$

L'*erreur itérative* peut être évalué en résolvant le système avec un second membre égal au résidu local d'itération:

$$\mathbf{u}_{\mathcal{M}} - \mathbf{u}_{\mathcal{M}}^{k_c} \;=\; \mathbf{A}_{\mathcal{M}}^{-1}(\mathbf{f}_{\mathcal{M}} - \mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{k_c}).$$

Il reste à comparer l'erreur itérative avec l'erreur d'approximation. Rappelons-nous tout d'abord que la définition de l'erreur d'approximation ne suppose pas nécessairement que $u_{\mathcal{M}}$ soit solution du système discret. En particulier, la même estimation est valable pour le résultat $u_{\mathcal{M}}^{k_c}$ de la résolution itérative incomplète. Introduisons l'élément de $V_{\mathcal{M}}$:

$$u_{\mathcal{M}}^{k_c} = T\mathbf{u}_{\mathcal{M}}^{k_c} = \sum [\mathbf{u}_{\mathcal{M}}^{k_c}]_i N_i,$$

On utilise alors l'estimation d'erreur *a posteriori*:

$$u - u_{\mathcal{M}}^{k_c} \approx T\mathbf{A}_{\mathcal{M}}^{-1}\mathcal{R}_{\mathcal{M}/2\to h}(res_{\mathcal{M}/2}). \tag{3.6}$$

avec $[res_{\mathcal{M}/2}]_i = (f, N_{\mathcal{M}/2,i}) - a(u_{\mathcal{M}}^{k_c}, N_{\mathcal{M}/2,i})$. Il est maintenant utile de transformer le résidu algébrique itératif $\mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{k_c} - \mathbf{f}_{\mathcal{M}}$ de façon similaire. On part de:

$$\mathbf{A}_{\mathcal{M}}(\mathbf{u}_{\mathcal{M}}^{k_c} - \mathbf{u}_{\mathcal{M}}) = \mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{k_c} - \mathbf{f}_{\mathcal{M}} - (\mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}} - \mathbf{f}_{\mathcal{M}}) = \mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{k_c} - \mathbf{f}_{\mathcal{M}},$$

ainsi

$$\mathbf{u}_{\mathcal{M}}^{k_c} - \mathbf{u}_{\mathcal{M}} = \mathbf{A}_{\mathcal{M}}^{-1}(\mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{k_c} - \mathbf{f}_{\mathcal{M}}) \quad \Leftrightarrow \quad u_{\mathcal{M}}^{k_c} - u_{\mathcal{M}} = T\mathbf{A}_{\mathcal{M}}^{-1}(\mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{k_c} - \mathbf{f}_{\mathcal{M}}). \tag{3.7}$$

Heuristique: *En assumant que, dans une certaine norme à spécifier plus tard,*

$$||u - u_{\mathcal{M}}|| \leq \varepsilon, \quad \varepsilon \text{ petit et positif},$$

et qu'après k_c itérations, nous avons

$$||u_{\mathcal{M}} - u_{\mathcal{M}}^{k_c}|| \leq \; 0.1 \; ||u - u_{\mathcal{M}}^{k_c}||, \tag{3.8}$$

alors

$$||u - u_{\mathcal{M}}^{k_c}|| \leq \; \frac{1}{0.9} \; ||u - u_{\mathcal{M}}||. \tag{3.9}$$

Comme cela a déjà été mentionné, la dernière inégalité (3.9) est, pour nous, un critère d'arrêt de la convergence itérative acceptable. On observe que ce test d'arrêt est réalisable puisque, au fur et à mesure que le nombre d'itérations augmente, la norme $||u - u_{\mathcal{M}}^{k_c}||$ est supposée converger vers $||u - u_{\mathcal{M}}||$ que l'on suppose ne pas valoir zéro alors que $||u_{\mathcal{M}} - u_{\mathcal{M}}^{k_c}||$ peut descendre jusqu'au zéro machine en itérant suffisamment de fois.
Malheureusement, évaluer les deux termes de (3.8) suppose de résoudre deux systèmes discrets avec la matrice A, un calcul qui est plus ou moins aussi coûteux que le système original à résoudre et est donc trop coûteux. Une solution possible est de résoudre approximativement le système d'erreur discrète sur une grille grossière. Ici, on propose de diminuer le coût du test d'arrêt, prenant ainsi le risque de diminuer son exactitude, en prenant la norme l^1 du second membre:

$$||\mathbf{f}_{\mathcal{M}} - \mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{k_c}||_{l^1} = \sum_i |[\mathbf{f}_{\mathcal{M}} - \mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{k_c}]_i|$$

On compte sur le fait que les cycles MG font décroître beaucoup de normes différentes du résidu avec la même pente. De fait, rien n'indique que le test d'arrêt proposé serait capable de fonctionner correctement si on l'utilise avec une autre méthode itérative que les MG.

Test d'arrêt 1: *En assumant qu'après k itérations de résolution,*

$$||\mathbf{f}_{\mathcal{M}} - \mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{k_c}||_{l^1} \leq \varepsilon'||\mathbf{f}_{\mathcal{M}} - \mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{(0)}||_{l^1}, \quad \varepsilon' \text{ petit et positif,} \tag{3.10}$$

et que nous avons

$$||\mathbf{f}_{\mathcal{M}} - \mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{k_c}||_{l^1} \leq \ 0.1\ ||\mathcal{R}_{\mathcal{M}/2\to h}(res_{\mathcal{M}/2})||_{l^1}, \tag{3.11}$$

alors, on arrête l'itération.

D'après (3.4), ε' est choisi valant $\frac{1}{10}$. En pratique, puisque le calcul du second membre de (3.11) peut nécessiter plus de temps CPU qu'un cycle, le test (3.10) prévoit que plusieurs itérations soient effectuées dans le but de diminuer le résidu itératif afin de satisfaire (3.10) avant que le second test (3.11) soit effectué.

Si le test (3.11) est négatif, plusieurs itérations sont encore effectuées avant que (3.11) soit appliqué une seconde fois, etc... Cette alternance de tests permet d'avoir un coût CPU moins important. Cela est inspiré par une méthode analogue utilisée par Arioli et ses collègues [61]. Notre formulation finale se résume comme suit :

Test d'arrêt 2:

1. *Application de cycles d'itération jusqu'à ce que*

$$||\mathbf{f}_{\mathcal{M}} - \mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{k_c}||_{l^1} \leq \frac{1}{10}||\mathbf{f}_{\mathcal{M}} - \mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{(0)}||_{l^1}.$$

2. *Si* $||\mathbf{f}_{\mathcal{M}} - \mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{(k_c)}||_{l^1} > \ 0.1\ ||\mathcal{R}_{\mathcal{M}/2\to\mathcal{M}}(res_{\mathcal{M}/2})||_{l^1}$, *alors* $\mathbf{u}_{\mathcal{M}}^{(0)} = \mathbf{u}_{\mathcal{M}}^{(k_c)}$ *retour à 1.*
3. *Arrêt des itérations.*

3.5.3 Application de la méthode FMG proposée

L'algorithme MG que l'on utilise est construit à partir:

- d'une séquence de maillages non-structurés qui ne sont pas nécessairement emboîtés. Les transferts inter-grilles basiques sont, de façon classique, l'interpolation P^1 (pour un transfert d'un maillage vers un maillage plus fin) et une accumulation pondérée par des coefficients barycentriques (pour un transfert d'un maillage vers un maillage plus grossier),

- d'un V-cycle en dents de scie avec 10 relaxations de Jacobi amorti comme lisseur,

- d'une encapsulation de ce cycle MG comme préconditionneur de la boucle GMRES.

Quand on parle d'un "cycle", on veut dire la combinaison d'un V-cycle MG avec un GMRES réactualisé. Le nombre assez élevé de balayages, avec l'utilisation du GMRES, est le prix que nous payons afin d'avoir une convergence robuste dans le cas où le coefficient $1/\rho$ présente des discontinuités de forte amplitude.

Trois contextes sont présentés pour montrer comment fonctionne la combinaison du FMG avec le test d'arrêt. La fonction à calculer n'est pas importante mais, pour fixer les idées, il s'agit du cas test de la tâche circulaire décrit dans le chapitre suivant. On retiendra seulement qu'il s'agit d'un cas peu raide, justifiant le choix de notre cyclage robuste GMRES-MG.

Sur la figure 3.5, la solution initiale est uniforme. On applique 90 cycles GMRES-MG. La norme l^1 du résidu préconditionné MG (commencé à l'itération 1) décroît de façon monotone et rapide de 1 à 10^{-8} (c'est la courbe avec des symboles $\times$). La courbe marquée par des symboles superposant $+$ et $\times$ représente la norme du résidu de léquation $||\mathbf{f}_{\mathcal{M}} - \mathbf{A}_{\mathcal{M}}\mathbf{u}_{\mathcal{M}}^{k_c}||_{l^1}$. Elle commence à moins que 1, présente une augmentation durant une phase et, ensuite, décroît jusqu'à environ 2.5 10^{-3}. La courbe marquée par des symboles $+$ représente la norme du résidu d'approximation $||\mathcal{R}_{\mathcal{M}/2\to h}(res_{\mathcal{M}/2})||_{l^1}$. Elle aussi augmente tout d'abord de façon similaire et, ensuite, diminue vers une valeur limite non-nulle. Ces deux courbes présentent une intersection à environ 70 cycles GMRES-MG. Ce nombre de cycle est vraisemblablement beaucoup trop grand et contient donc une forte marge de sécurité. Dans l'exemple choisi, on connaît la solution exacte ce qui fait qu'on peut aussi représenter la norme de l'erreur d'approximation $||u - u_{\mathcal{M}}^{(k_c)}||_{l^1}$ qui commence à un nombre proche de 1 et décroît vers une valeur limite égale à $||u - u_{\mathcal{M}}||_{l^1}$ et valant environ 5 10^{-5}. La courbe nous confirme qu'avec 70 cycles, cette valeur est bien approchée.

Sur la figure 3.6, le changement principal est que la solution initiale fournie

en interpolant la solution discrète calculée sur le précédent maillage, plus grossier. Les cycles sont arrêtés à 34. La norme l^1 du résidu préconditionné MG commence à une petite valeur et décroît rapidement jusqu'à 5 10^{-8} (courbe avec des symboles $\times$). Marquée par des symboles superposant $+$ et $\times$, la norme du résidu de l'équation $0.1||\mathcal{R}_{\mathcal{M}/2\to\mathcal{M}}(res_{\mathcal{M}/2})||_{l^1}$ commence à une valeur plus petite que dans le cas précédent et décroît de façon monotone d'une façon similaire au résidu de l'équation, tendant vers une valeur limite non-nulle. Ces deux courbes présentent une intersection à environ 22 cycles GMRES-MG. Contrairement au cas précédent, la norme de l'erreur d'approximation $||u - u_{\mathcal{M}}^{k_c}||_{l^1}$ commence à une petite valeur d'environ 10^{-4} et décroît jusqu'à environ 5 10^{-5} ce qui indique (1) que la convergence numérique n'est pas si bonne (ordre 1) entre deux maillages et (2) que, encore, l'erreur itérative à l'itération 22 est beaucoup plus petite que celle d'approximation.

Sur la figure 3.7, on donne un exemple de séquence FMG pour un problème légèrement plus facile (l'équation de Laplace avec des maillages uniformes). La convergence sur quatre maillages successifs est montrée. L'erreur d'approximation converge numériquement au second ordre. Avec les maillages 2 et 4, le second test est négatif lors de sa première application et une seconde convergence MG est appliquée. Les quatre phases sont complètes avec un total de 16 cycles ce qui donne un nombre moyen raisonnable de 4 cycles par phase FMG.

Conclusion du chapitre

Nous avons introduit un nouveau critère d'arrêt pour FMG et avons montré comment il s'insère dans la théorie existante. Une importante hypothèse heuristique est la convergence par MG des différents modes à des taux voisins. Cette hypothèse est essentielle, et rend illusoire toute tentative d'étendre la méthode directement à un autre type d'algorithme.
En revanche, les calculs préliminaires en FMG semblent indiquer un bon fonctionnement du critère, que nous vérifierons dans les trois chapitres qui suivent, aucun incident ne nous ayant conduit à changer ce critère.
Comme l'ont démontré les développements de ce chapitre, la méthode FMG n'est pas une méthode purement algébrique, mais fait intervenir des considérations d'approximation. Nous avons proposé un critère d'arrêt basé sur un estimateur de l'erreur d'approximation. Comme on l'imagine, ce type d'estimateur a sa place dans un algorithme combinant résolution et adaptation de maillage.

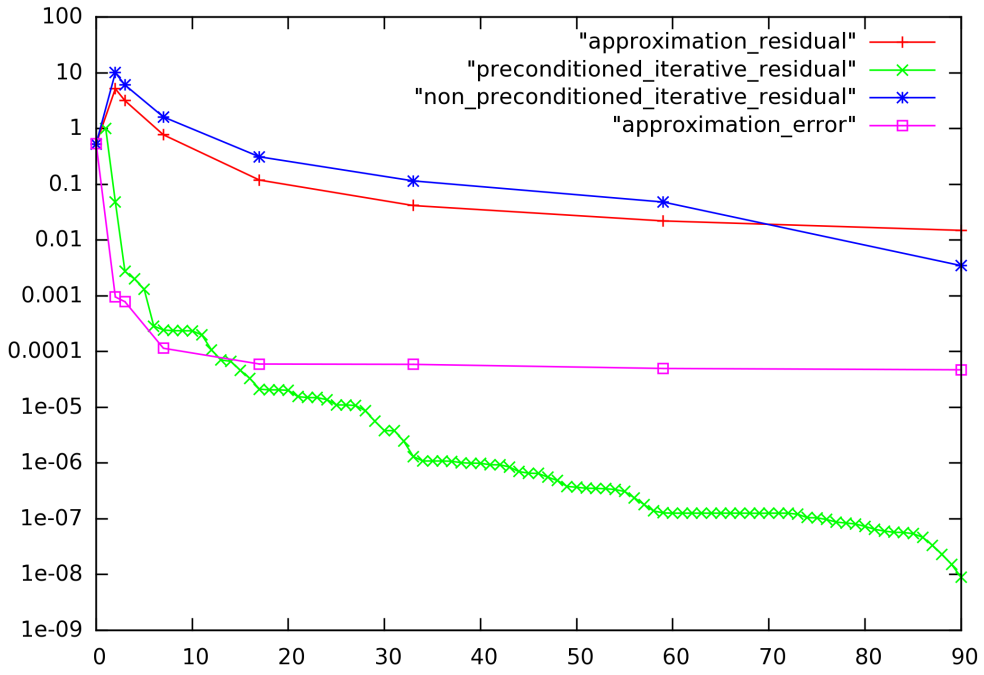

Figure 3.5: Convergence de l'itération GMRES-MG en norme l^1 du résidu itératif $||\mathbf{A}_h u_h^k - f_h||_{l^1}$ $(*)$, du résidu d'approximation $||\mathcal{R}_{h/2\to h}(res_{h/2})||_{l^1}$ $(+)$, du résidu itératif préconditionné $||MG(\mathbf{A}_h u_h^k - f_h)||/||f_h||(\times)$, et de la norme $||u - u_h||_{l^1}$ de l'erreur d'approximation $(\square)$, pour un problème de Poisson, en partant d'un champ uniforme $u_h^0 = 1$ à l'itération 0.

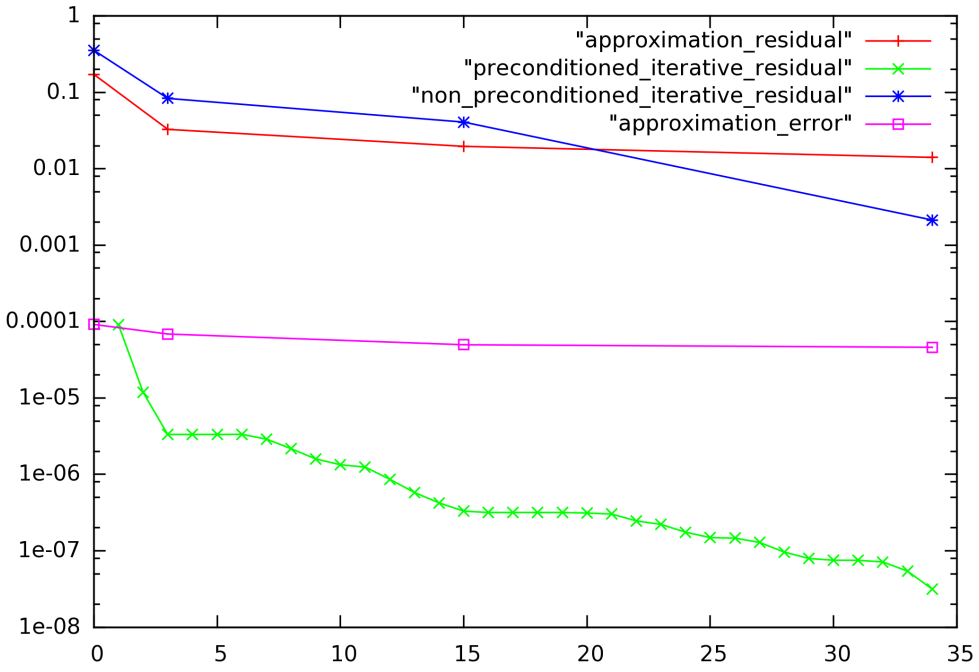

Figure 3.6: Convergence de l'itération GMRES-MG en norme l^1 du résidu itératif $||\mathbf{A}_h u_h^k - f_h||_{l^1}$ $(*)$, du résidu d'approximation $||\mathcal{R}_{h/2\to h}(res_{h/2})||_{l^1}$ $(+)$, du résidu itératif préconditionné $||MG(\mathbf{A}_h u_h^k - f_h)||/||f_h||(\times)$, et de la norme $||u - u_h||_{l^1}$ de l'erreur d'approximation $(\square)$, en partant de l'interpolation d'une solution convergée sur la grille juste plus grossière à l'itération 0.

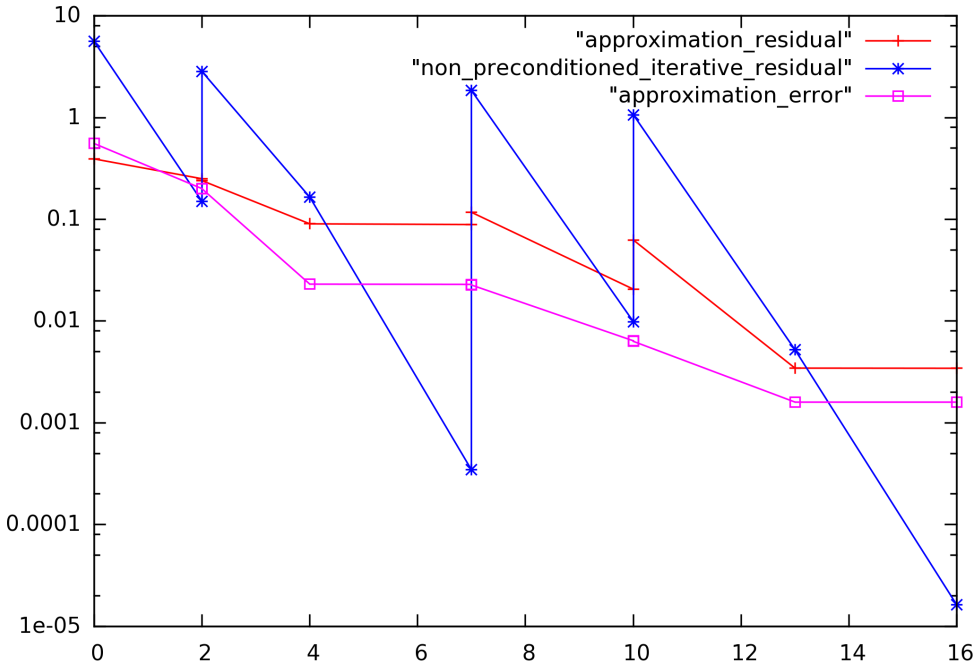

Figure 3.7: Convergence de l'itération GMRES-MG en norme l^1 du résidu itératif $||\mathbf{A}_h u_h^k - f_h||_{l^1}$ ($*$), du résidu d'approximation $||\mathcal{R}_{h/2\to h}(res_{h/2})||_{l^1}$ ($+$), du résidu itératif préconditionné $||MG(\mathbf{A}_h u_h^k - f_h)||/||f_h||(\times)$, et de la norme $||u - u_h||_{l^1}$ de l'erreur d'approximation ($\square$), en partant de l'interpolée de solution convergée sur la grille la plus grossière (grille 1) et en appliquant 4 phases FMG, de la grille 2 à la grille 5.

Chapter 4

FMGA basé-hessien

4.1 Avant-propos

Le but de ce chapitre est de proposer un couplage entre l'algorithme de résolution FMG et une méthode d'adaptation anisotrope basée sur une métrique continue et sur un hessien au sens suivant: l'erreur minimisée est une norme L^2 de l'erreur d'interpolation commise sur le champ solution.

Ce travail a été publié dans un article

A mesh-adaptative metric-based full multigrid for the Poisson problem, *International Journal for Numerical Methods in Fluids*,79-1, 30 53, 2015

que nous incluons dans ce chapitre.

Cet article contient une description de la méthode du critère d'arrêt, méthode déjà discutée dans le chapitre précédent. Le lecteur est invité à passer cette description (qui utilise les mêmes notations).

4.2 Introduction

Multi-grid methods (MG) can produce fast and robust solution algorithms. They apply to a large variety of models and approximations in Computational Mechanics.

MG uses intensively approximation properties. A simple local iteration is applied on the given grid. The iterative convergence is then accelerated by means of a set of coarser-grid corrections, ranging typically from a level which is just twice coarser than the given grid to a coarsest level containing just a few dozens degrees of freedom. The approach generally produces an iterative convergence which, when expressed in terms of (logarithm of) residual norm decreasing with iterations, is more or less of constant slope. Further, this slope does not depend on mesh size. In particular, an accurate discrete solution with N degrees of freedom is obtained with a number of operations bounded by $const.\ N\ Log(N)$, a complexity nearly optimal. Lastly, MG can be combined with a nested iteration producing the Full-Multi-grid (FMG) algorithm: FMG involves n phases, working from the 1-st coarsest mesh to the n-th finest mesh. The j-th phase of FMG solves the appproximate PDE on the j-th mesh. This j-th phase starts from an interpolation of the result of $j-1$-th mesh

and applies a certain number k_j of MG cycles with the available j coarser meshes, from 1-st to j-th. In an ideal case, the number k_j of cycles in each phase is the same. Then, FMG has an optimal complexity of *const.* N, predicted by theory and observed on many practical examples, see the reference book [76].

But this rosy picture needs some rectifications.

First, in many cases, the single local iteration of MG is frequently not sufficient to deal with singular or stiff configurations like discontinuities or boundary layers. It becomes necessary to use more sophisticated less local iterations and/or more sophisticated coarse mesh definition, as proposed by Algebraic MG [19][75] or by anisotropic mesh coarsening [66], [63], [39] and/or more adapted inter-grid transfers. Indeed, in some cases, the directly-coarser grid correction is not able to complement the fine grid iteration or may even work in a defavourable way. As a consequence, the best set of grids to apply for MG acceleration is not necessarily the best set of grids to apply for the FMG process. With the method proposed in this paper, many anisotropic strategies can be applied thanks to the way we parametrize our meshes but this issue is not addressed in this paper.

Second, for a lot of complex applications, it has been remarked that FMG does not work, in the sense that the usual stopping criterion produces a discrete final solution with an accuracy deteriorated by an insufficient iterative resolution. See for example [24]. Let us examine a possible FMG failure scenario. The two assumptions in FMG theory are (1) a MG convergence which does not depend of level fineness and (2) the asymptotic high-order convergence of the discrete solution to continuous one on the different meshes of FMG, including coarse ones. Assuming the MG cycling convergence is good, FMG failure can be explained by the lack of asymptotic convergence to continuous, either because meshes are still too coarse or because solution involves small details or singularities. As a consequence, when the basic FMG algorithm involving a fixed number of cycles per phase is applied, the solution produced at end of FMG may be inaccurate. Adjusting the number of cycles to the necessary convergence, if not done accurately, may result in a computational cost much larger than the one which the theory of FMG would let expect and may result in loosing the *const.*N complexity. Stopping criteria for iterative solvers have been the topic of many published works. In the case of quasi-Newton iteration, a typical work is [61]. It is commonly admitted that the best criterion is to stop the iteration when the iteration error between iterated approximated solution and converged approximated solution is smaller than the approximation error between converged approximated solution and exact solution [76],[61]. But computing the approximation error is computationally costly. In [61], its evaluation is replaced by an assumption concerning $O(h^2)$ mesh convergence. In the present paper, we propose to use an *a posteriori* residual as reference for stopping MG cycles.

MG and FMG have been combined with mesh adaptation in many works. Let us cite a pioneering work of R. Bank, [11], and a few more recent ones, such as [73],[17],[67],[64]. Adaptative works are most frequently based on mesh refinement by local division, producing embedded meshes. More generally, unstructured non-embedded MG and FMG have been penalized during years by the difficulty in

building and managing multiple coarse and fine unstructured meshes in particular for industrial applications. This difficulty is easier to address today, with the recent progress of mesh generation and adaptation, see e.g. [42]. Due to this progress, novel anisotropic strongly mesh-adaptative algorithms are now available. By strongly mesh-adaptative, we mean that an anisotropic mesh adaptation is strongly coupled with the solver thanks to a nonlinear fixed point iteration. Anisotropic mesh adaptors have been observed as carrying two important advantages. First, not only many computations are performed in much better conditions than with traditional methods but, also, they allow computations which were simply *not feasible* without anisotropic adaptation, like the propagation of a sonic boom from aircraft to ground [60]. Second, anisotropic mesh adaptors provide mesh convergence at high-order for singular problems [53]. For non-singular problems but rather heterogeneous problems, non-adaptative methods will produce higher order convergence only with very fine meshes. Anisotropic adaptation will give a high order numerical mesh convergence with a much smaller number of nodes.

The purpose of the present paper is to discuss some important issues in the combination of FMG and anisotropic mesh adaptation for a simple problem, the 2D Poisson problem. The plan is as follows. The next section introduces Riemannian metrics for defining what we call a continuous mesh model and the fixed-point mesh adaptation algorithm. Section 3 combines MG and mesh adaptation. Section 4 presents FMG and proposes a stopping criterion for it. Section 5 defines the complete proposed algorithm, combining FMG and anisotropic mesh adaptation. The paper is completed by several test cases and a discussion.

4.3 Mesh parametrization and mesh adaptation loop

4.3.1 Continuous mesh model

We recall shortly the continuous mesh framework, introduced in [54, 55]. This framework lies in the class of metric-based methods. A continuous mesh $\mathcal{M}$ of the computational domain Ω is identified to a Riemannian metric field [16] $\mathcal{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$. For all $\mathbf{x}$ of Ω, $\mathcal{M}(\mathbf{x})$ is a symmetric 2×2 matrix. Its diagonalisation writes:

$$\mathcal{M}(\mathbf{x}) = d(\mathbf{x})\ {}^t\mathcal{R}(\mathbf{x}) \begin{pmatrix} r_1^{-1}(\mathbf{x}) & \\ & r_2^{-1}(\mathbf{x}) \end{pmatrix} \mathcal{R}(\mathbf{x}), \tag{4.1}$$

The *total number of vertices* $\mathcal{C}$ is defined as:

$$\mathcal{C}(\mathcal{M}) = \int_\Omega d(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_\Omega \sqrt{\det(\mathcal{M}(\mathbf{x}))}\,\mathrm{d}\mathbf{x}.$$

A discrete mesh $\mathcal{H}$ of the same domain Ω is a *unit mesh with respect to* $\mathcal{M}$ if, to simplify, each edge $\mathbf{e} = \mathbf{ab}$ verifies $\int_0^1 \sqrt{{}^t\mathbf{ab}\,\mathcal{M}(\mathbf{a}+t\,\mathbf{ab})\,\mathbf{ab}}\,\mathrm{d}t \in \left[\frac{1}{\sqrt{2}}, \sqrt{2}\right]$.

Given a smooth function u, to each unit mesh $\mathcal{H}$ with respect to $\mathcal{M}$ corresponds a local interpolation error $|u - \Pi_{\mathcal{H}} u|$. In [54, 55], it is shown that this interpolation

error is well represented by the so-called continuous interpolation error related to $\mathcal{M}$ which is locally expressed in terms of the Hessian H_u of u as follows:

$$|u - \pi_{\mathcal{M}} u|(\mathbf{x}) \quad = \quad \frac{1}{10}\text{trace}(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})\,|H_u(\mathbf{x})|\,\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})) \tag{4.2}$$

where $|H_u|$ is deduced from H_u by taking the absolute values of its eigenvalues. We define as optimal metric the one which minimizes the right-hand side under the constraint of a total number of vertices $\mathcal{C}$ equal to a parameter N. After solving analytically this optimization problem, this defines the unique optimal metric $(\mathcal{M}_{opt}^{\mathbf{L}^p}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ as:

$$\mathcal{M}_{opt}^{\mathbf{L}^p} = D_{\mathbf{L}^p}\,(\det|H_u|)^{\frac{-1}{2p+2}}\,|H_u| \quad \text{with} \quad D_{\mathbf{L}^p} = N\left(\int_\Omega (\det|H_u|)^{\frac{p}{2p+2}}\right)^{-1}, \tag{4.3}$$

where $D_{\mathbf{L}^p}$ is a global normalization term set to obtain a continuous mesh with complexity N and $(\det|H_u|)^{\frac{-1}{2p+2}}$ is a local normalization term accounting for the sensitivity of the $\mathbf{L}^p$ norm. In the sequel, we choose $p = 2$.

4.3.2 Fixed-point mesh adaptation

In the case where the function u is the solution of a Partial Differential Equation, the Hessian-based method extends heuristically as follows. Given a discrete solution u_h to the PDE, a Hessian $H(u_h)$ is defined from it. The so-called optimal mesh $\mathcal{M}_{pde}$ is defined by:

$$\mathcal{M}_{pde} = \mathcal{M}_{opt}(H(u_h(\mathcal{M}_{pde})))$$

where, for any metric $\mathcal{M}$, $u_h(\mathcal{M})$ is the discrete PDE solution computed on a unit mesh for $\mathcal{M}$. We solve the non-linear problem giving the optimal mesh $\mathcal{M}_{pde}$ by applying the following loop:

Fixed point for adaptative PDE approximation:
1- compute the PDE approximate solution u_h on current mesh $\mathcal{M}$,
2- compute an approximate Hessian $H(u_h)$,
3- adapt with N nodes according to this Hessian, obtain $\mathcal{M} = \mathcal{M}_{opt}(H(u_h))$,
4- go to 1.

For remeshing phases, we used indifferently `MeshGems-Adapt` of Distene and in-house versions of `Yams` ([40, 41]). Due to the discrete and noisy character of remeshing, a strictly-fixed point cannot be obtained but, instead, the iteration needs to be stopped when further work would be useless. In contrast to the adaptation to an analytic function, the deviation to a known target is not available. However, it is possible to rely on the approximation of the interpolation error given by integral (4.2). In [4], it is proposed to stop the iteration when the difference between two approximate solution fields u_h is smaller than a positive quantity to choose cleverly since this difference does not converge to zero. The simple option of a fixed number of adaptation iterations is also a rather secure one.

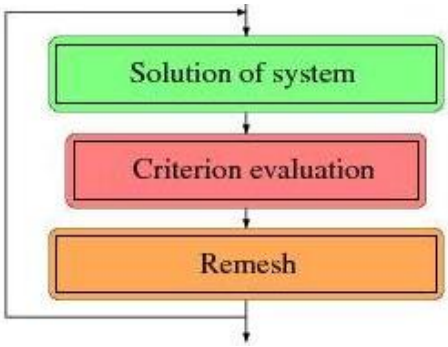

Figure 4.1: Mesh adaptation loop

4.4 The MG Anisotropic fixed-point

A MG solver uses several grids/meshes in order to iteratively converge to the discrete solution of the finer grid under study. Each iteration is called a cycle. A typical 3-grid cycle writes:

Linear 3-grid saw-tooth V-cycle:
- 1.-iterate on the finer grid, correct unknown, compute residual,
- 2.-define on medium grid a RHS as fine grid residual transfered to medium,
- 3.-iterate on the medium grid with this RHS, get medium correction, compute residual,
- 4.-define on coarse grid a RHS as medium grid residual transfered to coarse,
- 5.-iterate on the coarse grid with this RHS, get coarse correction,
- 6.-update the fine corrected unknown of step 1 with the two above corrections, transfered to fine grid.

In our case, transfers are defined as follows: correction transfers from coarse to fine are P_1 interpolated in triangles. Residual transfers, from fine to coarse, are accumulated on coarse nodes with barycentric weighting. A saw-tooth V-cycle with 10 Jacobi sweeps as pre-smoothing and without post-smoothing is applied.

As for previous section, we propose to apply the adaptative loop as an external one, the MG resolution by MG being an internal loop. The resulting algorithm is depicted in Figure 4.2.

The external loop of mesh adaptation is iterated five times for convergence of the coupling between PDE solution and adapted mesh. We now define in more details the ingredients of this loop.

4.4.1 Generating the different grids

In many cases, in particular when applying nested iteration, the coarse grids are available from previous computations. Otherwise, we want to emphasize that, as soon as a metric $\mathcal{M}$ has been computed for the current mesh, coarser meshes can

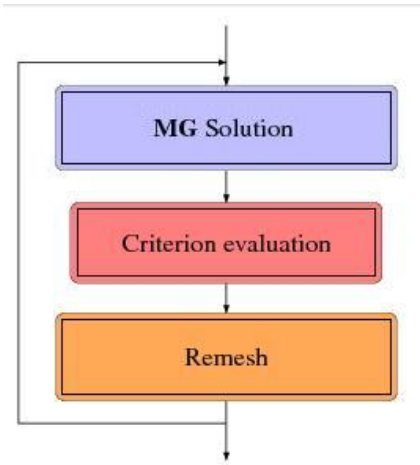

Figure 4.2: Mesh adaptation loop with Multigrid

be built from coarser metrics. We start from:

$$\mathcal{M}(x,y) = {}^t\mathcal{R}(x,y) \begin{pmatrix} \frac{1}{\Delta\xi^2(x,y)} & 0 \\ 0 & \frac{1}{\Delta\eta^2(x,y)} \end{pmatrix} \mathcal{R}(x,y)$$

in which $\Delta\xi(x,y) =$ is mesh size in the first caracteristic direction, $\Delta\eta(x,y) =$ is the mesh size in the second caracteristic direction, and $\mathcal{R}(x,y) =$ is the matrix of eigenvectors of $\mathcal{M}$. Then, coarser metrics can be built using the metric-based embedding:

$$\mathcal{M}_{coarser_1}(x,y) = {}^t\mathcal{R}(x,y) \begin{pmatrix} \frac{1}{4\Delta\xi^2(x,y)} & 0 \\ 0 & \frac{1}{4\Delta\eta^2(x,y)} \end{pmatrix} \mathcal{R}(x,y)$$

and even coarser:

$$\mathcal{M}_{coarser_2}(x,y) = {}^t\mathcal{R}(x,y) \begin{pmatrix} \frac{1}{16\Delta\xi^2(x,y)} & 0 \\ 0 & \frac{1}{16\Delta\eta^2(x,y)} \end{pmatrix} \mathcal{R}(x,y)$$

etc.

Remark: Anisotropic coarsening can also be applied. For example, assuming that the ordering of eigenvalues satisfies $\Delta\xi \leq \Delta\eta$, we can coarsen in an equivalent way to [39]:

$$\mathcal{M}_{coarser_1}(x,y) = {}^t\mathcal{R}(x,y) \begin{pmatrix} \frac{1}{4\Delta\xi^2(x,y)} & 0 \\ 0 & \frac{1}{Max(\Delta\eta^2(x,y),4\Delta\xi^2(x,y))} \end{pmatrix} \mathcal{R}(x,y).$$

This option has been tested but did not improve the results for the test cases which we present, probably because the involved meshes were not sufficiently stretched. $\square$.

4.4.2 Global fixed point

In the mesh-adapted MG, the adapted solution with a prescribed number of nodes N_k is obtained by encapsulating the MG cycle into the adaptation loop. For the adaptation convergence, we have chosen to uniformly apply 4 adaptations.

4.5 The FMG algorithm

The adaptative FMG is the succession of adaptative MG phases with transfer of the solution between each phase. At phase k_ϕ, the number of nodes is prescribed to be equal to N_{k_ϕ}. We have chosen the usual option of a new mesh size two times smaller in next FMG phase:

$$N_{k_\phi+1} = 4\ N_{k_\phi}. \tag{4.4}$$

The resulting approximation error will be presumably 4 times smaller. If, instead of a factor 4, we choose $N_{k_\phi+1} = 2\ N_{k_\phi}$, the error would be 2 times smaller and the necessary number of cycles would also be 2 times smaller. But two phases would be necessary for the same final accuracy, resulting to a similar global CPU effort. Therefore, the algorithm efficiency is not sensitive to the choice of $N_{k_\phi+1}/N_{k_\phi}$.

4.5.1 Global FMG convergence

FMG can be defined as the combination of a MG loop with a nested iteration. A first coarse mesh is used for a first evaluation of the solution. On the coarse mesh, in principle, a coarser level for acceleration is not necessary since the convergence of a standard iterative solution algorithm is rather fast. A finer mesh is built, generally by uniformly refining the first mesh. The previous solution is transfered to the new mesh, typically by interpolation. The two meshes are available for playing the role of two levels in order to solve fastly the problem with a two-grid iteration on the new mesh, starting with a good initial condition. This process is reiterated with a 3-grid solution on next mesh, etc...

Full-Multigrid:
- Solve on the coarsest mesh with a 1-grid iteration,
- Interpolate the solution to a slightly finer second mesh,
- Solve on the second mesh with a 2-grid iteration,
- Interpolate the new solution to a finer third mesh,
- Solve on the third mesh with a 3-grid iteration,
- etc...

We call *FMG k_ϕ-th phase* the k_ϕ-th nested iteration phase, using *k_ϕ-grid cycling*. Due to the initialization by the previous phase, a sufficient convergence at each phase k_ϕ can be obtained by a small number k_c of k_ϕ-grid cycles. Due to the ability of MG in exhibiting a convergence rate quasi-independant of number of level and grid size, an important gain is reachable. The prescription in stopping

iterative solutions is generally based on the following criterion:

Stopping criterion: *Iteration error should be smaller than the approximation error.*

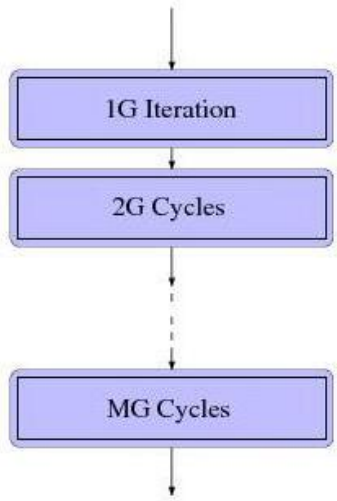

Figure 4.3: Each phase k_ϕ of the Full Multi-Grid algorithm is made of several k_ϕ-grid cycles

In the FMG theory presented in [49] and for second-order approximation schemes, it is sufficient that there exists a constant K such that, for any phase k_ϕ, the approximate solution u_{k_ϕ} computed at the end of phase satisfies:

$$||u - u_{k_\phi}|| \leq K h_{k_\phi}^2. \tag{4.5}$$

The error for u_{k_ϕ} is four times larger than for $u_{k_\phi - 1}$, which naively suggests a residual reduction by 4 at each phase. A rigorous analysis [49] shows that a sufficient condition for (4.5) is that each FMG cycling phase divides the residual by a factor more than $2r^\kappa$, where $r = (N_{k_\phi+1}/N_{k_\phi})^{1/d}$ is the (fixed) mesh size ratio between two phases, here chosen as $r = 2$ (d is the space dimension), and $\kappa = 2$ is the accuracy order of the scheme. This gives $2r^\kappa = 8$. Let us denote $h = h_{k_\phi}$ the mesh size at phase k_ϕ and $u_h^{k_c}$ the iterate obtained at this phase after k_c cycles have been applied. A conservative choice is to take a factor 10:

$$||\mathbf{A}_h u_h^{k_c} - f_h|| \leq \frac{1}{10} ||\mathbf{A}_h u_h^0 - f_h||. \tag{4.6}$$

The weakness of this theory lies in the central assumption (4.5). An evident first remark is that (4.5) is a mesh convergence assumption, which can be established only for a sufficiently fine mesh. This is true for *grids/levels* ℓ with ℓ sufficiently large. Conversely, for ℓ close to unity, the number of nodes is small and the above estimate

is clearly *not true*. Another major difficulty arises when (4.5) does not holds for the last fine grids. In that latter case, the dilemma of FMG is the following.

If the convergence test (4.6) is based on a fixed decrease of residual at each phase, then, the convergence, being not sufficient, will produce an inaccurate solution on the finer mesh. Otherwise, we apply a too severe way to stop convergence in each phase, then we get an accurate solution but we have increased the cost in a rather unpredictible way and have presumably lost the optimal $const.N$ complexity of FMG.

We are also motivated by a second issue: if a mesh adaptation loop is applied, the MG cycling between mesh updates will be initialised by a candidate solution obtained with a just slightly less adapted mesh of same fineness. This candidate solution can be already very close to the converged solution. Then, in order to avoid a rather large amount of unnecessary computing time, we need to recognize it and stop early the MG cycling.

Next two paragraphs deal with introducing a measure of approximation error and a control of iteration error in FMG in order to improve the cycling stopping test for FMG, consolidating the accuracy of FMG.

4.5.2 Approximation error estimate

Let us denote $\mathcal{M}_h$ the current mesh on which we try to iteratively solve the discrete system $A_h\mathbf{u}_h = f_h$. Let $u_h^k(x) = \sum \mathbf{u}_{h,i}^k N_i(x)$ be the approximate function at iteration k of a given iterative solver (N_i holds for the finite-element basis function related to node i). For a second-order accurate approximation, Arioli and co-workers propose, in [61], a stopping criterion based on:

$$||\mathbf{A}_h u_h^k - f_h||_{H^{-1}} \;\leq\; h^2\; ||f_h||_{H^{-1}}$$

where H^{-1} holds for the dual of the Sobolev space H_0^1. Using the h^2 factor assumes that the numerical mesh-convergence is close to scheme asymptotic convergence.

We propose here a method which does not *a priori* assume mesh convergence. Let us introduce some notations: let $V = H_0^1(\Omega)$, Ω be the computational domain. The continuous PDE system is written in short:

$$Au = f \quad or \quad u \in V \quad \forall\, \phi \in V \quad a(u,\phi) = (f,\phi)$$

with

$$a(u,\phi) = \int_\Omega \frac{1}{\rho} \nabla u \cdot \nabla \phi \; \mathrm{d}x\mathrm{d}y$$

in which ρ is a positive non-uniform coefficient which can be discontinuous. Let V_h be the usual P_1-continuous finite-element approximation space. The finite-element discretisation is written:

$$u_h \in V_h \quad \forall\, \phi_h \in V_h \quad a(u_h,\phi_h) = (f_h,\phi_h)$$

We introduce the components of u_h for the finite-element basis (N_i) as follows, with $T : \mathbb{R}^n \to V$:

$$u_h = \sum \mathbf{u}_{h,i} N_i \quad \Leftrightarrow \quad u_h = T\mathbf{u}_h.$$

Note that T is an isomorphism from $\mathbb{R}^n$ to V_h, $\mathbf{u}_h = T^{-1}u_h$. Let us denote T^* the adjoint of T for the L^2 scalar product:

$$T^* : V' \to \mathbb{R}^n \quad [T^*f]_i = (f, N_i).$$

Since the mass matrix of this finite-element is invertible, T^* is an isomorphism from V_h to $\mathbb{R}^n$. The *variational discretization*:

$$a(\sum \mathbf{u}_{h,j} N_j, N_i) = (f_h, N_i) \ \ \forall i \quad \Leftrightarrow \quad \sum a(N_j, N_i)\mathbf{u}_{h,j} = (f_h, N_i) \ \ \forall i$$

transforms into an algebraic one:

$$\mathbf{A}_h \mathbf{u}_h = \mathbf{f}_h, \quad \text{where} \quad [\mathbf{A}_h]_{ij} = a(N_j, N_i) \quad \text{and} \quad \mathbf{f}_h = T^* f_h. \tag{4.7}$$

The usual *a posteriori* error estimate writes:

$$u - u_h = A^{-1}(f - Au_h).$$

This estimate is made of two factors which we need to transform:
- the operator A is continuous and cannot be exactly inverted,
- the residual $f - Au_h$ involves Dirac layers, making it difficult to use in numerical codes.

It is natural to approximate A^{-1} by its discrete analog, which writes:

$$A^{-1} \approx T\mathbf{A}_h^{-1}T^*.$$

In order to get a smoother residual, we evaluate it on a finer mesh $\mathcal{M}_{h/2}$ obtained by dividing each element of $\mathcal{M}_h$ into four equal elements, (Figure 4.4). We denote $(N_{h/2,i})$ the canonic finite-element basis of $V_{h/2}$ defined on $\mathcal{M}_{h/2}$. The discrete function u_h^k belongs to $V_{h/2}$ but does not (in general) satisfy to the discrete system set on $\mathcal{M}_{h/2}$, in other words

$$res_{h/2} = -A_{h/2}T_{h/2}u_h^k + \mathbf{f}_{h/2} \neq 0$$

where $res_{h/2}$ can be expressed in terms of the variational notation:

$$[res_{h/2}]_i = (f_{h/2}, N_{h/2,i}) - a(u_h^k, N_{h/2,i}).$$

This residual is transfered on the current mesh by using the operator $\mathcal{R}_{h/2\to h}$ which we define as follows:

$$\forall\, i \text{ vertex of } \mathcal{M}_h, \ \ [\mathcal{R}_{h/2\to h}(res_{h/2})]_i = [res_{h/2}]_i + \sum_{T|i\in T} \frac{1}{2}([res_{h/2}]_{(H_j)} + [res_{h/2}]_{(H_k})$$

where H_j (resp. H_k) is the mid-size facing vertex j (resp. k). We propose to apply the following approximation:

$$T^*(f - Au_h^k) \approx \mathcal{R}_{h/2\to h}(res_{h/2}).$$

Finally, we propose the approximate *a posteriori* estimate:

$$u - u_h^k \approx (T\mathbf{A_h^{-1}}T^*)T^{-*}\mathcal{R}_{h/2\to h}(res_{h/2}) \approx T\mathbf{A_h^{-1}}\mathcal{R}_{h/2\to h}(res_{h/2}).$$

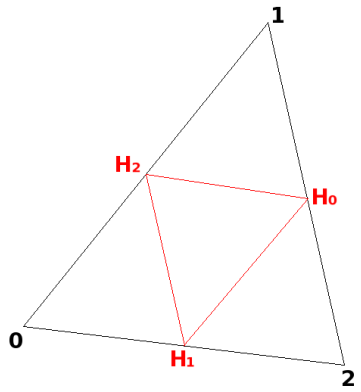

Figure 4.4: Integration nodes for residual evaluation on a finer mesh $\mathcal{M}_{h/2}$.

4.5.3 Iterative-Error-controlled FMG

Now, the algebraic system (4.7) is solved by a number k_c of cycles:

$$\mathbf{u}_h^0 = 0 \quad ; \quad \mathbf{u}_h^{k_c+1} = Iterate(\mathbf{u}_h^{k_c}) \quad ; \quad \mathbf{u}_h^\infty = \mathbf{u}_h$$

The *iterative error* can be evaluated by solving the system with a right-hand side equal to the local iteration residual:

$$\mathbf{u}_h - \mathbf{u}_h^{k_c} \;=\; \mathbf{A}_h^{-1}(\mathbf{f}_h - \mathbf{A}_h\mathbf{u}_h^{k_c}).$$

It remains to compare the iterative error with the above approximation error. Remember first that the approximation error above derivation did not use the assumption that u_h is the solution of the discrete system. In particular, the same estimate holds for the result $u_h^{k_c}$ of the incomplete iterative resolution. Let us introduce the element of V_h:

$$u_h^{k_c} = T\mathbf{u}_h^{k_c} = \sum [\mathbf{u}_h^{k_c}]_i N_i,$$

Then, we use the *a posteriori* error estimate:

$$u - u_h^{k_c} \approx T\mathbf{A}_h^{-1}\mathcal{R}_{h/2\to h}(res_{h/2}). \tag{4.8}$$

with $[res_{h/2}]_i = (f, N_{h/2,i}) - a(u_h^{k_c}, N_{h/2,i})$. It is now useful to transform the algebraic iterative residual $\mathbf{A}_h\mathbf{u}_h^{k_c} - \mathbf{f}_h$ in similar terms. We start from:

$$\mathbf{A}_h(\mathbf{u}_h^{k_c} - \mathbf{u}_h) = \mathbf{A}_h\mathbf{u}_h^{k_c} - \mathbf{f}_h - (\mathbf{A}_h\mathbf{u}_h - \mathbf{f}_h) = \mathbf{A}_h\mathbf{u}_h^{k_c} - \mathbf{f}_h,$$

thus

$$\mathbf{u}_h^{k_c} - \mathbf{u}_h = \mathbf{A}_h^{-1}(\mathbf{A}_h\mathbf{u}_h^{k_c} - \mathbf{f}_h) \quad \Leftrightarrow \quad u_h^{k_c} - u_h = T\mathbf{A}_h^{-1}(\mathbf{A}_h\mathbf{u}_h^{k_c} - \mathbf{f}_h). \tag{4.9}$$

Heuristics: *Assuming that, in some norm to specify later,*

$$||u - u_h|| \leq \varepsilon, \quad \varepsilon \textit{ small and positive},$$

and that after k_c solver iterations, we have

$$||u_h - u_h^{k_c}|| \leq \ 0.1 \ ||u - u_h^{k_c}||, \tag{4.10}$$

then

$$||u - u_h^{k_c}|| \leq \ \frac{1}{0.9} \ ||u - u_h||. \tag{4.11}$$

As already mentioned, the last statement (4.11) is, for us, an acceptable iterative convergence stopping criterion. We observe that stopping criterion (4.10) is realizable since, as iteration number increases, the norm $||u - u_h^{k_c}||$ is supposed to converge to $||u - u_h||$, assumed to be not zero while $||u_h - u_h^{k_c}||$ can be driven to machine zero by iterating over k. Unfortunately, evaluating the two terms of (4.10) involves solving two discrete systems with matrix A, a computation which is more or less computationally as costly as the original system to solve and is therefore too costly. A possible solution is to solve approximatively the discrete error system with a coarse grid. Here, we propose to decrease the cost of the stopping test, with some risk of decreasing its accuracy, by taking the following l^1 norm of the right-hand side:

$$||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{k_c}||_{l^1} = \sum_i |[\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{k_c}]_i|$$

We rely on the observation that MG-cycles decrease many different norms of the residual with about the same slope. Therefore, we do not claim that the proposed stopping criterion is able to work adequately when associated with another iteration than MG.

Stopping test 1: *Assume that, after k solver iterations,*

$$||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{k_c}||_{l^1} \leq \varepsilon' ||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{0}||_{l^1}, \qquad \varepsilon' \ \text{ small and positive,} \tag{4.12}$$

and that we have

$$||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{k_c}||_{l^1} \leq \ 0.1 \ ||\mathcal{R}_{h/2 \to h}(res_{h/2})||_{l^1}, \tag{4.13}$$

then stop the iteration.

According to (4.6), ε' is chosen to be $\frac{1}{10}$. In practice, since the computation of the RHS of (4.13) may need more CPU than a cycle, the test (4.12) means that several iterations are performed in order to decrease the iterative residual to satisfy (4.12) before the second test (4.13) is evaluated. If test (4.13) is negative, several iterations are again performed before a second test of (4.13) is again evaluated, etc. This splitting allows for a lower CPU cost. It is inspired by an analog one proposed by Arioli and co-workers [61]. Our final formulation is as follows:

Stopping test 2:

1. *Iterate cycling until* $||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{k_c}||_{l^1} \leq \frac{1}{10} ||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{0}||_{l^1}$.
2. *If* $||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{k_c}||_{l^1} > \ 0.1 \ ||\mathcal{R}_{h/2 \to h}(res_{h/2})||_{l^1}$, *then* $\mathbf{u}_h^0 = \mathbf{u}_h^{k_c}$ *go to 1.*
3. *Stop the iteration.*

4.5.4 Application to the proposed FMG

The MG algorithm which we use is built from:
- a sequence of unstructured meshes which are not necessarily embedded. The basic inter-grid transfers are classically the P^1 interpolation (for a transfer from a mesh to a finer mesh) and an accumulation weighted with barycentric coefficients (for a transfer from a mesh to a coarser mesh),
- a saw-tooth V-cycle with 10 damped-Jacobi relaxation as a smoother,
- an encapsulation of this MG cycle as a preconditioner of a GMRES loop. When we shall talk about "a cycle", we shall mean the combination of the MG V-cycle with the GMRES updating.
The rather high number of sweeps, together with the use of GMRES, is the price we pay in order to get a robust convergence in the case where the coefficient $1/\rho$ presents high-amplitude discontinuities.

Three contexts are now examined for showing how works the combination of FMG with the stopping criterion.

In Figure 4.5, the function to compute is not important but, to fix the ideas, it is the circular test case described in the sequel. The initial solution is uniform. We apply 90 GMRES-MG cycles. The preconditioned (by MG) residual l^1 norm (started at iteration 1) indeed decreases monotonely and fastly from 1 to 10^{-8} (multiply-shaped marked curve). Marked by plus-covered-by-multiply, the equation residual norm $||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{k_c}||_{l^1}$ starts from less than 1, shows an increasing phase and, then, decreases to about 2.5 10^{-3}. Marked by plus-symbols, the approximation residual norm $||\mathcal{R}_{h/2\to h}(res_{h/2})||_{l^1}$ also first increases in a similar way and, then, goes down to a non-zero limiting value. The two above curves intersect at about 70 GMRES-MG cycles. This is probably too many cycles, *i.e.* rather conservative. In the chosen example, we know the exact solution and can also depict the approximation error norm $||u - u_h^{k_c}||_{L^1}$ which starts from a number close to 1 and decreases to a limiting level $||u - u_h||_{L^1}$ of about 5 10^{-5}. We get confirmation that, with 70 cycles, this level is tightly approached.

In Figure 4.6, the main change is that the initial solution is provided by interpolating the discrete solution computed on previous coarser mesh. Cycles are stopped at 34. The preconditioned (by MG) residual l^1 norm starts with a small level and decreases fastly to 5 10^{-8} (multiply-shaped marked curve). Marked by plus-covered-by-multiply, the equation residual norm $||\mathbf{f}_h - \mathbf{A}_h \mathbf{u}_h^{k_c}||_{l^1}$ starts from about 0.5 and, monotonely (this time), decreases to about 2 10^{-3}. Marked by plus-symbols, the approximation residual norm $0.1||\mathcal{R}_{h/2\to h}(res_{h/2})||_{l^1}$ starts from a lower value than previous case and monotonely decreases in a similar way to equation residual, going down to a non-zero limiting value. The two above curves intersect at about 22 GMRES-MG cycles. In contrast to the previous case, the approximation error norm $||u - u_h^{k_c}||_{L^1}$ starts from a low level of about 10^{-4} and decreases to about 5 10^{-5}, indicating (1) that the numerical convergence is not so good (first-order) between the two meshes and (2) that, again, the iterative error at iteration 22 is much smaller than the approximation one.

In Figure 4.7, we give an example of FMG sequence for a slightly easier problem (Laplace equation with uniform meshes). The convergence on four successive meshes is shown. The approximation error is numerically converging at second order. With mesh 2 and mesh 4, the second test is negative and a second MG convergence is applied. The four phases are complete with a total of 16 cycles, that is a reasonable mean number of 4 cycles per FMG phase.

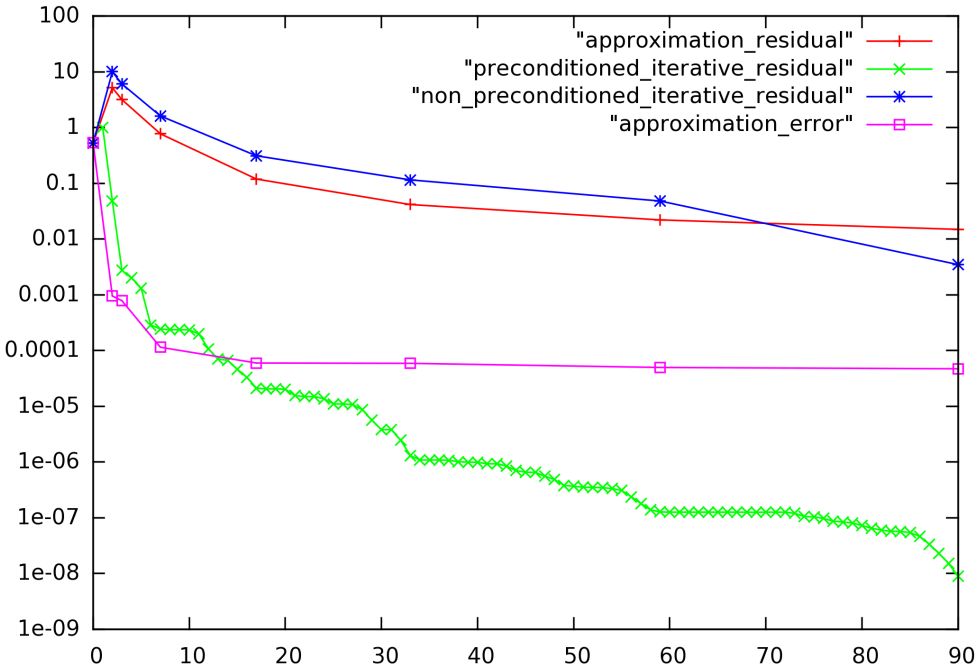

Figure 4.5: Convergence of the GMRES-MG-iterative l^1 residual norm $||\mathbf{A}_h u_h^k - f_h||_{l^1}$ (⋊), the approximation l^1 residual norm $||\mathcal{R}_{h/2\to h}(res_{h/2})||_{l^1}$ (+), the preconditioned residual norm (×), the norm $||u - u_h||_{L^1}$ of deviation to exact (□), for a Poisson problem, starting from a uniform field $u_h^0 = 1$ at iteration 0.

4.6 The FMG Anisotropicly adaptative algorithm

The synthesis of the above sections is the FMG anisotropicly adaptative algorithm. We insert the adaptation loop as an intermediate loop between FMG phases and MG cycles. Concerning the process of going to a larger number of nodes, we keep the previous meshes and define a finer one by a simple division of each element into four elements of same area. The global algorithm, sketched in Figure 4.8, writes:

Anisotropicly adaptative FMG:
step 0.- $k = 1$, start with a coarse mesh, with N_1 nodes,
step 1.- apply mesh-adaptation loop combining MG and adaptative remeshing with N_k nodes,
step 2.- transfer solution to a mesh with $N_{k+1} \approx 4N_k$ nodes, obtained by element

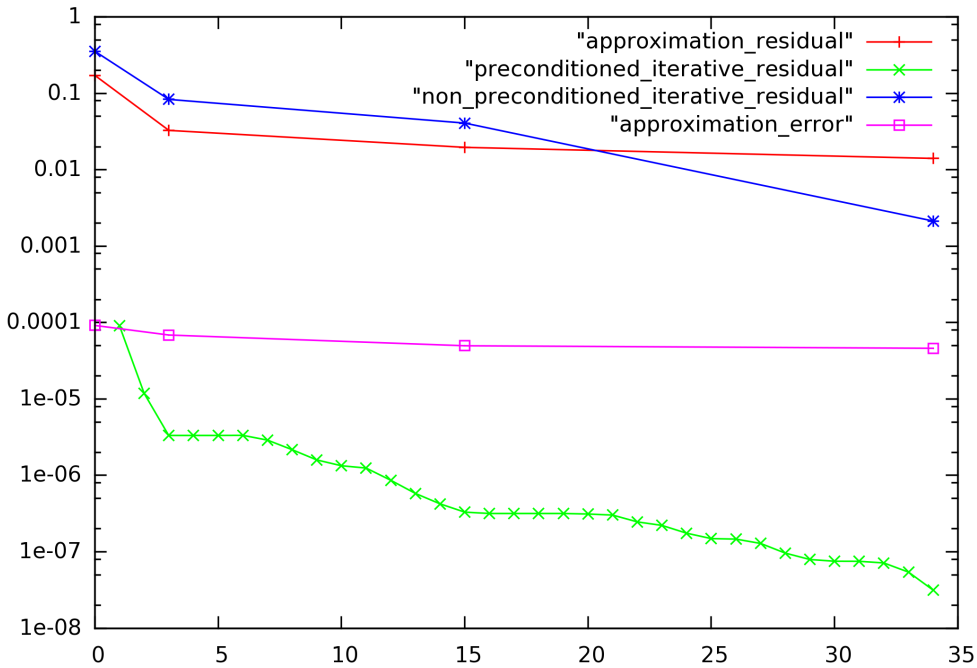

Figure 4.6: Convergence of the GMRES-MG-iterative l^1 residual norm $||\mathbf{A}_h u_h^k - f_h||_{l^1}$ ($\times$+), the approximation l^1 residual norm $||T^{-*}\mathcal{R}_{h/2\to h}(res_{h/2})||_{l^1}||_{l^1}$ (+), the preconditioned residual norm ($\times$), the norm $||u - u_h||_{L^1}$ of deviation to exact ($\square$), for a Poisson problem, starting from a coarser-grid interpolated solution at iteration 0.

division,
step 3.- if $k = kmax$, stop, else go to 1.

Let us re-visit the ways these loops are stopped. The *k-indexed external loop*, *i.e.* the FMG phase, will increase the number of mesh nodes. Theoretically, it should stop when some norm of the approximation error $|u - u_h|$ is smaller than a number prescribed by the user. This option is yet rather far from practice and is not studied in this work. The *intermediate loop* (step 1), *i.e.* the mesh adaptation loop, is stopped after 4 iterations. Only the finer grid is adapted, the coarser ones being kept. As concerns *inner loop*, *i.e.* the MG cycling loop, it is controlled by the stopping criterion defined in Section 6. The stopping criterion is used when solving on a new mesh. This happens in two different contexts: (1) after we have changed from a first mesh of N nodes to a finer mesh of $4N$ nodes between two FMG phases, as well as (2) after we have changed from a first mesh of N nodes to a second mesh of same number N of nodes, but more adapted.

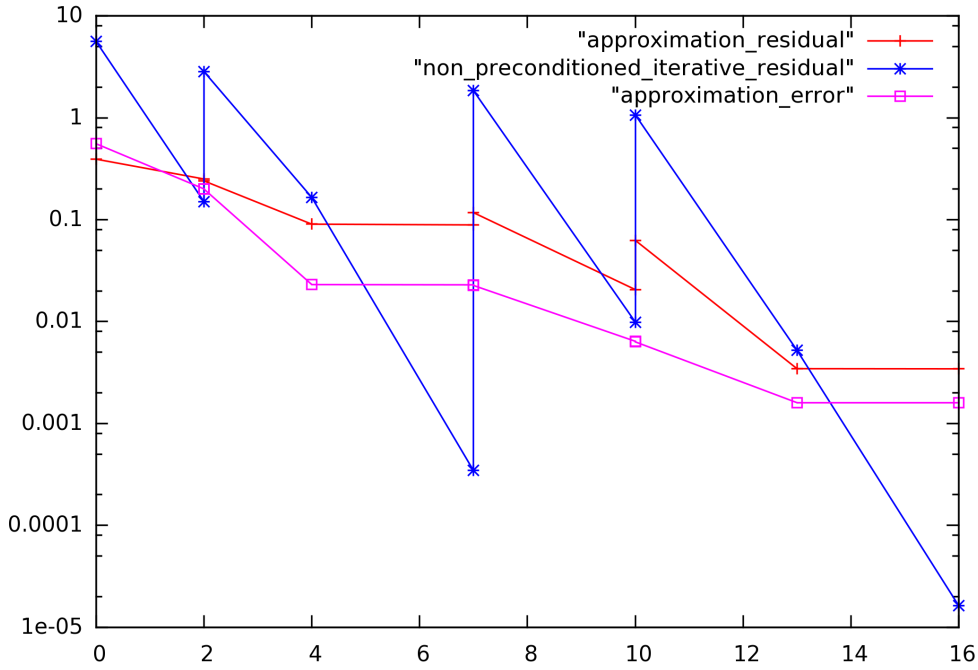

Figure 4.7: Convergence of the FMG-iterative l^1 residual norm $||\mathbf{A}_h u_h^k - f_h||_{l^1}$ ($\times$), the approximation l^1 residual norm $||T^{-*}\mathcal{R}_{h/2\to h}(res_{h/2})||_{l^1}||_{l^1}$ ($+$), the norm $||u-u_h||_{L^1}$ of deviation to exact ($\square$), for a Poisson problem, starting from the coarsest-grid (grid 1) solution and performing four FMG phases from grid 2 to grid 5.

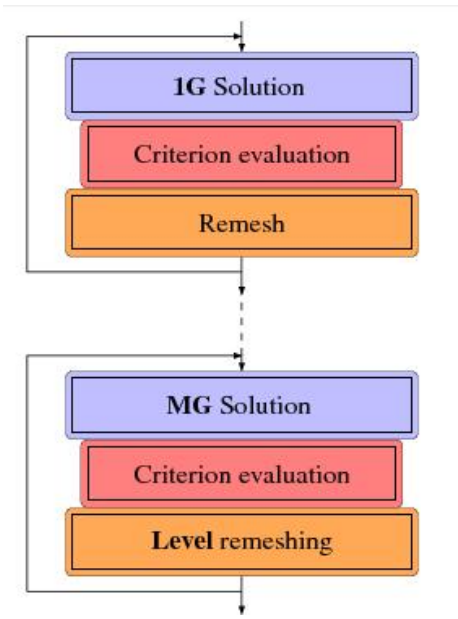

Figure 4.8: Mesh adaptative Full Multi-Grid

4.7 Examples

The proposed Adaptative Anisotropic FMG is applied to three test cases and compared with a pure FMG method applied with a sequence of embedded uniform meshes. Due to our restriction to a Poisson-like model, test cases will be rather trivial problems. However, we choose them in order to represent the three following typical difficulties of multiphase incompressible flows:
- boundary layers,
- discontinuous phase changes,
- Dirac layer source term from capillarity.
Our three simplified representations of these difficulties, in combination with the specification of mesh sizes (number of vertices), constitute a small benchmark for the performance of mesh adaptative methods, which could be used for the evaluation of various sensors.

Each of our three test cases has an analytic solution. We shall call *total approximation error* or simply *approximation error* the error between the analytic solution from one side and, from the other side, the discrete solution produced by the algorithm: since the GMRES-MG algorithm is not converged to machine-zero, our approximation error combines numerical scheme approximation error and iterative error.

4.7.1 A smooth boundary layer test case

For modelling the stiffness of a boundary layer, we consider a Poisson problem with a smooth solution presenting some anisotropic local variations. Let: $rhs(x,y) = \left[\alpha^2(exp(1/\alpha)-1)\right]^{-1} exp(x/\alpha)$ with $\alpha = 0.03$. We solve $-\Delta u = rhs$ with $\frac{\partial u}{\partial y}(x,0) = \frac{\partial u}{\partial y}(x,1) = 0$ and $u(0,y) = u(1,y) = 0$. Then $u(x,y) = [exp(1/\alpha)-1]^{-1} exp(x/\alpha) + x + [exp(1/\alpha)-1]^{-1}$. An example of approximated solution u_h is shown in Figure 4.9. That allows us to compute directly the norm of the approximation error defined by $||u-u_h||_{L^1} = \int_\Omega |u-u_h| dxdy$ which is depicted as a function of the number N of nodes of the mesh. For evaluation of our Hessian-based criterion, we draw the error of interpolation of u on the current meshes as a function of the number N of nodes of the meshes, in Figure 4.10. We observe a convergence of order two for the non-adaptative case and a similar convergence in the adaptative case. Since the criterion for adaptation which we adopted postulates that the interpolation error is a good representation of the approximation error, it is interesting to examine the convergence of the interpolation error of the exact solution on the meshes we used, see Figure 4.11. We observe that indeed both convergences are similar but that the interpolation error decreases to values which are smaller by a factor larger than two orders of magnitude. This observation has also been done for the other test cases and delivers a strong message saying that the interpolation error and its companion the Hessian criterion, while providing rather good adaptations, are not faithful representations of the actual approximation error.

Comparing the non-adaptative case and the adaptative one (Figure 4.10) for the

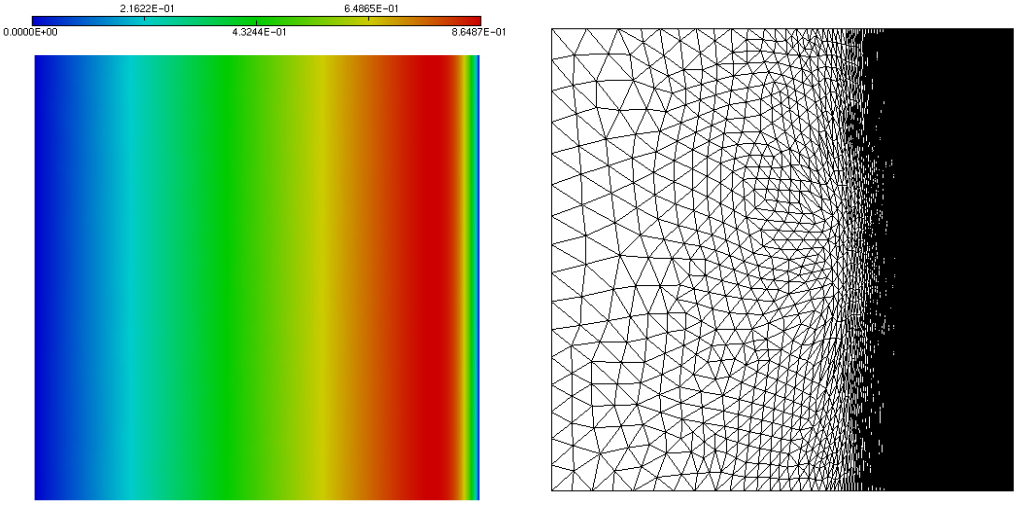

Figure 4.9: Boundary layer test case solution and adapted mesh

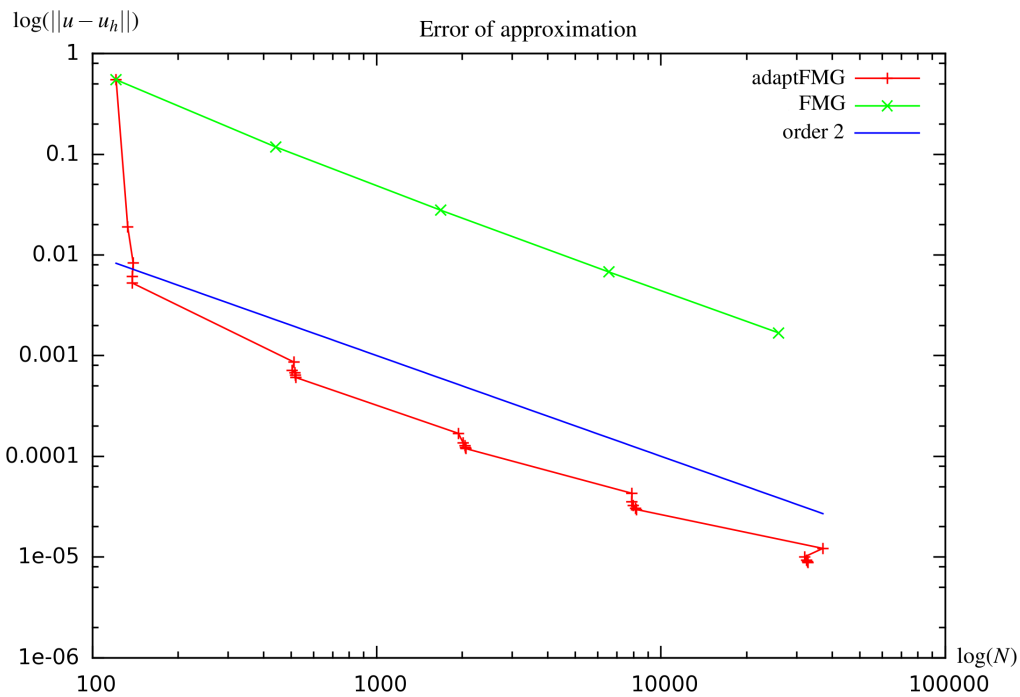

Figure 4.10: Boundary layer test case. Approximation error $||u-u_h||_{L^1}$ as a function of the number of mesh nodes. (+) non-adaptative FMG, (×) adaptative FMG. The straight line shows the second-order slope.

same number of vertices, we observe that the error $||u-u_h||_{L^1}$ is notably smaller in the adaptative case. We also draw this error in function of the CPU time in Figure 4.12. We distinguish mesh division phase with steep slopes from the mesh adaptation ones with less steep slopes. Thanks to the stopping criterion, the iterations 2

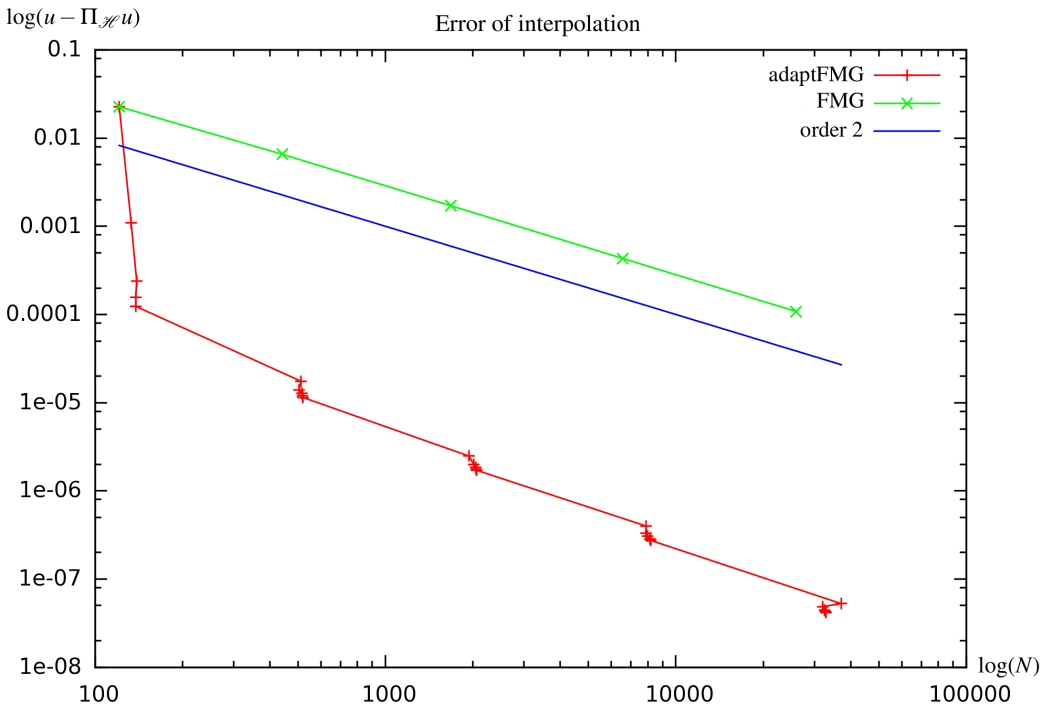

Figure 4.11: Boundary layer test case. Corresponding behavior of the interpolation error of exact solution $||u - \Pi_h u||_{L^1}$ on the same meshes as in Figure 4.10. The straight line shows the second-order slope.

to 4 of adaptation consume less cycles and, therefore, less CPU. For about 1000 seconds of a workstation, the obtained accuracy is 6 10^{-3} for the non-adaptative case and 10^{-5} for the adaptative one. The accuracy for the non-adaptative calculation with 750 seconds is obtained by adaptation with 10 seconds.

4.7.2 A singular internal layer case

The second test case exemplifies the singularity which is met in the simulation of multi-fluid flows with a large deviation between the densities ρ_1 and ρ_2 of each phase. In the case where a projection algorithm is applied, a Poisson problem with discontinuous coefficients has to be solved. An example can be found in [48]. We consider the equation of Poisson $-div(\frac{1}{\rho}\nabla u) = rhs$ with a discontinuous coefficient taking two different values $1/\rho_1$ and $1/\rho_2$ on two sub-domains Ω_1 and Ω_2 separated by an interface which is sufficiently smooth for having a normal vector. This is mathematically referred as a transmission problem and the solution is continuous across the interface but of discontinuous normal derivatives since:

$$1/\rho_1 \nabla u_1 \cdot \mathbf{n} = 1/\rho_2 \nabla u_2 \cdot \mathbf{n}$$

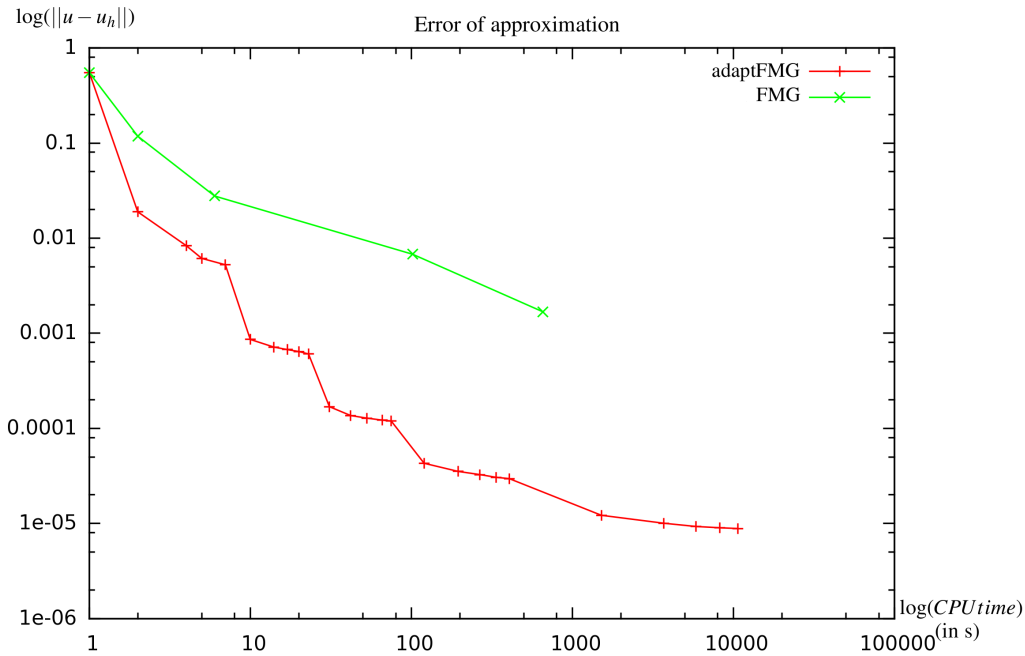

Figure 4.12: Boundary layer test case. Approximation error $||u-u_h||_{L^1}$ as a function of the CPU time. (+) non-adaptative FMG, (×) adaptative FMG.

where u_1 and u_2 are the restrictions of the solution u on Ω_1 and Ω_2. In our example, we define them as follows

$$u|_{\Omega_i} = u_i = \alpha_i + \beta_i(x^2 + y^2) \qquad i = 1, 2.$$

Further, Ω_2 is the disk of center $(0.5, 0.5)$ and of radius 0.2 in the computational domain $]0, 1[\times]0, 1[$ and we have:

$$\begin{aligned} 1/\rho_1 &= 1000. \;;\; \alpha_1 = 1.23579... \;;\; \beta_1 = -2.47158... \\ 1/\rho_2 &= 1. \;;\; \alpha_2 = 100. \;;\; \beta_2 = -2471.58... \end{aligned} \tag{4.14}$$

This is sketched in Figure 4.13. In the discrete model, the interface appears only as values of $1/\rho$ evaluated on the vertices of each grid. Convergence in terms of number of vertices is given in Figure 4.14. We observe that, without mesh adaptation (crosses +), the convergence order is around 1. This behavior can be explained by the singularity of the solution. In contrast, the overall convergence order of the adaptative process is about two. This is a usual bonus obtained by anisotropic mesh adaptation which has been noted in [59]. A short analysis of this property is proposed in [32]. The gain in efficiency is also good, see Figure 4.15.

We made a computation with iterative convergence to machine zero in order to measure a possible effect of the stopping criterion, see Figure 4.16. The result is only slightly different with, for medium meshes (between 1000 and 30,000 vertices),

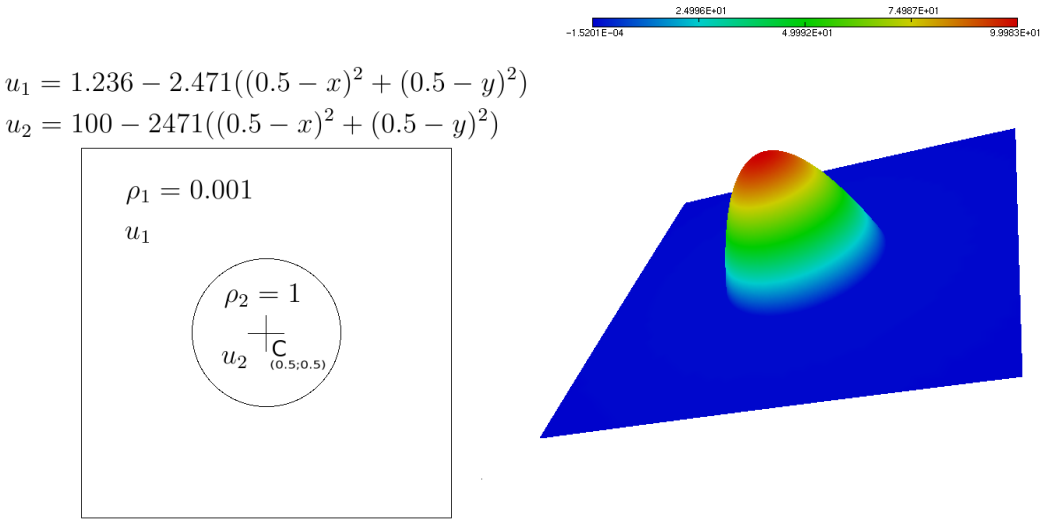

Figure 4.13: Singular internal layer: sketch of exact solution definition and a typical computation of it.

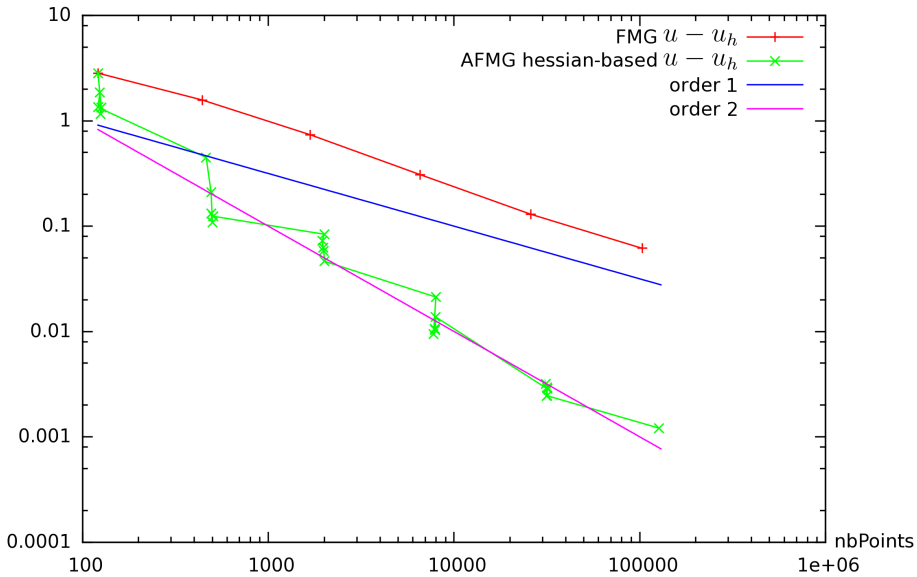

Figure 4.14: Singular internal layer: error as a function of number of vertices (with stopping criterion).

a better convergence order after refinement. But the overall accuracy is not notably changed. This gives a measure of the fiability of the stopping criterion for a singular context.

A last comparison helps to understand the essential difference between

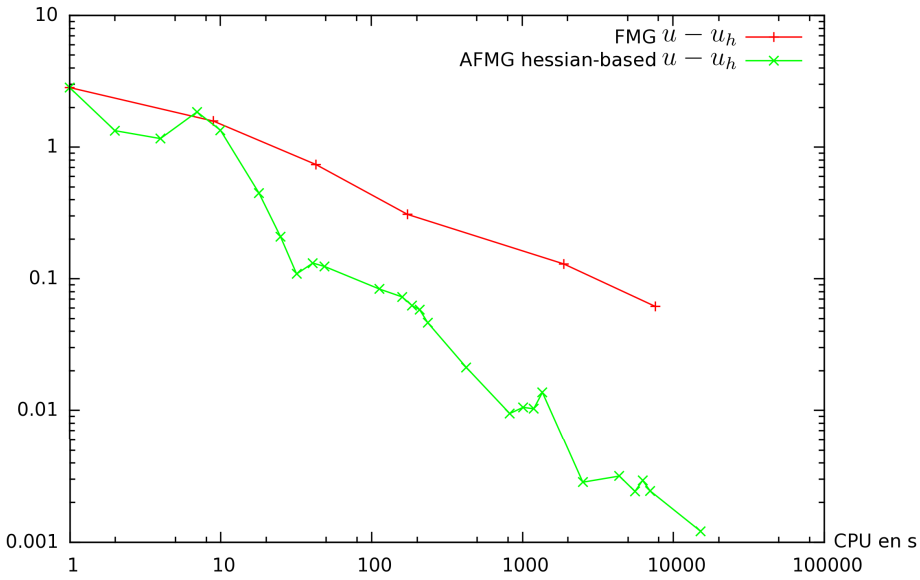

Figure 4.15: Singular internal layer: error as a function of CPU time.

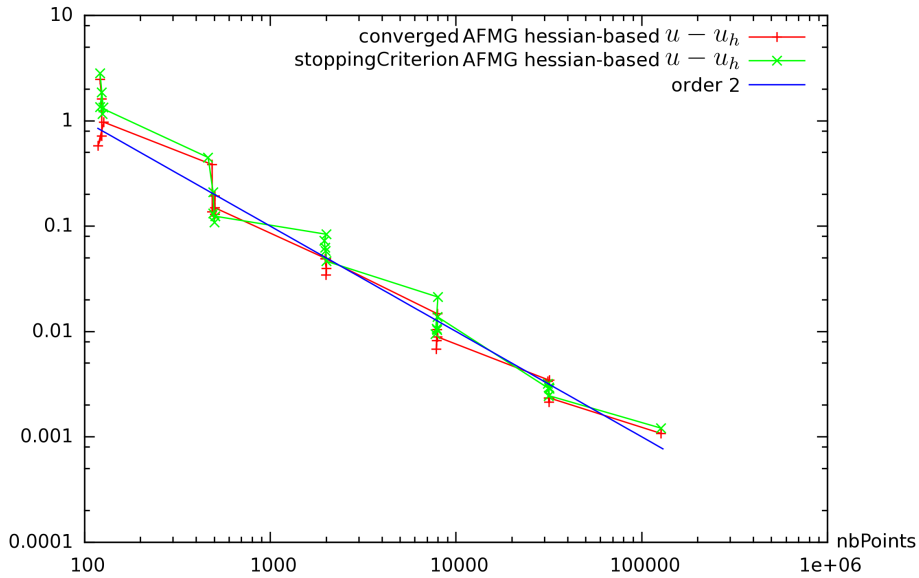

Figure 4.16: Singular internal layer: comparison between the use of the stopping criterion and the application of convergence to zero machine for solution MG iteration.

anisotropic and isotropic mesh adaptation. In [32, 59], the second-order convergence

on singularities of anisotropic adaptation was noted. In [29], some counter-examples demonstrate that isotropic adaptation will produce only first-order convergence in similar conditions. We have recomputed the adaptation sequence with same parameters (stopping criterion, four adaptations steps, same numbers of vertices) except that, in the Hessians of unknowns which are used in (7.12), both eigenvalues are replaced by the maximum of their absolute values. This produces an isotropic mesh adaptation. Figure 4.17 compares the error as function of vertices for this isotropic calculation and the error with the anisotropic one (already shown in Figures 4.14 and 4.16). We observe that isotropic adaptation does not provide a second-order convergence while the anisotropic adaptation does.

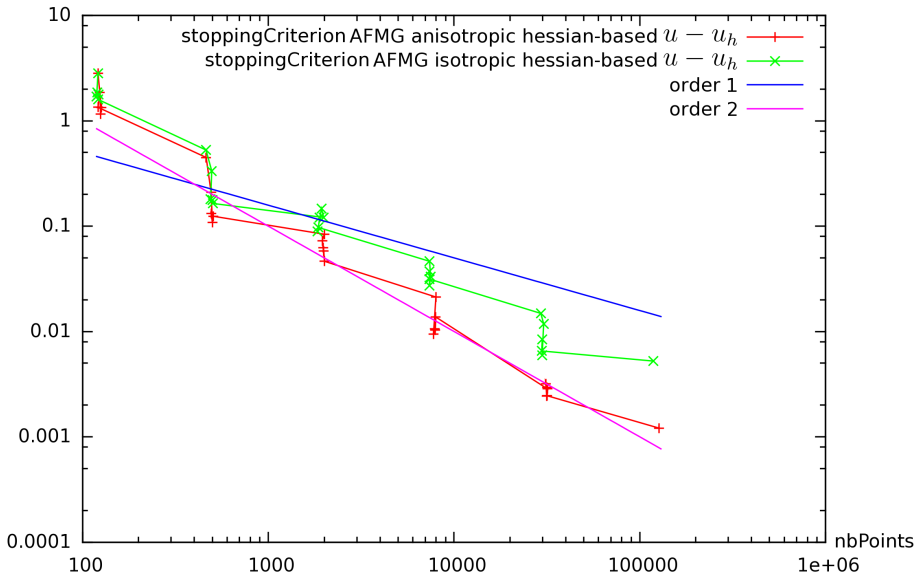

Figure 4.17: Singular internal layer: comparison between the use of anisotropic mesh adaptation and isotropic mesh adaptation.

4.7.3 Circular spot test case

Capillary models exhibit, along the interfaces, Dirac layer source terms for the pressure equation, see for example [52]. These terms imply a discontinuous pressure. For example, the pressure is equal to 1 on a disk at center and equal to 0 in the rest of the domain. This context is then even more singular than the previous one. But a very frequent approach is to combine a level-set approach with a thickening of the interface. Instead of considering a strictly discontinuous solution, we approach it by defining a thickness ε of the layer between the two uniform phases as shown in Figure 6.7. If (x, y) is located inside the thickness of the layer, $u_C(x, y)$ is given by: $u_C(x, y) = \frac{1}{2}[1 + \frac{\psi}{\varepsilon} + \frac{1}{\pi} sin(\frac{\pi\psi}{\varepsilon})]$ with $\psi = R - \sqrt{(x_C - x)^2 + (y_C - y)^2}$. The value of ε controls the thickness of the transition between $u_C = 1$ and $u_C = 0$

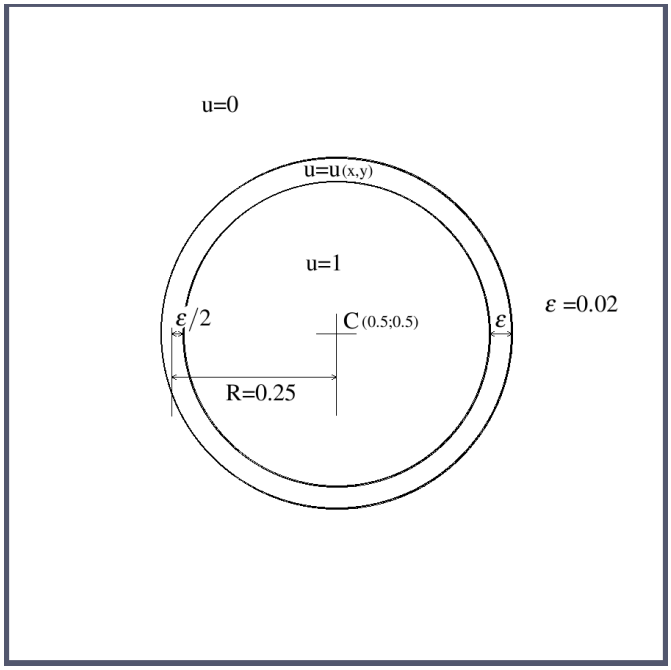

Figure 4.18: Circular spot test case: geometry

and is chosen equal to 0.02. Let $rhs = \Delta u_C$. We consider the Dirichlet problem $\Delta u = rhs$ in Ω, $u = 0$ on $\partial\Omega$. The right-hand side rhs is close to a Dirac distribution concentrated along the circle limiting the disk. In practical nonlinear situations as capillary models, that kind of feature is not *a priori* known. Then, we choose in our variational formulation to integrate the discrete RHS (rhs, N_i) (N_i: finite element basis function) on the given mesh without particular care of the quasi singularity of rhs, which means that a coarse mesh may produce a very inaccurate solution. Indeed, in our computations, we observe, in Figure 4.24, that very large errors are produced by coarse uniform meshes. With $10,000$ nodes, a 100 % L^1 error is still produced. Full second-order asymptotic convergence seems to be reached only after $30,000$ nodes are used. This behavior can be a strong handicap for 3D calculations where the number of nodes cannot be much increased. In contrast, the mesh adaptative computation produces much smaller errors with coarse meshes and always performs as well, or better. A mesh-adapted approximate solution u_h is shown in Figure 6.12. A second remark is that most gain of adaptation is obtained at about $1000-2000$ nodes while adaptation for finer meshes seems to bring no further acceleration (slope remains second-order). This phenomenon is observed also for the interpolation error of the exact solution onto the different meshes which are used. Our interpretation is that, once the layer around the circle is captured, no further adaptation is needed since a fine enough, but uniform, mesh is quasi-optimal in the vicinity of the layer described by a *sinus* function. Then, the subsequent efforts in adaptation are useless which explains that, after an interesting performance for 2000 nodes, the adaptation option does not amplify its CPU advantage.

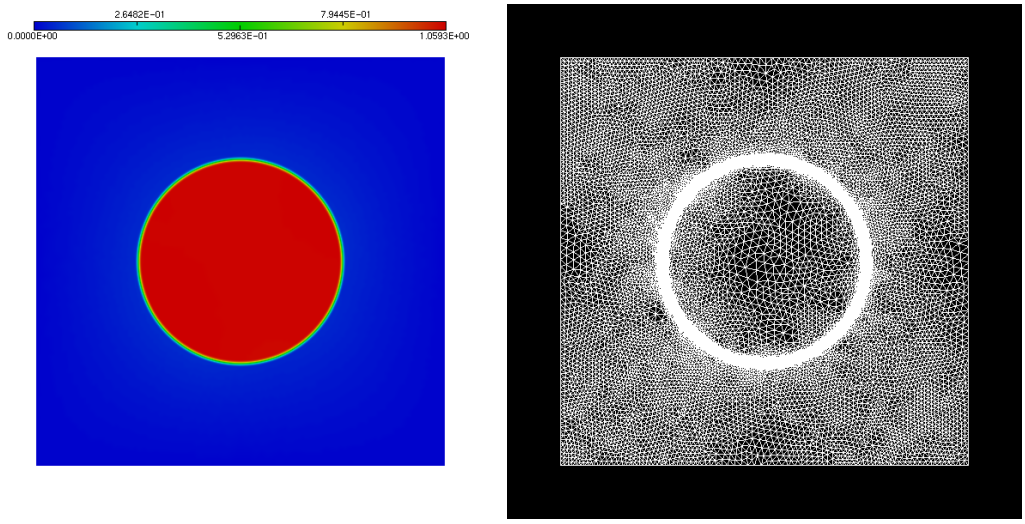

Figure 4.19: Circular spot test case solution and adapted mesh

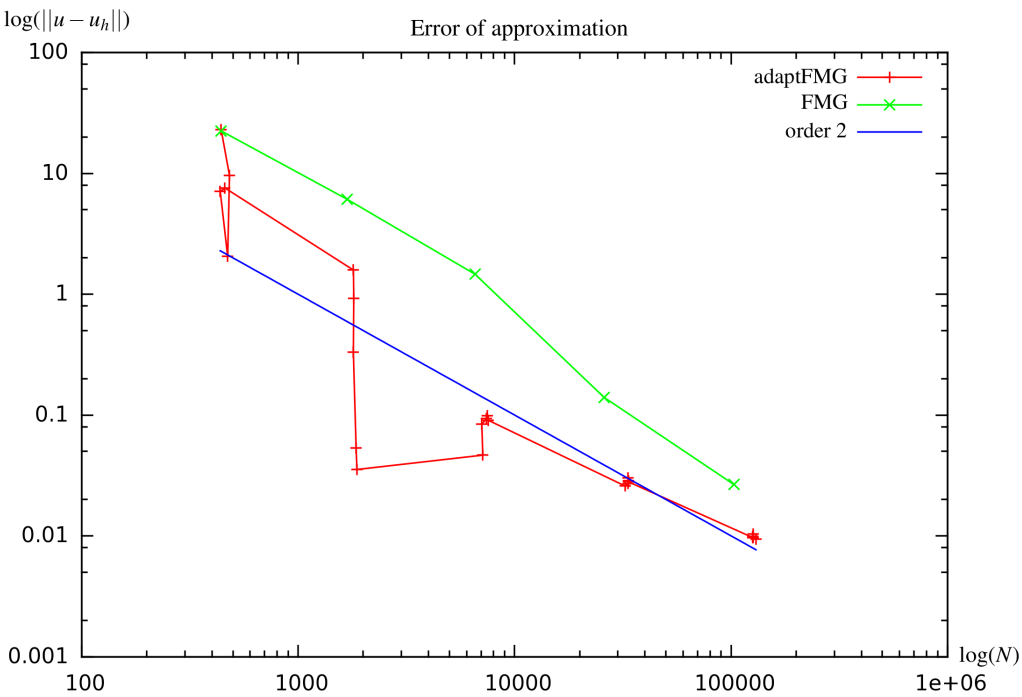

Figure 4.20: Circular spot test case: errors as functions of the number of mesh nodes. Approximation error $||u - u_h||_{L^1}$ as a function of the number of mesh nodes. ($\times$) non-adaptative FMG, ($+$) adaptative FMG. The straight line shows the second-order slope.

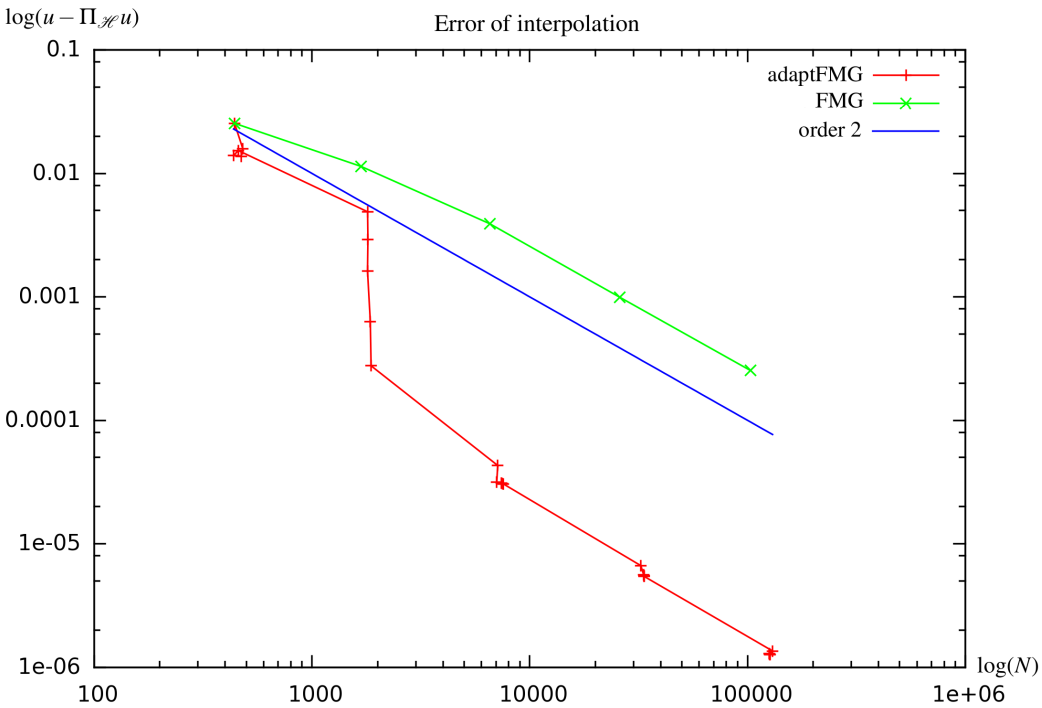

Figure 4.21: Circular spot test case. Behavior of the interpolation error of exact solution $||u - \Pi_h u||_{L^1}$ as a function of the number of mesh nodes, on the same meshes. The straight line shows the second-order slope.

4.7.4 Thinner circular test case

Now, the value of ε controlling the thickness of the transition between $u = 1$ and $u = 0$ is chosen equal to 0.001. An approximate (adapted) solution u_h is shown in Figure 4.23 (right). Now, due to the very thin definition of the Dirac-type right-hand-side, the brut force use of an uniform mesh of $100,000$ nodes does not allow the computation of a good solution, see Figure 4.23 (left). In the same figure is depicted the approximate mesh-adaptative solution with the same number of nodes. The adaptative L^1 approximation error norm is 0.3 with $10,000$ nodes and 0.04 with $100,000$ nodes.

4.8 Concluding remarks

We have proposed a combination of the well-established FMG method with an anisotropic mesh adaptation method. The mesh adaptation fixed point loop is introduced in the FMG process.

In order to master the extra computational complexity, an improved stopping criterion for MG cycling inside FMG is proposed. We believe that it is important for robustness and efficiency. We emphasize that this stopping criterion assumes that the iterative MG convergence rate is more or less norm-independant, a prop-

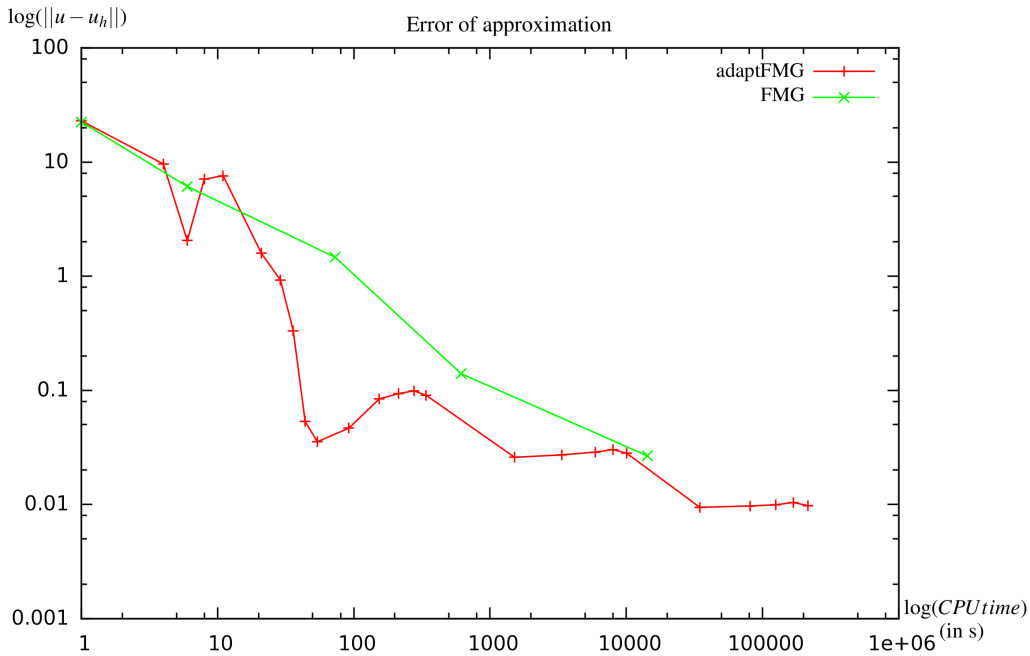

Figure 4.22: Circular spot test case: errors as a function of the CPU time. Approximation error $||u - u_h||_{L^1}$ as a function of the CPU time. ($\times$) non-adaptative FMG, ($+$) adaptative FMG. The straight line shows the second-order slope.

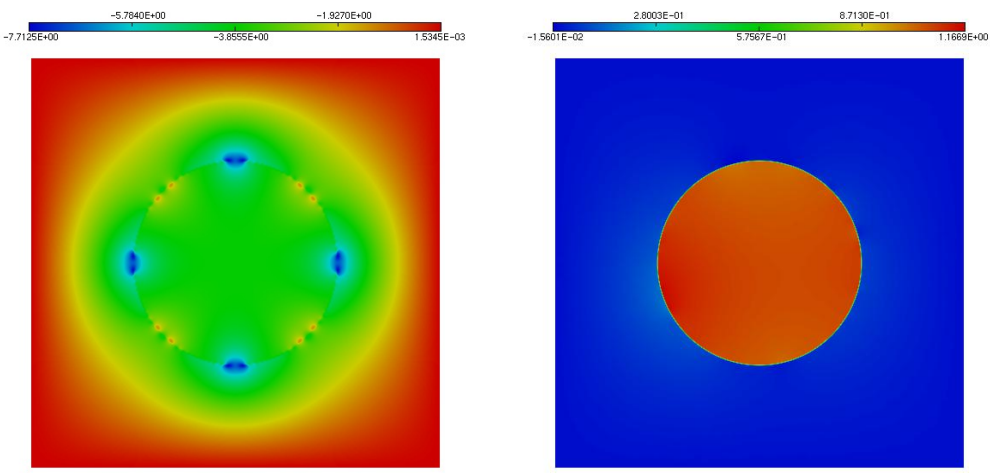

Figure 4.23: Thin-circular spot test case uniform-mesh solution and mesh-adaptative solution.

erty surely not enjoyed by many other (non-MG) solution algorithms for which our stopping criterion is definitively not recommanded.

Let us comment the choice of the main parameters in the global algorithm.

We have defined a robust MG scheme with a 10-sweeps Jacobi smoother, a saw-

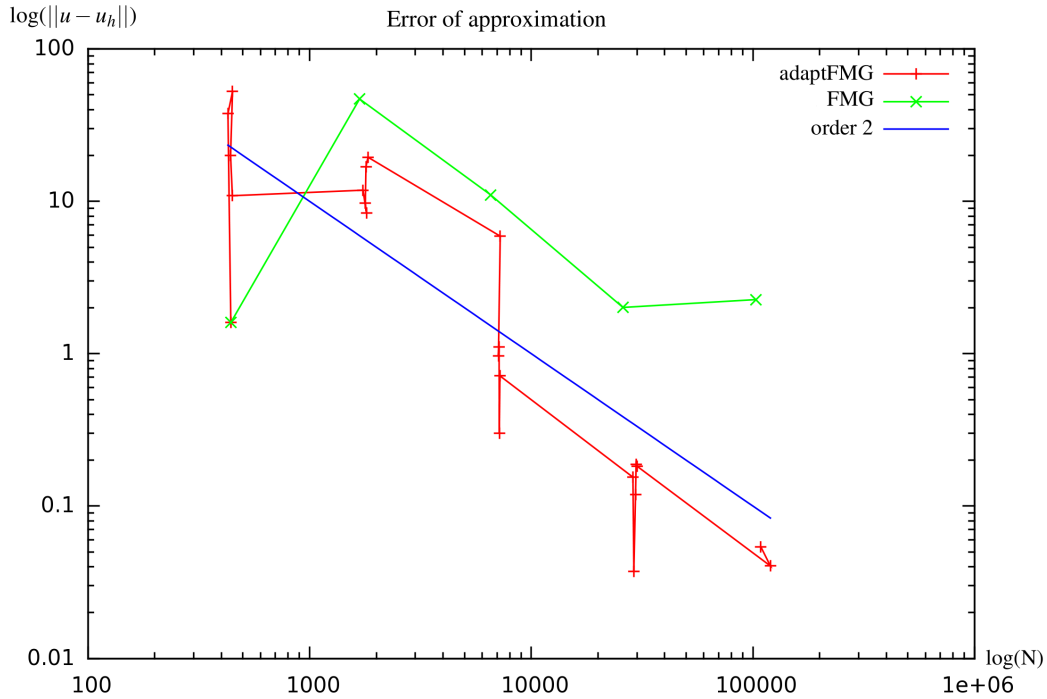

Figure 4.24: Thin circular spot test case: errors as functions of the number of mesh nodes. Approximation error $||u - u_h||_{L^1}$ as a function of the number of mesh nodes. ($\times$) non-adaptative FMG, ($+$) adaptative FMG. The straight line shows the second-order slope.

tooth V-cycle and a GMRES encapsulating the cycles. These options are chosen for computing robustly the case of discontinuous coefficients with ratios as high as 1000. Probably a better smoother can be introduced in order to improve the efficiency. Also, in the case of very stretched meshes (aspect ratio of thousands), anisotropicly generated coarse grids should be useful.

When building FMG, the mesh size ratio between two phases can indeed be more easily controlled when we use metrics. As far as strict arithmetic complexity is concerned, we remark the mesh size ratio is not a too sensitive parameter. Then, in the present 2D study, the classical choice of dividing mesh size by 2 ($N_{k_\phi}/N_{k_\phi+1} = 4$ times more nodes) is adopted. This option can be reconsidered in 3D applications where series of embedded meshes, each one with 8 times mode nodes than the previous one, involve much more vertices and are more difficult to manage.

Let us consider the stopping-criterion parameter. The theory makes it depend only of two factors: (1) the mesh ratio $N_{k_\phi}/N_{k_\phi+1} = 4$ between FMG phases and (2) the convergence order which is fixed to scheme's L^2-convergence order, i.e. 2. Convergence is order two for smooth solutions. For non-smooth problems, we conjecture that anisotropic mesh adaptation also carries second-order convergence (cf. test case 2). The FMG stopping criterion is chosen equal to 10., a conservative

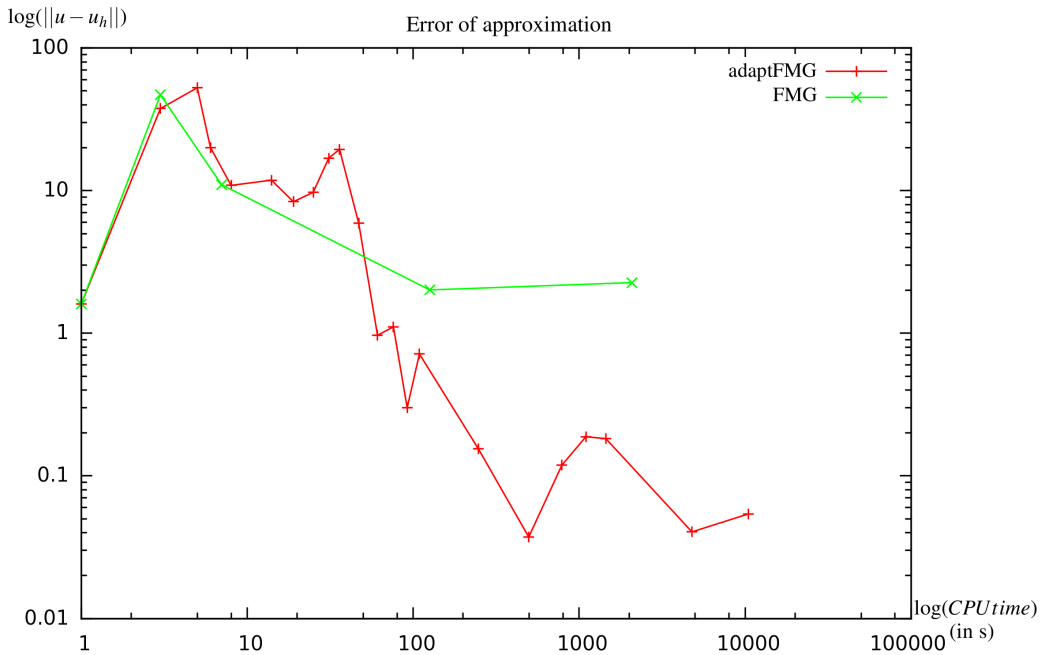

Figure 4.25: Thin circular spot test case: errors as a function of the CPU time. Approximation error $||u-u_h||_{L^1}$ as a function of the CPU time. ($\times$) non-adaptative FMG, (+) adaptative FMG. The straight line shows the second-order slope.

figure close to the 8. proposed by the theory.

The number of adaptation iterations is fixed to a minimal number of 4 once for all, in order to reduce the computational cost of adaptation while keeping most part of the error reduction.

The overall anisotropic adaptative FMG is of rather high programming/algorithmic complexity, due to the higher number of tests and embedded loops. The central question is therefore: does it enjoy a robust computational efficiency? In other words, is it computationally efficient for difficult problems?

The few numerical experiments tend to promote a positive answer. The four test cases are run with a unique set of parameters, *i.e.* without parameter tuning. In contrast to the non-adaptative case, with the use of anisotropic adaptation, we observe the early capturing of many different scales. A significative comparison between AFMG and FMG relies on the total approximation error as a function of CPU time. In most test cases, for a same CPU time, the mesh adaptative computation produces a lower or much lower approximation error. For a same CPU time, the number of nodes is much lower. A second measure concerns the asymptotic behavior in terms of number of unknowns and CPU. When it works, our FMG indeed shows $O(N)$ complexity, namely the considered norm of the total approximation error is proportional to the logarithm of the number of nodes N. This also approximatively holds for the AFMG version. For FMG, with some variations, the norm of total approximation error is also proportional to the logarithm of CPU time. Because

of our choice of limiting adaptation iteration to 4, we expect that this also holds for AFMG and we observe it, again approximatively. The discontinuous coefficient case desserves a particular mention since even for the L^1 norm of solution, the convergence on uniform meshes is first-order. The improvement in that case is of two orders of magnitude.

We have not proposed a smart stopping criterion for the adaptation loop. We plan to discuss this issue in a forthcoming paper in combination with a different mesh adaptation criterion.

Indeed, the present study relies on the Hessian-based mesh adaptative criterion. This simple and robust option has some limits which we have measured by comparing the convergence of the interpolation error and the convergence of the actual approximation error. The Hessian-based criterion is designed exactly for the interpolation error and, by the way, the interpolation error converges fastly to small values. This shows that the different approximations of u by u_h and of the Hessian of u_h do not introduce a too important penalty. The -expected- bad news is that the approximation error does not decrease as fastly and as low. We interpret this as an effect of the lack of consistency between the interpolation error and the approximation error. Introducing adjoint-based adaptation criteria may improve this issue. We plan to discuss this in a forthcoming paper using the benchmark proposed in the present work.

Of course, we are not sure that the extension to 3D will enjoy the same qualities. We expect that the gain provided by anisotropic mesh adaptation will be importantly larger as observed in several works, e.g. [4, 48]. Conversely extension of FMG to 3D carries a number of difficulties:
- FMG is less easy in 3D since a mesh ratio of 2 in terms of mesh size turns into a mesh ratio of 8 in terms of number of unknowns.
- The fabulous gain by MG happens for fine meshes. The 3D equivalent of a $10,000$ nodes 2D mesh is a $1,000,000$ nodes 3D mesh. Fortunately, our computations are already encouraging for $10,000$ nodes 2D meshes.
- Last but not least, useful computations with million nodes need to be done with parallel computing for which MG need some complex changes of strategy for coarse levels.
In our opinion, this parallel 3D extension desserves to be tried but for more practical physics. Extension to models of Continuum Mechanics can be envisaged as far as the application of MG satisfactorily works. In a forthcoming paper, 3D CFD applications are considered.

4.9 Acknowledgements

We thank Adrien Loseille and Frédéric Alauzet for the fruitful discussions we had together and help in using the softwares *yams* and *metrix*. Thanks also to Françoise Loriot from Distene. This work has been supported by French National Research Agency (ANR) through COSINUS program (projects ECINADS n^o ANR-09-COSI-

003 and MAIDESC n^o ANR-13-MONU-0010). HPC resources from GENCI-[CINES] (Grant 2010-x2010026386 and 2010-c2009025067) are also gratefully acknowledged.

Chapter 5

FMGA orienté-objectif

5.1 Introduction

Nous avons vu, dans le chapitre précédent, une méthode d'adaptation basée sur une minimisation de la norme de l'erreur d'interpolation: on cherche la métrique $\mathcal{M}$ qui minimise $||u - \Pi_{\mathcal{M}}||$ pour une norme donnée. Le principal défaut de cette méthode est qu'elle repose sur l'hypothèse qu'en interpolant le mieux possible l'inconnue discrète, on obtiendra une erreur d'approximation $u - u_{\mathcal{M}}$ la plus petite possible. Nous avons signalé au chapitre deux que cela est en toute rigueur faux. En pratique c'est une méthode heuristique pas si mauvaise que cela, notamment pour des modèles elliptiques (moins pour des modèles advectifs). La méthode dite *orientée-objectif (goal-oriented)* consiste à minimiser une véritable erreur d'approximation, à condition qu'il s'agisse d'une erreur sur une *fonctionnelle -scalaire donc- dépendant du champ inconnu.*
On consière pour simplifier une fonctionnelle $j_{goal} = (g, u)$. Le but de l'adaptation sera de minimiser $\delta j_{goal}(\mathcal{M}) = (g, u - u_{\mathcal{M}})$ par rapport au maillage. Faute de garantir la meilleure approximation de u, ni même la convergence vers u, cette approche nous assure d'obtenir la meilleure approximation de j_{goal} et la convergence de $j_{goal}(u_h)$ vers $j_{goal}(u)$.

5.2 Théorie

Partons de notre expression de δj_{goal}.

$$\delta j_{goal}(\mathcal{M}) = |(g, u - u_{\mathcal{M}})| = |(g, \Pi_{\mathcal{M}} u - u_{\mathcal{M}} + u - \Pi_{\mathcal{M}} u)|. \tag{5.1}$$

Définissons l'état adjoint discret $u^*_{g,\mathcal{M}}$:

$$\forall \psi_{\mathcal{M}} \in \ V_{\mathcal{M}}, \quad a(\psi_{\mathcal{M}}, u^*_{g,\mathcal{M}}) = (\psi_{\mathcal{M}}, g). \tag{5.2}$$

Puis:

$$(g, \Pi_{\mathcal{M}} u - u_{\mathcal{M}} + u - \Pi_{\mathcal{M}} u) = a(\Pi_{\mathcal{M}} u - u_{\mathcal{M}}, u^*_{g,\mathcal{M}}) + (g, u - \Pi_{\mathcal{M}} u)$$

et, en utilisant (6.3),

$$(g, \Pi_{\mathcal{M}} u - u_{\mathcal{M}} + u - \Pi_{\mathcal{M}} u) = a(\Pi_{\mathcal{M}} u - u, u^*_{g,\mathcal{M}}) + (u^*_{g,\mathcal{M}}, f - \Pi_{\mathcal{M}} f) + (g, u - \Pi_{\mathcal{M}} u).$$

donc

$$\delta j_{goal}(\mathcal{M}) \preceq |a(\Pi_{\mathcal{M}} u - u, u^*_{g,\mathcal{M}})| + |(u^*_{g,\mathcal{M}}, f - \Pi_{\mathcal{M}} f)| + |(g, u - \Pi_{\mathcal{M}} u)|.$$

Rappelons que u est inconnue. Le troisième terme $(g, u - \Pi_{\mathcal{M}} u)$, similaire au terme principal de l'adaptation basée hessien dans le chapitre 4, peut être approché explicitement de la même façon, *i.e.* en introduisant l'erreur d'interpolation continue:

$$\delta j_{goal}(\mathcal{M}) \preceq |a(\Pi_{\mathcal{M}} u - u, u^*_{g,\mathcal{M}})| + |(u^*_{g,\mathcal{M}}, f - \Pi_{\mathcal{M}} f)| + |(g, \pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}})|.$$

On procède de la même façon pour pour le deuxième terme $(f - \Pi_{\mathcal{M}} f, u^*_{g,\mathcal{M}})$ en introduisant l'erreur d'interpolation continue sur f comme on l'avait fait pour u:

$$\delta j_{goal}(\mathcal{M}) \preceq |a(\Pi_{\mathcal{M}} u - u, u^*_{g,\mathcal{M}})| + |(u^*_{g,\mathcal{M}}, \pi_{\mathcal{M}} f - f)| + |(g, \pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}})|.$$

Le premier terme est le plus difficile à traiter. La formule de Green nous donne:

$$a(\Pi_{\mathcal{M}} u - u, u^*_{g,\mathcal{M}}) = \sum_{K,element} \left(- \int_K (u - \Pi_{\mathcal{M}} u) div(\mu \nabla u^*_{g,\mathcal{M}}) \mathrm{d}\mathbf{x} + \int_{\partial K} \mu (u - \Pi_{\mathcal{M}} u) \nabla u^*_{g,\mathcal{M}} . \overrightarrow{n} \mathrm{d}\sigma \right)$$

Pour continuer notre analyse, nous utilisons le lemme suivant démontré dans [13].

Lemma 5.2.1
On considère une métrique $\mathcal{M}_0$ d'étirement borné par un nombre positif, et une suite de métriques de plus en plus fines proportionnelles à $\mathcal{M}_0$, ainsi que leurs maillages unité. Pour un couple de fonctions lisses (u, ϕ) où u n'est pas nécessairement une solution de (7.1) et, pour une fonction positive μ pas nécessairement continue, on a:

$$\left| \sum_{K,element} \int_{\partial K} \mu (u - \Pi_{\mathcal{M}} u) \nabla \varphi . \overrightarrow{n} \, d\sigma \right| \preceq const \int_{\Omega} \mu \; \rho(H(\varphi)) \; |u - \Pi_{\mathcal{M}} u| d\mathbf{x} \quad (5.3)$$

où $A \preceq B$ désigne une majoration asymptotiquement valide, i.e. $A \leq B + o(A)$. L'expression $\rho(H(\varphi))$ désigne la plus grande valeur propre (en valeur absolue) du hessien $H(\varphi)$ de φ. $\square$

La démonstration dans [13] fait dépendre la constante de l'étirement maximal des maillages considérés. En revanche, les calculs dans ce même papier sont réalisés avec $const. = 3$. Nous choisirons cette valeur, et considérerons donc que:

$$\left| \sum_{K,element} \int_{\partial K} \mu (u - \Pi_{\mathcal{M}} u) \nabla u^*_{g,\mathcal{M}} . \overrightarrow{n} \mathrm{d}\sigma \right| \leq 3 \int_{\Omega} \mu \rho(H(u^*_{g,\mathcal{M}})) \; |u - \Pi_{\mathcal{M}} u| \mathrm{d}\mathbf{x}.$$

Ainsi, on obtient:

$$\delta j_{goal}(\mathcal{M}) \preceq 3 \int_{\Omega} \mu \; \rho(H(\varphi)) \; |u - \Pi_{\mathcal{M}} u| \mathrm{d}\mathbf{x} + \left| \sum_{K,element} - \int_K (u - \Pi_{\mathcal{M}} u) div(\mu \nabla u^*_{g,\mathcal{M}}) \mathrm{d}\mathbf{x} \right| + |(u^*_{g,\mathcal{M}}, \pi_{\mathcal{M}} f - f)| + |(g, \pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}})|.$$

On fait maintenant l'hypothèse que μ est constante par élément ce qui nous donne:

$$\delta j_{goal}(\mathcal{M}) \preceq 3 \int_\Omega \mu\ \rho(H(\varphi))\ |u - \Pi_{\mathcal{M}} u| \mathrm{d}\mathbf{x} + |(u^*_{g,\mathcal{M}}, \pi_{\mathcal{M}} f - f)| + |(g, \pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}})|$$

et donc

$$\delta j_{goal}(\mathcal{M}) \ \preceq$$
$$\int_\Omega \left(\left[3\ \mu\ \rho(H(u^*_{g,\mathcal{M}})) + |g|\right]\ |\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}|\ +\ |u^*_{g,\mathcal{M}}|\ |\pi_{\mathcal{M}} f - f|\ \right)\ \mathrm{d}\mathbf{x}. \tag{5.4}$$

Sachant cela, on essaie de minimiser le second membre de cette inégalité au lieu de minimiser $\delta j_{goal}(\mathcal{M})$. Cela implique une difficulté car l'état adjoint $u^*_{g,\mathcal{M}}$ dépend de $\mathcal{M}$. On doit donc simplifier notre fonctionnelle δj_{goal} en gelant, pendant une partie de l'algorithme, l'état adjoint. Cela est justifié par le fait que, quand on change le paramètre $\mathcal{M}$, $u^*_{g,\mathcal{M}}$ est proche de sa limite continue (non nulle) et n'est que peu affecté par le changement de maillage, contrairement aux erreurs d'interpolation $|\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}|$ et $|\pi_{\mathcal{M}} f - f|$. De plus, au fur et à mesure que le point fixe du maillage converge, cet état adjoint change de moins en moins. On considère alors, pour une métrique donnée $\mathcal{M}_0$, le problème optimal suivant:

$$\min_{\mathcal{M}} \int_\Omega \left(\left[3\mu\rho(H(u^*_{g,\mathcal{M}_0})) + |g|\right]\ |\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}|\ +\ |u^*_{g,\mathcal{M}_0}|\ |\pi_{\mathcal{M}} f - f|\ \right)\ \mathrm{d}\Omega.$$

Cela produit une métrique optimale:

$$\mathcal{M}_{opt,\mathcal{M}_0} = \arg\min_{\mathcal{M}} |tr(\mathcal{M}^{-1/2} \left(\left[3\mu\rho(H(u^*_{g,\mathcal{M}_0})) + |g|\right]|H_u| + |u^*_{g,\mathcal{M}_0}||H_f|\right) \mathcal{M}^{-1/2})|.$$

On observe que:

$$H_{goal,0} = [3\mu\rho(H(u^*_{g,\mathcal{M}_0})) + |g|]\ |H_u|\ +\ |u^*_{g,\mathcal{M}_0}|\ |H_f|$$

est une matrice symétrique définie positive. Cela nous permet d'appliquer la même formule que dans le cas basé hessien en remplaçant simplement H_u par $H_{goal,0}$:

$$\mathcal{M}_{opt,\mathcal{M}_0} = \mathcal{K}_1(\ [3\mu\rho(H(u^*_{g,\mathcal{M}_0})) + |g|]\ |H_u|\ +\ |u^*_{g,\mathcal{M}_0}|\ |H_f|)$$

où $\mathcal{K}_1$ est définie par

$$\mathcal{K}_1(H) = D_{\mathbf{L}^1} (\det |H|)^{\frac{-1}{4}}\ |H| \ \text{ and } \ D_{\mathbf{L}^1} = N \left(\int_\Omega (\det |H|)^{\frac{1}{4}} \right)^{-1}. \tag{5.5}$$

Cette solution peut alors être introduite dans une boucle point-fixe:

Étape 1: calculer une solution discrète $u_{\mathcal{M}}$ sur le maillage $\mathcal{H}_{\mathcal{M}}$,
Étape 2: calculer l'état adjoint discret $u^*_{\mathcal{M}}$,
Étape 3: calculer la métrique optimale $\mathcal{M}^{opt}_{goal}(u_{\mathcal{M}})$,
Étape 4: prendre $\mathcal{M} = \mathcal{M}^{opt}_{goal}(u_{\mathcal{M}})$, générer un nouveau maillage $\mathcal{H}_{\mathcal{M}} = \mathcal{H}_{\mathcal{M}^{opt}_{goal}(u_{\mathcal{M}})}$ et retourner à l'étape 1, jusqu'à convergence.

L'adaptation de ce procédé au modèle d'Euler est étudiée dans [57] pour le cas stationnaire et dans [15] pour le cas instationnaire.

5.3 Cas tests

On teste cette méthode sur les mêmes cas tests que ceux effectués dans le chapitre précédent et on compare les résultats du cas goal-oriented avec ceux des cas uniforme et basé hessien.

5.3.1 Cas test circulaire discontinu

On reprend le cas test du cercle avec une discontinuité de densité du chapitre 4 illustré par la figure 4.13 et défini par la relation

$$1/\rho_1 \nabla u_1 \cdot \mathbf{n} = 1/\rho_2 \nabla u_2 \cdot \mathbf{n}$$

avec

$$u_i = \alpha_i + \beta_i (x^2 + y^2) \qquad i = 1, 2.$$

et

$$\begin{aligned} 1/\rho_1 &= 1000. \;;\; \alpha_1 = 1.23579... \;;\; \beta_1 = -2.47158... \\ 1/\rho_2 &= 1. \;;\; \alpha_2 = 100. \;;\; \beta_2 = -2471.58... \end{aligned} \tag{5.6}$$

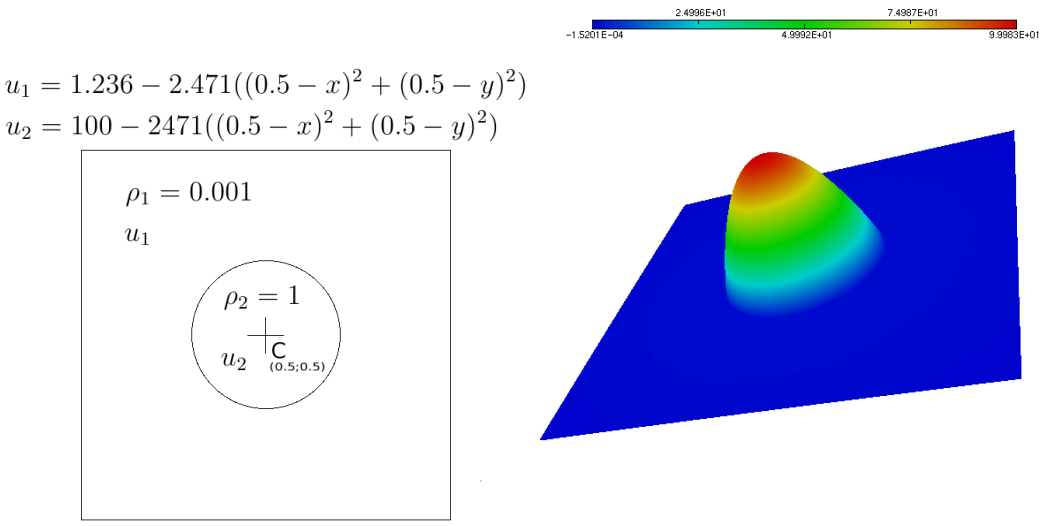

Figure 5.1: Discontinuité circulaire: définition de la solution exacte.

La figure 5.2 nous montre que la convergence goal-oriented est tout-à-fait comparable à la convergence basée hessien. Les résultats semblent de la même qualité sur les deux premières phases, puis les résultats goal-oriented sont un peu meilleurs sur les troisième et quatrième phases avant de se détériorer sur la cinquième pour finalement redevenir bien meilleurs que ceux du basé hessien dans la sixième.
La figure 5.3, en revanche, montre que la méthode goal-oriented est bien plus coûteuse en temps que la méthode basée hessiens, et cela pour un résultat pas franchement meilleur.

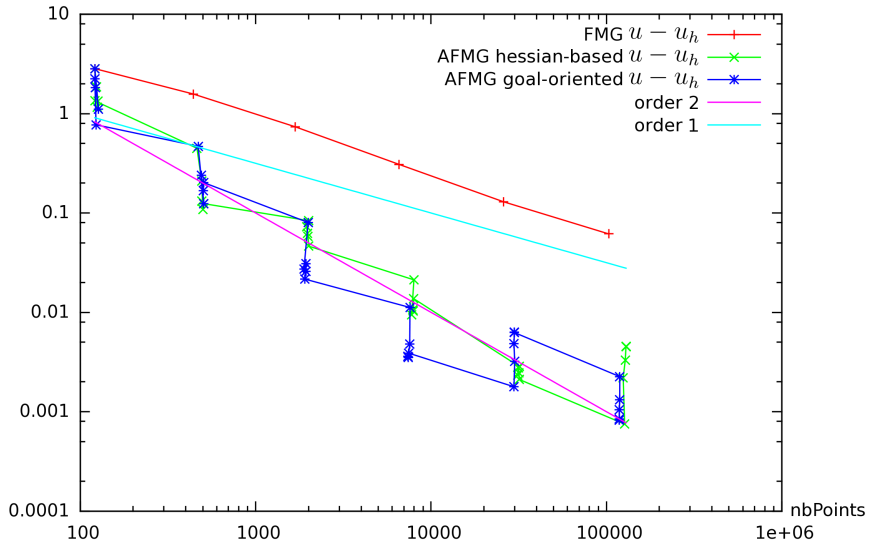

Figure 5.2: Convergences en nombre de points du cas test de la discontinuité circulaire.

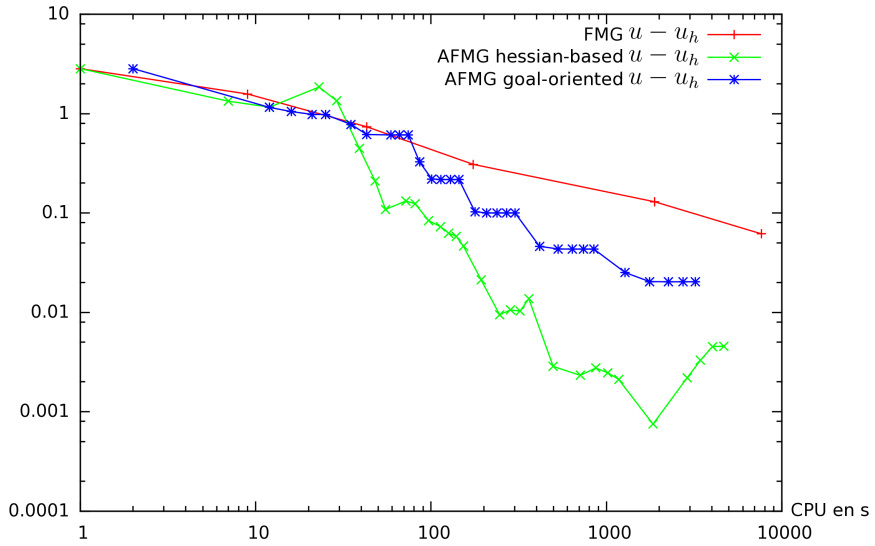

Figure 5.3: Convergences en temps CPU du cas test de la discontinuité circulaire.

5.3.2 Cas test circulaire continu avec interface fine

A nouveau, on considère un cas test déjà étudié précédemment avec le cercle défini par la figure avec, dans l'interface,

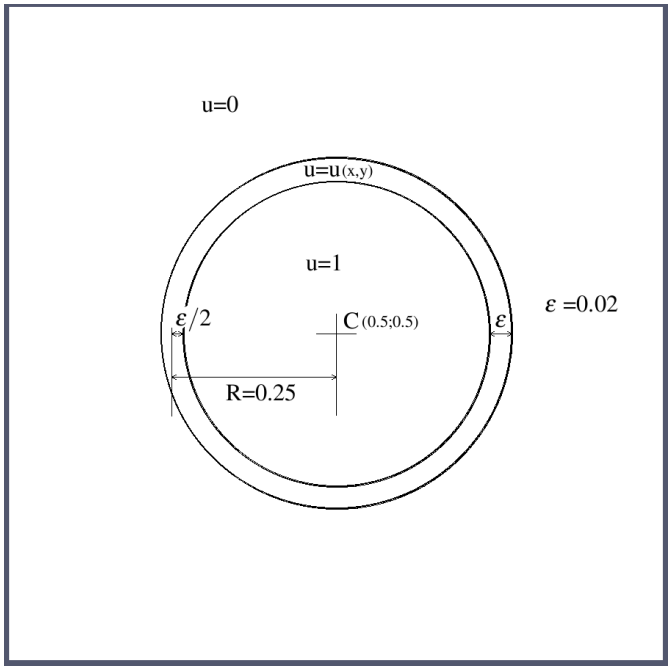

Figure 5.4: Tâche circulaire avec interface fine: géométrie.

$$u(x,y) = \frac{1}{2}[1 + \frac{\psi}{\varepsilon} + \frac{1}{\pi} sin(\frac{\pi\psi}{\varepsilon})] \text{ où } \psi = R - \sqrt{(x_C - x)^2 + (y_C - y)^2}.$$

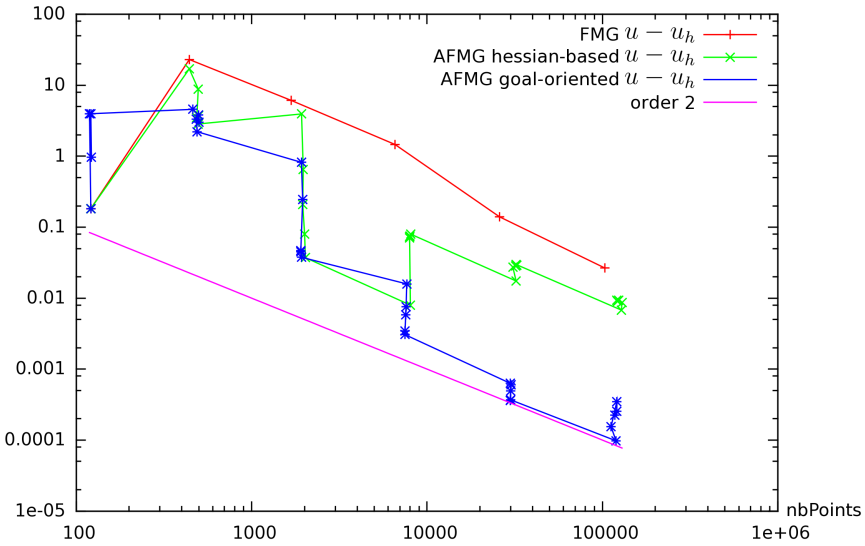

Figure 5.5: Convergences en nombre de points du cas test de la tâche circulaire à interface fine.

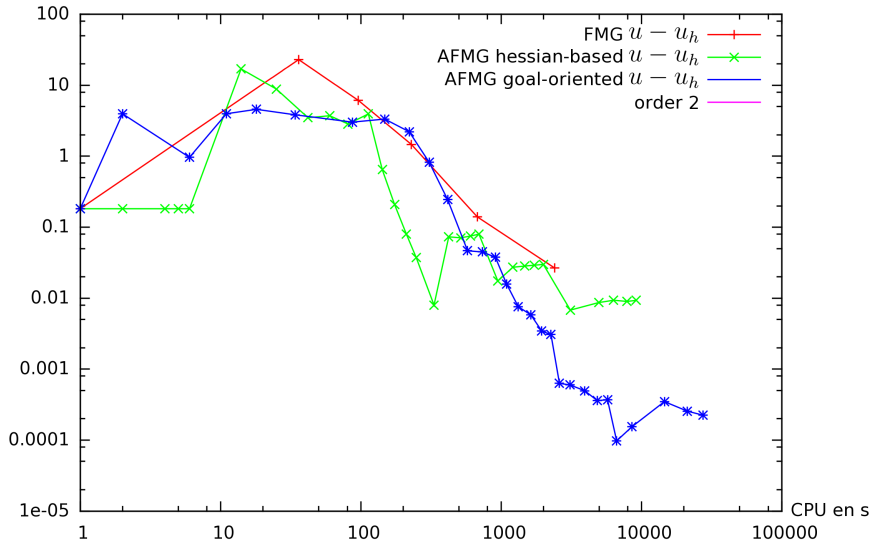

Figure 5.6: Convergences en temps CPU du cas test de la tâche circulaire à interface fine.

Contrairement au cas discontinu, on observe, sur la figure 5.5, une importante différence entre le basé hessien et le goal-oriented. A la fin de la dernière phase, l'erreur dans le cas basé hessien vaut 0.00929769 alors qu'elle n'est que de 0.00022429 dans le cas goal-oriented ce qui veut dire que la valeur de l'erreur a été divisée par plus de 40.
La figure 5.6 nous montre le temps de calcul pour les différentes méthodes. On observe ici que le goal-oriented donne les meilleurs résultats en donnant rapidement une erreur plus petite que les méthodes uniforme et basée hessien pour un même temps de calcul.

5.3.3 Couche limite

On considère à présent le cas test de la couche limite avec
$u(x,y) = [exp(1/\alpha) - 1]^{-1}\, exp(x/\alpha) + x + [exp(1/\alpha) - 1]^{-1}$. où $\alpha = 0.03$.

La figure 5.7 nous montre une convergence meilleure pour le goal-oriented que pour le basé hessien et, bien sûr, l'uniforme. Malheureusement, on observe, au fur et à mesure qu'on avance dans les phases, une perte de la convergence à l'ordre 2. Cette détérioration de la convergence est due au fait que, dans notre second membre à majorer

$$\delta j_{goal}(\mathcal{M}) \;\preceq \int_\Omega \left(\left[3\ \mu\ \rho(H(u^*_{g,\mathcal{M}})) + |g|\right]\ |\pi_\mathcal{M} u_\mathcal{M} - u_\mathcal{M}| \ +\ |u^*_{g,\mathcal{M}}|\ |\pi_\mathcal{M} f - f|\ \right)\ \mathrm{d}\mathbf{x}, \qquad (5.7)$$

le terme $|u^*_{g,\mathcal{M}}|\ |\pi_\mathcal{M} f - f|$ ne capture pas correctement les variations de f aux points

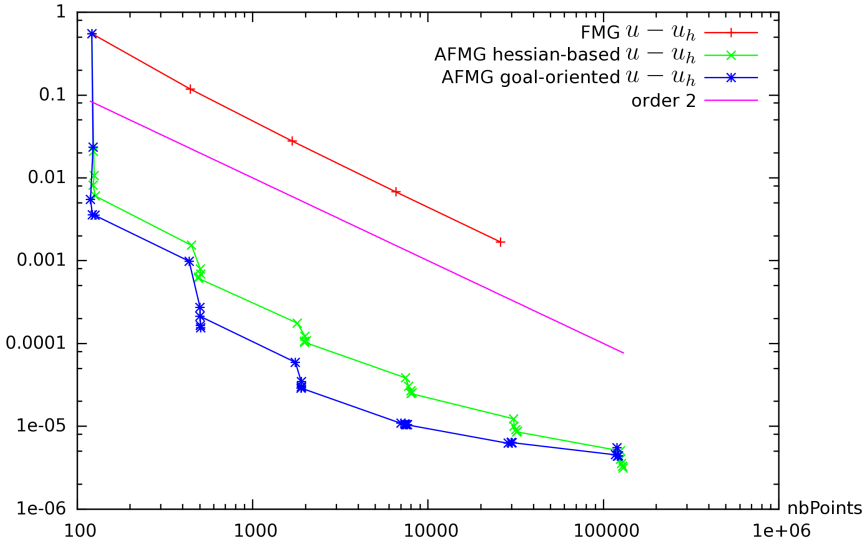

Figure 5.7: Convergence en nombre de points du test de la couche limite.

trop proches du bord parce que, en ces points, $u^*_{g,\mathcal{M}}$ tend vers 0. Nous reviendrons sur ce problème dans le prochain chapitre et une solution sera proposée.

Conclusion du chapitre

Nous avons mis en place une seconde méthode d'adaptation, basée sur une méthode goal-oriented. Cette approche prend *a priori* mieux en compte l'EDP à travers le calcul d'un état adjoint. En revanche le principe d'optimisation utilise n'assure pas une convergence globale du champ solution mais seulement celle d'une fonctionnelle scalaire dépendant de cette solution. Les résultats présentent certaines améliorations intéressantes par rapport au simple basé hessien dans certains cas. Malheureusement, dans d'autres cas, on a pu observer des détériorations. C'est pour cela que l'on va maintenant s'intéresser à une troisième méthode d'adaptation: la méthode orientée norme.

Chapter 6

FMGA orienté-norme

6.1 Avant propos

Ce chapitre contient un manuscrit “Anisotropic Norm-Oriented Mesh Adaptation for a Poisson problem “ soumis à un journal pour publication.

6.2 Abstract

We present a novel formulation for the mesh adaptation of the approximation of a Partial Differential Equation (PDE). The discussion is restricted to a Poisson problem. The proposed norm-oriented formulation extends the goal-oriented formulation since it is equation-based and uses an adjoint. At the same time, the norm-oriented formulation somewhat supersedes the goal-oriented one since it is basically a solution-convergent method. Indeed, goal-oriented methods rely on the reduction of the error in evaluating a chosen scalar output with the consequence that, as mesh size is increased (more degrees of freedom), only this output is proven to tend to its continuous analog while the solution field itself may not converge. A remarkable quality of goal-oriented metric-based adaptation is the mathematical formulation of the mesh adaptation problem under the form of the optimization, in the well-identified set of metrics, of a well-defined functional. In the new proposed formulation, we amplify this advantage. We search, in the same well-identified set of metrics, the minimum of a norm of the approximation error. The norm is prescribed by the user and the method allows addressing the case of multi-objective adaptation like, for example in aerodynamics, adaptating the mesh for drag, lift and moment in one shot. In this work, we consider the basic linear finite-element approximation and restrict our study to L^2 norm in order to enjoy second-order convergence. Numerical examples for the Poisson problem are computed.

Keywords: goal-oriented mesh adaptation, anisotropic mesh adaptation, adjoint, metric, Poisson problem, finite elements

6.3 Introduction

This paper addresses anisotropic mesh adaptation. We focus on methods which build a somewhat optimal mesh defined by a parametrization using a Riemannian

metric. A typical family of optimal metric-based methods for CFD is the family of Interpolation-based/Hessian-based methods. An attractive property of these methods is that they are founded on a mathematical optimization principle.

Iso-distribution /equi-repartition Hessian-based methods tend to minimize a Sup or L^∞ norm of the (main term of) interpolation error with respect to a metric considered in a subset of metrics with a prescribed number of vertices. We refer to the two pioneering works [25, 35]. Multiscale methods tend to minimize the L^p norm ($p < \infty$) of the interpolation error of one or several *sensors* depending on the CFD solution. Cf. [2, 77, 78, 51, 27, 3]. Sensors are fields chosen by the user according to their ability to take into account mesh-resolution difficulties of the flow to compute.

The Hessian-based methods are particularly well adapted to the finite-element approximation of second-order elliptic PDEs. It is true, by the projection theorem, that a norm of the approximation error is bounded by the analog norm of the interpolation error but this concerns the H^1 norm while Hessian-based method concentrate on L^∞ or L^p norms. More generally, while taking into account the features of the PDE *solution*, these methods do not take into account the features of the *PDE* itself. This is penalizing in the case of systems, for which several sensors need to be chosen and weighted by the user. However, if sensors are wisely chosen, a good convergence of the whole approximate solution field to the exact solution field is usually observed.

Taking into account the influence of the PDE on the error through an *equation-based estimate* has also been an important research topic. The formulation of *goal-oriented methods* is an important step for a more justified error evaluation. It has been introduced in [12]. It relies on an *a posteriori* estimate. A good synthesis concerning *a posteriori* estimates is [81], see also [36]. An interest of *a posteriori estimate* is that it is expressed in terms of the approximate solution, assumed to be available in a mesh adaptation loop. A second interest is that it does not require the use of higher-order (approximate) derivatives, in contrast to truncation analyses. These estimates show accurately where the mesh should be refined. A method for deducing a better anisotropic mesh from an *a posteriori* estimate is proposed in [38]. These methods cannot focus on an arbitrary user-specified error norm but relies on a particular one, specified by the variational formulation of the PDE. A more popular option is to chose, as accuracy target, a particular scalar output depending on the PDE solution. Any scalar output can be considered, except that difficulties can arise for the so-called non-admissible ones, according to [7]. An *a posteriori* estimate also allows for building *correctors* for goal-oriented analyses [44, 72]. In [79], the goal-oriented approach is cleverly combined with the correction strategy of [72] and with the Hessian-based metric approach, still minimizing the interpolation error of a user-prescribed sensor.

A priori estimates generally rely on Taylor series, either through divided differences or through polynomial approximation of functions. Then, approximations of higher-order derivatives of solution need to be built from the approximate solution.

This is a delicate job since there are not so many proofs ensuring that a good approximation of a higher-order derivative of the exact solution can be recovered from the low-order approximate solution. However, since the development of the first recovery methods (see for example [84]), many numerical experiments tend to show that the method is useful and rather reliable.

A remarkable feature of the goal-oriented metric-based adaptation of [56, 14] is the complete and coherent mathematical formulation of the mesh adaptation problem. Indeed, it takes the form of the optimization of a well-defined functional, namely the error for a prescribed scalar output, to be minimized with respect to a parameter, the metric, belonging to a well-identified and compact set. This strategy is applied to the discrete case in [83]. In [56, 14], in order to analytically solve the optimum, an a priori analysis is developed. It restricts to the main asymptotic term of the local error in order to exhibit more easily the dependance with respect to metric.

Goal-oriented methods have strongly impacted the applications but, due to its formulation, a goal-oriented method has two inherent limitations. First, it does not naturally extend to several scalar outputs. This "multi-target" issue is well-known and a proposition for addressing it is made in [50]. Second, because they are specialized to a given scalar output, the features of the solution field which are not influencing this output might be neglected by the automatic mesh improvement. When a goal-oriented method is used for producing a mesh-converging sequence, the convergence to continuous analog holds for the prescribed scalar output but generally does not hold for the whole flow field itself. To clarify this point, let us consider the mesh adaptative computation of a sonic boom footprint at ground. The functional depends only on pressure at ground. Now, many details of the flow on upper part of the aircraft do not influence the pressure at ground. This vanishing influence is taken into account by the adjoint state which also vanishes on these upper regions. Then, in these regions, the adapted mesh is not refined and the approximation of the flow field does not converge. See an illustration in [57]. Another limitation of a goal-oriented method is the scalar character of the error to reduce. It leads to use integrals of solution fields as in $(u - u_h, g)$, u being the exact solution, u_h its approximation and g a field prescribed by user. Now these integrals are generally not sufficiently sensitive to oscillating deviations between u and u_h.

In the new norm-oriented formulation proposed in this paper, the user can prescribe a norm or a semi-norm $|u - u_h|$ of the error in order to minimize it with respect to the mesh. As a typical example of semi-norm, this can be the sum of square deviations on particular outputs. Let us take an example in aerodynamics. The semi-norm $|u-u_h| \equiv |C_l(u)-C_l(u_h)|^2+|C_d(u)-C_d(u_h)|^2+|C_m(u)-C_m(u_h)|^2$ will account for minimizing the errors on lift, drag and moment measured from flow solution u_h with respect to mesh. The proposed method will ultimately address this kind of semi-norm, assuming that, as for the goal-oriented method, the possible issue of a non-admisssible norm according to [7] is solved. As for the goal-oriented method, the norm-oriented method takes into account the PDE features and, in the

case where a *norm* is prescribed, it produces an approximate solution field which does converge to the exact one in this norm. In this paper, the method is demonstrated with the usual linear finite-element method in 2D. This approximation is first-order accurate for H^1 norm but second-order accurate for L^2 norm which we shall consider here. The method relies on the use of a corrector field and on an *a priori* error estimate from which is extracted the asymptotically largest terms of the local error. After we have the best mesh for the prescribed norm, are we in the best of all possible worlds? Not really. We need that this norm be less than a prescribed level. The central difficulty, in our opinion, is the accurate evaluation of this error norm. Most estimates are conditioned by mesh convergence. Therefore, we compare our estimates with mesh convergence analysis and try to propose a least uncertain error evaluation. In our test cases, this evaluation is compared with the information produced by the analytic exact solution.

After a formulation of the problem, the derivation of two correctors is proposed in Sec.2. Next three sections are devoted to the three identified mesh adaptation formulations: Hessian-based, minimizing an interpolation error in Sec.3, goal-oriented formulation in Sec.4 and our proposal for a norm-oriented in Sec.5. Sec.6 is devoted to a numerical comparison between the two field-convergent formulations, *viz.* Hessian-based and norm-oriented and the paper is completed by a discussion of methods and numerical examples.

6.4 Two correctors for the Poisson problem

On the way for minimizing the error norm, let us say $|u - u_h|$, we have to replace $u - u_h$ by a so-called corrector which would be rather easy to evaluate.

6.4.1 Notations

Let $V = H_0^1(\Omega)$, Ω being a sufficiently smooth computational domain of $\mathbb{R}^2$. The continuous PDE system is written in short:

$$u \in V \mid \; Au = f \quad or \quad u \in V \mid \; \forall\, \phi \in V, \;\; a(u, \phi) = (f, \phi) \tag{6.1}$$

where

$$A = -div(\frac{1}{\rho}\nabla) \;\; ; \;\; a(u, \varphi) = \int \frac{1}{\rho}\nabla u \cdot \nabla\varphi \; dxdy$$

and where $\frac{1}{\rho}$ is a positive, possibly discontinuous, scalar field on Ω. Further, we assume that the bilinear form a is coercive in space V, *i.e.* there exists a positive α such that:

$$a(v, v) \geq \alpha |v|_V^2.$$

This model exemplifies the pressure equation in some multi-phase incompressible flow formulation, see e.g. [48]. Let $\Omega_h = \Omega$ for simplicity, τ_h a triangulation of Ω_h and V_h be the usual P_1-continuous finite-element approximation space related to τ_h:

$$V_h = \phi_h \in \mathcal{C}^0(\bar{\Omega}) \cap V, \phi_h|_T \text{ is affine } \forall T \in \tau_h.$$

The finite-element discretisation of (7.1) is written in variational and operational form:

$$u_h \in V_h \quad \text{and} \quad \forall\ \phi_h \in V_h, \quad a(u_h, \phi_h) = (f_h, \phi_h) \tag{6.2}$$

in such a way that u_h is a linear function of f_h which we denote:

$$u_h = A_h^{-1} f_h.$$

We denote by Π_h the usual interpolation operator:

$$\forall v \in \mathcal{C}^0(\bar{\Omega}) \cap V, \quad \Pi_h v \in V_h \quad \text{and,} \quad \forall\ \mathbf{x}_i \text{ vertex of } \Omega_h, \quad \Pi_h v(\mathbf{x}_i) = v(\mathbf{x}_i).$$

Scalar correctors, *i.e.* correctors for scalar outputs $j(u_h)$ depending on the solution, e.g. $j(u_h) = (g, u_h)$ with g prescribed, have been defined by Giles and Pierce, [44]. Our interest concerns the correction of the unknown field itself. Two options are now proposed.

6.4.2 A priori corrector for the PDE solution

A first rather simple *a priori* corrector can be derived from an analysis of the error RHS. We observe that:

$$a(u - u_h, \phi_h) = (f - f_h, \phi_h) \qquad \forall \phi_h \in V_h.$$

Assuming that the solution u is continuous, we get:

$$a(\Pi_h u - u_h, \phi_h) = a(\Pi_h u - u, \phi_h) + (f - f_h, \phi_h) \qquad \forall \phi_h \in V_h. \tag{6.3}$$

We call $\Pi_h u - u_h$ the *implicit error*. By implicit, we mean that it can be obtained through solving a discrete system. It differs from the approximation error by an interpolation error:

$$u - u_h \;=\; \Pi_h u - u_h \;\;+\;\; u - \Pi_h u. \tag{6.4}$$

In order to find an approximate of the implicit error, we need to evaluate the RHS of (6.3) for any test function ϕ. The second term of RHS of (6.3) is easy to evaluate (we know f and f_h). The first term of RHS of (6.3) can be transformed as follows (assuming that $\frac{1}{\rho}$ is constant on each element):

$$\begin{aligned} a(\Pi_h u - u, \phi_h) &= \sum_T \int_T \frac{1}{\rho} \nabla \phi_h \nabla (\Pi_h u - u)\ \mathrm{d}x\mathrm{d}y \\ &= \sum_T \int_{\partial T} (\Pi_h u - u) \frac{1}{\rho} \nabla \phi_h \cdot \mathbf{n}\ \mathrm{d}\sigma. \end{aligned}$$

Then, we get:

$$a(\Pi_h u - u_h, \phi_h) = \sum_{\partial T_{ij}} \frac{1}{\rho} \nabla (\phi_h|_{T_i} - \phi_h|_{T_j}) \cdot \mathbf{n}_{ij} \int_{\partial T_{ij}} (\Pi_h u - u)\ \mathrm{d}\sigma + (f - f_h, \phi_h) \tag{6.5}$$

where the sum is taken for all edges ∂T_{ij} separating triangles T_i and T_j of the triangulation. The unit vector $\mathbf{n}_{ij}$ normal to ∂T_{ij} is pointing outward T_i.

Now, we do not know u but u_h. In order to evaluate the interpolation error, we first approximate the Hessian of u by an approximation $H_h(u_h)$ in V_h of the Hessian of u_h. This is done with a Zienkiewicz-Zhu-type ([84]) recovery method defined in the seventh chapter of [62]. Then, the evaluation of $\Pi_h u - u$ is built on the edge e_{ij} as a quadratic function vanishing at both extremities of e_{ij} and of second-derivative in direction e_{ij} defined from $H_h(u_h)$. We replace $\Pi_h u - u$ by $\bar{\pi}_h u_h - u_h$ where $\bar{\pi}_h u_h - u_h$ is defined on edge ij as follows:

$$\forall\ \mathbf{x} \in\ e_{ij},\quad (\bar{\pi}_h u_h - u_h)(\mathbf{x}) = \frac{1}{2}(H_h(u_h)(\mathbf{x}_i) + H_h(u_h)(\mathbf{x}_j))(\mathbf{x} - \mathbf{x}_i)(\mathbf{x} - \mathbf{x}_j)$$

which allows a mid-edge integration on each triangle of Ω_h. We shall see, in the sequel, that it is useful to apply a similar estimate for the $f - f_h$ term, $f - f_h \approx -(\pi_h f_h - f_h)$. We define our *a priori implicit corrector* by:

$$a(\bar{u}'_{prio}, \phi_h) \ = \\ \sum_{\partial T_{ij}} \ (\frac{1}{\rho}\nabla\phi_h|_{T_i} - \nabla\phi_h|_{T_j}) \cdot \mathbf{n}_{ij} \ \int_{\partial T_{ij}} (\bar{\pi}_h u_h - u_h)\ \mathrm{d}\sigma \ - \ (\phi_h, \bar{\pi}_h f_h - f_h). \quad (6.6)$$

Then, we define our *a priori corrector* by:

$$u'_{prio} = \bar{u}'_{prio} - (\bar{\pi}_h u_h - u_h) \quad (6.7)$$

built in such a way that:

$$u'_{prio} \approx u - u_h.$$

This corrector is easy to compute but of *a priori* low accuracy.

6.4.3 Finer-grid defect correction corrector for the PDE solution

A second option for a corrector relies on using a fictitious finer grid. Let us assume that the approximation is in its asymptotic mesh convergence phase for the mesh Ω_h under study, of size h. Then, this will be also true for a strictly two-times finer embedding mesh $\Omega_{h/2}$ and, for our second-order accurate scheme applied to a sufficiently smooth problem, we would have:

$$u_h = A_h^{-1} f_h \quad , \quad u_{h/2} = A_{h/2}^{-1} f_{h/2} \quad \Rightarrow \quad u - u_{h/2} \approx \frac{1}{4}(u - u_h) \quad (6.8)$$

where u_h and $u_{h/2}$ are respectively the solutions on Ω_h and $\Omega_{h/2}$. We have also:

$$\Pi_h u - \Pi_h u_{h/2} \approx \frac{1}{4}(\Pi_h u - u_h).$$

This motivates the definition of a finer-grid Defect-Correction (DC) corrector as follows:

$$A_h \bar{u}'_{DC} = \frac{4}{3} R_{h/2 \to h}(A_{h/2} P_{h \to h/2} u_h - f_{h/2}) \quad (6.9)$$

where the residual transfer $R_{h/2\to h}$ accumulates on coarse grid vertices the values at fine vertices in neighboring coarse elements multiplied with barycentric weights and $P_{h\to h/2}$ linearly interpolates coarse values on fine mesh. In the case of local singularities, statement (6.8) is not true for uniform meshes but we have some hints that it holds almost everywhere for a sequence of adapted meshes, according to [59]. The DC corrector $\bar{u}'_{DC}$ approximates $\Pi_h u - u_h$ instead of $u - u_h$ and can be corrected as the previous one:

$$u'_{DC} = \bar{u}'_{DC} - (\pi_h u_h - u_h). \tag{6.10}$$

6.5 Interpolation error optimization

Main notions and notations are stated by recalling the basics of the Hessian-based approach.

6.5.1 Mesh parametrization

We propose to work in the continuous mesh framework, introduced in [54, 55]. The main idea of this framework is to model discrete meshes by continuous Riemannian metric fields. It allows us to define the adaptation problem as a differentiable optimization problem *i.e.,* to apply, on the class continuous metrics, a calculus of variations which cannot be applied on the class of discrete meshes. This framework lies in the class of metric-based methods. A continuous mesh $\mathcal{M}$ of the computational domain Ω is identified to a Riemannian metric field [16] $\mathcal{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ where $\mathcal{M}(\mathbf{x})$ is a symmetric 3×3 matrix. We define the *total number of vertices* of $\mathcal{M}$ as:

$$\mathcal{C}(\mathcal{M}) = \int_\Omega \sqrt{\det(\mathcal{M}(\mathbf{x}))}\,\mathrm{d}\mathbf{x}.$$

Given a continuous mesh $\mathcal{M}$, we shall say that a discrete mesh $\mathcal{H}$ of the same domain Ω is a *unit mesh with respect to* $\mathcal{M}$ if each triangle $K \in \mathcal{H}$, defined by its list of edges $(\mathbf{a}_i\mathbf{b}_i)_{i=1\ldots 3}$, verifies:

$$\forall i \in [1,3], \quad \int_0^1 \sqrt{{}^t\mathbf{a}_i\mathbf{b}_i\ \mathcal{M}(\mathbf{a}_i + t\,\mathbf{a}_i\mathbf{b}_i)\ \mathbf{a}_i\mathbf{b}_i}\ \mathrm{d}t \in \left[\frac{1}{\sqrt{2}}, \sqrt{2}\right].$$

The rest of the paper will try to find the best metric $\mathcal{M}$ from an error analysis which is asymptotic with respect to mesh size.

6.5.2 Interpolation-based optimal metric

Let u be any sufficiently smooth function defined on Ω. Let $\mathcal{M}$ be a mesh/metric of Ω. In the rest of the paper, $\mathcal{M}$ replaces h as the discretization index. We consider only meshes $\mathcal{M}$ involving enough nodes for justifying the replacement of the complete error by its main asymptotic part. The P^1 interpolation error $|\Pi_{\mathcal{M}} u - u|$ can be approximated in terms of second derivatives of u and of the metric $\mathcal{M}$ by the *continuous interpolation error:*

$$|\Pi_{\mathcal{M}} u - u| \approx |u - \pi_{\mathcal{M}} u|$$

with:

$$|u - \pi_{\mathcal{M}} u|(\mathbf{x}) \quad = \quad \frac{1}{10}\text{trace}(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})\,|H_u(\mathbf{x})|\,\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})) \tag{6.11}$$

where $|H_u|$ is deduced from H_u by taking the absolute values of its eigenvalues. Starting from:

$$\|u - \pi_{\mathcal{M}} u\|_{\mathbf{L}^p(\Omega_h)} = \left(\int_\Omega \left(\text{trace}\big(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})|H_u(\mathbf{x})|\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})\big)\right)^p \mathrm{d}\mathbf{x}\right)^{\frac{1}{p}}, \tag{6.12}$$

we define as optimal metric the one which minimizes the right hand side under the constraint of a total number of vertices equal to a parameter N. After solving analytically this optimization problem (see e.g. [25, 2, 77]), we get the unique optimal $(\mathcal{M}_{\mathbf{L}^p}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ as:

$$\mathcal{M}_{\mathbf{L}^p} = \mathcal{K}_p(H_u) \quad \text{with}$$
$$\mathcal{K}_p(H_u) = D_{\mathbf{L}^p}\,(\det|H_u|)^{\frac{-1}{2p+2}}\,|H_u| \quad \text{and} \quad D_{\mathbf{L}^p} = N\left(\int_\Omega (\det|H_u|)^{\frac{p}{2p+2}}\right)^{-1}, \tag{6.13}$$

where $D_{\mathbf{L}^p}$ is a global normalization term set to obtain a continuous mesh with complexity N and $(\det|H_u|)^{\frac{-1}{2p+2}}$ is a local normalization term accounting for the sensitivity of the $\mathbf{L}^p$ norm. In the case of an adaptation loop for solving a Partial Differential Equation, a continuous function u is not available but an approximate solution $u_{\mathcal{M}}$ is. In that case, the continuous interpolation error (6.11) is replaced by:

$$|u_{\mathcal{M}} - \pi_{\mathcal{M}} u_{\mathcal{M}}|(\mathbf{x}) \quad = \quad \frac{1}{10}\text{trace}(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})\,|H_{u_{\mathcal{M}}}(\mathbf{x})|\,\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})) \tag{6.14}$$

where $H_{u_{\mathcal{M}}}$ is an approximate Hessian evaluated with the recovery method.

According to the continuous mesh framework, statement (7.12) defines directly a continuous optimal metric. In practice, solving (7.12) is done by approximation, *i.e.* in a discrete context with a couple (mesh, solution) denoted $(\mathcal{H}_{\mathcal{M}}, u_{\mathcal{M}})$ and iteratively through the following fixed point:
Step 1: compute the discrete state $u_{\mathcal{M}}$ on mesh $\mathcal{H}_{\mathcal{M}}$,
Step 2: compute sensor $s_{\mathcal{M}} = s(u_{\mathcal{M}})$ and optimal metric $\mathcal{M}^{opt}_{inter} = \mathcal{K}_p(H_{\mathcal{M}}(s_{\mathcal{M}}))$,
Step 3: put $\mathcal{M} = \mathcal{M}^{opt}_{inter}$, generate a new mesh $\mathcal{H}_{\mathcal{M}} = \mathcal{H}_{\mathcal{M}^{opt}_{inter}}$ and go to 1, until convergence.

In the above algorithm, the continuous Hessian of s is replaced by an approximate Hessian $H_{\mathcal{M}}(s_{\mathcal{M}})$, evaluated by the patch-recovery approximation defined in [62]. In our Hessian-based numerical examples, the L^2 case, $p = 2$, has been considered. The above notation $\mathcal{K}_p$ will also be used in the next sections for $p = 1$.

6.6 Implicit a priori error estimate

In contrast to a corrector as defined in Section 6.4, an asymptotic upper bound of the approximation error should allow an easier error reduction by minimisation of its norm with respect to the metric. In our PDE discretisation notations, we henceforward replace in the discretization index h by the index $\mathcal{M}$ which holds for any unit mesh of the metric $\mathcal{M}$. The implicit *a priori* estimate (7.5) then writes:

$$\forall\ \phi \in\ V_{\mathcal{M}},\ \ a(\Pi_{\mathcal{M}} u - u_{\mathcal{M}}, \phi)\ =$$
$$\sum_{\partial T_{ij}} \frac{1}{\rho}(\nabla\phi|_{T_i} - \nabla\phi|_{T_j}) \cdot \mathbf{n}_{ij} \int_{\partial T_{ij}} (\Pi_{\mathcal{M}} u - u)\ \mathrm{d}\sigma\ -\ (\phi, \Pi_{\mathcal{M}} f - f) \quad (6.15)$$

where T_{ij} are the triangles of a unit mesh for $\mathcal{M}$ and the proposed corrector is expressed with the discrete solution:

$$\begin{aligned} a(\bar{u}'_{prio}, \phi) &= K(\mathcal{M}, \phi, u_{\mathcal{M}}) \quad \forall\ \phi \in\ V_{\mathcal{M}} \qquad \text{with} \\ K(\mathcal{M}, \phi, u_{\mathcal{M}}) &= \sum_{\partial T_{ij}} \frac{1}{\rho}(\nabla\phi|_{T_i} - \nabla\phi|_{T_j}) \cdot \mathbf{n}_{ij} \int_{\partial T_{ij}} (\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}})\ \mathrm{d}\sigma \\ &\quad -\ (\phi, \pi_{\mathcal{M}} f - f), \\ u'_{prio} &= \bar{u}'_{prio} - (\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}). \end{aligned} \quad (6.16)$$

We now restart from (7.5). The following result is proven in [13]:

Lemma 6.6.1
We assume that the metric anisotropy is bounded by a positive number. For any smooth couple of functions (u, ϕ), where u is not necessarily a solution of (7.1), we have the following bound:

$$|\int_\Omega \frac{1}{\rho} \nabla(u - \Pi_{\mathcal{M}} u) \nabla \Pi_{\mathcal{M}} \varphi d\mathbf{x}|\ \preceq\ K\ \int_\Omega \frac{1}{\rho}\ \bar{\rho}(H(\varphi))\ |u - \Pi_{\mathcal{M}} u| d\mathbf{x} \quad (6.17)$$

where $A \preceq B$ holds for a majoration asymptotically valid, i.e. $A \leq B + o(A)$. Expression $\bar{\rho}(H(\varphi))$ holds for the largest (in absolute value) eigenvalue of the Hessian $H(\varphi)$ of φ. $\square$

The next section shows two ways in using this estimate.

6.7 Equation-based adaptation

6.7.1 Scalar output "goal-oriented" analysis

The goal-oriented analysis relies on the minimization of the error $\delta j_{goal}(\mathcal{M})$ done in the evaluation of the scalar output $j = (g, u)$, error which we simplify as follows:

$$\delta j_{goal}(\mathcal{M}) = |(g, u - u_{\mathcal{M}})| = |(g, \Pi_{\mathcal{M}} u - u_{\mathcal{M}} + u - \Pi_{\mathcal{M}} u)|. \quad (6.18)$$

Let us define the discrete adjoint state $u^*_{g,\mathcal{M}}$:

$$\forall \psi_{\mathcal{M}} \in\ V_{\mathcal{M}}, \quad a(\psi_{\mathcal{M}}, u^*_{g,\mathcal{M}}) = (\psi_{\mathcal{M}}, g). \quad (6.19)$$

Then:

$$(g, \Pi_{\mathcal{M}} u - u_{\mathcal{M}} + u - \Pi_{\mathcal{M}} u) = a(\Pi_{\mathcal{M}} u - u_{\mathcal{M}}, u^*_{g,\mathcal{M}}) + (g, u - \Pi_{\mathcal{M}} u)$$

and, using (6.3),

$$(g, \Pi_{\mathcal{M}} u - u_{\mathcal{M}} + u - \Pi_{\mathcal{M}} u) = a(\Pi_{\mathcal{M}} u - u, u^*_{g,\mathcal{M}}) + (f - \Pi_{\mathcal{M}} f, u^*_{g,\mathcal{M}}) + (g, u - \Pi_{\mathcal{M}} u)$$

thus

$$\delta j_{goal}(\mathcal{M}) \approx |a(\Pi_{\mathcal{M}} u - u, u^*_{g,\mathcal{M}}) + (f - \Pi_{\mathcal{M}} f, u^*_{g,\mathcal{M}}) + (g, u - \Pi_{\mathcal{M}} u)|$$

Recall that u is unknown. The second term $\Pi_{\mathcal{M}} u - u$, similar to the main term of the Hessian-based adaptation in Section 6.5.2, can be explicitly approached in the same way, *i.e.* introducing the continuous interpolation error (6.14):

$$\delta j_{goal}(\mathcal{M}) \preceq |a(\Pi_{\mathcal{M}} u - u, u^*_{g,\mathcal{M}})| + |(f - \Pi_{\mathcal{M}} f, u^*_{g,\mathcal{M}})| + |g||\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}|$$

For the second $\Pi_{\mathcal{M}} u - u$ term, we first apply Lemma 6.6.1 and then also introduce the continuous interpolation error. We get

$$\delta j_{goal}(\mathcal{M}) \preceq \int_\Omega \left(\left[\frac{1}{\rho}\bar{\rho}(H(u^*_{g,\mathcal{M}})) + |g|\right] |\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}| + |u^*_{g,\mathcal{M}}| \, |\pi_{\mathcal{M}} f - f| \right) \mathrm{d}\Omega.$$

It is then reasonable to try to minimize the RHS of this inequality instead of the LHS. But this involves still some difficulty due to the dependancy of adjoint state $u^*_{g,\mathcal{M}}$ with respect to $\mathcal{M}$. We shall further simplify our functional by freezing, during a part of the algorithm, the adjoint state. The idea is that, when we change the parameter $\mathcal{M}$, $u^*_{g,\mathcal{M}}$ is close to its (non-zero) continuous limit and is not much affected, in contrast to the interpolation errors $|\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}|$ and $|\pi_{\mathcal{M}} f - f|$. We then consider, for a given $\mathcal{M}_0$, the following optimum problem:

$$\min_{\mathcal{M}} \int_\Omega \left(\left[\frac{1}{\rho}\bar{\rho}(H(u^*_{g,\mathcal{M}_0})) + |g|\right] |\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}| + |u^*_{g,\mathcal{M}_0}| \, |\pi_{\mathcal{M}} f - f| \right) \mathrm{d}\Omega.$$

This will produce an optimum:

$$\mathcal{M}_{opt,\mathcal{M}_0} = \arg\min_{\mathcal{M}} |tr(\mathcal{M}^{-1/2} \left(\left[\frac{1}{\rho}\bar{\rho} H(u^*_{g,\mathcal{M}_0}) + |g|\right] |H_u| + |u^*_{g,\mathcal{M}_0}||H_f| \right) \mathcal{M}^{-1/2})|.$$

Observing that, in the integrand,

$$H_{goal,0} = [\frac{1}{\rho}\bar{\rho}(H(u^*_{g,\mathcal{M}_0})) + |g|] \, |H_u| + |u^*_{g,\mathcal{M}_0}| \, |H_f|$$

is a positive symmetric matrix, we can apply the above calculus of variation and get:

$$\mathcal{M}_{opt,\mathcal{M}_0} = \mathcal{K}_1(\, [\frac{1}{\rho}\bar{\rho}(H(u^*_{g,\mathcal{M}_0})) + |g|] \, |H_u| + |u^*_{g,\mathcal{M}_0}| \, |H_f|)$$

where $\mathcal{K}_1$ is defined in (7.12). This solution can then be introduced in a fixed-point loop and will produce the solution of:

$$\mathcal{M}_{opt,goal} = \mathcal{K}_1(\, [\frac{1}{\rho}\bar{\rho}(H(u^*_{g,\mathcal{M}_{opt,goal}})) + |g|] \, |H_u| \; + \; |u^*_{g,\mathcal{M}_{opt,goal}}| \, |H_f|).$$

Let us precise how the discrete algorithm is organised:
Step 1: compute the discrete state $u_{\mathcal{M}}$ on mesh $\mathcal{H}_{\mathcal{M}}$,
Step 2: compute the discrete adjoint state $u^*_{\mathcal{M}}$,
Step 3: compute optimal metric $\mathcal{M}^{opt}_{goal}(u_{\mathcal{M}})$,
Step 4: put $\mathcal{M} = \mathcal{M}^{opt}_{goal}(u_{\mathcal{M}})$, generate a new mesh $\mathcal{H}_{\mathcal{M}} = \mathcal{H}_{\mathcal{M}^{opt}_{goal}(u_{\mathcal{M}})}$ and go to 1, until convergence.

The adaptation of this process to the Euler model of Gas Dynamics is studied in [57] for the steady case and in [15] for the unsteady case.

6.7.2 Norm-based functional

We are now interested by the minimization of the following expression with respect to the mesh $\mathcal{M}$:

$$\delta j(\mathcal{M}) = ||u - u_{\mathcal{M}}||^2_{L^2(\Omega)}. \tag{6.20}$$

Introducing $g_{\mathcal{M}} = u - u_{\mathcal{M}}$, we get a formulation similar to the goal-oriented formulation:

$$\delta j(\mathcal{M}) = (g_{\mathcal{M}}, u - u_{\mathcal{M}}). \tag{6.21}$$

Let us define the discrete adjoint state $u^*_{\mathcal{M}}$:

$$\forall \psi_{\mathcal{M}} \in \; V_{\mathcal{M}}, \quad a(\psi_{\mathcal{M}}, u^*_{\mathcal{M}}) = (\psi_{\mathcal{M}}, g_{\mathcal{M}}). \tag{6.22}$$

Then, similarly to Section 6.7.1, we have to solve the following optimum problem.

$$\min_{\mathcal{M}} \int_{\Omega} \left(\; [\frac{1}{\rho}\bar{\rho}(H(u^*_{\mathcal{M}})) + |g_{\mathcal{M}}|] \; |\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}| \; + \; |u^*_{\mathcal{M}}| \, |\pi_{\mathcal{M}} f - f| \; \right) \, \mathrm{d}\Omega.$$

Exactly as for Section 6.7.1, we freeze the dependancy of the adjoint state.

$$\min_{\mathcal{M}} \int_{\Omega} \left(\; [\frac{1}{\rho}\bar{\rho}(H(u^*_{\mathcal{M}_0})) + |g_{\mathcal{M}}|] \; |\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}| \; + \; |u^*_{\mathcal{M}_0}| \, |\pi_{\mathcal{M}} f - f| \; \right) \, \mathrm{d}\Omega.$$

$$\mathcal{M}_{opt,\mathcal{M}_0} = \mathcal{K}_1(\, [\frac{1}{\rho}\bar{\rho}(H(u^*_{\mathcal{M}_0})) + |g_{\mathcal{M}}|] \, |H_u| \; + \; |u^*_{\mathcal{M}_0}| \, |H_f|).$$

In practice, the $g_{\mathcal{M}}$ RHS in (7.24-7.26) is replaced by the corrector $u'_{prio,\mathcal{M}}$. In order to get the final norm-oriented optimum $\mathcal{M}_{opt,norm}$, we apply:

Step 1: first solve the linearised corrector system:

$$a(\bar{u}'_{prio,\mathcal{M}}, \phi) = \sum_{\partial T_{ij}} (\nabla\phi|_{T_i} - \nabla\phi|_{T_j}) \cdot \mathbf{n}_{ij} \int_{\partial T_{ij}} (\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}) \, \mathrm{d}\sigma \; - \; (\phi, \pi_{\mathcal{M}} f_{\mathcal{M}} - f_{\mathcal{M}}). \quad (6.23)$$

where mesh is unit for $\mathcal{M}$ and $\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}$ is expressed in terms of metric and Hessian, as in (6.6).

Put $u'_{prio,\mathcal{M}} = \bar{u}'_{prio,\mathcal{M}} - (\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}})$.

Step 2: Then, solve the adjoint system:

$$a(\psi, u^*_{prio,\mathcal{M}}) = (u'_{prio,\mathcal{M}}, \psi). \quad (6.24)$$

Step 3: Finally, put:

$$\mathcal{M}^{(\alpha+1)} = \mathcal{K}_1([|u'_{prio}| + \frac{1}{\rho}\bar{\rho} H(u^*_{prio})] \; |H_{u_{\mathcal{M}}}| + |u^*_{prio}||H_f|) \quad (6.25)$$

the three-step process being re-iterated until we get a fixed point $\mathcal{M}_{opt,norm} = \mathcal{M}^{(\infty)}$.

The new algorithm involves the solution of two extra linear systems with respect to a basic Hessian-based algorithm and one extra linear system with respect to the goal-oriented algorithm.

Remark: In contrast to the adjoint of the goal-oriented algorithm, these auxiliary variables are not consistent with a continuous adjoint (and do not converge towards it when mesh is refined). They are correctors and converge to zero. We assume that our data and solution are sufficiently regular for applying the Aubin-Nitsche L^2 analysis. This implies the second-order L^2 convergence of approximate solution to exact one, *i.e.* $|u_{\mathcal{M}} - u|_{L^2} = O(h^2)$. This tends to indicate that our corrector norm $|u'_{prio,\mathcal{M}}|_{L^2}$ is also $O(h^2)$. We write this in terms of the number N of vertices for metric $\mathcal{M}$ as follows:

$$|u'_{prio,\mathcal{M}}| = O(N^{-1}).$$

Further, following a Richardson expansion:

$$u'_{prio,\mathcal{M}} \approx N^{-1} u'_{prio} \quad (6.26)$$

where u'_{prio} is the limit (assumed to exist) of $N u'_{prio,\mathcal{M}}$ when mesh is refined. Similarly, we can put:

$$u^*_{prio,\mathcal{M}} \approx N^{-1} u^*_{prio}. \quad (6.27)$$

In particular, when passing from a mesh $\mathcal{M}_1$ with N_1 vertices, $\mathcal{M}_2$ with N_2 vertices, approximations of $u'_{prio,\mathcal{M}_2}$ and $u^*_{prio,\mathcal{M}_2}$ can be defined as

$$u'_{prio,\mathcal{M}_2} \approx \frac{N_1}{N_2} u'_{prio,\mathcal{M}_1} \quad ; \quad u^*_{prio,\mathcal{M}_2} \approx \frac{N_1}{N_2} u^*_{prio,\mathcal{M}_1}. \square \quad (6.28)$$

6.8 Numerical examples

We restrict our study to a benchmark of two-dimensional Poisson problems. We conjecture that the two following mesh adaptation methods produce L^2 convergent solutions to continuous. The first method, the Hessian-based method (with $p = 2$), is just heuristically relying on usual finite-element estimates. The second method, our novel norm-oriented method, is directly built on the minimisation of the L^2 error norm. We do not consider goal-oriented applications for which examples of computations can be found in [57] and [15]. As already remarked, the convergence of goal-oriented solutions to continuous is definitively questionable.

6.8.1 Numerical features

In [21], a mesh-adaptative full-multigrid (FMG) algorithm relying on the Hessian-based adaptation criterion is designed. We first describe in short this algorithm for the Hessian-based option. A sequence of numbers N_k of vertices is specified from a coarse mesh to finer one $N_0 = N, N_1 = 4N, N_2 = 16N, N_3 = 64N, \ldots$. For each mesh size N_k, a sequence of adapted meshes of size N_k is built by iterating the following loop:
(1) computing a solution,
(2) computing the optimal metric,
(3) building the adapted mesh.

In (1), a multi-grid V-cycle is applied to a sufficient convergence. In (2), approximations of the Hessians are performed as in [57]. When changing of mesh, an interpolation is applied in order to enjoy a good initial condition. A prescribed number of 4 adaptation iterations is applied at each mesh fineness N_k.

The extension of the above loop to norm-oriented adaptation consists of replacing the single Hessian evaluation by:
- the computation of the corrector, using MG and, as initial solution, the previous evaluation interpolated to current mesh and corrected according to (6.28),
- the computation of the adjoint, using MG and, as initial solution, the previous evaluation interpolated to current mesh and corrected according to (6.28),
- the evaluation of (6.25).

Let us discuss computer efficiency. In the demonstrator of [21], a particular feature is the stopping criterion of FMG which applies to the convergence of the solution of the unique solved system, *i.e.* the system under study, $u = A^{-1}f$. It is then possible to enjoy a better and better initial condition and control the iterative and approximation errors convergence. Consequently, it was possible, in [21], to show that mesh adaptation carries large improvement not only in terms of accuracy for a given number of vertices but also in terms of accuracy for a given computational time.

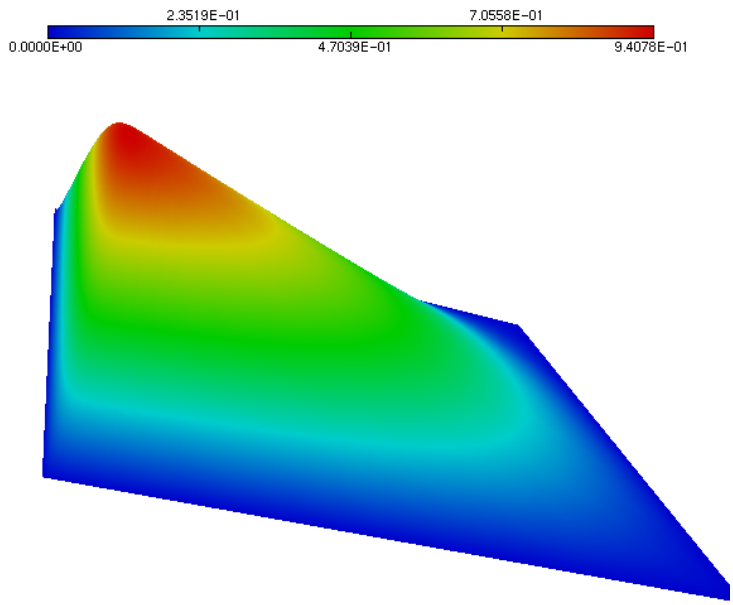

Figure 6.1: Fully 2D Boundary layer test case : sketch of the solution.

In contrast, the method proposed in this paper involves three systems to solve: (1) the system under study, $u = A^{-1}f$, (2) the corrector system, (3) the adjoint system. We shall give an idea of the performances for one test case.

6.8.2 2D Boundary layer

This test case is taken from [38]. We solve the Poisson problem $-\Delta u = f$ in $]0,1[\times]0,1[$ with Dirichlet boundary conditions and a right-hand side f chosen for having:

$$u(x,y) = [1 - e^{-\alpha x} - (1 - e^{-\alpha})x]4y(1-y).$$

The coefficient α is chosen equal to 100. The graph of the solution is depicted in Figure 6.1. Before applying our mesh adaptative algorithm, it is interesting to evaluate the accuracy of our correctors. We choose a 161×161 uniform mesh and show, in Figure 6.2 and Figure 6.3, the cut of $u - u_h$ compared with the cut of u'. We observe that the *a priori* corrector does its job in a correct but inaccurate manner while the DC one is rather accurate. We have also observed that the DC corrector does not consume notably more CPU than the *a priori* one. Therefore, we keep this option for the rest of the test case. In Figure 6.4, we show a set of FMG calculations for the considered test case. The numbers of vertices of the successive meshes are supported by the horizontal axis, from 120 vertices to 30,000

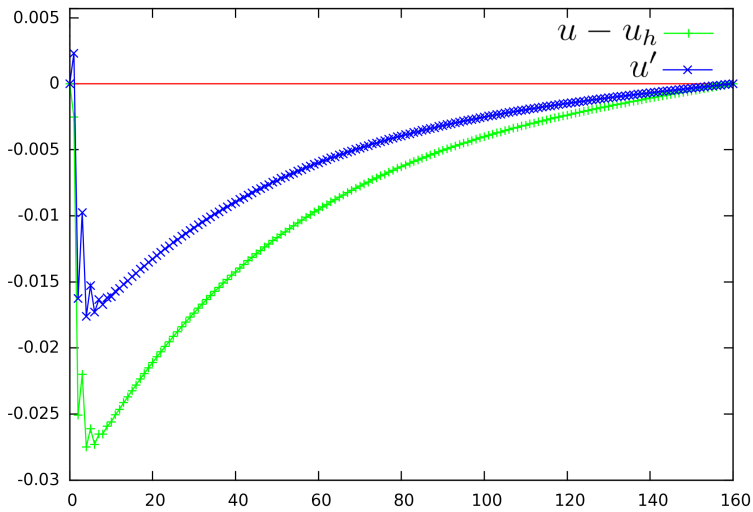

Figure 6.2: Fully 2D Boundary layer test case : comparison of error cuts for $y = 0.5$: plus signs (+) depict the approximation error $u - u_h$ and crosses ($\times$) depict the *a priori* corrector u'_{prio}. The corrector is able to correct about 60% of the approximation error.

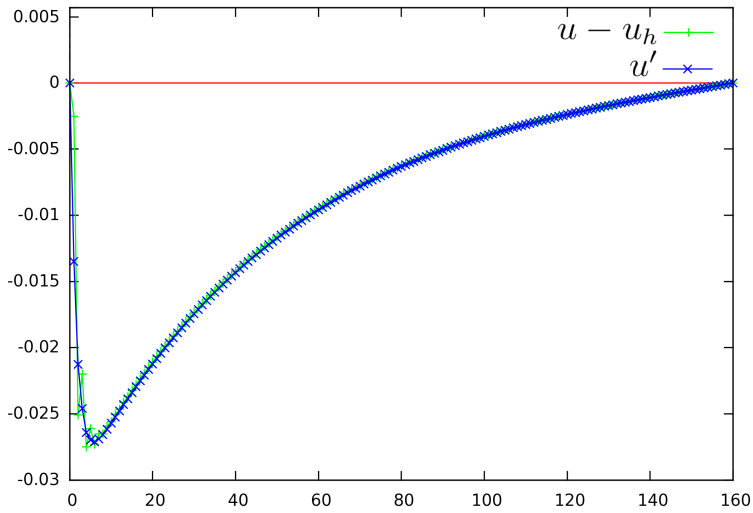

Figure 6.3: Fully 2D Boundary layer test case : comparison of error cuts for $y = 0.5$: plus signs (+) depict the approximation error $u - u_h$ and crosses ($\times$) depict the Defect-Correction corrector u'_{DC}. The corrector is able to correct about 95% of the approximation error.

vertices. The vertical axis gives the L^2-norm of the approximation error $|u - u_h|_{L^2}$ obtained on the mesh. Its variation with respect to number of vertices is compared in Figure 6.4 for the three following algorithms: (a) the uniform-mesh FMG, (b)

the Hessian-based adaptative FMG and (c) the norm-oriented adaptative FMG. We observe that both adaptation methods carry an important improvement with respect to uniform-grid FMG (25921 vertices on finest mesh). For essentially the same number of vertices (32318), the Hessian option gives an error divided by 47. The norm-oriented option appears as better with an error divided by 208 with 29485 vertices.

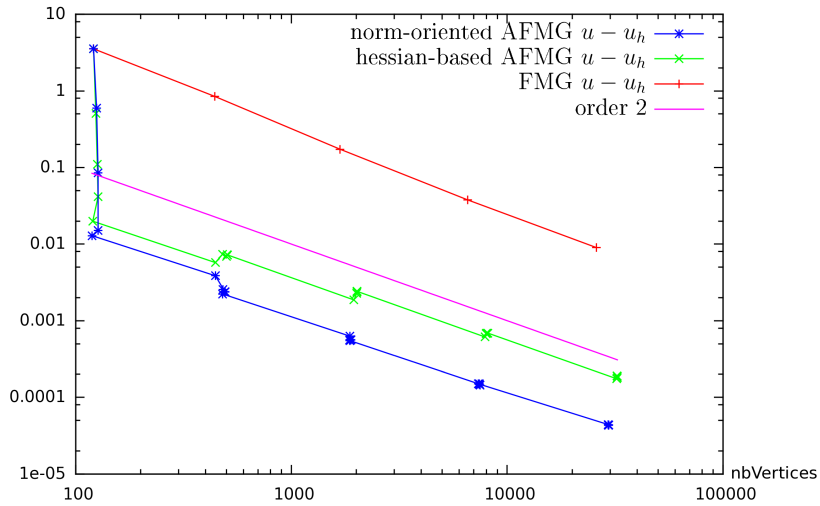

Figure 6.4: Fully 2D Boundary layer test case: convergence of the error norm $|u - u_h|_{L^2}$ as a function of number of vertices in the mesh for (+) non-adaptative FMG, (×) Hessian-based adaptative FMG, (∗) norm-oriented adaptative FMG.

We also propose an *a posteriori* measure of the correctors efficiency by comparing the convergence of our norm-oriented adaptation equipped with either corrector with the same method in which the corrector is replaced by $u - u_h$. Of course, that latter algorithm is not a mesh adaptation method since we assume that we already know the exact solution. In Figure 6.5, we observe that the error convergence for the three computations are very close to each other. This confirms the interest of the two proposed correctors.

Lastly, when using the proposed method in usual conditions, *i.e.* without knowing the analytic solution, it can be interesting to have an evaluation of the final error $u - u_h$ by using the second corrector which has been introduced and which appeared in experiments as more accurate. Figure 6.6 shows a comparison of the virtual error computed from the corrector with the actual error. We observe that the prediction is useful, giving the good order of magnitude. For the finer mesh of $30,000$ vertices, the predicted error is $2.14\ 10^{-5}$ while the actual error is $4.31\ 10^{-5}$. However, for most meshes, the prediction remains somewhat optimistic (smaller than the actual error).

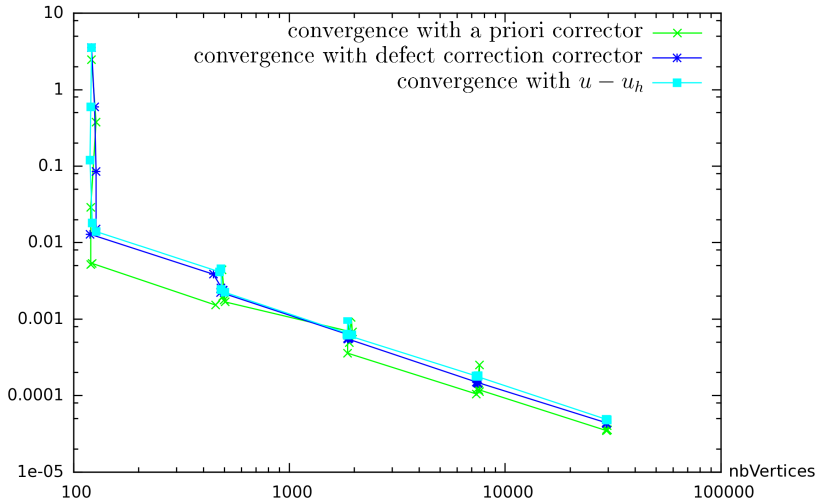

Figure 6.5: Fully 2D Boundary layer test case: convergence based on ($\times$) the *a priori* corrector or on the ($*$) Defect-Correction one, compared with ($\square$) a virtual adaptation controlled by $u - u_h$.

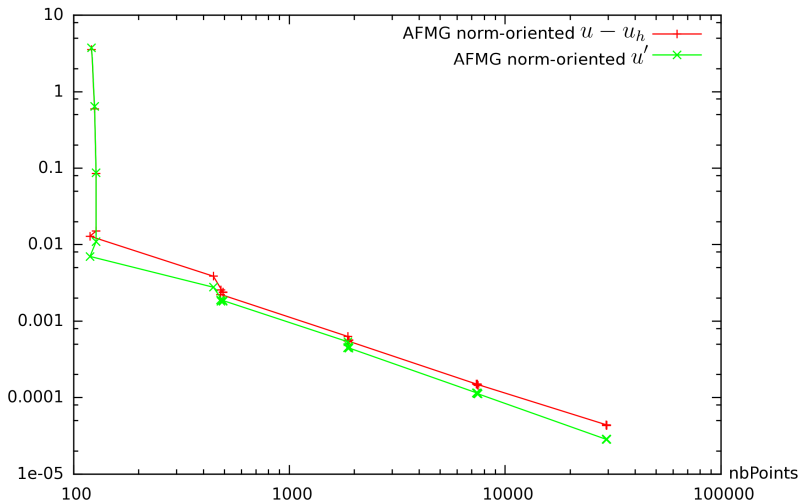

Figure 6.6: Fully 2D Boundary layer test case: comparison of the predicted error norm proposed by the method, curve with ($\times$), with the exact error norm ($+$).

6.8.3 Bubble-like test case with thick interface

We are interested by a Poisson problem the solution of which is a function u equal to 1 on a disk and to 0 in the rest of the domain. This function is the prototype of the pressure in a multi-fluid flow involving capillary forces. The source term is

a Dirac derivative. We smooth this computation by defining a thickness ε of an annular region separing the two subdomains (outside the disk, inside the disk) and in which u is smoothly varying from 0 to 1 as shown in Figure 6.7. If (x, y) is located

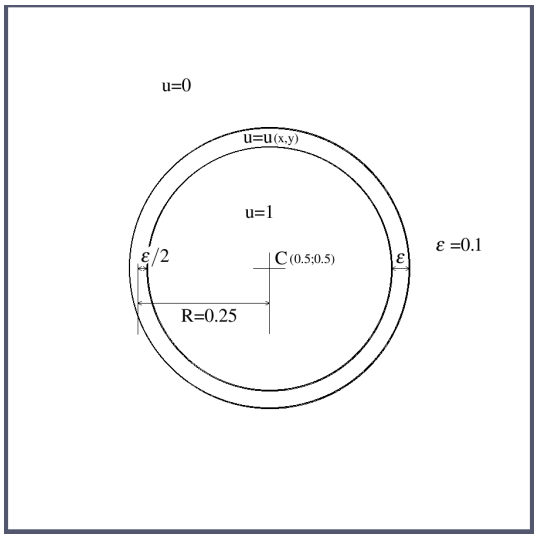

Figure 6.7: Circular-test-case-domain: sketch of the solution u.

inside the annular region, $u(x, y)$ is given by the formula: $u(x, y) = \frac{1}{2} + \frac{1}{2}sin(\frac{\pi\psi}{\varepsilon})$ with $\psi = 0.25 - \sqrt{(x_C - x)^2 + (y_C - y)^2}$. From this solution, a right-hand side f is computed. Given a mesh, vertex values of f_h are interlopated from the analytic f. As a result, for rather coarse meshes, the zone where f is not zero can be simply missed and f_h can be zero even in the neighborhood of the high values of f. We consider first a quite large thickness of $\varepsilon = 0.1$. An approximate solution u_h is shown in Figure 6.8. As for the previous test case, we first evaluate the accuracy of the

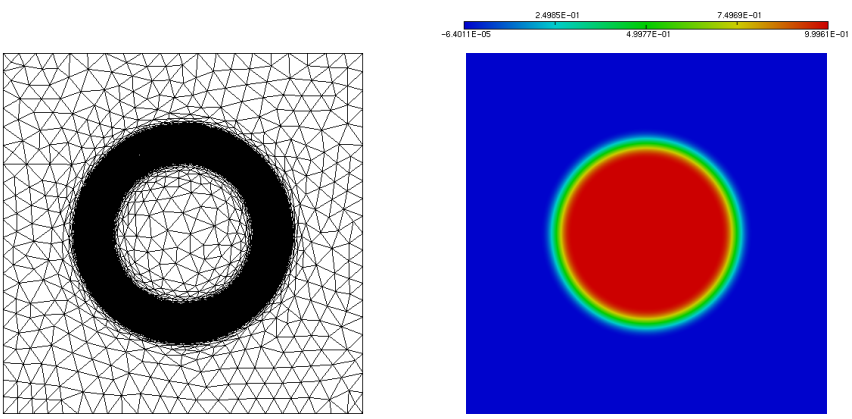

Figure 6.8: Circular layer test case: an adapted mesh and the corresponding numerical solution u_h.

corrector. We choose a uniform mesh 161×161 and show, in Figure 6.9 and Figure 6.10, the cut of $u - u_h$ compared with the cut after correction, that is $u - u_h - u'_h$. We observe that both *a priori* and Defect-Correction correctors do an accurate job.

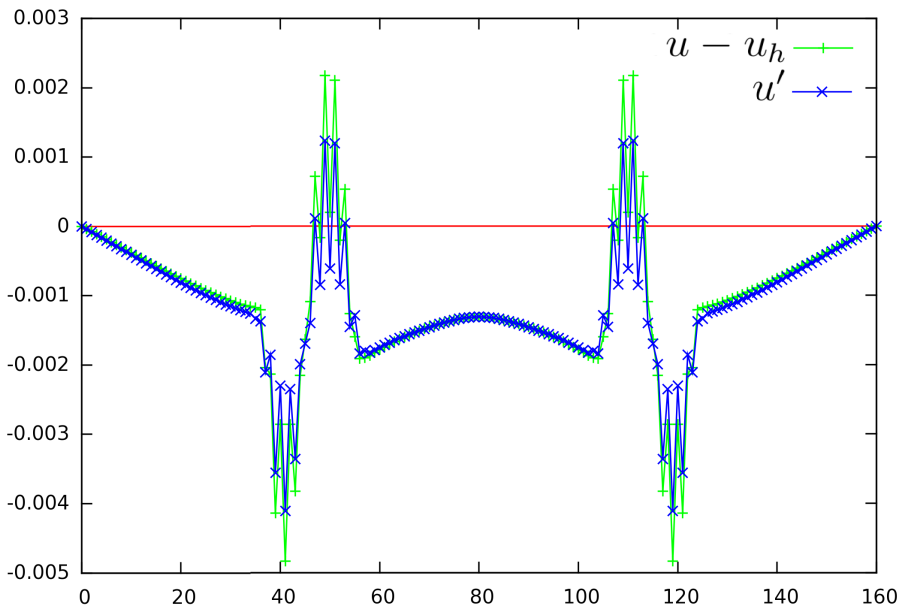

Figure 6.9: Thick bubble case: comparison of (+) error $u - u_h$ and (×) *a priori* corrector.

The three methods are again compared in Figure 6.11: standard FMG, Hessian-based adaptative FMG and norm-oriented adaptative FMG. For the Hessian-based calculation, we observe a tendancy for a slower convergence for finer meshes, finishing with an error which is worse than the uniform refinement. The proposed norm-oriented adaptative method behaves in a better way with a five times smaller error than for the uniform refinement.

6.8.4 Bubble-like test case with thin interface

In order to evaluate the robustness of the methods with respect to steeper gradients, we consider the same test case with a thinner transition: $\varepsilon = 0.02$.

The convergence of the three methods is shown in Figure 6.15. Due to the very thin support of the right-hand side f compared with the initial coarse mesh size, the three methods start with a zero f_h. Then, either with adaptation or uniform refinement, the error increases to several units. The convergence of the uniform FMG shows an acceptable slope but the error values are relatively huge. By comparison with the thick-bubble convergence, we may infer that the slope of uniform FMG will be second-order with even higher number of vertices and a 0.1 % error may be not attained for meshes of less than 10 millions nodes. The Hessian-based final

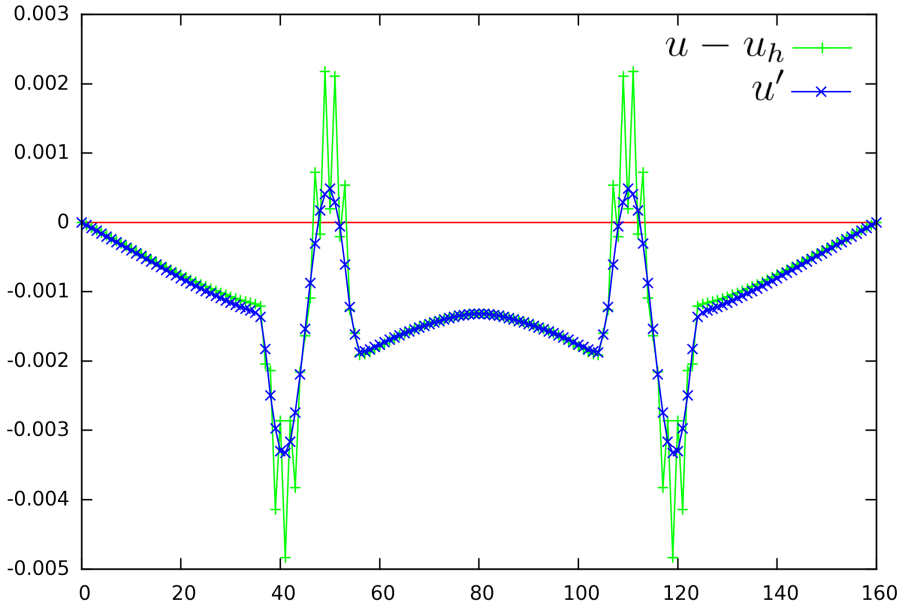

Figure 6.10: Thick bubble case: comparison of (+) error $u - u_h$ and (×) Defect-Correction corrector.

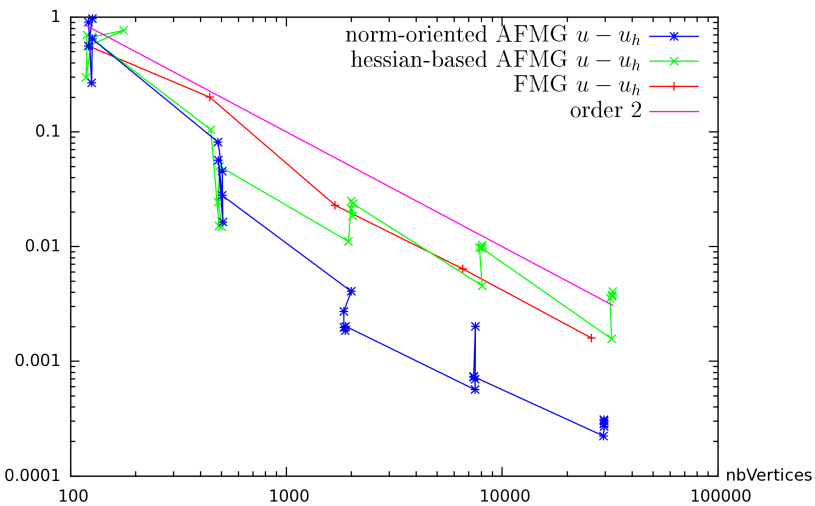

Figure 6.11: Thick bubble test case: convergence of the error norm $|u - u_h|_{L^2}$ as a function of number of vertices in the mesh for (+) non-adaptative FMG, (×) Hessian-based adaptative FMG and (∗) norm-oriented adaptative FMG.

result is a little better but globally disappointing. The norm-oriented convergence starts chaoticly before being monotone and second-order for meshes finer than 2000

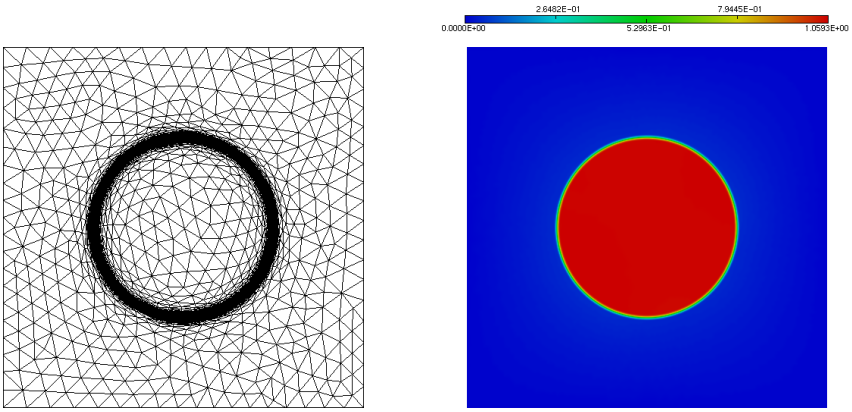

Figure 6.12: Thin bubble case: an adapted mesh and the corresponding numerical solution u_h.

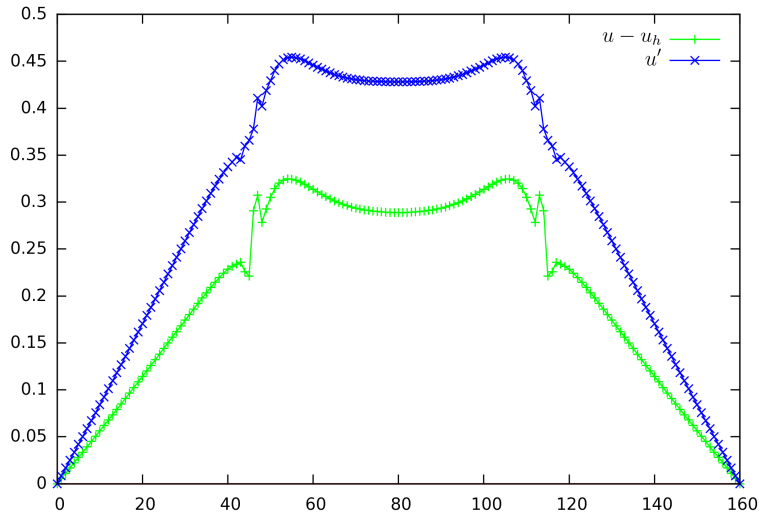

Figure 6.13: Thin bubble case: comparison of (+) error $u - u_h$ and (×) *a priori* corrector.

vertices. The final L^2 errors produced by the three methods are 0.17589 with 25921 vertices for the uniform FMG, 0.03773 with 32127 vertices for the Hessian-based adaptation and 0.000585 for 29742 vertices for the norm-oriented calculation, 300 times smaller than the first result. In order to show the behavior of the algorithms for more complex solutions, we have computed the case of a solution involving three bubbles of different sizes. A comparison, in Figure 6.16, of the solutions with and without adaptation demonstrates that the uniform FMG (results of top range) is unable to produce even a rough approximation with 6561 vertices. Only the solution for 25921 vertices is rather good, with an error L^1 norm of 0.50 but with

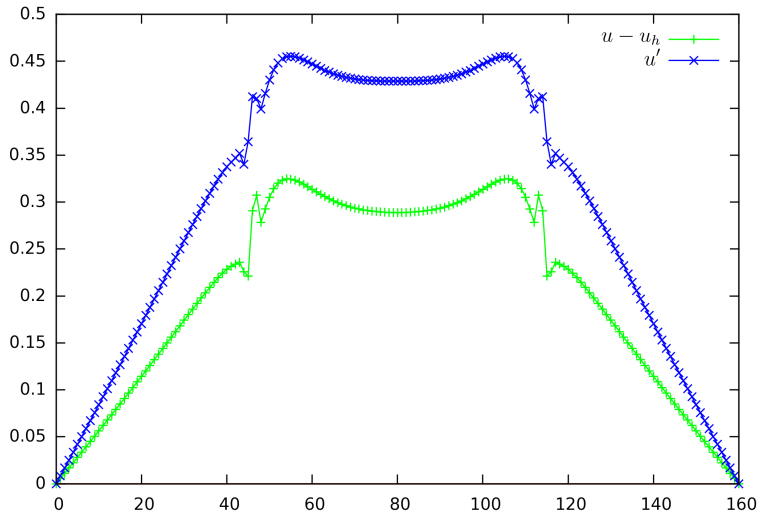

Figure 6.14: Thin bubble case: comparison of (+) error $u - u_h$ and (×) Defect-Correction corrector.

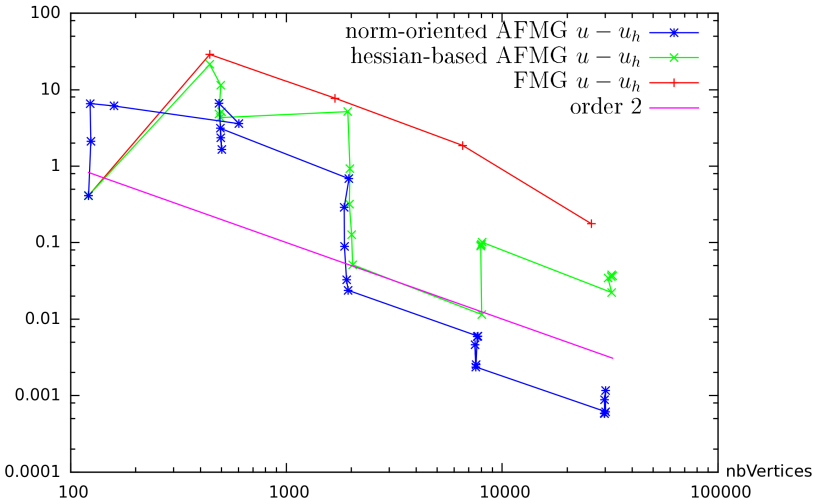

Figure 6.15: Thin bubble test case: convergence of the error norm $|u - u_h|_{L^2}$ as a function of number of vertices in the mesh for (+) non-adaptative FMG, (×) Hessian-based adaptative FMG and (∗) norm-oriented adaptative FMG.

mean values in bubbles showing still 10% deviations. In contrast, the norm-oriented mesh-adapted solutions (bottom range of Figure 6.16) show a fast convergence. For 1681 vertices, the solution has the same quality as the uniform 25921-vertices one (error is slightly larger, 0.54). For 6561 and 25921 vertices, the accuracy seems good. The apparent convergence order is greater than two. In contrast, we observe

that the Hessian-based calculation is rather disappointing.

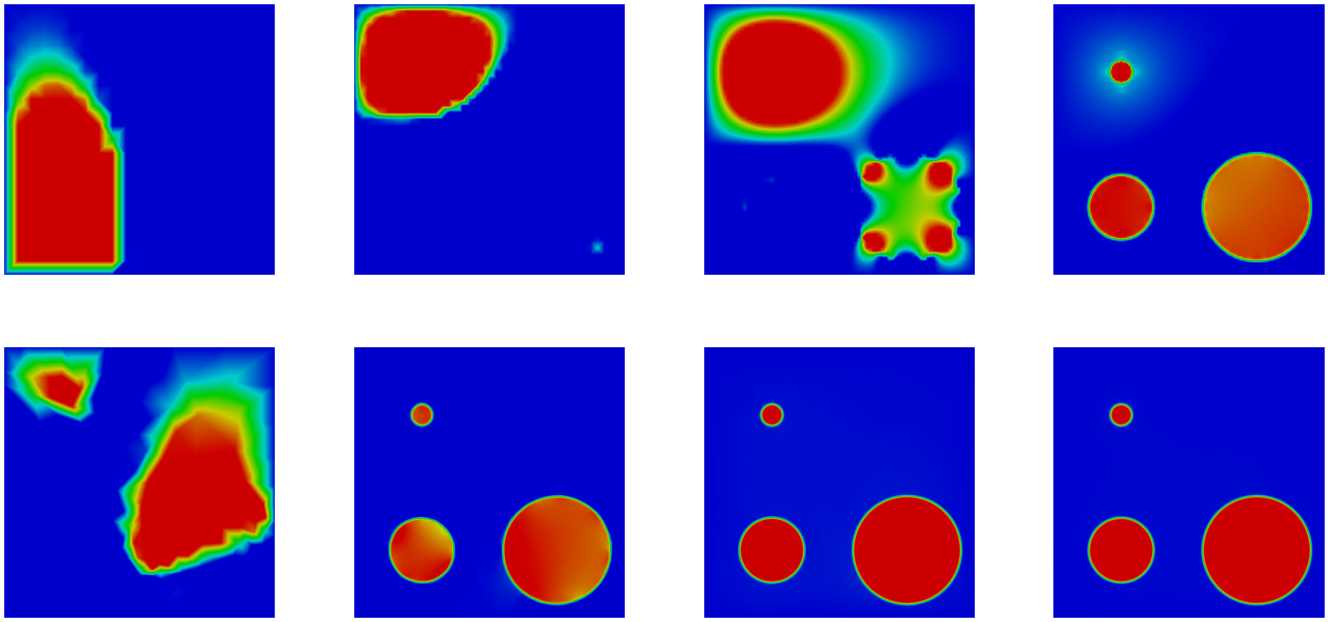

Figure 6.16: Multiple bubble test case: top: non-adaptative solutions for 441, 1681,6561, 25921 nodes, and, bottom, the mesh adaptative solution for about the same number of nodes (palette restricted to values between the minimum (0.) and the maximum (1.) of the exact solution.

6.8.5 Poisson problem with discontinuous coefficient

This test case exemplifies the singularity which is met in the simulation of multi-fluid flows with a large deviation between the densities ρ_1 and ρ_2 of each phase. In the case where a projection algorithm is applied, a Poisson problem with discontinuous coefficients has to be solved. An example can be found in [48]. We consider the equation of Poisson $-div(\frac{1}{\rho}\nabla u) = rhs$ with a discontinuous coefficient taking two different values $1/\rho_1$ and $1/\rho_2$ on two sub-domains Ω_1 and Ω_2 separated by an interface which is a sufficiently smooth curve for having a normal vector. This PDE is mathematically referred as a transmission problem and the solution is continuous across the interface but of discontinuous normal derivatives since:

$$1/\rho_1 \nabla u_1 \cdot \mathbf{n} = 1/\rho_2 \nabla u_2 \cdot \mathbf{n}$$

where u_1 and u_2 are the restrictions of the solution u on Ω_1 and Ω_2. In our example, we define them as follows

$$u|_{\Omega_i} = u_i = \alpha_i + \beta_i(x^2 + y^2) \qquad i = 1, 2.$$

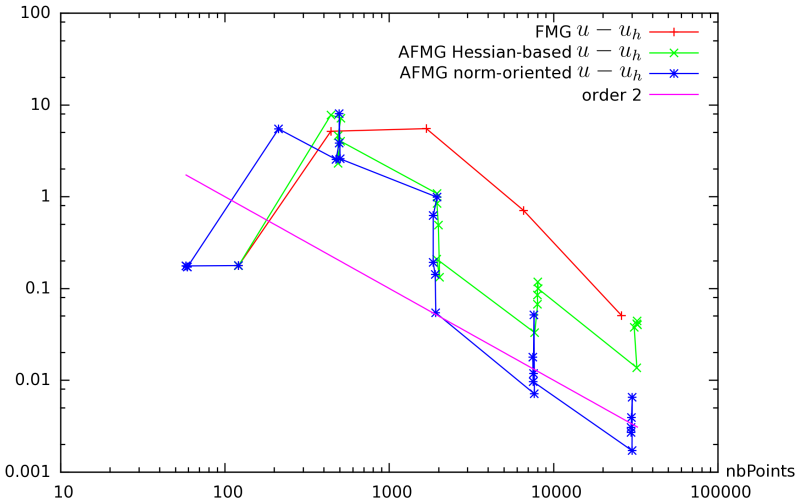

Figure 6.17: Multiple bubble test case: convergence of the error norm $|u - u_h|_{L^2}$ as a function of number of vertices in the mesh for (+) non-adaptative FMG, (×) Hessian-based adaptative FMG and (∗) norm-oriented adaptative FMG.

Further, Ω_2 is the disk of center $(0.5, 0.5)$ and of radius 0.2 in the computational domain $]0,1[\times]0,1[$ and we have:

$$\begin{aligned} 1/\rho_1 &= 1000. \;;\; \alpha_1 = 1.23579... \;;\; \beta_1 = -2.47158... \\ 1/\rho_2 &= 1. \;;\; \alpha_2 = 100. \;;\; \beta_2 = -2471.58... \end{aligned} \tag{6.29}$$

This is sketched in Figure 6.18. In the discrete model, the interface appears only as values of $1/\rho$ evaluated on the vertices of each grid. From the examination of cuts of the correctors in Figures 6.19 and 6.20, we observe that their accuracy is somewhat a disaster. However, the convergence of our process is not really affected as can be seen from the comparison of convergence with the correctors or with the exact error, Figure 6.21. A convergence in terms of number of vertices with non-adaptative and Hessian-based adaptive is given in Figure 6.22. We observe that, without mesh adaptation (crosses +), the convergence order is around 1. This behavior can be explained by the singularity of the solution. In contrast, the overall convergence order of the adaptative process is about two. This is a usual bonus obtained by anisotropic mesh adaptation which has been noted in [59]. A short analysis of this property is proposed in [32].

Although not optimised for CPU, the novel algorithm may give a good increase of efficiency. We present, in Figure 6.23, the behavior of the error as a function of CPU. Three curves are compared. The upper curve corresponds to the non-adaptative FMG calculation. An error of about 0.06 is obtained on 100000 vertices after 25000 seconds of CPU. We present also the Hessian-based adaptative convergence obtained in [21] in which the convergence of MG is adequately controlled by a stopping

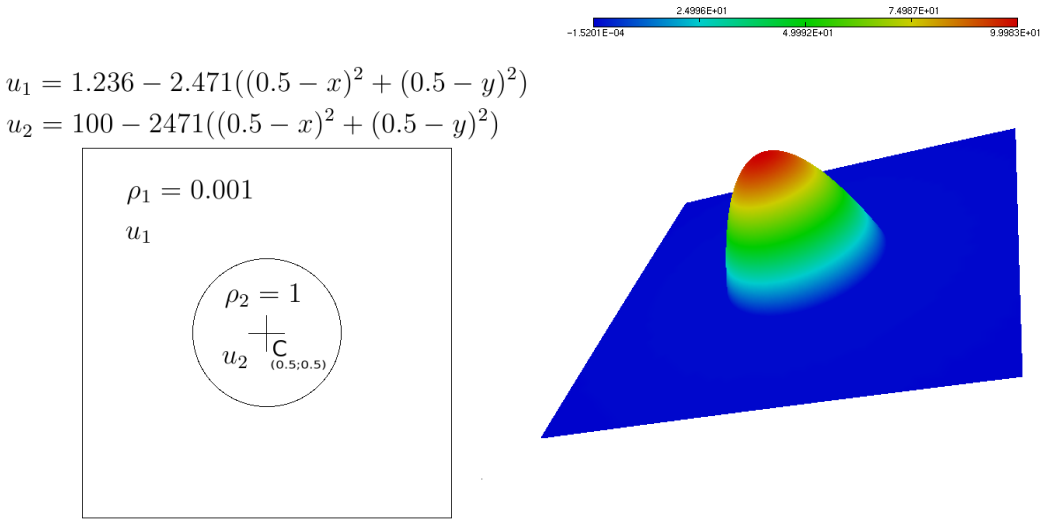

Figure 6.18: Poisson problem with discontinuous coefficient: sketch of exact solution definition and a typical computation of it.

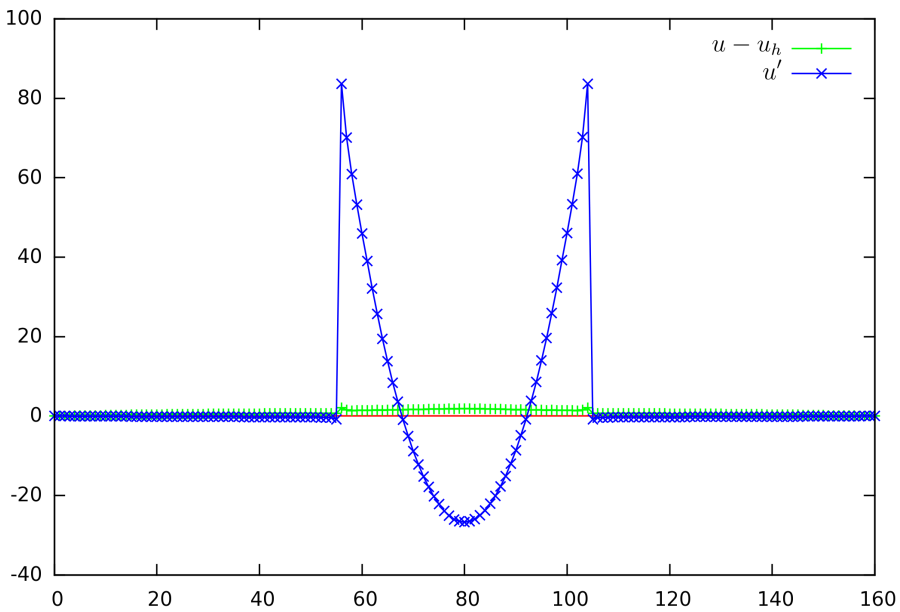

Figure 6.19: Poisson problem with discontinuous coefficient: comparison of (+) error $u - u_h$ and (×) *a priori* corrector.

criterion. Despite the remeshing and re-computing for adaptation, the total time is only about twice the non-adaptative FMG time, for an error of $0.001-0.002$, at least 30 times smaller. The third curve shows the CPU behavior of the norm-oriented algorithm. The total time is about ten times the non-adaptative FMG time, for an error of 0.001. After 25000 seconds of CPU, the error is about 10 times smaller than

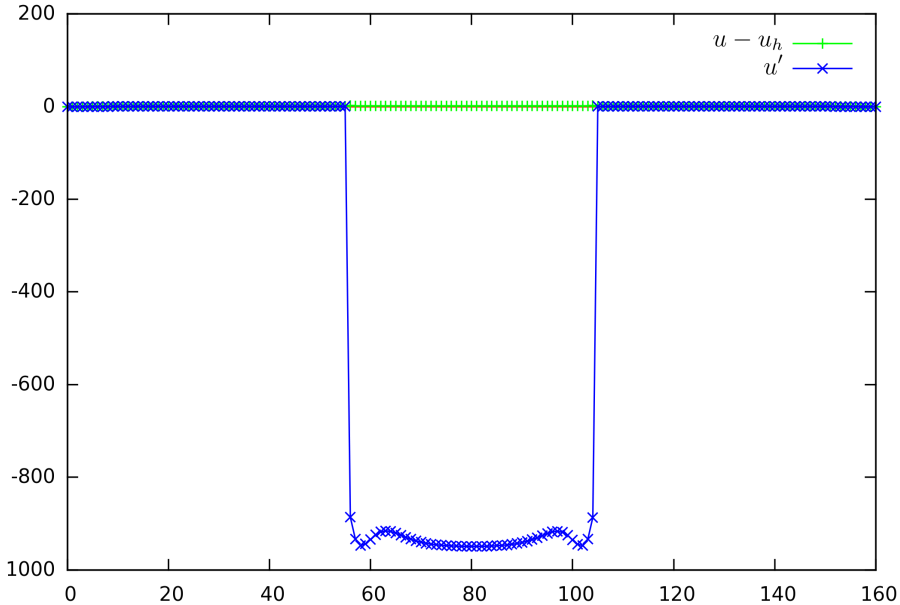

Figure 6.20: Poisson problem with discontinuous coefficient: comparison of (+) error $u - u_h$ and (×) Defect-Correction corrector.

with the non-adaptative FMG at same time.

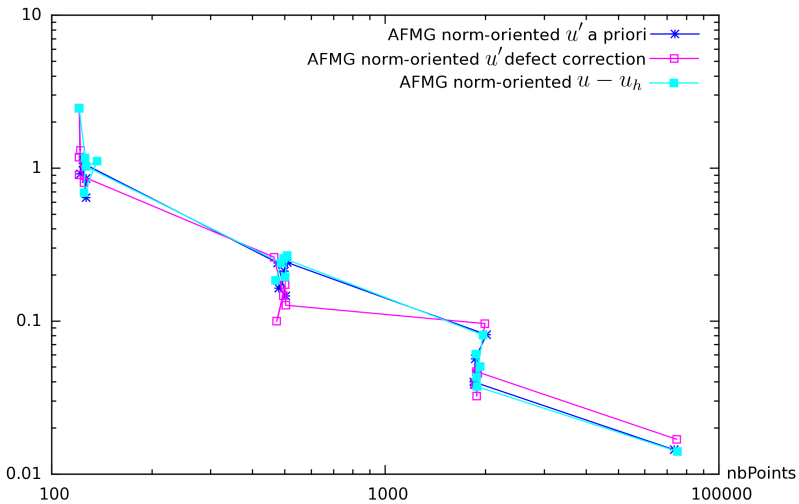

Figure 6.21: Poisson problem with discontinuous coefficient: convergence based on (×) the *a priori* corrector or on the (∗) Defect-Correction one, compared with (□) a virtual adaptation controlled by $u - u_h$.

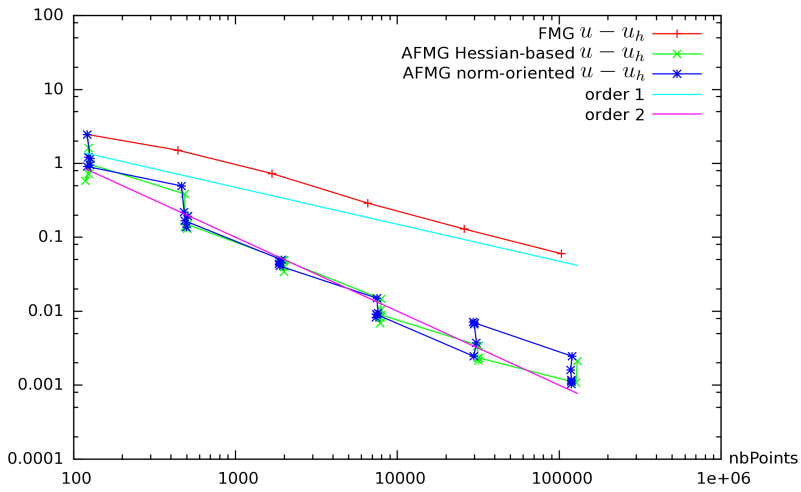

Figure 6.22: Poisson problem with discontinuous coefficient: convergence of the error norm $|u - u_h|_{L^2}$ as a function of number of vertices in the mesh for (+) non-adaptative FMG, (×) Hessian-based adaptative FMG and (∗) norm-oriented adaptative FMG.

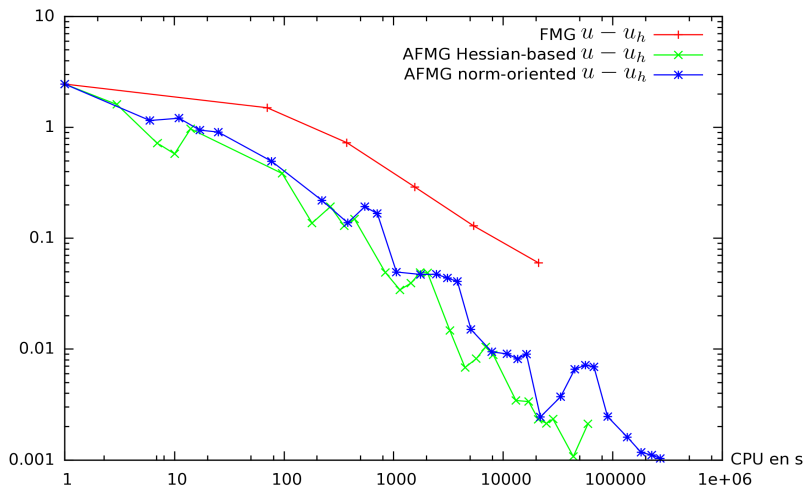

Figure 6.23: Poisson problem with discontinuous coefficient: efficiency analysis : CPU time for (+) non-adaptative FMG, (×) Hessian-based adaptative FMG and (∗) norm-oriented adaptative FMG.

6.8.6 A 1D Boundary layer

The new method has shown a good behavior for all test cases we tried except one

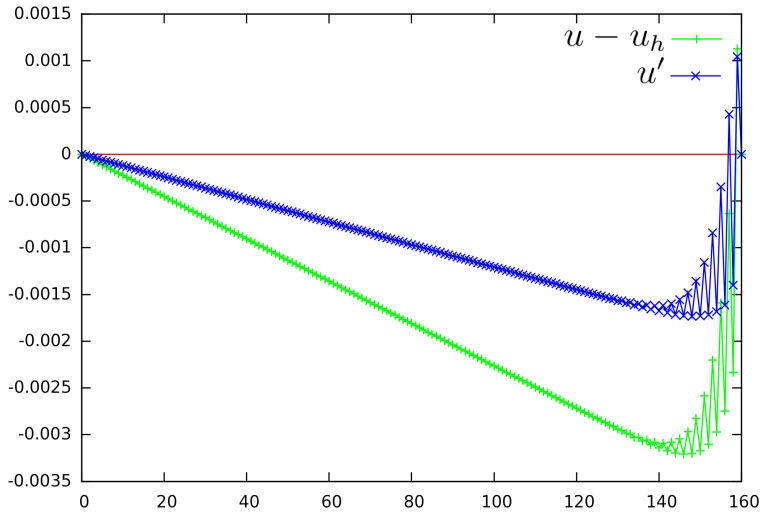

Figure 6.24: 1D Boundary Layer: comparison of error cuts for $y = 0.5$: plus signs (+) depict the approximation error $u - u_h$ and crosses ($\times$) depict the *a priori* corrector u'_{prio}. The corrector is able to correct about 55% of the approximation error.

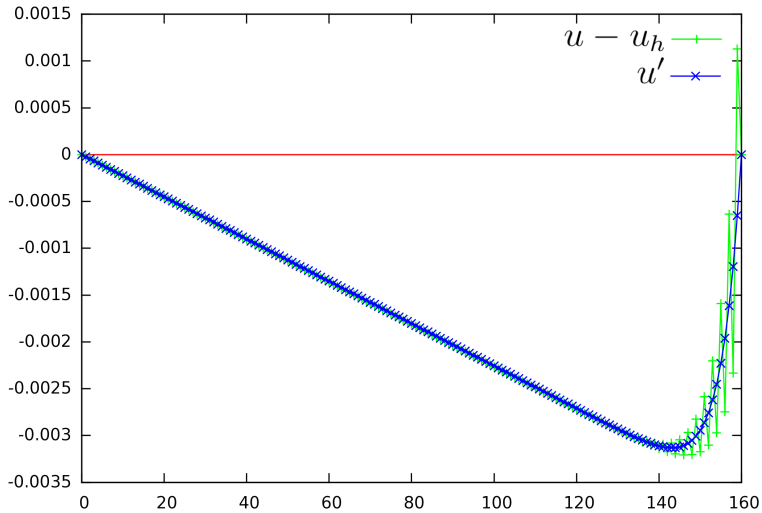

Figure 6.25: 1D Boundary Layer: comparison of error cuts for $y = 0.5$: plus signs (+) depict the approximation error $u - u_h$ and crosses ($\times$) depict the Defect-Correction corrector u'_{DC}. The corrector is able to correct about 80% of the approximation error.

which we describe now. It is a boundary layer case with a 1D solution: $u(x, y) = u(x)$ of a Dirichlet-Neumann problem $-\Delta u = rhs$ with

$$rhs(x, y) = (\alpha^2(exp(1/\alpha) - 1))^{-1} exp(x/\alpha) \quad ; \quad \alpha = 0.03.$$

We check first the correctors. Both seem adequate on a uniform grid, as shown with a horizontal cut depicted on Figure 6.24 and Figure 6.25.

In Figure 6.26, the non-adaptative FMG produces an approximation error of 0.003 (30000 vertices). This convergence is relatively satisfactory, being a second order convergence. However, in order to reach a 10^{-7} error level, several hundred millions vertices will be necessary with this sequence of uniform meshes. A second curve is obtained with the adaptative FMG with the Hessian-based criterion. Final convergence is disappointing since the slope is first-order. The same problem appears with our new algorithm. A deeper examination of adaptation criteria has shown that the high derivatives of the right-hand side f are very close to boundary $x = 1$. It could not be seen by the algorithm, because of the weighting by the adjoint u^*, which is zero at this boundary. We have replaced the norm-oriented optimum metric by its intersection with the metric based on the Hessian of f. Then, the convergence improved a lot. In contrast, introducing the same metric intersection in the other cases did not produce second order convergence.

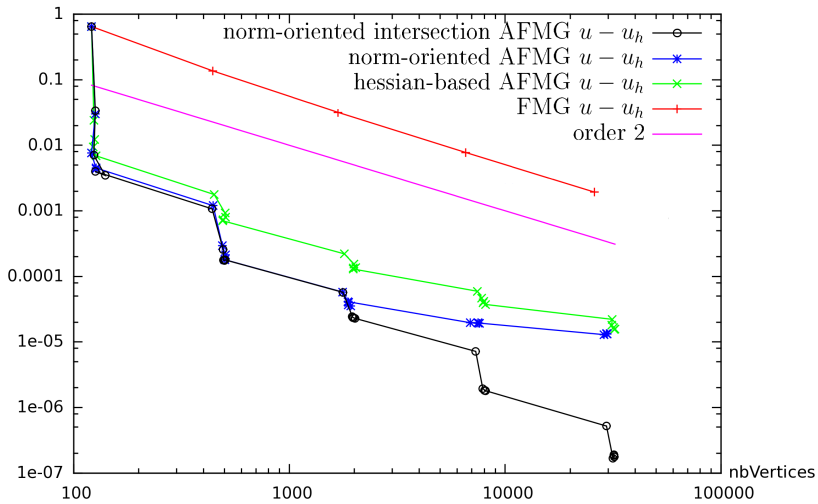

Figure 6.26: 1D Boundary Layer: convergence of the error norm $|u - u_h|_{L^2}$ as a function of number of vertices in the mesh for (+) non-adaptative FMG, (×) Hessian-based adaptative FMG, (∗) norm-oriented adaptative FMG, and (o) norm-oriented adaptative FMG with intersection with RHS Hessian.

6.9 Conclusion

The norm-oriented mesh adaptation method is an answer to a well formulated problem. Considering a numerical scheme, here the most used FEM, and prescribing a norm, we want to find the mesh giving the smallest approximation error in that norm for a given number of vertices. The norm-oriented mesh adaptation method

transforms the problem into an optimization problem which is mathematically well-posed. It relies on the following other features.

A corrector represents the approximation error. We give two examples of correctors. An *a priori* corrector is built from the variational discrete statement. A Defect-Correction corrector is built from a finer-mesh defect correction principle. These correctors appear as not very accurate but sufficiently accurate for our purpose. According to the type of approximation, at least the second one, Defect-Correction is extendable to many models and schemes.

The norm-oriented algorithm is presented as a natural extension of the goal-oriented algorithm which, in our formulation, is itself a natural extension of the Hessian-based algorithm. More precisely, while the Hessian-based algorithm solves only the PDE under study in the mesh-adaptation loop, the goal-oriented algorithm also solves an adjoint system (with linearised operator, transposed). The norm-oriented algorithm solves three systems, a corrector (linearised system with an adhoc RHS), an adjoint (linearised and transposed with the corrector as RHS) and the PDE itself. The three algorithms have in common an anisotropic *a priori* error analysis and a metric-based mesh parameterisation.

The Hessian-based method produces convergent solution fields but does not take into account the precise equation and discretization. The goal-oriented method takes into account equation and discretization but is too focused on a particular output and does not generally produce convergent solution fields. The norm-oriented method has the advantages of both.

In order to show the improvement obtained with respect to previous methods, we compare, in our experiments, the two field-convergent options, Hessian-based and norm-oriented. Our benchmark examines convergence to continuous thanks to the application of a Full-Multigrid (FMG) process. Approximation errors can then be compared as functions of the number of degrees of freedom. Although of larger complexity, the mesh adaptative algorithms are generally more efficient in terms of CPU time than the non-adaptative one. The advantage is renforced by the fact that the same level of error is reached with a much lower number of vertices. Secondly, although the elliptic context is known as favourable to Hessian-based methods, the norm-oriented approach behaves notably better than Hessian-based in terms of approximation error for a given number of vertices. This is observable in particular for singular or stiff contexts.

We have presented an example of comparison of computational effort but our algorithm is not optimised. It is, therefore, in several cases, less efficient than Hessian-based adaptative FMG.

The method is rather general and it is now applied to more complex PDE models from CFD (Euler, Navier-Stokes), see [58], involving dominant advection effects for which a Hessian-based approach is much less efficient.

Chapter 7

FMGA orienté-norme tensoriel

7.1 Introduction

This paper addresses anisotropic mesh adaptation for PDE's (Partial Differential Equations). We focus on methods which prescribe a somewhat optimal mesh under the form of a parametrization of it by a Riemannian metric. These Riemannian metrics can be considered under a continuous standpoint, as proposed in [54, 55]. A dual way is proposed in [30, 31] and relies on mesh-based tensorial statistics.

Continuous and *tensorial* metrics both rely on the parametrization of the mesh by a spatial field defining in any point of the computational domain a matrix giving information on mesh size in all the spatial directions.

Both methods solve an optimality system, the continuous metric builds a continuous optimality system which has, afterwards, to be discretised and solved, while the tensorial metric builds a discrete optimality system to be solved directly. Also, the continuous metric theory defines the ideal metric to be choosen. The resulting ideal mesh produced by metric optimization is a unit mesh for the ideal/optimal metric. In contrast, the tensorial metric obtained from optimization in [30, 31] defines the modification to apply to the current mesh in order to obtain the ideal mesh. Then the way to parameterize the final mesh with the two metrics is different, since the ideal mesh is of length unity (any edge has a length 1 for the metric) for the continuous metric, while the tensorial metric defines the ideal mesh from local directional amplifications of the background mesh. Similarly, the constraint imposing a prescribed number of nodes is formulated vertices by vertices for the continuous metric and edges by edges for the tensorial method.

Both methods apply to a typical family of optimal metric-based method for PDE's, the P_1-Interpolation-based/Hessian-based method. Continuous and tensorial Hessian-based methods involve the equi-distribution method, which turns out to finding the metric which minimizes a L^∞ norm of the interpolation error of one or several sensors depending on the PDE solution. Also involved in the continuous Hessian-based methods is the multiscale method, which minimizes the L^p interpolation error of the sensors. Cf. [25, 35, 2, 77, 46, 78, 51, 27, 3, 83]. These Hessian-based methods, while taking into account the features of the PDE solution, do not take into account the features of the PDE itself. However, if sensors are cleverly chosen, a good convergence of the whole approximate solution field to the exact solution field is usually observed.

Goal-oriented methods allow to take into account the PDE under study. A

combination with anisotropic Hessian-based adaption is proposed in [80]. Goal-oriented optimal methods [56, 14, 83], minimize with respect to the metric the approximation error committed on the evaluation of a scalar functional depending on the PDE solution. They do take into account the features of the PDE, typically through the use of an adjoint state. Goal-oriented methods needs also to rely on an error estimate (and on its sensitivity to mesh).

Several methods have been proposed for reducing the approximation error through an estimate. A pioneering approach is the work Becker and Rannacher [12] which rely, as many estimate-based work, on an *a posteriori* estimate. A good synthesis concerning *a posteriori* estimates is [81]. An interest of *a posteriori* estimate is that it is expressed in terms of the approximate solution, assumed to be available in a mesh adaption loop. A second interest is that it does not require the use of higher order (approximate) derivatives, in contrast to truncation analyses. These estimates show accurately where the mesh should be refined. However, deducing the best anisotropic mesh from an *a posteriori* estimate remains difficult (see however [83]).

A priori estimates rely quasi-systematically on Taylor series, either through divided differences, or through polynomial approximation of functions. Then approximations of higher order derivatives of solution need be built from the approximate solution. This is a delicate job since nothing ensures that a higher order derivative of the approximate solution is a good approximation of the corresponding higher order derivative of the exact solution. In contrast with the *a posteriori* option, the Taylor series can be easily used for proposing a somewhat optimal mesh. Further, *A priori* estimates can also provide correctors: an example is given in [23]. In [56, 14], in order to be able to solve the goal-oriented, metric-based, mesh optimization problem, the authors introduce an *a priori* analysis which restricts to the main asymptotic term of the local error. In this paper we use the tensorial formulation in order to build a novel *a priori* estimate for the Poisson equation which does not explicitly require the evaluation of higher-order derivatives.

With the metric-based goal-oriented formulation, metric-based mesh adaptation becomes a well-posed optimization problem for the reduction of a genuine approximation error. However, goal-oriented optimal methods are specialized to a given scalar output. Features of the solution field which are not related to this output may be neglected by the automatic mesh improvement. As a consequence, these methods do not provide a convergent solution field. In the present paper, we study a norm-oriented formulation (according to [23]) in which the user can prescribe a norm of error $|u - u_h|$ which the algorithm will minimize with respect to the metric parametrization of the mesh. As a consequence, with an adequate choice of the norm, the norm-oriented mesh adaptation produces convergent solution fields.

The continuous approach for Hessian-based, goal-oriented, and norm-oriented has been defined in papers like [56, 14, 23].

The purpose of this paper is to analyse the possible novelties which can be derived from the application of a tensorial method to Hessian-based, goal-oriented,

and norm-oriented problematics. The main feature of tensorial approach which we shall exploit is the tensorial inversion of main error term. In order to adapt this feature to L^1-Hessian, to goal-oriented, to norm-oriented problematics, we unify the parametrization by choosing the unit-mesh formulation and by measuring the number of nodes on a vertex basis.

In Section 2 we define the Poisson problem under study and propose a simple corrector for the discrete solution. Section 3 recall the main features of the continuous metric adaptation. Section 4 introduces the discrete context for tensorial metric optimization. In Section 5, we focalise on a particular family of errors, edge-based second-order errors. Section 6 gives the optimal metric for this family of errors. Numerical examples are presented in Section 7 and the paper is concluded by Section 8.

7.2 Poisson problem approximation

Let us introduce some notations: Let $V = H_0^1(\Omega)$, Ω being a smooth enough computational domain of $\mathbb{R}^2$. The continuous PDE system is written in short:

$$Au = f \quad or \quad u \in V, \quad \forall\, \phi \in V, \quad a(u, \phi) = (f, \phi). \tag{7.1}$$

To fix the ideas and simplify notations,

$$A = -\sum \frac{\partial}{\partial x_k} \frac{\partial}{\partial x_k}.$$

But the extension to a coercive case where $A = -\sum \frac{\partial}{\partial x_k}(a_{k\ell}(\mathbf{x})\frac{\partial}{\partial x_\ell})$ (where $a_{k\ell}$ is a scalar, possibly discontinuous, field) is not difficult. Let $\Omega_h = \Omega$ for simplicity, τ_h a triangulation of Ω_h, and V_h be the usual P_1-continuous finite-element approximation space related to τ_h:

$$V_h = \phi_h \in \mathcal{C}^0(\bar{\Omega}) \cap V, \phi_h|_T \text{ is affine } \forall T \in \tau_h.$$

The finite-element discretisation of (7.1) is written:

$$u_h \in V_h \quad \forall\, \phi_h \in V_h \quad a(u_h, \phi_h) = (f, \phi_h). \tag{7.2}$$

We are interested first in getting estimates of the approximation error $u_h - u$. Let N be the dimension of V_h, that is the number of vertices in τ_h. We observe that (7.2) is equivalent to computing the array $\mathbf{u}_h$ of the degrees of freedom of the discrete solution:

$$\mathbf{u}_h \in \mathbb{R}^N \ ; \ \mathbf{A}_h \mathbf{u}_h = \mathbf{f}_h. \tag{7.3}$$

From the above array we derive u_h by

$$u_h = \sum_{i=1,N} \mathbf{u}_{h,i} N_i(\mathbf{x})$$

where the N_i are the canonic finite-element basis of V_h:

$$N_i \in V_h, \quad N_i(\mathbf{x}_j) = 1 \text{ if } i = j,\ 0 \text{ else.}$$

We also introduce the interpolation operator Π_h:

$$for \quad v \in V \cap H^2(\Omega), \quad \Pi_h v \in V_h, \quad (\Pi_h v - v)(\mathbf{x}_i) = 0 \quad \forall \mathbf{x}_i \text{ vertex of } \tau_h.$$

An *a priori* error analysis can be applied in order to build a corrector, which is a signed approximate, and not a upper bound, of the approximation error. We start from the discrete above statement

$$a(u_h, \phi_h) = (f_h, \phi_h) \quad \forall \phi_h \in V_h.$$

and observe that for the exact solution satisfies:

$$a(u, \phi_h) = (f, \phi_h) \quad \forall \phi_h \in V_h.$$

Then

$$a(u_h, \phi_h) = a(u, \phi_h) + (f_h - f, \phi_h) \quad \forall \phi_h \in V_h.$$

Assuming that the solution u is sufficiently smooth, we get:

$$a(\Pi_h u - u_h, \phi_h) = a(\Pi_h u - u, \phi_h) + (f - f_h, \phi_h) \quad \forall \phi_h \in V_h. \tag{7.4}$$

We call $\Pi_h u - u_h$ the *implicit error*. It differs from the approximation error by an interpolation error:

$$u - u_h = u - \Pi_h u \quad + \quad \Pi_h u - u_h.$$

The rest of the section is devoted to finding an approximate of the implicit error. In practice, we need to evaluate the RHS of (7.4). The second term of (7.4)'s RHS is easy to evaluate (we know f and f_h). The first term of (7.4)'s RHS can be transformed as follows:

$$\begin{aligned} a(\Pi_h u - u, \phi_h) &= \sum_T \int_T \nabla \phi_h \nabla (\Pi_h u - u)\ \mathrm{d}x\mathrm{d}y \\ &= \sum_T \int_{\partial T} (\Pi_h u - u) \nabla \phi_h \cdot \mathbf{n}\ \mathrm{d}\sigma. \end{aligned}$$

Then we get:

$$\begin{aligned} a(\Pi_h u - u, \phi_h) &= K(\phi, u_h) \qquad \text{with} \\ K(\phi, u_h) &= \sum_{\partial T_{ij}} \nabla(\phi_h|_{T_i} - \phi_h|_{T_j}) \cdot \mathbf{n}_{ij} \int_{\partial T_{ij}} (\Pi_h u - u)\ \mathrm{d}\sigma \end{aligned} \tag{7.5}$$

where the last sum is taken for all edges $ij = \partial T_{ij}$ (2D case) separating triangles T_{ij}^+ and T_{ij}^- of the triangulation. The unit vector $\mathbf{n}_{ij}$ normal to ∂T_{ij} is pointing outward T_i.

In order to evaluate the interpolation error, we introduce the following interpolation estimator π:

$$for \quad v \in V \cap \mathcal{C}^2(\bar{\Omega}), \quad \pi v - v = H(v)\delta x \delta y$$

where $H(v)$ holds for the Hessian matrix of u and $\delta x \delta y$ hold for local mesh sizes in Cartesian directions. We could write $\Pi_h u - u \approx \pi u - u$, but we do not know u. Replacing u by u_h, the second derivatives of the unknown u are approximated by approximate second derivatives of the discrete solution as for example in [4]. This more or less assumes that the approximate second derivatives of the approximate solution are not a too bad approximation of the second derivatives of the unknown exact solution. This last statement is far from being satisfactorily stated. However, under this condition, our *corrector* is defined by:

$$\begin{aligned} a(u'_{prio}, \phi_h) &= K(\phi_h, u_h) \qquad \text{with} \\ K(\phi_h, u_h) &= \sum_{\partial T_{ij}} (\nabla \phi_h|_{T_i} - \nabla \phi_h|_{T_j}) \cdot \mathbf{n}_{ij} \int_{\partial T_{ij}} (\pi_h u_h - u_h) \, \mathrm{d}\sigma. \end{aligned} \tag{7.6}$$

In practice, the term $\pi_h u_h - u_h$ is built on the edge T_{ij} as a quadratic function vanishing at both extremities of T_{ij}, and of second derivative in direction T_{ij} equal to the approximate second derivative in same direction of u_h.

The above relation (7.6) is routinely used for building mesh adaption loops, see [15].

7.3 Continuous metric parametrization

7.3.1 Mesh parametrization

We recall the continuous mesh framework, introduced in [54, 55]. The main idea of this framework is to model discrete meshes by Riemannian metric fields. It allows us to define a differentiable optimization problem [1, 8], *i.e.,* to apply on the class continuous metrics a calculus of variations which cannot be applied on the class of discrete meshes. This framework lies in the class of metric-based methods. A continuous mesh $\mathcal{M}$ of the computational domain Ω is identified to a Riemannian metric field [16] $\mathcal{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$. For all $\mathbf{x}$ of Ω, $\mathcal{M}(\mathbf{x})$ is a symmetric 3×3 matrix having $(\lambda_i(\mathbf{x}))_{i=1,3}$ as eigenvalues along the principal directions $\mathcal{R}(\mathbf{x}) = (\mathbf{v}_i(\mathbf{x}))_{i=1,3}$. Sizes along these directions are denoted $(h_i(\mathbf{x}))_{i=1,3} = (\lambda_i^{-\frac{1}{2}}(\mathbf{x}))_{i=1,3}$ and the three *anisotropy quotients* r_i are defined by: $r_i = h_i^3 \, (h_1 h_2 h_3)^{-1}$. The diagonalisation of $\mathcal{M}(\mathbf{x})$ writes:

$$\mathcal{M}(\mathbf{x}) = d^{\frac{2}{3}}(\mathbf{x}) \, \mathcal{R}(\mathbf{x}) \begin{pmatrix} r_1^{-\frac{2}{3}}(\mathbf{x}) & & \\ & r_2^{-\frac{2}{3}}(\mathbf{x}) & \\ & & r_3^{-\frac{2}{3}}(\mathbf{x}) \end{pmatrix} {}^t\mathcal{R}(\mathbf{x}), \tag{7.7}$$

The *vertex density* d is equal to: $d = (h_1 h_2 h_3)^{-1} = (\lambda_1\lambda_2\lambda_3)^{\frac{1}{2}} = \sqrt{\det(\mathcal{M})}$. By integrating it, we define the *total number of vertices* $\mathcal{C}$:

$$\mathcal{C}(\mathcal{M}) = \int_\Omega d(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_\Omega \sqrt{\det(\mathcal{M}(\mathbf{x}))}\,\mathrm{d}\mathbf{x}. \tag{7.8}$$

Given a continuous mesh $\mathcal{M}$, we shall say, following [54, 55], that a discrete mesh $\mathcal{H}$ of the same domain Ω is a **unit mesh with respect to** $\mathcal{M}$, if each triangle $K \in \mathcal{H}$, defined by its list of edges $(\mathbf{e}_i)_{i=1\ldots3}$, verifies:

$$\forall i \in [1,3], \quad \ell_{\mathcal{M}}(\mathbf{e}_i) \in \left[\frac{1}{\sqrt{2}}, \sqrt{2}\right],$$

in which the length of an edge $\ell_{\mathcal{M}}(\mathbf{e}_i)$ is defined as follows:

$$\ell_{\mathcal{M}}(\mathbf{e}_i) = \int_0^1 \sqrt{{}^t\mathbf{ab}\,\mathcal{M}(\mathbf{a}+t\,\mathbf{ab})\,\mathbf{ab}}\,\mathrm{d}t, \quad \text{with } \mathbf{e}_i = \mathbf{ab},$$

The unit edge property of unit mesh writes also in short:

For a unit mesh $\mathbf{x}^{\mathcal{M}}$, any edge $\mathbf{x}_{ij}^{\mathcal{M}}$ satisfies $\left(\mathbf{x}_{ij}^{\mathcal{M}} - \mathbf{x}_{ij}^{\mathcal{M}}, \mathcal{M}(\mathbf{x}_{ij}^{\mathcal{M}} - \mathbf{x}_{ij}^{\mathcal{M}})\right) = 1.$

We want to emphasize that the set of all the discrete meshes that are unit meshes with respect to a unique $\mathcal{M}$ contains an infinite number of meshes, but these meshes have properties sufficiently close to each others so that we consider these meshes as an equivalence class of meshes. We henceforward denote by $\mathcal{M}$ both the metric and the corresponding unit mesh.

7.3.2 Optimal continuous metric

We recall, following [54, 55], the main features of the metric-based analysis initiated in several papers like [35, 25, 2]. The continuous interpolation error of a function u defined on the computational domain is denoted now:

$$u - \pi_{\mathcal{M}} u = |tr(\mathcal{M}^{-\frac{1}{2}}|H_u|\mathcal{M}^{-\frac{1}{2}})| \tag{7.9}$$

where H_u is the Hessian of u. Let denote also $\mathcal{M}$ a unit mesh for metric $\mathcal{M}$. We shall use the estimate

$$|u - \Pi_{\mathcal{M}} u| \approx \frac{1}{8}|u - \pi_{\mathcal{M}} u|. \tag{7.10}$$

Once we have a continuous tensorial error kernel, we consider minimizing:

$$j_p(\mathcal{M}) = \|u - \pi_{\mathcal{M}} u\|_{\mathbf{L}^p(\Omega_h)} \tag{7.11}$$

and we define as optimal metric the one which minimizes the right hand side under the constraint of a total number of vertices equal to a parameter N. In the case of a bounded p, after solving analytically this optimization problem, we get -without

using the fact that H is anything but a positive symmetric matrix- the unique optimal $(\mathcal{M}_{\mathbf{L}^p}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ as:

$$\mathcal{M}_{\mathbf{L}^p} = \mathcal{K}_p(1,H) \quad \text{with}$$
$$\mathcal{K}_p(1,H) = D_{\mathbf{L}^p}\,(\det(H))^{\frac{-1}{2p+2}}\,H \quad \text{and} \quad D_{\mathbf{L}^p} = N^{\frac{2}{2}}\left(\int_\Omega (\det(H))^{\frac{p}{2p+2}}\right)^{-\frac{2}{2}}, \tag{7.12}$$

where $D_{\mathbf{L}^p}$ is a global normalization term set to obtain a continuous mesh with complexity N and $(\det(H))^{\frac{-1}{2p+2}}$ is a local normalization term accounting for the sensitivity of the $\mathbf{L}^p$ norm.

A particular case: L^∞-norm/iso-distribution It is important to remark that error iso-distribution is taken into account by setting $p=\infty$, a limiting case for which we get:

$$(\det(H))^{\frac{-1}{2\infty+2}} = 1.$$

and

$$\mathcal{M}_{\mathbf{L}^\infty} = \mathcal{K}_\infty(1,H) \quad \text{with} \quad \mathcal{K}_\infty(1,H) = D_{\mathbf{L}^\infty}\,H$$

where $D_{\mathbf{L}^\infty}$ is defined from the specification of the number of nodes of the mesh.

Another way to see it is to write that the error is uniform, indeed:

$$\mathcal{M}_{\mathbf{L}^\infty}(\mathbf{x}) = \text{const.(indep. of } \mathbf{x})\ H$$

implies that:

$$\text{trace}\big(\mathcal{M}_{\mathbf{L}^\infty}^{-\frac{1}{2}}(\mathbf{x})H(\mathbf{x})\mathcal{M}_{\mathbf{L}^\infty}^{-\frac{1}{2}}(\mathbf{x})\big) = \text{const.(indep. of } \mathbf{x}).$$

Main case under study: L^1-norm optimisation The rest of the paper concentrates with the case:

$$p = 1$$

Replacing the optimal metric $\mathcal{M}_{\mathbf{L}^1}$ in the L^1 norm shows that second-order convergence is obtained for smooth contexts. This can also be extended to non-smooth ones, cf. [59].

Let k a sufficiently smooth scalar function defined on Ω. We shall be, in the sequel, interested in minimizing the right-hand side of:

$$|(k, u - \Pi_{\mathcal{M}} u)_\Omega| \approx \int_\Omega \text{trace}\big(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})|k(\mathbf{x})H(\mathbf{x})|\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})\big)\mathrm{d}\mathbf{x}. \tag{7.13}$$

The optimum metric is given by:

$$\mathcal{M}_{opt}^{1,k} = \mathcal{K}_1(k,H) \quad \text{with}$$
$$\mathcal{K}_1(k,H) = D_{opt}^{1,k}\,(\det|kH|)^{\frac{-1}{4}}\,|kH| \quad \text{and} \quad D_{opt}^{1,k} = N\left(\int_\Omega (\det|kH|)^{\frac{1}{4}}\right). \tag{7.14}$$

It is interesting to compare this result with the result of equidistribution, at least for the particular case of an interpolation error. We observe that:

$$\mathcal{M}_{opt}^{1,k} = \text{const.} \quad |k|^{\frac{3}{4}}|(\det|H|)^{\frac{-1}{4}}|H| \;=\; \text{const.} \;\; |H_k|$$
$$H_k = |k|^{\frac{3}{4}}|(\det|H|)^{\frac{-1}{4}}H \tag{7.15}$$

Which means that the error minimisation in $L^1_{weight,k}$ is equivalent to an equidistribution process with a matrix H corrected by a scalar factor $|k|^{\frac{3}{4}}|(\det|H|)^{\frac{-1}{4}}$:

$$\mathcal{M}_{opt}^{1,k} = \text{const.} \;\; \mathcal{K}_\infty(|k|^{\frac{3}{4}}|(\det|H|)^{\frac{-1}{4}}, H).$$

To synthetize, the continuous mesh/metric method yields the mesh adaptation solution under the form of a continuous KKT system involving the continuous initial PDE, its continuous adjoint, and a stationarity condition explicitly solved by (7.14). In practice, the KKT system is discretized and then solved.

The rest of the paper examines the approach closer to [30, 31], refered in this paper as a tensorial method, which consists in the direct building of a discrete KKT system.

7.4 Approximation of metric properties

In contrast to the continuous method, the tensorial method works only in a discrete context. Also, the optimality system of the tensorial formulation relies on an edge-based error modelling. Then most of the important discrete fields need to be cast in an *edge-based format*. We introduce a few notations for this.

7.4.1 Generic mesh notations

Given a mesh $\mathcal{H}_{\mathbf{x}}$, we can define the following partitions:

- a *mesh-vertex* is a vertex of numero i and coordinates $\mathbf{x}_i$ of an element of the mesh.

- when there is an *edge* between vertex i and vertex j, we denote $\mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i$.

- two tetrahedra m and n having a common face have face mn or face nm as common face.

- *elements* : triangles (i, j, k) or tetrahedra (i, j, k, l). Elements are divided in *sub-elements*: 6 *subtriangles* using medians and 24 *subtetrahedra* using median plans. The vertices of a subtetrahedron are : a mesh-vertex i , a center I_{ij} of an edge ij having i as extremity, the centroid g_{ijk} of a face ijk containing i and j, the element centroid G_{ijkl}. The measure of a subtetrahedron of the tetrahedron T is

$1/24\ meas(T)$.

- *cell* i : for a vertex i of the mesh, cell i is union of sub-elements having i as vertex of the sub-element. A cell measure is defined as
$$meas_{\mathbf{x}}(i) = \frac{1}{dim+1} \sum_{T_{\mathbf{x}} \ni i} \quad meas(T_{\mathbf{x}})$$
where $T_{\mathbf{x}}$ are elements of $\mathcal{H}_{\mathbf{x}}$ containing i.

- *2D-diamond* D_{ij} : union of the 4 subtriangles (of triangles ijk and ijl) having a side included in edge ij.

- *face-diamond* $\bar{D}_{mn}$, where m and n are two tetrahedra having a common face ijk : union of 6 subtetrahedra having a subtriangle of the common face ijk as face.

- *edge-diamond* D_{ij}: union of subtetrahedra having having a side included in edge ij.

The integral of a function e_{ij} defined on the edges can be approximated by:
$$err_{L^1} = \sum_i meas_{\mathbf{x}}(i)\Gamma(i)^{-1} \sum_j e_{ij}$$
or introducing the diamond partition $\Omega = \cup \bar{D}_{mn}$ where m and n are elements with a common face:
$$err_{L^1} = \frac{1}{3} \sum_{\bar{D}_{mn}} \ meas_{\mathbf{x}}(\bar{D}_{mn}) \ (e_{ij} + e_{ik} + e_{jk}).$$
where i, j, k are vertices of the face mn.

7.4.2 Discretizing an arbitrary continuous metric on a background mesh

In order to find the optimal metric we are given a background mesh $\mathbf{x}$. We assume that the unknown metric $\mathcal{M}$ is defined on the vertices $\mathcal{M}(\mathbf{x}_i) = \mathcal{M}^i$ of the background mesh and that it is P^1-continuously interpolated. The total number of nodes can be approximated on the mesh $\mathbf{x}$ by a quadrature of (7.8) as follows:
$$\mathcal{C}(\mathcal{M}) = \sum_i meas_{\mathbf{x}}(i)\sqrt{det(\mathcal{M}^i)}.$$
To simplify, *we assume that the unit mesh is a deformation of* $\mathbf{x}$*, and that* $\mathbf{x}_{ij}^{\mathcal{M}}$ *and* $\mathbf{x}_{ij}$ *are colinear*. Then we can derive from the unit-mesh property a relation between the edge lengths of unknown mesh and the edge lengths of the background mesh:
$$(\mathbf{x}_{ij}^{\mathcal{M}}, \mathcal{M}\mathbf{x}_{ij}^{\mathcal{M}}) = 1 = (\mathbf{x}_{ij}\frac{|\mathbf{x}_{ij}^{\mathcal{M}}|}{|\mathbf{x}_{ij}|}, \mathcal{M}\mathbf{x}_{ij}\frac{|\mathbf{x}_{ij}^{\mathcal{M}}|}{|\mathbf{x}_{ij}|}) = (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})\frac{|\mathbf{x}_{ij}^{\mathcal{M}}|^2}{|\mathbf{x}_{ij}|^2}$$
$$\Rightarrow \mathbf{x}_{ij}^{\mathcal{M}} \approx \mathbf{x}_{ij}(\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})^{-\frac{1}{2}}.$$
In order now to evaluate the approximation error provoked by the application of the unit-mesh, we need to define a *generic error model*.

7.5 Second-order error of a metric on a background mesh

To any given metric, *i.e.* to any given mesh, should correspond a numerical error field. Let us define a generic family of error field with values on mesh edges. We restrict to second-order or quadratic errors, on the model of P_1-interpolation error.

Definition : An *edge-based second-order (or quadratic) error* produced by the use of the unit mesh $\mathbf{x}_{\mathcal{M}}$ of metric $\mathcal{M}$ has an intensity defined on edge $\mathbf{x}_{ij}^{\mathcal{M}}$ by:

$$e_{ij}^{\mathcal{M}} = \bar{e}_{ij}\ |\mathbf{x}_{ij}^{\mathcal{M}}|^2.$$

in which $\bar{e}_{ij}$ depends only on location and direction of $\mathbf{x}_{ij}^{\mathcal{M}}$, typically:

$$e_{ij}^{\mathcal{M}} = \quad |\mathbf{x}_{ij}^{\mathcal{M}}|^2 \quad \bar{e}_{ij}(\tfrac{1}{2}(\mathbf{x}_i^{\mathcal{M}} + \mathbf{x}_j^{\mathcal{M}}), \tfrac{\mathbf{x}_{ij}^{\mathcal{M}}}{|\mathbf{x}_{ij}^{\mathcal{M}}|}).\square$$

Since we *a priori* do not know the optimal metric nor its mesh, it is useful to evaluate this error on a given background mesh $\mathbf{x}$. We use that the unit mesh is a deformation of $\mathbf{x}$ in such a way that $\mathbf{x}_{ij}^{\mathcal{M}}$ and $\mathbf{x}_{ij}$ are colinear. Then the intensity $e_{ij}^{\mathcal{M}}$ of the error with the unit mesh evaluated at middle of $\mathbf{x}_{ij}$ of the initial mesh writes:

$$e_{ij}^{\mathcal{M}} = \quad |\mathbf{x}_{ij}|^2 \quad (\mathbf{x}_{ij}, \mathcal{M}_{ij}\mathbf{x}_{ij})^{-1} \quad \bar{e}_{ij}(\frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j), \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|}) \tag{7.16}$$

where $\mathcal{M}_{ij}$ is evaluated on $\frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j)$. The mesh adaptation problem will be set as the research of the discrete metric, defined on mesh vertices and linearly interpolated, of a given number of nodes N

$$\mathcal{C}(\mathcal{M}) = N,$$

and minimizing the discrete error norm:

$$j(\mathcal{M}) = \sum_i meas_{\mathbf{x}}(i)\frac{1}{\Gamma(i)}\sum_{ij\ni i}\ e_{ij}^{\mathcal{M}}. \tag{7.17}$$

In Section 7.6 we determine the optimal mesh for this type of error, as far as $\bar{e}_{ij}$ is identified. The rest of the present section is devoted to the description of three examples of quadratic errors.

7.5.1 First example: interpolation error

The error committed in interpolating a smooth function on a P^1 mesh is a quadratic error. Indeed, the weighted P_1-interpolation error of a quadratic function u on $\mathbf{x}_{ij}^{\mathcal{M}}$ can be discretized similarly to (7.9),(7.10) as follows:

$$\int |g||u - \Pi_h u|\mathrm{d}\Omega \preceq \ \tfrac{1}{8}\ \textstyle\sum_i\ meas_{\mathbf{x}}(i)\Gamma(i)^{-1}\ \sum_j\ e_{ij}^{\mathcal{M},g,u}(\mathbf{x}_{ij})$$

with

$$e_{ij}^{\mathcal{M},g,u} = \quad |\mathbf{x}_{ij}^{\mathcal{M}}|^2 \quad |g_{ij}|\ |H_{ij}| \cdot \frac{\mathbf{x}_{ij}^{\mathcal{M}}}{|\mathbf{x}_{ij}^{\mathcal{M}}|} \cdot \frac{\mathbf{x}_{ij}^{\mathcal{M}}}{|\mathbf{x}_{ij}^{\mathcal{M}}|} \quad ,$$

and where $H_{ij} = H(\frac{1}{2}(\mathbf{x}_i^{\mathcal{M}} + \mathbf{x}_j^{\mathcal{M}}))$, $H(\mathbf{x})$ being the Hessian of u at point $\mathbf{x}$, and $g_{ij} = g(\frac{1}{2}(\mathbf{x}_i^{\mathcal{M}} + \mathbf{x}_j^{\mathcal{M}}))$. Here $\preceq$ holds for an inequality applying for a sufficiently fine mesh, with a multiplicative constant close to 1. It can be evaluated on a background mesh as follows:

$$e_{ij}^{\mathcal{M},g,u}(\mathbf{x}_{ij}) = \quad |\mathbf{x}_{ij}^{\mathcal{M}}|^2 \quad \bar{e_{ij}}(\mathbf{x}_{ij}) = \ (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})^{-1} \ \ |\mathbf{x}_{ij}|^2 \quad \bar{e_{ij}}(\mathbf{x}_{ij})$$

with:

$$\bar{e_{ij}}(\mathbf{x}_{ij}) = |g_{ij}(\mathbf{x}_{ij})|\ |H_{ij}(\mathbf{x}_{ij})| \cdot \frac{\mathbf{x}_{ij}^{\mathcal{M}}}{|\mathbf{x}_{ij}^{\mathcal{M}}|} \cdot \frac{\mathbf{x}_{ij}^{\mathcal{M}}}{|\mathbf{x}_{ij}^{\mathcal{M}}|} \quad = \quad |g_{ij}(\mathbf{x}_{ij})|\ |H_{ij}(\mathbf{x}_{ij})| \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|} \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|}.$$

Then this first example takes place into the context of (7.16)(7.17).

7.5.2 Goal-oriented error

Let u be the solution of (7.1) and $u_{\mathcal{M}}$ the discrete solution of (7.2) where the mesh is an unit mesh for metric $\mathcal{M}$. A typical goal-oriented analysis relies on the minimization of the error $\delta j_{goal}(\mathcal{M})$ committed in the evaluation of the scalar output $j = (g, u)$, error which we write as follows:

$$\delta j_{goal}(\mathcal{M}) = |(g, u - u_{\mathcal{M}})| = |(g, \Pi_{\mathcal{M}} u - u_{\mathcal{M}} + u - \Pi_{\mathcal{M}} u)|. \tag{7.18}$$

According to the Aubin-Nitsche analysis, this error is second-order with respect to mesh size. Let us define the discrete adjoint state u^*_{goal}:

$$\forall \psi_{\mathcal{M}} \in \ V_{\mathcal{M}}, \quad a(\psi_{\mathcal{M}}, u^*_{goal}) = (\psi_{\mathcal{M}}, g). \tag{7.19}$$

In the sequel, we use a fixed-point in which the adjoint is frozen with respect to the metric $\mathcal{M}$. Injecting (7.19) in (7.18) we get:

$$(g, \Pi_{\mathcal{M}} u - u_{\mathcal{M}} + u - \Pi_{\mathcal{M}} u) = a(\Pi_{\mathcal{M}} u - u_{\mathcal{M}}, u^*_{goal}) + (g, u - \Pi_{\mathcal{M}} u)$$

and, using (7.4),

$$(g, \Pi_{\mathcal{M}} u - u_{\mathcal{M}} + u - \Pi_{\mathcal{M}} u) = a(\Pi_{\mathcal{M}} u - u, u^*_{goal}) + (f - \Pi_{\mathcal{M}} f, u^*_{goal}) + (g, u - \Pi_{\mathcal{M}} u)$$

thus

$$\delta j_{goal}(\mathcal{M}) \approx |a(\Pi_{\mathcal{M}} u - u, u^*_{goal}) + (f - \Pi_{\mathcal{M}} f, u^*_{goal}) + (g, u - \Pi_{\mathcal{M}} u)|$$

or:

$$\delta j_{goal}(\mathcal{M}) \preceq |a(\Pi_{\mathcal{M}} u - u, u^*_{goal})| + |(f - \Pi_{\mathcal{M}} f, u^*_{goal})| + |g||u - \Pi_{\mathcal{M}} u| \tag{7.20}$$

The RHS of (7.20) involves three terms. The *second* and *third terms* give Hessian-like quadratic errors $e_{ij}^{\mathcal{M},u^*_{goal},f}$ and $e_{ij}^{\mathcal{M},g,u}$:

$$\begin{aligned}
&|(f-\Pi_{\mathcal{M}}f,u^*_{goal})|+|g||\pi_{\mathcal{M}}u_{\mathcal{M}}-u_{\mathcal{M}}| \\
&\qquad \preceq \sum_i meas_{\mathbf{x}}(i)\Gamma(i)^{-1}\sum_{ij\ni i}\left(e_{ij}^{\mathcal{M},u^*_{goal},f}+e_{ij}^{\mathcal{M},g,u}\right) \\
&\quad \preceq \sum_i meas_{\mathbf{x}}(i)\Gamma(i)^{-1}\sum_{ij\ni i}(\mathbf{x}_{ij},\mathcal{M}\mathbf{x}_{ij})^{-1}\quad |\mathbf{x}_{ij}|^2\left(\bar{e_{ij}}^{u^*_{goal},f}+\bar{e_{ij}}^{g,u}\right)
\end{aligned}$$

with

$$\bar{e_{ij}}^{u^*_{goal},f}(\mathbf{x}_{ij})=|u^*_{goal,ij}|\ |H^f_{ij}|\cdot\frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|}\cdot\frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|}$$

$$\bar{e_{ij}}^{g,u}(\mathbf{x}_{ij})=|g_{ij}|\ |H^u_{ij}|\cdot\frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|}\cdot\frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|}$$

and

$$u^*_{goal,ij}=u^*_{goal}(\frac{\mathbf{x}_i+\mathbf{x}_j}{2})$$

$$g_{ij}=g(\frac{\mathbf{x}_i+\mathbf{x}_j}{2})$$

$$H^f_{ij}=H^f(\frac{\mathbf{x}_i+\mathbf{x}_j}{2})$$

$$H^u_{ij}=H^u(\frac{\mathbf{x}_i+\mathbf{x}_j}{2}).$$

The *first term* of (7.20)'s RHS is more complex. It can be estimated in a different way from the continuous method presented in [13] and used in [23] (chapter 6 of this thesis). Indeed,

$$\begin{aligned}
|a(\Pi_{\mathcal{M}}u-u,u^*_{goal})| &= |\int_\Omega \nabla(\Pi_{\mathcal{M}}u-u)\nabla\Pi_{\mathcal{M}}u^*_{g,\mathcal{M}}\mathrm{d}\mathbf{x}| \\
&\preceq \sum_{\partial T_{mn}} |(\nabla u^*_{goal}|_{T_m}-\nabla u^*_{goal}|_{T_n})\cdot\mathbf{n}_{mn}|\quad\int_{\partial T_{mn}}|\Pi_{\mathcal{M}}u-u|\ \mathrm{d}\sigma. \quad (7.21)
\end{aligned}$$

Study of the 2D case. In the 2D case, ∂T_{mn} is exactly an edge ij. We introduce the interpolation error estimate on ij, and its measure . We get from (7.21):

$$|a(\Pi_{\mathcal{M}}u-u,u^*_{goal})|\preceq\sum_{ij}\kappa_{ij}(u^*_{goal})\ |\mathbf{x}_{ij}|^3(\mathbf{x}_{ij},\mathcal{M}\mathbf{x}_{ij})e^u_{ij}$$

where the sum is taken over the edges with

$$\kappa_{ij}(u^*_{goal})=|(\nabla u^*_{goal}|_{T_m}-\nabla u^*_{goal}|_{T_n})\cdot\mathbf{n}_{mn}|$$

We need know to identify the *local intensity of the error term* by comparing the RHS with an integral over the computational domain. This integral is taken as a sum over the diamond cells:

$$|a(\Pi_{\mathcal{M}}u - u, u^*_{goal})| \preceq \sum_{ij} |D_{ij}||D_{ij}|^{-1}\kappa_{ij}(u^*_{goal})\ |\mathbf{x}_{ij}|^3(\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})\bar{e}^u_{ij}$$

which shows that $|D_{ij}|^{-1}\kappa_{ij}(u^*_{goal})\ |\mathbf{x}_{ij}|^3(\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})\bar{e}^u_{ij}$ is the local error intensity. The cellwise error integral then writes:

$$\mathcal{E}^{\mathcal{M},a} = \sum_i \frac{1}{\Gamma(i)} \sum_{ij\ni i} |\mathbf{x}_{ij}|^2(\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})\bar{e}^a_{ij}$$

with

$$\bar{e}^a_{ij} = |\mathbf{x}_{ij}||D_{ij}|^{-1}\kappa_{ij}(u^*_{goal})\bar{e}^u_{ij}.$$

We observe that for a Cartesian mesh of mesh size Δx, term $|\mathbf{x}_{ij}|$ is $O(\Delta x)$, term $|D_{ij}|^{-1}$ is $O(\Delta x)^{-2}$, term $\kappa_{ij}(u^*_{goal})$ is $O(\Delta x)$ (non divided difference of normal gradient), and e^u_{ij}, which is a directional second derivative is $O(1)$. The error intensity is $O(1)$.

Study of the 3D case. The intersection ∂T_{mn} of two elements T_m and T_n is a common face with vertices i, j, k and an area $area(mn)$. The following quantity is again known:

$$\kappa_{mn}(u^*_{goal}) = |\left(\nabla u^*_{goal}\right)|_{T_m} \cdot \mathbf{n}_{mn} - \left(\nabla u^*_{goal}\right)|_{T_n} \cdot \mathbf{n}_{mn}|.$$

The remaining expression can be expressed in terms of interpolation errors:

$$\int_{\partial T_{mn}} |\Pi_{\mathcal{M}}u - u| \approx \frac{1}{3} area(mn)(e^{\mathcal{M},u}_{ij} + e^{\mathcal{M},u}_{ik} + e^{\mathcal{M},u}_{kj})$$

with (for $\alpha\beta$=ij,ik and kj):

$$e^{\mathcal{M},u}_{\alpha\beta} = \ (\mathbf{x}_{\alpha\beta}, \mathcal{M}\mathbf{x}_{\alpha\beta})^{-1} \quad |\mathbf{x}_{\alpha\beta}|^2\ \bar{e}^u_{\alpha\beta}$$

and:

$$\bar{e}^u_{\alpha\beta}(\mathbf{x}_{\alpha\beta}) = \ |H^u_{\alpha\beta}| \cdot \frac{\mathbf{x}_{\alpha\beta}}{|\mathbf{x}_{\alpha\beta}|} \cdot \frac{\mathbf{x}_{\alpha\beta}}{|\mathbf{x}_{\alpha\beta}|}.$$

We get:

$$|a(\Pi_{\mathcal{M}}u - u, u^*_{goal})| \ \preceq \ \sum_{\bar{D}_{mn}} \frac{area(mn)}{3}\ (e^{\mathcal{M},u}_{ij} + e^{\mathcal{M},u}_{ik} + e^{\mathcal{M},u}_{jk})\ \kappa_{mn}(u^*_{goal})$$

Let us convert the RHS into an edge-by-edge sum:

$$|a(\Pi_{\mathcal{M}}u - u, u^*_{goal})| \ \preceq \ \sum_{\bar{D}_{mn}} \qquad \sum_{\alpha\beta=ij,ik,jk} area(mn)\ \frac{1}{3}e^{\mathcal{M},u}_{\alpha\beta}\kappa_{mn}(u^*_{goal})$$

$$= \sum_{\text{edges } ij} \sum_{\bar{D}_{mn} \ni ij} area(mn)\ \frac{1}{3} e_{ij}^{\mathcal{M},u} \kappa_{mn}(u^*_{goal}) \ = \sum_{\text{edges } ij} e_{ij}^{\mathcal{M},a}\ |D_{ij}|$$

where we recognize the edge-by-edge integral of a field $e_{ij}^{\mathcal{M},a}$ defined on edges, with the notation:

$$e_{ij}^{\mathcal{M},a} = \frac{1}{3}\ \frac{1}{|D_{ij}|}\ e_{ij}^{\mathcal{M},u} \sum_{\bar{D}_{mn} \ni ij} area(mn)\ \kappa_{mn}(u^*_{goal}). \tag{7.22}$$

Equivalently (at the second order) we get the(7.16)(7.17) format:

$$|a(\Pi_{\mathcal{M}} u - u, u^*_{goal})| \ \preceq \ \sum_i meas_{\mathbf{x}}(i) \frac{1}{\Gamma(i)} \sum_{ij \ni i} e_{ij}^{\mathcal{M},a}.$$

We can then define:

$$\bar{e}_{ij}^{a} = \ (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij}) \quad |\mathbf{x}_{ij}|^{-2}\ e_{ij}^{\mathcal{M},a} = \frac{1}{3}\ \frac{1}{|D_{ij}|}\ \bar{e}_{ij}^{u} \sum_{\bar{D}_{mn} \ni ij} area(mn)\ \kappa_{mn}(u^*_{goal})$$

which does not depend on $\mathcal{M}$.

Synthesis. Finally, gathering the estimate of the three RHS, we get:

$$\delta j_{goal}(\mathcal{M}) \ \preceq \\ \sum_i meas_{\mathbf{x}}(i) \Gamma(i)^{-1} \sum_{ij \ni i} (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})^{-1} \quad |\mathbf{x}_{ij}|^2 \left(\bar{e}_{ij}^{a} + \bar{e}_{ij}^{u^*_{goal},f} + \bar{e}_{ij}^{g,u} \right)$$

which takes place in the context of (7.16)(7.17).

Remark: The *a priori estimates* at the starting of this analysis relies on edge-based terms which are essentially products of : $\kappa_{mn}(u^*_{goal})$, a second-order directional derivative, normal to edge in 2D, of the adjoint, times $\bar{e}_{ij}^{u}$, a second-order directional derivative in edge direction (in 2D). In the analysis proposed in [13] and used in [23] (chapter 6), the majoration of the directional adjoint derivative consists in using the largest eigenvalue $\rho(H_{u^*})$ of its Hessian. Further, the demonstration is obtained thanks to the assumption that the mesh stretching is bounded. *In the present study, the second directional derivative of u^*_{goal} is directly taken into account, which makes the estimate more accurate, and this is done without any extra assumption. This progress is the central output of the present work.*

7.5.3 Norm-oriented error

The norm-oriented analysis is defined in details in the case of the continuous metric method in [20]. In short, this method focusses on the minimization of the following norm with respect to the mesh $\mathcal{M}$:

$$\delta j(\mathcal{M}) = ||u - u_{\mathcal{M}}||^2_{L^2(\Omega)}. \tag{7.23}$$

Introducing $g_{\mathcal{M}} = u - u_{\mathcal{M}}$, we get a formulation similar to the goal-oriented formulation:

$$\delta j(\mathcal{M}) = (g_{\mathcal{M}}, u - u_{\mathcal{M}}). \tag{7.24}$$

But in the practical application $u - u_{\mathcal{M}}$ is not known. We approximate it by a function close to it, which we call a corrector. Let us define:

$$g_{\mathcal{M}} = \bar{u}'_{prio,\mathcal{M}} - (\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}})$$

in which $\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}$ is a Hessian-based approximation of the interpolation error and in which $\bar{u}'_{prio,\mathcal{M}}$ is the solution of:

$$a(\bar{u}'_{prio,\mathcal{M}}, \phi) = \sum_{\partial T_{ij}} (\nabla\phi|_{T_i} - \nabla\phi|_{T_j}) \cdot \mathbf{n}_{ij} \int_{\partial T_{ij}} (\pi_{\mathcal{M}} u_{\mathcal{M}} - u_{\mathcal{M}}) \, \mathrm{d}\sigma \; - \; (\phi, \pi_{\mathcal{M}} f_{\mathcal{M}} - f_{\mathcal{M}}). \tag{7.25}$$

Another example with a RHS evaluated on a two-times finer grid is given in [20].

Let us define the discrete adjoint state u^*_{norm}:

$$\forall \psi_{\mathcal{M}} \in \; V_{\mathcal{M}}, \quad a(\psi_{\mathcal{M}}, u^*_{norm}) = (\psi_{\mathcal{M}}, g_{\mathcal{M}}). \tag{7.26}$$

Then, similarly to previous section we shall minimize:

$$\delta j_{norm}(\mathcal{M}) \approx |a(\Pi_{\mathcal{M}} u - u, u^*_{norm}) + (f - \Pi_{\mathcal{M}} f, u^*_{norm}) + (g_{\mathcal{M}}, u - \Pi_{\mathcal{M}} u)|.$$

Turning now to the *tensorial formulation*, we minimize:

$$\mathcal{E}(\mathcal{M}) \; = \; \sum_i meas_{\mathbf{x}}(i)\Gamma(i)^{-1} \sum_{ij \ni i} (\mathbf{x}_{ij}, \mathcal{M}\mathbf{x}_{ij})^{-1} \; |\mathbf{x}_{ij}|^2 \left(\bar{e_{ij}}^{\mathcal{M},a} + \bar{e}_{ij}^{u^*_{norm},f} + \bar{e}_{ij}^{g,u} \right)$$

with

$$\bar{e_{ij}}^{u^*_{norm},f} = |u^*_{norm,ij}| \; |H^f_{ij}| \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|} \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|}$$

$$\bar{e_{ij}}^{g,u} = |g_{ij}| \; |H^u_{ij}| \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|} \cdot \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|}$$

$$\bar{e_{ij}}^{\mathcal{M},a} = \; |\mathbf{x}_{ij}| \; |D_{ij}|^{-1} \kappa_{ij}(u^*_{norm}) \; \bar{e}^u_{ij} \tag{7.27}$$

and with $\kappa_{mn}(u^*_{norm}) = |\,(\nabla u^*_{norm})\,|_{T_m} \cdot \mathbf{n}_{mn} - (\nabla u^*_{norm})\,|_{T_n} \cdot \mathbf{n}_{mn}|$. This again takes place in the context of (7.16)(7.17).

7.6 Optimal metric

The purpose is to minimize with respect to the metric for a given number of vertices N a functional of the form:

$$\mathcal{E}(\mathcal{M}) \;=\; \sum_i meas_{\mathbf{x}}(i)\Gamma(i)^{-1}\sum_{\mathbf{x}_{ij}} (\mathbf{x}_{ij})^2(\mathbf{x}_{ij},\mathcal{M}\mathbf{x}_{ij})^{-1}\;\bar{e}_{ij}$$

which takes into account the three different types of errors defined in the previous sections. We solve this in two steps as in [54, 55]: first we minimize the functional in a point of the computational domain and get a first property of the solution, second we finish determining the optimum by solving a sub-problem on the whole domain.

7.6.1 Pointwise optimal metric

The purpose of the pointwise metric optimisation is to look for the optimal stretching of the metric, independantly of mesh density. The number of vertices is fixed. We consider metric $\mathcal{M}_0$ such that the determinant, or product of eigenvalues is equal to unity, i.e. $\lambda_1\lambda_2\lambda_2 = 1$ or, equivalently $det(\mathcal{M}_0) = 1.$. We know that:

$$(\mathbf{x}_{ij})^2\;(\mathbf{x}_{ij},\mathcal{M}\mathbf{x}_{ij})^{-1}\;\;\bar{e}_{ij} = e_{ij}^{\mathcal{M}}\;\;\forall j.$$

In that expression, $(\mathbf{x}_{ij})^2$ and $(\mathbf{x}_{ij},\mathcal{M}\mathbf{x}_{ij})^{-1}$ are not vanishing for any couple of neighboring vertices i and j,

$$e_{ij}^{\mathcal{M}} = 0 \;\;\Leftrightarrow\;\; \bar{e}_{ij} = 0.$$

Now, for any i and any j belonging to $\Gamma(i)$ such that $\bar{e}_{ij} \neq 0$,

$$(\mathbf{x}_{ij})^{-2}\;(\mathbf{x}_{ij},\mathcal{M}\mathbf{x}_{ij})\;\;(\bar{e}_{ij})^{-1} = (e_{ij}^{\mathcal{M}})^{-1}.$$

Summing around the vertex i, it gives:

$$\sum_{\substack{j\in\Gamma(i)\\ |\bar{e}_{ij}\neq 0}} (\mathbf{x}_{ij})^{-2}\;(\bar{e}_{ij})^{-1}\;(\mathbf{x}_{ij},\mathcal{M}\mathbf{x}_{ij}) = \sum_{\substack{j\in\Gamma(i)\\ |\bar{e}_{ij}\neq 0}} (e_{ij}^{\mathcal{M}})^{-1}$$

For the sake of simplicity, let us denote: $D_i = \sum_{\substack{j\in\Gamma(i)\\ |\bar{e}_{ij}\neq 0}} (e_{ij}^{\mathcal{M}})^{-1}$.

We note that each $e_{ij}^{\mathcal{M}}$ is positive and therefore so is D_i. This implies:

$$D_i = \sum_{j\in\Gamma(i)} (\mathcal{M}\;\bar{e}_{ij}^{-\frac{1}{2}}|\mathbf{x}_{ij}|\mathbf{x}_{ij},\bar{e}_{ij}^{-\frac{1}{2}}|\mathbf{x}_{ij}|\mathbf{x}_{ij}) = \mathcal{M}\;:\;\sum_{j\in\Gamma(i)} \bar{e}_{ij}^{-\frac{1}{2}}|\mathbf{x}_{ij}|\mathbf{x}_{ij}\otimes\bar{e}_{ij}^{-\frac{1}{2}}|\mathbf{x}_{ij}|\mathbf{x}_{ij}.$$

Now, remembering that $A : B \;=\; tr({}^tA.B)$, it is interesting to choose (among other solutions):

$$\mathcal{M}^i \;=\; \frac{D_i}{dim}\left(\sum_{j\in\Gamma(i)} \bar{e}_{ij}^{-1}|\mathbf{x}_{ij}|^{-2}\mathbf{x}_{ij}\;\otimes\;\mathbf{x}_{ij}\right)^{-1}. \tag{7.28}$$

The optimal pointwise metric is then defined as:

$$\mathcal{M}_0^i \;=\; (det(\mathcal{M}^i))^{-\frac{1}{2}}\mathcal{M}^i. \tag{7.29}$$

7.6.2 Global optimal metric

The global optimal metric will be obtained by multiplying the pointwise metric by a scalar field to be determined:

$$\mathcal{M}^i_{opt} = C_i \; \mathcal{M}^i_0.$$

We search $(C_i)_i$ which minimizes

$$err_{L^1} = \sum_i meas_{\mathbf{x}}(i)\Gamma(i)^{-1} \sum_{\mathbf{x}_{ij}} \; (\mathbf{x}_{ij})^2 (\mathbf{x}_{ij}, C_i \mathcal{M}^i_0 \mathbf{x}_{ij})^{-1} \; \bar{e}_{ij}$$

or

$$err_{L^1} = \sum_i \; \alpha_i \; C_i^{-1} \quad ; \quad \text{with} \;\; \alpha_i = meas_{\mathbf{x}}(i)\Gamma(i)^{-1} \sum_{\mathbf{x}_{ij}} \; (\mathbf{x}_{ij})^2 (\mathbf{x}_{ij}, \mathcal{M}^i_0 \mathbf{x}_{ij})^{-1} \; \bar{e}_{ij}$$

while satisfying to the constraint: $\sum_i meas_{\mathbf{x}}(i)\sqrt{det(C_i \mathcal{M}^i_0)} = N$ or:

$$\sum_i \; \mu_i \; C_i^{\frac{dim}{2}} = N \;\; \text{with} \quad \mu_i \; = \; meas_{\mathbf{x}}(i)\sqrt{det(\mathcal{M}^i_0)}.$$

This can be simply solved by applying the variable change $d_i = \mu_i C_i^{\frac{dim}{2}}$, which gives:

$$\text{Min} \;\; \sum_i \eta_i d_i^{\frac{-2}{dim}} \;\; \text{under the constraint} \;\; \sum_i d_i = N, \tag{7.30}$$

with $\eta_i = \alpha_i \mu_i^{\frac{2}{dim}}$. The solution of (7.30) writes:

$$d_i = \left(\sum_j \eta_j^{\frac{dim}{2+dim}} \right)^{-1} \eta_i^{\frac{dim}{2+dim}} \; N.$$

Lemma: *The optimal metric is defined by:*

$$\mathcal{M}^i \; = \; C_i \; \mathcal{M}^i_0$$

with

$$\mathcal{M}^i_0 \; = \; (det(\mathcal{M}^i_1))^{-\frac{1}{2}} \mathcal{M}^i_1, \quad \mathcal{M}^i_1 \; = \; \frac{1}{dim} \left(\sum_{j \in \Gamma(i)} \; \bar{e}_{ij}^{-1} |\mathbf{x}_{ij}|^{-2} \mathbf{x}_{ij} \; \otimes \; \mathbf{x}_{ij} \right)^{-1},$$

$$C_i = \mu_i^{-\frac{2}{dim}} \left(\sum_j \eta_j^{\frac{dim}{2+dim}} \right)^{-\frac{2}{dim}} \eta_i^{\frac{2}{2+dim}} \; N^{\frac{2}{dim}},$$

$$\eta_i = \alpha_i \mu_i^{\frac{2}{dim}} \quad ; \quad \alpha_i = \frac{meas_{\mathbf{x}}(i)}{\Gamma(i)} \sum_{\mathbf{x}_{ij}} \; \frac{(\mathbf{x}_{ij})^2}{(\mathbf{x}_{ij}, \mathcal{M}^i_0 \mathbf{x}_{ij})} \; \bar{e}_{ij} \quad ; \quad \mu_i \; = \; meas_{\mathbf{x}}(i)\sqrt{det(\mathcal{M}^i_0)}. \square$$

7.7 Numerical examples

The analysis developed in this paper gives an answer to the same problem as in [22] in which the continuous approach were introduced and a series of test cases were presented for its evaluation. Our evaluation of the new method will then consist in recomputing these test cases and comparing the results with the results of [22]. We refer to [22] for a detailed presentation of each test case.

7.7.1 A 2D boundary layer test case

We study the 2D boundary layer test case for five different methods: uniform FMG, continuous Hessian-based adaptative FMG, tensorial Hessian-based adaptative FMG, continuous norm-oriented adaptative FMG and tensorial norm-oriented adaptative FMG. We can first compare the meshes obtained with the four different adaptative methods. At the begining, we have the uniform mesh given by Figure 7.1. Using this mesh, we compute an approximate solution and we use it the create an adapted mesh with the four methods:

- continuous Hessian-based adaptation gives Figure 7.2,
- tensorial Hessian-based adaptation gives Figure 7.3,
- continuous norm-oriented adaptation gives Figure 7.4,
- tensorial norm-oriented adaptation gives Figure 7.5.

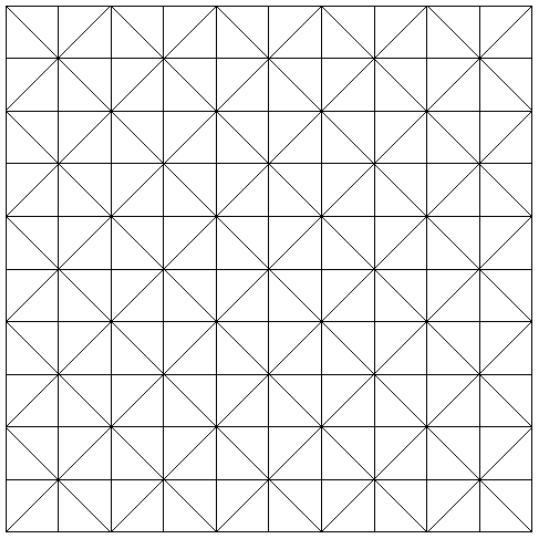

Figure 7.1: Initial uniform mesh.

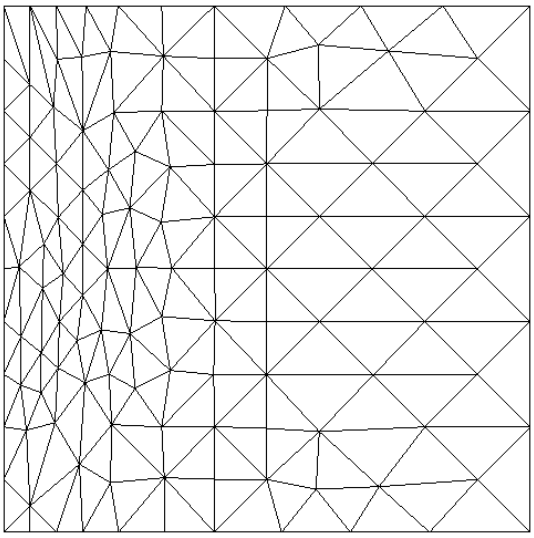

Figure 7.2: Adapted mesh obtained thanks to continuous Hessian-based adaptation.

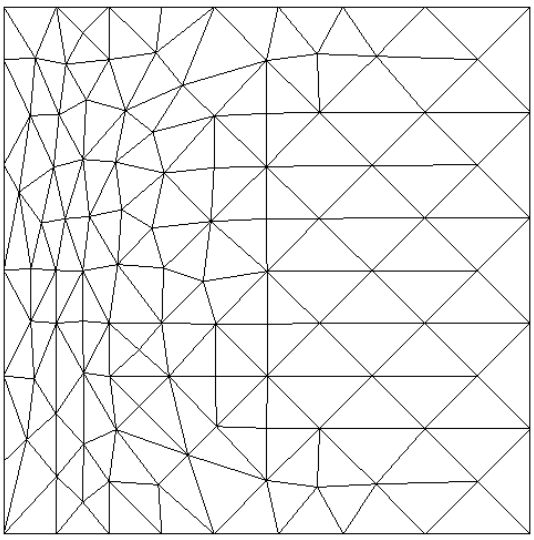

Figure 7.3: Adapted mesh obtained thanks to tensorial Hessian-based adaptation.

We have computed the results for the continuous case and for the tensorial case, using ten adaptations per phase. The convergences of the different methods are given in figures 7.6 and 7.7 in function of the number of points.

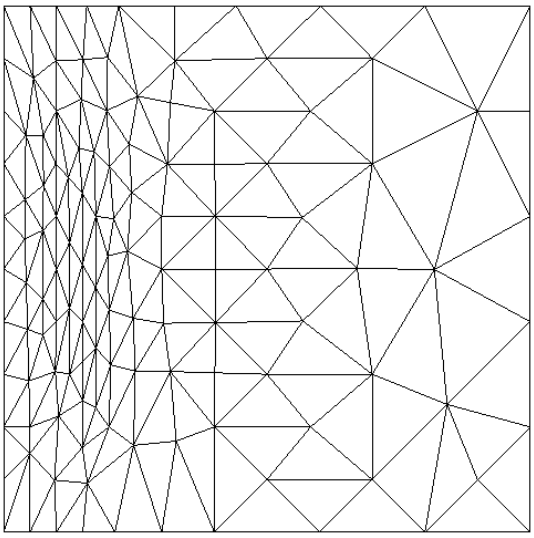

Figure 7.4: Adapted mesh obtained thanks to continuous norm-oriented adaptation.

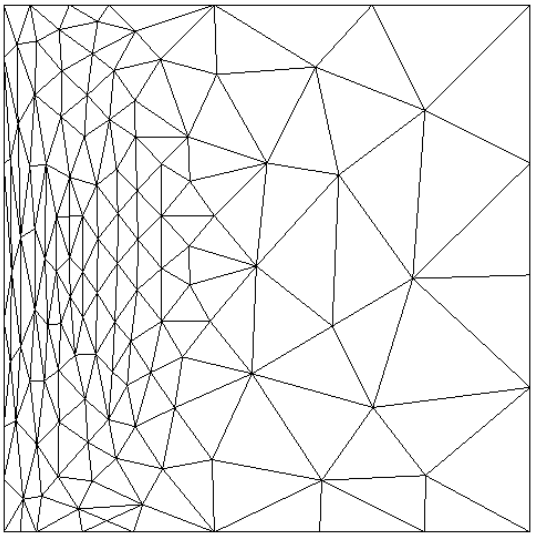

Figure 7.5: Adapted mesh obtained thanks to tensorial norm-oriented adaptation.

We can observe the uniform case in red, the Hessian-based continuous and tensoriel respectively in green and dark blue and the norm-oriented continuous and tensorial respectively in pink and clear blue, the black line being simply the order

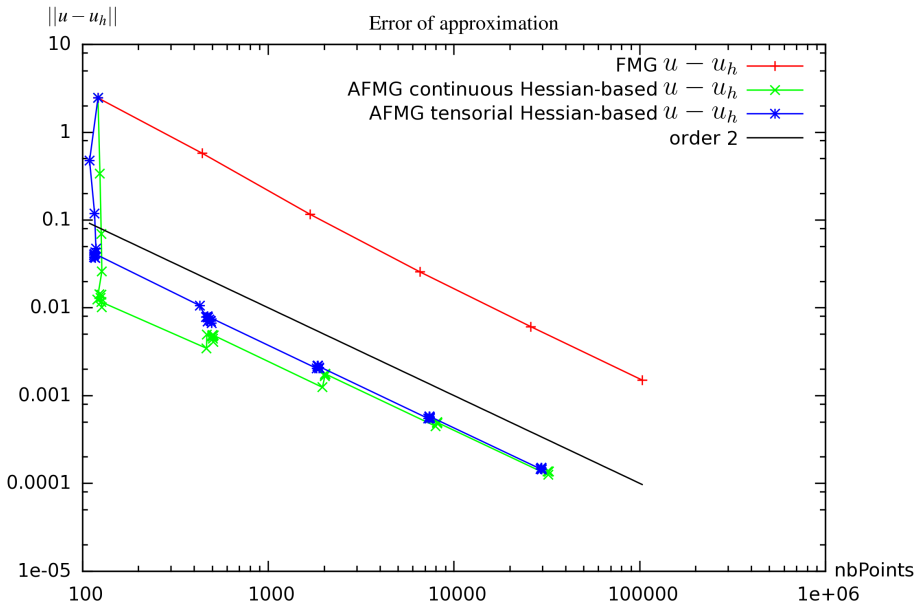

Figure 7.6: 2D boundary layer test case, Hessian-based methods: error convergence in terms of number of vertices.

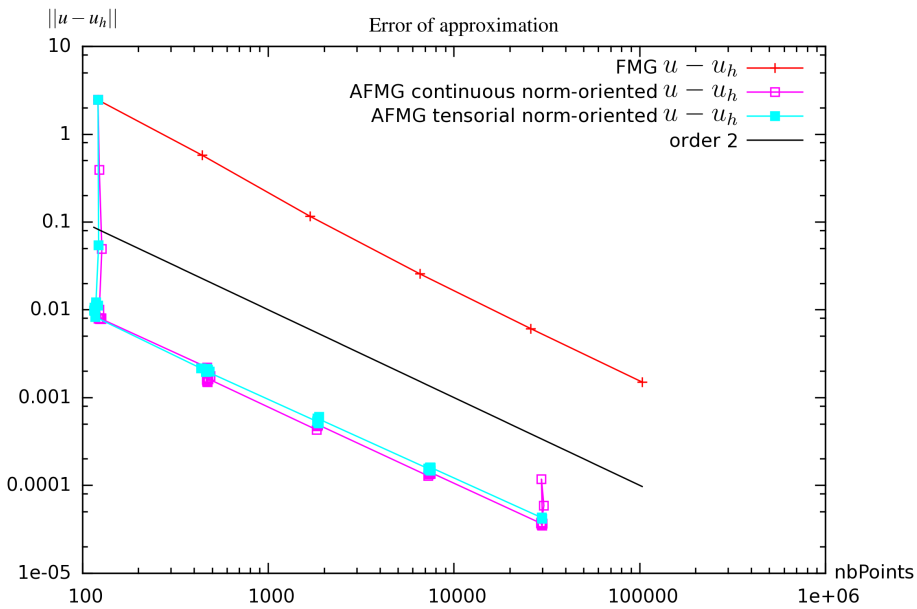

Figure 7.7: 2D boundary layer test case, norm-oriented methods: error convergence in terms of number of vertices.

2. The two Hessian-based cases are very similar and, in the same way, the two norm-oriented cases are very similar too. That assures us our tensorial method is

good, at least for this test case.

7.7.2 Poisson problem with discontinuous coefficient [22]

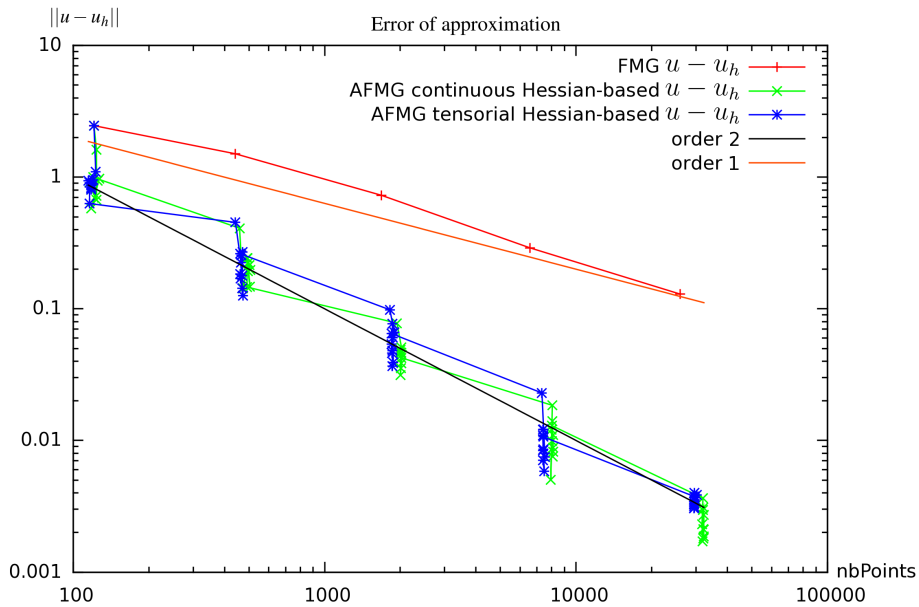

Figure 7.8: Poisson problem with discontinuous coefficient, Hessian-based methods: error convergence in terms of number of vertices.

Results of Figure 7.8 are also good but Figure 7.9 shows results which are a little disappointing. The two Hessian-based cases and the continuous norm-oriented cases present very good results, of order two. Unfortunately, the tensorial norm-oriented case present a result very different, of order one, whereas it should look very much like the continuous norm-oriented. It seems that the discontinuity does not permit to have a good tensorial norm-oriented adaptation or maybe it needs some rectifications.

7.7.3 Bubble-like test case with thin interface

We study now a bubble-like test case with thin interface. Figures 7.10 and 7.11 give us the results. In this case, the tensorial version and the continuous version perform with very similar efficiency.

7.7.4 Bubble-like test case with thick interface

We have also run the easier case with a thicker transition. Figures 7.12 and 7.13 depict the convergence curves.

Like in the previous test case, we observe that the tensorial version and the continuous version are very similar.

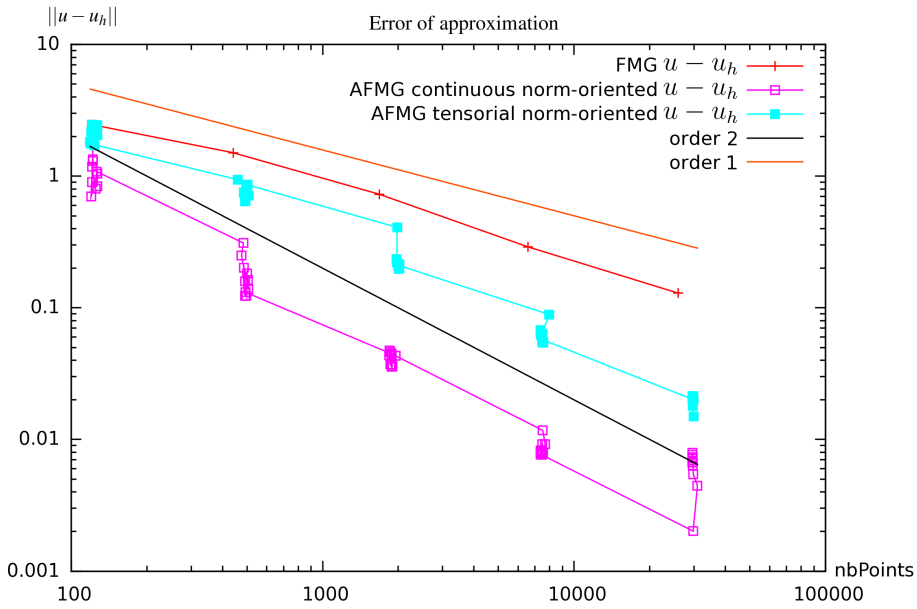

Figure 7.9: Poisson problem with discontinuous coefficient, norm-oriented methods: error convergence in terms of number of vertices.

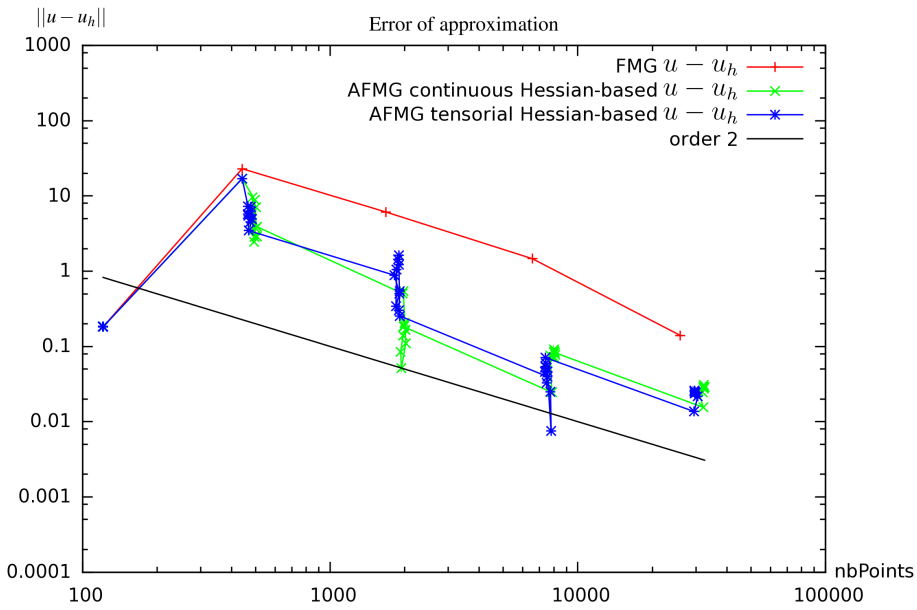

Figure 7.10: Bubble-like test case with thin interface, Hessian-based methods: error convergence in terms of number of vertices.

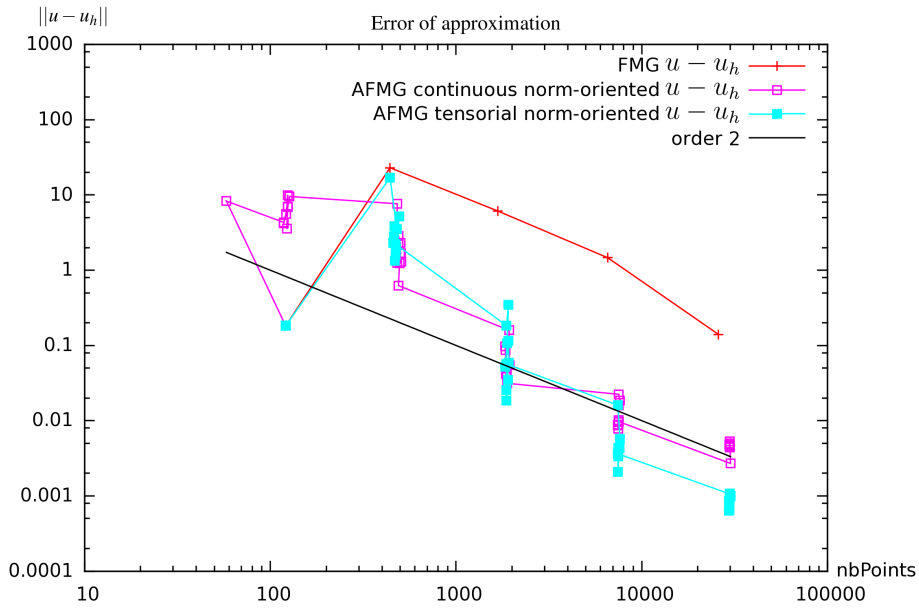

Figure 7.11: Bubble-like test case with thin interface, norm-oriented methods: error convergence in terms of number of vertices.

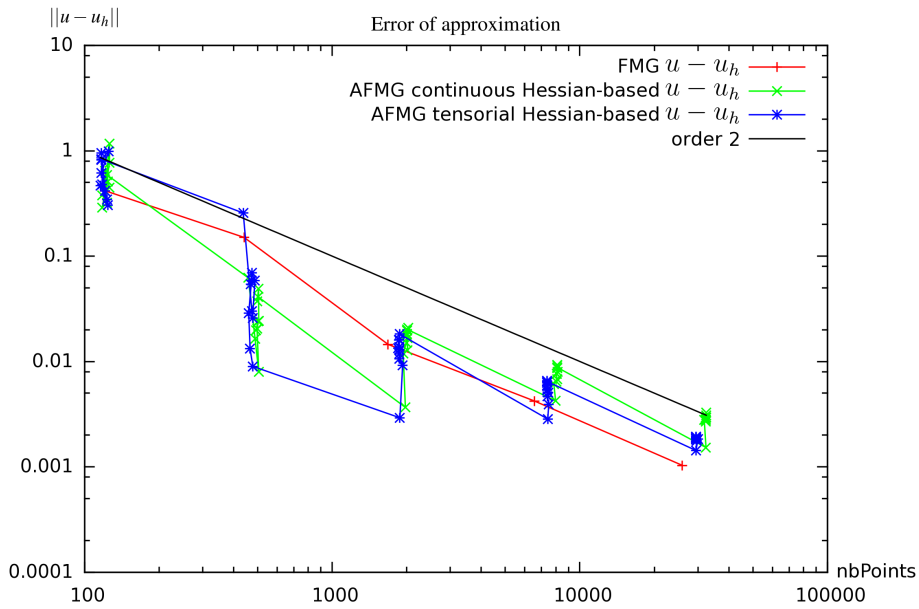

Figure 7.12: Bubble-like test case with thick interface, Hessian-based methods: error convergence in terms of number of vertices.

7.7.5 A 1D boundary layer test case

Figures 7.14 and 7.15 give us the results in the case of the 1D boundary layer. The two Hessian-based results are similar but we can observe an important difference

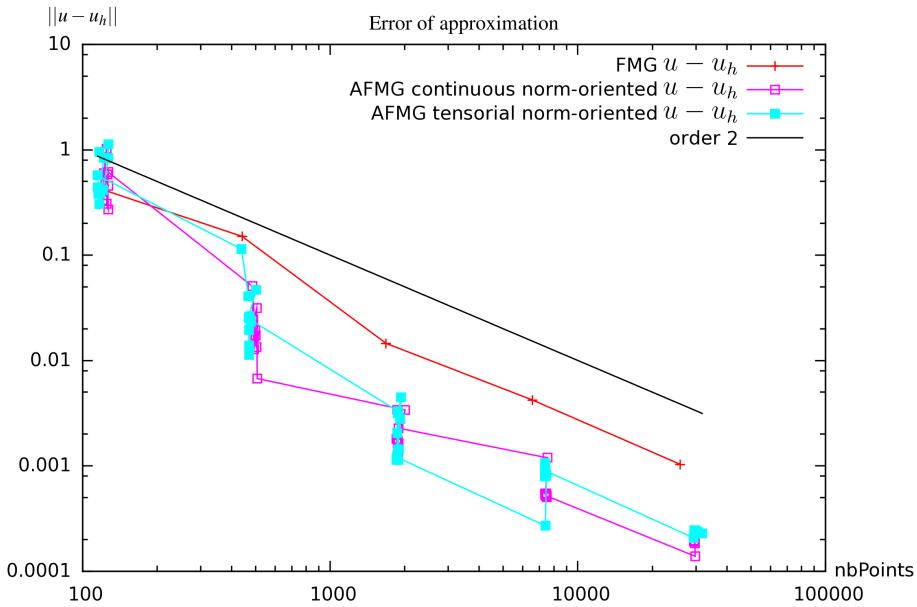

Figure 7.13: Bubble-like test case with thick interface, norm-oriented methods: error convergence in terms of number of vertices.

bewteen the two norm-oriented results. The continuous norm-oriented gives a bad convergence which loses the order two at the end of the computation whereas the tensorial norm-oriented remains of order two. Because of that, the tensorial norm-oriented is better than the continous norm-oriented.

7.8 Conclusion

We have proposed several extensions of the discrete tensorial metric method for the metric-based mesh adaptation of an elliptic model. First, it is formulated in terms of an equation defining an intrinseque optimal metric, giving the optimal adapted mesh as a unit mesh of the optimal metric. Second, the method is extended to the minimization of L^p norms. Third, it is extended to anisotropic goal-oriented mesh adaptation. It is also extended to the novel norm-oriented analysis.

The proposed tensorial approach assumes that the iterated mesh is locally of same edge directions as the background mesh, but this assumption is just a way of reasoning and not a constraint in adaptation. This is illustrated by the fact the tensorial method produces optimality systems which are essentially discretisations of the optimality systems given by the continuous metric method.

We compared the tensorial method with the continuous metric method. In the continuous metric method, discrete fields are theoretically mapped into a continuous one in order to define a continuous optimality system for the metric. In the tensorial treatment of Hessian-based, goal-, and norm-oriented error analysis, no continuous

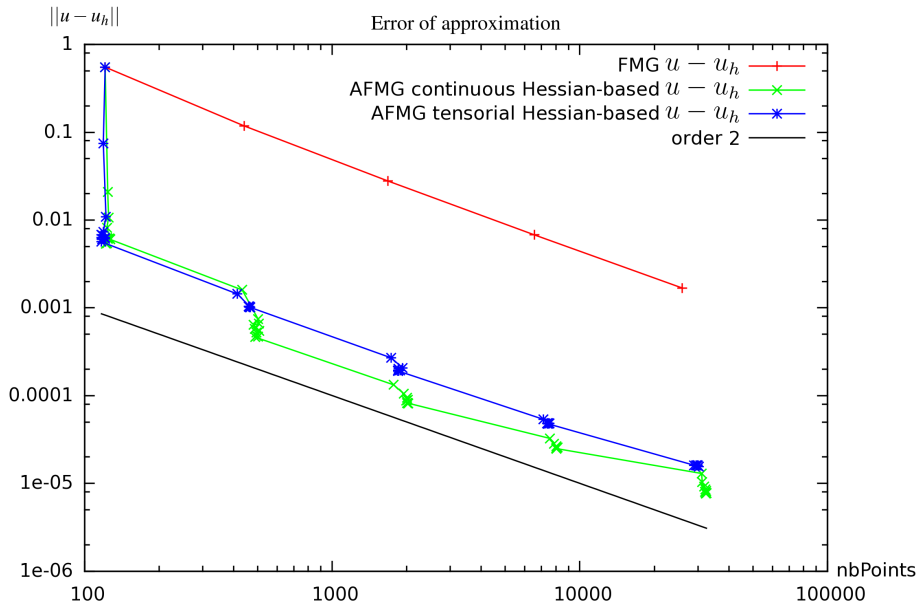

Figure 7.14: 1D boundary layer, Hessian-based methods: error convergence in terms of number of vertices.

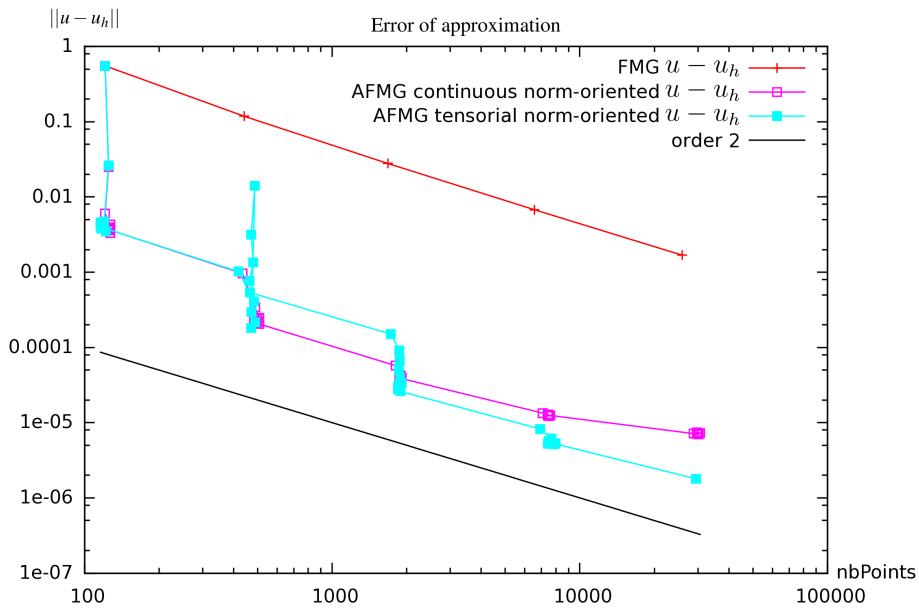

Figure 7.15: 1D boundary layer, norm-oriented methods: error convergence in terms of number of vertices.

context needs to be invoked. Further, the error analysis in the tensorial case does not require any anisotropy bound while the continuous analysis does (at least in

theoretical arguments).

Two-dimensional numerical experiments on a benchmark already used for continuous Hessian-based, goal-, and norm-oriented adaptation show that both continuous approach and tensorial approach behave similarly. The tensorial method appears as less smooth than the continuous one and improvements of this issue need to be applied. These preliminary outputs confirm the interest of the new approach. The comparison will be continued with strongly anisotropic mesh adaptation (shape aspect ratio larger than 100).

This work also proposes a 3D analysis. 3D experiments will soon be produced.

Chapter 8

Application aux écoulements incompressibles

8.1 Introduction

Une des missions de cette thèse est de soutenir le transfert des méthodes de maillage adaptatif développées par l'INRIA vers la société Lemma, co-support de la bourse de thèse. La société Lemma développe notamment dans le logiciel ANANAS des outils de simulation à la fois très innovatifs (adaptation du maillage) et fonctionnant sur des plateformes informatiques les plus modernes (parallélisme massif). Rappelons dans ce contexte quelques réalisations résultant de la collaboration de Lemma avec la recherche.

Dans sa thèse [47], Damien Guégan a travaillé à un démonstrateur de simulation par maillage adaptatif et parallélisme massif d'écoulement diphasiques, avec des critères d'adaptation basés hessiens.

Suite à la thèse de Hubert Alcin à l'INRIA [5, 6], le module Navier-Stokes a été enrichi par un algorithme de résolution de type Schwarz multiniveau par déflation.

Des critères basés adjoints ont été ensuite introduits dans certains modules de structure d'ANANAS.

Une certaine priorité étant mise sur l'efficacité, notre travail de thèse s'est focalisé sur l'adaptation de notre algorithme FMG dans le module Navier-Stokes d'ANANAS. Dans un premier temps, cette implémentation se fait indépendamment de l'algorithme de Schwarz, mais pourrait être combiné avec Schwartz (qui serait appliqué à la grille la plus grossière) pour atteindre un niveau supérieur de scalabilité.

Pour adapter l'algorithme FMG à un module Navier-Stokes parallèle, une difficulté essentielle est la mise au point d'une version parallèle du solveur elliptique en pression sans perte importante d'efficacité. *L'essentiel du travail de mise au point du démonstrateur expérimental présenté dans ce chapitre porte sur la transposition de FMG à un contexte partitionné pour une mise en œuvre avec MPI.*

Dans un premier paragraphe on introduit le modèle Navier-Stokes. Le second paragraphe définit une petite adaptation à l'instationnaire. Le troisième paragraphe

définit les conditions d'adaptation au cas partitionné. Le quatrième paragraphe présente quelques mesures d'efficacité. Un cinquième paragraphe fait la synthèse.

8.2 Modèle Navier-Stokes incompressible instationnaire

Le modèle considéré concerne les écoulement multi-fluides incompressibles. Sa formulation sous forme d'EDP est basée sur une formulation en courbe de niveau (*Level-Set*), comme dans [34, 69] incluse dans un modèle Navier-Stokes parallèle comme dans [52] et étendue au cas adaptatif comme dans [48].

8.2.1 Équations bi-fluides via Level-Set

On considère le modèle des équations de Navier-Stokes pour deux liquides incompressibles non-miscibles, non soumis à une tension interfaciale et se déplaçant dans une enceinte fermée Ω.

$$\rho\frac{\partial \boldsymbol{U}}{\partial t} + \rho\nabla\cdot(\boldsymbol{U}\otimes\boldsymbol{U}) = \nabla\cdot(\nu(\rho)\nabla\boldsymbol{U}) - \nabla p + \rho\boldsymbol{g} \quad \text{dans} \quad \Omega \tag{8.1}$$

$$\nabla\cdot\boldsymbol{U} = 0 \quad \text{dans} \quad \Omega \tag{8.2}$$

$$\frac{\partial \boldsymbol{\rho}}{\partial t} + \nabla\cdot(\rho\boldsymbol{U}) = 0 \quad \text{dans} \quad \Omega \tag{8.3}$$

où $\boldsymbol{U}$ désigne le vecteur vitesse du système des deux fluides, p sa pression, ρ sa masse volumique, et $\boldsymbol{g}$ le vecteur des forces de gravité. Dans cette formulation, la masse volumique prend seulement deux valeurs ρ_l et ρ_g strictement positives chacune et prise chacune dans un sous-domaine de Ω, les deux sous-domaines étant séparés par une interface. Le modèle est mathématiquement bien posé quand la viscosité $\nu(\rho)$ n'est pas nulle, voir par exemple [74]. L'interface est supposée être une surface suffisamment régulière, et en particulier, avoir une normale bien définie $\mathbf{n}$:

$$\mathbf{n}\delta_\rho = \frac{1}{\rho_l - \rho_g}\nabla\rho\,,$$

où δ_ρ désigne la couche de Dirac portée par cette interface. Pour fixer les idées, ρ_l est la masse volumique d'un liquide, et ρ_g est la masse volumique d'un gaz (le mouvement du gaz étant supposé garantir son incompressibilité). Typiquement, le rapport entre les deux masses volumiques peut atteindre un millier, ce qui représente une difficulté certaine au niveau de l'approximation, aussi bien qu'au niveau de la résolution, et ceci explique que nous ayons tout au long de cette étude utilisé un cyclage extrêmement robuste incluant un GMRES. Le système précédent est complété par une condition au bord, de type glissement ($\boldsymbol{U}\cdot\mathbf{n}_{\partial\Omega} = 0$ on $\partial\Omega$ où $\mathbf{n}_{\partial\Omega}$ est la normale au bord).

Dans la méthode Level-Set, on modifie la formulation (8.1-8.3). On considère une fonction lisse ϕ, la Level Set, dont le contour 0 est porté par l'interface. Les

deux sous-domaines sont donc repérés par la fonction de Heaviside H appliquée à ϕ:

$$H(\phi) = \begin{cases} 0 & \text{si } \phi < 0 \\ 1 & \text{si } \phi \geq 0\,. \end{cases}$$

$$\rho(\phi) = \rho_g + (\rho_l - \rho_g)H(\phi)$$

.

$$\phi_t + \nabla \cdot (\boldsymbol{U}\phi) = 0\,, \tag{8.4}$$

$$\rho(\phi)\frac{\partial \boldsymbol{U}}{\partial t} + \rho(\phi)\nabla \cdot (\boldsymbol{U} \otimes \boldsymbol{U}) + \nabla p - \rho(\phi)\boldsymbol{g} = 0 \tag{8.5}$$

$$\nabla \cdot \boldsymbol{U} = 0\,. \tag{8.6}$$

8.2.2 Approximation par éléments finis

Le système (8.5-8.6) est approché par une méthode de projection standard avec une approximation P_1-P_1 en éléments finis continus de degré un comme dans [52]. L'interface est advectée à l'aide de l'équation (8.4) qui est approchée par des volumes finis centrés-sommets [33].

Introduisons quelques notations supplémentaires. Soit Ω_h la discrétisation du domaine de calcul Ω inclus dans $\mathbb{R}^{dim}$ et soit $\mathcal{H}$ un maillage de Ω_h. Pour simplifier nous supposons que le domaine discret Ω_h est identique à Ω. On omet les indices de discrétisation spatiale $_h$. $V = \{\psi \in \mathcal{C}^0(\bar{\Omega}) \mid \psi|_K \text{ is affine } \forall K \in \mathcal{H}\}$ est l'espace usuel d'éléments finis P_1. V est engendré par les fonctions de base ψ_i décrites dans les chapitres précédents. On note $\mathbf{V} = V^{dim}$, Les variables multi-fluides discrètes sont:

$$\mathbf{U} = \sum_i \mathbf{U}_i\psi_i\,, \quad p = \sum_i p_i\psi_i \quad \text{and} \quad \phi = \sum_i \phi_i\psi_i\,.$$

Projecteur L^2: pour tout $u \in \mathbf{L}^2(\Omega)$, on note $\mathcal{P}u : \mathbf{L}^2 \mapsto V$ la fonction telle que pour tout sommet $\mathbf{x}_i$ de $\mathcal{H}$:

$$\mathcal{P}u(\mathbf{x}_i) = \frac{\int_\Omega u\psi_i \,\mathrm{d}\mathbf{x}}{\int_\Omega \psi_i \,\mathrm{d}\mathbf{x}}\,.$$

Pour tout $\mathbf{U} = (u, v) \in (\mathbf{L}^2(\Omega))^2$, on note $\mathcal{P}\mathbf{U} = (\mathcal{P}u, \mathcal{P}v)$ le projecteur dans $\mathbf{V}$. Cet opérateur $\mathcal{P}$ est utilisé pour transformer un champ discret constant par élément en un champ discret continu et affine par élément.

L'algorithme global d'avancement en temps s'écrit:

Étape 1: (Étape de prédiction) La vitesse est explicitement prédite via l'équation (8.5) en omettant le terme de pression:

$$\bar{\mathbf{U}}_i = \mathbf{U}_i^n - \frac{\Delta t}{|C_i|}\int_\Omega \psi_i\,(\nabla.(\mathbf{U} \otimes \mathbf{U}) - \mathbf{g})\,\mathrm{d}\mathbf{x}\,,$$

où $|C_i|$ est le volume de la cellule associée au sommet i:

$$|C_i| \;=\; \sum_j \int_\Omega \psi_i \psi_j \, \mathrm{d}\mathbf{x} \; .$$

Étape 2: (Étape de projection) Cette étape évalue la pression en utilisant la prédiction de la vitesse et met à jour la vitesse en imposant une contraint approchée de divergence nulle (8.6). A cet effet, on résout

$$\int \frac{1}{\rho} \nabla p^{n+1} . \nabla \psi \, \mathrm{d}\mathbf{x} \;=\; \frac{1}{\Delta t} \int \nabla \psi . \bar{\mathbf{U}} \, \mathrm{d}\mathbf{x} \quad \forall \; \psi \in V \, ,$$

dans V par éléments finis P_1. La correction de vitesse basée sur le gradient de pression est constante par élément. Elle est transférée dans $\mathbf{V}$ à l'aide de $\mathcal{P}$ et ajoutée à la vitesse :

$$\mathbf{U}^{n+1} = \bar{\mathbf{U}} + \Delta t \, \mathcal{P} \left(\frac{1}{\rho} \nabla p^{n+1} \right) \quad \text{et} \quad \mathbf{U}^{n+1} = 0 \;\; \text{sur } \partial\Omega \, .$$

Étape 3: (Advection de la Level-Set) La fonction ϕ est advectée à l'aide de l'équation (8.4) avec $\mathbf{U}^n$ comme vitesse pour passer du niveau en temps n au niveau en temps $n+1$. Cette équation est approchée par des volumes finis centrés-sommet précis au second ordre. De manière classique, afin d'éviter sur la Level-Set la formation de très forts (ou trop faibles) gradients, la Level-Set est remplacée régulièrement par une fonction $\tilde{\phi}^{n+1}$ définie comme la distance signée au contour $\{\phi^{n+1} = 0\}$. Nous renvoyons à [52] pour plus de détails.

Épaissement de l'interface. Pour de grands rapports de masse volumique de chaque cotè de l'interface, la manipulation de fonctions Heaviside $H(\phi)$ strictement discontinues pour définir la masse volumique conduit à des instabilités de l'approximation et quelquefois à des divergences du calcul. Dans l'équation (8.5), $H(\phi)$ est remplacée par une version "épaissie" $H_\eta(\phi)$ définie par :

$$H_\eta(\phi) = \begin{cases} 0 & \text{if } \; \phi < -\eta, \\ \frac{1}{2}(1 + \frac{\phi}{\eta} + \frac{1}{\pi}\sin(\frac{\pi\phi}{\eta})) & \text{if } |\phi| \leq \eta, \\ 1 & \text{if } \; \phi > \eta. \end{cases}$$

En pratique, le paramètre η est choisi de manière que $\eta = k \; \Delta x$ avec $k \approx 3$. Cet épaississement d'interface renforce comme attendu la stabilité globale mais introduit un paramètre de discrétisation supplémentaire à prendre en compte. On notera que dans les chapitres précédents, nous avons pris en compte cet épaississement d'interface dans nos cas tests.

Quelques remarques. Dans le modèle numérique présenté, l'avancement en temps peut se faire soit en semi-implicite en ne résolvant que l'équation pression, soit en implicite faiblement couplé en résolvant chaque étape de manière implicite, soit

encore en implicite couplé en ré-itérant à chaque pas de temps les étapes précédentes comme dans les algorithmes de type SIMPLE [70]. On s'intéresse ici à la première option, semi-implicite et on se pose la question des modalités de résolution par FMG du système dont la pression est la solution.

8.3 Une idée globale du module adaptatif

Dans l'algorithme de point fixe local instationnaire utilisé dans [48], l'intervalle en temps est préalablement divisé en un certain nombre de *sous-intervalles*:

$$[0,T] = [0,t_1] \cup [t_1,t_2] \cup ... \cup [t_k,t_{k+1}] \cup ... \cup [t_{nsi-1},t_{nsi}].$$

La simulation calcule chaque sous-intervalle l'un après l'autre.

Dans chaque sous-intervalle $[t_k,t_{k+1}]$**:**
- L'avancement en temps se fait à maillage spatial fixe. Mais ce maillage est adapté par point fixe, c'est-à-dire que dans chaque sous-intervalle, l'avancement en temps est ré-itéré entre deux mises à jour du maillage (sur la base d'une nouvelle métrique dépendant de l'avancement en temps précédent).

- (a) Dans la nouvelle version à laquelle cette thèse contribue, la génération d'un nouveau maillage adapté (toujours pour $[t_k,t_{k+1}]$) *est suivie de la génération de maillages plus grossiers en utilisant des métriques plus grossières.*

- (b) Les différents niveaux sont partitionnés en commençant par le niveau fin et en calquant la partition d'un niveau grossier sur celle du niveau directement plus fin.

- (c) L'avancement en temps est réalisé. *On résoud la projection en pression par FMG/FMV en utilisant les niveaux construits.*

- Les étapes (a)(b)(c) sont ré-itérées jusqu'à un couplage solution/maillage satisfaisant.

8.4 Vers MG en instationnaire : FMV

8.4.1 Motivations

Nous présentons une courte étude d'une modification assez simple de notre algorithme pour une adaptation à l'instationnaire. Rappelons le contexte de FMG stationnaire. le but est d'obtenir une solution assez précise en partant de grilles grossières en raffinant et en adaptant, et en s'arrêtant de raffiner quand le maillage et sa solution sont satisfaisants. Dans la plate-forme ANANAS, la résolution adaptative instationnaire se fait sur un intervalle en temps durant lequel

le maillage est figé, cf. [48]. On dispose donc des maillages et des inconnues au temps précédent, et on conserve le même maillage pour le niveau de temps à calculer.

On notera que l'EDP à résoudre n'est pas de type Cauchy-Kowalevska pour la pression, en d'autres termes que la pression ne sera pas avancée en temps sur la base d'une dérivée en temps de la pression. Il est néanmoins possible d'extrapoler à partir des instants précédents une condition d'initialisation de la pression pour l'algorithme itératif de l'étape de projection. Cette option n'est pas très robuste et n'a pas été envisagée dans ce travail. On utilisera seulement l'existence du maillage fin en transformant le FMG en un cyclage proche des cycles FMV. Bien que FMV désigne une méthode de cyclage, nous conservons cette appellation pour le nouveau solveur proposé.

8.4.2 Description de la méthode

Notre méthode de résolution FMV est très semblable à la méthode FMG. La seule différence réside dans l'assemblage du second membre de l'équation à résoudre.
Dans le cas de FMG, à chaque phase, on assemble le second membre sur le maillage courant, c'est-à-dire le maillage le plus fin de la phase.
On a l'équation

$$A_{\mathcal{M}} v = f_{\mathcal{M}}.$$

Dans le cas de FMV, à chaque phase, on assemble le second membre non pas sur le maillage courant mais sur le maillage le plus fin. Puis, on le transfère sur le maillage courant par accumulation. On résout donc

$$A_{\mathcal{M}} v = R_{\mathcal{M}_{fin} \to \mathcal{M}} f_{\mathcal{M}_{fin}}.$$

Pour être plus clair, on compare les seconds membres pour le cas test du cercle.

8.4.3 Exemple avec le cas test du cercle

On considère une suite de maillages uniformes, les mêmes que ceux utilisés dans les exemples en FMG. Le maillage le plus grossier a 121 points et le plus fin en a 103041. Ces deux maillages sont montrés par les figures 8.1 et 8.2.

On remarque que le maillage à 103041 points est tellement fin qu'on ne peut même pas voir ses mailles. Un zoom sur ce maillage est donné par la figure 8.3.

Le second membre $f_{\mathcal{M}}$ assemblé directement sur le maillage grossier est montré par la figure 8.4.

La figure 8.5 montre, au contraire, le second membre $f_{\mathcal{M}_{fin}}$ assemblé sur le maillage fin.

On constate que les deux seconds membres $f_{\mathcal{M}}$ et $f_{\mathcal{M}_{fin}}$ sont très différents, $f_{\mathcal{M}_{fin}}$ étant nécessairement une meilleure représentation du second membre analytique f.
Reste à transférer $f_{\mathcal{M}_{fin}}$ sur le maillage grossier. Une simple accumulation nous donne $R_{\mathcal{M}_{fin} \to \mathcal{M}} f_{\mathcal{M}_{fin}}$, représenté sur la figure 8.6

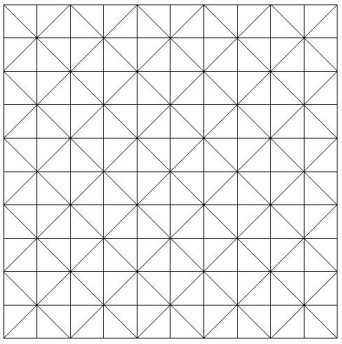

Figure 8.1: Maillage uniforme 121 points

Figure 8.2: Maillage uniforme 103041 points

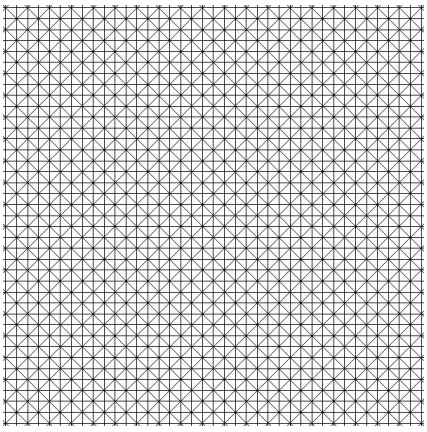

Figure 8.3: Zoom maillage uniforme 103041 points

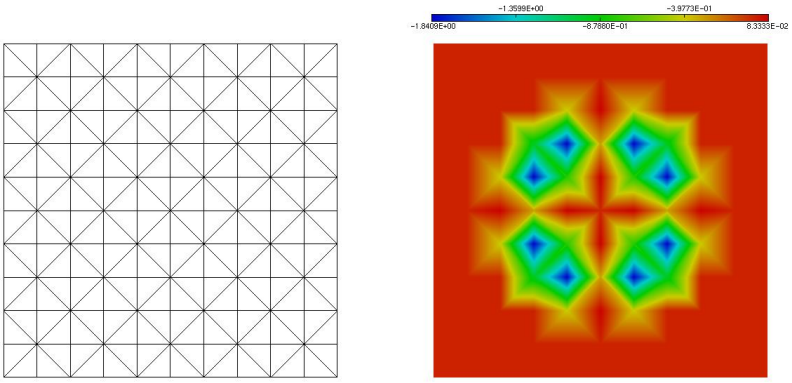

Figure 8.4: Second membre assemblage grossier

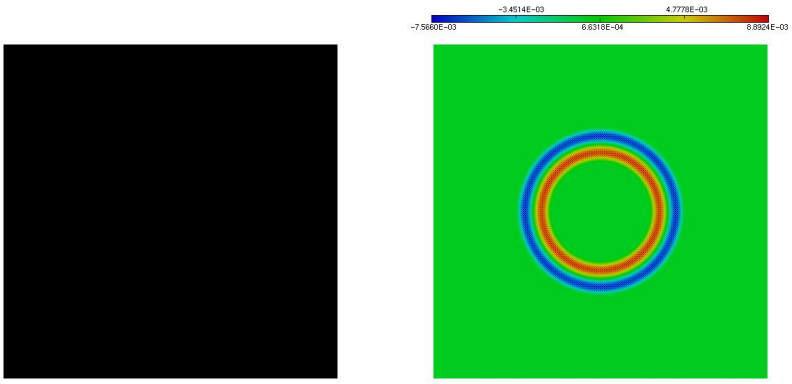

Figure 8.5: Second membre assemblage grossier

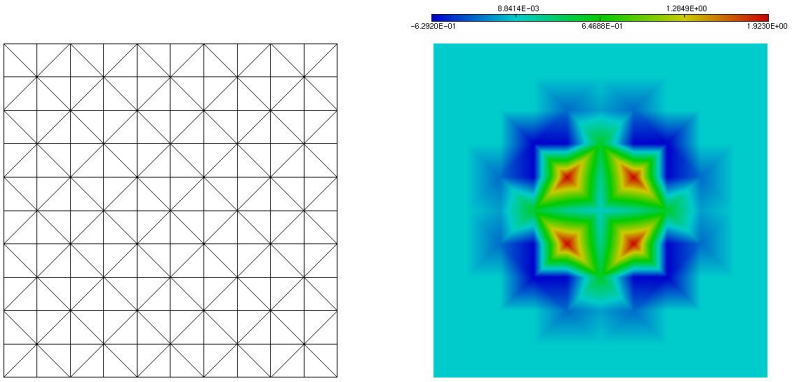

Figure 8.6: Second membre assemblage fin transféré sur grossier

On observe une grande différence avec $f_{\mathcal{M}}$. Ce nouveau second membre ressemble beaucoup plus à $f_{\mathcal{M}_{fin}}$.
A partir de là, on peut lancer notre résolution de la même façon que pour le FMG avec, à chaque phase, un second membre transféré depuis le maillage à 103041 points. Notons qu'au fur et à mesure que l'on avance dans la résolution et qu'on passe d'une phase à la suivante, le second membre transféré ressemble de plus en plus au second membre assemblé sur le maillage courant, jusqu'à ce qu'on arrive à la dernière phase où le maillage courant est lui-même le maillage le maillage à 103041 points.

Les courbes de convergence des méthodes FMG et FMV sont données par les figures 8.7 et 8.8 qui représentent l'erreur, respectivement en fonction du nombre de points et du temps CPU.

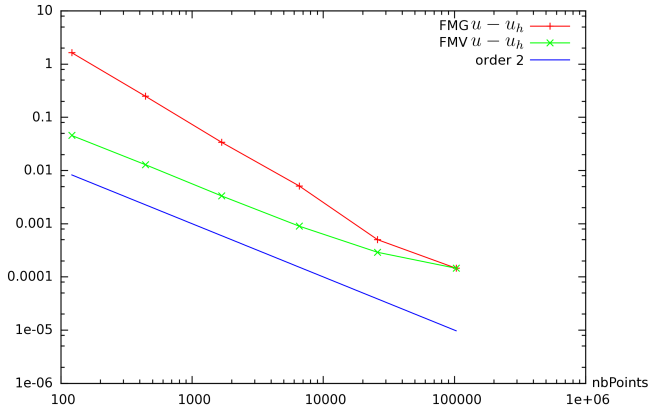

Figure 8.7: Convergence en nombre de points du FMG et du FMV

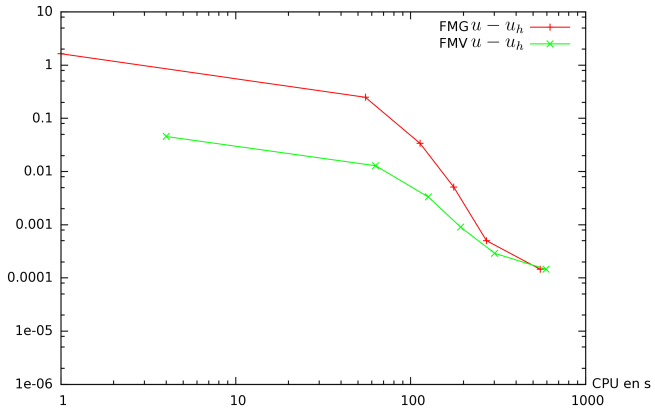

Figure 8.8: Convergence en temps CPU du FMG et du FMV

On observe, sur la figure 8.7, que les résultats, au début de la résolution, sont bien

meilleurs en FMV qu'en FMG mais que, plus on avance dans les phases, plus les deux courbes se rapprochent, l'avantage du FMV sur le FMG diminuant, pour finalement arriver à des résultats identiques à la dernière phase. Ceci est parfaitement logique puisque, au fur et à mesure qu'on utilise des maillages de plus en plus fins, le second membre assemblé dessus est de plus en plus proche du second membre assemblé sur le maillage à 103041 points. Finalement, à la dernière phase, le maillage utilisé pour la résolution est le maillage à 103041 points lui-même ce qui implique que, lors de cette phase, le même second membre est employé dans les deux méthodes ce qui amène très logiquement à des résultats identiques. Notons toutefois que ces résultats ne sont identiques que parce que, dans nos expériences, la résolution multigrille est convergée sur la dernière phase (tout comme elle l'est aussi sur les autres phases d'ailleurs) car, sans cette convergence complète, les résultats seraient différents du fait de la condition initiale, transférée depuis l'avant-dernière phase et donc différente entre les deux méthodes. Or, en pratique, seulement quelques cycles seront appliqués à chaque phase.
La figure 8.8 nous montre le temps de calcul pour les deux méthodes. On remarque que, pour la première phase, le calcul ne prend qu'une seconde en FMG alors qu'il en prend quatre en FMV. Cela n'a rien d'étonnant puisque l'assemblage du second membre sur un maillage plus fin prend nécessairement plus de temps et qu'il faut, de plus, le transférer sur le maillage grossier. Cette différence se fait aussi sentir à la dernière phase où on observe des résultats égaux mais décalés en temps CPU: il faut 550 secondes pour le FMG et 594 pour le FMV. Cependant, pour toutes les autres phases, le rendement est meilleur pour le FMV que pour le FMG. Notamment, si on arrête la résolution à l'avant-dernière phase au lieu de la dernière, on a un gain en efficacité par rapport au FMG.

8.5 MG en MPI

Dans le cadre de cette thèse, la version FMG MPI a été développée en 2D et 3D et validée sur poste de travail biprocesseur. La méthode s'applique au cas non-adaptatif aussi bien qu'au cas adaptatif. Nous résumons la description des principaux modules/algorithmes qui ont été nécessaires à ce développement:

- Génération de grilles grossières si elles ne sont pas disponibles au lancement de la phase de résolution de l'étape de projection.

- Partition des grilles grossières calquée sur la partition de la grille fine.

- Construction des opérateurs de transfert inter-grille parallèles.

8.6 Un cas test de validation/évaluation en 3D

Le nouvel algorithm a été validé en 2D et 3D. Nous donnons un exemple de validation 3D. Pour cela, on a utilisé le maillage d'un cube montré en Figure 8.9.

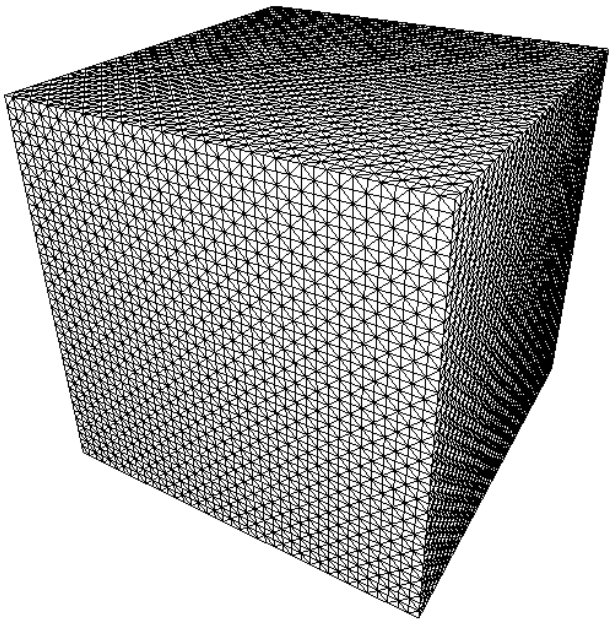

Figure 8.9: Maillage d'un cube en trois dimensions

La fonction u à approcher est définie par la formule suivante:

$$u = \alpha x(x-1)$$

avec, dans le cas présent, $\alpha = 1000000000$.
La résolution est faite en FMG uniforme avec quatre grilles ayant respectivement 216, 1331, 9261 et 68921 points.

On effectue 100 itérations à chaque phase. Cela nous permet d'obtenir une solution convergée. En effet, on a tracé la convergence du résidu sur chacune des quatre grilles et on observe qu'il y a bien convergence. Ces quatres courbes de convergence sont données par les Figures 8.10, 8.11, 8.12 et 8.13.

On peut voir une grande différence entre la courbe de convergence de la première phase et les suivantes avec une brusque descente de la courbe à la septième itération au lieu d'une descente progressive. Cela est dû au fait que, lors de la première phase, on effectue un GMRES avec un simple préconditionnement Jacobi alors qu'ensuite, le préconditionnement se fait avec des multigrilles. On observe également que la convergence ne ralentit pas d'une phase à l'autre, conséquence de l'utilisation des

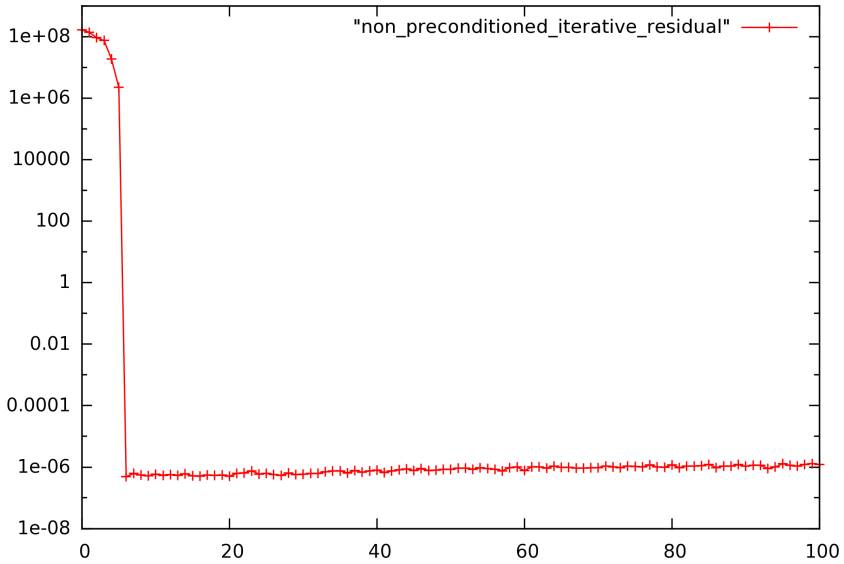

Figure 8.10: Convergence du résidu première phase

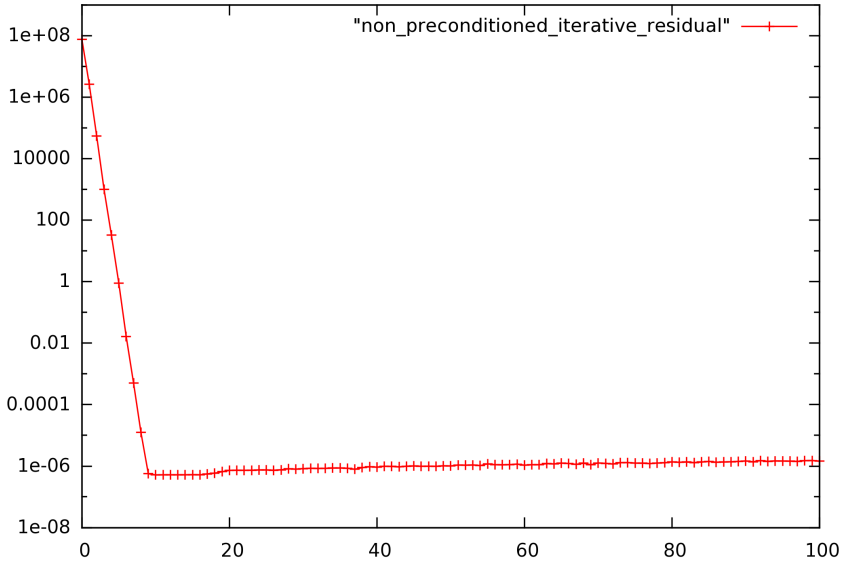

Figure 8.11: Convergence du résidu deuxième phase

multigrilles. La figure 8.14 montre la solution calculée. Les résultats sont identiques en calcul MPI sur deux processeurs. Les maillages de partition et les solutions correspondantes sont donnés par la figure 8.15.

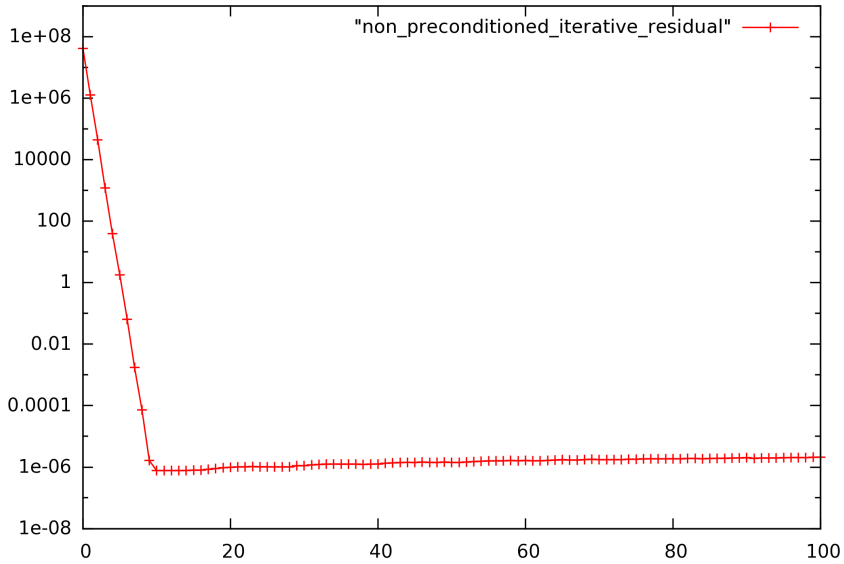

Figure 8.12: Convergence du résidu troisième phase

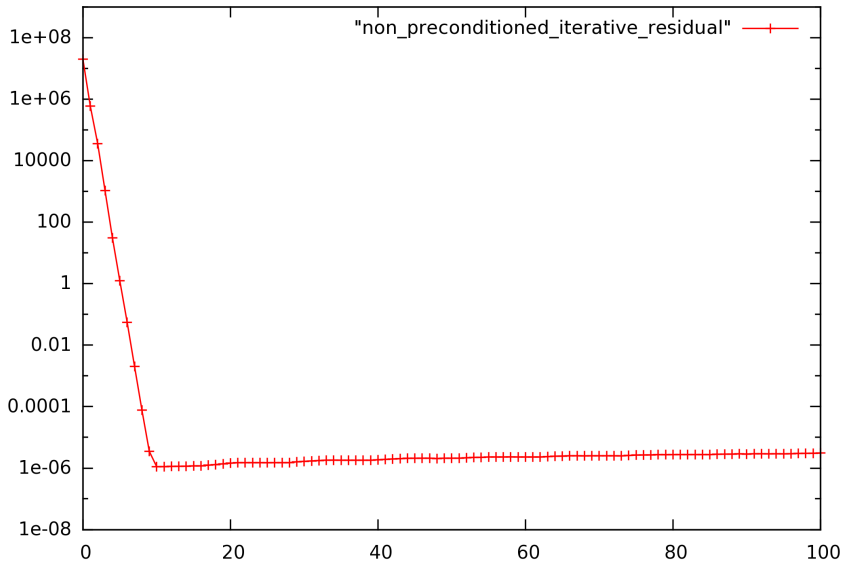

Figure 8.13: Convergence du résidu quatrième phase

Conclusion du chapitre

Nous avons résumé la description du transfert vers l'industrie de résultats de recherche effectués dans le cadre de cette thèse. Ce transfert reste à consolider, puis à tester sur des cas massivement parallèles. Une des premières applications sera la simulation de grandes structures dans un écoulement diphasique turbulent.

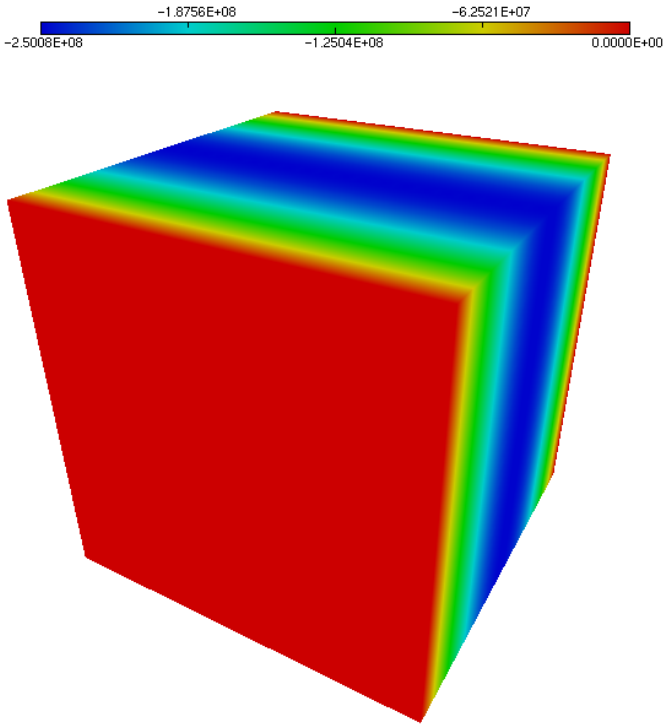

Figure 8.14: Solution calculée du cas en trois dimensions

Le nombre de points des maillages utilisés est de plusieurs dizaines de millions de sommets et dans ces niveaux de nombre de points les méthodes FMG apporteront un important progrès en efficacité.

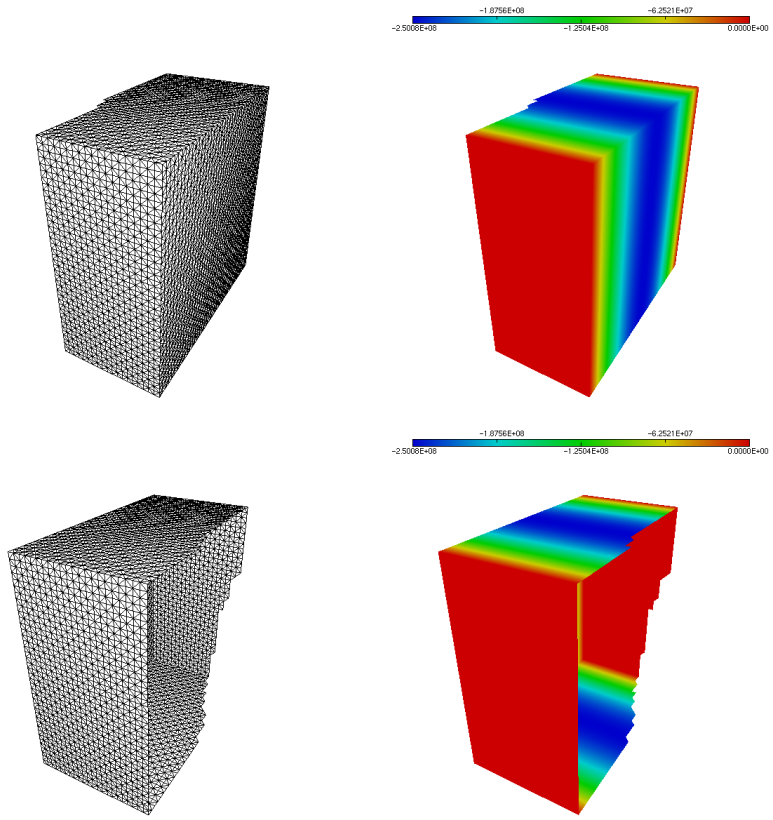

Figure 8.15: Cas 3D en MPI

Conclusion générale

L'étude présentée dans cette thèse s'est attaquée à la recherche d'une meilleure efficacité du calcul de phénomènes modélisés par des EDP elliptiques stationnaires multi-échelles. Par multi-échelles nous entendons que le champ de solution présente des grandes et petites échelles. Un exemple typique un peu extrême est le cas de champs discontinus. La situation physique cible est par exemple la modélisation d'un milieu fluide diphasique avec interface discontinue.

Les quatre ingrédients principaux de la simulation efficace de tels phénomènes sont les suivants:

(a)- la précision asymptotique formelle de l'approximation choisie ainsi que sa complexité (nombre d'opérations par inconnue),

(b)- l'adaptation de maillage afin d'obtenir le plus de précision possible pour un nombre de nœuds donné, et en particulier une convergence à l'ordre asymptotique formel.

(c)- la mise en œuvre d'algorithmes de résolution du système discret les plus efficaces possibles et en particulier de complexité linéaire (O(N)),

(d)- la mise en œuvre massivement parallèle du code de simulation complet avec le meilleur *speed up* et en particulier présentant des propriétés de scalabilité.

L'analyse de tous ces ingrédients doit permettre de relier asymptotiquement la précision recherchée avec les moyens mis en œuvre: nombre de processeurs, temps horloge.

Cette thèse est partie d'une approximation formellement à l'ordre deux (point (a)). Elle a cherché à contribuer aux trois autres points par les avancées suivantes:

- nous avons proposé une méthode MG efficace pour des discontinuités, controllée par un critère d'arrêt (point (c)),

- nous avons proposé des critères d'adaptation de maillage (point (b)) basés Hessien, orientés fonctionnelles, orientés norme, en version métrique continue et en version métrique tensorielle,

- nous avons développé une version parallèle MPI de ces algorithmes, notamment de la méthode FMG (point (d)).

Les méthodes développées apportent des réponses à plusieurs questions posées

dans le programme ANR MAIDESC. Elles ont trouvé divers débouchés, avec notamment des méthodes orientées normes pour la Mécanique des Fluides Compressibles mises au point par l'équipe INRIA GAMMA de Rocquencourt et constituant une des livraisons du projet Européen UMRIDA. Ces méthodes s'appliqueront à terme aux modèles CFD turbulents 3D.

Les méthodes développées donnent lieu à des transferts vers les activités industrielles, notamment à travers des développements informatiques dans le code ANANAS de la société Lemma.

De nombreuses extensions sont à envisager:

- nous avons signalé l'extension au 3D.
- l'adaptation à des phénomènes instationnaires (dans la suite de la thèse de Anca Belme) est un problème ouvert et représente un besoin industriel urgent.
- l'extension à des approximations d'ordre trois et plus, dans la perspective ouverte par la thèse d'Alexandre Carabias.
- l'extension à des couplages physiques (fluide-structure par exemple).

Tous ces développements augmenteront les capacités de la simulation numérique avec pour corollaire une meilleure maitrise par l'homme des processus naturels qui l'entourent et des dispositifs qu'il crée.

Bibliography

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008. (Cited on page 123.)

[2] A. Agouzal, K. Lipnikov, and Y. Vasilevskii. Adaptive generation of quasi-optimal tetrahedral meshes. *East-West Journal*, 7(4):223–244, 1999. (Cited on pages 90, 96, 119 and 124.)

[3] F. Alauzet. *Adaptation de maillage anisotrope en trois dimensions. Application aux simulations instationnaires en Mécanique des Fluides*. PhD thesis, Université Montpellier II, Montpellier, France, 2003. (in French). (Cited on pages 90 and 119.)

[4] F. Alauzet and A. Loseille. High order sonic boom modeling by adaptive methods. *J. Comp. Phys.*, 229:561–593, 2010. (Cited on pages 52, 78 and 123.)

[5] H. Alcin. *Algorithmes de résolution d'écoulements instationnaires et adjoints*. PhD thesis, Université de Nice, Nice, France, 2012. (Cited on page 147.)

[6] H. Alcin, B. Koobus, O. Allain, and A. Dervieux. Efficiency and scalability of a two-level schwarz algorithm for incompressible and compressible flows. *International Journal for Numerical Methods in Fluids*, 72(1):69 89, 2013. (Cited on page 147.)

[7] E. Arian and M.D. Salas. Admitting the inadmissible: Adjoint formulation for incomplete cost functionals in aerodynamic optimization. *AIAA Journal*, 37(1):37–44, 1999. (Cited on pages 90 and 91.)

[8] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magn. Reson. Med.*, 56(2):411–421, 2006. (Cited on page 123.)

[9] J. P. Aubin. Behaviour of the error of the approximate solution of boundary value problems for linear elliptic operators by Galerkin's and finite difference methods. *Ann. Scuola Norm. Sup. Pisa*, 21:599–637, 1967. (Cited on page 19.)

[10] N. S. Bakhvalov. On the convergence of a relaxation method with natural constraints on the elliptic operator. *USSR Computational Mathematics and Mathematical Physics*, 6(5):101–135, 1966. (Cited on page 27.)

[11] R.E. Bank. PLTMG: A software package for solving Elliptic Partial Differential Equations. Users' Guide 11.0. Technical report, Department of Mathematics, University of California at San Diego, 2012. (Cited on pages 4 and 50.)

[12] 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. (Cited on pages 20, 90 and 120.)

[13] A. Belme. *Aérodynamique instationnaire et méthode adjointe.* PhD thesis, Université de Nice Sophia Antipolis, Sophia Antipolis, France, 2011. (in French). (Cited on pages 82, 97, 130 and 132.)

[14] A. Belme, A. Dervieux, and F. Alauzet. A fully anisotropic goal-oriented mesh adaptation for unsteady flows. In *Proceedings of the 5th ECCOMAS CFD Conf.*, 2010. (Cited on pages 91 and 120.)

[15] A. Belme, A. Dervieux, and F. Alauzet. Time accurate anisotropic goal-oriented mesh adaptation for unsteady flows. *J. Comp. Phys.*, 231(19):6323–6348, 2012. (Cited on pages 83, 99, 101 and 123.)

[16] M. Berger. *A panoramic view of Riemannian geometry.* Springer Verlag, Berlin, 2003. (Cited on pages 51, 95 and 123.)

[17] M.E. Braaten and S.D. Connell. Three-dimensional unstructured adaptive multigrid scheme for the Navier-Stokes equations. *AIAA Journal*, 34:281–290, 1996. (Cited on pages 4 and 50.)

[18] A. Brandt. Multigrid guide with applications to fluid dynamics. Technical report, GMD Studien, 85, 1984. (Cited on page 27.)

[19] A. Brandt, S. F. McCormick, and J. W. Ruge. Algebraic Multigrid for automatic multigrid solutions with application to geodetic computations, Technical Report, Institute for Computational Studies, Fort Collins, Colorado, october 1982. (Cited on pages 4 and 50.)

[20] G. Brèthes. *Algorithmes multigrilles adaptatifs et scalables.* PhD thesis, Université de Nice, 2015. (Cited on pages 132 and 133.)

[21] G. Brèthes, O. Allain, and A. Dervieux. A mesh-adaptative metric-based Full-Multigrid for the Poisson problem. *I.J. Numer. Meth. Fluids*, 79(1), 2015. pre-print: http://www-sop.inria.fr/members/Gautier.Brethes/article-ADA-MG.pdf. (Cited on pages 101 and 112.)

[22] G. Brèthes and A. Dervieux. Anisotropic norm-oriented mesh adaptation for a poisson problem. Article soumis et chapitre 6 de cette thèse. (Cited on pages vii, 136 and 140.)

[23] G. Brèthes and A. Dervieux. Anisotropic Norm-Oriented mesh adaptation for a Poisson problem. *Submitted for publication*, 2015. pre-print: http://www-sop.inria.fr/members/Gautier.Brethes/article-NO.pdf. (Cited on pages 120, 130 and 132.)

[24] G. Carré and A. Dervieux. On the application of fmg to variational approximation of flow problems. *International Journal for Computational Fluid Dynamics*, 12:99–117, 1999. (Cited on pages 4, 36 and 50.)

[25] M.J. Castro-Díaz, F. Hecht, B. Mohammadi, and O. Pironneau. Anisotropic unstructured mesh adaptation for flow simulations. *Int. J. Numer. Meth. Fluids*, 25:475–491, 1997. (Cited on pages 90, 96, 119 and 124.)

[26] J. Céa. Approximation variationnelle des probls aux limites. *Annales de l'institut Fourier*, 14:345–444, 1964. (Cited on page 19.)

[27] L. Chen, P. Sun, and J. Xu. Optimal anisotropic meshes for minimizing interpolation errors in L^p-norm. *Math. Comp.*, 76(257):179–204, 2007. (Cited on pages 90 and 119.)

[28] Ph. Clément. Approximation by finite element functions using local regularization. *Revue Française d'Automatique, Informatique et Recherche Opérationnelle*, R-2:77–84, 1975. (Cited on page 20.)

[29] Y. Coudière, B. Palmerio, A. Dervieux, and D. Leservoisier. Accuracy barriers in mesh adaptation. RR-4528, INRIA, aug 2002. (Cited on page 71.)

[30] T. Coupez. Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing. *J. Comp. Phys.*, 230:2391–2405, 2011. (Cited on pages 119 and 126.)

[31] T. Coupez, G. Jannoun, N. Nassif, H.C. Nguyen, H. Digonnet, and E. Hachem. Adaptive time-step with anisotropic meshing for incompressible flows. *J. Comp. Phys.*, 241:195–211, 2013. (Cited on pages 119 and 126.)

[32] F. Courty, D. Leservoisier, P.L. George, and A. Dervieux. Continuous metrics and mesh adaptation. *Appl. Numer. Math.*, 56(2):117–145, 2006. (Cited on pages 68, 70 and 112.)

[33] C. Debiez and A. Dervieux. Mixed-Element-Volume MUSCL methods with weak viscosity for steady and unsteady flow calculations. *Comput. & Fluids*, 29:89–118, 2000. (Cited on page 149.)

[34] A. Dervieux and F. Thomasset. Multifluid incompressible flows by a finite element method. *Lecture Notes in Physics*, 11:158–163, 1981. (Cited on page 148.)

[35] J. Dompierre, M.G. Vallet, M. Fortin, Y. Bourgault, and W.G. Habashi. Anisotropic mesh adaptation: towards a solver and user independent CFD. In *AIAA 35th Aerospace Sciences Meeting and Exhibit*, AIAA-1997-0861, Reno, NV, USA, Jan 1997. (Cited on pages 90, 119 and 124.)

[36] A. Ern and M. Vohralík. Polynomial-degree-robust a posteriori estimates in a unified setting for conforming, nonconforming, discontinuous Galerkin, and

mixed discretizations. In *La Serena Numerica II, Octavo Encuentro de Anàlisis Numérico de Ecuaciones Diferenciales Parciales*, Enero 14 - 16, 2015. HAL Preprint 00921583, (2014). (Cited on page 90.)

[37] R. P. Fedorenko. The speed of convergence of one iterative process. *USSR Computational Mathematics and Mathematical Physics*, 4(3):227–235, 1964. (Cited on page 27.)

[38] L. Formaggia and S. Perotto. Anisotropic a priori error estimates for elliptic problems. *Numer. Math.*, 94:67–92, 2003. (Cited on pages 90 and 102.)

[39] J. Francescatto. *Modes multigrilles par agglomtion directionnelle pour le calcul d'ulements turbulents.* PhD thesis, Université de Nice - Sophia Antipolis, Nice, France, 1998. (in French). (Cited on pages 4, 50 and 54.)

[40] P.J. Frey. About surface remeshing. In *Proceedings of the 9th International Meshing Roundtable*, pages 123–136, New Orleans, LO, USA, 2000. (Cited on page 52.)

[41] P.J. Frey. `Yams`, a fully automatic adaptive isotropic surface remeshing procedure. RT-0252, INRIA, November 2001. (Cited on page 52.)

[42] P.J. Frey and P.L. George. *Mesh generation. Application to finite elements.* ISTE Ltd and John Wiley & Sons, 2nd edition, 2008. (Cited on pages 4 and 51.)

[43] M.B. Giles. On adjoint equations for error analysis and optimal grid adaptation in CFD. Technical Report NA-97/11, Oxford, 1997. (Cited on page 21.)

[44] M.B. Giles and N.A. Pierce. Improved lift and drag estimates using adjoint Euler equations. *AIAA paper*, 99-3293, 1999. (Cited on pages 21, 90 and 93.)

[45] M.B. Giles and E. Suli. *Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality*, pages 145–236. Cambridge University Press, 2002. (Cited on page 21.)

[46] C. Gruau and T. Coupez. 3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric. *Comput. Methods Appl. Mech. Engrg.*, 194(48-49):4951–4976, 2005. (Cited on page 119.)

[47] D. Guégan. *Modélisation numérique d'écoulements bifluides 3D instationnaires avec adaptation de maillage.* PhD thesis, Université de Nice Sophia Antipolis, Sophia Antipolis, France, 2007. (in French). (Cited on pages 1 and 147.)

[48] D. Guégan, O. Allain, A. Dervieux, and F. Alauzet. An L^∞-L^p mesh adaptive method for computing unsteady bi-fluid flows. *Int. J. Numer. Meth. Engng*, 84(11):1376–1406, 2010. (Cited on pages 67, 78, 92, 111, 148, 151 and 152.)

[49] W. Hackbusch. *Multigrid methods and applications.* Springer, 1985. (Cited on pages 27, 28, 31, 37 and 56.)

[50] R. Hartman. Multitarget error estimation and adaptivity in aerodynamic flow simulations. *SIAM Journal on Scientific Computing*, 31(1):708–731, 2008. (Cited on page 91.)

[51] W. Huang. Metric tensors for anisotropic mesh generation. *J. Comp. Phys.*, 204(2):633–665, 2005. (Cited on pages 90 and 119.)

[52] A.-C. Lesage, O. Allain, and A. Dervieux. On level set modelling of bi-fluid capillary flow. *Int. J. Numer. Meth. Fluids*, 53(8):1297–1314, 2007. (Cited on pages 71, 148, 149 and 150.)

[53] A. Loseille. *Adaptation de maillage anisotrope 3D multi-échelles et ciblée à une fonctionnelle pour la mécanique des fluides. Application à la prédiction haute-fidélité du bang sonique.* PhD thesis, Université Pierre et Marie Curie, Paris VI, Paris, France, 2008. (in French). (Cited on pages 5 and 51.)

[54] A. Loseille and F. Alauzet. Continuous mesh framework. Part I: well-posed continuous interpolation error. *SIAM Journal on Numerical Analysis*, 49(1):38–60, 2011. (Cited on pages 5, 9, 11, 14, 51, 95, 119, 123, 124 and 134.)

[55] A. Loseille and F. Alauzet. Continuous mesh framework. Part II: validations and applications. *SIAM Journal on Numerical Analysis*, 49(1):61–86, 2011. (Cited on pages 5, 9, 11, 14, 51, 95, 119, 123, 124 and 134.)

[56] A. Loseille, A. Dervieux, and F. Alauzet. A 3D goal-oriented anisotropic mesh adaptation applied to inviscid flows in aeronautics. In *48th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2010-1067, Orlando, FL, USA, Jan 2010. (Cited on pages 91 and 120.)

[57] A. Loseille, A. Dervieux, and F. Alauzet. Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations. *J. Comp. Phys.*, 229(8):2866–2897, 2010. (Cited on pages 83, 91, 99 and 101.)

[58] A. Loseille, A. Dervieux, and F. Alauzet. Anisotropic norm-oriented mesh adaptation for compressible flows. In *53rd AIAA Aerospace Sciences Meeting*, AIAA-2015-2037, Kissimmee, Florida, Jan 2015. (Cited on page 118.)

[59] A. Loseille, A. Dervieux, P.J. Frey, and F. Alauzet. Achievement of global second-order mesh convergence for discontinuous flows with adapted unstructured meshes. In *37th AIAA Fluid Dynamics Conference and Exhibit*, AIAA-2007-4186, Miami, FL, USA, Jun 2007. (Cited on pages 68, 70, 95, 112 and 125.)

[60] A. Loseille and R. Löhner. Adaptive anisotropic simulations in aerodynamics. In *Proceedings of 48th AIAA Aerospace Sciences Meeting, AIAA 2010-169, Orlando, FL, USA, January 2010*, 2010. (Cited on pages 5 and 51.)

[61] A.J. Wathen M. Arioli, D. Loghin. Stopping criteria for iterations in finite element methods. *Numerische Mathematik*, 99:381–410, 2005. (Cited on pages 4, 28, 39, 42, 50, 57 and 60.)

[62] F. Magoules. *Computational Fluid Dynamics*. CRC Press, Boca Raton, London, New York, Washington D.C., 2011. (Cited on pages 94 and 96.)

[63] D.J. Mavriplis. Adaptive mesh generation for viscous flows using Delaunay triangulation. *J. Comp. Phys.*, 90:271–291, 1990. (Cited on pages 4 and 50.)

[64] W.F. Mitchel. The hp-multigrid method applied to hp-adaptive refinement of triangular grids. *Numerical Linear Algebra with Applications*, 17:211–228, 2010. (Cited on pages 4 and 50.)

[65] E. Morano. *Resolution des equations d'euler par une methode multigrille stationnaire*. PhD thesis, Université de Nice, 1992. (Cited on page 32.)

[66] W.A. Mulder. A new multigrid approach to convection problems. *J. Comp. Phys.*, 83:303–323, 1989. (Cited on pages 4 and 50.)

[67] C. Nastase and D. J. Mavriplis. A parallel hp-multigrid solver for three-dimensional Discontinuous Galerkin discretizations of the Euler equations. *AIAA Aerospace Sciences Meeting, Reno, NV*, AIAA Paper 2007-0512, 2007. (Cited on pages 4 and 50.)

[68] J. Nitsche. Ein kriterium fr die quasi-optimalitat des ritzschen verfohrens. *Numer. Math.*, 11:346 348, 1968. (Cited on page 19.)

[69] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988. (Cited on page 148.)

[70] S.V. Patankar and D.B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15:1787–1806, 1972. (Cited on page 151.)

[71] N. A. Pierce and M. B. Giles. Adjoint and defect error bounding and correction for functional estimates. *Journal of Computational Physics*, 200:769–794, 2004. (Cited on page 22.)

[72] N.A. Pierce and M.B. Giles. Adjoint recovery of superconvergent functionals from PDE approximations. *SIAM Review*, 42(2):247–264, 2000. (Cited on pages 21, 22 and 90.)

[73] U. Rüde. Fully adaptive multigrid methods. *SIAM J. Numer. Anal.*, 30:230 248, 1993. (Cited on pages 4 and 50.)

[74] J. Simon. Non-homogenous viscous incompressible fluids: existence of velocity, density, and pressure. *SIAM journal on mathematical analysis*, 21(5):1093–1117, 1990. (Cited on page 148.)

[75] K. Stüben. Appendix A: An introduction to Algebraic Multigrid. In *Multigrid by U. Trottenberg, C.W. Osterlee and A. Schüller*, pages 413–532. Academic Press, San Diego, 2001. (Cited on pages 4 and 50.)

[76] U. Trottenberg, C. W. Osterlee, and A. Schüller. *Multigrid*. Academic Press, San Diego, 2001. (Cited on pages 3, 4, 27 and 50.)

[77] Y.V. Vasilevski and K.N. Lipnikov. An adaptive algorithm for quasi-optimal mesh generation. *Comput. Math. Math. Phys.*, 39(9):1468–1486, 1999. (Cited on pages 90, 96 and 119.)

[78] Y.V. Vasilevski and K.N. Lipnikov. Error bounds for controllable adaptive algorithms based on a Hessian recovery. *Computational Mathematics and Mathematical Physics*, 45(8):1374–1384, 2005. (Cited on pages 90 and 119.)

[79] D.A. Venditti and D.L. Darmofal. Adjoint error estimation and grid adaptation for functional outputs: application to quasi-one-dimensional flow. *J. Comp. Phys.*, 164(1):204–227, 2000. (Cited on page 90.)

[80] D.A. Venditti and D.L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *J. Comp. Phys.*, 187(1):22–46, 2003. (Cited on page 120.)

[81] R. Verfürth. *A Posteriori Error Estimation Techniques for Finite Element Methods*. Oxford University Press, Oxford, 2013. (Cited on pages 90 and 120.)

[82] P. Wesseling. *An introduction to multigrid methods*. John Wiley and Sons, 1992. (Cited on pages 27 and 28.)

[83] M. Yano and D. Darmofal. An optimization framework for anisotropic simplex mesh adaptation. *J. Comp. Phys.*, 231(22):7626–7649, 2012. (Cited on pages 91, 119 and 120.)

[84] O.C. Zienkiewicz and J.Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique. *Int. J. Numer. Meth. Engng*, 33(7):1331–1364, 1992. (Cited on pages 91 and 94.)

Ce document a été préparé à l'aide de l'éditeur de texte G Emacs et du logiciel de composition typographique LaTeX 2_ε.