HAL
open science

Algorithmes par décomposition de domaine et accélération multigrille pour le calcul d'écoulements compressibles

Victoria Dolean

To cite this version:

Victoria Dolean. Algorithmes par décomposition de domaine et accélération multigrille pour le calcul d'écoulements compressibles. Mathématiques [math]. Université Nice Sophia Antipolis, 2001. Français. NNT : . tel-00005614

HAL Id: tel-00005614

https://pastel.hal.science/tel-00005614

Submitted on 5 Apr 2004

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

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

UNIVERSITÉ DE NICE-SOPHIA ANTIPOLIS
FACULTÉ DES SCIENCES

THÈSE

présentée pour obtenir le titre de
Docteur en SCIENCES
(discipline : Mathématiques)

Victorita Dolean

ALGORITHMES PAR DÉCOMPOSITION DE DOMAINE ET ACCÉLÉRATION MULTIGRILLE POUR LE CALCUL D'ÉCOULEMENTS COMPRESSIBLES

Soutenue le 25 Avril 2001 devant le jury composé de :

Yves Achdou	:	rapporteur
Grégoire Allaire	:	rapporteur
Jean-Antoine Désidéri	:	directeur
Stéphane Lanteri	:	co-directeur
Frédéric Nataf	;	examinateur
Frédéric Poupaud	:	président
Patrick Vuillermoz	:	invité

Remerciements

Je voudrais d'abord remercier tous ceux qui ont contribué au bon déroulement de cette thèse: M. Jean-Antoine Désidéri qui a accepté de la diriger, les membres des équipes SINUS et CAIMAN de l'INRIA Sophia Antipolis et plus particulièrement, Stéphane Lanteri qui s'est beaucoup impliqué en étant un soutien réel à la fois professionnel et humain, et bien-sûr, le CNES qui a soutenu et contribué à la réalisation de cette thèse.

Je souhaite aussi remercier M. Frédéric Nataf qui a inspiré, par ses travaux et idées, une partie consistante de ce mémoire.

Mes remerciements se dirigent aussi vers M. Grégoire Allaire et M. Yves Achdou qui ont accepté d'être les rapporteurs de cette thèse, et vers M. Frédéric Poupaud qui a bien voulu présider son jury.

Bien que pas présent durant ces dernières années, mon entière gratitude va vers celui qui a été le cataliseur de la réaction qui a mené vers mon parcours de doctorant dans un pays étranger et sans le soutien duquel je n'aurais jamais réussi, mon ancien professeur de l'Université de Cluj-Napoca, M. Mihai Barbosu.

Je remercie ma famille qui a investi ses espoirs et attentes en moi, et en particulier, ma mère qui m'a révélé la première les secrets et la beauté des mathématiques.

Evidemment, nombreux sont ceux qui ont marqué ce long chemin, je leurs suis reconnaissante pour tout ce qu'ils ont apporté avec eux. Une pensée à part pour Max et Alice avec lesquels j'ai pu partager l'essentiel de ces dernières années.

Table des matières

Introduction générale **8**
Principes de base des méthodes de décomposition de domaine . . . 9
Décomposition de domaine en mécanique des fluides . . . 10
Équation scalaire de type advection-diffusion . . . 11
Système d'équations hyperboliques . . . 12
Système d'équations mixtes hyperboliques-paraboliques . . . 14
Objectif et plan de la thèse . . . 14

I Décomposition de domaine pour le système d'équations d'Euler **19**

1 Algorithme de Schwarz pour les équations Euler **21**
1.1 Décomposition de domaine pour des systèmes hyperboliques . . . 22
1.1.1 Position du problème . . . 22
1.1.2 Conditions de raccord aux interfaces et décomposition de domaine . . . 25
1.1.3 Algorithme de type Schwarz additif sans recouvrement . . . 27
1.1.4 Construction d'un système interface . . . 30
1.2 Analyse de convergence . . . 33
1.2.1 Le cas unidimensionnel . . . 33
1.2.2 Le cas bidimensionnel . . . 37
1.2.3 Le cas tridimensionnel . . . 50
1.3 Conclusion . . . 54

2 Résolution numérique des équations d'Euler **55**
2.1 Méthodes de discrétisation en espace et en temps . . . 55
2.1.1 Modèle mathématique . . . 55
2.1.2 Méthode de discrétisation en espace . . . 56
2.1.3 Conditions aux limites . . . 60
2.1.4 Intégration en temps . . . 61
2.2 Stratégie de parallélisation . . . 62
2.3 Algorithme de résolution par sous-domaine en discret . . . 63
2.4 Stratégie de résolution des systèmes locaux . . . 67
2.4.1 Généralités sur les méthodes multigrilles . . . 68
2.4.2 Méthode multigrille linéaire par agglomération . . . 68
2.5 Annexe : matrices locales et matrices de couplage pour Euler . . . 75
2.6 Résultats numériques . . . 78
2.6.1 Validations numériques de l'étude de convergence . . . 78

2.6.2 Calcul d'écoulements stationnaires . . . 85
2.7 Conclusion . . . 88

3 Une méthode de préconditionnement algébrique 93
3.1 Expression du préconditionneur . . . 93
3.2 Préconditionneur polynomial hiérarchique . . . 94
3.3 Résultats numériques . . . 94
3.3.1 Définition du cas test . . . 94
3.3.2 Plate-formes de calcul . . . 95
3.3.3 Étude sur un système linéaire . . . 95
3.3.4 Calcul de l'écoulement complet . . . 96
3.4 Conclusion . . . 110

4 Conditions d'interface optimisées 111
4.1 Introduction . . . 111
4.2 Diagonalisation de Smith . . . 112
4.2.1 Forme diagonale de Smith . . . 112
4.2.2 Application au système d'équations d'Euler . . . 115
4.2.3 Analyse de convergence . . . 119
4.2.4 Transformations préliminaires . . . 122
4.3 Conditions d'interface optimisées . . . 123
4.3.1 Un problème non-symétrique du troisième ordre . . . 123
4.3.2 Construction d'une classe plus générale de conditions d'interface . . . 128
4.3.3 Algorithme de résolution en discret . . . 138
4.4 Résultats numériques . . . 144
4.5 Conclusion . . . 145

II Décomposition de domaine pour le système d'équations de Navier-Stokes 151

5 Calcul d'écoulements stationnaires 153
5.1 Résolution numérique du système d'équations de Navier-Stokes . . . 153
5.1.1 Modèle mathématique . . . 154
5.1.2 Méthode de discrétisation en espace . . . 155
5.1.3 Conditions aux limites . . . 157
5.1.4 Intégration en temps . . . 158
5.2 Algorithme de résolution par sous-domaine . . . 160
5.2.1 Formulation en continu . . . 161
5.2.2 Formulation en discret . . . 161
5.2.3 Stratégie de résolution des systèmes locaux . . . 164
5.3 Résultats numériques . . . 165
5.3.1 Définition des cas tests . . . 165
5.3.2 Plateformes de calcul . . . 166
5.3.3 Cas test **TVIS1** . . . 166
5.3.4 Cas test **TVIS2** . . . 171
5.3.5 Cas test **TVIS3** . . . 174
5.4 Conclusion . . . 177

6 Calcul d'écoulements instationnaires **179**
6.1 Une méthode implicite précise au second ordre 179
6.2 Résultats numériques 183
6.3 Conclusion 197

Conclusion générale **199**

Références bibliographiques **204**

Introduction générale

Principes de base des méthodes de décomposition de domaine

Dans la résolution numérique d'un problème modélisé par une équation aux dérivées partielles, on est généralement confronté, après une étape de discrétisation, à la résolution d'un système linéaire. La grande taille de ce système ou bien le fait que celui-ci soit mal conditionné, sont deux raisons pour lesquelles une approche globale et une résolution directe ne sont plus appropriées. L'avènement des calculateurs parallèles et leur capacité à résoudre numériquement des problèmes requérant une grande quantité de calculs ont conduit à une recherche très active de méthodes de résolution de systèmes linéaires adaptées à ce type d'architecture. Les méthodes de décomposition de domaine en sont un des exemples les plus populaires. Les méthodes de décomposition de domaine sont des méthodes très flexibles de résolution des systèmes issus de la discrétisation d'équations aux dérivées partielles, qui utilisent les propriétés mathématiques de celles-ci pour un calcul itératif de la solution. De nos jours, les méthodes par décomposition de domaine sont généralement utilisées comme préconditionneurs pour des méthodes itératives de type Krylov comme la méthode du gradient conjugué ou les méthodes de minimisation de résidus généralisés telle que GMRES.

Les premières méthodes apparues dans la littérature visaient surtout la possibilité de résoudre des problèmes de grande taille par décomposition en problèmes de tailles réduites pouvant être traités par des calculateurs à faible capacité mémoire. Dans ce cas, la solution d'un sous-problème permettait de définir les conditions aux limites du sous-problème suivant et on procédait donc à un traitement en séquence des différents sous-problèmes. L'algorithme de Schwarz, initialement conçu pour prouver l'existence de la solution d'un problème de Poisson posé sur un domaine quelconque (voir par exemple Lions[Lio88]), est la toute première illustration de cette approche. Il repose sur une décomposition spatiale du domaine de calcul en sous-domaines recouvrants, la zone de recouvrement ainsi introduite permettant une définition simple des conditions aux limites aux frontières artificielles (aussi nommées « interfaces »). La forme originale de cet algorithme est souvent qualifiée de « multiplicative » parce que le traitement en séquence des sous-problèmes se traduit par un opérateur itératif s'exprimant comme le produit d'opérateurs associés aux résolutions locales.

Les méthodes de décomposition de domaine peuvent être classifiées suivant quelques critères (on se référera notamment aux ouvrages récents [QV99] et [SBG96] qui constituent deux références incontournables sur le sujet) :

- selon le type de décomposition en régions sur lesquelles on définit les sous-problèmes, on distingue les méthodes « avec recouvrement » ou « sans recouvrement ». Comme souligné plus haut, les premières permettent une actualisation très simple des solutions locales sur la base de conditions aux limites de type Dirichlet ou Neumann (ou une combinaison des deux) aux frontières artificielles. Elles se caractérisent aussi par une complexité arithmétique accrue du fait de la multiple définition

des degrés de liberté aux interfaces. Leur inconvénient principal est donc de compliquer quelque peu la mise en œuvre numérique, surtout lors de la résolution de problèmes tridimensionnels sur des géométries de formes complexes;

– selon l'interdépendance des résolutions locales à chaque itération, on aboutit à des formes « multiplicatives » (équivalentes à des méthodes itératives du type Gauss-Seidel par bloc) ou « additives » (équivalentes à des méthodes itératives de type Jacobi par bloc). Il est clair que les méthodes multiplicatives (comme l'algorithme de Schwarz sous sa forme originale) sont peu ou pas du tout adaptées aux architectures de calcul parallèles. Ces dernières années ont donc vu la mise au point de nombreuses méthodes additives caractérisées par un traitement simultané des sous-problèmes et donc plus à même d'exploiter efficacement les possibilités des calculateurs parallèles.

Les méthodes de décomposition de domaine sans recouvrement sont principalement de deux types : les méthodes de sous-structuration (ou méthodes du complément de Schur) et les méthodes de type Schwarz sans recouvrement. Les méthodes de sous-structuration s'apparentent aux méthodes d'élimination de Gauss par bloc. Elles consistent à ramener la résolution d'un problème global posé sur l'ensemble des degrés de liberté (d.d.l.) issus de la discrétisation du domaine de calcul, à la résolution d'un problème de taille moindre posé sur les d.d.l. « interfaces ». Ces derniers sont associés aux nœuds situés sur les interfaces entre sous-domaines voisins dans la partition du domaine de calcul. Le problème d'interface ainsi posé est alors résolu par une méthode itérative adaptée (généralement une méthode de Krylov). L'opérateur associé au problème d'interface et résultant du processus d'élimination de Gauss par bloc, est appelé complément de Schur. Du point de vue du problème continu, cela revient à la définition d'un nouvel opérateur, l'opérateur de Steklov-Poincaré, qui agit sur les variables d'interface [QV91]-[QV99] et dont la discrétisation donne le complément de Schur de l'opérateur du problème global. Les méthodes de type Schwarz sans recouvrement, quant à elles, peuvent être vues comme un cas particulier des méthodes de sous-structuration. En effet, on démontre facilement qu'une méthode de type Schwarz sans recouvrement est équivalente à une méthode de Jacobi appliquée aux seuls d.d.l. d'interfaces. Cependant par rapport au cas avec recouvrement, on est ici amené à formuler des conditions aux limites plus complexes pour retrouver un niveau d'efficacité correct, ou même assurer la convergence. La variante sans recouvrement de l'algorithme de Schwarz original a été introduite par Lions[Lio88]-[Lio89] et ensuite étudiée par Desprès[Des91] et Nataf *et al.*[NRdS94].

Notons enfin que le terme décomposition de domaine a selon [SBG96] quelques significations sensiblement différentes comme par exemple le processus de distribution des données entre les processeurs d'un ordinateur à mémoire distribuée, ou la séparation du domaine physique dans des régions où les phénomènes physiques sont modélisés par des équations aux dérivées partielles différentes (raccord de modèles), ou encore, du point de vue du préconditionnement, c'est le processus de séparation du problème initial en sous-problèmes de dimensions plus petites et donc plus faciles à résoudre, chaque résolution locale définissant une contribution à un inverse approché de l'opérateur global. D'une façon générale, les méthodes de décomposition de domaine permettent une parallélisation plus facile et performante en favorisant explicitement la localisation des données. Elles peuvent aussi conduire à des algorithmes présentant des propriétés de convergence supérieures.

Décomposition de domaine en mécanique des fluides

Les méthodes de décomposition de domaine ont été initialement développées pour des problèmes elliptiques du second ordre et profitent de la forte régularité des solutions de ce type d'équation, ainsi

que de la symétrie des opérateurs impliqués. On est alors capable de construire des méthodes quasi-optimales, c'est-à-dire de convergence faiblement dépendante du nombre de degrés de liberté (ou encore, de la taille du maillage sous-jacent), et « scalables », c'est-à-dire de convergence indépendante du nombre de sous-domaines. La situation est beaucoup moins claire pour les systèmes d'EDP représentatifs de la mécanique des fluides compressibles. Les opérateurs sont à dominante du premier ordre, non symétriques, à solutions comportant souvent des singularités. C'est dans ce cadre que se situe la présente étude dont l'objectif principal est la mise au point d'algorithmes par décomposition de domaine pour la résolution de systèmes hyperboliques (système d'équations d'Euler) ou mixtes hyperboliques-paraboliques (système des équations de Navier-Stokes) pour des applications en mécanique des fluides compressibles.

Dans l'application des méthodes de décomposition de domaine, il est impératif d'avoir des conditions appropriées aux interfaces entre les sous-domaines voisins pour que la méthode converge. Ce problème est résolu dans le cas elliptique, comme par exemple dans le cas de l'élasticité linéaire [Smi92] où l'on traite des opérateurs symétriques définis positifs, ou encore dans le cas d'opérateurs non symétriques qui ont une partie elliptique dominante [XC92], [CW92]. Dans [XZ98] on trouvera une étude exhaustive des méthodes par décomposition de domaine sans recouvrement pour des systèmes elliptiques, construites avec des conditions de Dirichlet ou de Neumann aux interfaces, ainsi que des méthodes de préconditionnement appropriées. Dans le cas de systèmes hyperboliques ou mixtes hyperboliques-paraboliques, le caractère non symétrique de l'opérateur associé impose la prise en compte de conditions de transmission spécifiques. Dans le cas où il y a un terme de diffusion et lorsque celui-ci est dominant, on peut utiliser avec de bonnes chances de succès les algorithmes conçus pour les systèmes linéaires symétriques avec quelques modifications dans l'étape de résolution. Quand la convection est dominante, ces méthodes sont moins efficaces ou bien divergent, et il faut alors modifier les conditions de raccord.

Équation scalaire de type advection-diffusion

Considérons tout d'abord le cas d'une équation scalaire modélisant l'advection et la diffusion d'une quantité $u(x,t)$. On note au passage que cette équation est souvent utilisée comme modèle simplifié des EDP représentatives de la mécanique des fluides. Un problème aux valeurs initiales et aux limites impliquant cette équation est classiquement donné par :

$$\left\{\begin{array}{lcll} \dfrac{\partial u}{\partial t}(x,t) & + & \vec{V}.\nabla u(x,t) - \nu\triangle u(x,t) = 0 \quad \nu \geq 0 & \text{dans } \Omega \\ \mathcal{C}u(x,t) & = & g(x,t) & \text{sur } \Gamma \\ u(x,t=0) & = & u_0(x) & \text{dans } \Omega \end{array}\right. \tag{1}$$

où Γ désigne la frontière du domaine de calcul $\Omega \subset \mathbb{R}^2$ et $\mathcal{C}$, l'opérateur des conditions aux limites. Il existe plusieurs algorithmes par décomposition de domaine appropriés à la résolution de (1). Un de ceux-ci, proposé par d'Hennezel *et al.*[dLTV92], consiste en la définition de deux types de sous-problèmes, respectivement avec des conditions de Dirichlet et de Neumann posées aux interfaces entre sous-domaines voisins. La résolution des sous-problèmes du même type se fait en parallèle, les échanges de valeurs aux interfaces ayant lieu depuis les sous-domaines posant des conditions de Neumann vers ceux posant des conditions de Dirichlet pour une interface donnée. Une autre approche, rencontrée dans [CQ95], [GGQ96] et [GGQ98], suit l'idée de résolution par « itération par sous-domaine »; dans ce cas, l'itération principale porte seulement sur les variables interfaces, chaque itération impliquant des résolutions locales dans les sous-domaines. Les auteurs montrent que cette méthode est équivalente à la formulation du problème en terme de l'opérateur de Steklov-Poincaré. Les conditions de transmission aux interfaces sont de type

Robin (c'est-à-dire, une combinaison d'une condition de Dirichlet et d'une condition de Neumann). Une méthode différente, visant à augmenter la vitesse de convergence, consiste à préconditionner le problème de Steklov-Poincaré posé à l'interface. Un préconditionneur dont l'optimalité est démontrée dans [ALTNV00], est construit à partir de résolutions locales de problèmes posés avec des conditions de type Robin aux interfaces. L'idée est de trouver un opérateur qui, dans le cas de deux sous-domaines, est l'inverse exact de l'opérateur de Steklov-Poincaré (les calculs qui permettent de justifier cette approche se font dans l'espace des phases).

Une stratégie alternative de résolution par décomposition de domaine pour l'équation d'advection-diffusion fait appel aux notions de conditions aux limites absorbantes. Les conditions aux limites absorbantes (ou conditions aux limites artificielles) ont été introduites pour la première fois par Engquist et Majda[EM77] dans un article désormais célèbre. Ce type de conditions aux limites a été utilisé pour la première fois dans la résolution des équations aux dérivées partielles définies sur un domaine non-borné. Celles-ci sont imposées sur une frontière artificielle de telle sorte que la solution dans le domaine restreint soit la plus proche possible de la solution dans le domaine infini. Leur application à l'équation d'advection-diffusion et aux équations de Navier-Stokes pour un fluide incompressible, a été faite par Halpern et Schatzmann dans [Hal86],[Hal88] et [HS89]. Par la suite, Nataf [Nat93] s'est aperçu que l'on rencontre une situation similaire dans les méthodes de décomposition de domaine : idéalement, on cherche à résoudre localement dans chaque sous-domaine indépendamment des sous-domaines voisins. Nataf *et al.* [NRdS94] ont par ailleurs démontré que dans ce contexte, l'utilisation de conditions de raccord absorbantes conduit à une vitesse de convergence optimale des méthodes de type Schwarz sans recouvrement. Cependant, ce type de conditions aux limites fait intervenir des opérateurs non-locaux qui doivent ensuite être approchés par des opérateurs aux dérivées partielles. Ceci a été réalisé dans le cas de l'équation d'advection-diffusion par Japhet dans [Jap98] ayant comme critère d'approximation, l'optimisation du taux de convergence de l'algorithme de type Schwarz dans lequel les conditions de transmission sont ces conditions optimisées. Plus précisément, on cherche des conditions de transmission à l'interface consistant à spécifier une expression de la forme :

$$\left(\frac{\partial}{\partial n_i} - c_1 \frac{\partial}{\partial \tau_i} - c_2 \frac{\partial^2}{\partial \tau_i^2}\right) u$$

où les n_i et les τ_i désignent respectivement les vecteurs unitaires normaux à l'interface et les vecteurs unitaires tangents. Les constantes c_i sont déterminées de sorte que le taux de convergence de l'algorithme de Schwarz sans recouvrement appliqué au problème décomposé avec de telles conditions de raccord soit optimal. Le problème de minimisation obtenu est tout d'abord résolu dans le cas d'une convection normale à l'interface, le résultat étant ensuite étendu au cas général.

Système d'équations hyperboliques

Dans la théorie des systèmes hyperboliques, indépendamment de la décomposition de domaine, il y a plusieurs aspects liés : la bonne définition du problème aux valeurs initiales et aux limites[Jef76], la linéarisation et la symétrisation (lorsque cette opération est possible) [Cle97]-[Les75] et l'approximation par des schémas spécifiques comme les schémas de type volumes finis traités dans [GR96]. Un exemple représentatif de traitement de systèmes de lois de conservation par décomposition de domaine est fourni par Quarteroni[Qua90] dans le cadre d'une approximation par méthode spectrale. Une autre approche qui a fait l'objet de plusieurs travaux ces dernières années, consiste à construire des conditions de raccord aux interfaces entre des sous-domaines non-recouvrants dans le contexte d'un algorithme de type Schwarz additif. Plus précisément, supposons que l'on cherche à résoudre le problème :

$$\left\{ \begin{array}{ll} \mathcal{L}W & \equiv A_0 W + \sum_{k=1}^{d} A_k \partial_{x_k} W = f \quad \text{dans} \quad \Omega \\ & \mathcal{C}W = g \quad \text{sur} \quad \partial\Omega \end{array} \right. \tag{2}$$

par une méthode de décomposition de domaine sur un découpage du domaine de calcul de la forme $\Omega = \bigcup_{i=1}^{k} \Omega_i$, $\Omega_i \cap \Omega_j = \Gamma_{ij}$ désignant l'interface entre les sous-domaines Ω_i et Ω_j si ces derniers sont voisins. Soit $W_i^{(p)}$, l'approximation de la solution W dans le sous-domaine Ω_i, à l'itération p. L'algorithme de type Schwarz additif s'écrit :

$$\left\{ \begin{array}{lcll} \mathcal{L}W_i^{(p+1)} & = & f & \text{dans } \Omega_i \\ \mathcal{B}_i W_i^{(p+1)} & = & \mathcal{B}_i W_{i-1}^{(p)} & \text{sur } \Gamma_{ij} \\ \mathcal{C}W_i^{(p+1)} & = & g & \text{sur } \partial\Omega \cup \partial\Omega_i \end{array} \right. \tag{3}$$

où $\mathcal{B}_i$ est un opérateur interface. Des conditions naturelles, déduites de la formulation faible du problème aux limites (2), sont trouvées dans [QS96]. Dans [GG93] on trouve une formulation du problème à l'interface à l'aide d'un opérateur de Steklov-Poincaré pour des équations de transport plus simples et dans le cas d'une décomposition en deux sous-domaines :

$$\begin{pmatrix} I & -S_2 \\ -S_1 & I \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \Phi_2 \end{pmatrix} = 0$$

où les Φ_i sont les flux entrants dans chaque sous-domaine et où les opérateurs S_i sont définis de telle sorte qu'ils conduisent aux flux sortants lorsqu'ils sont appliqués aux flux entrants. La généralisation de cette formulation au traitement de systèmes hyperboliques est immédiate. Clerc[Cle97] a étudié des conditions de raccord applicables à la résolution par décomposition de domaine de systèmes hyperboliques linéarisés et symétrisés (où symétrisables). On distingue tout d'abord des conditions naturelles, que l'on qualifiera dans la suite de ce manuscrit de « conditions de raccord classiques », qui découlent directement de la formulation faible du problème :

$$\mathcal{B}_i = A_{\mathbf{n}}^{-}$$

où $A_{\mathbf{n}} = A_1 n_1 + A_2 n_2 = T\Lambda T^{-1}$, $A_{\mathbf{n}}^{-} = T\Lambda^{-}T^{-1}$ et $\mathbf{n} = (n_1,n_2)$ est la normale sortante à l'interface. Il y a ensuite des conditions qui généralisent les précédentes :

$$\mathcal{B}_i = A_{\mathbf{n}}^{neg}$$

où les $A_{\mathbf{n}}^{neg}$ et $A_{\mathbf{n}}^{pos}$ sont des matrices dépendantes d'un paramètre qui garantit que les problèmes locaux sont bien posés. Ces matrices ont des propriétés similaires à celles caractérisant les matrices $A_{\mathbf{n}}^{-}$ et $A_{\mathbf{n}}^{+}$.

Dans notre étude, nous nous sommes notamment inspirés de la démarche suivie par Clerc[Cle97] avec l'objectif de l'étendre au système d'équations d'Euler, la principale différence résidant dans la définition d'un système d'interface exprimé en variables de flux et non pas en variables primales. Cette approche est avant tout motivée par le fait que les variables de flux $F(W)$ et $G(W)$ qui caractérisent le système d'équations d'Euler, contrairement aux variables d'état W, sont continues (se conservent) dans tout l'écoulement et donc, dans le contexte qui nous intéresse ici, aux interfaces entre sous-domaines. Une étude théorique via une analyse de Fourier combinée à une technique de diagonalisation classique, permet de déduire, comme dans le cas d'une équation d'advection-diffusion, que ces conditions naturelles assurent

de bonnes propriétés de convergence de l'algorithme de type Schwarz additif sans recouvrement. Par ailleurs, on a aussi étudié des conditions de raccord plus générales construites en utilisant la théorie de Smith pour la diagonalisation des matrices polynômiales[Gan65]. Une approche similaire a été adoptée dans un contexte différent par Gonzalez[Gon00] afin d'établir des conditions de raccord du second ordre, qualifiées dans la suite de « conditions de raccord optimisées », qui conduisent à une convergence optimale de l'algorithme de type Schwarz additif sans recouvrement. Dans cette étude, on propose une généralisation de la forme des conditions de raccord classiques dans l'espace de Fourier en raisonnant en termes des variables de Smith. Les nouvelles conditions, construites par une série de transformations algébriques, sont au plus du premier ordre (en dérivée tangentielle) et dépendent de quelques paramètres qui, s'ils sont convenablement choisis, permettent une meilleure convergence de l'algorithme de type Schwarz additif.

Système d'équations mixtes hyperboliques-paraboliques

On peut facilement étendre l'algorithme de type Schwarz additif sans recouvrement basé sur les conditions de raccord classiques au traitement du système d'équations de Navier-Stokes pour un fluide compressible. Dans ce cas, on doit tenir compte du fait qu'au terme de flux convectif présent dans le cas du système d'équations d'Euler, s'ajoute un terme de flux diffusif (flux visqueux). Une telle approche a été adoptée par Quarteroni et Stolcis[QS96] dans le cadre d'une méthode d'intégration en temps explicite.

Objectif et plan de la thèse

L'objectif général de cette thèse est la construction et l'évaluation d'algorithmes par décomposition de domaine pour la résolution des systèmes d'équations d'Euler et de Navier-Stokes (en régime laminaire) régissant l'écoulement d'un fluide compressible, le domaine d'application principal étant la dynamique des gaz. Dans la réalisation de cet objectif, notre démarche a été double : d'une part, nous avons cherché à appliquer les algorithmes en question dans des situations concrètes, à des fins de comparaison avec des méthodes plus largement utilisées pour la résolution des applications qui nous intéressent ici; d'autre part, nous avons étudié des solutions nouvelles, considérées comme plus prometteuses, nous limitant dans ce cas à des résultats préliminaires obtenus dans un cadre applicatif simplifié. Dans le premier cas, notre souci a été de proposer des solutions compétitives (en termes de temps de calcul) avec les méthodes existantes dans le domaine. Ainsi, nous avons évalué en détails des méthodes par décomposition de domaine basées sur des stratégies de résolution approchée des problèmes locaux. En effet, les méthodes de résolution couramment adoptées pour les systèmes linéaires issus de la discrétisation des systèmes d'équations d'Euler ou de Navier-Stokes sont très souvent de nature itératives (méthodes de relaxation de Jacobi ou de Gauss-Seidel, méthode de Krylov préconditionnées, méthodes multigrilles); les systèmes linéaires concernés sont rarement résolus de façon précise et, dans ce contexte, il est clair que l'utilisation de méthodes directes (factorisation LU) pour la résolution des problèmes locaux conduit à des méthodes de décomposition de domaine globalement trop coûteuses en temps de calcul et donc, peu ou pas compétitives vis-à-vis des méthodes existantes.

Cette thèse s'appuie sur plusieurs résultats issus de travaux précédemment menés au sein du projet SINUS de l'INRIA Sophia Antipolis. En particulier, le point de départ de la programmation des algorithmes proposés dans cette thèse est donné par un code de résolution numérique des systèmes d'équations d'Euler et de Navier-Stokes, qui intègre les ingrédients suivants :

– une formulation mixte éléments finis/volumes finis en maillages non-structurés triangulaires (cas

2D) basée sur des schémas décentrés TVD, pour la résolution des systèmes d'équations d'Euler et de Navier-Stokes en régime laminaire (voir notamment [Fez85], [FD89] et [FLLO89]);

- une méthode de type Euler implicite linéarisée [FS89]-[Ste88] pour l'intégration en temps des équations semi-discrétisées;
- un algorithme de résolution multigrille linéaire par agglomération de volumes pour la résolution itérative des systèmes linéaires. Cette méthode a essentiellement fait l'objet d'études visant son application à des problèmes bidimensionnels (voir notamment [Lal88], [LSA92], [Koo94] et [KLA94]).

Le manuscrit est structuré en deux parties. La première partie est consacrée à la résolution par décomposition de domaine du système d'équations d'Euler régissant l'écoulement d'un gaz parfait. Cette partie comporte les chapitres suivants :

- le chapitre 1 discute de la formulation et de l'analyse d'un algorithme de type Schwarz additif sur une décomposition sans recouvrement du domaine de calcul. Cet algorithme repose sur l'utilisation de conditions de raccord classiques qui découlent naturellement de la formulation variationnelle du problème et qui reviennent à imposer la continuité des flux normaux aux interfaces entre sous-domaines. La convergence de la méthode est étudiée en deux et trois dimensions d'espace, pour une décomposition en deux sous-domaines, en utilisant l'analyse de Fourier. Dans chaque cas, on démontre que l'algorithme est convergent et que, par ailleurs, il existe une valeur du nombre de Mach normal à l'interface pour laquelle le taux de convergence est optimal. Pour une large plage de valeurs du nombre de Mach de l'écoulement, le taux de convergence asymptotique (c'est-à-dire lorsque le pas d'espace tend vers 0) est sensiblement inférieur à 1;
- le chapitre 2 est consacré à la mise en œuvre et l'évaluation de l'algorithme proposé dans le chapitre 1, dans le cadre de la méthodologie numérique décrite plus haut. En particulier, la discrétisation en espace du système d'équations d'Euler repose sur une formulation mixte éléments finis/volumes finis en maillages triangulaires non-structurés. L'intégration en temps est réalisée au moyen d'une méthode d'Euler implicite linéarisée. Chaque pas de temps nécessite la résolution approchée d'un système linéaire de matrice creuse non-symétrique. Ici, on utilise l'algorithme de type Schwarz additif pour la réalisation d'un pas de temps implicite. D'un point de vue algébrique, l'algorithme de Schwarz peut être interprété comme une méthode de relaxation de Jacobi appliquée à la résolution d'un système linéaire dont la matrice a une structure par blocs particulière. Une technique de sous-structuration peut être appliquée à cette matrice afin d'obtenir un système interface. Dans notre cas, les variables aux interfaces sont les composantes de flux numériques décentrés. Il en résulte un algorithme par décomposition de domaine du type complément de Schur. Le système interface résultant est résolu au moyen d'une méthode GMRES. Une des originalités principales de notre approche consiste dans l'utilisation d'une méthode multigrille par agglomération de volumes pour la résolution itérative (approchée) des systèmes locaux intervenant dans l'algorithme par décomposition de domaine.
- le chapitre 3 porte sur l'étude numérique d'une technique de préconditionnement algébrique pour l'accélération de la résolution itérative du système d'interface qui caractérise l'algorithme par décomposition de domaine du type complément de Schur proposé au chapitre 2. On considère un préconditionneur de type polynomial dont l'utilisation est justifiée par un résultat théorique énoncé au chapitre 1, sur le caractère contractant de l'opérateur interface (matrice du complément de Schur). L'intérêt pratique de ce préconditionneur est là encore lié à l'adoption de stratégies itératives pour la résolution approchée des problèmes locaux;

– le chapitre 4 est avant tout motivé par le résultat de l'analyse de convergence menée au chapitre 1. En effet, cette analyse montre que l'utilisation de conditions de raccord classiques conduit à un algorithme exhibant de bonnes propriétés de convergence quel que soit le régime de l'écoulement. Le chapitre 4 vise alors deux objectifs : d'une part, une meilleure compréhension de l'impact des conditions d'interface retenues sur la convergence de l'algorithme et, d'autre part, la construction de conditions d'interfaces optimisées pour accélérer la convergence de la méthode de décomposition de domaine.

La seconde partie du manuscrit est consacrée à la résolution numérique du système d'équations de Navier-Stokes bidimensionnelles en régime laminaire. Elle est organisée en deux chapitres :

– le chapitre 5 traite de l'extension de la méthode de résolution par sous-domaine du système d'équations d'Euler proposée au chapitre 2, dans le but de permettre le traitement du système mixte hyperbolique/parabolique des équations de Navier-Stokes. Comme pour la résolution du système d'équations d'Euler, la méthode en question repose sur la formulation d'un algorithme de type Schwarz additif sur une décomposition sans recouvrement du domaine de calcul. Les conditions imposées aux interfaces entre sous-domaines voisins expriment, là encore, la continuité de flux normaux. Cependant, dans le cas présent, cette formulation doit être adaptée au fait que le vecteur flux mis en jeu combine des termes de flux convectif et des termes de flux diffusif (ou flux visqueux). Dans ce chapitre, les expériences numériques portent sur le calcul d'écoulements stationnaires;

– dans le chapitre 6, on s'intéresse au calcul d'écoulements instationnaires. L'objectif visé ici est avant tout la validation de la méthode de résolution par sous-domaine mise au point au chapitre 5 dans un contexte où la méthode d'intégration en temps influe sur la précision des résultats obtenus. Cette étude a été réalisée en utilisant une méthode d'intégration en temps de type implicite linéarisée, précise au second ordre.

Les études menées dans cette thèse ont donné lieu aux publications et présentations orales suivantes :

V. Dolean et S. Lanteri, *A domain decomposition approach to finite volume solutions of the Euler equations on triangular meshes.* Accepté pour publication dans *Int. J. Numer. Fluids.* en 2000. Une version longue est aussi parue comme rapport de recherche INRIA No. 3751 [1], octobre 1999.

V. Dolean, S. Lanteri et F. Nataf, *Convergence analysis of a Schwarz type domain decomposition method for the solution of the Euler equations*, rapport de recherche INRIA No. 3916 [2], avril 2000.

V. Dolean et S. Lanteri, *A non-overlapping domain decomposition method for solving the Navier-Stokes equations on unstructured triangular meshes*, rapport de recherche INRIA No. 3962 [3], juillet 2000.

V. Dolean et S. Lanteri, *A hybrid domain decomposition and multigrid method for the acceleration of compressible viscous flow calculations on unstructured triangular meshes*, Accepté pour publication dans *Comp. Fluid. Dyn. J.* en 2000.

V. Dolean, *Méthodes de résolution par décomposition de domaine pour des problèmes hyperboliques*, 30ème Congrès National d'Analyse Numérique (CANUM98), mai 1998, Arles.

1. URL : http://www.inria.fr/RRRT/RR-3751.html
2. URL : http://www.inria.fr/RRRT/RR-3916.html
3. URL : http://www.inria.fr/RRRT/RR-3962.html

V. Dolean, *A non-overlapping domain decomposition method for the solution of the Euler equations*, 8th International Symposium on Computational Fluid Dynamics (ISCFD99), septembre 1999, Bremen, Allemagne.

V. Dolean et S. Lanteri, *Une approche par sous-domaines pour la résolution des équations d'Euler par une méthode mixte éléments/volumes finis en maillages triangulaires*, Journée Décomposition de Domaine et Parallélisme, octobre 1999, CEA Saclay.

V. Dolean, S. Lanteri et F. Nataf, *Construction of non-overlapping domain decomposition algorithms for the solution of the Euler equations for compressible flows*, 13th International Conference on Domain Decomposition Methods, octobre 2000, University Claude Bernard Lyon 1.

Première partie

Décomposition de domaine pour le système d'équations d'Euler

Chapitre 1

Etude d'un algorithme de type Schwarz pour les équations d'Euler

L'objectif de ce chapitre est de formuler et d'étudier une méthode de décomposition de domaine pour la résolution du système d'équations d'Euler en deux dimensions d'espace. Dans un premier temps (section 1.1), on considère le cas d'un système hyperbolique général afin de situer le cadre de notre étude. Ce système est linéarisé puis symétrisé avant de rappeler un résultat de base sur l'existence et l'unicité d'une solution du problème aux limites correspondant. On formule ensuite un algorithme de type Schwarz additif sur une décomposition sans recouvrement du domaine de calcul. Cet algorithme est basé sur des conditions de raccord naturelles issues de la formulation variationnelle du problème aux limites. Ces conditions, qui sont aussi qualifiées de *classiques*, reviennent à imposer la continuité des flux normaux aux interfaces entre sous-domaines voisins.

D'autres travaux, portant sur la résolution par décomposition de domaine de problèmes du premier ordre, ont par exemple été réalisés par Gastaldi *et al.*[GGQ96]-Gastaldi[GG93] qui étudient la convergence d'un algorithme de type Schwarz additif pour une équation de transport scalaire, et Quarteroni[Qua90] qui construit une méthode de Schwarz pour un système hyperbolique et propose une analyse de convergence en dimension un. Un résultat démontré par Clerc[Cle97] montre, par une estimation d'énergie, que l'algorithme de type Schwarz additif basé sur des conditions de raccord classiques converge pour un système hyperbolique général, en dimension quelconque.

La deuxième partie de ce chapitre (section 2.1) est consacrée à l'adaptation et l'application de la méthode proposée dans la section 1.1 à la résolution du système d'équations d'Euler. Dans cette partie, une analyse de convergence est menée en considérant le système d'équations linéarisées et symétrisées et en se concentrant sur le cas d'une décomposition en deux sous-domaines d'un domaine infini (le plan ou l'espace réel). On applique alors une analyse de Fourier selon une technique désormais classique, utilisée pour la première fois par Nataf[Nat93] et ensuite par Japhet[Jap98] pour l'équation d'advection-diffusion, pour obtenir une expression du taux de convergence de l'algorithme de type Schwarz additif. On étudie ensuite le comportement de cette quantité en fonction du paramètre de fréquence et des autres paramètres physiques et numériques du problème.

1.1 Décomposition de domaine pour des systèmes hyperboliques

Dans cette section on formule un algorithme de type Schwarz additif sans recouvrement pour la résolution d'un système hyperbolique général.

1.1.1 Position du problème

1.1.1.1 Systèmes hyperboliques

Dans ce qui suit on rappelle quelques notions de la théorie des systèmes hyperboliques afin de bien situer l'étude et la formulation des algorithmes qui suivent. Nous considérons un système de lois de conservation de la forme :

$$\partial_t W + \sum_{i=1}^{d} \partial_{x_i} F_i(W) = 0, \quad W \in \mathbb{R}^p \tag{1.1}$$

où d est la dimension d'espace et p la dimension du système. Les fonctions de flux $F_i : \mathbb{R}^p \to \mathbb{R}^p$ sont supposées différentiables par rapport au vecteur d'état $W = W(x,t)$. Les flux sont dans le cas général des fonctions non-linéaires de W; on peut linéariser les équations pour étudier des petites perturbations au voisinage d'une solution stationnaire $W^\star(x)$ (qui vérifie donc $\sum_{i=1}^{d} \partial_{x_i} F_i(W^\star) = 0$).

Posons $W(x,t) = W^\star(x) + U(x,t)$; on a alors :

$$F_i(W) = F_i(W^\star) + \frac{\partial F_i}{\partial W}(W^\star)U + \cdots$$

ce qui conduit, au second ordre près, à :

$$\partial_t U + \sum_{i=1}^{d} \partial_{x_i} \left[\frac{\partial F_i}{\partial W}(W^*)U \right] = 0 \tag{1.2}$$

Par ailleurs, dans l'hypothèse de la régularité de la solution W, le système (1.1) peut s'écrire sous une forme *non-conservative* ou *quasi-linéaire* :

$$\partial_t W + \sum_{i=1}^{d} \frac{\partial F_i}{\partial W}(W)\partial_{x_i} W = 0 \tag{1.3}$$

ou encore :

$$\partial_t W + \sum_{i=1}^{d} A_i(W)\partial_{x_i} W = 0 \tag{1.4}$$

On rappelle la définition de base suivante.

Définition 1 Le système (1.4) est dit *strictement hyperbolique* en x si la matrice $\sum_{i=1}^{d} A_i(W)\nu_i$ est diagonalisable dans $\mathbb{R}$ et si ses valeurs propres sont distinctes quel que soit le vecteur unitaire $\nu = (\nu_1,...,\nu_d)$. Le système est dit *hyperbolique* au sens large si la matrice $\sum_{i=1}^{d} A_i(W)\nu_i$ est diagonalisable dans $\mathbb{R}$ et si ses valeurs propres ne sont pas toutes distinctes.

1.1.1.2 Discrétisation en temps implicite

On s'intéresse plus particulièrement au cas où le système (1.1) est intégré en temps par un schéma d'Euler implicite et où on linéarise le terme de flux implicite. On obtient alors :

$$\frac{\delta W}{\delta t} + \sum_{i=1}^{d} \partial_{x_i} \left[\frac{\partial F_i}{\partial W}(W^n)\delta W \right] = -\mathrm{div}(F(W^n)) \tag{1.5}$$

où $\delta W = W(x,t^{n+1}) - W(x,t^n) = W^{n+1} - W^n$. On suppose que δW est régulier et on écrit la forme non-conservative du système (1.5) :

$$\left[\frac{1}{\delta t} Id + \sum_{i=1}^{d} \partial_{x_i} \left[\frac{\partial F_i}{\partial W}(W^n) \right] \right] \delta W + \sum_{i=1}^{d} \left[\frac{\partial F_i}{\partial W}(W^n) \right] \partial_{x_i} \delta W = -\mathrm{div}(F(W^n)) \tag{1.6}$$

Le système (1.6) peut être symétrisé par multiplication par une matrice Σ qui, dans le cas où le système admet une entropie, est la matrice hessienne de cette entropie (voir par exemple Barth[Bar99]), pour conduire à un système du premier ordre de la forme :

$$A_0 \delta W + \sum_{i=1}^{d} A_i \partial_{x_i} \delta W = f \tag{1.7}$$

où :

$$\begin{cases} A_0 & = \Sigma \left[\frac{1}{\delta t} Id + \sum_{i=1}^{d} \partial_{x_i} \left[\frac{\partial F_i}{\partial W}(W^n) \right] \right] \\ A_i & = \Sigma \left[\frac{\partial F_i}{\partial W}(W^n) \right] \\ f & = -\Sigma \mathrm{div}(F(W^n)) \end{cases} \tag{1.8}$$

où les matrices Jacobiennes transformées A_i sont symétriques.

1.1.1.3 Espaces fonctionnels

Pour bien définir le problème on a besoin de quelques hypothèses de départ qui font intervenir les espaces fonctionnels suivants :

- $L^\infty(\Omega) = \{u : \Omega \to \mathbb{R};\ u$ est mesurable et $\exists\ C$ t.q. $|u(x)| \leq C$ p.p. dans $\Omega\}$
- $W^{1,\infty}(\Omega) = \{u : \Omega \to \mathbb{R}; \exists\ C$ et C_1 t.q. $|u(x)| \leq C$ et $|\partial_{x_i} u(x)| \leq C_1$ p.p. dans Ω, $i = 1,..,d\}$
- $H(\mathrm{div},\Omega) = \{u : \Omega \to \mathbb{R};\ u \in L^2(\Omega)$ et div u $\in L^2(\Omega)\}$
- $H^{1/2}(\partial\Omega) = \{u : \partial\Omega \to \mathbb{R};\ u \in L^2(\partial\Omega)$ et $\int_{\partial\Omega}\int_{\partial\Omega} \frac{(u(x)-u(y))^2}{|x-y|^d} ds(x)ds(y) < \infty\}$

On pose alors les hypothèses suivantes :

- $A_0 \in L^\infty(\Omega)^{p\times p}$;
- $A_i \in W^{1,\infty}(\Omega)^{p\times p}$;

- il existe une constante $C_0 > 0$ indépendante de x telle que :

$$(A_0^s - \frac{1}{2}\sum_{i=1}^{d} \partial_{x_i} A_i) \geq C_0 Id > 0 \quad \text{p.p. dans } \Omega \tag{1.9}$$

au sens des matrices symétriques définies positives, où A_0^s est la partie symétrique de A_0;

- la solution cherchée doit respecter : $W \in L^2(\Omega)^p$ et $\sum_{i=1}^{d} A_i \partial_{x_i} W \in L^2(\Omega)^p$.

On rencontre la condition (1.9) dans la définition des systèmes hyperboliques linéaires et symétriques de Friedrichs [Les75] comme une hypothèse qui assure que le problème est bien posé. Comme les opérateurs $A_i \in W^{1,\infty}(\Omega)^{p\times p}$ on peut écrire :

$$\sum_{i=1}^{d} \partial_{x_i} [A_i W] = \sum_{i=1}^{d} A_i \partial_{x_i} W + \left[\sum_{i=1}^{d} \partial_{x_i} A_i\right] W \quad \in L^2(\Omega)^p$$

ce qui revient à dire que $A_i W \in H(\text{div},\Omega)^p$. Cet espace peut être doté de la norme suivante :

$$\|(A_1 W,...,A_d W)\|_{H_{\text{div}}} = (\|(A_1 W,...,A_d W)\|_{L^2}^2 + \|\text{div}(A_1 W,...,A_d W)\|_{L^2}^2)^{1/2}$$

On peut donc définir la trace normale $A_{\mathbf{n}} W$ sur $\partial\Omega$ où $A_{\mathbf{n}} = \sum_{i=1}^{d} A_i \nu_i$ ($\mathbf{n} = (\nu_{\mathbf{1}},...,\nu_{\mathbf{p}})$ est le vecteur unitaire normal à $\partial\Omega$ et dirigé vers l'extérieur de $\partial\Omega$), comme une distribution de $H^{-1/2}(\partial\Omega)^p$. On s'attachera donc à chercher l'état W solution de (1.7) comme un élément de l'espace fonctionnel donné par :

$$\bar{H} = \{W \in L^2(\Omega)^p \text{ tel que } \sum_{i=1}^{d} A_i \partial_{x_i} W \in L^2(\Omega)^p \text{ et } W|_{\partial\Omega} \in L_A^{1/2}(\partial\Omega)\}$$
$$L_A^2(\partial\Omega) = \{W \text{ t.q. } \int_{\partial\Omega} |A_{\mathbf{n}}| W \cdot W d\sigma < \infty\}$$

L'espace des traces $L_{1/A}^2(\partial\Omega)$ défini ci-dessus peut-être doté de la norme :

$$\|W\|_{L_A^2} = (\int_{\partial\Omega} |A_{\mathbf{n}}| W \cdot W d\sigma)^{1/2}$$

tandis que l'espace dual $L_{1/A}^2(\partial\Omega) = \{\Phi \text{ tel que } \int_{\partial\Omega} |A_{\mathbf{n}}|^{-1} \Phi \cdot \Phi d\sigma < \infty\}$ est doté de la norme :

$$\|\Phi\|_{L_{1/A}^2} = (\int_{\partial\Omega} |A_{\mathbf{n}}|^{-1} \Phi \cdot \Phi d\sigma)^{1/2}$$

1.1.1.4 Conditions aux limites

Il est bien connu que l'on ne peut pas imposer toutes les composantes de W sur la frontière $\partial\Omega$. Pour bien poser le problème aux limites pour le système (1.7) on doit tenir compte du sens de propagation de l'information le long des caractéristiques. Plus précisément, le nombre et le type de conditions aux limites à imposer sur $\partial\Omega$ sont déduites de l'expression du système (1.7) en variables caractéristiques. Une discussion plus rigoureuse de la définition de conditions aux limites pour les systèmes hyperboliques de la dynamique des gaz est par exemple donnée dans Godlewski et Raviart[GR96].

On décompose donc l'opérateur $A_{\mathbf{n}}$ en parties positive et négative $A_{\mathbf{n}} = A_{\mathbf{n}}^+ + A_{\mathbf{n}}^-$. On suppose par la suite que $\det(A_{\mathbf{n}}) \neq 0$. La diagonalisation de $A_{\mathbf{n}}$ s'écrit $A_{\mathbf{n}} = T\Lambda T^{-1}$ avec :

$$\left\{ \begin{array}{c} A_{\mathbf{n}}^{\pm} = T\Lambda_{\mathbf{n}}^{\pm}T^{-1} \\ \Lambda_{\mathbf{n}}^{\pm} = \text{diag}(\lambda_i^{\pm})_{1\leq i\leq p} \quad \text{avec} \quad \lambda_i^{\pm} = \dfrac{1}{2}(\lambda_i \pm |\lambda_i|) \\ \text{et vérifiant} \quad A_{\mathbf{n}}^{+} = -A_{-\mathbf{n}}^{-} \end{array} \right.$$

Pour que le problème soit bien posé, on a besoin de conditions aux limites du type :

$$A_{\mathbf{n}}^- W = A_{\mathbf{n}}^- g \tag{1.10}$$

les $A_{\mathbf{n}}^-$ définissant l'information *entrante* dans le domaine Ω. On a le résultat suivant d'existence et unicité de la solution du problème défini avec les conditions aux limites mentionnées plus haut :

Théorème 1 *Le problème :*

$$\left\{ \begin{array}{rcll} \mathcal{L}W = A_0 W + \displaystyle\sum_{i=1}^{d} A_i \partial_{x_i} W & = & f & \textit{dans } \Omega \\ A_{\mathbf{n}}^- W & = & A_{\mathbf{n}}^- g & \textit{sur } \partial\Omega \end{array} \right. \tag{1.11}$$

avec $f \in L^2(\Omega)^p$ et $g \in L^2_A(\partial\Omega)$, admet une solution unique $W \in \bar{H}$. Si $f = 0$ on a l'estimation :

$$C_0 \|W\|^2_{L^2(\Omega)} + \|A_{\mathbf{n}}^+ W\|^2_{L^2_{1/A}(\partial\Omega)} \leq \|A_{\mathbf{n}}^- g\|^2_{L^2_{1/A}(\partial\Omega)} \tag{1.12}$$

Une démonstration de ce théorème est faite dans [Cle97].

1.1.2 Conditions de raccord aux interfaces et décomposition de domaine

La résolution par décomposition de domaine consiste en la définition de sous-problèmes tels que la solution obtenue localement soit la restriction de la solution globale au sous-domaine considéré. Il faut pour cela définir des conditions aux limites et des conditions de raccord appropriées à la nature hyperbolique des équations. Nous considérons donc le problème aux limites (global) suivant :

$$\left\{ \begin{array}{l} \mathcal{L}W \equiv A_0 W + \displaystyle\sum_{k=1}^{d} A_k \partial_{x_k} W = f \quad \text{dans } \Omega \\ \text{Conditions aux limites sur } \partial\Omega \end{array} \right. \tag{1.13}$$

Soit une décomposition $\Omega = \displaystyle\bigcup_{i=1}^{N} \Omega_i$ en tranches successives et W_i la solution du problème local :

$$\begin{cases} \mathcal{L}W_i = f|_{\Omega_i} \\ \text{Conditions aux limites sur } \ \partial\Omega \cap \partial\Omega_i \\ \text{Conditions de raccord aux interfaces } \ \partial\Omega_i \cap \partial\Omega_j \end{cases} \tag{1.14}$$

On peut alors énoncer le théorème suivant :

Théorème 2 *On prolonge les W_i solutions de (1.14) par 0 au domaine Ω entier. Une condition nécessaire et suffisante pour que $\sum_{i=1}^{N} W_i$ soit la solution du problème (1.13) est que les conditions de raccord sur les interfaces $\Gamma = \partial\Omega_i \cap \partial\Omega_j$ soient telles que :*

$$\begin{cases} A^-_{\mathbf{n_i}} W_i + A^+_{\mathbf{n_j}} W_j = 0 \\ \textit{et} \\ A^+_{\mathbf{n_i}} W_i + A^-_{\mathbf{n_j}} W_j = 0 \end{cases} \tag{1.15}$$

où $\mathbf{n_i}$ et $\mathbf{n_j}$ sont les normales extérieures aux frontières $\partial\Omega_i$ et $\partial\Omega_j$ respectivement.

Démonstration. On se limite au cas d'une décomposition en deux sous-domaines $\Omega = \Omega_1 \bigcup \Omega_2$. On note $\mathbf{n}_{\partial\mathbf{\Omega}}$ la normale à la frontière du domaine global Ω, $\mathbf{n} = \mathbf{n_1} = -\mathbf{n_2}$ la normale à l'interface et $W = W_1 + W_2$. On choisit comme espace des fonctions test :

$$V = \{X \in L^2(\Omega)^p \ \text{ t.q. } \quad \sum_{i=1}^{d} A_i X \in H(\text{div},\Omega)^p, \quad A_{\mathbf{n}} X \in L^2_A(\Gamma)\}$$

On intègre par parties tout d'abord sur le domaine global Ω et ensuite sur chacun des sous-domaines Ω_1 et Ω_2 :

$$\begin{aligned}
\int_{\Omega} [A_0 W + \sum_{k=1}^{d} A_k \partial_{x_k} W] X &= \int_{\Omega} [A_0^T X - \sum_{k=1}^{d} \partial_{x_k}(A_k X)] W + \int_{\partial\Omega} (A_{\mathbf{n}_{\partial\mathbf{\Omega}}} W) X \\
&= \int_{\Omega_1} [A_0^T X - \sum_{k=1}^{d} \partial_{x_k}(A_k X)] W_1 + \int_{\partial\Omega_1 \cap \partial\Omega} (A_{\mathbf{n}_{\partial\mathbf{\Omega}}} W_1) X \\
&+ \int_{\Omega_2} [A_0^T X - \sum_{k=1}^{d} \partial_{x_k}(A_k X)] W_2 + \int_{\partial\Omega_2 \cap \partial\Omega} (A_{\mathbf{n}_{\partial\mathbf{\Omega}}} W_2) X \\
&= \int_{\Omega_1} [A_0 W_1 + \sum_{k=1}^{d} A_k \partial_{x_k} W_1] X - \int_{\Gamma} (A_{\mathbf{n_1}} W_1) X \\
&+ \int_{\Omega_2} [A_0 W_2 + \sum_{k=1}^{d} A_k \partial_{x_k} W_2] X - \int_{\Gamma} (A_{\mathbf{n_2}} W_2) X \\
&= \int_{\Omega_1} [A_0 W_1 + \sum_{k=1}^{d} A_k \partial_{x_k} W_1] X \\
&+ \int_{\Omega_2} [A_0 W_2 + \sum_{k=1}^{d} A_k \partial_{x_k} W_2] X \\
&+ \int_{\Gamma} [A_{\mathbf{n}} W_2 - A_{\mathbf{n}} W_1] X
\end{aligned}$$

Si W_1 et W_2 sont les solution des problèmes locaux on en déduit :

$$\int_\Gamma [A_{\mathbf{n}} W_1 - A_{\mathbf{n}} W_2] X = 0 \quad \forall X \in V$$

Donc une condition nécessaire est suffisante pour que $W_1 + W_2$ soit la solution du problème global (1.13) est :

$$A_{\mathbf{n}} W_1 = A_{\mathbf{n}} W_2 \quad \text{on} \quad \Gamma \tag{1.16}$$

Puisque $A_{\mathbf{n}}$ est régulière, (1.16) implique que :

$$W_1 = W_2 \quad \text{sur} \quad \Gamma \tag{1.17}$$

par suite :

$$\left(T \Lambda_{\mathbf{n}}^{\pm} T^{-1}\right) W_1 = \left(T \Lambda_{\mathbf{n}}^{\pm} T^{-1}\right) W_2 \quad \text{sur} \quad \Gamma \tag{1.18}$$

ce qui donne :

$$\left\{ \begin{array}{lcl} A_{\mathbf{n}}^{-} W_1 & = & A_{\mathbf{n}}^{-} W_2 \\ & \text{et} & \\ A_{\mathbf{n}}^{+} W_1 & = & A_{\mathbf{n}}^{+} W_2 \end{array} \right. \tag{1.19}$$

Réciproquement, puisque $A_{\mathbf{n}} = A_{\mathbf{n}}^{+} + A_{\mathbf{n}}^{-}$, les conditions (1.19) impliquent (1.16), ce qui conclue la preuve.

Remarque 1 Les conditions de raccord découplées (1.19) sont utilisées comme conditions aux limites dans la résolution des problèmes locaux. Pour que les problèmes locaux soient bien posés, il suffit d'imposer les quantités associées aux caractéristiques entrantes dans un sous-domaine.

1.1.3 Algorithme de type Schwarz additif sans recouvrement

Pour simplifier la présentation, on suppose que le domaine de calcul est de forme rectangulaire, $\Omega = [x_a,x_b] \times [y_a,y_b]$, et on considère une décomposition sans recouvrement en bandes verticales où les sous-domaines sont définis par $\Omega_i =]\gamma_{i-1},\gamma_i[\times]y_a,y_b[$, $2 \leq i \leq N-1$, $\Omega_1 =]x_a,\gamma_1[\times]y_a,y_b[$, $\Omega_N =]\gamma_{N-1},x_b[\times]y_a,y_b[$ (voir la figure 1.1). Les interfaces ou les frontières artificielles sont les droites $x = \gamma_i$ sur lesquelles on va imposer les conditions de raccord. Les normales externes aux interfaces du sous-domaine Ω_i sont données par $\mathbf{n}_{i,l} = (-1,0)$ et $\mathbf{n}_{i,r} = (1,0)$. Par suite, $A_{\mathbf{n}_{i,l}}^{-} = -A_{\mathbf{n}_{i-1,r}}^{+} = -A^{+}$ et $A_{\mathbf{n}_{i-1,r}}^{-} = A^{-}$. On propose maintenant un algorithme de type Schwarz additif où les conditions posées aux interfaces entre sous-domaines voisins sont construites à partir des conditions (1.19).

ALGORITHME 1 Algorithme de type Schwarz additif sans recouvrement.

Soit $W_i^{(0)}$ pour $i = 1{,}2$ une approximation initiale de la solution du système d'équations d'Euler au pas de temps $n\Delta t$ dans le sous-domaine Ω_i. Les approximations $W_i^{(p+1)}$ pour $i = 1{,}2$ de W dans les sous-domaines Ω_i à l'itération $k+1$ (où k désigne l'itération de l'algorithme de type Schwarz) sont les solutions des problèmes :

$$\left\{\begin{array}{lcll} \mathcal{L}W_i^{(k+1)} & = & f & \text{dans } \Omega_i \\ A^+W_i^{(k+1)} & = & A^+W_{i-1}^{(k)} & \text{sur } \Gamma_{i,l} \\ A^-W_i^{(k+1)} & = & A^-W_{i+1}^{(k)} & \text{sur } \Gamma_{i,r} \\ A_{\mathbf{n}}^-W_i^{(k+1)} & = & A_{\mathbf{n}}^-g & \text{sur } \partial\Omega \cup \partial\Omega_i \end{array}\right. \tag{1.20}$$

où $\Gamma_{i,l}$ (respectivement $\Gamma_{i,r}$) désigne la droite $x = \gamma_{i-1}$ (respectivement $x = \gamma_i$). Par ailleurs, $W_i^{(0)} = W_i^n$ (où n est l'itération en temps) et $W_i^{n+1} = W_i^{(K)}$ où K est le nombre d'itérations de l'algorithme (1.20) pour arriver à la convergence.

Clairement, l'itération(1.20) définit un algorithme de type Schwarz additif puisque celle-ci repose sur une actualisation des valeurs à l'interface en utilisant celle de l'itération précédente dans chaque sous-domaine. En général les algorithmes sans recouvrement convergent moins bien que ceux avec domaines recouvrants mais ils présentent l'avantage de la facilité de l'implémentation informatique.

Ce type d'algorithme a été étudié par Nataf[Nat93] et Nataf *et al.*[NRdS94] pour des problèmes de convection-diffusion. En particulier, ces auteurs ont considéré l'utilisation de conditions d'ordre élevé, inspirées du concept de condition absorbante pour les calculs en domaine non-borné [EM77], afin d'améliorer la convergence d'un l'algorithme de type Schwarz de la forme 1.

FIG. 1.1 – *définition d'une décomposition en bandes verticales*

Pour les systèmes hyperboliques en dimension quelconque, on dispose notamment du résultat de convergence suivant qui a été démontré par Clerc[Cle97] :

Théorème 3 *On note par $E_i^{(k)} = W_i^{(k)} - W_i$, le vecteur erreur associé à la restriction de la solution globale du problème 1.13 au sous-domaine i. Alors la méthode de Schwarz converge au sens suivant :*

$$\begin{cases} \lim\limits_{k\to\infty} \|E_i^{(k)}\|_{L^2(\Omega)^p} = 0 \\ \lim\limits_{k\to\infty} \|\sum A_j \partial_j E_i^{(k)}\|_{L^2(\Omega)^p} = 0 \end{cases}$$

Dans ce qui suit, on rappelle brièvement la démonstration de ce théorème. Le vecteur erreur $E_i^{(k)}$ doit vérifier la version dite homogène de l'algorithme (1.20) :

$$\begin{cases} \mathcal{L}E_i^{(k+1)} & = & 0 & \text{dans } \Omega_i \\ A^+E_i^{(k+1)} & = & A^+E_{i-1}^{(k)} & \text{sur } \Gamma_{i,l} \\ A^-E_i^{(k+1)} & = & A^-E_{i+1}^{(k)} & \text{sur } \Gamma_{i,r} \\ A_{\mathbf{n}}^-E_i^{(k+1)} & = & 0 & \text{sur } \partial\Omega \cup \partial\Omega_i \end{cases}$$

En utilisant l'estimation (1.12) on obtient :

$$C_0\|E_i^{(k)}\|^2_{L^2(\Omega_i)} + \|A_{\mathbf{n}}^+E_i^{(k)}\|^2_{L^2_{1/A}(\partial\Omega_i)} \leq \|A_{\mathbf{n}}^-E_i^{(k)}\|^2_{L^2_{1/A}(\partial\Omega_i)} \tag{1.21}$$

On introduit les vecteurs :

$$\begin{aligned} F^{(k)} &= \sum_{i=1}^N \int_{\Omega_i} C_0 E_i^{(k)} \cdot E_i^{(k)} = \sum_{i=1}^N C_0 \|E_i^{(k)}\|^2_{L^2(\Omega_i)} \\ B^{(k)} &= \sum_{i=1}^N \left[\int_{\Gamma_{i,r}} \left(-A^-E_i^{(k)} \cdot E_i^{(k)} + A^+E_{i+1}^{(k)} \cdot E_{i+1}^{(k)} \right) + \int_{\Gamma_{i,l}} \left(A^+E_i^{(k)} \cdot E_i^{(k)} - A^-E_{i-1}^{(k)} \cdot E_{i-1}^{(k)} \right) \right] \end{aligned}$$

En observant que :

$$\begin{aligned} \int_{\Gamma_{i,l}} A^+E_i^{(k)} \cdot E_i^{(k)} &= \int_{\Gamma_{i,l}} A^+E_{i-1}^{(k-1)} \cdot E_{i-1}^{(k-1)} = \int_{\Gamma_{i-1,r}} A^+E_{i-1}^{(k-1)} \cdot E_{i-1}^{(k-1)} \\ \int_{\Gamma_{i,r}} A^-E_i^{(k)} \cdot E_i^{(k)} &= \int_{\Gamma_{i,r}} A^-E_{i+1}^{(k-1)} \cdot E_{i+1}^{(k-1)} = \int_{\Gamma_{i+1,l}} A^-E_{i+1}^{(k-1)} \cdot E_{i+1}^{(k-1)} \end{aligned}$$

et que $A^-E_i^{(k)}|_{\partial\Omega\cap\partial\Omega_i} = 0$, et si on somme (1.21) sur tous les sous-domaines, on obtient :

$$0 \leq F^{(k)} + B^{(k)} \leq B^{(k-1)}$$

ce qui montre que la suite $\{B^{(k)}\}$ est décroissante puisque $F^{(k)} > 0$, donc convergente et, par suite, $F^{(k)} \to 0$; $F^{(k)}$ est aussi le terme d'une série convergente :

$$0 \leq \sum_k^\infty F^{(k)} \leq B^{(0)}$$

ce qui prouve que $\|E_i^{(k)}\|_{L^2} \to 0\,\forall i$. Mais $\sum_j A_j \partial_j E_i^{(k)} = -E_i^{(k)}$ et donc ce dernier terme converge aussi vers zéro.

1.1.4 Construction d'un système interface

Dans cette section, on reformule l'algorithme (1.20) comme la résolution itérative d'un système interface dont les inconnues sont des flux normaux. Adoptant le formalisme de Nataf *et al.*[NRdS94], on introduit des opérateurs définis comme :

$$S^i : L^2_A(\Gamma_{i,l})^p \times L^2_A(\Gamma_{i,r})^p \times L^2(\Omega_i)^p \times L^2(\Omega_i)^p \quad \longrightarrow \quad L^2_{1/A}(\Gamma_{i,l})^p \times L^2_{1/A}(\Gamma_{i,r})^p$$

tel que :

$$S^i \;:\; (\Phi_{i,l}\;,\;\Phi_{i,r}\;,\;f\;,\;g) \quad \longrightarrow \quad \left(A^- W_i|_{\Gamma_{i,l}}\;,\;A^+ W_i|_{\Gamma_{i,r}}\right) \tag{1.22}$$

pour $2 \leq i \leq N-1$ et :

$$\begin{array}{rcl} S^1 & : & L^2_A(\Gamma_{1,r})^p \times L^2(\Omega_i)^p \times L^2(\Omega_i)^p \quad \longrightarrow \quad L^2_{1/A}(\Gamma_{1,r})^p \\ S^N & : & L^2_A(\Gamma_{N,l})^p \times L^2(\Omega_i)^p \times L^2(\Omega_i)^p \quad \longrightarrow \quad L^2_{1/A}(\Gamma_{N,l})^p \end{array}$$

avec :

$$\left\{ \begin{array}{rcl} S^1 & : & (\Phi_{1,r}\;,\;f\;,\;g) \quad \longrightarrow \quad A^+ W_1|_{\Gamma_{1,r}} \\ S^N & : & (\Phi_{N,l}\;,\;f\;,\;g) \quad \longrightarrow \quad A^- W_N|_{\Gamma_{N,l}} \end{array} \right.$$

pour $i = 1$ et $i = N$. Dans les expressions ci-dessus, $L^2_{1/A}(\Gamma)$ désigne l'espace dual de $L^2_A(\Gamma)$. Dans chaque cas, W_i est la solution du problème aux limites :

$$\left\{ \begin{array}{rcll} \mathcal{L} W_i & = & f & \text{dans } \Omega_i \\ A^+ W_i & = & \Phi_{i,l} & \text{sur } \Gamma_{i,l} \\ A^- W_i & = & \Phi_{i,r} & \text{sur } \Gamma_{i,r} \\ A^-_{\mathbf{n}} W_i & = & A^-_{\mathbf{n}} g & \text{sur } \partial\Omega \cup \partial\Omega_i \end{array} \right. \tag{1.23}$$

Les opérateurs d'interface ainsi définis sont linéaires et vérifient :

$$\left\{ \begin{array}{rclclcl} A^- W_i|_{\Gamma_{i,l}} & = & S^i(\Phi_{i,r},0,0,0)|_{\Gamma_{i,l}} & + & S^i(0,\Phi_{i,l},0,0)|_{\Gamma_{i,l}} & + & S^i(0,0,f,g)|_{\Gamma_{i,l}} \\ A^+ W_i|_{\Gamma_i} & = & S^i(\Phi_{i,r},0,0,0)|_{\Gamma_{i,r}} & + & S^i(0,\Phi_{i,l},0,0)|_{\Gamma_{i,r}} & + & S^i(0,0,f,g)|_{\Gamma_{i,r}} \end{array} \right.$$

Afin de simplifier le formalisme et d'isoler le rôle de chaque argument, on introduit les opérateurs suivants :

$$\left\{ \begin{array}{rcl} S^i_{rr}, S^i_{rl} & : & L^2_A(\Gamma_{i,l})^p \quad \longrightarrow \quad L^2_{1/A}(\Gamma_{i,l})^p \\ S^i_{lr}, S^i_{ll} & : & L^2_A(\Gamma_{i,r})^p \quad \longrightarrow \quad L^2_{1/A}(\Gamma_{i,r})^p \end{array} \right.$$

tels que :

$$\left\{ \begin{array}{rcl} S^i_{rr}(\Phi_{i,r}) & = & S^i(\Phi_{i,r}\;,\;0\;,\;0\;,\;0)|_{\Gamma_{i,l}} \\ S^i_{rl}(\Phi_{i,l}) & = & S^i(0\;,\;\Phi_{i,l}\;,\;0\;,\;0)|_{\Gamma_{i,l}} \\ S^i_{lr}(\Phi_{i,r}) & = & S^i(\Phi_{i,r}\;,\;0\;,\;0\;,\;0)|_{\Gamma_{i,r}} \\ S^i_{rl}(\Phi_{i,l}) & = & S^i(0\;,\;\Phi_{i,l}\;,\;0\;,\;0)|_{\Gamma_{i,r}} \end{array} \right.$$

On peut alors reformuler l'algorithme de type Schwarz en termes des flux $\Phi_{i,l}$ et $\Phi_{i,r}$ définis précédemment :

$$\Phi^{(k+1)} = \mathcal{S}(\Phi^{(k)}) + \mathcal{G} \tag{1.24}$$

où :

$$\Phi = (\Phi_{1,r},...,\Phi_{N-1,r},\Phi_{2,l},...,\Phi_{N,l})^T$$

$$\mathcal{S} = \begin{bmatrix} 0 & S^2_{rr} & \cdots & S^2_{rl} & \cdots & 0 \\ 0 & 0 & S^3_{rr} & \cdots & S^3_{rl} & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & S_N \\ S_1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & S^2_{lr} & \cdots & S^2_{ll} & \cdots & 0 \\ \vdots & & & & & \vdots \\ 0 & \cdots & S^{N-1}_{lr} & \cdots & S^{N-1}_{ll} & 0 \end{bmatrix} \qquad \mathcal{G} = \begin{bmatrix} S^1(0,f,g)|_{\Gamma_{1,r}} \\ S^2(0,0,f,g)|_{\Gamma_{2,r}} \\ \vdots \\ S^{N-1}(0,0,f,g)|_{\Gamma_{N-1,r}} \\ S^2(0,0,f,g)|_{\Gamma_{2,l}} \\ \vdots \\ S^{N-2}(0,0,f,g)|_{\Gamma_{N-2,l}} \\ S^{N-1}(0,f,g)|_{\Gamma_{N-1,l}} \\ S^N(0,f,g)|_{\Gamma_{N,l}} \end{bmatrix}$$

L'algorithme (1.20) peut donc être interprété comme une méthode de relaxation de Jacobi appliquée au système interface $(I - \mathcal{S})(\Phi) = \mathcal{G}$. On peut tout aussi bien accélérer la résolution de ce système en lui appliquant une méthode de type Krylov comme par exemple la méthode GMRES[SS86].

Dans le cas d'une décomposition en deux sous-domaines, la convergence de l'algorithme (1.20) peut aussi être déduite du fait que les opérateurs d'interface construits ci-dessus sont des contractions; cette démonstration peut êêtre faite en adaptant une idée rencontrée dans le cas scalaire dans [GG93]. On donne ci-dessous les grandes lignes de cette démonstration.

Soit une décomposition $\Omega = \Omega_1 \cup \Omega_2$, $\Gamma = \Omega_1 \cap \Omega_2$ étant l'interface entre les deux sous-domaines. Comme précédemment, on note $\mathbf{n}$ la normale à l'interface dirigée de Ω_1 vers Ω_2. On définit les opérateurs S_1 et S_2 tels que :

$$\begin{cases} S_1\Phi_1 = A^+_{\mathbf{n}} W_1 \\ S_2\Phi_2 = A^-_{\mathbf{n}} W_2 \end{cases}$$

où W_1 et W_2 sont les solutions des problèmes aux limites suivants :

$$\begin{cases} \mathcal{L}W_1 = f_1 \quad \text{dans } \Omega_1 \\ A^-_{\mathbf{n}} W_1 = \Phi_1 \quad \text{sur } \Gamma \\ + \ \text{Conditions aux limites sur } \partial\Omega_1 \end{cases}$$

et :

$$\begin{cases} \mathcal{L}W_2 = f_2 \quad \text{dans } \Omega_2 \\ A^+_{\mathbf{n}} W_2 = \Phi_2 \quad \text{sur } \Gamma \\ + \ \text{Conditions aux limites sur } \partial\Omega_2 \end{cases}$$

Dans le cas d'un algorithme de type Schwarz additif, on cherche à obtenir $\Phi = (\Phi_1, \Phi_2)^T$ par une itération de la forme (compte tenu de la définition de la normale $\mathbf{n}$) :

$$\left\{ \begin{array}{lllllll} A_{\mathbf{n}}^- W_1^{(k+1)} & = & \Phi_1^{(k+1)} & = & A_{\mathbf{n}}^- W_2^{(k)} & = & S_2 \Phi_2^{(k)} \\ A_{\mathbf{n}}^+ W_2^{(k+1)} & = & \Phi_2^{(k+1)} & = & A_{\mathbf{n}}^+ W_1^{(k)} & = & S_1 \Phi_1^{(k)} \end{array} \right. \tag{1.25}$$

Pour cela on définit un opérateur similaire à celui de Steklov-Poincaré :

$$S : L_A^2(\Gamma)^p \times L_A^2(\Gamma)^p \to L_{1/A}^2(\Gamma)^p \times L_{1/A}^2(\Gamma)^p$$

par les relations suivantes :

$$\begin{array}{rcl} (S\Phi)|_{\Gamma_1} & = & \Phi_1 - S_2 \Phi_2 \\ (S\Phi)|_{\Gamma_2} & = & \Phi_2 - S_1 \Phi_1 \end{array}$$

où on a noté par Γ_i l'interface Γ en tant que partie du domaine Ω_i. En utilisant les relations (1.25) et la forme (1.24) de l'itération interface, on voit que l'on est conduit à poser le système interface suivant :

$$S\Phi = \begin{pmatrix} I & -S_2 \\ -S_1 & I \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \Phi_2 \end{pmatrix} = 0 \quad \text{sur } \Gamma$$

On remarque par ailleurs que l'on peut aussi écrire (1.25) sous la forme :

$$\left\{ \begin{array}{lllll} \Phi_1^{(k+1)} & = & S_2 \Phi_2^{(\frac{k+1}{2})} & = & S_2 S_1 \Phi_1^{(k)} \\ \Phi_2^{(k+1)} & = & S_1 \Phi_1^{(\frac{k+1}{2})} & = & S_1 S_2 \Phi_2^{(k)} \end{array} \right. \tag{1.26}$$

c'est-à-dire que l'on peut aussi interpréter notre algorithme de Schwarz additif comme une itération de type Richardson appliquée au système :

$$\bar{S}\Phi = \begin{pmatrix} I - S_2 S_1 & 0 \\ 0 & I - S_1 S_2 \end{pmatrix} \Phi = \begin{pmatrix} I & S_2 \\ S_1 & I \end{pmatrix} \cdot \begin{pmatrix} I & -S_2 \\ -S_1 & I \end{pmatrix} \Phi = TS\Phi = 0 \tag{1.27}$$

qui peut lui-même être interprété comme un système interface d'opérateur S préconditionné par l'opérateur T.

Remarque 2 *On peut démontrer que les opérateurs $S_2 S_1$ et $S_1 S_2$ sont des contractions, les itérations (1.26) étant alors toujours convergentes. Ceci résulte de l'utilisation de l'estimation d'énergie du théorème 1 dans le cas $f = 0$ (le cas où f est non-nul peut être traité d'une façon analogue) :*

$$\left\{ \begin{array}{lllll} \|S_1 \Phi_1\|^2_{L^2_{1/A}(\Gamma)} & = & \displaystyle\int_\Gamma A_{\mathbf{n}}^+ W_1 \cdot W_1 d\sigma < -\int_\Gamma A_{\mathbf{n}}^- W_1 \cdot W_1 d\sigma & = & \|\Phi_1\|^2_{L^2_{1/A}(\Gamma)} \\ \|S_2 \Phi_2\|^2_{L^2_{1/A}(\Gamma)} & = & \displaystyle -\int_\Gamma A_{\mathbf{n}}^- W_2 \cdot W_2 d\sigma < \int_\Gamma A_{\mathbf{n}}^+ W_2 \cdot W_2 d\sigma & = & \|\Phi_2\|^2_{L^2_{1/A}(\Gamma)} \end{array} \right.$$

Dans [GG93] on trouve un résultat plus fort, démontré dans le cas scalaire : il existe $\eta > 0$ dépendant des domaines Ω_i et des opérateurs A_i t.q. :

$$\begin{cases} \|S_2 S_1 \Phi_1\|^2_{L^2_{1/A}(\Gamma)} \leq \dfrac{1}{\eta+1} \|\Phi_1\|^2_{L^2_{1/A}(\Gamma)} \\ \|S_1 S_2 \Phi_2\|^2_{L^2_{1/A}(\Gamma)} \leq \dfrac{1}{\eta+1} \|\Phi_2\|^2_{L^2_{1/A}(\Gamma)} \end{cases} \tag{1.28}$$

On déduit des délations (1.26) et (1.28) que :

$$\|\Phi^{(k+1)}\|^2_{L^{-1/2}_A(\Gamma)} \leq \frac{1}{(\eta+1)^{(k+1)}} \|\Phi_0\|^2_{L^{-1/2}_A(\Gamma)}$$

1.2 Analyse de convergence

Dans la section précedente on a vu que l'algorithme de type Schwarz additif sans recouvrement appliqué aux équations d'Euler est convergent. Dans ce qui suit on essaie d'evaluer quantitativement le taux de convergence dans les cas des équations aux cœfficients constants. Cette étude est menée en une, deux et trois dimensions d'espace, pour une décomposition en deux sous-domaines, en appliquant une analyse de Fourier aux équations linéarisées (pour les dimensions supérieures à deux). En dépit du fait que les conditions appliquées aux interfaces sont des plus simples, on démontre que l'algorithme converge et on met en évidence, dans des situations d'écoulement particulières, un taux de convergence optimal. Différentes simulations numériques portant sur la résolution du système d'équations d'Euler bidimensionnelles, permettent d'exhiber un comportement qualitativement en accord avec les résultats de l'analyse.

1.2.1 Le cas unidimensionnel

On considère le système d'équations d'Euler 1D en variables primitives, linéarisées autour d'un état constant $W^* = (\bar{\rho}\ ,\ \bar{u}\ ,\ \bar{p})$ et résolues sur un intervalle $[a\ ,\ b]$ de la droite réelle :

$$\beta W + A(\bar{W})\frac{\partial W}{\partial x} = f \quad \Rightarrow \quad \beta \begin{pmatrix} \rho \\ u \\ p \end{pmatrix} + \begin{pmatrix} \bar{u} & \bar{\rho} & 0 \\ 0 & \bar{u} & \dfrac{1}{\bar{\rho}} \\ 0 & \bar{\rho}\bar{c}^2 & \bar{u} \end{pmatrix} \partial_x \begin{pmatrix} \rho \\ u \\ p \end{pmatrix} = f \tag{1.29}$$

où $W = (\rho\ ,\ u\ ,\ p)$ est le vecteur des variables primitives. On fait l'hypothèse que $0 < \bar{u} < \bar{c}$. En ce qui concerne les conditions aux limites en a et b, pour que le problème aux limites formé avec le système considéré soit bien posé, il suffit selon [GR96] d'imposer $(\rho\ ,\ u)$ en $x = a$ (par exemple) et ρ en $x = b$. On passe ensuite en variables caractéristiques afin de découpler complètement les équations de transport pour chaque variable. On note les nouvelles variables par $\bar{W} = T^{-1}W$ où :

$$
\begin{aligned}
A(\bar{W}) &= \begin{pmatrix} \bar{u} & \bar{\rho} & 0 \\ 0 & \bar{u} & \dfrac{1}{\bar{\rho}} \\ 0 & \bar{\rho}\bar{c}^2 & \bar{u} \end{pmatrix} = T(\bar{W})\Lambda(\bar{W})T^{-1}(\bar{W}) \\
&= \begin{pmatrix} 1 & 1 & 1 \\ -\dfrac{\bar{c}}{\bar{\rho}} & 0 & \dfrac{\bar{c}}{\bar{\rho}} \\ \bar{c}^2 & 0 & \bar{c}^2 \end{pmatrix} \mathrm{diag}(\bar{u}-\bar{c}\ ,\ \bar{u}\ ,\ \bar{u}+\bar{c}) \begin{pmatrix} 0 & -\dfrac{\bar{\rho}}{2\bar{c}} & \dfrac{1}{2\bar{c}^2} \\ 1 & 0 & -\dfrac{1}{\bar{c}^2} \\ 0 & \dfrac{\bar{\rho}}{2\bar{c}} & \dfrac{1}{2\bar{c}^2} \end{pmatrix}
\end{aligned}
$$

et on obtient :

$$
\begin{cases}
\bar{W}_1 &= -\dfrac{\rho}{2\bar{c}}u + \dfrac{1}{2\bar{c}^2}p \\
\bar{W}_2 &= \rho - \dfrac{1}{\bar{c}^2}p \\
\bar{W}_3 &= \dfrac{\rho}{2\bar{c}}u + \dfrac{1}{2\bar{c}^2}p
\end{cases}
$$

On se place maintenant dans un contexte de résolution par sous-domaines non recouvrants où les conditions de raccord à l'interface sont de type classique, c'est-à-dire qu'on impose la continuité des variables caractéristiques à l'interface [QS96].

Remarque 3 *On sait également que si au lieu d'imposer les quantités physiques aux frontières on écrit des conditions aux limites en variables caractéristiques, alors le système est équivalent à trois équations scalaires pour lesquelles les conditions aux limites et aux interfaces sont aussi découplées. Ceci conduit à une convergence de l'algorithme de Schwarz en un nombre d'itérations égal au nombre de sous-domaines (cas dans lequel l'analyse de convergence qui suit, pour deux sous-domaines, n'a pas de sens, le taux de convergence étant alors nul).*

Soit une décomposition en deux sous-domaines non-recouvrants $[a\ ,\ \gamma]$ et $[\gamma\ ,\ b]$. Pour l'étude de convergence, on écrit des équations locales en termes des vecteurs erreurs $(e^i)^{(p)} = (\bar{W}^i)^{(p)} - \bar{W}^i$ où l'indice i désigne le sous-domaine et p l'itération de l'algorithme de Schwarz. Les conditions aux limites transcrites en termes des composantes des vecteurs erreur deviennent :

$$
\begin{cases}
(e_1+e_2+e_3)(a) = 0 \\
(e_3-e_1)(a) = 0
\end{cases}
\qquad (e_1+e_2+e_3)(b) = 0 \tag{1.30}
$$

Le système étant transcrit en variables caractéristiques, dans l'algorithme de Schwarz on retrouve des conditions de type Dirichlet pour les composantes de l'erreur correspondant aux valeurs propres négatives ou positives selon le sous-domaine considéré. Dans la suite, on note l'état constant par $(\rho\ ,\ u\ ,\ p)$, puisqu'il n'y a plus de confusion dans les notations. L'algorithme de Schwarz conduit à résoudre dans chaque sous-domaine les équations suivantes :

$$\Omega_1 = [a\ ,\ \gamma]$$

$$\left\{\begin{array}{lcl} \beta(e_1^1)^{(p+1)} - (c-u)(e_1^1)_x^{(p+1)} & = & 0 \text{ dans } [a\ ,\ \gamma] \\ \beta(e_2^1)^{(p+1)} + u(e_2^1)_x^{(p+1)} & = & 0 \text{ dans } [a\ ,\ \gamma] \\ \beta(e_3^1)^{(p+1)} + (c+u)(e_3^1)_x^{(p+1)} & = & 0 \text{ dans } [a\ ,\ \gamma] \\ (e_1^1 + e_2^1 + e_3^1)^{(p+1)}(a) & = & 0 \\ (e_3^1 - e_1^1)^{(p+1)}(a) & = & 0 \\ (e_1^1)^{(p+1)}(\gamma) & = & (e_1^2)^{(p)}(\gamma) \end{array}\right. \tag{1.31}$$

$$\Omega_2 = [\gamma\ ,\ b]$$

$$\left\{\begin{array}{lcl} \beta(e_1^1)^{(p+1)} - (c-u)(e_1^2)_x^{(p+1)} & = & 0 \text{ dans } [\gamma\ ,\ b] \\ \beta(e_2^2)^{(p+1)} + u(e_2^2)_x^{(p+1)} & = & 0 \text{ dans } [\gamma\ ,\ b] \\ \beta(e_3^2)^{(p+1)} + (c+u)(e_3^2)_x^{(p+1)} & = & 0 \text{ dans } [\gamma\ ,\ b] \\ (e_1^2 + e_2^2 + e_3^2)^{(p+1)}(b) & = & 0 \\ (e_2^2)^{(p+1)}(\gamma) & = & (e_2^1)^{(p)}(\gamma) \\ (e_3^2)^{(p+1)}(\gamma) & = & (e_3^1)^{(p)}(\gamma) \end{array}\right.$$

où les dernières relations traduisent la transmission de l'information à l'interface par des conditions de type Dirichlet. En résolvant les problèmes dans les deux sous-domaines on obtient l'itération à l'interface :

$$\left\{\begin{array}{l} (e_2^2)^{(p+1)}(\gamma) = -2(e_1^1)^{(p)}(\gamma)e^{-\dfrac{\beta c}{u(c-u)}(\gamma - a)} \\ (e_3^2)^{(p+1)}(\gamma) = (e_1^1)^{(p)}(\gamma)e^{-\dfrac{2\beta c}{c^2-u^2}(\gamma - a)} \end{array}\right. \tag{1.32}$$

$$(e_1^1)^{(p+1)}(\gamma) = -(e_2^2)^{(p)}(\gamma)e^{-\dfrac{\beta c}{u(c-u)}(b-\gamma)} - (e_3^2)^{(p)}(\gamma)e^{-\dfrac{2\beta c}{c^2-u^2}(b-\gamma)}$$

que l'on peut ensuite exprimer sous forme matricielle :

$$\begin{pmatrix} (e_1^1)(\gamma) \\ (e_2^2)(\gamma) \\ (e_3^2)(\gamma) \end{pmatrix}^{(p+1)} = \mathcal{S} \begin{pmatrix} (e_1^1)(\gamma) \\ (e_2^2)(\gamma) \\ (e_3^2)(\gamma) \end{pmatrix}^{(p)}$$

où la matrice d'itération est :

$$\mathcal{S} = \begin{pmatrix} 0 & -e^{-\frac{\beta c}{u(c-u)}(b-\gamma)} & -e^{-\frac{2\beta c}{c^2-u^2}(b-\gamma)} \\ -2e^{-\frac{\beta c}{u(c-u)}(\gamma-a)} & 0 & 0 \\ e^{-\frac{2\beta c}{c^2-u^2}(\gamma-a)} & 0 & 0 \end{pmatrix} \tag{1.33}$$

Le taux de convergence de l'algorithme de Schwarz est défini comme le rayon spectral de la matrice d'itération ci-dessus :

$$\rho^2_{\text{Schwarz}} = \rho(\mathcal{S}) = \left| 2e^{-\frac{\beta c}{u(c-u)}(b-a)} - e^{-\frac{2\beta c}{c^2-u^2}(b-a)} \right| \tag{1.34}$$

On vérifie facilement que $\rho^2_{\text{Schwarz}} < 1$ et donc l'algorithme de Schwarz est convergent. En reprenant la même démarche pour le système discrétisé, dans le contexte d'une méthode de différences finies basée sur une schéma décentré du premier ordre, on obtient l'itération matricielle à l'interface :

$$\begin{pmatrix} e_1^2 \\ e_2^1 \\ e_3^1 \end{pmatrix}^{(p+1)} = \mathcal{S}_\Gamma \begin{pmatrix} e_1^2 \\ e_2^1 \\ e_3^1 \end{pmatrix}^{(p)}$$

avec :

$$\mathcal{S}_d = \begin{pmatrix} 0 & -\left(\frac{\lambda_1'\lambda_2'}{(h+\lambda_1')(h+\lambda_2')}\right)^K & -\left(\frac{\lambda_1'\lambda_3'}{(h+\lambda_1')(h+\lambda_3')}\right)^K \\ -2\left(\frac{\lambda_1'\lambda_2'}{(h+\lambda_1')(h+\lambda_2')}\right)^L & 0 & 0 \\ \left(\frac{\lambda_1'\lambda_3'}{(h+\lambda_1')(h+\lambda_3')}\right)^L & 0 & 0 \end{pmatrix}$$

où $\lambda_i' = \dfrac{|\lambda_i|/\beta}{h+|\lambda_i|/\beta}$; les λ_i sont les valeurs propres de A, $h = \dfrac{(b-a)}{N}$ est le pas de la discrétisation par différences finies, L et K étant les nombres de points respectivement dans le premier et deuxième sous-domaine, $L + K = N + 1$. Le taux de convergence discret est donné par :

$$\rho^2_{\text{Schwarz discret}}(N) = \rho(\mathcal{S}_d) = \left| 2\left(\frac{\lambda_1'\lambda_2'}{(h+\lambda_1')(h+\lambda_2')}\right)^{N+1} - \left(\frac{\lambda_1'\lambda_3'}{(h+\lambda_1')(h+\lambda_3')}\right)^{N+1} \right| \tag{1.35}$$

ou bien, si on suppose $b - a = \beta = 1$ et si on note $\mathcal{M} = \dfrac{u}{c}$:

$$\rho^2_{\text{Schwarz discret}}(\mathcal{M},N) = \left| 2\left(\frac{N^2\mathcal{M}(1-\mathcal{M})}{1+N^2\mathcal{M}(1-\mathcal{M})+N}\right)^N - \left(\frac{N^2(1-\mathcal{M}^2)}{1+N^2(1-\mathcal{M}^2)+2N}\right)^N \right| \tag{1.36}$$

On vérifie de nouveau facilement que $\rho^2_{\text{Schwarz discret}}(\mathcal{M},N) < 1$ et donc l'algorithme de Schwarz appliqué au système d'équations d'Euler discrétisé est convergent. La figure 1.2 représente l'évolution du taux de convergence en fonction du nombre de points N de la discrétisation, pour différentes valeurs du nombre de Mach de l'écoulement $\mathcal{M}$. On note que le taux de convergence tend très rapidement (dès que $N > 5$) vers une valeur asymptotique indépendamment de la valeur de $\mathcal{M}$.

FIG. 1.2 – *Système d'équations d'Euler 1D discrétisé*
Taux de convergence de l'algorithme de Schwarz (décomposition en deux sous-domaines)

1.2.2 Le cas bidimensionnel

La forme conservative du système d'équations d'Euler qui modélisent l'écoulement d'un fluide compressible en dimension deux est donnée par :

$$\frac{\partial W}{\partial t} + \vec{\nabla}.\vec{\mathcal{F}}(W) = 0 \quad , \quad W = \left(\rho \ , \ \rho\vec{U} \ , \ E\right)^T \quad , \quad \vec{\nabla} = \left(\frac{\partial}{\partial x} \ , \ \frac{\partial}{\partial y}\right)^T \tag{1.37}$$

où $W = W(\vec{x},t)$, $\vec{x}$ et t désignent respectivement les variables d'espace et temporelle; $\vec{\mathcal{F}}(W)$ désigne le flux conservatif dont les composantes sont données par :

$$F_x(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(E+p) \end{pmatrix} \quad , \quad F_y(W) = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}$$

Dans les expressions ci-dessus, ρ est la densité, $\vec{U} = (u \ , \ v)^T$ est le vecteur vitesse, E est l'énergie totale par unité de volume, p est la pression. La pression est déduite des autres variables en utilisant la loi d'état des gaz parfaits :

$$p = (\gamma - 1)(E - \frac{1}{2}\rho \parallel \vec{U} \parallel^2)$$

où γ est le rapport des chaleurs spécifiques. On suppose que ces équations sont intégrées en temps à l'aide d'une méthode d'Euler implicite linéarisée (la section 2.1.4 décrit une telle approche dans un contexte discret particulier). Les équations sont donc linéarisées autour d'un état $\bar{W}$. On peut alors écrire la forme non-conservative suivante des équations :

$$\mathcal{L}W := \frac{1}{\Delta t}W + A_1\partial_x W + A_2\partial_y W = f \tag{1.38}$$

où $A_1 = A_1(\bar{W})$ et $A_2 = A_2(\bar{W})$ sont les matrices Jacobiennes des flux :

$$A_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2}(\gamma-3)u^2 + \frac{1}{2}(\gamma-1)v^2 & -(\gamma-3)u & -(\gamma-1)v & \gamma-1 \\ -uv & v & u & 0 \\ u\frac{(\gamma-1)(\gamma-2)(u^2+v^2)-2c^2}{2(\gamma-1)} & (\frac{3}{2}-\gamma)u^2 + \frac{1}{2}v^2 + \frac{c^2}{\gamma-1} & -(\gamma-1)uv & \gamma u \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ -uv & v & u & 0 \\ \frac{1}{2}(\gamma-1)u^2 + \frac{1}{2}(\gamma-3)v^2 & -(\gamma-1)u & -(\gamma-3)v & \gamma-1 \\ v\frac{(\gamma-1)(\gamma-2)(u^2+v^2)-2c^2}{2(\gamma-1)} & -(\gamma-1)uv & (\frac{3}{2}-\gamma)v^2 + \frac{1}{2}u^2 + \frac{c^2}{\gamma-1} & \gamma v \end{pmatrix}$$

On peut transformer ce système en lui applicant une matrice T (on peut toujours supposer qu'une telle matrice existe[Kro91]) telle que :

$$\begin{cases} B_1 = T^{-1}A_1T = \text{diag}(\sigma_1\ ,\ \sigma_2\ ,\ \sigma_3\ ,\ \sigma_4) \\ B_2 = T^{-1}A_2T \quad \text{est une matrice symétrique} \end{cases}$$

où les σ_i sont les valeurs propres de la matrice A_1. En utilisant le changement de variables $\widetilde{W} = T^{-1}W$ on obtient une version symétrisée du système (2.3) :

$$\widetilde{\mathcal{L}}\ \widetilde{W} = \beta\widetilde{W} + B_1\partial_x\widetilde{W} + B_2\partial_y\widetilde{W} = T^{-1}f \quad , \quad \beta = \frac{1}{\Delta t} \tag{1.39}$$

Dans le cas du système d'équations d'Euler la matrice T est donnée par :

$$T = \begin{pmatrix} 1 & 1 & 0 & 1 \\ u-c & u+c & 0 & u \\ v & v & c\sqrt{2} & v \\ \frac{1}{2}(u^2+v^2) - cu + \frac{c^2}{\gamma-1} & \frac{1}{2}(u^2+v^2) + cu + \frac{c^2}{\gamma-1} & cv\sqrt{2} & \frac{1}{2}(u^2+v^2) \end{pmatrix}$$

et les matrices du système symétrisé ont les expressions suivantes :

$$B_1 = \text{diag}(u-c\ ,\ u+c\ ,\ u\ ,\ u) \qquad B_2 = \begin{pmatrix} v & 0 & \frac{c}{\sqrt{2}} & 0 \\ 0 & v & \frac{c}{\sqrt{2}} & 0 \\ \frac{c}{\sqrt{2}} & \frac{c}{\sqrt{2}} & v & 0 \\ 0 & 0 & 0 & v \end{pmatrix}$$

Dans la suite, on note simplement W (respectivement $\mathcal{L}$) au lieu de $\widetilde{W}$ (respectivement $\widetilde{\mathcal{L}}$). On réalise une étude de la convergence de l'algorithme de Schwarz appliqué au système symétrique linéarisé (1.39). On se situe donc dans le cadre théorique de la section 1.1.

On considère le cas d'une décomposition en deux sous-domaines caractérisée par $\Omega_1 = \mathbb{R}_- \times \mathbb{R}$ et $\Omega_2 = \mathbb{R}_+ \times \mathbb{R}$ séparés par l'interface $x = 0$; $\mathbf{n} = (\mathbf{1},\mathbf{0})$ désigne le vecteur normal à l'interface $x = 0$, dirigée de Ω_1 vers Ω_2. Soient :

$$\begin{cases} M_n &= \dfrac{\mathbf{V.n}}{c} = \dfrac{u}{c} \\ M_t &= \dfrac{\mathbf{V.t}}{c} = \dfrac{v}{c} \end{cases}$$

respectivement les nombres de Mach normal et tangentiel à l'interface $x = 0$. Par ailleurs, en tout point de $\Omega_1 \bigcup \Omega_2$, le nombre de Mach local peut s'exprimer par :

$$M = \frac{\sqrt{u^2+v^2}}{c} = \sqrt{M_u^2 + M_v^2}$$

On note par $(E_i^{(p)})(x) = (W_i^{(p)} - W_i)(x)$ le vecteur erreur dans le sous-domaine Ω_i, $i = 1,2$, à l'itération p de l'algorithme de Schwarz (1.20). Cet algorithme, formulé en termes des vecteurs d'erreur locaux, s'écrit simplement comme :

$$\begin{array}{lll} \Omega_1 & : & \begin{cases} \mathcal{L}E_1^{(p+1)} &= 0 \text{ pour } x < 0 \\ (E_1^{(p+1)})_j &= (E_2^{(p)})_j \text{ pour } \sigma_j < 0\ ,\ x = 0 \\ E_1^{(p+1)} & \quad \text{borné quand } x \longrightarrow -\infty \end{cases} \\ \Omega_2 & : & \begin{cases} \mathcal{L}E_2^{(p+1)} &= 0 \text{ pour } x > 0 \\ (E_2^{(p+1)})_j &= (E_1^{(p)})_j \text{ pour } \sigma_j > 0\ ,\ x = 0 \\ E_2^{(p+1)} & \quad \text{borné quand } x \longrightarrow +\infty \end{cases} \end{array} \tag{1.40}$$

où $(E_i^{(p+1)})_j$ désigne la $j$ème composante du vecteur d'erreur $E_i^{(p+1)}$. Pour évaluer le taux de convergence de l'algorithme de Schwarz, on doit résoudre des problèmes aux limites locaux. Dans le cas présent où l'on considère la résolution du système d'équations d'Euler, ceci ne peut être fait directement. L'outil

mathématique qui va nous permettre de contourner cette difficulté est l'analyse de Fourier. Plus précisément, on applique à (1.40) une transformée de Fourier dans la direction tangentielle à l'interface, y dans ce cas (la variable de Fourier est notée k). Soit :

$$\hat{E}(x,k) = \mathcal{F}E(x,y) = \int_{\mathbb{R}} e^{-iky} E(x,y) dy$$

le symbole de Fourier du vecteur erreur. On note au passage que cette démarche est justifiée si les matrices B_1 et B_2 sont constantes ce qui est bien le cas ici puisqu'elles résultent d'une linéarisation autour d'un état $\bar{W}$. Ceci conduit à :

$$\begin{array}{lll} \Omega_1 & : & \left\{ \begin{array}{lll} \dfrac{d}{dx}\hat{E}_1^{(p+1)} & = & -M(k)\hat{E}_1^{(p+1)} \\ M(k) & = & B_1^{-1}(\beta \mathrm{Id} + ikB_2) \\ (\hat{E}_1^{(p+1)})_j & = & (\hat{E}_2^{(p)})_j \text{ pour } \sigma_j < 0 \ , \ x = 0 \end{array} \right. \\ \Omega_2 & : & \left\{ \begin{array}{lll} \dfrac{d}{dx}\hat{E}_2^{(p+1)} & = & -M(k)\hat{E}_2^{(p+1)} \\ M(k) & = & B_1^{-1}(\beta \mathrm{Id} + ikB_2) \\ (\hat{E}_2^{(p+1)})_j & = & (\hat{E}_1^{(p)})_j \text{ pour } \sigma_j > 0 \ , \ x = 0 \end{array} \right. \end{array} \tag{1.41}$$

Dans le cas du système d'équations d'Euler on a :

$$M(k) = \begin{pmatrix} \dfrac{a}{u-c} & 0 & \dfrac{ikc}{\sqrt{2}(u-c)} & 0 \\ 0 & \dfrac{a}{u+c} & \dfrac{ikc}{\sqrt{2}(u+c)} & 0 \\ \dfrac{ikc}{\sqrt{2}u} & \dfrac{ikc}{\sqrt{2}u} & \dfrac{a}{u} & 0 \\ 0 & 0 & 0 & \dfrac{a}{u} \end{pmatrix}$$

avec $a = \beta + ikv$. Les valeurs propres et les vecteurs propres de la matrice $M(k)$ ont les expressions suivantes :

$$\lambda_1(k) = \frac{-au - cR(k)}{c^2 - u^2}, \qquad V_1(k) = \begin{pmatrix} -\dfrac{(R(k)+a)(c+u)}{\sqrt{2}} \\ \dfrac{(R(k)-a)(c-u)}{\sqrt{2}} \\ ik(c^2-u^2) \\ 0 \end{pmatrix}$$

$$\lambda_2(k) = \frac{-au + cR(k)}{c^2 - u^2}, \qquad V_2(k) = \begin{pmatrix} \dfrac{(R(k)-a)(c+u)}{\sqrt{2}} \\ -\dfrac{(R(k)+a)(c-u)}{\sqrt{2}} \\ ik(c^2-u^2) \\ 0 \end{pmatrix}$$

$$\lambda_{3,4}(k) = \frac{a}{u}, \qquad V_3(k) = \begin{pmatrix} -\dfrac{iku}{\sqrt{2}} \\ \dfrac{iku}{\sqrt{2}} \\ a \\ 0 \end{pmatrix}, \; V_4(k) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

où $R(k) = \sqrt{a^2 + k^2(c^2 - u^2)}$ (on vérifie aisément que les quantités ci-dessus sont inchangées si on choisit la racine négative de la quantité auxiliaire $a^2 + k^2(c^2 - u^2)$). Il en résulte des problèmes locaux qui sont de simples équations différentielles ordinaires dont les solutions peuvent être exprimées comme des combinaisons linéaires des vecteurs propres de $M(k)$:

$$\hat{E}_i(x,k) = \sum_{j=1}^{4} \alpha_j^i e^{-\lambda_j(k)x} V_j(k) \tag{1.42}$$

où les $\lambda_j(k)$ sont les valeurs propres de $M(k)$ et en supposant que les vecteurs propres $V_j(k)$ sont linéairement indépendants.

Remarque 4 *Un calcul simple montre que cette hypothèse n'est pas vérifiée seulement lorsque $v = 0$ pour $k^2 = \dfrac{\beta^2}{u^2}$ (dans ce cas, deux des valeurs propres coïncident et la matrice $M(k)$ n'est pas diagonalisable par vecteurs propres). Pour obtenir un résultat général valable dans tous les cas de figure, il faudrait appliquer rigoureusement l'approche proposée dans [EM77]. Ici , le cas posant problème étant spécifique à une longueur d'onde particulière, on va quand même adopter cette approche en supposant que $k^2 \neq \dfrac{\beta^2}{u^2}$.*

Dans ce qui suit on impose que les solutions locales soient bornées à l'infini (à $-\infty$ pour $\hat{E}_1$ et $+\infty$ pour $\hat{E}_2$). On en déduit que dans les décompositions de $\hat{E}_1(x,k)$ (respectivement $\hat{E}_2(x,k)$) on ne doit retenir que les vecteurs propres associés à des valeurs propres négatives (respectivement positives). On peut supposer par exemple que $u > 0$ ce qui implique $M_n > 0$, le cas $u < 0$ pouvant en être déduit.

Théorème 4 *Dans l'hypothèse où $M_n > 0$, le signe des parties réelles des valeurs propres dépend de la valeur de M_n comme suit :*

$$\begin{cases} 0 < M_n < 1 \Rightarrow \Re(\lambda_1) < 0, \Re(\lambda_{2,3,4}) > 0 \\ M_n > 1 \Rightarrow \Re(\lambda_{1,2,3,4}) > 0 \end{cases}$$

Démonstration. On vérifie que $\Re(\lambda_{3,4}) > 0, \forall M_n$. D'autre part, les calculs conduisent à des formules de $\Re(\lambda_1)$ et $\Re(\lambda_2)$ pour $M_n \neq 1$:

$$\begin{cases} \Re(\lambda_{1,2}) & = & -\dfrac{\beta M_n \pm \dfrac{1}{\sqrt{2}}\sqrt{A+\sqrt{B}}}{c(1-M_n^2)} \\ A & = & \beta^2 + k^2(c^2 - u^2 - v^2) \\ B & = & A^2 + 4k^2\beta^2 v^2 \end{cases}$$

En effet, on voit que $\Re(\lambda_1)$ est de signe opposé à $1 - M_n^2$. Par ailleurs, si on suppose que $\Re(\lambda_2)$ peut changer de signe suivant la valeur de k pour M_n et M_t donnés, cela implique qu'il existe une valeur de k tel que $\beta M_n = \dfrac{1}{\sqrt{2}}\sqrt{A+\sqrt{B}}$, c'est-à-dire que $(c^2M^2k^2 + \beta^2 M_n^2)(1 - M_n^2) = 0$, ce qui est impossible dans le domaine considéré ($M_n \neq 1$). Donc $\Re(\lambda_2(k))$ a forcément le signe de $\Re(\lambda_2(0))$ qui est positif.

On en déduit que le cas le plus intéressant du point de vue de l'étude de la convergence de l'algorithme de Schwarz est le cas d'un écoulement subsonique. Dans le cas d'un écoulement supersonique, la méthode converge en une itération et il n'est donc pas nécessaire d'évaluer le taux de convergence. Lorsque l'écoulement est subsonique, les solutions des problèmes locaux (1.42) deviennent :

$$\begin{cases} \hat{E}_1(x,k) & = & \alpha_1 e^{-\lambda_1 x} V_1(k) \\ \hat{E}_2(x,k) & = & \alpha_2 e^{-\lambda_2 x} V_2(k) + \alpha_3 e^{-\lambda_3 x} V_3(k) + \alpha_4 e^{-\lambda_4 x} V_4(k) \end{cases} \tag{1.43}$$

A partir des conditions d'interface utilisées dans (1.41) on en déduit pour $x = 0$:

$$\begin{cases} (\hat{E}_1^{(p+1)})_1 & = & (\hat{E}_2^{(p)})_1 \\ (\hat{E}_2^{(p+1)})_{2,3,4} & = & (\hat{E}_1^{(p)})_{2,3,4} \end{cases} \tag{1.44}$$

ce qui est équivalent à :

$$\begin{cases} \alpha_1^{(p+1)}(V_1(k))_1 = \alpha_2^{(p)}(V_2(k))_1 + \alpha_3^{(p)}(V_3(k))_1 + \alpha_4^{(p)}(V_4(k))_1 \\ \alpha_2^{(p+1)}(V_2(k))_{2,3,4} + \alpha_3^{(p+1)}(V_3(k))_{2,3,4} + \alpha_4^{(p+1)}(V_4(k))_{2,3,4} = \alpha_1^{(p)}(V_1(k))_{2,3,4} \end{cases} \tag{1.45}$$

Les relations ci-dessus conduisent aux itérations suivantes valables à l'interface $x = 0$:

$$\left\{\begin{array}{lcl} \alpha_1^{(p+1)} & = & \mathcal{T}_1 \begin{pmatrix} \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix}^{(p)} = \mathcal{T}_1\mathcal{T}_2\alpha_1^{(p-1)} \\ \begin{pmatrix} \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix}^{(p+1)} & = & \mathcal{T}_2\alpha_1^{(p)} = \mathcal{T}_2\mathcal{T}_1 \begin{pmatrix} \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix}^{(p-1)} \end{array}\right. \tag{1.46}$$

Le taux de convergence de l'algorithme de Schwarz, dans le cas d'une décomposition en deux sous-domaines, est donc donné par :

$$\rho^2_{\mathsf{Schwarz}} = \max\left(\rho(\mathcal{T}_1\mathcal{T}_2)\ ,\ \rho(\mathcal{T}_2\mathcal{T}_1)\right) \tag{1.47}$$

Les itérations (1.45) s'expriment encore comme :

$$\begin{array}{l} \alpha_1^{(p+1)} = \mathcal{T}_1 \begin{pmatrix} \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix}^{(p)} = -\left[\dfrac{R(k)-a}{R(k)+a}\right]\alpha_2^{(p)} + \left[\dfrac{iku}{(R(k)+a)(c+u)}\right]\alpha_3^{(p)} \\ \begin{pmatrix} \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix}^{(p+1)} = \mathcal{T}_2\alpha_1^{(p)} = \begin{pmatrix} \left[-\dfrac{(R(k)-a)(ac+uR(k))}{(R(k)+a)(ac-uR(k))}\right]\alpha_1^{(p)} \\ \left[\dfrac{2ik(c^2-u^2)(c-u)R(k)}{(R(k)+a)(ac-uR(k))}\right]\alpha_1^{(p)} \\ 0 \end{pmatrix} \end{array} \tag{1.48}$$

Nous pouvons calculer le taux de convergence à partir de l'équation (1.48) :

$$\rho^2_{\mathsf{Schwarz}}(k) = \left|\frac{R(k)-a}{(R(k)+a)^2}\cdot\frac{R(k)(c-3u)-a(c+u)}{c+u}\right| \tag{1.49}$$

Cette quantité dépend de 4 paramètres : β, k, M_n et M_t c'est-à-dire, du pas de temps, du paramètre de fréquence et des nombres de Mach normal et tangentiel à l'interface :

$$\left\{\begin{array}{l} \rho^2_{\mathsf{Schwarz}}(k) = \left|\dfrac{R_1(k)-a_1}{(R_1(k)+a_1)^2}\cdot\dfrac{R_1(k)(1-3M_n)-a_1(1+M_n)}{1+M_n}\right| \\ a_1 = \dfrac{\beta}{c} + ikM_t \\ R_1(k) = \sqrt{a_1^2 + k^2(1-M_n^2)} \end{array}\right. \tag{1.50}$$

Pour démontrer la convergence de l'algorithme de Schwarz c'est-à-dire que $\rho^2_{\text{Schwarz}} < 1$, $\forall u, v, c$ tels que $M < 1$ et $0 < u < c$, on va utiliser le fait que pour λ , $\alpha \in \mathbb{C}$, $\mathbb{C}$ désignant l'ensemble des nombres complexes, on a :

$$\left|\frac{\lambda - \alpha}{\lambda + \alpha}\right| < 1 \Leftrightarrow \left(\frac{\lambda}{\alpha}\right) \subset \{z \in \mathbb{C} | \Re(z) > 0\}$$

Dans un premier temps, on applique le résultat ci-dessus pour $\lambda = R_1(k)$ et $\alpha = a_1$ avant de considérer le cas où $\lambda = R_1(k)(1-M_n)$ et $\alpha = 2M_nR_1(k)+a_1(1+M_n)$. On constate que si $\Re(\frac{R_1(k)}{a_1}) > 0$ alors on a :

$$\begin{cases} \dfrac{R_1(k) - a_1}{R_1(k) + a_1} < 1 \\ \dfrac{R_1(k)(1 - 3M_n) - a_1(1 + M_n)}{R_1(k) + a_1} < 1 \end{cases} \Rightarrow \rho_{\text{Schwarz}}(k)^2 < 1$$

Cette dernière assertion s'obtient aisément après quelques manipulations algébriques et en tenant compte du fait que $\Re(R_1(k)) > 0$, $\Im(R_1(k)) > 0$, $\Re(a_1) > 0$, et $\Im(a_1) > 0$

On observe aussi que, sous certaines conditions, le taux de convergence s'annule pour une valeur non nulle du paramètre de fréquence. Ainsi, lorsque $M_n < \frac{1}{3}$, $M_t = 0$ et pour k_0 tel que :

$$k_0^2 = \left(\frac{\beta}{c}\right)^2 \frac{8M_n}{(1 - M_n)(1 - 3M_n)^2}$$

on a $\rho_{\text{Schwarz}}(k_0) = 0$. Par ailleurs :

$$\begin{aligned} \rho_\infty(M_n,M_t) = \lim_{k\to\infty} \rho_{\text{Schwarz}}(k) &= \sqrt{1 - \frac{8M_n(1 - M_n^2 - M_t^2)}{(1 + M_n)^3}} \\ &= \sqrt{\left(\frac{1 - 3M_n}{1 + M_n}\right)^2 + \frac{8M_nM_t^2}{(1 + M_n)^3}} \end{aligned} \tag{1.51}$$

qui montre que, dans le cas particulier où $M_n = \frac{1}{3}$ et $M_t = 0$, cette limite s'annule. Il faut aussi remarquer que $\rho_\infty(M_n,M_t)$ est strictement inférieur à 1 pour toute valeur de $M = \sqrt{M_n^2 + M_t^2}$ telle que $0 < M < 1$ avec $M_n > 0$ puisque :

$$1 - \rho_\infty(M_n,M_t)^2 = \frac{8M_n(1 - M_n^2 - M_t^2)}{(1 + M_n)^3} > 0$$

Même lorsque $M_t \neq 0$, la valeur asymptotique du taux de convergence admet un minimum pour $M_n = \dfrac{\sqrt{1 - M_t^2}}{\sqrt{1 - M_t^2} + \sqrt{4 - M_t^2}} \in]0,1/3[$ cette valeur étant d'autant plus proche $\frac{1}{3}$ que M_t est petit. On a représenté le comportement de cette limite sur la figure 1.3 en fonction du nombre de Mach normal à l'interface pour différentes valeurs du nombre de Mach tangentiel.

Ce résultat est quelque peu surprenant et certainement inattendu. Pour le moment, on retient de ce qui précède que ce comportement de type super-convergence est obtenu pour un écoulement tel que $M < 1$ et $v = 0$, ce qui constitue une situation particulière probablement idéale. On note en passant

FIG. 1.3 – *Taux de convergence asymptotique de l'algorithme de Schwarz dans le cas deux sous-domaines*

que nous avons limité notre analyse au cas d'un écoulement subsonique qui est le cas le plus intéressant du point de vue de la propagation de l'information à l'interface entre les deux sous-domaines.

Pour faciliter l'interprétation des résultats de cette étude, on introduit le nombre d'onde adimensionné défini par $\bar{k} = \dfrac{ck}{\beta}$. L'expression du taux de convergence en fonction de cette quantité est donnée par :

$$\begin{cases} \rho^2_{\mathsf{Schwarz}}(k) = \left| \dfrac{\bar{R}(k) - \bar{a}}{(\bar{R}(k) + \bar{a})^2} \cdot \dfrac{\bar{R}(k)(1 - 3M_n) - \bar{a}(1 + M_n)}{1 + M_n} \right| \\ \bar{a} = 1 + i\bar{k}M_t \\ \bar{R}(k) = \sqrt{\bar{a}^2 + \bar{k}^2(1 - M_n^2)} \end{cases}$$

Pour répresenter graphiquement le taux de convergence il faut aussi introduire le nombre de courant (CFL) de maille :

$$\mathsf{CFL}_h = \frac{(M+1)c}{\beta h} = \frac{(M+1)c\Delta t}{h}$$

où h répresente le pas du maillage. Si l'on tient compte du fait que la longueur d'onde maximale est $k_{max} = \dfrac{\pi}{h_{min}}$, on obtient la longueur d'onde adimensionnée maximale par :

$$\bar{k}_{max} = \frac{\pi}{M+1}(\mathsf{CFL}_h)_{max}$$

L'évolution du taux de convergence en fonction du paramètre de fréquence adimensionné $\bar{k}$ (qui est proportionnel à h^{-1} c'est-à-dire au nombre de points de la discrétisation) est représentée sur les figures 1.4 à 1.7 pour différentes valeurs de M_n et M_t et pour une valeur fixée du paramètre $(\mathrm{CFL}_h)_{max} = 100$. On notera que nous n'avons considéré que des cas de figure caractérisés par $M_n^2 + M_t^2 < 1$. On remarque que la convergence de l'algorithme de Schwarz se détériore et devient moins sensible à la valeur du nombre de Mach normal lorsque le nombre de Mach tangentiel tend vers 1.

FIG. 1.4 – *Taux de convergence de l'algorithme de Schwarz dans le cas deux sous-domaines*
Haut : $M_t = 0$ - Bas : $M_t = 0.1$

FIG. 1.5 – *Taux de convergence de l'algorithme de Schwarz dans le cas deux sous-domaines*
Haut : $M_t = 0.2$ - Bas : $M_t = \dfrac{1}{3}$

FIG. 1.6 – *Taux de convergence de l'algorithme de Schwarz dans le cas deux sous-domaines*
Haut : $M_t = 0.4$ - Bas : $M_t = 0.5$

FIG. 1.7 – *Taux de convergence de l'algorithme de Schwarz dans le cas deux sous-domaines*
Haut : $M_t = 0.6$ - Bas : $M_t = 0.8$

On conclut cette section par la remarque suivante:

Remarque 5 *Pour des plus grands nombres de sous-domaines, on ne peut pas facilement évaluer le taux de convergence en utilisant l'approche adoptée ci-dessus puisque, dans ce cas, on aboutit au calcul du rayon spectral d'une matrice de dimension* $4(N-1) \times 4(N-1)$ *où* N *est le nombre de sous-domaines. Il est néanmoins possible d'obtenir des estimations sous la forme d'inégalités comme dans Nataf et Nier[NN97].*

1.2.3 Le cas tridimensionnel

Dans cette section on cherche à savoir comment est affecté le résultat (1.50) lorsque la dimension spatiale augmente et lorsque l'analyse départ d'une autre formulation du système d'équations d'Euler (ces deux points sont considérés simultanément dans ce qui suit). On étudie la convergence de l'algorithme de Schwarz (1.20) appliqué à la résolution du système d'équations d'Euler en trois dimensions d'espace exprimé en variables primitives. Comme dans la section précédente, on se limite au cas d'une décomposition en deux sous-domaines. La forme quasi-linéaire des équations est donnée par :

$$\partial_t W + A_1(W)\partial_x W + A_2(W)\partial_y W + A_3(W)\partial_z W = 0 \tag{1.52}$$

où le vecteur d'état est $W = (\rho\ ,\ u\ ,\ v\ ,\ w\ ,\ p)^T$; les matrices Jacobiennes des vecteurs flux, définies par $A_1(W) = \dfrac{\partial F_1(W)}{\partial W}$, $A_2(W) = \dfrac{\partial F_2(W)}{\partial W}$ et $A_3(W) = \dfrac{\partial F_3(W)}{\partial W}$, ont pour expressions :

$$A_1(W) = \begin{pmatrix} u & \rho & 0 & 0 & 0 \\ 0 & u & 0 & 0 & \dfrac{1}{\rho} \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ 0 & \rho\, c^2 & 0 & 0 & u \end{pmatrix} \quad A_2(W) = \begin{pmatrix} v & 0 & \rho & 0 & 0 \\ 0 & v & 0 & 0 & 0 \\ 0 & 0 & v & 0 & \dfrac{1}{\rho} \\ 0 & 0 & 0 & v & 0 \\ 0 & 0 & \rho\, c^2 & 0 & v \end{pmatrix}$$

$$A_3(W) = \begin{pmatrix} w & 0 & 0 & \rho & 0 \\ 0 & w & 0 & 0 & 0 \\ 0 & 0 & w & 0 & 0 \\ 0 & 0 & 0 & w & \dfrac{1}{\rho} \\ 0 & 0 & 0 & \rho\, c^2 & w \end{pmatrix}$$

Comme précédemment, le point de départ de l'analyse de convergence est donné par le système linéarisé :

$$\mathcal{L}W := \frac{1}{\Delta t}W + A_1\partial_x W + A_2\partial_y W + A_3\partial_z W = f \tag{1.53}$$

auquel on applique le changement de variables $\widetilde{W} = T^{-1}W$ défini par la matrice :

$$T = \begin{pmatrix} \dfrac{\rho}{c\sqrt{2}} & \dfrac{\rho}{c\sqrt{2}} & 0 & 0 & 1 \\ -\dfrac{1}{\sqrt{2}} & \dfrac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \dfrac{c\rho}{\sqrt{2}} & \dfrac{c\rho}{\sqrt{2}} & 0 & 0 & 0 \end{pmatrix}$$

qui est telle que $A_1 = T\Lambda T^{-1}$ (T est la matrice dont les colonnes sont les vecteurs propres de la matrice A_1, qui diagonalise A_1 et qui symétrise A_2 et A_3) et où :

$$\Lambda = \begin{pmatrix} u-c & 0 & 0 & 0 & 0 \\ 0 & u+c & 0 & 0 & 0 \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ 0 & 0 & 0 & 0 & u \end{pmatrix}$$

est la matrice diagonale des valeurs propres associées. Le système (1.53) devient :

$$\widetilde{\mathcal{L}}\widetilde{W} = \beta\widetilde{W} + B_1\partial_x\widetilde{W} + B_2\partial_y\widetilde{W} + B_3\partial_z\widetilde{W} = T^{-1}f \quad , \quad \beta = \frac{1}{\Delta t} \tag{1.54}$$

où $B_2 = T^{-1}A_2T$ et $B_3 = T^{-1}A_3T$:

$$B_2 = \begin{pmatrix} v & 0 & \frac{c}{\sqrt{2}} & 0 & 0 \\ 0 & v & \frac{c}{\sqrt{2}} & 0 & 0 \\ \frac{c}{\sqrt{2}} & \frac{c}{\sqrt{2}} & v & 0 & 0 \\ 0 & 0 & 0 & v & 0 \\ 0 & 0 & 0 & 0 & v \end{pmatrix} \quad B_3 = \begin{pmatrix} w & 0 & 0 & \frac{c}{\sqrt{2}} & 0 \\ 0 & w & 0 & \frac{c}{\sqrt{2}} & 0 \\ 0 & 0 & w & 0 & 0 \\ \frac{c}{\sqrt{2}} & \frac{c}{\sqrt{2}} & 0 & w & c \\ 0 & 0 & 0 & 0 & w \end{pmatrix}$$

Dans la suite on note simplement W (respectivement $\mathcal{L}$) au lieu de $\widetilde{W}$ (respectivement $\widetilde{\mathcal{L}}$). On considère une décomposition en deux sous-domaines avec une interface définie par le plan $x = 0$. La formulation de l'algorithme de Schwarz est similaire à l'équation (1.40). On procède alors à une transformée de Fourier dans les directions y et z (les variables de Fourier sont notées k_1 et k_2). Soit :

$$\hat{E}(x,k_1,k_2) = \mathcal{F}E(x,y,z) = \int_{\mathbb{R}}\int_{\mathbb{R}} e^{-i(k_1y+k_2z)}E(x,y,z)dydz$$

le symbole de Fourier du vecteur erreur où on a supposé que les matrices $B_{1,2,3}$ sont constantes, ce qui est bien le cas puisque les équations ont été linéarisées autour d'un état $\bar{W}$. Dans chaque sous-domaine l'équation différentielle ordinaire résultante est de la forme :

$$\frac{d}{dx}\hat{E}_i^{p+1}(x,k_1,k_2) = -M(k_1,k_2)\hat{E}_i^{p+1}(x,k_1,k_2)$$

où la matrice $M(k_1,k_2)$ est donnée par :

$$\begin{pmatrix} \frac{a}{u-c} & 0 & \frac{1}{\sqrt{2}}\frac{ik_1c}{u-c} & \frac{1}{\sqrt{2}}\frac{ik_2c}{u-c} & 0 \\ 0 & \frac{a}{u+c} & \frac{1}{\sqrt{2}}\frac{ik_1c}{u+c} & \frac{1}{\sqrt{2}}\frac{ik_2c}{u+c} & 0 \\ \frac{1}{\sqrt{2}}\frac{ik_1c}{u} & \frac{1}{\sqrt{2}}\frac{ik_1c}{u} & \frac{a}{u} & 0 & 0 \\ \frac{1}{\sqrt{2}}\frac{ik_2c}{u} & \frac{1}{\sqrt{2}}\frac{ik_2c}{u} & 0 & \frac{a}{u} & 0 \\ 0 & 0 & 0 & 0 & \frac{a}{u} \end{pmatrix}$$

avec $a = \beta + ik_1 v + ik_2 w$. La solution dans chaque sous-domaine est exprimée comme une combinaison linéaire des vecteurs propres de $M(k_1,k_2)$. Comme dans le cas bidimensionnel, on impose que les solutions locales soient bornées à l'infini ce qui conduit à une prise en compte, dans leur expression, des vecteurs qui correspondent à des valeurs propres dont la partie réelle a le bon signe. De même, pour des raisons similaires à celles avancées dans le cas bidimensionnel, on se limite au seul cas d'étude qui présente un intérêt pour la convergence. On suppose donc que l'écoulement est subsonique c'est-à-dire que $M < 1$ et que $0 < u < c$. Tous calculs faits, on obtient les expressions suivantes pour les valeurs propres de la matrice $M(k_1,k_2)$:

$$\begin{cases} \lambda_1(k_1,k_2) & = & \dfrac{-au - cR(k_1,k_2)}{c^2 - u^2} \\ \lambda_2(k_1,k_2) & = & \dfrac{-au + cR(k_1,k_2)}{c^2 - u^2} \\ \lambda_{3,4,5}(k_1,k_2) & = & \dfrac{a}{u} \end{cases}$$

les vecteurs propres associés étant donnés par :

$$V_1(k_1,k_2) = \begin{pmatrix} -\dfrac{(R(k_1,k_2)+a)(c+u)}{\sqrt{2}} \\ \dfrac{(R(k_1,k_2)-a)(c-u)}{\sqrt{2}} \\ ik_1(c^2-u^2) \\ ik_2(c^2-u^2) \\ 0 \end{pmatrix} \quad V_2(k_1,k_2) = \begin{pmatrix} \dfrac{(R(k_1,k_2)-a)(c+u)}{\sqrt{2}} \\ -\dfrac{(R(k_1,k_2)+a)(c-u)}{\sqrt{2}} \\ ik_1(c^2-u^2) \\ ik_2(c^2-u^2) \\ 0 \end{pmatrix}$$

$$V_3(k_1,k_2) = \begin{pmatrix} -\dfrac{ik_1 u}{\sqrt{2}} \\ \dfrac{ik_1 u}{\sqrt{2}} \\ a \\ 0 \\ 0 \end{pmatrix} \quad V_4(k_1,k_2) = \begin{pmatrix} 0 \\ 0 \\ -k2 \\ k1 \\ 0 \end{pmatrix} \quad V_5(k_1,k_2) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

où $R(k_1,k_2) = \sqrt{a^2 + (k_1^2 + k_2^2)(c^2 - u^2))}$. Ces vecteurs propres sont linéairement indépendants sauf dans le cas particulier $v = w = 0$, pour $k_1^2 + k_2^2 = \dfrac{c^2}{u^2}$, qu'on éliminera dans la suite de notre étude.

En suivant un raisonnement similaire à celui du cas bidimensionnel, sous l'hypothèse $0 < u < c$ de départ, on a que $\Re(\lambda_1) < 0$ et $\Re(\lambda_{2,3,4,5}) > 0$ ce qui conduit à l'expression suivante de l'itération de Schwarz à l'interface. Pour le premier sous-domaine :

$$\begin{aligned} \alpha_1^{p+1}(V^-(k_1,k_2))_1 &= \alpha_2^{(p)}(V_1^+(k_1,k_2))_1 + \alpha_3^{(p)}(V_2^+(k_1,k_2))_1 \\ &+ \alpha_4^{(p)}(V_3^+(k_1,k_2))_1 + \alpha_5^{(p)}(V_4^+(k_1,k_2))_1 \end{aligned}$$

et pour le deuxième sous-domaine :

$$\left\{\begin{array}{lcl}
\alpha_2^{p+1}(V_1^+(k_1,k_2))_2 & + & \alpha_3^{p+1}(V_2^+(k_1,k_2))_2 + \alpha_4^{p+1}(V_3^+(k_1,k_2))_2 + \alpha_5^{p+1}(V_4^+(k_1,k_2))_2 \\
 & = & \alpha_1^{(p)}(V^-(k_1,k_2))_2 \\
\alpha_2^{p+1}(V_1^+(k_1,k_2))_3 & + & \alpha_3^{p+1}(V_2^+(k_1,k_2))_3 + \alpha_4^{p+1}(V_3^+(k_1,k_2))_3 + \alpha_5^{p+1}(V_4^+(k_1,k_2))_3 \\
 & = & \alpha_1^{(p)}(V^-(k_1,k_2))_3 \\
\alpha_3^{p+1}(V_1^+(k_1,k_2))_4 & + & \alpha_3^{p+1}(V_2^+(k_1,k_2))_4 + \alpha_4^{p+1}(V_3^+(k_1,k_2))_4 + \alpha_5^{p+1}(V_4^+(k_1,k_2))_4 \\
 & = & \alpha_1^{(p)}(V^-(k_1,k_2))_4 \\
\alpha_3^{p+1}(V_1^+(k_1,k_2))_5 & + & \alpha_3^{p+1}(V_2^+(k_1,k_2))_5 + \alpha_4^{p+1}(V_3^+(k_1,k_2))_5 + \alpha_5^{p+1}(V_4^+(k_1,k_2))_5 \\
 & = & \alpha_1^{(p)}(V^-(k_1,k_2))_5
\end{array}\right.$$

où les indices des vecteurs désignent les composantes de ces derniers. En remplaçant les expressions des vecteurs propres dans les relations ci-dessus et en résolvant les équations résultantes pour obtenir les α_i il vient :

$$\alpha_1^{p+1} = -\left[\frac{R(k_1,k_2)-a}{R(k_1,k_2)-a}\right]\cdot\left[\frac{c-u}{c+u}\right]\alpha_2^{(p)} - \left[\frac{ik_1\sqrt{2}(c-u)}{R(k_1,k_2)+a}\right]\alpha_3^{(p)} - \left[\frac{ik_2\sqrt{2}(c-u)}{R(k_1,k_2)+a}\right]\alpha_4^{(p)}$$

et :

$$\left\{\begin{array}{lcl}
\alpha_2^{p+1} & = & -\left[\dfrac{R(k_1,k_2)-a}{R(k_1,k_2)+a}\right]\alpha_1^{(p)} \\
\alpha_3^{p+1} & = & -\left[\dfrac{ik_1\sqrt{2}u}{R(k_1,k_2)+a}\right]\alpha_1^{(p)} \\
\alpha_4^{p+1} & = & -\left[\dfrac{ik_2\sqrt{2}u}{R(k_1,k_2)+a}\right]\alpha_1^{(p)} \\
\alpha_5^{p+1} & = & 0
\end{array}\right.$$

Finalement, le taux de convergence de l'algorithme de Schwarz est donné par :

$$\rho^2_{\mathsf{Schwarz}}(k_1,k_2) = \left|\frac{R(k_1,k_2)-a}{(R(k_1,k_2)+a)^2}\cdot\frac{R(k_1,k_2)(c-3u)-a(c+u)}{c+u}\right|$$

On voit donc que le taux de convergence obtenu dans le cas tridimensionnel est quasiment identique à celui caractérisant la convergence dans le cas bidimensionnel à ceci près qu'il dépend ici de deux paramètres de fréquence. En particulier, on retrouve le même comportement asymptotique du taux de convergence lorsque k_1 et k_2 tendent vers l'infini, et pour des nombres de Mach tangentiels au plan $x=0$ nuls :

$$\begin{aligned}
\rho_\infty(M_n,M_{t1},M_{t2}) = \lim_{k_1\to\infty,k_2\to\infty}\rho_{\mathsf{Schwarz}}(k_1,k_2) &= \sqrt{1-\frac{8M_n(1-M_n)(1-M_n^2-M_{t1}^2-M_{t2}^2)}{(1+M_n)^3}} \\
&= \sqrt{\left(\frac{1-3M_n}{1+M_n}\right)^2+\frac{8M_n(M_{t1}^2+M_{t2}^2)}{(1+M_n)^3}}
\end{aligned} \tag{1.55}$$

où $M_n = \frac{u}{c}$ désigne le nombre Mach normal à l'interface; $M_{t1} = \frac{v}{c}$ et $M_{t2} = \frac{w}{c}$ correspondent, quant à eux, aux nombres de Mach suivant le plan tangentiel à l'interface.

1.3 Conclusion

L'analyse de convergence préliminaire réalisée dans cette section a montré que l'algorithme de type Schwarz additif proposé dans la section 1.1.3 pour la résolution du système d'équations d'Euler, est convergent même si les conditions de raccord imposées aux interfaces entre les sous-domaines restent des plus simples; ces conditions sont ici qualifiées de *classiques* par comparaison aux conditions plus sophistiquées, dites *optimisées* étudiées par Japhet *et al.* pour une équation d'advection-diffusion, et récemment adaptées au système d'équations d'Euler par Gonzalez[Gon00]. L'analyse de convergence quantitative, plus approfondie, a été limitée à une décomposition en deux sous-domaines du domaine de calcul et à des équations linéarisées autour d'un état constant. L'étude des cas à deux et trois dimensions d'espace a notamment montré que le résultat obtenu pour le taux de convergence de la méthode est indépendant de la dimension spatiale. Le résultat le plus notable est certainement que le taux de convergence reste asymptotiquement strictement inférieur à 1. Ce point motive l'étude présentée au chapitre 4 où l'on tente notamment d'expliquer le bon comportement de la méthode en dépit de l'utilisation de conditions d'interfaces « classiques ». Le résultat le plus surprenant concerne le comportement de super-convergence prédit théoriquement pour une valeur du Mach normal égale à $M_n = \frac{1}{3}$ quelque soit la dimension spatiale.

Nous verrons dans le chapitre 2 qui suit que des expériences numériques ont permis de retrouver un comportement qualitativement en accord avec la théorie, à savoir, qu'il existe une valeur du nombre de Mach de référence pour laquelle la convergence est optimale et cette valeur est indépendante du pas d'espace (c'est-à-dire du nombre d'inconnues). L'analyse de convergence menée se doit d'être complétée par l'étude d'une décomposition à plus de 2 sous-domaines et notamment du cas à N sous-domaines afin d'évaluer les propriétés de scalabilité de la méthode. Néanmoins, cet aspect de la méthode est étudié numériquement dans les sections 2.6.1 et 2.6.2 du chapitre 2.

Chapitre 2

Résolution numérique par décomposition de domaine du système d'équations d'Euler

2.1 Méthodes de discrétisation en espace et en temps

Dans cette section, on décrit les caractéristiques de la méthodologie numérique utilisée pour la résolution du système d'équations d'Euler. Une attention plus particulière est portée aux aspects qui relèvent d'un traitement particulier lors de la mise en œuvre de la méthode de décomposition de domaine introduite dans la section 1.1 et notamment, la formulation volumes finis utilisant des flux convectifs décentrés et le schéma d'Euler implicite linéarisé pour l'intégration en temps.

2.1.1 Modèle mathématique

Soit $\Omega \subset \mathbb{R}^2$ le domaine de calcul et Γ sa frontière. Γ se décompose en une frontière de type paroi solide Γ_w et une frontière à l'infini Γ_∞ : $\Gamma = \Gamma_w \cup \Gamma_\infty$. Soit $\vec{n}$ la normale sortante en tout point de Γ. La forme conservative du système d'équations d'Euler est donnée par :

$$\frac{\partial W}{\partial t} + \vec{\nabla}.\vec{\mathcal{F}}(W) = 0 \quad , \quad W = \left(\rho \ , \ \rho\vec{U} \ , \ E\right)^T \quad , \quad \vec{\nabla} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)^T \tag{2.1}$$

où $W = W(\vec{x},t)$; $\vec{x}$ et t désignent respectivement les variables spatiale et temporelle tandis que $\vec{\mathcal{F}}(W) = (F_x(W) \ , \ F_y(W))^T$ est le flux convectif dont les composantes sont données par :

$$F_x(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(E+p) \end{pmatrix} \quad , \quad F_y(W) = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}$$

Dans les expressions ci-dessus, ρ désigne la masse volumique (ou densité), $\vec{U} = (u \ , \ v)^T$ le vecteur vitesse, E l'énergie totale par unité de volume et p la pression. La pression est déduite des autres variables par la loi d'état des gaz parfaits :

$$p = (\gamma - 1)(E - \frac{1}{2}\rho \parallel \vec{U} \parallel^2)$$

où γ est le rapport des chaleurs spécifiques ($\gamma = 1.4$ pour un gaz diatomique).

2.1.2 Méthode de discrétisation en espace

Soit le domaine de calcul $\Omega \subset \mathbb{R}^2$, borné par la frontière Γ qui contient une partie de type solide et une autre artificielle qui limite la partie infinie du domaine physique. Celui ci, qui est est supposé polygonal est discrétisé par une triangulation $\mathcal{T}_h$. Un sommet de $\mathcal{T}_h$ est noté s_i et l'ensemble des sommets voisins de s_i par $N(s_i)$. On associe à chaque sommet s_i une surface de contrôle (ou cellule médiane) notée C_i qui est construite comme l'union de contributions locales provenant de l'ensemble des triangles dont s_i et un sommet. La contribution d'un triangle donné est obtenue en joignant son centre de gravité G aux millieux I des segments incidents à s_i (voir la figure 2.1). La frontière de C_i est notée ∂C_i et le vecteur normal sortant en tout point de ∂C_i par $\vec{\nu}_i = (\nu_{ix}, \nu_{iy})$. L'union de toutes ces cellules constitue une seconde discrétisation de Ω souvent qualifiée de discrétisation duale de $\mathcal{T}_h$:

$$\Omega_h = \bigcup_{i=1}^{N_V} C_i \quad , \quad N_V : \text{nombre de sommets de } \mathcal{T}_h$$

FIG. 2.1 – *La cellule médiane sur un maillage triangulaire*

La méthode d'approximation en espace adoptée ici est basée sur les éléments suivants :

- une formulation volumes finis combinée à des schémas décentrés pour le calcul des flux convectifs;
- une technique MUSCL (« Monotonic Upwind Schemes for Conservation Laws ») initialement introduite par van Leer[VL79] et étendue au cas de maillages triangulaires non-structurés par Fezoui[Fez85].

On cherche en fait à approcher la solution du système des équations d'Euler par une méthode de volumes finis. Un schéma de volumes finis bidimensionnels est basé sur la forme intégrale des équations d'Euler qui donne en utilisant ensuite la formule de Gauss-Ostrodgadski, pour chaque cellule C_i:

$$\begin{aligned} \iint_{C_i} \frac{\partial W}{\partial t} d\vec{x} \quad &+ \sum_{j \in N(s_i)} \int_{\partial C_{ij}} \vec{\mathcal{F}}(W).\vec{\nu}_i d\sigma & < 1 > \\ &+ \int_{\partial C_i \cap \Gamma_w} \vec{\mathcal{F}}(W).\vec{n}_i d\sigma + \int_{\partial C_i \cap \Gamma_\infty} \vec{\mathcal{F}}(W).\vec{n}_i d\sigma & < 2 > \\ &= 0 \end{aligned} \tag{2.2}$$

où $\partial C_{ij} = \partial C_i \cap \partial C_j$ et où on a noté par Γ_b et Γ_∞, les parties solide et "infinie" de la frontière du domaine de calcul. Une approximation volumes finis du terme $< 1 >$ *conservative* et *consistante*, précise au premier ordre, s'écrit simplement comme :

$$< 1 > = \; W_i^{n+1} - W_i^n + \Delta t \sum_{j \in N(s_i)} \Phi_{\mathcal{F}}(W_i^n, W_j^n, \vec{\nu}_{ij}) \tag{2.3}$$

où $\Phi_{\mathcal{F}}$ désigne une fonction de flux numérique qui vérifie les propriétés :

$$\begin{cases} \Phi_{\mathcal{F}}(W_i, W_j, \vec{\nu}_{ij}) & = & -\Phi_{\mathcal{F}}(W_i, W_j, -\vec{\nu}_{ij}) \\ \Phi_{\mathcal{F}}(W_i, W_i, \vec{\nu}_{ij}) & = & \mathcal{F}(W_i)\vec{\nu}_{ij} \end{cases}$$

pour tout W_i et W_j et tout vecteur $\vec{\nu}_{ij}$. Le flux numérique est une approximation du flux à travers la frontière de chaque cellule:

$$\Phi_{ij} \equiv \Phi_{\mathcal{F}}(W_i, W_j, \vec{\nu}_{ij}) \approx \int_{\partial C_{ij}} \vec{\mathcal{F}}(W).\vec{\nu}_i d\sigma \tag{2.4}$$

On note :

$$\vec{\nu}_{ij} = \int_{\partial C_{ij}} \vec{\nu} d\sigma = \vec{\nu}_1 + \vec{\nu}_2 \tag{2.5}$$

Les normales $\vec{\nu}_1$ et $\vec{\nu}_2$ sont représentés sur la figure 2.2.

FIG. 2.2 – *Interface ∂C_{ij} séparant les sommets s_i et s_j*

Dans cette étude, le calcul numérique des flux convectifs repose sur l'utilisation de schémas décentrés particulièrement bien adaptés à la nature hyperbolique du système d'équations d'Euler. La construction des schémas numériques correspondants est détaillée dans de nombreux ouvrages comme celui de Toro[Tor97]. On introduit les notations suivantes :

$$\vec{\mathcal{F}}(W) = \begin{pmatrix} F(W) \\ G(W) \end{pmatrix} \quad \text{et} \quad \mathcal{F}(W, \vec{\nu}) = \vec{\mathcal{F}}(W) \cdot \vec{\nu} = \nu_x F(W) + \nu_y G(W)$$

ainsi que la matrice Jacobienne $\mathcal{A}$ définie par :

$$\mathcal{A}(W, \vec{\nu}) = \nu_x \frac{\partial F(W)}{\partial W} + \nu_y \frac{\partial G(W)}{\partial W}$$

On sait que $\mathcal{A}$ est diagonalisable et a pour valeurs propres :

$$\begin{cases} \lambda_1(W,\vec{\nu}) &= \lambda_2(W,\vec{\nu}) = \vec{U}\cdot\vec{\nu} \\ \lambda_3(W,\vec{\nu}) &= \lambda_1(W,\vec{\nu}) + c\|\vec{\nu}\| \\ \lambda_4(W,\vec{\nu}) &= \lambda_1(W,\vec{\nu}) - c\|\vec{\nu}\| \end{cases}$$

On rappelle que $c = \sqrt{\dfrac{\gamma p}{\rho}}$ est la vitesse du son. La diagonalisation de $\mathcal{A}$ s'écrit :

$$\mathcal{A}(W,\vec{\nu}) = \mathcal{T}(W,\vec{\nu})\Lambda(W,\vec{\nu})\mathcal{T}^{-1}(W,\vec{\nu})$$

Le calcul du flux numérique Φ_{ij} à l'interface entre deux cellules peut être évalué par une méthode de Godunov ce qui veut dire qu'on considère un problème de Riemann monodimensionnel (dans la direction de la normale $\vec{\nu}_{ij}$) local à l'interface, à partir des deux états constants W_i et W_j respectivement à gauche et à droite de ∂C_{ij} :

$$\begin{cases} \dfrac{\partial W}{\partial t} + \dfrac{\partial}{\partial \nu}\left[\vec{\mathcal{F}}(W)\cdot\vec{\nu}\right] = 0 \\ W(\vec{X},t) = \begin{cases} W_i & \text{si} \quad \vec{X}\in C_i \\ W_j & \text{si} \quad \vec{X}\in C_j \end{cases} \end{cases} \tag{2.6}$$

La méthode de calcul du flux numérique de Roe[Roe81] consiste à approcher la solution de ce problème de Riemann. Le terme $\dfrac{\partial}{\partial \nu}\left[\vec{\mathcal{F}}(W)\cdot\vec{\nu}\right]$ dans l'équation (2.6) est linéarisé et remplacé par $\mathcal{A}_{ij}\dfrac{\partial W}{\partial \nu}$. La matrice Jacobienne de Roe est construite de façon à satisfaire les trois propriétés suivantes :

1. respecter l'hyperbolicité du système, c'est-à-dire que la diagonalisation conduit à des valeurs propres réelles et des vecteurs propres linéairement indépendants ;
2. être consistant avec la matrice jacobienne $\mathcal{A}$:

$$\mathcal{A}_{ij}(W_i,W_i) = \mathcal{A}(W_i)$$

3. assurer un principe de conservation à travers des discontinuités :

$$\left(\vec{\mathcal{F}}(W_i) - \vec{\mathcal{F}}(W_j)\right)\cdot\vec{\nu}_{ij} = \mathcal{A}_{ij}(W_i - W_j)$$

La matrice Jacobienne de Roe est évaluée comme :

$$\mathcal{A}_{ij} \equiv \mathcal{A}(\widetilde{W}_{ij},\vec{\nu}_{ij}) \tag{2.7}$$

où $\widetilde{W}_{ij}$ est la moyenne de Roe des états W_i et W_j. On note:

$$W_i = \begin{pmatrix} \rho_i \\ \rho_i u_i \\ \rho_i v_i \\ E_i \end{pmatrix} \quad , \quad W_j = \begin{pmatrix} \rho_j \\ \rho_j u_j \\ \rho_j v_j \\ E_j \end{pmatrix} \quad , \quad W_{ij} = \begin{pmatrix} \widetilde{\rho}_{ij} \\ \widetilde{\rho}_{ij}\widetilde{u}_{ij} \\ \widetilde{\rho}_{ij}\widetilde{v}_{ij} \\ \widetilde{E}_{ij} \end{pmatrix}$$

et :

$$\widetilde{\rho}_{ij} = \frac{\sqrt{\rho_i}\rho_i + \sqrt{\rho_j}\rho_j}{\sqrt{\rho_i} + \sqrt{\rho_j}} \quad , \quad \widetilde{u}_{ij} = \frac{\sqrt{\rho_i}u_i + \sqrt{\rho_j}u_j}{\sqrt{\rho_i} + \sqrt{\rho_j}} \quad , \quad \widetilde{v}_{ij} = \frac{\sqrt{\rho_i}v_i + \sqrt{\rho_j}v_j}{\sqrt{\rho_i} + \sqrt{\rho_j}} \quad , \quad \widetilde{H}_{ij} = \frac{\sqrt{\rho_i}H_i + \sqrt{\rho_j}H_j}{\sqrt{\rho_i} + \sqrt{\rho_j}}$$

où H est l'enthalpie totale par unité de volume définie par :

$$H = \frac{E+p}{\rho} = \frac{\gamma p}{(\gamma-1)\rho} + \frac{u^2+v^2}{2}$$

La fonction de flux numérique Φ_{ij} s'écrit alors :

$$\Phi_{ij} = \Phi_{\mathcal{F}}(W_i,W_j,\vec{\nu}_{ij}) = \frac{\vec{\mathcal{F}}(W_i) + \vec{\mathcal{F}}(W_j)}{2}.\vec{\nu}_{ij} - d(W_i,W_j,\vec{\nu}_{ij}) \tag{2.8}$$

où $d(W_i,W_j,\vec{\nu}_{ij})$ représente un terme de diffusion numérique :

$$d(W_i,W_j,\vec{\nu}_{ij}) = |\mathcal{A}(\widetilde{W}_{ij},\vec{\nu}_{ij})|\frac{W_i - W_j}{2}$$

En d'autres termes, la fonction de flux numérique (2.8) définit un schéma décentré pour le calcul du flux convectif (2.4). De plus, la troisième propriété de la matrice de Roe permet de simplifier le calcul du flux. En effet, sachant que :

$$\begin{cases} \mathcal{A}^+(W,\vec{\nu}) & = & \mathcal{T}(W,\vec{\nu})\Lambda^+(W,\vec{\nu})\mathcal{T}^{-1}(W,\vec{\nu}) \\ \Lambda^+(W,\vec{\nu}) & = & \text{diag}(\lambda_k^+(W,\vec{\nu})) \\ \lambda_k^+(W,\vec{\nu}) & = & \max(\lambda_k(W,\vec{\nu}),0) \end{cases}$$

$$\begin{cases} \mathcal{A}^-(W,\vec{\nu}) & = & \mathcal{T}(W,\vec{\nu})\Lambda^-(W,\vec{\nu})\mathcal{T}^{-1}(W,\vec{\nu}) \\ \Lambda^-(W,\vec{\nu}) & = & \text{diag}(\lambda_k^-(W,\vec{\nu})) \\ \lambda_k^-(W,\vec{\nu}) & = & \min(\lambda_k(W,\vec{\nu}),0) \end{cases}$$

la fonction de flux numérique s'écrit :

$$\Phi_{\mathcal{F}}(W_i,W_j,\vec{\nu}_{ij}) = \vec{\mathcal{F}}(W_j).\vec{\nu}_{ij} - \mathcal{A}^+(\widetilde{W}_{ij},\vec{\nu})(W_j - W_i) \tag{2.9}$$

ou bien :

$$\Phi_{\mathcal{F}}(W_i,W_j,\vec{\nu}_{ij}) = \vec{\mathcal{F}}(W_i).\vec{\nu}_{ij} + \mathcal{A}^-(\widetilde{W}_{ij},\vec{\nu})(W_j - W_i) \tag{2.10}$$

Le schéma est alors plus simple et surtout moins coûteux à implémenter.

La technique MUSCL[VL79] pour l'extension à une approximation plus précise de (2.8) repose sur une méthode d'interpolation linéaire des états W_{ij} et W_{ji} à l'interface entre les cellules C_i et C_j :

$$\bar{W}_{ij} = \bar{W}_i + \frac{1}{2}(\vec{\nabla}\bar{W})_i.s_i\vec{s}_j \quad , \quad \bar{W}_{ji} = \bar{W}_j - \frac{1}{2}(\vec{\nabla}\bar{W})_j.s_i\vec{s}_j \tag{2.11}$$

où $\bar{W} = \left(\rho \ , \ \vec{U} \ , \ p\right)^T$ — en d'autres termes, l'interpolation est réalisée sur les variables physiques et non pas sur les variables conservatives. Les états interpolés (2.11) sont alors utilisés comme arguments de la fonction de flux numérique (2.3). Les gradients nodaux $(\vec{\nabla}\bar{W})_i$ sont obtenus par une moyenne pondérée des gradients P1 (centrés) calculés sur chaque triangle du support élément fini du sommet s_i :

$$\left(\vec{\nabla}\bar{W}\right)_i = \frac{\displaystyle\iint_{C_i} \vec{\nabla}\bar{W}|_T d\vec{x}}{\displaystyle\iint_{C_i} d\vec{x}} = \frac{1}{\text{aire}(C_i)} \sum_{T\epsilon C_i} \frac{\text{aire}(T)}{3} \sum_{s_{i_k}\in T} \bar{W}_{i_k} \vec{\nabla} N_{i_k}^T \tag{2.12}$$

où $N_{i_k}^T(x,y,z)$ est la fonction de base P1 définie au sommet s_{i_k} et associée au triangle T. La construction donnée par les équations (2.11) et (2.12) résulte en une approximation demi-décentrée qui peut cependant conduire à la présence d'oscillations non-physiques dans la solution exprimant une perte de monotonie. Une solution classique à ce problème consiste à réaliser un compromis entre des schémas précis au premier ordre et au second ordre avec l'introduction d'une procédure de limitation dans le calcul des états interpolés (2.11) (voir par exemple [FD89] pour plus de détails).

2.1.3 Conditions aux limites

Le terme $< 2 >$ de l'équation (2.2) est associé aux conditions aux limites du problème. Les simulations numériques considérées dans les sections 2.6.1 et 2.6.2 mettent en jeu les conditions aux limites suivantes qui sont ici prises en compte dans la formulation faible du problème :

- *paroi solide*. On impose sur Γ_w une condition de glissement $\vec{U}.\vec{n} = 0$. Cette condition est introduite dans le terme correspondant de l'équation (2.2); plus précisément, on prend en compte l'intégrale de pression :

$$\int_{\partial C_i \cap \Gamma_w} \vec{\mathcal{F}}(W).\vec{n}_i d\sigma = p_i \parallel \vec{n} \parallel \begin{pmatrix} 0 \\ \bar{n}_{ix} \\ \bar{n}_{iy} \\ 0 \end{pmatrix} \tag{2.13}$$

  dans l'assemblage des flux élémentaires pour l'équation discrète associée au sommet S_i.

- *frontière infinie*. Sur Γ_∞, on suppose que l'écoulement est uniforme et caractérisé par le vecteur d'état :

$$\rho_\infty = 1 \ \ , \ \ \vec{U}_\infty = (u_\infty \ , \ v_\infty)^T \quad \text{avec} \quad \parallel \vec{U}_\infty \parallel = 1 \ \ , \ \ p_\infty = \frac{1}{\gamma M_\infty^2} \tag{2.14}$$

  où M_∞ est le nombre de Mach de l'écoulement uniforme. Dans ce cas, une décomposition de flux amont/aval est utilisée entre l'information à l'infini (W_∞) et le vecteur d'état W_i associé au sommet $s_i \in \Gamma_\infty$. En pratique, l'intégrale correspondante dans le terme $< 2 >$ est évaluée avec une version non-reflexive de la décomposition de flux de Steger et Warming[SW81] :

$$\int_{\partial C_i \cap \Gamma_\infty} \vec{\mathcal{F}}(W).\vec{n}_i d\sigma = \mathcal{A}^+(W_i,\vec{n}_{i\infty}).W_i + \mathcal{A}^-(W_i,\vec{n}_{i\infty}).W_\infty \tag{2.15}$$

2.1.4 Intégration en temps

Puisque $W(\vec{x},t)$ est constant sur chaque cellule C_i dans la formulation volumes finis adoptée ici, on obtient le système d'équations différentielles ordinaires suivant :

$$\text{aire}(C_i)\frac{dW_i}{dt} + \Psi(W)_i = 0 \quad , \quad i = 1, \cdots, N_V \tag{2.16}$$

où $W_i = W(\vec{x}_i,t)$et :

$$\Psi(W)_i = \sum_{j \in N(s_i)} \Phi_{\mathcal{F}}(W_{ij},W_{ji},\vec{\nu}_{ij}) + \int_{\partial C_i \cap \Gamma} \vec{\mathcal{F}}(W).\vec{n}_i d\sigma \tag{2.17}$$

Les schémas explicites pour l'intégration de (2.16) sont généralement caractérisés par une condition de stabilité exprimée par un nombre de CFL (« Courant-Friedrichs-Lewy »). Une stratégie d'intégration en temps efficace est ici obtenue au moyen d'une formulation implicite linéarisée telle que celle décrite par Fezoui et Stoufflet[FS89] et brièvement rappelée dans ce qui suit. Tout d'abord, une variante implicite de l'équation (2.16) s'écrit :

$$\frac{\text{aire}(C_i)}{\Delta t^n}\delta W_i^{n+1} + \Psi(W^{n+1})_i = 0 \quad , \quad i = 1, \cdots, N_V \tag{2.18}$$

où $\delta W_i^{n+1} = W_i^{n+1} - W_i^n$. Ensuite, l'application d'une linéarisation au premier ordre du terme de flux nodal $\Psi(W^{n+1})_i$ conduit à la formulation :

$$\left(\frac{\text{aire}(C_i)}{\Delta t^n} + \frac{\partial \Psi(W^n)}{\partial W}\right)\delta W^{n+1} = -\Psi(W^n) \tag{2.19}$$

En pratique, on remplace le calcul exact de la matrice Jacobienne du flux du second ordre $\dfrac{\partial \Psi(W^n)}{\partial W}$ par une matrice Jacobienne approchée $J(W^n)$ qui résulte d'une différentiation analytique du flux du premier ordre (2.8) :

$$P(W^n)\delta W^{n+1} = \left(\frac{\text{aire}(C_i)}{\Delta t^n} + J(W^n)\right)\delta W^{n+1} = -\Psi(W^n) \tag{2.20}$$

Le calcul de la dérivée $\dfrac{\partial \Psi(W^n)}{\partial W}$ est détaillée dans la section 5.1.4 dans la cadre de la résolution numérique du système d'équations de Navier-Stokes. De cette simplification il résulte un schéma d'intégration en temps de type Euler implicite linéarisé qui est en fait une méthode de Newton modifiée (voir Fezoui et Stoufflet[FS89] pour plus de détails). Une conséquence importante est que cette formulation n'est en général pas caractérisée par une convergence quadratique lorsque le pas de temps tend vers l'infini. Cependant, la matrice $P(W^n)$ est creuse (à largeur de bande minimale) et possède les propriétés appropriées (diagonale dominance dans le cas scalaire linéaire) qui autorisent l'utilisation de méthodes de relaxation (Jacobi ou Gauss-Seidel) pour la résolution approchée du système linéaire défini par l'équation (2.19). Par ailleurs, une méthode efficace pour obtenir une solution précise au second ordre dans le cas d'un calcul d'écoulement stationnaire, tout en préservant les propriétés de la matrice Jacobienne basée sur le flux du premier ordre, consiste à utiliser les flux élémentaires précis au second ordre (voir les équations (2.11) et (2.12)) pour l'assemblage du membre de droite de l'équation (2.19).

Dans ce qui suit, on adapte l'algorithme de résolution par sous-domaine proposé dans la section 1.1 à la résolution du système d'équations d'Euler dans le contexte de la méthodologie numérique décrite

précédemment. Tout d'abord, on décrit la stratégie adoptée pour la parallélisation SPMD (« Single Program Multiple Data ») des méthodes décrites aux sections 2.1.2 et 2.1.4. Ensuite, on construit un système interface posé sur des variables d'interface qui sont ici directement reliées aux flux numériques décentrés (2.8).

2.2 Stratégie de parallélisation

La stratégie de parallélisation de la méthodologie de résolution globale définie dans les sections 2.1.2 et 2.1.4 combine un partitionnement du domaine de calcul et une programmation dans un modèle par transfert de messages. Cette stratégie est décrite en détails dans [FL94] (voir aussi [Lan96] pour la résolution des systèmes d'équations d'Euler et de Navier-Stokes tridimensionnelles). Le maillage de calcul est décomposé en plusieurs sous-maillages (ou sous-domaines) à l'aide d'algorithmes de partitionnement appropriés[FSL95]. Un code essentiellement identique au code séquentiel est exécuté sur chaque processeur (chaque processeur est en charge d'un sous-maillage). La mise en œuvre de cette stratégie implique alors la programmation explicite d'appels à des fonctions de transfert de messages pour deux types d'opérations :

- des communications *locales* aux interfaces entre sous-domaines (entre sous-maillages voisins). De telles opérations interviennent dans la boucle principale (boucle en temps) pour l'échange des variables d'état W_i et des gradients nodaux $\left(\vec{\nabla} W\right)_i$, et dans la boucle de résolution du système linéaire (2.20) pour l'échange des variations δW_i; ici, la méthode de Jacobi est sélectionnée pour cette résolution car elle est naturellement parallélisable au contraire de la méthode de Gauss-Seidel dont la parallélisation est délicate voire impossible dans le cas de calculs en maillages non-structurés;
- des communications *globales* correspondant à des opérations de réduction pour le calcul de quantités scalaires telles que les résidus linéaires et non-linéaires ou le minimum des pas de temps locaux.

Pour ce qui concerne le partitionnement du maillage de calcul, deux options peuvent être considérées. La première est basée sur l'introduction d'une zone de recouvrement aux interfaces entre sous-domaines et est particulièrement bien adaptée à la parallélisation de la formulation mixte éléments finis/volumes finis décrite dans la section 2.1.2. Son principal inconvénient est d'introduire des opérations arithmétiques redondantes dues à la duplication de sommets, triangles et segments entre deux sous-domaines voisins. Une faible proportion de ces opérations peut être éliminée par renumérotation locale des listes d'entités géométriques. La seconde stratégie consiste à limiter l'interface à une liste de segments (seuls les segments et les sommets associés sont dupliqués dans deux sous-domaines voisins). Dans ce cas, il n'y a pas d'opération arithmétique redondante. La première stratégie est caractérisée par l'échange aux interfaces de quantités physiques assemblées (W_i , $\left(\vec{\nabla} W\right)_i$ et δW_i) alors que dans la seconde option, des quantités semi-assemblées sont échangées ($\Psi(W)_i$, $\left(\vec{\nabla} W\right)_i$ et δW_i). En pratique, la programmation parallèle est plus simple lorsque l'on opte pour la première stratégie alors que l'efficacité parallèle est meilleure lorsque la parallélisation est basée sur la seconde approche (voir notamment [Lan96] pour une évaluation détaillée des deux stratégies dans le cadre de la résolution des systèmes d'équations d'Euler et de Navier-Stokes tridimensionnelles).

Dans cette étude, sur la base de la méthode de décomposition de domaine proposée dans la section 1.1, il est intéressant d'opter pour un partitionnement avec recouvrement du domaine de calcul. En effet, on vérifie aisément que dans ce cas, l'interface entre deux sous-domaines voisins est non-recouvrante du

point de vue de la discrétisation duale de Ω en termes des cellules médianes : si Ω_1 et Ω_2 sont voisins alors (voir la figure 2.3) :

$$\Gamma = \Omega_1 \cap \Omega_2 = \bigcup_{C_{1_k} \in \Omega_1, C_{2_k} \in \Omega_2} \partial C_{1_k} \cap \partial C_{2_k}$$

2.3 Algorithme de résolution par sous-domaine en discret

Pour construire un système interface de la forme (1.24), il est nécessaire de passer par une étape préliminaire qui porte sur l'introduction d'une variable redondante définie à l'interface entre deux cellules de contrôle (voir la figure 2.3), en s'inspirant d'une stratégie notamment adoptée par Clerc[Cle97]. Néanmoins, l'approche considérée ici diffère de celle proposée dans [Cle97] de par l'usage qu'il est fait de cette variable redondante; en effet, comme détaillé plus bas, cette variable redondante est utilisée pour définir une variable d'interface sous la forme d'un flux numérique (normal) à l'interface entre deux cellules de contrôle appartenant à deux sous-domaines voisins.

FIG. 2.3 – *Définition d'une variable redondante à l'interface* $\Gamma = \Omega_1 \cap \Omega_2$

Soit $[s_i,s_j]$ un segment tel que la cellule C_i (associée au sommet s_i) et la cellule C_j (associée au sommet s_j) appartiennent à deux sous-domaines voisins notés Ω_1 et Ω_2. On formule un algorithme de type Schwarz additif en posant les conditions d'interface suivantes, localement dans chaque sous-domaine :

$$\left\{ \begin{array}{rcl} \mathcal{A}_{ij}^{-}(W_1^*)^{(k+1)} & = & \mathcal{A}_{ij}^{-}(W_2^*)^{(k)} \\ \mathcal{A}_{ij}^{+}(W_2^*)^{(k+1)} & = & \mathcal{A}_{ij}^{+}(W_1^*)^{(k)} \end{array} \right. \tag{2.21}$$

où on a noté $\mathcal{A}_{ij}^{\pm} \equiv \mathcal{A}^{\pm}(\widetilde{W}_{ij}^{n},\vec{\nu}_{ij})$. Par ailleurs, W_1^* et W_2^* désignent des variables redondantes dans chaque sous-domaine, définies à l'interface, et vérifiant les relations suivantes :

$$\left\{ \begin{array}{rcl} \mathcal{A}_{ij}^{+} W_1^* & = & \mathcal{A}_{ij}^{+} W_i \\ \mathcal{A}_{ij}^{-} W_2^* & = & \mathcal{A}_{ij}^{-} W_j \end{array} \right. \tag{2.22}$$

Dans la suite, on écrit simplement W_i au lieu de $W_i^{(k+1)}$. Dans chaque sous-domaine, pour ce qui concerne les équations qui correspondent aux sommets W_i (sous-domaine Ω_1) et W_j (sous-domaine Ω_2), on remplace la contribution du voisin, exprimée par le couplage avec $W_j \in \Omega_2$ pour $W_i \in \Omega_1$ (respectivement W_i pour W_j), par la nouvelle variable W_1^* (respectivement W_2^*).

Il est clair que les conditions 2.21 expriment lors de la convergence de l'algorithme de Schwarz la continuité des flux normaux à l'interface $\Gamma = \Omega_1 \cap \Omega_2$ et entrainenent la continuité de la variable W^*: $W_1^* = W_2^*$. Ce qui veut dire que dans l'optique de l'équivalence algorithme de Schwarz-résolution d'un problème sous-structuré on peut directement introduire une variable auxiliaire notée W^* qui est telle que :

$$\begin{array}{rcl} \left(\mathcal{A}_{ij}^{-} W_j\right) |_{s_j} & = & \left(\mathcal{A}_{ij}^{-} W^*\right) |_{\frac{s_i + s_j}{2}} \\ & \text{et} & \\ \left(\mathcal{A}_{ij}^{+} W_i\right) |_{s_i} & = & \left(\mathcal{A}_{ij}^{+} W^*\right) |_{\frac{s_i + s_j}{2}} \end{array} \tag{2.23}$$

et on définit :

$$\Phi = |\mathcal{A}_{ij}| W^* = \mathcal{A}_{ij}^{+} W_i - \mathcal{A}_{ij}^{-} W_j \tag{2.24}$$

comme étant une nouvelle inconnue du problème qui correspond ici à un flux à travers l'interface.

On peut alors écrire :

$$\Phi = \left(\mathcal{T}(\widetilde{W}_{ij}^{n},\vec{\nu}_{ij})\ |\Lambda(\widetilde{W}_{ij}^{n}),\vec{\nu}_{ij}|\ \mathcal{T}^{-1}(\widetilde{W}_{ij}^{n},\vec{\nu}_{ij})\right) W^*$$

$$\Updownarrow$$

$$W^* = \left(\mathcal{T}(\widetilde{W}_{ij}^{n},\vec{\nu}_{ij})\ |\Lambda(\widetilde{W}_{ij}^{n},\vec{\nu}_{ij})|^{-1}\ \mathcal{T}^{-1}(\widetilde{W}_{ij}^{n},\vec{\nu}_{ij})\right) \Phi$$

où $\Lambda(\widetilde{W}_{ij}^{n},\vec{\nu}_{ij})$ est la matrice diagonale dont les éléments sont les valeurs propres de $\mathcal{A}(\widetilde{W}_{ij}^{n},\vec{\nu}_{ij})$ et $\mathcal{T}(\widetilde{W}_{ij}^{n},\vec{\nu}_{ij})$ est la matrice dont les colonnes sont les vecteurs propres associés. En notant $\mathcal{T}_{ij} = \mathcal{T}(\widetilde{W}_{ij}^{n},\vec{\nu}_{ij})$ et $\Lambda_{ij}^{\pm} = \Lambda(\widetilde{W}_{ij}^{n},\vec{\nu}_{ij})$, les parties positive et négative du flux numérique ainsi défini sont données par :

$$\begin{array}{rcl} \Phi^{\pm} & = & \mathcal{A}_{ij}^{\pm} W^* \\ & = & \left(\mathcal{T}_{ij}\ \Lambda_{ij}^{\pm}\ \mathcal{T}_{ij}^{-1}\right) W^* \\ & = & \left(\mathcal{T}_{ij}\ \Lambda_{ij}^{\pm}\ |\Lambda_{ij}^{-1}|\ \mathcal{T}_{ij}^{-1}\right) \Phi \end{array} \tag{2.25}$$

ce que l'on peut encore écrire sous la forme condensée :

$$\Phi^{\pm} = P^{\pm}\Phi \tag{2.26}$$

Les quantités définies ci-dessus ont la signification des flux entrant (Φ^-) et sortant (Φ^+) à l'interface d'un sous-domaine donné lorsque la normale sortante est donnée par $\vec{\nu}_{ij}$.

Par ailleurs, le flux élémentaire linéarisé associé aux cellules de contrôle C_i et C_j peut être exprimé comme (voir aussi l'équation (2.10)) :

$$\left\{ \begin{array}{rcl} \Phi(W_i,W_j,\vec{\nu}_{ij}) & = & \left(\mathcal{A}(W_i^n,\vec{\nu}_{ij}) - \mathcal{A}^-(\widetilde{W}_{ij},\vec{\nu}_{ij})\right) W_i + \mathcal{A}^-(\widetilde{W}_{ij},\vec{\nu}_{ij})W_j \\ \Phi(W_j,W_i,\vec{\nu}_{ji}) & = & -\Phi(W_i,W_j,\vec{\nu}_{ij}) \end{array} \right. \tag{2.27}$$

sachant que $\Phi(W_i,W_j,\vec{\nu}_{ij})$ contribue au flux total au sommet s_i de Ω_1 (soit $\Psi(W)_i|_{\Omega_1}$), et $\Phi(W_j,W_i,\vec{\nu}_{ji})$ contribue au flux total au sommet s_j de Ω_2 (soit $\Psi(W)_j|_{\Omega_2}$).

Par ailleurs, dans les relations ci-dessus, la linéarisation s'entend au sens de (2.19), c'est-à-dire que, pour être rigoureux, il conviendrait de noter δW_i^{n+1} (respectivement δW_j^{n+1}) au lieu de W_i (respectivement W_j). En faisant l'approximation suivante à l'interface Γ :

$$\mathcal{A}(W_i^n,\vec{\nu}_{ij}) - \mathcal{A}^-(\widetilde{W}_{ij},\vec{\nu}_{ij}) \cong \mathcal{A}^+(\widetilde{W}_{ij},\vec{\nu}_{ij})$$

et en notant $\mathcal{A}_i = \mathcal{A}(W_i^n,\vec{\nu}_{ij})$ et $\mathcal{A}_j = \mathcal{A}(W_j^n,\vec{\nu}_{ij})$, on peut combiner les équation (2.23), (2.25) et (2.26) pour reformuler (2.27) comme :

$$\left\{ \begin{array}{rcl} \Phi(W_i,W^\star,\vec{\nu}_{ij}) & = & \left(\mathcal{A}_i - \mathcal{A}_{ij}^-\right) W_i + \mathcal{A}_{ij}^- W^* \\ & = & \left(\mathcal{A}_i - \mathcal{A}_{ij}^-\right) W_i + P^-\Phi \quad \text{(dans } \Omega_1) \\ \Phi(W_j,W^\star,\vec{\nu}_{ij}) & = & -\mathcal{A}_j^- W_j - \mathcal{A}_{ij}^+ W^* \\ & = & -\mathcal{A}_j^- W_j - P^+\Phi \quad \text{(dans } \Omega_2) \end{array} \right. \tag{2.28}$$

A partir des flux élémentaires (2.28), et aussi des flux élémentaires (inchangés) pour les sommets purement internes à Ω_1 et Ω_2 (pour simplifier la présentation, on se limite ici au cas d'une décomposition en deux sous-domaines), on peut construire un système linéaire implicite (analogue de (2.20)) où l'on distingue les inconnues purement internes à un sous-domaine (vecteurs d'état) des inconnues à l'interface (flux normaux) :

$$\begin{pmatrix} \mathcal{M}_1 & 0 & \mathcal{M}_{1\Phi} \\ 0 & \mathcal{M}_2 & \mathcal{M}_{2\Phi} \\ \mathcal{F}_{1\Phi} & \mathcal{F}_{2\Phi} & \mathrm{Id} \end{pmatrix} \begin{pmatrix} W_1 \\ W_2 \\ \Phi \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ 0 \end{pmatrix} \tag{2.29}$$

où $\mathcal{M}_1$ (respectivement $\mathcal{M}_2$) est la matrice implicite qui couple les sommets internes à Ω_1 (respectivement Ω_2) tandis que $\mathcal{F}_{1\Phi}$, $\mathcal{F}_{2\Phi}$, $\mathcal{M}_{1\Phi}$ et $\mathcal{M}_{2\Phi}$ sont des matrices de couplage entre les inconnues internes et les inconnues à l'interface Γ. Ces différentes matrices sont détaillées dans l'annexe 2.5. A ce stade, les inconnues internes peuvent être éliminées du système linéaire (2.29), par application d'un principe de sous-structuration, au profit des inconnues à l'interface pour aboutir à la formulation d'un système interface :

$$\begin{array}{rcl} S\Phi & = & [Id - (\mathcal{F}_{1\Phi}\mathcal{M}_1^{-1}\mathcal{M}_{1\Phi} + \mathcal{F}_{2\Phi}\mathcal{M}_2^{-1}\mathcal{M}_{2\Phi})]\Phi \\ & & \begin{array}{rcl} = & g \\ = & -[\mathcal{F}_{1\Phi}\mathcal{M}_1^{-1}b_1 + \mathcal{F}_{2\Phi}\mathcal{M}_2^{-1}b_2] \end{array} \end{array} \tag{2.30}$$

Une fois le système (2.30) résolu pour les valeurs de Φ, on obtient les valeurs des inconnues purement internes par résolution en parallèle des systèmes locaux suivants :

$$\left\{ \begin{array}{rcl} W_1 & = & \mathcal{M}_1^{-1}(b_1 - \mathcal{M}_{1\Phi}\Phi) \\ W_2 & = & \mathcal{M}_2^{-1}(b_2 - \mathcal{M}_{2\Phi}\Phi) \end{array} \right. \tag{2.31}$$

Remarque 6 On aurait pu adopter une autre linéarisation du flux élémentaire (2.28) de façon à utiliser exactement les quantités mises en jeu dans les équations (2.23), (2.24) et (2.25) sans autre approximation. On aurait pu, par exemple, adopter la linéarisation :

$$\left\{ \begin{array}{rcl} \Phi(W_i, W^\star, \vec{\nu}_{ij}) & = & \left(\mathcal{A}_i - \mathcal{A}_{ij}^-\right) W_i + \mathcal{A}_{ij}^- W^\star \\ \\ \Phi(W_j, W^\star, \vec{\nu}_{ij}) & = & -\left(\mathcal{A}_j - \mathcal{A}_{ij}^+\right) W_j - \mathcal{A}_{ij}^+ W^\star \end{array} \right. \tag{2.32}$$

On notera néanmoins que les expressions ci-dessus conduisent à une formulation non-conservative pour le segment interface en question. Il existe par ailleurs une troisième façon de traiter ce problème qui consiste à adopter dans tout le domaine de calcul (c'est-à-dire pour les segments internes comme pour les segments interface) une autre linéarisation du flux numérique qui combine les deux possibilités (2.28) et (2.32). Dans cette approche, le flux numérique est conservatif et est exprimé comme :

$$\left\{ \begin{array}{rcl} \Phi(W_i, W^\star, \vec{\nu}_{ij}) & = & \mathcal{A}_{ij}^+ W_i + \mathcal{A}_{ij}^- W^\star \\ \\ & = & \mathcal{A}_{ij}^+ W_i + P^- \Phi \\ \\ \Phi(W_j, W^\star, \vec{\nu}_{ij}) & = & -\mathcal{A}_{ij}^- W_j - \mathcal{A}_{ij}^+ W^\star \\ \\ & = & -\mathcal{A}_{ij}^- W_j - P^+ \Phi \end{array} \right. \tag{2.33}$$

Comme on a vu dans la partie théorique concernant la résolution de ces équations, l'algorithme de Schwarz est rien d'autre qu'une méthode de Jacobi par blocs appliquée à une système sous-structuré (défini à l'aide des variables d'interface). En tenant compte de cette équivalence on obtient la méthode itérative suivante appliquée au système 2.30.

ALGORITHME 2 algorithme de type Jacobi pour résoudre le système interface $S\Phi = g$ dans le cas d'une décomposition en 2 sous-domaines.

1. Initialisation : $\Phi = \Phi^{(0)}$

2. Calcul de $g = g_1 + g_2$ (incluant des étapes de communication pour assembler les contributions locales) avec $g_i = \mathcal{F}_i x_i$ où x_i est obtenue par la résolution du système local (i.e. posé dans Ω_i) $\mathcal{M}_i x_i = b_i$

3. POUR $k = 0, \cdots, K$ FAIRE

- Sous-domaine Ω_1 en parallèle :

$$y_1 = \mathcal{M}_{12}\Phi^{(k)} \quad \text{et} \quad \Phi_1 = \mathcal{F}_1 v_1$$

  où v_1 est est obtenue par la résolution du système local $\mathcal{M}_1 v_1 = y_1$
- Sous-domaine Ω_2 en parallèle :

$$y_2 = \mathcal{M}_{21}\Phi^{(k)} \quad \text{et} \quad \Phi_2 = \mathcal{F}_2 v_2$$

  où v_2 est obtenue par la résolution du système local $\mathcal{M}_2 v_2 = y_2$
- Assemblage des contributions locales (incluant des étapes de communication) :

$$\Phi^{k+1} = \Phi_1 + \Phi_2 + g$$

4. SI $\|\Phi^{(k+1)} - \Phi^{(k)}\| < \epsilon$ ALORS sortir de la boucle principale SINON retour en **3**

Ensuite on peut accélérer cette résolution en utilisant une méthode de type Krlylov, comme le GMRES.

2.4 Stratégie de résolution des systèmes locaux

La méthode de résolution par sous-domaine proposée dans la section 2.3 fait apparaître des étapes de résolution locales indépendantes (c'est-à-dire parallèles) dans chaque sous-domaine (voir par exemple l'algorithme 2). Une caractéristique de l'étude numérique réalisée à la section 2.6.2 réside dans l'utilisation de stratégies de résolution itératives pour les sous-problèmes en question. Il s'agit là d'une variation importante par rapport à ce qui est généralement fait dans le domaine où des méthodes de factorisation sont utilisées pour résoudre exactement les systèmes locaux. Deux raisons principales motivent la stratégie adoptée ici :

- l'utilisation de méthodes de factorisation dans chaque sous-domaine, bien qu'effectuées en parallèle, rend le coût d'une itération de la méthode de décomposition de domaine trop élevé pour que cette dernière soit compétitive avec des stratégies de résolution alternatives. En effet, comme souligné dans la section 2.1.4, les systèmes linéaires issus du schéma du type Euler implicite linéarisé pour l'intégration en temps sont dans la grande majorité des cas résolus de façon approchée avec une précision relativement faible (2 ou 3 ordres de grandeur de réduction du résidu linéaire initial). En particulier, dans ces cas, résoudre plus précisément les systèmes linéaires en question ne permet pas d'améliorer la convergence non-linéaire vers l'état stationnaire;
- les méthodes de résolution directes sont gourmandes en en particulier lorsque le nombre de sous-domaines reste modéré (comme c'est le cas pour tous les résultats numériques présentés dans cette étude) et généralement coûteuses en temps CPU.

Deux stratégies de résolution ont été considérées dans cette étude :

- une stratégie *monogrille* basée sur la méthode de relaxation de Jacobi pour la résolution approchée du système linéaire (2.19), c'est-à-dire dans le contexte d'une stratégie de résolution globale;
- une stratégie *multigrille* appliquée à la résolution des systèmes linéaires locaux qui caractérisent la méthode de décomposition de domaine proposée dans cette étude.

2.4.1 Généralités sur les méthodes multigrilles

La méthode multigrille considérée ici a pour but d'accélérer la résolution itérative de systèmes linéaires tels que ceux résultant de l'utilisation du schéma d'Euler implicite linéarisé décrit dans la section 2.1.4. D'un point de vue algorithmique, à chaque itération en temps, on construit l'approximation (par la méthode décrite dans la section 2.1.2) du second membre et des éléments constitutifs de la matrice implicite. Dans le cas d'approximations par éléments finis sur des maillages non-structurés, aucune factorisation spatiale de la matrice ne peut être effectuée et la largeur de bande est inconnue a priori. C'est pourquoi on résout par relaxation. L'analyse de Fourier (en espace), ou analyse modale, de systèmes linéaires représentatifs des équations à résoudre après discrétisation d'équations aux dérivées partielles, permet d'ordonner les composantes de l'erreur itérative suivant les valeurs d'un (ou plusieurs) paramètres fréquentiels, la valeur de la plus haute fréquence étant liée au pas de discrétisation en espace, ou, à l'inverse, au nombre de degrés de liberté. Un principe de base concernant les méthodes itératives classiques, telles que l'itération de Jacobi, est le suivant : l'itération agit avec la plus grande efficacité sur les composantes de l'erreur de hautes fréquences ; à l'inverse, ce sont les composantes de basses fréquences qui persistent et sont la manifestation de la raideur du système. Par contre, ces modes de basses fréquences, qui sont la représentation discrète de fonctions *lisses* des coordonnées d'espace, peuvent être interpolés sans grande perte de précision sur des grilles de moindre finesse.

La méthode multigrille[Hac85]-[Bri87]- [Wes91]-[Dés98] est issue de cette observation. On construit a priori une hiérarchie de niveaux de grille, associés à des intervalles de fréquences différents. Une méthode itérative de type classique, dite *lisseur* est utilisée pour atténuer efficacement les modes de hautes fréquences de l'erreur associés à la discrétisation la plus fine ; le problème résiduel est ensuite reformulé sur une grille plus grossière, sur laquelle on lisse à nouveau avant de transférer le problème sur une grille encore plus grossière, et ainsi de suite, jusqu'à ce que le problème devienne trivial. On construit ensuite à l'inverse des approximations sur les différentes grilles de dimensions croissantes par prolongement (et éventuellement lissage). En procédant de la sorte, les phases de lissage associées aux différentes grilles éliminent efficacement les composantes de l'erreur itérative suivant les différentes fréquences, jusqu'à la plus basse qui est éliminée par résolution directe d'un système trivial. Dans le cas d'un problème modèle linéaire elliptique, la théorie permet d'établir que la complexité de la méthode multigrille est proportionnelle au nombre de degrés de liberté. Cela signifie que le coût de résolution du système à la précision fixée par l'erreur d'approximation est directement proportionnel au nombre d'inconnues du problème.

2.4.2 Méthode multigrille linéaire par agglomération

Les méthodes multigrilles *géométriques* (par opposition aux méthodes multigrilles *algébriques*) utilisent donc différentes discrétisations du domaine de calcul. La grille fine est le support de la solution discrète recherchée ; les grilles grossières correspondent à des discrétisations moins fines du domaine de calcul et permettent d'accélérer la résolution itérative. Le choix des discrétisations grossières en conjugaison avec l'utilisation de maillages non-structurés n'est pas aisé et a déjà fait l'objet de nombreux travaux.

Une première approche consiste, à partir d'une discrétisation grossière, à générer par un processus récursif des discrétisations de plus en plus fines, par subdivision des éléments de la discrétisation grossière initiale. Les différents niveaux ainsi obtenus sont donc totalement emboîtés. Mavriplis et Jameson[MJ87]-[MJ88b] ont, quant à eux, développé une méthode utilisant des maillages non emboîtés et, plus récemment, une procédure développée par Guillard[Gui93] consiste à générer une séquence de maillages éléments finis par suppression de points et reconstruction d'une nouvelle triangulation. Ces différentes approches ont permis de construire des algorithmes de résolution efficaces et robustes dans de nombreux domaines d'application. Cependant, le fait qu'elles nécessitent de générer et gérer l'ensemble des maillages pour chaque application, rendent leur utilisation en contexte industriel assez lourde.

La méthode multigrille par agglomération repose sur une technique permettant de construire des discrétisations grossières en utilisant la seule donnée du maillage élément fini fin. Cette méthode a été mise au point par Lallemand[Lal88] pour la résolution du système d'équations d'Euler, puis étendue par Koobus[Koo94] et Carré[Car95] au traitement du système d'équations de Navier-Stokes couplé à un modèle de turbulence $k-\epsilon$. Le principe de base de la méthode est de construire une grille grossière par regroupement ou agglomération des cellules de contrôle de la grille immédiatement plus fine. On peut ainsi construire une séquence de grilles grossières dont les cellules de contrôle ou agglomérats sont emboîtées. Cette spécificité facilite la construction des opérateurs permettant le transfert de quantités, telles que le vecteur solution ou le vecteur résidu, entre les différentes grilles.

2.4.2.1 Construction des agglomérats

Plusieurs stratégies peuvent être envisagées pour construire ces agglomérats. La plus classique utilise un algorithme de type « glouton » adapté au contexte d'un maillage en cellules de contrôle (i.e. le maillage dual du maillage éléments finis dans le cas du maillage fin de départ). La version décrite ici est celle proposée par Lallemand[Lal88]. La procédure est appliquée indifféremment au (sous-)maillage dual du (sous-)maillage fin de départ et à un (sous-)maillage grossier lui-même constitué d'agglomérats.

ALGORITHME 3 *Algorithme d'agglomération de type « glouton » (construction du niveau grossier $k+1$ à partir du niveau fin k).*

INITIALISATION *du compteur des zones grille grossière : $j=0$*
POUR *chaque zone grille fine C_i^k ($i=1,\cdots,n_k$)* FAIRE
  SI *C_i^k appartient déjà à une nouvelle zone* ALORS
    - *Passer à la zone grille fine suivante*
  SINON
    - *Créer un nouveau groupe contenant C_i^k (et $j=j+1$)*
    - *Mettre dans ce groupe toutes les zones de G^k voisines de C_i^k (et non déjà agglomérées)*
    SI *le nouveau groupe créé ne contient que C_i^k* ALORS
      - *Détruire ce groupe et mettre C_i^k dans un groupe existant (et $j=j-1$) contenant au moins une cellule de contrôle voisine de C_i^k*
    FIN SI
  FIN SI
FIN FAIRE

Les cellules de contrôle des niveaux grossiers sont généralement désignées par le terme *macro-cellules*, *macro-éléments* ou encore *zones*. Nous illustrons ci-dessous le fonctionnement de la procédure d'agglomération sur le maillage d'un profil d'aile NACA0012. (voir la figure 2.4).

FIG. 2.4 – *Illustration de la procédure d'agglomération*

2.4.2.2 Opérateurs de transfert

Le choix des opérateurs de transfert doit s'effectuer en fonction de critères de simplicité (des opérateurs d'ordre élevé entraînant des coûts de résolution importants) et de robustesse de la méthode. Différents travaux [Hem90]-[Wes91] on montré qu'une condition nécessaire à la définition des opérateurs de transfert est que la somme de leurs ordres de précision soit supérieure ou égale à l'ordre des équations différentielles résolues. En notant $\bar{M}$ l'opérateur de la grille grossière et M l'opérateur de la grille fine, on a :

$$\bar{M} = r.M.p$$

où $O(r)+O(p) = O(M)$, $O(.)$ représentant l'ordre de précision des différents opérateurs. Cette relation, proposée par Hemker[Hem90], est une condition nécessaire au non développement de hautes fréquences sur la grille fine. Par exemple, la résolution des équations paraboliques devrait nécessiter l'emploi d'opérateurs de prolongement et de restriction linéaires (ou du second ordre pour la projection dans le cas d'une injection canonique). La construction d'opérateurs d'ordre élevé dans notre contexte (c'est-à-dire d'une méthode de discrétisation de type volumes finis) est une tâche difficile. Pour des raisons de simplicité et de coût CPU, on a retenu des opérateurs précis uniquement au premier ordre.

Soit $[W_k]$ le vecteur d'état global défini sur le k-ième niveau de discrétisation tel que $[W_k]\left(Z_l^k\right)$ désigne le vecteur des variables conservatives associé à la l-ième macro-cellule $\left(Z_l^k\right)$ de la grille G_k ($l = 1, \cdots, N_k$, où N_k est le nombre total de macro-cellules de la grille G_k).

- *Opérateur de restriction de la solution* : $r^s_{k,k+1}$. Cet opérateur est utilisé pour transférer le vecteur d'état global $[W_k]$ depuis le niveau fin k vers le niveau grossier $k+1$ (voir la figure 2.5). Il s'exprime comme une moyenne pondérée des solutions du niveau k :

$$\boxed{\texttt{POUR}\ j = 1, \cdots, N_{k+1} \quad : \quad \left[r^s_{k,k+1}\left(W_k\right)\right]\left(Z_j^{k+1}\right) = \frac{\sum\limits_{l \in U^k(j)} \text{aire}\left(Z_l^k\right) W_k\left(Z_l^k\right)}{\text{aire}\left(Z_j^{k+1}\right)}}$$

  où $U^k(j)$ désigne l'ensemble des indices l des macro-cellules Z_l^k du niveau k avec $Z_l^{k+1} = \bigcup\limits_{l \in U^k(j)} Z_l^k$. Le vecteur d'état ainsi transférer est uniquement utilisé pour construire la matrice implicite sur le niveau $k+1$.

FIG. 2.5 – *Opérateur de restriction de la solution :* $r^s_{k,k+1}$

– *Opérateur de restriction du second membre* : $r^{rhs}_{k,k+1}$. Cet opérateur est utilisé pour transférer le vecteur second-membre (résidu) $[b_k]$ depuis le niveau fin k vers le niveau grossier $k+1$ (voir la figure 2.6). Le second membre sur le niveau $k+1$ est défini comme la somme des seconds membres du niveau k :

$$\boxed{\texttt{POUR}\ j = 1, \cdots, N_{k+1} \quad : \quad \left[r^{rhs}_{k,k+1}\left(b_k\right)\right]\left(Z^{k+1}_j\right) = \sum_{l \in U^k(j)} b_k\left(Z^k_l\right)}$$

FIG. 2.6 – *Opérateur de restriction du second membre :* $r^{rhs}_{m,m+1}$

– *Opérateur de prolongement de la correction* : $p^s_{k+1,k}$. Cet opérateur est utilisé pour transférer le vecteur correction depuis le niveau grossier $k+1$ vers le niveau fin k (voir la figure 2.7). Il est défini comme une injection naturelle de la correction du niveau grossier :

$$\boxed{\texttt{POUR}\ l = 1, \cdots, N_k \quad : \quad \left[p^s_{k+1,k}\left(e_{k+1}\right)\right]\left(Z^k_j\right) = e_{k+1}\left(Z^{k+1}_l\right)}$$

FIG. 2.7 – *Opérateur de prolongement de la correction :* $p^s_{k+1,k}$

2.4.2.3 Approximation des termes convectifs sur les grilles grossières

On rappelle que la matrice implicite du système linéaire (2.34) est construite en partie à partir d'une linéarisation du flux convectif précis au premier ordre en espace. En fait, l'approximation volumes finis grille fine des termes convectifs est suffisamment générale pour s'appliquer directement aux macro-cellules d'un niveau grossier. Aussi, le calcul des flux convectifs s'effectue de façon identique sur les grilles grossières. Les variables conservatives sont transférées entre les différentes grilles à l'aide de l'opérateur de transfert $r^s_{k,k+1}$ défini précédemment. Les vecteurs normaux à la frontière d'une macro-cellule résultent, quant à eux, de la sommation des vecteurs normaux appartenant aux cellules de contrôle de la grille fine qui

possèdent une frontière commune avec la macro-cellule. Pour plus de détails on consultera par exemple Carré[Car95].

2.4.2.4 Algorithme multigrille de type V-cycle

Nous décrivons ici l'algorithme V-cycle pour la résolution d'un système linéaire de la forme :

$$AX = b \tag{2.34}$$

Dans le contexte qui nous intéresse ici, la méthode multigrille linéaire a pour but d'accélérer la résolution itérative du système linéaire (2.34) par une méthode de relaxation qui peut être Jacobi ou Gauss-Seidel. Ces méthodes jouent donc le rôle de lisseur dans l'algorithme multigrille. Soit $\{G_k\}_{1\leq k\leq N_g}$ une hiérarchie de grilles volumes finis ($\{G_1\}$ étant la grille la plus fine et $\{G_{N_g}\}$ la grille la plus grossière). Le système (2.34) est posé sur la grille la plus fine soit encore :

$$A_{(1)}X_{(1)} = b_{(1)} \tag{2.35}$$

On note :

- X^α : α-ième itéré multigrille approchant X sur la grille fine avec X^0=0;
- $X^0_{(k)}$: itéré initial de la méthode de relaxation pour résoudre le système sur la grille G_k;
- $p_{k+1,k}$ et $r_{k,k+1}$ correspondent respectivement aux opérateurs de prolongation et de restriction;
- ν_1 et ν_2 désignent les nombres de relaxations des étapes de pré- et post-lissage (ν_g désigne le nombre d'itérations de la méthode de relaxation sur la grille la plus grossière);
- ω est un paramètre de sous-relaxation;
- $A_{ij\,(k)}$ est la matrice résultante de la discrétisation sur la grille G_k;
- $b_{i(k)}$ correspond au second membre du système linéaire sur G_k.

L'algorithme définissant un V-cycle est alors le suivant.

`ALGORITHME 4` algorithme multigrille V-cycle pour la résolution d'un système linéaire $AX = b$.

1. Initialisation : $X^0_{(1)} = X^\alpha$.

2. Phase descendante : détermination des corrections sur les grilles grossières.

`POUR` $k = 1$ au niveau grossier $k = N_g - 1$ `FAIRE`

- Itéré initial : $X^0_{i(k)} = \begin{cases} A_{ii\,(k)}^{-1} b_{i(k)}, & \text{si } k > 1 \\ X`_i^\alpha, & \text{si } k = 1 \end{cases}$
- Pré-lissages (ν_1 relaxations pour $k < N_g$ et ν_g relaxations sur le niveau le plus grossier i.e. $k = N_g$) :

- pour **Jacobi**

$$\overline{X_{i(k)}^{l}} = A_{ii(k)}^{-1}\left[b_{i(k)} - \sum_{j\neq i} A_{ij(k)} X_{j(k)}^{l-1}\right] \quad , \quad l = 1,...,\nu_1$$

- pour **Gauss-Seidel**

$$\overline{X_{i(k)}^{l}} = A_{ii(k)}^{-1}\left[b_{i(k)} - \sum_{j<i} A_{ij(k)} X_{j(k)}^{l} - \sum_{j>i} A_{ij(k)} X_{j(k)}^{l-1}\right] \quad , \quad l = 1,...,\nu_1$$

– Mise-à-jour : $X_{i(k)}^{l} = (1-\omega)\, X_{i(k)}^{l-1} + \omega \overline{X_{i(k)}^{l}}$

– Restriction : $b_{i(k+1)} = r_{k,k+1}\left(b_{i(k)} - \sum_{j\neq i; j=i} A_{ij(k)} X_{j(k)}^{\nu_1}\right)$

`FIN FAIRE`

3. Phase montante : prolongement des corrections du niveau N_g vers la grille fine.

`POUR` $k = N_g$ au niveau fin $k = 1$ `FAIRE`

– Prolongement et mise-à-jour : $X_{i(k)} = X_{i(k)}^{\nu_1} + p_{k+1,k}\left(X_{i(k+1)}\right)$
  Itéré initial : $X_{i(k)}^{0} = X_{i(k)}$
– Post-lissages (ν_2 relaxations) :

- pour **Jacobi**

$$\overline{X_{i(k)}^{l}} = A_{ii(k)}^{-1}\left[b_{i(k)} - \sum_{j\neq i} A_{ij(k)} X_{j(k)}^{l-1}\right] \quad , \quad l = 1,...,\nu_2$$

- pour **Gauss-Seidel**

$$\overline{X_{i(k)}^{l}} = A_{ii(k)}^{-1}\left[b_{i(k)} - \sum_{j<i} A_{ij(k)} X_{j(k)}^{l} - \sum_{j>i} A_{ij(k)} X_{j(k)}^{l-1}\right] \quad , \quad l = 1,...,\nu_2$$

– Mise-à-jour : $X_{i(k)}^{l} = (1-\omega)\, X_{i(k)}^{l-1} + \omega \overline{X_{i(k)}^{l}}$ puis $X_{i(k)} = X_{i(k)}^{\nu_2}$

4. Réactualisation de la solution multigrille : $X^{\alpha+1} = X_{(1)}$

5. `SI` $\|X^{\alpha+1} - X^{\alpha}\| < \epsilon$ `ALORS` sortir de la boucle principale `SINON` retour en **2.**

2.5 Annexe : matrices locales et matrices de couplage pour Euler

On précise ici les expressions des différents termes de la matrice du système linéaire (2.29). Pour simplifier, on considère le cas d'une décomposition en deux sous-domaines Ω_1 et Ω_2. On rappelle l'expression de cette matrice :

$$\mathcal{M} = \begin{pmatrix} \mathcal{M}_1 & 0 & \mathcal{M}_{1\Phi} \\ 0 & \mathcal{M}_2 & \mathcal{M}_{2\Phi} \\ \mathcal{F}_{1\Phi} & \mathcal{F}_{2\Phi} & \mathsf{Id} \end{pmatrix}$$

où $\mathcal{M}_1$ (respectivement $\mathcal{M}_2$) est la matrice implicite qui couple les sommets internes à Ω_1 (respectivement Ω_2) tandis que $\mathcal{F}_{1\Phi}$, $\mathcal{F}_{2\Phi}$, $\mathcal{M}_{1\Phi}$ et $\mathcal{M}_{2\Phi}$ sont des matrices de couplage entre les inconnues internes et les inconnues à l'interface $\Gamma = \Omega_1 \bigcup \Omega_2$.

Le schéma implicite décrit dans la section 2.1.4 est basé sur la linéarisation du flux numérique précis au premier ordre (2.8). Pour un segment $[s_i,s_j]$, une version implicite du flux élémentaire est formellement écrite :

$$\Phi_{ij}^{n+1} = \Phi_{\mathcal{F}}(W_i^n,W_j^n,W_i^{n+1},W_j^{n+1},\vec{\nu}_{ij})$$

En notant $U = W_i^n$, $V = W_j^n$, $W = W_i^{n+1}$ et $Z = W_j^{n+1}$ et en écrivant un développement de Taylor au premier ordre du flux élémentaire implicite, on obtient :

$$\Phi_{\mathcal{F}}(U,V,W,Z,\vec{\nu}_{ij}) = \Phi_{\mathcal{F}}(U,V,\vec{\nu}_{ij}) + \left(\frac{\partial \Phi}{\partial U}\right)(W-U) + \left(\frac{\partial \Phi}{\partial V}\right)(Z-V) \tag{2.36}$$

où :

$$\Phi_{\mathcal{F}}(U,V,\vec{\nu}_{ij}) = \Phi_{\mathcal{F}}(W_i^n,W_j^n,\vec{\nu}_{ij})$$

correspond au flux explicite (2.8). L'expression (2.36) peut-être simplifiée dans le cas où la fonction de flux numérique est de la forme :

$$\Phi_{\mathcal{F}}(U,V,\vec{\nu}_{ij}) = H_1(U,V,\vec{\nu}_{ij})U + H_2(U,V,\vec{\nu}_{ij})V$$

Par exemple, pour la fonction de flux numérique associée au solveur de Riemann approché de Roe[Roe81] on a :

$$\begin{aligned} \Phi_{\mathcal{F}}(W_i,W_j,\vec{\nu}_{ij}) &= \mathcal{F}(W_i,\vec{\nu}_{ij}) &+ \mathcal{A}^-(\widetilde{W}_{ij},\vec{\nu}_{ij})(W_j - W_i) \\ &= \mathcal{A}(W_i,\vec{\nu}_{ij})W_i &+ \mathcal{A}^-(\widetilde{W}_{ij},\vec{\nu}_{ij})(W_j - W_i) \end{aligned}$$

par suite :

$$\begin{cases} H_1(W_i,W_j,\vec{\nu}_{ij}) &= \mathcal{A}(W_i,\vec{\nu}_{ij}) - \mathcal{A}^-(\widetilde{W}_{ij},\vec{\nu}_{ij}) \\ H_2(W_i,W_j,\vec{\nu}_{ij}) &= \mathcal{A}^-(\widetilde{W}_{ij},\vec{\nu}_{ij}) \end{cases}$$

En d'autres mots, on obtient une linéarisation approchée (et donc une matrice Jacobienne approchée) si on suppose que :

$$\frac{\partial \Phi}{\partial U} \approx H_1(U,V,\vec{\nu}_{ij}) \quad \text{et} \quad \frac{\partial \Phi}{\partial V} \approx H_2(U,V,\vec{\nu}_{ij})$$

On notera au passage que pour la fonction de flux numérique associée au schéma de Steger et Warming[SW81] (voir l'expression (2.15)), on obtient :

$$\begin{cases} H_1(U,V,\vec{\nu}_{ij}) &= \mathcal{A}^+(U,\vec{\nu}_{ij}) \\ H_2(U,V,\vec{\nu}_{ij}) &= \mathcal{A}^-(V,\vec{\nu}_{ij}) \end{cases}$$

On peut maintenant préciser l'équation nodale de base correspondant au produit matrice-vecteur :

$$\mathcal{M}\,(\delta W_1\ ,\ \delta W_2\ ,\ \delta\Phi)^T = (b_1\ ,\ b_2\ ,\ 0)^T$$

où $\delta W_{1,2} = W_{1,2}^{n+1} - W_{1,2}^n$. Pour simplifier, on ne prend pas en compte les termes associés aux flux convectifs calculés aux frontières physiques du domaine de calcul. On note $e_{ij}^I = [s_i^I,s_j^I]$ un segment interface (voir la figure 2.3). On suppose ici que $N(s_i)$ désigne l'ensemble des sommets s_k voisins de s_i^I qui sont internes à un sous-domaine (par exemple Ω_1; dans ce cas s_k est l'extrémité d'un segment purement interne à Ω_1) alors que $N^I(s_i)$ décrit les sommets s_j^I voisins de s_i^I qui correspondent à des segments interfaces (par suite s_j^I est un sommet situé dans Ω_2 pour l'exemple considéré).

Inconnues internes : ce sont les composantes des vecteurs δW_1 et δW_2 (ces vecteurs sont formés de blocs de 4 éléments, chaque bloc étant associé à un sommet). Pour un sommet s_i interne à Ω_1 (ou Ω_2) on obtient :

$$(D_{1,2})_i^n(\delta W_{1,2})_i + \sum_{j\in N(s_i)} (E_{1,2})_{ij}^n(\delta W_{1,2})_j + \sum_{j\in N^I(s_i)} (\mathcal{M}_{12,21})_{i,e_{ij}^I}^n(\delta\Phi)_{e_{ij}^I} = (b_{1,2})_i^n \tag{2.37}$$

Dans le cas du système d'équations d'Euler bidimensionnelles, $(D_{1,2})_i^n$, $(\mathcal{M}_{12,21})_{i,e_{ij}^I}^n$ et $(E_{1,2})_{ij}^n$ sont des matrices 4×4 :

$$\begin{cases} (D_{1,2})_i^n &= \dfrac{\text{aire}(C_i)}{\Delta t^n} + \displaystyle\sum_{j\in N(i)} \left[\mathcal{A}(W_i^n,\vec{\nu}_{ij}) - \mathcal{A}^-(\widetilde{W}_{ij}^n,\vec{\nu}_{ij})\right] \\ (E_{1,2})_{ij}^n &= \mathcal{A}^-(\widetilde{W}_{ij}^n,\vec{\nu}_{ij}) \\ (\mathcal{M}_{12})_{i,e_{ij}^I}^n &= P^-(\widetilde{W}_{ij}^n,\vec{\nu}_{ij}) \\ (\mathcal{M}_{21})_{i,e_{ij}^I}^n &= -\ P^+(\widetilde{W}_{ij}^n,\vec{\nu}_{ij}) \\ (b_{1,2})_i^n &= -\displaystyle\sum_{j\in N(s_i)} \Phi_{\mathcal{F}}(W_i^n,W_j^n,\vec{\nu}_{ij}) \end{cases} \tag{2.38}$$

où $P^-(\widetilde{W}_{ij}^n,\vec{\nu}_{ij})$ et $P^+(\widetilde{W}_{ij}^n,\vec{\nu}_{ij})$ sont données par les équations (2.25) et (2.26).

Inconnues interfaces : ce sont les composantes du vecteur $\delta\Phi$ (ce vecteur est formé de blocs de 4 éléments, chaque bloc étant associé à un segment interface, voir la figure 2.3). On suppose que chaque segment interface $e_{ij}^I = [s_i^I,s_j^I]$ est tel que s_i^I appartient à Ω_1 et s_j^I à Ω_2. Alors, pour chaque segment e_{ij}^I on a :

$$(\delta\Phi)_{e_{ij}^I} + (\mathcal{F}_1)_{e_{ij}^I,i}^n(\delta W_1)_i + (\mathcal{F}_2)_{e_{ij}^I,j}^n(\delta W_2)_j = 0 \tag{2.39}$$

avec :

$$\left\{ \begin{array}{lcl} (\mathcal{F}_1)^n_{e^I_{ij},i} & = & \mathcal{A}^-(\widetilde{W}^n_{ij},\vec{\nu}_{ij}) \\ (\mathcal{F}_2)^n_{e^I_{ij},j} & = & -\mathcal{A}^+(\widetilde{W}^n_{ij},\vec{\nu}_{ij}) \end{array} \right. \tag{2.40}$$

2.6 Résultats numériques

Cette section comprend deux parties. Dans un premier temps (section 2.6.1), on cherche à valider numériquement l'étude de convergence menée à la section 1.2. Ensuite (section 2.6.2), on s'intéresse au calcul d'écoulements stationnaires autour d'un profil d'aile.

2.6.1 Validations numériques de l'étude de convergence

Les objectifs de cette section sont, d'une part, d'étudier numériquement la convergence de l'algorithme de Schwarz proposé dans la section 1.1.3 et, d'autre part, de valider le résultat de convergence (1.50). Pour cela, on utilise la méthodologie de résolution numérique du système d'équations d'Euler bidimensionnelles décrite dans la section 2.1. On se concentre ici sur la résolution du système linéaire résultant du premier pas de temps implicite par la méthode décrite dans la section 2.3; plus précisément, on résout le système interface $S\Phi = g$ (voir l'équation (2.30)) par l'une de deux méthodes suivantes :

- l'algorithme 2 de type Richardson;
- la méthode de GMRES sans réinitialisation[SS86].

2.6.1.1 Définition des cas tests

Les seuils de résolution appliqués aux systèmes locaux et au système interface sont respectivement donnés par $\varepsilon_l = \varepsilon_i = 10^{-10}$. Le nombre de Courant est fixé à la valeur 1000. Les expériences numériques portent sur deux géométries : un domaine rectangulaire de taille $[0{,}8] \times [0{,}1]$ et un profil d'aile `NACA0012`. Pour la première de ces géométries, deux types de discrétisation ont été adoptés : une triangulation structurée (c'est-à-dire obtenue à partir d'une grille différences finies cartésienne, voir la figure 2.8) et une triangulation non-structurée (voir la figure 2.9). Les caractéristiques des maillages structurés et non-structurés sont respectivement données dans les tables 2.1 et 2.2. Les Maillages RU2 et RU3 ont été obtenus par division uniforme du maillage RU1. Pour chaque géométrie, l'initialisation du calcul est donnée par l'écoulement uniforme caractérisé par $\rho_0 = 1$, $u_0 = 1$, $v_0 = 0$ et $p_0 = \dfrac{1}{\gamma M^2}$ où M désigne le nombre de Mach de référence. Pour la première géométrie, une condition de glissement est appliquée sur les faces horizontales alors que les faces verticales définissent des frontières amont et aval de l'écoulement. Pour ce qui concerne le profil d'aile `NACA0012`, trois triangulations non-structurées ont été utilisées (voir la table 2.3 et la figure 2.10 pour une vue partielle du maillage N1). Les maillages N2 et N3 ont été obtenus par division uniforme du maillage N1.

TAB. 2.1 – *Caractéristiques des triangulations structurées du domaine rectangulaire*

Maillage	# Nœuds	# Triangles	# Segments
RS1	4000	7562	11561
RS2	16000	31442	47441
RS3	64000	126962	190961

2.6.1.2 Domaine rectangulaire : triangulations structurées

On s'intéresse à la résolution du système linéaire issu du premier pas de temps pour plusieurs régimes d'écoulement correspondant à des nombres de Mach de référence compris entre 0.1 et 0.9 et pour des décompositions en 2 sous-domaines des maillages RS1 à RS3 de la table 2.1. Le système interface est

FIG. 2.8 – *Triangulation structurée d'un domaine rectangulaire*

FIG. 2.9 – *Triangulation non-structurée d'un domaine rectangulaire*

TAB. 2.2 – *Caractéristiques des triangulations non-structurées du domaine rectangulaire*

Maillage	# Nœuds	# Triangles	# Segments
RU1	3740	7041	10780
RU2	14520	28164	42683
RU3	57203	112656	169858

TAB. 2.3 – *Caractéristiques des triangulations non-structurées autour du profil d'aile* `NACA0012`

Maillage	# Nœuds	# Triangles	# Segments
N1	3114	6056	9170
N2	12284	24224	36508
N3	48792	96896	145688

FIG. 2.10 – *Triangulation non-structurée autour du profil d'aile* `NACA0012`

résolu par l'algorithme 2. Les résultats obtenus sont résumés sur la figure 2.11 qui visualise, pour chaque maillage, le nombre total d'itérations de Richardson pour réduire le résidu initial normalisé à la valeur seuil $\varepsilon_i = 10^{-10}$. De cette première série d'expériences numériques on conclut que la convergence observée est au moins qualitativement en accord avec la prédiction théorique (voir les équations (1.50) et (1.51)). Par ailleurs, la valeur $M = \sqrt{M_t^2 + M_n^2} = 0.6$ conduit toujours à la meilleure convergence c'est-à-dire au nombre minimal d'itérations à convergence. On notera que la valeur $u_0 = 1$ et le fait que l'on utilise une triangulation structurée du domaine rectangulaire, ne se traduisent pas ici en $M_n = M$ et $M_t = 0$ du fait de la formulation volumes finis (et plus précisément, à cause de la forme des cellules de contrôle) adoptée pour la discrétisation en espace. En d'autres termes, une comparaison plus rigoureuse entre la valeur théorique $M_n = \dfrac{1}{3}$ (voir l'équation (1.51)) et la valeur observée de M_n issue de cette série d'expériences numériques nécessiterait un calcul effectif de la valeur de M_n.

La figure 2.12 visualise la convergence du premier système linéaire pour $M = \dfrac{1}{3}$ et pour plusieurs décompositions en bandes verticales. Lorsque le nombre de sous-domaines passe de 2 à 16, le nombre d'itérations de Richardson augmente de 33 à 109 pour le maillage RS1, et de 132 à 541 pour la maillage RS3. Sur la base de cette série d'expériences numériques, il semblerait donc que le facteur de croissance du nombre d'itérations soit peut sensible à la taille du maillage sous-jacent ce qui est en accord avec le résultat de l'équation (1.51).

FIG. 2.11 – *Convergence du système interface*
Domaine rectangulaire (triangulations structurées) : décomposition en 2 sous-domaines
Axe X : nombre de Mach ($M = \sqrt{M_t^2 + M_n^2}$) - Axe Y : # itérations de Richardson

2.6.1.3 Domaine rectangulaire : triangulations non-structurées

Comme précédemment, on s'intéresse à la résolution du système linéaire issu du premier pas de temps pour plusieurs régimes d'écoulement correspondant à des nombres de Mach de référence compris entre

FIG. 2.12 – *Convergence du système interface :* $M = \sqrt{M_t^2 + M_n^2} = 0.3$
Domaine rectangulaire (triangulations structurées)
Axe X : # sous-domaines - Axe Y : # itérations de Richardson

0.1 et 0.9 et pour des décompositions en 2 sous-domaines des maillages RU1 à RU3 de la table 2.2. Les résultats obtenus sont résumés sur la figure 2.13 qui visualise, pour chaque maillage, le nombre total d'itérations de Richardson pour réduire le résidu initial normalisé à la valeur seuil $\varepsilon_i = 10^{-10}$. Cette seconde série d'expériences numériques confirme les observations faites dans le cas de triangulations structurées. La valeur $M = \sqrt{M_t^2 + M_n^2} = 0.6$ est encore celle qui conduit à la meilleure convergence. La figure 2.14 visualise la convergence du premier système linéaire pour $M = \dfrac{1}{3}$ et pour plusieurs décompositions en bandes verticales. Lorsque le nombre de sous-domaines passe de 2 à 16, le nombre d'itérations de Richardson augmente de 38 à 188 pour le maillage RU1, et de 47 à 192 pour la maillage RU3.

2.6.1.4 Écoulement autour d'un profil d'aile NACA0012

On s'intéresse à la résolution du système linéaire issu du premier pas de temps pour plusieurs régimes d'écoulement correspondant à des nombres de Mach de référence compris entre 0.1 et 0.8 et pour des décompositions en 4 sous-domaines des maillages N1 à N3 de la table 2.3. Cette fois, on utilise la méthode de GMRES pour la résolution du système interface. Les résultats obtenus sont résumés sur la figure 2.15 qui visualise, pour chaque maillage, le nombre total d'itérations de GMRES pour réduire le résidu initial normalisé à la valeur seuil $\varepsilon_i = 10^{-10}$. On observe que la valeur de M conduisant à la meilleure convergence est légèrement supérieure à 0.6. Le comportement général reste qualitativement en accord avec le résultat théorique des équations (1.50) et (1.51). Enfin, la figure 2.16 compare les convergences obtenues pour des décompositions en 4 et 8 sous-domaines du maillage N3 (pour la décomposition en 8 sous-domaines, on visualise également les résultats obtenus en utilisant l'algorithme de type Richardson). On notera que l'accélération de la convergence avec l'utilisation de la méthode GMRES est plus notable pour des faibles nombres de Mach.

FIG. 2.13 – *Convergence du système interface*
Domaine rectangulaire (triangulations non-structurées) : décomposition en 2 sous-domaines
Axe X : nombre de Mach ($M = \sqrt{M_t^2 + M_n^2}$) - Axe Y : # itérations de Richardson

FIG. 2.14 – *Convergence du système interface :* $M = \sqrt{M_t^2 + M_n^2} = \dfrac{1}{3}$
Domaine rectangulaire (triangulations non-structurées)
Axe X : # sous-domaines - Axe Y : # itérations de Richardson

FIG. 2.15 – *Convergence du système interface*
Écoulement externe autour d'un profil `NACA0012` : décomposition en 4 sous-domaines
Axe X : nombre de Mach ($M = \sqrt{M_t^2 + M_n^2}$) - Axe Y : # itérations de GMRES

FIG. 2.16 – *Convergence du système interface*
Écoulement externe autour d'un profil `NACA0012` : décomposition en 4 et 8 sous-domaines
Axe X : nombre de Mach ($M = \sqrt{M_t^2 + M_n^2}$) - Axe Y : # itérations de GMRES

2.6.2 Calcul d'écoulements stationnaires

2.6.2.1 Définition des cas tests

Dans cette section on s'intéresse à l'évaluation de la méthode de résolution par sous-domaine du système d'équations d'Euler proposée dans la section 1.1.3, dans le contexte de la simulation d'écoulements stationnaires. La géométrie sélectionnée est celle du profil d'aile `NACA0012` (voir la table 2.3 et la figure 2.10). Les situations suivantes ont été considérées :

S1 : l'écoulement subsonique caractérisé par un nombre de Mach de référence égal à 0.3 et un angle d'incidence de $0°$. Dans ce cas, l'extension au second ordre de la précision en espace (voir les équations (2.11) et (2.12)) n'utilise pas de procédure de limitation. Le pas de temps (local) est calculé à partir d'une valeur constante du nombre de Courant CFL = 1000.

S2 : l'écoulement transsonique caractérisé par un nombre de Mach de référence égal à 0.85 et un angle d'incidence de $0°$. Dans ce cas, on utilise une procédure de limitation basée sur le limiteur de Van Albada (voir par exemple Fezoui et Dervieux[FD89]) pour éviter les oscillations non-physiques dues à la présence d'un choc dans la solution. Par ailleurs, le pas de temps (local) est calculé à partir d'un nombre de Courant variable donné par la loi CFL=$5 \times k_t$ où k_t désigne l'itération en temps.

2.6.2.2 Plateformes de calcul et notations

Les expériences numériques ont été réalisées sur une grappe de 12 PC sous système LINUX (tous équipés d'un processeur `Pentium Pro` cadencé à 200 Mhz et de 128 Mb de mémoire RAM) interconnectés par deux switchs `FastEthernet` d'un débit théorique de 100 Mbit/s. La programmation parallèle repose sur l'implémentation MPICH de l'environnement MPI. Le langage de programmation est FORTRAN 77 et le compilateur G77 a été utilisé (avec options d'optimisation maximales). Les calculs sont effectués en double précision. Dans les tables ci-dessous, les notations suivantes on été adoptées :

- N_p désigne le nombre de processeurs (soit aussi le nombre de sous-domaines) utilisés pour la simulation parallèle;
- N_g est le nombre total de niveaux dans la hiérarchie multigrille (niveau fin et niveaux grossiers);
- « CPU » est le temps CPU de la simulation; plus précisément il s'agit du maximum des temps relevés sur chaque nœud de calcul;
- « REEL » est le temps réel de la simulation (le temps total écoulé). La distinction entre temps CPU et temps total écoulé est pertinente pour les simulations effectuées sur les grappes de PC comme celle considérée ici. Dans ce cas, la différence entre les valeurs « REEL » et « CPU » représente principalement les temps de communication et les temps d'inactivité liés aux défauts de répartition de la charge de calcul;
- « %CPU » est égal au rapport « CPU/REEL » et constitue une mesure de l'efficacité parallèle pour les simulations effectuées ici;
- $S(N_p)$ est l'accélération parallèle qui est dans tous les cas évaluée à partir des temps réels.

2.6.2.3 Écoulement subsonique (cas test S1)

Les lignes iso-Mach stationnaires pour ce cas test sont visualisées sur la figure 2.19. Les résultats numériques présentés ici sont caractérisés par les points suivants :

- les simulations numériques portent sur le maillage N3;

- le calcul démarre d'une situation d'écoulement uniforme;
- les résolutions locales induites par la méthode de décomposition de domaine sont toujours effectuées de façon approchée.

Le troisième des points ci-dessus est fondamental si l'on souhaite démontrer que la méthode de résolution par sous-domaine est compétitive par rapport à l'approche globale classique. En effet, on rappelle que, dans cette dernière (décrite dans la section 2.1.4), les systèmes linéaires obtenus à chaque pas de temps ne sont jamais résolus avec une grande précision (du fait de l'utilisation d'une méthode de Newton modifiée, basée sur des matrices Jacobiennes approchées). Un des objectifs de cette section de résultats numériques est donc de vérifier que les résolutions locales approchées n'influent pas ou très peu sur la convergence non-linéaire vers l'état stationnaire. On notera que la robustesse des méthodes de Krylov telle que GMRES, lorsque les produits matrice-vecteur sont calculés de façon approchée, a fait l'objet d'études récentes (voir notamment Bouras et Fraysse [BF00a] et aussi, Bouras *et al.* [BF00b] pour une discussion dans le contexte de méthodes de décomposition de domaine du type complément de Schur). Dans notre étude, ces résolutions locales approchées sont obtenues au moyen de la méthode multigrille linéaire décrite dans la section 2.4, le lisseur étant une méthode de Gauss-Seidel par point.

La figure 2.17 visualise la convergence non-linéaire vers l'état stationnaire en termes de la décroissance du résidu (normalisé à sa valeur initiale) de l'énergie totale par unité de volume. Les stratégies de résolution suivantes sont comparées :

GLOB : une stratégie de résolution globale où, à chaque pas de temps, le système linéaire (2.20) est résolu par une méthode de Jacobi (naturellement parallélisable) avec un seuil de résolution linéaire fixé à $\varepsilon_g = 10^{-1}$;

DDM1 : une stratégie de résolution par sous-domaine où, à chaque pas de temps, le système interface (2.30) est résolu par une méthode de GMRES sans ré-initialisation et non-préconditionnée, avec un seuil de résolution linéaire fixé à $\varepsilon_i = 10^{-1}$. Par ailleurs, les systèmes locaux sont résolus par application d'1 V-cycle avec $\nu_1 = 4$ pré-lissages, $\nu_2 = 4$ post-lissages et $\nu_g = 4$ lissages sur le niveau le plus grossier. Le nombre de niveaux grossiers est fixé à 3 (donc $N_g = 4$).

DDM2 : une stratégie identique à **DDM1** sauf pour la résolution des systèmes locaux qui consiste ici à appliquer 4 V-cycles avec $\nu_1 = 2$ pré-lissages, $\nu_2 = 2$ post-lissages et $\nu_g = 2$ lissages sur le niveau le plus grossier.

Les nombres effectifs de pas de temps jusqu'à l'obtention de l'état stationnaire (résidu initial de l'énergie par unité de volume divisé par un facteur 10^6) et les temps d'exécution correspondants sont résumés dans la table 2.4. Ces résultats nous conduisent à faire les premiers commentaires suivants :

- la convergence non-linéaire vers l'état stationnaire n'est pas affectée par les résolutions locales approchées. Le résultat de référence est ici fourni par la stratégie de résolution globale basée sur la méthode de relaxation de Jacobi (98 pas de temps quel que soit le nombre de sous-domaines). On a notamment pris soin de vérifier qu'une augmentation de l'effort de résolution linéaire (en diminuant le seuil de résolution ε_g de 10^{-1} à 10^{-2}) ne se traduit pas en une réduction notable du nombre de pas de temps nécessaires à l'obtention de l'état stationnaire (alors que les temps d'exécution augmentent de façon conséquente). En comparaison, la stratégie de résolution par sous-domaine basée sur 1 V-cycle ($\nu_1 = \nu_2 = \nu_g = 4$) résulte toujours en un nombre inférieur de pas de temps avec une légère dégradation lorsque l'on augmente le nombre de sous-domaines;

FIG. 2.17 – *Écoulement non-visqueux autour d'un profil d'aile* NACA0012 *: cas test* **S1**
Convergence non-linéaire (CFL = 1000, maillage N3, $N_p = 8$)
Résolution globale (Jacobi, $\varepsilon_g = 10^{-1}$) vs. résolution par sous-domaine (GMRES, $\varepsilon_i = 10^{-1}$)
Axe X : itérations en temps - Axe Y : résidu de l'énergie par unité de volume

– l'option basée sur des résolutions locales approchées par 1 V-cycle ($\nu_1 = \nu_2 = \nu_g = 4$) conduit à une stratégie de résolution par sous-domaine plus performante que la stratégie de résolution globale. On note une réduction de l'ordre de 18% du temps total de simulation lorsque $N_p = 12$. Comme on pouvait s'y attendre, la stratégie de résolution par sous-domaine démontre de meilleures efficacités parallèles du fait de la réduction des temps de communication par rapport à la stratégie de résolution globale.

2.6.2.4 Écoulement transsonique (cas test S2)

Les lignes iso-Mach stationnaires pour ce cas test sont visualisées sur la figure 2.19. Les résultats numériques présentés ici sont caractérisés par les points suivants :

– les simulations portent sur le maillage N3;
– le calcul démarre d'une situation d'écoulement uniforme;
– les résolutions locales induites par la méthode de décomposition de domaine sont toujours effectuées de façon approchée.

La figure 2.18 visualise la convergence non-linéaire vers l'état stationnaire en termes de la décroissance du résidu (normalisé à sa valeur initiale) de l'énergie totale par unité de volume. Les stratégies de résolution suivantes sont comparées :

GLOB1 : une stratégie de résolution globale où, à chaque pas de temps, le système linéaire (2.20) est résolu par une méthode de Jacobi avec un seuil de résolution linéaire fixé à $\varepsilon_g = 10^{-1}$;

TAB. 2.4 – *Écoulement non-visqueux autour d'un profil d'aile* NACA0012*: cas test* **S1**
Temps d'exécution sur une grappe de PC (CFL = 1000, maillage N3, $N_p = 8$)
Résolution globale (Jacobi, $\varepsilon_g = 10^{-1}$) vs. résolution par sous-domaine (GMRES, $\varepsilon_i = 10^{-1}$)

Méthode	N_p	# it	CPU	REEL	% CPU
GLOB	4	98	2914 sec	2995 sec	97.2
	8	98	1992 sec	2252 sec	88.5
	12	98	1071 sec	1207 sec	88.9
DDM1	4	93	2686 sec	2748 sec	97.8
	8	94	1728 sec	1802 sec	95.9
	12	96	911 sec	982 sec	92.7
DDM2	8	102	3870 sec	4144 sec	93.4

GLOB2 : une stratégie identique à **GLOB1** mais avec un seuil résolution linéaire fixé à $\varepsilon_g = 10^{-2}$;

DDM1 : une stratégie de résolution par sous-domaine où, à chaque pas de temps, le système interface (2.30) est résolu par une méthode de GMRES sans ré-initialisation et non-préconditionnée, avec un seuil de résolution linéaire fixé à $\varepsilon_i = 10^{-1}$. Par ailleurs, les systèmes locaux sont résolus par application d'1 V-cycle avec $\nu_1 = 4$ pré-lissages, $\nu_2 = 4$ post-lissages et $\nu_g = 4$ lissages sur le niveau le plus grossier. Le nombre de niveaux grossiers est fixé à 3 (donc $N_g = 4$).

DDM2 : une stratégie identique à **DDM1** mais avec un seuil de résolution linéaire fixé à $\varepsilon_i = 10^{-2}$.

Les nombres effectifs de pas de temps jusqu'à l'obtention de l'état stationnaire (résidu initial de l'énergie par unité de volume divisé par un facteur 10^6) et les temps d'exécution sont résumés dans la table 2.5. Cette seconde série de résultats confirme les observations faites pour le cas test **S1**, en particulier :

- il est clair que fixer le seuil de résolution linéaire à $\varepsilon_g = 10^{-1}$ dans la stratégie de résolution globale ne suffit pas à garantir une convergence correcte vers l'état stationnaire. En fait, la réduction ciblée du résidu de l'énergie n'a pas été atteinte après 200 pas de temps (en particulier, les temps d'exécution associés dans la table 2.5 ne sont pas vraiment exploitables puisque la convergence n'a pas été obtenue). La convergence vers l'état stationnaire a été possible en fixant le seuil de résolution linéaire à $\varepsilon_g = 10^{-2}$;
- parallèlement, réduire le seuil de résolution linéaire de $\varepsilon_i = 10^{-1}$ à $\varepsilon_i = 10^{-2}$ dans la stratégie de résolution par sous-domaine ne change rien à la convergence non-linéaire vers l'état stationnaire. Le nombre de pas de temps nécessaires à l'obtention de l'état stationnaire pour $\varepsilon_i = 10^{-1}$ est même inférieur à celui caractérisant la convergence de la stratégie de résolution globale **GLOB2** (pour laquelle $\varepsilon_g = 10^{-2}$);
- une comparaison des temps de calcul associés à la stratégie de résolution globale **GLOB2** et à la stratégie de résolution par sous-domaine **DDM1** montre que cette dernière est 3 fois plus rapide pour le calcul de l'écoulement considéré ici.

2.7 Conclusion

Dans ce chapitre, on a proposé une mise en œuvre de la méthode de décomposition de domaine introduite dans la section 1.1 dans l'optique de la résolution numérique du système d'équations d'Euler

FIG. 2.18 – *Écoulement non-visqueux autour d'un profil d'aile* `NACA0012` *: cas test* **S2**
Convergence non-linéaire (CFL $= 5 \times kt$, maillage N3, $N_p = 8$)
Résolution globale (Jacobi, $\varepsilon_g = 10^{-1}$ et $\varepsilon_g = 10^{-2}$)
vs. résolution par sous-domaine (GMRES, $\varepsilon_i = 10^{-1}$ et $\varepsilon_i = 10^{-2}$)

TAB. 2.5 – *Écoulement non-visqueux autour d'un profil d'aile* `NACA0012` *: cas test* **S2**
Temps d'exécution sur une grappe de PC (CFL $= 5 \times k_t$, maillage N3, $N_p = 8$)
Résolution globale : Jacobi, $\varepsilon_g = 10^{-1}$ et $\varepsilon_g = 10^{-2}$
Résolution par sous-domaine : GMRES, $\varepsilon_i = 10^{-1}$ et $\varepsilon_i = 10^{-2}$

Méthode	N_p	# it	CPU	REEL	% CPU
GLOB1	8	> 200	1825 sec	2017 sec	90.5
GLOB2	8	149	4524 sec	5095 sec	88.7
DDM1	8	127	1631 sec	1700 sec	96.0
DDM2	8	126	2830 sec	2951 sec	95.9

bidimensionnelles modélisant l'écoulement d'un fluide compressible parfait. Pour ce faire, on s'est appuyé sur un code de calcul existant qui combine les ingrédients suivants :

- une formulation mixte éléments finis/volumes finis en maillages non-structurés triangulaires basée sur des schémas décentrés TVD (voir notamment [Fez85], [FD89] et [FLLO89]);
- une méthode de type Euler implicite linéarisée [FS89]-[Ste88] pour l'intégration en temps des équations semi-discrétisées.

Chaque pas de temps nécessite la résolution d'un système linéaire de matrice creuse à structure irrégulière et non-symétrique. Dans la méthodologie numérique de départ, ces systèmes linéaires sont résolus de façon inexacte à l'aide d'une méthode de relaxation de Jacobi ou de Gauss-Seidel (stratégie de résolution globale). L'approche adoptée ici (stratégie de résolution par sous-domaine) consiste à résoudre ces systèmes linéaires à l'aide d'un algorithme de type Schwarz additif de la forme (1.20), introduit dans la section 1.1.3 du chapitre 1. D'un point de vue algébrique, l'algorithme de Schwarz peut être interprété comme une méthode de relaxation de Jacobi appliquée à la résolution d'un système linéaire dont la matrice a une structure par blocs particulière (voir l'équation (2.29)). Une technique de sous-structuration est alors appliquée à cette matrice afin d'obtenir une formulation de type complément de Schur. Le système interface résultant est résolu au moyen d'une méthode GMRES sans réinitialisation.

La mise en œuvre de la méthode de décomposition de domaine proposée ici est notamment caractérisée par les deux points principaux suivants :

- les inconnues aux interfaces sont les composantes de flux numériques décentrés calculés entre deux cellules de contrôle appartenant à deux sous-domaines voisins. L'idée est alors de modifier l'expression de la linéarisation du flux numérique de Roe, en introduisant une variable redondante définie à l'interface entre les deux cellules de contrôle (qui est aussi une partie de l'interface entre les deux sous-domaines voisins compte tenu de la stratégie de partitionnement adoptée), et de prendre en compte le sens de propagation de l'information pour la définition des contributions aux problèmes locaux;
- les systèmes locaux qui interviennent dans l'algorithme par décomposition de domaine sont toujours résolus de façon inexacte à l'aide d'une méthode multigrille par agglomération de volumes particulièrement bien adaptée au contexte discret à la base du code de calcul de départ. Alors que la stratégie globale utilise la méthode de relaxation de Jacobi pour une résolution parallèle des systèmes linéaires, la stratégie par sous-domaine permet d'opter pour une méthode de Gauss-Seidel par point qui joue ici le rôle de lisseur dans la méthode multigrille.

Les expériences numériques rassemblées dans ce chapitre ont permis :

- d'une part, de valider l'étude de convergence réalisée au chapitre 1.2. On a ainsi mis en évidence un comportement qualitativement en accord avec la théorie, à savoir, que pour un maillage donné, il existe une valeur du nombre de Mach de référence (et donc du Mach normal à l'interface) pour laquelle la convergence est optimale et cette valeur est indépendante du pas d'espace;
- d'autre part, d'évaluer la stratégie par sous-domaine dans le cadre de la résolution d'écoulements stationnaires autour d'un profil d'aile `NACA0012`. On a notamment vérifié que l'option de résolution approchée des systèmes locaux ne dégrade pas la convergence linéaire (résolution du système linéaire pour un pas de temps donné) et non-linéaire (convergence vers l'état stationnaire).

Pour ce qui concerne le deuxième des points ci-dessus, la table 2.5 montre que les nombres de pas de temps nécessaires au calcul de l'écoulement stationnaire qui caractérisent les stratégies de résolution par sous-domaine, sont sensiblement inférieurs à ceux résultant des stratégies de résolution globales. Ce comportement s'explique ici par le fait que, pour chaque pas de temps implicite et pour les seuils de résolution linéaires considérés ($\varepsilon_g = 10^{-1}$ et $\varepsilon_g = 10^{-2}$ pour les stratégies de résolution globales, et $\varepsilon_i = 10^{-1}$ et $\varepsilon_i = 10^{-2}$ pour les stratégies de résolution par sous-domaine), les solutions locales obtenues par la méthode multigrille contribuent à une solution globalement plus convergée que celle fournie par la méthode de relaxation de Jacobi, résultant du traitement d'un spectre plus large de l'erreur itérative.

FIG. 2.19 – *Lignes iso-Mach stationnaires pour le cas test subsonique (haut) et transsonique (bas)*

Chapitre 3

Evaluation d'une méthode de préconditionnement algébrique

Dans ce chapitre, on s'intéresse à la mise au point d'une technique de préconditionnement algébrique simple pour l'accélération de la résolution itérative du système interface (2.30) qui caractérise la méthode de décomposition de domaine développée au chapitre 2 pour la résolution du système d'équations d'Euler. Le caractère algébrique signifie que le préconditionneur n'est construit qu'à partir de la seule connaissance de la structure de l'opérateur interface. Ici, compte tenu des propriétés de l'opérateur interface, on opte pour un préconditionneur de type polynomial. On propose une mise en œuvre particulière qui repose sur la méthode multigrille par agglomération de volumes présentée dans la section 2.4 et qui vise à réduire le coût d'application de ce préconditionneur. Ce chapitre est essentiellement consacré à une évaluation numérique de la méthode de décomposition de domaine résultante.

3.1 Expression du préconditionneur

Dans la section 2.3, on a montré que la formulation de l'algorithme de type Schwarz additif sans recouvrement, dans le contexte des méthodes de discrétisation en espace et d'intégration en temps respectivement décrites aux sections 2.1.2 et 2.1.4, conduit à la résolution d'un système global de la forme :

$$Mx = b \quad \text{avec} \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_{N_{\mathsf{sd}}} \\ \Phi \end{pmatrix} \quad \text{et} \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_{N_{\mathsf{sd}}} \\ 0 \end{pmatrix} \tag{3.1}$$

qui, par application d'une technique de sous-structuration, se ramène à la résolution du système interface :

$$S\Phi = g \quad \text{avec} \quad S = \mathsf{Id} - \sum_{k=1}^{N_{\mathsf{sd}}} \mathcal{F}_k \mathcal{M}_k^{-1} \mathcal{M}_{k\Phi} = \mathsf{Id} - A \quad \text{et} \quad g = -\sum_{k=1}^{N_{\mathsf{sd}}} \mathcal{F}_k \mathcal{M}_k^{-1} b_k \tag{3.2}$$

Dans (3.1), x_k pour $k = 1, \cdots, N_{\mathsf{sd}}$ désigne un vecteur de $4 \times N_k$ éléments où N_k est le nombre de nœuds internes au sous-domaine k, alors que dans (3.2), Φ est une vecteur de $4 \times N_i$ éléments où N_i est le nombre total de segments interface, c'est-à-dire de segments dont les nœuds extrémités appartiennent à des sous-domaines différents (mais voisins). Ici, on se place dans le cas où le système (3.2) est résolu par une méthode GMRES sans ré-initialisation de la base de Krylov (méthode « GMRES complète »).

Le choix du préconditionneur tient compte du fait que la matrice A résulte de la discrétisation d'un opérateur contractant (par conséquent, le rayon spectral de A est inférieur à 1); ce point est discuté dans la section 1.1.4 (voir notamment la remarque 2). On choisi donc d'utiliser un préconditionneur polynomial qui consiste à calculer l'inverse de l'opérateur interface par :

$$(\mathrm{Id} - A)^{-1} = \mathrm{Id} + \sum_{l=1}^{p} A^p \tag{3.3}$$

On obtient par exemple un préconditionneur polynomial du premier ordre par l'expression :

$$P_1 \equiv \mathrm{Id} + A \approx S^{-1} \tag{3.4}$$

et des préconditionneurs d'ordres plus élevés, et donc de meilleures approximations de l'inverse de S, en tenant compte des termes suivants dans la série (3.3). On adopte la notation générique P_p dans le cas où l'on tient compte de p termes dans (3.3).

3.2 Préconditionneur polynomial hiérarchique

L'application à gauche du préconditionneur (3.4) au système interface (3.2) conduit à :

$$P_1 S\Phi = P_1 g \quad \Leftrightarrow \quad (\mathrm{Id} - A^2)\Phi = g' \tag{3.5}$$

ce qui montre que l'on a quasiment doublé la complexité d'une itération de la méthode GMRES si l'on considère que le calcul du produit de la matrice A par le vecteur Φ domine largement le coût d'une itération. Par conséquent, on ne diminuera le temps total de résolution que si l'application de ce type de préconditionneur résulte en une réduction de plus d'un facteur 2 du nombre d'itérations de la méthode GMRES nécessaires à la résolution du système (3.2) à un seuil donné. Pour pallier ce problème d'efficacité, nous proposons et nous évaluons ici une solution qui repose sur une approche hiérarchique pour l'évaluation du préconditionneur P_p . Plus précisément, (3.3) montre que l'application du préconditionneur fait intervenir la résolution de problèmes locaux à travers le calcul des produits matrice-vecteur avec la matrice A. L'idée est alors de différencier la complexité de la résolution (et donc la précision du calcul des inverses $\mathcal{M}_k^{-1}$ dans (3.2)) suivant que l'on évalue le produit Sx ou le produit $P_p x$. En pratique, on utilise la méthode multigrille par agglomération de volumes décrite dans la section 2.4 pour la résolution approchée des problèmes locaux induits par l'application de S et P_p à un vecteur x. On obtient ainsi un préconditionneur polynomial que l'on qualifie de hiérarchique et que l'on note $P_{p-\mathrm{mgV}}(\nu_1,\nu_2,\nu_g,N_c)$ pour signifier que l'application du préconditionneur consiste en N_c V-cyles de caractéristiques ν_1 pré-lissages, ν_2 post-lissages et ν_g lissages sur le niveau le plus grossier (le lisseur est une méthode de Gauss-Seidel par point).

3.3 Résultats numériques

3.3.1 Définition du cas test

La géométrie sélectionnée est celle du profil d'aile `NACA0012` (voir la table 2.3 et la figure 2.10). On considère de nouveau les situations suivantes :

S1 : l'écoulement subsonique caractérisé par un nombre de Mach de référence égal à 0.3 et un angle d'incidence de $0°$. Dans ce cas, l'extension au second ordre de la précision en espace (voir les

équations (2.11) et (2.12)) n'utilise pas de procédure de limitation. Le pas de temps (local) est calculé à partir d'une valeur constante du nombre de Courant fixée à 1000.

S2 : l'écoulement transsonique caractérisé par un nombre de Mach de référence égal à 0.85 et un angle d'incidence de $0°$. Dans ce cas, on utilise une procédure de limitation pour éviter les oscillations non-physiques dues à la présence d'un choc dans la solution. Le limiteur de Van Albada (voir par exemple Fezoui et Dervieux[FD89]) a été adopté ici. Par ailleurs, le pas de temps (local) est calculé à partir d'un nombre de Courant variable donné par la loi CFL=$5 \times k_t$ où k_t désigne l'itération en temps.

3.3.2 Plate-formes de calcul

Les expériences numériques ont été réalisées sur une grappe de 14 PC sous système LINUX (tous équipés de cartes mères comprenant 2 processeurs `Pentium Pro` cadencés à 500 Mhz et 512 Mb de mémoire RAM partagée par les deux processeurs) inter-connectés par deux switchs `FastEthernet` d'un débit théorique de 100 Mbit/s. La programmation parallèle repose sur l'implémentation MPICH de l'environnement MPI. Le langage de programmation est FORTRAN 77 et le compilateur G77 a été utilisé (avec options d'optimisation maximales). Les calculs sont effectués en double précision. Dans les tables ci-dessous, les notations sont celles déjà adoptées dans le cadre de la résolution du système d'équations d'Euler (voir la section 2.6.2.2).

3.3.3 Étude sur un système linéaire

On considère ici la résolution du système linéaire issu du premier pas de temps. On se limite ici à l'étude du cas test **S1** (écoulement subsonique). Afin d'isoler le rôle du préconditionneur, on résout quasi-exactement (seuil de résolution linéaire fixé à $\varepsilon = 10^{-10}$) les systèmes linéaires induits par l'application de l'opérateur S et on compare différentes stratégies de préconditionnement à gauche. Les résultats obtenus sont résumés dans les tables 3.1 (maillage N1), 3.2 (maillage N2) et 3.3 (maillage N3). Dans ces tables, « N.A. » fait référence aux mesures sans application du préconditionneur dans la résolution du système interface.

Les calculs effectués avec le maillage N1 montrent une sensible amélioration, à la fois en termes du nombre d'itérations de la méthode GMRES et du temps total d'exécution, avec l'utilisation du préconditionneur polynomial proposé ici. On constate cependant que réaliser de $N_c = 2$ V-cycles dans l'application du préconditionneur polynomial du premier ordre ne se traduit pas en une diminution notable du nombre d'itérations. Par contre, le préconditionneur polynomial du second ordre basé sur $N_c = 1$ V-cycle permet une réduction d'un facteur de l'ordre de 2.3 du nombre d'itérations par rapport aux calculs sans préconditionneur; parallèlement, les temps d'exécution sont réduits d'un facteur proche de 1.9 dans chaque cas de valeur du nombre de processeurs (ou encore du nombre de sous-domaines) N_p. Les calculs réalisés avec le maillage N2 donnent des résultats comparables. En particulier, ceux-ci confirment que le préconditionneur $P_{1-\mathsf{mgV}}(2,2,2,2)$ n'est pas supérieur au préconditionneur $P_{1-\mathsf{mgV}}(2,2,2,1)$. Sur ce maillage, les gains en nombres d'itérations liés au préconditionneur $P_{2-\mathsf{mgV}}(2,2,2,1)$ se traduisent en des gains du même ordre en temps d'exécution. Enfin, les calculs effectués avec le maillage N3 permettent de faire quelques remarques supplémentaires. On note tout d'abord que lorsque le nombre de sous-domaines passe de $N_p = 4$ à $N_p = 24$, le nombre d'itérations de la méthode GMRES non-préconditionnée augmente de 57 à 98 (soit une variation de 72%); l'utilisation du préconditionneur $P_{1-\mathsf{mgV}}(2,2,2,1)$ fait que le nombre d'itérations varie de 36 à 57 (soit une variation de 58%) alors que l'application du préconditionneur $P_{2-\mathsf{mgV}}(2,2,2,1)$ est caractérisée par une variation de 65% (le nombre d'itérations passe de 23 à 38). Les variations correspondantes pour les calculs réalisés avec le maillage N2 sont respectivement

égales à 52%, 61% et 48%. Il semble donc difficile, à ce stade de la discussion, de tirer des conclusions quant aux propriétés de scalabilité de la méthode de décomposition de domaine développée dans cette étude (avec ou sans préconditionneur) si ce n'est que, d'une façon générale, celle-ci est peu sensible à l'augmentation du nombre de sous-domaines (pour le maillage N3, lorsque ce dernier augmente d'un facteur 6, l'augmentation du nombre d'itérations n'excède pas 1.8).

TAB. 3.1 – *Influence du préconditionneur* $P_{p-mgV}(\nu_1,\nu_2,\nu_g,N_c)$
Maillage N1 : convergence du premier système linéaire

Préconditionneur	N_p	N_g	# iter GMRES	CPU	REEL
N.A.	4	3	42	22 s	23 s
$P_{1-\mathrm{mgV}}(2,2,2,1)$	4	3	24	14 s	15 s
$P_{1-\mathrm{mgV}}(2,2,2,2)$	4	3	24	14 s	15 s
$P_{2-\mathrm{mgV}}(2,2,2,1)$	4	3	18	11 s	12 s
N.A.	8	3	56	12 s	13 s
$P_{1-\mathrm{mgV}}(2,2,2,1)$	8	3	32	13 s	14 s
$P_{1-\mathrm{mgV}}(2,2,2,2)$	8	3	30	13 s	14 s
$P_{2-\mathrm{mgV}}(2,2,2,1)$	8	3	23	6 s	7 s

TAB. 3.2 – *Influence du préconditionneur* $P_{p-mgV}(\nu_1,\nu_2,\nu_g,N_c)$
Maillage N2 : convergence du premier système linéaire

Préconditionneur	N_p	N_g	# iter GMRES	CPU	REEL
N.A.	4	4	50	194 s	203 s
$P_{1-\mathrm{mgV}}(2,2,2,1)$	4	4	28	119 s	121 s
$P_{1-\mathrm{mgV}}(2,2,2,2)$	4	4	28	114 s	116 s
$P_{2-\mathrm{mgV}}(2,2,2,1)$	4	4	21	93 s	95 s
N.A.	8	4	66	114 s	117 s
$P_{1-\mathrm{mgV}}(2,2,2,1)$	8	4	37	68 s	70 s
$P_{1-\mathrm{mgV}}(2,2,2,2)$	8	4	36	69 s	71 s
$P_{2-\mathrm{mgV}}(2,2,2,1)$	8	4	27	53 s	54 s
N.A.	16	4	76	51 s	69 s
$P_{1-\mathrm{mgV}}(2,2,2,1)$	16	4	45	31 s	45 s
$P_{1-\mathrm{mgV}}(2,2,2,2)$	16	4	42	30 s	42 s
$P_{2-\mathrm{mgV}}(2,2,2,1)$	16	4	31	24 s	34 s

3.3.4 Calcul de l'écoulement complet

Nous considérons maintenant le calcul de l'écoulement stationnaire complet. On considère dans un premier temps les cas test **S1**. Cette fois, le principe de la résolution approchée des systèmes locaux est adopté dans l'application de S (opérateur interface) comme dans celle de $P_{p-\mathrm{mgV}}(\nu_1,\nu_2,\nu_g,N_c)$ (préconditionneur du système interface), à l'instar de ce qui avait été fait dans la section 2.6.2. On réalise différents calculs qui se différencient notamment par les valeurs choisies des seuils de résolution linéaire des systèmes locaux (ε_l) et interface (ε_i) dans l'application de l'opérateur S. Les tables 3.4 à

TAB. 3.3 – *Influence du préconditionneur* $P_{p-mgV}(\nu_1,\nu_2,\nu_g,N_c)$
Maillage N3 : convergence du premier système linéaire

Préconditionneur	N_p	N_g	# iter GMRES	CPU	REEL
N.A.	4	5	57	1238 s	1246 s
$P_{1-\mathrm{mgV}}(2,2,2,1)$	4	5	36	815 s	822 s
$P_{1-\mathrm{mgV}}(2,2,2,2)$	4	5	31	730 s	737 s
$P_{2-\mathrm{mgV}}(2,2,2,1)$	4	5	23	539 s	546 s
N.A.	8	5	73	717 s	725 s
$P_{1-\mathrm{mgV}}(2,2,2,1)$	8	5	45	464 s	469 s
$P_{1-\mathrm{mgV}}(2,2,2,2)$	8	5	39	417 s	424 s
$P_{2-\mathrm{mgV}}(2,2,2,1)$	8	5	28	306 s	310 s
N.A.	16	5	82	318 s	391 s
$P_{1-\mathrm{mgV}}(2,2,2,1)$	16	5	49	202 s	250 s
$P_{1-\mathrm{mgV}}(2,2,2,2)$	16	5	44	190 s	233 s
$P_{2-\mathrm{mgV}}(2,2,2,1)$	16	5	33	147 s	186 s
N.A.	24	5	98	238 s	246 s
$P_{1-\mathrm{mgV}}(2,2,2,1)$	24	5	57	152 s	156 s
$P_{1-\mathrm{mgV}}(2,2,2,2)$	24	5	52	139 s	145 s
$P_{2-\mathrm{mgV}}(2,2,2,1)$	24	5	38	106 s	111 s

3.6 résument les résultats de calculs effectués avec le maillage N3 pour des décompositions en $N_p = 7$, $N_p = 14$ et $N_p = 28$ sous-domaines. Dans ces tables, « # iter » désigne le nombre de pas de temps pour atteindre la solution stationnaire. Dans tous les cas, on a choisi de fixer $\varepsilon_i = \varepsilon_l$. Pour compléter ces mesures, on a représenté sur les figures 3.1 et 3.2 l'évolution du nombre effectif d'itérations de la méthode GMRES en fonction de l'itération en temps, pour une décomposition en $N_p = 14$ sous-domaines.

On constate que l'utilisation du préconditionneur $P_{p-\mathrm{mgV}}(\nu_1,\nu_2,\nu_g,N_c)$ est sans effet positif sur les temps d'exécution pour $\varepsilon_i = \varepsilon_l = 10^{-1}$. Le rôle du préconditionneur est d'autant plus important que l'on réduit les seuils de résolution linéaire. Ainsi, dans le cas $\varepsilon_i = \varepsilon_l = 10^{-2}$, les gains en termes du temps total d'exécution, entre un calcul sans préconditionneur et le même calcul réalisé en appliquant le préconditionneur $P_{2-\mathrm{mgV}}(2,2,2,1)$, sont respectivement de 16% pour $N_p = 7$, 24% pour $N_p = 14$ et 20% pour $N_p = 28$. Pour cette valeur des paramètres ε_l et ε_i, on réalise le même nombre de pas de temps (98) pour toutes les configurations testées. On note cependant que, bien que les nombres de pas de temps effectués pour $\varepsilon_i = \varepsilon_l = 10^{-1}$ sont sensiblement supérieurs dans certains cas (entre 98 et 105), les temps d'exécution correspondant sont toujours largement inférieurs à ceux qui caractérisent $\varepsilon_i = \varepsilon_l = 10^{-2}$. Pour $\varepsilon_i = \varepsilon_l = 10^{-3}$, le préconditionneur $P_{2-\mathrm{mgV}}(2,2,2,1)$ conduit à une réduction de 40% du temps total d'exécution pour $N_p = 7$ et de 37% pour $N_p = 28$.

Afin de préciser l'influence du préconditionneur $P_{p-\mathrm{mgV}}(\nu_1,\nu_2,\nu_g,N_c)$ sur le comportement de la méthode de décomposition de domaine lorsque le nombre de sous-domaines augmente, la table 3.7 résume les extrema du nombre d'itérations de la méthode GMRES au cours du calcul de l'écoulement stationnaire dans le cas $\varepsilon_i = \varepsilon_l = 10^{-3}$. Cette table est complétée par les figures 3.3 et 3.4; sur chacune de ces dernières, on a représenté l'évolution du nombre effectif d'itérations de la méthode GMRES en fonction de l'itération en temps, pour différentes valeurs du nombre de sous-domaines N_p. On observe tout d'abord

que le préconditionneur $P_{p-\mathrm{mgV}}(\nu_1,\nu_2,\nu_g,N_c)$ (notamment le préconditionneur $P_{2-\mathrm{mgV}}(2,2,2,1)$) a une sorte d'effet stabilisateur sur la variation du nombre d'itérations de la méthode GMRES au cours de la simulation. Par exemple, pour $N_p = 7$, la variation maximale dans le cas non-préconditionné atteint 60% alors qu'elle est de 44% lorsqu'on adopte le préconditionneur $P_{2-\mathrm{mgV}}(2,2,2,1)$. D'une façon générale, on ne peut que noter une faible dépendance de la méthode de décomposition de domaine développée dans cette étude, vis-à-vis du nombre de sous-domaines. Ce commentaire, qui est valable avec ou sans application du préconditionneur, doit néanmoins être modéré par le fait que les calculs réalisés ici portent sur des nombres de sous-domaines $N_p \leq 28$ ce qui n'est peut-être pas suffisant pour juger convenablement des propriétés de scalabilité de la méthode.

TAB. 3.4 – *Influence du préconditionneur* $P_{p-mgV}(\nu_1,\nu_2,\nu_g,N_c)$
Calcul d'un écoulement complet (cas test **S1**) : maillage N3 et $N_p = 7$
Résolutions locales : V-cycle avec $\nu_1 = \nu_2 = \nu_g = 2$

Préconditionneur	N_g	ε_i	ε_l	# iter	CPU	REEL	% CPU
N.A.	5	10^{-1}	10^{-1}	98	826 s	867 s	95.2
$P_{1-\mathrm{mgV}}(2,2,2,1)$	5	10^{-1}	10^{-1}	98	820 s	854 s	96.0
$P_{1-\mathrm{mgV}}(2,2,2,2)$	5	10^{-1}	10^{-1}	98	966 s	1009 s	95.7
$P_{2-\mathrm{mgV}}(2,2,2,1)$	5	10^{-1}	10^{-1}	98	924 s	979 s	94.4
N.A.	5	10^{-2}	10^{-2}	98	2810 s	2927 s	96.0
$P_{1-\mathrm{mgV}}(2,2,2,1)$	5	10^{-2}	10^{-2}	98	2683 s	2785 s	96.3
$P_{1-\mathrm{mgV}}(2,2,2,2)$	5	10^{-2}	10^{-2}	98	2842 s	2909 s	97.7
$P_{2-\mathrm{mgV}}(2,2,2,1)$	5	10^{-2}	10^{-2}	98	2354 s	2457 s	95.8
N.A.	5	10^{-3}	10^{-3}	98	7866 s	8088 s	97.2
$P_{1-\mathrm{mgV}}(2,2,2,1)$	5	10^{-3}	10^{-3}	98	6569 s	6737 s	97.5
$P_{1-\mathrm{mgV}}(2,2,2,2)$	5	10^{-3}	10^{-3}	98	5527 s	5880 s	93.4
$P_{2-\mathrm{mgV}}(2,2,2,1)$	5	10^{-3}	10^{-3}	98	4602 s	4773 s	96.4

L'ensemble des calculs précédents ont aussi été effectués pour le cas test **S2** (écoulement transsonique). Les tables 3.8 à 3.10 résument les résultats de calculs effectués avec le maillage N3 pour des décompositions en $N_p = 7$, $N_p = 14$ et $N_p = 28$ sous-domaines. Pour compléter ces mesures, on a représenté sur les figures 3.5 et 3.6 l'évolution du nombre effectif d'itérations de la méthode GMRES en fonction de l'itération en temps, pour une décomposition en $N_p = 14$ sous-domaines. Contrairement au cas test **S1**, on note cette fois que des gains sont obtenus même lorsque les seuils de résolution linéaire sont fixés à $\varepsilon_i = \varepsilon_l = 10^{-1}$. Dans ce cas, le préconditionneur $P_{1-\mathrm{mgV}}(2,2,2,1)$ fournit les meilleurs résultats. Lorsque $\varepsilon_i = \varepsilon_l = 10^{-2}$, les gains en termes du temps total d'exécution, entre un calcul sans préconditionneur et le même calcul réalisé en appliquant le préconditionneur $P_{2-\mathrm{mgV}}(2,2,2,1)$, sont respectivement de 23% pour $N_p = 7$, 26% pour $N_p = 14$ et 18% pour $N_p = 28$, soit des gains du même ordre que ceux obtenus pour le cas test **S1**. Pour $\varepsilon_i = \varepsilon_l = 10^{-3}$, le préconditionneur $P_{2-\mathrm{mgV}}(2,2,2,1)$ conduit à une réduction de 41% du temps total d'exécution pour $N_p = 7$ et de 33% pour $N_p = 28$.

L'étude de scalabilité (table 3.11 et figures 3.7 à 3.8) conduit à des observations similaires à celles faites pour le cas test **S1**. On note cependant des variations plus importantes du nombre effectif d'itérations de la méthode GMRES au cours de la simulation; cette remarque est aussi valable lorsqu'on applique le préconditionneur $P_{p-\mathrm{mgV}}(\nu_1,\nu_2,\nu_g,N_c)$.

TAB. 3.5 – *Influence du préconditionneur* $P_{p-mgV}(\nu_1,\nu_2,\nu_g,N_c)$
Calcul d'un écoulement complet (cas test **S1**) : maillage N3 et $N_p = 14$
Résolutions locales : V-cycle avec $\nu_1 = \nu_2 = \nu_g = 2$

Préconditionneur	N_g	ε_i	ε_l	# iter	CPU	REEL	% CPU
N.A.	5	10^{-1}	10^{-1}	104	461 s	486 s	94.8
$P_{1-\mathrm{mgV}}(2,2,2,1)$	5	10^{-1}	10^{-1}	105	524 s	559 s	93.7
$P_{1-\mathrm{mgV}}(2,2,2,2)$	5	10^{-1}	10^{-1}	101	510 s	538 s	94.8
$P_{2-\mathrm{mgV}}(2,2,2,1)$	5	10^{-1}	10^{-1}	99	503 s	534 s	94.2
N.A.	5	10^{-2}	10^{-2}	98	1626 s	1726 s	94.2
$P_{1-\mathrm{mgV}}(2,2,2,1)$	5	10^{-2}	10^{-2}	98	1580 s	1719 s	91.9
$P_{1-\mathrm{mgV}}(2,2,2,2)$	5	10^{-2}	10^{-2}	98	1484 s	1593 s	93.1
$P_{2-\mathrm{mgV}}(2,2,2,1)$	5	10^{-2}	10^{-2}	98	1237 s	1303 s	94.9
N.A.	5	10^{-3}	10^{-3}	98	3754 s	3986 s	94.2
$P_{1-\mathrm{mgV}}(2,2,2,1)$	5	10^{-3}	10^{-3}	98	3062 s	3265 s	93.8
$P_{1-\mathrm{mgV}}(2,2,2,2)$	5	10^{-3}	10^{-3}	98	2861 s	2998 s	95.4
$P_{2-\mathrm{mgV}}(2,2,2,1)$	5	10^{-3}	10^{-3}	98	2150 s	2304 s	93.3

TAB. 3.6 – *Influence du préconditionneur* $P_{p-mgV}(\nu_1,\nu_2,\nu_g,N_c)$
Calcul d'un écoulement complet (cas test **S1**) : maillage N3 et $N_p = 28$
Résolutions locales : V-cycle avec $\nu_1 = \nu_2 = \nu_g = 2$

Préconditionneur	N_g	ε_i	ε_l	# iter	CPU	REEL	% CPU
N.A.	5	10^{-1}	10^{-1}	103	220 s	423 s	52.0
$P_{1-\mathrm{mgV}}(2,2,2,1)$	5	10^{-1}	10^{-1}	100	227 s	457 s	49.7
$P_{1-\mathrm{mgV}}(2,2,2,2)$	5	10^{-1}	10^{-1}	100	245 s	491 s	49.9
$P_{2-\mathrm{mgV}}(2,2,2,1)$	5	10^{-1}	10^{-1}	99	240 s	461 s	52.0
N.A.	5	10^{-2}	10^{-2}	98	769 s	1337 s	57.5
$P_{1-\mathrm{mgV}}(2,2,2,1)$	5	10^{-2}	10^{-2}	98	652 s	1115 s	58.5
$P_{1-\mathrm{mgV}}(2,2,2,2)$	5	10^{-2}	10^{-2}	98	758 s	1243 s	61.0
$P_{2-\mathrm{mgV}}(2,2,2,1)$	5	10^{-2}	10^{-2}	98	632 s	1064 s	59.4
N.A.	5	10^{-3}	10^{-3}	98	1977 s	2774 s	71.2
$P_{1-\mathrm{mgV}}(2,2,2,1)$	5	10^{-3}	10^{-3}	98	1312 s	1977 s	66.3
$P_{1-\mathrm{mgV}}(2,2,2,2)$	5	10^{-3}	10^{-3}	98	1441 s	2217 s	64.5
$P_{2-\mathrm{mgV}}(2,2,2,1)$	5	10^{-3}	10^{-3}	98	1112 s	1764 s	63.0

FIG. 3.1 – *Calcul d'un écoulement complet (cas test* **S1***) : maillage N3 et* $N_p = 14$

Influence du préconditionneur $P_{p-\mathrm{mgV}}(\nu_1,\nu_2,\nu_g,N_c)$: $\varepsilon_i = \varepsilon_l = 10^{-1}$ (haut) - $\varepsilon_i = \varepsilon_l = 10^{-2}$ (bas)

Résolutions locales : V-cycle avec $\nu_1 = \nu_2 = \nu_g = 2$

Prec 1 : $P_{1-\mathrm{mgV}}(2,2,2,1)$ - Prec 2 : $P_{1-\mathrm{mgV}}(2,2,2,2)$ - Prec 3 : $P_{2-\mathrm{mgV}}(2,2,2,1)$

Axe X : itérations en temps - Axe Y : # itérations de GMRES

FIG. 3.2 – *Calcul d'un écoulement complet (cas test* **S1***) : maillage N3 et* $N_p = 14$
Influence du préconditionneur $P_{p-\mathrm{mgV}}(\nu_1,\nu_2,\nu_g,N_c)$ - $\varepsilon_i = \varepsilon_l = 10^{-3}$
Résolutions locales : V-cycle avec $\nu_1 = \nu_2 = \nu_g = 2$
Prec 1 : $P_{1-\mathrm{mgV}}(2,2,2,1)$ - Prec 2 : $P_{1-\mathrm{mgV}}(2,2,2,2)$ - Prec 3 : $P_{2-\mathrm{mgV}}(2,2,2,1)$
Axe X : itérations en temps - Axe Y : # itérations de GMRES

TAB. 3.7 – *Influence du préconditionneur* $P_{p-mgV}(\nu_1,\nu_2,\nu_g,N_c)$
Calcul d'un écoulement complet (cas test **S1**) : maillage N3, étude de scalabilité
$\varepsilon_i = \varepsilon_l = 10^{-3}$ - Résolutions locales : V-cycle avec $\nu_1 = \nu_2 = \nu_g = 2$

Préconditionneur	N_p	# iter GMRES min	# iter GMRES max
N.A.	7	20	32
$P_{1-\mathrm{mgV}}(2,2,2,1)$	7	14	22
$P_{1-\mathrm{mgV}}(2,2,2,2)$	7	12	18
$P_{2-\mathrm{mgV}}(2,2,2,1)$	7	9	13
N.A.	14	21	28
$P_{1-\mathrm{mgV}}(2,2,2,1)$	14	13	19
$P_{1-\mathrm{mgV}}(2,2,2,2)$	14	12	16
$P_{2-\mathrm{mgV}}(2,2,2,1)$	14	9	12
N.A.	28	24	33
$P_{1-\mathrm{mgV}}(2,2,2,1)$	28	15	19
$P_{1-\mathrm{mgV}}(2,2,2,2)$	28	14	18
$P_{2-\mathrm{mgV}}(2,2,2,1)$	28	11	14

FIG. 3.3 – *Calcul d'un écoulement complet (cas test* **S1***) : étude de scalabilité, maillage N3*
GMRES non-préconditionné (haut) et préconditionné par $P_{1-\mathrm{mgV}}(2,2,2,1)$ (bas)

$\varepsilon_i = \varepsilon_l = 10^{-3}$ - Résolutions locales : V-cycle avec $\nu_1 = \nu_2 = \nu_g = 2$

Axe X : itérations en temps - Axe Y : # itérations de GMRES

FIG. 3.4 – *Calcul d'un écoulement complet (cas test* **S1***) : étude de scalabilité, maillage N3*
GMRES préconditionné par $P_{1-\mathrm{mgV}}(2,2,2,2)$ (haut) et $P_{2-\mathrm{mgV}}(2,2,2,1)$ (bas)
$\varepsilon_i = \varepsilon_l = 10^{-3}$ - Résolutions locales : V-cycle avec $\nu_1 = \nu_2 = \nu_g = 2$
Axe X : itérations en temps - Axe Y : # itérations de GMRES

TAB. 3.8 – *Influence du préconditionneur* $P_{p-mgV}(\nu_1,\nu_2,\nu_g,N_c)$
Calcul d'un écoulement complet (cas test **S2**) : maillage N3 et $N_p = 7$
Résolutions locales : V-cycle avec $\nu_1 = \nu_2 = \nu_g = 4$

Préconditionneur	N_g	ε_i	ε_l	# iter	CPU	REEL	% CPU
N.A.	5	10^{-1}	10^{-1}	112	1065 s	1156 s	92.1
$P_{1-\mathrm{mgV}}(2,2,2,1)$	5	10^{-1}	10^{-1}	100	856 s	897 s	95.4
$P_{1-\mathrm{mgV}}(2,2,2,2)$	5	10^{-1}	10^{-1}	97	946 s	1011 s	93.6
$P_{2-\mathrm{mgV}}(2,2,2,1)$	5	10^{-1}	10^{-1}	106	942 s	1028 s	91.6
N.A.	5	10^{-2}	10^{-2}	97	2395 s	2507 s	95.5
$P_{1-\mathrm{mgV}}(2,2,2,1)$	5	10^{-2}	10^{-2}	94	1863 s	1967 s	94.7
$P_{1-\mathrm{mgV}}(2,2,2,2)$	5	10^{-2}	10^{-2}	98	2195 s	2289 s	95.9
$P_{2-\mathrm{mgV}}(2,2,2,1)$	5	10^{-2}	10^{-2}	97	1841 s	1904 s	96.7
N.A.	5	10^{-3}	10^{-3}	94	4833 s	5071 s	95.3
$P_{1-\mathrm{mgV}}(2,2,2,1)$	5	10^{-3}	10^{-3}	93	3370 s	3476 s	97.0
$P_{1-\mathrm{mgV}}(2,2,2,2)$	5	10^{-3}	10^{-3}	93	3712 s	3834 s	96.7
$P_{2-\mathrm{mgV}}(2,2,2,1)$	5	10^{-3}	10^{-3}	91	2884 s	3011 s	95.8

TAB. 3.9 – *Influence du préconditionneur* $P_{p-mgV}(\nu_1,\nu_2,\nu_g,N_c)$
Calcul d'un écoulement complet (cas test **S2**) : maillage N3 et $N_p = 14$
Résolutions locales : V-cycle avec $\nu_1 = \nu_2 = \nu_g = 4$

Préconditionneur	N_g	ε_i	ε_l	# iter	CPU	REEL	% CPU
N.A.	4	10^{-1}	10^{-1}	116	502 s	554 s	90.6
$P_{1-\mathrm{mgV}}(2,2,2,1)$	4	10^{-1}	10^{-1}	112	458 s	497 s	92.1
$P_{1-\mathrm{mgV}}(2,2,2,2)$	4	10^{-1}	10^{-1}	112	518 s	572 s	90.5
$P_{2-\mathrm{mgV}}(2,2,2,1)$	4	10^{-1}	10^{-1}	104	468 s	513 s	91.2
N.A.	4	10^{-2}	10^{-2}	105	1237 s	1294 s	95.5
$P_{1-\mathrm{mgV}}(2,2,2,1)$	4	10^{-2}	10^{-2}	101	1019 s	1077 s	94.6
$P_{1-\mathrm{mgV}}(2,2,2,2)$	4	10^{-2}	10^{-2}	102	1082 s	1147 s	94.3
$P_{2-\mathrm{mgV}}(2,2,2,1)$	4	10^{-2}	10^{-2}	101	908 s	953 s	95.3
N.A.	4	10^{-3}	10^{-3}	89	2337 s	2473 s	94.5
$P_{1-\mathrm{mgV}}(2,2,2,1)$	4	10^{-3}	10^{-3}	89	1595 s	1691 s	94.3
$P_{1-\mathrm{mgV}}(2,2,2,2)$	4	10^{-3}	10^{-3}	89	1749 s	1865 s	93.8
$P_{2-\mathrm{mgV}}(2,2,2,1)$	4	10^{-3}	10^{-3}	89	1407 s	1482 s	95.0

TAB. 3.10 – *Influence du préconditionneur* $P_{p-mgV}(\nu_1,\nu_2,\nu_g,N_c)$
Calcul d'un écoulement complet (cas test **S2**) : maillage N3 et $N_p = 28$
Résolutions locales : V-cycle avec $\nu_1 = \nu_2 = \nu_g = 4$

Préconditionneur	N_g	ε_i	ε_l	# iter	CPU	REEL	% CPU
N.A.	4	10^{-1}	10^{-1}	121	269 s	501 s	53.7
$P_{1-\mathrm{mgV}}(2,2,2,1)$	4	10^{-1}	10^{-1}	115	247 s	444 s	55.6
$P_{1-\mathrm{mgV}}(2,2,2,2)$	4	10^{-1}	10^{-1}	115	280 s	484 s	57.8
$P_{2-\mathrm{mgV}}(2,2,2,1)$	4	10^{-1}	10^{-1}	111	267 s	469 s	57.0
N.A.	4	10^{-2}	10^{-2}	105	570 s	941 s	60.6
$P_{1-\mathrm{mgV}}(2,2,2,1)$	4	10^{-2}	10^{-2}	102	469 s	744 s	63.0
$P_{1-\mathrm{mgV}}(2,2,2,2)$	4	10^{-2}	10^{-2}	104	521 s	842 s	61.9
$P_{2-\mathrm{mgV}}(2,2,2,1)$	4	10^{-2}	10^{-2}	100	471 s	773 s	61.0
N.A.	4	10^{-3}	10^{-3}	89	1097 s	1596 s	68.7
$P_{1-\mathrm{mgV}}(2,2,2,1)$	4	10^{-3}	10^{-3}	87	742 s	1119 s	66.3
$P_{1-\mathrm{mgV}}(2,2,2,2)$	4	10^{-3}	10^{-3}	88	848 s	1312 s	64.6
$P_{2-\mathrm{mgV}}(2,2,2,1)$	4	10^{-3}	10^{-3}	88	663 s	1073 s	61.8

TAB. 3.11 – *Influence du préconditionneur* $P_{p-mgV}(\nu_1,\nu_2,\nu_g,N_c)$
Calcul d'un écoulement complet (cas test **S2**) : maillage N3, étude de scalabilité
$\varepsilon_i = \varepsilon_l = 10^{-3}$ - Résolutions locales : V-cycle avec $\nu_1 = \nu_2 = \nu_g = 2$

Préconditionneur	N_p	# iter GMRES min	# iter GMRES max
N.A.	7	5	19
$P_{1-\mathrm{mgV}}(2,2,2,1)$	7	3	10
$P_{1-\mathrm{mgV}}(2,2,2,2)$	7	3	10
$P_{2-\mathrm{mgV}}(2,2,2,1)$	7	2	8
N.A.	14	6	20
$P_{1-\mathrm{mgV}}(2,2,2,1)$	14	3	11
$P_{1-\mathrm{mgV}}(2,2,2,2)$	14	3	11
$P_{2-\mathrm{mgV}}(2,2,2,1)$	14	3	8
N.A.	28	6	21
$P_{1-\mathrm{mgV}}(2,2,2,1)$	28	3	12
$P_{1-\mathrm{mgV}}(2,2,2,2)$	28	3	12
$P_{2-\mathrm{mgV}}(2,2,2,1)$	28	3	9

FIG. 3.5 – *Calcul d'un écoulement complet (cas test* **S2***) : maillage N3 et* $N_p = 14$

Influence du préconditionneur $P_{p-\mathrm{mgV}}(\nu_1,\nu_2,\nu_g,N_c)$: $\varepsilon_i = \varepsilon_l = 10^{-1}$ (haut) - $\varepsilon_i = \varepsilon_l = 10^{-2}$ (bas)

Résolutions locales : V-cycle avec $\nu_1 = \nu_2 = \nu_g = 2$

Prec 1 : $P_{1-\mathrm{mgV}}(2,2,2,1)$ - Prec 2 : $P_{1-\mathrm{mgV}}(2,2,2,2)$ - Prec 3 : $P_{2-\mathrm{mgV}}(2,2,2,1)$

Axe X : itérations en temps - Axe Y : # itérations de GMRES

FIG. 3.6 – *Calcul d'un écoulement complet (cas test* **S2***) : maillage N3 et* $N_p = 14$
Influence du préconditionneur $P_{p-\mathrm{mgV}}(\nu_1,\nu_2,\nu_g,N_c)$ - $\varepsilon_i = \varepsilon_l = 10^{-3}$
Résolutions locales : V-cycle avec $\nu_1 = \nu_2 = \nu_g = 2$
Prec 1 : $P_{1-\mathrm{mgV}}(2,2,2,1)$ - Prec 2 : $P_{1-\mathrm{mgV}}(2,2,2,2)$ - Prec 3 : $P_{2-\mathrm{mgV}}(2,2,2,1)$
Axe X : itérations en temps - Axe Y : # itérations de GMRES

FIG. 3.7 – *Calcul d'un écoulement complet (cas test* **S2***) : étude de scalabilité, maillage N3*
GMRES non-préconditionné (haut) et préconditionné par $P_{1-\mathrm{mgV}}(2,2,2,1)$ (bas)

$\varepsilon_i = \varepsilon_l = 10^{-3}$ - Résolutions locales : V-cycle avec $\nu_1 = \nu_2 = \nu_g = 2$

Axe X : itérations en temps - Axe Y : # itérations de GMRES

FIG. 3.8 – *Calcul d'un écoulement complet (cas test* **S2***) : étude de scalabilité, maillage N3*
GMRES préconditionné par $P_{1-\mathrm{mgV}}(2,2,2,2)$ (haut) et $P_{2-\mathrm{mgV}}(2,2,2,1)$ (bas)
$\varepsilon_i = \varepsilon_l = 10^{-3}$ - Résolutions locales : V-cycle avec $\nu_1 = \nu_2 = \nu_g = 2$
Axe X : itérations en temps - Axe Y : # itérations de GMRES

3.4 Conclusion

Les résultats numériques préliminaires présentés dans la section 2.6.2 du chapitre 2, portant sur le calcul d'écoulements stationnaires autour d'un profil d'aile NACA0012, ont montré qu'il n'était pas nécessaire de résoudre avec précision les systèmes linéaires résultant de la méthode d'intégration implicite linéarisée (section 2.1.4 de ce même chapitre) pour obtenir une convergence optimale vers la solution stationnaire (au sens du temps total de calcul). Cette remarque s'applique aussi bien à la stratégie de résolution globale basée sur la méthode de relaxation de Jacobi qu'à la stratégie de résolution par sous-domaine proposée dans la section 2.3. Pour cette dernière, l'approche préconisée consiste à résoudre les systèmes locaux par un nombre constant de cycles d'une méthode multigrille, et les systèmes interfaces par une méthode GMRES avec un seuil de convergence généralement élevé ($\varepsilon_i = 10^{-1}$ pour la plupart des calculs réalisés dans la section 2.6.2). En pratique, cela se traduit notamment par un nombre d'itérations de la méthode GMRES relativement faible et ceci en dépit d'une résolution sans application d'un préconditionneur.

Bien que le contexte numérique rappelé ci-dessus ne semble pas justifier une attention particulière quant à l'amélioration de l'efficacité de la résolution des systèmes interfaces, il est toujours intéressant de s'attacher à construire des stratégies de préconditionnement permettant d'une part, de réduire le nombre d'itérations de la méthode GMRES pour résoudre un système donné et, d'autre part, de limiter la dépendance de la résolution sur la granularité de la décomposition (propriété de « scalabilité » des méthodes de décomposition de domaine). Une contrainte qu'il est important de prendre en compte dans cette recherche concerne le coût d'application du préconditionneur. D'une façon générale, il s'agit de trouver le bon compromis entre l'efficacité numérique du préconditionneur et le gain résultant en temps total de calcul. Il est en particulier indispensable de construire un préconditionneur adapté aux plate-formes de calcul parallèles.

Dans ce chapitre, on s'est intéressé à la mise au point d'une technique de préconditionnement algébrique simple pour l'accélération de la résolution itérative des systèmes interfaces issus de la méthode de décomposition de domaine développée dans la section 2.3 du chapitre 2. Compte tenu des propriétés de l'opérateur interface mis en jeu, on a opté pour un préconditionneur de type polynomial et on a proposé une mise en œuvre particulière qui repose sur la méthode multigrille par agglomération de volumes présentée dans la section 2.4. Pour réduire le coût d'application de ce préconditionneur, la stratégie adoptée consiste à fixer des complexités différentes pour la résolution des systèmes locaux suivant que l'on considère l'application de l'opérateur interface ou celle du préconditionneur, lors du calcul des produits matrice-vecteur intervenant dans la méthode GMRES. En pratique, on a choisi de fixer le seuil de résolution ε_l des systèmes locaux pour ce qui concerne l'application de l'opérateur interface alors que l'application du préconditionneur repose sur la réalisation d'un nombre constant de cycles de la méthode multigrille. Cette stratégie a permis de mettre en évidence des réductions des temps de calcul même si les systèmes interfaces ne sont pas résolus de façon précise. Néanmoins, les gains observés sont d'autant plus importants que le seuil de résolution des systèmes interfaces ε_i diminue (de 10^{-1} à 10^{-3} dans les expériences numériques considérées ici).

Chapitre 4

Conditions d'interface optimisées

4.1 Introduction

Ce chapitre est avant tout motivé par les conclusions de la section 1.2 du chapitre 1 où l'on a mené une étude théorique, reposant sur une analyse de Fourier combinée à une technique de diagonalisation classique, qui a permis de montrer que les conditions de raccord naturelles (ou conditions de raccord classiques) assurent de bonnes propriétés de convergence de l'algorithme de type Schwarz additif sans recouvrement de la forme (1.20). Partant de ce résultat préliminaire, on vise dans ce chapitre deux objectifs principaux :

- d'une part, on cherche à mieux comprendre l'impact des conditions de raccord classiques sur le comportement de l'algorithme de Schwarz de la forme (1.20). Cette étape va nous conduire à mettre en évidence une forme particulière des itérations à l'interface;
- d'autre part, on s'intéresse à la construction d'une nouvelle forme de conditions d'interface à des fins d'optimisation du taux de convergence. Pour cela, on généralise la forme des itérations à l'interface issue des conditions de raccord classiques en introduisant un certain nombre de paramètres qui constituent les paramètres d'optimisation.

Pour la réalisation de ces deux objectifs, on fait appel à la théorie de Smith pour la diagonalisation des matrices polynomiales[Gan65]. Cette technique nous permet notamment de transcrire les itérations à l'interface de l'algorithme de Schwarz en une forme simplifiée intrinsèque au problème. Cette forme simplifiée permet de donner une explication au bon comportement de l'algorithme de Schwarz basé sur les conditions d'interface classiques. On notera qu'une approche similaire a été considérée dans un contexte différent par Gonzalez[Gon00] afin de dériver des conditions de raccord du second ordre, qualifiées de conditions de raccord optimisées, qui conduisent à une convergence optimale de l'algorithme de type Schwarz additif sans recouvrement pour ce type de problème.

Dans l'approche adoptée ici, on commence par exprimer la forme de l'algorithme de Schwarz basé sur les conditions d'interface classiques, en variables de Smith au lieu des variables physiques du problème. On propose ensuite une généralisation de la forme des conditions de raccord. Les nouvelles conditions, construites par une série de transformations algébriques, sont au plus du premier ordre en dérivée partielle dans la direction tangentielle à l'interface et dépendent de quelques paramètres qui, s'ils sont convenablement choisis, permettent une meilleure convergence de l'algorithme de type Schwarz additif. Dans cette étude préliminaire, on s'intéresse plus particulièrement à des conditions d'ordre zéro (c'est-à-dire sans dérivée dans la direction tangentielle à l'interface). Des expériences numériques permettent de retrouver

un comportement qualitativement en accord avec la théorie en dépit des approximations numériques qui caractérisent la mise en œuvre de l'algorithme.

4.2 Diagonalisation de Smith

Dans le cas d'un système d'équations aux dérivées partielles à coefficients constants, si on procède à une transformation de Laplace partielle dans la diréction x, on obtient un système dont le symbôle est une matrice polynômiale en λ (variable duale de x). Le but de cette section est d'introduire les principes de base de la théorie de Smith[Gan65] qui traite de la diagonalisation (et de la factorisation) de matrices polynômiales. Ensuite, nous nous proposons d'appliquer cette théorie pour mieux analyser et comprendre la convergence d'un algorithme de type Schwarz additif de la forme (1.20), c'est-à-dire basé sur des conditions de raccord classiques, appliqué à la résolution du système des équations d'Euler linéarisé. Nous nous limitons au cas à deux dimensions d'espace et nous considérons toujours une décomposition en deux sous-domaines non-recouvrants du plan réel.

4.2.1 Forme diagonale de Smith

On rappelle maintenant quelques résultats d'algèbre linéaire qui permettent de transformer une matrice polynomiale en une matrice diagonale dite « forme diagonale de Smith »[Gan65].

Définition 2 Soit $P(\lambda)$ une matrice polynomiale. On définit un ensemble d'opérations algébriques élémentaires applicables à $P(\lambda)$: la permutation de deux lignes (ou colonnes), l'addition d'une ligne avec une autre ligne multipliée par un polynôme en λ (ou la même opération avec deux colonnes) et la multiplication d'une ligne (ou colonne) par une constante non nulle.

Deux matrices polynomiales $P(\lambda)$ et $M(\lambda)$ sont dites à lignes et colonnes équivalentes s'il existe une suite finie des opérations élémentaires ci-dessus, sur les lignes et les colonnes de $P(\lambda)$, telles que :

$$M(\lambda) = E(\lambda)P(\lambda)F(\lambda)$$

où $E(\lambda)$ et $F(\lambda)$ sont des produits de matrices polynomiales élémentaires issues respectivement d'opérations sur les lignes et sur les colonnes de $P(\lambda)$.

Les matrices $E(\lambda)$ et $F(\lambda)$ de la factorisation précédente ont un déterminant constants (c'est-à-dire non polynomiaux). Les déterminants de $P(\lambda)$ et de $M(\lambda)$ sont égaux à une constante non nulle près.

Le résultat principal de la théorie de factorisation de Smith[Gan65] est donné par le théorème suivant.

Théorème 5 Toute matrice polynomiale $A(\lambda)$ inversible est équivalente, dans le sens défini précédemment, à une matrice diagonale de la forme :

$$D(\lambda) = \begin{pmatrix} \mathcal{L}_{m,1}(\lambda) & 0 & . & 0 \\ 0 & \mathcal{L}_{m,2}(\lambda) & . & 0 \\ . & . & . & . \\ 0 & 0 & . & \mathcal{L}_{m,m}(\lambda) \end{pmatrix} \tag{4.1}$$

où $\mathcal{L}_{m,j}(\lambda)$ est un polynôme scalaire tel que $\mathcal{L}_{m,j-1}(\lambda)$ est un diviseur de $\mathcal{L}_{m,j}(\lambda)$ et $\mathcal{L}_{m,j}(\lambda) = \dfrac{\phi_j}{\phi_{j-1}}$ où ϕ_j est le plus grand diviseur commun de tous les $j \times j$ sous-déterminants de $A(\lambda)$.

$D(\lambda)$ est appelée *forme diagonale de Smith* de $A(\lambda)$ et vérifie $\det(A(\lambda)) = \det(D(\lambda))$ et $A(\lambda) = E(\lambda)D(\lambda)F(\lambda)$.

Démonstration. D'abord on va démontrer que, dans le cas où cette factorisation existe, les éléments de la forme diagonale de Smith ont bien la forme $\mathcal{L}_{m,j}(\lambda) = \dfrac{\phi_j}{\phi_{j-1}}$. Pour cela on va prouver qu'étant données deux matrices polynômiales $L_1(\lambda)$ et $L_2(\lambda)$ de même taille telles que $L_1(\lambda) = E(\lambda)L_2(\lambda)F(\lambda)$ où $E^{\pm}(\lambda)$ et $F^{\pm}(\lambda)$ sont des matrices carrées polynômiales, les plus grands diviseurs communs des mineurs d'ordre j, $\phi_{1,j}$ et $\phi_{2,j}$ de $L_1(\lambda)$ et $L_2(\lambda)$ sont égaux. On sait que les mineurs d'ordre j de $L_1(\lambda)$ peuvent être exprimés comme combinaisons linéaires des mineurs de même ordre de $L_2(\lambda)$. On en déduit que $\phi_{2,j}$ est un diviseur de $\phi_{1,j}$. Mais l'équation $L_2(\lambda) = E^{-1}(\lambda)L_1(\lambda)F^{-1}(\lambda)$ implique que $\phi_{1,j}$ est un diviseur de $\phi_{2,j}$, donc $\phi_{1,j} = \phi_{2,j}$.

On peut maintenant appliquer cette observation aux matrices polynômiales $A(\lambda)$ et $D(\lambda)$. On en déduit qu'on doit démontrer la première assertion seulement dans le cas où A est diagonale. De part la structure de D, il est clair que le plus grand diviseur commun des mineurs de D d'ordre j est $\mathcal{L}_{m,1}(\lambda)\mathcal{L}_{m,2}(\lambda)...\mathcal{L}_{m,j}(\lambda)$, la conclusion sur la forme des termes diagonaux étant évidente.

Il faut ensuite démontrer que toute matrice polynômiale peut être réduite à sa forme diagonale par une suite de transformations élémentaires où les éléments de la diagonale respectent la propriété précédente. On va réaliser cette démonstration par induction par rapport à la dimension m de la matrice. Pour $m = 1$ le résultat est évident. On choisit un m générique et on suppose le résultat démontré pour $m - 1$. On suppose aussi que l'élément (1,1) de la matrice A est non-nul et qu'il a le dégré minimal parmi tous les éléments non-nuls de A. Par un nombre fini de transformations élémentaires on peut réduire $A(\lambda)$ à la forme suivante:

$$A_1(\lambda) = \begin{bmatrix} a_{11}^{(1)}(\lambda) & 0 & \dots & 0 \\ 0 & a_{22}^{(1)}(\lambda) & \dots & a_{2m}^{(1)}(\lambda) \\ \vdots & \vdots & & \vdots \\ 0 & a_{m2}^{(1)}(\lambda) & \dots & a_{mm}^{(1)}(\lambda) \end{bmatrix}$$

Cette réduction n'est quand même pas triviale. Pour ce qui concerne les éléments de la première ligne, on divise $a_{j1}(\lambda)$ par $a_{11}(\lambda)$: $a_{j1}(\lambda) = a_{11}(\lambda)b_{j1}(\lambda) + r_{j1}(\lambda)$, où $r_{j1}(\lambda)$ est le polynôme reste dont le dégré est inferieur à celui de $a_{11}(\lambda)$ ou $r_{j1}(\lambda) = 0$. On additionne à la j-ième colonne la première multipliée par $-b_{j1}(\lambda)$; ensuite, si le reste $r_{j,1}$ est non nul on le met sur la première place de la ligne. S'il reste des éléments non nuls sur la ligne on applique à nouveau le processus précédent en divisant par $r_{j,1}(\lambda)$. Comme le dégré du polynôme reste decroît, après un nombre fini de pas on peut se ramener au cas où tous les éléments de la ligne sdont nuls mis à part le premier. On procède de la même façon sur la première colonne.

Si on suppose maintenant qu'il existe un $a_{ij}^{(1)}(\lambda) \neq 0$ qui n'est pas divisible par $a_{11}^{(1)}(\lambda)$ alors on ajoute à la première ligne la i-ème et on ré-applique la démarche ci-dessus, ce qui nous permet de nous rammener à la forme:

$$A_2(\lambda) = \begin{bmatrix} a_{11}^{(2)}(\lambda) & 0 & \dots & 0 \\ 0 & a_{22}^{(2)}(\lambda) & \dots & a_{2m}^{(2)}(\lambda) \\ \vdots & \vdots & & \vdots \\ 0 & a_{m2}^{(2)}(\lambda) & \dots & a_{mm}^{(2)}(\lambda) \end{bmatrix}$$

où le dégré de $a_{11}^{(2)}(\lambda)$ est inferieur à celui de $a_{11}^{(1)}(\lambda)$. S'il y a encore des éléments $a_{ij}^{(1)}(\lambda)$ qui sont pas divisibles par $a_{11}^{(2)}(\lambda)$ on repète la procedure précedente jusqu'a ce que, après un nombre fini de pas on obtient la matrice:

$$A_3(\lambda) = \begin{bmatrix} a_{11}^{(3)}(\lambda) & 0 & \dots & 0 \\ 0 & a_{22}^{(3)}(\lambda) & \dots & a_{2m}^{(3)}(\lambda) \\ \vdots & \vdots & & \vdots \\ 0 & a_{m2}^{(3)}(\lambda) & \dots & a_{mm}^{(3)}(\lambda) \end{bmatrix}$$

dans laquelle chaque $a_{ij}^{(3)}$ est divisible par $a_{11}^{(3)}$. En appliquant maintenant l'hypothèse d'induction à la matrice:

$$A_4(\lambda) = \frac{1}{a_{11}^{(3)}(\lambda)} \begin{bmatrix} a_{22}^{(3)}(\lambda) & \dots & a_{2m}^{(3)}(\lambda) \\ \vdots & & \vdots \\ a_{m2}^{(3)}(\lambda) & \dots & a_{mm}^{(3)}(\lambda) \end{bmatrix}$$

de taille $m-1$ on obtient la démonstration complète du théorème.

4.2.1.1 Algorithme de Schwarz appliqué à un système en variables de Smith

On va maintenant utiliser cette forme diagonale d'une matrice $A(\lambda)$ issue de la transformation de Laplace partielle d'un système d'équations auquel on applique un algorithme de résolution de type Schwarz additif. On se situe dans le cas bidimensionnel et pour une décomposition en deux sous-domaines verticaux où l'interface est donnée par la droite $x = 0$.

ALGORITHME 5 Algorithme de type Schwarz additif.

Soit $W_i^{(0)}$ pour $i = 1{,}2$ une approximation initiale de la solution du système d'équations $A(\lambda)W = f$ dans le sous-domaine Ω_i. Les approximations $W_i^{(p+1)}$ pour $i = 1{,}2$ de W dans les sous-domaines Ω_i à l'itération $p+1$ (où p désigne l'itération de l'algorithme de type Schwarz) sont les solutions des problèmes :

$$\begin{aligned} \Omega_1 \quad : \quad & \begin{cases} AW_1^{(p+1)} & = & f_1 \\ \mathcal{C}_1 W_1^{(p+1)} & = & \mathcal{C}_1 W_2^p \end{cases} \\ \Omega_2 \quad : \quad & \begin{cases} AW_2^{(p+1)} & = & f_2 \\ \mathcal{C}_2 W_2^{(p+1)} & = & \mathcal{C}_2 W_1^p \end{cases} \end{aligned} \tag{4.2}$$

En utilisant la factorisation de $A(\lambda) = E(\lambda)D(\lambda)F(\lambda)$ l'algorithme devient :

ALGORITHME 6 Algorithme de type Schwarz additif en variables de Smith. *Dans les mêmes conditions et avec les mêmes notations que dans l'algorithme précédent on obtient:*

$$
\begin{array}{lll}
\Omega_1 & : & \left\{ \begin{array}{lll} D\bar{W}_1^{(p+1)} & = & \bar{f}_1 \\ \mathcal{B}_1\bar{W}_1^{(p+1)} & = & \mathcal{B}_1\bar{W}_2^p \end{array} \right. \\
\Omega_2 & : & \left\{ \begin{array}{lll} D\bar{W}_2^{(p+1)} & = & \bar{f}_2 \\ \mathcal{B}_2\bar{W}_2^{(p+1)} & = & \mathcal{B}_2\bar{W}_1^p \end{array} \right.
\end{array}
\tag{4.3}
$$

où $\bar{f}_{1,2} = E^{-1}F_{1,2}$, $\bar{W} = F^{-1}W$ et $\mathcal{B}_{1,2} = F^{-1}\mathcal{C}_{1,2}$, E^{-1} et F^{-1} étant des matrices polynômiales en λ. Les opérateurs d'interface $\mathcal{B}_i$ sont des polynômes en λ.

Cette dernière forme de l'algorithme va nous servir par la suite dans l'étape de dérivation du taux de convergence mais aussi, dans l'interprétation du comportement de l'algorithme de Schwarz et enfin, dans la construction des nouvelles conditions d'interface. L'avantage principal à utiliser la diagonalisation de Smith réside dans le fait que celle-ci repose seulement sur la connaisaance des valeurs propres généralisées, sans faire appel aux vecteurs propres (ce qui est en fait plus adapté au traitement de systèmes incompletement paraboliques comme le système d'équations de Navier-Stokes), la matrice F étant connue d'une façon exacte quelle que soit l'expression des valeurs propres généralisées. Cette forme de l'algorithme est intrinsèque au problème en fournissant ainsi des informations précises sur le comportement des algorithmes de résolution appliqués à celui-ci.

4.2.2 Application au système d'équations d'Euler

Le point de départ de notre analyse est donc donné par le système linéarisé exprimé, pour l'instant, en termes des variables conservatives :

$$
\mathcal{L}W := \beta W + A_1\partial_x W + A_2\partial_y W = f \quad , \quad \beta = \frac{1}{\Delta t}
\tag{4.4}
$$

En effectuant une transformation de Laplace dans la direction x (λ étant la variable de Laplace correspondante) et une transformation de Fourier dans la direction y (k étant la variable de Fourier correspondante) on obtient le symbole de l'opérateur aux dérivées partielles $\mathcal{L}$ qui est une matrice polynomiale en λ notée $A_{(\lambda,k)}(W)$:

$$
A_{(\lambda,k)}(W) = \begin{pmatrix}
\beta & \lambda & ik & 0 \\
A_{21} & (3-\gamma)u\lambda + ikv + \beta & iku - (\gamma-1)v\lambda & (\gamma-1)\lambda \\
A_{31} & -ik(\gamma-1)u + v\lambda & -ikv(\gamma-3) + \beta + \lambda u & ik(\gamma-1) \\
A_{41} & A_{42} & A_{43} & \gamma u\lambda + ikv\gamma + \beta
\end{pmatrix}
\tag{4.5}
$$

avec :

$$\begin{cases} A_{21}(\lambda,k) &= -ikuv + \frac{1}{2}((\gamma-3)u^2 + (\gamma-1)v^2)\lambda \\ A_{31}(\lambda,k) &= -\lambda uv + \frac{1}{2}((\gamma-1)u^2 + (\gamma-3)v^2)ik \\ A_{41}(\lambda,k) &= \frac{u((\gamma^2-3\gamma+2)(u^2+v^2)-2c^2)}{2(\gamma-1)}\lambda + \frac{v((\gamma^2-3\gamma+2)(u^2+v^2)-2c^2)}{2(\gamma-1)}ik \\ A_{42}(\lambda,k) &= \left((\frac{3}{2}-\gamma)u^2 + \frac{1}{2}v^2 + \frac{c^2}{\gamma-1}\right)\lambda - (\gamma-1)ikuv \\ A_{43}(\lambda,k) &= -(\gamma-1)uv\lambda + \left((\frac{3}{2}-\gamma)v^2 + \frac{1}{2}u^2 + \frac{c^2}{\gamma-1}\right)ik \end{cases}$$

L'application de cette théorie dans le contexte qui nous intéresse ici consiste donc à chercher une factorisation de Smith de $A(\lambda,k)$ vue comme une matrice polynomiale en λ :

$$A(\lambda,k) = E(\lambda,k)D_s(\lambda,k)F(\lambda,k)$$

où $D_s(\lambda,k)$ représente la forme diagonale de Smith de $A(\lambda,k)$; $E(\lambda,k)$ est déduite des opérations sur les lignes et $F(\lambda,k)$ des opérations sur les colonnes.

Après quelques manipulations algébriques on obtient l'expression de la forme diagonale de Smith de la matrice $A(\lambda,k)$:

$$D_s(\lambda,k) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \mathcal{G}(\lambda,k) & 0 \\ 0 & 0 & 0 & \mathcal{G}(\lambda,k)\mathcal{L}(\lambda,k) \end{pmatrix} \tag{4.6}$$

où :

$$\begin{cases} \mathcal{L}(\lambda,k) &= -(c^2-u^2)\lambda^2 + 2u(\beta+ikv)\lambda + c^2k^2 + (\beta+ikv)^2 \\ \mathcal{G}(\lambda,k) &= \lambda u + (\beta+ikv) \end{cases} \tag{4.7}$$

et :

$$F(\lambda,k) = \begin{pmatrix} \beta & \lambda & ik & 0 \\ 0 & F_{22} & F_{23} & (\gamma-1)\beta^2 \\ 0 & F_{32} & F_{33} & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{4.8}$$

avec :

$$\left\{\begin{array}{lcl} F_{22}(\lambda,k) & = & u(c^2-u^2-v^2)\lambda^2 \\ & + & ((c-u^2-v^2)iku+\frac{1}{2}\beta(2c^2-(\gamma+3)u^2-(\gamma+1)v^2))\lambda \\ & - & (\beta\gamma+ikv)\beta u \\ F_{23}(\lambda,k) & = & -((u^2+v^2-c^2)ik+\beta v)u\lambda \\ & - & v(c-u^2-v^2)k^2+\frac{1}{2}\beta(2c^2-(\gamma+1)u^2-(\gamma+3)v^2)ik-\beta^2\gamma v \\ F_{32}(\lambda,k) & = & ((u^2+v^2-c^2)ik+\beta v)\lambda+ik\beta u \\ F_{33}(\lambda,k) & = & -(u^2+v^2-c^2)k^2+2ik\beta v+\beta^2 \end{array}\right. \tag{4.9}$$

$$E(\lambda,k)=\begin{pmatrix} 1 & 0 & 0 & 0 \\ E_{21} & \frac{\lambda}{\beta^2} & E_{23} & E_{24} \\ E_{31} & \frac{ik}{\beta^2} & \frac{1}{\beta^2} & 0 \\ E_{41} & E_{42} & E_{43} & E_{44} \end{pmatrix} \tag{4.10}$$

avec :

$$\left\{\begin{array}{lcl} E_{21}(\lambda,k) & = & \dfrac{((\gamma-3)u^2+(\gamma-1)v^2)\lambda-2uvik}{2\beta} \\ E_{23}(\lambda,k) & = & \dfrac{((u^2+v^2-c^2)ik+\beta v)\lambda+\beta uik}{\beta^2F_{33}} \\ E_{24}(\lambda,k) & = & F_{33}^{-1} \\ E_{31}(\lambda,k) & = & \dfrac{-2uv\lambda+ik((\gamma-1)u^2+(\gamma-3)v^2)}{2\beta} \\ E_{41}(\lambda,k) & = & -\dfrac{(\lambda u+ikv)(2c^2-(\gamma-1)(\gamma-3)(u^2+v^2))}{\beta(\gamma-1)} \\ E_{42}(\lambda,k) & = & \dfrac{\beta+\gamma(\lambda u+ikv)}{\beta^2(\gamma-1)} \\ E_{43}(\lambda,k) & = & \gamma\dfrac{\lambda u((u^2+v^2-c^2)ik+\beta v)-k^2v(u^2+v^2-c^2)+ik\beta(u^2+2v^2)+\beta^2v}{\beta^2(\gamma-1)F_{33}} \\ E_{44}(\lambda,k) & = & \dfrac{u\gamma}{(\gamma-1)F_{33}} \end{array}\right. \tag{4.11}$$

Les formules ci-dessus ont aussi été obtenues par Gonzalez[Gon00] et constituent le point de départ à sa recherche de conditions de transmission d'ordre élevé pour le système d'équations d'Euler.

4.2.2.1 Algorithme de type Schwarz additif

On va maintenant utiliser cette forme diagonale de la matrice $A(\lambda,k)$ pour calculer le taux de convergence de l'algorithme de type Schwarz additif dans le cas bidimensionnel et pour une décomposition en deux sous-domaines verticaux où l'interface est donnée par la droite $x=0$. Le résultat de cette analyse confirme celui obtenu par la diagonalisation par vecteurs propres (voir la section 1.2.2 du chapitre 1).

Par ailleurs, dans la section suivante, ce résultat servira de point de départ à la recherche de conditions d'interface *généralisées* (ou *optimisées*) en vue de l'accélération de la convergence de l'algorithme de base considéré ici.

ALGORITHME 7 Algorithme de type Schwarz additif en variables de Smith (forme basée sur le vecteur solution $W^{(p)}$). Soit $W_i^{(0)}$ pour $i = 1{,}2$ une approximation initiale de la solution du système d'équations d'Euler au pas de temps $n\Delta t$ dans le sous-domaine Ω_i. Les approximations $W_i^{(p+1)}$ pour $i = 1{,}2$ de W dans les sous-domaines Ω_i à l'itération $p+1$ (où p désigne l'itération de l'algorithme de type Schwarz) sont les solutions des problèmes :

$$\begin{array}{lcl} \Omega_1 & : & \left\{ \begin{array}{lcl} D_s F W_1^{(p+1)} & = & \bar{f}_1 \\ \left(T^{-1} W_1^{(p+1)}\right)_j & = & \left(T^{-1} W_2^{(p)}\right)_j \quad \text{pour} \quad \lambda_j(A_1) < 0 \end{array} \right. \\ \Omega_2 & : & \left\{ \begin{array}{lcl} D_s F W_2^{(p+1)} & = & \bar{f}_2 \\ \left(T^{-1} W_2^{(p+1)}\right)_j & = & \left(T^{-1} W_1^{(p)}\right)_j \quad \text{pour} \quad \lambda_j(A_1) > 0 \end{array} \right. \end{array} \tag{4.12}$$

où T est la matrice qui diagonalise A_1 c'est-à-dire $A_1 = T\Lambda T^{-1}$ avec :

$$T = \begin{pmatrix} 1 & 1 & 0 & 1 \\ u-c & u+c & 0 & u \\ v & v & c\sqrt{2} & v \\ \frac{1}{2}(u^2+v^2) - cu + \frac{c^2}{\gamma-1} & \frac{1}{2}(u^2+v^2) + cu + \frac{c^2}{\gamma-1} & cv\sqrt{2} & \frac{1}{2}(u^2+v^2) \end{pmatrix}$$

et :

$$\Lambda = \text{diag}(u-c\ ,\ u+c\ ,\ u\ ,\ u)$$

Comme dans la section 1.2.2, précédemment, les conditions de transmission pour l'algorithme de type Schwarz ci-dessus sont des conditions de Dirichlet pour les caractéristiques entrantes dans le sous-domaine considéré.

Soit $W = (w_1\ ,\ w_2\ ,\ w_3\ ,\ w_4)^T$ le vecteur des variables conservatives. La factorisation de la matrice $A(\lambda,k)$ nous suggère le changement de variables $\bar{W} = FW$ où $\bar{W}$ est le vecteur des variables de Smith. Avec ce changement de variables, les équations dans chaque sous-domaine deviennent:

$$D_s \bar{W} = \bar{f} \quad \Leftrightarrow \quad \left\{ \begin{array}{c} \beta w_1 + \lambda w_2 + ikw_3 = \bar{f}_1 \\ F_{22} w_2 + F_{23} w_3 + (\gamma-1)\beta^2 w_4 = \bar{f}_2 \\ \mathcal{G}\bar{w}_s \equiv \mathcal{G}(F_{32} w_2 + F_{33} w_3) = \bar{f}_3 \\ \mathcal{G}\mathcal{L}\bar{w}_2 = \bar{f}_4 \end{array} \right. \tag{4.13}$$

A cause de la structure de la matrice D_s on ne va retenir que deux variables de Smith, w_2 et w_s, les autres étant déduites de celles-ci à partir des relations (4.13). Ce sont les seules variables qui jouent un rôle dans les calculs qui suivent. Par ailleurs, les variables physiques sont exprimées en fonction des variables de Smith à travers la relation matricielle $W = F^{-1}\bar{W}$.

4.2.3 Analyse de convergence

Dans cette partie on se propose de calculer le taux de convergence de l'algorithme de Schwarz en utilisant la nouvelle forme des équations en variables de Smith. Ceci nous permettra, d'une part, de retrouver le résultat obtenu dans la section 1.2.2 du chapitre 1 et, d'autre part, d'employer cette expression de l'algorithme pour la généralisation des conditions de transmission.

On considère le cas d'une décomposition en deux sous-domaines caractérisée par $\Omega_1 = \mathbb{R}_- \times \mathbb{R}$ et $\Omega_2 = \mathbb{R}_+ \times \mathbb{R}$ séparés par l'interface $x = 0$; $\mathbf{n} = (\mathbf{1},\mathbf{0})$ désigne le vecteur normal à l'interface $x = 0$, dirigée de Ω_1 vers Ω_2. Soient :

$$\begin{cases} M_n &= \dfrac{\mathbf{V.n}}{c} = \dfrac{u}{c} \\ M_t &= \dfrac{\mathbf{V.t}}{c} = \dfrac{v}{c} \end{cases}$$

respectivement les nombres de Mach normal et tangentiel à l'interface $x = 0$. Par ailleurs, en tout point de $\Omega_1 \bigcup \Omega_2$, le nombre de Mach local peut s'exprimer par :

$$M = \frac{\sqrt{u^2 + v^2}}{c} = \sqrt{M_n^2 + M_t^2}$$

Comme dans la section 1.2.2 on va faire l'hypothèse que l'écoulement est subsonique, c'est-à-dire que $M < 1$, qui correspond au seul cas intéressant pour notre étude. Ceci signifie aussi que $\dfrac{|u|}{c} < 1$ et $\dfrac{|v|}{c} < 1$ puisque $M^2 = \dfrac{u^2 + v^2}{c}$. On suppose aussi que l'écoulement est tel que $u > 0$, autrement dit on a $0 < u < c$ (le résultat pour le cas $u < 0$ peut se déduire facilement de celui obtenu pour $u > 0$).

Dans ce qui suit il est préférable de travailler en termes de vecteurs erreur. On note donc par $(E_i^{(p)})(x,k) = (W_i^{(p)} - W_i)(x,k) = (e_1^i \ , \ e_2^i \ , \ e_3^i \ , \ e_4^i)^T$ le vecteur erreur dans le sous-domaine Ω_i à l'itération p de l'algorithme de type Schwarz. En variables de Smith, le vecteur erreur s'écrit $(\bar{E}_i^{(p)})(x,k) = F(W_i^{(p)} - W_i)(x,k)$.

L'algorithme de type Schwarz additif 7 peut être reformulé en utilisant le vecteur erreur en tenant compte du fait que les variables w_3 et w_4 peuvent être déduites diréctement des deux premières rélations satisfaites à l'interieur des sous-domaines données par 4.13.

`ALGORITHME 8` Algorithme de type Schwarz additif en variables de Smith (forme basée sur le vecteur erreur $E^{(p)}$). Étant donné le changement de variables $\bar{E} = FE$, l'algorithme de type Schwarz additif s'écrit :

$$
\begin{array}{lll}
\Omega_1 & : & \left\{ \begin{array}{lll}
\mathcal{G}\left((\bar{e}_s^1)^{(p+1)}\right) & = & 0 \\
\mathcal{GL}\left((\bar{e}_2^1)^{(p+1)}\right) & = & 0 \\
(B\bar{E}_1^{(p+1)})_j & = & (B\bar{E}_2^{(p)})_j \quad \text{pour} \quad \lambda_j(A_1) < 0
\end{array} \right. \\
\Omega_2 & : & \left\{ \begin{array}{lll}
\mathcal{G}\left((\bar{e}_s^2)^{(p+1)}\right) & = & 0 \\
\mathcal{GL}\left((\bar{e}_2^2)^{(p+1)}\right) & = & 0 \\
(B\bar{E}_2^{(p+1)})_j & = & (B\bar{E}_1^{(p)})_j \quad \text{pour} \quad \lambda_j(A_1) > 0
\end{array} \right.
\end{array}
\tag{4.14}
$$

où $B(\lambda,k)$ est une matrice 4×2. Les deux dernières colonnes de $T^{-1}F^{-1}(\lambda,k)$ se comportent comme un opérateur interface qui agit sur les variables de Smith $\bar{w}_s$ et w_2.

En effectuant les calculs on en déduit les composantes de B :

$$
\left\{ \begin{array}{lll}
b_{11} & = & \left[\dfrac{\beta^2}{2\beta^2c^2F_{33}}\right]\left[u(u^2-c^2)\lambda^2-(\beta+ikv)(c-u)(c+2u)\lambda-(k^2c^3+(\beta+ikv)^2(c-u))\right] \\
b_{12} & = & \left[\dfrac{1}{2\beta^2c^2F_{33}}\right]\left[(\beta v-ik(c^2-u^2-v^2))\mathcal{G}+ik\beta cu\right] \\
b_{21} & = & \left[\dfrac{\beta^2}{2\beta^2c^2F_{33}}\right]\left[u(u^2-c^2)\lambda^2-(\beta+ikv)(c+u)(c-2u)\lambda+(k^2c^3+(\beta+ikv)^2(c+u)\right] \\
b_{22} & = & \left[\dfrac{1}{2\beta^2c^2F_{33}}\right]\left[(\beta c-ik(c^2-u^2-v^2))\mathcal{G}-ik\beta cu\right] \\
b_{31} & = & \left[\dfrac{ik\beta\sqrt{2}}{2\beta cF_{33}}\right]\left[\lambda(c^2-u^2)-u(\beta+ikv)\right] \\
b_{32} & = & \left[\dfrac{\sqrt{2}}{2\beta cF_{33}}\right]\left[\beta+ikv\right] \\
b_{41} & = & -\left[\dfrac{\beta^2}{\beta^2c^2F_{33}}\right]\left[u\mathcal{L}\right] \\
b_{42} & = & -\left[\dfrac{1}{\beta^2c^2F_{33}}\right]\left[(\beta v-ik(c^2-u^2-v^2))\mathcal{G}-ik\beta c^2\right]
\end{array} \right.
\tag{4.15}
$$

En prenant en compte le signe des valeurs propres de A_1 à l'interface, on en déduit la forme des itérations de Schwarz dans chaque sous-domaine en termes des composantes du vecteur erreur sur les variables de Smith :

$$
\begin{array}{lll}
\Omega_1 & : & \left\{ \; b_{11}(\bar{e}_2^1)^{(p+1)} = b_{11}(\bar{e}_2^2)^{(p)} + b_{12}(\bar{e}_s^2)^{(p)} \right. \\
\Omega_2 & : & \left\{ \begin{array}{lll}
b_{21}(\bar{e}_2^2)^{(p+1)} + b_{22}(\bar{e}_s^2)^{(p+1)} & = & b_{21}(\bar{e}_2^1)^{(p)} \\
b_{31}(\bar{e}_2^2)^{(p+1)} + b_{32}(\bar{e}_s^2)^{(p+1)} & = & b_{31}(\bar{e}_2^1)^{(p)} \\
b_{41}(\bar{e}_2^2)^{(p+1)} + b_{42}(\bar{e}_s^2)^{(p+1)} & = & b_{41}(\bar{e}_2^1)^{(p)}
\end{array} \right.
\end{array}
\tag{4.16}
$$

Étant donnée la forme des équations (4.14), les solutions locales peuvent être exprimées comme des combinaisons linéaires d'exponentielles :

$$\left\{\begin{array}{rcl} \bar{e}_2^1(x,k) & = & \alpha_1 e^{\lambda_{\mathcal{L}_1} x} \\ \bar{e}_2^2(x,k) & = & \alpha_2 e^{\lambda_{\mathcal{G}} x} + \alpha_3 e^{\lambda_{\mathcal{L}_2} x} \\ \bar{e}_s^2(x,k) & = & \alpha_4 e^{\lambda_{\mathcal{G}} x} \end{array}\right. \tag{4.17}$$

où $\lambda_{\mathcal{G}} = \mathcal{F}_k \Lambda_{\mathcal{G}}$, $\lambda_{\mathcal{L}_1} = \mathcal{F}_k \Lambda_{\mathcal{L}_1}$ et $\lambda_{\mathcal{L}_2} = \mathcal{F}_k \Lambda_{\mathcal{L}_2}$ sont des opérateurs pseudodifférentiels qui factorisent $\mathcal{G}$ et $\mathcal{L}$ dont les symboles sont donnés par les relations (4.7) :

$$\left\{\begin{array}{rcl} \mathcal{G} & = & u(\partial_x - \Lambda_{\mathcal{G}}) \\ \mathcal{L} & = & -(c^2 - u^2)(\partial_x - \Lambda_{\mathcal{L}_1})(\partial_x - \Lambda_{\mathcal{L}_2}) \end{array}\right. \tag{4.18}$$

où, par abus de notation, $\mathcal{G}$ et $\mathcal{L}$ désignent ici des opérateurs différentiels pour la variable x. Les expressions de $\lambda_{\mathcal{G}}$, $\lambda_{\mathcal{L}_1}$ et $\lambda_{\mathcal{L}_2}$ sont données par :

$$\left\{\begin{array}{rcl} \lambda_{\mathcal{G}} & = & -\dfrac{\beta + ikv}{u} \\ \lambda_{\mathcal{L}_1} & = & \dfrac{(\beta + ikv)u + cR(k)}{c^2 - u^2} \\ \lambda_{\mathcal{L}_2} & = & \dfrac{(\beta + ikv)u - cR(k)}{c^2 - u^2} \end{array}\right. \tag{4.19}$$

où $R(k) = \sqrt{(\beta + ikv)^2 + k^2(c^2 - u^2)}$. Ces valeurs propres sont en général distinctes. Le seul cas où on l'on a $\lambda_{\mathcal{G}} = \lambda_{\mathcal{L}_2}$ est lorsque $v = 0$ et $k = \dfrac{\beta}{u}$.

En remplaçant les expressions des solutions (4.17) dans les relations (4.16) et en tenant compte du fait que B est un opérateur pseudodifférentiel, on peut exprimer l'itération de l'algorithme de Schwarz à l'aide des coefficients α_i :

$$\begin{pmatrix} \alpha_1^{(p+1)} \\ \alpha_2^{(p+1)} \\ \alpha_3^{(p+1)} \\ \alpha_4^{(p+1)} \end{pmatrix} = \begin{pmatrix} 0 & \dfrac{c(R(k)-a)}{(c+u)R(k)} & \dfrac{R(k)-a}{R(k)+a} & \dfrac{iku(c-u)}{\beta c R(k)(R(k)+a)} \\ -\dfrac{2cu(R(k)-a)}{(c+u)R_2(k)} & 0 & 0 & 0 \\ \dfrac{R(k)-a}{R(k)+a}\dfrac{R_1(k)}{R_2(k)} & 0 & 0 & 0 \\ -\dfrac{2c\beta(a^2-k^2u^2)(R(k)-a)}{ik(c+u)R_2(k)} & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha_1^p \\ \alpha_2^p \\ \alpha_3^p \\ \alpha_4^p \end{pmatrix} \tag{4.20}$$

où $a = \beta + ikv$, $R(k) = \sqrt{a^2 + k^2(c^2 - u^2)}$, $R_1(k) = ac + uR(k)$ et $R_2(k) = ac - uR(k)$. Tous calculs faits, on en déduit le taux de convergence de l'algorithme de Schwarz qui est le rayon spectral de la matrice d'itération :

$$\rho^2_{\text{Schwarz}}(k) = \left| \frac{R(k) - a}{(R(k) + a)^2} \cdot \frac{R(k)(c - 3u) - a(c + u)}{c + u} \right| \tag{4.21}$$

On a démontré dans la section 1.2.2 du chapitre 1 que l'expression ci-dessus du taux de convergence est telle que $\rho^2_{\text{Schwarz}}(k) < 1$ $\forall u, v, c$ pour $M < 1$ et $0 < u < c$ et l'algorithme considéré est donc

convergent. En conclusion, on a retrouvé ici, par une méthode d'analyse différente, les résultats obtenus dans la section 1.2.2. D'une certaine façon, on a validé la méthode d'analyse adoptée ici. Par ailleurs, les itérations à l'interface exprimées sous la forme 4.16 en variables de Smith, va nous permettre de justfier le bon comportement de l'algorithme de Schwarz additif appliqué à la résolution du système d'équations d'Euler (voir la section suivante).

4.2.4 Transformations préliminaires

Dans ce qui suit, on souhaite justifier le bon comportement de l'algorithme de Schwarz considéré précédemment. Ce faisant, on cherche aussi à construire une forme générale de conditions d'interface qui conduisent à une meilleure convergence de l'algorithme de type Schwarz additif.

La demarche adoptée est la suivante : tout d'abord, on reformule l'expression des itérations à l'interface (4.16) en exprimant une des variables de Smith en fonction de l'autre; on obtient ainsi une itération à l'interface à deux niveaux (p et $p-1$). On considère ensuite un problème non-symétrique du troisième ordre générique (qui inclut notamment le système des équations d'Euler sous forme factorisée de Smith) que l'on cherche à résoudre par un algorithme de Schwarz additif basé sur des conditions d'interface à deux niveau qui sont une généralisation de celles exhibées précédemment. Cette forme d'itération pourrait être l'explication du bon taux de convergence de l'algorithme de Schwarz. En effet, si on évalue le taux de convergence comme dans la section 4.2.3, on s'aperçoit que celui-ci peut-être minimisé en jouant sur les paramètres qui définissent les conditions d'interface généralisées. Dans une dernière étape, on s'attache donc à optimiser le taux de convergence de l'algorithme de Schwarz.

On commence tout d'abord par écrire l'itération à l'interface (4.16) à l'aide de la variable de Smith e_2^i en éliminant la variable e_s^2 de la première équation (par exemple) et ensuite, en remplaçant l'expression obtenue dans les autres équations. On obtient :

$$(\bar{e}_s^2)^{(p+1)} = \frac{b_{21}}{b_{22}}\left((\bar{e}_2^1)^{(p)} - (\bar{e}_2^2)^{(p+1)}\right)$$

qui conduit à une nouvelle forme des itérations (4.16) :

$$\begin{array}{lll} \Omega_1 & : & \left\{ \; b_{11}b_{22}(\bar{e}_2^1)^{(p+1)} \;=\; (b_{11}b_{22} - b_{21}b_{12})(\bar{e}_2^2)^{(p)} + b_{21}b_{12}(\bar{e}_2^1)^{(p-1)} \right. \\ \Omega_2 & : & \left\{ \begin{array}{rcl} (b_{31}b_{22} - b_{21}b_{32})(\bar{e}_2^2)^{(p+1)} & = & (b_{31}b_{22} - b_{21}b_{32})(\bar{e}_2^1)^{(p)} \\ (b_{41}b_{22} - b_{21}b_{42})(\bar{e}_2^2)^{(p+1)} & = & (b_{41}b_{22} - b_{21}b_{42})(\bar{e}_2^1)^{(p)} \end{array} \right. \end{array} \tag{4.22}$$

La différence principale entre les itérations (4.22) et la forme habituelle des itérations (4.16) basées sur des conditions de transmission classiques, est que pour le premier sous-domaine l'équation contient non seulement les valeurs correspondant au deuxième sous-domaine à l'itération p, mais aussi la valeur correspondant au premier sous-domaine à l'itération $p-1$, selon le modèle suivant :

$$\mathcal{B}_1(\bar{e}_2^1)^{(p+1)} = (\mathcal{B}_1 + \mathcal{B}_2)(\bar{e}_2^2)^{(p)} - \mathcal{B}_2(\bar{e}_2^1)^{(p-1)} \tag{4.23}$$

Pour le deuxième sous-domaine, on a des itérations de la forme :

$$\mathcal{B}_i(\lambda,k)(\bar{e}_2^2)^{(p+1)} = \mathcal{B}_i(\lambda,k)(\bar{e}_2^1)^{(p)} \quad \text{pour} \quad i = 3,4$$

où $\mathcal{B}_i(\lambda,k) = p_i(k)\lambda^2 + q_i(k)\lambda + r_i(k), i = 1,4$, sont des polynômes du second degré en λ qui sont les symboles des opérateurs pseudodifférentiels (λ correspondant à $\dfrac{\partial}{\partial x}$), et $p_i(k), q_i(k), r_i(k)$ sont des polynômes en ik.

4.3 Conditions d'interface optimisées

Les conditions de la forme 4.23, qui conduisent à une itération à deux niveaux à l'interface, pourraient être une des raisons du bon comportement de l'algorithme de type Schwarz additif sans recouvrement 8. En prenant un peu de recul vis-à-vis du problème particulier qu'on étudie ici, on se propose maintenant de regarder l'effet de cette forme de conditions d'interface sur la convergence de l'algorithme de type Schwarz additif 6 appliqué à un problème plus général que l'on exprime ici sous forme factorisée de Smith.

4.3.1 Un problème non-symétrique du troisième ordre

On considère maintenant un type de problème qui est représentatif du système d'équations d'Euler sous forme factorisée de Smith. Dans ce cas, l'opérateur impliqué peut être factorisé de la façon suivante (voir aussi l'équation (4.18)) :

$$\mathcal{G}\mathcal{L} = (\partial_x - \lambda_0(k))\,(\partial_x - \lambda_1(k))\,(\partial_x - \lambda_2(k)) \tag{4.24}$$

avec $\Re(\lambda_1(k)) > 0$, $\Re(\lambda_0(k)) < 0$ et $\Re(\lambda_2(k)) < 0$.

Les hypothèses sur les λ_i adoptées ci-dessus correspondent dans le cas du système d'équations d'Euler à un écoulement subsonique. On applique l'algorithme de type Schwarz additif au problème défini par cet opérateur en utilisant des conditions de transmission de la forme (4.23) pour un des sous-domaines :

$$\begin{array}{lll}
\Omega_1 & : & \left\{\begin{array}{lll} \mathcal{G}\mathcal{L}(e_1^{p+1}) & = & 0 \ \text{ pour } \ x<0 \\ \mathcal{B}_1(e_1^{p+1}) & = & (\mathcal{B}_1+\mathcal{B}_2)(e_2^p) - \mathcal{B}_2(e_1^{p-1}) \end{array}\right. \\
\Omega_2 & : & \left\{\begin{array}{lll} \mathcal{G}\mathcal{L}(e_2^{p+1}) & = & 0 \quad \text{ pour } \ x>0 \\ \mathcal{B}_3(e_2^{p+1}) & = & \mathcal{B}_3(e_1^p) \\ \mathcal{B}_4(e_2^{p+1}) & = & \mathcal{B}_4(e_1^p) \end{array}\right.
\end{array} \tag{4.25}$$

où les $\mathcal{B}_i$ sont des opérateurs aux dérivées partielles du second ordre :

$$\mathcal{B}_i = p_i(k)\partial_x^2 + q_i(k)\partial_x + r_i(k), \quad i = 1,4 \tag{4.26}$$

A cause de la forme des opérateurs $\mathcal{L}$ et $\mathcal{G}$ et puisque les solutions locales sont supposées bornées à $\pm\infty$, ces dernières peuvent être écrites sous la forme :

$$\left\{\begin{array}{lll} e_1(x,k) & = & \alpha_1 e^{\lambda_1(k)x} \\ e_2(x,k) & = & \alpha_2 e^{\lambda_2(k)x} + \alpha_3 e^{\lambda_0(k)x} \end{array}\right. \tag{4.27}$$

En remplaçant les expression des solutions locales dans les conditions de transmission on en déduit la forme des itérations à l'interface :

$$\Omega_1 \quad : \quad \begin{cases} [p_1\lambda_1^2 + q_1\lambda_1 + r_1]\alpha_1^{(p+1)} & = & [(p_1+p_2)\lambda_2^2 + (q1+q_2)\lambda_2 + r_1 + r_2]\alpha_2^{(p)} \\ & + & [(p_1+p_2)\lambda_0^2 + (q_1+q_2)\lambda_0 + r_1 + r_2]\alpha_3^{(p)} \\ & - & [p_2\lambda_1^2 + q_2\lambda_1 + r_2]\alpha_1^{(p-1)} \end{cases}$$

$$\Omega_2 \quad : \quad \begin{cases} [p_3\lambda_2^2 + q_3\lambda_2 + r_3]\alpha_2^{(p+1)} & + & [p_3\lambda_0^2 + q_3\lambda_0 + r_3]\alpha_3^{(p+1)} \\ & = & [p_3\lambda_1^2 + q_3\lambda_1 + r_3]\alpha_1^{(p)} \\ [p_4\lambda_2^2 + q_4\lambda_2 + r_4]\alpha_2^{(p+1)} & + & [p_4\lambda_0^2 + q_4\lambda_0 + r_4]\alpha_3^{(p+1)} \\ & = & [p_4\lambda_1^2 + q_4\lambda_1 + r_4]\alpha_1^{(p)} \end{cases} \tag{4.28}$$

En définissant le taux de convergence comme précédemment on obtient :

$$\rho^2_{\text{Schwarz}}(k) = \left| 1 - \frac{[\lambda_1 - \lambda_0][\lambda_1 - \lambda_2][A(r_1+r_2) - B(q_1+q_2) + C(p1+p2)]}{[p_1\lambda_1^2 + q_1\lambda_1 + r_1][A\lambda_0\lambda_2 + B(\lambda_0+\lambda_2) + C]} \right| \tag{4.29}$$

où :

$$\begin{cases} A(k) & = & p_3(k)q_4(k) - p_4(k)q_3(k) \\ B(k) & = & p_3(k)r_4(k) - p_4(k)r_3(k) \\ C(k) & = & q_3(k)r_4(k) - q_4(k)r_3(k) \end{cases}$$

Si on écrit l'itération à l'interface pour le deuxième sous-domaine (voir les expressions (4.25)) en utilisant la forme des opérateurs $\mathcal{B}_3$ et $\mathcal{B}_4$ (4.26), et si on procède successivement aux opérations suivantes :

- on multiplie la première relation par r_4 et la deuxième par r_3,
- on multiplie la première relation par q_4 et la deuxième par q_3,
- on multiplie la première relation par p_4 et la deuxième par p_3,

on obtient des relations équivalentes à l'aide de $A(k)$, $B(k)$ et $C(k)$:

$$\begin{cases} [B(k)\partial_x^2 + C(k)\partial_x]e_2^{(p+1)} & = & [B(k)\partial_x^2 + C(k)\partial_x]e_1^{(p)} \\ [A(k)\partial_x^2 - C(k)]e_1^{(p+1)} & = & [A(k)\partial_x^2 - C(k)]e_2^{(p)} \\ [A(k)\partial_x + B(k)]e_1^{(p+1)} & = & [A(k)\partial_x + B(k)]e_2^{(p)} \end{cases}$$

On peut alors remplacer la forme initiale des itérations à l'interface par une des deux premières relations ci-dessus (la troisième étant une combinaison linéaire des deux autres). On notera que le résultat précédent fournit l'expression du taux de convergence de l'algorithme de Schwarz pour n'importe quel problème qui en variables de Smith peut être écrit sous la forme $\mathcal{GL} = 0$ et dont la factorisation obéit aux conditions des équations (4.24).

On voit que le taux de convergence dépend des données caractéristiques du problème (les valeurs propres $\lambda_i(k)$ dans le cas présent) et des conditions d'interface données par les polynômes p_i, q_i, r_i A, B, C. On peut donc directement évaluer l'influence de ces derniers sur le comportement de l'algorithme.

Remarque 7 Parmi les différents choix de conditions de transmission données par les opérateurs $\mathcal{B}_i$, on peut distinguer le cas particulier suivant : $\mathcal{B}_3$, $\mathcal{B}_4$ ou une combinaison linèaire des deux est egal à $\mathcal{L}$. Prenons par exemple $\mathcal{B}_3 = \mathcal{L}$. Étant donné que $\mathcal{B}_3 e_1(x,k) = 0$ et que $\mathcal{B}_3 e_2(x,k) = \alpha_3 \mathcal{B}_3 e^{\lambda_0 x}$, l'itération à l'interface conduit à $\alpha_3^{(p+1)} = 0$, ce qui signifie que les solutions locales s'écrivent :

$$\begin{cases} e_1(x,k) &= \alpha_1 e^{\lambda_1 x} \\ e_2(x,k) &= \alpha_2 e^{\lambda_2 x} \end{cases}$$

ce qui permet de nous ramener à un problème du second ordre au lieu d'un problème du troisième ordre, λ_0 ne jouant plus aucun rôle dans le problème. On doit remarquer que dans le cas d'une équation du deuxième ordre les itérations à deux niveaux ne contribuent pas à l'amélioration de la convergence de l'algorithme il faut donc eviter ces cas.

Remarque 8 En général, les p_i et A sont des constantes (ou peuvent être réduites à des constantes) et $\lambda_{1,2} = \mathcal{O}(|k|)$. Les autres paramètres impliqués dans l'expression du taux de convergence sont des polynômes en ik. On peut alors en déduire le comportement asymptotique quand k tend vers l'infini sous certaines hypothèses sur p_i,q_i,r_i et A,B,C. Par exemple, si :

$$q_1 \ , \ r_1 + r_2 \ , \ q_2 \ , \ B \ , \ \lambda_0 \ \text{ sont des constantes} \tag{4.30}$$

et si $r_1 = \mathcal{O}(k^2)$ et $C = \mathcal{O}(k^2)$ (ce qui correspond, pour le système d'équations d'Euler, au cas où $v = 0$) on obtient :

$$\begin{cases} p_1\lambda_1^2 + q_2\lambda_1 + r_1 &= \mathcal{O}(p_1\lambda_1^2 + r_1) &= \mathcal{O}(k^2) \\ A\lambda_2\lambda_0 + B(\lambda_0 + \lambda_2) + C &= \mathcal{O}(C) &= \mathcal{O}(k^2) \\ (\lambda_1 - \lambda_0)(\lambda_1 - \lambda_2) &= \mathcal{O}(2\lambda_1^2) &= \mathcal{O}(k^2) \\ A(r_1 + r_2) - B(q_1 + q_2) + C(p_1 + p_2) &= \mathcal{O}(C(p_1 + p_2)) &= \mathcal{O}(k^2) \end{cases} \tag{4.31}$$

et par conséquent on a :

$$\lim_{k\to\infty} \rho^2_{\text{Schwarz}}(k) = \lim_{k\to\infty} \left| \frac{r_1 - \lambda_1^2(p_1 + 2p_2)}{r_1 + \lambda_1^2 p_1} \right| \tag{4.32}$$

A ce stade, on note que l'on peut toujours choisir r_1 de sorte que $p_1 + p_2 < \dfrac{r_1}{\lambda_1^2} \quad \forall k$. Autrement dit, on peut toujours faire un choix favorable de conditions de raccord de façon à obtenir ;

$$\lim_{k\to\infty} \left| \frac{r_1 - \lambda_1^2(p_1 + 2p_2)}{r_1 + \lambda_1^2 p_1} \right| < 1$$

De plus, en choisissant r_1 tel que $p_1 + 2p_2 = \lim\limits_{k\to\infty} \dfrac{r_1}{\lambda_1^2}$, la limite asymptotique du taux de convergence s'annule (ce qui est un résultat déjà obtenu dans la section 1.2.2 du chapitre 1 pour le système d'équations d'Euler). Cependant, sans aucune hypothèse particulière sur les données du problème (par exemple si v est non nul), on ne peut pas se prononcer a priori sur le comportement du taux de convergence pour des conditions construites dans un cadre général.

Dans le cas présent, des conditions de raccord classiques, on peut identifier les opérateurs qui se trouvent dans les relations (4.22), obtenues pour le système d'équations d'Euler, en utilisant les conditions de transmission classiques pour l'algorithme de Schwarz. Mais avant de procéder à cette identification, on remarque qu'il est possible de simplifier les itérations (4.24) si on tient compte du fait que $\mathcal{G}\bar{e}_s^2 = 0$. En utilisant les expressions (4.14) et (4.15) on obtient :

$$\left\{ \begin{array}{lcll} b_{12}(\bar{e}_s^2) & = & -ik\beta cu(\bar{e}_s^2) & \equiv \quad \bar{b}_{12}(\bar{e}_s^2) \\ b_{22}(\bar{e}_s^2) & = & ik\beta cu(\bar{e}_s^2) & \equiv \quad \bar{b}_{22}(\bar{e}_s^2) \\ b_{42}(\bar{e}_s^2) & = & ik\beta c^2(\bar{e}_s^2) & \equiv \quad \bar{b}_{42}(\bar{e}_s^2) \end{array} \right. \tag{4.33}$$

Par conséquent les itérations (4.22) peuvent être reformulées en remplaçant b_{12} par $\bar{b}_{12}$, b_{22} par $\bar{b}_{22}$ et b_{42} par $\bar{b}_{42}$:

$$\begin{array}{lll} \Omega_1 & : & \{ \; b_{11}\bar{b}_{22}(\bar{e}_2^1)^{(p+1)} \;=\; (b_{11}\bar{b}_{22} - b_{21}\bar{b}_{12})(\bar{e}_2^2)^{(p)} + b_{21}\bar{b}_{12}(\bar{e}_2^1)^{(p-1)} \\ \Omega_2 & : & \left\{ \begin{array}{lcl} (b_{31}\bar{b}_{22} - b_{21}b_{32})(\bar{e}_2^2)^{(p+1)} & = & (b_{31}\bar{b}_{22} - b_{21}b_{32})(\bar{e}_2^1)^{(p)} \\ (b_{41}\bar{b}_{22} - b_{21}\bar{b}_{42})(\bar{e}_2^2)^{(p+1)} & = & (b_{41}\bar{b}_{22} - b_{21}\bar{b}_{42})(\bar{e}_2^1)^{(p)} \end{array} \right. \end{array} \tag{4.34}$$

On en déduit les relations :

$$\left\{ \begin{array}{lcl} \mathcal{B}_1 & = & \bar{b}_{22}b_{11} \\ \mathcal{B}_2 & = & -\bar{b}_{12}b_{21} \\ \mathcal{B}_1 + \mathcal{B}_2 & = & \bar{b}_{22}b_{11} - \bar{b}_{12}b_{21} \\ \mathcal{B}_3 & = & \bar{b}_{22}b_{31} - \bar{b}_{32}b_{21} \\ \mathcal{B}_4 & = & \bar{b}_{22}b_{41} - \bar{b}_{42}b_{21} \end{array} \right. \tag{4.35}$$

ainsi que les expressions des polynômes $p_i(k)$ et $q_i(k)$:

$$\begin{cases} p_1 &= -u(c^2-u^2) \\ q_1 &= -(\beta+ikv)(c-u)(c+2u) \\ r_1 &= -[k^2c^3+(\beta+ikv)^2(c-u)] \end{cases} \qquad \begin{cases} p_2 &= -u(c^2-u^2) \\ q_2 &= -(\beta+ikv)(c+u)(c-2u) \\ r_2 &= [k^2c^3+(\beta+ikv)^2(c+u)] \end{cases}$$

$$\begin{cases} p_3 &= (\beta+ikv)u(c-u) \\ q_3 &= -[k^2cu(c-u)-(\beta+ikv)^2(c-2u)] \\ r_3 &= -(\beta+ikv)[k^2c(c-u)+(\beta+ikv)^2] \end{cases}$$

$$\begin{cases} p_4 &= u(c+u)(c-u)^2 \\ q_4 &= (\beta+ikv)(c-u)(c^2-2u^2) \\ r_4 &= -[k^2c^2(c^2-u^2)+(\beta+ikv)^2(c^2+cu-u^2)] \end{cases} \tag{4.36}$$

$$\begin{cases} p_1+p_2 &= -2u(c^2-u^2) \\ q_1+q_2 &= -2(\beta+ikv)(c^2-2u^2) \\ r_1+r_2 &= 2u(\beta+ikv)^2 \end{cases}$$

et enfin, les expressions de A, B et C :

$$\begin{cases} A &= u(c-u)^2 \\ B &= -(\beta+ikv)(c-u)u \\ C &= [u(\beta+ikv)^2+k^2c^2(c-u)] \end{cases} \tag{4.37}$$

Les expressions ci-dessus de A, B et C tiennent compte d'une simplification par un facteur multiplicatif $A_0(k) = uc[(\beta+ikv)^2+k^2(c^2-u^2)]$ qui n'a aucune incidence sur le résultat final (en particulier, sur l'expression du taux de convergence).

En résumé, il est donc acquis qu'avec des conditions de transmission basées sur les expressions (4.36) et (4.37), on est sûr d'obtenir un taux de convergence de l'algorithme de type Schwarz additif, qui est strictement inférieur à 1. De plus, si on se place de nouveau dans le cas général (c'est-à-dire en considérant des conditions de raccord autres que les conditions classiques), et si on fait des hypothèses sur les opérateurs qui interviennent, on devrait être capable d'obtenir un comportement asymptotique similaire.

Dans ce qui suit on cherche à construire des conditions de raccord qui conduisent à une meilleure convergence de l'algorithme de type Schwarz additif. Pour cela, on se propose de généraliser la forme des conditions de transmission classiques exprimées en variables de Smith, en préservant les propriétés qui fournissent déjà un bon comportement du taux de convergence.

4.3.2 Construction d'une classe plus générale de conditions d'interface

Les nouvelles conditions d'interface sont tout d'abord construites en variables de Smith. Concrètement, les nouvelles conditions d'interface sont traduites par une matrice T_1 qui remplace la matrice T dans les itérations (4.12). Pour une utilisation pratique des nouvelles conditions, on doit ensuite exprimer celles-ci en termes des variables physiques en remontant à la forme initiale du problème via un certain nombre de transformations algébriques. Par ailleurs, dans cette démarche, on cherche à obtenir des conditions de complexité raisonnable en variables physiques c'est à dire sans dérivées normales et sans dérivées tangentielles d'ordre supérieurs à deux. Cette restriction permettra de réduire naturellement le nombre de paramètres qui interviennent dans la forme la plus générale des itérations à l'interface.

En regardant de plus près les expressions (4.36) des polynômes $p_i(k)$ et $q_i(k)$ qui interviennent dans les conditions d'interface, on peut en écrire une forme plus générale comme suit :

$$\left\{\begin{array}{lcl} p_1 & = & -u(c^2-u^2) \\ q_1 & = & (\beta+ikv)e_1 \\ r_1 & = & (\beta+ikv)^2 e_2 + k^2 e_3 \end{array}\right. \qquad \left\{\begin{array}{lcl} p_2 & = & -u(c^2-u^2) \\ q_2 & = & (\beta+ikv)f_1 \\ r_2 & = & (\beta+ikv)^2 f_2 + k^2 f_3 \end{array}\right.$$

$$\left\{\begin{array}{lcl} p_3 & = & (\beta+ikv)u(c-u) \\ q_3 & = & k^2 a_1 + (\beta+ikv)^2 a_2 \\ r_3 & = & (\beta+ikv)[k^2 a_3 - (\beta+ikv)^2] \end{array}\right. \qquad \left\{\begin{array}{lcl} p_4 & = & u(c+u)(c-u)^2 \\ q_4 & = & (\beta+ikv)b_1 \\ r_4 & = & k^2 b_2 + (\beta+ikv)^2 b_3 \end{array}\right. \tag{4.38}$$

où les a_i, b_i, e_i, f_i sont des paramètres réels. En utilisant les relations ci-dessus combinées aux relations 4.35, on en déduit :

$$\left\{\begin{array}{lcl} b_{i1} & = & \left[\dfrac{\beta^2}{2\beta^2 c^2 F_{33}}\right](p_i\lambda^2 + q_i\lambda + r_i), i = 1{,}2 \\ \mathcal{B}_3 & = & \beta^2(c+u)[p_3\lambda^2 + q_3\lambda + r_3] \\ \mathcal{B}_4 & = & ikc\beta^3[p_4\lambda^2 + q_4\lambda + r_4] \\ & & \\ b_{31} & = & \left[\dfrac{ik\beta\sqrt{2}}{2\beta c F_{33}}\right]\left[\dfrac{\mathcal{B}_3 + \bar{b}_{32}b_{21}}{\bar{b}_{22}}\right] \\ b_{41} & = & -\left[\dfrac{\beta^2}{\beta^2 c^2 F_{33}}\right]\left[\dfrac{\mathcal{B}_4 + \bar{b}_{42}b_{21}}{\bar{b}_{22}}\right] \end{array}\right.$$

Dans les expressions ci-dessus, les b_{i2} sont donnés par (4.15). Les conditions d'interface en variables physiques peuvent être retrouvées à l'aide de la relation matricielle suivante :

$$
\begin{aligned}
T_1 W = B\bar{F}W &= \begin{bmatrix} b_{12} & b_{11} \\ b_{22} & b_{21} \\ b_{32} & b_{31} \\ b_{42} & b_{41} \end{bmatrix} \begin{bmatrix} 0 & F_{32} & F_{33} & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} W \\
&= \begin{bmatrix} 0 & b_{11}+b_{12}F_{32} & b_{12}F_{33} & 0 \\ 0 & b_{21}+b_{22}F_{32} & b_{22}F_{33} & 0 \\ 0 & b_{31}+b_{32}F_{32} & b_{32}F_{33} & 0 \\ 0 & b_{41}+b_{42}F_{32} & b_{42}F_{33} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}
\end{aligned}
\tag{4.39}
$$

Ces conditions d'interface dépendent de la variable λ. Il faut donc utiliser les relations suivantes, issues de (4.13), qui sont vérifiées à l'intérieur de chaque sous-domaine, afin de pouvoir éliminer λ :

$$
\begin{cases}
\beta w_1 + \lambda w_2 + ikw_3 &= 0 \\
F_{22}w_2 + F_{23}w_3 + (\gamma-1)\beta^2 w_4 &= 0
\end{cases}
$$

Ceci nous permet d'exprimer la forme finale des conditions d'interface en variables physiques, qui est donnée ici par la matrice T_1. On va de plus imposer que cette matrice ne contienne que des éléments qui sont des polynômes en k dont le degré est au plus deux. Avec ces exigences, on peut fixer quelques paramètres comme suit :

$$
\begin{cases}
e_1 &= 2u^2 - cu - c^2 \\
e_3 &= e_2(c^2-u^2) - cu^2 - uc^2 + u^3 \\
f_1 &= 2u^2 + cu - c^2 \\
f_3 &= f_2(c^2-u^2) + cu^2 - uc^2 + u^3 \\
a_3 &= -c(c-u) \\
a_1 &= a_2(c^2-u^2) - c^3 + uc^2 + 2cu^2 - 2u^3 \\
b_1 &= -2cu^2 - uc^2 + 2u^3 + c3 \\
b_2 &= b_3(c^2-u^2) - cu^3 + c^3u - c^2u^2 + u^4
\end{cases}
\tag{4.40}
$$

On obtient de cette façon des conditions d'interface qui dépendent de cinq paramètres qui sont ici donnés par a_2, e_2, f_2 et b_3.

Remarque 9 On retrouve la forme classique des conditions d'interface en choisissant $a_2 = c - 2u$, $e_2 = -(c-u)$, $f_2 = c + u$ et $b_3 = u^2 - c^2 - cu$.

En résumé, la forme matricielle des conditions d'interface est maintenant donnée par :

$$\hat{T}_1 = \begin{bmatrix} \dfrac{q^2\gamma_1 + 2cu}{4c^2} & \dfrac{e_2 - u\gamma}{2c^2} & -\dfrac{v\gamma_1}{2c^2} & \dfrac{\gamma_1}{2c^2} \\ \dfrac{q^2\gamma_1 - 2cu}{4c^2} & \dfrac{f_2 - u\gamma}{2c^2} & -\dfrac{v\gamma_1}{2c^2} & \dfrac{\gamma_1}{2c^2} \\ -\dfrac{\bar{a} + ikcuv}{ikc^2u\sqrt{2}} & \dfrac{(f_2 - c - u))(\beta + ikv)}{ikc^2u\sqrt{2}} & -\dfrac{a_2(c+u) - (c^2 - 2u^2)}{c^2u\sqrt{2}} & 0 \\ \dfrac{-q^2\gamma_1 + 2c^2}{2c^2} & -\dfrac{f_2c + b_3 - \gamma u^2}{uc^2} & \dfrac{v\gamma_1}{c^2} & -\dfrac{\gamma_1}{c^2} \end{bmatrix} \tag{4.41}$$

avec :

$$\begin{cases} \gamma_1 &= \gamma - 1 \\ q^2 &= u^2 + v^2 \\ \bar{a} &= \beta(c+u)(a_2 - c + 2u) \end{cases}$$

Remarque 10 Afin d'obtenir des conditions d'interface indépendantes, il est nécessaire d'imposer :

$$\det(\hat{T}_1) \neq 0 \quad \Leftrightarrow \quad a_2(c+u) - (c^2 - 2u^2) \neq 0 \ \text{ et } \ (e_2 + f_2)(c-u) + 2b_3 \neq 0$$

On notera que la matrice obtenue désigne en fait le symbole de Fourier de l'opérateur interface T_1 de l'expression (4.39). Comme il n'apparaît que des quantités polynomiales (ou des fractions rationnelles de polynômes), les conditions de transmission seront données par des opérateurs aux dérivées partielles, sachant que ik est la transformée de Fourier de $\dfrac{\partial}{\partial y}$.

4.3.2.1 Conditions d'ordre 0

Dans cette section, on considère plus particulièrement le cas de conditions de raccord n'impliquant pas de dérivée normale ou tangentielle, c'est-à-dire celles qui sont obtenues en posant $a_2 = c - 2u$ et $f_2 = c + u$. En introduisant ces choix dans l'expression (4.29) on obtient le taux de convergence suivant :

$$\rho^2_{\text{Schwarz}}(\alpha,\sigma,k) = \left| 1 - \frac{2[2\alpha + \sigma - 1][ac + uR(k)]R(k)}{\alpha[R(k) + a][R(k)\sigma + a)(c+u)]} \right| \tag{4.42}$$

où $R(k) = \sqrt{(\beta + ikv)^2 + k^2(c^2 - u^2)}$ et $a = \beta + ikv$ et où on a noté :

$$\begin{cases} \alpha &= -\dfrac{cu - u^2 + b_3}{c^2} \\ \sigma &= -\dfrac{e_2(c-u) - u(2c-u)}{c^2} \end{cases}$$

La propriété d'indépendance des conditions d'interface s'écrit (en fonction des nouveaux paramètres) :

$$2\alpha + \sigma - 1 \neq 0$$

Dans le cas des conditions d'interface classiques on a $\alpha = 1$, $\sigma = 1$.

Une fois la forme générale des conditions d'interface obtenue, on doit déterminer l'ensemble des valeurs admissibles pour les paramètres subsistants, c'est-à-dire, les valeurs des paramètres qui conduisent à des problèmes locaux bien posés. Enfin, on peut envisager la recherche de valeurs particulières des paramètres en optimisant le taux de convergence de l'algorithme de type Schwarz additif.

Remarque 11 Les calculs précédents, et par conséquent, la construction des nouvelles conditions d'interface, ont été faits sous l'hypothèse $c > u > 0$; dans le cas contraire ($0 > u > -c$) on aboutit à une autre forme de conditions.

Il s'agit maintenant d'identifier les ensembles de valeurs admissibles pour les paramètres intervenant dans la définition des conditions d'interface. L'objectif est d'obtenir des problèmes locaux bien posés d'une part, et un algorithme de type Schwarz additif convergent d'autre part. Dans un premier temps, on cherche à résoudre le problème posé dans le demi plan gauche :

$$\begin{cases} \beta W_1 + A_1 \partial_x W_1 + A_2 \partial_y W_1 = 0 \quad \text{dans} \quad \Omega_1 =]-\infty,0[\times \mathbb{R} \\ (T_1 W_1)_j = \Phi_1 \quad \text{pour } j \text{ tel que} \quad \lambda_j(A_1) < 0 \quad \text{et} \quad x = 0 \end{cases}$$

En utilisant le changement de variables $W = T\bar{W}$, T étant la matrice qui diagonalise A_1, on obtient :

$$\begin{cases} \beta \bar{W}_1 + B_1 \partial_x \bar{W}_1 + B_2 \partial_y \bar{W}_1 = 0 \quad \text{dans} \quad \Omega_1 =]-\infty,0[\times \mathbb{R} \\ (T_1 T \bar{W}_1)_j = \Phi_1 \quad \text{pour } j \text{ tel que} \quad \lambda_j(A_1) < 0 \quad \text{et} \quad x = 0 \end{cases}$$

où :

$$B_1 = \text{diag}(u-c, u+c, u, u) \qquad \text{et} \quad B_2 = \begin{bmatrix} v & 0 & \frac{c}{\sqrt{2}} & 0 \\ 0 & v & \frac{c}{\sqrt{2}} & 0 \\ \frac{c}{\sqrt{2}} & \frac{c}{\sqrt{2}} & v & 0 \\ 0 & 0 & 0 & v \end{bmatrix}$$

Les nouvelles conditions d'interface sont donc exprimées à l'aide de la matrice $C = T_1 T$:

$$C = \begin{bmatrix} \frac{\sigma+1}{2} & -\frac{1}{2}\frac{(\sigma-1)(u+c)}{c-u} & 0 & -\frac{1}{2}\frac{u(\sigma-1)}{c-u} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{(\alpha-1)(c-u)}{u} & \frac{(u+c)(\alpha-1)}{u} & 0 & \alpha \end{bmatrix} \tag{4.43}$$

En procédant comme à la section 1.2, c'est-à-dire en effectuant une transformée de Fourier dans la direction tangentielle à l'interface, on obtient la solution locale à Ω_1 comme une combinaison linéaire des vecteurs propres de la matrice $M(k) = B_1^{-1}(\beta Id + ikB_2)$. Comme la solution locale doit être bornée à $-\infty$, on ne garde que les vecteurs propres qui correspondent à des valeurs propres à partie réelle négative. On obtient ainsi :

$$\hat{W}_1(x,k) = \beta_1 V_1(k) e^{-\lambda_1(k)x}$$

où :

$$V_1(k) = \begin{bmatrix} \dfrac{(R(k)+a)(c+u)}{\sqrt{2}} \\ -\dfrac{(R(k)-a)(c-u)}{\sqrt{2}} \\ ik(c^2-u^2) \\ \\ 0 \end{bmatrix}$$

En utilisant la condition en $x=0$, soit $(CW_1(k))_1 = \hat{\Phi}_1$, on en déduit :

$$\hat{W}_1(x,k) = -\frac{\sqrt{2}}{[c+u][R(k)\sigma+a]} V_1(k) e^{-\lambda_1(k)x} \hat{\Phi}_1 \tag{4.44}$$

Pour obtenir un problème local bien posé pour une donnée à l'interface Φ quelconque, il faut démontrer que la solution locale est unique et bornée par une quantité qui dépend de la donnée à l'interface. Compte tenu de l'expression (4.44), il faut tout d'abord imposer que $[R(k)\sigma+a] \neq 0, \forall k$. Cette condition a lieu $\forall \sigma \in \mathbb{R}$ lorsque $v \neq 0$, et pour $\sigma \in]-\infty, -1[\cup]0, +\infty[$ si $v=0$. Dans ces conditions, on s'aperçoit que les composantes de $\hat{W}_1(x,k)$ sont bornées dans le sens de la norme L_∞ pour tout k.

En procédant d'une façon similaire pour le problème posé dans le demi plan droit on obtient :

$$\begin{cases} \beta \bar{W}_2 + B_1 \partial_x \bar{W}_2 + B_2 \partial_y \bar{W}_2 = 0 \quad \text{dans} \quad \Omega_2 =]0,\infty[\times\mathbb{R} \\ (T_1 T \bar{W}_2)_j = (\hat{\Phi}_2, \hat{\Phi}_3, \hat{\Phi}_4)^T \quad \text{pour } j \text{ tel que} \quad \lambda_j(A_1) > 0 \quad \text{et} \quad x=0 \end{cases}$$

Le symbole de Fourier de la solution est dans ce cas donné par :

$$\hat{W}_2(x,k) = \beta_2 V_2(k) e^{-\lambda_2(k)x} + \beta_3 V_3(k) e^{-\lambda_3(k)x} + \beta_4 V_4(k) e^{-\lambda_4(k)x}$$

où :

$$V_2(k) = \begin{bmatrix} \dfrac{(R(k)-a)(c+u)}{\sqrt{2}} \\ -\dfrac{(R(k)+a)(c-u)}{\sqrt{2}} \\ ik(c^2-u^2) \\ \\ 0 \end{bmatrix} \quad V_3(k) = \begin{bmatrix} -\dfrac{iku}{\sqrt{2}} \\ \dfrac{iku}{\sqrt{2}} \\ a \\ \\ 0 \end{bmatrix} \quad V_4(k) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

En utilisant maintenant les conditions en $x=0$ on obtient :

$$\hat{W}_2(x,k) = D_2 V_2(k) e^{-\lambda_2(k)x} + D_3 V_3(k) e^{-\lambda_3(k)x} + D_4 V_4(k) e^{-\lambda_3(k)x} \tag{4.45}$$

où :

$$\begin{cases} D_2 &= \dfrac{-a\sqrt{2}\hat{\Phi}_2 + iku\hat{\Phi}_3}{[ac - R(k)u][R(k)+a]} \\ D_3 &= \dfrac{ik\sqrt{2}[c^2-u^2]\hat{\Phi}_2 + [R(k)+a][c-u]\hat{\Phi}_3}{[ac - R(k)u][R(k)+a]} \\ D_4 &= \dfrac{-2[\alpha-1][au+cR(k)]\hat{\Phi}_2 - iku[\alpha-1][c-u]\sqrt{2})\hat{\Phi}_3 + u[R(k)+a]\hat{\Phi}_4}{u\alpha[R(k)+a]} \end{cases}$$

Dans ce cas, on doit tout d'abord imposer $\alpha \neq 0$. Ensuite il faut étudier le comportement de la solution locale dans le voisinage de la fréquence k_0 qui annule $[ac - uR(k)]$, ce qui correspond au cas ou les vecteurs propres de la matrice $M(k)$ de la section 1.2.2 du chapitre 1.2 ne sont pas linéairement indépendants. Pour cela on voit que $\hat{W}_2(x,k)$ peut s'écrire:

$$\begin{cases} \hat{W}_2(x,k) &= D_2V_2(k)e^{-\lambda_3(k)x}\left[e^{(\lambda_3(k)-\lambda_2(k))x}-1\right] + [D_2V_2(k)+D_3V_3(k)+D_4V_4(k)]\,e^{-\lambda_3(k)x} \\ &= D_2'V_2(k)e^{-\lambda_3(k)x}\left[\dfrac{e^{(\lambda_3(k)-\lambda_2(k))x}-1}{ac-uR(k)}\right] + De^{-\lambda_3(k)x} \\ D &= \begin{pmatrix} -\left[\dfrac{R(k)-a}{R(k)+a}\right] \\ 1 \\ 0 \\ \dfrac{-2[\alpha-1][au+cR(k)]}{u\alpha[R(k)+a]} \end{pmatrix}\hat{\Phi}_2 + \begin{pmatrix} -\dfrac{iku\sqrt{2}}{[R(k)+a]} \\ 0 \\ 1 \\ \dfrac{-iku[c-u][\alpha-1]\sqrt{2}}{u\alpha[R(k)+a]} \end{pmatrix}\hat{\Phi}_3 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \dfrac{1}{\alpha} \end{pmatrix}\hat{\Phi}_4 \end{cases}$$

A ce stade, on constate que les composantes du vecteur D sont bornées dans le sens de la norme L_∞ pour tout k et que :

$$\lim_{k\to k_0} \frac{[e^{(\lambda_3(k)-\lambda_2(k))x}-1]}{[ac-uR(k)]} = u(c^2-u^2)$$

ce qui prouve que $\hat{W}(x,k)$ a des composantes bornées pour tout k sous les hypothèses faites sur les paramètres α et σ, y compris au voisinage de la valeur k_0 où le dénominateur de ces termes s'annule. On peut en déduire que le problème local est donc bien posé.

4.3.2.2 Optimisation du taux de convergence

Dans cette section, on se propose d'optimiser le taux de convergence donné par la formule 4.42, par rapport au paramètres α et σ, sur un intervalle de fréquences $[0\ ,\ k_{max}]$. On se situe d'abord dans le cas où la vitesse tangentielle à l'interface est nulle (donc $M_t = 0$). Dans un premier temps, pour simplifier l'analyse, on peut fixer un des paramètres de façon à vérifier la condition naturelle $\rho(\alpha,\sigma,0) = 0$, ce qui conduit à $\alpha = 1$. Dans ce cas, le taux de convergence prend la forme :

$$\rho^2(\sigma,k) = \left|\frac{[R(k)-a][(c\sigma-u\sigma-2u)R(k)-a(c+u)]}{[c+u][R(k)\sigma+a][R+a]}\right| \tag{4.46}$$

Pour que l'algorithme de type Schwarz converge, c'est-à-dire pour que $\rho(\sigma,k) < 1$, un calcul simple montre que l'on doit avoir $\sigma \in]-\infty, -1[\cup[\dfrac{u}{c}\ ,\ \infty[$. On est ensuite amené à résoudre le problème d'optimisation suivant :

$$\min_{\sigma} \max_{k\in[0,k_{max}]} \rho(\sigma,k) \tag{4.47}$$

Puisque pour $v = 0$, $a = \beta$ est une constante, on peut considérer comme variable d'optimisation $R = \sqrt{a^2 + k^2(c^2 - u^2)} \in [a,\infty[$. Dans ce qui suit, on va démontrer un résultat similaire à celui obtenu dans le cas de l'équation d'advection-diffusion par Japhet[Jap98] sur l'optimisation du taux de convergence de l'algorithme de type Schwarz. Mais tout d'abord, il faut remarquer que lorsque $\sigma = \dfrac{2u}{c-u}$, la limite du taux de convergence s'annule à l'infini pour toute valeur du nombre de Mach normal à l'interface M_n. Ce comportement est illustré sur la figure 4.1 ci-dessous.

FIG. 4.1 – *Taux de convergence asymptotique pour* $\sigma = \dfrac{2u}{c-u}$

Théorème 6 *Il existe une valeur* $\sigma \in \left]\dfrac{2u}{c-u}\ ,\ \dfrac{c+3u}{c-u}\right]$ *qui optimise le taux de convergence, fonction de* $R_{int} \in [a\ ,\ R_{max}]$ *où* $R_{max} = \sqrt{a^2 + k_{max}^2(c^2 - u^2)}$. *De plus,* R_{int} *représente une valeur intermédiaire de* R *qui annule le taux de convergence, c'est-à-dire* $\rho(R_{int},\sigma(R_{int})) = 0$, *et* $R_s(R_{int})$ *correspond à un point de maximum local pour le taux de convergence, c'est-à-dire* $\dfrac{d\rho_R(R_s,\sigma(R_{int}))}{dR} = 0$.

Le problème d'optimisation 4.47 est alors équivalent à :

$$\min_{\mathbb{R}_{int}\in[a\ ,\ R_{max}]} \max_{R\in[a\ ,\ R_{max}]} \rho(\sigma(R_{int}),R) \tag{4.48}$$

le minimum étant atteint pour R_{int} *qui est la solution de l'équation :*

$$\rho(R_s(R_{int}),\sigma(R_{int})) = \rho(R_{max},\sigma(R_{int})) \tag{4.49}$$

Démonstration. Si $\sigma \in \left[\dfrac{2u}{c-u}\ ,\ \dfrac{c+3u}{c-u}\right]$, on voit que l'équation $\rho(R,\sigma) = 0$ admet une solution $R_{int} = \dfrac{a(c+u)}{c\sigma - u\sigma - 2u} > a$. On peut ensuite exprimer le paramètre σ comme :

$$\sigma(R_{int}) = \frac{2R_{int}u + a(c+u)}{R_{int}(c-u)} \tag{4.50}$$

On cherche les possibles points de maximum locaux solutions de l'équation $\dfrac{d\rho_R(R,\sigma(R_{int}))}{dR} = 0$ ce qui conduit à :

$$R_s(R_{int}) = a\frac{R_{int}u + \sqrt{R_{int}(c+u)(ac+R_{int}u)}}{R_{int}u + a(c+u)}$$

On montre facilement que $a \leq R_s(R_{int}) \leq R_{int}$. Cette assertion combinée à l'expression du taux de convergence et de sa dérivée montrent que $\rho(R,R_{int})$ est une fonction croissante sur $[0\ ,\ R_s(R_{int})] \cup [R_{int}\ ,\ R_{max}]$ et décroissante sur $[R_s(R_{int})\ ,\ R_{int}]$ donc $R_s(R_{int})$ est bien un point de maximum local. Ceci veut dire que :

$$\max_{R\in[a,R_{max}]} \rho(R,R_{int}) = \max\{\rho(R_s(R_{int}),R_{int})\ ,\ \rho(R_{max},R_{int})\}$$

En évaluant ρ dans ce point de maximum on obtient :

$$\rho(R_s(R_{int}),R_{int}) = (c-u)\left(\frac{\sqrt{R_{int}(ac+R_{int}u)} - a\sqrt{c+u}}{2R_{int}u + a(c+u)}\right)^2$$

qui est une fonction croissante de R_{int} tandis que $\rho(R_{max},R_{int})$ est décroissante par rapport à la même variable, ce qui nous conduit à la conclusion :

$$\max\{\rho(R_s(R_{int}),R_{int})\ ,\ \rho(R_{max},R_{int})\} = \begin{cases} \rho(R_{max},R_{int}), R_{int} & \leq\ R_0 \\ \rho(R_s(R_{int}),R_{int}), R_{int} & >\ R_0 \end{cases}$$

où R_0 est la solution de l'équation $\rho(R_s(R_{int}),R_{int}) = \rho(R_{max},R_{int})$ (cette solution existe car, si on note $g(R_{int}) = \rho(R_s(R_{int}),R_{,}int) - \rho(R_{max},R_{int})$, on a $g(a) < 0$ et $g(R_{max}) > 0$ et les fonctions sont continues). Donc, dû à la monotonie de ces fonctions, le minimum :

$$\min_{R_{int}\in[a\ ,\ R_{max}]} \max\{\rho(R_s(R_{int})\ ,\ R_{int}),\rho(R_{max},R_{int})\}$$

est atteint en R_0. La valeur optimale de σ est obtenue en remplaçant R_0 dans la formule 4.50.

Notons ρ_{class} le taux de convergence (4.21) associé aux conditions de raccord classiques. De même, ρ_{opt} désigne le taux de convergence (4.46) associé aux conditions de raccord optimisées (c'est-à-dire pour la valeur de σ déduite du théorème 6).

Les figures 4.2 à 4.5 représentent les évolutions des taux de convergence ρ_{class} et ρ_{opt} pour différentes valeurs du nombre de Mach M_n normal à l'interface. On peut conclure cette analyse par la remarque suivante.

Remarque 12 Si le nombre de Mach normal à l'interface est inférieur à $\dfrac{1}{3}$ alors ρ_{class} s'annule déjà pour une fréquence k_0 (voir la section 1.2.2 du chapitre 1.2) et il est donc plus difficile d'optimiser

dans le domaine des basses fréquences en obtenant un gain important. Par contre, comme pour les hautes fréquences, le taux de convergence ρ_{class} se dégrade, on peut observer un gain plus sensible après optimisation. Si $M_n = \dfrac{1}{3}$, on peut produire des améliorations dans le domaine des basses fréquences mais, pour les hautes fréquences, comme $\lim\limits_{k\to\infty} \rho_{\text{class}}(k) = 0$ on ne peut évidemment pas faire mieux que les conditions de raccord classiques. Pour ce qui concerne le cas $M_n > \dfrac{1}{3}$, le gain est clair pour toutes les plages de fréquences car le taux de convergence associé aux conditions de raccord classiques est croissant et il atteint rapidement sa valeur asymptotique maximale.

FIG. 4.2 – *Comparaison des taux de convergence classique et optimisé :* $M_n = 0.1$

Dans le cas où v est non nul à l'interface, on ne peut plus adopter la même stratégie car, avec un seul paramètre, on ne peut plus annuler le taux de convergence. Si on arrive à le faire avec deux paramètres alors on perd le bon comportement autour de $k = 0$. On peut néanmoins trouver une valeur de σ comme solution de l'équation :

$$\rho(k_s,\sigma) = \rho(k_{max},\sigma)$$

où k_s est un point de maximum local (quand celui-ci existe), ou minimiser directement $\rho(k_{max},\sigma)$ dans le cas où ρ est strictement monotone par rapport à k, ce qui se passe notamment quand M_t devient important par rapport à M_n (voir les figures 4.6 à 4.11 qui suivent). D'un point de vue pratique, pour des grandes fréquences, on peut utiliser la valeur de σ qui minimise $\lim\limits_{k\to\infty} \rho(\sigma,k)$:

$$\sigma_{\text{opt}} = \frac{M_n + \sqrt{M_n^2 + M_t^2}}{1 - \sqrt{M_n^2 + M_t^2}} \tag{4.51}$$

qui correspond aux figures 4.6 à 4.11. Dans ce cas on distingue à nouveau plusieurs cas. On s'aperçoit que si M_t est comparable à M_n alors le gain est moindre dans le voisinage de $\dfrac{1}{3}$ que dans les autres cas.

FIG. 4.3 – *Comparaison des taux de convergence classique et optimisé :* $M_n = 0.3$

FIG. 4.4 – *Comparaison des taux de convergence classique et optimisé :* $M_n = 0.6$

FIG. 4.5 – *Comparaison des taux de convergence classique et optimisé :* $M_n = 0.8$

Si par contre M_t devient important par rapport à M_n alors l'écart entre les deux types de condition tend à diminuer si $M_n < \frac{1}{3}$ en restant important pour $M_n > \frac{1}{3}$.

4.3.3 Algorithme de résolution en discret

Pour simplifier la présentation, on considère de nouveau le cas d'une décomposition du domaine Ω en deux sous-domaines. Soit $[s_i,s_j]$ un segment tel que la cellule de contrôle C_i (associée au sommet s_i) et la cellule de contrôle C_j (associée au sommet s_j) appartiennent respectivement à Ω_1 et Ω_2. On rappelle que l'interface est définie par (voir aussi la figure 2.3 page 63 du chapitre 2) :

$$\Gamma = \Omega_1 \cap \Omega_2 = \bigcup_{C_{1_k} \in \Omega_1, C_{2_k} \in \Omega_2} \partial C_{1_k} \cap \partial C_{2_k}$$

Si on définit W^* comme étant l'inconnue définie à l'interface entre les deux sous-domaines, plus précisément au milieu du segment $[s_i,s_j]$, le système linéaire à résoudre s'écrit sous la forme :

$$\begin{pmatrix} \mathcal{M}_1 & 0 & \mathcal{M}_{12} \\ 0 & \mathcal{M}_2 & \mathcal{M}_{21} \\ \mathcal{F}_{1B} & \mathcal{F}_{2B} & \mathcal{F}_{BB} \end{pmatrix} \begin{pmatrix} W_1 \\ W_2 \\ W^* \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ 0 \end{pmatrix} \tag{4.52}$$

où $\mathcal{M}_1$ (respectivement $\mathcal{M}_2$) est la matrice implicite qui couple les sommets internes à Ω_1 (respectivement Ω_2) tandis que $\mathcal{F}_{1B}$, $\mathcal{F}_{2B}$, $\mathcal{M}_{12}$ et $\mathcal{M}_{21}$ sont des matrices de couplage entre les inconnues internes et les inconnues à l'interface Γ.

Dans ce qui suit on décide de dupliquer l'inconnue interface W^* en W_1^* et W_2^*. On est alors amené à considérer la nouvelle forme du vecteur d'inconnues donnée par $(W_1\ ,\ W_2\ ,\ W_1^*\ ,\ W_2^*)^T$. On peut ré-écrire le système précédent en séparant le bloc $\mathcal{F}_{BB}$ en $\mathcal{F}_{1BB}$ et $\mathcal{F}_{2BB}$ (ceci correspond à une décomposition d'un bloc en partie négative et positive du type $A_{\mathbf{n}} = A_{\mathbf{n}}^+ + A_{\mathbf{n}}^-$) :

FIG. 4.6 – *Comparaison des taux de convergence classique et optimisé :* $M_n = 0.1$, $M_t = 0.1$

FIG. 4.7 – *Comparaison des taux de convergence classique et optimisé :* $M_n = 0.1$, $M_t = 0.2$

FIG. 4.8 – *Comparaison des taux de convergence classique et optimisé :* $M_n = 0.3$, $M_t = 0.3$

FIG. 4.9 – *Comparaison des taux de convergence classique et optimisé :* $M_n = 0.3$, $M_t = 0.5$

FIG. 4.10 – *Comparaison des taux de convergence classique et optimisé :* $M_n = 0.6$, $M_t = 0.6$

FIG. 4.11 – *Comparaison des taux de convergence classique et optimisé :* $M_n = 0.8$, $M_t = 0.5$

$$\begin{pmatrix} \mathcal{M}_1 & \mathcal{M}_{12} & 0 & 0 \\ \mathcal{F}_{1B} & \mathcal{F}_{1BB} & \mathcal{F}_{2B} & \mathcal{F}_{2BB} \\ 0 & 0 & \mathcal{M}_2 & \mathcal{M}_{21} \\ 0 & -\mathrm{Id} & 0 & \mathrm{Id} \end{pmatrix} \begin{pmatrix} W_1 \\ W_1^* \\ W_2 \\ W_2^* \end{pmatrix} = \begin{pmatrix} b_1 \\ 0 \\ b_2 \\ 0 \end{pmatrix} \tag{4.53}$$

La dernière ligne du système ci-dessus, qui exprime en fait la continuité de W^* à travers l'interface, peut-être ré-écrite à l'aide de deux lignes qui représentent des conditions de flux entrant dans chaque sous-domaine, en accord avec la nature hyperbolique du système d'équations d'Euler. Il s'en suit :

$$\begin{pmatrix} \mathcal{M}_1 & \mathcal{M}_{12} & 0 & 0 \\ \mathcal{F}_{1B} & \mathcal{F}_{1BB} & \mathcal{F}_{2B} & \mathcal{F}_{2BB} \\ 0 & 0 & \mathcal{M}_2 & \mathcal{M}_{21} \\ 0 & -C_1 & 0 & C_1 \\ 0 & C_2 & 0 & -C_2 \end{pmatrix} \begin{pmatrix} W_1 \\ W_1^* \\ W_2 \\ W_2^* \end{pmatrix} = \begin{pmatrix} b_1 \\ 0 \\ b_2 \\ 0 \\ 0 \end{pmatrix} \tag{4.54}$$

où, dans le cas des conditions de raccord classiques, $C_1 = A_{\mathbf{n}}^-$ et $C_2 = A_{\mathbf{n}}^+$. Suivant une démarche adoptée par Nataf[Nat01], les opérateurs C_1 et C_2 peuvent être associées à des conditions plus générales, notées $\bar{C}_i(\alpha_1,\alpha_2,\sigma,\delta)$, , à l'instar de celles proposées dans la section précédente. On s'intéresse dans ce qui suit à la prise en compte de ces conditions dans ce type de formulation. Pour cela on additionne la deuxième ligne du système ci-dessus à la quatrième puis à la cinquième ligne. On obtient alors :

$$\begin{pmatrix} \mathcal{M}_1 & \mathcal{M}_{12} & 0 & 0 \\ \mathcal{F}_{1B} & \mathcal{F}_{1BB} - \bar{C}_1 & \mathcal{F}_{2B} & \mathcal{F}_{2BB} + \bar{C}_1 \\ 0 & 0 & \mathcal{M}_2 & \mathcal{M}_{21} \\ \mathcal{F}_{1B} & \mathcal{F}_{1BB} + \bar{C}_2 & \mathcal{F}_{2B} & \mathcal{F}_{2BB} - \bar{C}_2 \end{pmatrix} \begin{pmatrix} W_1 \\ W_1^* \\ W_2 \\ W_2^* \end{pmatrix} = \begin{pmatrix} b_1 \\ 0 \\ b_2 \\ 0 \end{pmatrix} \tag{4.55}$$

A ce stade, il est classique de reformuler l'algorithme de type Schwarz additif comme une méthode de Jacobi par bloc :

$$\begin{aligned} &\begin{pmatrix} \mathcal{M}_1 & \mathcal{M}_{12} & 0 & 0 \\ \mathcal{F}_{1B} & \mathcal{F}_{1BB} - \bar{C}_1 & 0 & 0 \\ 0 & 0 & \mathcal{M}_2 & \mathcal{M}_{21} \\ 0 & 0 & \mathcal{F}_{2B} & \mathcal{F}_{2BB} - \bar{C}_2 \end{pmatrix} \begin{pmatrix} W_1 \\ W_1^* \\ W_2 \\ W_2^* \end{pmatrix}^{(p+1)} = \\ &\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -\mathcal{F}_{2B} & -\mathcal{F}_{2BB} - \bar{C}_1 \\ 0 & 0 & 0 & 0 \\ -\mathcal{F}_{1B} & -\mathcal{F}_{1BB} - \bar{C}_2 & 0 & 0 \end{pmatrix} \begin{pmatrix} W_1 \\ W_1^* \\ W_2 \\ W_2^* \end{pmatrix}^{(p)} + \begin{pmatrix} b_1 \\ 0 \\ b_2 \\ 0 \end{pmatrix} \end{aligned} \tag{4.56}$$

Si on note par A_D la matrice du membre de gauche et par A_E celle du membre de droite, l'itération de Jacobi s'écrit :

$$\begin{pmatrix} W_1 \\ W_1^* \\ W_2 \\ W_2^* \end{pmatrix}^{(p+1)} = A_D^{-1} A_E \begin{pmatrix} W_1 \\ W_1^* \\ W_2 \\ W_2^* \end{pmatrix}^{(p)} + A_D^{-1} \begin{pmatrix} b_1 \\ 0 \\ b_2 \\ 0 \end{pmatrix} \tag{4.57}$$

Afin d'aboutir à une itération qui ne porte que sur les inconnues à l'interface, on introduit la matrice suivante :

$$\Pi = \begin{pmatrix} 0 & 0 & -\mathcal{F}_{2B} & -\mathcal{F}_{2BB} - \bar{C}_1 \\ -\mathcal{F}_{1B} & -\mathcal{F}_{1BB} - \bar{C}_2 & 0 & 0 \end{pmatrix}$$

On peut alors écrire :

$$\begin{pmatrix} W_1 \\ W_1^* \\ W_2 \\ W_2^* \end{pmatrix}^{(p+1)} = A_D^{-1} \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \Pi \begin{pmatrix} W_1 \\ W_1^* \\ W_2 \\ W_2^* \end{pmatrix}^{(p)} + A_D^{-1} \begin{pmatrix} b_1 \\ 0 \\ b_2 \\ 0 \end{pmatrix} \tag{4.58}$$

On multiplie l'équation précédente par Π et on note :

$$\Phi = \Pi \begin{pmatrix} W_1 \\ W_1^* \\ W_2 \\ W_2^* \end{pmatrix}$$

le nouveau vecteur d'inconnues à l'interface qui doit satisfaire l'équation :

$$\left[\mathsf{Id} - A_D^{-1} \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \right] \Phi = \Pi A_D^{-1} \begin{pmatrix} b_1 \\ 0 \\ b_2 \\ 0 \end{pmatrix} \tag{4.59}$$

On peut maintenant détailler une nouvelle forme de l'algorithme de type Schwarz additif appliqué à la résolution du système d'équations d'Euler.

`ALGORITHME 9` Algorithme de type Schwarz additif en variables physiques pour la prise en compte de conditions de transmission générales.

Soit $W_i^{(0)}$ pour $i = 1{,}2$ une approximation initiale de la solution des équations d'Euler au pas de temps $n\Delta t$ dans le sous-domaine Ω_i. On suppose aussi données les valeurs initiales de $\Phi_i^{(0)}$ pour $i = 1{,}2$. Les nouvelles solutions $W_i^{(p+1)}$ et $\Phi_i^{(p+1)}$ pour $i = 1{,}2$ à l'itération $p+1$ sont obtenues par les étapes suivantes.

– Résolution des problèmes locaux :

$$\begin{cases} M_1 W_1^{(p+1)} + M_{12}(W_1^*)^{(p+1)} = b_1 \\ (\mathcal{F}_{1BB} - \bar{C}_1)(W_1^*)^{(p+1)} + \mathcal{F}_{1B} W_1^{(p+1)} + \Phi_1^{(p)} = 0 \end{cases}$$
$$\begin{cases} M_2 W_2^{(p+1)} + M_{21}(W_2^*)^{(p+1)} = b_2 \\ (\mathcal{F}_{2BB} + \bar{C}_2)(W_2^*)^{(p+1)} + \mathcal{F}_{2B} W_2^{(p+1)} + \Phi_2^{(p)} = 0 \end{cases} \tag{4.60}$$

– Calcul et échanges des valeurs de flux à l'interface :

$$\begin{cases} \Phi_1^{(p+1)} &= (\mathcal{F}_{2BB} - \bar{C}_2)(W_2^*)^{(p)} + \mathcal{F}_{2B} W_2^{(p)} \\ \Phi_2^{(p+1)} &= (\mathcal{F}_{1BB} + \bar{C}_1)(W_1^*)^{(p)} + \mathcal{F}_{1B} W_1^{(p)} \end{cases} \tag{4.61}$$

Dans le cas du système d'équations d'Euler, les matrices qui interviennent dans ces itérations sont telles que :

$$\begin{cases} \mathcal{F}_{1BB} & \equiv & A_{\mathbf{n}}^{+} \\ \mathcal{F}_{2BB} & \equiv & -A_{\mathbf{n}}^{-} \\ \mathcal{F}_{1B} & \equiv & -A_{\mathbf{n}}^{+} \\ \mathcal{F}_{2B} & \equiv & A_{\mathbf{n}}^{-} \end{cases}$$

Les nouvelles conditions de transmission sont données par $\bar{C}_1 = A_{\mathbf{n}}^{-}$ et $\bar{C}_2 = A_{\mathbf{n}}^{+}$ avec $A_{\mathbf{n}} = T_1^{-1}\Lambda T_1$; dans ces expressions, T_1 désigne la matrice définissant nouvelles conditions de raccord (voir aussi l'expression de $\hat{T}_1$ (4.41) page 130).

Dans l'hypothèse de solutions régulières du système d'équations d'Euler, on peut aussi utiliser :

$$\begin{cases} \mathcal{F}_{1BB} & \equiv & \Lambda^{+}T^{-1} \\ \mathcal{F}_{2BB} & \equiv & -\Lambda^{-}T^{-1} \\ \mathcal{F}_{1B} & \equiv & -\Lambda^{+}T^{-1} \\ \mathcal{F}_{2B} & \equiv & \Lambda^{-}T^{-1} \end{cases}$$

les nouvelles conditions de raccord s'exprimant comme $\bar{C}_1 = \Lambda^{-}\bar{T}_1$ et $\bar{C}_2 = \Lambda^{+}\bar{T}_1$.

4.4 Résultats numériques

Dans cette section, on se situe dans le cadre de la méthodologie numérique décrite au chapitre 2 (voir notamment les méthodes d'approximation en espace et d'intégration en temps décrites dans les sections 2.1.2 et 2.1.4, ainsi que la stratégie de mise en œuvre de la méthode de décomposition de domaine décrite dans la section 2.2). Dans ce qui suit, on cherche avant tout à valider numériquement l'analyse faite dans la section 4.3.2.2, l'objectif étant de mettre en évidence la supériorité des conditions d'interface optimisées. Le système interface discret (4.59) est résolu par un algorithme de Jacobi qui est équivalent à l'algorithme de type Schwarz additif 9 page 143.

On s'intéresse à la résolution du système linéaire issu du premier pas de temps pour plusieurs régimes d'écoulement correspondant à des nombres de Mach de référence compris entre 0.1 et 0.8 et pour une décomposition en deux sous-domaines d'un maillage triangulaire structuré contenant 200×20 points (le domaine de calcul est le rectangle $[0,2] \times [0,1]$). Les résultats obtenus sont résumés sur les figures ci-dessous qui comparent le comportement des deux types de condition en termes du nombre d'itérations de Jacobi nécessaires pour réduire le résidu linéaire (normalisé à sa valeur initiale) d'un facteur $\varepsilon_i = 10^{10}$.

Dans un premier temps, on teste le cas d'une vitesse normale à l'interface (c'est-à-dire que l'on évalue les conditions d'interface optimisées obtenues en supposant que $v = 0$ à l'interface) pour quatre valeurs du nombre de Mach global $M = 0.1$, 0.3 , 0.6 et 0.8 (voir les figures 4.12, 4.13, 4.14), dont l'estimation théorique du taux a été faite dans la section 4.3.2.2 (voir les figures 4.2 à 4.5). On note que l'on retrouve le comportement qualitatif estimé analytiquement avec la réserve que dans le cas où le gain théorique est petit, à cause des approximations numériques, celui-ci peut ne pas être retrouvé numériquement. De

plus, l'approximation spatiale par volumes finis fournit en discret une forme de l'interface qui n'est pas rectiligne (voir la figure 2.3 page 63) ce qui nous éloigne un peu du cadre de l'analyse théorique de la section 4.3.2.2.

FIG. 4.12 – *Comparaison de la convergence: conditions classiques et optimisés :* $M_n = 0.1$
Axe X: nombre d'itérations, Axe Y: résidu linéaire

On teste ensuite le cas où la vitesse n'est plus supposée normale à l'interface (c'est-à-dire que l'on évalue les conditions d'interface optimisées déduites de la valeur (4.51)). Les courbes de référence sont celles représentées pour des valeurs de $M_t \neq 0$ dans les figures 4.6 à 4.11. On voit sur les courbes de convergence de l'algorithme de Schwarz en discret (figures 4.15 à 4.20) que les estimations théoriques sont tout à fait confirmées dans le sens où l'on obtient un gain important pour les grandes valeurs du nombre de Mach de référence de l'écoulement et moindre pour celles inférieures à $\frac{1}{3}$.

4.5 Conclusion

Dans ce chapitre, on s'est intéressé à l'accélération de la convergence de l'algorithme de type Schwarz additif sans recouvrement de la forme (1.20) en modifiant les conditions de raccord appliquées aux interfaces entre sous-domaines voisins. On a ainsi introduit des conditions de raccord généralisées qui ont été construites en utilisant la théorie de Smith pour la diagonalisation des matrices polynomiales[Gan65] et plus précisément obtenues par une série de transformations algébriques; elles sont au plus du premier ordre en dérivée partielle dans la direction tangentielle à l'interface et dépendent de quelques paramètres qui, s'ils sont convenablement choisis, permettent une meilleure convergence de l'algorithme de type Schwarz additif. Ici, on a étudié plus en détails des conditions d'ordre zéro (c'est-à-dire sans dérivée dans la direction tangentielle à l'interface) et on a proposé des conditions d'interface « optimisées » dont la mise en œuvre est particulièrement aisée.

Les expériences numériques présentées dans ce chapitre ont permis de retrouver qualitativement

FIG. 4.13 – *Comparaison de la convergence: conditions classiques et optimisés :* $M_n = 0.3$
Axe X: nombre d'itérations, Axe Y: résidu linéaire

FIG. 4.14 – *Comparaison de la convergence: conditions classiques et optimisés :* $M_n = 0.6$
Axe X: nombre d'itérations, Axe Y: résidu linéaire

FIG. 4.15 – *Comparaison de la convergence: conditions classiques et optimisés :* $M_n = 0.1$, $M_t = 0.1$
Axe X: nombre d'itérations, Axe Y: résidu linéaire

FIG. 4.16 – *Comparaison de la convergence: conditions classiques et optimisés :* $M_n = 0.3$, $M_t = 0.2$
Axe X: nombre d'itérations, Axe Y: résidu linéaire

FIG. 4.17 – *Comparaison de la convergence: conditions classiques et optimisés :* $M_n = 0.6$, $M_t = 0.4$
Axe X: nombre d'itérations, Axe Y: résidu linéaire

FIG. 4.18 – *Comparaison de la convergence: conditions classiques et optimisés :* $M_n = 0.6$, $M_t = 0.6$
Axe X: nombre d'itérations, Axe Y: résidu linéaire

FIG. 4.19 – *Comparaison de la convergence: conditions classiques et optimisés :* $M_n = 0.6$, $M_t = 0.7$
Axe X: nombre d'itérations, Axe Y: résidu linéaire

FIG. 4.20 – *Comparaison de la convergence: conditions classiques et optimisés :* $M_n = 0.8$, $M_t = 0.5$
Axe X: nombre d'itérations, Axe Y: résidu linéaire

un comportement en accord avec la théorie même si, d'un point de vue discret, on n'a pas reproduit identiquement les résultats obtenus en continu. Il y au moins deux explications possibles aux variations observées entre la théorie et la pratique des conditions de raccord optimisées :

- l'approximation spatiale par volumes finis fournit en discret une forme de l'interface qui n'est pas rectiligne (voir la figure 2.3 page 63). De ce fait, on ne peut par exemple pas garantir que le nombre de Mach tangentiel est nul lorsque la composante v de la vitesse de référence de l'écoulement est nulle. Une stratégie possible pour corriger cet écart avec le contexte théorique consisterait à considérer une décomposition sans recouvrement du maillage triangulaire initial[Lan96];
- l'analyse théorique des conditions d'interface généralisées n'a été menée que sur le système d'équations d'Euler intégrées en temps mais pas en espace (analyse en continu). Clairement, une analyse en discret s'impose pour mieux affiner les gains entre conditions d'interface « classiques » et « optimisées ».

Les résultats préliminaires obtenus ici sont néanmoins encourageants puisqu'ils conduisent à une très bonne convergence de l'algorithme de type Schwarz additif sans recouvrement pour certains régimes d'écoulement.

Deuxième partie

Décomposition de domaine pour le système d'équations de Navier-Stokes

Chapitre 5

Calcul d'écoulements stationnaires

Dans ce chapitre, on présente une extension de la méthode de résolution par sous-domaine du système d'équations d'Euler proposée au chapitre 1, dans le but de permettre le traitement du système mixte hyperbolique/parabolique des équations de Navier-Stokes. Comme pour la résolution du système d'équations d'Euler, la méthode en question repose sur la formulation d'un algorithme de type Schwarz additif sur une décomposition sans recouvrement du domaine de calcul. Les conditions imposées aux interfaces entre sous-domaines voisins expriment, là encore, la continuité de flux normaux. Cependant, dans le cas présent, cette formulation doit être adaptée au fait que le vecteur flux mis en jeu combine des termes de flux convectif et des termes de flux diffusif (ou flux visqueux).

Dans un premier temps (section 5.1), on décrit les caractéristiques de la méthodologie de départ, utilisée pour la simulation numérique d'écoulements de fluides compressibles visqueux en régime laminaire. Puis, dans la section 5.2, on formule l'algorithme de résolution par sous-domaine en continu avant de présenter son adaptation au contexte discret de la section 5.1. La section 5.3 est consacrée à l'évaluation de la méthode de résolution par sous-domaine introduite dans la section 5.2 dans le cadre du calcul de plusieurs écoulements stationnaires autour d'un profil d'aile `NACA0012`. Enfin, dans la section 6, on s'intéresse à la résolution d'écoulements instationnaires. Pour ce faire, on introduit au préalable un schéma d'intégration en temps implicite précis au second ordre qui est particulièrement bien adapté à la méthode de résolution par sous-domaine qui fait l'objet de ce chapitre.

L'adaptation de cette formulation au contexte discret, décrit dans la section 5.1, n'est pas immédiate. Plus précisément, la méthode de discrétisation mixte éléments finis/volumes finis en maillages triangulaires ne facilite pas la prise en compte de la condition de continuité des flux normaux aux interfaces. Dans l'approche considérée ici, les termes de flux convectif et diffusif sont traités séparément; pour le flux convectif, on préserve la stratégie utilisée pour la résolution du système d'équations d'Euler (section 2.3), alors que, dans un souci de simplification, on a choisi d'imposer à la fois la continuité des composantes normales et tangentielles du flux diffusif.

5.1 Résolution numérique du système d'équations de Navier-Stokes

Dans cette section, on commence par rappeler le système d'équations aux dérivées partielles qui modélise l'écoulement d'un fluide visqueux en régime laminaire. On aborde ensuite les aspects liés à la résolution numérique de ces équations. La méthodologie adoptée est une extension de celle utilisée pour la résolution du système d'équations d'Euler (voir notamment les sections 2.1.2 et 2.1.4 de la partie I); on se contente donc de détailler les ingrédients liés à la prise en compte du terme de flux diffusif. Comme

pour la résolution du système d'équations d'Euler, la méthodologie numérique décrite ici sert de point de départ (plus précisément, elle correspond à une stratégie de résolution globale) au développement de la stratégie de résolution par sous-domaine qui fait l'objet de ce chapitre.

5.1.1 Modèle mathématique

Soit $\Omega \subset \mathbb{R}^2$ le domaine de calcul et Γ sa frontière. On rappelle que Γ est décomposée en une frontière de type paroi solide Γ_w et une frontière à l'infini Γ_∞ : $\Gamma = \Gamma_w \cup \Gamma_\infty$. Soit $\vec{n}$ la normale sortante en tout point de Γ. Les conditions aux limites appliquées sur Γ, ainsi que leur prise en compte dans la résolution numérique, sont discutées dans la section 5.1.3. La forme conservative du système d'équations de Navier-Stokes qui modélise l'écoulement d'un fluide compressible visqueux en régime laminaire est donnée par :

$$\frac{\partial W}{\partial t} + \frac{\partial F_x(W)}{\partial x} + \frac{\partial F_y(W)}{\partial y} = \frac{1}{\mathsf{Re}}\left(\frac{\partial R_x(W)}{\partial x} + \frac{\partial R_y(W)}{\partial y}\right) \quad , \quad W = \left(\rho \ , \ \rho\vec{U} \ , \ E\right)^T \tag{5.1}$$

où $W = W(\vec{x},t)$; $\vec{x}$ et t désignent respectivement les variables spatiale et temporelle tandis que $\vec{\mathcal{F}}(W) = (F_x(W) \ , \ F_y(W))^T$ est le flux convectif dont les composantes sont données par :

$$F_x(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(E+p) \end{pmatrix} \quad , \quad F_y(W) = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(E+p) \end{pmatrix} \tag{5.2}$$

Dans les expressions ci-dessus, ρ désigne la masse volumique (ou densité), $\vec{U} = (u \ , \ v)^T$ le vecteur vitesse, E l'énergie totale par unité de volume et p la pression. Par ailleurs, $\vec{\mathcal{R}}(W) = (R_x(W) \ , \ R_y(W))^T$ désigne le flux diffusif (ou flux visqueux) dont les composantes sont données par :

$$R_x\,(W) = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\,\tau_{xx} + v\,\tau_{xy} + \dfrac{\gamma}{\mathsf{Pr}}\dfrac{\partial e}{\partial x} \end{pmatrix} \quad , \quad R_y\,(W) = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\,\tau_{xy} + v\,\tau_{yy} + \dfrac{\gamma}{\mathsf{Pr}}\dfrac{\partial e}{\partial y} \end{pmatrix} \tag{5.3}$$

La pression p et l'énergie interne spécifique sont déduites des autres variables par les lois d'état suivantes :

$$\begin{cases} p &= (\gamma - 1)(E - \dfrac{1}{2}\rho \parallel \vec{U} \parallel^2) \\ e &= \dfrac{E}{\rho} - \dfrac{1}{2}(u^2 + v^2) = C_v T = \dfrac{p}{\rho(\gamma - 1)} \end{cases}$$

où T désigne la température; γ est le rapport des chaleurs spécifiques ($\gamma = \dfrac{C_v}{C_p} = 1.4$ pour un gaz diatomique). Dans les expressions des termes de flux diffusif, τ_{xx},τ_{xy} et τ_{yy} désignent les composantes du tenseur des contraintes de Cauchy :

$$\left\{\begin{array}{lcl} \tau_{xx} & = & \dfrac{2}{3}\left(2\dfrac{\partial u}{\partial x}-\dfrac{\partial v}{\partial y}\right) \\ \tau_{yy} & = & \dfrac{2}{3}\left(2\dfrac{\partial v}{\partial y}-\dfrac{\partial u}{\partial x}\right) \\ \tau_{xy} & = & \left(\dfrac{\partial u}{\partial y}+\dfrac{\partial v}{\partial x}\right) \end{array}\right.$$

Deux nombres sans dimension apparaissent dans les équations ci-dessus, le nombre de Reynolds et le nombre de Prandtl (qui ne fait intervenir que des caractéristiques physiques du fluide), dont les expressions sont données par :

$$\left\{\begin{array}{lcl} \mathsf{Re} & = & \dfrac{\rho_0 U_0 L_0}{\mu} \\ \mathsf{Pr} & = & \dfrac{\mu C_p}{k} \end{array}\right.$$

où μ et k sont respectivement des coefficients de viscosité et de conductivité thermique et où l'indice 0 est utilisé pour identifier des quantités caractéristiques de l'écoulement.

5.1.2 Méthode de discrétisation en espace

Le domaine de calcul Ω est discrétisé par une triangulation $\mathcal{T}_h$. La méthode d'approximation en espace est basée sur les éléments suivants :

- une formulation volumes finis combinée à des schémas décentrés pour le calcul des flux convectifs (voir les équations (2.3) à (2.9) de la section 2 partie I);
- une technique MUSCL pour l'extension au second ordre du calcul du flux convectif (voir les équations (2.11) et (2.12) de la section 2 partie I);
- une formulation éléments finis classique (élément P1 de type Lagrange) aboutissant à un schéma centré pour le calcul du flux diffusif.

Une formulation variationnelle générale associée à l'équation (5.1) s'écrit comme :

$$\iint_{S_i}\left(W_t+\frac{\partial F_x(W)}{\partial x}+\frac{\partial F_y(W)}{\partial y}\right)\Phi_i d\vec{x}=\frac{1}{\mathsf{Re}}\iint_{S_i}\left(\frac{\partial R_x(W)}{\partial x}+\frac{\partial R_y(W)}{\partial y}\right)\Phi_i d\vec{x} \qquad (5.4)$$

où S_i est le support de définition de la fonction test Φ_i. Cette fonction test est ici choisie différemment suivant le flux considéré :

- pour le terme de flux convectif, la fonction test est donnée par la fonction caractéristique associée à la cellule de contrôle C_i :

$$\Phi_i=\chi\left(C_i\right)\ ,\ \ S_i=C_i$$

- pour le terme de flux diffusif, la fonction test est donnée par la fonction de base P1 (linéaire et continue sur le support S_i) associée au sommet s_i :

$$\Phi_i=\phi_i \ \text{ et } \ S_i=\bigcup_{\tau\ ,\ s_i\in\tau}\tau\equiv K(s_i) \ \text{ avec } \ \phi_i\left(s_j\right)=\delta_{ij}$$

On déduit alors de l'équation (5.4) :

– pour le terme de flux convectif :

$$\begin{aligned}\iint_{C_i}\left(\frac{\partial F(W)}{\partial x}+\frac{\partial G(W)}{\partial y}\right)\chi_i d\vec{x} &= \int_{\partial C_i\cap\Gamma}\left(F_x\left(W\right)\eta_x+F_y\left(W\right)\eta_y\right)\chi_i dl \\ &- \iint_{C_i}\left(F_x\left(W\right)\frac{\partial\chi_i}{\partial x}+F_y\left(W\right)\frac{\partial\chi_i}{\partial y}\right)d\vec{x}\end{aligned} \tag{5.5}$$

Dans l'expression (5.5), le second terme du membre de droite est égal à zéro puisque la fonction χ_i est constante sur C_i;

– pour le terme de flux diffusif :

$$\begin{aligned}\iint_{K(s_i)}\frac{1}{\mathrm{Re}}\left(\frac{\partial R(W)}{\partial x}+\frac{\partial S(W)}{\partial y}\right)\phi_i d\vec{x} &= \frac{1}{\mathrm{Re}}\int_{K(s_i)\cap\Gamma}\left(R_x\left(W\right)n_x+R_y\left(W\right)n_y\right)\phi_i dl \\ &- \frac{1}{\mathrm{Re}}\iint_{K(s_i)}\left(R_x\left(W\right)\frac{\partial\phi_i}{\partial x}+R_y\left(W\right)\frac{\partial\phi_i}{\partial y}\right)d\vec{x}\end{aligned} \tag{5.6}$$

La consistance de la formulation mixte éléments finis/volumes finis introduite ci-dessus a été abordée dans [AD93] et établie rigoureusement pour un problème modèle dans [Mer94]. Par ailleurs, Rostand et Stoufflet[RS88] ont étudié différentes formulations pour l'approximation du système d'équations de Navier-Stokes : une formulation éléments finis de type Galerkin[AD84], une formulation mixte éléments finis/volumes finis centrée[MJ88a] et la formulation adoptée dans cette étude. Les auteurs montrent notamment que ces trois formulations sont équivalentes pour la discrétisation du système d'équations d'Euler linéarisé (c'est-à-dire lorsque les termes de flux convectif sont des fonctions linéaires de W).

On obtient donc l'équation discrète associée au sommet s_i en calculant :

– le flux convectif sur le contour $\partial C_i\cap\Gamma$,
– le flux diffusif sur le support de la fonction de base ϕ_i. Par ailleurs, on ne tient pas compte ici des intégrales de bord sur $\partial C_i\cap\Gamma_w$ et $\partial C_i\cap\Gamma_\infty$ pour des raisons données dans la section 5.1.3.2.

Compte tenu de ce qui précède, l'équation (5.4) devient :

$$\begin{aligned}\iint_{S_i}W_t\Phi_i d\vec{x} &+ \int_{\partial C_i}\vec{\mathcal{F}}(W).\vec{\eta}dl \\ &= -\frac{1}{\mathrm{Re}}\sum_{\tau\in K(s_i)}\iint_{\tau}\left(R_x\left(W\right)\frac{\partial\phi_i}{\partial x}+R_y\left(W\right)\frac{\partial\phi_i}{\partial y}\right)d\vec{x}\end{aligned} \tag{5.7}$$

où $\vec{\mathcal{F}}(W)=\left(F_x(W)\ ,\ F_y(W)\right)^T$.

5.1.2.1 Calcul du terme de flux convectif

Le calcul numérique du terme de flux convectif est identique à ce qui a été adopté pour la résolution du système d'équations d'Euler (voir en particulier les équations 2.3 à 2.12).

5.1.2.2 Calcul du terme de flux diffusif

Le calcul du flux visqueux repose sur une approximation éléments finis P1 des termes correspondants de l'équation (5.7); plus précisément, en supposant que les variables physiques ρ, $\vec{U}$ et p sont linéaires sur un triangle τ, on obtient :

$$\begin{aligned} \iint_{\tau} \left(R_x(W) \frac{\partial \phi_i}{\partial x} + R_y(W) \frac{\partial \phi_i}{\partial y} \right) d\vec{x} &= \text{aire}(\tau) \left(R_x(\tau) \frac{\partial \phi_i}{\partial x}|_\tau + R_y(\tau) \frac{\partial \phi_i}{\partial y}|_\tau \right) \\ &= \Upsilon_{\tau,i}(\tau) \end{aligned} \tag{5.8}$$

où $R_x(\tau)$ et $R_y(\tau)$ sont des vecteurs constants sur le triangle τ calculés en utilisant $\vec{U}(\tau) = \frac{1}{3} \sum_{s_i \in \tau} \vec{U}(s_i)$ (puisque les composantes de $\vec{U}$ sont supposées linéaires sur τ).

5.1.3 Conditions aux limites

5.1.3.1 Conditions à l'infini

Dans cette étude, on se limite à la résolution numérique d'écoulements externes autour d'un obstacle (plus précisément, d'un profil d'aile). Dans chaque cas, le maillage utilisé est tel que le bord infini Γ_∞, qui constitue une frontière artificielle du domaine de calcul, est situé assez loin de la frontière Γ_w définissant le contour de l'obstacle. Dans ces conditions, on peut raisonnablement supposer que les effets visqueux sont négligeables sur Γ_∞; par conséquent on ne tient pas compte de l'intégrale de bord correspondante dans (5.6). Comme pour la résolution du système d'équations d'Euler, on considère que l'écoulement dans cette région est représentable par un champ uniforme :

$$\rho_\infty = 1 \ , \ \vec{U}_\infty = (u_\infty \ , \ v_\infty)^T \quad \text{avec} \quad \| \vec{U}_\infty \| = 1 \ , \ p_\infty = \frac{1}{\gamma M_\infty^2} \tag{5.9}$$

Le terme de flux correspondant dans la formulation variationnelle (5.5) est calculé au moyen de la décomposition de flux de Steger et Warming[SW81] donnée par l'expression (2.15).

5.1.3.2 Conditions sur la paroi solide

Sur la frontière Γ_w, on impose une condition classique de non-glissement :

$$\vec{U} = 0$$

et une condition de paroi isotherme :

$$T = T_w$$

où la température d'arrêt T_w est donnée par :

$$T_w = T_\infty \left(1 + \frac{\gamma - 1}{2} M_\infty^2 \right)$$

où M_∞ est le nombre de Mach de l'écoulement uniforme. Les conditions ci-dessus sont appliquées de façon forte, c'est-à-dire que pour un sommet s_i de Γ_w, $\vec{U}(s_i)$ est fixé à 0 et $T(s_i)$ à T_w; la densité ρ_i reste inchangée puisque l'équation correspondante est hyperbolique (en d'autres termes, on suppose un

flux de masse nul à travers $\partial C_i \cap \Gamma_w$). Enfin, l'énergie totale par unité de volume et la pression sont actualisées via les relations :

$$E_i = \rho_i C_v T_w \quad \text{et} \quad p_i = (\gamma - 1)\, E_i$$

5.1.4 Intégration en temps

En supposant que $W(\vec{x},t)$ est constant sur chaque cellule C_i, on obtient le système d'équations différentielles ordinaires suivant :

$$\text{aire}(C_i)\frac{dW_i^n}{dt} + \Psi(W^n)_i = 0 \quad , \quad i = 1, \cdots, N_V \tag{5.10}$$

où $W_i^n = W(\vec{x}_i,t^n)$, $t^n = n\Delta t^n$ et :

$$\Psi(W^n)_i = \sum_{j \in N(s_i)} \Phi_{\mathcal{F}}(W_{ij},W_{ji},\vec{\nu}_{ij}) + \int_{\partial C_i \cap \Gamma_\infty} \vec{\mathcal{F}}(W).\vec{n}_i dl + \frac{1}{\text{Re}} \sum_{\tau \in K(s_i)} \Upsilon_{\tau,i} \tag{5.11}$$

La méthode d'intégration en temps du système d'équations différentielles ordinaires (5.10) est celle décrite dans la section 2.1.4 pour la résolution du système d'équations d'Euler; on utilise donc la méthode implicite linéarisée donnée par les équations (2.18) à (2.20).

5.1.4.1 Linéarisation du terme de flux convectif

La contribution du terme de flux convectif à la matrice Jacobienne $P(W^n)$ de l'équation (2.20) est basée sur une linéarisation approchée du flux numérique précis au premier ordre (2.8). Pour un segment $[s_i,s_j]$, une version implicite du flux numérique est formellement écrite :

$$\Phi_{ij}^{n+1} = \Phi_{\mathcal{F}}(W_i^n,W_j^n,W_i^{n+1},W_j^{n+1},\vec{\nu}_{ij})$$

En posant $U = W_i^n$, $V = W_j^n$, $W = W_i^{n+1}$ et $Z = W_j^{n+1}$ et en utilisant un développement de Taylor au premier ordre du flux implicite on obtient :

$$\Phi_{\mathcal{F}}(U,V,W,Z,\vec{\nu}_{ij}) = \Phi_{\mathcal{F}}(U,V,\vec{\nu}_{ij}) + \left(\frac{\partial \Phi}{\partial U}\right)(W - U) + \left(\frac{\partial \Phi}{\partial V}\right)(Z - V) \tag{5.12}$$

où :

$$\Phi_{\mathcal{F}}(U,V,\vec{\nu}_{ij}) = \Phi_{\mathcal{F}}(W_i^n,W_j^n,\vec{\nu}_{ij})$$

est le flux explicite. L'expression (5.12) peut être simplifiée dans le cas où la fonction de flux numérique prend la forme :

$$\Phi_{\mathcal{F}}(U,V,\vec{\nu}_{ij}) = H_1(U,V,\vec{\nu}_{ij})U + H_2(U,V,\vec{\nu}_{ij})V$$

Par exemple, pour la fonction de flux numérique associée au solveur de Riemann approché de Roe[Roe81] on a :

$$\begin{aligned} \Phi_{\mathcal{F}}(W_i,W_j,\vec{\nu}_{ij}) &= \mathcal{F}(W_i,\vec{\nu}_{ij}) + \mathcal{A}_R^-(W_i,W_j,\vec{\nu}_{ij})(W_j - W_i) \\ &= \mathcal{A}(W_i,\vec{\nu}_{ij})W_i + \mathcal{A}_R^-(W_i,W_j,\vec{\nu}_{ij})(W_j - W_i) \end{aligned}$$

par suite :

$$\begin{cases} H_1(U,V,\vec{\nu}_{ij}) &= \mathcal{A}(U,\vec{\nu}_{ij}) - \mathcal{A}_R^-(U,V,\vec{\nu}_{ij}) \\ H_2(U,V,\vec{\nu}_{ij}) &= \mathcal{A}_R^-(U,V,\vec{\nu}_{ij}) \end{cases}$$

En d'autres termes, on obtient une linéarisation approchée (et donc une matrice Jacobienne approchée) si on suppose :

$$\frac{\partial \Phi}{\partial U} \approx H_1(U,V,\vec{\nu}_{ij}) \quad \text{et} \quad \frac{\partial \Phi}{\partial V} \approx H_2(U,V,\vec{\nu}_{ij})$$

On notera de même que dans le cas de la fonction de flux numérique de Steger et Warming[SW81] (voir l'équation (2.15)) on a :

$$\begin{cases} H_1(U,V,\vec{n_{i\infty}}) &= \mathcal{A}^+(U,\vec{n}_{i\infty}) \\ H_2(U,V,\vec{n_{i\infty}}) &= \mathcal{A}^-(U,\vec{n}_{i\infty}) \end{cases}$$

5.1.4.2 Linéarisation du terme de flux diffusif

Comme précédemment, une version implicite du terme de flux diffusif (5.8), exprimant la contribution du triangle τ au flux nodal global associé au sommet s_i, est formellement écrite :

$$\Upsilon_{\tau,i}^{n+1} = \Upsilon_{\tau,i}\left(W_{k1}^n, W_{k2}^n, W_{k3}^n, W_{k1}^{n+1}, W_{k2}^{n+1}, W_{k3}^{n+1}\right)$$

où s_{k1}, s_{k2} et s_{k3} sont les sommets du triangle τ (l'un deux étant s_i). En procédant comme pour la linéarisation du terme de flux convectif on obtient :

$$\begin{aligned} \Upsilon_{\tau,i}^{n+1} &= \Upsilon_{\tau,i}\left(W_{k1}^n, W_{k2}^n, W_{k3}^n\right) + \left(\frac{\partial \Upsilon_{\tau,i}}{\partial W_{k1}^n}\right)\left(W_{k1}^{n+1} - W_{k1}^n\right) \\ &+ \left(\frac{\partial \Upsilon_{\tau,i}}{\partial W_{k2}^n}\right)\left(W_{k2}^{n+1} - W_{k2}^n\right) + \left(\frac{\partial \Upsilon_{\tau,i}}{\partial W_{k3}^n}\right)\left(W_{k3}^{n+1} - W_{k3}^n\right) \end{aligned}$$

Dans les expressions ci-dessus, $\Upsilon_{\tau,i}\left(W_{k1}^n, W_{k2}^n, W_{k3}^n\right)$ est donné par :

$$\begin{aligned} \Upsilon_{\tau,i}\left(W_{k1}^n, W_{k2}^n, W_{k3}^n\right) &= \text{aire}\,(\tau)\left(R_x\left(W_{k1}^n, W_{k2}^n, W_{k3}^n\right)\frac{\partial \phi_i}{\partial x}\,|_\tau + \right. \\ &\qquad \left. R_y\left(W_{k1}^n, W_{k2}^n, W_{k3}^n\right)\frac{\partial \phi_i}{\partial y}\,|_\tau\right) \end{aligned} \tag{5.13}$$

Par suite, la linéarisation du flux visqueux peut aussi s'écrire comme :

$$\Upsilon_{\tau,i}^{n+1} = \Upsilon_{\tau,i}^n + \left(\frac{\partial \Upsilon_{\tau,i}}{\partial W_i}\right)\delta W_i^{n+1} + \sum_{k\in\tau, k\neq i}\left(\frac{\partial \Upsilon_{\tau,i}}{\partial W_k}\right)\delta W_k^{n+1} \tag{5.14}$$

Le calcul des termes de l'équation (5.14) se fait de façon exacte à partir de l'expression des composantes des flux discrets (5.13) (voir [FLLO89] pour plus de détails).

5.1.4.3 Résolution numérique

On décrit ici la stratégie de résolution référencée « stratégie de résolution globale » dans la section de résultats numériques 5.3.

En prenant en compte les diverses contributions des sections précédentes, l'intégration implicite du système d'équations de Navier-Stokes se déroule de la façon suivante :

- Phase « physique » ou phase explicite. Cette étape porte sur l'évaluation du membre de droite $\delta\hat{W}_i$ du système linéaire :

$$\delta\hat{W}_i = -\left(\sum_{j\in N(s_i)} \Phi^n_{ij} + \int\limits_{\partial C_i\cap\partial\Gamma_\infty} \Phi^{SW}\left(W^n_i, W_\infty, \vec{\eta}\right) + \frac{1}{\mathsf{Re}}\sum_{\tau\in K(s_i)} \Upsilon^n_{\tau,i}\right) \tag{5.15}$$

- Phase « mathématique » ou phase implicite. Cette étape consiste d'une part en l'assemblage des différentes contributions à la matrice implicite $P\left(W^n\right)$ et, d'autre part, en la résolution itérative du système linéaire :

$$P\left(W^n\right)\delta W^{n+1} = \delta\hat{W} \tag{5.16}$$

par une méthode de relaxation comme Jacobi ou Gauss-Seidel. La matrice $P\left(W^n\right)$ est creuse, de structure irrégulière et non-symétrique. Chaque terme de cette matrice est ici un bloc dense de taille 4×4. La i-ème ligne du système linéaire s'écrit :

$$P^n_{ii}\delta W^{n+1}_i + \sum_{j\in N(s_i)} M^n_{ij}\delta W^{n+1}_j + \sum_{\tau\in K(s_i)}\sum_{k,k\neq i} M^n_{ik}\delta W^{n+1}_k = \delta\hat{W}_i$$

Les blocs formant la sous-matrice diagonale ont pour expression :

$$P^n_{ii} = \sigma^n_i\,\mathsf{Id} + \sum_{j\in N(s_i)} H^n_{1,ij} + \int\limits_{\partial C_i\cap\partial\Gamma_\infty} \mathcal{A}^+\left(W^n_i, \vec{\eta}\right) + \frac{1}{\mathsf{Re}}\sum_{\tau\in K(s_i)}\left(\frac{\partial\Upsilon^n_{\tau,i}}{\partial W_i}\right) \tag{5.17}$$

où $\sigma^n_i = \dfrac{\mathsf{aire}(C_i)}{\Delta t^n}$. Les blocs constituant les matrices extra-diagonales sont quant à eux donnés par :

$$P^n_{ij} = H^n_{2,ij} \quad , \quad P^n_{ik} = \frac{1}{\mathsf{Re}}\left(\frac{\partial\Upsilon^n_{\tau,i}}{\partial W_k}\right)$$

On notera que le système linéaire est écrit sous forme « delta » et par suite :

$$W^{n+1} = W^n + \delta W^{n+1}$$

5.2 Algorithme de résolution par sous-domaine

Dans cette section, on s'intéresse à la formulation d'un algorithme de résolution par sous-domaine et à son adaptation au contexte discret défini dans la section 5.1.

5.2.1 Formulation en continu

La formulation de l'algorithme de résolution par sous-domaine dans le cas continu s'inspire de celle proposée par Quarteroni et Stolcis[QS96]. On suppose ici que le domaine de calcul $\Omega \subset \mathbb{R}^2$ est décomposé en sous-domaines non recouvrants Ω_i avec $\Omega_i \cap \Omega_j = \Gamma_{ij}$ si Ω_j est un sous-domaine voisin de Ω_i. La solution de l'équation (5.1) peut alors être reformulée comme :

$$\left\{\begin{array}{lll} \dfrac{\partial W}{\partial t} + \vec{\nabla}.\vec{\mathcal{F}}(W) = \dfrac{1}{\mathsf{Re}}\vec{\nabla}.\vec{\mathcal{R}}(W) & \text{pour} & \vec{x} \in \Omega_i \\ \left[\left(\vec{\mathcal{F}}(W) - \dfrac{1}{\mathsf{Re}}\vec{\mathcal{R}}(W)\right)\right]_{\Gamma_{ij}}.\vec{n}_{ij} = 0 & \text{pour} & \vec{x} \in \Gamma_{ij} \\ + \text{ Conditions de paroi sur } \Gamma_w & & \\ + \text{ Conditions à l'infini sur } \Gamma_\infty & & \end{array}\right. \tag{5.18}$$

où $\vec{\mathcal{F}}(W) = (F_x(W)\ ,\ F_y(W))^T$ et $\vec{\mathcal{R}}(W) = (R_x(W)\ ,\ R_y(W))^T$; $\vec{n}_{ij}$ désigne le vecteur normal en tout point de Γ_{ij} dirigé de Ω_i vers Ω_j; par ailleurs, $[a]_{\Gamma_{ij}}$ désigne le saut de la quantité a à l'interface Γ_{ij} :

$$[a]_{\Gamma_{ij}} = a\ |_{\Gamma_{ij}^+} \ - \ a\ |_{\Gamma_{ij}^-}$$

où Γ_{ij}^+ et Γ_{ij}^- désignent respectivement les parties droite et gauche de l'interface Γ_{ij}. Dans (5.18), la condition d'interface sur Γ_{ij} exprime donc la continuité du flux normal total. Cette propriété de continuité du flux est une conséquence naturelle du fait que le vecteur d'état W est une solution faible de l'équation (5.1) de même que les solutions locales $W_{|\Omega_i}$. Une argumentation similaire à celle proposée dans la section 1.1.2 dans le cadre de la résolution du système d'équations d'Euler, et plus particulièrement dans la démonstration du théorème 2, permet d'aboutir à cette forme de condition d'interface.

5.2.2 Formulation en discret

Remarquons tout d'abord que la condition d'interface apparaissant dans (5.18) se réduit à :

$$\left[\vec{\mathcal{F}}(W)\right]_{\Gamma_{ij}}.\vec{n}_{ij} = 0 \tag{5.19}$$

lorsque l'on fait tendre le nombre de Reynolds vers l'infini. Cette situation correspond à la résolution du système d'équations d'Euler pour un fluide non-visqueux. On notera, qu'en vertu de la propriété d'homogénéité des termes de flux du système d'équations d'Euler, la condition (5.19) n'est rien d'autre qu'une reformulation des conditions (1.15). La construction d'une version discrète de l'algorithme de Schwarz défini par les équations (5.18) pour chaque sous-domaine, est une extension directe de ce qui a été proposé pour la résolution du système d'équations d'Euler. Le point de départ est encore donné par la stratégie de parallélisation décrite dans la section 2.2. Cependant, pour des raisons présentées plus loin, on décide ici de traiter séparément les termes de flux convectif et diffusif dans la condition d'interface de (5.18). De façon évidente, on a :

$$\left\{\begin{array}{l} \left[\vec{\mathcal{F}}(W)\right]_{\Gamma_{ij}}.\vec{n}_{ij} = 0 \\ \qquad \text{et} \\ \left[\vec{\mathcal{R}}(W)\right]_{\Gamma_{ij}}.\vec{n}_{ij} = 0 \end{array}\right. \Rightarrow \left[\left(\vec{\mathcal{F}}(W) - \frac{1}{\mathsf{Re}}\vec{\mathcal{R}}(W)\right)\right]_{\Gamma_{ij}}.\vec{n}_{ij} = 0 \tag{5.20}$$

L'approche adoptée consiste alors à introduire des inconnues interfaces différentes pour les termes de flux convectif et diffusif.

5.2.2.1 Continuité des flux convectifs

La traitement de la première des conditions du terme de droite de (5.20) est identique à ce qui est décrit dans la section 2.3.

5.2.2.2 Continuité des flux diffusifs

L'objectif de cette section est de construire l'inconnue interface discrète associée à la seconde condition du terme de gauche de (5.20). Tout d'abord, on notera que la formulation mixte éléments finis/volumes finis adoptée pour la discrétisation en espace (voir la section 5.1.2) ne facilite pas la prise en compte de la condition de continuité des flux diffusifs normaux à l'interface Γ_{ij}. En effet, si la formulation volumes finis utilisée pour le calcul des termes de flux convectif est directement compatible avec l'évaluation d'une inconnue du type flux normal à l'interface, il n'en va pas de même des termes de flux diffusif dont le support de calcul est le triangle. On notera au passage que la stratégie de partitionnement adoptée ici implique qu'un triangle interface est partagé par plusieurs sous-domaines (au plus trois). En fait, une méthode de discrétisation purement volumes finis pour le traitement des termes de flux convectif et diffusif, telle que celle proposée par Rostand et Stoufflet[RS88], aurait probablement été mieux adaptée à la réalisation de notre objectif.

Afin de faciliter la mise en œuvre de la stratégie de résolution par sous-domaine, on impose la continuité de toutes les composantes du flux au lieu des composantes normales seulement, ce que l'on traduit par :

$$\left[\vec{\mathcal{R}}(W)\right]_{\Gamma_{ij}} = 0 \tag{5.21}$$

Par conséquent, pour un triangle interface donné τ (i.e. un triangle dont les sommets sont situés dans la zone de recouvrement), les composantes du flux discret (5.8) sont considérées comme des inconnues du problème. Dans ces conditions, il s'agit maintenant de préciser les deux points suivants :

- la définition d'un couplage approprié entre les composantes du flux diffusif qui sont calculées sur le triangle τ, et les inconnues nodales correspondantes (i.e. les composantes du vecteur des variables conservatives W pour chacun des sommets du triangle τ);
- la construction d'une linéarisation associée du flux diffusif qui doit être prise en compte dans la méthode d'intégration implicite en temps décrite dans la section 5.1.4.

En utilisant l'expression (5.8) du flux diffusif discret on peut écrire :

$$\Upsilon_{\tau,i}(\tau) = \text{aire}\,(\tau)\left(\begin{pmatrix} 0 \\ r_2(\tau) \\ r_3(\tau) \\ r_4(\tau) \end{pmatrix} \frac{\partial \phi_i}{\partial x}\,|_\tau + \begin{pmatrix} 0 \\ r_3(\tau) \\ s_3(\tau) \\ s_4(\tau) \end{pmatrix} \frac{\partial \phi_i}{\partial y}\,|_\tau\right) \tag{5.22}$$

où $r_i(\tau)$ et $s_i(\tau)$ sont les composantes des vecteurs de flux discrets $R_x(\tau)$ et $R_y(\tau)$ qui sont constants sur le triangle τ. En accord avec les expressions des composantes de $R_x(W)$ et $R_y(W)$ (5.3), on peut encore écrire :

$$\Upsilon_{\tau,i}(\tau) = \text{aire}\,(\tau) \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \partial_x\phi_i & \partial_y\phi_i & 0 & 0 & 0 \\ 0 & \partial_x\phi_i & 0 & \partial_y\phi_i & 0 \\ 0 & 0 & \partial_x\phi_i & 0 & \partial_y\phi_i \end{pmatrix} \begin{pmatrix} r_2(\tau) \\ r_3(\tau) \\ r_4(\tau) \\ s_3(\tau) \\ s_4(\tau) \end{pmatrix} \tag{5.23}$$

où $\partial_x\phi_i = \dfrac{\partial\phi_i}{\partial x}$ et $\partial_y\phi_i = \dfrac{\partial\phi_i}{\partial y}$. L'expression ci-dessus peut être interprétés comme une linéarisation des flux diffusifs à l'interface (i.e. pour un triangle interface) alternative à la linéarisation classique décrite dans la section 5.1.4.2, où les quantités $r_i(\tau)$ et $s_i(\tau)$ définissent les nouvelles variables recherchées. Par suite, le couplage direct entre les variables nodales situées dans la zone de recouvrement (i.e. les variables W_{ki} associées aux trois sommets du triangle interface) est remplacé par un couplage entre ces variables et les composantes du flux interface via les blocs matriciels 4×5 de l'expression (5.23). Inversement, le couplage entre les nouvelles variables interfaces et les variables nodales peut s'écrire, à partir de la linéarisation classique, comme :

$$\begin{cases} r_m(\tau) = \displaystyle\sum_{k_i\in\tau} \frac{\partial r_m(\tau)}{\partial W_{k_i}} W_{k_i} & \text{pour} \quad m = 2{,}3{,}4 \\ s_m(\tau) = \displaystyle\sum_{k_i\in\tau} \frac{\partial s_m(\tau)}{\partial W_{k_i}} W_{k_i} & \text{pour} \quad m = 3{,}4 \end{cases} \tag{5.24}$$

5.2.2.3 Formulation d'un système interface

On se place ici dans le cas d'une décomposition en deux sous-domaines $\Omega = \Omega_1 \cup \Omega_2$. A partir des flux élémentaires (2.28) définis dans la section 2.3, des flux (5.24) et aussi des flux convectifs et diffusifs élémentaires (inchangés) pour les sommets purement internes à Ω_1 et Ω_2, on peut construire un système linéaire implicite (analogue de (2.20)) où l'on distingue les inconnues purement internes à un sous-domaine (vecteurs d'état) des inconnues à l'interface (flux normaux) :

$$\begin{pmatrix} \mathcal{M}_1 & 0 & \mathcal{M}_{1c} & \mathcal{M}_{1v} \\ 0 & \mathcal{M}_2 & \mathcal{M}_{2c} & \mathcal{M}_{2v} \\ \mathcal{F}_{1c} & \mathcal{F}_{2c} & \mathsf{Id} & 0 \\ \mathcal{F}_{1v} & \mathcal{F}_{2v} & 0 & \mathsf{Id} \end{pmatrix} \begin{pmatrix} W_1 \\ W_2 \\ \Phi_c \\ \Phi_v \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ 0 \\ 0 \end{pmatrix} \tag{5.25}$$

où $\mathcal{M}_1$ (respectivement $\mathcal{M}_2$) est la matrice implicite qui couple les sommets internes à Ω_1 (respectivement Ω_2) tandis que $\mathcal{F}_{1c}$, $\mathcal{F}_{2c}$, $\mathcal{F}_{1v}$, $\mathcal{F}_{2v}$, $\mathcal{M}_{1c}$, $\mathcal{M}_{1v}$, $\mathcal{M}_{2c}$ et $\mathcal{M}_{2v}$ sont des matrices de couplage entre les inconnues internes et les inconnues à l'interface Γ. A ce stade, les inconnues internes peuvent être éliminées du système linéaire (5.25), par application d'un principe de sous-structuration, au profit des inconnues à l'interface pour aboutir à la formulation d'un système interface :

$$S \begin{pmatrix} \Phi_c \\ \Phi_v \end{pmatrix} = \begin{pmatrix} (\mathsf{Id} - S_{1c})\,\Phi_c - S_{1v}\Phi_v \\ -S_{2c}\Phi_c + (\mathsf{Id} - S_{2v})\,\Phi_v \end{pmatrix} = g \tag{5.26}$$

avec :

$$\begin{cases} S_{1c} &= \mathcal{F}_{1c}\mathcal{M}_1^{-1}\mathcal{M}_{1c} + \mathcal{F}_{2c}\mathcal{M}_2^{-1}\mathcal{M}_{2c} \\ S_{1v} &= \mathcal{F}_{1c}\mathcal{M}_1^{-1}\mathcal{M}_{1v} + \mathcal{F}_{2c}\mathcal{M}_2^{-1}\mathcal{M}_{2v} \\ S_{2c} &= \mathcal{F}_{1v}\mathcal{M}_1^{-1}\mathcal{M}_{1c} + \mathcal{F}_{2v}\mathcal{M}_2^{-1}\mathcal{M}_{2c} \\ S_{2v} &= \mathcal{F}_{1v}\mathcal{M}_1^{-1}\mathcal{M}_{1v} + \mathcal{F}_{2v}\mathcal{M}_2^{-1}\mathcal{M}_{2v} \\ g &= -\begin{pmatrix} [\mathcal{F}_{1c}\mathcal{M}_1^{-1}b_1 + \mathcal{F}_{2c}\mathcal{M}_2^{-1}b_2] \\ \\ [\mathcal{F}_{1v}\mathcal{M}_1^{-1}b_1 + \mathcal{F}_{2v}\mathcal{M}_2^{-1}b_2] \end{pmatrix} \end{cases}$$

Une fois le système (5.26) résolu pour les valeurs de $(\Phi_c,\Phi_v)^T$, on obtient les valeurs des inconnues purement internes par résolution en parallèle des systèmes locaux suivants :

$$\begin{cases} W_1 &= \mathcal{M}_1^{-1}(b_1 - \mathcal{M}_{1c}\Phi_c - \mathcal{M}_{1v}\Phi_v) \\ W_2 &= \mathcal{M}_2^{-1}(b_2 - \mathcal{M}_{2c}\Phi_c - \mathcal{M}_{2v}\Phi_v) \end{cases} \tag{5.27}$$

5.2.3 Stratégie de résolution des systèmes locaux

La méthode de résolution par sous-domaine proposée dans la section 5.2 fait apparaître des étapes de résolution locales indépendantes (i.e. parallèles) dans chaque sous-domaine. Comme pour la résolution du système d'équations d'Euler, une composante importante de notre étude concerne l'utilisation d'une stratégie de résolution approchée pour ces systèmes locaux. Pour ce faire, on utilise la méthode multigrille par agglomération de volumes introduite dans la section 2.4 adaptée à la résolution du système d'équations de Navier-Stokes. La principale difficulté dans cette adaptation réside dans l'approximation des termes diffusifs, de nature elliptique, sur les grilles grossières. Comme pour les termes convectifs, la contribution des termes de flux diffusifs à la matrice Jacobienne est déterminée à partir de la discrétisation des équations sur les différentes grilles grossières générées par le processus d'agglomération. L'absence de triangulation sur ces grilles ne nous permet pas de conserver la formulation variationnelle de la grille fine. Toutefois, afin de garder une approximation éléments finis sur ces niveaux, on définit des fonctions de base grilles grossières par sommation des fonctions de base de la grille fine. Soit I^h l'ensemble des indices i du niveau fin. On définit une partition de I^h :

$$I^h = (1, \cdots, i, \cdots, n_h) = I_1 \cup I_2 \cup \cdots \cup I_{n_H} \quad n_H << n_h$$

où chaque I_J identifie l'ensemble des indices des sommets s_k^h (i.e. des cellules de contrôle C_k^h) du maillage fin initial définissant l'agglomérat C_J^H du maillage grossier généré par agglomération (n_H est le nombre total d'agglomérats du maillage grossier). Les fonctions de base grossières, notées φ_J, pour chaque sous-ensemble I_J sont définies par :

$$\varphi_J = \sum_{j \in I_J} \varphi_j \quad J = 1, \cdots, n_H$$

Ces fonctions de base grossières, définies initialement par Koobus[Koo94] (voir aussi [KLA94]), ne sont plus totalement affines par morceaux. En effet, le passage sur la grille grossière conduit à des fonctions de base qui possèdent une partie constante. De ce fait, le nouveau problème sur la grille grossière n'est pas une bonne approximation du problème d'origine sur la grille fine. La consistance de la discrétisation grossière déduite de la définition ci-dessus avec la formulation éléments finis adoptée sur

la grille fine, a été étudiée par Koobus[Koo94] pour un problème de Poisson et par Carré[Car95] pour les équations de Navier-Stokes. Plus précisément, Carré[Car95] a mené cette étude de consistance pour la linéarisation de la formulation mixte éléments finis/volumes finis appliquée aux équations de Navier-Stokes monodimensionnelles. Ces études mettent en évidence une inconsistance entre les deux discrétisations. En particulier, le système linéaire obtenu sur la grille grossière n'est donc pas une approximation du problème d'origine sur la grille fine. Koobus[Koo94] propose de corriger ce problème via l'introduction d'un facteur correctif appliqué au système linéaire de la grille grossière :

$$\mho_n = \frac{2\,(n-1)^2}{(2n-1)^2} \tag{5.28}$$

où n est une estimation du nombre de sommets dans n'importe quelle direction d'espace, évaluée comme $\sqrt[d]{n_H}$ où d est la dimension spatiale et n_H le nombre total d'agglomérats de la grille grossière considérée. Cette expression est déduite empiriquement de l'expression obtenue dans le cas d'un maillage cartésien (orthogonal et régulier) pour lequel on montre que l'adjonction d'un facteur correctif permet de retrouver exactement la consistance

5.3 Résultats numériques

5.3.1 Définition des cas tests

Dans cette section, on s'intéresse à l'évaluation de la méthode de résolution par sous-domaine du système d'équations de Navier-Stokes proposée dans la section 5.2, dans le contexte de la simulation d'écoulements visqueux laminaires stationnaires. La géométrie sélectionnée est celle du profil d'aile `NACA0012` comme dans le cas de la résolution du système d'équations d'Euler. Deux maillages ont été utilisés dont les caractéristiques sont données dans la table 5.1. Les situations suivantes ont été considérées :

TVIS1 : l'écoulement transsonique caractérisé par un nombre de Mach de référence égal à 0.85, un nombre de Reynolds fixé à 2000 et un angle d'incidence de $0°$. Le pas de temps (local) est calculé à partir d'un nombre de Courant variable donné par la loi CFL=$500 \times k_t$ où k_t désigne l'itération en temps.

TVIS2 : l'écoulement subsonique caractérisé par un nombre de Mach de référence égal à 0.8, un nombre de Reynolds fixé à 73 et un angle d'incidence de $10°$. Le pas de temps (local) est calculé à partir d'un nombre de Courant variable donné par la loi CFL=$500 \times k_t$.

TVIS3 : l'écoulement supersonique caractérisé par un nombre de Mach de référence égal à 2.0, un nombre de Reynolds fixé à 106 et un angle d'incidence de $10°$. Le pas de temps (local) est calculé à partir d'un nombre de Courant variable donné par la loi CFL=$50 \times k_t$.

Dans chacun des cas ci-dessus, l'extension au second ordre de précision dans le calcul des flux numériques convectifs (voir les équations (2.11) et (2.12)) n'utilise pas de procédure de limitation.

TAB. 5.1 – *Caractéristiques des triangulations non-structurées autour du profil d'aile* `NACA0012`

Mesh	# Sommets	# Triangles	# Segments
N1	48792	96896	145688
N2	194480	387584	582064

Les résultats numériques présentés ici sont tous caractérisés par les points suivants :

- le calcul démarre d'une situation d'écoulement uniforme;
- les résolutions locales induites par la méthode de décomposition de domaine sont toujours effectuées de façon approchée.

Les lignes iso-Mach stationnaires pour chacun des cas tests considérés sont montrées sur les figures 5.1 et 5.2 (calculs effectués sur le maillage N2).

5.3.2 Plateformes de calcul

Les expériences numériques ont été réalisées sur une grappe de 14 PC sous système LINUX (tous équipés de cartes mères comprenant 2 processeurs `Pentium Pro` cadencés à 500 Mhz et 512 Mb de mémoire RAM partagée par les deux processeurs) inter-connectés par deux switchs `FastEthernet` d'un débit théorique de 100 Mbit/s. La programmation parallèle repose sur l'implémentation MPICH de l'environnement MPI. Le langage de programmation est FORTRAN 77 et le compilateur G77 a été utilisé (avec options d'optimisation maximales). Les calculs sont effectués en double précision. Dans les tables ci-dessous, les notations sont celles déjà adoptées dans le cadre de la résolution du système d'équations d'Euler (voir la section 2.6.2.2).

5.3.3 Cas test TVIS1

Les stratégies de résolution suivantes ont été utilisées :

GLOB : stratégie de résolution globale où, à chaque pas de temps, le système linéaire (5.16) est résolu par une méthode de Jacobi avec un seuil de résolution linéaire fixé à $\varepsilon_g = 10^{-1}$. On a vérifié ici qu'une augmentation de l'effort de résolution linéaire (en diminuant le seuil de résolution ε_g de 10^{-1} à 10^{-2}) ne s'est pas traduit en une réduction notable du nombre de pas de temps nécessaires à l'obtention de l'état stationnaire (alors que les temps d'exécution augmentent de façon conséquente);

DDM1 : stratégie de résolution par sous-domaine où, à chaque pas de temps, le système interface (5.26) est résolu par une méthode de GMRES sans ré-initialisation et non-préconditionnée, avec un seuil de résolution linéaire fixé à $\varepsilon_i = 10^{-1}$. Les systèmes linéaires locaux sont résolus par l'application d'un nombre variable de V-cycles, le seuil de résolution linéaire étant fixé à $\varepsilon_l = 10^{-1}$. Le V-cycle sélectionné est caractérisé par $\nu_1 = 2$ pré-lissages, $\nu_2 = 2$ post-lissages et $\nu_g = 2$ lissages sur le niveau le plus grossier.

DDM2 : stratégie de résolution par sous-domaine où, à chaque pas de temps, le système interface (5.26) est résolu par une méthode de GMRES sans ré-initialisation et non-préconditionnée, avec un seuil de résolution linéaire fixé à $\varepsilon_i = 10^{-1}$. Les systèmes linéaires locaux sont résolus par l'application d'un nombre constant de 3 V-cycles de la méthode multigrille par agglomération. Le V-cycle sélectionné est caractérisé par $\nu_1 = 2$ pré-lissages, $\nu_2 = 2$ post-lissages et $\nu_g = 2$ lissages sur le niveau le plus grossier (le nombre de niveaux grossiers est donné dans les tables).

Les nombres effectifs de pas de temps jusqu'à l'obtention de l'état stationnaire (résidu initial de l'énergie par unité de volume divisé par un facteur 10^{10}) et les temps d'exécution correspondants sont résumés dans les tables 5.2 (calculs effectués sur le maillage N1) et 5.3 (calculs effectués sur le maillage N2). La table 5.3 fournit un ensemble de résultats pour des calculs effectués sur le maillage N2 avec $N_p = 16$. Dans la table 5.2, l'accélération parallèle est calculée relativement au temps d'exécution mesuré

FIG. 5.1 – *Écoulement visqueux laminaire autour d'un profil d'aile* `NACA0012`
Lignes iso-Mach stationnaires : cas test **TVIS1** (haut) et cas test **TVIS2** (bas)

FIG. 5.2 – *Écoulement visqueux laminaire autour d'un profil d'aile* NACA0012
Lignes iso-Mach stationnaires : cas test **TVIS3**

pour $N_p = 4$. Les convergences non-linéaires vers l'état stationnaire en termes de la décroissance du résidu (normalisé à sa valeur initiale) de l'énergie totale par unité de volume sont montrées sur la figure 5.3.

Ces résultats nous conduisent à faire les premiers commentaires suivants :

- comme on pouvait s'y attendre, la stratégie de résolution par sous-domaine s'avère plus efficace que la stratégie de résolution globale lorsque le nombre de sous-domaines augmente. Ainsi, pour $N_p = 24$ et pour les calculs effectués sur le maillage N1, le temps de calcul réel pour la stratégie de résolution globale est de 429 secondes contre 239 secondes pour la stratégie de résolution par sous-domaine basée sur un nombre constant de V-cycles pour les résolutions locales. Dans ce cas, on obtient donc une réduction de 44% du temps total de simulation;
- la caractéristique la plus remarquable de la méthode de résolution par sous-domaine est certainement l'efficacité parallèle qu'elle démontre. L'efficacité parallèle est ici évaluée à l'aide du rapport « % CPU ». On remarque donc que ce rapport décroît sensiblement pour la stratégie de résolution globale lorsque le nombre de sous-domaines augmente, alors qu'il reste relativement élevé dans le cas de la méthode de résolution par sous-domaine. Dans les conditions de la comparaison faite au point précédent, on note une amélioration de l'efficacité parallèle de l'ordre de 16%;
- les mesures obtenues avec le maillage fin N2 montrent que la stratégie de résolution par sous-domaine basée sur une tolérance de résolution linéaire cible est bien plus coûteuse que celle utilisant un nombre constant de V-cycles pour les résolutions locales. Cependant, dans ce dernier cas,on note une réduction du temps total de calcul de seulement 5 % (contre 27% pour les calculs effectués sur le maillage N1 avec $N_p = 16$). Des gains plus importants sont prévisibles pour des plus grands nombres de sous-domaines sachant que le coût total de simulation reste dominé par le coût des

TAB. 5.2 – *Écoulement visqueux laminaire autour d'un profil d'aile* `NACA0012` *: cas test* **TVIS1**
Temps d'exécution pour le calcul de l'écoulement stationnaire (maillage N1)
Stratégie de résolution globale basée sur la méthode de relaxation de Jacobi
Stratégies de résolution par sous-domaine : système interface résolu par GMRES

Méthode	N_p	N_g	# it	CPU	REEL	% CPU	$S(N_p)$
GLOB	4	1	73	1527 sec	1611 sec	95.0	1.0
	6	1	73	976 sec	1081 sec	90.5	1.5
	8	1	73	755 sec	856 sec	88.0	1.9
	12	1	73	495 sec	615 sec	80.5	2.6
	16	1	73	384 sec	512 sec	75.0	3.1
	24	1	73	272 sec	429 sec	63.5	3.8
DDM1	4	5	71	1679 sec	1751 sec	96.0	1.0
	6	5	72	1030 sec	1071 sec	96.0	1.6
	8	5	73	869 sec	926 sec	94.0	1.9
	12	5	74	599 sec	648 sec	92.5	2.7
	16	5	76	446 sec	493 sec	90.5	3.5
	24	4	78	286 sec	333 sec	86.0	5.3
DDM2	4	5	74	1296 sec	1320 sec	98.0	1.0
	6	5	75	844 sec	871 sec	97.0	1.5
	8	5	75	663 sec	681 sec	97.0	1.9
	12	5	79	489 sec	517 sec	94.5	2.5
	16	5	78	338 sec	372 sec	91.0	3.5
	24	4	77	212 sec	239 sec	89.0	5.5

TAB. 5.3 – *Écoulement visqueux laminaire autour d'un profil d'aile* `NACA0012` *: cas test* **TVIS1**
Temps d'exécution pour le calcul de l'écoulement stationnaire (maillage N2)
Stratégies de résolution globale basée sur la méthode de relaxation de Jacobi
Stratégie de résolution par sous-domaine : système interface résolu par GMRES

Méthode	N_p	N_g	# it	CPU	REEL	% CPU
GLOB	16	1	151	3705 sec	4127 sec	90.0
DDM1	16	5	149	8338 sec	8535 sec	97.4
DDM2	16	5	161	3841 sec	3928 sec	98.0

FIG. 5.3 – *Écoulement visqueux laminaire autour d'un profil d'aile* `NACA0012` *: cas test* **TVIS1**
Convergences non-linéaires des stratégies de résolution globale et par sous-domaine (maillage N1)
Axe X : itération en temps - Axe Y : résidu en échelle logarithmique

FIG. 5.4 – *Écoulement visqueux laminaire autour d'un profil d'aile* `NACA0012`
Cas test **TVIS1** (maillage N1)
Convergence linéaire de la stratégie de résolution par sous-domaine ($\varepsilon_i = 10^{-1}$/solveur local : 3 V(2,2,2)-cycles)
Axe X : itération en temps - Axe Y : # itérations GMRES

résolutions locales qui diminuera sensiblement dans ce cas. Malgré cela, l'efficacité parallèle atteint 98% pour la méthode de résolution par sous-domaine alors qu'elle est ne dépasse pas 90 % pour la stratégie de résolution globale;

– dans cette étude, le système interface (5.26) est résolu par une méthode GMRES non préconditionnée. Dans ce contexte, il est intéressant d'évaluer les propriétés de scalabilité de la méthode de résolution par sous-domaine sachant que celles-ci ne peuvent être que meilleures avec la prise en compte d'un préconditionneur. Une réponse partielle à cette question est donnée sur la figure 5.4 qui représente la convergence linéaire à chaque pas de temps, en termes du nombre effectif d'itérations de GMRES, dans le cas d'une stratégie basée sur une nombre constant de 3 V-cycles pour chaque résolution locale. On peut voir sur cette figure que le nombre d'itérations de GMRES augmente faiblement lorsque le nombre de sous-domaines passe de $N_p = 4$ à $N_p = 24$. Ce commentaire doit cependant être modéré par le fait que les systèmes interfaces sont ici résolus avec une faible précision ce qui demande en moyenne 6 itérations de GMRES.

5.3.4 Cas test TVIS2

Les stratégies de résolution sélectionnées sont essentiellement celles mises en œuvre pour le cas test **TVIS1** sauf pour la méthode de résolution par sous-domaine pour laquelle on a seulement considéré le cas d'une complexité constante pour les résolutions locales. Plus précisément, les stratégies de résolution suivantes ont été utilisées :

GLOB1 : stratégie de résolution globale où, à chaque pas de temps, le système linéaire (5.16) est résolu par une méthode de Jacobi avec un seuil de résolution linéaire fixé à $\varepsilon_g = 10^{-1}$.

GLOB2 : stratégie de résolution globale où, à chaque pas de temps, le système linéaire (5.16) est résolu par une méthode de Jacobi avec un seuil de résolution linéaire fixé à $\varepsilon_g = 10^{-2}$.

DDM : stratégie de résolution par sous-domaine où, à chaque pas de temps, le système interface (5.26) est résolu par une méthode de GMRES sans ré-initialisation et non-préconditionnée, avec un seuil de résolution linéaire fixé à $\varepsilon_i = 10^{-1}$. Les systèmes linéaires locaux sont résolus par l'application d'un nombre constant de 3 V-cycles de la méthode multigrille par agglomération. Le V-cycle sélectionné est caractérisé par $\nu_1 = 2$ pré-lissages, $\nu_2 = 2$ post-lissages et $\nu_g = 2$ lissages sur le niveau le plus grossier (le nombre de niveaux grossiers est donné dans les tables).

Les nombres effectifs de pas de temps jusqu'à l'obtention de l'état stationnaire (résidu initial de l'énergie par unité de volume divisé par un facteur 10^{10}) et les temps d'exécution correspondants sont résumés dans les tables 5.4 (calculs effectués sur le maillage N1) et 5.5 (calculs effectués sur le maillage N2). Par ailleurs, la table 5.5 fournit un ensemble de résultats pour des calculs effectués sur le maillage N2 avec $N_p = 16$. Dans la table 5.4, l'accélération parallèle est calculée relativement au temps d'exécution mesuré pour $N_p = 4$. Les convergences non-linéaires vers l'état stationnaire sont représentées sur la figure 5.5.

Pour ce qui concerne les calculs réalisés sur le maillage N1, les résultats obtenus confirment les commentaires faits pour le cas test **TVIS1**. On notera que la méthode de résolution par sous-domaine conduit toujours à un nombre inférieur de pas de temps pour atteindre l'état stationnaire sauf pour le cas $N_p = 12$. Dans ce cas, on peut suspecter que le partitionnement du maillage N1 a conduit à des sous-domaines de formes plus irrégulières avec un impact direct sur la qualité des maillages grossiers construits localement via le principe d'agglomération. Dans ces conditions, une stratégie de résolution

locale basée sur plus de 3 V-cycles est probablement nécessaire pour obtenir une convergence non-linéaire en accord avec celle observée pour les autres valeurs de N_p. Les calculs réalisés sur le maillage N2 montrent que la stratégie de résolution par sous-domaine est 3.3 fois moins coûteuse que la stratégie de résolution globale. Pour cette dernière, une réduction de la tolérance de résolution linéaire de $\varepsilon_g = 10^{-1}$ à $\varepsilon_g = 10^{-2}$ conduit à une convergence plus rapide vers l'état stationnaire (en termes du nombre de pas de temps) au détriment d'une augmentation de 5% du temps total de simulation.

Enfin, la figure 5.6 représente la convergence linéaire à chaque pas de temps, en termes du nombre effectif d'itérations de GMRES, dans le cas d'une stratégie basée sur une nombre constant de 3 V-cycles pour chaque résolution locale. On note cette fois une dépendance un peu plus nette du nombre d'itérations de GMRES sur le nombre de sous-domaines (par rapport à ce qui avait été observé pour le cas test **TVIS1**, figure 5.4) même si ce nombre reste relativement faible compte tenu de la faible précision de résolution des systèmes interfaces.

TAB. 5.4 – *Écoulement visqueux laminaire autour d'un profil d'aile* `NACA0012` *: cas test* **TVIS2**
Temps d'exécution pour le calcul de l'écoulement stationnaire (maillage N1)
Stratégie de résolution globale basée sur la méthode de relaxation de Jacobi
Stratégie de résolution par sous-domaine : système interface résolu par GMRES

Méthode	N_p	N_g	# it	CPU	REEL	% CPU	$S(N_p)$
GLOB1	4	1	90	1805 sec	1892 sec	95.5	1.0
	6	1	90	1164 sec	1270 sec	91.5	1.5
	8	1	90	876 sec	993 sec	88.5	1.9
	12	1	90	598 sec	723 sec	83.0	2.6
	16	1	90	450 sec	594 sec	76.0	3.2
	24	1	90	322 sec	488 sec	66.0	3.9
DDM	4	5	84	1287 sec	1311 sec	98.0	1.0
	6	5	86	879 sec	896 sec	98.0	1.5
	8	5	84	641 sec	658 sec	97.5	2.0
	12	5	94	528 sec	552 sec	95.5	2.4
	16	5	84	313 sec	331 sec	94.5	4.0
	24	4	85	235 sec	261 sec	90.0	5.0

TAB. 5.5 – *Écoulement visqueux laminaire autour d'un profil d'aile* `NACA0012` *: cas test* **TVIS2**
Temps d'exécution pour le calcul de l'écoulement stationnaire (maillage N2)
Stratégies de résolution globale basée sur la méthode de relaxation de Jacobi
Stratégie de résolution par sous-domaine : système interface résolu par GMRES

Méthode	N_p	N_g	# it	CPU	REEL	% CPU
GLOB1	16	1	125	6211 sec	6903 sec	90.0
GLOB2	16	1	117	6527 sec	7255 sec	90.0
DDM	16	5	116	2043 sec	2090 sec	97.5

FIG. 5.5 – *Écoulement visqueux laminaire autour d'un profil d'aile* `NACA0012` *: cas test* **TVIS2**
Convergences non-linéaires des stratégies de résolution globale et par sous-domaine (maillage N1)
Axe X : itération en temps - Axe Y : résidu en échelle logarithmique

FIG. 5.6 – *Écoulement visqueux laminaire autour d'un profil d'aile* `NACA0012`
Cas test **TVIS2** (maillage N1)
Convergence linéaire de la stratégie de résolution par sous-domaine ($\varepsilon_i = 10^{-1}$/solveur local : 3 V(2,2,2)-cycles)
Axe X : itération en temps - Axe Y : # itérations GMRES

5.3.5 Cas test TVIS3

Les stratégies de résolution suivantes ont été utilisées :

GLOB1 : stratégie de résolution globale où, à chaque pas de temps, le système linéaire (5.16) est résolu par une méthode de Jacobi avec un seuil de résolution linéaire fixé à $\varepsilon_g = 10^{-1}$.

GLOB2 : stratégie de résolution globale où, à chaque pas de temps, le système linéaire (5.16) est résolu par une méthode de Jacobi avec un seuil de résolution linéaire fixé à $\varepsilon_g = 10^{-2}$.

DDM1 : stratégie de résolution par sous-domaine où, à chaque pas de temps, le système interface (5.26) est résolu par une méthode de GMRES sans ré-initialisation et non-préconditionnée, avec un seuil de résolution linéaire fixé à $\varepsilon_i = 10^{-1}$. Les systèmes linéaires locaux sont résolus par l'application d'un nombre variable de V-cycles, le seuil de résolution linéaire étant fixé à $\varepsilon_l = 10^{-1}$. Le V-cycle sélectionné est caractérisé par $\nu_1 = 2$ pré-lissages, $\nu_2 = 2$ post-lissages et $\nu_g = 2$ lissages sur le niveau le plus grossier.

DDM2 : stratégie de résolution par sous-domaine où, à chaque pas de temps, le système interface (5.26) est résolu par une méthode de GMRES sans ré-initialisation et non-préconditionnée, avec un seuil de résolution linéaire fixé à $\varepsilon_i = 10^{-1}$. Les systèmes linéaires locaux sont résolus par l'application d'un nombre constant de 3 V-cycles de la méthode multigrille par agglomération. Le V-cycle sélectionné est caractérisé par $\nu_1 = 2$ pré-lissages, $\nu_2 = 2$ post-lissages et $\nu_g = 2$ lissages sur le niveau le plus grossier (le nombre de niveaux grossiers est donné dans les tables).

Les nombres effectifs de pas de temps jusqu'à l'obtention de l'état stationnaire (résidu initial de l'énergie par unité de volume divisé par un facteur 10^{10}) et les temps d'exécution correspondants sont résumés dans les tables 5.6 (calculs effectués sur le maillage N1) et 5.7 (calculs effectués sur le maillage N2). La table 5.7 fournit un ensemble de résultats pour des calculs effectués sur le maillage N2 avec $N_p = 16$. Dans la table 5.6, l'accélération parallèle est calculée relativement au temps d'exécution mesuré pour $N_p = 4$.

Les convergences non-linéaires vers l'état stationnaire sont montrées sur la figure 5.7 (calculs effectués sur le maillage N1 pour $N_p = 24$). Sur cette figure, il est clair que la stratégie de résolution globale basée sur $\varepsilon_g = 10^{-1}$ pour le seuil de résolution linéaire, est loin d'être celle qui conduit à la convergence la plus rapide vers l'état stationnaire.

Comme pour les autres cas tests, la méthode de résolution par sous-domaine basée sur un nombre constant de V-cyles pour les résolutions locales est la plus efficace. Par exemple, pour les calculs réalisés sur le maillage N1, une comparaison avec la stratégie de résolution globale avec pour valeur du seuil de résolution linéaire $\varepsilon_g = 10^{-2}$, montre une réduction du temps total de simulation de l'ordre de 66% lorsque $N_p = 24$ (470 sec pour la stratégie de résolution globale contre 161 sec pour la méthode de résolution par sous-domaine utilisant 3 V-cycles pour les résolutions locales). La même comparaison pour les calculs réalisés avec le maillage N2 montre que la stratégie de résolution par sous-domaine est 4.2 fois plus rapide que la stratégie de résolution globale. Pour cette dernière, la réduction du seuil de résolution linéaire de $\varepsilon_g = 10^{-1}$ à $\varepsilon_g = 10^{-2}$ permet une convergence plus rapide vers l'état stationnaire (en termes du nombre de pas de temps) au détriment d'une augmentation de 36% du temps total de simulation. Finalement, la figure 5.8 montre le nombre effectif d'itérations de GMRES à chaque pas de temps pour la stratégie de résolution par sous-domaine basée sur un nombre constant de V-cycles pour les résolutions locales. On note que, comme pour les autres cas tests, le nombre d'itérations de GMRES n'augmente que faiblement lorsqu'on augmente le nombre de sous-domaines de $N_p = 4$ à $N_p = 24$.

TAB. 5.6 – *Ecoulement visqueux laminaire autour d'un profil d'aile* `NACA0012` *: cas test* **TVIS3**
Temps d'exécution pour le calcul de l'écoulement stationnaire (maillage N1)
Stratégies de résolution globale basée sur la méthode de relaxation de Jacobi
Stratégies de résolution par sous-domaine : système interface résolu par GMRES

Méthode	N_p	N_g	# it	CPU	REEL	% CPU	$S(N_p)$
GLOB1	4	1	155	1399 sec	1463 sec	95.5	1.0
	6	1	155	916 sec	999 sec	91.5	1.5
	8	1	155	670 sec	754 sec	89.0	2.0
	12	1	155	454 sec	552 sec	82.5	2.6
	16	1	155	323 sec	434 sec	74.0	3.4
	24	1	155	244 sec	367 sec	66.5	4.0
GLOB2	4	1	66	1690 sec	1769 sec	98.0	1.0
	6	1	66	1103 sec	1212 sec	91.0	1.5
	8	1	66	804 sec	937 sec	86.0	1.9
	12	1	66	555 sec	689 sec	80.5	2.5
	16	1	66	413 sec	552 sec	74.5	3.2
	24	1	66	295 sec	470 sec	63.0	3.8
DDM1	4	5	57	1163 sec	1202 sec	97.0	1.0
	6	5	58	764 sec	790 sec	96.5	1.5
	8	5	57	554 sec	586 sec	94.5	2.0
	12	5	63	419 sec	455 sec	92.0	2.6
	16	5	57	251 sec	284 sec	88.0	4.2
	24	4	58	177 sec	206 sec	86.0	5.8
DDM2	4	5	57	967 sec	979 sec	99.0	1.0
	6	5	58	646 sec	661 sec	98.0	1.5
	8	5	57	454 sec	472 sec	96.0	2.1
	12	5	62	350 sec	369 sec	95.0	2.7
	16	5	57	225 sec	243 sec	92.5	4.0
	24	4	58	145 sec	161 sec	90.0	6.1

TAB. 5.7 – *Ecoulement visqueux laminaire autour d'un profil d'aile* `NACA0012` *: cas test* **TVIS2**
Temps d'exécution pour le calcul de l'écoulement stationnaire (maillage N2)
Stratégies de résolution globale basée sur la méthode de relaxation de Jacobi
Stratégies de résolution par sous-domaine : système interface résolu par GMRES

Méthode	N_p	N_g	# it	CPU	REEL	% CPU
GLOB1	16	1	208	4693 sec	5242 sec	89.5
GLOB2	16	1	93	6420 sec	7152 sec	90.0
DDM1	16	5	93	2206 sec	2475 sec	89.0
DDM2	16	5	91	1677 sec	1715 sec	98.0

FIG. 5.7 – *Ecoulement visqueux laminaire autour d'un profil d'aile* `NACA0012` *: cas test* **TVIS3**
Convergences non-linéaires des stratégies de résolution globale et par sous-domaine (maillage N1)
Axe X : itération en temps - Axe Y : résidu en échelle logarithmique

FIG. 5.8 – *Ecoulement visqueux laminaire autour d'un profil d'aile* `NACA0012`
Cas test **TVIS3** (maillage N1)
Convergence linéaire de la stratégie de résolution par sous-domaine ($\varepsilon_i = 10^{-1}$/solveur local : 3 V(2,2,2)-cycles)
Axe X : itération en temps - Axe Y : # itérations GMRES

5.4 Conclusion

Dans ce chapitre, on a proposé et étudié d'un point de vue numérique une méthode par sous-domaine pour le calcul de solutions stationnaires du système d'équations de Navier-Stokes modélisant l'écoulement d'un fluide compressible visqueux en régime laminaire. Comme pour la résolution du système d'équations d'Euler (voir le chapitre 1), la formulation en continu de cette méthode consiste en un algorithme de type Schwarz additif sur une décomposition sans recouvrement du domaine de calcul. Les conditions imposées aux interfaces entre sous-domaines voisins sont issues de la formulation faible du problème aux valeurs initiales et aux limites associé (conditions d'interface « naturelles » ou « classiques »). Elles expriment la continuité des composantes du flux normal à l'interface sachant que celles-ci combine des termes de flux convectif et des termes de flux diffusif (ou flux visqueux).

La discrétisation en espace du système d'équations de Navier-Stokes s'appuie sur une formulation mixte éléments finis/volumes finis en maillages triangulaires non-structurés. L'intégration en temps des équations semi-discrétisées fait appel à un schéma implicite linéarisé qui conduit à la résolution d'un système linéaire de matrice creuse à structure irrégulière et non-symétrique. On distingue alors deux approches pour résoudre ce système :

- une stratégie globale qui utilise une méthode de relaxation de Jacobi naturellement parallélisable;
- une stratégie par sous-domaine qui repose sur la méthode proposée dans ce chapitre.

La mise en œuvre de la stratégie de résolution par sous-domaine, dans le cadre de la méthode de discrétisation mixte éléments finis/volumes finis, présente deux caractéristiques principales :

- alors que le traitement de la partie hyperbolique des conditions de raccord est précisément celui adopté pour la résolution du système d'équations d'Euler (voir la section 2.3 du chapitre 2), on a choisi ici d'imposer la continuité de toutes les composantes du flux diffusif au lieu des composantes normales seulement. Ce choix a été fait dans un souci de simplification du traitement en discret de la condition de raccord des termes de flux diffusif;
- afin de permettre une résolution itérative efficace des problèmes locaux qui caractérisent la stratégie par sous-domaine, on utilise une méthode multigrille linéaire par agglomération de volumes où le lisseur est une méthode de relaxation de Gauss-Seidel par point.

Les deux stratégies de résolution (globale et par sous-domaine) on été évaluées en détails par leur application au calcul d'écoulements stationnaires autour d'un profil d'aile `NACA0012`. Différentes situations (écoulements de référence subsonique, transsonique et supersonique) ont été considérées afin de pouvoir juger quelque peu du comportement de la stratégie de résolution par sous-domaine en fonction du régime de l'écoulement. Les figures 5.4, 5.6 et 5.8 tendent à montrer que la méthode en question à un comportement plutôt peu sensible à l'écoulement considéré (sous la réserve que les situations sélectionnées sont caractérisées par des valeurs relativement faibles du nombre de Reynolds). Comme on pouvait s'y attendre, l'efficacité numérique de la méthode est meilleure lorsque l'écoulement de référence est supersonique.

L'utilisation d'une méthode multigrille pour la résolution des systèmes locaux conduit à une stratégie de résolution par sous-domaine compétitive (en termes de temps de calcul) avec la méthode de résolution globale qui a servi de référence aux expériences numériques réalisées dans cette étude. Des gains substantiels ont été obtenus lorsqu'on procède à des résolutions approchées des systèmes locaux. Ici, cette option n'a été justifiée que numériquement et est principalement motivée par le fait que, dans la méthode de

résolution globale, on opte déjà pour une résolution inexacte des systèmes résultant du schéma implicite linéarisé. On notera que ce type d'approche fait actuellement l'objet de plusieurs études, tant théoriques que numériques, comme par exemple celles menées au CERFACS[BF00a]-[BF00b]. Par ailleurs, l'efficacité parallèle démontrée par la stratégie de résolution par sous-domaine laisse entrevoir des gains plus importants pour des calculs effectués sur des plus grands nombres de sous-domaines ($N_p > 24$).

Chapitre 6

Calcul d'écoulements instationnaires

Dans ce chapitre on s'intéresse à l'application de la méthode de résolution par sous-domaine du système d'équations de Navier-Stokes proposée dans la section 5.2, à la simulation d'écoulements visqueux laminaires instationnaires. L'objectif visé ici est avant tout la validation de la stratégie de résolution par sous-domaine dans un contexte où la méthode d'intégration en temps influe sur la précision des résultats obtenus. Cette étude nécessite une étape préliminaire qui a pour objet l'adaptation de la méthode implicite linéarisée décrite dans la section 5.1.4 afin d'augmenter la précision en temps.

6.1 Une méthode implicite précise au second ordre

La méthode d'intégration en temps décrite dans la section 5.1.4 repose sur le schéma d'Euler implicite que nous écrivons ici sous la forme vectorielle suivante :

$$D\frac{W_h^{n+1} - W_h^n}{\Delta t} + \frac{\partial \Psi_h^p(W_h^n)}{\partial W_h}\left(W_h^{n+1} - W_h^n\right) = -\Psi_h^p(W_h^n) \tag{6.1}$$

où la matrice diagonale D est définie par :

$$D_i = \text{aire}(C_i)$$

et où on convient que :

- W_h^n désigne la solution discrète aux nœuds du maillage triangulaire $\mathcal{T}_h$ calculée au temps $t^n = t^0 + n\Delta t$;
- $\Psi_h^p(W_h^n)$ désigne le vecteur flux nodal précis à l'ordre p, dont les composantes pour $p = 2$ sont données par l'équation (5.11).

Dans le cas $p = 2$, la matrice Jacobienne $\dfrac{\partial \Psi_h^2(W_h^n)}{\partial W_h}$ est difficile, voire impossible, à calculer. Par ailleurs, l'approximation au second ordre des flux numériques (voir les équations (2.4) à (2.12)) conduit à une matrice Jacobienne de largeur de bande accrue (la matrice obtenue est creuse, irrégulière et non-symétrique) nécessitant une capacité mémoire importante pour son stockage. Face à cette situation, la solution généralement adoptée (comme c'est le cas dans la présente étude) consiste à remplacer le calcul de la matrice Jacobienne exacte par celui d'une matrice Jacobienne approchée suivant un certain nombre de simplifications. La plus importante de ces simplifications consiste à baser le calcul de la matrice Jacobienne sur la linéarisation d'un flux nodal précis au premier ordre. Dans ces conditions, on obtient le schéma hybride suivant :

$$D\frac{W_h^{n+1}-W_h^n}{\Delta t}+\frac{\partial\Psi_h^1(W_h^n)}{\partial W_h}\left(W_h^{n+1}-W_h^n\right)=-\Psi_h^2(W_h^n) \tag{6.2}$$

On rappelle que dans le cas présent, le flux nodal est la somme d'un terme de flux convectif et d'un terme de flux diffusif. En pratique, la matrice Jacobienne approchée $\frac{\partial\Psi_h^1(W_h^n)}{\partial W_h}$ est alors basée sur :

- une linéarisation approchée du terme de flux convectif (2.4) qui est précis au premier ordre,
- une linéarisation exacte du flux diffusif discret (5.8) qui est, quant à lui, précis au second ordre.

Il est clair qu'à convergence l'équation (6.2) conduit à une approximation précise au second ordre du système d'équations de Navier-Stokes stationnaire. La méthode (6.2) définit une méthode de Newton inexacte qui n'est pas caractérisée par une convergence quadratique lorsque $\Delta t \longrightarrow +\infty$. En revanche les systèmes linéaires obtenus sont plus faciles à résoudre (en particulier, de largeur de bande minimale). On peut notamment démontrer que pour certains problèmes modèles linéaires, la matrice Jacobienne en question est à diagonale dominante (voir par exemple Fezoui $et\ al.$[FLLO89]), autorisant ainsi l'utilisation de méthodes de relaxation de Jacobi ou Gauss-Seidel pour résoudre les systèmes linéaires correspondants.

L'idée est alors de construire un schéma d'intégration implicite précis au moins au second ordre en temps, basé sur l'utilisation de la matrice Jacobienne approchée $\frac{\partial\Psi_h^1(W_h^n)}{\partial W_h}$. Pour ce faire, nous adoptons dans cette étude la méthode proposée par Martin et Guillard[MG96] dont on rappelle les principes de base dans ce qui suit. Supposons que l'on cherche à résoudre un système d'équations aux dérivées partielles écrit ici sous la forme abstraite suivante :

$$\left\{\begin{array}{lcl}\dfrac{\partial W}{\partial t}+\mathcal{L}(W) & = & 0\\ W(t=0) & = & W^0\end{array}\right. \tag{6.3}$$

où la solution W est cherchée dans un espace de Hilbert noté H et on suppose que $W^0 \in H$; d'autre part, $\mathcal{L}(W)$ désigne un opérateur aux dérivées partielles dont le domaine de définition est $D(H)$, supposé inclus dans H. Une version semi-discrète de (6.3) est obtenue en définissant un espace d'approximation de dimension finie V_h inclus dans H et un opérateur $\Psi_h\ :\ V_h \longrightarrow V_h$ conduisant au système d'équations différentielles ordinaires :

$$\left\{\begin{array}{lcl}\dfrac{dW_h}{dt}+\Psi_h(W_h) & = & 0\\ W_h(t=0) & = & W_h^0\end{array}\right. \tag{6.4}$$

Dans ce contexte, il est classique de supposer qu'il existe une restriction stable $R_h\ :\ H \longrightarrow V_h$ telle que :

$$\lim_{h\longrightarrow 0}\parallel R_hW-W\parallel_H=0 \quad \forall W\in H \quad \text{et} \quad \parallel R_h\parallel=\sup_{\|W\|=1}\parallel R_hW\parallel_H\leq C$$

L'opérateur Ψ_h^p désigne une approximation d'ordre p de l'opérateur $\mathcal{L}$ si l'erreur de troncature définie par $T_h(W)=\parallel R_h\mathcal{L}(W)-\Psi_h^p(R_hW)\parallel_H$ est telle que :

$$T_h(W)<C_1(W)h^p \quad \forall W\in\mathcal{D} \tag{6.5}$$

où $\mathcal{D}$ est un sous-ensemble dense de H. L'équation (6.4) est alors discrétisée en temps en utilisant :

$$\frac{dW}{dt} \approx \frac{aW(t+\Delta t) - V^\star\left[W(t),W(t-\Delta t),\cdots,W(t-j\Delta t)\right]}{\Delta t} \tag{6.6}$$

où $V^\star\left[W(t),W(t-\Delta t),\cdots,W(t-j\Delta t)\right]$ désigne une fonction linéaire à j arguments. La formule de discrétisation (6.6) est une approximation d'ordre p de la dérivée temporelle si l'erreur de troncature associée vérifie :

$$\left|\left|\frac{dW}{dt} - \frac{aW(t+\Delta t) - V^\star\left[W(t),W(t-\Delta t),\cdots,W(t-j\Delta t)\right]}{\Delta t}\right|\right| < C_2(W)\Delta t^p \tag{6.7}$$

En combinant les éléments précédents, une discrétisation implicite en temps de (6.3) est donc donnée par :

$$\frac{aW_h^{n+1} - V^\star\left[W_h^n,W_h^{n-1},\cdots,W_h^{n-j}\right]}{\Delta t} + \Psi_h^p(W_h^{n+1}) = 0 \tag{6.8}$$

On note $G_h(\Delta t) \ : \ V_h \longrightarrow V_h$ l'opérateur qui associe à $V^\star\left[W_h^n,W_h^{n-1},\cdots,W_h^{n-j}\right]$, la solution W_h^{n+1} du système d'équations non-linéaires (6.8). L'erreur de troncature locale associée à (6.8) est définie par :

$$T_h = \frac{R_hW(t+\Delta t) - G_h(\Delta t)R_hV^\star\left[W(t),W(t-\Delta t),\cdots,W(t-j\Delta t)\right]}{\Delta t} \tag{6.9}$$

et on peut vérifier qu'en vertu de (6.5) et (6.7), T_h définit une approximation d'ordre p de (6.3) si l'opérateur $\left(a+\Delta t\Psi_h^p\right)^{-1}$ est uniformément borné[1]. Pour résoudre (6.8), on fait appel à une méthode de Newton qui pose alors deux problèmes principaux : le calcul de la matrice Jacobienne $\dfrac{\partial\Psi_h^p(W_h^n)}{\partial W_h}$ et la résolution d'un système linéaire de la forme $\left(a\mathrm{Id} + \Delta t\dfrac{\partial\Psi_h^p(W_h^n)}{\partial W_h}\right)X = b$. Pour les raisons soulevées au début de cette section, on opte ici pour la résolution de (6.8) par une méthode de Newton modifiée reposant sur le choix d'une approximation d'ordre m ($m \leq p$) du terme de flux nodal. On est alors amené à construire une matrice Jacobienne approchée $\dfrac{\partial\Psi_h^m(W_h^n)}{\partial W_h}$ qui est utilisée dans une itération de type « défaut corrigé » (notée DeC dans la suite) :

$$\left\{\begin{array}{rcl} \mathcal{W}_h^0 & = & \mathcal{W}^0\left[W_h^n,W_h^{n-1},\cdots,W_h^{n-j}\right] \\ \mathcal{W}_h^{s+1} & = & \mathcal{W}_h^s - \Delta t\left[a\mathrm{Id} + \Delta t\dfrac{\partial\Psi_h^m(W_h^n)}{\partial W_h}\right]^{-1} \times \\ & & \left(\dfrac{a\mathcal{W}_h^s - V^\star\left[W_h^n,W_h^{n-1},\cdots,W_h^{n-j}\right]}{\Delta t} + \Psi_h^p(\mathcal{W}_h^s)\right) \end{array}\right. \tag{6.10}$$

Chaque itération de la forme (6.10) définit un opérateur $G_h^s(\Delta t)$ qui associe à $V^\star$ le vecteur $\mathcal{W}_h^s$. En notant T_h^s l'erreur de troncature associée, on déduit de (6.10) que T_h^s vérifie :

$$T_h^s = A^s\left(T_h^0 - T_h\right) + T_h \tag{6.11}$$

1. On convient que $a+\Delta t\Psi_h^p$ est l'opérateur non-linéaire qui à U associe $aU + \Delta t\Psi_h^p(U)$.

avec :

$$A = \mathrm{Id} - \left[a\mathrm{Id} + \Delta t \frac{\partial \Psi_h^m(W_h^n)}{\partial W_h}\right]^{-1} \left[a + \Delta t \Psi_h^p\right]$$

ou T_h^0 est l'erreur de troncature associée à l'initialisation de la méthode de Newton :

$$T_h^0 = \frac{R_h W(t+\Delta t) - R_h \mathcal{W}^0\left[W(t), W(t-\Delta t), \cdots, W(t-j\Delta t)\right]}{\Delta t}$$

Si le rayon spectral $\rho(A) < 1$, (6.11) montre que l'erreur de troncature $T_h^s \longrightarrow T_h$. Cependant, (6.11) montre aussi qu'il n'est pas utile de converger complètement l'itération DeC (6.10) pour obtenir une solution précise à l'ordre p. En effet, il suffit de réaliser s_p itérations de telle sorte que le premier terme de (6.10) soit d'ordre p, c'est-à-dire $s_p = \frac{\frac{\ln(T_h)}{T_h^0}}{\ln(\rho(A))}$. L'étape suivante consiste à déterminer une valeur utilisable en pratique du nombre d'itérations s_p pour l'obtention d'une solution précise à l'ordre p. On suppose que $\mathcal{W}^0$ est une approximation d'ordre l de $W(t+\Delta t)$. La proposition suivante est énoncée dans [MG96].

Proposition 1 L'erreur de troncature T_h^s est au moins d'ordre p dès que $s_p \geq \dfrac{p-l+1}{m+1}$.

On s'intéresse ici plus particulièrement au cas où $m = 1$. On en déduit les corollaires suivants.

Corollaire 1 Soit Ψ_h^2 une approximation du second ordre de l'opérateur $\mathcal{L}$, alors :

- si $\mathcal{W}_h^0 = 0$, $s_p \geq 2$ itérations DeC (6.10) sont nécessaires à l'obtention d'une solution précise au second ordre en temps et en espace;
- si $\mathcal{W}_h^0 = W_h^n$, $s_p \geq 1$ itération DeC (6.10) suffit à obtenir une solution précise au second ordre en temps et en espace.

Corollaire 2 Soit Ψ_h^3 une approximation du troisième ordre de l'opérateur $\mathcal{L}$, alors :

- si $\mathcal{W}_h^0 = W_h^n$, $s_p \geq 2$ itérations DeC (6.10) sont nécessaires à l'obtention d'une solution précise au troisième ordre en temps et en espace;
- si $\mathcal{W}_h^0 = 2W_h^n - W_h^{n-1}$ (c'est-à-dire si $\mathcal{W}_h^0$ est une approximation du second ordre de $W(t+\Delta t)$), $s_p \geq 1$ itération DeC (6.10) suffit à obtenir une solution précise au troisième ordre en temps et en espace.

Les corollaires 1 et 2 montrent donc que, si l'initialisation des itérations DeC (6.10) est correctement choisie, des approximations précises au second ordre ou au troisième ordre de la solution sont a priori obtenues au prix d'une seule inversion d'un système linéaire dont la matrice est la Jacobienne d'un flux précis au premier ordre. Cependant, des expériences numériques menées dans [MG96] portant sur le calcul de l'écoulement instationnaire dans une géométrie de moteur à piston, montrent qu'il est préférable de réaliser deux itérations DeC au prix de l'inversion de deux systèmes linéaires dont la matrice est la Jacobienne d'un flux précis au premier ordre. En résumé, la méthode implicite linéarisée adoptée en pratique comporte donc deux étapes :

$$\begin{aligned}
\left[\frac{3}{2}\mathrm{Id} + \Delta t \frac{\partial \Psi_h^1(W_h^n)}{\partial W_h}\right](U - W_h^n) &= -\left(\frac{3}{2}W_h^n + \Delta t \Psi_h^2(W_h^n)\right) + \left(2W_h^n - \frac{1}{2}W_h^{n-1}\right) \\
\left[\frac{3}{2}\mathrm{Id} + \Delta t \frac{\partial \Psi_h^1(W_h^n)}{\partial W_h}\right](W_h^{n+1} - U) &= -\left(\frac{3}{2}U + \Delta t \Psi_h^2(U)\right) + \left(2W_h^n - \frac{1}{2}W_h^{n-1}\right)
\end{aligned} \tag{6.12}$$

6.2 Résultats numériques

Le cas test considéré ici consiste en la simulation numérique de l'écoulement instationnaire autour d'un profil d'aile NACA0012. Le profil d'aile est mis initialement en forte incidence dans un écoulement uniforme subsonique (voir aussi [Lan91]). Les simulations numériques utilisent le maillage N1 de la table 5.1. Le nombre de Mach de référence est fixé à 0.2, le nombre de Reynolds étant égal à 2100. L'angle d'incidence du profil d'aile est fixé à $20°$. L'écoulement est simulé jusqu'au temps adimensionné $t = 56$.

Les lignes iso-densité instationnaires à différents temps compris entre $t = 52$ et $t = 56$ sont montrées sur les figures 6.1 à 6.3. Les figures 6.4 à 6.7 représentent l'évolution du coefficient de portance entre les temps $t = 0$ et $t = 56$. La simulation correspondant aux figures 6.1 à 6.3 a été réalisée avec la méthode implicite linéarisée d'ordre 2 (6.12) en utilisant un pas de temps global calculé à partir d'un CFL fixé à 100. On observe qu'un régime périodique de détachement de tourbillons est bien établi à partir du temps $t = 32$. Les courbes d'évolution du coefficient de portance permettent de comparer les solutions obtenues, d'une part, avec la méthode implicite linéarisée précise au premier ordre en temps (c'est-à-dire la méthode initialement décrite dans la section 5.1.4) et, d'autre part, avec la méthode implicite linéarisée précise au second ordre présentée dans ce qui précède. Ces courbes montrent clairement des comportements différents; d'une façon générale, la méthode d'intégration en temps précise au premier ordre tend à diffuser notablement les caractéristiques principales de l'écoulement.

Dans ce qui suit, nous considérons tout d'abord un certain nombre de simulations visant à justifier l'adoption de la méthode implicite linéarisée précise au second ordre, dans le cadre de la stratégie de résolution globale (c'est-à-dire que les systèmes linéaires sont résolus par la méthode de relaxation de Jacobi). Ensuite, nous procédons à une comparaison entre la stratégie globale et la stratégie par sous-domaine pour la résolution des systèmes linéaires intervenant dans les deux étapes (6.12).

Influence du pas de temps. Le calcul d'un écoulement instationnaire nécessite l'utilisation d'un pas de temps global obtenu comme le minimum des pas de temps calculés localement sur chaque cellule de contrôle. Le nombre de Courant (CFL) est alors utilisé comme un facteur multiplicatif du pas de temps global. La méthode implicite linéarisée pour l'intégration en temps des équations semi-discrétisées étant inconditionnellement stable, on peut a priori utiliser des grandes valeurs du pas de temps, c'est-à-dire des grandes valeurs du CFL. Cependant, une grande valeur du pas de temps se traduit aussi par une diffusion numérique plus importante et il s'agit donc de trouver une valeur du CFL qui conduise à une solution physiquement acceptable. Dans la situation qui nous intéresse ici, une telle évaluation ne peut se faire que numériquement. Nous avons donc réalisé trois simulations correspondant à des valeurs du CFL de 10, 100 et 1000, en utilisant à chaque fois la méthode implicite linéarisée précise au second ordre. La figure 6.8 montre l'évolution du coefficient de portance entre les temps $t = 44$ et $t = 56$ dans chaque cas de valeur du CFL. Les solutions obtenues pour CFL = 10 et CFL = 100 sont presque identiques alors que la valeur CFL = 1000 conduit à une solution éronnée. Des simulations supplémentaires permettraient d'obtenir une valeur plus précise du CFL maximum, néanmoins, dans la suite, nous utiliserons toujours la valeur CFL = 100.

Influence du seuil de résolution linéaire. On évalue ici l'influence de la précision de résolution des systèmes linéaires intervenant dans les deux étapes (6.12) sur la précision globale de la solution. On utilise donc la méthode implicite linéarisée précise au second ordre avec un pas de temps global calculé

FIG. 6.1 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* `NACA0012`
Lignes iso-densité : $t = 52$ (haut) et $t = 53$ (bas)

FIG. 6.2 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* NACA0012
Lignes iso-densité : $t = 54$ (haut) et $t = 55$ (bas)

FIG. 6.3 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* `NACA0012`
Lignes iso-densité : $t = 56$

FIG. 6.4 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* `NACA0012`
Évolution du coefficient de portance entre les temps $t = 8$ et $t = 20$
Axe X : temps - Axe Y : coefficient de portance
Algorithme implicite monogrille (Jacobi) : comparaison ordre 1/ordre 2, CFL = 100

FIG. 6.5 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* NACA0012
Évolution du coefficient de portance entre les temps $t = 20$ et $t = 32$
Axe X : temps - Axe Y : coefficient de portance
Algorithme implicite monogrille (Jacobi) : comparaison ordre 1/ordre 2, CFL = 100

FIG. 6.6 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* NACA0012
Évolution du coefficient de portance entre les temps $t = 32$ et $t = 44$
Axe X : temps - Axe Y : coefficient de portance
Algorithme implicite monogrille (Jacobi) : comparaison ordre 1/ordre 2, CFL = 100

FIG. 6.7 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* `NACA0012`
Évolution du coefficient de portance entre les temps $t = 44$ et $t = 56$
Axe X : temps - Axe Y : coefficient de portance
Algorithme implicite monogrille (Jacobi) : comparaison ordre 1/ordre 2, CFL = 100

FIG. 6.8 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* `NACA0012`
Évolution du coefficient de portance entre les temps $t = 44$ et $t = 56$
Axe X : temps - Axe Y : coefficient de portance
Algorithme implicite ordre 2 monogrille (Jacobi) : influence du nombre de Courant (CFL)

à partir de la valeur CFL = 100 et on considère trois simulations correspondant aux valeurs 10^{-1}, 10^{-2} et 10^{-3} du seuil de résolution linéaire ε_g. La figure 6.9 représente l'évolution du coefficient de portance entre les temps $t = 44$ et $t = 56$ dans chaque cas de valeur du paramètre ε_g. Cette figure montre que la valeur $\varepsilon_g = 10^{-2}$ conduit à une solution acceptable; en particulier il n'est pas nécessaire de résoudre plus précisément les systèmes linéaires ($\varepsilon_g = 10^{-3}$) au détriment d'un temps de calcul considérablement plus important.

FIG. 6.9 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* NACA0012
Évolution du coefficient de portance entre les temps $t = 44$ et $t = 56$
Axe X : temps - Axe Y : coefficient de portance
Algorithme implicite ordre 2 monogrille (Jacobi) : CFL = 100, influence du seuil de résolution linéaire ε_g

On conclut donc des deux séries d'expériences numériques précédentes que la configuration de référence d'utilisation de la méthode implicite linéarisée précise au second ordre est caractérisée par les valeurs CFL = 100 et $\varepsilon_g = 10^{-2}$.

On réalise maintenant une série d'expériences numériques visant à évaluer la stratégie par sous-domaine décrite dans la section 5.2, appliquée à la résolution des systèmes linéaires intervenant dans les deux étapes (6.12). On s'intéresse de près à l'influence de certains paramètres tels que les seuils de résolution des systèmes locaux et interfaces ou encore, la complexité de résolution des systèmes locaux. Plus précisément, on distingue dans ce qui suit les stratégies de résolution suivantes :

GLOB : méthode implicite linéarisée d'ordre 2, CFL = 100, stratégie de résolution globale basée sur la méthode de relaxation de Jacobi, $\varepsilon_g = 10^{-2}$.

DDM1 : méthode implicite linéarisée d'ordre 2, CFL = 100, stratégie de résolution par sous-domaine; système interface résolu par GMRES, $\varepsilon_i = 10^{-1}$; systèmes locaux résolus par MG-V(2,2,2), Jacobi comme lisseur, $\varepsilon_l = 10^{-1}$.

DDM2 : méthode implicite linéarisée d'ordre 2, CFL = 100, stratégie de résolution par sous-domaine; système interface résolu par GMRES, $\varepsilon_i = 10^{-1}$; systèmes locaux résolus par MG-V(2,2,2), Gauss-Seidel comme lisseur, $\varepsilon_l = 10^{-1}$.

DDM3 : méthode implicite linéarisée d'ordre 2, CFL = 100, stratégie de résolution par sous-domaine; système interface résolu par GMRES, $\varepsilon_i = 10^{-1}$; systèmes locaux résolus par MG-V(2,2,2), Jacobi comme lisseur, $N_c = 2$.

DDM4 : méthode implicite linéarisée d'ordre 2, CFL = 100, stratégie de résolution par sous-domaine; système interface résolu par GMRES, $\varepsilon_i = 10^{-1}$; systèmes locaux résolus par MG-V(4,4,4), Gauss-Seidel comme lisseur, $N_c = 1$.

Dans un premier temps, on évalue les stratégies de résolution ci-dessus du point de vue de la précision des solutions qu'elles fournissent. A la fin de cette section, on s'intéresse aux performances parallèles et compare les différentes stratégies en termes des temps de calcul.

Comparaison à seuils de résolution linéaire fixés. La figure 6.10 compare les stratégies de résolution **GLOB** et **DDM1**. Ces courbes montrent que la stratégie **DDM1**, qui est basée sur des seuils de résolution linéaire respectivement fixés à $\varepsilon_i = 10^{-1}$ pour la résolution des systèmes interface par la méthode GMRES et $\varepsilon_l = 10^{-1}$ pour la résolution des systèmes locaux par la méthode multigrille par agglomération, conduit à une solution très comparable à celle résultant de la stratégie de résolution globale de référence. On voit donc l'avantage à utiliser une méthode multigrille pour la résolution des systèmes locaux : à seuils de résolution égaux, la solution obtenue par la méthode multigrille est de meilleure qualité que celle calculée par la méthode monogrille; pour cette dernière, il est nécessaire de résoudre plus précisément les systèmes linéaires comme l'illustre la figure 6.9.

Simulations à complexités de résolution linéaire constantes. Plutôt que de régir les résolutions locales en fixant le seuil de résolution linéaire et en laissant le nombre de V-cycles varier d'un sous-domaine à un autre, on peut préférer fixer ce nombre à une valeur constante durant la simulation (c'est-à-dire, une valeur valable à chaque pas de temps et pour tous les sous-domaines). La figure 6.11 compare les stratégies de résolution **DDM1** et **DDM3**. La stratégie **DDM3** est caractérisée par un nombre de V(2,2,2)-cycles fixé à 2, le lisseur étant une méthode de relaxation de Jacobi. Les solutions obtenues sont là encore très comparables. La figure 6.12 compare les stratégies de résolution **DDM3** et **DDM4**. La stratégie **DDM4** est caractérisée par un nombre de V(4,4,4)-cycles fixé à 1, le lisseur étant une méthode de relaxation de Gauss-Seidel.

Influence du nombre de sous-domaines. On utilise de nouveau la stratégie de résolution **DDM4** et on réalise trois simulations pour des nombres de sous-domaines N_p respectivement fixés à 8, 16 et 24. Les figures 6.13 à 6.15 représentent l'évolution du nombre effectif d'itérations de la méthode de GMRES en fonction de l'itération en temps. Pour $N_p = 8$ et $N_p = 16$, le nombre d'itérations de GMRES varie essentiellement entre 2 et 3; pour $N_p = 24$, un nombre important de pas de temps ne demandent qu'une itération de GMRES pour la résolution du système interface. D'une façon générale, la résolution de ce système est peu sensible au nombre de sous-domaines. Ce comportement favorable (nombre réduit d'itérations de GMRES et quasi-indépendance vis-à-vis du nombre de sous-domaines) est ici dû, d'une part, au fait que les systèmes interfaces ne sont pas résolus avec une grande précision

FIG. 6.10 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* `NACA0012`
Évolution du coefficient de portance entre les temps $t = 44$ et $t = 56$
Axe X : temps - Axe Y : coefficient de portance
Implicite ordre 2, CFL = 100
Stratégie de résolution globale : Jacobi, $\varepsilon_g = 10^{-2}$ (stratégie **GLOB**)
Stratégie de résolution par sous-domaine **DDM1** :
- solveur interface GMRES $\varepsilon_i = 10^{-1}$
- solveur local MG-V(2,2,2) et $\varepsilon_l = 10^{-1}$ (Jacobi comme lisseur)

FIG. 6.11 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* `NACA0012`
Évolution du coefficient de portance entre les temps $t = 44$ et $t = 56$
Axe X : temps - Axe Y : coefficient de portance
Implicite ordre 2, CFL = 100
Stratégie de résolution par sous-domaine **DDM1** :
- solveur interface GMRES $\varepsilon_i = 10^{-1}$
- solveur local MG-V(2,2,2) et $\varepsilon_l = 10^{-1}$ (Jacobi comme lisseur)

Stratégie de résolution par sous-domaine **DDM3** :
- solveur interface GMRES $\varepsilon_i = 10^{-1}$
- solveur local MG-V(2,2,2) et $N_c = 2$ (Jacobi comme lisseur)

FIG. 6.12 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* `NACA0012`
Évolution du coefficient de portance entre les temps $t = 44$ et $t = 56$
Axe X : temps - Axe Y : coefficient de portance
Implicite ordre 2, CFL = 100
Stratégie de résolution par sous-domaine **DDM3** :
- solveur interface GMRES $\varepsilon_i = 10^{-1}$
- solveur local MG-V(2,2,2) et $N_c = 2$ (Jacobi comme lisseur)
Stratégie de résolution par sous-domaine **DDM4** :
- solveur interface GMRES $\varepsilon_i = 10^{-1}$
- solveur local MG-V(4,4,4) et $N_c = 1$ (Gauss-Seidel comme lisseur)

(le seuil correspondant est fixé à $\varepsilon_i = 10^{-1}$) et, d'autre part, au relativement bon conditionnement des systèmes linéaires du fait de l'utilisation d'un pas de temps global.

Comparaison des temps de calcul. Les différentes stratégies de résolution considérées précédemment ont été testées pour des nombres de sous-domaines égaux à 8, 16 et 24. Les temps de calcul obtenus sont rassemblés dans la table 6.1. On constate tout d'abord que l'efficacité parallèle de la stratégie de résolution globale se dégrade notablement et plus rapidement qu'à l'occasion des expériences numériques menées en stationnaire (voir la section 5). Cette dégradation de l'efficacité parallèle résulte du poids plus important donné ici aux phases de résolution linéaire à l'aide de la méthode de relaxation de Jacobi du fait, d'une part, d'un seuil de résolution plus contraignant ($\varepsilon_g = 10^{-2}$) et, d'autre part, de la résolution de deux systèmes au lieu d'un à chaque pas de temps avec la méthode implicite linéarisée précise au second ordre. Pour $N_p = 24$, les différentes stratégies de résolution par sous-domaine démontrent des efficacités parallèles 20% supérieures à celle de la stratégie de résolution globale. Comme dans le cas stationnaire, seules les stratégies de résolution par sous-domaine basées sur une complexité constante pour la résolution des systèmes locaux (stratégies **DDM3** et **DDM4**) deviennent compétitives avec la stratégie de résolution globale lorsque le nombre de sous-domaines croît. Compte tenu des résultats obtenus ici, on peut raisonnablement prévoir que les stratégies de résolution par sous-domaines **DDM3** et **DDM4** seront plus performantes que la stratégie **GLOB** pour des valeurs de $N_p \geq 32$.

FIG. 6.13 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* NACA0012
Nombre d'itérations de GMRES pour la résolution des systèmes interface
Axe X : itération en temps - Axe Y : # itérations GMRES
Implicite ordre 2, CFL = 100, $N_p = 8$
Stratégie de résolution par sous-domaine **DDM4** :
- solveur interface GMRES $\varepsilon_i = 10^{-1}$
- solveur local MG-V(4,4,4) et $N_c = 1$ (Gauss-Seidel comme lisseur)

FIG. 6.14 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* NACA0012
Nombre d'itérations de GMRES pour la résolution des systèmes interface
Axe X : itération en temps - Axe Y : # itérations GMRES
Implicite ordre 2, CFL = 100, $N_p = 16$
Stratégie de résolution par sous-domaine **DDM4** :
- solveur interface GMRES $\varepsilon_i = 10^{-1}$
- solveur local MG-V(4,4,4) et $N_c = 1$ (Gauss-Seidel comme lisseur)

TAB. 6.1 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* NACA0012
Temps d'exécution : $t = 56$

Méthode	N_p	N_g	# it	CPU	REEL	% CPU
GLOB	8	1	4662	16852 s	19029 s	88.5
-	16	1	4662	8608 s	11580 s	74.4
-	24	1	4662	6050 s	9021 s	67.0
DDM1	8	4	4661	33551 s	35629 s	94.1
-	16	4	4661	17918 s	19162 s	93.5
-	24	4	4660	10867 s	12238 s	88.8
DDM2	8	4	4661	29988 s	31547 s	95.0
-	16	4	4661	14951 s	17031 s	87.8
-	24	4	4660	10444 s	11979 s	87.2
DDM3	8	4	4662	27904 s	29195 s	95.6
-	16	4	4662	13247 s	14634 s	90.5
-	24	4	4660	8328 s	9664 s	86.1
DDM4	8	4	4662	26457 s	27787 s	95.2
-	16	4	4662	12752 s	14114 s	90.3
-	24	4	4660	8247 s	9546 s	86.4

FIG. 6.15 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* `NACA0012`
Nombre d'itérations de GMRES pour la résolution des systèmes interface
Axe X : itération en temps - Axe Y : # itérations GMRES
Implicite ordre 2, CFL = 100, $N_p = 24$
Stratégie de résolution par sous-domaine **DDM4** :
- solveur interface GMRES $\varepsilon_i = 10^{-1}$
- solveur local MG-V(4,4,4) et $N_c = 1$ (Gauss-Seidel comme lisseur)

Apport de la méthode implicite linéarisée précise au second ordre. L'objectif de ce dernier paragraphe de résultats est de démontrer l'intérêt de la méthode implicite linéarisée précise au second ordre. Pour cela, on compare les solutions obtenues avec cette méthode pour CFL = 100 (solution de référence) avec celles qui résultent de l'utilisation de la méthode implicite linéarisée précise au premier ordre pour différentes valeurs du CFL. Ici, il s'agit en premier lieu d'évaluer la valeur du CFL la plus acceptable, c'est-à-dire celle qui permet d'obtenir une solution comparable à la solution de référence. La figure 6.16 montre l'évolution du coefficient de portance entre les temps $t = 44$ et $t = 56$, correspondant à l'utilisation de la méthode implicite linéarisée précise au premier ordre pour différentes valeurs du CFL. Cette figure inclut aussi la solution de référence. La comparaison des différentes solutions montre qu'il semble difficile de dépasser CFL=10 avec la méthode précise au premier ordre. Les temps de calcul sont résumés dans la table 6.2. En conclusion, la méthode implicite linéarisée précise au second ordre permet de réduire le temps de calcul total d'un facteur proche de 3.

FIG. 6.16 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* NACA0012
Évolution du coefficient de portance entre les temps $t = 44$ et $t = 56$
Axe X : temps - Axe Y : coefficient de portance
Algorithme implicite monogrille (Jacobi) : comparaison ordre 1/ordre 2

6.3 Conclusion

Le calcul d'écoulements instationnaires tient de nos jours une part de plus en plus importante dans les applications de la mécanique des fluides compressibles (voir par exemple Martin[Mar96] et Viozat[Vio98]). Les simulations numériques portent généralement sur des durées longues afin de bien appréhender les différents régimes (transitoires et permanents) qui caractérisent les écoulements en question. Dans ce contexte, il est nécessaire de disposer de méthodes de calcul précises (en espace et en temps) et efficaces

TAB. 6.2 – *Écoulement visqueux laminaire instationnaire autour d'un profil d'aile* NACA0012
Stratégie de résolution globale : algorithme implicite monogrille (Jacobi)
Temps d'exécution : $t = 56$, $N_p = 16$

Méthode	CFL	N_g	# it	CPU	REEL	% CPU
Ordre 1	10	1	46592	27285 s	32027 s	85.2
Ordre 2	100	1	4662	8608 s	11580 s	74.4

(permettant l'utilisation de grands pas de temps). Ce chapitre est une contribution dans ce domaine où l'on s'est intéressé principalement aux aspects relevant de l'efficacité numérique et informatique des méthodes. Pour ce faire, on a adapté et appliqué la méthode de résolution par sous-domaine du système d'équations de Navier-Stokes proposée dans la section 5.2 à la simulation d'écoulements visqueux laminaires instationnaires. L'objectif visé ici était avant tout la validation de la stratégie de résolution par sous-domaine dans un contexte où la méthode d'intégration en temps influe sur la précision des résultats obtenus.

Cette étude a nécessité une étape préliminaire qui a eu pour objet la prise en compte d'une méthode implicite linéarisée précise au second ordre en temps, initialement proposée par Martin et Guillard[MG96]. Chaque pas de temps conduit à la résolution de deux systèmes linéaires (au lieu d'un seul dans le cas de la méthode implicite linéarisée utilisée pour le calcul d'écoulements stationnaires) dont la matrice est la Jacobienne d'un flux précis au premier ordre. Ces systèmes linéaires sont alors résolus soit par une stratégie globale reposant sur la méthode de relaxation de Jacobi, soit à l'aide de la stratégie par sous-domaine proposée dans cette étude. On notera au passage que ces systèmes linéaires sont à priori mieux conditionnés que ceux caractérisant le calcul d'écoulements stationnaires du fait de l'utilisation d'un pas de temps global. Dans ce contexte, la stratégie de résolution globale basée sur la méthode de relaxation de Jacobi permet de traiter efficacement les systèmes linéaires en question, tout au moins pour des tailles de problèmes modérées. Ici encore, les systèmes locaux qui caractérisent la stratégie par sous-domaine sont résolus de façon inexacte à l'aide d'une méthode multigrille linéaire par agglomération de volumes.

Une étude numérique détaillée a été réalisée dans le cadre du calcul de l'écoulement instationnaire autour d'un profil d'aile NACA0012 mis initialement en forte incidence dans un écoulement uniforme subsonique. Dans un premier temps, nous avons évalué la valeur maximale du nombre de courant utilisable avec la méthode implicite linéarisée précise au second ordre, sans effet notable sur la précision de la solution calculée (influence du terme d'erreur de nature diffusive). Pour le cas test considéré, la valeur CFL = 100 a été retenue. De son côté, la méthode implicite linéarisée précise au premier ordre ne permet pas de dépasser la valeur CFL = 10 pour ce même cas test. Dans ces conditions, des calculs réalisés avec la stratégie de résolution globale sur $N_p = 16$ nœuds de calculs, dont les résultats sont résumés dans la table 6.2, on permis de mettre en évidence un gain de l'ordre de 3 lié à l'utilisation de la méthode implicite linéarisée précise au second ordre. On notera au passage dans cette table que l'efficacité parallèle se dégrade lorsqu'on utilise cette méthode en raison de l'action combinée de deux facteurs : d'une part les systèmes linéaires obtenus pour la valeur CFL = 100 sont plus difficiles à résoudre à un même seuil que ceux caractérisant la valeur CFL = 10 et, d'autre part, la méthode de Jacobi bien que naturellement parallélisable voit son efficacité parallèle se dégrader rapidement lorsque le nombre de relaxations augmente. De ce point de vue, la stratégie de résolution par sous-domaine proposée ici est caractérisée par des efficacités parallèles en moyennes 20% supérieures à celles démontrées par la stratégie de résolution globale pour une décomposition en $N_p = 24$ sous-domaines (voir la table 6.1).

Pour cette valeur de N_p, les deux stratégies sont globalement de coût équivalent mais il est clair que la supériorité de la stratégie de résolution par sous-domaine sera d'autant plus flagrante que le nombre de sous-domaines augmentera.

Conclusion générale

La mécanique des fluides numérique a réalisé des progrès spectaculaires ces vingt dernières années, autant par la sophistication accrue des méthodes d'approximation (solveurs de Riemann, schémas TVD, etc), des modèles physiques considérés (de l'aérodynamique au traitement de la combustion, l'hypersonique, la turbulence, les écoulements multifluides, multiphasiques, tridimensionnels, ou couplés à d'autres disciplines), que dans les techniques de résolution des grands systèmes linéaires ou non-linéaires discrets (méthodes multigrilles, méthodes par décomposition de domaine, calcul parallèle, etc). Malgré ces progrès évidents, le sujet continue de retenir l'intérêt des chercheurs de plusieurs disciplines (mécaniciens, numériciens, spécialistes de calcul haute performance, etc) autant que celui des ingénieurs. Cette thèse a visé à contribuer à la diminution des coûts de calcul d'écoulements représentatifs de la mécanique des fluides compressibles, en considérant plus particulièrement l'étape de résolution des systèmes discrets issus des modèles d'équations d'Euler et de Navier-Stokes en deux dimensions d'espace.

En général, la complexité des calculs provient de deux origines principales. Elle est tout d'abord liée aux phénomènes physiques que l'on cherche à modéliser. La turbulence par exemple, fait intervenir de nombreuses échelles en interaction que l'on ne peut, ou ne souhaîte pas toutes résoudre finement. Dans d'autres applications, si l'écoulement est réactif, le modèle physique doit être étendu pour prendre en compte l'évolution d'espèces chimiques, et couplé à un modèle représentant la réaction chimique en jeu (une combustion par exemple). Deuxièmement, la complexité du calcul provient souvent de la géométrie du domaine. De nos jours, le monde industriel a de plus en plus besoin de simuler des phénomènes physiques complexes dans des géométries réalistes c'est-à-dire représentatives, sinon proches d'une véritable configuration. La nécessité de construire des méthodes numériques efficaces est particulièrement impérative lorsque la simulation numérique est réalisée de façon répétitive pour optimiser la forme de la géométrie, ou tout autre paramètre. Dans cette étude, les écoulements considérés ne font pas appel à des modélisations complexes (turbulence, combustion, etc). En revanche, la complexité de la géométrie du domaine de calcul est une composante importante de l'étude qui se traduit par la mise au point de méthodes de résolution adaptées à des discrétisation de type éléments finis en maillages triangulaires non-structurés.

Les deux caractéristiques principales que l'on peut attendre d'une méthode de résolution sont la rapidité de calcul de la solution et la minimisation de la capacité mémoire nécessaire à sa mise en œuvre. Une première réponse passe par l'utilisation de plateformes de calcul parallèles. Celles-ci permettent non seulement un gain en temps d'exécution car des opérations peuvent être distribuées sur plusieurs unités de calcul travaillant en parallèle, mais aussi une augmentation considérable de la mémoire disponible par aggrégation des mémoires locales associées à chaque unité de calcul. D'un autre point de vue, on peut chercher à mettre au point une méthode de résolution itérative dont la convergence est peu sensible à l'augmentation du nombre d'inconnues du problème. Les méthodes de décomposition de domaine sont très étudiées depuis le début des années 80. La raison principale de l'engouement pour ces méthodes

réside justement dans la possibilité de construire des algorithmes parfaitement adaptés aux plateformes de calcul parallèles. De plus, lorsqu'elles sont développées pour la résolution de problèmes elliptiques du second ordre, on est souvent capable de construire des méthodes quasi-optimales, c'est-à-dire de convergence faiblement dépendante du nombre de degrés de liberté (ou encore, de la taille du maillage sous-jacent), et « scalables », c'est-à-dire de convergence indépendante du nombre de sous-domaines, comme en témoignent les nombreuses méthodes proposées dans la littérature (on consultera par exemple l'ouvrage de référence dû à Smith, Bjorstad et Gropp[SBG96]).

La situation est cependant moins claire pour les systèmes d'EDP représentatifs de la mécanique des fluides compressibles. Les opérateurs sont à dominante du premier ordre, non symétriques, à solutions comportant souvent des singularités. Ainsi, la construction d'algorithmes par décomposition de domaine adaptés à ces modèles d'EDP a été jusqu'ici peu abordée en comparaison des travaux réalisés en mécanique des structures par exemple. La présente étude vise à combler quelque peu ce déficit en adoptant une démarche double :

- d'une part, on a proposé des contributions visant à améliorer des méthodologies existantes. Dans ce cas, les stratégies de résolution développées ont été appliquées au calcul d'écoulements stationnaires et instationnaires représentatifs avec un souci de comparaison (en termes d'efficacité parallèle comme de coût de calcul total) aux méthodes plus largement utilisées pour la résolution des applications qui nous intéressent ici;
- d'autre part, on a étudié des solutions nouvelles, considérées comme plus prometteuses, nous limitant dans ce cas à des résultats préiminaires obtenus dans un cadre applicatif simplifié.

Pour ce qui concerne le premier aspect de l'étude, nous avons notamment évalué en détails des méthodes par décomposition de domaine basées sur des stratégies de résolution approchée des problèmes locaux. En effet, les méthodes de résolution couramment adoptées pour les systèmes linéaires issus de la discrétisation des systèmes d'équations d'Euler ou de Navier-Stokes sont très souvent de nature itératives (méthodes de relaxation de Jacobi ou de Gauss-Seidel, méthode de Krylov préconditionnées, méthodes multigrilles); les systèmes linéaires concernés sont rarement résolus de façon précise et, dans ce contexte, il est clair que l'utilisation de méthodes directes (factorisation LU) pour la résolution des problèmes locaux conduit à des méthodes de décomposition de domaine globalement trop coûteuses en temps de calcul et donc, peu ou pas compétitives vis-à-vis des méthodes existantes. On a démontré ici, à travers un grand nombre d'expériences numériques, que l'on peut construire des stratégies par sous-domaine efficaces, tant d'un point de vue parallèle que numérique, basées sur l'application de quelques cycles d'une méthode multigrille pour la résolution des problèmes locaux.

Ainsi, les chapitres 2 (partie I), 5 et 6 (partie II) ont été respectivement consacrés à la résolution numérique des systèmes d'équations d'Euler et de Navier-Stokes en régine laminaire. Dans chaque cas, la discrétisation en espace repose sur une formulation mixte éléments finis/volumes finis en maillages triangulaires non-structurés et l'utilisation de schémas décentrés pour le calcul des termes de flux convectif. L'intégration en temps des équations semi-discrétisées fait appel à un schéma implicite linéarisé. Dans ce contexte, on a développé et évalué des stratégies de résolution par sous-domaines en considérant les étapes suivantes (communes à la résolution des systèmes d'équations d'Euler et de Navier-Stokes) :

- dans un premier temps, on a considéré les systèmes d'équations d'Euler ou de Navier-Stokes en continu (plus précisément, intégrées en temps par un schéma implicite, et linéarisées) et on a formulé un algorithme de type Schwarz additif sur une décomposition sans recouvrement du domaine de calcul. Les conditions imposées aux interfaces entre sous-domaines voisins sont issues de la

formulation faible du problème aux valeurs initiales et aux limites associé (conditions d'interface « naturelles » ou « classiques »). Elles expriment la continuité des composantes du flux normal à l'interface;

- la seconde étape a consisté à adapter l'algorithme de type Schwarz au contexte discret de notre étude. Pour ce qui concerne les termes de flux convectif (partie « hyperbolique » des équations), les conditions de raccord sont données par les composantes de flux numériques décentrés calculés entre deux cellules de contrôle associées aux nœuds situés de part et d'autre de l'interface entre sous-domaines voisins. Les problèmes locaux sont bien posés si on impose seulement les composantes liées à l'information entrante dans un sous-domaine donné. Dans le cas des termes de flux diffusifs (partie « elliptique » des équations), on a imposé la continuité de toutes les composantes au lieu des composantes normales seulement pour simplifier la mise en œuvre de la méthode dans le cadre de la formulation mixte éléments finis/volumes finis à la base de notre étude;
- d'un point de vue algébrique, l'algorithme de type Schwarz peut être interprété comme une méthode de relaxation de Jacobi appliquée à la résolution d'un système linéaire dont la matrice a une structure par blocs particulière. Il s'agit en fait d'un système discret similaire à celui caractérisant la stratégie de résolution globale mais dont l'expression a été étendue à la prise en compte d'inconnues spécifiques à l'interface. Alors que les inconnues associées à chaque nœud interne à un sous-domaine sont les composantes du vecteur de variables conservatives, les inconnues définies aux nœuds interfaces sont les composantes d'un flux numérique. On peut alors appliquer une technique de sous-structuration à la matrice de ce système pour aboutir à la formulation d'un problème interface comme dans les méthodes de décomposition de domaine de type complément de Schur. Le système interface résultant est résolu au moyen d'une méthode GMRES sans réinitialisation;
- une composante essentielle de notre étude concerne l'utilisation d'une méthode multigrille par agglomération de volumes pour une résolution approchée des systèmes locaux qui caractérisent la méthode de décomposition de domaine de type complément de Schur. Cette méthode multigrille est particulièrement bien adaptée au contexte discret considéré ici. En particulier, les discrétisations grossières sont construites automatiquement, suivant le principe d'agglomération de volumes, à partir de la seule donnée du maillage dual du maillage triangulaire fin de départ.

Un nombre important d'expériences numériques ont permis de montrer que l'option de résolution approchée des systèmes locaux ne dégrade pas la convergence linéaire (résolution de systèmes linéaires pour un pas de temps donné) et non-linéaire (convergence vers l'état stationnaire). De même, dans le cas de calculs d'écoulements instationnaires, on a observé que la précision de la solution physique n'est pas affectée par la résolution approchée des systèmes locaux.

L'aspect le plus « prospectif » de cette étude est développé dans le chapitre 4 pour la résolution du système d'équations d'Euler en deux dimensions d'espace. En effet, dans ce chapitre on s'est intéressé plus particulièrement à la forme des conditions de raccord imposées aux interfaces entre sous-domaines voisins dans le cadre d'un algorithme de type Schwarz additif sans recouvrement de la forme (1.20). Partant du résultat de convergence obtenu dans la section 1.2 du chapitre 1, lorsque les conditions en question sont celles résultant naturellement de la formulation faible du problème aux valeurs initiales et aux limites associé (c'est-à-dire les conditions d'interface « classiques »), on a cherché à construire des conditions de raccord plus générales en vue de l'optimisation du taux de convergence de l'algorithme. Il s'agit donc ici d'une alternative à la méthode d'accélération de convergence algébrique proposée dans le chapitre 3.

Dans un premier temps, des conditions de raccord généralisées ont été construites en utilisant la théorie de Smith pour la diagonalisation des matrices polynômiales[Gan65], en s'inspirant de travaux récents dûs à Japhet[Jap98] et Japhet *et al.*[JNR98] pour la résolution d'un problème de convection-diffusion, et aussi à Gonzalez[Gon00] pour la résolution du système d'équations d'Euler. Les conditions en question sont au plus du premier ordre en dérivée partielle dans la direction tangentielle à l'interface et dépendent de quelques paramètres qui, s'ils sont convenablement choisis, permettent une meilleure convergence de l'algorithme de type Schwarz additif. Elles se matérialisent par un opérateur matriciel (opérateur « interface ») qui remplace celui caractérisant la diagonalisation classique (par vecteurs propres) du système d'équations d'Euler lnéarisé et symétrisé. De ce fait, la mise en œuvre des conditions de raccord généralisées est a priori aisée.

On a ensuite particularisé l'étude en considérant des conditions de raccord d'ordre zéro (c'est-à-dire sans dérivée dans la direction tangentielle à l'interface) dont on a déduit des conditions d'interface « optimisées ». L'absence de dérivée tangentielle facilite un peu plus la mise en œuvre qui s'est faite ici dans le cadre de la méthodologie numérique de type volumes finis décrite au chapitre 2. Cependant, cette dernière est caractérisée par une forme de l'interface qui n'est pas rectiligne contrairement à l'hypothèse faite en théorie. D'une façon générale, les expériences numériques présentées dans ce chapitre ont permis de retrouver qualitativement un comportement en accord avec la théorie même si, d'un point de vue discret, on n'a pas reproduit identiquement les résultats obtenus en continu. Ces résultats préliminaires tout à fait encourageants laissent présager une bonne marge de manoeuvre quant à l'accélération de la convergence de l'algorithme de type Schwarz additif dans un contexte discret plus favorable.

Bibliographie

[AD84] Angrand (F.) et Dervieux (A.). – Some explicit triangular finite element schemes for the Euler equations. *Int. J. Numer. Meth. Fluids*, vol. 4, 1984, pp. 749–764.

[AD93] Arminjon (P.) et Dervieux (A.). – Construction of TVD-like artificial viscosities on two-dimensinal arbitrary FEM grids. *J. Comput. Phys.*, vol. 106, 1993, pp. 176–198.

[ALTNV00] Achdou (Y.), Le Tallec (P.), Nataf (F.) et Vidrascu (M.). – A domain decomposition preconditionner for an advection-diffusion problem. *Comp. Meth. in Appl. Mech. and Eng.*, vol. 184, 2000, pp. 145–170.

[Bar99] Barth (T.). – *Numerical methods for gasdynamics systems on unstructured meshes*, pp. 195–285. – Springer Verlag, 1999, *Lecture Notes in Computatational Science and Engineering*, volume 5.

[BF00a] Bouras (A.) et Fraysse (V.). – *A relaxing strategy for inexact matrix-vector products for Krylov methods*. – Rapport technique nTR/PA/00/15, CERFACS, 2000.

[BF00b] Bouras (A.) et Fraysse (V.). – *A relaxing strategy for inner-outer linear solvers in domain decomposition methods*. – Rapport technique nTR/PA/00/17, CERFACS, 2000.

[Bri87] Briggs (W.-L.). – *A multigrid tutotial*. – SIAM, 1987.

[Car95] Carré (G.). – *Simulation numérique d'écoulements turbulents compressibles stationnaires par méthodes multigrilles*. – Thèse en Sciences de l'Ingénieur, Université de Nice-Sophia Antipolis, novembre 1995.

[Cle97] Clerc (S.). – *Etude de schémas décentrés implicites pour le calcul numérique en mécanique des fluides. Résolution par décomposition de domaine*. – Thèse en Analyse Numérique, Université de Paris VI, 1997.

[CQ95] Carlenzoli (C.) et Quarteroni (A.). – Adaptive domain decomposition methods for advection-diffusion problems. *In : Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations*, éd. par Babuska (I.) et al. pp. 169–199. – Springer Verlag.

[CW92] Cai (X.-C.) et Widlund (O.B.). – Domain decomposition algorithms for indefinite elliptic problems. *SIAM J. Sci. Stat. Comput.*, vol. 13, 1992, pp. 243–259.

[Des91] Desprès (B.). – *Domain decomposition methods and the Helmholtz problem*, pp. 44–52. – SIAM, 1991, *Mathematical and Numerical Aspects of Wave Propagation Phenomena*.

[Dés98] Désidéri (J.-A.). – *Modèles discrets et schémas itératifs. Application aux algorithmes multigrilles et multidomaines*. – Hermes, 1998.

[dLTV92] d'Hennezel (F.), Le Tallec (P.) et Vidrascu (M.). – *A parallel algorithm for advection-diffusion problem using domain decomposition*. – Rapport technique n 33, STPA-ONERA, 1992.

[EM77] Engquist (B.) et Majda (A.). – Absorbing boundary conditions for the numerical simulation of waves. *Math. Comp.*, vol. 31, 1977, pp. 629–651.

[FD89] Fezoui (L.) et Dervieux (A.). – Finite element non-oscillatory schemes for compressible flows. *In : 8th France-U.S.S.R.-Italy Joint Symposium on Computational Mathematics and Applications*. – Pavie, Italie, 1989.

[Fez85] Fezoui (F.). – Résolution des équations d'euler par un schéma de van leer en éléments finis. – Rapport de recherche n RR-358, INRIA, 1985.

[FL94] Farhat (C.) et Lanteri (S.). – Simulation of compressible viscous flows on a variety of mpps : computational algorithms for unstructured dynamic meshes and performance results. *Comp. Meth. in Appl. Mech. and Eng.*, vol. 119, 1994, pp. 35–60.

[FLLO89] Fezoui (L.), Lanteri (S.), Larrouturou (B.) et Olivier (C.). – Résolution numérique des équations de Navier-Stokes pour un fluide compressible en maillage triangulaire. – Rapport de recherche n RR-1033, INRIA, 1989.

[FS89] Fezoui (L.) et Stoufflet (B.). – A class of implicit upwind schemes for Euler simulations with unstructured meshes. *J. of Comp. Phys.*, vol. 84, 1989, pp. 174–206.

[FSL95] Farhat (C.), Simon (H.) et Lanteri (S.). – TOP/DOMDEC : a software tool for mesh partitioning and parallel processing and applications to CSM and CFD computations. *Comput. Syst. Eng.*, vol. 6, n1, 1995, pp. 13–26.

[Gan65] Gantmacher (F.-R.). – *Théorie des matrices*. – Dunod, 1965.

[GG93] Gastaldi (F.) et Gastaldi (L.). – On a domain decomposition for the transport equation : theory and finite element approximation. *IMA J. Numer. Anal.*, vol. 14, 1993, pp. 111–135.

[GGQ96] Gastaldi (F.), Gastaldi (L.) et Quarteroni (A.). – Adaptive domain decomposition methods for advection-dominated equations. *East-West J. Numer. Math.*, vol. 4, 1996, pp. 165–206.

[GGQ98] Gastaldi (F.), Gastaldi (L.) et Quarteroni (A.). – Adn and arn domain decomposition methods for advection-diffusion equations. *In : 9th International Conference on Domain Decompositon Methods in Science and Engineering*, éd. par Bjorstad (P.-E.), Espedal (M.-S.) et Keyes (D.-E.). pp. 334–341. – John Wiley & Sons.

[Gon00] Gonzalez (T.). – *Contributions aux conditions d'interface et conditions aux limites pour le système d'équations Euler compressible*. – Thèse en Analyse Numérique, Université Paris VI, 2000.

[GR96] Godlewski (E.) et Raviart (P.-A.). – *Numerical approximation of hyperbolic systems of conservation laws*. – Springer Verlag, 1996, *Applied Mathematical Sciences*, volume 118.

[Gui93] Guillard (H.). – Node-nested multigrid with delaunay coarsening. – Rapport de recherche n RR-1898, INRIA, 1993.

[Hac85] Hackbusch (W.). – *Multigrid methods and applications*. – Springer Verlag, 1985, *Springer series in Computationl Mathematics*, volume 4.

[Hal86] Halpern (L.). – Artificial boundary conditions for the advection-diffusion equation. *Math. Comp.*, vol. 174, 1986, pp. 425–438.

[Hal88] Halpern (L.). – Conditions aux limites artificielles pour un système incomplètement parabolique. *C. R. Acad. Sci. Paris*, vol. 307, 1988, pp. 413–416.

[Hem90] Hemker (P.-W). – On the order of prolongations and restrictions in multigrid procedures. *J. Comput. Appl. Math.*, vol. 32, 1990, pp. 423–429.

[HS89] Halpern (L.) et Schatzman (M.). – Artificial boundary conditions for incompressible viscous flows. *SIAM J. Matrix Anal. Appl.*, vol. 20, 1989, pp. 308–353.

[Jap98] Japhet (C.). – *Méthode optimisée d'ordre 2. Application à l'équation d'advection-diffusion*. – Thèse en Analyse Numérique, Université Paris XIII, 1998.

[Jef76] Jeffrey (A.). – *Quasilinear hyperbolic systems and waves.* – Pitman Publishing Limited, 1976.

[JNR98] Japhet (C.), Nataf (F.) et Roux (F.X.). – The optimized order 2 method with a coarse grid preconditioner. application to convection-diffusion problems. *In : 9th International Conference on Domain Decompositon Methods in Science and Engineering*, éd. par Bjorstad (P.-E.), Espedal (M.-S.) et Keyes (D.-E.). pp. 279–286. – John Wiley & Sons.

[KLA94] Koobus (B.), Lallemand (M.-H.) et A. (Dervieux). – Unstructured volume agglomeration mg: solution of the Poisson equation. *Int. J. Numer. Meth. Fluids*, vol. 18, 1994, pp. 27–42.

[Koo94] Koobus (B.). – *Algorithmes multigrille et algorithmes implicites pour les écoulements compressibles turbulents.* – Thèse de Mathématiques, Université de Nice-Sophia Antipolis, 1994.

[Kro91] Kroner (D.). – Absorbing boundary conditions for the linearized Euler equations in 2d. *Math. Comp.*, vol. 57, 1991, pp. 153–167.

[Lal88] Lallemand (M.-H.). – *Schémas décentrés multigrilles pour la résolution des équations d'Euler en éléments finis.* – Thèse en Mathématiques, Université de Provence, 1988.

[Lan91] Lanteri (S.). – *Simulation d'écoulements aérodynamiques instationnaires sur une architecture S.I.M.D. massivement parallèle.* – Thèse en Sciences de l'Ingénieur, Université de Nice-Sophia Antipolis, décembre 1991.

[Lan96] Lanteri (S.). – Parallel solutions of compressible flows using overlapping and non-overlapping mesh partitioning strategies. *Parallel Computing*, vol. 22, 1996, pp. 943–968.

[Les75] Lesaint (P.). – *Sur la résolution des systèmes hyperboliques du premier ordre par des méthodes d'éléments finis.* – Thèse en Analyse Numérique, Université de Paris VI, 1975.

[Lio88] Lions (P.L.). – On the Schwarz alternationg method i. *In : 1st International Symposium on Domain Decompositon Methods for Partial Differential Equations*, éd. par Glowinski (R.), Golub (G.H.), Meurant (G.A.) et Périaux (J.). pp. 1–42. – SIAM.

[Lio89] Lions (P.L.). – On the Schwarz alternationg method iii: a variant for nonoverlapping subdomains. *In : 3rd International Symposium on Domain Decompositon Methods for Partial Differential Equations*, éd. par Chan (T.), Glowinski (R.), Périaux (J.) et Widlund (O.-B.). pp. 202–223. – SIAM.

[LSA92] Lallemand (M.-H.), S. (Steve) et A. (Dervieux). – Unstructured multigridding by volume agglomeration : current status. *Computers & Fluids*, vol. 21, 1992, pp. 397–433.

[Mar96] Martin (R.). – *Développement de méthodes de calcul efficaces pour des écoulements instationnaires en géométrie déformable.* – Thèse en Sciences de l'Ingénieur, Université de Nice-Sophia Antipolis, 1996.

[Mer94] Mer (K.). – Variational analysis of a mixte finite element/finite volume scheme on general triangulations. – Rapport de recherche n RR-2213, INRIA, 1994.

[MG96] Martin (R.) et Guillard (V.). – A second order defect correction scheme for unsteady problems. *Computers & Fluids*, vol. 25, 1996, pp. 9–27.

[MJ87] Mavriplis (D. J.) et Jameson (A.). – Multigrid solution of the two-dimensional Euler equations on unstructured meshes. *AIAA paper*, no 87-0353, 1987.

[MJ88a] Martinelli (L.) et Jameson (A.). – Validation of a multigrid method for the Reynolds averaged Navier-Stokes equations. *AIAA paper*, no 88-0414, 1988.

[MJ88b] Mavriplis (D. J.) et Jameson (A.). – Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes. *AIAA paper*, no 88-3707, 1988.

[Nat93] Nataf (F.). – *On the use of open boundary conditions in block Gauss-Seidel methods for the convection-diffusion equation.* – Rapport technique n RI284, Centre de Mathématiques Appliquées, Ecole Polytechnique, 1993.

[Nat01] Nataf (F.). – *Domain decomposition methods for non-symmetric problems*. – Saxe-Coburg Publications, 2001, *Parallel and Distributed Processing for Computational Mechanics III*. à paraître.

[NN97] Nataf (F.) et Nier (F.). – Convergence rate of some domain decomposition methods for overlapping and nonoverlapping subdomains. *Numer. Math.*, vol. 75, 1997, pp. 357–377.

[NRdS94] Nataf (F.), Rogier (F.) et de Sturler (E.). – *Optimal interface conditions for domain decomposition methods*. – Rapport technique n RI301, Centre de Mathématiques Appliquées, Ecole Polytechnique, 1994.

[QS96] Quarteroni (A.) et Stolcis (L.). – *Homogeneous and heterogeneous domain decomposition methods for compressible flow at high Reynolds numbers*. – Rapport technique n 33, CRS4, 1996.

[Qua90] Quarteroni (A.). – Domain decomposition methods for systems of conservation laws : spectral collocation approximation. *SIAM J. Sci. Stat. Comput.*, vol. 11, 1990, pp. 1029–1052.

[QV91] Quarteroni (A.) et Valli (A.). – *Theory and application of Steklov-Poincaré operators for boundary value problems*, pp. 179–203. – Kluwer Academic Publishers, 1991, *Applied and Industrial Mathematics*.

[QV99] Quarteroni (A.) et Valli (A.). – *Domain decomposition methods for partial differential equations*. – Oxford University Press, 1999, *Numerical Mathematics and Scientific Computation*.

[Roe81] Roe (P.L.). – Approximate Riemann solvers, parameter vectors and difference schemes. *J. of Comp. Phys.*, vol. 43, 1981, pp. 357–371.

[RS88] Rostand (P.) et Stoufflet (B.). – Finite volume Galerkin methods for viscous gas dynamics. – Rapport de recherche n RR-863, INRIA, 1988.

[SBG96] Smith (B.), Bjorstad (P.) et Gropp (W.). – *Domain decomposition and parallel multilevel methods for elliptic partial differential equations*. – Cambridge University Press, 1996.

[Smi92] Smith (B.). – An optimal domain decomposition preconditionner for the finite element solution of linear elasticity. *SIAM J. Sci. Stat. Comput.*, vol. 13, 1992, pp. 364–379.

[SS86] Saad (Y.) et Schultz (H.). – Gmres : Generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM J. Sci. Stat. Comput.*, vol. 7, 1986, pp. 856–869.

[Ste88] Steve (H.). – *Schémas implicites linéarisés décentrés pour la résolution des équations d'Euler en plusieurs dimensions*. – Thèse Mathématiques, Université de Provence, juillet 1988.

[SW81] Steger (J.) et Warming (R. F.). – Flux vector splitting for the inviscid gas dynamic with applications to finite difference methods. *J. of Comp. Phys.*, vol. 40, 1981, pp. 263–293.

[Tor97] Toro (E.). – *Riemann solvers and numerical methods for fluid dynamics*. – Springer, 1997.

[Vio98] Viozat (C.). – *Calcul d'écoulements stationnaires et instationnaires à petit nombre de Mach et en maillages étirés*. – Thèse en Sciences de l'Ingénieur, Université de Nice-Sophia Antipolis, octobre 1998.

[VL79] Van Leer (B.). – Towards the ultimate conservative difference scheme V : a second-order sequel to Godunov's method. *J. of Comp. Phys.*, vol. 32, 1979, pp. 361–370.

[Wes91] Wesseling (P.). – *An introduction to multigrid methods*. – John Wiley & Sons, 1991.

[XC92] Xu (J.) et Cai (X.-C.). – A preconditionned gmres method for nonsymmetric or indefinite problems. *Math. Comp.*, vol. 59, 1992, pp. 311–319.

[XZ98] Xu (J.) et Zou (J.). – Non-overlapping domain decomposition methods. *SIAM Review*, vol. 40, n4, 1998, pp. 857–914.

Algorithmes par décomposition de domaine et accélération multigrille pour le calcul d'écoulements compressibles

Cette thèse porte sur l'étude d'algorithmes par décomposition de domaine sans recouvrement pour la résolution de systèmes d'EDP en mécanique des fluides compressibles. La première partie traite de la résolution des équations d'Euler. On considère le système linéarisé résultant d'une méthode d'Euler implicite pour l'intégration en temps, que l'on résout en formulant un algorithme de type Schwarz additif basé sur des conditions de raccord naturelles. La convergence de cet algorithme est étudiée au moyen d'une analyse de Fourier. Les expériences numériques réalisées dans un cadre discret d'approximation mixte éléments/volumes finis en maillages triangulaires confirment la théorie. En discret, cet algorithme s'interprète comme une méthode de Jacobi appliquée à un système linéaire dont la matrice a une structure par blocs. Une technique de sous-structuration peut être appliquée à cette matrice pour obtenir un système interface (algorithme de type complément de Schur) résolu par une méthode GMRES. Dans cette étude, on utilise une méthode multigrille par agglomération de volumes pour la résolution itérative des systèmes locaux. Deux techniques d'accéleration de la convergence sont étudiées: un préconditionnement algébrique du système interface et la construction de conditions d'interface optimisées pour l'algorithme de Schwarz. Ces dernières sont dérivées en utilisant la factorisation de Smith des matrices polynômiales issues d'une transformation de Laplace partielle du système d'équations d'Euler, et en optimisant le taux de convergence de l'algorithme de Schwarz. Des tests numériques confirment la viabilité de ces conditions. La deuxième partie traite de l'extension de l'algorithme de type complément de Schur basée sur des conditions de raccord naturelles à la résolution des équations de Navier-Stokes pour le calcul d'écoulements laminaires stationnaires et instationnaires.

Mots clefs : Equations d'Euler - Equations de Navier-Stokes - Eléments finis - Volumes finis - Décomposition de domaine - Algorithmes de Schwarz - Multigrille - Calcul parallèle.

Domain decomposition and multigrid algorithms for the calculation of compressible flows

This thesis deals with non-overlapping domain decomposition algorithms for the resolution of systems of EDPs modelling compressible fluid flows. The first part of the thesis is concerned with the resolution of the Euler equations. We consider the linearized system resulting from a backward Euler method for time integration. In order to solve this system, an additive Schwarz type algorithm based on natural interface conditions is formulated. The convergence of this algorithm is analysed by means of a Fourier transform technique. Numerical experiments performed in the framework of a mixed finite element/finite volume method on triangular meshes confirm the results of the convergence analysis. Algebraically speaking, the Schwarz algorithm is equivalent to a Jacobi iteration applied to a linear system whose matrix has a block structure. A substructuring technique can be applied to this matrix in order to obtain an interface system (Schur complement type algorithm) which is solved using a GMRES method. In this study, the local problems are solved iteratively by using a multigrid by agglomeration method. Two convergence acceleration techniques are studied: an algebraic preconditioning technique for the interface system and the construction of optimized interface conditions for the Schwarz algorithm. The latter are derived using the Smith factorization of a polynomial matrix resulting from a partial Laplace transform of the Euler equations, and by optimizing the convergence rate of the Schwarz algorithm. Numerical tests confirm their superiority with respect to classical interface conditions. The second part of the thesis is concerned with the extension of the Schur type algorithm based on classical interface conditions to the solution of the Navier-Stokes equations in view of the calculation of compressible laminar steady and unsteady flows.

Key words : Euler equations - Navier-Stokes equations - Finite elements - Finite volumes - Domain decomposition methods - Schwarz algorithms - Multigrid methods - Parallel computing.