Résolution des équations de Navier-Stokes linéarisées pour l'aéroélasticité, l'optimisation de forme et l'aéroacoustique

Aloïs Bissuel

université PARIS-SACLAY

NNT : 2018SACLX019

Thèse de doctorat

Résolution des équations de Navier-Stokes linéarisées pour l'aéroélasticité, l'optimisation de forme et l'aéroacoustique

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'École Polytechnique

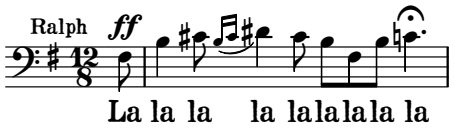
École doctorale n°574 mathématiques Hadamard (EDMH)
Spécialité de doctorat: Mathématiques appliquées

Thèse présentée et soutenue à Palaiseau, le 22 janvier 2018, par

Aloïs Bissuel

Composition du Jury :

Victorita Dolean Maître de conférence, Université Côte d'Azur (LJAD)	Rapporteur
Rémi Abgrall Professeur, Universität Zürich (I-Math)	Rapporteur
Luc Giraud Directeur de recherche, INRIA Bordeaux	Président du jury
Marc Massot Professeur, École Polytechnique (CMAP)	Examinateur
Nicole Spillane Chargée de recherche, École Polytechnique (CMAP)	Examinateur
Grégoire Allaire Professeur, École Polytechnique (CMAP)	Directeur de thèse
Laurent Daumas Ingénieur, Dassault Aviation	Examinateur



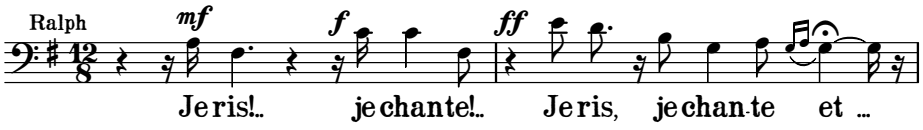
Bizet, *La jolie fille de Perth*, Acte 2

Remerciements

Une thèse, comme tout travail scientifique, n'est jamais réalisée dans l'isolement. J'aimerais ici sincèrement remercier tous ceux qui ont, de près ou de loin, participé à son bon déroulement.

Tout d'abord, je souhaite remercier Dassault Aviation de m'avoir proposé cette aventure. L'encadrement scientifique chez Dassault Aviation est, je le crois, d'une qualité rare. Mes premiers mots seront pour mon encadrant Laurent Daumas, grand représentant de l'ovalie, et fin connaisseur du Sud, de l'orthographe et de l'Équipe. J'ai eu la chance d'être son premier thésard, et d'avoir été encadré avec autant d'attention. Michel Mallet, qui m'a fait l'honneur de m'accepter dans son équipe et d'accueillir avec un intérêt peu dissimulable toutes mes avancées. Frédéric Chalot, sans qui AeTher ne serait rien. Zdenek Johan, grand défenseur de la méthode linéarisée et qui n'a jamais manqué de me surprendre par sa connaissance parfaite des cas tests. Sébastien Barré, qui m'a transmis le virus de l'acoustique avec une pédagogie et un enthousiasme contagieux. Enfin, Nicolas Forestier, qui possède un art redoutable d'animer l'open space, et sans qui tout cela n'aurait pas eu lieu. Sans oublier les autres collègues avec qui j'ai eu le plaisir de travailler, Pierre-Élie, Flavien, Ximun, Gilbert, et tous les autres. Je ne puis terminer sans citer le co-thésard Pierre Yser, compagnon d'open-space, de ligne d'eau et d'Arduino, dont l'analyse clairvoyante et intransigeante de la société a toujours permis des discussions passionnantes.

Je voudrais également remercier Grégoire Allaire, mon directeur de thèse, pour l'excellent encadrement de mon travail, qui m'a permis de prendre du recul scientifique sur mes travaux. Sa disponibilité sans faille, quelques soient les circonstances, m'a toujours époustouflé. Sans Nicole Spillane et fine connaisseuse de la décomposition de domaine, cette thèse ne serait pas tout à fait ce qu'elle est. Je lui suis infiniment reconnaissant pour tous ses conseils de relecture avisés et bienveillants et pour m'avoir fait partager son exigence typographique, certainement supérieure à la mienne !



Bizet, *La jolie fille de Perth*, Acte 2

Enfin, ces trois années de thèse ont été également l'occasion de faire autre chose que des mathématiques. Les citations musicales qui ponctuent chaque début de chapitres en sont un témoignage parfois espiègle. Charles-Henri, Jean-Matthieu et Pierre, puis Sarah, ont été mes compagnons de route de colocation, qui ont dû supporter (mais je leur ai bien rendu, je crois) mon exigence culinaire.

Enfin, puisque je suis en vérité le dernier diplômé de la famille – quoique j'ai pu dire – j'ai eu deux frères qui m'ont montré assez brillamment la voie dans beaucoup de domaines. Enfin je ne serai jamais assez reconnaissant envers mes parents pour l'ouverture au monde des arts qu'ils m'ont offerte.

Table des matières

Remerciements **2**

1 Introduction **7**
1.1 Applications 9
1.1.1 Optimisation de forme aérodynamique 9
1.1.2 Aeroélasticité 10
1.1.3 Aéroacoustique 11
1.2 État de l'art 14
1.3 Résumé des travaux 15
1.3.1 Solveur linéaire parallèle 15
1.3.2 Schéma de discrétisation 19
1.4 Publications et communications 22

2 Le code Aether **24**
2.1 Les équations de Navier-Stokes 25
2.1.1 Les équations de Navier-Stokes sous forme conservative 25
2.1.2 Variables entropiques et symétrisation 26
2.2 Éléments finis et stabilisation 28
2.2.1 Rappels sur les éléments finis 28
2.2.2 Stabilisation des éléments finis 34
2.3 Les équations de Navier-Stokes linéarisées 35
2.3.1 Notation matricielle 35
2.3.2 Différenciation automatique 38
2.4 Parallélisation 40

I Solveur linéaire parallèle **44**

3 Le solveur GMRES **45**
3.1 Solveur itératif ou solveur direct ? 45
3.2 GMRES 46
3.2.1 Description de l'algorithme 46
3.2.2 Parallélisation 51
3.2.3 Préconditionnement 52

3.2.4 Amélioration des redémarrages par la déflation 53
3.3 La méthode Block-GMRES 58
3.3.1 Intérêts attendus 58
3.3.2 Description 59
3.3.3 Déflation des valeurs propres 61
3.3.4 Résultats 63
3.3.5 La déflation des seconds membres 70
3.4 Conclusion 71

4 Préconditionnement **73**
4.1 Introduction 73
4.2 Les préconditionneurs existant dans AeTher 74
4.2.1 Préconditionnement bloc diagonal 74
4.2.2 block-SOR 76
4.2.3 ILU(0) 76
4.3 Parallélisation du préconditionnement 78
4.3.1 Schwarz additif 79
4.3.2 Conditions de Schwarz optimisées 83
4.3.3 Méthodes de sous-structuration 84
4.4 Déflation d'espace grossier 85
4.4.1 Les limites du préconditionnement Schwarz additif . . 85
4.4.2 Déflation de vecteurs 86
4.4.3 Espace de Nicolaides 89
4.4.4 Extensions 91
4.5 Préconditionnement ILU(k) 96
4.5.1 Remplissage de l'ILU 96
4.5.2 Résultats 99
4.6 Conclusion 108

II Schéma de discrétisation **110**

5 Stabilisation **111**
5.1 Forme de la matrice de stabilisation 112
5.1.1 Stabilisation d'une équation d'advection 1D 112
5.1.2 Stabilisation d'un système d'équations d'advection 1D 119
5.1.3 Cas multidimensionnel 121
5.1.4 Variables entropiques 123
5.1.5 Calcul pratique de la matrice de stabilisation 124
5.2 Nouvelle matrice de stabilisation 128
5.2.1 Calcul de la matrice de stabilisation complète 128
5.2.2 Navier-Stokes non-linéaire 130
5.2.3 Résultats en aéroélasticité sur la maquette DTP . . . 136
5.2.4 Résultats en linéarisé basse vitesse 139

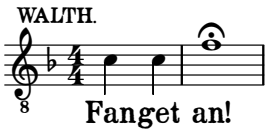
5.2.5 Résultats en aéroacoustique 139
5.3 Conclusion 141

6 Conditions aux limites **144**
6.1 Conditions aux limites de Dirichlet non homogènes 144
6.1.1 Imposition d'une variation de pression 145
6.1.2 Imposition d'une onde incidente par les caractéristiques 147
6.1.3 Validation des conditions aux limites incidentes 149
6.1.4 Imposition de la vitesse de paroi en aéroélasticité . . . 155
6.2 Conditions aux limites transparentes 156
6.2.1 Présentation 156
6.2.2 Modèle 1D 157
6.2.3 Résultats en 2D et en 3D 168
6.3 Résultats industriels d'aéroacoustique 170
6.3.1 Présentation de l'étude SFWA 170
6.3.2 Résultats sans écoulement 172
6.3.3 Résultats avec écoulement 176
6.4 Conclusion 178

7 Conclusion et perspectives **180**

A Calcul des valeurs de Ritz harmoniques **184**
A.1 Simplification du problème 184
A.2 Calcul du vecteur $\mathbf{f}_m$ 185

Bibliographie **186**



Wagner, *Die Meistersinger von Nürnberg*, 1. Akt, 2. Sz

Chapitre 1

Introduction

Le travail de conception des avions utilise de plus en plus la simulation numérique pour prédire les performances et le comportement d'un avion en vol [153]. En aérodynamique, le calcul numérique de l'écoulement de l'air autour de l'avion permet de prédire les performances de l'aéronef, et ainsi guide le dessin des formes externes de l'avion.

L'extraordinaire expansion des moyens informatiques a permis l'utilisation de modèles de plus en plus précis en calcul aérodynamique. Le champ d'application des méthodes se fait plus large. La description des avions est plus fine (accessoires extérieurs, supports de volets, etc). La précision des résultats est meilleure.

La conception d'un avion se fait habituellement en trois phases. La première phase, dite d'avant-projet, cherche à trouver la configuration de l'avion répondant le mieux au positionnement de marché visé (rayon d'action, masse utile et nombre de passagers, longueur de piste nécessaire au décollage et à l'atterrissage, etc.). C'est durant cette phase que la définition initiale de l'avion est trouvée : nombre de réacteurs et leur placement approximatif, dimensions et formes générales de l'avion (longueur du fuselage, envergure de l'aile et du plan horizontal, surface alaire, flèche, etc), masse, quantité de carburant. Les outils de conception sont simples afin de pouvoir tester un grand nombre de configurations potentielles : méthode des doublets [91] pour l'aérodynamique, modèle de poutre pour la structure de l'aile, modèle simple des moteurs pour trouver les performances de l'avion, modèles semi-empiriques en aéroacoustique, etc.

Vient ensuite la phase de conception préliminaire, durant laquelle les formes aérodynamiques sont définies conjointement avec une première ébauche de la structure. Les performances de l'avion sont affinées par rapport à la phase d'avant-projet. Les outils de conception sont précis : des calculs Navier-Stokes sont menés pour connaître l'aérodynamique précise de l'avion et l'améliorer. Durant cette phase, l'optimisation automatique de forme aéro-

dynamique est utilisée. Elle a pour but d'améliorer un critère (par exemple la traînée) sous des contraintes aérodynamique (portance minimale requise) ou de forme (volume de l'aile, envergure, etc.) par modification de variables dites géométriques décrivant la forme de l'avion. Comme la configuration initiale n'est pas loin de l'optimum (en d'autres termes, il ne s'agit pas ici d'exploration de concepts), l'optimisation est réalisée par une méthode utilisant des gradients. Le calcul des gradients de la fonction coût par rapport aux variables géométriques peut se faire par différences finies, mais il est bien plus efficace de le calculer directement en utilisant les équations de Navier-Stokes linéarisées, comme par exemple, par une méthode adjointe. Des premiers calculs aéroacoustiques sont réalisés durant cette phase, pour que les éventuelles contraintes de niveau de bruit soient prises en compte, afin d'influer la forme de l'avion ou d'anticiper des besoins de traitements acoustiques du moteur. Des calculs de propagation de bruit de réacteur sont effectués afin de prévoir l'empreinte sonore d'un avion à partir de données fournies par le motoriste sur le réacteur seul. Les équations de l'acoustique sont celles d'Euler linéarisées, et on peut donc utiliser le solveur Navier-Stokes linéarisé pour les résoudre.

La dernière phase de conception de l'avion, dite détaillée, vise à terminer la définition de la structure, des systèmes (hydrauliques, électriques, etc), et à consolider certains résultats aérodynamiques en vue de la certification de l'appareil. À la fin de cette phase, toutes les pièces de l'avion ont été dessinées, et celui-ci peut être mis en production. C'est durant cette phase que des calculs de flottement sont effectués. Le flottement, également connu en anglais sous le terme *flutter*, est un couplage catastrophique entre l'écoulement aérodynamique et les modes structuraux de l'avion (principalement de l'aile et du plan horizontal), qui conduit à un transfert d'énergie vers la structure sur certains modes. S'ils sont excités, ils se mettent à croître jusqu'à ce qu'un phénomène non-linéaire aérodynamique ou structurel arrête leur augmentation d'amplitude. Cela peut mener à la rupture de la structure de l'avion. Lorsque les modèles structuraux et aérodynamiques ont suffisamment convergés, un campagne de calcul aéroélastique est menée. Elle vise à déterminer la frontière d'apparition du phénomène de flottement et nécessite un très grand nombre de calculs linéarisés – jusqu'à plusieurs dizaines de milliers – qui doivent être réalisés en un temps raisonnable.

Ainsi, l'aéroélasticité, l'optimisation de forme aérodynamique et l'aéroacoustique sont trois applications qui demandent la résolution des équations de Navier-Stokes linéarisées. La discrétisation de ces équations fournit un système linéaire de très grande taille (plusieurs dizaines voire centaines de millions d'inconnues) de la variable complexe (pour l'aéroélasticité et l'aéroacoustique) ou de la variable réelle (pour l'optimisation de forme aérodynamique). Vu leur taille, ces systèmes sont nécessairement résolus en parallèle sur plusieurs processeurs en même temps. Afin que l'approche linéarisée utilisable dans un contexte industriel, la méthode de résolution des

systèmes linéaires correspondant doit être très efficace et robuste.

1.1 Applications

Comme on l'a expliqué précédemment, les équations de Navier-Stokes linéarisées sont utilisées pour trois applications, l'optimisation de forme aérodynamique, l'aéroélasticité et l'aéroacoustique, qui sont chacune détaillées dans cette partie

1.1.1 Optimisation de forme aérodynamique

L'optimisation de forme est utilisée lors de la phase de conception préliminaire, pour aider l'ingénieur aérodynamicien à concevoir les formes de l'avion. L'optimisation automatique de forme aérodynamique permet de minimiser une fonction coût sous contraintes, en jouant sur des variables géométriques qui paramétrisent la forme. La fonction coût est issue des données aérodynamiques. C'est typiquement la traînée de l'avion, ou un écart à une répartition de pression objectif. Les contraintes peuvent être d'ordre aérodynamique (respecter une portance minimale) ou concernant la forme (envergure maximale de l'aile, épaisseur ou volume minimal). On appelle observation le calcul d'une quantité (comme la portance) sur le champ aérodynamique, qui entre en compte dans la formulation de la fonction coût ou d'une contrainte. Enfin, l'optimisation automatique de forme nécessite une méthode de déformation de la surface. Dassault Aviation, comme d'autres [6], a fait le choix de paramétrer la forme aérodynamique.

La minimisation de la fonction coût se fait à l'aide d'une méthode utilisant un gradient. Cela demande le calcul des dérivées partielles des observations aérodynamiques par rapport aux variables géométriques. Il faut donc connaître les dérivées partielles du champ aérodynamique par rapport aux variables géométriques. Deux méthodes existent pour calculer ces dérivées. La méthode directe calcule pour chaque variation de variable géométrique la variation de toutes les observations. La méthode adjointe, issue de la théorie du contrôle optimal [99, 114, 126] utilise un système adjoint associé à une observation pour obtenir en un seul calcul toutes les variations de variables géométriques. La méthode directe est plus efficace lorsqu'il y plus d'observations que de variables, et la méthode adjointe est plus intéressante lorsqu'il y plus de variables que d'observations. Pour plus de détails, on pourra consulter [4, 37, 42].

Le gradient des équations de la mécanique des fluides peut s'obtenir de deux façons. Soit les équations sont linéarisées analytiquement et résolues à l'aide de schéma *ad hoc*, soit le schéma numérique de résolution des équations non-linéaires est linéarisé par différenciation automatique (la section 2.3.2 traite spécifiquement cette notion). La première méthode, qui est utilisée dans [86, 126, 147], est appelée *gradient continu*. La deuxième [6] est dénommée

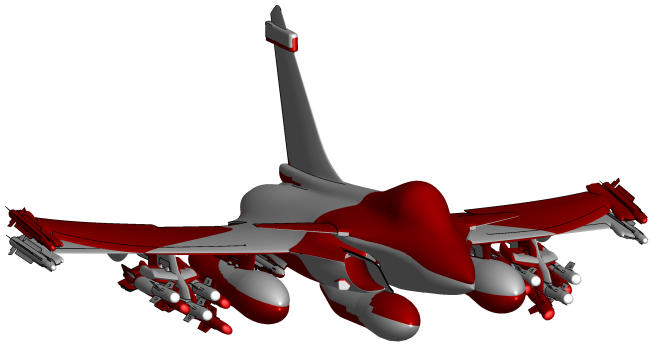


FIGURE 1.1 – Mode symétrique de voilure sur un Rafale avec emports.

gradient discret. Ces deux techniques ont chacune leurs avantages et leurs inconvénients. Sans volonté d'exhaustivité, la méthode du gradient continu permet une analyse théorique du schéma, mais la définition des conditions aux limites est technique. A contrario, la méthode du gradient discret ne demande pas de travail théorique important pour les conditions aux limites, mais rend délicate l'analyse du schéma. Par exemple, le gradient de la stabilisation du schéma non-linéaire doit être inclus dans le linéarisé. L'analyse théorique de ce terme est compliquée (sur son impact pratique, on pourra consulter les résultats de la section 5.2.3). Enfin, on notera que l'utilisation du gradient discret permet toujours de diminuer la fonction coût sur le maillage choisi, tandis que la méthode du gradient continu ne garantit cette diminution que dans la limite d'une discrétisation infiniment fine.

1.1.2 Aeroélasticité

Le flottement (*flutter* en anglais) est un phénomène aéroélastique, c'est-à-dire de couplage entre le fluide et la structure de l'avion. Il correspond à un transfert d'énergie du fluide vers la structure. Si ce transfert d'énergie est supérieur à l'amortissement des modes structuraux, ceux-ci deviennent auto-entretenus et leur amplitude augmente, parfois jusqu'à destruction de l'aéronef. La vitesse de l'avion influe sur la quantité d'énergie que le fluide apporte aux modes structuraux.

Afin d'éviter de devoir coupler les calculs de mécanique des fluides et de structure, les forces aérodynamiques générées par chaque mode sont calculées séparément, avant d'être recombinées par la méthode p-k [85], que l'on trouvera également décrite dans [51]. Les calculs sont effectués sur des modes de structure. La figure 1.1 montre le déplacement associé à un mode de voilure symétrique sur un Rafale avec emports.

Les efforts aérodynamiques associés à chaque mode doivent être calculés sur un ensemble de dix fréquences et au moins six vitesses de l'avion. Plus de

400 modes sont évalués ainsi, pour une campagne d'évaluation de flottement sur un avion. Cela fait un total de 24000 calculs à réaliser.

Cela exclut l'utilisation d'une méthode temporelle, où l'on calculerait l'écoulement aérodynamique autour de l'avion en mouvement à la fréquence souhaitée, car elle serait trop coûteuse en temps de calcul. Une méthode linéarisée fréquentielle, telle que décrite par Lesoine dans [97], permet de diminuer substantiellement le temps de calcul. L'approche linéarisée fréquentielle est utilisée chez Dassault Aviation [38], mais également par le DLR [111, 158] à l'aide du code Tau [60].

Un calcul d'ordre de grandeur permet de trouver un temps acceptable pour une utilisation industrielle de l'approche linéarisée. Si chacun des 24000 calculs demande une minute pour sa résolution, le temps total de calcul de la campagne d'aéroélasticité est d'un peu plus de 16 jours, ce qui est raisonnable et compatible avec une utilisation industrielle.

1.1.3 Aéroacoustique

L'aéroacoustique traite de l'acoustique en présence d'écoulements (propagation et production de bruit), et sert à évaluer l'impact sonore des avions au décollage, à l'atterrissage ou encore au sol. Plusieurs types de sources acoustiques génèrent le bruit d'un avion. Les sources aérodynamiques, dues à la turbulence, émettent un bruit sur un large spectre, qui est dû aux nombreuses échelles spatiales de la turbulence. Les sources issues des machines tournantes, principalement les réacteurs, ont un spectre resserré sur des fréquences caractéristiques (et leurs harmoniques) qui sont directement liées à la vitesse de rotation et au nombre de pales de la machine. La propagation du bruit de ces sources peut se faire efficacement à l'aide d'une méthode fréquentielle, d'autant qu'elles sont généralement connues par leurs composantes fréquentielles.

Les équations de propagation acoustique s'obtiennent en considérant une petite perturbation de l'écoulement dans les équations d'Euler. En l'absence d'écoulement porteur, les équations d'Euler linéarisées se réduisent à une équation d'Helmholtz, dont la résolution ne demande pas nécessairement une discrétisation du volume. Une équation de type Helmholtz peut être trouvée lorsque l'écoulement porteur est potentiel [1]. Des couches de cisaillement présentes dans l'écoulement porteur nécessitent d'utiliser les équations d'Euler linéarisées pour prendre correctement en compte les effets de réfraction associés.

Dans cette thèse, on s'est intéressé à la propagation acoustique de bruit de soufflante de réacteur, dont la source peut être modélisée par une décomposition modale sur quelques fréquences. Une approche linéarisée fréquentielle est bien sûr très adaptée. Pour résoudre ces problèmes, certains codes utilisent cependant une approche temporelle, comme dans [106, 128] ou encore [117].

Une étude de propagation aéroacoustique pour un bruit de machine

tournante part d'une description de la source, qui est connue fréquence par fréquence, et décomposée en mode sur chacune de celles-ci. Ensuite, chacune de ces composantes donne lieu à un calcul de propagation aéroacoustique depuis le réacteur jusqu'à l'endroit d'intérêt en utilisant les équations d'Euler linéarisées autour d'un écoulement porteur. Enfin, le bruit total est recomposé par une somme pondérée et décorrélée de toutes ces composantes.

Selon son but, un calcul de propagation aéroacoustique peut chercher à déterminer le bruit en champ proche ou en champ lointain. Dans le premier cas, il s'agit d'évaluer le bruit sur une surface proche de la tuyère du réacteur afin de détecter d'éventuels problèmes de fatigue acoustique (des matériaux) causés par un niveau sonore trop élevé. Le bruit en champ lointain sert à calculer la signature sonore d'un avion au décollage ou à l'atterrissage, afin de quantifier la gêne acoustique des habitants riverains d'un aéroport.

Dans ces calculs en champ lointain, la propagation acoustique utilisant les équations d'Euler linéarisées est réalisée jusqu'à une surface d'interpolation située une certaine distance (typiquement une dizaine de longueur d'onde) afin d'une part que les termes acoustiques évanescents de champ proche soient éliminés, et d'autre part d'être loin des gradients de vitesse de propagation induits par l'écoulement – en d'autres termes être dans une zone de champ porteur uniforme non perturbé par l'avion. La propagation depuis cette surface d'interpolation, dite de Kirchhoff, jusqu'aux micros virtuels à grande distance se fait par une méthode intégrale basée sur des fonctions de Green. Ces dernières sont faciles à exprimer puisque l'écoulement est uniforme.

L'aéroacoustique possède une particularité qui la démarque des deux autres applications précédemment mentionnées. Les calculs de propagation demandent un maillage différent de celui utilisé pour calculer le champ porteur. Les maillages Navier-Stokes sont très structurés près de la surface de l'avion, afin de capturer correctement la couche limite. Une discrétisation adéquate de cette couche limite demande de nombreux points près de la surface. Cela conduit à des éléments très étirés dans cette zone, comme on présenté sur la figure 1.2 en haut. Un calcul de propagation acoustique demande un maillage isotrope, afin de respecter une discrétisation suffisante (qui dépend de la longueur d'onde calculée) dans toutes les directions possibles de propagation des ondes.

La solution du calcul Navier-Stokes non linéaire est projetée sur le maillage acoustique. Ce maillage est trop grossier pour discrétiser correctement la couche limite, comme on le voit sur la figure 1.2, image du milieu, ligne du bas. De plus, l'épaisseur de la couche limite est en pratique très petite devant les longueurs d'ondes calculées, et ainsi son effet sur la propagation acoustique est minimal et négligé. De surcroît, il serait trop coûteux de la mailler convenablement. Il faut donc supprimer la couche limite de l'écoulement porteur, ce qui est fait par un lissage volumique appliqué aux éléments proches de la paroi. Cette solution lissée sur le maillage acoustique est en tout point similaire à la solution du calcul non-linéaire original. Enfin, la

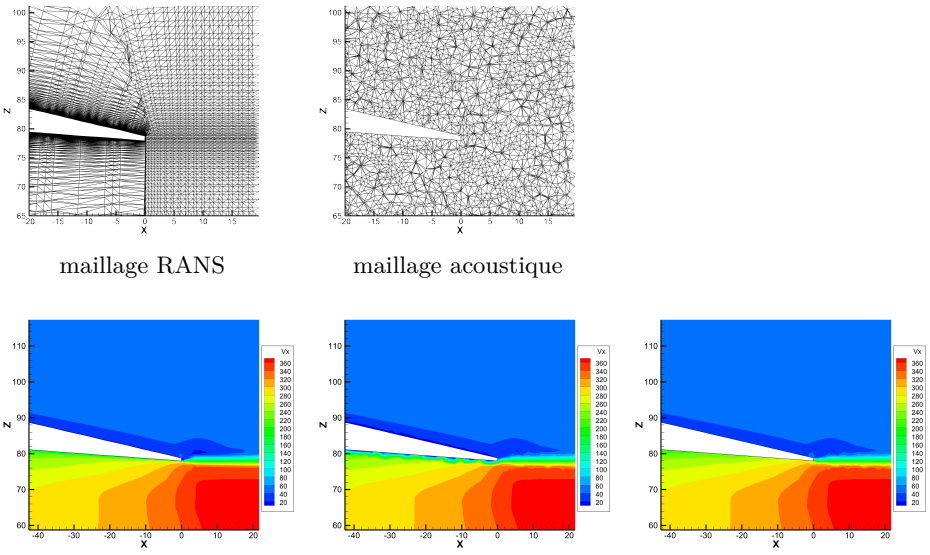


FIGURE 1.2 – Haut : maillages utilisés pour le calcul Navier-Stokes du champ porteur par RANS et pour la propagation acoustique. Bas : Effet de la projection de la solution RANS sur le maillage acoustique, puis effet du lissage de la couche limite.

suppression de la couche limite impose d'utiliser une condition de glissement à la paroi.

1.2 État de l'art

Le code AeTher, développé par Dassault Aviation, utilise la méthode des éléments finis stabilisés par SUPG pour résoudre les équations de Navier-Stokes compressibles. Il est décrit plus en détails dans le chapitre 2. Le code AeTher a été linéarisé par différenciation automatique, en utilisant le logiciel Tapenade [76].

Le code AeTher non-linéaire utilise l'algorithme GMRES [137] pour résoudre les problèmes implicites à chaque pseudo-pas de temps. Ce même algorithme a été repris pour les systèmes linéaires issus de la discrétisation des équations de Navier-Stokes linéarisées. Ces systèmes linéaires sont plus délicats à faire converger, et les tolérances de convergence sont beaucoup plus faibles. Pour l'application au linéarisé, le solveur GMRES a été doté d'une déflation des vecteurs propres associés aux plus petites valeurs propres selon la méthode de Morgan [115] améliorée par des astuces numériques [132]. Tout comme le code non-linéaire, les systèmes linéaires sont préconditionnés par une méthode de Schwarz additif, dont les solveurs locaux sont des préconditionneurs de type SOR par bloc (BSOR) ou ILU(0).

Le code AeTher est écrit en Fortran 77. L'intégralité des travaux décrits dans cette thèse a été implémentée dans ce code. Certains de ces travaux ont été industrialisés, et sont mis à la disposition de tous les utilisateurs du code linéarisé. AeTher utilise la bibliothèque MPI pour la parallélisation sur cluster de calcul haute performance. Toutes les simulations de cette thèse ont été réalisées sur les clusters de calcul de Dassault Aviation, en particulier sur le cluster Bull. Il se compose de lame possédant 2 processeurs Xeons *Ivy Bridge* de 12 cœurs chacun. Sur chaque cœur est lancé un processus MPI, qui s'occupe d'un sous-domaine. Des calculs industriels nécessitant 512 sous-domaines ont été lancés.

Dans le cadre des programmes avions, le code linéarisé est utilisé industriellement pour l'optimisation de formes [153]. Le temps de résolution des systèmes linéaires est par contre trop long pour envisager une utilisation industrielle pour une campagne d'aéroélasticité. En aéroacoustique, des premiers calculs avaient été menés avec le code Navier-Stokes linéarisé, mais avec des conditions limites particulières. Le plan d'imposition était traité comme une paroi limite vibrante. L'onde acoustique était générée par une vitesse non nulle imposée à cette paroi. Ce type de condition limite pose problème, car elle est complètement réfléchissante.

1.3 Résumé des travaux

Durant cette thèse, deux approches ont été suivies pour améliorer la résolution des équations de Navier-Stokes linéarisées. La première est évidemment les méthodes numériques de résolution de systèmes linéaires. La deuxième a été de travailler sur les schémas de discrétisation qui conduisent à ces systèmes. Ces deux axes ne sont pas opposés, bien au contraire. En effet, un problème mal discrétisé peut être difficile à résoudre car la solution du système linéaire n'est pas physique, et améliorer le solveur au prix de grands efforts serait un travail perdu.

Après un chapitre d'introduction sur le code AeTher, la thèse est divisée en deux grandes parties introduites précédemment, divisées chacune en deux chapitres. Dans les sections suivantes, les apports principaux de la thèse sont détaillés.

1.3.1 Solveur linéaire parallèle

Les maillages utilisés par Dassault Aviation pour résoudre les équations de Navier-Stokes sont très fins. Ils contiennent de plusieurs millions à quelques dizaines de millions de nœuds. Ainsi, les problèmes ne peuvent être résolus sur un seul processeur, mais doivent être partagés sur plusieurs dizaines voire centaines de processeur sur un super-calculateur. Les équations de Navier-Stokes ayant cinq inconnues, les systèmes linéaires issus de la discrétisation des équations de Navier-Stokes linéarisées ont plusieurs dizaines à centaines de millions d'inconnues. La taille de ces problèmes, qui conduit à un impératif de parallélisation, a guidé ce travail vers des solveurs linéaires qui soient très efficaces et qui se parallélisent bien.

L'algorithme GMRES est une méthode itérative de résolution de systèmes linéaires non symétriques. Inventé par Saad et Schultz [137], il se base sur la minimisation du résidu sur l'espace de Krylov généré par le résidu initial. Un espace de Krylov est généré par les itérés successifs de la matrice avec le vecteur initial choisi. L'orthonormalisation des itérés successifs par une méthode d'Arnoldi fournit une matrice de forme Hessenberg qui représente l'action de la matrice du système linéaire sur la base de Krylov construite. Afin d'améliorer les redémarrages, une déflation des petites valeurs propres est effectuée en ajoutant au début de l'espace de Krylov les vecteurs générant l'espace des vecteurs propres à éliminer, suivant une idée de Morgan [115].

Dans le chapitre 3, une extension de l'algorithme GMRES aux systèmes linéaires à plusieurs seconds membres a été testée. Une campagne de flottement nécessite la résolution, à la même fréquence et autour du même écoulement de référence, de nombreux modes structuraux qui chacun fournissent un second membre. La méthode block-GMRES est en tout point semblable à l'algorithme GMRES, mais s'applique à des vecteurs blocs, c'est-à-dire ayant plusieurs colonnes. Quelques modifications concernent l'orthonormalisation

de la base de Krylov : l'étape de normalisation du vecteur dans la méthode d'Arnoldi se transforme en une décomposition QR pour la méthode bloc associée.

L'algorithme block-GMRES peut être plus rapide pour résoudre s seconds membres simultanément que s résolutions successives par l'algorithme GMRES, pour deux raisons. La première est d'ordre informatique : les accès à la mémoire sont regroupés, ce qui accélère (entre autre) les produits matrice-vecteurs. De même, les opérations parallèles (comme les assemblages de vecteurs, les sommes globales pour terminer un produit scalaire) sont regroupées. La deuxième raison est mathématique. L'espace de Krylov est plus riche, et la solution de chaque second membre peut utiliser l'espace de Krylov généré par les autres seconds membres.

Comme cette technique a été identifiée dans le cas standard (non bloc) comme étant primordiale pour la convergence de l'algorithme GMRES, la déflation des plus petites valeurs propres a également été implémentée, en étendant les idées de Morgan [115] au cas bloc.

Les tests numériques effectués ont montré que la méthode block-GMRES n'apportait pas les gains espérés. Cet algorithme est plus lent ou moins robuste, selon les cas tests. Pour une méthode de Krylov, augmenter la taille de l'espace de recherche permet d'être plus robuste, au prix d'un ralentissement de la méthode dû au temps d'orthonormalisation accru. Or, l'augmentation du nombre de seconds membres résolus simultanément augmente d'autant la taille de l'espace de Krylov. Ainsi, si la résolution simultanée des seconds membres ne diminue pas grandement le nombre d'itérations requis pour atteindre la convergence, la méthode block-GMRES ne sera pas compétitive face à l'algorithme GMRES utilisant une taille réduite d'espace de Krylov.

Pour accélérer la méthode block-GMRES, la technique dite de déflation de second membre se propose d'éliminer les seconds membres qui auraient convergés avant les autres ou seraient une combinaison linéaire des autres. Cependant, une exploration préliminaire de cette technique ne laisse pas penser qu'il y ait de forts gains à tirer de cette technique sur les cas tests étudiés.

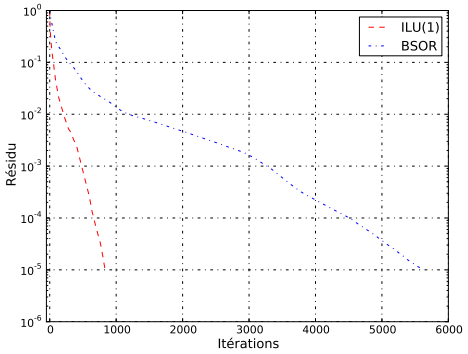
L'autre pan de cette partie sur les solveurs linéaires est traité dans le chapitre 4 et concerne le préconditionnement, qui consiste à transformer le système linéaire en un système équivalent numériquement plus facile à résoudre. Pour ce faire, on peut multiplier à gauche ou à droite le système par une matrice $\mathbf{M}$ approchant l'inverse de la matrice $\mathbf{A}$ du système linéaire. L'équation $\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b}$ sera plus facile à résoudre que le système initial. Un équilibre nécessaire existe entre qualité du préconditionnement et temps d'application. Un préconditionneur si bon qu'il réduit à quelques dizaines par exemple le nombre d'itérations pour résoudre un système est inutile si son temps de création et d'application est largement supérieur à la méthode itérative. Il faut donc trouver un préconditionnement efficace et rapide. Un deuxième défi du préconditionnement est sa parallélisation.

Puisque la matrice $\mathbf{A}$ est distribuée sur plusieurs processeurs, il faut trouver un moyen de calculer une inverse approchée de suffisamment bonne qualité. Pour cela, le code AeTher utilise une méthode issue de la décomposition de domaine, appelée méthode de Schwarz additif [44, 156]. Dans sa version la plus simple, la méthode de Schwarz additif applique l'inverse des matrices locales $\mathbf{A}_i$ (*i.e.* la restriction de la matrice globale $\mathbf{A}$ à un sous-domaine i) à la restriction sur le sous-domaine i du vecteur qui est préconditionné. La méthode de Schwarz additive s'utilise en préconditionnement en remplaçant l'inverse exacte de la matrice locale par une inverse approchée.

L'avantage de la méthode de Schwarz additif tient dans sa simplicité, notamment d'implémentation, mais elle découple partiellement les sous-domaines, ce qui peut conduire à une perte de performance lorsque le découpage augmente. Pour rendre plus robuste la méthode de Schwarz au nombre de sous-domaine, la méthode de Schwarz à deux niveaux a été implémentée durant cette thèse. Elle consiste à ajouter un problème grossier commun à tous les sous-domaines, qui permet un transfert de l'information entre tous les sous-domaines. La méthode repose sur un espace grossier, dont une base est notée $\mathbf{Z}$ composé de vecteurs dont on veut enlever l'influence sur la convergence du solveur de Krylov. La matrice du problème grossier, notée $\mathbf{E}$, est définie par $\mathbf{E} = \mathbf{Z}^T\mathbf{A}\mathbf{Z}$. Le préconditionneur à deux niveaux est $\mathbf{P} = \mathbf{I} - \mathbf{A}\mathbf{Z}\mathbf{E}^{-1}\mathbf{Z}^T$ et on résout le système $\mathbf{P}\mathbf{A}\widehat{\mathbf{x}} = \mathbf{P}\mathbf{b}$. Enfin, la solution est reconstruite en y rajoutant la contribution de l'espace grossier : $\mathbf{x} = \mathbf{Z}\mathbf{E}^{-1}\mathbf{Z}^T\mathbf{b} + \left(\mathbf{I} - \mathbf{Z}\mathbf{E}^{-1}\mathbf{Z}^T\mathbf{A}\right)\widehat{\mathbf{x}}$.

Une bonne définition de l'espace grossier est cruciale pour l'intérêt de la méthode. Le premier espace testé a été celui de Nicolaides [120], qui est composé des vecteurs constants par sous-domaines. Ils composent le noyau des problèmes de Neumann dans chaque sous-domaine pour l'opérateur laplacien. De bonnes performances annoncées pour les équations de Navier-Stokes dans [5] ainsi que la simplicité de la méthode nous ont poussé à tester cet espace. Pour deux cas tests, l'un 2D et l'autre 3D, aucune accélération significative de la convergence n'a été constatée, même si le cas 2D a montré une amélioration de l'extensibilité. L'espace GDSW (pour *Generalized Dryja-Smith-Widlund*), plus complexe a été également testé. Il se base sur une partition de l'interface entre les sous-domaine en faces, arêtes et coins. Les vecteurs grossiers de l'espace GDSW sont générés par l'extension harmonique d'une impulsion sur chacune de ces faces, arêtes et coins. L'utilisation de cet espace ajoute beaucoup de difficultés, tant pour la création des vecteurs que la formation de la matrice du problème grossier. En 2D, il n'a pas donné de résultats suffisamment intéressants pour que son extension délicate à des problèmes 3D industriels soit menée.

Enfin, l'effet d'autres préconditionneurs locaux ainsi que l'impact du recouvrement dans la méthode de Schwarz additif ont été testés, à l'aide de la bibliothèque PETSc [12, 13, 14]. Elle propose entre autre des solveurs et



Préconditionneur	BSOR	ILU(1)
Temps (s)	420	98

FIGURE 1.3 – Cas DTP (cas test II). À gauche, courbe de convergence de l'algorithme GMRES pour le préconditionnement BSOR et ILU(1). À droite, temps de résolution.

des préconditionneurs « clef en main ». Elle a permis de tester, sans autre effort que celui de l'interfacer avec le code AeTher, le recouvrement des sous-domaines dans la méthode de Schwarz additif et d'autres préconditionneurs locaux, tel que l'ILU(k). Ce dernier est une décomposition LU incomplète – d'où le nom d'ILU – en autorisant un certain niveau de remplissage dénoté par k. Il s'est avéré que ce préconditionneur local accélère grandement la résolution des systèmes linéaires. Par rapport au préconditionneur SOR par bloc (BSOR), l'utilisation de l'ILU(k) (également par bloc) a diminué jusqu'à un facteur dix le nombre d'itérations pour atteindre la convergence et le temps de résolution. Pour des calculs haute vitesse (c'est-à-dire en vol de croisière transsonique), l'ILU(1) est recommandé. La figure 1.4 présente les courbes de convergence ainsi que le temps de résolution pour le cas DTP (cas test II) pour le préconditionnement BSOR et ILU(1). Sur ce cas, le nombre d'itérations est divisé par six et le temps de résolution par quatre. Pour des applications basse vitesse en configuration hypersustentée, l'ILU(3) donne en général de meilleurs résultats. Le recouvrement des sous-domaines n'a pas apporté de gain suffisant sur le nombre d'itérations pour que les résolutions soient accélérées. L'utilisation de l'ILU(k) a permis de ramener le temps de résolution de chaque calcul d'une campagne d'aéroélasticité à l'ordre de la minute, ce qui rend possible l'utilisation industrielle de cette approche.

L'intégration de PETSc dans AeTher a été réalisée plus subtilement que d'utiliser la bibliothèque PETSc comme une boîte noire, à laquelle on fournirait un système linéaire à résoudre et le nom du solveur et du préconditionneur à utiliser pour la résolution. En effet, l'algorithme GMRES de PETSc ne permettait pas une déflation des plus petites valeurs propres satisfaisante pour des problèmes dans $\mathbb{R}$, et cette déflation n'existe pas pour des problèmes complexes. La méthode GMRES codée dans AeTher a été conservée, en réduisant l'utilisation de PETSc à des opérations sur les grands

vecteurs, à savoir les produits matrices-vecteurs, les produits scalaires et les combinaisons linéaires de vecteurs.

1.3.2 Schéma de discrétisation

La deuxième partie de cette thèse expose le travail effectué sur le schéma de discrétisation. Deux chapitres la composent, le premier (le chapitre 5) se concentre sur la stabilisation des éléments finis, et le deuxième (le chapitre 6) sur les conditions aux limites de Dirichlet.

Le code AeTher utilise la méthode des éléments finis pour discrétiser les équations de Navier-Stokes. Des éléments finis standards ne sont pas stables pour des équations d'advection-diffusion dont le nombre de Péclet est supérieur à un, ce qui interdit leur utilisation pour les équations de Navier-Stokes. Pour ce faire, le code AeTher utilise la méthode de stabilisation appelée SUPG [24]. Elle consiste en la modification des fonctions test. Si $\mathcal{L}_{\text{NS}} : \mathbf{V} \mapsto \widetilde{\mathbf{A}}_i \mathbf{V}_{,i} + \left(\widetilde{\mathbf{K}}_{ij} \mathbf{V}_{,j}\right)_{,i}$ est l'opérateur de Navier-Stokes, et $\widetilde{\mathbf{A}}_i$ la jacobienne du flux d'Euler selon la direction i, la stabilisation SUPG s'écrit :

$$\sum_{\Omega^e} \int_{\Omega^e} \left(\mathbf{W} + \boldsymbol{\tau} \widetilde{\mathbf{A}}_i \mathbf{W}_{,i}\right) \cdot \mathcal{L}_{\text{NS}}(\mathbf{V}) \, \mathrm{d}\mathbf{x} = 0 \tag{1.1}$$

où $\boldsymbol{\tau}$ est la matrice de stabilisation, paramètre central de la méthode SUPG, définie sur chaque élément Ω^e du maillage.

La construction de cette matrice de stabilisation dans AeTher suit celle donnée par Mallet dans [104]. La démonstration de cette construction est reprise dans cette thèse. Sur un système d'advection-diffusion 1D, l'ajout d'une viscosité artificielle bien choisie à un schéma de différences finies centrées permet d'obtenir un résultat exact aux nœuds. Cet ajout de viscosité se transpose pour une formulation élément fini par l'ajout d'un terme constant par élément aux fonctions test. L'extension à un système d'équations d'advection est immédiate, si l'opérateur d'advection est diagonalisable. On se ramène alors dans l'espace de ces vecteurs propres à des systèmes découplés d'advection. L'utilisation des variables entropiques pour formuler les équations d'Euler permet d'être dans ce cas, en se plaçant dans un espace bien choisi.

Une approximation est réalisée dans la construction théorique de la matrice $\boldsymbol{\tau}$ lors du passage à plusieurs dimensions. Pour ce faire, il faut généraliser la notion de valeur absolue d'une matrice carrée à une matrice rectangulaire $\mathbf{B}$. Mallet [104] propose d'utiliser $\sqrt{\mathbf{B}^T \mathbf{B}}$.

La matrice de stabilisation dans AeTher suit la construction de Mallet, en y ajoutant une approximation sur l'extension multidimensionnelle de $\boldsymbol{\tau}$. Les termes croisés, entre autres, y sont négligés.

Durant cette thèse, nous avons cherché à quantifier l'impact de ces approximations en reprenant littéralement la formule proposée par Mallet,

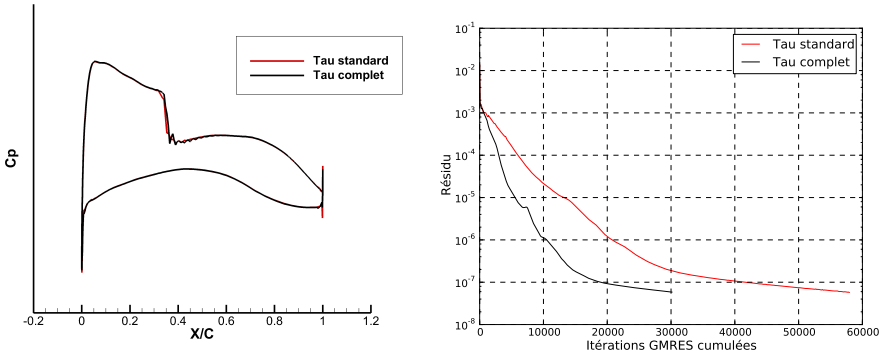


FIGURE 1.4 – À gauche, coupe de pression sur le cas Falcon croisière (cas test V). À droite, convergence du résidu non-linéaire en fonction du nombre cumulé d'itérations GMRES sur le cas Falcon décollage (cas test VI).

sans simplifications. Cela rend le calcul de $\boldsymbol{\tau}$ bien plus complexe, puisqu'alors une matrice de taille 5×5 doit être diagonalisée numériquement avec LAPACK pour pouvoir calculer l'inverse de sa racine carrée. Cette nouvelle forme de matrice de stabilisation, que l'on appelle complète, a été testée sur des calculs non-linéaires pour déterminer un écoulement stationnaire, et linéarisés pour les applications sus-citées. Les calculs non-linéaires ont montré une réduction substantielle du nombre d'itérations GMRES nécessaire à la résolution des problèmes linéaires implicites à chaque pseudo pas de temps. Ils ont permis de constater que la matrice complète introduit moins de viscosité numérique, à tel point qu'il est délicat de l'utiliser pour des calculs transsoniques. La figure 1.4 illustre ces deux effets sur deux cas tests différents. Enfin, des simulations à basse vitesse montrent une réduction de la sensibilité du code aux variations de normales du maillage surfacique au niveau du fuselage. En linéarisé, la réduction du nombre d'itérations n'a pas été retrouvée. L'impact de la différentiation de la stabilisation est également discutée.

Le chapitre 6 – deuxième chapitre de cette partie consacrée au schéma de discrétisation – s'intéresse aux conditions aux limites de Dirichlet. Tout d'abord, une méthode d'imposition des conditions de Dirichlet non homogènes à des variables non triviales du calcul est présentée. Le code AeTher utilise les variables entropiques pour exprimer les équations de Navier-Stokes. De nombreuses propriétés mathématiques intéressantes sont apportées par ces variables, mais leur utilisation complique l'imposition des conditions aux limites. Prenons l'exemple de la pression. On choisit de l'imposer par la première variable entropique V_1, qui est la seule n'étant pas uniquement proportionnelle à la vitesse ou la température. On inverse l'expression de la pression en fonction des cinq variables entropiques pour isoler la première variable $V_1 = f_p(V_2, \ldots, V_5, p)$. La linéarisation de cette relation donne

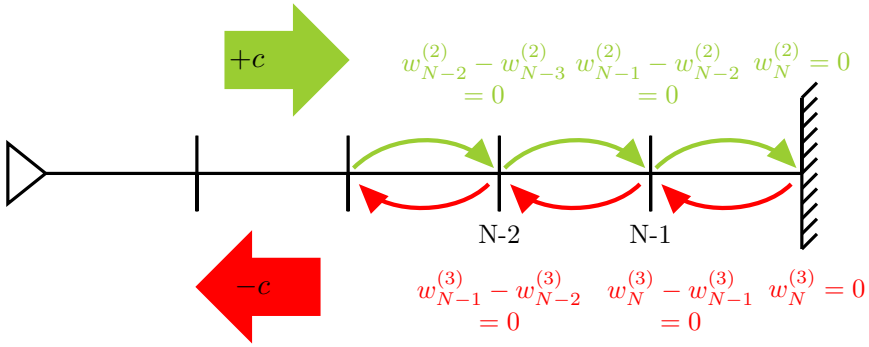


FIGURE 1.5 – Décentrement différent des caractéristiques $w^{(2)}$ et $w^{(3)}$ avec la discrétisation SUPG pour une vitesse de fluide nulle. Schéma repris de la figure 6.9.

$$dV_1 = \sum_{i=2}^{5} \frac{\partial f_p}{\partial V_i} dV_i + \frac{\partial f_p}{\partial p} dp \,,$$

qui permet d'éliminer la variable V_1 et imposer la variation de pression dp demandée. Une méthode de transformation algébrique du système linéaire à résoudre est donnée.

Les conditions aux limites de Dirichlet non homogènes ont été implémentées pour trois variables. Le travail sur les conditions non homogènes a été motivé par l'aéroacoustique. Tout d'abord, l'imposition d'une variation de pression a été développée. Cette condition aux limites a permis l'utilisation sur des cas industriels du code Navier-Stokes linéarisé en aéroacoustique. Pour offrir un contrôle sur l'énergie acoustique injectée, une condition aux limites sur la variable caractéristique entrante a été testée. Une caractérisation angulaire de cette condition par une méthode originale de cavité résonante a montré qu'elle reste performante pour des angles à la normale de la surface inférieurs à 50°. La dernière utilisation des conditions de Dirichlet non homogènes a été d'imposer une vitesse imaginaire pure à la paroi de l'avion pour les calculs d'aéroélasticité. Auparavant, la vitesse était imposée par pénalisation, ce qui faussait légèrement les courbes de convergence pour la première itération.

Ensuite, une propriété étonnante en aéroacoustique des conditions aux limites de Dirichlet homogènes sur toutes les variables a été expliquée. Ces conditions aux limites simples se sont révélées être transparentes pour les ondes sortantes dans les calculs de propagation. L'explication est simple. La stabilisation SUPG apporte un décentrement de la discrétisation de chaque caractéristique. Ce décentrement est total en 1D sans terme temporel. Comme le montre le schéma de la figure 1.5, la caractéristique sortante $w^{(2)}$ a une discrétisation décentrée. Elle atteint la condition de Dirichlet homogène en N sans que sa valeur aux nœuds précédents soit influencée par cette condition.

La caractéristique entrante $w^{(3)}$ est uniformément nulle sur tout le segment de discrétisation. L'ajout du terme fréquentiel complique l'analyse, mais ne change pas les conclusions et permet en outre d'étudier analytiquement la performance du schéma stabilisé par SUPG. En deux ou trois dimensions, la matrice $\boldsymbol{\tau}$ ne permet pas de décentrer parfaitement chaque caractéristique. La démonstration n'est plus exacte, mais des expériences numériques en 2D et en 3D montrent que la propriété de transparence se conserve bien. L'influence de la matrice $\boldsymbol{\tau}$ complète sur la réflexion de ces conditions homogènes est négative : les réflexions sur les parois augmentent fortement. Cela montre que le contenu spectral de la matrice complète n'est pas bon et ne permet pas un bon décentrement de toutes les caractéristiques en 3D.

Le chapitre sur les conditions aux limites se conclut par des simulations sur un cas test industriel, qui est une tuyère modèle utilisée dans le cadre du projet européen *Clean Sky* [34]. Dans les cas sans écoulement, une comparaison très favorable est réalisée avec un code de propagation par éléments de frontière utilisant les équations d'Helmholtz, sur des modes acoustiques de plus en plus complexes. Un calcul avec écoulement montre des résultats prometteurs, même si la comparaison avec le code éléments frontières n'est alors plus possible.

1.4 Publications et communications

Plusieurs communications scientifiques ont été réalisées durant cette thèse. Elles sont listées ci-dessous.

Articles de revues

[21] Bissuel, A., Allaire, G., Daumas, L., Barré, S. et Rey, F. Linearized Navier-Stokes equations for Aeroacoustics using Stabilized Finite Elements : Boundary Conditions and Industrial Application to Aft-Fan Noise Propagation. *Computers and Fluids*, 2017. En cours de revue.

N.B. Cet article a été traduit pour composer la majeure partie du chapitre 6 de cette thèse.

Actes de conférences

[23] Bissuel, A., Allaire, G., Daumas, L. Mallet, M. et Chalot, F. Solving linear systems with multiple right-hand sides with GMRES : an application to aircraft design, *VII European Congress on Computational Methods in Applied Sciences and Engineering*. doi :10.7712/100016.2339.7593, `https://doi.org/10.7712/100016.2339.7593`.

Communication orale

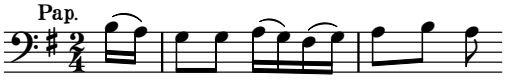
[22] Bissuel, A., Allaire, G., Daumas, L., Chalot, F., Barré, S. et Rey, F. Linearized Navier-Stokes for Aeroacoustics : Assessment of Aft Fan Noise Radiated from Business Jet Engine Nozzles, *Finite Elements in Flow 2017*.

Posters

Bissuel, A., Allaire, G., Daumas, L., Improving the convergence of the linearized Navier-Stokes equations, *Journée HPCC-DD*, 7 avril 2016, Université Paris 13 – Villetaneuse.

Bissuel, A., Allaire, G., Daumas, L., Use of the linearized Navier-Stokes equations for shape optimization, flutter and aeroacoustics, *Séminaire mécanique des fluides numérique*, 30-31 janvier 2017, Institut Henri Poincaré, Paris.

Bissuel, A., Allaire, G., Daumas, L., Barré, S. et Rey, F. Équations de Navier-Stokes linéarisées pour l'aéroacoustique : propagation du bruit de moteur d'un avion d'affaire, *Rencontres maths-industrie : acoustique numérique et traitement du signal audio*, 13 mars 2017, École Polytechnique, Palaiseau.



Mozart, *Die Zauberflöte*, Acte 1 sc. 4

Chapitre 2

Le code Aether

Le code AeTher (pour AeroThermodynamique) est utilisé pour résoudre les équations de Navier-Stokes pour un fluide compressibles exprimées en variables entropiques à l'aide de la méthode des éléments finis stabilisés par SUPG (pour *Streamline Upwind Petrov Galerkin*). La formulation élément fini permet l'utilisation de maillages non-structurés très adaptés à des configurations complexes [26]. L'utilisation des variables entropiques symétrise les équations de Navier-Stokes et permet d'obtenir des propriétés thermodynamiques [104, 82]. Cette formulation facilite l'ajout d'équations de chimie pour traiter les espèces réactives apparaissant dans des écoulements hypersoniques [28]. De nombreux modèles de turbulence (k-ε, Spalart-Allmaras, *etc*) sont disponibles pour les calculs stationnaires utilisant les équations RANS (pour *Reynolds Averaged Navier-Stokes*). Les calculs temporels en DES/LES (*Detached Eddy Simulation*/*Large Eddy Simulation*) sont possibles grâce à la méthode classique ZDES [41] (*Zonal Detached Eddy Simulation*), et également grâce à une approche originale utilisant la méthode VMS (*Variational MultiScale*) [98, 160]. Des fonctions de forme d'ordre élevé sont utilisables pour améliorer la convergence spatiale du schéma [121].

Les principaux ingrédients numériques du code, qui sont d'importance pour le reste de la thèse, sont détaillés dans ce chapitre.

Conventions de notation

Dans tout ce manuscrit, les quantités dont l'index est précédé d'une virgule indique, lorsque l'index désigne sans ambiguïté une dimension spatiale ou temporelle, la dérivée de cette quantité par rapport à la dimension indexée. Par exemple, $\mathbf{U}_{,i} = \frac{\partial \mathbf{U}}{\partial x_i}$ est la dérivée par rapport à la i$^{\text{e}}$ direction du vecteur $\mathbf{U}$. De même, $\mathbf{V}_{,t} = \frac{\partial \mathbf{V}}{\partial t}$ est la dérivée temporelle de $\mathbf{V}$.

La convention de sommation d'Einstein est employée. Ainsi, dans toute expression où deux quantités partagent le même index, une sommation implicite sur cet index est effectuée. Par exemple,

$$\widetilde{\mathbf{A}}_i \mathbf{V}_{,i} = \sum_{i=1}^{3} \widetilde{\mathbf{A}}_i \mathbf{V}_{,i} \,,$$

ou encore

$$\frac{\partial \xi_i}{\partial x_j}\frac{\partial \xi_i}{\partial x_k}\mathbf{A}_j\mathbf{A}_k = \sum_{i,j,k} \frac{\partial \xi_i}{\partial x_j}\frac{\partial \xi_i}{\partial x_k}\mathbf{A}_j\mathbf{A}_k \,.$$

La convention d'Einstein sera principalement utilisée lorsque les indices utilisés désignent des directions de l'espace. Il n'y aura donc pas de risque de confusion sur les bornes de la somme à effectuer. Par exemple, l'équation précédente, i, j et k sont des dimensions, et parcourent dans un cas tridimensionnel chacun l'ensemble $\{1, 2, 3\}$.

2.1 Les équations de Navier-Stokes

2.1.1 Les équations de Navier-Stokes sous forme conservative

Le système d'équations de Navier-Stokes décrit le comportement d'un fluide visqueux newtonien compressible. Il est composé de cinq équations (quatre en 2D) de conservation : une équation scalaire pour la masse, une équation vectorielle pour la quantité de mouvement et une équation scalaire pour l'énergie. Si l'on note $\mathbf{x} = (x_i)$ les coordonnées de l'espace (et i parcourt le nombre de dimensions), ρ la densité du fluide, ici l'air, $\mathbf{u} = (u_i)$ la vitesse de l'air et e_t son énergie totale, les équations de Navier-Stokes compressibles s'écrivent, en utilisant la convention de sommation d'Einstein

$$\left\{ \begin{array}{c} \dfrac{\partial \rho}{\partial t} + \dfrac{\partial \rho u_j}{\partial x_j} = 0 \\ \dfrac{\partial \rho u_i}{\partial t} + \dfrac{\partial \rho u_i u_j}{\partial x_j} + \dfrac{\partial p}{\partial x_i} = \dfrac{\partial}{\partial x_j}(2\mu S_{ij}) \\ \dfrac{\partial \rho e_t}{\partial t} + \dfrac{\partial \rho e_t u_j}{\partial x_j} + \dfrac{\partial p u_j}{\partial x_j} = \dfrac{\partial}{\partial x_j}(2\mu S_{ij} u_i) - \dfrac{\partial q_j}{\partial x_j} \,, \end{array} \right. \tag{2.1}$$

où le tenseur $\mathbf{S}$ est la partie déviatrice du tenseur des taux de déformation, donc $S_{ij} = 1/2(u_{i,j} + u_{j,i}) - 1/3 u_{k,k}\delta_{ij}$, $u_{i,j} = \frac{\partial u_i}{\partial x_j}$ et μ désigne la viscosité moléculaire dynamique. Le vecteur $\mathbf{q}$ est le flux de chaleur, calculé à l'aide de la loi de Fourier, $\mathbf{q} = -\kappa \nabla T$, où κ est la conductivité thermique du gaz, et T la température. Enfin, le système est fermé par une loi d'état. Pour les gaz parfaits, elle s'écrit $p/\rho = RT$, où R est la constante universelle des gaz parfait. On pourra consulter [61] et [143, 144] pour une présentation générale des propriétés mathématiques des équations d'Euler.

On définit le vecteur des variables conservatives $\mathbf{U}$ comme

$$\mathbf{U} = \rho \begin{pmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ e_t \end{pmatrix} . \tag{2.2}$$

Les équations de Navier-Stokes compressibles (2.1) se mettent sous forme vectorielle

$$\mathbf{U}_{,t} + \boldsymbol{\mathcal{F}}^{\text{Eul}}_{i,i} = \boldsymbol{\mathcal{F}}^{\text{Diff}}_{i,i} . \tag{2.3}$$

La notation indicielle $\mathbf{U}_{,t}$ désigne la dérivée partielle temporelle, *i.e.* $\frac{\partial \mathbf{U}}{\partial t}$ $\partial \mathbf{U}/\partial t$. De même, $\boldsymbol{\mathcal{F}}^{\text{Eul}}_{i,i} = \frac{\partial \boldsymbol{\mathcal{F}}^{\text{Eul}}_{i}}{\partial x_i}$

Les flux d'Euler $\boldsymbol{\mathcal{F}}^{\text{Eul}}_{i}$ et les flux diffusifs $\boldsymbol{\mathcal{F}}^{\text{Diff}}_{i}$ s'identifient depuis l'équation (2.1) :

$$\boldsymbol{\mathcal{F}}^{\text{Eul}}_{i} = \begin{pmatrix} \rho u_i \\ \rho u_i u_1 + p\delta_{1i} \\ \rho u_i u_2 + p\delta_{2i} \\ \rho u_i u_3 + p\delta_{3i} \\ \rho u_i e_t + p u_i \end{pmatrix} \qquad \text{et} \qquad \boldsymbol{\mathcal{F}}^{\text{Diff}}_{i} = \begin{pmatrix} 0 \\ 2\mu S_{1i} \\ 2\mu S_{2i} \\ 2\mu S_{3i} \\ 2\mu S_{ij} u_i - q_i \end{pmatrix} . \tag{2.4}$$

Si l'on note $\mathbf{A}_i$ la matrice jacobienne du flux d'Euler $\boldsymbol{\mathcal{F}}^{\text{Eul}}_{i}$, alors $\boldsymbol{\mathcal{F}}^{\text{Eul}}_{i,i} = \mathbf{A}_i \mathbf{U}_{,i}$. De même, on définit les matrices $\mathbf{K}_{ij}$ de diffusivité telles que $\mathbf{K}_{ij}\mathbf{U}_{,j} = \boldsymbol{\mathcal{F}}^{\text{Diff}}_{i}$. En réinjectant ces définitions dans (2.3), on obtient une écriture matricielle des équations de Navier-Stokes compressibles :

$$\mathbf{U}_{,t} + \mathbf{A}_i \mathbf{U}_{,i} = \left(\mathbf{K}_{ij}\mathbf{U}_{,j}\right)_{,i} . \tag{2.5}$$

Les matrices $\mathbf{A}_i$ et $\mathbf{K}_{ij}$ dépendent de $\mathbf{U}$. L'équation (2.5) est donc bien non linéaire.

2.1.2 Variables entropiques et symétrisation

L'utilisation des variables conservatives pose deux problèmes [104]. Premièrement, le vecteur $\mathbf{U}$ des variables conservatives n'est pas homogène dans ses unités. Ainsi la norme L^2 de $\mathbf{U}$ n'a aucun sens physique (l'expression $\rho^2(1 + \|\mathbf{u}\|^2 + e_t^2)$ est dimensionnellement fausse). Deuxièmement, les matrices $\mathbf{A}_i$ et la matrice $\mathbf{K} = (\mathbf{K}_{ij})$ définie par bloc n'ont aucune propriété intrinsèque intéressante.

Un changement de variables dit entropique est utilisé. Il se place dans le cadre théorique proposé par Harten [75] et Tadmor [154]. Soit la fonction d'entropie généralisée $\mathcal{H}(\mathbf{U})$ telle que définie par Hughes, Franca et Mallet [82] et Mallet [104] :

$$\mathcal{H}(\mathbf{U}) = -\rho s\,,$$

où s est l'entropie. Pour un gaz parfait divariant, l'entropie s'écrit $s = c_p \ln \frac{T}{T_0} - R \ln \frac{p}{p_0} + s_0$, où (p_0, T_0) définissent l'état de référence associé à l'entropie de référence s_0. Ici, c_p et c_v sont les chaleurs spécifiques respectivement à pression et à volume constant. On définit les variables $\mathbf{V}$ dites entropiques par

$$\mathbf{V} = \left(\frac{\partial \mathcal{H}}{\partial \mathbf{U}}\right)^T .$$

Pour un gaz parfait divariant, si l'on note $h = c_p T$ l'enthalpie massique, on exprime plus simplement le vecteur $\mathbf{V}$ des variables entropiques :

$$\mathbf{V} = \frac{1}{T} \begin{pmatrix} h - Ts - \frac{\|\mathbf{u}\|^2}{2} \\ u_1 \\ u_2 \\ u_3 \\ -1 \end{pmatrix} . \tag{2.6}$$

Soit $\widetilde{\mathbf{A}}_0$ la jacobienne de $\mathbf{U}$ par rapport à $\mathbf{V}$. Alors pour toute variable α, $\frac{\partial \mathbf{U}}{\partial \alpha} = \frac{\partial \mathbf{U}}{\partial \mathbf{V}} \frac{\partial \mathbf{V}}{\partial \alpha} = \widetilde{\mathbf{A}}_0 \frac{\partial \mathbf{V}}{\partial \alpha}$. On en déduit que les équations de Navier-Stokes sous forme matricielle (2.5) prennent la forme suivante avec les variables entropiques :

$$\widetilde{\mathbf{A}}_0 \mathbf{V}_{,t} + \widetilde{\mathbf{A}}_i \mathbf{V}_{,i} = \left(\widetilde{\mathbf{K}}_{ij} \mathbf{V}_{,j}\right)_{,i} . \tag{2.7}$$

Les tildes au dessus des matrices indiquent qu'elles s'appliquent à des vecteurs de variables entropiques. À l'aide de la matrice de passage $\widetilde{\mathbf{A}}_0$, on peut relier les expressions des matrices de diffusivité et de flux Euler des variables conservatives et entropiques :

$$\widetilde{\mathbf{A}}_i = \mathbf{A}_i \widetilde{\mathbf{A}}_0 \quad \text{et} \quad \widetilde{\mathbf{K}}_{ij} = \mathbf{K}_{i,j} \widetilde{\mathbf{A}}_0 \,.$$

Une propriété fondamentale du changement de variables entropiques est que les matrices $\widetilde{\mathbf{A}}_0$, $\widetilde{\mathbf{A}}_i$ et $\widetilde{\mathbf{K}} = \left(\widetilde{\mathbf{K}}_{ij}\right)$ sont symétriques, $\widetilde{\mathbf{A}}_0$ est définie positive, et $\widetilde{\mathbf{K}}$ est semi-définie positive. De plus, la formulation entropique (2.7) permet d'imposer l'inégalité thermodynamique de Clausius-Duhem [82, 104]. Enfin, l'utilisation des variables entropiques permet facilement d'ajouter au systèmes des équations de chimie [28].

Les variables entropiques nécessitent des changements de variables non-linéaires pour trouver les variables de travail naturelles en aérodynamique (pression, vitesse, etc). Cela complexifie la mise en place des conditions aux limites, détaillée dans le chapitre 6.

2.2 Éléments finis et stabilisation

La méthode des éléments finis permet de discrétiser des équations aux dérivées partielles mises sous forme faible. Ce cadre mathématique permet, pour certains types d'équations, de garantir l'existence et l'unicité des solutions. Il permet également d'établir des estimations d'erreur a priori et a posteriori. Quelques rappels de base sur les éléments finis et leur implémentation sont donnés dans cette section, qui n'a pas pour but d'être exhaustive. Les bases mathématiques peuvent se trouver dans des ouvrages de références, comme celui d'Ern et Guermond [50], Zienkiewicz et Taylor [162, 161] ou encore Hughes [79].

2.2.1 Rappels sur les éléments finis

Forme faible

On reprend les équations de Navier-Stokes non linéaires définies en (2.7), qu'on cherche à résoudre sur un domaine Ω dont la frontière Γ est suffisamment régulière. Pour simplifier le propos et l'établissement de la forme faible, on utilise la forme stationnaire de ces équations en enlevant le premier terme de (2.7). Pour les détails de l'intégration en temps dans les calculs instationnaires, on se réfèrera à [145, 146] et, pour un aperçu plus récent, à [98, 160]. On se place tout de suite dans le cas discret pour faciliter l'introduction de la stabilisation définie aux éléments [80]. On partitionne le domaine Ω en éléments Ω^e. L'exposant e désigne l'élément considéré. On utilise des éléments de Lagrange P1. Les fonctions considérées sont donc continues, et linéaires par morceaux sur chaque élément du maillage. Soit l'espace $\mathcal{V}$ de ces fonctions :

$$\mathcal{V} = \left\{ \mathbf{V} \mid \mathbf{V} \in \left(\mathcal{C}^0(\Omega)\right)^{n_{\text{dl}}}, \mathbf{V}_{|\Omega^e} \in (\mathcal{P}_1(\Omega^e))^{n_{\text{dl}}} \right\} . \tag{2.8}$$

Ici, n_{dl} désigne le nombre d'équations scalaires résolues en même temps en chaque point de l'espace. Il vaut donc cinq en 3D et quatre en 2D. L'espace $\mathcal{C}^0(\Omega)$ est celui des fonctions continues sur Ω et $\mathcal{P}_1(\Omega^e)$ l'ensemble des fonctions linéaire sur l'élément Ω^e.

Afin de garder des notations simples, et sans perte de généralité, on ne considèrera que des conditions de Dirichlet sur l'interface du domaine. L'utilisation des variables entropiques empêche un traitement simple des conditions aux limites. Des détails sont exposés dans le chapitre 6 et dans [145]. Soit $\mathbf{q}(V)$ une fonction non linéaire des variables entropiques qui calcule sur un point de l'interface la ou les grandeurs naturelles (comme la pression, la densité ou la vitesse) à imposer en ce point. La solution aux équations de Navier-Stokes non linéaires sera cherchée dans l'espace

$$\mathcal{V}_{\mathbf{g}} = \{ \mathbf{V} \in \mathcal{V} \mid \mathbf{q}(\mathbf{V}) = \mathbf{g} \text{ sur } \Gamma \} . \tag{2.9}$$

Les fonctions test seront prises dans un espace tangent :

$$\mathcal{V}_{\mathbf{0}} = \{\mathbf{V} \in \mathcal{V} \mid \mathbf{q}'(\mathbf{V}) = \mathbf{0} \text{ sur } \Gamma\} \,, \tag{2.10}$$

Pour obtenir la formulation faible, on multiplie à gauche l'équation de Navier-Stokes (2.7) par une fonction test $\mathbf{W} \in \mathcal{V}_{\mathbf{0}}$ quelconque et l'on intègre sur l'ensemble du domaine de calcul Ω :

$$\forall\, \mathbf{W} \in \mathcal{V}_{\mathbf{0}},\ \int_\Omega \mathbf{W} \cdot \left(\widetilde{\mathbf{A}}_i \mathbf{V}_{,i} - \left(\widetilde{\mathbf{K}}_{ij} \mathbf{V}_{,j} \right)_{,i} \right) \mathrm{d}\mathbf{x} = 0 \,. \tag{2.11}$$

On réalise une intégration par partie sur l'équation (2.11), afin de transférer un ordre de dérivation sur la fonction test $\mathbf{W}$

$$\begin{aligned} \forall\, \mathbf{W} \in \mathcal{V}_{\mathbf{0}},\ & \int_\Omega \mathbf{W}_{,i} \cdot \left(-\boldsymbol{\mathcal{F}}_i^{\mathrm{Eul}}(\mathbf{V}) + \boldsymbol{\mathcal{F}}_i^{\mathrm{Diff}}(\mathbf{V}) \right) \mathrm{d}\mathbf{x} \\ & + \int_\Gamma \mathbf{W} \cdot \left(\boldsymbol{\mathcal{F}}_i^{\mathrm{Eul}}(\mathbf{V}) - \boldsymbol{\mathcal{F}}_i^{\mathrm{Diff}}(\mathbf{V}) \right) n_i \, \mathrm{d}\Gamma = 0 \,. \end{aligned} \tag{2.12}$$

On rappelle que $\boldsymbol{\mathcal{F}}_i^{\mathrm{Diff}}(\mathbf{V})$ et $\boldsymbol{\mathcal{F}}_i^{\mathrm{Eul}}(\mathbf{V})$ sont les flux diffusifs et d'Euler exprimés en fonction des variables entropiques, $\Gamma = \partial\Omega$ est la frontière du domaine Ω, et $\mathbf{n} = (n_i)$ est le vecteur normal unitaire sortant à Ω défini en tout point de Γ.

Il est important de noter que l'équation (2.12) n'a pas été mise sous la forme habituelle de l'égalité entre une forme bilinéaire et une forme linéaire $a(w, v) = b(w)$. En effet, l'équation (2.12) est non linéaire, car les flux d'Euler et de diffusion dépendent non linéairement de $\mathbf{V}$. Ce problème non linéaire est résolu par une méthode de Newton (avec pseudo pas de temps, cf. [146, 145]. Ainsi, notons $\mathbf{E}(\mathbf{W}, \mathbf{V})$ le résidu non nul de (2.12) :

$$\begin{aligned} \mathbf{E}(\mathbf{W}, \mathbf{V}) = & \int_\Omega \mathbf{W}_{,i} \cdot \left(-\boldsymbol{\mathcal{F}}_i^{\mathrm{Eul}}(\mathbf{V}) + \boldsymbol{\mathcal{F}}_i^{\mathrm{Diff}}(\mathbf{V}) \right) \mathrm{d}\mathbf{x} \\ & + \int_\Gamma \mathbf{W} \cdot \left(\boldsymbol{\mathcal{F}}_i^{\mathrm{Eul}}(\mathbf{V}) - \boldsymbol{\mathcal{F}}_i^{\mathrm{Diff}}(\mathbf{V}) \right) n_i \, \mathrm{d}\Gamma \,. \end{aligned} \tag{2.13}$$

Le terme $\mathbf{E}(\mathbf{W}, \mathbf{V})$ dépend non-linéairement de $\mathbf{V}$ mais est linéaire en $\mathbf{W}$. Pour l'annuler, on utilise une méthode de Newton en cherchant un incrément de solution $\boldsymbol{\delta}\mathbf{V}$ tel que

$$\forall\, \mathbf{W} \in \mathcal{V}_{\mathbf{0}},\ \frac{\partial \mathbf{E}(\mathbf{W}, \mathbf{V})}{\partial \mathbf{V}} \boldsymbol{\delta}\mathbf{V} = -\mathbf{E}(\mathbf{W}, \mathbf{V}) \,. \tag{2.14}$$

On note que l'expression du produit de la jacobienne de $\mathbf{E}$ par $\boldsymbol{\delta}\mathbf{V}$ peut s'écrire de manière explicite, en se rappelant que $\widetilde{\mathbf{A}}_i$ est la jacobienne de $\boldsymbol{\mathcal{F}}_i^{\mathrm{Eul}}$ par rapport à $\mathbf{V}$ et que $\boldsymbol{\mathcal{F}}_i^{\mathrm{Diff}} = \widetilde{\mathbf{K}}_{ij} \mathbf{V}_{,j}$:

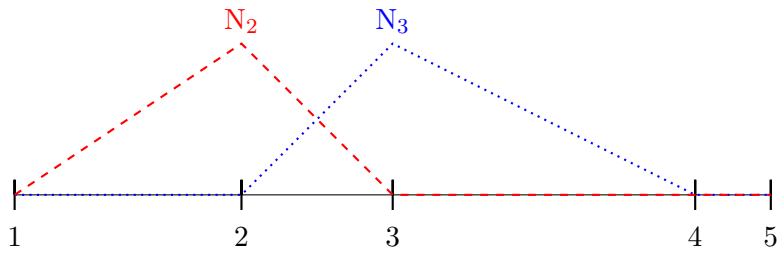


FIGURE 2.1 – Deux fonctions de forme linéaires sur un maillage 1D

$$\begin{aligned}
&\frac{\partial \mathbf{E}(\mathbf{W},\mathbf{V})}{\partial \mathbf{V}}\boldsymbol{\delta}\mathbf{V} = \\
&\int_\Omega \mathbf{W}_{,i} \cdot \left(-\widetilde{\mathbf{A}}_i \boldsymbol{\delta}\mathbf{V}_{,i} + \widetilde{\mathbf{K}}_{ij}\boldsymbol{\delta}\mathbf{V}_{,j} + \left(\frac{\partial \widetilde{\mathbf{K}}_{ij}}{\partial \mathbf{V}}\boldsymbol{\delta}\mathbf{V}\right)\mathbf{V}_{,j}\right) \mathrm{d}\mathbf{x} \\
+ &\int_\Gamma \mathbf{W} \cdot \left(\widetilde{\mathbf{A}}_i \boldsymbol{\delta}\mathbf{V}_{,i} - \widetilde{\mathbf{K}}_{ij}\boldsymbol{\delta}\mathbf{V}_{,j} - \left(\frac{\partial \widetilde{\mathbf{K}}_{ij}}{\partial \mathbf{V}}\boldsymbol{\delta}\mathbf{V}\right)\mathbf{V}_{,j}\right) n_i \,\mathrm{d}\Gamma \,.
\end{aligned} \tag{2.15}$$

Maillage, fonctions de forme et discrétisation

Les espaces discrets $\mathcal{V}_\mathbf{a}$ et $\mathcal{V}_\mathbf{0}$ sont définis sur le maillage du domaine Ω, partitionné en éléments Ω^e. Les fonctions de forme constituant la base qui génère les espaces $\mathcal{V}_\mathbf{a}$ et $\mathcal{V}_\mathbf{0}$ sont les fonctions N_a, continues et linéaires sur chaque élément, valant 1 sur le nœud a du maillage et 0 sur tous les autres nœuds. Ces fonctions de forme définissent l'élément fini de Lagrange P1. La figure 2.1 présente l'exemple de deux fonctions de forme sur un maillage à une dimension. Si $\mathbf{x}_b$ sont les coordonnées d'un nœud b du maillage, alors la fonction N_a vérifie la propriété fondamentale suivante :

$$\mathrm{N}_a(\mathbf{x}_b) = \delta_{ab} = \begin{cases} 1 & \text{si } a = b\,, \\ 0 & \text{sinon}\,. \end{cases} \tag{2.16}$$

La relation (2.16) montre qu'on peut exprimer directement toute fonction $\mathbf{V} \in \mathcal{V}$ dans la base des N_a :

$$\mathbf{V}(x) = \sum_{a=1}^{n} \mathrm{N}_a(x)\,\mathbf{V}(\mathbf{x}_a)\ . \tag{2.17}$$

On constate que résoudre le problème (2.14) revient à chercher $\boldsymbol{\delta}\mathbf{V} \in \mathcal{V}_\mathbf{0}$ tel que

$$\forall\, \mathbf{W} \in \mathcal{V}_\mathbf{0},\ \mathbf{a}(\mathbf{W}, \boldsymbol{\delta}\mathbf{V}) = \mathbf{b}(\mathbf{W})\ , \tag{2.18}$$

où $\mathbf{a}$ est bilinéaire et $\mathbf{b}$ est linéaire. On va chercher à écrire ce problème sous forme matricielle dans la base des fonctions de forme (N_a) de $\mathcal{V}$. Les vecteurs $\mathbf{W}$ et $\boldsymbol{\delta}\mathbf{V}$ sont exprimés dans cette base :

$$\boldsymbol{\delta}\mathbf{V} = \sum_{a=1}^{n} \mathrm{N}_a \boldsymbol{\delta}\mathbf{V}_a$$
$$\mathbf{W} = \sum_{a=1}^{n} \mathrm{N}_a \mathbf{W}_a \,,$$

où $\boldsymbol{\delta}\mathbf{V}_a = \boldsymbol{\delta}\mathbf{V}(\mathbf{x}_a)$ est la valeur de $\boldsymbol{\delta}\mathbf{V}$ au sommet a. On remarque que $\boldsymbol{\delta}\mathbf{V}_a$ est un vecteur des cinq inconnues en 3D (quatre en 2D) décrites à la section 2.1.2. Soient $(\mathbf{e}^m)$ les vecteurs de la base canonique de $\mathbb{R}^5$. Alors, en notant δV_a^m la m-ième composante de $\boldsymbol{\delta}\mathbf{V}_a$, on obtient avec une sommation implicite sur m :

$$\boldsymbol{\delta}\mathbf{V} = \sum_{a=1}^{n} \mathrm{N}_a \mathbf{e}^m \delta V_a^m$$
$$\mathbf{W} = \sum_{a=1}^{n} \mathrm{N}_a \mathbf{e}^m W_a^m \,.$$

On définit une matrice $\mathbf{A}$ et un vecteur $\mathbf{b}$ par leur coefficients

$$\mathbf{A}_{kl} = \mathbf{a}(\mathrm{N}_k \mathbf{e}^m, \mathrm{N}_l \mathbf{e}^p) \,, \quad b_k^m = \mathbf{b}(\mathbf{e}^m \mathrm{N}_k) \,. \tag{2.19}$$

On notera que $\mathbf{A}_{kl}$ est une matrice de dimension 5×5 (4×4 en 2D) dont les coefficients sont donnés par $A_{kl}^{mp} = \mathbf{a}(\mathrm{N}_k \mathbf{e}^m, \mathrm{N}_l \mathbf{e}^p)$. La matrice globale $\mathbf{A}$ est donc définie par blocs. Le problème (2.18) s'écrit sous la forme matricielle suivante, où $\boldsymbol{\delta}\mathbf{V}$ est recherchée dans $\mathcal{V}_\mathbf{0}$:

$$\forall\, \mathbf{W} \in \mathcal{V}_\mathbf{0},\ \mathbf{W}^T \mathbf{A} \boldsymbol{\delta}\mathbf{V} = \mathbf{W}^T \mathbf{b} \,. \tag{2.20}$$

Le passage de l'espace $\mathcal{V}$ à $\mathcal{V}_\mathbf{0}$, c'est-à-dire l'imposition des conditions aux limites de Dirichlet homogènes, est effectué par modification de la matrice $\mathbf{A}$ et du vecteur $\mathbf{b}$ selon la procédure décrite dans le chapitre 6.

L'équation (2.20) est valable pour tout $\mathbf{W} \in \mathcal{V}$ et donc on se ramène à un système linéaire :

$$\mathbf{A} \boldsymbol{\delta}\mathbf{V} = \mathbf{b} \,. \tag{2.21}$$

Intégration par élément et assemblage

La matrice $\mathbf{A}$ et le vecteur résidu $\mathbf{b}$ sont définis comme une somme d'intégrales volumiques et surfaciques sur le domaine Ω et sa frontière Γ. Ces intégrales peuvent être décomposées comme la somme d'intégrales sur le volume des éléments (et leur frontière quand ceux-ci sont au bord). Notons

$\mathcal{B}_a$ l'ensemble des éléments du maillage partageant le nœud a. Les fonctions de formes N_a sont définies sur le support compact des éléments de $\mathcal{B}_a$. Alors le terme A_{kl} de la matrice défini par l'équation (2.19) s'écrit comme une somme sur les éléments :

$$\mathbf{A}_{kl} = \sum_{\Omega^e \in \mathcal{B}_k \cap \mathcal{B}_l} \mathbf{a}\Big(\mathbf{e}^m \mathrm{N}_{k|\Omega^e}, \mathbf{e}^p \mathrm{N}_{l|\Omega^e}\Big) \ . \tag{2.22}$$

En effet, l'intégrale d'un produit de deux fonctions de forme (ou de leur dérivée, ou d'une combinaison) peut être non nulle si et seulement si l'intersection de leur support compact est non nulle, ce qui est possible si et seulement si les deux nœuds associés à ces fonctions de forme appartiennent à (au moins) un même élément.

Cela justifie le calcul de matrices élémentaires $\mathbf{A}^e$ (et de résidus élémentaires $\mathbf{b}^e$) de l'élément Ω^e définies par

$$\mathbf{A}^e_{k'l'} = \mathbf{a}\Big(\mathbf{e}^m \mathrm{N}_{k'|\Omega^e}, \mathbf{e}^p \mathrm{N}_{l'|\Omega^e}\Big) \ . \tag{2.23}$$

Les indices k' et l' sont donnés en numérotation locale, *i.e.* $k' \in \{1; 2; 3\}$ pour un élément P1 linéaire en 2D. Ils correspondent aux nœuds globaux k et l.

L'opération d'assemblage de la matrice globale se fait en sommant les contributions locales appartenant aux matrices élémentaires :

$$\mathbf{A}_{kl} = \sum_{\Omega^e \in \mathcal{B}_k \cap \mathcal{B}_l} \mathbf{A}^e_{k'l'} \ . \tag{2.24}$$

La matrice $\mathbf{A}$ est creuse, c'est-à-dire qu'il est plus efficace de stocker ses éléments non nuls plutôt que l'intégralité de ses éléments. On note $\mathcal{S}$ son masque, qui est l'index de ses éléments blocs non nuls :

$$\mathcal{S} = \{(i, j) \, , \mathbf{A}_{ij} \neq 0\} \ . \tag{2.25}$$

Le masque de la matrice peut être interprété sous forme de graphe [135]. On définit le graphe associé à la matrice comme suit : si $(i, j) \in \mathcal{S}$, alors on relie les sommets i et j du graphe par une arête. Pour les éléments finis de Lagrange P1, le graphe associé à la matrice et le maillage sont identiques.

Élément de référence

Tous les termes du système linéaire sont calculés à partir d'intégrales effectuées sur les éléments. Pour simplifier l'intégration, un élément de référence Ω^{ref} est défini. Tous les éléments du maillage seront obtenus à partir de cet élément de référence par une transformation géométrique. On note $\boldsymbol{\xi}$ le vecteur des coordonnées de l'espace de l'élément de référence. L'élément de référence en 2D est présenté sur la figure 2.2.

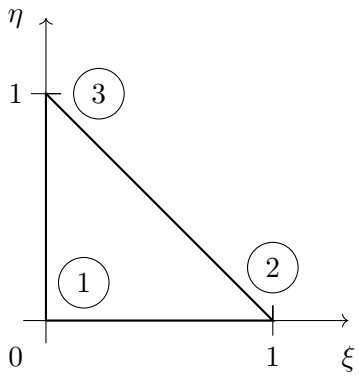


FIGURE 2.2 – Élément de référence en 2D dans le système de coordonnées (ξ, η). Les numéros locaux de l'élément sont entourés.

Considérons un élément Ω^e. Notons $\mathbf{x}_a$ les coordonnées de ses nœuds a. Notons $\mathbf{x}^e(\boldsymbol{\xi})$ la transformation géométrique de l'élément de référence vers l'élément considéré. On prend comme hypothèse que les éléments sont isoparamétriques, c'est-à-dire que cette transformation est prise dans l'espace des fonctions de forme N_a. Ainsi, pour tout point de coordonnées $\mathbf{x}$ dans l'élément Ω^e et de coordonnées $\boldsymbol{\xi}$ dans Ω^{ref}, on a

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_{a \in \Omega^e} \mathrm{N}_a(\boldsymbol{\xi})\, \mathbf{x}_a \,, \tag{2.26}$$

où les fonctions de forme N_a sont définies sur l'élément de référence, et prennent donc comme argument les coordonnées $\boldsymbol{\xi}$. De même, l'interpolation d'une variable dans un élément se fait à partir des fonctions de forme définies sur Ω^{ref} :

$$\mathbf{V} = \sum_{a \in \Omega^e} \mathrm{N}_a(\boldsymbol{\xi})\, \mathbf{V}_a \,.$$

On peut également effectuer les intégrales directement dans l'élément de référence. Le changement de variable se note

$$\int_{\Omega^e} \mathbf{f}(\mathbf{x})\, \mathrm{d}\Omega^e = \int_{\Omega^{\mathrm{ref}}} \mathbf{f}(\boldsymbol{\xi}) \left| \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \right| \mathrm{d}\Omega^{\mathrm{ref}} \,, \tag{2.27}$$

où $\left|\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}\right|$ désigne le déterminant de la jacobienne de la transformation vers l'élément courant.

On note $\mathbf{J} = \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}}$ la jacobienne de la transformation vers l'élément de référence (*i.e.* $\boldsymbol{\xi}(\mathbf{x})$). Pour l'obtenir, il suffit d'inverser la jacobienne de la transformation vers l'élément courant qui s'écrit

$$\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} = \sum_{a \in \Omega^e} \frac{\partial \mathrm{N}_a}{\partial \boldsymbol{\xi}} \mathbf{x}_a \,.$$

Cette jacobienne est de petite taille 3×3 et est donc inversible analytiquement. Enfin, la jacobienne $\mathbf{J}$ de la transformation est utile pour transformer les gradients de fonctions de forme :

$$\frac{\partial \mathrm{N}_a(\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial \mathrm{N}_a(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}} = \frac{\partial \mathrm{N}_a(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \mathbf{J}^{\,}.$$

Enfin, les intégrales sur les éléments peuvent être rapidement approchées par des formules de quadrature. Pour plus de détails, on pourra consulter [50, 162].

2.2.2 Stabilisation des éléments finis

La discrétisation présentée dans le chapitre précédent utilise le même espace pour les fonctions de forme et les fonctions tests. Cette discrétisation est dite de Galerkin standard, et est instable pour les équations à advection dominante, comme l'on montré Brooks et Hughes dans [24] à l'aide d'un exemple 1D que nous reprendrons dans le chapitre 5.

Une solution est d'utiliser des fonctions tests prises dans un autre espace que les fonctions de forme. La formulation SUPG (*Streamline Upwind Petrov-Galerkin*) introduite par Brooks et Hughes [24] modifie les fonctions tests $\mathbf{W}$ par l'ajout d'un terme $\boldsymbol{\tau}\widetilde{\mathbf{A}}_i\mathbf{W}_{,i}$. La matrice $\boldsymbol{\tau}$, dite de stabilisation, est définie aux éléments, et doit être symétrique définie positive. Les équations stabilisées sont de la forme, si l'on désigne par $\mathcal{L} : \mathbf{V} \mapsto \widetilde{\mathbf{A}}_{,i}\mathbf{V}_i + \left(\widetilde{\mathbf{K}}_{ij}\mathbf{V}_{,j}\right)_{,i}$ l'opérateur associées aux équations de Navier-Stokes :

$$\sum_{\Omega^e} \int_{\Omega^e} \left(\mathbf{W} + \boldsymbol{\tau}\widetilde{\mathbf{A}}_{,i}\mathbf{W}_{,i}\right) \cdot \mathcal{L}\mathbf{V} \mathrm{d}\mathbf{x} = \mathbf{0}\,. \tag{2.28}$$

La formulation GLS (*Galerkin Least Squares*) introduite par Hughes, Franca et Hubert [81] consiste à inclure dans la modification des fonctions de test le terme visqueux. Ainsi les fonctions de pondération seront de la forme $\mathbf{W} + \boldsymbol{\tau}\mathcal{L}(\mathbf{W})$, où l'opérateur Navier-Stokes stationnaire est noté $\mathcal{L} : \mathbf{V} \mapsto \widetilde{\mathbf{A}}_i\mathbf{V}_{,i} - \left(\widetilde{\mathbf{K}}_{ij}\mathbf{V}_{,j}\right)_{,i}$. La formulation GLS tire son nom d'une interprétation en termes de moindres carrés [53]. Elle peut s'écrire

$$\sum_{\Omega^e} \int_{\Omega^e} (\mathbf{W} + \boldsymbol{\tau}\mathcal{L}\mathbf{W}) \cdot \mathcal{L}\mathbf{V} \mathrm{d}\mathbf{x} = \mathbf{0}\,.$$

Cette formulation pose problème pour des éléments dont les fonctions de forme sont linéaires. En effet, le terme de diffusion croisé

$$\int \left(\widetilde{\mathbf{K}}_{ij}\mathbf{W}_{,j}\right)_{,i}^T \boldsymbol{\tau} \left(\widetilde{\mathbf{K}}_{ij}\mathbf{V}_{,j}\right)_{,i} \mathrm{d}\mathbf{x}\,,$$

n'est pas calculable car les dérivées secondes des fonctions de forme sont nulles. La formulation GLS se réduit donc automatiquement à la formulation SUPG qui ajoute seulement le terme de convection dans la stabilisation.

Ces deux formulations sont consistantes, puisque le terme de stabilisation rajouté est multiplié par le résidu. Lorsque la solution converge, l'influence du terme de stabilisation tend vers zéro.

La matrice de stabilisation $\boldsymbol{\tau}$ est homogène à l'inverse d'un temps. Pour les équations d'Euler, elle s'écrit sous la forme [104] :

$$\boldsymbol{\tau} = \widetilde{\mathbf{A}}_0^{-1} \left(\frac{\partial \xi_i}{\partial x_j} \frac{\partial \xi_i}{\partial x_k} \mathbf{A}_j \mathbf{A}_k \right)^{-\frac{1}{2}} . \tag{2.29}$$

Le caractère symétrique défini positif de cette matrice est prouvé dans [104], et sera redémontré dans le chapitre 5.

2.3 Les équations de Navier-Stokes linéarisées

2.3.1 Notation matricielle

Comme on l'a vu précédemment, l'approche retenue pour résoudre les équations de Navier-Stokes linéarisées est d'utiliser le gradient discret du schéma non-linéaire, par rapport aux variables entropiques pour obtenir la matrice du système linéaire et par rapport aux coordonnées pour le second membre. Ce processus est très simple à expliquer à fréquence nulle. Dans ce cas, le gradient obtenu est équivalent à celui apporté par une méthode de différences finies.

Les équations linéarisées à fréquence nulle

On dispose d'un solveur permettant de résoudre les équations de Navier-Stokes stationnaires qui sont non-linéaires. Son principe est de minimiser le résidu noté $\mathbf{E}(\mathbf{V}, \mathbf{x})$, introduit dans la section précédente, qui dépend des variables entropiques $\mathbf{V}$ et des coordonnées des sommets du maillage $\mathbf{x}$. Une solution approchée $\mathbf{V}_0$ des équations de Navier-Stokes sur un maillage défini par des coordonnées $\mathbf{x}_0$ est trouvée lorsque $\mathbf{E}(\mathbf{V}_0, \mathbf{x}_0) = \mathbf{0}$.

On souhaite connaître l'impact $\boldsymbol{\delta}\mathbf{V}$ sur la solution d'une perturbation des coordonnées $\boldsymbol{\delta}\mathbf{x}$. On résout donc le problème

$$\mathbf{E}(\mathbf{V}_0 + \boldsymbol{\delta}\mathbf{V}, \mathbf{x}_0 + \boldsymbol{\delta}\mathbf{x}) = \mathbf{0} . \tag{2.30}$$

On effectue un développement limité à l'ordre 1 sur l'équation (2.30) pour obtenir

$$\mathbf{E}(\mathbf{V}_0 + \boldsymbol{\delta}\mathbf{V}, \mathbf{x}_0 + \boldsymbol{\delta}\mathbf{x}) = \mathbf{0} \approx \underbrace{\mathbf{E}(\mathbf{V}_0, \mathbf{x}_0)}_{=\mathbf{0}} + \frac{\partial \mathbf{E}}{\partial \mathbf{V}} \boldsymbol{\delta}\mathbf{V} + \frac{\partial \mathbf{E}}{\partial \mathbf{x}} \boldsymbol{\delta}\mathbf{x} .$$

On obtient donc le système linéaire suivant

$$\frac{\partial \mathbf{E}}{\partial \mathbf{V}} \boldsymbol{\delta}\mathbf{V} = -\frac{\partial \mathbf{E}}{\partial \mathbf{x}} \boldsymbol{\delta}\mathbf{x}\,. \tag{2.31}$$

Dans le reste du manuscrit, on notera ce système $\mathbf{Ay} = \mathbf{b}$, où par identification $\mathbf{A} = \frac{\partial \mathbf{E}}{\partial \mathbf{V}}$, la solution $\mathbf{y} = \boldsymbol{\delta}\mathbf{V}$ et le second membre $\mathbf{b} = -\frac{\partial \mathbf{E}}{\partial \mathbf{x}} \boldsymbol{\delta}\mathbf{x}$. On obtient bien une variation de variables aérodynamiques $\boldsymbol{\delta}\mathbf{V}$ à partir d'une variation de maillage $\boldsymbol{\delta}\mathbf{x}$.

Plusieurs remarques sont à formuler. La première est que la formule (2.31) tient uniquement si $\mathbf{E}(\mathbf{V}_0, \mathbf{x}_0) = \mathbf{0}$. Si la solution de référence $\mathbf{V}_0$ est mal calculée, à savoir que $\mathbf{E}(\mathbf{V}_0, \mathbf{x}_0)$ ne peut plus être considéré comme très petit, par manque d'itérations pseudo-temporelle ou de modélisation déficiente, alors l'équation (2.31) ne sera plus vérifiée.

Le déplacement $\boldsymbol{\delta}$ des nœuds du maillage est issu d'une propagation dans le volume d'un déplacement de la peau de l'avion. Pour l'aéroélasticité, le mouvement issu d'un mode structurel est défini à la peau, et dans le cas de l'optimisation de forme utilisant les gradients directs, les variables de définition de la forme définissent un déplacement à la peau. La propagation dans le volume de ce déplacement est effectuée par un opérateur linéarisé de déformation de maillage, qui utilise un opérateur Laplacien pondéré par l'inverse du volume afin de ne pas inverser les mailles très petites de la couche limite.

Notons que l'un des avantages de l'approche linéarisée est que $\boldsymbol{\delta}\mathbf{x}$ n'a pas a être une variation admissible du maillage, c'est-à-dire que $\mathbf{x}_0 + \boldsymbol{\delta}\mathbf{x}$ peut ne pas être un maillage bien défini et posséder des éléments retournés ou avoir des nœuds à l'extérieur du domaine. Cela permet une gestion plus facile des intersections géométriques par exemple.

Enfin, le gradient $\boldsymbol{\delta}\mathbf{V}$ peut se calculer par différences finies, à des fins de comparaison ou de validation. Soit $\mathbf{f}$ la fonction associant à des coordonnées $\mathbf{x}$ la solution $\mathbf{V}$ des équations de Navier-Stokes stationnaires. Alors $\mathbf{E}(\mathbf{f}(\mathbf{x}), \mathbf{x}) = \mathbf{0}$. La différence finie décentrée d'ordre un de $\mathbf{f}$ donne

$$\boldsymbol{\delta}\mathbf{V} \approx \frac{\mathbf{f}(\mathbf{x}_0 + h\boldsymbol{\delta}\mathbf{x}) - \mathbf{f}(\mathbf{x})}{h} \tag{2.32}$$

pour un paramètre scalaire h petit. Cela correspond à effectuer deux résolutions des équations de Navier-Stokes, l'une sur le maillage déformé $\mathbf{x}_0 + h\boldsymbol{\delta}\mathbf{x}$ et l'autre sur le maillage initial, et à effectuer la différence des solutions que l'on renorme. Trouver le paramètre h (ou la norme du déplacement $\boldsymbol{\delta}\mathbf{x}$) suffisamment petit pour que $\mathbf{f}$ puisse être considérée comme linéaire et suffisamment grand pour que la variation de $\mathbf{f}$ soit significative et non noyée dans le bruit numérique est un art délicat qui nécessite une connaissance de l'aérodynamique et des performances numériques du code de calcul. La différence finie centrée d'ordre deux est définie par

$$\boldsymbol{\delta}\mathbf{V} \approx \frac{\mathbf{f}(\mathbf{x}_0 + h\boldsymbol{\delta}\mathbf{x}) - \mathbf{f}(\mathbf{x}_0 - h\boldsymbol{\delta}\mathbf{x})}{2h} \tag{2.33}$$

La différence finie d'ordre deux affiche une précision en $\mathrm{O}(h^2)$ (résultat que l'on trouve avec un développement limité de f) contrairement à la différence finie d'ordre 1 qui a une précision en $\mathrm{O}(h)$, mais nécessite deux calculs non-linéaires (en plus de celui pour la configuration de référence).

Les équations linéarisées fréquentielles

Pour les applications d'aéroélasticité et d'aéroacoustique, une solution complexe de pulsation ω est recherchée. Pour la trouver, il faut repartir d'un résidu non-linéaire $\mathbf{E}$ qui dépend encore de la solution $\mathbf{V}$ et des coordonnées du maillage $\mathbf{x}$, mais également de la variation temporelle de la solution $\dot{\mathbf{V}}$ et de la vitesse de déplacement du maillage $\dot{\mathbf{x}}$: $\mathbf{E}\Big(\mathbf{V}, \dot{\mathbf{V}}, \mathbf{x}, \dot{\mathbf{x}}\Big)$. La solution de référence est la même :

$$\mathbf{E}(\mathbf{V}_0, \mathbf{0}, \mathbf{x}_0, \mathbf{0}) = \mathbf{0}\,.$$

Cette fois-ci, la perturbation de maillage induit une vitesse de maillage $\boldsymbol{\delta}\dot{\mathbf{x}}$, et également une perturbation de $\dot{\mathbf{V}}$:

$$\mathbf{E}\Big(\mathbf{V}_0 + \boldsymbol{\delta}\mathbf{V}, \boldsymbol{\delta}\dot{\mathbf{V}}, \mathbf{x}_0 + \boldsymbol{\delta}\mathbf{x}, \boldsymbol{\delta}\dot{\mathbf{x}}\Big) = \mathbf{0}\,.$$

Un développement limité du premier ordre donne :

$$\begin{aligned}
&\mathbf{E}\Big(\mathbf{V}_0 + \boldsymbol{\delta}\mathbf{V}, \boldsymbol{\delta}\dot{\mathbf{V}}, \mathbf{x}_0 + \boldsymbol{\delta}\mathbf{x}, \boldsymbol{\delta}\dot{\mathbf{x}}\Big) = \mathbf{0} \\
&\approx \underbrace{\mathbf{E}(\mathbf{V}_0, \mathbf{0}, \mathbf{x}_0, \mathbf{0})}_{=\mathbf{0}} + \frac{\partial \mathbf{E}}{\partial \mathbf{V}}\boldsymbol{\delta}\mathbf{V} + \frac{\partial \mathbf{E}}{\partial \dot{\mathbf{V}}}\boldsymbol{\delta}\dot{\mathbf{V}} + \frac{\partial \mathbf{E}}{\partial \mathbf{x}}\boldsymbol{\delta}\mathbf{x} + \frac{\partial \mathbf{E}}{\partial \dot{\mathbf{x}}}\boldsymbol{\delta}\dot{\mathbf{x}}\,.
\end{aligned} \tag{2.34}$$

Or la perturbation de maillage est harmonique :

$$\boldsymbol{\delta}\dot{\mathbf{x}} = \mathrm{j}\omega\boldsymbol{\delta}\mathbf{x}\,.$$

De même, la perturbation de la variation temporelle de la solution est harmonique :

$$\boldsymbol{\delta}\dot{\mathbf{V}} = \mathrm{j}\omega\boldsymbol{\delta}\mathbf{V}\,.$$

On peut donc simplifier l'équation (2.34), ce qui donne

$$\left(\frac{\partial \mathbf{E}}{\partial \mathbf{V}} + \mathrm{j}\omega\frac{\partial \mathbf{E}}{\partial \dot{\mathbf{V}}}\right)\boldsymbol{\delta}\mathbf{V} = -\left(\frac{\partial \mathbf{E}}{\partial \mathbf{x}} + \mathrm{j}\omega\frac{\partial \mathbf{E}}{\partial \dot{\mathbf{x}}}\right)\boldsymbol{\delta}\mathbf{x}\,. \tag{2.35}$$

Le terme $\frac{\partial \mathbf{E}}{\partial \dot{\mathbf{V}}}$ provient de l'intégration sur les éléments de $\widetilde{\mathbf{A}}_0\mathbf{I}$ (premier terme de l'équation (2.7)). La dépendance du résidu à la vitesse de maillage se calcule par une formulation ALE (pour *Arbitrary Lagrangian Eulerian*) des équations de Navier-Stokes [97].

2.3.2 Différenciation automatique

Dans le code AeTher, les termes des jacobiennes comme $\frac{\partial \mathbf{E}}{\partial \mathbf{V}}$ sont issus de la différenciation automatique du code Navier-Stokes. La différenciation automatique permet d'obtenir par calcul formel la dérivée d'une fonction et ainsi de s'affranchir des problèmes de précision numérique liés à l'utilisation des différences finies. D'une manière générale, une fonction informatique est décomposée en opérations arithmétiques élémentaires dont la dérivée est connue (par exemple $(uv)' = u'v + v'u$).

Soit une fonction informatique $f(\mathbf{X}, \mathbf{Y})$, où $\mathbf{X}$ et $\mathbf{Y}$ sont les vecteurs respectivement d'entrée et de sortie de la fonction.

Deux types de dérivées sont recherchés : l'une par rapport aux entrées de f et l'autre par rapport à ses sorties. La première dérivée est appelée dérivée tangente ou directe, la deuxième, dérivée inverse, dérivée mode rétrograde, dérivée mode adjoint, ou encore simplement gradient.

Une notation plus mathématique est de considérer $\mathbf{f}(\mathbf{X}) = \mathbf{Y}$, où $\mathbf{f}$ est un vecteur de fonctions scalaires (f_i). On note $\frac{\partial \mathbf{f}}{\partial \mathbf{X}}$ la jacobienne de $\mathbf{f}$.

La dérivée dite directe ou tangente consiste à évaluer la différentielle de $\mathbf{f}$ dans la direction $\boldsymbol{\delta}\mathbf{X}$. Cela correspond au calcul de $\frac{\partial \mathbf{f}}{\partial \mathbf{X}}\boldsymbol{\delta}\mathbf{X}$. Cela permet de calculer la sensibilité de la sortie $\mathbf{Y}$ par rapport à une variation d'entrée $\boldsymbol{\delta}\mathbf{X}$ autour d'une entrée $\mathbf{X}$. Le nom direct vient de la méthode de calcul d'une telle dérivée : la fonction f est décomposée en opérations mathématiques (ou informatiques) élémentaires. La dérivée directe est calculée par la règle de dérivation des fonctions composées [76].

La dérivée inverse ou gradient évalue à sortie constante la sensibilité des entrées par rapport à une variation des sorties $\boldsymbol{\delta}\mathbf{Y}$. Cela revient à l'évaluation de $\frac{\partial \mathbf{f}}{\partial \mathbf{X}}^T \boldsymbol{\delta}\mathbf{Y}$. La méthode de calcul inverse est plus complexe. Comme cette dérivée est évaluée à sortie $\mathbf{Y}$ constante, il faut d'abord calculer la sortie de f, puis calculer pour chaque opération élémentaire la sensibilité de son entrée par rapport à une variation de sa sortie. Cette variation de sortie a été calculée en combinant les variations d'entrées pour les opérations élémentaires suivantes. Ainsi, ce calcul de dérivée se réalise dans le sens inverse de l'exécution normale du programme f, ce qui justifie le nom d'inverse ou d'adjoint. Pour plus de détails, on pourra également consulter [76]. La figure 2.3 schématise pour une fonction mathématique simple $f(x, y, z) = xy + \sin z$ le calcul de f au point $(x = -1/2, y = \sqrt{2}, z = \pi/4)$ puis de la dérivée directe (ou directionnelle) suivant $(x' = 0, y' = 2, z' = \sqrt{2})$ et enfin du gradient pour $\dot{f} = 1$. On remarquera que le calcul des valeurs de la dérivée inverse nécessite, pour une opération donnée, les valeurs d'entrée de cette opération.

Enfin, notons que si la variation d'entrée $\boldsymbol{\delta}\mathbf{X}$ est le vecteur canonique $\mathbf{e}_j$ alors la méthode directe calcule $\frac{\partial \mathbf{f}}{\partial \mathbf{X}}\mathbf{e}_j = \frac{\partial f_j}{\partial \mathbf{X}}$ qui est la j$^{\text{e}}$ colonne de la jacobienne $\frac{\partial \mathbf{f}}{\partial \mathbf{X}}$ de $\mathbf{f}$. De même, si la variation de sortie $\boldsymbol{\delta}\mathbf{Y}$ est le i$^{\text{e}}$ vecteur canonique $\mathbf{e}_i$, la méthode inverse calcule $\frac{\partial \mathbf{f}}{\partial \mathbf{X}}^T \mathbf{e}_i = \frac{\partial \mathbf{f}}{\partial X_i}^T$ qui est la i$^{\text{e}}$ ligne (transposée) de la jacobienne de $\mathbf{f}$.

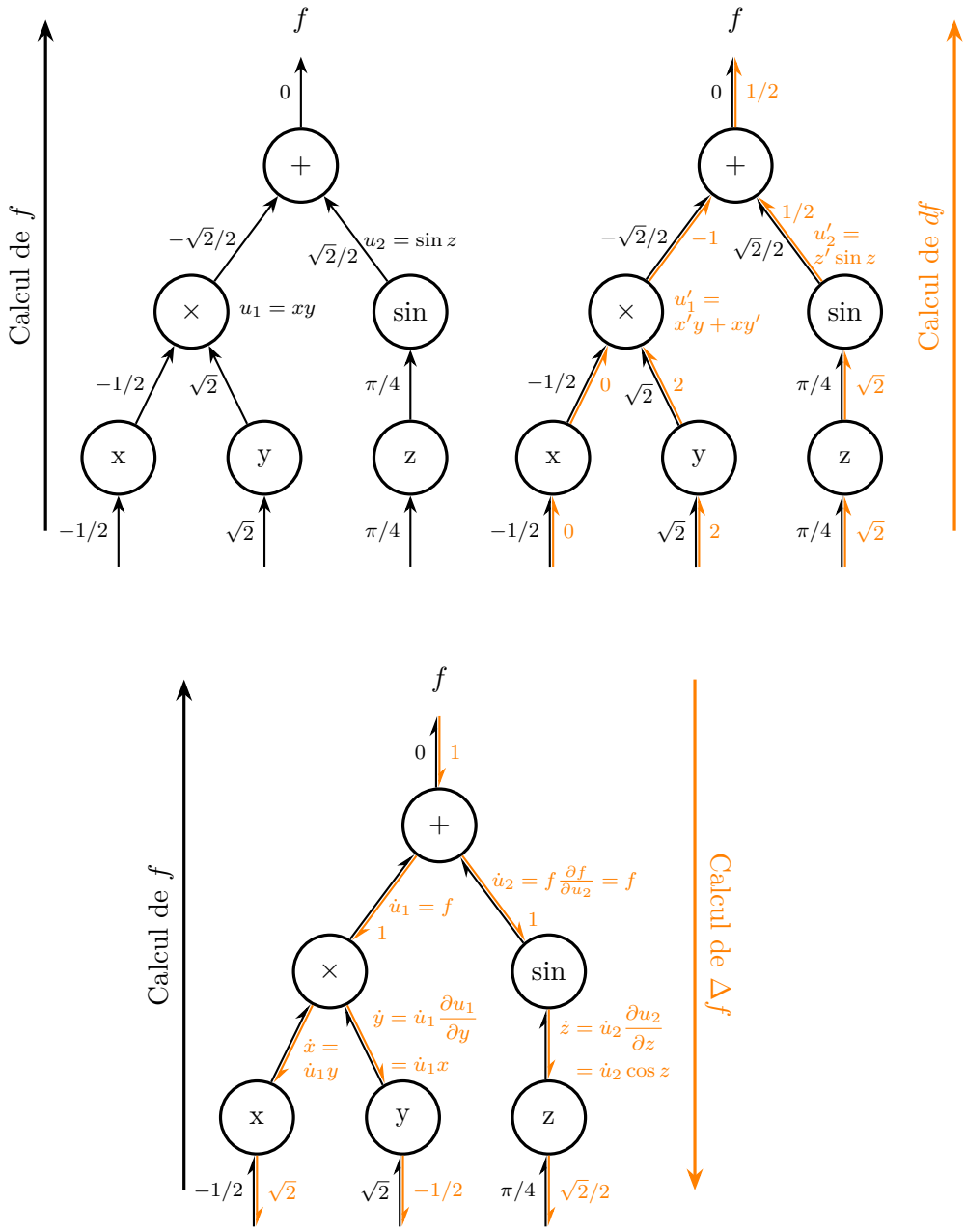


FIGURE 2.3 – En haut, calcul de $f(x, y, z) = xy + \sin z$ à gauche et calcul de la dérivée directionnelle de f à droite. En bas, calcul du gradient de f. La notation $\dot{x}$ réfère à la dérivée inverse de la variable d'entrée x. On consultera [76] pour des explications plus détaillées.

Deux grandes techniques existent pour la méthode de calcul de ces dérivées. Pour les langages qui la supportent (notamment le C++), la surcharge d'opérateur permet de transformer les opérateurs mathématiques des opérations élémentaires composant f en des opérateurs qui s'appliquent à la fois sur les données originales et sur les valeurs dérivées. Pour les autres langages, une méthode est de réécrire automatiquement le code source après analyse. C'est la seule méthode pour calculer les gradients en mode inverse, puisque la surcharge d'opérateur ne permet pas de parcourir le programme en sens inverse. Le logiciel TAPENADE développé par l'INRIA [76], qui utilise la modification de code source, est utilisé pour la linéarisation du code AeTher.

2.4 Parallélisation

Les problèmes d'aérodynamique couramment résolus chez Dassault Aviation sont trop volumineux pour tenir sur un seul processeur. Il est alors nécessaire de répartir le problème sur plusieurs processeurs, afin de le résoudre en parallèle.

Le maillage est réparti en effectuant une partition sur les éléments. Pour cette raison, un tel découpage est dit *aux éléments*. Cela implique que les nœuds de l'interface entre les sous-domaines sont dupliqués, comme on peut le voir sur la figure 2.5. Les arêtes (et les faces en 3D) sont également dupliquées. Ce découpage est appelé à recouvrement minimal, car seuls les nœuds de l'interface sont dupliqués, et non ceux de l'intérieur des sous-domaines.

Le découpage des maillages est assuré par METIS [90]. Cette librairie utilise des algorithmes multi-niveaux pour générer des partitions de graphe (et donc de maillages) ainsi que des renumérotations qui limitent le remplissage (voir section 4.5.1). La figure 2.4 montre les sous-domaines créés par le découpage du maillage de l'aile RAE2822 (cas test I). La qualité d'un découpage tient dans l'équilibrage de la charge de travail de chaque processeur (en anglais *load balancing*). D'une manière grossière, cela vise à ce que les sous-domaines aient tous une taille similaire, que l'on mesure par le nombre d'éléments, de nœuds ou encore d'arêtes. Une vision plus fine demande une étude du code de calcul pour lequel le maillage est découpé. En effet, si tous les sous-domaines ont le même nombre d'éléments, ils n'ont pas nécessairement le même nombre de nœuds. Cela tient au nombre non constant d'éléments partageant un même sommet.

Prenons trois opérations typiques du code de calcul éléments finis résolvant un problème linéaire par une méthode itérative : la construction du système linéaire, la multiplication d'un vecteur par une matrice, et le produit scalaire de deux vecteurs. Comme on l'a vu dans la section 2.2.1, la méthode des éléments finis construit le système linéaire par assemblage de contributions élémentaires, c'est-à-dire aux éléments. Le stockage d'une matrice creuse dépend du nombre d'arêtes du graphe, donc le produit matrice-vecteur

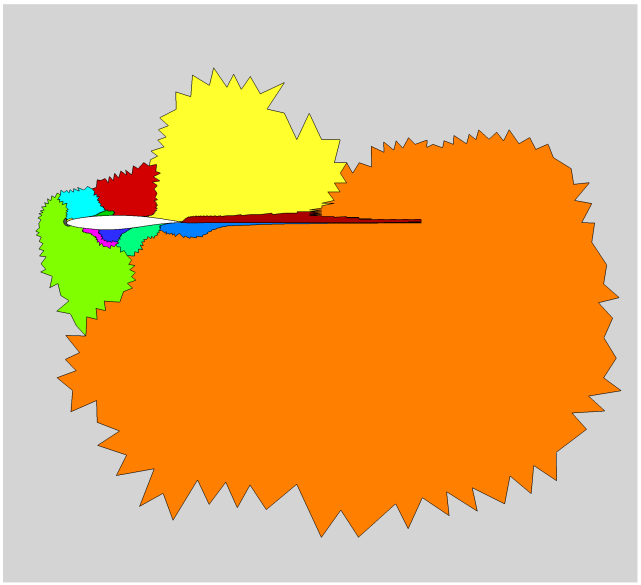


FIGURE 2.4 – Maillage 2D de l'aile RAE2822 (cas test I) découpé par METIS en 16 sous-domaines. Zoom près de l'aile.

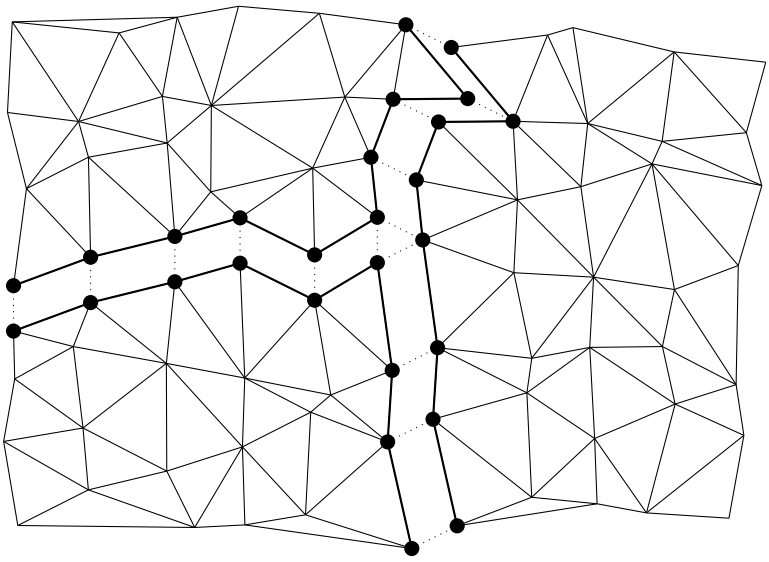


FIGURE 2.5 – Exemple de découpage à recouvrement minimal d'un maillage. Les nœuds et les arêtes partagées par plusieurs domaines sont en gras.

est essentiellement une somme de termes définis sur les arêtes. Enfin, un produit scalaire de deux vecteurs dépend du nombre de sommets du graphe. L'équilibrage parfait de ces trois opérations nécessite un nombre identique d'éléments, d'arêtes et de nœuds par sous-domaine.

Un code Navier-Stokes linéarisé ne crée qu'une fois un système linéaire par calcul, et la résolution de ce système par une méthode itérative est très coûteuse et demande de nombreux produits matrice-vecteurs et produits scalaires (*cf.* chapitre 3). Pour un tel code, l'équilibrage en éléments est moins important que l'équilibrage des arêtes et des nœuds. Notons enfin que le nombre d'éléments, d'arêtes et de nœuds n'est pas indépendant dans un maillage composé de simplexes. L'équilibrage d'une quantité n'est donc pas indépendant de l'équilibrage des autres.

La figure 2.5 rappelle que pour un découpage aux éléments sans recouvrement, les nœuds et les arêtes des bords sont dupliqués. Les vecteurs sont stockés sous format assemblé, tandis que la matrice du système linéaire est stockée sous format non assemblé. On va détailler ici l'impact de ce choix sur les produits scalaires et les produits matrice-vecteur, deux opérations courantes d'un solveur itératif.

Soit V l'ensemble des n sommets du maillage, que l'on identifiera à $[1;n] \cap \mathbb{N}$ l'ensemble des nombres entiers de 1 à n. Le maillage est découpé en N sous-domaines, et l'on note V_i l'ensemble des sommets de chaque sous-domaine. Leur union génère V. Leur intersection deux à deux n'est pas nécessairement nulle, car ils partagent des nœuds s'ils sont voisins. Soit un nœud $k \in V$. On note $\mathcal{M}_k = \{i/k \in V_i\}$ les sous-domaines auxquels k appartient. L'ensemble $\mathcal{M}_k$ n'est jamais vide. On note $m_k = |\mathcal{M}_k|$ le nombre de sous-domaines V_i auquel il appartient. Si k est un nœud intérieur d'un sous-domaine, alors $m_k = 1$.

Soient $\mathbf{x}$ et $\mathbf{y}$ deux vecteurs définissant chacun un champ sur le maillage. On note x_k et y_k la valeur réelle qu'ils prennent au nœud k. Le produit scalaire canonique de $\mathbf{x}$ et de $\mathbf{y}$ est défini par

$$(\mathbf{x}, \mathbf{y}) = \sum_{k \in V} x_k y_k \,.$$

La parallélisation naïve du produit scalaire qui consisterait à sommer la contribution locale de chaque sous- domaine, indiqué par la formule suivante, est fausse :

$$(\mathbf{x}, \mathbf{y}) \neq \sum_{i=1}^{N} \sum_{k \in V_i} x_k y_k \,.$$

En effet, pour un nœud k sur une interface entre les sous-domaines, le terme $x_k y_k$ serait compté m_k fois au lieu d'une. Pour contrer cela, il faut pondérer les contributions locales par $1/m_k$. Cela conduit à la formule suivante pour le produit scalaire distribué :

$$(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} \sum_{k \in V_i} \frac{1}{m_k} x_k y_k \,. \tag{2.36}$$

La somme sur i est une opération de somme globale. La matrice $\mathbf{A}$ n'est pas assemblée. Notons $\mathbf{A}^i$ la matrice locale issue de l'assemblage des contributions élémentaires du sous-domaine i. Soit $\mathbf{u}$ un vecteur. On note $\mathbf{v} = \mathbf{Au}$. Alors

$$\forall j \in V, v_j = \sum_{k \in V} A_{jk} u_k \,.$$

On décompose cette somme par les sous-domaines $\mathcal{M}_j$ auxquels j appartient. Par définition de l'assemblage,

$$\forall (j,k) \in V^2, A_{jk} = \sum_{i \in \mathcal{M}_j \cap \mathcal{M}_k} A^i_{jk} \,.$$

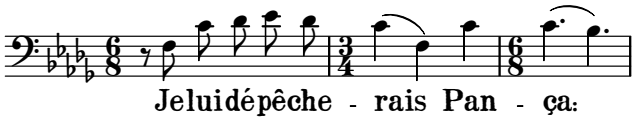
La décomposition du produit matrice-vecteur s'effectue alors :

$$\begin{aligned} \forall j \in V, v_j &= \sum_{k \in V} A_{jk} u_k \\ &= \sum_{k \in V} \left(\sum_{i \in \mathcal{M}_j \cap \mathcal{M}_k} A^i_{jk} \right) u_k \\ &= \sum_{i \in \mathcal{M}_j} \sum_{k \in V_i} A^i_{jk} u_k \,. \end{aligned}$$

La permutation de somme s'explique par le fait que $i \in \mathcal{M}_j \cap \mathcal{M}_k \iff \exists\, i, (j,k) \in V_i$. Ainsi, la somme sur i, comme elle est restreinte aux seuls $i \in \mathcal{M}_j$, indique une opération d'assemblage pour les nœuds de la frontière entre les sous-domaines. Le produit matrice-vecteur s'effectue par des produits matrice-vecteur locaux effectués en parallèle, et dont le résultat est assemblé à la frontière entre les sous-domaines.

Première partie

Solveur linéaire parallèle



Ravel, *Don Quichotte à Dulcinée*, 1. Chanson romanesque

Chapitre 3

Le solveur GMRES

3.1 Solveur itératif ou solveur direct ?

Comme indiqué dans la section 2.3.1, la résolution des équations de Navier-Stokes linéarisées implique la résolution d'un système linéaire. Ce système présente plusieurs particularités. Premièrement, il est de très grande taille. Un maillage Navier-Stokes 3D standard possède une dizaine de millions de nœuds. Comme cinq inconnues (cf. 2.1) sont présentes à chacun de ces nœuds, le système linéaire possède plus de cinquante millions d'inconnues. Ensuite, ce système est issu d'une discrétisation des équations par une méthode éléments finis. Pour des éléments finis de Lagrange P1, le graphe de remplissage de la matrice [135] est identique au graphe du maillage. Le nombre de sommets voisins d'un autre définit le nombre de blocs non-nuls de chaque ligne et colonne de la matrice, et est de l'ordre de plusieurs dizaines. Cela est à comparer à la dizaine de millions de nœuds du maillage. La matrice est donc extrêmement creuse.

Il existe deux grandes catégories de méthodes pour résoudre des systèmes linéaires de grandes tailles : les méthodes itératives et les méthodes directes. Les méthodes itératives, comme leur nom l'indique, créent une suite d'approximations de la solution. L'utilisateur définit une tolérance de résolution, qui est la norme maximale du résidu $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$ à partir de laquelle l'utilisateur est satisfait de la solution. La méthode directe calcule la solution exacte (aux erreurs numériques près) $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$.

Les avantages et les inconvénients de ces deux classes de méthodes sont faciles à comprendre. Les méthodes itératives se basent sur la création de suite d'approximations soit par itérés de la matrice (comme pour les méthodes basées sur des espaces de Krylov), soit par multiplication par des parties de la matrice (méthodes de relaxation type Jacobi, Gauss-Seidel et SOR). Ces méthodes utilisent donc des produits matrices-vecteurs, pour lesquels

les formats de stockage peuvent être optimisés [135, chapitre 3]. Le coût en mémoire de ces algorithmes est uniquement celui de la matrice creuse et d'un certain nombre de vecteurs. Pour les problèmes abordés dans cette thèse, le nombre de vecteurs composant l'espace de Krylov est de l'ordre de la centaine.

Les solveurs directs calculent l'application de $\mathbf{A}^{-1}$ à un vecteur. En général, cela passe par une décomposition de la matrice $\mathbf{A}$, par exemple une décomposition LU [10, 39]. Les facteurs de cette décomposition sont facilement inversibles. Une fois cette décomposition calculée, on peut résoudre quasi-instantanément de nombreux seconds membres différents pour une même matrice. Le temps de résolution ne dépend pas du conditionnement de la matrice $\kappa(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^{-1}\|$. À l'inverse, certaines méthodes itératives, comme le gradient conjugué pour des matrices symétriques définies positives, ont un taux de convergence dépendant directement de $\kappa(\mathbf{A})$. Les méthodes directes ont un important désavantage de mémoire requise. Pour un système plein, la décomposition LU coûte autant en mémoire que la matrice. Une méthode directe est alors très adaptée. Par contre, la décomposition LU d'une matrice creuse donne des facteurs $\mathbf{L}$ et $\mathbf{U}$ beaucoup plus remplis que la matrice $\mathbf{A}$ initiale, comme cela est présenté dans la section 4.5.1. Il existe des solveurs directs comme MUMPS [7, 9], adaptés aux matrices creuses. Le coût de stockage des facteurs pleins est alors identique à celui de la matrice. Pour conclure, la parallélisation des méthodes directes est délicate, comme le montre la parallélisation d'une méthode de préconditionnement ILU [135]. Les méthodes itératives sont très simples à paralléliser, comme cela est montré dans la section 3.2.2.

3.2 GMRES

Cette section présente l'algorithme GMRES avec déflation des petites valeurs propres utilisé par Dassault Aviation pour la résolution des systèmes linéaires Navier-Stokes linéarisées.

3.2.1 Description de l'algorithme

L'algorithme GMRES a été introduit par Saad et Schultz [137]. Il se base sur la minimisation du résidu au sens des moindres carrés sur l'espace de Krylov orthonormalisé via une méthode d'Arnoldi.

Méthode d'Arnoldi et espace de Krylov

On appelle espace de Krylov $\mathcal{K}_m(\mathbf{A}, \mathbf{r})$ l'espace vectoriel engendré par les itérés successifs du vecteur $\mathbf{r}$ par la matrice $\mathbf{A}$ jusqu'à la puissance $m-1$ [135].

$$\mathcal{K}_m(\mathbf{A}, \mathbf{r}) = \text{vect}\left\{\mathbf{r}, \mathbf{Ar}, \mathbf{A}^2\mathbf{r}, \ldots, \mathbf{A}^{m-1}\mathbf{r}\right\} \tag{3.1}$$

C'est dans cet espace que l'algorithme GMRES va chercher une solution approchée au système linéaire $\mathbf{Ax} = \mathbf{b}$ pour $\mathbf{r} = \mathbf{b} - \mathbf{Ax}_0$, où $\mathbf{x}_0$ est le point de départ de la méthode itérative. La solution approchée $\mathbf{x}_0$ est toujours choisie comme étant le vecteur nul, de sorte que le résidu initial est égal au second membre.

On cherche une base orthonormée de cet espace de Krylov, que l'on obtient par une simple procédure d'orthonormalisation d'Arnoldi, présentée dans l'algorithme 1.

Algorithme 1 Méthode d'Arnoldi pour $\mathcal{K}_m(\mathbf{A}, \mathbf{r})$

$\mathbf{v}_1 \leftarrow \mathbf{r}/\|\mathbf{r}\|$
for $j = 1,m$ **do**
  $\mathbf{w}_j \leftarrow \mathbf{Av}_j$
  for $i = 1,j$ **do**
    $h_{ij} = (\mathbf{w}_j, \mathbf{v}_i)$
  end for
  $\mathbf{w}_j \leftarrow \mathbf{w}_j - \sum_{i=1}^{j} h_{ij}\mathbf{v}_i$
  $h_{j+1,j} \leftarrow \|\mathbf{w}_j\|$
  $\mathbf{v}_{j+1} \leftarrow \mathbf{w}_j/h_{j+1,j}$
end for

On note $\overline{\mathbf{H}}_k$ la matrice d'Hessenberg de taille $(k+1) \times k$ composée des coefficients (h_{ij}). C'est la matrice créée après k étapes de la méthode d'Arnoldi. On note également $\mathbf{H}_k$ la matrice de taille $k \times k$ composée des mêmes coefficients. La matrice $\overline{\mathbf{H}}_k$ a une ligne de plus de que $\mathbf{H}_k$, composée uniquement de zéros et d'un seul terme non nul $h_{k+1,k}$. On remarque que $\mathbf{H}_k$ est triangulaire supérieure plus une diagonale inférieure.

Notons $\mathbf{V}_{k+1}$ la matrice colonne de vecteurs $(\mathbf{v}_j)_{j=1,k+1}$ formés après k étapes de la méthode d'Arnoldi. Alors, la relation fondamentale de la méthode d'Arnoldi s'écrit

$$\mathbf{AV}_k = \mathbf{V}_{k+1}\overline{\mathbf{H}}_k \tag{3.2}$$

La démonstration se fait simplement en remarquant que

$$\mathbf{w}_j = \mathbf{Av}_j - \sum_{i=1}^{j} h_{ij}\mathbf{v}_i = h_{j+1,j}\mathbf{v}_{j+1} \tag{3.3}$$

On obtient alors

$$\mathbf{Av}_j = \sum_{i=1}^{j+1} h_{ij}\mathbf{v}_i \tag{3.4}$$

La relation d'Arnoldi est intéressante car elle montre que l'action de la matrice $\mathbf{A}$ de grande taille sur les vecteurs $\mathbf{v}_i$ peut se réduire à celle d'une matrice $\overline{\mathbf{H}}_k$ beaucoup plus petite. Les tailles classiques d'espace de Krylov sont de l'ordre de la centaine, à comparer avec la matrice $\mathbf{A}$ dont l'ordre est de plusieurs dizaines de millions dans notre cas. La matrice $\overline{\mathbf{H}}_k$ condense donc l'information de la matrice que l'on a pu observer après k multiplications de vecteurs par celle-ci. Enfin, notons que la matrice d'Hessenberg $\mathbf{H}_k$ peut se calculer uniquement à l'aide de $\mathbf{A}$ et de $\mathbf{V}_k$:

$$\mathbf{V}_k^T \mathbf{A} \mathbf{V}_k = \mathbf{H}_k \tag{3.5}$$

En effet, en partant de la relation d'Arnoldi (3.2), et en notant $\mathbf{I}_k$ la matrice identité,

$$\begin{aligned} \mathbf{V}_k^T \mathbf{A} \mathbf{V}_k &= \mathbf{V}_k^T \mathbf{V}_{k+1} \overline{\mathbf{H}}_k \\ &= \begin{pmatrix} & 0 \\ \mathbf{I}_k & \vdots \\ & 0 \end{pmatrix} \overline{\mathbf{H}}_k \\ &= \mathbf{H}_k \end{aligned}$$

Méthode d'Arnoldi modifiée

La méthode d'Arnoldi précédemment introduite effectue en une seule fois la projection du nouveau vecteur $\mathbf{w}_j$ sur l'espace orthogonal à l'hyperplan de base $\mathbf{V}_j$. On peut montrer [135] [63, sec. 5.2.8] qu'en arithmétique inexacte, cette méthode est sensible aux erreurs numériques qui peuvent conduire à une perte d'orthogonalité entre les vecteurs. La forme modifiée de la méthode d'Arnoldi, qui effectue la projection sur l'orthogonal de chacun des vecteurs de la base $\mathbf{V}_j$ à la suite, est plus robuste aux erreurs numériques. Cela donne l'algorithme 2.

Algorithme 2 Méthode d'Arnoldi modifiée

$\mathbf{v}_1 \leftarrow \mathbf{r}/\|\mathbf{r}\|$
for $j = 1,m$ **do**
    $\mathbf{w}_j \leftarrow \mathbf{A}\mathbf{v}_j$
    for $i = 1,j$ **do**
        $h_{ij} = (\mathbf{w}_j, \mathbf{v}_i)$
        $\mathbf{w}_j \leftarrow \mathbf{w}_j - h_{ij}\mathbf{v}_i$
    end for
    $h_{j+1,j} \leftarrow \|\mathbf{w}_j\|$
    $\mathbf{v}_{j+1} \leftarrow \mathbf{w}_j / h_{j+1,j}$
end for

Pour encore améliorer la précision de la méthode, on peut effectuer une deuxième passe d'orthonormalisation après la première [135]. Enfin, une dernière méthode d'orthonormalisation d'une famille de vecteurs est basée sur les réflexions d'Householder, et est encore plus précise en arithmétique inexacte que la méthode d'Arnoldi, au prix d'un coût de calcul plus élevé [63].

L'algorithme GMRES

L'algorithme GMRES consiste en la minimisation, au sens des moindres carrés, du résidu $\mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{b}$, sur l'espace de Krylov $\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$. Une combinaison linéaire des vecteurs de la base de Krylov est recherchée. On note $\mathbf{y}_m \in \mathbb{R}^m$ ses coordonnées. La solution $\mathbf{x}_m$ est de la forme

$$\mathbf{x}_m = \mathbf{x}_0 + \mathbf{V}_m \mathbf{y}_m \tag{3.6}$$

Le résidu final s'exprime

$$\mathbf{r}_m = \mathbf{b} - \mathbf{A}\mathbf{x}_m = \mathbf{b} - \mathbf{A}\mathbf{V}_m \mathbf{y}_m \tag{3.7}$$

Si l'on prend comme solution initiale $\mathbf{x}_0 = \mathbf{0}$, alors le résidu initial $\mathbf{r}_0$ qui sert à construire l'espace de Krylov vaut $\mathbf{r}_0 = \mathbf{b}$. Dans la méthode d'Arnoldi (algorithmes 1 et 2), le vecteur initial $\mathbf{v}_1 = \mathbf{r}_0$ est normalisé. Notons alors β la norme du résidu initial $\mathbf{r}_0$. À l'aide de la relation d'Arnoldi (3.2) et des notations introduites précédemment, on obtient

$$\begin{aligned} \mathbf{r}_m &= \mathbf{b} - \mathbf{A}\mathbf{x}_m \\ &= \mathbf{r}_0 - \mathbf{A}\mathbf{V}_m \mathbf{y}_m \\ &= \beta \mathbf{v}_1 - \mathbf{V}_{m+1} \overline{\mathbf{H}}_m \mathbf{y}_m \\ &= \mathbf{V}_{m+1} \Big(\beta \mathbf{e}_1 - \overline{\mathbf{H}}_m \mathbf{y}_m \Big) \end{aligned} \tag{3.8}$$

où $\mathbf{e}_1$ est le premier vecteur de la base canonique de $\mathbb{R}^m$. On obtient la relation entre le résidu complet et celui dans la base réduite :

$$\|\mathbf{r}_m\|_2 = \Big\| \beta \mathbf{e}_1 - \overline{\mathbf{H}}_m \mathbf{y}_m \Big\|_2 \tag{3.9}$$

En effet, la base $\mathbf{V}_{m+1}$ est orthonormale, donc elle ne change pas la norme l_2 d'un vecteur. La minimisation du résidu $\mathbf{r}_m$ revient à un simple problème de moindres carrés de taille $(m+1) \times m$:

$$\min_{\mathbf{y}_m \in \mathbb{R}^m} \Big\| \beta \mathbf{e}_1 - \overline{\mathbf{H}}_m \mathbf{y}_m \Big\|_2 \tag{3.10}$$

Ce système peut facilement être résolu : la matrice $\overline{\mathbf{H}}_m$ est de type Hessenberg, c'est-à-dire triangulaire supérieure avec une diagonale inférieure.

Cette matrice peut est rendue triangulaire supérieure à l'aide de m rotations de Givens notées $(\mathbf{\Omega}_i)$. Les rotations de Givens, qui servent à annuler un terme dans une matrice, sont décrites par exemple dans [63]. La rotation $\mathbf{\Omega}_i$ annule le terme $h_{i+1,i}$ de $\overline{\mathbf{H}}_m$. Les matrices de rotation de Givens sont orthonormales, donc ne changent pas la norme du problème de minimisation. Notons $\mathbf{Q}_i = \prod_{j=i}^{1} \mathbf{\Omega}_i$ la transformation totale, produit des rotations. Alors

$$\mathbf{Q}_m \overline{\mathbf{H}}_m = \overline{\mathbf{R}}_m = \begin{pmatrix} & \mathbf{R}_m & \\ 0 & \dots & 0 \end{pmatrix} \tag{3.11}$$

où $\mathbf{R}_m$ est une matrice triangulaire supérieure de taille $m \times m$. La même transformation appliquée au second membre $\beta \mathbf{e}_1$ du problème de minimisation donne un vecteur $\mathbf{g}_{m+1}$ de coordonnées (γ_i) :

$$\mathbf{Q}_m \beta \mathbf{e}_1 = \mathbf{g}_{m+1} = \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_m \\ \gamma_{m+1} \end{pmatrix} = \begin{pmatrix} \mathbf{g}_m \\ \gamma_{m+1} \end{pmatrix} \tag{3.12}$$

Le problème de minimisation devient

$$\begin{aligned} \min_{\mathbf{y}_m \in \mathbb{R}^m} \left\| \beta \mathbf{e}_1 - \overline{\mathbf{H}}_m \mathbf{y}_m \right\|_2 &= \left\| \mathbf{g}_{m+1} - \overline{\mathbf{R}}_m \mathbf{y}_m \right\|_2 \\ &= \left\| \begin{pmatrix} \mathbf{g}_m \\ \gamma_{m+1} \end{pmatrix} - \begin{pmatrix} & \mathbf{R}_m & \\ 0 & \dots & 0 \end{pmatrix} \mathbf{y}_m \right\|_2 \\ &= \left\| \begin{pmatrix} \mathbf{g}_m \\ \gamma_{m+1} \end{pmatrix} - \begin{pmatrix} \mathbf{R}_m \mathbf{y}_m \\ 0 \end{pmatrix} \right\|_2 \end{aligned} \tag{3.13}$$

La matrice $\mathbf{R}_m$ est triangulaire supérieure, et est inversible (sauf si la méthode a atteint la solution exacte, voir la notion de *lucky breakdown* dans [135]). Par une méthode de remontée triangulaire [63], on obtient les coordonnées de la solution

$$\mathbf{y}_m = \mathbf{R}_m^{-1} \mathbf{g}_m \tag{3.14}$$

La norme du résidu vaut exactement

$$\|\mathbf{r}_m\|_2 = \min_{\mathbf{y}_m \in \mathbb{R}^m} \left\| \beta \mathbf{e}_1 - \overline{\mathbf{H}}_m \mathbf{y}_m \right\|_2 = |\gamma_{m+1}| \tag{3.15}$$

Le résultat de (3.15) est particulièrement intéressant. Il montre que la valeur du problème de minimisation peut être trouvée sans résoudre ce problème. La transformation successive du second membre $\beta \mathbf{e}_1$ par les rotations de Givens donne, en dernière coordonnée, la valeur du résidu à l'itération GMRES courante, sans avoir à calculer la solution $\mathbf{y}_k = \mathbf{R}_k^{-1} \mathbf{g}_k$. C'est pourquoi au cours des itérations de l'algorithme GMRES les vecteur

résidu $\mathbf{r}$ et solution $\mathbf{x}$ ne sont pas mis à jour avant d'avoir atteint le seuil de convergence souhaité ou un point de redémarrage.

L'algorithme 3 présente cette version du GMRES avec m itérations. Si, à la fin des ces itérations, la tolérance sur la norme du résidu n'est pas atteinte, l'algorithme est redémarré en prenant comme nouveau point de départ la solution approchée obtenue à la fin des itérations. Cela consiste à recommencer l'algorithme 3 en remplaçant les deux premières lignes par $\mathbf{x}_0 \leftarrow \mathbf{x}_m$ et $\mathbf{r}_0 \leftarrow \mathbf{r}_m$. L'algorithme peut s'arrêter avant d'avoir effectué les m itérations si la norme du résidu est plus petite que la taille demandée. De plus, le vecteur nul est pris comme point de départ de la méthode itérative. Le résidu initial est donc égal au second membre du système linéaire.

Algorithme 3 GMRES(m) avec redémarrage

$\mathbf{x}_0 \leftarrow \mathbf{0}$
$\mathbf{r}_0 \leftarrow \mathbf{b}$
$\beta \leftarrow \|\mathbf{r}_0\|$
$\mathbf{v}_1 \leftarrow \mathbf{r}_0/\beta$
$\mathbf{g} \leftarrow \beta \mathbf{e}_1$
for $j = 1,m$ **do**
    $\mathbf{w}_j \leftarrow \mathbf{A}\mathbf{v}_j$
    for $i = 1,j$ **do**
        $h_{ij} = (\mathbf{w}_j, \mathbf{v}_i)$
        $\mathbf{w}_j \leftarrow \mathbf{w}_j - h_{ij}\mathbf{v}_i$
    end for
    $h_{j+1,j} \leftarrow \|\mathbf{w}_j\|$
    $\mathbf{v}_{j+1} \leftarrow \mathbf{w}_j/h_{j+1,j}$
    Calcul de la rotation de Givens $\mathbf{\Omega}_j$ annulant $h_{j+1,j}$
    Application des rotations de Givens $\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_j$ sur $\overline{\mathbf{H}}$
    $\mathbf{g} \leftarrow \mathbf{\Omega}_j \mathbf{g}$
    If $|g_{j+1}| \leq \varepsilon$ **exit loop**
end for
$\mathbf{y}_m \leftarrow \mathbf{H}^{-1}\mathbf{g}$
Application des rotations de Givens inverses $\mathbf{\Omega}_m^{-1}, \ldots, \mathbf{\Omega}_1,^{-1}$ à $\mathbf{y}_m$
$\mathbf{x}_m \leftarrow \mathbf{x}_0 + \mathbf{V}_m \mathbf{y}_m$
$\mathbf{r}_m \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}_m$

3.2.2 Parallélisation

Comme expliqué dans le paragraphe 2.4, les maillages utilisés par Dassault Aviation sont trop volumineux pour que les calculs soient effectués sur un seul processeur. Les données sont donc distribuées sur plusieurs processeurs.

L'algorithme GMRES est très facilement parallélisable. Les données distribuées sont la matrice $\mathbf{A}$ et les vecteurs de même taille tels que la base $\mathbf{V}_m$

de l'espace de Krylov. Les produits matrice-vecteur sont parallélisés comme indiqué dans le paragraphe 2.4. L'autre opération parallèle est les produits scalaires pour la méthode d'Arnoldi d'orthonormalisation. La parallélisation de cette opération est triviale : les produits scalaires partiels sont effectués par processeur, puis ils sont sommés sur l'ensemble des processeurs. Une attention particulière est à porter aux nœuds d'interface, qui sont dupliqués, et dont la contribution doit être réduite d'autant (voir paragraphe 2.4).

3.2.3 Préconditionnement

Le préconditionnement, qui sera abordé plus en détails dans le chapitre 4, est une technique pour accélérer la résolution d'un système linéaire par une méthode itérative. Il consiste à transformer la matrice pour la rapprocher autant que possible de l'identité. Pour ce faire, on se cherche une matrice $\mathbf{M}$, dont l'inverse est simple à calculer et approche $\mathbf{A}^{-1}$. Ainsi, $\mathbf{AM}^{-1} \approx \mathbf{I}$ et le système linéaire sera plus simple à résoudre.

Le préconditionnement à gauche consiste à prémultiplier le système par $\mathbf{M}^{-1}$. On obtient donc

$$\mathbf{M}^{-1}\mathbf{Ax} = \mathbf{M}^{-1}\mathbf{b} \tag{3.16}$$

Le préconditionnement à droite se déduit simplement de la relation $\mathbf{Ax} = \mathbf{AM}^{-1}\mathbf{Mx}$. Un changement de variable permet d'obtenir

$$\begin{cases} \mathbf{AM}^{-1}\mathbf{y} = \mathbf{b} \\ \mathbf{x} = \mathbf{M}^{-1}\mathbf{y} \end{cases} \tag{3.17}$$

Les modifications à appliquer à l'algorithme GMRES pour préconditionner la matrice sont minimales. Les produits matrice-vecteur (ligne 7 de l'algorithme 3) sont modifiés : la matrice $\mathbf{A}$ est changée en $\mathbf{AM}^{-1}$ ou $\mathbf{M}^{-1}\mathbf{A}$ suivant le sens de préconditionnement.

Dans le cas d'un préconditionnement à gauche, le résidu initial $\mathbf{r}_0 = \mathbf{b} - \mathbf{Ax}_0$ doit être prémultiplié par $\mathbf{M}^{-1}$: $\mathbf{r}_0 = \mathbf{M}^{-1}(\mathbf{b} - \mathbf{Ax}_0)$. Le préconditionnement à droite impose de repasser dans l'espace initial pour la mise à jour du vecteur solution. La ligne 21 est remplacée par $\mathbf{x}_m \leftarrow \mathbf{x}_0 + \mathbf{M}^{-1}\mathbf{V}_m\mathbf{y}_m$.

Dans les deux cas, l'espace de Krylov sur lequel est minimisé le résidu n'est plus le même. Pour un préconditionneur à gauche, l'espace de Krylov $\mathcal{K}(\mathbf{A}, \mathbf{r})$ (cf (3.1)) devient $\mathcal{K}(\mathbf{M}^{-1}\mathbf{A}, \mathbf{M}^{-1}\mathbf{r})$. À droite, l'espace de Krylov est $\mathcal{K}(\mathbf{AM}^{-1}, \mathbf{r})$. Ainsi, le résidu GMRES qui est donné par l'algorithme est celui du système préconditionné.

Pour rendre le préconditionnement parallèle, on utilise la méthode de Schwarz additif dont les solveurs locaux sont un préconditionneur. Cette méthode est expliquée en détails dans le chapitre 4.

3.2.4 Amélioration des redémarrages par la déflation

Le rédémarrage simple de l'algorithme GMRES consiste à boucler l'algorithme 3 jusqu'à convergence du système linéaire, en fixant m qui définit la taille de l'espace de Krylov. Cette permet de limiter la taille mémoire, et de ne pas avoir à stocker un espace de Krylov dont la dimension est le nombre d'itérations nécessaire à atteindre la convergence. Sur nos cas industriels, il faut des milliers, voire des dizaines de milliers d'itérations pour converger le système linéaire associé, et il serait impossible de stocker autant de vecteurs.

Le redémarrage simple, s'il limite la taille mémoire, peut poser des problèmes de convergence. La figure 3.1 montre l'impact du redémarrage sur la convergence. En première approche, il est simple de comprendre pourquoi le redémarrage bloque la convergence. Tout l'espace de Krylov généré au cycle précédent est supprimée lors du redémarrage, alors que les informations qu'il contient peuvent être recyclées [124] Comme indiqué par la relation d'Arnoldi (3.2), l'information contenue dans la matrice $\overline{\mathbf{H}}_m$ représente l'action sur la base de Krylov $\mathbf{V}_m$ de l'opérateur $\mathbf{A}$. La matrice $\overline{\mathbf{H}}_m$ est donc un condensé de la matrice $\mathbf{A}$ sur l'espace généré par $\mathbf{V}_m$.

Un redémarrage standard ne conserve rien de l'espace de Krylov $\mathcal{K}_m(\mathbf{A}, \mathbf{r})$. L'information contenue dans $\overline{\mathbf{H}}_m$ est également perdue. Cela explique aisément pourquoi ce redémarrage simple dégrade fortement la convergence. À l'inverse, conserver une partie bien choisie de cette information lors des redémarrages doit permettre de se rapprocher d'un GMRES sans redémarrage.

Certaines méthodes itératives ont une convergence dépendante du contenu spectral de l'opérateur. La méthode du gradient conjugué, qui s'applique à des problèmes symétriques définis positifs, a une borne sur sa convergence qui dépend du conditionnement spectral $\kappa = \lambda_{\max}/\lambda_{\min}$, où $\lambda_{\max}$ et $\lambda_{\min}$ sont respectivement la plus grande et plus petite valeur propre de la matrice. La borne sur la convergence de cette méthode est $\left(\sqrt{\kappa} - 1\right)/\left(\sqrt{\kappa} + 1\right)$ [135]. Les bornes de convergence du GMRES sont plus complexes et ne font pas intervenir que les valeurs propres [48].Les matrices normales (i.e. dont les vecteurs propres sont orthogonaux) permettent d'obtenir des bornes de convergence de l'algorithme GMRES ne dépendant que des valeurs propres, mais la non-normalité d'un opérateur peut introduire des plateaux de convergence [119]. Greenbaum [66] utilise des opérateurs non normaux pour démontrer qu'il est possible, pour une courbe de convergence donnée, de créer une matrice aux valeurs propres arbitraires et un second membre pour lesquels la méthode GMRES donnerait cette même courbe de convergence.

Néanmoins, pour certains opérateurs, enlever l'influence des petites valeurs propres aide la convergence initiale après un redémarrage [43]. Comme on le verre plus tard, cette constatation expérimentale est vérifiée sur les matrices Navier-Stokes linéarisées que nous résolvons.

Garder une partie de l'espace de Krylov précédent pour ne pas entièrement perdre l'information qu'il contient est l'idée de base des méthodes de

redémarrage améliorées pour le GMRES. Deux questions se posent. Pour minimiser l'effet du redémarrage, quelle partie de l'espace de Krylov conserver ? Comment agencer les vecteurs conservés afin de garder la structure d'espace de Krylov ? Ceci permettra de ne pas modifier la base de l'algorithme GMRES.

Dans l'article décrivant pour la première fois l'algorithme GMRES [137], Saad et Schultz remarquent que le spectre de la matrice d'Hessenberg $\mathbf{H}_m$ est une approximation de celui de la matrice $\mathbf{A}$. La méthode des puissances itérées [136] montre également que les plus grandes valeurs propres sont très facilement captées par des itérations successives d'un vecteur avec la matrice. Ainsi, l'espace de Krylov $\mathcal{K}_m$ contiendra rapidement les vecteurs propres de $\mathbf{A}$ de grande valeur propre. Il n'est donc pas intéressant de garder ces vecteurs lors du redémarrage. À l'inverse, on cherchera à conserver les plus petites valeurs propres.

Valeurs propres et valeurs de Ritz harmoniques

Les méthodes d'Arnoldi permettent de calculer des approximations des valeurs propres de l'opérateur $\mathbf{A}$. La méthode de Rayleigh-Ritz [136] consiste à résoudre le problème aux valeurs propres de $\mathbf{A}$ tel que son résidu soit orthogonal à un espace de Krylov généré par $\mathbf{A}$. Si l'on paramètre le vecteur propre approché $\mathbf{x}$ de valeur propre λ par $\mathbf{x} = \mathbf{V}_m\mathbf{y}_m$, on obtient à l'aide de la définition de la matrice d'Hessenberg (3.5)

$$\begin{array}{rrcl} & \mathbf{A}\mathbf{x} - \lambda\mathbf{x} & \perp & \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0) \\ \Leftrightarrow & \mathbf{V}_m^T\left(\mathbf{A}\mathbf{V}_m\mathbf{y}_m - \lambda\mathbf{V}_m\mathbf{y}_m\right) & = & \mathbf{0} \\ \Leftrightarrow & \mathbf{H}_m\mathbf{y}_m & = & \lambda\mathbf{y}_m \end{array} \tag{3.18}$$

Ainsi, les valeurs de Ritz, par définition les valeurs propres de $\mathbf{H}_m$, sont des approximations des valeurs propres de l'opérateur $\mathbf{A}$. Pour identifier les petites valeurs propres de $\mathbf{A}$, on peut identifier les plus grandes valeurs propres de $\mathbf{A}^{-1}$. Les valeurs de Ritz harmoniques [64] sont définies comme étant les valeurs propres inverses approchées de $\mathbf{A}^{-1}$ dont le résidu est orthogonal à $\mathbf{A}\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$. On pose $\mathbf{u} = \mathbf{A}\mathbf{V}_m\mathbf{y}_m$ et $\lambda \in \mathbb{C}$.

$$\begin{array}{rrcl} & (\mathbf{A}^{-1}\mathbf{u} - \lambda^{-1}\mathbf{u}) & \perp & \mathbf{A}\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0) \\ \Leftrightarrow & \mathbf{V}_m^T\mathbf{A}^T(\mathbf{A}^{-1}\mathbf{u} - \lambda^{-1}\mathbf{u}) & = & 0 \\ \Leftrightarrow & \mathbf{V}_m^T\mathbf{A}^T(\mathbf{A}^{-1}\mathbf{A}\mathbf{V}_m\mathbf{y}_m - \lambda^{-1}\mathbf{A}\mathbf{V}_m\mathbf{y}_m) & = & 0 \\ \Leftrightarrow & \mathbf{V}_m^T\mathbf{A}^T(\mathbf{V}_m\mathbf{y}_m - \lambda^{-1}\mathbf{A}\mathbf{V}_m\mathbf{y}_m) & = & 0 \end{array}$$

On utilise ensuite la transposée de la relation d'Arnoldi (3.2) pour simplifier cette expression :

$$
\begin{array}{rrcl}
 & (\mathbf{A}^{-1}\mathbf{u} - \lambda^{-1}\mathbf{u}) & \perp & \mathbf{A}\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0) \\
\Leftrightarrow & \mathbf{V}_m^T\mathbf{A}^T(\mathbf{V}_m\mathbf{y}_m - \lambda^{-1}\mathbf{A}\mathbf{V}_m\mathbf{y}_m) & = & 0 \\
\Leftrightarrow & \overline{\mathbf{H}}_m^T\mathbf{V}_{m+1}^T\mathbf{V}_m\mathbf{y}_m - \lambda^{-1}\overline{\mathbf{H}}_m^T\mathbf{V}_{m+1}^T\mathbf{A}\mathbf{V}_m\mathbf{y}_m & = & 0 \\
\Leftrightarrow & \mathbf{H}_m^T\mathbf{y}_m - \lambda^{-1}\overline{\mathbf{H}}_m^T\overline{\mathbf{H}}_m\mathbf{y}_m & = & 0 \\
\Leftrightarrow & \overline{\mathbf{H}}_m^T\overline{\mathbf{H}}_m\mathbf{y}_m & = & \lambda\mathbf{H}_m^T\mathbf{y}_m
\end{array}
\tag{3.19}
$$

Dans l'annexe A, on montre que ce problème aux valeurs propres généralisées se transforme en la recherche des valeurs propres de la matrice $\mathbf{H}_m + h_{m+1,m}^2\mathbf{f}_m\mathbf{e}_m^T$, où $\mathbf{f}_m$ est la dernière colonne de $\mathbf{H}_m^{-T}$.

La méthode dite *GMRES with deflated restarting* introduite par Morgan [115] répond aux deux questions précédentes. Il est possible de choisir un espace contenant les vecteurs propres approchés de $\mathbf{A}$ de plus petites valeurs propres, tout en conservant une structure d'espace de Krylov. Les paires de Ritz harmoniques $(\theta, \mathbf{p})$ sont les valeurs et vecteurs propres de $\mathbf{H}_m + h_{m+1,m}^2\mathbf{H}_m^{-T}\mathbf{e}_m\mathbf{e}_m^T$ [115]. Elles convergent vers les valeurs propres et vecteurs propres de $\mathbf{A}$ lorsque la méthode GMRES converge [64].

L'utilisation des paires de Ritz harmoniques (plutôt que les valeurs propres de $\mathbf{H}_m$) est intéressante non seulement car elles convergent rapidement vers les plus petites valeurs propres de $\mathbf{A}$ [64] mais aussi car elles permettent de conserver la structure de l'espace de Krylov, comme montré par Morgan [115].

On note k le nombre de plus petites valeurs de Ritz harmoniques que l'on veut conserver parmi m lors d'un redémarrage avec déflation. Soit $\mathbf{P}_k$ la matrice de taille $k \times m$ constituée des k vecteurs de Ritz harmoniques $\mathbf{p}$ sélectionnés et orthonormalisés. Le problème aux valeurs propres généralisées est réel et peut donner des vecteurs de Ritz harmonique complexes, conjugués par paire. Dans ce cas, il est équivalent d'utiliser deux vecteurs réels formés respectivement de la partie réelle et imaginaire d'un des vecteurs conjugués [43]. La matrice $\overline{\mathbf{P}}_{k+1}$ de taille $(k+1) \times (m+1)$ est formée par la matrice $\mathbf{P}_k$ à laquelle on a ajouté des zéros en dernière ligne, et en dernier vecteur le résidu GMRES $\mathbf{g} - \overline{\mathbf{H}}_m\mathbf{y}_m$ orthonormalisé par rapport aux premières colonnes. La notation surlignée indique que sa dernière dimension est $m+1$. La dernière colonne de $\overline{\mathbf{P}}_{k+1}$ peut être calculée autrement, comme le montrent Röllin et Fichtner [132]. Le résidu GMRES est remplacé par $\begin{pmatrix} -\beta\mathbf{f}_m^T & 1 \end{pmatrix}^T$, qui lui est colinéaire [132]. Ainsi, avant son orthonormalisation, $\overline{\mathbf{P}}_{k+1}$ est définie par

$$
\overline{\mathbf{P}}_{k+1} = \begin{pmatrix} \mathbf{P}_k & -\beta\mathbf{f}_m \\ 0 \cdots 0 & 1 \end{pmatrix} \tag{3.20}
$$

On redéfinit la base orthonormale de l'espace de Krylov $\mathbf{V}_{k+1}$ et la matrice d'Hessenberg comme suit :

$$\overline{\mathbf{H}}_k^{\text{new}} = \overline{\mathbf{P}}_{k+1}^T \overline{\mathbf{H}}_m \mathbf{P}_k \tag{3.21}$$
$$\mathbf{V}_{k+1}^{\text{new}} = \mathbf{V}_{m+1} \overline{\mathbf{P}}_{k+1} \tag{3.22}$$

Comme $\mathbf{P}_{k+1}$ et $\mathbf{P}_k$ sont composés de vecteurs orthonormaux entre eux, la relation d'Arnoldi est conservée sur ces nouveaux espaces :

$$\mathbf{A}\mathbf{V}_k^{\text{new}} = \mathbf{V}_{k+1}^{\text{new}} \overline{\mathbf{H}}_k^{\text{new}} \tag{3.23}$$

En effet,

$$\begin{aligned}
\mathbf{V}_{k+1}^{\text{new}} \overline{\mathbf{H}}_k^{\text{new}} &= \mathbf{V}_{m+1} \overline{\mathbf{P}}_{k+1} \overline{\mathbf{P}}_{k+1}^T \overline{\mathbf{H}}_m \mathbf{P}_k \\
&= \mathbf{V}_{m+1} \overline{\mathbf{H}}_m \mathbf{P}_k \\
&= \mathbf{A}\mathbf{V}_m \mathbf{P}_k \\
&= \mathbf{A}\mathbf{V}_k^{\text{new}}
\end{aligned}$$

Par rapport à la relation d'Arnoldi standard, on remarque que la matrice $\overline{\mathbf{H}}_k^{\text{new}}$ n'est pas triangulaire supérieure, mais pleine. On peut la rendre triangulaire supérieure par $O(k^2)$ rotations de Givens. Morgan, dans [115], prouve que l'espace engendré par les colonnes $\mathbf{V}_{k+1}^{\text{new}}$ est bien un espace de Krylov. Le résidu dans l'espace réduit, comme prouvé dans [132] sera

$$\mathbf{g} = \left(\mathbf{V}_{k+1}^{\text{new}}\right)^T \mathbf{r}_m = \left(\mathbf{V}_{k+1}^{\text{new}}\right)^T (\mathbf{b} - \mathbf{A}\mathbf{x}_m) \tag{3.24}$$

Algorithme 4 Redémarrage du GMRES(m) avec déflation de k vecteurs propres

1: Application de l'algorithme GMRES(m) ▷ voir algorithme 3
2: Calcul des k premiers vecteurs propres $\mathbf{p}$ de $\mathbf{H}_m + h_{m+1,m}^2 \mathbf{H}_m^{-T} \mathbf{e}_m \mathbf{e}_m^T$
3: $\mathbf{P}_k \leftarrow (\mathbf{p}_i)_{i=1,\dots,k}$
4: Orthonormalisation de $\mathbf{P}_k$
5: $\overline{\mathbf{P}}_{k+1} \leftarrow \begin{pmatrix} \mathbf{P}_k & \\ 0 \dots 0 & \mathbf{g} - \overline{\mathbf{H}}_m \mathbf{y}_m \end{pmatrix}$
6: Orthonormalisation du dernier vecteur de $\overline{\mathbf{P}}_{k+1}$
7: $\overline{\mathbf{H}}_k \leftarrow \overline{\mathbf{P}}_{k+1}^T \overline{\mathbf{H}}_m \mathbf{P}_k$
8: $\mathbf{V}_{k+1} \leftarrow \mathbf{V}_{m+1} \overline{\mathbf{P}}_{k+1}$
9: $\mathbf{g} \leftarrow \left(\mathbf{V}_{k+1}^{\text{new}}\right)^T \mathbf{r}_m$

Les figures 3.1 et 3.2 présentent la convergence du résidu normalisé pour l'algorithme GMRES avec ou sans déflation des plus petites valeurs propres pour le cas RAE2822 qui est présenté dans la section 3.3.4. Pour la figure 3.1, l'algorithme GMRES est préconditionné par ILU(1) (*cf.* chapitre

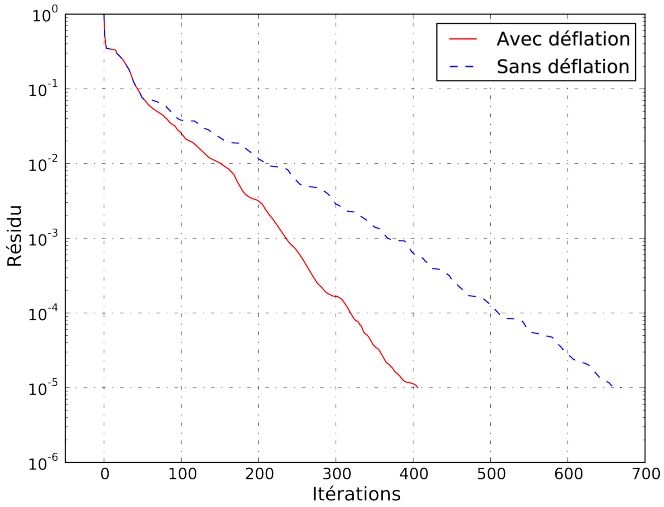


FIGURE 3.1 – Résidu adimensionné issu de l'algorithme GMRES (50 vecteurs de Krylov) préconditionné par ILU(1) avec ou sans déflation. Cas RAE2822 (cas test I).

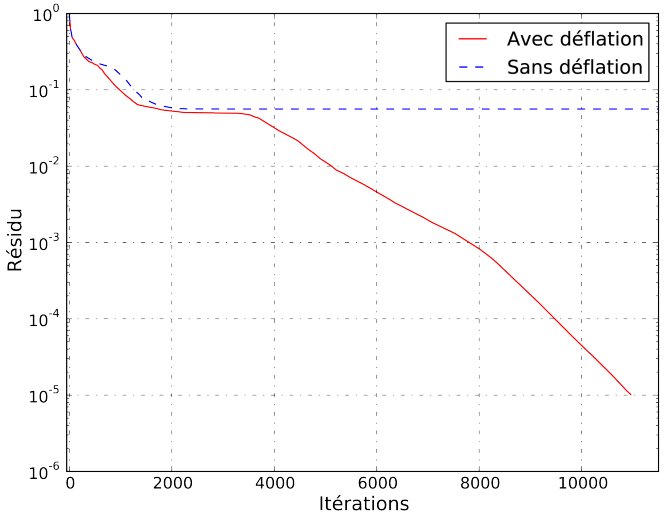


FIGURE 3.2 – Résidu adimensionné issu de l'algorithme GMRES (250 vecteurs de Krylov) préconditionné par BSOR avec ou sans déflation. Cas RAE2822 (cas test I).

4). La convergence est plus rapide pour l'algorithme avec déflation des plus petites valeurs propres. Sans déflation, la méthode itérative demande quelques itérations après chaque redémarrage avant de pouvoir diminuer le résidu. C'est un cas où la déflation accélère l'obtention de la convergence. Sur la figure 3.2, l'algorithme GMRES est préconditionné par BSOR, qui est moins performant. Avec déflation, la courbe de convergence présente un plateau à partir de la 2000^{e} itération jusqu'à la 3500^{e} environ. Durant cette phase, la norme des plus petites valeurs de Ritz harmoniques trouvées diminue à chaque redémarrage. Une fois la plus petite trouvée, la diminution du résidu reprend, car l'effet néfaste du vecteur propre correspondant sur la convergence a été éliminé. Sans déflation, l'algorithme atteint le plateau à peu près au même moment, mais il reste bloqué. Ce cas montre que la déflation de petites valeurs propres permet de faire converger certains cas difficiles.

3.3 La méthode Block-GMRES

Maintenant que nous avons présenté l'algorithme GMRES avec déflation existant dans le code AeTher avant le début de ces travaux, nous allons introduire une extension de cette algorithme qui a été codée et testée durant cette thèse.

3.3.1 Intérêts attendus

L'algorithme block-GMRES est une extension de l'algorithme GMRES pour résoudre des systèmes linéaires à plusieurs seconds membres. Pour la génération de base de données de *flutter* (voir la section 1.1), les efforts aérodynamiques pour chacune des centaines de formes modales doivent être calculés, et ce pour une fréquence et un point de vol (altitude, vitesse, incidence) donnés. Chaque forme modale donne lieu à un second membre $\frac{\partial \mathbf{E}}{\partial \mathbf{x}}$, qu'il faudra résoudre avec une même matrice $\frac{\partial \mathbf{E}}{\partial \mathbf{V}}$.

Plutôt que de relancer plusieurs fois l'algorithme GMRES avec un second membre différent mais la même matrice, un algorithme permettant de résoudre plusieurs second membres en même temps peut être avantageux. L'algorithme block-GMRES a deux avantages, qui sont détaillés dans les deux paragraphes suivant.

Le premier avantage est d'ordre purement informatique. La méthode block-GMRES est basée sur l'utilisation de vecteurs blocs de s colonnes. Le produit d'une matrice avec un bloc vecteur de s colonnes est plus rapide que s produits matrices-vecteurs, pour des raisons de temps d'accès à la mémoire. Les opérations parallèles, comme l'assemblage ou l'orthonormalisation, sont regroupées pour toutes les colonnes du vecteur.

L'autre avantage est d'ordre mathématique. Comme on va le voir, l'espace de Krylov pour une méthode par bloc est s fois plus grand, et chaque second

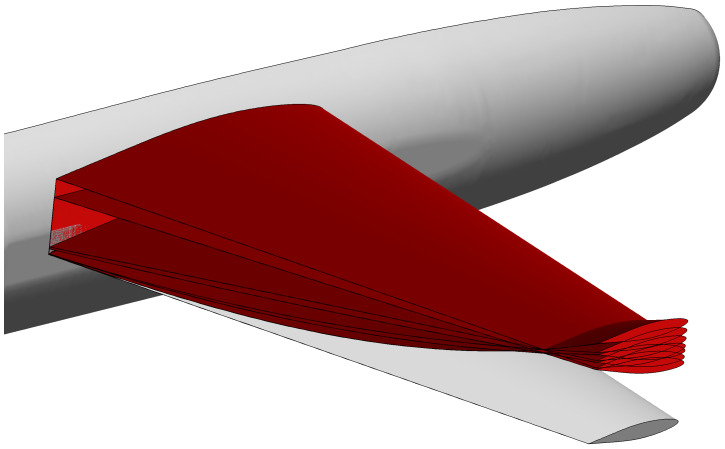


FIGURE 3.3 – Différents modes propres structuraux de la voilure de la maquette DTP

membre utilise les itérés des autres seconds membres pour trouver leur solution. On peut ainsi penser que la résolutions des s second membres en même temps prendra moins d'itérations que s résolutions indépendantes. Comme le montre la figure 3.3, les modes propres structuraux d'une aile sont tous très ressemblants. La réponse aérodynamique pour chacun de ces modes est également très proche. Une méthode par bloc sera peut-être capable de trouver les similitudes de chacune de ces solutions pour accélérer la résolution.

3.3.2 Description

L'algorithme du block-GMRES est très similaire au GMRES standard. Seules quelques différences dans la méthode d'Arnoldi sont à noter.

Définitions

On appelle un vecteur bloc un vecteur à s colonnes de longueur N. Un produit matrice-vecteur bloc nécessite $N^2 \times s$ opérations, soit s fois plus que pour un produit matrice-vecteur.

Deux vecteurs blocs $\mathbf{u}$ et $\mathbf{v}$ sont dits orthonormaux lorsque

$$\mathbf{u}^T\mathbf{v} = \mathbf{I}_s \tag{3.25}$$

$\mathbf{I}_s$ est la matrice identité de taille s. Lorsque l'on considère des vecteurs complexes, le transconjugué $\mathbf{u}^H$ (vecteur transposé et conjugué) est utilisé. La norme utilisée pour les vecteurs blocs est la norme de Frobénius, notée $\|\cdot\|_F$. Si l'on note $\mathbf{v}_1, \cdots, \mathbf{v}_s$ les s colonnes du vecteur bloc $\mathbf{v}$, alors

$$\|\mathbf{v}\|_F = \sqrt{\sum_{i=1}^{s} \|\mathbf{v}_i\|_2^2} \tag{3.26}$$

Espace de Krylov de vecteur bloc et méthode d'Arnoldi

L'algorithme du block-GMRES est basé sur la minimisation du résidu sur l'espace de Krylov bloc. Ce dernier est toujours défini comme l'espace généré par les itérés d'un vecteur bloc $\mathbf{r}$ initial avec la matrice $\mathbf{A}$. L'espace de Krylov bloc de degré m est noté $\mathcal{B}_m(\mathbf{A}, \mathbf{r})$ et a pour définition

$$\mathcal{B}_m(\mathbf{A}, \mathbf{r}) = \text{bloc vect} \left\{ \mathbf{r}, \mathbf{A}\mathbf{r}, \mathbf{A}^2\mathbf{r}, \ldots, \mathbf{A}^{m-1}\mathbf{r} \right\} \tag{3.27}$$

La notation « bloc vect » indique l'espace vectoriel généré par les m vecteurs blocs, qui est donc de dimension au maximum $m \times s$. En d'autres termes, la définition de l'opérateur bloc vect est la suivante [70] :

$$\mathbf{u} \in \mathcal{B}_m(\mathbf{A}, \mathbf{r}) \iff \exists(\gamma_k) \in \mathbb{R}^{(s\times s)\times m}, \mathbf{u} = \sum_{i=0}^{m-1} \mathbf{A}^i \mathbf{r} \gamma_i \tag{3.28}$$

Cela veut dire que chaque colonne du vecteur bloc $\mathbf{u}$ est une combinaison linéaire de toutes les colonnes des vecteurs blocs $\mathbf{A}^i\mathbf{r}$. Ainsi, cela permet à la résolution de chaque second membre de profiter de la contribution de tous les autres seconds membres.

La méthode d'Arnoldi pour les vecteurs blocs est similaire à la méthode d'Arnoldi modifiée présentée dans l'algorithme 2. Vu la définition d'orthonormalité entre deux vecteurs blocs donnée par l'équation (3.25), un vecteur bloc $\mathbf{w}$ est normal si et seulement si ses colonnes sont orthonormales entre elles. Ainsi, dans la procédure d'Arnoldi modifiée, la normalisation des vecteurs se fait par une décomposition QR de la matrice $\mathbf{w}$. Cette décomposition peut se faire de manière efficace par une méthode d'Arnoldi appliquée aux vecteurs colonnes de $\mathbf{w}$ [63].

La méthode d'Arnoldi bloc produit une matrice d'Hessenberg, définie par des blocs $\boldsymbol{\eta}_{ij}$ de taille $s \times s$, qui sont issus de l'orthogonalisation du nouveau vecteur par rapport aux précédents ou de sa normalisation par décomposition QR (facteur R).

L'algorithme block-GMRES

Une fois cette base de Krylov calculée, la méthode block-GMRES trouve une nouvelle approximation de la solution $\mathbf{x}_m$ dans l'espace affine $\mathbf{x}_0 +$

Algorithme 5 Méthode d'Arnoldi bloc modifiée

1: $\mathbf{v}_1\rho_1 \leftarrow \mathrm{QR}(\mathbf{v}_1)$ ▷ Décomposition QR de $\mathbf{v}_1$
2: **for** $j = 1,m$ **do**
3: $\quad\mathbf{w}_j \leftarrow \mathbf{A}\mathbf{v}_j$
4: $\quad$**for** $i = 1,j$ **do**
5: $\quad\quad\boldsymbol{\eta}_{ij} = \mathbf{v}_i^T\mathbf{w}_j$
6: $\quad\quad\mathbf{w}_j \leftarrow \mathbf{w}_j - \boldsymbol{\eta}_{ij}\mathbf{v}_i$
7: $\quad$**end for**
8: $\quad\mathbf{v}_{j+1}\boldsymbol{\eta}_{j+1,j} \leftarrow \mathrm{QR}(\mathbf{w}_j)$
9: **end for**

$\mathcal{B}_m(\mathbf{A}, \mathbf{r}_0)$ en minimisant le résidu.

L'algorithme block-GMRES est similaire à l'algorithme GMRES. La matrice $\overline{\mathbf{H}}_m$ devient une matrice d'Hessenberg par bloc. Pour obtenir directement le résidu à chaque itération en rendant la matrice d'Hessenberg diagonale, des simples rotations de Givens ne suffisent plus. Utiliser des réflexions de Householder est plus rapide et plus efficace [71]. Une réflexion de Householder permet de mettre à zéro une partie d'une colonne d'une matrice [63]. Il suffit de s réflexions de Householder annulant $2s$ termes par colonnes de la matrice de Hessenberg pour rendre cette dernière diagonale. Si des rotations de Givens avaient été utilisées, il en aurait fallu $O(s^2)$, au détriment de la rapidité et de l'erreur numérique [71]. L'algorithme 6 détaille les étapes de la méthode block-GMRES.

3.3.3 Déflation des valeurs propres

L'algorithme block-GMRES souffre tout autant que le GMRES des redémarrages standard. L'extension de la déflation des petites valeurs propres à l'algorithme par bloc se fait sans problème. L'extension a été réalisée par Morgan dans [116]. Il étend le résultat obtenu dans [115] au cas de vecteurs blocs : les vecteurs de Ritz harmoniques suivis d'un vecteur de résidu, tous orthonormalisés, forment bien un espace de Krylov.

Les vecteurs de Ritz harmoniques, pour le cas du block-GMRES, sont les vecteurs propres de la matrice suivante :

$$\mathbf{H}_m + \mathbf{f}_m\boldsymbol{\eta}_{m+1,m}^T\boldsymbol{\eta}_{m+1,m}\tilde{\mathbf{e}}_m^T \tag{3.29}$$

Le vecteur $\tilde{\mathbf{e}}_m$ est défini comme étant le vecteur de dimension $s \times ms$, où le m$^{\mathrm{e}}$ bloc est la matrice identité de taille s et les autres blocs sont nuls. La démonstration de cette formule suit exactement celle donnée dans l'annexe A, en portant une attention particulière à l'ordre des multiplications (le scalaire $h_{m+1,m}$ étant remplacé par une matrice $\boldsymbol{\eta}_{m+1,m}$). Le vecteur bloc $\mathbf{f}_m$ est défini de manière similaire : $\mathbf{f}_m = \mathbf{H}_m^{-T}\tilde{\mathbf{e}}_m$. Il correspond aux s dernières colonnes de la matrice $\mathbf{H}_m^{-T}$. De la même façon que dans le cas du GMRES

Algorithme 6 Block-GMRES(m) avec redémarrage

1: $\mathbf{v}_0\boldsymbol{\rho}_0 = QR(\mathbf{r}_0)$ ▷ factorisation QR du résidu initial
2: $\mathbf{g} \leftarrow \mathbf{e_1}\boldsymbol{\rho}_0$ ▷ Second membre du problème de moindre carré
3: **for** $j = 1,m$ **do**
4: $\quad \mathbf{w}_j \leftarrow \mathbf{A}\mathbf{v}_j$
5: $\quad$ **for** $i = 1,j$ **do**
6: $\quad\quad \boldsymbol{\eta}_{i,j} \leftarrow \mathbf{w}_j^T\mathbf{v}_i$
7: $\quad\quad \mathbf{w}_j \leftarrow \mathbf{w}_j - \mathbf{v}_i\boldsymbol{\eta}_{i,j-1}$
8: $\quad$ **end for**
9: $\quad \mathbf{v}_j\boldsymbol{\eta}_{j,j-1} = QR(\tilde{\mathbf{v}})$
10: $\quad$ Appliquer les précédentes réflexions de Householder à $(\boldsymbol{\eta}_{k,j}), k = 1,\ldots,j-1$
11: $\quad$ Créer et appliquer la réflexion de Householder $\mathbf{Q}_n$ sur $\begin{pmatrix}\boldsymbol{\eta}_{j-1,j-1}\\ \boldsymbol{\eta}_{j,j-1}\end{pmatrix}$
12: $\quad$ Appliquer la réflexion de Householder $\mathbf{Q}_j$ à $\mathbf{g}$
13: $\quad$ Calcul du résidu : norme du dernier bloc de $\mathbf{g}$
14: $\quad$ **If** résidu inférieur à ε **exit loop**
15: **end for**
16: $\mathbf{y}_j \leftarrow \mathbf{H}_j^{-1}\mathbf{g}$ ▷ Inversion triangulaire
17: Appliquer les réflexions de Householder inverses à $\mathbf{y}_j$
18: $\mathbf{x}_j \leftarrow \mathbf{x}_0 + \mathbf{V}_{j+1}\mathbf{y}_j$ ▷ Nouvelle solution approchée
19: $\mathbf{r}_j \leftarrow \mathbf{V}_{j+1}\mathbf{k}_j$ ▷ Nouveau résidu

standard, on peut calculer ce vecteur à l'aide de l'inverse des transformations ayant servi à rendre $\mathbf{H}_m$ triangulaire supérieure. Cette fois-ci, les réflexions de Householder inverses sont appliquées sur le dernier bloc de $\mathbf{H}_m$.

Les étapes principales du calcul sont rappelées par l'algorithme 7. On constate qu'elles sont toutes similaires à celles du redémarrage dans le cas standard (voir algorithme 4). La principale différence est dans le calcul du résidu dans la base du nouvel espace de Krylov, noté $\mathbf{g}$. Plutôt que de calculer $\mathbf{g} = (\mathbf{V}_k^{new})^T \mathbf{r}_m$ comme indiqué dans [132], une astuce numérique issue de [125] a été utilisée. On remplit $\mathbf{g}$ avec les coefficients d'orthonormalisation du résidu réduit dans la base de Krylov par rapport aux coordonnées dans la même base des vecteurs de Ritz harmoniques. Par les propriétés de la décomposition QR, on a, en notant $\mathbf{g}^{new}$ le résidu réduit dans la nouvelle base :

$$\mathbf{g} - \overline{\mathbf{H}}_m \mathbf{y}_m = \overline{\mathbf{P}}_{k+1} \mathbf{g}^{new} \tag{3.30}$$

Cette formulation est équivalente à celle utilisée pour la déflation de l'algorithme GMRES standard (voir ligne 9 de l'algorithme 4). En effet, $\overline{\mathbf{P}}_{k+1}$ est orthonormale, donc

$$\mathbf{g}^{new} = \overline{\mathbf{P}}_{k+1}^T \Big(\mathbf{g} - \overline{\mathbf{H}}_m \mathbf{y}_m \Big) \tag{3.31}$$

Ainsi,

$$\begin{aligned}
\mathbf{g}^{new} &= \overline{\mathbf{P}}_{k+1}^T \Big(\mathbf{g} - \overline{\mathbf{H}}_m \mathbf{y}_m \Big) \\
&= \overline{\mathbf{P}}_{k+1}^T \mathbf{V}_{m+1}^T \mathbf{V}_{m+1} \Big(\mathbf{g} - \overline{\mathbf{H}}_m \mathbf{y}_m \Big) \\
&= \left(\mathbf{V}_{k+1}^{new}\right)^T \Big(\mathbf{r}_0 - \mathbf{V}_{m+1} \overline{\mathbf{H}}_m \mathbf{y}_m \Big) \\
&= \left(\mathbf{V}_{k+1}^{new}\right)^T \left(\mathbf{r}_0 - \mathbf{A}\mathbf{V}_m \mathbf{y}_m \right) \\
&= \left(\mathbf{V}_{k+1}^{new}\right)^T \mathbf{r}_m
\end{aligned}$$

On retrouve bien la formule précédente. Cette formulation est plus économique, car elle remplace $k+1$ « produits scalaires » de vecteurs blocs de longueur N par une orthonormalisation d'un vecteur bloc de longueur $(m+1) \times s$

3.3.4 Résultats

L'algorithme block-GMRES a été testé sur des cas tests 2D et 3D présentés ci-dessous.

Algorithme 7 Redémarrage avec déflation de k vecteurs de Ritz harmoniques du block GMRES

1: Calcul des k premiers vecteurs de Ritz harmoniques $\mathbf{p}_i$ de $\mathbf{H}_m$
2: Orthonormalisation de $\mathbf{P}_k = (\mathbf{p}_i)_{i=1,k}$
3: Ajout de $\mathbf{g} - \overline{\mathbf{H}}_m \mathbf{y}_m$ à $\mathbf{P}_k$ qui devient $\overline{\mathbf{P}}_{k+1}$
4: Orthonormalisation de la dernière colonne $\overline{\mathbf{P}}_{k+1}$ par rapport aux premières colonnes
5: Stockage des coefficients d'orthonormalisation dans $\mathbf{g}$
6: $\mathbf{V}_{k+1}^{new} \leftarrow \mathbf{V}_{n+1}\overline{\mathbf{P}}_{k+1}$
7: $\overline{\mathbf{H}}_k^{new} \leftarrow \overline{\mathbf{P}}_{k+1}^T \overline{\mathbf{H}}_m \mathbf{P}_k$

Cas test 2D

Cas test I (Profil RAE2822) :
Le profil RAE2822 est un profil 2D transsonique. Il a fait l'objet d'une étude détaillée en soufflerie [35] et a été utilisé dans de nombreux tests numériques [107]. Le maillage utilisé est très fin. Il comprend une zone structurée très fine autour de l'aile, qui permet un calcul précis de la couche limite et du sillage. Ce maillage possède environ 35 000 nœuds. Les équations de Navier-Stokes 2D ont quatre variables par nœud (voir section 2.1). Un calcul Navier-Stokes sur ce maillage utilise donc 140 000 inconnues. Une vue rapprochée de ce maillage est présentée sur la figure 3.4. Le maillage de ce cas est découpé en 8 sous-domaines

Les résultats d'utilisation de l'algorithme du block-GMRES sont présentés dans le tableau 3.1. Quatre seconds membres au maximum ont été utilisés. Ils correspondent respectivement à une augmentation de l'épaisseur, une rotation de prise d'incidence, un mouvement vertical ainsi qu'une rotation de la partie postérieure du profil, qui correspond à un mouvement d'aileron. Les courbes de convergence sont présentées sur la figure 3.5. Les pointillés indiquent le nombre de seconds membres résolus simultanément. Plus les pointillés sont fins, plus il y a de seconds membres.

La première ligne du tableau 3.1, notée « Krylov », désigne la taille de l'espace Krylov par bloc. Il indique le nombre maximum d'itérés avec la matrice $\mathbf{A}$. Cela correspond également à l'indice m de la définition de l'espace de Krylov par bloc $\mathcal{B}_m(\mathbf{A}, \mathbf{r})$ dans l'équation (3.27). Les différents essais sont rangés par taille décroissante de ce paramètre. La deuxième ligne notée s présente le nombre de seconds membres résolus en même temps. Les troisièmes et quatrièmes lignes présentent le temps en seconde de résolution respectivement pour tous les seconds membres et par second membre. La cinquième ligne indique le nombre d'itérations nécessaire pour la convergence du système. Cela donne le nombre de produits matrice-vecteur bloc pour obtenir la solution. La ligne suivante est le nombre moyen de produits matrice-

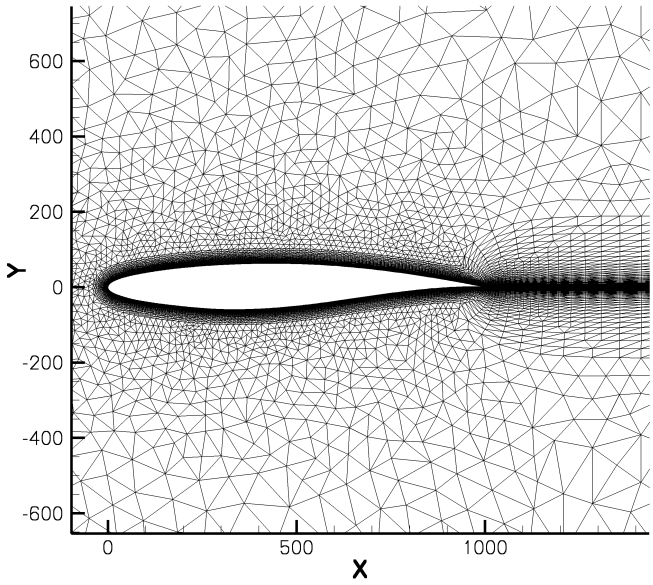


FIGURE 3.4 – Vue rapprochée du maillage 2D du profil RAE2822 (cas test I)

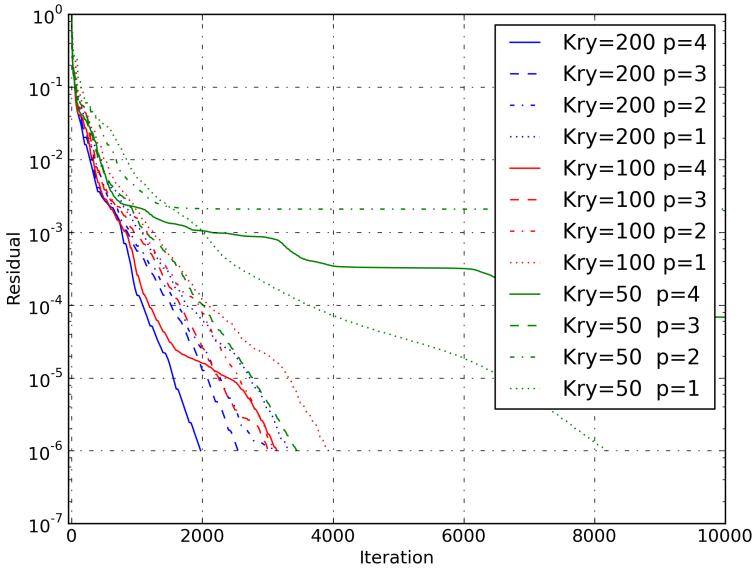


FIGURE 3.5 – Courbes de convergence de l'algorithme block-GMRES pour profil RAE2822 (cas test I). Abscisse : nombre de produits matrice-vecteur bloc. Ordonnée : résidu relatif.

Krylov	200	200	200	200	100	100	100	100
s	4	3	2	1	4	3	2	1
Temps t (s)	356	244	135	35	272	149	61	25
t/s (s)	89	81	68	35	68	50	30	25
Itérations	1974	2546	3174	3315	3140	3021	3092	3931
Itérations/t	5,5	10,4	23,4	95,2	11,6	20,2	51,1	155,2
Itérations*s/t	22,2	31,3	46,9	95,2	46,2	60,7	102,2	155.2

Krylov	50	50	50	50
s	4	3	2	1
Temps (s)	—	80	—	33
t/s (s)	—	27	—	33
Itérations	—	3444	—	8185
Itérations/t	—	43,0	—	245,3
Itérations*s/t	—	128,9	—	245,3

TABLEAU 3.1 – cas RAE2822 (cas test I) : temps de résolution des systèmes avec le block-GMRES. Un tiret indique que le système n'a pas convergé

vecteur bloc par seconde. Enfin, la dernière ligne est simplement la précédente multipliée par le nombre de seconds membres résolus simultanément pour montrer le nombre moyen de produits matrice-vecteur (simple colonne) par seconde.

Les résultats présentés sur la figure 3.5 et le tableau 3.1 montrent tout d'abord que, pour une taille d'espace de Krylov bloc donnée, augmenter le nombre de seconds membres permet de résoudre les systèmes en moins d'itérations. Cela se vérifie parfaitement pour une taille d'espace de Krylov de 200, et également pour 100, à l'exception de la résolution à quatre seconds membres qui présente une inflexion de la courbe de convergence autour de 2000 itérations. Enfin, pour un espace de Krylov à 50 vecteurs blocs, on note que la résolution de 3 seconds membres demande beaucoup moins d'itérations que pour un seul. Cependant, pour cette taille d'espace de Krylov, la convergence n'a pas été possible pour $s = 2$ ou 4.

La diminution du nombre d'itérations ne va pas de pair avec le temps de résolution par second membre, comme le montre la troisième ligne du tableau 3.1. Certaines opérations de l'algorithme block-GMRES sont d'une complexité quadratique en s, comme les produits scalaires de vecteurs blocs ou les opérations sur la matrice d'Hessenberg (réflexions de Householder, etc). La diminution du nombre d'itérations n'est pas suffisante pour contrer

	800 vecteurs			400 vecteurs		
Krylov	200	400	800	100	200	400
s	4	2	1	4	2	1
Temps t (s)	356	208	129	272	135	177
t/s (s)	89	104	129	68	68	177
Itérations	1974	2336	2819	3140	3174	3023
Itérations/t	5,5	11,2	21,8	11,6	23,4	17,1
Itérations*s/t	22,2	22,5	21,8	46,2	46,9	17,1

TABLEAU 3.2 – cas RAE2822 (cas test I) : temps de résolution à espace mémoire égal.

l'augmentation de temps de calcul par opération. La seule exception est pour 50 vecteurs bloc de Krylov, où la résolution de trois seconds membres en même temps diminue de plus de 50% le nombre d'itérations, ce qui permet gagner 10% en temps de calcul par second membre.

La taille mémoire d'un espace de Krylov bloc $\mathcal{B}_m(\mathbf{A}, \mathbf{R})$ à itérés m fixé dépend linéairement du nombre de seconds membres s. Une comparaison à taille mémoire d'espace de Krylov fixée, *i.e.* à $m \times s$ constant est présentée sur le tableau 3.2. On constate pour 800 vecteurs en mémoire une diminution du temps de résolution par second membre. Cela est en grande partie dû à la diminution du nombre d'itérations, mais également à l'augmentation de l'efficacité algorithmique de l'algorithme bloc. En effet, les orthonormalisations et les assemblages parallèles sont groupés. Cela se voit dans la dernière ligne du tableau 3.2. Cette ligne peut s'interpréter comme le nombre de produits matrice-vecteur (à une seule colonne) par seconde en moyenne. On voit que les opérations sur l'ensemble des vecteurs colonne de l'espace de Krylov se font légèrement plus vite lorsque s augmente.

Cas test 3D

Cas test II (Maquette DTP) :
Le cas test 3D est un calcul sur la géométrie de la maquette DTP. Cette maquette représente un fuselage simplifié d'avion, auquel est attachée une aile à profil symétrique (figure 3.6). L'aile en flèche a une demi-envergure d'un mètre. Cette maquette a servi pour des tests en soufflerie pour des expériences de flutter, et de cas test pour la prévision numérique de ce phénomène [38]. Le maillage de ce cas test est représentatif de ceux utilisés par Dassault Aviation pour ses simulations. Il comprend plus de six millions de nœuds, ce qui correspond pour un calcul 3D à 30 millions d'inconnues. Ce cas est découpé en 512 sous-domaines. La configuration aérodynamique

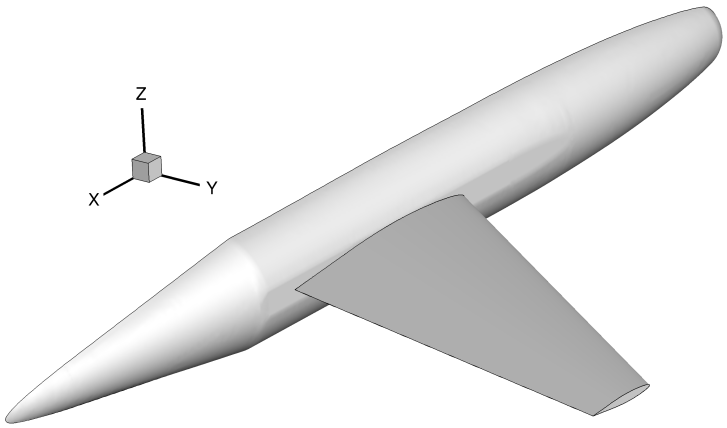


FIGURE 3.6 – Vue générale de la maquette DTP (cas test II).

de référence autour de laquelle ont été réalisés les calculs linéarisés est un nombre de Mach de 0.85 et une incidence de 0°.

Afin de générer des mouvements proches des modes propres de l'aile, des polynômes des coordonnées x, y et z ont été utilisés. La figure 3.3 montre sept d'entre eux. Ils sont une combinaison de mouvements de flexion et de torsion de l'aile. Leur relative ressemblance laisse penser que leur résolution combinée sera source de gain de temps.

La figure 3.7 montre la courbe de convergence obtenue avec l'algorithme block-GMRES. L'espace de Krylov contient 50 vecteurs blocs. Les traits fins sont les résidus de chaque second membre séparé. Le trait épais montre le résidu global de l'algorithme block-GMRES, qui est une norme de Frobénius du vecteur bloc résidu. La formule (3.26) relie les résidus de chaque second membre au résidu global. Le résidu atteint un plateau de convergence à partir de 2500 itérations. Au bout de 4000 itérations, le résidu a diminué de seulement 2,5 ordres de grandeur, ce qui est loin des cinq ordres généralement demandés pour assurer une bonne qualité des résultats. L'algorithme GMRES standard demande moins de 4000 itérations pour converger à la précision demandée. De nombreuses autres combinaisons de taille d'espace de Krylov et de nombre de seconds membres résolus ont été testées, et aucune n'est satisfaisante : la convergence est impossible ou trop lente.

En conclusion, l'algorithme block-GMRES n'a pas donné satisfaction, pour des raisons simples. La déflation des petites valeurs propres dans l'algorithme GMRES permet de diminuer énormément la taille de l'espace de Krylov. Actuellement, cela permet d'utiliser des espaces de Krylov à 100 vecteurs ou moins pour des applications industrielles. Le coût de l'ortho-

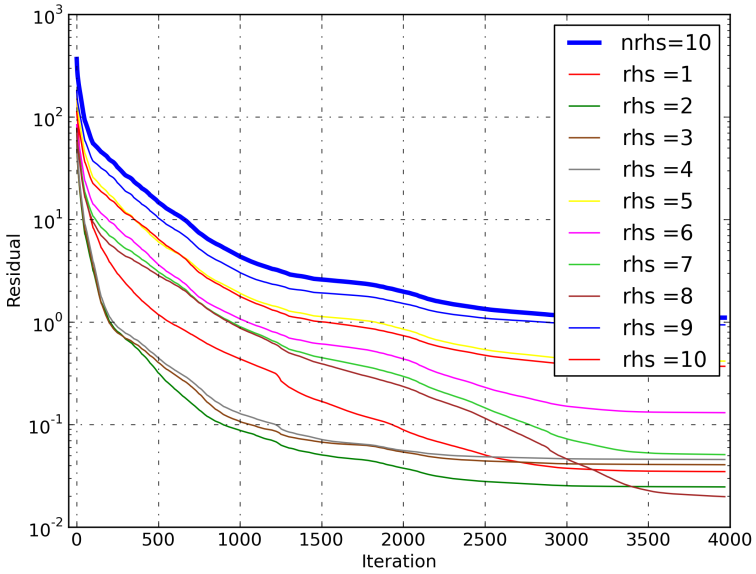


FIGURE 3.7 – Maquette DTP (cas test II) : résidu de l'algorithme block-GMRES pour 10 seconds membres. 50 vecteurs de Krylov. Trait épais : résidu global. Trait fin : résidu partiel de chaque second membre

normalisation des vecteurs dans la méthode d'Arnoldi est proportionnel au carré de la taille de l'espace de Krylov. Un petit espace de Krylov permet d'accélérer significativement les résolutions.

La taille de l'espace de Krylov bloc utilisé par l'algorithme block-GMRES est de $m \times s$ vecteurs colonnes, où m et s sont respectivement le nombre d'itérés avec la matrice $\mathbf{A}$ et le nombre de seconds membres. Les opérations d'orthonormalisation, même si elles sont groupées, dépendent quadratiquement de cette taille. Pour avoir des résolutions très rapides, il faut impérativement garder cette taille petite. Cela impose de limiter le nombre de seconds membres résolus simultanément, ainsi que l'ordre m de l'espace de Krylov bloc. Les conséquences sont doubles : le gain espéré par la résolution simultanée de plusieurs seconds membres ne peut être atteint, et la précision de l'estimation des petites valeurs propres de $\mathbf{A}$ est faible, donc la convergence est mauvaise.

Cette conclusion rejoint les limites soulevées par Morgan au point 1 de la partie 2.2 dans [116]. Le block-GMRES ne peut être compétitif pour un problème difficile dont la convergence est limitée par les petites valeurs propres. Cette affirmation est corroborée par l'efficacité de la déflation pour le GMRES. Alors, explique Morgan, le block-GMRES ne sera pas compétitif par rapport au GMRES notamment si l'espace mémoire est limité.

3.3.5 La déflation des seconds membres

Durant la résolution de plusieurs seconds membres par l'algorithme block-GMRES, il peut arriver que les résidus soient quasiment linéairement dépendants. Il est alors possible de résoudre uniquement les seconds membres indépendants, et de reconstruire la solution à l'aide de la relation de dépendance linéaire pour les autres seconds membres.

La déflation des seconds membres consiste à identifier les systèmes qui dépendent linéairement des autres, et à les éliminer de l'espace de Krylov généré. Cette méthode a été utilisée pour la première fois pour une méthode de type quasi-minimum residual (QMR) par bloc dans [55].

Les dépendances linéaires exactes sont rares, et conduiraient à l'arrêt de la méthode block-GMRES, puisque la normalisation du vecteur par décomposition QR serait impossible. On peut trouver ces quasi-dépendances linéaires en utilisant une décomposition QR avec permutation des colonnes, appelées factorisation QR révélatrice du rang (*rank revealing QR-factorizations*). Cette méthode consiste à trouver la matrice de permutation $\boldsymbol{\pi}$ de taille $s \times s$ et le rang s_0 tels que la matrice $\mathbf{R}$ ait un bloc inférieur droit de petite norme, c'est-à-dire :

$$\mathbf{r}\boldsymbol{\pi} = \mathbf{QR} = \mathbf{Q}\begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{R}_{22} \end{pmatrix} \tag{3.32}$$

La matrice $\mathbf{R}$ est définie par bloc et la sous-matrice carrée $\mathbf{R}_{22}$ est de taille $s_0 \times s_0$. Cette dernière est de petite norme, *i.e.* $\|\mathbf{R}_{22}\| < \epsilon$ [63]. Cela veut dire que les $s - s_0$ derniers vecteurs de $\mathbf{A}\boldsymbol{\pi}$ ont une très petite composante orthonormale à l'espace vectoriel généré par les s_0 premiers vecteurs. Suivant le choix de la norme et de la tolérance ϵ, on peut dire que le vecteur $\mathbf{r}$ est de rang s_0. Différents algorithmes de factorisation QR révélatrice du rang existent. Ce sont en général des heuristiques pour trouver la bonne permutation de colonnes. On pourra consulter [20, 32] pour une vue d'ensemble sur ces méthodes. Les premiers algorithmes sont ceux de Golub [62] et de Chan [29]. Notons enfin que si un second membre converge avant les autres, sa norme est très petite relativement aux autres colonnes. Il sera donc identifié comme quasi-linéairement dépendant des autres.

La déflation peut se produire à deux moments dans l'algorithme du block-GMRES : soit au redémarrage, en identifiant une perte de rang dans le résidu de départ $\mathbf{r}_0$, soit au cours des itérations d'Arnoldi, cette fois-ci dans le nouveau vecteur $\mathbf{w}_j$. D'après Gutknecht [70], ces dernières sont moins probables. Robbé et Sadkane [131] étudient la déflation à chaque itération en utilisant deux critères au choix : soit la décomposition QR révélatrice du rang est appliqué à l'ensemble de l'espace de Krylov bloc $\mathcal{B}_m$ (critère W), soit elle est appliquée au résidu $\mathbf{r}$. Quelle que soit la méthode, les vecteurs éliminés doivent être gardés pour reconstruire la solution, au risque d'empêcher la convergence [131].

Cette forme de déflation peut être combinée avec la déflation des petites valeurs propres, telle que décrit dans la section 3.3.3. Agullo, Giraud et Jing [2] prouvent qu'une déflation des petites valeurs propres au sens de Morgan [116] s'associe avec la déflation utilisant le critère R de [131]. En effet, ils prouvent que les vecteurs de Ritz harmoniques constituent un espace de Krylov.

La déflation des seconds membres en combinaison avec celles des petites valeurs propres n'a pas été implantée pour plusieurs raisons. Des tests ont été effectués sur les résidus initiaux après chaque redémarrage, afin d'identifier s'ils avaient des colonnes quasi-dépendantes. Cela ne s'est jamais produit. De plus, les seconds membres ont des convergences très similaires, comme le montre la figure 3.7. Aucun second membre n'atteint un résidu négligeable devant les autres. Enfin, sur les cas tests 3D utilisés, le block-GMRES a toujours montré des problèmes de convergence, alors que les exemples de [131] convergent très bien.

Il existe une autre méthode de résolution de systèmes linéaires, appelée GCRO-DR [124], qui permet de s'affranchir de la contrainte du GMRES avec déflation, à savoir que les vecteurs propres de la matrice doivent être compris dans la base de Krylov précédente, et reforment un nouvel espace de Krylov [115]. Elle permet d'utiliser des vecteurs quelconques, et ainsi un recyclage des informations spectrales de l'opérateur entre plusieurs résolutions successives. Elle est entièrement équivalente au GMRES avec déflation, si ce recyclage n'est pas utilisé. Cette méthode a été appliquée à des problèmes de résolution des équations de Navier-Stokes linéarisées avec une discrétisation par volumes finis [159]. Les gains de cette méthode pourraient cependant être limités, car des résolutions successives d'un même système linéaire n'ont pas forcément été accélérées [159].

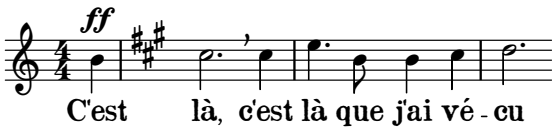
3.4 Conclusion

Ce chapitre a présenté la méthode itérative de résolution de système linéaire utilisée dans AeTher, appelée GMRES. Cette méthode minimise le résidu sur la base de l'espace de Krylov. Cet espace est généré par les itérés successifs de la solution initiale avec la matrice du système. Il serait trop coûteux, tant en temps de calcul qu'en espace mémoire, de constituer une base de Krylov contenant tous les vecteurs nécessaires pour atteindre le seuil de convergence demandé. Ainsi, des redémarrages sont effectués en prenant comme approximation initiale la solution approchée lorsque la taille maximale de l'espace de Krylov est atteinte. Ces redémarrage gênent la convergence, car ils ne retiennent aucune autre information sur le système que le vecteur de solution. L'information sur le spectre de la matrice contenue dans l'espace de Krylov et la matrice d'Hessenberg associée peut être réutilisée. On génère des approximations des vecteurs propres de plus petite valeur propre

afin d'éliminer leur action sur le système par un procédé appelé déflation. Une bonne implémentation de cette méthode est capitale pour garantir la convergence de l'algorithme GMRES.

Dans cette thèse, une extension de l'algorithme GMRES, appelé block-GMRES pour résoudre plusieurs seconds membres simultanément a été testée. L'utilisation de cette méthode a été pressentie en aéroélasticité, puisque plusieurs formes modales doivent être calculées à la même fréquence et au même point de vol, c'est-à-dire que tous les systèmes linéaires correspondants ont la même matrice. L'algorithme block-GMRES permet *a priori* d'exploiter les similitudes entre les seconds membres et de générer des espaces de Krylov plus riches, puisque les solutions de chacun des seconds membres pourront se servir de l'information générée par tous les autres. Enfin, les opérations informatiques sont plus efficaces par leur regroupement, ce qui diminue les temps d'accès à la mémoire ou encore le temps de communication parallèle. L'implémentation astucieuse de la déflation des plus petits vecteurs propres a été transposée dans le cadre de cette thèse au cas block-GMRES.

Les tests numériques de cette méthode ne sont malheureusement pas convaincants. La richesse accrue de l'espace de Krylov par bloc ne s'accompagne pas d'une accélération de la convergence. Pire encore, une perte de robustesse est constatée. L'accélération informatique est bien là, mais ne contrebalance pas le ralentissement de l'orthonormalisation due à un plus grande nombre de vecteurs de Krylov. L'algorithme GMRES avec déflation des petites valeurs propres permet d'utiliser une petite taille d'espace de Krylov (typiquement une centaine). Pour être compétitif en mémoire et en temps de calcul, il faudrait pour l'algorithme block-GMRES diminuer fortement le nombre maximal d'itérés, au prix d'une perte de précision sur l'approximation des plus petites valeurs propres pour la déflation. Si les vecteurs résidus finissent par être quasi colinéaires, la déflation de second membre les élimine et permet d'accélérer la méthode. Cette colinéarité n'a pas été détectée lors de test effectués avec une factorisation QR révélant le rang.



Duparc, *La vie antérieure*

Chapitre 4

Préconditionnement

4.1 Introduction

Le préconditionnement consiste à transformer le système linéaire en une forme équivalente qui est numériquement plus facile à résoudre. Comme expliqué dans la partie 3.2.3, la matrice $\mathbf{A}$ du système linéaire $\mathbf{Ax} = \mathbf{b}$ est multipliée à droite ou à gauche par une matrice $\mathbf{M}^{-1}$ choisie afin que leur produit approche l'identité. La section 3.2 a montré que la méthode GMRES est très sensible aux petites valeurs propres de la matrice. Un bon préconditionnement permet d'éviter d'avoir de trop petites valeurs propres nuisibles à la convergence.

Une bonne matrice de préconditionnement est également rapide à former et le produit de cette matrice par un vecteur est peu coûteux. En effet, dans la méthode GMRES, la matrice de préconditionnement est appliquée après chaque produit matrice-vecteur dans le cas d'un préconditionnement à gauche (ou avant lors d'un préconditionnement à droite). Les résolutions avec l'algorithme GMRES nécessitent des centaines voire des milliers d'itérations. Il faut donc que l'application du préconditionnement soit rapide.

De même, le temps de création du préconditionneur doit être limité. Le cas limite est $\mathbf{M}^{-1} = \mathbf{A}^{-1}$, qui correspond à l'utilisation d'une méthode directe pour inverser la matrice $\mathbf{A}$. L'algorithme GMRES aura alors besoin d'une seule itération pour converger. L'inverse d'une matrice creuse est généralement plein. Un préconditionnement de bonne qualité nécessitera a priori beaucoup de mémoire.

Il est donc nécessaire de trouver un bon compromis entre qualité de préconditionnement, et coûts de formation et d'application, tant en mémoire qu'en temps. Cela fait dire à Saad dans [135] que

> *Finding a good preconditioner to solve a given sparse linear system is often viewed as a combination of art and science. Theoretical*

results are rare and some methods work surprisingly well, often despite expectations.

Il existe de nombreuses méthodes pour approcher l'inverse d'une matrice. Elles sont abordées dans [18, 135]. Une façon naturelle de concevoir un préconditionneur est d'utiliser une méthode d'inversion de matrice que l'on dégrade afin qu'elle soit rapide. On peut citer les méthodes dites SOR (*Successive Over Relaxation*). La matrice de préconditionnement associée correspond à une passe de l'algorithme SOR [135]. Les méthodes ILU [112], abordées plus longuement plus tard dans ce chapitre, sont une adaptation de la factorisation LU à des matrices creuses, où l'on garde uniquement les facteurs qui rentrent dans le squelette de la matrice à factoriser. Enfin, toute méthode de résolution de système linéaire peut être utilisée à une convergence très faible pour approcher l'effet de l'inverse de la matrice sur un vecteur. Par exemple, l'utilisation de méthodes de multi-grilles algébriques (AMG pour *Algebraic Multi Grid*) connaissent un grand succès comme préconditionneur pour d'autres solveurs itératifs [152]. Leur inconvénient majeur est qu'une méthode de résolution approchée ne garantit pas de toujours obtenir le même préconditionneur. Il faut alors utiliser un solveur qui accepte la variation du précondionnement au cours des itérations, comme par exemple le *flexible* GMRES [134].

D'autres méthodes ne sont pas issues de solveur ou de factorisations. Par exemple, le préconditionneur SPAI (*SParse Approximate Inverse*) [19, 65] cherche à minimiser au sens d'une certaine norme l'écart entre la matrice préconditionnée et l'identité, sous contrainte d'un même squelette que la matrice.

Enfin, le préconditionnement peut se baser sur une matrice plus simple que $\mathbf{A}$. Les méthodes de volumes finis d'ordre élevé conduisent à des matrices plus pleines qu'avec des éléments finis linéaires. La travail de factorisation approchée est alors bien plus coûteux. Utiliser une matrice issue d'une discrétisation d'ordre plus faible permet d'avoir un préconditionnement plus léger et néanmoins efficace [96, 102].

4.2 Les préconditionneurs existant dans AeTher

Plusieurs préconditionneurs existent dans AeTher. Le premier, le préconditionnement diagonal par bloc d'inconnues aux nœuds, est toujours utilisé. Un autre préconditionnement, le block-SOR ou l'ILU(0), peut être utilisé en plus de celui-ci.

4.2.1 Préconditionnement bloc diagonal

Le préconditionnement diagonal approche l'inverse d'une matrice par l'inverse de sa diagonale, qui est triviale à calculer. Si l'on note $\mathbf{D}$ la diagonale

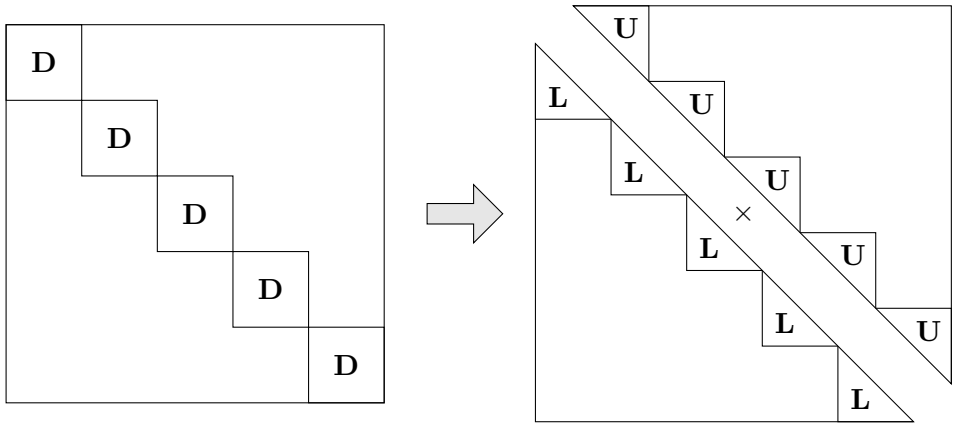


FIGURE 4.1 – Décomposition LU de la diagonale par bloc

de $\mathbf{A}$, le système préconditionné à gauche s'écrit

$$\mathbf{D}^{-1}\mathbf{A}\mathbf{x} = \mathbf{D}^{-1}\mathbf{b} \tag{4.1}$$

Cela correspond à renormer les lignes de la matrice. Le préconditionnement à droite correspond à une renormalisation des colonnes. Pour normer les lignes et les colonnes de $\mathbf{A}$, si les termes de $\mathbf{D}$ sont tous positifs, on peut préconditionner à gauche et à droite comme suit

$$\begin{cases} \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{y} = \mathbf{D}^{-\frac{1}{2}}\mathbf{b} \\ \mathbf{x} = \mathbf{D}^{-\frac{1}{2}}\mathbf{y} \end{cases} \tag{4.2}$$

La matrice issue de la discrétisation par la méthode des élément finis des équations de Navier-Stokes est définie par bloc, comme exposé dans la section 2.2.1. Il est alors naturel d'extraire la matrice des blocs diagonaux. L'inverse de cette matrice diagonale par blocs se calcule par une décomposition LU de ses blocs, qui est illustrée par la figure 4.1. On obtient deux matrices, notées $\mathbf{D_L}$ et $\mathbf{D_U}$, telles que $\mathbf{D_L}\mathbf{D_U} = \mathbf{D}$. Le préconditionnement droite-gauche diagonal par bloc s'obtient naturellement :

$$\begin{cases} {\mathbf{D_L}}^{-1}\mathbf{A}{\mathbf{D_U}}^{-1}\mathbf{y} = {\mathbf{D_L}}^{-1}\mathbf{b} \\ \mathbf{x} = {\mathbf{D_U}}^{-1}\mathbf{y} \end{cases} \tag{4.3}$$

Le préconditionnement diagonal par bloc est systématiquement utilisé dans le code AeTher, avant l'application d'un autre préconditionneur. Plutôt que d'appliquer le préconditionnement diagonal par bloc à chaque produit matrice-vecteur, la matrice est modifiée. Cette stratégie est possible car le préconditionnement diagonal ne change pas le squelette de la matrice, et le produit matriciel ${\mathbf{D_L}}^{-1}\mathbf{A}{\mathbf{D_U}}^{-1}$ peut être calculé en une seule boucle sur tous les éléments de $\mathbf{A}$.

4.2.2 block-SOR

La méthode SOR (*Successive Over Relaxation*) est une méthode itérative de résolution de système linéaire par point fixe [135]. Elle peut également être mise sous la forme d'une matrice de préconditionnement. Comme indiqué dans la section précédente, le code AeTher modifie systématiquement la matrice $\mathbf{A}$ par un préconditionneur bloc diagonal. Sa diagonale est donc l'identité par bloc. On peut alors décomposer additivement cette matrice comme suit :

$$\mathbf{A} = \mathbf{I} + \mathbf{L} + \mathbf{U} \tag{4.4}$$

Les matrices $\mathbf{L}$ et $\mathbf{U}$ sont les termes de la matrice $\mathbf{A}$ respectivement situés strictement au-dessous et au-dessus de sa diagonale par bloc. Dans ce cas, la matrice de préconditionnement SOR est définie [135] :

$$\mathbf{M}_{SOR} = (\mathbf{I} + \omega\mathbf{L})\,(\mathbf{I} + \omega\mathbf{U}) \tag{4.5}$$

où $\omega > 0$ est appelé paramètre de relaxation. La matrice $\mathbf{M}_{SOR}$ réalise une factorisation approchée de $\mathbf{A}$:

$$\mathbf{M}_{SOR} = (\mathbf{I} + \omega\mathbf{L})\,(\mathbf{I} + \omega\mathbf{U}) = \mathbf{I} + \omega(\mathbf{L} + \mathbf{U}) + \omega^2\mathbf{L}\mathbf{U} \tag{4.6}$$

Si $\omega \approx 1$, alors $\mathbf{M}_{SOR} \approx \mathbf{A} + \mathbf{L}\mathbf{U}$. Cette factorisation approchée est d'autant meilleure que les termes extra-diagonaux de $\mathbf{A}$ (qui entrent en jeu dans le produit $\mathbf{L}\mathbf{U}$) sont petits. Si cette dernière hypothèse n'est pas satisfaite, choisir ω petit permettra de diminuer l'influence du terme $\omega^2\mathbf{L}\mathbf{U}$, mais alors $\mathbf{I} + \omega(\mathbf{L} + \mathbf{U})$ ne sera plus une aussi bonne approximation de $\mathbf{A}$.

4.2.3 ILU(0)

Le code AeTher est également doté d'un préconditionneur de type ILU(0) (pour *Incomplete LU*). La décomposition LU complète d'une matrice creuse est possible. Le solveur direct MUMPS utilise cette méthode [7, 9]. Néanmoins, les matrices $\mathbf{L}$ et $\mathbf{U}$ sont alors beaucoup plus remplies que la matrice originale, et leur calcul nécessite beaucoup plus d'espace mémoire et de temps qu'une décomposition incomplète, présentée ci-dessous.

La décomposition ILU(0) est présentée dans l'algorithme 8. Il consiste à réaliser un pivot de Gauss pour trouver la décomposition LU de $\mathbf{A}$, en ne conservant que les termes qui rentrent dans le squelette $\mathcal{S}$ de $\mathbf{A}$. À la fin de l'algorithme, les matrices $\mathbf{L}$ et $\mathbf{U}$ sont stockées respectivement dans les parties triangulaires inférieure et supérieure de la matrice $\mathbf{A}$ fournie. La version bloc de l'algorithme découle immédiatement de l'algorithme 8 en considérant que $\mathbf{a}_{ij} \in \mathbb{R}^{5\times5}$ (en 2D $\mathbb{R}^{4\times4}$) au lieu de $a_{ij} \in \mathbb{R}$. Une factorisation LU des blocs diagonaux permet de calculer l'inversion $\mathbf{a}_{kk}^{-1}\mathbf{a}_{ik}$. L'une des premières

Algorithme 8 Décomposition ILU(0) de la matrice $\mathbf{A}$ de masque $\mathcal{S}$

```
1: for i = 2, ... n do
2:     for k = 1, ... i − 1 / (i, k) ∈ S do
3:         a_ik ← a_kk^{-1} a_ik
4:         for j = k + 1, ... n / (i, j) ∈ S do
5:             a_ij ← a_ij − a_ik a_kj
6:         end for
7:     end for
8: end for
```

analyses du préconditionnement ILU(0) est due à Meijerink et van der Worst [112].

L'algorithme 8 possède plusieurs variantes d'ordre de passage dans la matrice à l'aide de ses trois coefficients i,j et k. Les variantes sont couramment appelées jki, jik et kji [113, 63, 135]. Certaines sont plus efficaces que d'autres en fonction du format de stockage de la matrice et des capacités de calcul vectoriel de la machine. L'algorithme 8 est présenté sous la variante ikj.

Sur le masque $\mathcal{S}$ de $\mathbf{A}$, le résidu $\mathbf{R} = \mathbf{A} - \mathbf{LU}$ de la factorisation est nul [33]. Autrement dit, si l'on note $\mathbf{R} = (r_{ij})$ ce résidu matriciel, alors

$$\forall\,(i,j) \in \mathcal{S}, r_{ij} = 0 \tag{4.7}$$

Cette propriété sert aux auteurs de [33] pour utiliser une méthode de point fixe diminuant la norme l_2 de ce résidu afin de calculer une décomposition ILU de façon approchée, compatible avec une parallélisation en mémoire partagée.

Une décomposition ILU n'est pas exacte (sauf cas simples particuliers, voir [135]). En effet, la matrice produit $\mathbf{LU}$ contient des éléments supplémentaires, c'est-à-dire $\exists (i,j) \notin \mathcal{S}, r_{ij} \neq 0$. L'erreur de décomposition peut être répartie autrement, comme pour le MILU (pour *Modified ILU*) [69, 135]. Si cee résidu est sommé par ligne à la fin de l'algorithme et ajouté à la diagonale de $\mathbf{U}$, alors $\mathbf{A1} = \mathbf{LU1}$, où $\mathbf{1}$ est le vecteur rempli uniquement de 1. Le préconditionnement MILU n'introduira pas d'erreur pour les fonctions constantes. Cela peut être intéressant pour certains problèmes physiques.

Le préconditionneur décrit par l'algorithme 8 s'appelle ILU(0) car il n'introduit pas de remplissage supplémentaire dans le squelette des décompositions $\mathbf{L}$ et $\mathbf{U}$. Pour améliorer la qualité de la factorisation, on peut introduire des termes supplémentaires. Cette stratégie sera évoquée dans la section 4.5.1.

La numérotation des nœuds affecte la qualité du préconditionnement ILU(0) (et des ses variantes avec remplissage) [10, 39, 135]. L'ordre dans lequel est effectuée l'élimination gaussienne importe sur la création de termes

supplémentaires qui ne sont pas considérés dans le cas d'une factorisation inexacte. Si peu de termes supplémentaires sont créés, alors la factorisation inexacte est de meilleure qualité. L'élimination gaussienne trouve son équivalent dans la théorie des graphes [10, 133]. De nombreuses heuristiques ont été développées pour trouver un ordre des nœuds qui minimise le remplissage. Parmi ces méthodes de renumérotation, on peut citer le *Reverse Cuthill-McKee ordering* (RCM) [36], le *Minimum Degree ordering* [8] ou encore *Nested Dissection* [101], qui trouve écho dans la décomposition de domaine [90].

Remarque 4.1. Dans le cas d'AeTher, le préconditionnement ILU(0) n'a jamais correctement fonctionné pour les applications Navier-Stokes linéarisées. L'un des apports de cette thèse, décrit dans la partie 4.5.1, est de comprendre que cela était dû au remplissage insuffisant de l'ILU. Notons que différents algorithmes de renumérotation ont été testés pour l'ILU(0) (et sa variante avec remplissage introduit dans la section 4.5.1), sans apporter de bénéfice lors de la résolution.

4.3 Parallélisation du préconditionnement

Les méthodes de préconditionnement présentées dans la section précédente ne traitaient que du cas séquentiel, c'est-à-dire lorsque la matrice complète est disponible. Comme expliqué dans la section 2.4, la matrice du problème linéarisé est découpée en blocs, chacun étant sur un processeur différent.

Algorithme 9 Résolution du système triangulaire inférieur $\mathbf{Lx} = \mathbf{b}$

1: $\mathbf{x} \leftarrow \mathbf{0}$
2: **for** $i = 1, \ldots n$ **do**
3: **for** $k = 1, \ldots i-1$ **do**
4: $x_i \leftarrow x_i - L_{ij}x_j$
5: **end for**
6: $x_i \leftarrow L_{ii}^{-1}b_i$
7: **end for**

L'application d'un préconditionneur ILU ou BSOR nécessite la résolution de deux systèmes triangulaires. L'algorithme 9 permet de résoudre un système triangulaire inférieur. C'est un algorithme séquentiel, qui parcourt successivement les lignes du système en utilisant les x_j des lignes précédemment traitées. Cet algorithme est difficilement parallélisable. Une interdépendance complexe se crée entre les blocs. La résolution d'une ligne i peut faire intervenir les inconnues x_j déjà résolues mais stockées par d'autres processeurs. Une implémentation efficace se fait par coloriage des domaines et renumérotation des inconnues pour que les inconnues de bord (*i.e.* partagées par plusieurs domaines) soient à la fin. On consultera par exemple [135, chap. 12] pour plus

détails. On retiendra qu'une résolution parallèle de systèmes triangulaires est d'implémentation délicate [33] et nécessite beaucoup de communications entre les processeurs. Une technique plus simple de parallélisation de préconditionnement est la méthode de Schwarz additif, présentée dans la section suivante.

4.3.1 Schwarz additif

Cette introduction à la méthode de Schwarz additif suit la présentation donnée dans [44, 73], où l'on trouvera les preuves associées. On pourra également consulter les *review paper* suivants [93] pour une discussion sur l'architecture matérielle et son impact sur les temps de calcul et [30] pour un aperçu plus récent, ou le livre [156].

Méthode de Schwarz additif pour le problème continu

La méthode de Schwarz additif est basée sur des travaux d'Hermann Schwarz concernant la résolution de l'équation de Poisson [142] sur l'union de deux domaines simples $\Omega = \Omega_1 \cup \Omega_2$ présentés sur la figure 4.2. Sur chaque domaine, une résolution indépendante est effectuée, en prenant comme condition limite la valeur de la solution précédente sur les autres domaines là où ils se recouvrent. C'est une méthode itérative, dont Schwarz a prouvé la convergence.

Soit le problème suivant, dit global

$$\begin{cases} \Delta u = f & \text{sur} \quad \Omega \\ u = 0 & \text{sur} \quad \Gamma \end{cases} \tag{4.8}$$

Le domaine Ω est décomposé en deux sous-domaines Ω_1 et Ω_2 qui se recouvrent. Soit Γ la frontière de $\Omega = \Omega_1 \cup \Omega_2$. On note $\Gamma_1 = \partial\Omega_1 \cap \Omega_2$ la frontière de Ω_1 comprise dans l'intérieur de Ω_2. De même, on définit $\Gamma_2 = \partial\Omega_2 \cap \Omega_1$. Les frontières entre les sous-domaines sont également appelées interfaces parallèles. On note $\left(u_1^k\right)$ et $\left(u_2^k\right)$ deux suites d'approximations de la solution u de (4.8) sur respectivement Ω_1 et Ω_2, dont le premier terme est la fonction nulle et que l'on définit itérativement par :

$$\begin{cases} \Delta u_1^{k+1} = f & \text{sur} \quad \Omega_1 \\ u_1^{k+1} = 0 & \text{sur} \quad \Gamma \cap \partial\Omega_1 \\ u_1^{k+1} = u_2^k & \text{sur} \quad \Gamma_1 \end{cases} \quad \text{puis} \quad \begin{cases} \Delta u_2^{k+1} = f & \text{sur} \quad \Omega_2 \\ u_2^{k+1} = 0 & \text{sur} \quad \Gamma \cap \partial\Omega_2 \\ u_2^{k+1} = u_1^{k+1} & \text{sur} \quad \Gamma_2 \end{cases} \tag{4.9}$$

On remarque que chaque problème continu prend comme condition limite de Dirichlet la valeur précédente dans l'autre domaine. Cet algorithme est séquentiel.

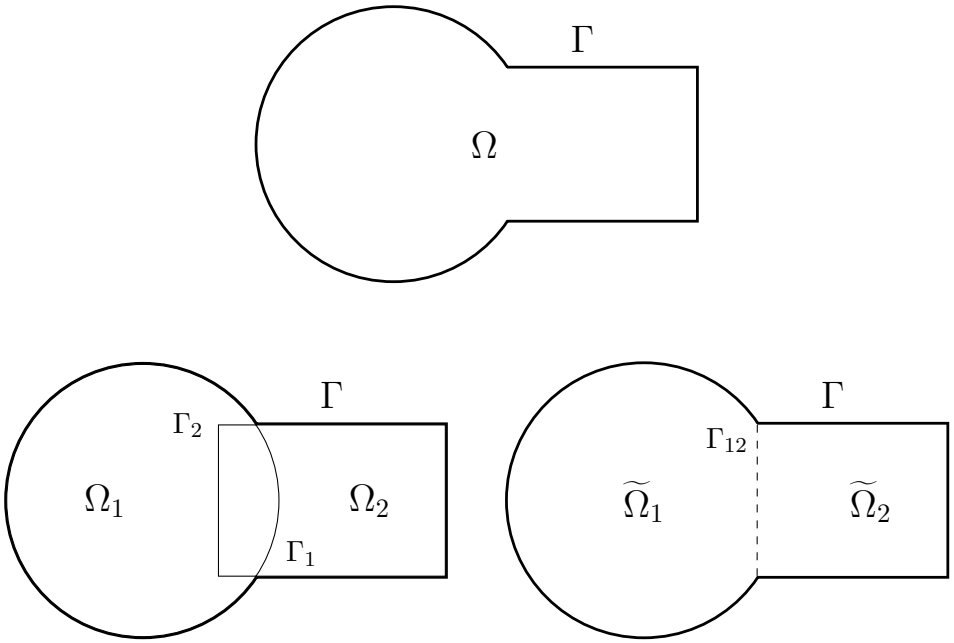


FIGURE 4.2 – Illustration du domaine global Ω et des sous-domaines locaux avec recouvrement et sans recouvrement.

Pour rendre cet algorithme parallèle, les conditions limites de Dirichlet sont prises à l'itération précédente :

$$\begin{cases} \Delta u_1^{k+1} = f & \text{sur} \quad \Omega_1 \\ u_1^{k+1} = 0 & \text{sur} \quad \Gamma \cap \partial\Omega_1 \\ u_1^{k+1} = u_2^k & \text{sur} \quad \Gamma_1 \end{cases} \quad \text{et} \quad \begin{cases} \Delta u_2^{k+1} = f & \text{sur} \quad \Omega_2 \\ u_2^{k+1} = 0 & \text{sur} \quad \Gamma \cap \partial\Omega_2 \\ u_2^{k+1} = u_1^k & \text{sur} \quad \Gamma_2 \end{cases} \tag{4.10}$$

Les résolutions locales dans (4.10) se font indépendamment sur chaque sous-domaine. Les méthodes continues (4.9) et (4.10) se font sur des fonctions indépendantes qui chacune converge sur son espace de définition vers la solution u. Pour obtenir une fonction globale (*i.e.* définie sur Ω) notée u^k, il faut définir la recombinaison des fonctions locales, par exemple une somme pondérée $\chi_1 u_1 + \chi_2 u_2$, avec $\chi_1 = 1$ sur $\Omega_1 \setminus (\Omega_1 \cap \Omega_2)$, $\chi_2 = 1$ sur $\Omega_2 \setminus (\Omega_1 \cap \Omega_2)$ et $\chi_1 + \chi_2 = 1$ sur $\Omega_1 \cap \Omega_2$.

Pour simplifier la recombinaison, soit deux domaines $\widetilde{\Omega}_i \subset \Omega_i$ dont l'union forme le domaine complet $\widetilde{\Omega}_1 \cup \widetilde{\Omega}_2 = \Omega$ et dont l'intersection est nulle $\widetilde{\Omega}_1 \cap \widetilde{\Omega}_2 = \emptyset$. L'intersection de leur adhérence est une frontière notée Γ_{12}. On considère que $\Gamma_{12} \subset \widetilde{\Omega}_1$ par exemple. Un exemple d'une telle décomposition est donnée sur la figure 4.2. On peut alors choisir les poids χ_i de recombinaison tels que $\chi_i = 1$ sur $\widetilde{\Omega}_i$. Soit les opérateurs d'extension E_i et $\widetilde{E}_i$ tels que $E_i : \Omega_i \to \Omega$ étende une fonction de Ω_i vers Ω par des zéros sur Ω/Ω_i et

$\widetilde{E}_i : \widetilde{\Omega}_i \to \Omega$ étende une fonction de $\widetilde{\Omega}_i$ par des zéros sur Ω. On remarque que les opérateurs $\widetilde{E}_i$ forment une partition de l'unité, c'est-à-dire que pour toute fonction v définie sur Ω, $\widetilde{E}_1\Big(v_{|\widetilde{\Omega}_1}\Big) + \widetilde{E}_2\Big(v_{|\widetilde{\Omega}_2}\Big) = v$. L'algorithme de Schwarz additif s'écrit alors

$$\begin{cases} \Delta u_1^{k+1} = f & \text{sur} \quad \Omega_1 \\ u_1^{k+1} = 0 & \text{sur} \quad \Gamma \cap \partial\Omega_1 \\ u_1^{k+1} = u_2^k & \text{sur} \quad \Gamma_1 \end{cases} \quad \text{et} \quad \begin{cases} \Delta u_2^{k+1} = f & \text{sur} \quad \Omega_2 \\ u_2^{k+1} = 0 & \text{sur} \quad \Gamma \cap \partial\Omega_2 \\ u_2^{k+1} = u_1^k & \text{sur} \quad \Gamma_2 \end{cases} \tag{4.11}$$
$$u^{k+1} = \widetilde{E}_1\Big(u_1^{k+1}\Big) + \widetilde{E}_2\Big(u_2^{k+1}\Big)$$

On initialise la suite $\Big(u^k\Big)$ par u^0 comme étant la fonction nulle sur Ω. Pour pouvoir écrire l'algorithme (4.11) sous forme de point fixe, afin d'identifier sa version algébrique et en tirer un préconditionneur, on réécrit (4.11) avec le résidu de l'équation globale (4.8) et on cherche des corrections locales v_i^{k+1} à la solution globale u^k

$$r^k \leftarrow f - \Delta u^k$$
$$\begin{cases} \Delta v_1^{k+1} = r^k & \text{sur} \quad \Omega_1 \\ v_1^{k+1} = 0 & \text{sur} \quad \Gamma \end{cases} \quad \text{et} \quad \begin{cases} \Delta v_2^{k+1} = r^k & \text{sur} \quad \Omega_2 \\ v_2^{k+1} = 0 & \text{sur} \quad \Gamma \end{cases} \tag{4.12}$$
$$u^{k+1} = u^k + \widetilde{E}_1\Big(v_1^{k+1}\Big) + \widetilde{E}_2\Big(v_2^{k+1}\Big)$$

La condition aux limites de Dirichlet inhomogène sur u_i^{k+1} devient alors une condition homogène sur v^{k+1}.

Méthode de Schwarz additif pour le cas discret

Des définitions supplémentaires sont nécessaires pour discrétiser le cas continu. On considère des éléments finis de Lagrange P1, ce qui permet d'identifier la discrétisation de l'espace avec celle des fonctions. L'ensemble de cardinal n des degrés de liberté V qui discrétisent l'espace global Ω est découpé en parties V_i de taille n_i correspondant à Ω_i et en $\widetilde{V}_i$ pour $\widetilde{\Omega}_i$. La discrétisation du problème global (4.8) donne le système $\mathbf{Au} = \mathbf{b}$.

L'opérateur de restriction de V dans V_i se note $\mathbf{R}_i$. C'est une matrice de taille $n_i \times n$, de colonnes composées de zéros et d'un seul élément valant 1. Sa transposée est l'opérateur de prolongation de V_i dans V, et est la discrétisation de l'opérateur d'extension E_i. De même, on note $\widetilde{\mathbf{R}}_i$ l'opérateur de restriction de V_i dans $\widetilde{V}_i$.

On note $\mathbf{A}_i = \mathbf{R}_i\mathbf{A}\mathbf{R}_i^T$. Sous des réserves de correspondances entre Ω_i et V_i que l'on détaillera ci-après, les systèmes $\mathbf{A}_i\mathbf{v}_i^k = \mathbf{R}_i\Big(\mathbf{b} - \mathbf{A}\mathbf{u}^k\Big)$ sont la discrétisations des problèmes locaux de (4.12). Leur solution s'obtient en inversant $\mathbf{A}_i$. Alors, l'algorithme (4.12) se discrétise en

$$\begin{aligned}\mathbf{u}^{k+1} &= \mathbf{u}^k + \widetilde{\mathbf{R}}_1^T \mathbf{A}_1^{-1} \mathbf{R}_1 \Big(\mathbf{b} - \mathbf{A}\mathbf{u}^k\Big) + \widetilde{\mathbf{R}}_2^T \mathbf{A}_2^{-1} \mathbf{R}_2 \Big(\mathbf{b} - \mathbf{A}\mathbf{u}^k\Big) \\ &= \mathbf{u}^k + \left(\sum_{i=1}^{2} \widetilde{\mathbf{R}}_i^T \mathbf{A}_i^{-1} \mathbf{R}_i\right) \Big(\mathbf{b} - \mathbf{A}\mathbf{u}^k\Big) \end{aligned} \tag{4.13}$$

C'est une méthode itérative pour résoudre un problème de point fixe. Si elle converge, notons $\mathbf{u}^\infty = \lim_{k\to\infty} \mathbf{u}^k$. Alors

$$\left(\sum_{i=1}^{2} \widetilde{\mathbf{R}}_i^T \mathbf{A}_i^{-1} \mathbf{R}_i\right) (\mathbf{b} - \mathbf{A}\mathbf{u}^\infty) = \mathbf{0}$$

Si l'on pose $\mathbf{M}_{RAS}^{-1} = \sum_{i=1}^{2} \widetilde{\mathbf{R}}_i^T \mathbf{A}_i^{-1} \mathbf{R}_i$, alors $\mathbf{u}^\infty$ est solution du système préconditionné suivant :

$$\mathbf{M}_{RAS}^{-1} \mathbf{A}\mathbf{u} = \mathbf{M}_{RAS}^{-1} \mathbf{b} \tag{4.14}$$

On obtient le préconditionneur RAS pour *Restricted Additive Schwarz* [44]. Le terme *Restricted* indique que c'est l'extension $\widetilde{\mathbf{R}}_i^T$ depuis $\widetilde{V}$ qui est utilisée. Si on se place dans le cas d'une fonction de pondération χ_i plus générale, et que l'on note $\mathbf{D}_i$ la discrétisation de χ_i sur V_i (*i.e.* sur Ω_i), le préconditionneur WRAS (pour *Weighted Restricted Additive Schwarz*) [56] est défini comme

$$\mathbf{M}_{WRAS}^{-1} = \sum_i \mathbf{R}_i^T \mathbf{D}_i \mathbf{A}_i^{-1} \mathbf{R}_i \tag{4.15}$$

Si les matrices $\mathbf{D}_i$ sont composées uniquement de 0 et de 1, on a $\mathbf{R}_i^T \mathbf{D}_i \mathbf{R}_i = \widetilde{R}_i^T \widetilde{R}_i$. Dans ce cas, $\mathbf{M}_{RAS}^{-1} = \mathbf{M}_{WRAS}^{-1}$. Cette égalité conduit à abandonner la terminologie WRAS et à utiliser le nom de RAS même lorsque des poids $\mathbf{D}_i$ quelconques sont utilisés. On rappelle que ces matrices de poids $\mathbf{D}_i$ sont la discrétisation des χ_i et, vu l'utilisation de fonctions de formes interpolantes, leur somme vaut 1 en chaque nœud :

$$\sum_i \mathbf{R}_i^T \mathbf{D}_i \mathbf{R}_i = \mathbf{I} \tag{4.16}$$

où $\mathbf{I}$ est la matrice identité de rang n. Cette propriété conduit à appeler les $\mathbf{D}_i$ partitions de l'unité.

Revenons aux matrices $\mathbf{A}_i = \mathbf{R}_i \mathbf{A} \mathbf{R}_i^T$ et aux problèmes locaux (4.12) qui leur sont associés. Ces problèmes locaux continus ont des conditions aux limites de Dirichlet homogènes. Comme on le rappellera dans le chapitre 6, imposer des conditions de Dirichlet dans une formulation élément fini utilisant des éléments finis de Lagrange revient à éliminer les inconnues de cette interface. En d'autres termes, on supprime les lignes et les colonnes associées dans la matrice. Cela veut dire que si $\mathbf{A}_i$ est la discrétisation

du problème continu sur Ω_i, alors V_i ne contient pas les inconnues des interfaces parallèles (*i.e.* entre les sous-domaines). Tout se passe comme si Ω_i contenait une bande d'éléments supplémentaire au niveau des interfaces parallèles. Ainsi, même pour une décomposition de domaine à recouvrement minimal (partitionnement des éléments, donc les domaines ne se touchent que par des surfaces), le problème continu est à recouvrement non nul et donc converge [44]. Une conséquence triviale de cette différence de définition entre les domaines continus et discrets est que les matrices $\mathbf{A}_i$ ne sont pas les matrices locales (à un domaine) de la discrétisation par éléments finis. Il y a nécessairement une phase d'assemblage aux interfaces, puisque par définition les $\mathbf{A}_i = \mathbf{R}_i \mathbf{A} \mathbf{R}_i^T$ sont issues de la matrice globale assemblée.

Enfin, cette différence entre domaines continus et discrets n'est possible que pour des conditions aux limites de Dirichlet. D'autres méthodes, brièvement présentées ci-dessous, utilisent des conditions de Neumann ou de Robin. Dans ce cas-là, l'espace discret coïncide géométriquement avec l'espace continu qu'il discrétise.

Finalement, on peut définir un préconditionneur appelé AS pour *Additive Schwarz*, en enlevant la partition de l'unité $\mathbf{D}_i$ au préconditionneur RAS :

$$\mathbf{M}_{AS}^{-1} = \sum_i \mathbf{R}_i^T \left(\mathbf{R}_i \mathbf{A} \mathbf{R_i}^T \right)^{-1} \mathbf{R}_i \tag{4.17}$$

Historiquement, c'est le premier préconditionnement utilisé. Il converge moins bien que le préconditionneur RAS car il correspond à une addition du résidu dans la méthode itérative associée (4.13) [47]. Le préconditionneur AS a l'avantage d'être symétrique, contrairement au RAS, et doit donc être choisi si un solveur symétrique (type CG) est utilisé. La découverte accidentelle du préconditionneur RAS formulée à l'aide de l'opérateur d'extension $\widetilde{\mathbf{R}}_i^T$ est due à Cai et Sarkis [25]. La différence entre les préconditionnements RAS et AS est traitée en détail dans [47]. On pourra également consulter [56] pour une théorie algébrique du préconditionneur RAS. Le préconditionnement RAS permet en général une meilleure convergence. Sur des applications aérodynamiques, on pourra consulter [139, 27].

4.3.2 Conditions de Schwarz optimisées

La méthode de Schwarz additif se base sur l'utilisation de conditions aux limites de Dirichlet pour les problèmes locaux résolus parallèlement. Seule la continuité de la solution est donc imposée à l'interface. La convergence est assurée uniquement si le recouvrement est non nul [44]. Pour développer une méthode à recouvrement nul, Lions [100] utilise une condition de Robin à l'interface, à savoir que $\left(\frac{\partial}{\partial \mathbf{n}} + \alpha \right) u$ doit être égal de part et d'autre de l'interface, où $\mathbf{n}$ est un vecteur normal à l'interface et α un paramètre de relaxation. Cela est imposé itérativement comme pour les méthodes de

Schwarz additif standard. Ces conditions d'interface ont été appliquées avec succès aux équations d'Helmholtz [44, 17, 103].

En revanche, elles ne sont pas optimales. Les conditions optimales sont les conditions limites parfaitement absorbantes [44, 87], présentées par exemple dans [157]. Elles utilisent des opérateurs non-locaux, appelés Dirichlet-to-Neumann (DtN). Elles ne sont donc pas facilement implémentables.

Ces opérateurs peuvent être approchés localement sur une interface plane infinie. Pour cela, on prend la transformée de Fourier de l'opérateur dans le plan de l'interface, que l'on approche par les premiers termes de sa décomposition de Taylor [57, 87]. L'opérateur frontière obtenu est une somme de dérivées dans le plan de l'interface. On peut également optimiser les coefficients de cette somme directement, afin d'améliorer le fonctionnement de ces conditions limites sur l'ensemble du spectre. On parle de conditions de Schwarz optimisées [57, 103].

Les conditions optimisées amènent à une correspondance entre espace discret et espace continu, car ces conditions n'entraînent pas l'élimination des inconnues de bord (cf. section 4.3.1). Lorsque le recouvrement entre les domaines est suffisant, on peut simplement rajouter à la matrice locale la discrétisation de l'opérateur différentiel à l'interface [150, 151]. Lorsque les domaines sont à recouvrement minimal, on est obligé d'utiliser une inconnue supplémentaire représentant, par exemple pour des conditions de Robin, la valeur de la dérivée normale de la solution approchée [44, 57].On peut également utiliser un système dont les inconnues partagées sont dupliquées, au prix de devoir appliquer les opérateurs de transmission pour chaque produit matrice-vecteur [150, section 3.9] [151].

Bien que des applications des conditions de Schwarz optimisées en mécaniques des fluides existent [45, 73], elles n'ont pas été testées dans le cadre de cette thèse.

4.3.3 Méthodes de sous-structuration

La méthode de Schwarz additif n'est pas la seule méthode de parallélisation d'un problème linéaire. D'autres méthodes, dites de sous-structuration, se servent de compléments de Schur locaux pour éliminer les inconnues intérieures des domaines, et résolvent le problème uniquement via les inconnues d'interface. La méthode Neumann-Neumann, introduite par De Roek et Le Tallec [40], demande la continuité de ces inconnues d'interface, alors que la méthode FETI due à Farhat et Roux [52] duplique ces inconnues et impose leur égalité par des multiplicateurs de Lagrange. Elles ont également une interprétation continue [156]. La méthode Neumann-Neumann résout des problèmes de Dirichlet par sous-domaine, et utilise la différence des dérivées normales des solutions locales aux interfaces entre les sous-domaines comme conditions aux limites de problèmes de Neumann locaux. Cette étape de préconditionnement permet d'imposer faiblement la continuité de la dérivée

normale. La méthode FETI procède à l'inverse : des problèmes de Neumann locaux avec le même flux aux interfaces pour tous les sous-domaines sont résolus pour fournir une valeur à l'interface. Des problèmes de Dirichlet locaux avec comme conditions aux limites la différence de ces valeurs à l'interface permet de corriger la solution précédente et d'imposer faiblement la continuité de la solution dans l'étape de préconditionnement. On pourra se référer à [156] pour plus de détails, notamment le chapitre 1 pour une bonne introduction à ces deux méthodes.

Dans le cadre de l'élasticité linéaire, ces deux méthodes trouvent une explication physique [148]. De même, ces méthodes peuvent être utilisées en préconditionnement de méthodes itératives. Des méthodes de sous-structurations ont déjà été utilisées pour des problèmes en mécanique des fluides [15, 59].

4.4 Déflation d'espace grossier

4.4.1 Les limites du préconditionnement Schwarz additif

Le préconditionnement de Schwarz additif souffre d'un problème d'extensibilité (*scalability* en anglais). Ce terme, en calcul parallèle, désigne la capacité d'une méthode à ne pas se dégrader quand le nombre de domaines augmente. Plus précisément, on distingue l'extensibilité forte et l'extensibilité faible. La première notion se définit par la variation du temps de résolution d'un problème de taille fixe en fonction du nombre de domaines. L'extensibilité forte est parfaite quand le temps de résolution est inversement proportionnel au nombre de domaines. L'extensibilité faible se définit par la variation du temps de calcul lorsque la taille du problème augmente linéairement avec le nombre de domaines. L'extensibilité faible est atteinte lorsque le temps de résolution est constant. Une introduction fournie à la scalabilité est donnée par Keyes dans [92].

Ces deux notions répondent à des questions pratiques légèrement différentes. L'extensibilité forte intéresse particulièrement les utilisateurs de la méthode de décomposition de domaine. Elle permet aux industriels de répondre à la question suivante : quelle machine de calcul dois-je acheter pour que mes simulations actuelles durent moins de X jours ? Ou encore, si j'achète une machine deux fois plus grosse, mes calculs iront-ils deux fois plus vite ? L'extensibilité faible permet de vérifier qu'une méthode de décomposition de domaine sera toujours intéressante dans quelques années, quand les problèmes seront de plus grande taille, et les machines de calcul aussi.

La méthode de Schwarz additif souffre d'un problème d'extensibilité faible et forte [44, chapitre 4]. Cette méthode découple partiellement les domaines en effectuant une résolution locale dans chaque domaine et en ne communiquant qu'à l'interface. Intuitivement, l'information d'un domaine mettra autant d'itérations à atteindre un autre qu'il y a de domaines les séparant. Des

résultats théoriques plus précis existent sur le préconditionnement de Schwarz additif pour des matrices symétriques définies positives [135, Th. 14.6] [156].

Pour améliorer l'extensibilité des méthodes de décomposition de domaine, les méthodes dites à deux niveaux introduisent un espace grossier partagé par tous les domaines, qui permet de communiquer instantanément l'information dans tout le domaine de calcul. Cet espace grossier est constitué en général de vecteurs locaux à un ou plusieurs domaines, qui approchent les plus petites valeurs propres des opérateurs locaux [44]. Les méthodes de construction de l'espace grossier sont nombreuses : Nicolaides [120] utilise le noyau de l'opérateur Laplacien, qui sont les fonctions constantes par domaine, l'espace grossier GDSW [77] utilise les solutions de problèmes de Dirichlet locaux avec une impulsion sur les faces, les arêtes et les coins de la décomposition, ou encore l'espace GenEO [148, 149] basé sur les vecteurs propres basse fréquence de problèmes aux valeurs propres généralisés dans les sous-domaines.

L'espace grossier des préconditionnements à deux niveaux s'interprète comme la grille grossière dans les méthodes multigrille. Tang *et al.* listent les parallèles entre les deux méthodes dans [155].

Enfin, on peut tracer un parallèle intéressant entre les méthodes de Schwarz à deux niveaux et les solveurs itératifs par blocs. L'idée est de laisser au solveur itératif le choix de la meilleure combinaison linéaire entre les différents préconditionneurs locaux. L'algorithme GMRES avec plusieurs préconditionnements est introduit dans [68], et appliquée au cas spécifique du préconditionnement par la méthode de Schwarz additif dans [67], où les différents préconditionneurs sont les préconditionneurs locaux. Une version du block-GMRES avec déflation des seconds membres et des petites valeurs propres est présentée dans [3].

Dans le cadre de cette thèse, le préconditionnement à deux niveaux a été testé à la lumière des résultats encourageants d'Alcin *et al* [5] sur les équations de Navier-Stokes temporelles.

4.4.2 Déflation de vecteurs

En algèbre linéaire, la déflation de vecteurs consiste [44, 155] à enlever l'influence de vecteurs choisis sur la convergence du solveur de Krylov, par projection sur l'orthogonal de l'espace généré par ces vecteurs. On suivra la présentation de Frank et Vuik [54], en prenant le même espace à gauche et à droite.

Soit $\mathbf{Z} \in \mathbb{R}^{N\times m}$ la matrice dont les colonnes sont les m vecteurs de déflation. Le choix de ces vecteurs sera abordé plus tard. Le nombre de ces vecteurs est choisi très faible devant la dimension du système, *i.e.* $m \ll N$. On note $\mathbf{E} = \mathbf{Z}^T\mathbf{A}\mathbf{Z}$ la matrice carrée de taille $m \times m$, qui définit le système dit grossier. Cette matrice permet de définir deux projecteurs $\mathbf{P}$ et $\mathbf{Q}$:

$$\begin{aligned} \mathbf{P} &= \mathbf{I} - \mathbf{A}\mathbf{Z}\mathbf{E}^{-1}\mathbf{Z}^T \\ \mathbf{Q} &= \mathbf{I} - \mathbf{Z}\mathbf{E}^{-1}\mathbf{Z}^T\mathbf{A}\,, \end{aligned} \tag{4.18}$$

où $\mathbf{I}$ désigne la matrice identité de rang n. Les relations $\mathbf{P}^2 = \mathbf{P}$ et $\mathbf{Q}^2 = \mathbf{Q}$ montrent que $\mathbf{P}$ et $\mathbf{Q}$ sont des projecteurs. Enfin, de manière triviale, $\mathbf{PA} = \mathbf{AQ}$. La caractérisation des projecteurs se fait à l'aide de l'identité suivante :

$$\begin{aligned} \mathbf{PAZ} = \mathbf{0} = \mathbf{Z}^T\mathbf{P} \\ \mathbf{QZ} = \mathbf{0} = \mathbf{Z}^T\mathbf{AQ}\,. \end{aligned}$$

On en déduit que $\mathbf{P}$ est le projecteur sur $(\operatorname{Vect}\mathbf{Z})^\perp$ parallèlement à $\operatorname{Vect}(\mathbf{AZ})$, et que $\mathbf{Q}$ est le projecteur sur $\left(\operatorname{Vect}\mathbf{A}^T\mathbf{Z}\right)^\perp$ parallèlement à $\operatorname{Vect}\mathbf{Z}$. Le système $\mathbf{Ax} = \mathbf{b}$ peut être écrit de façon équivalente comme suit :

$$\mathbf{PAx} + (\mathbf{I} - \mathbf{P})\,\mathbf{Ax} = \mathbf{Pb} + (\mathbf{I} - \mathbf{P})\,\mathbf{b}\,. \tag{4.19}$$

Les projecteurs $\mathbf{P}$ et $\mathbf{I}-\mathbf{P}$ sont associés, car $\mathbf{P}(\mathbf{I} - \mathbf{P}) = \mathbf{0}$. En multipliant d'une part l'équation (4.19) par $\mathbf{P}$ et d'autre part par $\mathbf{I}-\mathbf{P}$, on décompose le système linéaire original en sa composante selon $(\operatorname{Vect}\mathbf{Z})^\perp$ et sa composante selon $\operatorname{Vect}(\mathbf{AZ})$:

$$\begin{array}{ccc} \mathbf{Ax} & = & \mathbf{b} \\ & \Longleftrightarrow & \\ \mathbf{PAx} = \mathbf{Pb} & \text{et} & (\mathbf{I} - \mathbf{P})\,\mathbf{Ax} = (\mathbf{I} - \mathbf{P})\,\mathbf{b}\,. \end{array} \tag{4.20}$$

La deuxième équation se simplifie, car $\mathbf{I} - \mathbf{P} = \mathbf{AZE}^{-1}\mathbf{Z}^T$:

$$\begin{aligned} (\mathbf{I} - \mathbf{P})\,\mathbf{Ax} = (\mathbf{I} - \mathbf{P})\,\mathbf{b} &\iff \mathbf{AZE}^{-1}\mathbf{Z}^T\mathbf{Ax} = \mathbf{AZE}^{-1}\mathbf{Z}^T\mathbf{b} \\ &\iff \mathbf{ZE}^{-1}\mathbf{Z}^T\mathbf{Ax} = \mathbf{ZE}^{-1}\mathbf{Z}^T\mathbf{b}\,. \end{aligned} \tag{4.21}$$

La recomposition de la solution s'effectue en remarquant que $\mathbf{ZE}^{-1}\mathbf{Z}^T\mathbf{Ax} = (\mathbf{I} - \mathbf{Q})\,\mathbf{x}$. Ainsi, $\mathbf{x}$ se décompose comme suit :

$$\mathbf{x} = (\mathbf{I} - \mathbf{Q})\,\mathbf{x} + \mathbf{Qx}\,. \tag{4.22}$$

L'opérateur $\mathbf{I}-\mathbf{Q}$ est un projecteur sur $\operatorname{Vect}\mathbf{Z}$ parallèlement à $(\operatorname{Vect}\mathbf{AZ})^\perp$. Le premier terme de la somme, qui correspond à la composante selon $\operatorname{Vect}\mathbf{Z}$, se calcule directement comme on l'a vu :

$$\begin{aligned} (\mathbf{I} - \mathbf{Q})\,\mathbf{x} &= \mathbf{ZE}^{-1}\mathbf{Z}^T\mathbf{Ax} \\ &= \mathbf{ZE}^{-1}\mathbf{Z}^T\mathbf{b}\,. \end{aligned}$$

L'autre terme, $\mathbf{Qx}$, est issu de la résolution de la première équation de (4.20). En effet, en réinjectant l'équation (4.4.2) dans l'équation (4.22), on obtient, après multiplication à gauche par $\mathbf{A}$:

$$\mathbf{Ax} = \mathbf{b} = \mathbf{AZE}^{-1}\mathbf{Z}^T\mathbf{b} + \mathbf{AQx}\,. \tag{4.23}$$

On isole $\mathbf{b}$ et en utilisant la relation $\mathbf{PA} = \mathbf{AQ}$:

$$\begin{array}{rcl} \mathbf{AQx} & = & \left(\mathbf{I} - \mathbf{AZE}^{-1}\mathbf{Z}^T\right)\mathbf{b} \\ & \Longleftrightarrow & \\ \mathbf{PAx} & = & \mathbf{Pb}\,. \end{array} \tag{4.24}$$

Ainsi, on résout le système préconditionné $\mathbf{PA}\widehat{\mathbf{x}} = \mathbf{Pb}$ par l'algorithme GMRES, puis on reconstruit la solution suivant la formule $\mathbf{x} = \mathbf{ZE}^{-1}\mathbf{Z}^T\mathbf{b} + \mathbf{Q}\widehat{\mathbf{x}}$. Si le système est de plus préconditionné à gauche par une matrice $\mathbf{M}^{-1}$, cela revient à résoudre le système $\mathbf{M}^{-1}\mathbf{PA}\widehat{\mathbf{x}} = \mathbf{M}^{-1}\mathbf{Pb}$.

Cette décomposition de la solution peut être évitée en appliquant le préconditionneur dit *balancing preconditionner* dû à Mandel [105] défini par $\mathbf{P}_B = \mathbf{M}^{-1}\mathbf{P} + \mathbf{ZE}^{-1}\mathbf{Z}^T$. Le problème grossier est résolu à chaque application du préconditionnement. Cela apporte une stabilité plus grande, pour un coût de calcul légèrement plus élevé (deux applications de $\mathbf{E}^{-1}$ par itération au lieu d'une) [94].

L'utilisation du préconditionnement $\mathbf{P}$ demande l'application de l'inverse de la matrice $\mathbf{E}$. C'est par construction une matrice de petite taille ($m \times m$). Ce calcul d'inverse peut se faire par une méthode directe, qui est éventuellement parallélisée. On pourra consulter [5] pour une étude de la parallélisation optimale de la factorisation exacte par MUMPS [7, 9]. Enfin, on peut résoudre de manière approchée ce système [118]. Dans ce cas-là, certaines formes du préconditionnement à deux niveaux sont plus appropriées [49].

Pour un système défini positif, si $\mathbf{Z}$ est composé de vecteurs propres de $\mathbf{A}$, les valeurs propres correspondantes sont remplacées par des 0 dans le spectre de la matrice après déflation $\mathbf{PA}$ [54]. Ce même résultat est étendu aux matrices non symétriques dans [49]. Cela justifie la dénomination de déflation à cette méthode.

Ainsi, le système linéaire $\mathbf{PA}\widehat{\mathbf{x}} = \mathbf{Pb}$ est singulier, mais consistant, puisque la même projection est appliquée à la matrice et au second-membre. La convergence de l'algorithme GMRES est garantie pour les systèmes singuliers lorsqu'ils sont consistants, *i.e.* lorsque le second membre appartient à l'image de l'opérateur. Dans [58], la condition pour que l'algorithme GMRES converge est que $\mathbf{Z}$ soit invariant par $\mathbf{A}$. Lorsque les vecteurs de déflations sont vecteurs propres de $\mathbf{A}$, la convergence de GMRES est assurée [49].

Enfin, lorsque l'espace grossier est composé de vecteurs propres de l'opérateur, l'algorithme GMRES appliqué au système $\mathbf{PA}\widehat{\mathbf{x}} = \mathbf{Pb}$ est identique à l'algorithme GMRES avec déflation des mêmes vecteurs propres introduit dans 3.2.4. Il est prouvé dans [58] l'équivalence entre la résolution par

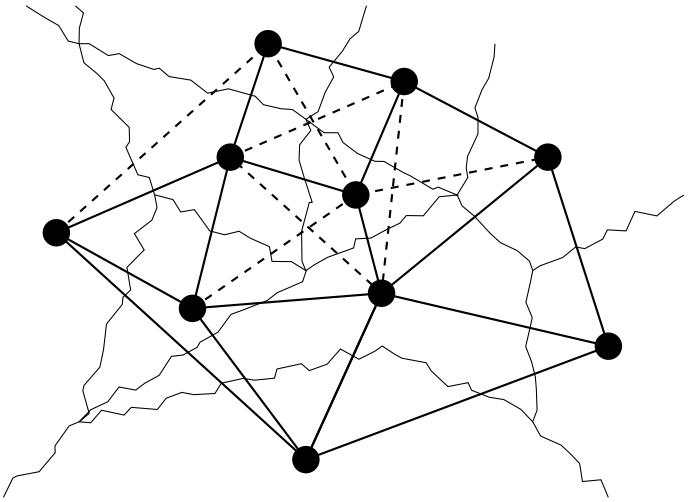


FIGURE 4.3 – Grille grossière induite par l'espace grossier de Nicolaides. Les traits pleins indiquent une connexion entre deux domaines par une face, les traits en pointillés par un sommet (ou une arête en 3D).

GMRES de $\mathbf{PA}\hat{\mathbf{x}} = \mathbf{Pb}$ et celle du système standard avec un GMRES sur un espace de Krylov augmenté des vecteurs propres utilisés. On pourra également consulter [43] pour une comparaison des deux approches.

4.4.3 Espace de Nicolaides

Le premier espace grossier de déflation a été introduit par Nicolaides [120]. Il a été utilisé avec succès dans [5] pour des problèmes de mécanique des fluides, ce qui a motivé l'étude du préconditionnement à deux niveaux au cours de cette thèse. L'espace grossier de Nicolaides est composé des vecteurs constants valant l'unité sur chaque domaine. Cet espace a été conçu pour l'équation de Poisson, et correspond au noyau de l'opérateur discrétisé sur un sous-domaine avec des conditions de Neumann au bord. Pour le système d'équations de Navier-Stokes, qui possède cinq inconnues en 3D, cinq vecteurs constants par domaines ont été utilisés, chacun correspondant à une des cinq inconnues. L'opérateur grossier $\mathbf{E}$ est donc une matrice de taille $5N \times 5N$, où N est le nombre de domaines. L'autre interprétation de cet opérateur grossier est qu'il correspond à l'agglomération de $\mathbf{A}$ sur un maillage grossier qui relierait les domaines de calcul partageant au moins un nœud. Ce maillage virtuel est illustré sur la figure 4.3. Les vecteurs $\mathbf{Z}$ doivent générer par leur somme le vecteur unité $\mathbf{1}$ sur tout le domaine [44, 138]. On se place ici dans le cadre de domaines avec recouvrement minimal.

La matrice $\mathbf{E} = \mathbf{Z}^T\mathbf{AZ}$ du problème grossier peut se construire de plusieurs manières. La méthode naïve serait d'effectuer le produit de tous les vecteurs composant les colonnes de $\mathbf{Z}$ avec la matrice $\mathbf{A}$, puis de calculer

le produit scalaire avec $\mathbf{Z}^T$. Cette méthode n'est pas extensible avec le découpage car $\mathbf{Z}$ a un nombre de colonnes égal à cinq fois le nombre de domaines.

Une méthode plus rapide pour construire $\mathbf{E}$ est de calculer les contributions locales à chaque domaine de cette matrice puis de l'assembler. La diagonale de $\mathbf{E}$ est un bloc 5×5 (en 3D) qui est la somme par blocs des lignes et colonnes de la matrice $\mathbf{A}_i$ servant à construire le préconditionneur de Schwarz additif (*cf.* section 4.3.1). Les termes extra-diagonaux nécessitent des opérations de communication entre les processeurs, mais qui peuvent être réduites à une seule. Chaque processeur connaît à l'interface la valeur des vecteurs de $\mathbf{Z}$ appartenant au domaine voisin. En effet, les supports des vecteurs de $\mathbf{Z}$ ne se recouvrent pas : il suffit de décider à l'avance à quel processeur appartiennent les nœuds de l'interface.

La contribution d'un domaine Ω_i aux termes extra-diagonaux de $\mathbf{E}$ s'énonce comme suit : les éléments (i, j) de la ligne i de $\mathbf{E}$ sont la somme des termes de $\mathbf{A}_i$ qui correspondent à des arêtes reliant un nœud appartenant au processeur voisin j vers les nœuds intérieurs de Ω_i, tandis que les éléments (j, i) de la la colonne i de $\mathbf{E}$ sont la somme des termes de $\mathbf{A}_i$ qui correspondent à des arêtes reliant un nœud intérieur vers un nœud appartenant au domaine voisin j. L'assemblage, qui peut s'effectuer une fois par une opération de réduction globale (en MPI, une somme globale à l'aide de la routine MPI_AllReduce), de ces contributions locales aux termes diagonaux et extra-diagonaux de $\mathbf{E}$ donne la matrice complète du problème grossier. Comme cette matrice est de taille $5N \times 5N$, l'opération d'assemblage nécessite une taille mémoire et un temps croissant avec le nombre de domaines. Ceci n'est pas extensible, mais demande beaucoup moins d'opérations que la méthode naïve précédente.

La résolution d'un problème grossier est demandée à chaque application du projecteur $\mathbf{P}$ lors de la multiplication par $\mathbf{E}^{-1}$. Comme ce problème est de petite taille et doit être résolu de nombreuses fois, il a été décidé de le résoudre exactement à l'aide d'un solveur direct, MUMPS [7, 9]. Lorsque la taille du problème augmente avec le découpage, cette résolution peut être parallélisée, sur typiquement 1% des processeurs [5]. Cette résolution peut aussi être approchée, par exemple avec une factorisation approchée. Dans ce cas-là, le projecteur perd de son efficacité, ce qui peut nuire à la qualité de la solution. On pourra consulter [94] pour l'étude de l'impact de la factorisation approchée sur la solution.

Dans un premier temps, la déflation d'espace de Nicolaides a été appliquée au cas test I, qui est le profil RAE2822. Les courbes de convergence pour le système projeté par $\mathbf{P}$ sont présentées sur la figure 4.4. Sans déflation, le nombre d'itérations nécessaires pour diminuer le résidu de 6 ordres de grandeur augmente légèrement avec le découpage, de 4% au maximum. Avec déflation, comme on peut le voir sur le graphe du haut, le nombre d'itérations est sensiblement plus faible, et diminue avec l'augmentation du nombre de domaines. Une grande partie de cette diminution du nombre d'itérations

s'explique par le résidu initial plus fort, comme indiqué par le graphe du bas de la figure 4.4. Le résidu initial avec déflation ne varie que peu avec le nombre de domaines. Ainsi, cette augmentation de résidu initial est la conséquence de la projection du système sur l'ensemble des fonctions constantes. Le véritable apport de la déflation se voit sur le graphe du bas, où l'on constate que la déflation diminue la taille du plateau de stagnation de la convergence autour de la 2000e itération, et améliore également le taux asymptotique de convergence pour le cas à 64 domaines.

Le cas test II, plus industriel, est celui de la maquette DTP, présentée plus tôt à la section 3.3.4. Vu le grand nombre d'éléments du maillage, celui-ci est découpé en 512 sous-domaines. La déflation de Nicolaides a donné sur ce cas des résultats beaucoup moins positifs, comme le montrent les graphes de la figure 4.5. Le nombre d'itérations diminue grâce à la déflation par un espace de Nicolaides. L'interprétation est similaire au cas 2D précédent pour deux domaines : le résidu initial est un peu augmenté, sans modifier le comportement de la convergence. De surcroît, la résolution du système grossier lors de la projection augmente le coût de chaque itération GMRES.

L'espace grossier de Nicolaides n'a pu apporter d'accélération de la convergence des systèmes linéarisés. Ces résultats nettement moins bons que ceux présentés dans [5] sont peut-être dûs aux équations différentes que l'on résout : Alcin *et al.* s'intéressent à des problèmes de mécanique des fluides instationnaires. Le terme temporel ajoute une contribution sur la diagonale de la matrice, ce qui peut changer les propriétés de l'opérateur.

4.4.4 Extensions

Des espaces grossiers plus complexes existent. Pour un opérateur elliptique (donc symétrique défini positif), une stratégie robuste pour créer un espace grossier est d'utiliser les vecteurs propres de petites valeurs propres des opérateurs Dirichlet-to-Neumann locaux, ces vecteurs propres sur le bord étant étendus harmoniquement à l'intérieur du domaine [148, 46]. En effet, ceux-ci correspondent aux composantes basse fréquence de l'erreur que la méthode de Schwarz additif a du mal à éliminer. Sa généralisation aux opérateurs symétriques définis positifs s'appelle GenEO (pour *Generalized Eigenproblems in the Overlap*) [148, 149]. Ses vecteurs sont solutions d'un problème aux valeurs propres généralisé défini sur l'interface. L'espace GenEO a été étendu au contrôle des plus petites et des plus grandes valeurs propres du systèmes préconditionné par la résolution de deux problèmes aux valeurs propres généralisés [73].

D'autres stratégies pour construire des espaces grossiers optimaux existent. L'espace GDSW (pour *Generalized Dryja-Smith-Widlund*) permet de borner le conditionnement de l'opérateur symétrique défini positif préconditionné [77]. Les vecteurs grossiers sont associés à chacune des faces, arêtes ou sommets de la frontière entre les domaines de parallélisation. La valeur du

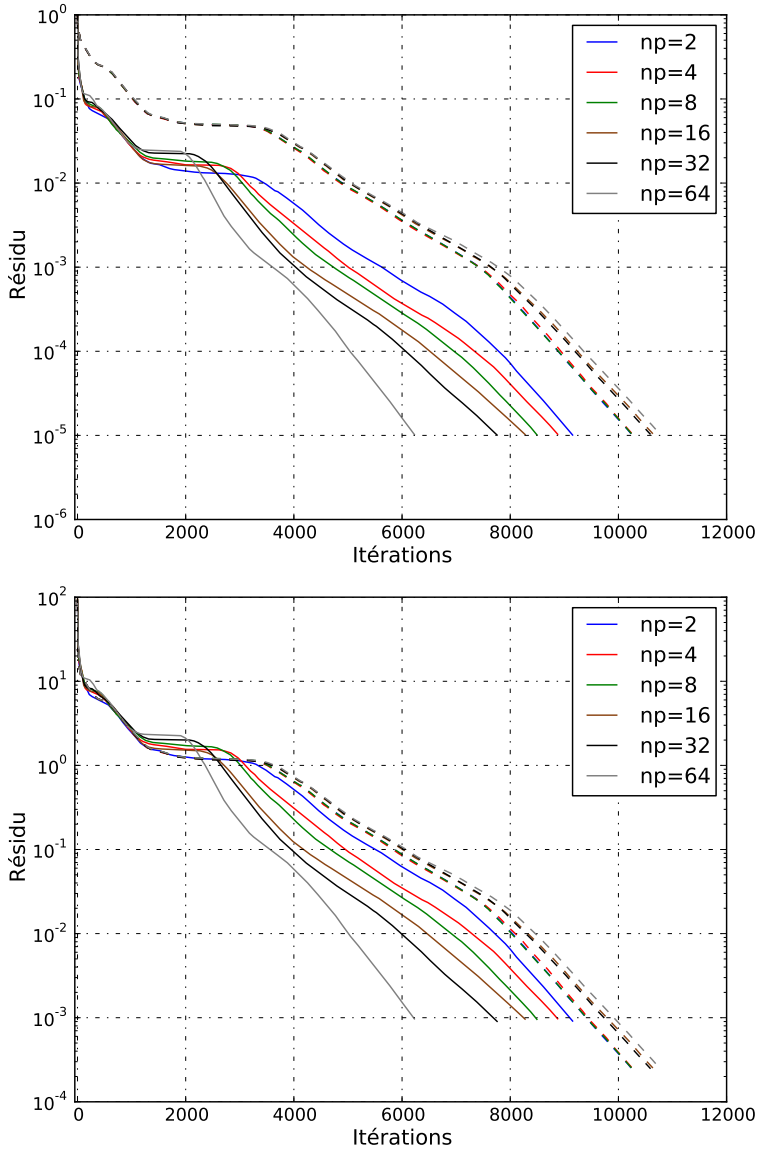


FIGURE 4.4 – Résidu GMRES normalisé sur la figure du haut et non normalisé sur la figure du bas pour le cas RAE2822 (cas test I) avec ou sans déflation par un espace grossier de Nicolaides, en fonction du nombre de sous-domaines. Les traits pleins désignent la résolution du système avec déflation, tandis que les pointillés correspondent au système original.

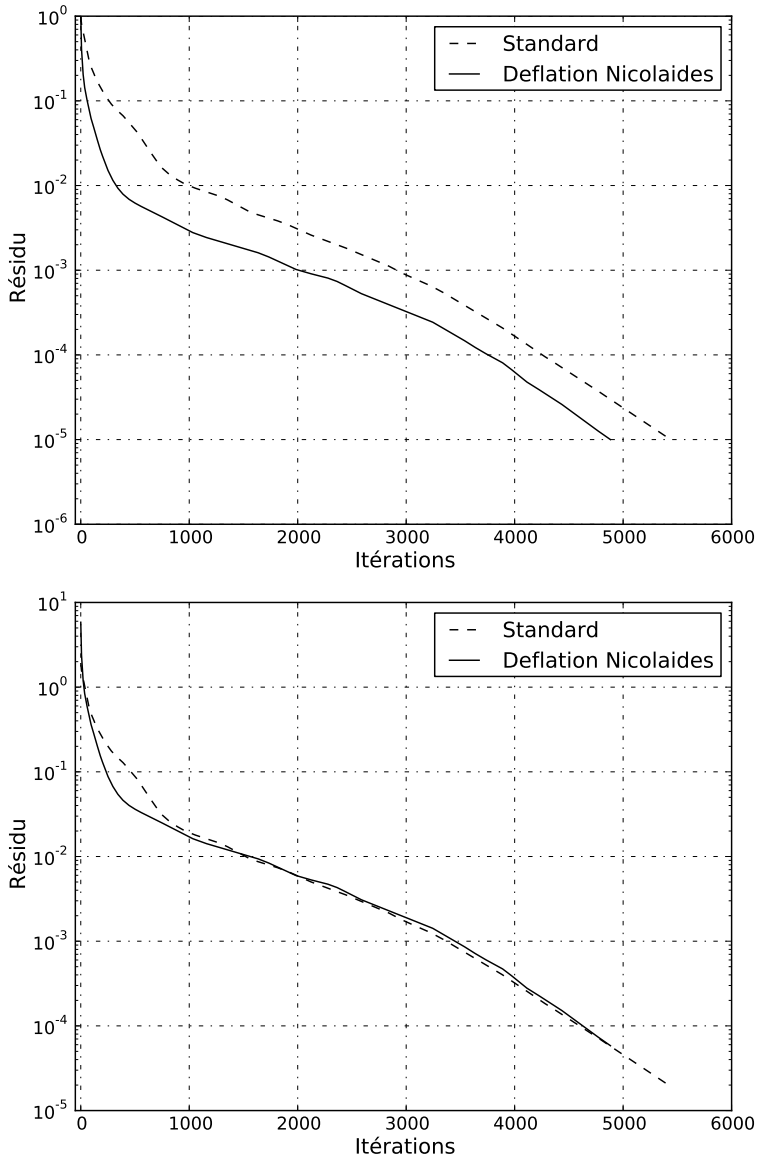


FIGURE 4.5 – Résidu GMRES normalisé sur le graphe du haut et non normalisé sur le graphe du bas pour le cas test II DTP avec ou sans déflation par un espace grossier de Nicolaides. Maillage découpé en 512 sous-domaines.

TABLEAU 4.1 – Nombre de supports pour les vecteurs de l'espace grossier GDSW en incluant ou non les modes de coins en fonction du découpage. Cas test I (RAE2822)

Domaines	4	8	16	32	64
GDSW	9	19	45	124	266
GDSW sans coins	6	13	30	77	163

vecteur grossier dans l'espace est donnée par une extension harmonique de la valeur 1 imposée sur cette partie de la frontière, et des conditions de Dirichlet homogènes ailleurs. La construction de l'espace grossier s'effectue en trois étapes. Premièrement, la frontière entre les sous-domaines doit être partitionnée en faces, arêtes et sommets. Ensuite, l'extension harmonique des vecteurs à partir de ces supports se fait localement sur chaque domaine. Enfin, la matrice du problème grossier est calculée et assemblée.

Le nombre de vecteurs augmente donc avec la complexité du découpage du problème, même si contrairement à l'espace grossier de Nicolaides, les vecteurs sont partagés entre plusieurs domaines. Comme pour les vecteurs de Nicolaides, il a été décidé de créer un vecteur par inconnue (4 en 2D, 5 en 3D) par support. Les résultats de la déflation GDSW pour le cas 2D RAE2822 (cas test I) sont présentés sur la figure 4.6. On constate que pour un faible nombre de domaines, la déflation avec un espace GDSW facilite la résolution en supprimant le plateau de convergence initial. Par contre, l'augmentation du nombre de domaines n'a pas d'effet positif sur le nombre d'itérations nécessaire à la convergence, quoiqu'elle améliore le taux de convergence asymptotique.

Le tableau 4.1 donne le nombre de supports de la base de vecteurs GDSW pour le même cas 2D en fonction du nombre de domaines. La ligne GDSW donne le nombre de faces et de coins de l'interface entre les sous-domaines, tandis que la ligne suivante donne le nombre de faces. On constate que le nombre de supports augmente de manière non-linéaire avec le nombre de domaines. L'espace GDSW utilise quatre fois plus de vecteurs que l'espace de Nicolaides pour le découpage en 64 domaines (266 supports contre 64). Pour contenir l'augmentation de la dimension de l'espace grossier, l'idée d'éliminer les modes de coins introduite dans [94] a été reprise. Elle permet de faire en sorte que le nombre de vecteurs soit une fonction quasi-linéaire du nombre de domaines, comme la dernière ligne du tableau 4.1 le montre. De surcroît, la courbe en pointillé de la figure 4.6 permet de constater que la perte de ces modes n'entraîne pas de différence majeure sur la convergence.

L'extension à des cas 3D n'a pu être effectuée avec succès pour plusieurs raisons. Tout d'abord, la résolution de l'extension harmonique des vecteurs

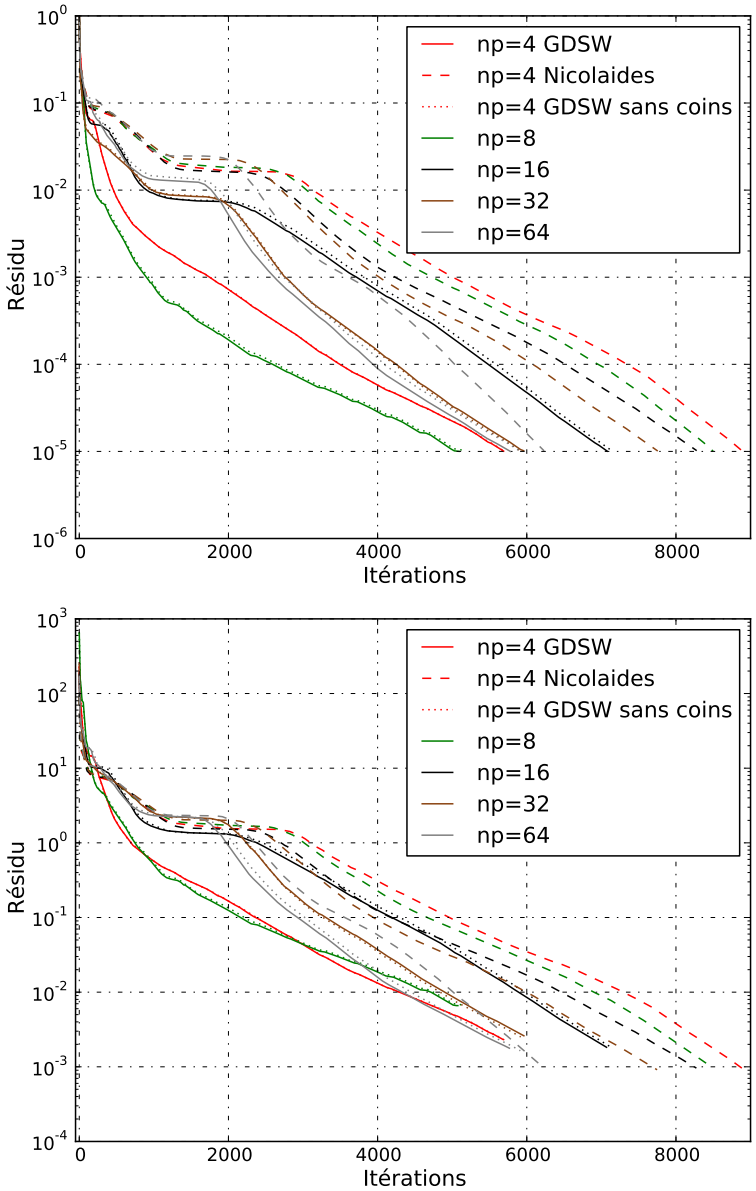


FIGURE 4.6 – Résidu GMRES normalisé sur la figure du haut et non normalisé sur la figure du bas pour le cas test I (RAE2822) avec déflation par un espace grossier GDSW ou de Nicolaides, en fonction du nombre de sous-domaines. Les traits pleins désignent la déflation avec GDSW, tandis que les tirets correspondent à la déflation de Nicolaides. Les pointillés indiquent la déflation GDSW sans modes de coins.

depuis leur support était effectuée dans notre implémentation pour une raison de simplicité par l'algorithme GMRES. Or, il s'est avéré que ces résolutions nécessitaient beaucoup d'itérations chacune, au point que chaque processeur réalisait autant d'itérations (locales) pour l'extension des vecteurs que pour la résolution du système global. Ces résolutions itératives locales auraient pu être remplacées par un solveur direct. La deuxième raison est le temps de construction de la matrice grossière $\mathbf{E} = \mathbf{Z}^T\mathbf{A}\mathbf{Z}$. Chaque vecteur de l'espace grossier a un support sur plusieurs domaines (deux s'il est associé à une face, trois ou plus pour les autres). La construction astucieuse de $\mathbf{E}$ présentée dans la section précédente pour l'espace de Nicolaides ne pouvait être utilisée. Par défaut, la matrice du problème grossier était construite par la multiplication de chaque vecteur de $\mathbf{Z}$ par $\mathbf{A}$, puis en prenant le produit scalaire du résultat par les colonnes de $\mathbf{Z}$. Cet algorithme séquentiel n'est pas extensible : il nécessite m^2 produits scalaires de vecteurs (où m est la dimension de l'espace grossier généré par $\mathbf{Z}$) et m est une fonction non-linéaire du nombre de domaines. Une méthode plus astucieuse de génération de $\mathbf{E}$ par le calcul des contributions locales par domaine a été testée, mais n'a pas donné les mêmes résultats que la méthode naïve de référence. Combinée à l'utilisation d'un solveur direct pour les extensions harmoniques, une autre méthode de génération de la matrice grossière utilise les compléments de Schur locaux [94]. Cela présuppose d'utiliser un solveur direct pour les problèmes locaux d'extension harmonique.

Sur le cas 2D présenté précédemment (*cf.* figure 4.6), la déflation GDSW n'a pas apporté de gain par rapport à la déflation de Nicolaides, qui elle-même n'était pas intéressante par rapport au système standard. De plus, l'espace grossier de Nicolaides n'apportait que peu de gain de convergence sur le cas test 3D. Cela permet de penser que la déflation par l'espace GDSW en 3D ne présente pas d'intérêt pour nos problèmes et que l'absence d'implémentation fonctionnelle en 3D n'est pas à déplorer.

4.5 Préconditionnement ILU(k)

4.5.1 Remplissage de l'ILU

Le préconditionnement ILU(0) introduit dans la section 4.2.3 n'est pas efficace pour les systèmes linéaires issus de la discrétisation des équations de Navier-Stokes linéarisées. La décomposition LU de $\mathbf{A}$ sur son masque $\mathcal{S}$ conduit à éliminer des termes. Utiliser un masque $\mathcal{S}_{LU}$ pour la décomposition ILU plus large que celui de la matrice $\mathbf{A}$ peut améliorer la qualité du préconditionnement.

Il existe plusieurs méthodes pour augmenter raisonnablement le masque utilisé pour la décomposition LU. Saad a proposé la méthode ILUT(p,τ) [135]. Lors de la factorisation, cette méthode élimine les termes dont la norme est inférieure à τ et ne garde que les p plus grands termes par colonne.

L'idée est que les petits termes ne participent que peu à la précision de la décomposition. Le paramètre p permet de borner l'empreinte mémoire de la décomposition. Le masque de la décomposition est augmenté lors de la factorisation. Pour améliorer la précision numérique et la fiabilité, la méthode ILUTP(p,τ) ajoute un pivot pour permuter les lignes et les colonnes.

Une autre stratégie est de garder certains termes de la factorisation uniquement en fonction du graphe associé à la matrice. Pour l'ILU(1), seuls les remplissages provenant des termes initiaux de la matrice sont autorisés [31]. On définit un niveau de remplissage l_{ij}, initialisé à 0 si $(i,j) \in \mathcal{S}$ et à $+\infty$ sinon, et qui pour toute entrée nouvelle qui ne soit pas dans $\mathcal{S}$ est mise à jour par la formule suivante :

$$l_{ij} = \min_{1 \leq k < \min(i,j)} \{l_{ik} + l_{kj} + 1\} \tag{4.25}$$

Le niveau ainsi défini a une interprétation (dont la preuve est donnée dans [84]) en théorie des graphes : $l_{ij} = k$ est équivalent à dire que le plus petit chemin de remplissage entre i et j est de longueur $k+1$. Un chemin de remplissage est un chemin reliant i à j sur le graphe de la matrice dont les nœuds sont de numéro inférieur à i et à j. On rappelle que les nœuds de numéros inférieurs à $\min(i,j)$ correspondent à des variables éliminées avant i et j. Pour plus de détails sur l'interprétation en théorie des graphes de l'élimination gaussienne, on pourra se référer à [10, 133].

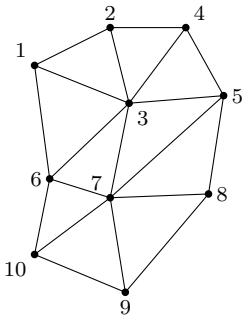
Le niveau l_{ij} permet de définir le masque $\mathcal{S}^k$ sur lequel est effectué la factorisation ILU(k) :

$$\mathcal{S}^k = \{(i,j)\,, l_{ij} \leq k\} \tag{4.26}$$

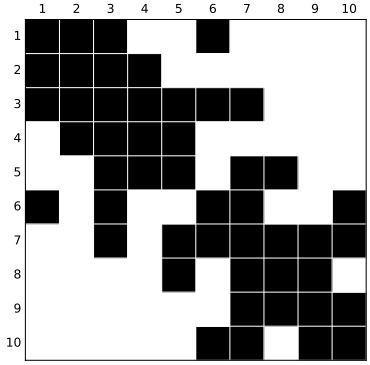
Par définition, $\mathcal{S}^0 = \mathcal{S}$. Le masque $\mathcal{S}^1$ de la décomposition ILU(1) a une interprétation simple : il correspond, s'il n'y a pas d'annulations, au masque de la matrice $\mathbf{LU}$ produit des facteurs $\mathbf{L}$ et $\mathbf{U}$ de la décomposition ILU(0) de $\mathbf{A}$ [135, sec. 10.3.3]. Ce fonctionnement est illustré sur la figure

L'identification des éléments de niveaux $l_{ij} \leq k$ qui constituent $\mathcal{S}^k$ peut se faire lors d'une analyse dite statique réalisée avant la factorisation de la matrice. Cela permet de préparer l'allocation mémoire des éléments du masque $\mathcal{S}^k$ du préconditionnement, puis d'appliquer l'algorithme 8 en utilisant $\mathcal{S}^k$ comme masque. Notons que d'autres règles existent pour définir un niveau l_{ij} [84].

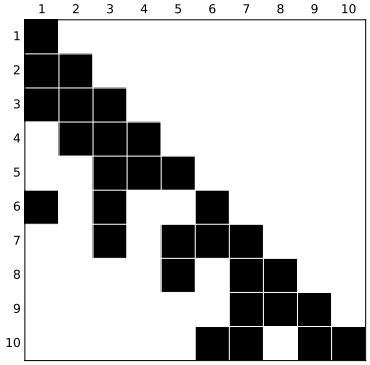
Pour tester rapidement différents préconditionnements (qui approchent la résolution des problèmes locaux dans la méthode de Schwarz additif), la bibliothèque de calcul scientifique PETSc [12, 13, 14] a été utilisée. Elle a pour but de fournir une solution « clef en main » de résolution de problèmes de calcul scientifique, en proposant une large palette de solveurs, de préconditionneurs, *etc.* Elle permet de résoudre des problèmes en parallèle. Cette bibliothèque a été choisie notamment pour sa capacité à évaluer simplement



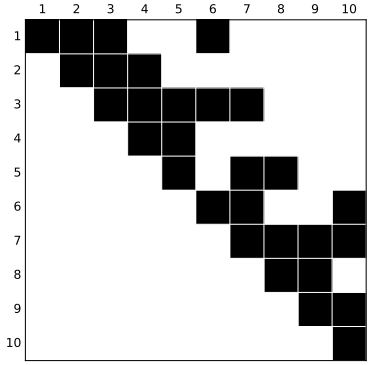
Maillage exemple



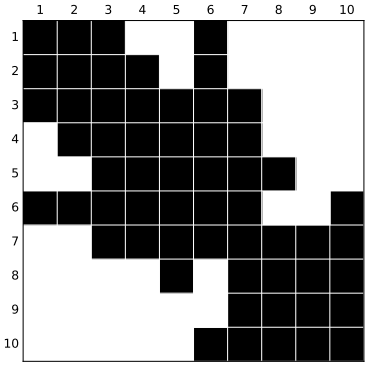
Masque de **A**



Masque de **L**



Masque de **U**



Masque de **LU**

FIGURE 4.7 – Première ligne : maillage exemple et masque de **A** associé. Deuxième ligne : masque des facteurs **L** et **U** de la décomposition ILU(0) de **A**. En bas, masque du produit **LU** qui le masque $\mathcal{S}^1$ de la décomposition ILU(1).

l'effet du recouvrement sur le préconditionnement de Schwarz additif. Elle a également permis de tester l'ILU(k) parallélisé par la méthode de Schwarz additif.

Dans le cadre de cette thèse, une adaptation du code AeTher a été réalisée afin de pouvoir utiliser cette bibliothèque. Le principal travail d'adaptation a été de créer un numérotation globale des nœuds compatible avec les exigences de parallélisation de PETSc. En effet, cette bibliothèque demande un découpage algébrique sans recouvrement du système linéaire. Elle suppose donc que les n_1 premières inconnues sont situées sur le processeur 1, les n_2 suivantes sur le processeur 2, *etc.* La deuxième adaptation a été de ne pas utiliser les solveurs de PETSc. Le GMRES avec déflation des petites valeurs propres, décrit dans le chapitre 3, fourni par la bibliothèque PETSc n'a pas donné satisfaction, et n'était pas disponible pour les systèmes linéaires à nombres complexes. Le choix a été fait d'utiliser PETSc uniquement comme boîte à outil d'algèbre linéaire pour réaliser les opérations de base de l'algorithme GMRES s'appliquant aux vecteurs, comme les produits matrice-vecteur, les produits scalaires, ou les combinaisons linéaires de vecteurs. Les vecteurs, la matrice, le préconditionnement sont gérés par PETSc, et l'algorithme GMRES implémenté dans AeTher reste le même. Seul le vecteur solution à la fin de la résolution par la méthode GMRES est rapatrié sous un format compatible d'AeTher (découpage en sous-domaine avec recouvrement minimal) avec une renumérotation adaptée.

Enfin, signalons que la bibliothèque PETSc est prévue pour utiliser des nombres réels ou des nombres complexes, ce choix étant décidé à la compilation. Pour les applications d'aéroélasticité, la bibliothèque compilée pour utiliser des nombres complexes a été utilisée, même pour les cas à fréquence nulle. Dans ce cas-là, on fournit à PETSc des nombres complexes à partie imaginaire nulle. L'utilisation de nombres complexes par rapport à des réels demande deux fois plus de mémoire pour stocker ces nombres et les opérations arithmétiques sont plus coûteuse. Cette perte de performance en mémoire et en temps n'est pas grave car les calculs à fréquence nulle ne représentent que 10 % environ du volume de calculs à réaliser pour une étude d'aéroélasticité. En optimisation de forme aérodynamique, où tous les calculs sont réels, la bibliothèque PETSc utilisée est compilée pour les nombres réels.

4.5.2 Résultats

Le préconditionnement ILU(k) a été utilisé avec succès sur les cas d'aéroélasticité ainsi que d'optimisation. On présente ici trois exemples d'utilisation, deux en aéroélasticité et un en optimisation.

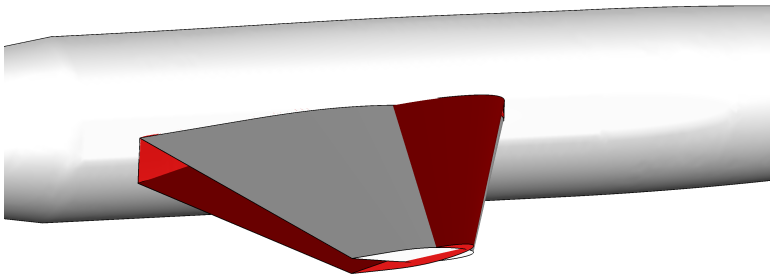


FIGURE 4.8 – Mode de tangage de voilure sur la maquette DTP (cas test II). Voilure déformée en rouge

Aéroélasticité

Sur la maquette DTP (cas test II), le préconditionnement ILU(k) a permis une très forte accélération des calculs d'aéroélasticité. La maillage contient plus de 6 millions de noœuds et est découpé en 512 domaines. Le mouvement retenu est un tangage de l'aile, présenté sur la figure 4.8. La maquette est montrée dans son intégralité sur la figure 3.6. La configuration aérodynamique de référence est de 0° d'incidence et Mach 0,88. L'écoulement est transsonique sur une large partie du profil. Le choc de recompression génère un décollement à son pied, comme le montre la figure 4.9.

L'effet du recouvrement entre sous-domaines du préconditionnement Schwarz additif a été testé, ainsi que l'effet du remplissage du préconditionneur local ILU en comparant l'ILU(0) et l'ILU(1). Les courbes de convergence de ces différents préconditionnements comparés à la référence BSOR sont données sur la figure 4.10. On constate que la méthode GMRES préconditionnée par Schwarz additif avec ILU(0) ne converge pas quel que soit le recouvrement. Le recouvrement noté « d » sur la figure 4.10 correspond à un recouvrement minimal pour d = 1, une absence de recouvrement pour d = 0 (qui correspond à un préconditionnement type Jacobi par bloc [135]) et l'inclusion du premier voisin pour d = 2. Même sans recouvrement, le préconditionnement ILU(1) converge plus rapidement. Cela est un exemple de cas où la méthode de Schwarz additive utilisée comme méthode de résolution ne pourrait pas converger, mais utilisée comme préconditionneur permet d'accélérer la convergence. L'utilisation d'un recouvrement minimal améliore encore la vitesse de convergence, mais l'augmentation du recouvrement (ILU(1) avec d=2) n'apporte pas de gain significatif de convergence. Le temps de résolution des systèmes linéarisés est présenté sur le tableau 4.2. Le meilleur temps de résolution est obtenu par l'ILU(1) avec recouvrement minimal. Le faible gain d'itérations apporté par le plus grand recouvrement ne compense pas le coût supplémentaire de communication et de calcul.

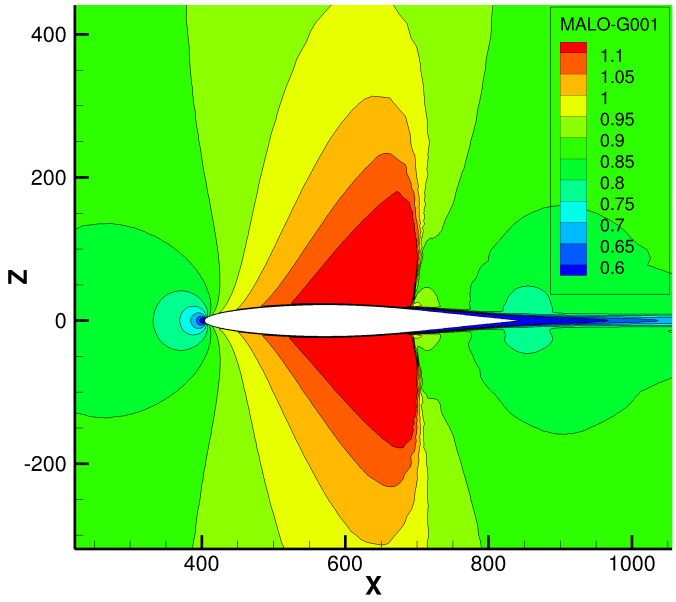


FIGURE 4.9 – Nombre de Mach local sur la maquette DTP (cas test II) pour un écoulement incident à 0° et Mach 0,88. Coupe réalisée au milieu de l'envergure.

TABLEAU 4.2 – Temps de résolution du système pour les différents préconditionnements présentés. Maquette DTP (cas test II).

Préconditionnement Recouvrement	BSOR 1	ILU(1) 0	ILU(1) 1	ILU(1) 2
Temps (s)	420	144	98	107

L'utilisation de l'ILU(2) apporte également un faible gain d'itérations qui donne un temps de résolution presque identique à l'ILU(1).

D'autres cas tests à haute vitesse (écoulement transsonique) ont permis d'arriver aux mêmes conclusions. Ainsi, suite aux travaux de cette thèse, le préconditionneur par défaut pour les calculs linéarisés dans AeTher sur des cas à haute vitesse est l'ILU(1) parallélisé par un Schwarz additif à recouvrement minimal.

Le deuxième exemple d'utilisation de l'ILU(k) est celui d'un Falcon en configuration hypersustentée pour le décollage.

Cas test III (cas Falcon décollage) :
Les becs sont complètement sortis, les volets partiellement braqués, et l'incidence est de 8°. Une vue générale de la configuration de l'avion est présentée sur la figure 4.11. Le maillage de cette configuration comporte plus de 8 millions de nœuds, soit plus de 40 millions d'inconnues. Le maillage est découpé en 512 sous-domaines. Du fait de l'incidence et des dispositifs hypersustentateurs, l'écoulement sur la voilure est complexe. La figure 4.12 montre le facteur de forme incompressible H_i qui est le rapport entre l'épaisseur de déplacement et l'épaisseur de quantité de mouvement. Une couche limite turbulente est décollée pour une valeur de H_i au-delà de 2,4. L'image à droite de la figure 4.12 montre des lignes de frottements sur l'extrados qui confirment la présence d'un décollement de bord de fuite au niveau de l'aileron. Un déplacement linéarisé de tangage est utilisé pour générer le second membre du système.

L'aérodynamique complexe du cas Falon décollage rend difficile la convergence du système linéarisé. Ce système converge avec un préconditionnement BSOR dont le paramètre de relaxation ω est réglé à 0,5. Les préconditionneurs ILU avec remplissage permettent la convergence pour un degré de remplissage k supérieur à 3. Le préconditionnement par une factorisation incomplète ILU(1) ou ILU(2) donnait des résidus initiaux extrêmement grand (supérieurs à 10^{10}) et des systèmes ne convergeant pas avec la méthode GMRES. Les courbes de convergence pour les différents préconditionnemenst ILU(3), ILU(4) et BSOR sont présentées sur la figure 4.13. Le préconditionnement ILU(k) divise au moins par dix le nombre d'itérations nécessaire à la convergence du système. Augmenter le degré de remplissage de l'ILU améliore un peu la convergence. L'influence du recouvrement est nette entre « d =

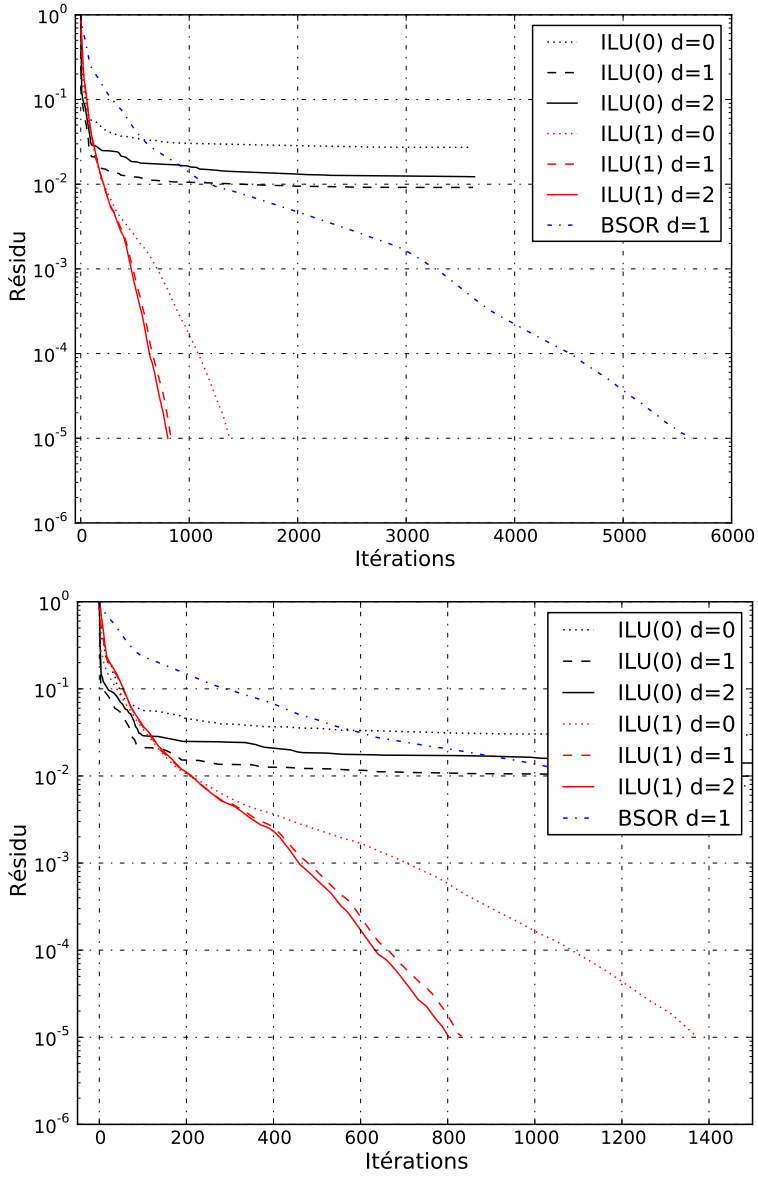


FIGURE 4.10 – Résidu normalisé du système linéarisé selon le remplissage de l'ILU et le recouvrement noté « d » sur le cas test II (maquette DTP). En bas, détail des 1500 premières itérations.

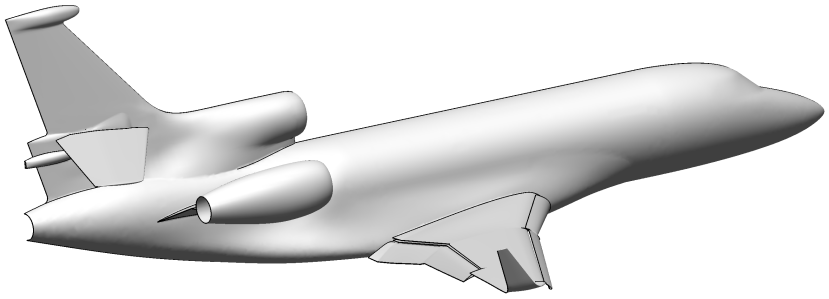


FIGURE 4.11 – Vue générale du cas Falcon décollage (cas test III).

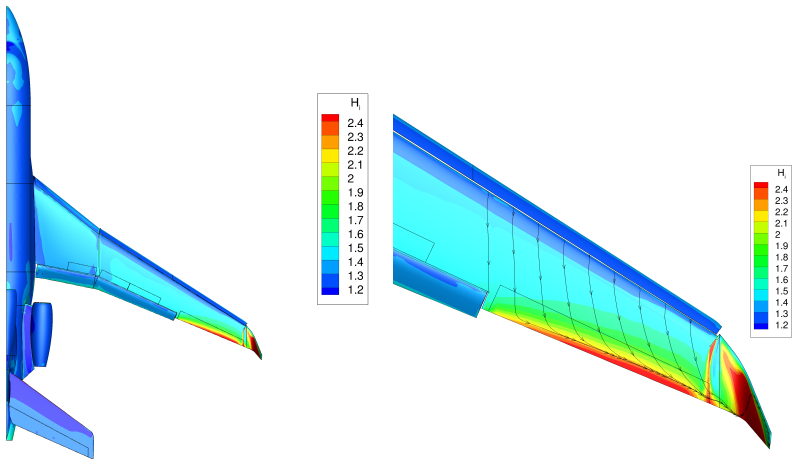


FIGURE 4.12 – Vue du facteur de forme incompressible H_i pour le cas Falcon décollage (cas test III). À droite, détail sur l'extrémité de l'aile et lignes de frottement indiquant le décollement de bord de fuite.

TABLEAU 4.3 – Temps de résolution du système pour les différents préconditionnements présentés. Cas Falcon décollage (cas test III)

Préconditionnement	BSOR	ILU(3)	ILU(3)	ILU(3)
Recouvrement	1	0	1	2
Temps (s)	1102	298	222	216
Préconditionnement		ILU(4)	ILU(4)	ILU(4)
Recouvrement		0	1	2
Temps (s)		362	300	295

0 » qui ne correspond à aucun recouvrement du préconditionnement (*i.e.* un préconditionneur de Jacobi par bloc) et le recouvrement minimal noté « d = 1 ». Au-delà du recouvrement minimal, l'amélioration de la convergence est moins sensible.

Le temps de résolution pour tous ces systèmes est présenté sur le tableau 4.3. On en déduit que le coût d'application du préconditionnement ILU(k) est supérieur à celui du BSOR, mais reste suffisamment petit pour permettre une accélération par cinq du temps de résolution.

Optimisation de forme aérodynamique

En optimisation, l'utilisation du préconditionnement ILU(k) a permis la convergence de systèmes qui jusqu'alors était impossible, comme sur le cas test suivant.

Une optimisation de traînée a été réalisée sur une forme de type Falcon, représentée sur la figure 4.14. Ce cas sera désigné par la suite « Falcon générique ». La variable à optimiser est la traînée, notée Cx, sous contrainte d'une portance Cz ne diminuant pas. Les efforts de pression et de frottement ont été pris en compte. Les variables de formes sont des vrillages locaux de 6 sections de l'aile. Cette optimisation est réalisée pour un point de vol de croisière, à savoir Mach 0,8 et 2° d'incidence.

Pour avoir une référence du gradient des variables d'observation par rapport aux variables de forme, une différence finie centrée (intitulée « DF » dans les résultats) a été réalisée. La comparaison avec ces gradients de référence ne peut être bonne qu'en gardant dans le système linéarisé le gradient de la stabilisation (voir le chapitre 5), au prix d'une plus grande difficulté à faire converger les systèmes linéaires. Sans l'utilisation du précondionnement ILU(1), le système linéaire incluant le gradient de stabilisation ne convergeait pas en approche adjointe ou directe.

La figure 4.15 montre les courbes de convergence des systèmes linéaires. Les courbes nommées « ILU(1) » et « BSOR » incluent le gradient de la stabilisation. Le préconditionnement BSOR n'est pas assez performant pour

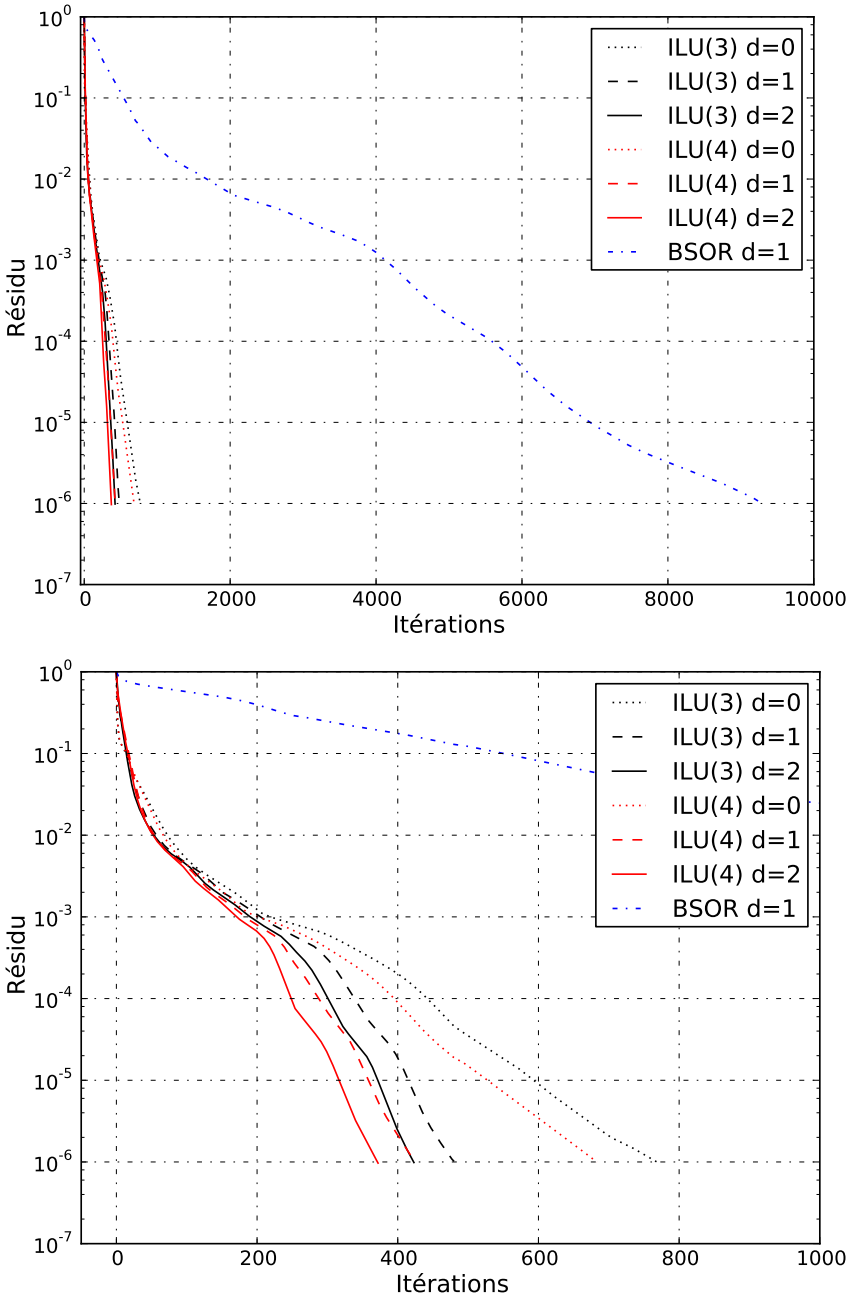


FIGURE 4.13 – Résidus adimensionnés pour le cas Falcon décollage (cas test III). En bas, zoom sur les 1000 premières itérations. « d » désigne le recouvrement des domaines pour le préconditionnement de Schwarz additif.

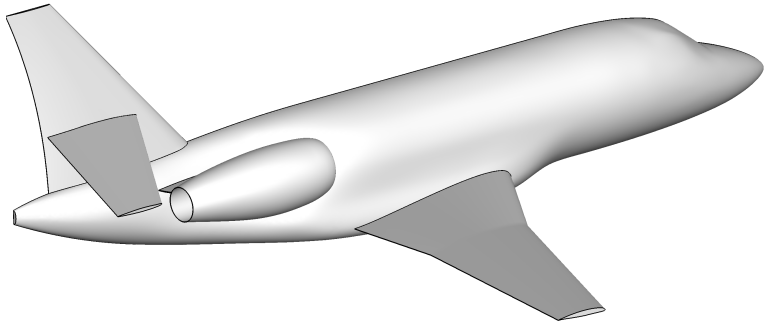


FIGURE 4.14 – Vue d'ensemble du Falcon générique utilisée pour l'optimisation.

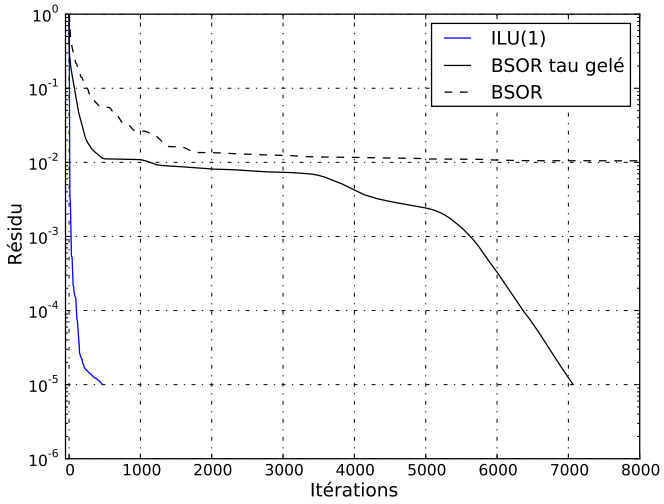


FIGURE 4.15 – Courbes de convergence du système adjoint de la traînée pour le cas d'optimisation sur Falcon générique.

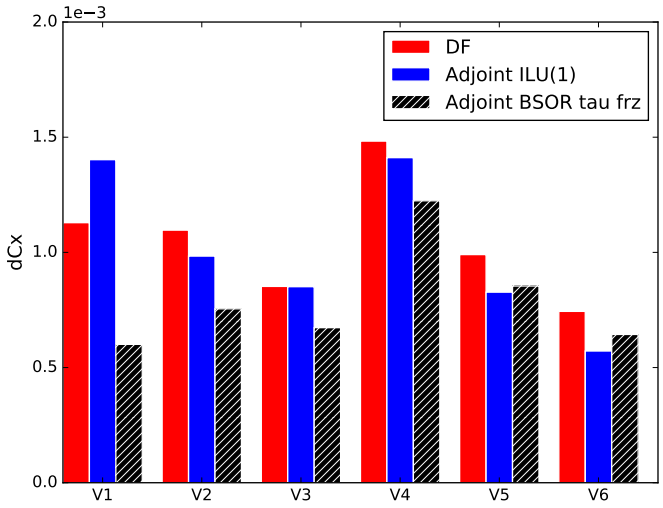


FIGURE 4.16 – Gradient de traînée sur les 6 variables géométriques de l'optimisation sur Falcon générique. Différences finies centrées notées « DF »

faire converger un système avec gradient de la stabilisation. La courbe « BSOR tau gelé » désigne un calcul sans gradient de stabilisation préconditionné par BSOR. L'utilisation du préconditionnement ILU(1) permet d'accélérer grandement la résolution.

Le gradient de traînée induits par les 6 variables géométriques est tracé sur la figure 4.16. Les gradients calculés par approche adjointe sont assez proches de la référence apportée par la différence finie centrée. Contrairement à d'autres cas industriels, considérer le gradient de la stabilisation n'apporte pas un gain évident de précision sur le gradient de la traînée.

4.6 Conclusion

Ce chapitre a permis d'apprécier l'importance du préconditionnement pour la résolution itérative des systèmes linéaires. La parallélisation du préconditionnement s'effectue grâce à des idées issues des méthodes de décomposition de domaine. La méthode la plus intuitive est de préconditionner la matrice distribuée par la somme de l'inverse de matrices locales $\mathbf{A}_i$, méthode que l'on appelle Schwarz additif.

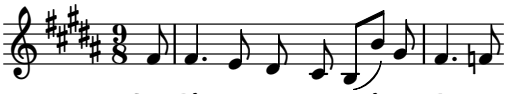
L'utilisation de l'ILU avec remplissage partiel pour approcher l'inverse des matrices locales dans le préconditionnement de Schwarz additif a permis de fortement réduire le nombre d'itérations et le temps nécessaires à la convergence des systèmes linéarisés. Son utilisation industrielle donne grande satisfaction. Les systèmes linéaires en aéroélasticité sont aujourd'hui résolus quotidiennement sur nos calculateurs en des temps de l'ordre de la minute. Par ailleurs, l'ILU(k) a été nécessaire pour résoudre les systèmes associant la

turbulence linéarisée [38].

Les préconditionneurs de Schwarz additif à deux niveaux ont été testés, à l'aide de divers espaces grossiers. L'espace grossier de Nicolaides, simplement défini comme la fonction constante par domaine, n'a pas donné satisfaction. Cet espace grossier est conçu pour les équations de Poisson, mais la méthode ne s'étend donc pas aux équations de Navier-Stokes linéarisées. L'espace grossier GDSW a été également testé. Sa définition et son implémentation sont plus complexes. L'implémentation en 3D n'a pas été menée à terme, mais les résultats en 2D laissent présager peu d'intérêt pour cet espace. Les préconditionneurs de Schwarz à deux niveaux n'ont pas donné satisfaction pour les espaces grossiers testés.

Deuxième partie

Schéma de discrétisation



Berlioz, *Songes d'une nuit d'été*, 2. Le spectre de la rose

Chapitre 5

Stabilisation

Ce chapitre s'intéresse à la stabilisation des éléments finis par la méthode SUPG (pour *Streamline Upwind Petrov-Galerkin*) [24] que le code AeTher utilise. Cette méthode de stabilisation consiste en la modification des fonctions test, afin de décentrer la discrétisation des termes d'advection. Si l'on note $\mathcal{L}_{\text{Adv}} : \mathbf{V} \mapsto \widetilde{\mathbf{A}}_i \mathbf{V}_{,i}$ l'opérateur des flux d'Euler et $\mathcal{L} : \mathbf{V} \mapsto \widetilde{\mathbf{A}}_i \mathbf{V}_{,i} + \left(\widetilde{\mathbf{K}}_{ij} \mathbf{V}_{,j}\right)_{,i}$ l'opérateur des équations de Navier-Stokes stationnaires, alors la contribution d'un élément Ω^e à la forme faible stabilisée est :

$$\int_{\Omega^e} (\mathbf{W} + \boldsymbol{\tau} \mathcal{L}_{\text{Adv}} \mathbf{W})^T \cdot (\mathcal{L}\mathbf{V}) \, \mathrm{d}\Omega^e \,. \tag{5.1}$$

La matrice $\boldsymbol{\tau}$, dite de stabilisation, est d'une importance capitale pour la méthode. Elle est définie aux éléments et doit être symétrique et définie positive. La valeur de ses coefficients règle la viscosité numérique introduite dans le calcul. Il existe plusieurs façons de construire la matrice $\boldsymbol{\tau}$. Dans AeTher, cette matrice est construite de manière algébrique, en utilisant les propriétés des opérateurs de convection exprimés en variables entropiques. Cette construction, proposée par Mallet [104], est expliquée en détail dans la première section de ce chapitre.

Ce n'est pas la seule méthode pour construire $\boldsymbol{\tau}$. Par exemple, Le Beau introduit dans [16] une matrice de stabilisation diagonale, multiple de l'identité. Le coefficient scalaire est proportionnel à un rapport de longueur de l'élément sur le rayon spectral de l'opérateur de flux d'Euler. Une définition similaire est proposée dans [78]. On pourra consulter [89] pour un panorama complet des formes du paramètre τ en 1D. La formulation SUPG peut s'interpréter de multiples manières. Hughes *et al.* expliquent le lien fort entre l'erreur commise à une petite échelle et la stabilisation SUPG dans [80], qui permet de s'interroger sur le bien-fondé d'une matrice de stabilisation constante sur chaque élément. Par exemple, Knobloch dans [95] propose une définition du

paramètre τ utilisant les informations des éléments voisins.

La deuxième section de ce chapitre est consacré à l'essai d'une matrice $\boldsymbol{\tau}$ qui est construite sans approximations sur la formule proposée dans [83, 104].

5.1 Forme de la matrice de stabilisation

La stabilisation des équations de Navier-Stokes par la méthode SUPG exige de bien définir la matrice de stabilisation $\boldsymbol{\tau}$. Cette section suit la présentation de Mallet dans [104] et permet de mieux comprendre la nécessité de la stabilisation des équations d'advection ainsi que la construction de cette matrice, dont la formule a été donnée dans la section 2.2.2 et est redonnée ici :

$$\boldsymbol{\tau} = \widetilde{\mathbf{A}}_0^{-1} \left(\frac{\partial \xi_i}{\partial x_j} \frac{\partial \xi_i}{\partial x_k} \mathbf{A}_j \mathbf{A}_k \right)^{-\frac{1}{2}}, \tag{5.2}$$

où $\widetilde{\mathbf{A}}_0$ est la matrice de passage des variables entropiques vers les variables conservatives, $\mathbf{A}_j$ sont les matrices de flux d'Euler et $\frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}}$ est la jacobienne de passage des coordonnées vers celles de l'élément de référence.

5.1.1 Stabilisation d'une équation d'advection 1D

Différence finie centrée et méthode *upwind*

Soit le problème modèle d'advection-diffusion en une dimension suivant :

$$\begin{cases} a u_{,x} - k u_{,xx} = 0 \quad \text{sur }]0, L[\\ u(0) = g_0 \\ u(L) = g_L \end{cases} \tag{5.3}$$

Ici, $a > 0$ représente la vitesse d'advection, et k le coefficient de diffusion. Ces deux grandeurs sont constantes sur tout le domaine. Ce problème admet une solution analytique qui dépend du nombre de Péclet $Pe = \frac{aL}{k}$:

$$u(x) = c_1 + c_2 \exp\left(Pe \frac{x}{L} \right) \tag{5.4}$$

où $c_{1,2}$ sont des constantes dépendant des conditions limites.

On peut chercher une solution approchée à ce problème par des méthodes de différences finies. On découpe le domaine spatial en N éléments de longueur h, délimités par $x_0, x_1, \ldots, x_N$. On note $u_i \simeq u(x_i)$ la discrétisation de u.

Lemme 5.1. *La discrétisation de l'équation d'advection-diffusion* (5.3) *par un schéma de différences finies centrées, utilisant comme approximation*

discrète des dérivées spatiales

$$\begin{cases} u_{i,x} = \dfrac{u_{i+1} - u_{i-1}}{2h} \\ u_{i,xx} = \dfrac{u_{i+1} - 2u_i + u_{i-1}}{h^2}\,, \end{cases} \tag{5.5}$$

donne une solution oscillante si le nombre de Péclet local $\alpha = \frac{ah}{2k}$ est strictement supérieur à 1.

Démonstration. La discrétisation du problème continu (5.3) par le schéma des différences finies centrées, dont les dérivées sont définies dans l'équation (5.5) est

$$a\frac{u_{i+1} - u_{i-1}}{2h} - k\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = 0\,, \tag{5.6}$$

soit en regroupant les termes, et en divisant l'équation par $\frac{k}{h^2}$ pour faire apparaître α :

$$u_{i+1}(\alpha - 1) + 2u_i + u_{i-1}(-\alpha - 1) = 0\,. \tag{5.7}$$

La suite (u_i) a une solution de la forme

$$u_i = c_- x_-^{\,i} + c_+ x_+^{\,i}\,, \tag{5.8}$$

où x_- et x_+ sont racines du polynôme caractéristique de l'équation de récurrence. Ces racines sont réelles et distinctes. Elles valent

$$\begin{cases} x_- = 1 \\ x_+ = \dfrac{1+\alpha}{1-\alpha}\,. \end{cases} \tag{5.9}$$

La solution discrète u_i s'exprime en fonction de constantes c_1 et c_2 dépendant des conditions limites :

$$u_i = c_1 + c_2\left(\frac{1+\alpha}{1-\alpha}\right)^i\,. \tag{5.10}$$

Si $\alpha > 1$, le terme de croissance géométrique $\frac{1+\alpha}{1-\alpha}$ est négatif. La solution discrète va donc présenter des oscillations numériques.

□

La méthode de différence finie décentrée (*upwind*) permet de s'affranchir de ces oscillations. La discrétisation décentrée du terme d'advection fait cette fois-ci uniquement intervenir le terme « au vent » de u_i (d'où le nom anglais de *upwind*), qui est donc u_{i-1}.

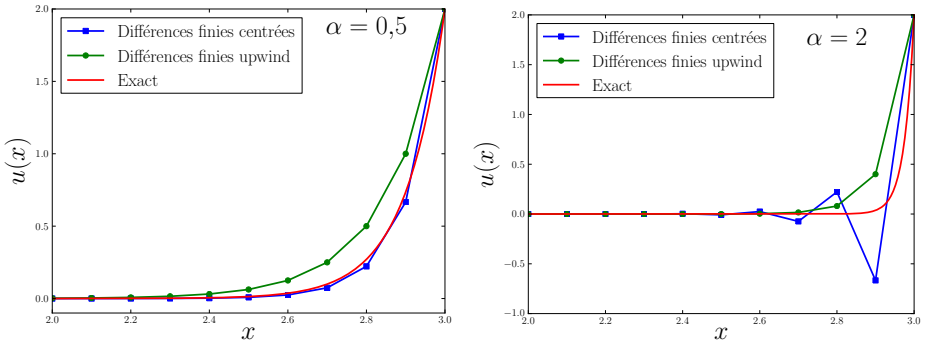


FIGURE 5.1 – Comparaison des méthodes de différences finies centrées et *upwind* pour deux valeurs du nombre de Péclet local α. Zoom sur la fin du segment $[0, 3]$.

Lemme 5.2. *La discrétisation de l'équation d'advection-diffusion* (5.3) *par un schéma de différences finies décentrées, utilisant comme approximation discrète des dérivées spatiales*

$$\begin{cases} u_{i,x} = \dfrac{u_i - u_{i-1}}{h} \\ u_{i,xx} = \dfrac{u_{i+1} - 2u_i + u_{i-1}}{h^2}\,, \end{cases} \tag{5.11}$$

ne présente pas d'oscillations numériques.

Démonstration. L'équation (5.3) devient la relation suivante de récurrence :

$$-u_{i+1} + u_i(\alpha + 2) + u_{i-1}(-\alpha - 1) = 0\,. \tag{5.12}$$

Les deux racines du polynômes sont cette fois-ci

$$\begin{cases} x_1 = 1 \\ x_2 = 1 + 2\alpha\,. \end{cases}$$

On obtient alors la discrétisation de u pour les différences finies décentrées, qui est monotone en i et donc ne présente pas d'oscillations pour tout α :

$$u_i = c_1 + c_2(1 + 2\alpha)^i\ . \tag{5.13}$$

□

La figure 5.1 donne les résultats dans le cas de différences finies centrées et décentrées, et les compare à la solution analytique, pour $L = 3$, $g_0 = 0$ et $g_L = 2$. Si le nombre de Péclet local α est supérieur à 1, on retrouve pour la méthode centrée les oscillations prédites. Les différences finies décentrées ne présentent pas d'oscillations numériques quelque soit le nombre de Péclet,

mais prédisent une couche limite plus épaisse que la solution analytique. On constate que la méthode *upwind* est surdiffusive, alors que la méthode centrée est légèrement sous-diffusive. On va maintenant chercher à corriger la sous-diffusivité du schéma centré en lui rajoutant une viscosité artificielle pour obtenir la solution exacte aux nœuds.

Viscosité artificielle

Théorème 5.3. *Il existe une viscosité artificielle $\tilde{k}$, fonction de α et de h et vérifiant $0 < \tilde{k} \leq ah/2$, qui rajoutée à la viscosité physique du problème d'advection-diffusion* (5.3)*, permet d'obtenir la solution exacte aux nœuds lorsque ce nouveau problème est discrétisé par le schéma des différences finies centrées.*

Démonstration. L'équation d'advection-diffusion devient

$$au_{,x} - \left(k + \tilde{k}\right) u_{,xx} = 0\,. \tag{5.14}$$

Si l'on pose le nouveau nombre de Péclet local $\tilde{\alpha} = \frac{ah}{2(k+\tilde{k})}$, on obtient par un développement similaire la solution du schéma centré :

$$u_i = c_1 + c_2\left(\frac{1+\tilde{\alpha}}{1-\tilde{\alpha}}\right)^i\,. \tag{5.15}$$

On veut qu'il soit égal à la solution exacte, qui vaut

$$u(x_i) = c_3 + c_4 \exp\left(Pe\frac{x_i}{L}\right)\,. \tag{5.16}$$

De plus, $x_i = hi$. Ainsi on doit avoir

$$\forall i \in [1, N-1],\ c_1 + c_2\left(\frac{1+\tilde{\alpha}}{1-\tilde{\alpha}}\right)^i = c_3 + c_4\exp(2i\alpha)\,. \tag{5.17}$$

Nécessairement, $c_1 = c_3$ et $c_2 = c_4$. Ainsi

$$\left(\frac{1+\tilde{\alpha}}{1-\tilde{\alpha}}\right) = \exp(2\alpha) \iff \tilde{\alpha} = \frac{e^{2\alpha}-1}{e^{2\alpha}+1}$$

Cela permet d'exprimer $\tilde{\alpha}$ en fonction de α :

$$\tilde{\alpha} = \tanh(\alpha)\,. \tag{5.18}$$

Or, si l'on pose le nombre sans dimension $\tilde{\xi} = \frac{2\tilde{k}}{ah}$, $\tilde{\alpha}$ s'écrit

$$\tilde{\alpha} = \frac{1}{\frac{1}{\alpha} + \tilde{\xi}}\,. \tag{5.19}$$

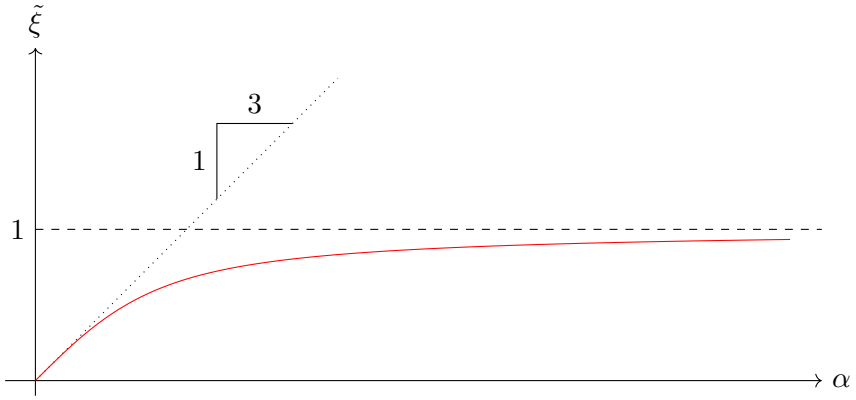


FIGURE 5.2 – Fonction d'amortissement $\tilde{\xi}$ de la correction visqueuse en fonction du nombre de Péclet local α

Ainsi, on obtient la définition suivante de la fonction $\tilde{\xi}$, qui est également tracée sur la figure 5.2 :

$$\tilde{\xi} = \coth \alpha - \frac{1}{\alpha} . \tag{5.20}$$

En utilisant la relation $\tilde{\xi}\alpha = \tilde{k}/k$, la viscosité artificielle $\tilde{k}$ pour rendre le schéma exact aux nœuds vaut

$$\tilde{k} = k\alpha\tilde{\xi}(\alpha) . \tag{5.21}$$

□

La viscosité artificielle $\tilde{k}$ dépend donc de la viscosité physique, du nombre de Péclet local α et d'une fonction d'amortissement $\tilde{\xi}$. Lorsque le nombre de Péclet α tend vers l'infini, $\tilde{\xi}(\alpha)$ tend vers 1. La viscosité artificielle vaut alors $k\alpha = \frac{ah}{2}$, qui est bien le résultat trouvé pour les équations d'advection seules. On va retrouver cette formule dans le cas des éléments finis, abordé dans la section suivante.

Méthode des éléments finis de Galerkin et de Petrov-Galerkin

Dans cette partie, on cherche à appliquer les résultats précédents à la méthode des éléments finis. Des fonctions de forme de Lagrange P1 sont utilisées pour la discrétisation. On introduit l'espace discret $\mathcal{V}^h$:

$$\mathcal{V}^h = \left\{ v \in \mathcal{C}^0([0, L]) \ , \forall\, i, v_{|[x_i, x_{i+1}]} \in \mathcal{P}_1([x_i, x_{i+1}]) \right\} , \tag{5.22}$$

où $\mathcal{C}^0([0; L])$ désigne l'espace des fonctions continues de $[0; L]$ dans $\mathbb{R}$, et $\mathcal{P}_1([x_i, x_{i+1}])$ sont les fonctions linéaire de $[x_i, x_{i+1}]$ dans $\mathbb{R}$.

La solution discrète est cherchée dans l'espace $\mathcal{V}_a^h$, sous-ensemble de $\mathcal{V}^h$ dont les membres vérifient les conditions aux limites en 0 et en L demandées à la solution du problème (5.3) :

$$\mathcal{V}_a^h = \left\{ v \in \mathcal{V}^h,\ v(0) = g_0,\ v(L) = g_L \right\}, \tag{5.23}$$

L'espace affine $\mathcal{V}_a^h$ a un espace vectoriel associé, noté $\mathcal{V}_0^h$ dont les fonctions vérifient des conditions de Dirichlet homogènes au bord :

$$\mathcal{V}_0^h = \left\{ v \in \mathcal{V}^h,\ v(0) = 0,\ v(L) = 0 \right\}. \tag{5.24}$$

Le problème d'advection-diffusion (5.3) discrétisé par des éléments finis de Lagrange P1, c'est-à-dire que la solution discrète u^h est recherchée dans $\mathcal{V}_a^h$ et les fonctions de pondération w^h appartiennent à $\mathcal{V}_0^h$, s'écrit :

$$\forall\, w^h \in \mathcal{V}_0^h, \int_0^L -w^h_{,x} a u^h + w^h_{,x} k u^h_{,x} \mathrm{d}x = 0 \tag{5.25}$$

Lemme 5.4. *La discrétisation du problème d'advection-diffusion* (5.3) *par la méthode des éléments finis de Lagrange P1* (5.25) *est identique à celle donnée par le schéma aux différences finies centrées*

Démonstration. Soit l'élément de référence le segment $[-1, 1]$. Les fonctions de formes sur l'élément de référence sont $\mathrm{N}_1 = \frac{1-x}{2}$ et $\mathrm{N}_2 = \frac{1+x}{2}$. Soit un élément Ω^e de longueur h. Calculons les matrices élémentaires de base sur cet élément :

$$\begin{aligned}
M_{ij} &= \int_{\Omega^e} \mathrm{N}_i \mathrm{N}_j \mathrm{d}x = \int_{\Omega^{\mathrm{ref}}} \mathrm{N}_i \mathrm{N}_j \frac{h}{2} \mathrm{d}x\,, & \mathbf{M} &= \frac{h}{3} \begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{pmatrix}, \\
N_{ij} &= \int_{\Omega^e} \mathrm{N}_{i,x} \mathrm{N}_j \mathrm{d}x\,, & \mathbf{N} &= \frac{1}{2} \begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix}, \\
P_{ij} &= \int_{\Omega^e} \mathrm{N}_{i,x} \mathrm{N}_{j,x} \mathrm{d}x\,, & \mathbf{P} &= \frac{1}{h} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.
\end{aligned}$$

La matrice élémentaire correspondant à la forme faible discrète (5.25) sera

$$-a\mathbf{N} + k\mathbf{P} = \begin{pmatrix} \frac{a}{2} + \frac{k}{h} & \frac{a}{2} - \frac{k}{h} \\ -\frac{a}{2} - \frac{k}{h} & -\frac{a}{2} + \frac{k}{h} \end{pmatrix}. \tag{5.26}$$

On en déduit qu'après assemblage, la discrétisation pour h constant et égal à h sera après division par h

$$u_{i-1}\left(-\frac{a}{2h} - \frac{k}{h^2}\right) + u_i \frac{2k}{h^2} + u_{i+1}\left(\frac{a}{2h} - \frac{k}{h^2}\right) = 0\,.$$

On reconnaît exactement la discrétisation par la méthode des différences finies centrées. □

Les éléments finis de Galerkin standard, où les fonctions de forme et test sont prises dans le même espace, seront donc instables pour des nombres de Péclet locaux supérieurs à 1.

Théorème 5.5. *La modification des fonctions tests w^h en $w^h + \tau a w^h_{,x}$, où $\tau = \frac{h}{2a}$ est le paramètre de stabilisation défini à l'élément, dans la formulation faible de Galerkin standard* (5.25) *conduit à une discrétisation décentrées upwind.*

Démonstration. La contribution d'un élément Ω^e de longueur h à la formulation stabilisée s'écrit, après intégration par partie

$$\int_{\Omega^e} -w^h_{,x} a u^h + w^h_{,x} k u^h_{,x} + w^h_{,x} a \tau \Big(a u^h_{,x} - k u^h_{,xx} \Big) \, \mathrm{d}x \,. \tag{5.27}$$

Or $u^h \in \mathcal{V}^h_a$. Ainsi, $u^h_{,xx} = 0$. On en déduit que (5.27) se simplifie en

$$\int_{\Omega^e} -w^h_{,x} a u^h + w^h_{,x} k u^h_{,x} + w^h_{,x} a \tau a u^h_{,x} \mathrm{d}x \,. \tag{5.28}$$

Cette équation correspond à la matrice élémentaire :

$$-a\mathbf{N} + k\mathbf{P} + \tau a^2 \mathbf{P} = \begin{pmatrix} a + \frac{k}{h} & -\frac{k}{h} \\ -a - \frac{k}{h} & \frac{k}{h} \end{pmatrix} . \tag{5.29}$$

On retrouve bien la discrétisation du schéma de différences finies décentrées, si les éléments sont de longueurs constantes h, après assemblage de la matrice globale. □

On remarque que si $a < 0$, le terme diffusif τa^2 pourrait devenir négatif, car $\tau = \frac{h}{2a}$. Dans ce cas, utiliser $\tau = \frac{h}{-2a} = \frac{h}{2|a|}$ permet de garder une viscosité ajoutée positive et de décentrer correctement le terme d'advection quelque soit sa direction.

Corollaire 5.6. *Si l'on pose $\tau = k\alpha\tilde{\xi}(\alpha)$, la solution de l'équation d'advection-diffusion* (5.3) *discrétisée par les éléments finis stabilisé par SUPG est exacte aux nœuds*

Démonstration. Ce n'est que la conséquence du théorème 5.3 appliqué aux résultats issus de la démonstration du théorème 5.5 □

On note que pour les fonctions de forme P1 utilisées, la perturbation $p^h\Big(w^h\Big) = \tau a w^h_{,x}$ est constante par élément. Les fonctions tests, représentées sur la figure 5.3, seront donc discontinues.

Pour résumer, on a trouvé dans cette section que les différences finies centrées sont instables pour des nombres de Péclet locaux supérieur à 1. Pour pallier cette instabilité, on discrétise de manière décentrée l'opérateur d'advection. Alors que la méthode centrée n'est pas assez diffusive, la discrétisation décentrée est légèrement trop dissipative. Rajouter une viscosité

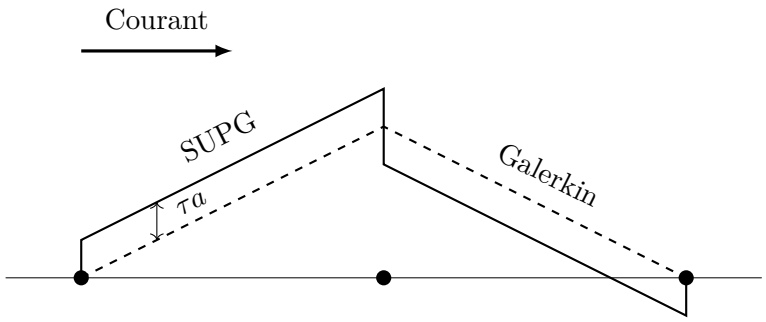


FIGURE 5.3 – Fonction test sur un élément pour la méthode de Galerkin et la méthode SUPG

artificielle bien choisie à la discrétisation centrée permettait d'obtenir un schéma optimal, c'est-à-dire exact aux nœuds. Pour des éléments finis, la méthode de Galerkin standard, où les fonctions de forme et de tests sont identiques sur l'élément, conduit à un schéma identique aux différences finies centrées. La stabilisation SUPG permet d'ajouter un terme visqueux au schéma. Le choix du paramètre τ conduit à retrouver un décentrement ou un schéma exact aux nœuds. La modification des fonctions tests est définie à l'élément, ce qui corrige parfaitement la discrétisation de l'advection quelque soit la taille de l'élément.

5.1.2 Stabilisation d'un système d'équations d'advection 1D

On considère le système d'advection pure unidimensionnel suivant

$$\mathbf{A}\mathbf{U}_{,x} = \mathbf{0}\,. \tag{5.30}$$

On fait l'hypothèse que la matrice d'advection $\mathbf{A}$ est symétrique. À ce titre, $\mathbf{A}$ est diagonalisable. Les variables exprimées dans la base de ces vecteurs propres sont appelées caractéristiques, par analogie avec les équations d'Euler.

Théorème 5.7. *La matrice de stabilisation*

$$\boldsymbol{\tau} = \frac{h}{2}|\mathbf{A}|^{-1}\,,$$

permet un décentrement parfait de la discrétisation du système d'advection pure par la méthode SUPG exprimé dans les variables caractéristiques

Démonstration. On note $\mathbf{S}$ la matrice dont les colonnes sont vecteurs propres de $\mathbf{A}$, et $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_i)$ les valeurs propres associées. Ainsi, $\mathbf{AS} = \mathbf{S}\boldsymbol{\Lambda}$. On définit alors la valeur absolue de la matrice $\mathbf{A}$ comme $|\mathbf{A}| = \mathbf{S}\,\mathrm{diag}(|\lambda_i|)\,\mathbf{S}^T$.

En négligeant les termes de bord, la forme faible du terme advectif pur (5.30) est

$$-\int_\Omega \mathbf{W}_{,x} \cdot (\mathbf{AU})\, \mathrm{d}\Omega = 0\,. \tag{5.31}$$

On se place d'abord dans le cadre des éléments finis de Galerkin, c'est-à-dire que $\mathbf{U}$ et $\mathbf{W}$ sont prises dans un espace affine et un espace vectoriel discrétisé par les mêmes fonctions de forme.

Comme $\mathbf{S}^T\mathbf{AS} = \mathbf{\Lambda}$, on pose $\mathbf{Y} = \mathbf{SU}$ et $\mathbf{Z} = \mathbf{SW}$ la solution et la fonction test dans l'espace des variables que l'on appellera caractéristiques par analogie avec les équations d'Euler. Dans ces coordonnées, le système d'équations s'écrit

$$-\int_\Omega \mathbf{Z}_{,x} \cdot (\mathbf{\Lambda Y})\, \mathrm{d}\Omega = 0\,. \tag{5.32}$$

C'est un système diagonal d'équations, composé d'équations scalaires d'advection à la vitesse λ_i :

$$-\int_\Omega z^{(i)}_{,x} \lambda_i y^{(i)} \mathrm{d}\Omega = 0\,. \tag{5.33}$$

où $z^{(i)}$ et $y^{(i)}$ sont les composantes de $\mathbf{Z}$ et de $\mathbf{Y}$.

Pour décentrer la discrétisation de ces équations, on rajoute aux $z^{(i)}$ le terme SUPG $t_i \lambda_i z^{(i)}_{,x}$, avec $t_i = \frac{h}{2|\lambda_i|}$. La forme faible stabilisée donne pour l'élément Ω^e

$$\int_{\Omega^e} z^{(i)}_{,x} \lambda_i y^{(i)} + z^{(i)}_{,x} \lambda_i t_i \lambda_i y^{(i)}_{,x} \mathrm{d}x = 0 \tag{5.34}$$

Sous forme matricielle, on obtient un système diagonal

$$\int_{\Omega^e} \mathbf{Z}^T_{,x} \mathbf{\Lambda Y} + (\mathbf{T\Lambda Z}_{,x}) \cdot \mathbf{\Lambda Y}_{,x} \mathrm{d}x = 0 \tag{5.35}$$

où $\mathbf{T} = \mathrm{diag}(t_i)$ est la matrice diagonale de stabilisation. Dans les variables premières $\mathbf{U}$ et $\mathbf{W}$, ce système s'écrit

$$\int_{\Omega^e} \mathbf{W}^T_{,x} \mathbf{S\Lambda S}^T \mathbf{U} + \left(\mathbf{T\Lambda S}^T \mathbf{W}_{,x}\right) \cdot \mathbf{\Lambda S}^T \mathbf{U}_{,x} \mathrm{d}x =$$
$$\int_{\Omega^e} \mathbf{W}^T_{,x} \mathbf{AU} + \mathbf{W}^T_{,x} \mathbf{S\Lambda T\Lambda S}^T \mathbf{U}_{,x} \mathrm{d}x\,.$$

Or $\mathbf{S\Lambda} = \mathbf{AS}$ et $\mathbf{S}^T\mathbf{A} = \mathbf{\Lambda S}^T$. Ainsi

$$\mathbf{S\Lambda T\Lambda S}^T = \mathbf{ASTS}^T\mathbf{A}\,.$$

On note $\boldsymbol{\tau} = \mathbf{STS}^T$. D'où

$$\int_{\Omega^e} \mathbf{W}^T_{,x} \mathbf{AU} + (\boldsymbol{\tau} \mathbf{AW}_{,x}) \cdot \mathbf{AU}_{,x} \mathrm{d}x = 0\,. \tag{5.36}$$

Ainsi $\boldsymbol{\tau}$ est bien la matrice de stabilisation de la méthode SUPG. On rappelle que $\mathbf{T} = \frac{h}{2}|\mathbf{\Lambda}|^{-1}$. On retrouve bien

$$\boldsymbol{\tau} = \frac{h}{2}|\mathbf{A}|^{-1}\,.$$

□

Une autre façon d'écrire $\boldsymbol{\tau}$ est de remarquer que $|\mathbf{A}| = \left(\mathbf{A}^2\right)^{1/2}$ et que $\frac{h}{2} = \frac{\partial x}{\partial \xi}$, si ξ est la coordonnée dans l'élément de référence $[-1, 1]$. Ainsi

$$\boldsymbol{\tau} = \left(\left(\frac{\partial \xi}{\partial x}\right)^2 \mathbf{A}^2\right)^{-\frac{1}{2}} \tag{5.37}$$

Remarque 5.1. L'extension de ce résultat à l'équation d'advection-diffusion $\mathbf{A}\mathbf{U}_{,x} - \mathbf{K}\mathbf{U}_{,xx} = \mathbf{0}$ se fait facilement si l'on suppose que $\mathbf{K}$ est également diagonalisable par $\mathbf{S}$. On retrouve alors des équations scalaires d'advection-diffusion dont la stabilisation optimale a été donnée par le corollaire 5.6 à la section précédente : il suffit de moduler les valeurs propres de $\boldsymbol{\tau}$, à savoir les t_i par la fonction de pondération $\tilde{\xi}$ en fonction du nombre de Péclet de chaque variable caractéristique.

5.1.3 Cas multidimensionnel

Lemme 5.8. *La viscosité articielle rajoutée par la stabilisation SUPG vaut*

$$\mathbf{K} = |\mathbf{B}|\,, \tag{5.38}$$

où $\mathbf{B} = \frac{\partial x}{\partial \xi}\mathbf{A}$ *est l'opérateur d'advection exprimé dans l'élément de référence, et* $|\mathbf{B}|$ *désigne la valeur absolue de* $\mathbf{B}$*, définie par* $|\mathbf{B}| = \mathbf{S}\,\mathrm{diag}(|\lambda_i|)\,\mathbf{S}^T$*, si* $\mathbf{S}$ *sont les vecteurs propres de* $\mathbf{B}$ *et* (λ_i) *les valeurs propres associées.*

Démonstration. Le terme SUPG ajouté est de la forme

$$\int_{\Omega^e} \mathbf{W}_{,x}^T \mathbf{A}\boldsymbol{\tau}\mathbf{A}\mathbf{U}_{,x}\mathrm{d}x\,,$$

qui correspond à un terme visqueux après intégration par partie. Par identification, la valeur de la viscosité $\mathbf{K}$ est

$$\mathbf{K} = \mathbf{A}\boldsymbol{\tau}\mathbf{A} = \mathbf{A}\frac{h}{2}\left(\mathbf{A}^2\right)^{1\frac{1}{2}}\mathbf{A} = \frac{h}{2}|\mathbf{A}|\,,$$

ce qui fournit le résultat demandé. □

Dans le cas multidimensionnel, le terme d'advection $\mathbf{A}\mathbf{U}_{,x}$ devient $\mathbf{A}^T\nabla\mathbf{U}$, où

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \end{pmatrix} ,$$

où les matrices $\mathbf{A}_i$ sont symétriques. Dans l'élément de référence, ce terme d'advection s'écrit

$$\mathbf{B}_i \mathbf{U}_{,\xi_i} ,$$

où $\mathbf{B}_i = \frac{\partial \xi_i}{\partial x_k} \mathbf{A}_k$. De même, on note, dans le cas de trois dimensions spatiales :

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \mathbf{B}_3 \end{pmatrix}$$

Lemme 5.9 (Extension multi-dimensionnelle)**.** *L'extension à plusieurs dimensions de la matrice de stabilisation est donnée par*

$$\boldsymbol{\tau} = \left(\mathbf{B}^T \mathbf{B}\right)^{-\frac{1}{2}} , \tag{5.39}$$

ce qui permet d'avoir une viscosité rajoutée par la stabilisation SUPG valant

$$\mathbf{K} = \sqrt{\mathbf{B}\mathbf{B}^T} , \tag{5.40}$$

qui est une extension proposée de la valeur absolue matricielle à une matrice rectangulaire.

Démonstration. L'extension de la valeur absolue à une matrice rectangulaire n'est évidemment pas exacte. On notera cependant qu'elle est bien définie puisque $\mathbf{B}\mathbf{B}^T$ est symétrique et positive. La formule donnée de $\boldsymbol{\tau}$ permet bien de retrouver celle de $\mathbf{K}$ car

$$\begin{aligned} \left(\mathbf{B}\tau\mathbf{B}^T\right)^2 &= \mathbf{B}\left(\mathbf{B}^T\mathbf{B}\right)^{-\frac{1}{2}} \mathbf{B}^T\mathbf{B}\left(\mathbf{B}^T\mathbf{B}\right)^{-\frac{1}{2}} \mathbf{B}^T \\ &= \mathbf{B}\left(\mathbf{B}^T\mathbf{B}\right)^{\frac{1}{2}} \left(\mathbf{B}^T\mathbf{B}\right)^{-\frac{1}{2}} \mathbf{B}^T \\ &= \mathbf{B}\mathbf{B}^T = \mathbf{K}^2 . \end{aligned}$$

Lorsque $\mathbf{B}\mathbf{B}^T$ n'est pas définie positive, on prend l'inverse de sa racine carrée sur un espace supplémentaire à son noyau. □

On en déduit la forme finale de la matrice de stabilisation :

$$\boldsymbol{\tau} = \left(\mathbf{B}^T\mathbf{B}\right)^{-\frac{1}{2}} = \left(\frac{\partial \xi_i}{\partial x_j}\frac{\partial \xi_i}{\partial x_k}\mathbf{A}_j\mathbf{A}_k\right)^{-\frac{1}{2}} , \tag{5.41}$$

où une somme est prise implicitement sur les indices i, j et k.

Dans cette section, il n'a pas fait été usage d'une définition spectrale de $\boldsymbol{\tau}$. En une dimension, la définition de la matrice de stabilisation par les vecteurs propres de $\mathbf{A}$ permettait un décentrement parfait du terme d'advection par caractéristiques. L'extension au cas multidimensionnel demande d'approcher la valeur absolue d'une matrice non carrée par l'utilisation d'une racine carrée. Cela fait perdre le décentrement par caractéristique, sauf si les matrices $\mathbf{A}_i$ sont diagonalisables dans une même base.

5.1.4 Variables entropiques

Pour étendre le résultat précédent aux variables entropiques, considérons les équations d'Euler temporelles suivant :

$$\widetilde{\mathbf{A}}_0 \mathbf{V}_{,t} + \widetilde{\mathbf{A}}_i \mathbf{V}_{,i} = \mathbf{0}\,. \tag{5.42}$$

Les matrices $\widetilde{\mathbf{A}}_i$ sont symétriques. Si l'on appliquait directement le résultat de la section 5.1.2, en utilisant la décomposition spectrale de la matrice $\widetilde{\mathbf{A}}_i$ comme on l'a fait dans la preuve du théorème 5.7, le système exprimé dans les variables caractéristiques ne serait pas diagonal. En effet, si $\mathbf{T}$ est la matrice des vecteurs propres de $\widetilde{\mathbf{A}}_i$, rien n'indique que $\mathbf{T}^T \widetilde{\mathbf{A}}_0 \mathbf{T}$ soit une matrice diagonale.

Pour assurer le découplage des variables, on utilise une factorisation de Cholesky de $\widetilde{\mathbf{A}}_0 = \mathbf{L}\mathbf{L}^T$, qui est symétrique définie positive, avant de poser le changement de variables $\widehat{\mathbf{V}} = \mathbf{L}^T \mathbf{V}$. Les équations d'Euler temporelles exprimées avec ces nouvelles variables s'écivent

$$\widehat{\mathbf{V}}_{,t} + \widehat{\mathbf{A}}_i \widehat{\mathbf{V}}_{,i} = \mathbf{0}\,,$$

où les matrices $\widehat{\mathbf{A}}_i = \mathbf{L}^{-1} \widetilde{\mathbf{A}}_i \mathbf{L}^{-T} = \mathbf{L}^{-1} \mathbf{A}_i \mathbf{L}$ sont symétriques.

Théorème 5.10. *La matrice de stabilisation des équations d'Euler exprimées en variables entropiques est*

$$\boldsymbol{\tau} = \widetilde{\mathbf{A}}_0^{-1} \left(\frac{\partial \xi_i}{\partial x_j} \frac{\partial \xi_i}{\partial x_k} \mathbf{A}_j \mathbf{A}_k \right)^{-1/2} .$$

Démonstration. Pour se placer dans les hypothèses de la section précédente, on utilise les équations d'Euler exprimées avec les variables $\widehat{\mathbf{V}}$:

$$\widehat{\mathbf{V}}_{,t} + \widehat{\mathbf{A}}_i \widehat{\mathbf{V}}_{,i} = \mathbf{0}\,,$$

D'après la preuve du lemme 5.9, la matrice de stabilisation de ces équations est

$$\widehat{\boldsymbol{\tau}} = \left(\widehat{\mathbf{B}}_i \widehat{\mathbf{B}}_i \right)^{-\frac{1}{2}} ,$$

où $\widehat{\mathbf{B}}_i = \mathbf{L}^{-1}\mathbf{B}_i\mathbf{L}$. On en déduit que

$$\widehat{\boldsymbol{\tau}} = \mathbf{L}^{-1}(\mathbf{B}_i\mathbf{B}_i)^{-1/2}\mathbf{L}\,.$$

Le changement de variable $\mathbf{V} \mapsto \widehat{\mathbf{V}} = \mathbf{L}^T\mathbf{V}$ appliqué à la formulation faible stabilisée permet de retrouver la matrice de stabilisation $\boldsymbol{\tau}$ des équations d'Euler exprimées en variables entropiques :

$$\begin{aligned}\boldsymbol{\tau} &= \mathbf{L}^{-T}\widehat{\boldsymbol{\tau}}\mathbf{L}^{-1}\\ &= \mathbf{L}^{-T}\mathbf{L}^{-1}(\mathbf{B}_i\mathbf{B}_i)^{-1/2}\mathbf{L}\mathbf{L}^{-1}\\ &= \widetilde{\mathbf{A}}_0^{-1}(\mathbf{B}_i\mathbf{B}_i)^{-1/2}\,.\end{aligned}$$

L'expression des opérateurs de convection dans l'élément courant donne

$$\boldsymbol{\tau} = \widetilde{\mathbf{A}}_0^{-1}\left(\frac{\partial\xi_i}{\partial x_j}\frac{\partial\xi_i}{\partial x_k}\mathbf{A}_j\mathbf{A}_k\right)^{-1/2}\,.$$

□

On retrouve bien la forme de la matrice de stabilisation donnée dans [83, 104, 145] et rappelée dans l'équation (5.2).

L'extension aux équations de Navier-Stokes se fait en utilisant la remarque 5.1. On diagonalise la matrice $\boldsymbol{\tau}$ pour en tirer la base de ses vecteurs propres. La diagonale de la matrice de diffusion exprimée dans cette base permet d'obtenir une estimation de la diffusion dans la direction de chacun des vecteurs propres de $\boldsymbol{\tau}$. C'est ce coefficient de diffusion qui servira à exprimer un nombre de Péclet nécessaire à moduler chacune des valeurs propres de $\boldsymbol{\tau}$. Plus de détails sont donnés dans [104, 83] ainsi que dans la section suivante.

5.1.5 Calcul pratique de la matrice de stabilisation

La méthode de calcul de $\boldsymbol{\tau}$ que l'on va détailler ici utilise les simplifications apportées par l'expression des opérateurs d'advection exprimées dans le repère des lignes de courant.

On se place dans le repère tangent aux lignes de courant. Une illustration de ce repère est donnée dans la figure 5.4. On note $\mathbf{s}$ le vecteur de coordonnées $\mathbf{s} = (s, n, t)$ exprimé dans la base $(\mathbf{e}_s, \mathbf{e}_n, \mathbf{e}_t)$, où $\mathbf{e}_s$ est le vecteur unitaire tangent aux lignes de courant. On indexe par des lettres grecques les vecteurs unitaires de la cette base : $(\mathbf{e}_\alpha)\,, \alpha \in (s, n, t)$. On note $\mathbf{R} = \frac{\partial\mathbf{x}}{\partial\mathbf{s}}$ la matrice de rotation du repère tangent aux lignes de courant vers les coordonnées $\mathbf{x}$. $\mathbf{Q}$ est la matrice de rotation permettant d'exprimer les variables entropiques dans le repère des lignes de courant :

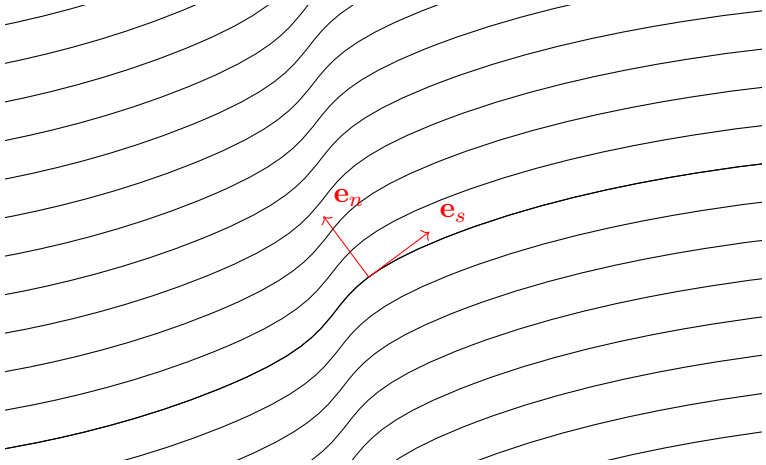


FIGURE 5.4 – Illustration du repère tangent $(\mathbf{e}_s, \mathbf{e}_n)$ en deux dimensions.

$$\mathbf{Q} = \begin{pmatrix} 1 & & \\ & \mathbf{R} & \\ & & 1 \end{pmatrix} . \tag{5.43}$$

Les opérateurs d'advection dans les directions (s, n, t) s'écrivent :

$$\widetilde{\mathbf{A}}_\alpha = \mathbf{Q}^T \widetilde{\mathbf{A}}_i \mathbf{Q} R_{i\alpha} \,, \tag{5.44}$$

où $R_{i\alpha}$ est le terme de $\mathbf{R}$ à la ligne i et la colonne α.

Soit $\widetilde{\mathbf{S}}$ la solution du problème aux valeurs propres généralisé $\widetilde{\mathbf{A}}_s \widetilde{\mathbf{S}} = \widetilde{\mathbf{A}}_0^s \widetilde{\mathbf{S}} \mathbf{\Lambda}_s$, où $\widetilde{\mathbf{A}}_s$ est la jacobienne du flux d'Euler dans la direction du courant et $\widetilde{\mathbf{A}}_0^s = \mathbf{Q}^T \widetilde{\mathbf{A}}_0 \mathbf{Q}$ la matrice de changement de variable exprimée dans le repère lié aux lignes de courant. On remarque que $\mathbf{\Lambda}_s = \mathrm{diag}(u, u, u, u + c, u - c)$.

On note de même $\mathbf{\Lambda}_\alpha$ les matrices telles que $\widetilde{\mathbf{A}}_\alpha \widetilde{\mathbf{S}} = \widetilde{\mathbf{A}}_0^s \widetilde{\mathbf{S}} \mathbf{\Lambda}_\alpha$. Pour $\alpha \neq s$, ces matrices ne sont pas diagonales, mais sont diagonales avec quatre termes extra-diagonaux.

Théorème 5.11. *La matrice de stabilisation issue de la formule* (5.2) *peut se réécrire sous la forme*

$$\boldsymbol{\tau} = \left(\mathbf{Q}\widetilde{\mathbf{S}}\mathbf{X}\right) \mathbf{\Lambda}^{-\frac{1}{2}} \left(\mathbf{Q}\widetilde{\mathbf{S}}\mathbf{X}\right)^T , \tag{5.45}$$

où $\mathbf{X}$ *est la matrice des vecteurs propres de* $\left(\frac{\partial \xi_i}{\partial s_\alpha} \frac{\partial \xi_i}{\partial s_\beta} \mathbf{\Lambda}_\alpha \mathbf{\Lambda}_\beta\right)$ *de valeurs propres* $\mathbf{\Lambda}$.

Démonstration. En reprenant les notations de la section 5.1.4,

$$\boldsymbol{\tau} = \mathbf{L}^{-T} \widehat{\boldsymbol{\tau}} \mathbf{L}^{-1} .$$

On rappelle que $\widehat{\boldsymbol{\tau}} = \left(\frac{\partial \xi_i}{\partial x_j} \frac{\partial \xi_i}{\partial x_k} \widehat{\mathbf{A}}_j \widehat{\mathbf{A}}_k\right)^{-1/2}$.

Soit $\widetilde{\mathbf{A}}_0^s$ la matrice de changement de variables entropiques exprimées dans le repère lié aux lignes de courant : $\widetilde{\mathbf{A}}_0^s = \mathbf{Q}^T \widetilde{\mathbf{A}}_0 \mathbf{Q}$. Soit $\mathbf{L}_s = \mathbf{Q}^T \mathbf{L} \mathbf{Q}$ un facteur de Cholesky de $\widetilde{\mathbf{A}}_0^s$. On peut alors définir $\widehat{\mathbf{A}}_\alpha$ par

$$\begin{aligned}
\widehat{\mathbf{A}}_\alpha &= \mathbf{L}_s^{-1} \widetilde{\mathbf{A}}_\alpha \mathbf{L}_s^{-T} \\
&= \mathbf{Q}^T \mathbf{L}^{-1} \mathbf{Q} \widetilde{\mathbf{A}}_\alpha \mathbf{Q}^T \mathbf{L}^{-T} \mathbf{Q} \\
&= \mathbf{Q}^T \mathbf{L}^{-1} \widetilde{\mathbf{A}}_i R_{\alpha i} \mathbf{L}^{-T} \mathbf{Q} \\
&= \mathbf{Q}^T \widehat{\mathbf{A}}_i \mathbf{Q} R_{\alpha i} \,.
\end{aligned}$$

On en déduit une écriture de $\widehat{\boldsymbol{\tau}}$ en fonction de $\widehat{\mathbf{A}}_\alpha$:

$$\begin{aligned}
\widehat{\boldsymbol{\tau}} &= \left(\frac{\partial \xi_i}{\partial x_j} \frac{\partial \xi_i}{\partial x_k} \widehat{\mathbf{A}}_j \widehat{\mathbf{A}}_k \right)^{-1/2} \\
&= \left(\frac{\partial \xi_i}{\partial x_j} \frac{\partial x_j}{\partial s_\alpha} \frac{\partial \xi_i}{\partial x_k} \frac{\partial x_k}{\partial s_\beta} \mathbf{Q} \widehat{\mathbf{A}}_\alpha \widehat{\mathbf{A}}_\beta \mathbf{Q}^T \right)^{-1/2} \\
&= \mathbf{Q} \left(\frac{\partial \xi_i}{\partial s_\alpha} \frac{\partial \xi_i}{\partial s_\beta} \widehat{\mathbf{A}}_\alpha \widehat{\mathbf{A}}_\beta \right)^{-1/2} \mathbf{Q}^T \,.
\end{aligned}$$

Soit $\widehat{\mathbf{S}}$ est la matrice de vecteurs propres de $\widehat{\mathbf{A}}_s$ et $\boldsymbol{\Lambda}_s$ la matrice des valeurs propres associées.

On pose $\boldsymbol{\Lambda}_\alpha = \widehat{\mathbf{S}}^T \widehat{\mathbf{A}}_\alpha \widehat{\mathbf{S}}$. Les matrices $\boldsymbol{\Lambda}_\alpha$ ne sont pas diagonales si $\alpha \neq s$. Elles sont symétriques par définition.

On peut réécrire $\widehat{\boldsymbol{\tau}}$ comme suit :

$$\widehat{\boldsymbol{\tau}} = \mathbf{Q} \widehat{\mathbf{S}} \left(\frac{\partial \xi_i}{\partial s_\alpha} \frac{\partial \xi_i}{\partial s_\beta} \boldsymbol{\Lambda}_\alpha \boldsymbol{\Lambda}_\beta \right)^{-1/2} \widehat{\mathbf{S}}^T \mathbf{Q}^T \,.$$

Soit $\mathbf{X}$ la matrice des vecteurs propres de $\frac{\partial \xi_i}{\partial s_\alpha} \frac{\partial \xi_i}{\partial s_\beta} \boldsymbol{\Lambda}_\alpha \boldsymbol{\Lambda}_\beta$ et $\boldsymbol{\Lambda}$ les valeurs propres associées. La matrice $\mathbf{X}$ est orthonormale.

On obtient une écriture e $\widehat{\boldsymbol{\tau}}$ faisant apparaître ses valeurs propres :

$$\widehat{\boldsymbol{\tau}} = \mathbf{Q} \widehat{\mathbf{S}} \mathbf{X} \boldsymbol{\Lambda}^{-1/2} \mathbf{X}^T \widehat{\mathbf{S}}^T \mathbf{Q}^T \,.$$

D'où

$$\boldsymbol{\tau} = \mathbf{L}^{-T} \mathbf{Q} \widehat{\mathbf{S}} \mathbf{X} \boldsymbol{\Lambda}^{-1/2} \mathbf{X}^T \widehat{\mathbf{S}}^T \mathbf{Q}^T \mathbf{L}^{-1} \,.$$

La matrice $\widetilde{\mathbf{S}}$ solution du problème aux valeurs propres généralisé $\widetilde{\mathbf{A}}_s \widetilde{\mathbf{S}} = \widetilde{\mathbf{A}}_0^s \widetilde{\mathbf{S}} \boldsymbol{\Lambda}_s$ est reliée à $\widehat{\mathbf{S}}$ qui diagonalise $\widehat{\mathbf{A}}_s$ par la relation

$$\widetilde{\mathbf{S}} = \mathbf{L}_s^{-T} \widehat{\mathbf{S}} \,.$$

En effet,

$$\begin{aligned}\widetilde{\mathbf{A}}_s\widetilde{\mathbf{S}} &= \widetilde{\mathbf{A}}_0^s\widetilde{\mathbf{S}}\mathbf{\Lambda}_s \\ &= \mathbf{L}_s\mathbf{L}_s^T\widetilde{\mathbf{S}}\mathbf{\Lambda}_s\,.\end{aligned}$$

En posant $\widehat{\mathbf{S}} = \mathbf{L}_s^T\widetilde{\mathbf{S}}$ et en multipliant à gauche la relation précédente par $\mathbf{L}_s^{-1}$, on trouve bien que $\widehat{\mathbf{S}}$ diagonalise $\widehat{\mathbf{A}}_s$. Cela permet également de justifier la définition des $\mathbf{\Lambda}_\alpha$ donnée avant l'écriture du théorème.

Pour revenir à $\boldsymbol{\tau}$, comme $\mathbf{L}_s^T = \mathbf{Q}^T\mathbf{L}^T\mathbf{Q}$, on en déduit que

$$\begin{aligned}\mathbf{L}^{-T}\mathbf{Q}\widehat{\mathbf{S}} &= \mathbf{L}^{-T}\mathbf{Q}\mathbf{L}_s^T\widetilde{\mathbf{S}} \\ &= \mathbf{L}^{-T}\mathbf{Q}\mathbf{Q}^T\mathbf{L}^T\mathbf{Q}\widetilde{\mathbf{S}} \\ &= \mathbf{Q}\widetilde{\mathbf{S}}\,.\end{aligned}$$

On retrouve bien

$$\boldsymbol{\tau} = \left(\mathbf{Q}\widetilde{\mathbf{S}}\mathbf{X}\right)\mathbf{\Lambda}^{-\frac{1}{2}}\left(\mathbf{Q}\widetilde{\mathbf{S}}\mathbf{X}\right)^T\,. \tag{5.46}$$

□

Remarque 5.2. Dans l'expression (5.45), les matrices $\widetilde{\mathbf{S}}$ et $\mathbf{Q}$ se calculent analytiquement, puisque $\mathbf{Q}$ est la rotation permettant le changement de repère (dépendant de l'écoulement local) et $\widetilde{\mathbf{S}}$ correspond à la matrice de passage de la base des variables caractéristiques à celle des variables entropiques. Seuls $\mathbf{X}$ et $\mathbf{\Lambda}$ nécessitent des calculs supplémentaires.

Remarque 5.3. Pour simplifier le calcul de $\mathbf{X}$ et $\mathbf{\Lambda}$, une des hypothèses prises dans le code AeTher est de ne pas considérer les termes croisés $\mathbf{\Lambda}_\alpha\mathbf{\Lambda}_\beta$, où $\alpha \neq \beta$. La matrice à diagonaliser est alors de la forme $\frac{\partial\xi_i}{\partial s_\alpha}^2\mathbf{\Lambda}_\alpha^2$ qui est diagonale avec quatre termes extra-diagonaux. Quelques itérations de Jacobi [63] servent à approcher avec suffisamment de précision $\mathbf{\Lambda}$ et $\mathbf{x}$.

Pour définir une correction visqueuse, on utilise la projection des matrices $\widetilde{\mathbf{K}}_{ij}$ sur les modes de $\boldsymbol{\tau}$. Cela donne des σ_i qui sont de la dimension d'une viscosité. On pose $\mathbf{\Psi} = \mathbf{Q}\widetilde{\mathbf{S}}\mathbf{X}$. Ses colonnes sont notées $\mathbf{\Psi}_i$. Les σ_i sont définis par

$$\sigma_i = \mathbf{\Psi}_i^T\sum_{j,k=1}^{3}\left(\frac{\partial\xi_j}{\partial x_l}\frac{\partial\xi_k}{\partial x_m}\widetilde{\mathbf{K}}_{lm}\right)\mathbf{\Psi}_i\,. \tag{5.47}$$

Le nombre de Péclet local α_i pour chaque mode est

$$\alpha_i = \frac{\sqrt{\lambda_i}}{\sigma_i}\,, \tag{5.48}$$

car les valeurs propres de $\boldsymbol{\tau}$ sont $\lambda_i^{-1/2}$. On applique la correction visqueuse $\widetilde{\xi}(\alpha_i)$ aux valeurs propres de $\boldsymbol{\tau}$ pour obtenir la matrice de stabilisation des équations de Navier-Stokes :

$$\boldsymbol{\tau}_{\mathrm{NS}} = \left(\mathbf{Q}\widetilde{\mathbf{S}}\mathbf{X}\right) \boldsymbol{\Lambda}^{-\frac{1}{2}} \operatorname{diag}\left(\widetilde{\xi}(\alpha_i)\right) \left(\mathbf{Q}\widetilde{\mathbf{S}}\mathbf{X}\right)^T \tag{5.49}$$

5.2 Nouvelle matrice de stabilisation

Cette thèse a été l'occasion d'explorer une matrice de stabilisation plus complète que celle actuellement utilisée dans AeTher.

5.2.1 Calcul de la matrice de stabilisation complète

Dans la partie 5.1.5, la méthode de calcul de la matrice $\boldsymbol{\tau}$ a été détaillée. Des approximations sont faites pour calculer le terme $\left(\frac{\partial \xi_i}{\partial s_\alpha}\frac{\partial \xi_i}{\partial s_\beta}\boldsymbol{\Lambda}_\alpha\boldsymbol{\Lambda}_\beta\right)^{-1/2}$ par une méthode spectrale. Les termes croisés $\boldsymbol{\Lambda}_\alpha\boldsymbol{\Lambda}_\beta$ pour $\alpha \neq \beta$ sont notamment négligés.

L'idée retenue a été d'implémenter l'intégralité des termes de $\boldsymbol{\tau}$ dans la formule (5.2) proposée par Mallet [104]. Deux méthodes de calcul sont proposées, l'une effectuant une diagonalisation directe de la somme des carrés des opérateurs de convection pour obtenir directement $\boldsymbol{\tau}$, l'autre passant par le repère lié au lignes de courant, introduit à la section 5.1.5, pour faciliter numériquement la diagonalisation

Construction directe

La première idée pour construire une matrice $\boldsymbol{\tau}$ complète a été d'appliquer la formule (5.2) à la lettre, c'est-à-dire de calculer la racine carré de $\frac{\partial \xi_i}{\partial x_j}\frac{\partial \xi_i}{\partial x_k}\mathbf{A}_j\mathbf{A}_k$ par sa diagonalisation, puis de multiplier à gauche par $\widetilde{\mathbf{A}}_0^{-1}$. Cette méthode n'est pas astucieuse, car elle ne profite pas des propriétés mathématiques des matrices $\widetilde{\mathbf{A}}_i$ et $\widetilde{\mathbf{A}}_0$. Le terme $\frac{\partial \xi_i}{\partial x_j}\frac{\partial \xi_i}{\partial x_k}\mathbf{A}_j\mathbf{A}_k$ n'est pas symétrique, car les matrices $\mathbf{A}_j$ ne le sont pas, ce qui complique sa diagonalisation [63].

Lemme 5.12. *Soit* $\mathbf{M} = \frac{\partial \xi_i}{\partial x_j}\frac{\partial \xi_i}{\partial x_k}\mathbf{A}_j^T\widetilde{\mathbf{A}}_0^{-1}\mathbf{A}_k$*, symétrique par définition. On diagonalise numériquement* $\mathbf{L}^T\mathbf{M}\mathbf{L}$ *en* $\mathbf{L}^T\mathbf{M}\mathbf{L} = \widehat{\mathbf{T}}\boldsymbol{\Lambda}\widehat{\mathbf{T}}^T$*. Si l'on note* $\mathbf{T} = \mathbf{L}^{-T}\widehat{\mathbf{T}}$*,*

$$\boldsymbol{\tau} = \mathbf{T}\boldsymbol{\Lambda}^{-1/2}\mathbf{T}^T\,. \tag{5.50}$$

Démonstration. On rappelle que $\mathbf{L}$ est le facteur de décomposition de Cholesky de $\widetilde{\mathbf{A}}_0$ en $\widetilde{\mathbf{A}}_0 = \mathbf{L}\mathbf{L}^T$, et que $\widehat{\mathbf{B}}_i = \frac{\partial \xi_i}{\partial x_j}\mathbf{L}^{-1}\widetilde{\mathbf{A}}_j\mathbf{L}^{-T}$.

La matrice de stabilisation $\widehat{\boldsymbol{\tau}}$ introduite dans la section 5.1.3 pour le problème définit sur l'élément de référence est

$$\begin{aligned}
\widehat{\boldsymbol{\tau}} &= \left(\widehat{\mathbf{B}}_i\widehat{\mathbf{B}}_i\right)^{-1/2} \\
&= \left(\mathbf{L}^{-1}\widetilde{\mathbf{B}}_i^T\widetilde{\mathbf{A}}_0^{-1}\widetilde{\mathbf{B}}_i\mathbf{L}^{-T}\right)^{-1/2} \\
&= \left(\frac{\partial\xi_i}{\partial x_j}\frac{\partial\xi_i}{\partial x_k}\mathbf{L}^{-1}\widetilde{\mathbf{A}}_j\widetilde{\mathbf{A}}_0^{-1}\widetilde{\mathbf{A}}_k\mathbf{L}^{-T}\right)^{-1/2} \\
&= \left(\frac{\partial\xi_i}{\partial x_j}\frac{\partial\xi_i}{\partial x_k}\mathbf{L}^{T}\mathbf{A}_j^{T}\widetilde{\mathbf{A}}_0^{-1}\mathbf{A}_k\mathbf{L}\right)^{-1/2} \qquad (5.51)
\end{aligned}$$

On constate que $\widehat{\boldsymbol{\tau}} = \left(\mathbf{L}^T\mathbf{M}\mathbf{L}\right)^{-1/2}$.
Enfin,

$$\boldsymbol{\tau} = \mathbf{L}^{-T}\widehat{\boldsymbol{\tau}}\mathbf{L}^{-1} = \mathbf{L}^{-T}\widehat{\mathbf{T}}\boldsymbol{\Lambda}^{-1/2}\widehat{\mathbf{T}}\mathbf{L}^{-1} = \mathbf{T}\boldsymbol{\Lambda}^{-1/2}\mathbf{T}^T$$

□

Cette méthode a été testée numériquement, en utilisant la bibliothèque LAPACK [11] pour calculer la factorisation de Cholesky de $\widetilde{\mathbf{A}}_0 = \mathbf{L}\mathbf{L}^T$, et pour la diagonalisation de la matrice symétrique $\mathbf{L}^T\mathbf{M}\mathbf{L}$.

Cette méthode n'a pas donnée entière satisfaction. En effet, malgré le caractère symétrique et positif de $\mathbf{L}^T\mathbf{M}\mathbf{L} = \widehat{\mathbf{B}}_i\widehat{\mathbf{B}}_i$, la diagonalisation de cette matrice par la routine DSYEVR prévue pour les matrices symétriques donnait sur certains éléments des valeurs propres négatives. Ces matrices ont été extraites, et la diagonalisation de celles-ci par d'autres logiciel comme Matlab donnait des valeurs propres différentes, surtout pour les plus petites et les plus grandes. Ce problème de conditionnement numérique a été résolu en changeant d'approche pour le calcul de $\boldsymbol{\tau}$.

Construction dans le repère lié à l'écoulement

La deuxième méthode de construction de la matrice $\boldsymbol{\tau}$ complète a été de suivre est de suivre exactement la formule (5.45) du théorème 5.11 sans approximations (comme celle précisée dans la remarque 5.3).

On utilise les opérateurs de flux d'Euler dans la repère lié aux lignes de courant. On diagonalise l'opérateur de flux exprimé dans le sens du courant à l'aide des variables caractéristiques, qui sont connues analytiquement, pour obtenir les $\boldsymbol{\Lambda}_\alpha$. Tous les termes, y compris les termes croisés, sont considérés dans $\frac{\partial\xi_i}{\partial s_\alpha}\frac{\partial\xi_i}{\partial s_\beta}\boldsymbol{\Lambda}_\alpha\boldsymbol{\Lambda}_\beta$. Cette matrice est diagonalisée numériquement par la routine DSYEVR de LAPACK.

Cette approche été retenue, plutôt que la précédente, car elle est plus stable numériquement. La diagonalisation numérique n'a donnée aucune valeur propre négative.

Cette nouvelle matrice de stabilisation, que l'on qualifiera de complète, est beaucoup plus coûteuse à calculer que l'ancienne matrice . Pour cette dernière, quelques itérations de Jacobi sont nécessaire pour approcher la diagonalisation d'une matrice quasi diagonale. La matrice complète nécessite de diagonaliser une matrice 5×5 pour tous les éléments du maillage.

Ce surcoût est négligeable pour des calculs linéarisé, vu que la construction de la matrice prend peu de temps par rapport à la résolution du système linéaire. Par contre, la résolution des équations de Navier-Stokes non-linéaires demande à chaque pas de temps la construction d'un système linéaire implicite. Dans ce cas, le surcoût de la matrice complète n'est plus négligeable.

5.2.2 Navier-Stokes non-linéaire

Dans cette partie, on a appliqué la matrice de stabilisation complète à des calculs non-linéaires. Il est plus aisé pour un aérodynamicien d'interpréter des solutions des équations de Navier-Stokes plutôt que de leur forme linéarisée. Enfin, pouvoir se servir de cette forme de $\boldsymbol{\tau}$ pour le calcul du champ moyen permet d'être consistant avec le calcul linéarisé si la même stabilisation y est utilisée.

Le problème non-linéaire est résolu à l'aide d'une méthode d'avance en temps fictif, et à chaque pseudo pas de temps, la variation de la solution est calculée d'une façon implicite par la résolution d'un système linéaire. Comme pour AeTher linéarisé, ce système est résolu par une méthode GMRES, avec cependant une précision demandée beaucoup plus faible. Pour plus de précisions, on consultera [145, 146].

La matrice complète de stabilisation a été testée sur deux cas transsoniques, et un cas d'approche basse vitesse.

Aile M6 transsonique

Cas test IV (Aile M6) :
L'aile M6 est une maquette de soufflerie développée par l'ONERA dont les résultats expérimentaux [141] servent de référence pour les codes de mécanique des fluides [88]. Le maillage utilisé est très fin et permet de résoudre la couche limite avec précision. Il contient plus de 700 000 nœuds, soit plus de 3,5 millions d'inconnues. Les conditions d'essais sont une incidence de 3° et une vitesse de Mach 0,84. L'écoulement est donc transsonique.

Une coupe de pression à la peau est présentée sur la figure 5.5. On remarque que les chocs sont un peu plus marqués avec le tau complet, et ailleurs les courbes sont quasiment identiques. La convergence du résidu non-linéaire en fonction des itérations pseudo-temporelles est donnée sur la figure 5.6. Le calcul utilisant le tau complet demande plus d'itérations en pseudo-temps pour converger et atteindre un résidu limite. Cependant, le nombre d'itérations GMRES nécessaire à la résolution des problèmes

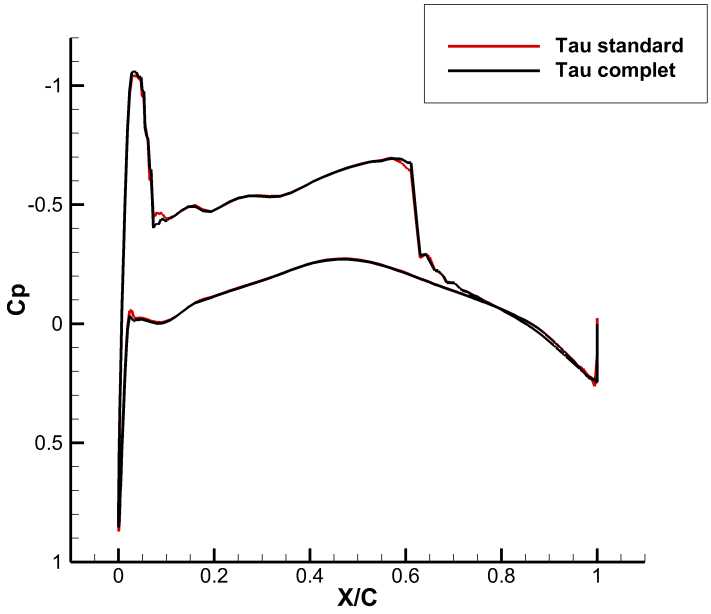


FIGURE 5.5 – Coupe du coefficient de pression sur l'aile M6 (cas test IV) pour des calculs utilisant le tau standard et le tau complet.

implicites linéaires est plus faible pour le calcul utilisant la matrice de stabilisation complète. Cela est illustrée sur la figure 5.7, qui en abscisse indique le nombre cumulé d'itérations GMRES ayant servi à résoudre les problèmes implicites. La figure 5.8 présente ce même résultat en indiquant le nombre d'itérations nécessaires à l'algorithme GMRES pour résoudre les problèmes implicites à chaque pas de temps. Le système stabilisé avec la matrice $\boldsymbol{\tau}$ complète demande un nombre d'itérations en général inférieur à chaque résolution, et est en tout cas moins impacté par les changements de CFL. Cela a permis d'utiliser une stratégie de montée plus agressive du CFL afin que la convergence en pas de pseudo-temps soit aussi rapide que le calcul utilisant la matrice de stabilisation standard.

Falcon croisière

La matrice de stabilisation a été testée sur un cas industriel, qui est un Falcon en croisière à Mach 0,8.

Cas test V (Falcon croisière) :
Le cas Falcon croisère représente un avion Falcon en croisière à Mach 0,8. Le maillage de discrétisation contient 7 millions de nœuds, soit plus de 35 millions d'inconnues. Il est découpé en 512 sous-domaines. Ce cas désigne un calcul non-linéaire

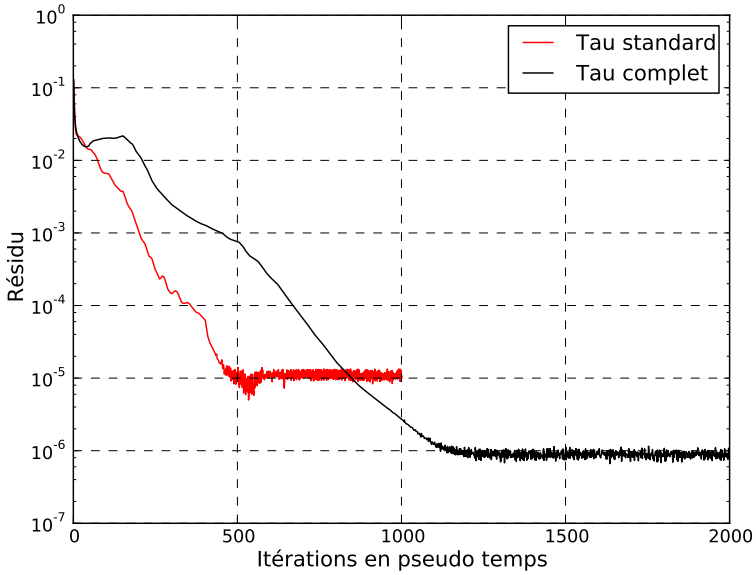


FIGURE 5.6 – Convergence pour le cas aile M6 (cas test IV) du résidu non-linéaire en fonction des itérations pseudo-temporelles.

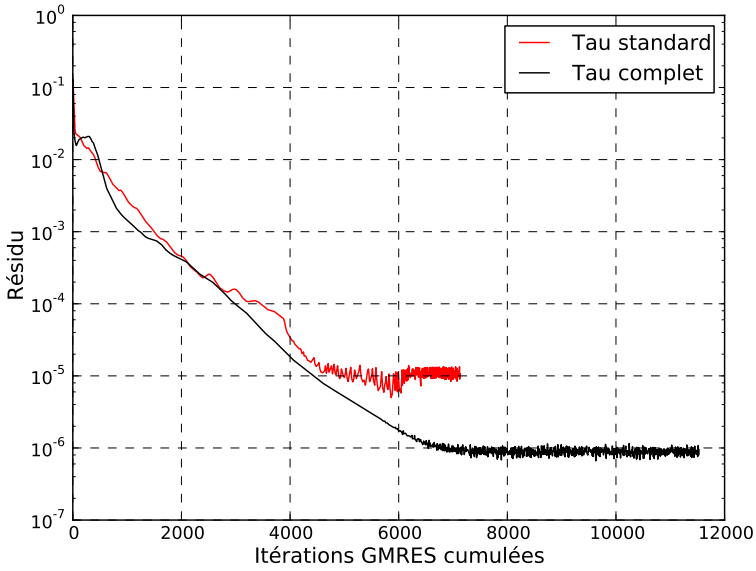


FIGURE 5.7 – Convergence pour le cas aile M6 (cas test IV) du résidu non-linéaire en fonction du nombre cumulé d'itérations GMRES

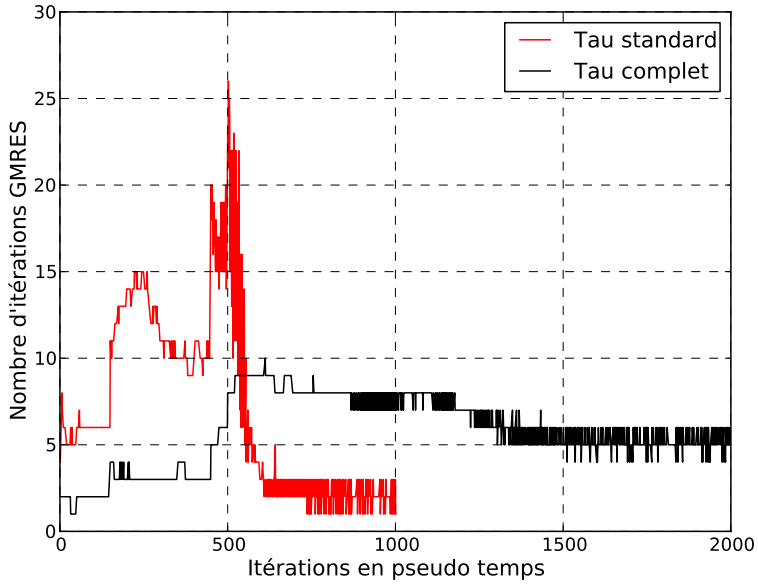


FIGURE 5.8 – Nombre d'itérations GMRES nécessaires pour résoudre le problème implicite à chaque pas de temps pour le cas aile M6 (cas test IV).

La convergence de ce cas avec la matrice $\boldsymbol{\tau}$ complète a été délicate, puisqu'il a fallu redémarrer le calcul à partir d'une solution convergée, et faire attention à la montée en CFL, sous peine de divergence du calcul. Aucune comparaison des courbes de convergence n'est donc possible.

Une coupe de pression sur l'aile montrée sur la figure 5.9 montre un choc toujours mieux défini, mais des oscillations en aval de celui-ci. Pour des raisons de confidentialités, les valeurs de l'axe des ordonnées ne sont pas données.

La matrice de stabilisation complète est donc sous dissipative, ce qui lui permet de mieux capturer les chocs, mais rend le schéma légèrement plus instable, expliquant les oscillations numériques. La diffusion numérique trop faible augmente la difficulté de convergence, d'où l'obligation de redémarrer d'une solution convergée.

Falcon décollage

La matrice complète de stabilisation a été testée sur un cas basse vitesse, où elle a pu démontrer son intérêt. Un calcul sur un avion de type Falcon en configuration de décollage à 10° d'incidence a été réalisé.

Cas test VI (Falcon décollage non-linéaire) :
Le maillage et la configuration aérodynamique ont été présentés dans le cas test III à la section 4.5.2. Cette fois-ci, le calcul réalisé est non-linéaire.

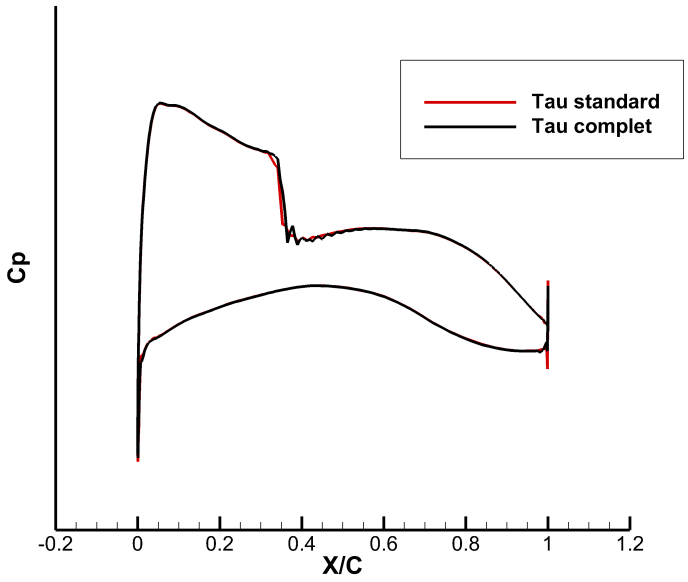


FIGURE 5.9 – Coupe de pression sur l'aile. Cas Falcon croisière (cas test V).

La convergence du résidu non-linéaire en fonction des itérations en pseudo temps se trouve sur la figure 5.10. Cette fois-ci les deux calculs convergent de manière très similaire. La figure 5.11 montre que l'utilisation de la matrice complète de stabilisation permet de réduire le nombre d'itérations GMRES pour résoudre les problèmes implicites.

L'intérêt de la matrice $\boldsymbol{\tau}$ complète s'est révélé lors de la comparaison des cartographies de pression sur le fuselage de l'avion. Le code AeTher est très sensible à la qualité du maillage sur le fuselage, qui est en général bien plus grossier que sur l'aile, où les phénomènes aérodynamiques les plus importants se passent. Le maillage est non seulement moins fin que sur l'aile, mais il est également non structuré. La cartographie de pression présente un mouchetage inexpliqué sur le fuselage dans les calculs basse vitesse. Il s'est avéré que l'utilisation de la matrice complète de stabilisation a considérablement réduit ces oscillations parasites de pression, et a amélioré grandement la qualité de la solution. La figure 5.12 montre que le mouchetage disparaît presque intégralement dans le sillage de l'aile sous les réacteurs, ainsi que sur le nez de l'avion. Le mouchetage a été identifié comme provenant d'une variation de l'écart de la normale des éléments surfaciques à la normales de la CAO. Un maillage suffisamment régulier permet d'éliminer les bosses qui sont à l'origine de ces variations locales de normale. Le mouchetage est un signe que le code stabilisée avec la matrice $\boldsymbol{\tau}$ standard est très sensible au variation

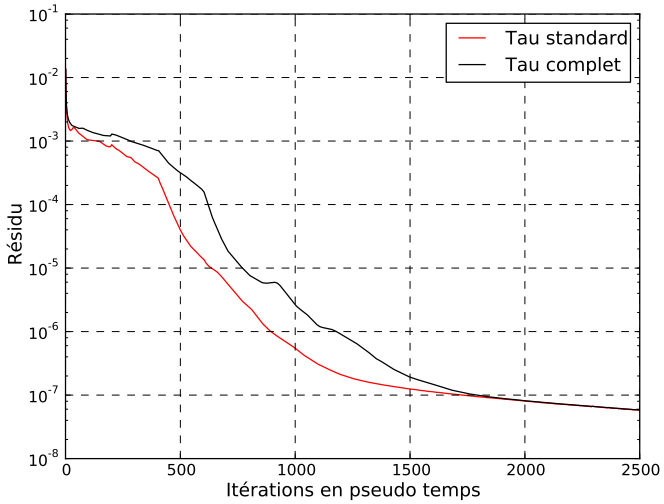


FIGURE 5.10 – Convergence du résidu non linéaire en fonction des itérations en pseudo temps. Cas Falcon décollage (cas test VI).

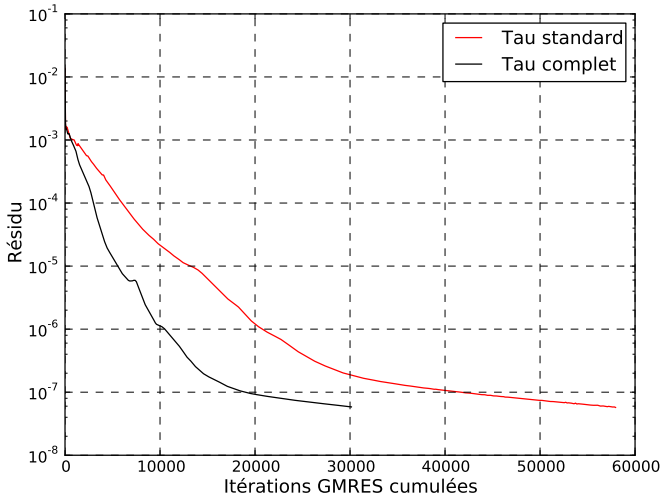


FIGURE 5.11 – Convergence du résidu non linéaire en fonction des itérations GMRES cumulées. Cas Falcon décollage (cas test VI).

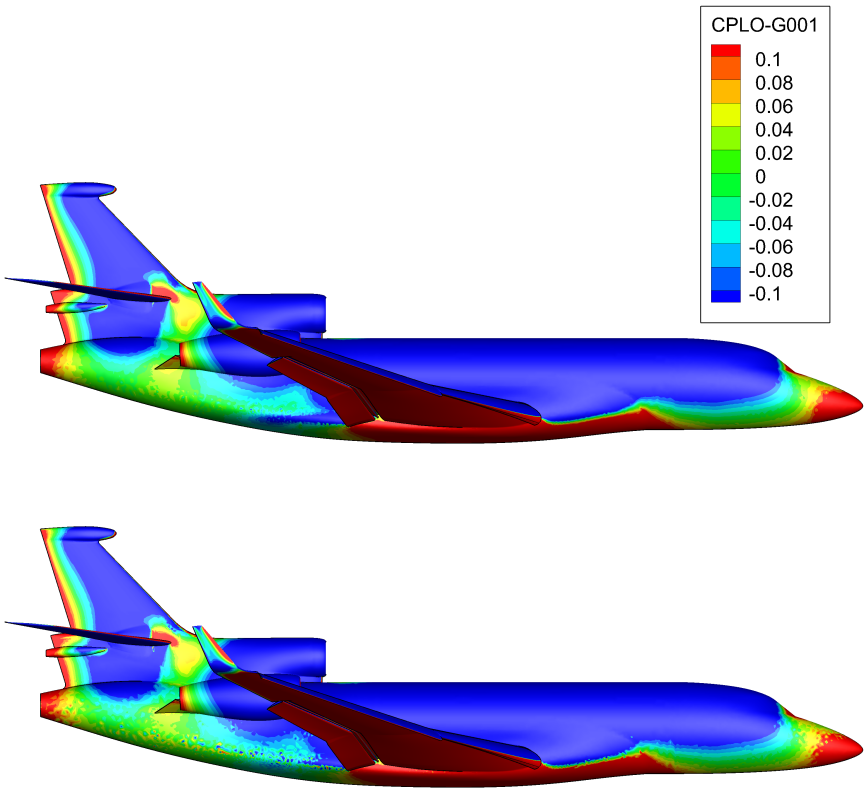


FIGURE 5.12 – Champ de pression locale sur le fuselage. En haut, $\boldsymbol{\tau}$ complet. En bas, $\boldsymbol{\tau}$ standard. Cas Falcon décollage (cas test VI).

des normales des facettes du maillage surfacique, ce qui peut être intéressant dans certains cas.

En conclusion de cette étude de l'utilisation de la matrice de stabilisation complète pour les équations de Navier-Stokes non-linéaires, on a vu que cette stabilisation était un peu moins diffusive que la matrice $\boldsymbol{\tau}$ standard. Cela permet de capturer mieux les chocs dans les écoulements transsoniques, au risque d'introduire des oscillations numériques en aval des chocs. Ce manque de viscosité complique également la convergence pour les cas haute vitesse. Les systèmes linéaires des problèmes implicites sont résolus plus facilement, ce qui n'est pas forcément cohérent avec une viscosité numérique plus faible.

5.2.3 Résultats en aéroélasticité sur la maquette DTP

Les deux matrices de stabilisation ont été comparées sur un cas d'aéroélasticité. La maquette DTP, présentée à la section 3.3.4, a fourni ce cas test, appelé cas test II. Pour rappel, lepoint de vol est de Mach 0,88 à 0° d'incidence. Le mouvement choisi est un tangage de voilure à fréquence nulle.

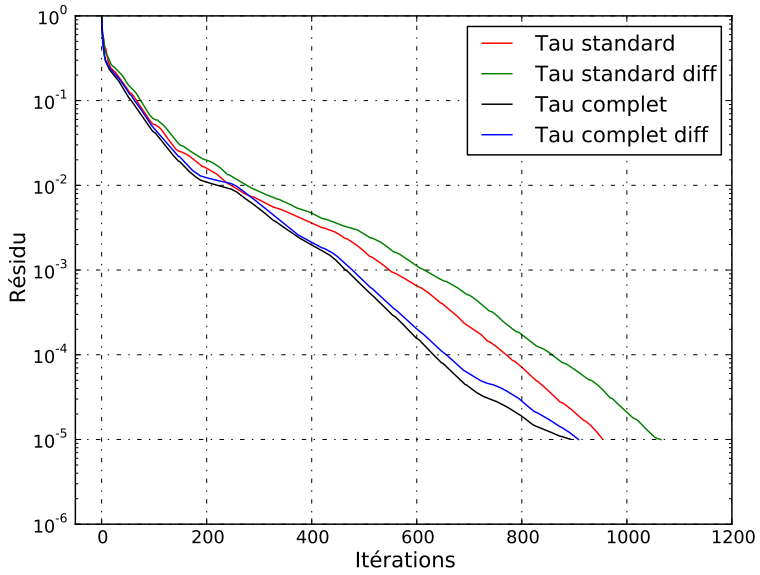


FIGURE 5.13 – Coubre de convergence du cas DTP (cas test II). Résidu normalisé.

Le terme de stabilisation SUPG intervient dans le résidu non-linéaire. Ainsi, une différenciation de ce terme entre en compte dans la formulation linéarisée, à la fois dans la matrice et le second membre du système linéaire. Comme expliqué dans la partie 4.5.2, la différenciation du terme de stabilisation est nécessaire afin d'obtenir les bons gradients de fonction coût. En aéroélasticité, l'utilisation de ce terme est plus discuté, et le choix est donné à l'utilisateur de ne pas considérer les gradients de la stabilisation pour la formation du système linéaire.

Le terme de stabilisation différenciée nécessite d'obtenir le gradient de la matrice $\boldsymbol{\tau}$. La routine de construction de cette matrice a donc été différenciée automatiquement avec le logiciel Tapenade [76]. Cela nécessite notamment d'obtenir le gradient de la diagonalisation du terme $\frac{\partial \xi_i}{\partial s_\alpha}\frac{\partial \xi_i}{\partial s_\beta}\boldsymbol{\Lambda}_\alpha\boldsymbol{\Lambda}_\beta$. Pour ce faire, la routine DSYEVR de LAPACK a été différenciée à l'aide de Tapenade. Cette routine différenciée calcule la diagonalisation d'une matrice symétrique, ainsi que la variation de ses valeurs et vecteurs propres dans la direction d'une perturbation symétrique de cette matrice. Cette routine ignore complètement les problèmes de différenciabilité de valeurs propres multiples.

La figure 5.13 montre la convergence du résidu normalisé par l'algorithme GMRES, pour des calculs utilisant une matrice de stabilisation standard ou complète, et différenciée ou non. On constate que le système avec stabilisation complète converge plus rapidement qu'avec la matrice standard. La prise en compte avec la matrice standard du terme de stabilisation différencié rend la convergence légèrement plus difficile, comme constaté auparavant

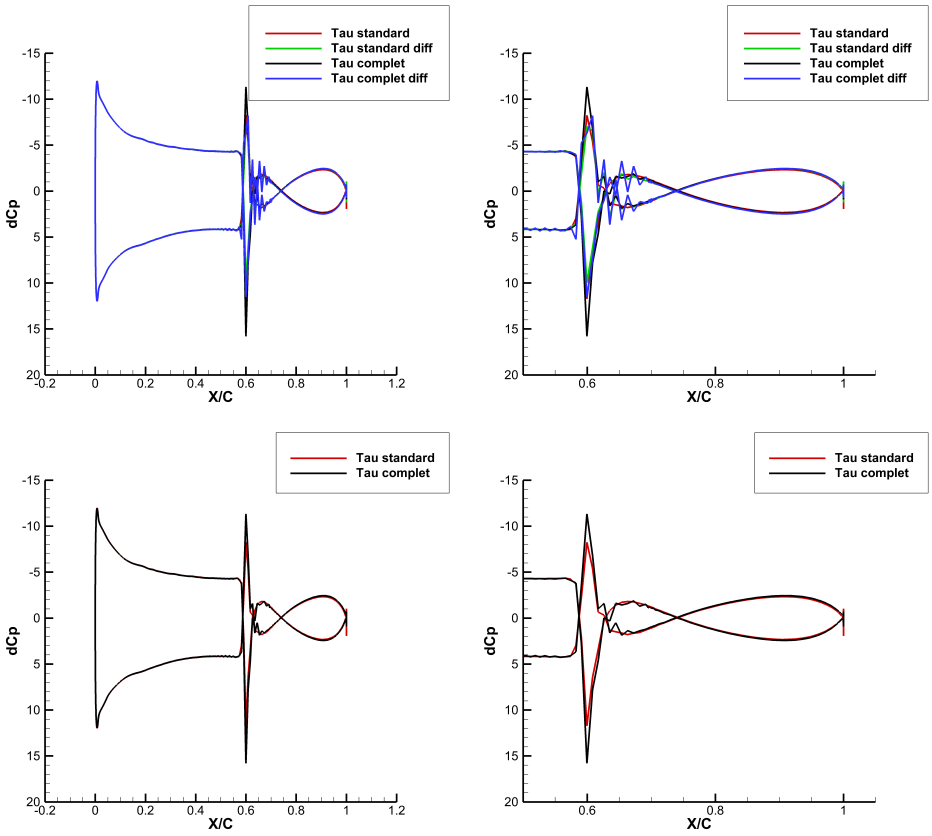


FIGURE 5.14 – Variation de la pression sur une coupe de l'aile. Cas DTP (cas test II). Ligne du bas : calculs sans $\boldsymbol{\tau}$ différencié. Ligne du haut : calculs avec $\boldsymbol{\tau}$ différencié. Colonne de gauche : coupe sur toute la corde de l'aile. Colonne de droite : zoom sur la partie antérieure de l'aile.

sur d'autres cas tests. Pour la matrice $\boldsymbol{\tau}$ complète, la convergence ne change presque pas, ce qui est inhabituel.

Des coupes de pression à la surface de l'aile sont présentées sur la figure 5.14. La ligne du bas présente les calculs sans différenciation de la stabilisation. On constate encore que le $\boldsymbol{\tau}$ complet est moins visqueux et introduit des instabilités numériques en aval du choc. La ligne du haut présente les mêmes résultats, et ajoute les calculs avec différenciation de la stabilisation. Pour la matrice $\boldsymbol{\tau}$ standard, la différenciation ajoute un petit peu d'instabilité derrière le choc. Par contre, le calcul avec différenciation de la matrice complète donne des résultats très bruités, particulièrement en aval mais également en amont du choc.

La matrice complète n'améliore pas la vitesse de convergence, et augmente l'instabilité de la solution derrière les chocs. Son utilisation en linéarisé sur des applications transsoniques ne semble pas intéressante.

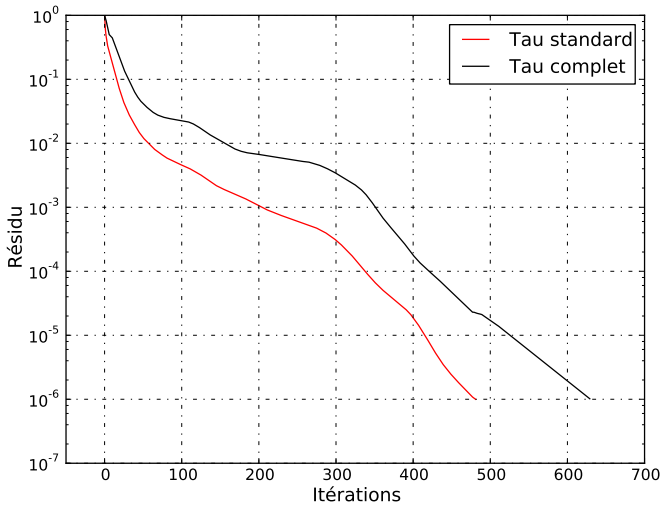


FIGURE 5.15 – Convergence du résidu normalisé par l'algorithme GMRES. Cas Falcon décollage linéarisé (cas test VII).

5.2.4 Résultats en linéarisé basse vitesse

La section 5.2.2 a montré l'intérêt de la matrice de stabilisation dans les configurations à basse vitesse. On reprend ce cas test en effectuant cette fois-ci un calcul linéarisé.

Cas test VII (Falcon décollage linéarisé) :
Le maillage et la configuration aérodynamique sont identiques à ceux du cas test VI. On utilise un mouvement de tangage à fréquence nulle pour créer le second membre. Le calcul non-linéaire de référence autour duquel les calculs linéarisés ont été réalisés utilise la matrice standard de stabilisation.

Sur ce cas, la prise en compte de la différenciation de la stabilisation, quelque soit la forme choisie de la matrice, a empêché la convergence. La figure 5.15 présente la convergence des systèmes linéaires résolu par l'algorithme GMRES, utilisant la matrice $\boldsymbol{\tau}$ standard ou complète. On constate cette fois-ci que la matrice complète ralentit légèrement la convergence. Une cartographie de la variation de pression sur le fuselage de l'avion est montrée sur la figure 5.16. On constate un mouchetage moins important sur le nez de l'avion, ainsi que sur le fuselage près de l'entrée d'air central, lorsque la forme complète de la matrice de stabilisation est utilisée.

5.2.5 Résultats en aéroacoustique

La nouvelle matrice de stabilisation a également été testée en aéroacoustique. Le cas test, nommé SFWA, est celui d'une tuyère modèle présentée

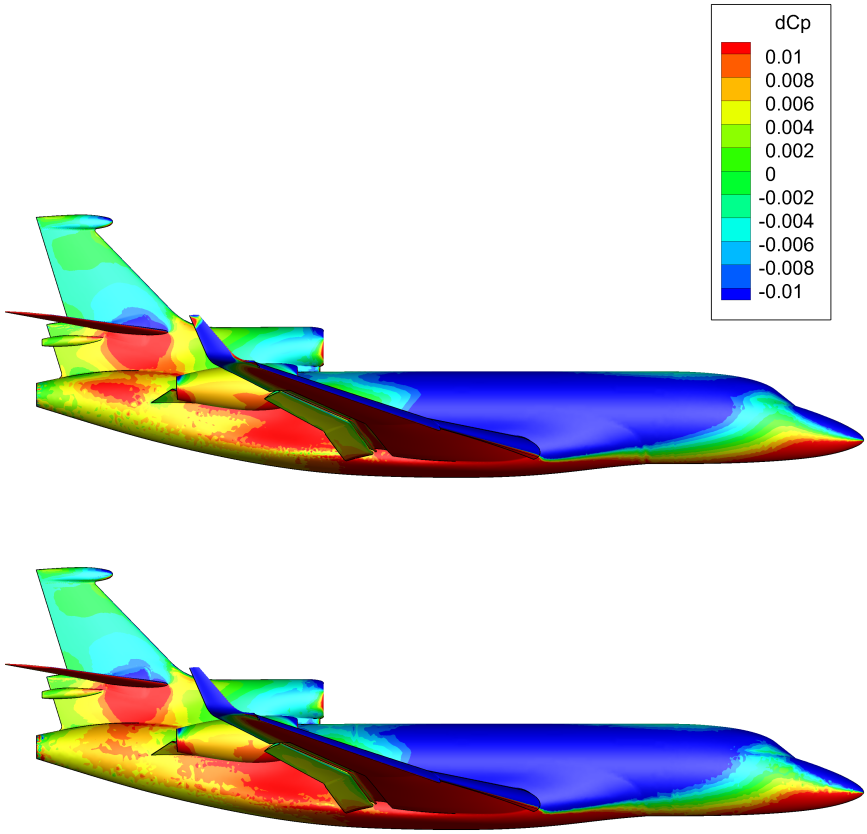


FIGURE 5.16 – Cartographie de la variation de pression sur le fuselage du Falcon en configuration décollage (cas test VII). Matrice complète de stabilisation en haut, et standard en bas.

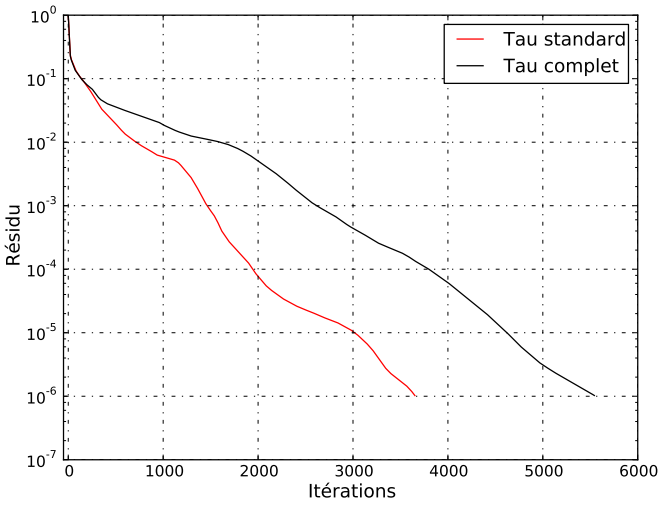


FIGURE 5.17 – Courbe de convergence du résidu normalisé pour cas SFWA.

dans la section 6.3.1, et l'interprétation des résultats fait référence à la section 6.2.

Les courbes de convergence sont présentées sur la figure 5.17. Le calcul utilisant la matrice complète de stabilisation converge beaucoup plus lentement. Cela est très facilement expliqué par des coupe de la variation de pression, données sur la figure 5.18. On constate immédiatement que l'utilisation de la matrice complète induit des rebonds très importants sur le bord. En d'autres termes, elle permet un moins bon décentrement des caractéristiques suivant toutes les directions.

5.3 Conclusion

Dans ce chapitre, on a d'abord détaillé la construction de la matrice $\boldsymbol{\tau}$, qui est au cœur de la stabilisation SUPG. La discétisation centrée d'un terme d'advection conduit à un schéma instable. Une des techniques pour obtenir un schéma stable est de décentrer cette discrétisation. Cela peut se faire par l'ajout d'une viscosité artificielle, qui dépend du nombre de Péclet local dans le cas d'une équation d'advection diffusion. Une façon commode d'implémenter cette viscosité artificielle pour la méthode des éléments finis est de modifier les fonctions de pondérations en y ajoutant un terme défini à l'élément. L'extension aux systèmes d'équations en une dimension ne pose pas de problèmes, en diagonalisant l'opérateur de convection. Par contre, le passage à plusieurs dimensions spatiales nécessite des approximations, puisqu'alors une définition spectrale de la matrice de stabilisation n'est plus possible. De surcroît, le terme de métrique permettant de décentrer la discrétisation des opérateurs de convection dans chaque direction est délicat,

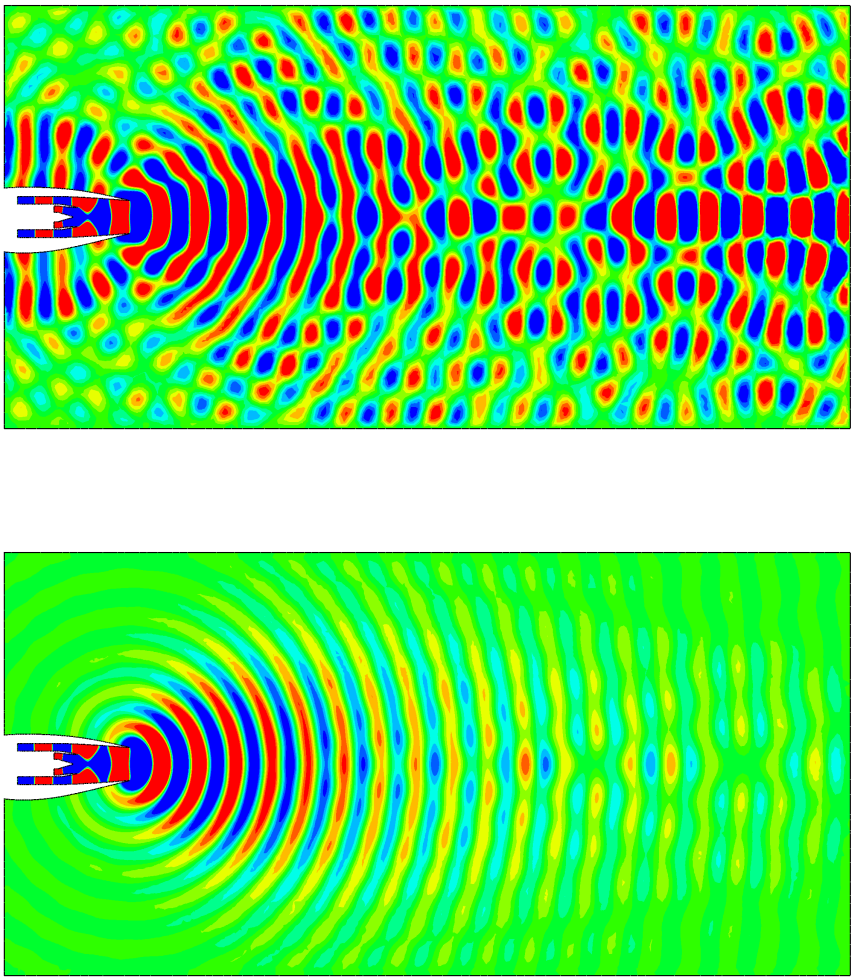


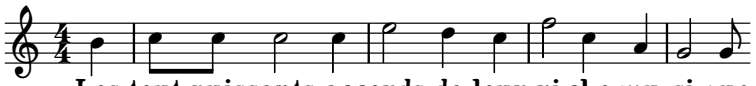
FIGURE 5.18 – Partie réelle de la variation de pression pour un mode plan à 2 kHz sans écoulement sur le cas SFWA. En haut, matrice $\boldsymbol{\tau}$ complète. En bas, matrice standard.

si ce n'est impossible à trouver.

L'approximation retenue par Mallet [104] est de passer de la valeur absolue d'une matrice $\mathbf{B}$ en une dimension à $\sqrt{\mathbf{B}\mathbf{B}^T}$ lorsque $\mathbf{B}$ est une matrice rectangulaire dans le cas à plusieurs dimensions. Cette méthode est retenue dans le code AeTher, mais avec des simplifications supplémentaires. La deuxième section de ce chapitre a été l'occasion de tester cette approximation en suivant littéralement la formule de $\boldsymbol{\tau}$ donnée par Mallet. Le calcul de l'inverse d'une racine carrée d'opérateur est réalisé par une diagonalisation numérique à l'aide de la bibliothèque LAPACK. Afin d'éviter des problèmes de stabilité numérique, les opérateurs de flux sont exprimés dans la base des variables caractéristiques dans un repère lié aux lignes de courant.

Cette forme, que nous avons appelée complète, de la matrice de stabilisation a été testée sur les équations de Navier-Stokes non-linéaires et également linéarisées. Les calculs non-linéaires ont montré que la matrice complète est un peu moins visqueuse et donc moins stabilisatrice. La résolution des problèmes implicites linéaires est également plus facile avec cette matrice. Enfin, le mouchetage apparaissant sur la répartition de pression sur le fuselage disparaît avec la stabilisation complète. L'étude de cette forme de la matrice de stabilisation en linéarisé donne des résultats moins probants. Les conclusions physiques tirées de l'utilisation pour les équations de Navier-Stokes de la matrice complète restent valable : elle est moins stabilisatrice, et tend à lisser le bruit numérique présent sur le fuselage. L'amélioration de convergence observée en non linéaire n'est pas retrouvée. Cela est très certainement dû au terme temporel supplémentaire présent dans le système implicite. Enfin, la matrice complète a été testée en aéroacoustique. On observe que les conditions de Dirichlet homogènes devienne très réfléchissantes, alors qu'elle sont normalement transparentes comme expliqué dans le chapitre 6. Cela montre que la matrice complète de stabilisation ne permet pas un décentrement convenable de la discrétisation des termes advectifs par caractéristique. La transformation d'une condition de Dirichlet homogène en une condition aux limites transparente par la stabilisation SUPG est un bon test de la qualité de la matrice $\boldsymbol{\tau}$.

Toutes ces observations permettent de conclure que l'idée de suivre à la lettre la définition (5.2) n'est pas bonne. Cette formule représente une tentative d'extension approchée en 3D d'une définition exacte en une dimension. Il s'avère que les approximations supplémentaires retenues dans AeTher pour la construction de $\boldsymbol{\tau}$ améliorent la qualité de la stabilisation. Une définition optimale de la matrice de stabilisation se situe certainement entre la formule (5.2) et l'approche choisie dans AeTher.



Duparc, *La vie antérieure*

Chapitre 6

Conditions aux limites

Ce chapitre présente le travail réalisé au cours de cette thèse sur les conditions aux limites. La quasi-intégralité de ces résultats est reprise d'un article soumis lors de la thèse [21]. La première partie de ce chapitre s'intéresse à l'imposition de conditions de Dirichlet non homogènes à des quantités dépendant non linéairement des variables entropiques. La deuxième partie explique une propriété surprenante de type « condition transparente » des conditions de Dirichlet homogènes dans le code AeTher, et la dernière section présente des résultats d'aéroacoustique sur une configuration industrielle.

6.1 Conditions aux limites de Dirichlet non homogènes

Cette section est dévolue à la présentation de l'imposition de conditions aux limites de Dirichlet non homogènes. Trois exemples sont détaillés. Les deux premiers, qui montrent comment imposer une variation de pression, concernent l'aéroacoustique, tandis que le dernier, qui détaille l'imposition d'une vitesse de paroi non nulle, s'applique à l'aéroélasticité.

Les conditions limites de Dirichlet non homogènes sont compliquées par l'utilisation des variables entropiques détaillées dans la section 2.1.2. Les variables de travail naturelles en aérodynamique (comme la pression, les vitesses, ou encore le nombre de Mach) ne sont pas les variables du calcul. Dans ce chapitre, on suivra le travail de Shakib [145], qui a expliqué l'imposition de conditions de Dirichlet homogènes pour des calculs Navier-Stokes non linéaires utilisant les variables entropiques. Ce travail se démarque de celui de Shakib par la généralisation à des conditions non homogènes.

6.1.1 Imposition d'une variation de pression

On rappelle la définition des variables entropiques introduite dans la section 2.1.2 :

$$\mathbf{V} = \begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{pmatrix} = \begin{pmatrix} \mu \\ \frac{u_x}{T} \\ \frac{u_y}{T} \\ \frac{u_z}{T} \\ -\frac{1}{T} \end{pmatrix},$$

où μ est le potentiel chimique, T la température et (u_x, u_y, u_z) sont les vitesses. Pour imposer une variation de pression dp, une relation linéaire entre dp et les variations des variables entropiques dV_i doit être trouvée. On peut la trouver en linéarisant l'expression reliant la pression aux variables entropiques, donnée ci-dessous :

$$p = \exp\left(\frac{1}{R}\left(V_1 - \frac{V_2^2 + V_3^2 + V_4^2}{2V_5} + C\right) - \frac{\gamma}{\overline{\gamma}}\left(1 + \ln\left(-V_5\right)\right)\right), \tag{6.1}$$

où $\gamma = c_p/c_v$ est le coefficient adiabatique et $\overline{\gamma} = \gamma - 1$, et C est une constante dépendant uniquement de l'entropie de référence choisie. La pression dépend donc non linéairement des variables entropiques. Il convient de choisir une des variables entropiques pour imposer la variation de pression. Les variables V_2 à V_5 dépendent uniquement des vitesses et de la température. Il ne paraît donc « pas physique » d'imposer la variation de pression avec une autre variable que V_1. Isolons donc cette variable dans l'équation (6.1) :

$$\begin{aligned} V_1 &= R\ln p + \frac{V_2^2 + V_3^2 + V_4^2}{2V_5} + C + \frac{R\gamma}{\overline{\gamma}}\left(1 + \ln\left(-V_5\right)\right) \\ &= f_p(V_2, V_3, V_4, V_5, p)\,, \end{aligned} \tag{6.2}$$

On dérive cette expression pour obtenir

$$dV_1 = \sum_{i=2}^{5} \frac{\partial f_p}{\partial V_i} dV_i + \frac{\partial f_p}{\partial p} dp\,. \tag{6.3}$$

On obtient une relation linéaire entre la variation des variables entropiques dV_i et la variation de la pression dp, dont les coefficients dépendent non linéairement de la valeur des variables entropiques autour de laquelle la linéarisation a été effectuée :

$$\begin{aligned} \frac{\partial f_p}{\partial V_2} &= \frac{V_2}{V_5}\,, \qquad \frac{\partial f_p}{\partial V_3} = \frac{V_3}{V_5}\,, \qquad \frac{\partial f_p}{\partial V_4} = \frac{V_4}{V_5}\,, \\ \frac{\partial f_p}{\partial V_5} &= -\frac{V_2^2 + V_3^2 + V_4^2}{2V_5^2} + \frac{R\gamma}{\overline{\gamma}}\frac{1}{V_5}\,, \qquad \frac{\partial f_p}{\partial p} = \frac{R}{p}\,. \end{aligned}$$

On réécrit (6.3) sous forme vectorielle :

$$\mathbf{dV} = \mathbf{SdV} + \begin{pmatrix} \frac{Rdp}{p} \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \mathbf{SdV} + \alpha \mathbf{e}_1 \,, \tag{6.4}$$

où $\mathbf{e}_1$ est le premier vecteur de la base canonique de $\mathbb{R}^5$ et $\alpha = Rdp/p$. On note que seul α dépend de dp à imposer. La condition aux limites de Dirichlet homogène sur la variation de la pression correspond à $\alpha = 0$. La matrice $\mathbf{S}$ est définie par

$$\mathbf{S} = \begin{pmatrix} 0 & \frac{\partial f_p}{\partial V_2} & \frac{\partial f_p}{\partial V_3} & \frac{\partial f_p}{\partial V_4} & \frac{\partial f_p}{\partial V_5} \\ 0 & 1 & & & \\ \vdots & & \ddots & & \\ & & & & \\ 0 & & & & 1 \end{pmatrix} . \tag{6.5}$$

La matrice $\mathbf{S}$ s'applique aux cinq variables entropiques d'un seul nœud du maillage (voir la section 2.2.1), et remplace la première variable par l'équation (6.3). Pour imposer une conditions aux limites sur un système complet, on indexe la matrice $\mathbf{S}_i$ et le paramètre α_i par le numéro i du nœud. La matrice de transformation $\mathbf{S}_i$ doit agir sur un vecteur d'inconnues de taille $5 \times N_{nd}$ (N_{nd} étant le nombre de nœuds du maillage) et est donc définie par bloc :

$$\mathbf{S} = \begin{pmatrix} \mathbf{I}_5 & & & & & & \\ & \ddots & & & & \mathbf{0} & \\ & & \mathbf{I}_5 & & & & \\ & & & \mathbf{S_i} & & & \\ & & & & \mathbf{I}_5 & & \\ & \mathbf{0} & & & & \ddots & \\ & & & & & & \mathbf{I}_5 \end{pmatrix} . \tag{6.6}$$

Une réindexation similaire doit être effectuée pour le vecteur $\mathbf{e}_1$ qui se note alors $\mathbf{e}_{i,1}$. La modification des inconnues du système linéaire $\mathbf{Ax} = \mathbf{b}$ pour imposer les conditions de Dirichlet non homogènes au nœud i s'écrit

$$\mathbf{A}(\mathbf{S}_i\mathbf{x} + \alpha_i \mathbf{e}_{i,1}) = \mathbf{b} \,. \tag{6.7}$$

La transformation matricielle $\mathbf{M} \to \mathbf{MS}_i$ correspond à une combinaison linéaire des colonnes de $\mathbf{M}$, afin d'éliminer l'inconnue (la première variable du nœud i) imposé par la condition de Dirichlet non homogène. Le terme $\mathbf{A}\alpha_i\mathbf{e}_{i,1}$, qui s'ajoutera au second membre, permet de reporter sur les nœuds voisins du nœud i l'effet de l'inconnue imposée. Ainsi, l'équation (6.7) s'interprète

comme l'imposition des conditions aux limites non homogènes sur les fonctions de forme. Les fonctions test doivent également vérifier la condition aux limites de Dirichlet homogène associée. Pour cela, on multiplie à gauche le système par $\mathbf{S}_i^T$, ce qui donne

$$\mathbf{S}_i^T \mathbf{A} \mathbf{S}_i \mathbf{x} = \mathbf{S}_i^T (\mathbf{b} - \alpha_i \mathbf{A} \mathbf{e}_{i,1}) \ . \tag{6.8}$$

La multiplication à gauche du système linéaire par $\mathbf{S}_i^T$ élimine la ligne (ici la première du nœud i) dont l'inconnue correspondante est déjà déterminée. Cela permet en outre que cette inconnue reste nulle.

On obtient ainsi la procédure pour modifier le système linéaire afin d'imposer la condition aux limites de Dirichlet de pression non homogène au nœud i. Cette procédure est appliquée pour tous les nœuds de la frontière où s'applique une telle condition de Dirichlet. Une implémentation plus efficace pour la méthode des éléments finis de cette procédure consiste à modifier les matrices élémentaires, quand l'un des nœuds de l'élément considéré est sur la frontière. La combinaison de lignes et de colonnes se fait alors sur une matrice pleine, ce qui est bien plus aisé qu'après distribution de ces matrices élémentaires sur la matrice globale creuse stockée dans un format adapté.

Remarque 6.1. La matrice $\mathbf{S}_i^T \mathbf{A} \mathbf{S}_i$ est singulière, car la ligne et la colonne correspondant à la première variable du nœud i sont nulles. Pour simplifier l'utilisation de préconditionneurs, on remplace le terme nul sur la diagonal par 1.

Remarque 6.2. Après imposition des conditions aux limites de Dirichlet non homogènes, les inconnues du système linéaire sont prises dans un espace vérifiant des conditions de Dirichlet homogènes associées. Pour pouvoir retrouver à des fins de *post-process* la valeur des variables imposées, il faut appliquer la relation (6.3).

6.1.2 Imposition d'une onde incidente par les caractéristiques

Les conditions aux limites telles que décrites dans la section précédente imposent une variation de la pression sur la frontière. Pour l'ingénieur acousticien, cela pose problème car cette condition aux limites ne permet pas de contrôler l'énergie acoustique introduite dans le domaine. En effet, imposer une variation de pression sur une interface revient à imposer la somme de deux ondes acoustiques, l'une entrante dans le domaine et l'autre sortante. Aucun contrôle n'est donné pour séparer ces deux quantités. Ainsi, avec ces conditions aux limites, deux calculs sur deux configurations légèrement différentes seront difficilement comparables, puisque l'onde acoustique sortante ne sera pas la même dans les deux calculs, ce qui fera varier l'énergie introduite dans le calcul.

De nombreuses techniques existent pour contrôler précisément l'énergie introduite dans le domaine de calcul. Elles sont toutes issues de la méthode

des conditions aux limites transparentes, qui sont décrites en détail dans [157]. Ce type de conditions aux limites permet d'approcher par un calcul dans un domaine fini un problème dans un domaine infini. La modélisation du comportement en champ lointain de la solution se fait via ces conditions limites. En acoustique, cela s'obtient par des conditions qui laissent passer toutes les ondes sortantes sans réflexions parasites et, en l'absence de source à l'infini, n'en laissent rentrer aucune.

L'utilisation des caractéristiques de l'équation d'Euler est une méthode simple pour obtenir des conditions aux limites transparentes. Soit un plan infini, de normale unitaire $\mathbf{n}$ (que l'on considéra entrante dans le domaine si ce plan est une frontière), et $\mathbf{s}$, $\mathbf{t}$ deux vecteurs complétant la base orthonormale. La diagonalisation du jacobien du flux d'Euler selon la direction $\mathbf{n}$ donne 5 valeurs propres qui sont $(\mathbf{u}\cdot\mathbf{n}, \mathbf{u}\cdot\mathbf{n}, \mathbf{u}\cdot\mathbf{n}, \mathbf{u}\cdot\mathbf{n}-c, \mathbf{u}\cdot\mathbf{n}+c)$. Les coordonnées dans la base générée par ces vecteurs propres sont appelées variables caractéristiques. L'expression de ces variables en fonction de la variation des variables primales $(d\rho, \mathbf{du}, dp)$

$$\delta\mathbf{W} = \begin{pmatrix} dW_1 \\ dW_2 \\ dW_3 \\ dW_4 \\ dW_5 \end{pmatrix} = \begin{pmatrix} d\rho - \frac{1}{c^2}dp \\ \mathbf{s}\cdot\mathbf{du} \\ \mathbf{t}\cdot\mathbf{du} \\ \mathbf{n}\cdot\mathbf{du} + \frac{1}{\rho c}dp \\ -\mathbf{n}\cdot\mathbf{du} + \frac{1}{\rho c}dp \end{pmatrix}. \tag{6.9}$$

Les quantités précédées par d sont relatives à l'onde acoustique et non à l'écoulement moyen. Les trois premières variables caractéristiques correspondent à la convection à une vitesse $\mathbf{u}\cdot\mathbf{n}$ respectivement d'une onde d'entropie et d'ondes de vorticité. La variable caractéristique dW_4 est une onde acoustique convectée se propageant à la vitesse $\mathbf{u}\cdot\mathbf{n}-c$, tout comme la variable caractéristique dW_5 à la vitesse $\mathbf{u}\cdot\mathbf{n}+c$.

On se place dans le cadre d'une tuyère orientée dans la direction $\mathbf{e}_x$. Le plan limite est une section de la tuyère orthonormale à son axe. La vitesse de l'écoulement porteur $\mathbf{u}$ est donc dirigée selon $\mathbf{n}=\mathbf{e}_x$ que l'on considèrera rentrant. On note $u=\mathbf{u}\cdot\mathbf{n}$.

Imposer le mode acoustique rentrant seul implique d'imposer la caractéristique rentrante de valeur propre $u+c$, et de laisser les autres caractéristiques libres. La théorie modale de la propagation acoustique dans les tubes que l'on trouvera dans [130] permet de calculer la variation de pression dp et la vitesse acoustique du du mode à imposer, et ainsi calculer la caractéristique rentrante dW_4. C'est pourquoi on appellera dans le reste de ce manuscrit ce type de conditions des conditions aux limites caractéristiques incidentes ou encore caractéristiques.

Imposer la caractéristique dW_4 ajoute une dépendance affine entre la variation de pression dp et la variation de la vitesse normale. On remplace dans (6.3) la variation de pression dp par la relation affine introduite par la

définition de dW_4 :

$$dp = \rho c \, dW_4 - \rho c \, du_x \, . \tag{6.10}$$

De la définition des variables entropiques, $u_x = -V_2/V_5$. Ainsi $du_x = -1/V_5 \, dV_2 + V_2/V_5 \, dV_5$. Ainsi,

$$dp = \rho c \, dW_4 + \frac{\rho c}{V_5} dV_2 - \rho c \frac{V_2}{V_5^2} dV_5 \, . \tag{6.11}$$

On injecte cette expression de dp dans l'équation (6.3) pour obtenir les dérivées partielles de la fonction implicite f_{W_4} :

$$dV_1 = \sum_{i=2}^{5} \frac{\partial f_{W_4}}{\partial V_i} dV_i + \frac{\partial f_{W_4}}{\partial W_4} dW_4 \, . \tag{6.12}$$

Pour un gaz parfait, l'équation d'état $p/\rho = RT$ et la définition de la vitesse du son $c^2 = \gamma RT$ permettent de simplifier l'expression des dérivées partielles de f_{W_4} :

$$\begin{gathered} \frac{\partial f_{W_4}}{\partial V_2} = \frac{V_2}{V_5} - c \, , \qquad \frac{\partial f_{W_4}}{\partial V_3} = \frac{V_3}{V_5} \, , \qquad \frac{\partial f_{W_4}}{\partial V_4} = \frac{V_4}{V_5} \, , \\ \frac{\partial f_{W_4}}{\partial V_5} = -\frac{V_2^2 + V_3^2 + V_4^2}{2V_5^2} + \frac{R\gamma}{\overline{\gamma}} \frac{1}{V_5} + c \frac{V_2}{V_5} \, , \qquad \frac{\partial f_{W_4}}{\partial W_4} = \frac{\gamma R}{c} \, . \end{gathered} \tag{6.13}$$

6.1.3 Validation des conditions aux limites incidentes

Les conditions aux limites incidentes ont été testées sur un problème très simple que l'on appellera cavité accordée. Il s'agit d'une géométrie unidimensionnelle, de longueur L, où l'on impose un signal rentrant par une face, et l'autre face est parfaitement réfléchissante. Un schéma d'un tel dispositif est donné sur la figure 6.1. L'accord de la cavité s'effectue en réglant la fréquence f de l'onde ou la longueur L de la cavité afin que l'onde réfléchie par la paroi du fond soit en phase avec l'onde incidente, créant ainsi une onde stationnaire constructive. L'écoulement porteur est évidemment nul au sein de la cavité.

Onde plane

La validation la plus simple s'effectue avec une onde plane se propageant dans l'axe de la cavité. Dans ce cas-là, l'accord de la cavité se fait en choisissant une fréquence telle que la longueur de la cavité soit un multiple de $\lambda/2$, où λ désigne la longueur d'onde. Utiliser une condition aux limites de pression totale telle que décrite dans la section 6.1.1 sur le plan d'entrée impose à cet endroit la somme des ondes incidente et réfléchie. Imposer seulement l'onde incidente par les conditions limites décrites dans la section

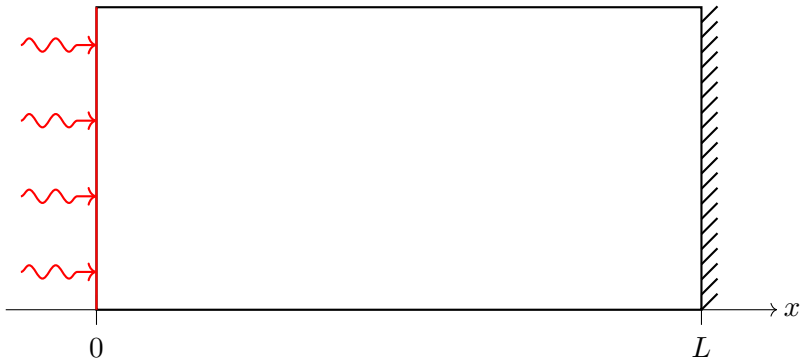


FIGURE 6.1 – Schéma de la cavité accordée. L'onde est injectée en $x = 0$.

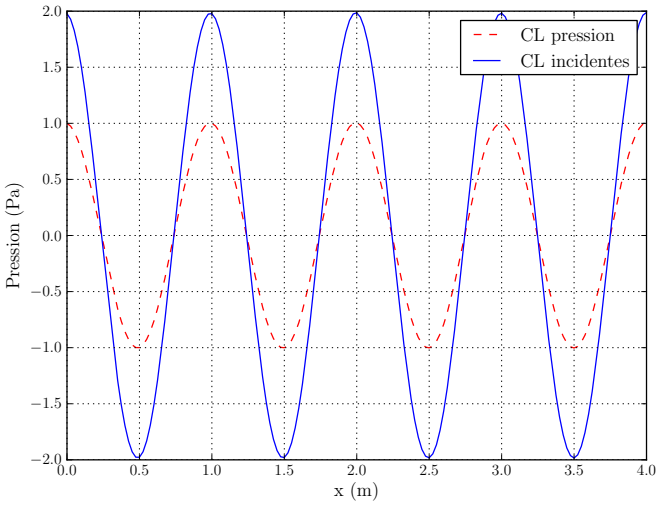


FIGURE 6.2 – Partie réelle de la variation de pression dans une cavité de 4 m de long. Onde plane à 340 Hz.

6.1.2 conduira à une amplitude de l'onde stationnaire deux fois supérieure. La figure 6.2 présente la partie réelle de l'onde stationnaire dans une cavité de 4 m de long accordée à 340 Hz. Pour les deux conditions aux limites utilisées, une amplitude d'un Pascal a été utilisée. Comme attendu, les conditions aux limites incidentes doublent l'amplitude de l'onde stationnaire par rapport à celle induite par les conditions aux limites totales.

Modes d'ordre élevé dans un tube

D'autres modes que des ondes planes existent dans un tube. Il existe une infinité de modes que l'on appellera d'ordre élevé qui ont chacun une fréquence de coupure en-deçà de laquelle ils ne se propagent pas. Ils permettent de tester la qualité des conditions aux limites incidentes pour une onde dont la

propagation n'est pas normale à la frontière. On cherche ici à accorder une cavité dans laquelle un de ces modes complexes serait en résonance. Cette cavité est un tube de rayon R_0 et de longueur L. Un système de coordonnées orthoradial (x, r, θ) est utilisé. Le tube est fermé en son fond, en $x = L$, par une surface parfaitement réfléchissante. Les modes sont introduits dans le plan de coordonnées $x = 0$ que l'on nomme plan modal. Les parois radiales du tubes sont parfaitement réfléchissantes. La formule permettant de déterminer pour une forme modale donnée toutes les fréquences de résonance est plus complexe que pour l'onde plane, où il suffit de choisir que la longueur de la cavité soit multiple de la demi-longueur d'onde. Nous allons déterminer cette formule. La pression du mode complexe (m, n) est donnée par la formule suivante [130] :

$$p_{m,n}(x, r, \theta) = p_0 J_m(k_{m,n} r)\, e^{i(\omega t - m\theta + k_x x)} \tag{6.14}$$

Ici, p_0 est l'amplitude du mode, J_m est la fonction de Bessel de premier type d'ordre m. Le nombre d'onde radial $k_{m,n}$ est tel que $k_{m,n} R_0 = \chi_{m,n}$ est le n-ième zéro de la dérivée de la fonction de Bessel J'_m, qui est une conséquence de la réflexion parfaite sur la paroi radiale du tube. Cette définition conduit à indexer le mode plan par $(m = 0, n = 1)$. Les nombres d'ondes axiaux k_x et radiaux $k_{m,n}$ sont reliés par la relation de dispersion au nombre d'onde $k = 2\pi f/c$:

$$k^2 = k_x^2 + k_{m,n}^2 \tag{6.15}$$

La pression totale dans le tube est la somme des deux modes entrant et réfléchi. Leur amplitude est égale, car la surface en $x = L$ est parfaitement réfléchissante. Ces deux modes diffèrent seulement par leur signe du nombre d'onde axial k_x et une différence de phase φ. La pression totale dans le tube s'écrit donc

$$P_{tot} = p_0 J_m(k_{m,n} r)\, e^{i\omega t} \Big(e^{i(k_x x - m\theta)} + e^{i(-k_x x - m\theta + \varphi)} \Big) \tag{6.16}$$

La cavité résonnante est bouchée en $x = L$. La réflexion parfaite sur ce mur demande $\frac{\partial P_{tot}}{\partial x} = 0$. On en déduit que $\varphi = -2k_x L$. La cavité est accordée si l'onde réfléchie est la même que l'onde incidente dans le plan d'entrée $x = 0$, *i.e.* si :

$$1 = e^{-2ik_x L}$$

Cela définit entièrement le nombre d'onde axial k_x par la longueur L de la cavité et p, le nombre (entier) de demi longueurs d'onde que la cavité contient :

$$k_x = \frac{p\pi}{L} \tag{6.17}$$

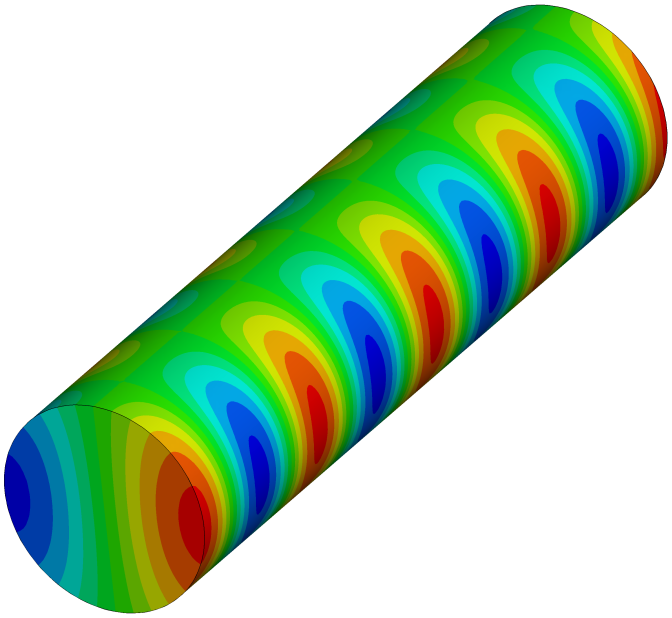


FIGURE 6.3 – Partie réelle de la variation de pression pour un mode (1,1) accordé à 4 λ_x dans une cavité cylindrique.

On combine cette définition avec celle de $k = 2\pi/f$ dans la relation de dispersion (6.15) et on obtient

$$\left(\frac{2\pi f}{c}\right)^2 = \left(\frac{p\pi}{L}\right)^2 + k_{m,n}^2$$

Cette relation permet d'exprimer la fréquence en fonction du nombre de demi longueurs d'onde dans la cavité :

$$f(p) = \sqrt{\left(\frac{pc}{2L}\right)^2 + f_c^2} \tag{6.18}$$

Ici, $f_c = \frac{k_{m,n}c}{2\pi}$ est la fréquence de coupure du mode (m, n) considéré. On constate que pour p très grand, la fréquence du mode se rapproche de celle du mode plan, à savoir $f \sim \frac{pc}{2L}$. Cette formule permet d'obtenir la fréquence pour accorder un mode à une cavité en fonction du nombre de périodes souhaité. Par exemple, la partie réelle de la variation de pression pour un mode (1,1) avec $p = 8$, c'est-à-dire un accord de la cavité à 4 λ_x, est donnée sur la figure 6.3.

La relation de dispersion (6.15) est également une relation de Pythagore qui permet de définir un angle α tel que présenté sur la figure 6.4. Comme $k_{m,n}$ dépend seulement du mode considéré, l'angle α décroît lorsque la fréquence croît. À la fréquence de coupure $p = 0$, donc α vaut 90°. On notera que l'angle α n'est pas un angle physique, car de tous les nombres d'onde de la relation de

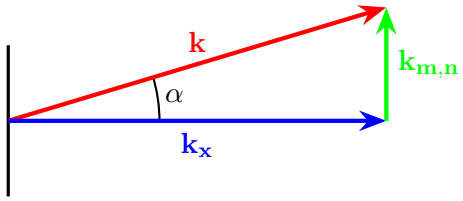


FIGURE 6.4 – Relation de dispersion et définition de l'angle α

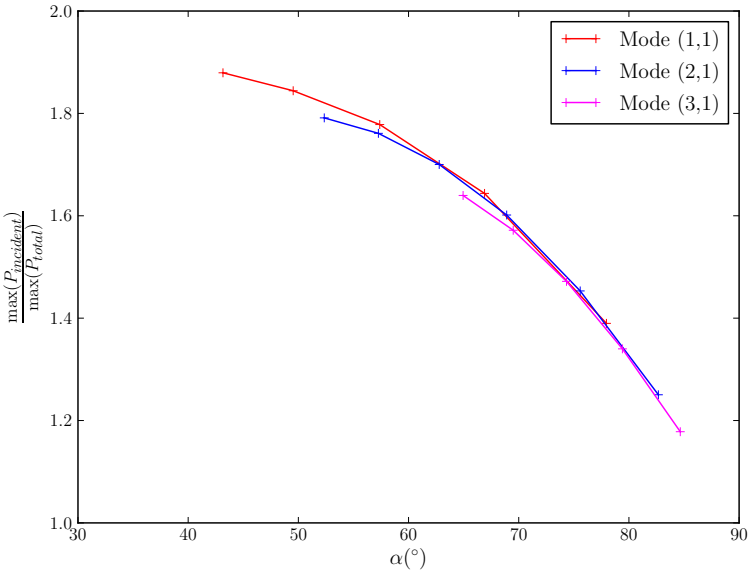


FIGURE 6.5 – Perte de pression en fonction de α

dispersion (6.15), seul k_x est réellement un vecteur. Une interprétation plus juste serait de considérer $k_{m,n}$ comme un vecteur « tournant », et qu'alors α serait la demi ouverture d'un cône de révolution obtenue par rotation de la figure 6.4.

Cette définition d'angle permet d'étudier le comportement des conditions aux limites caractéristiques en fonction de l'angle α du mode imposé à la normale du plan modal d'entrée. Les conditions aux limites transparentes par la méthode des caractéristiques se dégradent lorsque l'onde incidente n'est pas normale à la surface [157]. On peut donc supposer que les conditions aux limites incidentes de la section 6.1.2 connaissent le même problème.

La figure 6.5 présente le rapport du maximum de la pression dans la cavité d'un mode imposé par les conditions aux limites incidentes par rapport à celle d'une mode imposé par des conditions aux limites de pression totale, en fonction de l'angle α, ou de manière équivalente, du rapport $\frac{k_x}{k}$. On rappelle que si les conditions aux limites incidentes sont parfaites, ce rapport vaut 2. Ce résultat avait été trouvé pour des ondes planes. Pour des modes

tournants, il est inférieur à 2, et se dégrade au fur et à mesure que l'angle α augmente. On retrouve donc le résultat intuité que les conditions aux limites caractéristiques sont plus performantes quand les ondes sont normales à la surface. Il n'a pas été possible d'obtenir des points pour des angles inférieurs à 40°. Ils correspondent à p (et donc k_x) très grand, c'est-à-dire une longueur d'onde λ_x dans la direction x très faible. La résonance exacte est alors difficile à obtenir, car elle nécessite une grande précision dans la valeur de λ_x. De surcroît, le maillage impose une limite fréquentielle, et empêche de choisir des grandes valeurs de p.

Plusieurs commentaires sont à formuler sur les résultats de la figure 6.5. Premièrement, on constate que les conditions aux limites incidentes tendent à imposer la pression totale dans la limite des angles très grands. Ce fonctionnement surprenant est délicat à interpréter. Une piste pour éclaircir ce phénomène est de se rappeler que les conditions aux limites incidentes imposent une combinaison affine des variations de pression et de la vitesse normale. Or, un mode proche de sa fréquence de coupure a une vitesse acoustique presque tangente au plan d'entrée. Dans le cas limite d'un mode calculé à sa fréquence de coupure, son nombre radial k_x est nul, donc la vitesse acoustique normale l'est également. Alors, la caractéristique dW_4 est entièrement déterminée par la pression. Dans ce cas là, la contribution des conditions de Dirichlet non homogènes au second membre du système linéaire sera la même lorsqu'on impose la pression totale et la caractéristique incidente (car $\frac{\partial f_{W_4}}{\partial W_4} dW_4 = \frac{\partial f_p}{\partial p} \delta p$). Cela n'est qu'une partie de la réponse, car les matrices $\mathbf{S}$ de transformation du système linéaire ne sont pas les mêmes.

La deuxième remarque sur la figure 6.5 est d'ordre pratique pour l'ingénieur utilisant les conditions aux limites incidentes, c'est-à-dire l'écart entre l'énergie incidente voulue et celle effectivement appliquée. Il est important pour lui de quantifier l'erreur commise sur le calcul due aux conditions limites. La figure 6.5 permet de constater que pour des angles α inférieurs à 50°, la perte de pression acoustique est inférieure à 20%, qui est acceptable en pratique. Enfin, notons que la perte de signal pour des angles supérieurs à 50% est délicate à utiliser quantitativement. En effet, pour de tels angles, l'onde sortante est très certainement en partie réfléchie par la condition limite d'entrée, qui elle même n'injecte pas assez d'énergie dans le système. Il devient donc délicat de quantifier à partir de la figure 6.5 l'amplitude du mode incident que l'on impose par les conditions aux limites. Une expérience numérique pour mesurer précisément cela aurait été de placer en $x = L$ une condition aux limites parfaitement absorbante. Or le code AeTher ne dispose de ce type de conditions. Pour contourner ce problème, on pourrait déraffiner progressivement le maillage le long du tube pour que les modes soient détruits avant le bout du conduit.

6.1.4 Imposition de la vitesse de paroi en aéroélasticité

L'autre application durant cette thèse des conditions aux limites de Dirichlet non homogènes a été l'aéroélasticité. Pour les calculs dit statiques, sans terme fréquentiel, on impose le déplacement via un terme *ALE* dans le volume (voir section 2.3.1). Lorsque ce déplacement est considéré comme étant harmonique, on impose au fluide à la paroi une vitesse imaginaire pure valant $j\omega\mathbf{dx}$, où $\mathbf{dx}$ est le déplacement du mode structural considéré. Pour l'imposer, on rappelle la définition des variables entropiques faisant intervenir la vitesse du fluide :

$$\begin{pmatrix} V_2 \\ V_3 \\ V_4 \end{pmatrix} = \begin{pmatrix} \frac{u_x}{T} \\ \frac{u_y}{T} \\ \frac{u_z}{T} \end{pmatrix} = -V_5 \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}$$

Les variables dérivées valent donc

$$\begin{pmatrix} dV_2 \\ dV_3 \\ dV_4 \end{pmatrix} = -V_5 \begin{pmatrix} du_x \\ du_y \\ du_z \end{pmatrix} - dV_5 \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = -V_5 \begin{pmatrix} du_x \\ du_y \\ du_z \end{pmatrix} + \frac{dV_5}{V_5} \begin{pmatrix} V_2 \\ V_3 \\ V_4 \end{pmatrix} \tag{6.19}$$

Les variables du_x, du_y et du_z sont imposées et valent

$$\mathbf{du} = \begin{pmatrix} du_x \\ du_y \\ du_z \end{pmatrix} = j\omega\boldsymbol{\delta}\mathbf{x} \tag{6.20}$$

où $\boldsymbol{\delta}\mathbf{x}$ est le déplacement du mode structural imposé (voir section 2.3.1). La transformation à imposer aux variables de travail est donc

$$\mathbf{dV} \mapsto \mathbf{SdV} - V_5 \begin{pmatrix} 0 \\ j\omega\delta x \\ j\omega\delta y \\ j\omega\delta z \\ 0 \end{pmatrix} \tag{6.21}$$

La matrice de transformation $\mathbf{S}$ est alors définie par

$$\mathbf{S} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{V_2}{V_5} \\ 0 & 0 & 0 & 0 & \frac{V_3}{V_5} \\ 0 & 0 & 0 & 0 & \frac{V_4}{V_5} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.22}$$

Avant l'utilisation de cette condition aux limites, la vitesse imaginaire pure était imposée par pénalisation à la paroi. Pour comparer ces deux façons d'imposer la conditions aux limites, des calculs sur la maquette DTP

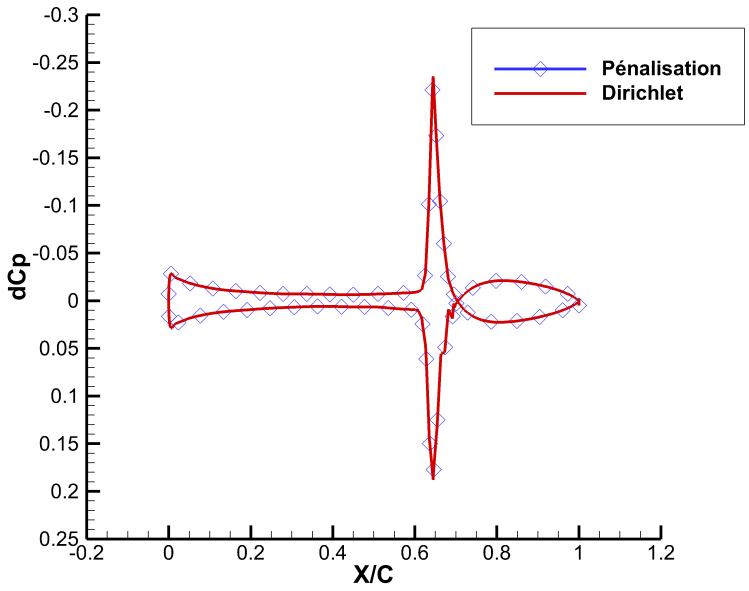


FIGURE 6.6 – Partie imaginaire du dCp à mi-envergure de la maquette DTP pour un mouvement de tangage à 30 Hz à Mach 0,88. Imposition de la vitesse à la paroi par pénalisation ou par condition de Dirichlet non homogène.

présentée à la section 3.3.4 ont été effectués. La figure 6.6 présente la partie imaginaire de la variation de pression à la peau à mi-envergure pour un mouvement de tangage à 30 Hz réalisé à Mach 0,88. Les résultats sont identiques pour les deux méthodes. La figure 6.7 montre la convergence du résidu non adimensionné. Les deux courbes ne peuvent se distinguer, à l'exception du résidu initial. La pénalisation introduit en effet un résidu très fort aux nœuds de la paroi. Ce résidu local disparaît après la première itération, à partir de laquelle les deux courbes de convergence sont superposées. Les conditions limites de Dirichlet non homogènes pour imposer la vitesse à la paroi en aéroélasticité sont désormais toujours utilisées dans AeTher linéarisé.

6.2 Conditions aux limites transparentes

6.2.1 Présentation

Les premiers calculs aéroacoustiques utilisant AeTher et effectués avant cette thèse ont été d'abord réalisés en déraffinant le maillage loin de l'avion, afin d'atténuer les ondes acoustiques avant qu'elles n'atteignent le bord. Sur la frontière représentant l'infini, des conditions de Dirichlet homogènes

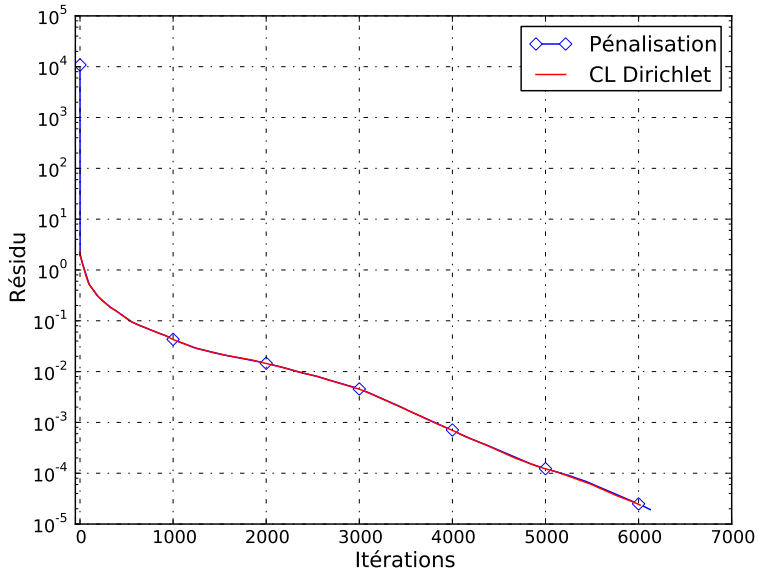


FIGURE 6.7 – Résidu GMRES non adimensionné. Imposition de la vitesse à la paroi par pénalisation ou par condition de Dirichlet non homogène.

sur toutes les variables étaient imposées. Il a été remarqué que les ondes atténuées arrivant sur cette frontière n'étaient pas, ou peu réfléchies. Le même phénomène a été confirmé sur des plus petits domaines de calcul, pour lesquels les ondes n'étaient presque pas atténuées lorsqu'elle atteignent le bord. Il s'est donc avéré que des conditions de Dirichlet homogènes sur toutes les variables entropiques se comportaient comme des conditions aux limites transparentes.

Une des contributions de cette thèse a été de comprendre le mécanisme expliquant cette propriété contre-intuitive. Un modèle 1D a permis de montrer que c'est la stratégie de stabilisation SUPG adoptée dans AeTher qui produit ce résultat surprenant.

6.2.2 Modèle 1D

Pour comprendre les conditions aux limites transparentes à l'infini, on utilise les équations d'Euler linéarisées fréquentielles, qui s'écrivent en 1D

$$j\omega\widetilde{\mathbf{A}}_0\mathbf{V} + \widetilde{\mathbf{A}}_1\mathbf{V}_{,x} = \mathbf{0}\,, \tag{6.23}$$

afin d'étudier leur propriétés une fois discrétisé sur un segment 1D.

Explication sans terme de temps

Pour simplifier l'analyse, on s'intéresse d'abord uniquement aux équations d'Euler linéarisées sans terme fréquentiel, qui se réduisent au terme convectif.

Théorème 6.1. *La discrétisation des équations d'Euler stationnaires linéarisées par la méthode SUPG, avec la matrice de stabilisation*

$$\boldsymbol{\tau} = \widetilde{\mathbf{A}}_0^{-1} \left(\frac{\partial \xi}{\partial x}^2 \mathbf{A}_1^2 \right)^{-\frac{1}{2}} ,$$

qui est la réduction à une dimension de la définition (5.2)*, est identique à une discrétisation par un schéma de différences finies décentrées sur les variables caractéristiques.*

Démonstration. C'est une conséquence des théorèmes 5.7 et 5.10 sans utiliser l'approximation effectuée dans le lemme 5.9 développée pour le passage au cas multidimensionnel.

Pour fixer les idées, on rappelle les grandes lignes de la preuve.

La formulation stabilisée par SUPG des équations d'Euler linéarisées s'écrit

$$\int_{\Omega^e} \mathbf{Y}^H \widetilde{\mathbf{A}}_1 \mathbf{V}_{,x} + \mathbf{Y}_{,x}^H \widetilde{\mathbf{A}}_1 \boldsymbol{\tau} \widetilde{\mathbf{A}}_1 \mathbf{V}_{,x} \mathrm{d}\Omega^e = 0 \,, \tag{6.24}$$

où les fonctions tests sont notées cette fois-ci $\mathbf{Y}$. On reprend les notations introduites dans la section 5.1.5. Soient $\mathbf{L}$ le facteur de Cholesky de $\widetilde{\mathbf{A}}_0$, et $\widehat{\mathbf{S}}$ la matrice orthonormale de diagonalisation de $\widehat{\mathbf{A}}_1 = \mathbf{L}^{-1} \widetilde{\mathbf{A}}_1 \mathbf{L}^{-T}$, et $\boldsymbol{\Lambda}$ ses valeurs propres associées. De plus, $\widetilde{\mathbf{S}} = \mathbf{L}^{-T} \widehat{\mathbf{S}}$ diagonalise $\boldsymbol{\tau}$:

$$\boldsymbol{\tau} = \frac{\partial x}{\partial \xi} \widetilde{\mathbf{S}} \left| \boldsymbol{\Lambda} \right|^{-1} \widetilde{\mathbf{S}}^T .$$

De plus,

$$\begin{aligned} \widetilde{\mathbf{S}}^T \widetilde{\mathbf{A}}_1 \widetilde{\mathbf{S}} &= \widehat{\mathbf{S}}^T \mathbf{L}^{-1} \widetilde{\mathbf{A}}_1 \mathbf{L}^{-T} \widehat{\mathbf{S}} \\ &= \widehat{\mathbf{S}}^T \widehat{\mathbf{A}}_1 \widehat{\mathbf{S}} \\ &= \boldsymbol{\Lambda} \,. \end{aligned}$$

Enfin, remarquons que

$$\begin{aligned} \widetilde{\mathbf{S}}^T \widetilde{\mathbf{A}}_1 \boldsymbol{\tau} \widetilde{\mathbf{A}}_1 \widetilde{\mathbf{S}} &= \boldsymbol{\Lambda} \mathbf{S}^{-1} \boldsymbol{\tau} \mathbf{S}^{-T} \boldsymbol{\Lambda} \\ &= \frac{\partial x}{\partial \xi} \boldsymbol{\Lambda} \left| \boldsymbol{\Lambda} \right|^{-1} \boldsymbol{\Lambda} \\ &= \frac{\partial x}{\partial \xi} \left| \boldsymbol{\Lambda} \right| . \end{aligned}$$

Ainsi, si l'on pose $\mathbf{W} = \widetilde{\mathbf{S}}^{-1} \mathbf{V}$ (attention, $\mathbf{W}$ désignait auparavant les fonctions test) et $\mathbf{Z} = \widetilde{\mathbf{S}}^{-1} \mathbf{Y}$ qui sont les variables caractéristiques, la formulation stabilisée (6.24) devient

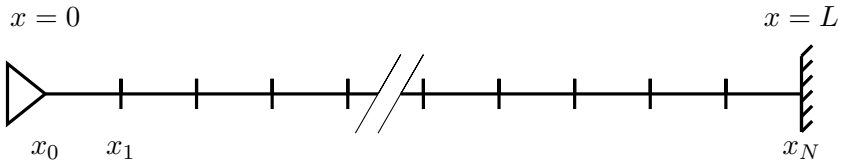


FIGURE 6.8 – Schéma du segment 1D.

$$\int_{\Omega^e} \mathbf{Z}^H \mathbf{\Lambda} \mathbf{W}_{,x} + \mathbf{Z}_{,x}^H \frac{\partial x}{\partial \xi} |\mathbf{\Lambda}| \mathbf{W}_{,x} \mathrm{d}\Omega^e = 0 \,. \tag{6.25}$$

On retrouve le résultat de la section 5.1.2, à savoir qu'il existe une base dans lequel le système d'advection 1D est diagonal. La fin de la preuve du théorème 5.7 permet de conclure. □

La vitesse de convection de chaque variable caractéristique est donnée par les coefficients de $\mathbf{\Lambda} = \mathrm{diag}(u, u + c, u - c)$, où u est la vitesse de l'écoulement porteur et c la célérité du son.

Corollaire 6.2. *Pour une vitesse d'advection u positive et subsonique, la discrétisation des variables caractéristiques annoncée dans le théorème 6.1 par des éléments finis de longueur uniforme h est*

$$\begin{cases} u \dfrac{w_i^{(1)} - w_{i-1}^{(1)}}{h} = 0 \\ (u + c) \dfrac{w_i^{(2)} - w_{i-1}^{(2)}}{h} = 0 \\ (u - c) \dfrac{w_{i+1}^{(3)} - w_i^{(3)}}{h} = 0 \,. \end{cases} \tag{6.26}$$

Démonstration. Immédiat par application du théorème 5.5. □

Considérons maintenant un segment de droite $[0, L]$ discrétisé en $N + 1$ points numérotés de 0 à N. Un signal acoustique est injecté en $x = 0$ et une condition aux limites de Dirichlet homogène sur toutes les variables est imposée en $x = L$. Ce segment est représenté sur la figure 6.8.

Théorème 6.3. *La condition aux limites de Dirichlet homogènes sur toutes les variables imposées en $x = L$ pour les équations d'Euler linéarisées avec une vitesse d'écoulement porteur positive et subsonique est transparente pour les deux premières variables caractéristiques $w^{(1)}$ et $w^{(2)}$ et impose la troisième variable caractéristique $w^{(3)}$ à être uniformément nulle sur tout le segment.*

Démonstration. C'est une conséquence directe de la discrétisation décentrée par caractéristique, tel que décrit par le théorème 5.5 et son corollaire 6.2. Une condition aux limites de Dirichlet homogène sur toutes les variables

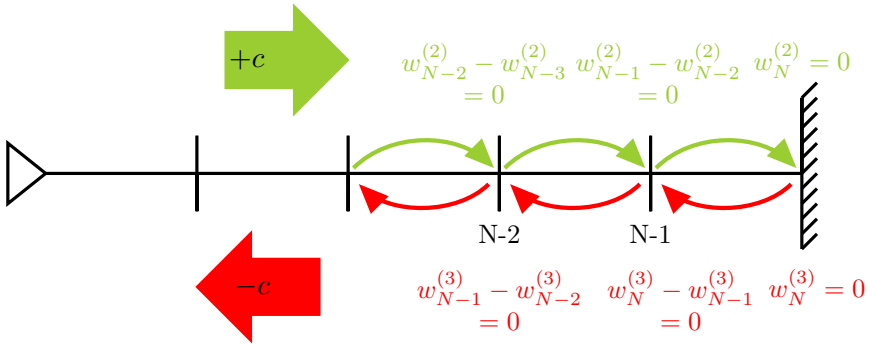


FIGURE 6.9 – Décentrement différent des caractéristiques $w^{(2)}$ et $w^{(3)}$ avec la discrétisation SUPG pour une vitesse de fluide nulle

est également une conditions de Dirichlet homogène pour les trois variables caractéristiques. Les équations (6.26) suivies par les variables caractéristiques discrétisées suffisent à montrer la propriété demandée. Les deux premières variables ne vont pas voir la condition aux limite en x_N avant d'atteindre ce point, alors que la dernière sera imposée à zéro. Cet argument est représenté sur la figure 6.9. En x_{N-1}, les équations suivies par les variables caractéristiques sont

$$\begin{cases} u\dfrac{w^{(1)}_{N-1} - w^{(1)}_{N-2}}{h} = 0 \\ (u+c)\dfrac{w^{(2)}_{N-1} - w^{(2)}_{N-2}}{h} = 0 \\ (u-c)\dfrac{0 - w^{(3)}_{N-1}}{h} = 0\,. \end{cases}$$

Cela montre qu'en x_{N-1} les deux premières caractéristiques ne sont pas affectées par la condition de Dirichlet homogène, alors que la troisième variable est imposée à 0.

Cela prouve le caractère transparent des conditions aux limites de Dirichlet homogènes sur toutes les variables quand une méthode de stabilisation SUPG bien choisie est utilisée. □

Ajout du terme temporel

Dans la partie précédente, on s'est cantonné à un terme d'advection pure. Maintenant, nous allons étudier l'impact sur la discrétisation de l'ajout du terme fréquentiel, à la fois sur les conditions aux limites transparentes, ainsi que sur la stabilité et la précision du schéma. Les propriétés du schéma élément fini Galerkin standard ainsi que le schéma élément fini stabilisé par

SUPG seront étudiées. Les équations d'Euler linéarisées fréquentielles à la pulsation ω s'écrivent

$$j\omega\widetilde{\mathbf{A}}_0\mathbf{V} + \widetilde{\mathbf{A}}_1\mathbf{V}_{,1} = \mathbf{0}\,. \tag{6.27}$$

Lemme 6.4. *La méthode des éléments finis de Galerkin appliquées aux équations d'Euler linéarisées fréquentielles* (6.27) *donne la discrétisation suivantes des variables caractéristiques :*

$$\begin{cases} j\omega\left(\frac{1}{6}w_{i-1}^{(1)} + \frac{2}{3}w_i^{(1)} + \frac{1}{6}w_{i+1}^{(1)}\right) + u\frac{w_{i+1}^{(1)} - w_{i-1}^{(1)}}{2h} = 0 \\ j\omega\left(\frac{1}{6}w_{i-1}^{(2)} + \frac{2}{3}w_i^{(2)} + \frac{1}{6}w_{i+1}^{(2)}\right) + (u+c)\frac{w_{i+1}^{(2)} - w_{i-1}^{(2)}}{2h} = 0 \\ j\omega\left(\frac{1}{6}w_{i-1}^{(3)} + \frac{2}{3}w_i^{(3)} + \frac{1}{6}w_{i+1}^{(3)}\right) + (u-c)\frac{w_{i+1}^{(3)} - w_{i-1}^{(3)}}{2h} = 0\,, \end{cases} \tag{6.28}$$

où par exemple $w_{i+1}^{(2)}$ désigne la valeur de la deuxième variable caractéristique au nœud $i+1$.

Démonstration. Vu que

$$\begin{aligned} \widetilde{\mathbf{S}}^T\widetilde{\mathbf{A}}_0\widetilde{\mathbf{S}} &= \widehat{\mathbf{S}}^T\mathbf{L}^{-1}\mathbf{L}\mathbf{L}^T\mathbf{L}^{-T}\widehat{\mathbf{S}} \\ &= \widehat{\mathbf{S}}^T\widehat{\mathbf{S}} \\ &= \mathbf{I}\,, \end{aligned}$$

la forme faible des équations (6.27) exprimées dans variables caractéristiques sur un élément Ω^e est

$$\int_{\Omega^e} \mathbf{Z}^H j\omega\mathbf{I}\mathbf{W} + \mathbf{Z}^H\mathbf{\Lambda}\mathbf{W}_{,x}\mathrm{d}x = 0\,. \tag{6.29}$$

Les résultats d'intégration élémentaires utilisés dans la preuve du théorème 5.5 permettent de conclure. □

Lemme 6.5. *La méthode des éléments finis stabilisées par la méthode SUPG appliquées aux équations d'Euler linéarisées fréquentielles* (6.27) *donne la discrétisation suivantes des variables caractéristiques :*

$$\begin{cases} j\omega\left(\frac{5}{12}w_{i-1}^{(1)} + \frac{2}{3}w_i^{(1)} - \frac{1}{12}w_{i+1}^{(1)}\right) + u\frac{w_i^{(1)} - w_{i-1}^{(1)}}{h} = 0 \\ j\omega\left(\frac{5}{12}w_{i-1}^{(2)} + \frac{2}{3}w_i^{(2)} - \frac{1}{12}w_{i+1}^{(2)}\right) + (u+c)\frac{w_i^{(2)} - w_{i-1}^{(2)}}{h} = 0 \\ j\omega\left(-\frac{1}{12}w_{i-1}^{(3)} + \frac{2}{3}w_i^{(3)} + \frac{5}{12}w_{i+1}^{(3)}\right) + (u-c)\frac{w_{i+1}^{(3)} - w_i^{(3)}}{h} = 0\,. \end{cases} \tag{6.30}$$

Démonstration. Les équations stabilisées par la méthode SUPG, écrites en variables entropiques, se notent

$$\int_{\Omega^e} \left(\mathbf{Y} + \boldsymbol{\tau}\widetilde{\mathbf{A}}_1\mathbf{Y}_{,x}\right)^H \cdot \left(j\omega\widetilde{\mathbf{A}}_0\mathbf{V} + \widetilde{\mathbf{A}}_1\mathbf{V}_{,x}\right) d\Omega^e = 0\,. \tag{6.31}$$

Le terme $\boldsymbol{\tau}\widetilde{\mathbf{A}}_1\mathbf{Y}_{,x}$ qui modifie les fonctions de pondération ajoute deux contributions à la discrétisation, correspondant à l'opérateur de convection et l'opérateur fréquentiel. Simplifions ce dernier terme :

$$\begin{aligned}\widetilde{\mathbf{S}}^T\widetilde{\mathbf{A}}_1\boldsymbol{\tau}j\omega\widetilde{\mathbf{A}}_0\widetilde{\mathbf{S}} &= j\omega\boldsymbol{\Lambda}\widetilde{\mathbf{S}}^{-1}\boldsymbol{\tau}\widetilde{\mathbf{S}}^{-T}\\ &= j\omega\frac{\partial x}{\partial\xi}\boldsymbol{\Lambda}|\boldsymbol{\Lambda}|^{-1}\\ &= j\omega\frac{\partial x}{\partial\xi}\,\mathrm{sgn}(\boldsymbol{\Lambda})\ ,\end{aligned}$$

où $\mathrm{sgn}(\boldsymbol{\Lambda}) = \mathrm{diag}(\mathrm{sgn}(\lambda_i))$, et la fonction signe est définie par $\mathrm{sgn}(x)$ vaut 1 si $x \leq 0$ et -1 sinon. Les équations stabilisées exprimées en variables caractéristiques sont encore diagonales :

$$\int_{\Omega^e} \mathbf{Z}^H j\omega\mathbf{W} + \mathbf{Z}^H_{,x}j\omega\frac{\partial x}{\partial\xi}\mathrm{sgn}\left(\boldsymbol{\Lambda}\right)\mathbf{W} + \mathbf{Z}^H\boldsymbol{\Lambda}\mathbf{W}_{,x} + \mathbf{Z}^H_{,x}\frac{\partial x}{\partial\xi}|\boldsymbol{\Lambda}|\mathbf{W}_{,x}d\Omega^e = \mathbf{0} \tag{6.32}$$

Les résultats d'intégration élémentaires utilisés dans la preuve du théorème 5.5 permettent de conclure. □

Remarque 6.3. Les deux termes apportés par la stabilisation ont chacun modifié la discrétisation pour passer de l'équation (6.28) à (6.30). Le terme convectif est parfaitement décentré, et le terme fréquentiel n'est plus symétrique.

Pour les deux discrétisations (6.28) et (6.30), la relation de récurrence de la deuxième variable caractéristique est étudiée. Une solution de la forme

$$w_i^{(2)} = \alpha_-^i w_0^- + \alpha_+^i w_0^+\,, \tag{6.33}$$

est recherchée, où α_- et α_+ sont les racines du polynôme caractéristique associé à la relation de récurrence.

Lemme 6.6. *Pour les éléments finis non stabilisés, les racines $\alpha_{\mathrm{Gal}+}$ et $\alpha_{\mathrm{Gal}-}$ sont données par*

$$\alpha_{\mathrm{Gal}\pm} = \frac{-2 \pm \sqrt{3 - 9\zeta^2}}{1 - 3j\zeta}\,, \tag{6.34}$$

où l'on a introduit le paramètre réduit $\zeta = \frac{u+c}{\omega h} = \frac{\lambda}{2\pi h}$, et λ est la longueur d'onde

Démonstration. On rappelle la relation de récurrence de la deuxième variable de (6.28), en ayant enlevé l'exposant 2 pour alléger les notations :

$$w_{i-1}\left(\frac{j\omega}{6}-\frac{u+c}{2h}\right)+w_i\frac{2j\omega}{3}+w_{i+1}\left(\frac{j\omega}{6}+\frac{u+c}{2h}\right)=0\,. \tag{6.35}$$

En utilisant le paramètre ζ, elle se réécrit

$$w_{i-1}\left(\frac{1}{4}+\frac{3}{4}j\zeta\right)+w_i+w_{i+1}\left(\frac{1}{4}-\frac{3}{4}j\zeta\right)=0\,. \tag{6.36}$$

Le polynôme caractéristique associé est

$$\left(\frac{1}{4}+\frac{3}{4}j\zeta\right)+X+X^2\left(\frac{1}{4}-\frac{3}{4}j\zeta\right)\,, \tag{6.37}$$

dont les racines sont les $\alpha_{\mathrm{Gal}\pm}$ recherchés. □

Lemme 6.7. *Pour les éléments finis stabilisés par la méthode SUPG, les racines $\alpha_{\mathrm{SUPG}+}$ et $\alpha_{\mathrm{SUPG}-}$ sont données par*

$$\alpha_{\mathrm{SUPG}\pm}=4-6j\zeta\pm\sqrt{21-36j\zeta-36\zeta^2}\,. \tag{6.38}$$

Démonstration. La relation de récurrence sur la deuxième variable caractéristique de (6.30) exprimée à l'aide de ζ conduit au polynôme quadratique

$$(-5-12j\zeta)+(-8+12j\zeta)\,X+X^2\,. \tag{6.39}$$

□

Lemme 6.8 (Propriétés des racines α)**.** *Dans la limite $\zeta\to\infty$, qui est celle d'une discrétisation infiniment fine on a*

$$\begin{array}{ll}\alpha_{\mathrm{Gal}+}\longrightarrow -1 & \alpha_{\mathrm{SUPG}+}\longrightarrow -j\infty\\ \alpha_{\mathrm{Gal}-}\longrightarrow 1 & \alpha_{\mathrm{SUPG}-}\longrightarrow 1\end{array} \tag{6.40}$$

De plus, pour $\zeta>1/\sqrt{3}$, $|\alpha_{\mathrm{Gal}+}|=|\alpha_{\mathrm{Gal}-}|=1$.
Enfin, la partie réelle de $\alpha_{\mathrm{Gal}+}$ est toujours négative.

Remarque 6.4. La définition de $\alpha_{\mathrm{SUPG}\pm}$ utilise la branche $z=\rho e^{j\theta}\mapsto\sqrt{\rho}e^{j\theta/2}e^{j\pi/2}$ de la fonction multiforme racine carrée, afin que les deux α_- tendent vers 1.

Remarque 6.5. Dans tous nos calculs, ζ est toujours supérieur à 1, car $\zeta=1$ correspond à environ 6 points par longueur d'onde, ce qui est trop peu pour bien discrétiser une onde avec des éléments linéaires.

Théorème 6.9. *La discrétisation par éléments finis non stabilisés conduit à des oscillations non physiques.*

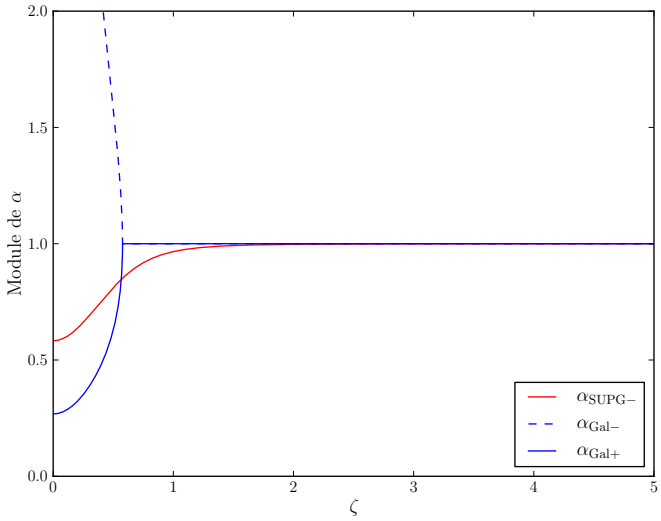


FIGURE 6.10 – Module des coefficients α introduits dans (6.34) et (6.38)

Démonstration. La solution est recherchée sous la forme donnée dans l'équation (6.33). La partie réelle de $\alpha_{\mathrm{Gal}+}$ est négative. La partie de la solution $\alpha^i_{\mathrm{Gal}+} w_0^+$ portée par ce terme sera oscillante d'un nœud à l'autre. C'est un comportement non physique, car la période de l'oscillation est alors fixée par le maillage, et elle existe même si $h \to 0$. □

Le module des différents coefficients de récurrence $\alpha_\pm$ est tracé dans la figure 6.10. Les figures 6.11 et 6.12 donnent leur partie réelle et leur chemin sur le plan complexe en fonction de ζ. Le coefficient $\alpha_{\mathrm{SUPG}+}$ n'est jamais présenté sur tous les graphes, car pour $\zeta = 0$, $|\alpha_{\mathrm{SUPG}+}| \approx 8{,}5$. De plus, son module croît avec ζ et un équivalent à l'infini est $|\alpha_{\mathrm{SUPG}+}| \sim 12\zeta$.

Ainsi, la partie réelle de $\alpha_{\mathrm{SUPG}-}$ est toujours positive, comme celle de $\alpha_{\mathrm{Gal}-}$, contrairement à $\alpha_{\mathrm{Gal}+}$ qui a une partie réelle négative. Notons également que $|\alpha_{\mathrm{SUPG}-}| < 1$, qui peut se voir graphiquement sur la figure 6.10, et est également prouvé pour les grands ζ par l'équivalent donné dans l'équation (6.44).

On peut maintenant montrer que la propriété de non réflexion d'une condition de Dirichlet homogène existe toujours lorsque l'on stabilise par la méthode SUPG les équations d'Euler linéarisées fréquentielles.

Théorème 6.10. *Une condition de Dirichlet homogène sur toutes les variables est transparente pour la caractéristique sortante : elle n'induit qu'une perturbation locale (couche limite exponentiellement décroissante) de cette variable caractéristique.*

La démonstration est reportée après celle du lemme 6.11 et permettra de clarifier l'énoncé du théorème.

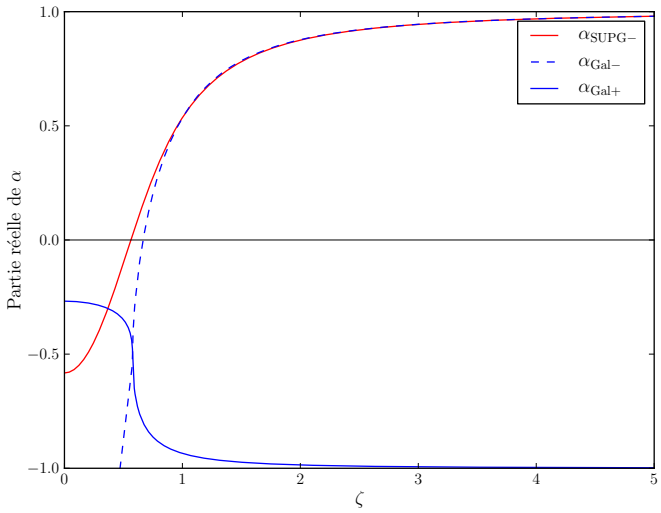


FIGURE 6.11 – Partie réelle des coefficients α introduits dans (6.34) et (6.38)

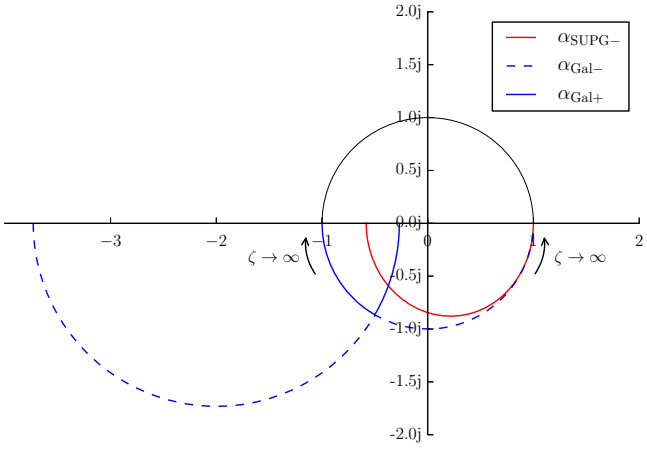


FIGURE 6.12 – Variation des coefficients α dans le plan complexe pour $\zeta > 0$

On discrétise les équations sur le même segment que dans le cas précédent sans terme temporel. On pourra consulter la figure 6.8. Soit un segment de longueur L, discrétisé en N éléments de longueur h. Les nœuds ainsi définis sont numérotés de 0 à N. Une condition de Dirichlet homogène sur toutes les variables est imposée au nœud N. Pour simplifier l'explication, on impose au nœud 0 une condition aux limites caractéristique qui fixe la seconde variable caractéristique à 1 et les autres à 0. Une condition en pression totale pourrait également être appliquée, mais alors elle couplerait les variables caractéristiques et les conditions aux limites., ce qui complique l'explication sans en changer la forme. Ce choix permet de continuer à n'étudier que la deuxième variable caractéristique.

La démonstration de la transparence des conditions aux limites demande le calcul de la solution.

Lemme 6.11. *Les coefficients w_0^+ et w_0^- définissant la solution discrète $w_i^{(2)} = \alpha_-^i w_0^- + \alpha_+^i w_0^+$ sont donnés par*

$$\begin{cases} w_0^+ = \dfrac{\alpha_-^N}{\alpha_-^N - \alpha_+^N} \\ w_0^- = \dfrac{\alpha_+^N}{\alpha_+^N - \alpha_-^N}\,, \end{cases} \tag{6.41}$$

et valables pour les deux discrétisations proposées

Démonstration. Les conditions aux limites du problème sur la seconde variable sont donc :

$$\begin{cases} w_0 = 1 \\ w_N = 0\,. \end{cases} \tag{6.42}$$

On réinjecte la forme du terme de récurrence $w_i = \alpha_-^i w_0^- + \alpha_+^i w_0^+$:

$$\begin{cases} w_0^- + w_0^+ = 1 \\ \alpha_-^N w_0^- + \alpha_+^N w_0^+ = 0\,. \end{cases}$$

Si $\alpha_+^N \neq \alpha_-^N$, on retrouve l'initialisation demandée.

Dans le cas du SUPG, les équivalents de $|\alpha_{\text{SUPG}-}|$ et $|\alpha_{\text{SUPG}-+}|$ pour ζ grand montrent que $|\alpha_{\text{SUPG}-}| < |\alpha_{\text{SUPG}-+}|$. Une étude numérique permet d'obtenir les mêmes conclusions pour tout ζ. Ainsi, $\alpha_{\text{SUPG}-}^N \neq \alpha_{\text{SUPG}+}^N$.

Pour la discrétisation par éléments finis de Galerkin, $\alpha_{\text{Gal}+}$ et $\alpha_{\text{Gal}-}$ convergent respectivement vers -1 et 1. L'utilisation d'équivalents permet d'écrire

$$\alpha_{\text{Gal}+}^N - \alpha_{\text{Gal}-}^N = (-1)^N - 1 + j\frac{N}{\zeta}\left(\frac{(-1)^N}{3} - 1\right) + o\left(\frac{1}{\zeta^2}\right)\,. \tag{6.43}$$

et ainsi de montrer que $\alpha^N_{\text{Gal}+} - \alpha^N_{\text{Gal}-}$ n'est jamais nul, mais est hautement dépendant de la parité de N. □

Preuve du théorème 6.10. D'après les équivalents fournis en résultats du lemme 6.8, on constate que $|w_0^+| \ll 1$ et que $|w_0^-| \sim 1$.

Ainsi, la solution SUPG est seulement portée par la racine $\alpha_{\text{SUPG}-}$ de module proche de 1. La composante divergente apportée par $\alpha_{\text{SUPG}+}$ n'est présente que pour annuler la solution tout près de la condition limite en N. Ainsi, $|\alpha^i_{\text{SUPG}-} w_0^-| = O(1)$. L'autre morceau de la solution $(\alpha^i_{\text{SUPG}+} w_0^+)$ crée une couche limite exponentiellement décroissante proche de la condition de Dirichlet, et est donc négligeable à l'intérieur du domaine. □

Remarque 6.6. Pour le schéma Galerkin standard, on rappelle que $\alpha_{\text{Gal}+} \to -1$, $\alpha_{\text{Gal}-} \to 1$ et que $\alpha^N_{\text{Gal}+} - \alpha^N_{\text{Gal}-} = (-1)^N - 1 + j\frac{N}{\zeta}\left(\frac{(-1)^N}{3} - 1\right) + o\left(\frac{1}{\zeta^2}\right)$. Ainsi, $|w_0^+| \approx |w_0^-|$. Cela implique que la solution est plus ou moins également distribuée sur une composante oscillante (due à $\alpha_{\text{Gal}+}$) et une composante sinusoïdale (due à $\alpha_{\text{Gal}-}$). En d'autres termes, l'amplitude du signal physique propagé par $\alpha_{\text{Gal}-}$ n'est pas l'amplitude réelle de l'onde de pression, et le bruit numérique est du même ordre de grandeur que le signal réel.

Corollaire 6.12. *La diffusion et la dispersion du schéma SUPG peuvent être étudiées par l'analyse de $\alpha_{\text{SUPG}-}$. La diffusion du schéma est liée au module de $\alpha_{\text{SUPG}-}$, et la dispersion à son argument. En utilisant les notations de Landau, pour $\zeta \to +\infty$,*

$$\begin{aligned} |\alpha_{\text{SUPG}-}| &= 1 - \frac{1}{24\zeta^4} + o\left(\frac{1}{\zeta^5}\right) \\ \arg(\alpha_{\text{SUPG}-}) &= -\frac{1}{\zeta} - \frac{11}{720\zeta^5} + o\left(\frac{1}{\zeta^6}\right) \end{aligned} \tag{6.44}$$

Preuve et commentaires. La partie physique de la solution est portée par le terme $w_0^- \alpha^i_{\text{SUPG}-}$ au point i. La viscosité numérique du schéma atténue l'amplitude de l'onde au cours de sa propagation. L'écart du module de $\alpha_{\text{SUPG}-}$ à 1 permet de quantifier cette viscosité.

La dispersion du schéma qualifie l'écart de la phase calculée de l'onde avec la phase théorique. Elle s'évalue avec l'argument de $\alpha_{\text{SUPG}-}$. Il est normal que le premier terme de l'équivalent de $\arg(\alpha_{\text{SUPG}-})$ soit $-1/\zeta$. En effet, rappelons que $\zeta = \frac{\lambda}{2\pi h}$, et après $i \approx \lambda/h$ nœuds, l'argument de w_i doit diminuer de 2π. □

Remarque 6.7. Le schéma SUPG est légèrement diffusif et dispersif. Il est intéressant de noter qu'il n'y a pas de terme en $1/\zeta^2$ dans le module de $\alpha_{\text{SUPG}-}$, ni de terme en $1/\zeta^3$ pour son argument, alors qu'ils sont attendus dans un tel développement. Cela montre la haute précision du schéma stabilisé par la méthode SUPG.

6.2.3 Résultats en 2D et en 3D

Dans la section précédente, on a prouvé la transparence d'une condition aux limites de Dirichlet homogène. Le cœur de cette preuve tient dans le décentrement parfait des dérivées spatiales des variables caractéristiques. Cela est possible car le système d'équations exprimé dans les variables caractéristiques est diagonal. Autrement dit, il existe une base qui diagonalise l'opérateur de convection et la matrice de stabilisation.

Le passage au cas multidimensionnel rend impossible cette preuve, notamment à cause de la perte de la dernière propriété. En effet, il n'existe pas de base diagonalisant les trois opérateurs de convection d'Euler simultanément, comme on l'a vu dans la section 5.1.5. Il est donc impossible de réduire les équations stabilisées à un système diagonal. La stabilisation n'est donc pas parfaite dans toutes les directions.

Un deuxième problème rencontré lors du passage à plusieurs dimensions est la notion de taille d'élément. En 1D, le terme de stabilisation est un terme diffusif multiplié par la longueur de l'élément considéré. Cela amène des annulations avec la discrétisation du terme de convection, et permet donc un décentrement parfait. En plusieurs dimensions, la définition de la longueur de l'élément n'est plus très claire, et ne permet pas des annulations parfaites comme en 1D.

Même si la démonstration formelle de la transparence des conditions de Dirichlet homogènes n'est plus possible, il est intéressant de caractériser les imperfections de ces conditions aux limites transparentes. Pour ce faire, une expérience numérique très simple a été mise en place. Elle consiste en la propagation acoustique en deux dimensions d'une source ponctuelle monopolaire dans un domaine rectangulaire. Comme la puissance acoustique est conservée sur tout cercle de rayon r centré sur la source, l'intensité acoustique est proportionnelle à $1/r$. L'intensité acoustique pour des ondes circulaires étant proportionnelle au carré de la pression acoustique, on a $p \propto 1/\sqrt{r}$.

La figure 6.14 montre le module de la pression corrigé de l'atténuation géométrique en 2D, sur deux domaines rectangulaires tel qu'expliqué par la figure 6.13. Le premier calcul a été réalisé sur un petit domaine rectangulaire, où le monopôle acoustique était placé en son centre. Sur les bords du rectangle, des conditions aux limites de Dirichlet homogènes jouent le rôle de conditions transparentes. Seule la moitié supérieure du domaine est présentée sur la figure 6.14. On y voit clairement des motifs d'interférences, qui prouvent que de l'énergie est bien réfléchie à l'intérieur du domaine.

Ce premier calcul a servi à obtenir une référence, à partir de laquelle les différences dues à la modification du domaine peuvent être analysées. Le domaine a été agrandi par une extension symétrique à gauche et à droite. Les résultats sur ce grand domaine de calcul sont représentés en-dessous et sur les côtés de la moitié supérieure du petit domaine sur la figure 6.14.

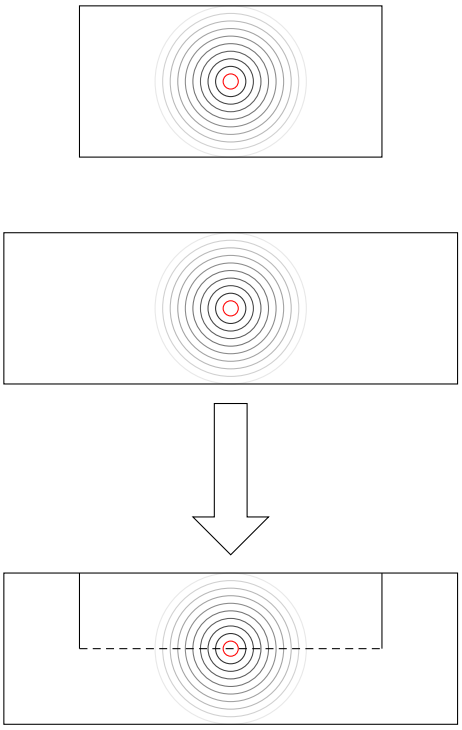


FIGURE 6.13 – Explication de la figure 6.14. Un calcul est réalisé sur un petit domaine, un deuxième sur un domaine plus grand en largeur. Les images des résultats sur les deux domaines sont superposées, en ne montrant que la partie supérieure du petit domaine.

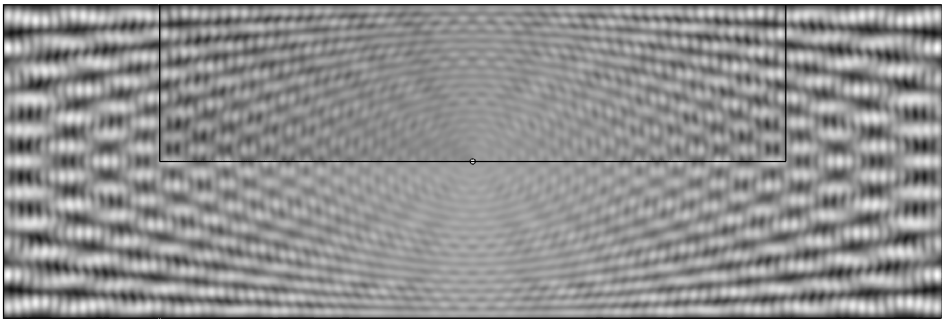


FIGURE 6.14 – Superposition de deux domaines de calcul rectangulaires contenant un monopôle acoustique en leur centre et des conditions limites transparentes à leur bord. Le champ tracé est la pression corrigée pour l'atténuation géométrique en 2D $\sqrt{r}|p|$. La moitié du plus petit domaine est superposée sur le plus grand (voir la figure 6.13).

Premièrement, on remarque que les franges d'interférences sont presque identiques sur le petit et le grand rectangle. On en déduit que ce ne sont pas les bords latéraux qui reflètent le plus d'énergie. Cela permet également de mettre hors de cause les coins. Cette observation est importante, car les coins sont délicats à gérer lors de l'implémentation classique de conditions aux limites transparentes par la méthode des caractéristiques, puisqu'il est impossible de définir proprement la normale à la surface dans les coins.

Ce sont donc les parois horizontales qui sont principalement responsables des réflexions. On note que les franges d'interférences deviennent de plus en plus fortes au fur et à mesure que l'on s'éloigne latéralement de la source. Cela montre que lorsque l'angle de l'onde incidente avec la normale de la paroi augmente, les ondes acoustiques sont plus réfléchies. Cela est attendu pour des conditions aux limites caractéristiques [157].

Aucune expérience numérique de la sorte n'a été menée pour les cas tridimensionnels. On se bornera à dire que des calculs sur des configurations industrielles où les bords du domaine sont extrêmement proches de la source ont été menés, sans que des réflexions indésirables trop significatives ne se produisent . La figure 6.15 montre, sur le cas test industriel présenté dans la section 6.3, la partie réelle de la variation de pression dans une coupe effectuée dans le domaine tridimensionnel de calcul. Ce domaine de calcul possède un bord cylindrique, autour de l'axe de révolution $y = 0$, $z = 0$. Les bords de l'image montrent l'étendue réelle du domaine de calcul. Les bords représentant l'infini sont donc très proches de la tuyère. Des interférences sont visibles, mais ne changent pas radicalement la solution en champ proche. Les conditions aux limites de Dirichlet homogènes, qui se sont révélées transparentes ont ainsi permis de radicalement diminuer la taille du domaine de calcul et ainsi le temps de calcul, sans devoir recourir à des conditions aux limites plus complexes [157].

6.3 Résultats industriels d'aéroacoustique

6.3.1 Présentation de l'étude SFWA

Le code AeTher linéarisé a été utilisé sur un cas test industriel tiré du projet européen *Clean Sky* [34]. Le projet Clean Sky vise à développer des briques technologiques innovantes pour la conception d'avions plus respectueux de l'environnement, qui consommeraient moins de carburant et seraient plus silencieux que les avions actuels. Dans le cadre de SFWA (*Smart Fixed Wing Aircraft*), l'une des six plateformes du projet Clean Sky, Dassault Aviation a travaillé sur un concept innovant d'arrière corps doté d'une gouverne de profondeur en U qui masque le bruit des réacteurs [129]. Une maquette de ce concept, munie de turbines propulsées à l'air comprimé permettant de simuler un réacteur a été testée en soufflerie. L'intérêt de ces turbine est double. Ce sont de véritables modèles réduits de réacteurs (où l'apport d'énergie par

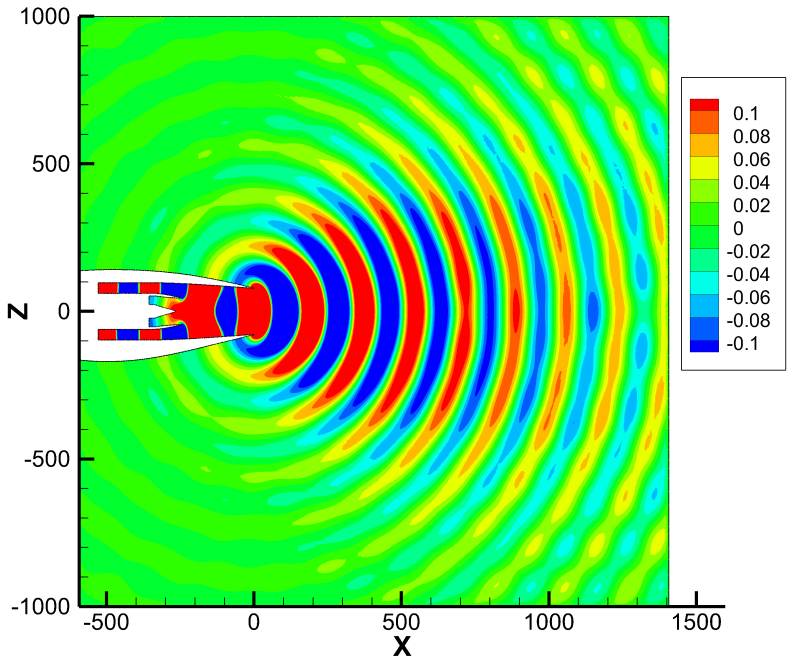


FIGURE 6.15 – Partie réelle de la variation de pression en Pascal pour un mode plan à 2 kHz sans écoulement. Coupe effectuée dans le maillage 3D.

FIGURE 6.16 – Maquette d'un avion avec PH en U en soufflerie. Crédit photographique : Dassault Aviation – Clean Sky 1 – SFWA – DNW

de l'air comprimé remplace la combustion). Ainsi, elles produisent un bruit similaire à celui d'un réacteur. De plus, elles génèrent un jet, qui réfracte ce bruit. Ces essais ont permis de quantifier la réduction de bruit apportée par cet empennage en U par rapport à un empennage en croix typique.

Les calculs présentés dans cette partie ont tous été réalisé sur le modèle réduit de réacteur. Ce cas test est dénommé SFWA. Il est à l'échelle 1/4,7. Certaines parties ne sont cependant pas tout à fait à l'échelle. Le conduit annulaire est plus fin dans le modèle réduit. Le pylône de support est plus gros, pour le passage des tubes amenant l'air comprimé nécessaire au fonctionnement de la turbine. Une image de la géométrie de la tuyère du modèle réduit de réacteur ayant servi au calcul est présentée sur la figure 6.17. À fin de référence, le diamètre de la tuyère est de 20 cm environ.

Pour minimiser l'influence du maillage sur les résultats, le maillage volumique a été délibérément choisi très fin. Il est conçu pour des calculs de propagation acoustique à des fréquences allant jusqu'à 9 kHz, même si aucun calcul durant cette thèse n'a été réalisé à cette fréquence. Comme le conduit annulaire est très fin, les modes d'ordres élevés, qui ont un nombre d'onde radial très haut, nécessitent une discrétisation très fine de l'espace dans le plan modal et l'intérieur de la tuyère. À partir de la sortie de la tuyère, la taille des éléments augmente progressivement pour atteindre la longueur limite de propagation. Près de la frontière infinie, une augmentation de la taille des éléments permet d'atténuer un petit peu plus les réflexions sur la condition aux limites.

6.3.2 Résultats sans écoulement

Les premiers calculs ont été effectués sans écoulement. Dans ce cas-là, la propagation acoustique suit les équations d'Helmholtz. Un code interne à

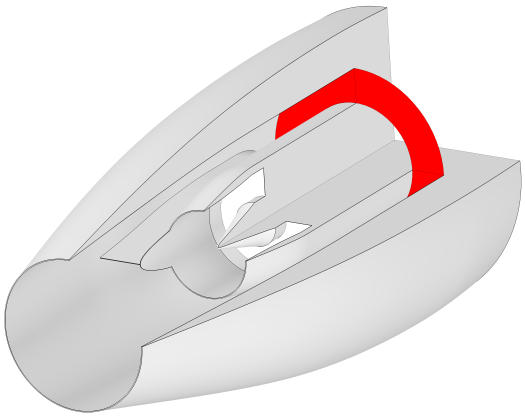


FIGURE 6.17 – Géométrie de la tuyère du réacteur utilisée pour le calcul, avec une découpe pour montrer l'intérieur. Le disque rouge est le plan modal où sont injectés les modes acoustiques.

Dassault Aviation qui résout ces équations par une méthode d'élément de frontière (BEM pour *Boundary Element Method*) [140] a servi de point de comparaison.

Mode plan à 2 kHz

Tout d'abord, une onde plane à 2 kHz a été injectée dans le plan modal. Ce cas plus simple que les autres réalisés par la suite a permis une première évaluation de la performance du code AeTher linéarisé pour la propagation acoustique. La directivité en champ lointain de ce mode est calculée par une méthode intégrale à base de fonction de Green, à partir de données du calcul volumique interpolées sur une surface dite de Kirchhoff située à une certaine distance de la tuyère. La comparaison de la directivité obtenue par la méthode à élément de frontière et le code AeTher est présentée sur la figure 6.18. L'angle de directivité est relatif à l'axe de la tuyère. Un angle de 0° indique que le point considéré est dans le jet du moteur, tandis qu'un point devant l'entrée d'air du réacteur aura un angle de 180°. On constate que les deux méthodes donnent des résultats très proches. Dans le lobe principal, elles diffèrent de moins d'un décibel. Pour les points écartés de plus de 120° de l'axe, les écarts entre les méthodes s'expliquent principalement par l'erreur de diffraction due à l'ouverture vers l'amont de la surface d'interpolation de Kirchhoff.

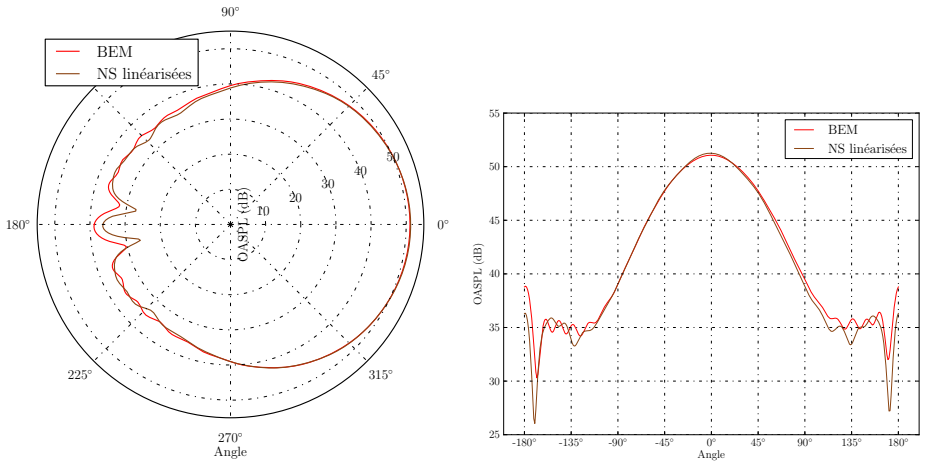


FIGURE 6.18 – Directivité en champ lointain pour le mode plan à 2 kHz. Comparaison entre Navier-Stokes linéarisé et le code BEM.

Modes d'ordre élevé à 2 kHz et 5 kHz

Comme on l'a expliqué dans la section 6.1.3, d'autres modes qu'une onde plane peuvent se propager dans un tube. Leur propagation est plus difficile à calculer, et leur directivité plus complexe. Le mode (1,1) (pour $(n = 1, m = 1)$) à 2 kHz a été calculé en premier. La variation de pression à imposer dans le plan d'entrée est tracée sur la figure 6.19. La directivité de ce mode est présentée sur la figure 6.20. La directivité issue du calcul Navier-Stokes linéarisé est à moins d'un décibel sur les deux lobes du calcul par éléments de frontière. La seule différence majeure entre les deux méthodes se situe dans la capture du creux d'interférence entre les deux lobes. Le code Navier-Stokes linéarisé ne permet pas d'obtenir une interférence aussi nette que l'approche BEM. Cela pourrait venir de la diffusion numérique dans le volume, ou encore d'un système linéaire pas assez convergé, ou enfin de l'effet de la diffraction due à l'ouverture de la surface d'interpolation vers l'amont. Le code Navier-Stokes arrive quand même à résoudre 20 dB d'écart (ou un rapport 10 en amplitude de pression) entre le maximum des lobes et le minimum dans le creux.

Les plus hautes fréquences sont plus difficiles à calculer. La directivité du mode (1,1) à 5 kHz est très différente de celle à 2 kHz. La figure 6.21 montre la directivité de ce mode à cette fréquence. Les résultats d'AeTher commencent à s'éloigner de ceux de l'approche BEM. Le code AeTher ne donne pas une image parfaite des deux lobes principaux, et ne capture presque pas le lobe secondaire autour de -70°. Les principales caractéristiques de la directivité sont cependant transcrites. La figure 6.22 montre la partie réelle de la variation de pression sur une coupe du volume pour ce mode. Elle permet de se rendre compte de la complexité de la propagation à l'intérieur

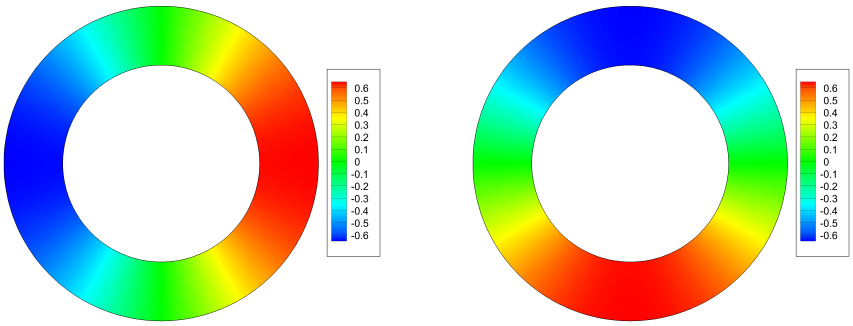


FIGURE 6.19 – Parties réelle (gauche) et imaginaire (droite) de la variation de pression en Pascal d'un mode (1,1) imposé dans le plan d'entrée

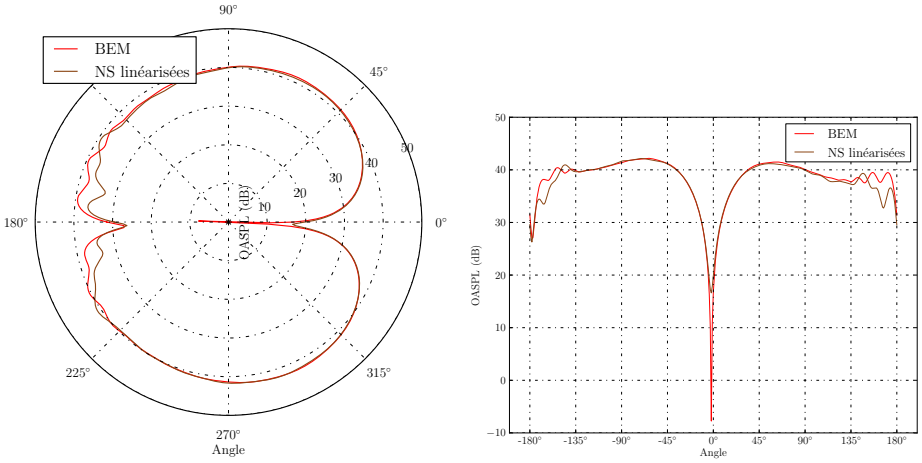


FIGURE 6.20 – Directivité en champ lointain d'un mode (1,1) à 2 kHz sans écoulement. Comparaison entre Navier-Stokes linéarisé et approche Helmholtz BEM.

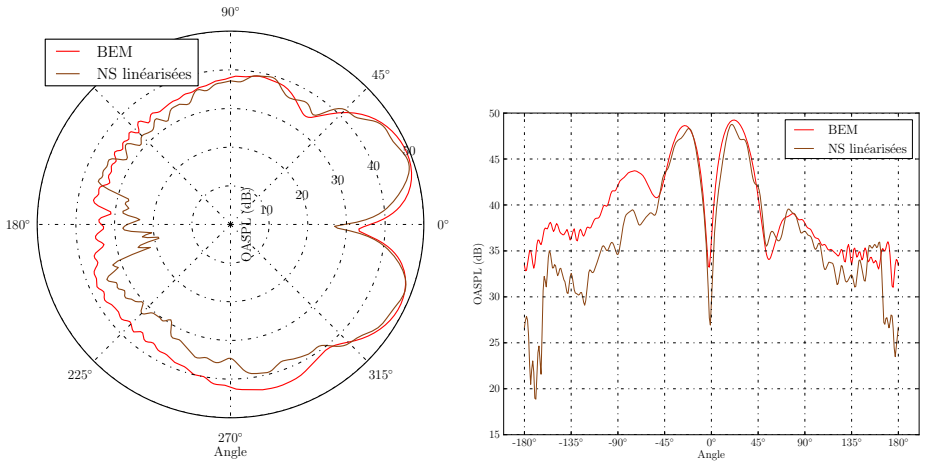


FIGURE 6.21 – Directivité en champ lointain d'un mode (1,1) à 5 kHz sans écoulement. Comparaison entre Navier-Stokes linéarisé et approche Helmholtz BEM.

de la tuyère, ainsi que des franges d'interférences apparaissant plus loin de la sortie.

6.3.3 Résultats avec écoulement

La propagation du son change considérablement en la présence d'un écoulement moyen. La réfraction des ondes sonores par les gradients de vitesse et de température altère la direction de propagation. Les ondes sonores sont réfractées vers l'extérieur du jet.

Le champ moyen pour ce cas a été calculé avec une méthode RANS. Comme on l'a indiqué dans la section 6.3.1, le moteur est un modèle réduit. Pour pouvoir fonctionner, il est alimenté en air comprimé à basse température. La détente dans la turbine refroidit l'air du jet autour de -100°C. La vitesse du son est donc très basse dans le jet, qui est donc localement supersonique. La figure 6.23 montre le nombre de Mach local de l'écoulement.

Un calcul de propagation acoustique a été effectué autour de cet écoulement moyen, pour un mode plan à 2 kHz. La partie réelle de la variation de pression est montrée sur la figure 6.24. On constate que les ondes sonores sont réfractées vers l'extérieur du jet, par rapport au cas sans écoulement présenté sur la figure 6.15. Les oscillations de forte amplitude qui matérialisent le jet correspondent à des instabilités de Kelvin-Helmholtz qui sont amplifiées à cette fréquence au cours de leur convection. Ces instabilités sont présentes dans la couche de cisaillement, et elles créent une onde de forte amplitude qui ne rayonne pas à la fréquence du calcul. Au fur et à mesure que le jet se mélange dans l'air ambiant, les couches de cisaillement deviennent plus épaisses, et les ondes de Kelvin-Helmholtz sont à nouveau stables et

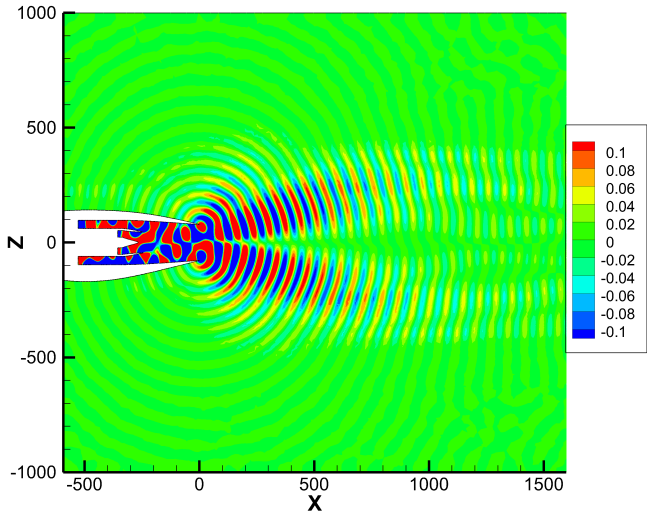


FIGURE 6.22 – Partie réelle de la variation de pression en Pascal pour le mode (1,1) à 5 kHz sans écoulement dans une coupe du maillage.

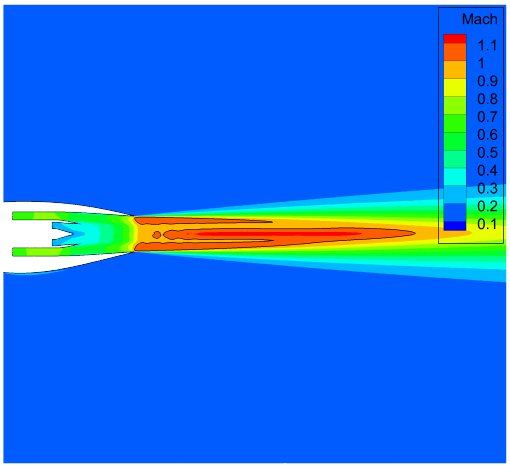


FIGURE 6.23 – Nombre de Mach local de l'écoulement produit par le modèle réduit de réacteur. Il simule alors un réacteur taille réelle à pleine poussée

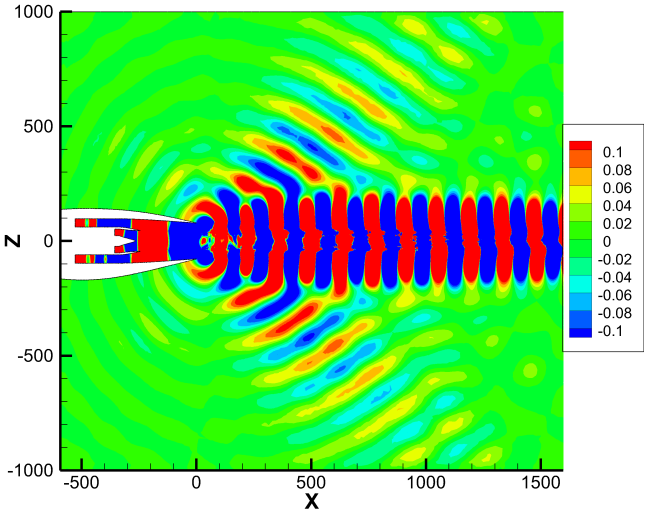


FIGURE 6.24 – Partie réelle de la variation de pression en Pascal pour un mode plan à 2 kHz. L'écoulement moyen correspond à la pleine puissance du réacteur

décroissent exponentiellement. Les conditions aux limites à l'infini n'étant pas parfaitement transparentes, il faut s'assurer que ces ondes d'instabilités sont suffisamment amorties avant d'atteindre le bord. Cette contrainte a été dimensionnante pour déterminer l'étendue du domaine de calcul.

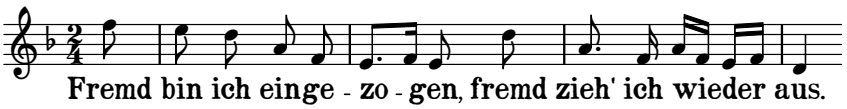
6.4 Conclusion

Les variables entropiques sont utilisées dans AeTher pour la résolution des équations de Navier-Stokes afin d'obtenir des propriétés intéressantes de symétrie d'opérateur et de thermodynamique. Ces variables sont cependant éloignées des variables naturelles de l'aérodynamique. L'imposition des conditions aux limites s'en trouve sérieusement complexifiée. En se plaçant dans le cadre proposé par Shakib dans [145] pour les conditions de Dirichlet homogènes, cette thèse a étendu ce cadre pour permettre l'imposition de conditions aux limites de Dirichlet non homogènes pour des variables non triviales du calcul. La linéarisation de la fonction non-linéaire reliant la variable d'intérêt aux variables entropiques fournit les coefficients nécessaires à l'élimination de la variable par laquelle on impose la condition aux limites.

Les conditions aux limites de Dirichlet non homogènes ont permis l'utilisation industrielle en aéroacoustique du code AeTher linéarisé. Des modes acoustiques complexes peuvent être injectés dans le calcul, avec un contrôle précis de leur amplitude. La comparaison des résultats avec ceux donnés

par une approche BEM sur un cas test industriel confirme le bien-fondé des conditions aux limites. Pour l'aéroacoustique, deux façons d'imposer le mode en entrée ont été proposées. La première est d'imposer l'amplitude totale du mode au niveau du plan d'entrée, ce qui a pour inconvénient de ne pas permettre de contrôler l'énergie injectée réellement dans le système. Pour pallier ce problème, on a proposé d'imposer la variable caractéristique entrante, en laissant libres les autres variables caractéristiques. Ainsi, seule l'onde entrante est imposée. Cela a été confirmé en imposant une onde plane dans une cavité résonnante. Les conditions aux limites utilisant les variables caractéristiques sont moins performantes lorsque l'onde incidente n'est plus normale à la paroi [157]. Afin de vérifier le comportement angulaire des conditions aux limites caractéristiques, une étude sur une cavité résonnante pour des modes d'ordre élevé a montré comme attendu qu'elles se dégradaient avec l'angle, et a également donné un critère utile à l'ingénieur pour connaître la validité de son calcul.

Dans ce chapitre, le comportement surprenant des conditions aux limites de Dirichlet homogènes sur toutes les variables a été éclairci. Elles jouent le rôle de conditions aux limites transparentes dans les calculs d'aéroacoustique sans traitement particulier. Un modèle 1D a permis de montrer que la discrétisation SUPG telle qu'implémentée dans le code AeTher décentre chacune des variables caractéristiques. Lorsqu'on ne considère que le terme advectif des équations d'Euler, l'explication est simplifiée et tient uniquement au décentrement des variables caractéristiques. L'ajout du terme fréquentiel complique la démonstration, qui fournit deux résultats connexes intéressants. Le premier est l'étude théorique de la diffusion et de la dispersion du schéma SUPG, qui sont très favorables. Enfin, on a prouvé que la discrétisation des équations d'Euler linéarisées fréquentielles par la méthode des éléments finis stabilisés par SUPG n'est pas stable. L'utilisation de conditions de Dirichlet homogènes à l'infini sur toutes les variables sélectionne la partie stable du schéma, tandis que la partie croissante assure l'annulation localisée de la caractéristique sortante au bord infini. La construction de la matrice de stabilisation ne permet pas une extension parfaite de ce résultat si l'espace est à plusieurs dimensions. Malgré cela, une étude numérique en 2D montre qu'elles reflètent davantage les ondes arrivant à grande incidence sur la surface, comme attendu d'une condition aux limites caractéristique. Enfin, leur utilisation pratique sur des cas industriels 3D donne entière satisfaction, car ces conditions aux limites permettent de réduire significativement la taille du domaine de calcul sans provoquer trop de réflexions parasites nuisibles à la qualité des résultats.



Schubert, *Winterreise*, Gute Nacht

Chapitre 7

Conclusion et perspectives

Cette thèse a eu pour objet d'améliorer la résolution des équations de Navier-Stokes linéarisées, pour l'optimisation de forme, l'aéroélasticité et l'aéroacoustique. Si l'approche Navier-Stokes linéarisée était déjà fonctionnelle chez Dassault Aviation, son utilisation restait peu compatible d'une application industrielle dans un cycle de conception, tant sur le plan de la rapidité que de la robustesse. Pour résoudre ce problème, deux angles d'approche se sont naturellement dégagés. Le premier, purement numérique, s'intéresse uniquement à la résolution du système linéaire. Le deuxième est la formulation du schéma conduisant à ce système linéaire.

La résolution des systèmes linéaires dans AeTher se fait par l'algorithme GMRES. Les méthodes itératives de résolution de systèmes, dont fait partie GMRES, sont peu gourmandes en mémoire et facilement parallélisables. Elles sont donc prisées pour la résolution de très grand systèmes. Leur principal inconvénient est leur convergence qui peut être lente ou alors jamais atteinte. Ce désavantage peut être contré par l'utilisation d'un préconditionneur, qui transforme le système linéaire en un équivalent plus simple numériquement à résoudre par une méthode itérative. La partie numérique de cette thèse s'est intéressée à l'algorithme GMRES et au préconditionnement du système linéaire à résoudre.

Les campagnes de calcul d'aéroélasticité demandent de nombreux calculs correspondant à divers modes structuraux, à plusieurs fréquences et points de vol. Les systèmes linéaires correspondant à des modes calculés à la même fréquence et au même point de vol ont une même matrice, mais des seconds membres différents. Ainsi, il semble intéressant de grouper la résolution de ces problèmes. Dans le chapitre 3, l'algorithme GMRES avec déflation des plus petites valeurs propres pour améliorer le redémarrage a été présenté en détails. Dans cette thèse, l'extension de cette méthode à plusieurs seconds membres a été implémentée et testée. Cet algorithme, dit block-GMRES, permet a priori d'exploiter la similitude entre les seconds membres

et d'explorer un espace de Krylov plus grand. De plus, le regroupement des opérations informatiques accélère l'exécution de celles-ci. Les tests numériques ont montré que la méthode block-GMRES ne semble pas adaptée à notre utilisation. L'accélération de la résolution par la richesse de l'espace de Krylov de taille accrue ne s'est pas produite. De plus, l'augmentation de la taille de l'espace a ralenti considérablement les itérations. L'algorithme block-GMRES n'a donc pas été retenu.

Le second chapitre du volet numérique de cette thèse s'est intéressé au préconditionnement, nécessaire au bon fonctionnement d'un solveur itératif. La parallélisation du préconditionnement est réalisée par la méthode de Schwarz additif. Elle consiste à appliquer le préconditionnement par sous-domaines, de manière locale. Cette méthode découple partiellement les sous-domaines de parallélisation. Lorsque l'on augmente le découpage du problème, la qualité du préconditionnement peut alors se dégrader. Pour pallier ce problème, les méthodes de Schwarz à deux niveaux introduisent un espace grossier, partagé par tous les sous-domaines, et leur permet de communiquer globalement l'information. Deux espaces grossiers, l'un simple et l'autre complexe, ont été testés, mais n'ont pas donné satisfaction. Finalement, l'utilisation de la factorisation incomplète avec remplissage ILU(k) comme préconditionneur local a permis d'accélérer les résolutions jusqu'à un facteur dix. L'utilisation du préconditionnement ILU(1) est désormais standard dans AeTher linéarisé. Il a également permis de faire converger des cas jusqu'alors impossibles à résoudre.

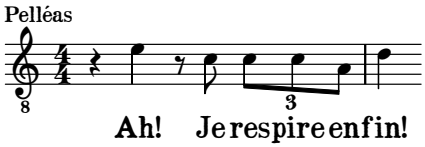
Le deuxième axe de cette thèse est la formulation du schéma numérique. AeTher utilise des éléments finis stabilisés par la méthode SUPG. Au cœur de cette méthode se trouve la matrice $\boldsymbol{\tau}$ de stabilisation, dont la construction fait l'objet du premier chapitre de cette partie. La démonstration de la construction de la matrice de stabilisation s'est appuyée sur des cas de plus en plus complexes, ce qui permet de mieux comprendre la formule de $\boldsymbol{\tau}$ retenue. Ensuite, une matrice de stabilisation suivant littéralement cette expression a été testée, sans les approximations retenues pour la matrice $\boldsymbol{\tau}$ originale. Pour les calculs non-linéaires, cette forme de matrice s'est révélée moins visqueuse, donc moins stabilisante. Les systèmes linéaires des problèmes implicites à chaque pseudo pas de temps sont plus faciles à résoudre. Enfin, la répartition de pression sur le fuselage de l'avion est plus satisfaisante avec cette matrice. L'utilisation de cette forme de matrice $\boldsymbol{\tau}$ en linéarisé a montré les mêmes avantages, à l'exception de la vitesse de résolution. Sur une application d'aéroacoustique, l'utilisation de la forme complète de la matrice de stabilisation a montré une dégradation de la transparence des conditions aux limites de Dirichlet homogènes, ce qui montre que cette forme n'est pas encore complètement comprise. On peut donc conclure que l'application littérale de la formule de $\boldsymbol{\tau}$ ne donne pas toujours une matrice de stabilisation meilleure que les approximations choisies historiquement dans AeTher pour sa construction.

L'étude des conditions aux limites de Dirichlet conclut cette partie sur la formulation. Tout d'abord, l'application de conditions de Dirichlet non homogènes à des variables non triviales du calcul est détaillée. Le code AeTher utilise en effet des variables entropiques pour exprimer les équations de Navier-Stokes. Les variables naturelles pour imposer les conditions aux limites du calcul dépendent non linéairement des variables entropiques. La différenciation des formules de dépendance permet d'obtenir une relation linéaire entre les variables linéarisées nécessaires pour imposer les conditions aux limites de Dirichlet. Cette thèse a introduit la méthode pour utiliser des conditions de Dirichlet non homogènes. L'imposition d'une variation de pression a rendu possibles les calculs aéroacoustiques avec un contrôle précis du signal sur le plan modal. Des conditions caractéristiques ont également été implémentées, et le comportement angulaire de celles-ci a été testé rigoureusement dans une cavité cylindrique résonante. Dans un deuxième temps, la propriété surprenante de transparence des conditions de Dirichlet homogènes sur toutes les variables a été élucidée. Un modèle 1D montre que c'est le décentrement parfait par caractéristiques apporté par la stabilisation SUPG qui est à l'origine de ce comportement. En 2D ou 3D, cette propriété ne se dégrade pas trop. Ainsi, les calculs sur des cas industriels peuvent être réalisés sur des domaines petits, limitant la taille du maillage et le temps de calcul.

En conclusion, cette thèse a permis plusieurs avancées dans l'utilisation industrielle des équations de Navier-Stokes linéarisées. La factorisation incomplète ILU(k) a divisé le temps de calcul d'aéroélasticité par un facteur de 5 à 10. Une campagne complète de calculs de flottement est désormais envisageable en un temps raisonnable. La mise en place des conditions aux limites de Dirichlet non homogènes a permis d'utiliser le code AeTher linéarisé pour la propagation acoustique. Sur le plan théorique, cette thèse a apporté l'explication de la transparence de simples conditions de Dirichlet homogènes sur toutes les variables. Enfin, cette thèse a confirmé que le solveur itératif utilisé doit être précis et robuste vis à vis du conditionnement de la matrice. L'algorithme GMRES avec une déflation des petites valeurs propres bien implémentée donne entière satisfaction. Cela signifie en creux que tout changement radical de solveur pour cette application nécessitera du temps pour le perfectionner et le rendre robuste. En clair, l'utilisation de solveur « boîte noire » paraît difficile pour AeTher linéarisé. De plus, on a montré que la forme de la matrice de stabilisation utilisée dans AeTher est cruciale aux bonnes propriétés du code.

Le travail sur la résolution des équations de Navier-Stokes linéarisées est loin d'être terminé. Le champ des possibles est encore vaste, même si cette thèse a ouvert des voies, et montré que d'autres ne sont pas viables. Des idées nouvelles restent à explorer, tant sur la résolution des systèmes linéaires que sur les schémas. Si le couple solveur itératif et préconditionneur fonctionne bien actuellement, rien ne présage du son bon fonctionnement

dans quelques années quand la taille des problèmes (et des maillages) et celle des calculateurs aura été multipliée par dix. Tout n'a pas été testé en décomposition de domaines. Les méthodes de sous-structuration, ou les conditions de Schwarz optimisées sont attrayantes, mais n'ont pu être testées faute de temps. La méthode AMG (pour *Algebraic MultiGrid*) [152], qui est une version purement algébrique des méthodes multigrilles [72, 108] et peut être utilisée pour résoudre un système linéaire ou comme préconditionneur, n'a pu être testée faute de temps. Elle a été utilisée avec succès pour la résolution des équations de Navier-Stokes discrétisées par des éléments finis stabilisés par SUPG [122]. L'influence négative de la renumérotation sur la qualité de préconditionneur n'a pas été expliquée. Une renumérotation inspirée du préconditionnement *line-implicit* [123, 109, 110, 122] couplée au préconditionnement ILU(k) pourrait peut-être améliorer la convergence. D'autres types de conditions aux limites transparentes existent, comme les opérateurs DtN [74], et pourraient être testés en aéroacoustique, tant pour les conditions à l'infini que pour imposer un mode incident [127].



Debussy, *Pelléas et Mélisande*, Acte 3 sc. 3

Annexe A

Calcul des valeurs de Ritz harmoniques

A.1 Simplification du problème

Le calcul des valeurs de Ritz harmoniques utilise plusieurs astuces numériques qui sont détaillées dans cette section.

En reprenant la notation de la section 3.2, le problème aux valeurs propres généralisées (3.19) se change en un problème standard de valeurs propres car $\mathbf{H}_m$ est inversible

$$\begin{aligned} \overline{\mathbf{H}}_m^T \overline{\mathbf{H}}_m \mathbf{y}_m &= \lambda \mathbf{H}_m^T \mathbf{y}_m \\ \Leftrightarrow \quad \mathbf{H}_m^{-T} \overline{\mathbf{H}}_m^T \overline{\mathbf{H}}_m \mathbf{y}_m &= \lambda \mathbf{y}_m \end{aligned} \tag{A.1}$$

Il reste à simplifier $\mathbf{H}_m^{-T} \overline{\mathbf{H}}_m^T \overline{\mathbf{H}}_m = \left(\overline{\mathbf{H}}_m \mathbf{H}_m^{-1}\right)^T \overline{\mathbf{H}}_m$. De plus, la matrice $\overline{\mathbf{H}}_m$ est la matrice $\mathbf{H}_m$ à laquelle on a rajouté une ligne contenant un seul terme, $h_{m=1,m}$:

$$\overline{\mathbf{H}} = \begin{pmatrix} & \mathbf{H}_m & \\ 0 \quad \cdots & 0 & h_{m+1,m} \end{pmatrix}$$

Ainsi

$$\overline{\mathbf{H}}_m \mathbf{H}_m^{-1} = \begin{pmatrix} & \mathbf{H}_m & \\ 0 \quad \cdots & 0 & h_{m+1,m} \end{pmatrix} \mathbf{H}_m^{-1} = \begin{pmatrix} \mathbf{I}_m \\ h_{m+1,m} \mathbf{e}_m^T \mathbf{H}_m^{-1} \end{pmatrix} \tag{A.2}$$

On en déduit

$$\mathbf{H}_m^{-T}\overline{\mathbf{H}}_m^T\overline{\mathbf{H}}_m = \begin{pmatrix}\mathbf{I}_m & h_{m+1,m}\mathbf{H}_m^{-T}\mathbf{e}_m\end{pmatrix}\begin{pmatrix}\mathbf{H}_m \\ 0 \quad \cdots \quad 0 \quad h_{m+1,m}\end{pmatrix}$$
$$= \mathbf{H}_m + h_{m+1,m}^2\mathbf{H}_m^{-T}\mathbf{e}_m\mathbf{e}_m^T \tag{A.3}$$

On note $\mathbf{f}_m$ la dernière colonne de $\mathbf{H}_m^{-T}$. On obtient finalement

$$\mathbf{H}_m^{-T}\overline{\mathbf{H}}_m^T\overline{\mathbf{H}}_m = \mathbf{H}_m + h_{m+1,m}^2\mathbf{f}_m\mathbf{e}_m^T \tag{A.4}$$

Cela correspond à une modification de $\mathbf{H}_m$ par une matrice de rang 1

A.2 Calcul du vecteur $\mathbf{f}_m$

Le vecteur $\mathbf{f}_i$ est la dernière colonne de la matrice $\mathbf{H}_m^{-T}$. Elle se calcule très facilement à l'aide des rotations de Givens qui rendent cette dernière triangulaire supérieure. En reprenant la notation de l'équation (3.11), on remarque que seules $m-1$ rotations de Givens sont nécessaires pour rendre $\mathbf{H}_m$ triangulaire (alors qu'il en faut m pour la matrice $\overline{\mathbf{H}}_m$). Ainsi, comme la matrice $\mathbf{Q}_{m-1}$ est orthonormale,

$$\begin{aligned} & \mathbf{Q}_{m-1}\mathbf{H}_m &=& \mathbf{R}_m \\ \Leftrightarrow \quad & \mathbf{H}_m^{-T} &=& \mathbf{Q}_{m-1}^{-1}\mathbf{R}_m^{-T} \end{aligned} \tag{A.5}$$

L'inverse de la transformation $\mathbf{Q}_{m-1}$ consiste simplement à appliquer les transformations de Givens inverses dans leur ordre opposé, *i.e.*

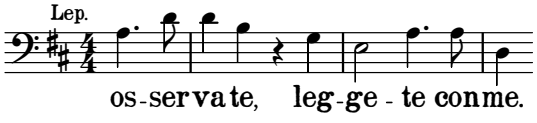
$$\mathbf{Q}_{m-1}^{-1} = \mathbf{\Omega}_1^{-1}\mathbf{\Omega}_2^{-1}\cdots\mathbf{\Omega}_{m-1}^{-1} \tag{A.6}$$

L'inverse d'une rotation de Givens est triviale à appliquer, car elle est représentée par une matrice orthonormale [63].

Enfin, la matrice $\mathbf{R}_m$ est triangulaire supérieure. Son inverse l'est également. La méthode de remontée d'une matrice triangulaire supérieure permet de montrer que $\mathbf{R}_m^{-1}\mathbf{e}_m = \frac{1}{R_{mm}}$, où R_{mm} est le terme en position (m,m) de $\mathbf{R}_m$. Le calcul du vecteur $\mathbf{f}_m$ est alors immédiat :

$$\mathbf{f}_m = \mathbf{H}_m^{-T}\mathbf{e}_m = \mathbf{Q}_{m-1}^{-1}\mathbf{R}_m^{-T}\mathbf{e}_m = \mathbf{Q}_{m-1}^{-1}\frac{1}{R_{mm}}\mathbf{e}_m \tag{A.7}$$

Pour conclure, le coefficient R_{mm} n'est pas disponible directement dans l'implémentation actuelle du GMRES, car m rotations de Givens sont appliquées à $\overline{\mathbf{H}}_m$. Il faut donc utiliser l'inverse de la dernière rotation $\mathbf{\Omega}_m$ pour retrouver R_{mm}.



Mozart, *Don Giovanni*, Acte 1 sc. 5

Bibliographie

[1] Agarwal A. & Dowling A.P. Low-frequency acoustic shielding by the silent aircraft airframe. *AIAA Journal*, 45(2), February 2007.

[2] Agullo E., Giraud L. & Jing Y.F. Block GMRES method with inexact breakdowns and deflated restarting. *Research Report RR-8503*, INRIA, 2014. URL `https://hal.inria.fr/hal-00963704v2`.

[3] Al Daas H., Grigori L., Hénon P. & Ricoux P. Enlarged GMRES for reducing communication. *Research Report RR-9049*, Inria Paris, mars 2017. URL `https://hal.inria.fr/hal-01497943`.

[4] Alauzet F., Borel-Sandou S., Daumas L., Dervieux A., Dinh Q., Kleinveld S. & A. Loseille Y. Mesri G.R. "multi-model design strategies applied to sonic boom reduction. *European Journal of Computational Mechanics*, 1 :1–20, 2007.

[5] Alcin H., Koobus B., Allain O. & Dervieux A. 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.

[6] Alonso J., Martins J., Reuther J., Haimes R. & Crawford C. High-Fidelity Aero-Structural Design Using a Parametric CAD-Based Model. In *AIAA-2003-3429*. AIAA, 2003. URL `http://dx.doi.org/10.2514/6.2003-3429`.

[7] Amestoy P., Duff I., Koster J. & L'Excellent J.Y. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal of Matrix Analysis and Applications*, 23(1), 2001.

[8] Amestoy P.R., Davis T.A. & Duff I.S. An approximate minimum degree ordering algorithm. *SIAM Journal on Matrix Analysis and Applications*, 17(4) :886–905, 1996.

[9] Amestoy P.R., Guermouche A., L'Excellent J.Y. & Pralet S. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2) :136–156, 2006.

[10] Amestoy P.R., L'Excellent J. & Daydé M. Linear algebra and sparse direct methods. 2004.

[11] Anderson E., Bai Z., Bischof C., Blackford S., Demmel J., Dongarra J., Du Croz J., Greenbaum A., Hammarling S., McKenney A. & Sorensen D. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, troisième édition, 1999. ISBN 0-89871-447-8 (paperback).

[12] Balay S., Abhyankar S., Adams M.F., Brown J., Brune P., Buschelman K., Dalcin L., Eijkhout V., Gropp W.D., Kaushik D., Knepley M.G., McInnes L.C., Rupp K., Smith B.F., Zampini S., Zhang H. & Zhang H. PETSc web page, 2016. URL `http://www.mcs.anl.gov/petsc`.

[13] Balay S., Abhyankar S., Adams M.F., Brown J., Brune P., Buschelman K., Dalcin L., Eijkhout V., Kaushik D., Knepley M.G., McInnes L.C., Gropp W.D., Rupp K., Smith B.F., Zampini S., Zhang H. & Zhang H. PETSc users manual. *Rapport technique ANL-95/11 - Revision 3.7*, Argonne National Laboratory, 2016.

[14] Balay S., Gropp W.D., McInnes L.C. & Smith B.F. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A.M. Bruaset & H.P. Langtangen (éditeurs), *Modern Software Tools in Scientific Computing*, p. 163–202. Birkhauser Press, 1997.

[15] Barth T.J., Chan T.F. & Tang W.P. A parallel non-overlapping domain-decomposition algorithm for compressible fluid flow problems on triangulated domains. *Contemporary Mathematics*, 218, 1998.

[16] Beau G.L., Ray S., Aliabadi S. & Tezduyar T. SUPG finite element computation of compressible flows with the entropy and conservation variables formulations. *Computer Methods in Applied Mechanics and Engineering*, 104(3) :397 – 422, 1993. ISSN 0045-7825. URL `http://dx.doi.org/10.1016/0045-7825(93)90033-T`.

[17] Benamou J.D. & Després B. A domain decomposition method for the helmholtz equation and related optimal control problems. *Journal of Computational Physics*, 136(1) :68–82, 1997.

[18] Benzi M. Preconditioning techniques for large linear systems : a survey. *Journal of computational Physics*, 182(2) :418–477, 2002.

[19] Benzi M. & Tuma M. A comparative study of sparse approximate inverse preconditioners. *Applied Numerical Mathematics*, 30(2-3) :305–340, 1999.

[20] Bischof C.H. & Quintana-Ortí G. Computing Rank-Revealing QR Factorizations of Dense Matrix. *ACM Transactions on Mathematical Softwares*, 24(2) :226–253, June 1998.

[21] Bissuel A., Allaire G., Daumas L., Barré S. & Rey F. Linearized Navier-Stokes Equations for Aeracoustics using Stabilized Finite Elements : Boundary Conditions and Industrial Application to Aft-Fan Noise Propagation. *Computers and Fluids*, 2017. En cours de revue.

[22] Bissuel A., Allaire G., Daumas L., Chalot F., Barré S. & Rey F. Linearized Navier-Stokes for Aeracoustics : Assessment of Aft Fan Noise Radiated from Business Jet Engine Nozzles. In *Finite Elements in Flow 2017*. IACM - CIMNE, 2017.

[23] Bissuel A., Allaire G., Daumas L., Mallet M. & Chalot F. Solving linear systems with multiple right-hand sides with GMRES : an application to aircraft design. In M. Papadrakakis, V. Papadopoulos, G. Stefanou & V. Plevris (éditeurs), *VII European Congress on Computational Methods in Applied Sciences and Engineering*, p. 7358–7371. ECCOMAS Congress 2016, 2016. URL `http://dx.doi.org/10.7712/100016.2339.7593`.

[24] Brooks A.N. & Hughes T.J.R. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1-3) :199–259, September 1982.

[25] Cai X.C. & Sarkis M. A restricted additive Schwarz preconditionner for general sparse linear systems. *SIAM journal on Scientific Computing*, 21 :239–247, 1999.

[26] Chalot F. *Encyclopaedia of Computational Mechanics*, chapitre Industrial aerodynamics. Wiley, 2004.

[27] Chalot F., Chevalier G., Dinh Q.V. & Giraud L. *Some Investigations of Domain Decomposition Techniques in Parallel CFD*, p. 595–602. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999. ISBN 978-3-540-48311-3. URL `http://dx.doi.org/10.1007/3-540-48311-X_84`.

[28] Chalot F., Hughes T. & Shakib F. Symmetrization of conservation laws with entropy for high-temperature hypersonic computations. *Computing Systems in Engineering*, 1(2) :495 – 521, 1990. ISSN 0956-0521. URL `http://dx.doi.org/10.1016/0956-0521(90)90032-G`.

[29] Chan T.F. Rank-revealing QR factorizations. *Linear Algebra and its Applications*, 1987.

[30] Chan T.F. & Mathew T.P. Domain decomposition algorithms. *Acta numerica*, 3 :61–143, 1994.

[31] Chan T.F. & Van der Vorst H.A. Approximate and incomplete factorizations. In *Parallel numerical algorithms*, p. 167–202. Springer, 1997.

[32] Chandrasekaran S. & I. I. On Rank-Revealing QR Factorisations. *Research Report RR-880*, YALEU/DCS, 1991.

[33] Chow E. & Patel A. Fine-grained parallel incomplete ILU factorization. *SIAM J. Sci. Comput.*, 37(2) :C169–C193, 2015.

[34] Clean Sky project. http ://www.cleansky.eu. URL `http://www.cleansky.eu`.

[35] Cook P.H., Firmin M.C.P. & McDonald M.A. Aerofoil RAE 2822 : pressure distributions, and boundary layer and wake measurements. *AGARD Report AR-138*, AGARD, 1977.

[36] Cuthill E. & McKee J. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th national conference*, p. 157–172. ACM, 1969.

[37] Daumas L. *Optimisation aérodynamique dans le cadre de la conception multidisciplinaire en contexte aéronautique.* Thèse de doctorat, Montpellier 2, 2005.

[38] Daumas L., Forestier N., Bissuel A., Broux G., Chalot F., Johan Z. & Mallet M. Industrial frequency-domain linearized Navier-Stokes calculations for aeroelastic problems in the transonic flow regime. In *IFASD*. 2017.

[39] Davis T.A., Rajamanickam S. & Sid-Lakhdar W.M. A survey of direct methods for sparse linear systems. *Acta Numerica*, 25 :383–566, 2016.

[40] De Roeck Y.H. & Le Tallec P. Analysis and test of a local domain decomposition preconditioner. In *Fourth international symposium on domain decomposition methods for partial differential equations*, tome 4. 1991.

[41] Deck S. Recent improvements in the zonal detached eddy simulation (ZDES) formulation. *Theoretical and Computational Fluid Dynamics*, 26(6) :1–28, December 2011.

[42] Dinh Q.V., Rogé G., Sevin C. & Stoufflet B. Shape Optimisation in Computational Fluid Dynamics. *European Journal of Finite Elements*, 5 :569–594, 1996.

[43] Dinkla R.M. *GMRES(m) with deflation appplied to nonsymmetric systems arising from fluid mechanics problems.* Thèse de maîtrise, Faculty of Aerospace Engineering, Delft University of Technology, 2009.

[44] Dolean V., Jolivet P. & Nataf F. *An introduction to domain decomposition methods : algorithms, theory, and parallel implementation.* SIAM, 2015.

[45] Dolean V., Lanteri S. & Nataf F. Optimized interface conditions for domain decomposition methods in fluid dynamics. *International Journal for Numerical Methods in Fluids*, 40(12) :1539–1550, 2002. ISSN 1097-0363. URL `http://dx.doi.org/10.1002/fld.410`.

[46] Dolean V., Nataf F., Scheichl R. & Spillane N. Analysis of a two-level Schwarz method with coarse spaces based on local Dirichlet–to–Neumann maps. *Computer Methods in Applied Mathematics*, 12(4) :391–414, 2012. URL `https://hal.archives-ouvertes.fr/hal-00586246`.

[47] Efstathiou E. & Gander M.J. Why Restricted Additive Schwarz converges faster than Additive Schwarz. *BIT*, 43(1) :1–10, 2002.

[48] Embree M. How descriptive are GMRES convergence bounds ? *Rapport technique*, Oxford University Computing Laboratory, 1999.

[49] Erlangga Y.A. & Nabben R. Deflation and balancing preconditioners for Krylov subspace methods applied to nonsymmetric matrices. *SIAM Journal on Matrix Analysis and Applications*, 30(2) :684–699, 2008.

[50] Ern A. & Guermond J.L. *Éléments finis : théorie, applications, mise en œuvre*. Numéro 36 in Mathématiques & Applications. Springer-Verlag, 2002.

[51] Fanion T. *Étude de la simulation numérique des phénomènes d'aéroélasticité dynamique. Application au problème du flottement des avions.* Thèse de doctorat, Université Paris Dauphine, 2001.

[52] Farhat C. & Roux F.X. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32(6) :1205–1227, 1991.

[53] Franca L.P. & Hughes T.J.R. Two classes of mixed finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 69(1) :89–129, July 1988.

[54] Frank J. & Vuik C. On the construction of deflation-based preconditioners. *SIAM Journal on Scientific Computing*, 23(2) :442–462, 2001.

[55] Freund R.W. & Malhotra M. A block QMR algorithm for non-hermitian linear systems with multiple right-hand sides. *Linear Algebra and its Applications*, 254(1) :119 – 157, 1997. ISSN 0024-3795. URL `http://dx.doi.org/10.1016/S0024-3795(96)00529-0`. Proceeding of the Fifth Conference of the International Linear Algebra Society.

[56] Frommer A. & Szyld D.B. An algebraic convergence theory for restricted additive schwarz methods using weighted max norms. *SIAM Journal on Numerical Analysis*, 39(2) :463–479, 2001. URL `http://dx.doi.org/10.1137/S0036142900370824`.

[57] Gander M.J. Optimized Schwarz methods. *SIAM J. Numer. Anal.*, 44(2) :699–731, 2006.

[58] Gaul A., Gutknecht M.H., Liesen J. & Nabben R. A framework for deflated and augmented Krylov subspace methods. *SIAM Journal on Matrix Analysis and Applications*, 34(2) :495–518, 2013.

[59] Gerardo-Giorda L., Le Tallec P. & Nataf F. A robin–robin preconditioner for advection–diffusion equations with discontinuous coefficients. *Computer Methods in Applied Mechanics and Engineering*, 193(9) :745–764, 2004.

[60] Gerhold T., Galle M., Friedrich O., Evans J., Gerhold T., Galle M., Friedrich O. & Evans J. Calculation of complex three-dimensional configurations employing the dlr-tau-code. In *35th Aerospace Sciences Meeting and Exhibit*, p. 167. 1997.

[61] Godlewski E. & Raviart. *Numerical approximation of hyperbolic systems of conservation laws*. Springer-Verlag New York, 1996.

[62] Golub G.H. Numerical methods for solving linear least squares problems. *Numerische Mathematik*, 7(3) :206–216, 1965. URL `http://dx.doi.org/10.1007/BF01436075`.

[63] Golub G.H. & Loan C.F.V. *Matrix computations*. Johns Hopkins University Press, deuxième édition, 1990.

[64] Goossens S. & Roose D. Ritz and harmonic Ritz values and the convergence of FOM and GMRES. *Numerical Linear Algebra with Applications*, 6(4) :281–293, 1999. ISSN 1099-1506. URL `http://dx.doi.org/10.1002/(SICI)1099-1506(199906)6:4<281::AID-NLA158>3.0.CO;2-B`.

[65] Gould N.I. & Scott J. On approximate-inverse preconditioners. *Rapport technique*, SCAN-9508223, 1995.

[66] Greenbaum A., Pták V. & Strakoš Z. Any nonincreasing convergence curve is possible for GMRES. *SIAM journal on matrix analysis and applications*, 17(3) :465–469, 1996.

[67] Greif C., Rees T. & Szyld D.B. Additive schwarz with variable weights. In *Domain Decomposition Methods in Science and Engineering XXI*, p. 779–787. Springer, 2014.

[68] Greif C., Rees T. & Szyld D.B. Gmres with multiple preconditioners. *SeMA Journal*, p. 1–19, 2017.

[69] Gustafsson I. A class of first order factorization methods. *BIT Numerical Mathematics*, 18(2) :142–156, 1978.

[70] Gutknecht M.H. Block Krylov space methods for linear systems with multiple right-hand sides : an introduction. 2006.

[71] Gutknecht M.H. & Schmelzer T. Updating the QR decomposition of block tridiagonal and block Hessenberg matrices. *Applied Numerical Mathematics*, 58(6) :871–883, 2008.

[72] Haase G. & Langer U. Multigrid methods : from geometrical to algebraic versions. *Modern methods in scientific computing and applications*, 75 :103–153, 2002.

[73] Haferssas R. *Coarse space for domain decomposition method with optimized transmission conditions*. Thèse de doctorat, Université Pierre et Marie Curie ; Laboratoire Jacques-Louis Lions (UPMC) ; Inria Paris, 2016.

[74] Harari I., Barbone P.E. & Montgomery J.M. Finite element formulatiosn for exterior problems : application to hybrid methods, non-reflecting boundary conditions and infinite elements. *Internation Journal for Numerical Methods in Engineering*, 40 :2791–2805, 1997.

[75] Harten A. On the symmetric form of systems of conservation laws with entropy. *Journal of Computational Physics*, 49(1) :151–164, January 1983.

[76] Hascoët L. & Pascual V. The Tapenade Automatic Differentiation tool : Principles, Model, and Specification. *ACM Transactions On Mathematical Software*, 39(3), 2013. URL `http://dx.doi.org/10.1145/2450153.2450158`.

[77] Heinlein A., Klawonn A. & Rheinbach O. A Parallel Implementation of a Two-Level Overlapping Schwarz Method with Energy-Minimizing Coarse Space Based on Trilinos. *SIAM Journal on Scientific Computing*, 38(6) :C713–C747, 2016.

[78] Hsu M.C., Bazilevs Y., Calo V., Tezduyar T. & Hughes T. Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. *Computer Methods in Applied Mechanics and Engineering*, 199(13) :828 – 840, 2010. ISSN 0045-7825. URL `http://dx.doi.org/10.1016/j.cma.2009.06.019`.

[79] Hughes T.J.R. *The Finite Element Method : Linear Static and Dynamic Finite Element Analysis*. Dover, deuxième édition, 2000.

[80] Hughes T.J.R., Feijoo G.R., Mazzei L. & Quincy J.B. The variational multiscale method - a paradigm for computational method mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166(1-2) :3–24, November 1998.

[81] Hughes T.J.R., Franca L.P. & Hulbert G.M. A new finite element formulation for computational fluid dynamics : VIII the Galerkin Least-Squares method for advective-diffusive equations. *Computer Methods in Applied Mechanics and Engineering*, 73 :173–189, May 1989.

[82] Hughes T.J.R., Franca L.P. & Mallet M. A new finite element formulation for computational fluid dynamics : I symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics. *Computer Methods in Applied Mechanics and Engineering*, 54(2) :223–234, February 1986.

[83] Hughes T.J.R. & Mallet M. A new finite element formulation for computational fluid dynamics : III. the generalized streamline operator for multidimensional advective-diffusive systems. *Computer Methods in Applied Mechanics and Engineering*, 58(3) :305 – 328, 1986. URL `https://doi.org/10.1016/0045-7825(86)90152-0`.

[84] Hysom D. & Pothen A. A scalable parallel algorithm for incomplete factor preconditioning. *SIAM Journal on Scientific Computing*, 22(6) :2194–2215, 2001.

[85] Irwin C.A.K. & Guyett P.R. The subcritical response and flutter of a swept-wing model. *R. & M. No. 3497*, Aeronautical Research Council, 1965.

[86] Jameson A. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3) :233–260, Sep 1988. ISSN 1573-7691. URL `http://dx.doi.org/10.1007/BF01061285`.

[87] Japhet C. & Nataf F. The best interface conditions for domain decomposition : absorbing boundary conditions. *Artificial Boundary Conditions, with Applications to Computational Fluid Dynamics Problems*, p. 348–373, 2001.

[88] Johan Z., Mathur K.K., Johnsson S.L. & Hughes T.J.R. A case study in parallel computation : Viscous flow around an ONERA M6 wing. *International Journal for Numerical Methods in Fluids*, 21(10) :877–884, 1995. ISSN 1097-0363. URL `http://dx.doi.org/10.1002/fld.1650211008`.

[89] John V. & Knobloch P. On spurious oscillations at layers diminishing (SOLD) methods for convection–diffusion equations : Part I – a review. *Computer Methods in Applied Mechanics and Engineering*, 196(17) :2197 – 2215, 2007. ISSN 0045-7825. URL `http://dx.doi.org/10.1016/j.cma.2006.11.013`.

[90] Karypis G. & Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1) :359–392, 1998.

[91] Katz J. & Plotkin A. *Low Speed Aerodynamics – From Wing Theory to Panel Methods*. Mc Graw-Hill, 1991.

[92] Keyes D. How scalable is domain decomposition in practice. In *Proceedings of the 11th International Conference on Domain Decomposition Methods*, p. 286–297. 1998.

[93] Keyes D.E. & Gropp W.D. A Comparison of Domain Decomposition Techniques for Elliptic Partial Differential Equations and their Parallel Implementation. *SIAM Journal on Scientific and Statistical Computing*, 8(2) :s166–s202, 1987.

[94] Klawonn A. & Rheinbach O. Deflation, Projector Preconditioning, and Balancing in Iterative Substructuring Methods : Connections and New Results. *SIAM Journal on Scientific Computing*, 34(1) :A459–A484, 2012. URL `http://dx.doi.org/10.1137/100811118`.

[95] Knobloch P. On the choice of the SUPG parameter at outflow boundary layers. *Advances in Computational Mathematics*, 31(4) :369,

Apr 2008. ISSN 1572-9044. URL `http://dx.doi.org/10.1007/s10444-008-9075-6`.

[96] Langer S., Schwöppe A. & Kroll N. The dlr flow solver tau - status and recent algorithmic developments. In A. Scitech (éditeur), *52nd Aerospace Sciences Meeting*. 2004.

[97] Lesoinne M., Sarkis M., Hetmaniuk U. & Farhat C. A linearized method for the frequency analysis of three-dimensional fluid/structure interaction problems in all flow regimes. *Computer Methods in Applied Mechanics and Engineering*, 190(24) :3121 – 3146, 2001. ISSN 0045-7825. URL `http://dx.doi.org/S0045-7825(00)00385-6`.

[98] Levasseur V. *Simulation des grandes échelles en éléments finis stabilisés : une approche variationnelle multi-échelles*. Thèse de doctorat, Université Paris VI Pierre et Marie Curie, 2007.

[99] Lions J.L. *Contrôle optimal de systèmes gouvernés par les équations aux dérivées partielles*. Dunod, 1968.

[100] Lions P.L. On the Schwarz alternating method III : a variant for nonoverlapping subdomains. In T. Chan, R. Glowinski, J. Periaux & O. Widlund (éditeurs), *Domain Decomposition Methods for Partial Differential Equations*, p. 202–223. SIAM, 1990.

[101] Lipton R.J., Rose D.J. & Tarjan R.E. Generalized nested dissection. *SIAM journal on numerical analysis*, 16(2) :346–358, 1979.

[102] Mack C.J. & Schmid P.J. A preconditionned Krylov technique for global hydrodynamic stability analysis of large-scale compressible flows. *Journal of Comp. Phys.*, 229(3) :541–560, 2010.

[103] Magoulès F., Roux F.X. & Salmon S. Optimal discrete transmission conditions for a nonoverlapping domain decomposition method for the helmholtz equation. *SIAM Journal on Scientific Computing*, 25(5) :1497–1515, 2004.

[104] Mallet M. *A Finite Element Method for Computational Fluid Dynamics*. Thèse de doctorat, Stanford University, 1985.

[105] Mandel J. Balancing domain decomposition. *International Journal for Numerical Methods in Biomedical Engineering*, 9(3) :233–241, 1993.

[106] Manoha E., Guenanff R., Redonnet S. & Terracol M. Acoustic scattering from complex geometries. In *10th AIAA/CEAS Aeroacoustics Conference*. AIAA, 2004.

[107] Martin L. *Conception aérodynamique robuste*. Thèse de doctorat, Université de Toulouse, Université Toulouse III-Paul Sabatier, 2010.

[108] Mavriplis D.J. Multigrid techniques for unstructured meshes. *Rapport technique ICASE Report No. 95-27*, ICASE NASA, 1995.

[109] Mavriplis D.J. Directional coarsening and smoothing for anisotropic Navier-Stokes problems. *Electronic Transactions on Numerical Analysis*, 6 :182–197, 1997.

[110] Mavriplis D.J. Directional agglomeration multigrid techniques for high-reynolds-number viscous flows. *AIAA journal*, 37(10) :1222–1230, 1999.

[111] McCracken A.J., Timme S., Badcock K.J. & Eberthardsteiner J. Accelerating convergence of the cfd linear frequency domain method by a preconditioned linear solver. In *6th European Congress on Computational Methods in Applied Sciences and Engineering*. 2012.

[112] Meijerink J.A. & van der Vorst H.A. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics of computation*, 31(137) :148–162, 1977.

[113] Mohamed A.G., Fox G.C. & Laszewski G. Blocked LU factorization on a multiprocessor computer. *Computer-Aided Civil and Infrastructure Engineering*, 8(1) :45–56, 1993.

[114] Mohammadi B. & Pironneau O. *Applied Shape Optimization for Fluids*. Oxford University Press, 2001.

[115] Morgan R.B. GMRES with deflated restarting. *SIAM J. Sci. Statist. Comput*, 24, 2002.

[116] Morgan R.B. Restarted block-GMRES with deflation of eigenvalues. *Applied Numerical Mathematics*, 54(2) :222–236, 2005.

[117] Mosson A., Binet D. & Caprile J. Simulation of the installation effects of the aircraft engine rear fan noise with ACTRAN/DGM. In *20th AIAA/CEAS Aeroacoustics Conference*. AIAA, 2014. URL `http://dx.doi.org/10.2514/6.2014-3188`.

[118] Nabben R. & Vuik C. A comparison of deflation and the balancing preconditioner. *SIAM Journal on Scientific Computing*, 27(5) :1742–1759, 2006.

[119] Nachtigal N.M., Reddy S.C. & Trefethen L.N. How fast are nonsymmetric matrix iterations ? *SIAM Journal on Matrix Analysis and Applications*, 13(3) :778–795, 1992.

[120] Nicolaides R.A. Deflation of conjugate gradients with applications to boundary value problems. *SIAM Journal on Numerical Analysis*, 24(2) :355–365, 1987.

[121] Normand P.E. *Application de méthodes d'ordre élevé en éléments finis pour l'aérodynamique*. Thèse de doctorat, Université de Bordeaux, 2011.

[122] Okusanya T.O. *Algebraic Multigrid for Stabilized Finite Elements Discretizations of the Navier-Stokes Equations*. Thèse de doctorat, MIT, 2002.

[123] Pandya M.J., Diskin B., Thomas J. & Frink N.T. Assessment of preconditioner for a usm3d hierarchical adaptive nonlinear iteration method (hanim) (invited). In *AIAA SciTech Forum*. American Institute of Aeronautics and Astronautics, Jan 2016. URL `http://dx.doi.org/10.2514/6.2016-0860`.

[124] Parks M.L., De Sturler E., Mackey G., Johnson D.D. & Maiti S. Recycling Krylov subspaces for sequences of linear systems. *SIAM Journal on Scientific Computing*, 28(5) :1651–1674, 2006.

[125] Pinel X. & Montagnac M. Block Krylov methods to solve adjoint problems in aerodynamic design optimization. *AIAA journal*, 2013.

[126] Pironneau O. *Optimal shape design for elliptic systems*. Springer Verlag New-York, 1984.

[127] Redon E., Ouedraogo B., Dhia A.S.B.B., Meerci J.F. & Chambeyron C. Transparent boundary condition for acoustic propagation in lined guide with mean flow. In *Acoustics08*. 2008.

[128] Redonnet S., Desquesnes G., Manoha E. & Parzani C. Numerical study of acoustic installation effects with a computational aeroacoustics method. *AIAA Journal*, 48(5) :929–937, May 2010. ISSN 0001-1452. URL `http://dx.doi.org/10.2514/1.42153`.

[129] Rey F. Design and test of innovative after-bodies for bizjets. *Greener aviation*, October 2016.

[130] Rienstra S.W. & Hirschberg W. *An Introduction to Acoustics*. Eindhoven University of Technology, 2004.

[131] Robbé M. & Sadkane M. Exact and inexact breakdowns in the block GMRES method. *Linear Algebra and its Applications*, p. 265–285, 2006.

[132] Röllin S. & Fichtner W. Improving the accuracy of GMRES with deflated restarting. *SIAM Journal on Scientific Computing*, 30(1) :232–245, 2007.

[133] Rose D.J. & Tarjan R.E. Algorithmic aspects of vertex elimination on directed graphs. *SIAM Journal on Applied Mathematics*, 34(1) :176–197, 1978. URL `http://dx.doi.org/10.1137/0134014`.

[134] Saad Y. A flexible inner-outer preconditioned gmres algorithm. *SIAM Journal on Scientific Computing*, 14(2) :461–469, 1993.

[135] Saad Y. *Iterative Methods for Sparse Linear Systems*. SIAM, deuxième édition, 2003.

[136] Saad Y. *Numerical methods for large eigenvalue problems*. SIAM, deuxième édition, 2011.

[137] Saad Y. & Schultz M.H. GMRES : a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3) :856–869, July 1986.

[138] Sarkis M. Partition of unity coarse spaces : enhanced versions, discontinuous coefficients and applications to elasticity. *Domain decomposition methods in science and engineering*, p. 149–158, 2003.

[139] Sarkis M. & Koobus B. A scaled and minimum overlap restricted additive Schwarz method with application to aerodynamics. *Comput. Methods Appli. Mech. Engrg.*, 184(2) :391–400, 2000.

[140] Sauter S. & Schwab C. *Boundary Element Method*. Springer Series in Computational Mathematics. Springer, 2011.

[141] Schmitt V. & Charpin F. Pressure distributions on the ONERA-M6-Wing at transonic Mach numbers. *Experimental data base for computer program assessment*, 4, 1979.

[142] Schwarz H.A. Über einen Grenzübergang durch alternierendes Verfahren. *Viertaljahrsschrift der Naturforschenden Gesellschaft in Zürich*, (15) :272–286, 1870.

[143] Serres D. *Systèmes de lois de conservation. I Hyperbolicité, entropies, ondes de choc*. Diderot Éditeur, Arts et Science, 1996.

[144] Serres D. *Systèmes de lois de conservation. II Structures géométriques, oscillations et problèmes mixtes*. Diderot Éditeur, Arts et Science, 1996.

[145] Shakib F. *Finite element analysis of the compressible Euler and Navier-Stokes equations*. Thèse de doctorat, Stanford, 1988.

[146] Shakib F., Hughes T.J. & Johan Z. A new finite element formulation for computational fluid dynamics : X. the compressible euler and navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 89(1) :141 – 219, 1991. ISSN 0045-7825. URL `https://doi.org/10.1016/0045-7825(91)90041-4`.

[147] Soemarwoto B., Jameson A., Martins A., Oskam B. & Laban M. Adaptive aerodynamic optimization of regional jet aircraft. In *40th AIAA Aerospace Sciences Meeting & Exhibit*, p. 260. 2002.

[148] Spillane N. *Méthodes de décomposition de domaine robustes pour les problèmes symétriques définis positifs*. Thèse de doctorat, Paris 6, 2014.

[149] Spillane N., Dolean V., Hauret P., Nataf F., Pechstein C. & Scheichl R. Abstract robust coarse spaces for systems of pdes via generalized eigenproblems in the overlaps. *Numerische Mathematik*, 126(4) :741–770, 2014.

[150] St-Cyr A., Gander M.J. & Thomas S.J. Optimized multiplicative, additive, and restricted additive schwarz preconditioning. *SIAM Journal on Scientific Computing*, 29(6) :2402–2425, 2007.

[151] St-Cyr A., Gander M.J. & Thomas S.J. Optimized restricted additive schwarz methods. In *Domain Decomposition Methods in Science and Engineering XVI*, p. 213–220. Springer, 2007.

[152] Stüben K. A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, 128(1) :281 – 309, 2001. ISSN 0377-0427. URL `http://dx.doi.org/10.1016/S0377-0427(00)00516-1`.

[153] Stoufflet B. Computational science and engineering achievements in the designing of aircraft. In *SIAM Conference on Computational Science and Engineering*. SIAM, Atlanta, 2017.

[154] Tadmor E. Skew-selfadjoint form for systems of conservation laws. *Journal of Mathematical Analysis and Applications*, 103(2) :428–442, October 1984.

[155] Tang J.M., Nabben R., Vuik C. & Erlangga Y.A. Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. *Journal of scientific computing*, 39(3) :340–370, 2009.

[156] Toselli A. & Widlund O.B. *Domain decomposition methods : algorithms and theory*, tome 34. Springer, 2005.

[157] Tsynkov S. Numerical solution of problems on unbounded domains. A review. *Appl. Num. Math.*, 27 :465 – 532, 1998.

[158] Widhalm M., Dwight R., Thormann R. & Hübner A. Efficient computation of dynamic stability data with a linearized frequency domain solver. In *5th European Congress on Computational Methods in Applied Sciences and Engineering*. 2010.

[159] Xu S., Timme S. & Badcock K.J. Krylov subspace recycling for linearised aerodynamics analysis using DLR-TAU. In *International Forum on Aeroelasticity and Structural Dynamics (IFASD)*. 2015.

[160] Yser P. *Simulation numérique aéroacoustique d'écoulements par une approche LES d'ordre élevé en éléments finis non structurés*. Thèse de doctorat, École Centrale Lyon, 2016.

[161] Zienkiewicz O. & Taylor R. *The Finite Element Method : Fluid Dynamics*, tome 3. Butterworth-Heinemann, cinquième édition, 2000.

[162] Zienkiewicz O. & Taylor R. *The Finite Element Method : The basis*, tome 1. Butterworth-Heinemann, cinquième édition, 2000.

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

Titre : Résolution des équations de Navier-Stokes linéarisées pour l'aéroélasticité, l'optimisation de forme et l'aéroacoustique

Mots clefs : Navier-Stokes linéarisées, systèmes linéaires, éléments finis

Résumé : Les équations de Navier-Stokes linéarisées sont utilisées dans l'industrie aéronautique pour l'optimisation de forme aérodynamique, l'aéroélasticité et l'aéroacoustique. Deux axes ont été suivis pour accélérer et rendre plus robuste la résolution de ces équations. Le premier est l'amélioration de la méthode itérative de résolution de systèmes linéaires utilisée, et le deuxième la formulation du schéma numérique conduisant à ce système linéaire.
Dans cette première partie, si l'extension de l'algorithme GMRES avec déflation spectrale à des systèmes à plusieurs seconds membres a été testée et ne s'est pas révélée compétitive, l'amélioration du préconditionnement de la méthode GMRES par l'utilisation d'un préconditionnement ILU(k) parallélisé par une méthode de Schwarz additive a permis une accélération du temps de résolution allant jusqu'à un facteur dix, ainsi que la convergence de cas jusqu'alors impossibles à résoudre.
La deuxième partie présente d'abord un travail sur la stabilisation SUPG du schéma élément fini utilisé. La forme proposée de la matrice de stabilisation, dite complète, a donné des résultats encourageants en non-linéaire qui ne se sont pas transposés en linéarisé. Une étude sur les conditions aux limites de Dirichlet clôt cette partie. Une méthode algébrique d'imposition de conditions non homogènes sur des variables non triviales du calcul, qui a permis l'application industrielle à l'aéroacoustique, y est détaillée. De plus, la preuve est apportée que le caractère transparent d'une condition de Dirichlet homogène sur toutes les variables s'explique par le schéma SUPG.

Title : Solving the linearized Navier-Stokes equations for aeroelasticity, shape optimization and aeroacoustics

Keywords : Linearized Navier-Stokes, linear systems, finite elements

Abstract : The linearized Navier-Stokes equations are solved in the aerospace industry for aerodynamic shape optimisation, flutter calculations and aeroacoustics. To improve the robustness and the speed of the solver, two complementary paths were taken. The first is a work on the iterative methods used to solve linear systems, and the second is the improvement of the numerical scheme leading to these linear systems.
In the first part, the extension to multiple right-hand sides of the GMRES algorithm with spectral deflation was tested and proved uncompetitive. The use of the ILU(k) preconditioner parallelised with an additive Schwarz method for improving the preconditioning of the GMRES method gave a tenfold reduction of the time needed to solve the systems, and also enabled the convergence of some very difficult cases which were impossible to solve until now.
The second part starts with a work on the SUPG method used to stabilise the finite element scheme. A new way of computing the stabilisation matrix gave promising results on non-linear cases, which were not replicated for linear cases. A study on Dirichlet boundary conditions concludes this part. An algebraic method to impose non-homogeneous Dirichlet boundary conditions on non-trivial variables is presented. It enabled the use in an industrial context of linearized Navier-Stokes for aeroacoustics. Moreover, the transparent behaviour of a homogeneous Dirichlet boundary condition on all variables is proved to be due to the SUPG stabilisation.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France