

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

THÈSE PRÉSENTÉE À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DU
DOCTORAT EN GÉNIE
Ph.D.

PAR
BEN HAJ ALI, Amine

CALCUL DE HAUTE PERFORMANCE EN AÉROÉLASTICITÉ ET EN
ÉCOULEMENTS TURBULENTS TRIDIMENSIONNELS

MONTREAL, LE 23 OCTOBRE 2008

CETTE THÈSE A ÉTÉ ÉVALUÉE

PAR UN JURY COMPOSÉ DE :

M. Azzeddine Soulaïmani, directeur de thèse
Département de génie mécanique, École de technologie supérieure

M. Saad Bennis, président du jury
Département de génie de la construction, École de technologie supérieure

M. Christian Masson
Département de génie mécanique, École de technologie supérieure

M. Marius Paraschivoiu
Department of mechanical and industrial engineering, Concordia University

M. Emmanuel Lorin, examinateur invité
Faculty of Science, University of Ontario I.T.

ELLE A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 22 AOÛT 2008

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Je tiens à exprimer mes remerciements les plus vifs

- À mon directeur de recherche professeur Azzeddine Soulaïmani Ph.D., Ing. Professeur Soulaïmani a encadré ce travail ce qui m'a permis de profiter de son expérience, de sa réputation dans le domaine, de ses directives et de son soutien aussi bien scientifique que moral. Pour cela et pour bien d'autres choses je tiens à lui exprimer ma profonde gratitude.
- Au professeur Saad Bennis Ph.D., Département de génie de la construction de École de technologie supérieure
- Au professeur Christian Masson Ph.D., Ing. Département de génie mécanique, École de technologie supérieure
- Au professeur Marius Paraschivoiu Ph.D. Department of mechanical and industrial engineering, Concordia University
- Au professeur Emmanuel Lorin Ph.D. Faculty of Science, University of Ontario I.T.

Je tiens aussi à exprimer mes remerciements les plus sincères à tous les membres du groupe de recherche Granit.

Je dédie ce travail à mon père Hamdane, ma mère Alia, mes frères Samir, Mourad et Mounir, ma chère épouse Farah et à toute personne qui m'est chère.

CALCUL DE HAUTE PERFORMANCE EN AÉROÉLASTICITÉ ET EN ÉCOULEMENTS TURBULENTS TRIDIMENSIONNELS

BEN HAJ ALI, Amine

RÉSUMÉ

Dans le cadre de cette thèse, nous étudions de nouvelles méthodes numériques de résolution des problèmes d'aéroélasticité et des écoulements turbulents tridimensionnels. Un code de calcul parallèle PFES360, basé sur une approche de décomposition fonctionnelle, est développé dans le but de résoudre des problèmes multiphysiques de grandes tailles.

Le recours aux maillages purement tétraédriques pour la discrétisation des équations gouvernantes dans le cas d'un écoulement turbulent requiert l'utilisation des éléments anisotropiques extrêmement aplatis. La forte distorsion de ces éléments affecte dramatiquement le conditionnement du système. Dans ces conditions, les méthodes standards deviennent incapables de stabiliser la solution numérique. Deux nouvelles définitions de la matrice τ de la méthode SUPG (Steamline Upwinding Petrov-Galerkin) et de l'opérateur de capture de chocs sont alors introduites.

Les nouvelles définitions et les méthodes développées ont été implémentées dans PFES360. Des cas tests documentés dans la littérature ont été menés afin de mettre en évidence la performance de ces méthodes. Des comparaisons des résultats avec les valeurs théoriques (plaque plane) et expérimentales (Agard 445.6 et Onera M6) sont présentées.

Le présent travail nous a permis de constater que la réussite de ce genre de simulations dépend de la pertinence méthode de stabilisation, de la précision du modèle de turbulence et essentiellement de la qualité du maillage. En plus, il est important d'assurer la positivité de la viscosité turbulente pour éviter les problèmes de convergence.

HIGH FIDELITY COMPUTATIONAL AEROELASTICITY AND THREE-DIMENSIONAL TURBULENT FLOW

BEN HAJ ALI, Amine

ABSTRACT

The main purpose of this thesis is the developement of new numerical procedures for simulating three-dimensional turbulent flows and aeroelasticity problems. In order to solve such large size multi-physics problems a parallel code called PFES360 based on a functional decomposition is developed.

In the case of a turbulent flow the discretization of the governing equations on anisotropic tetrahedral grids requires the use of extremely deformed elements. The distortion of these elements dramatically affects the system conditioning. Therefore, standard methods become unable to stabilize the numerical solution. New definitions of the matrix τ and the shock capturing operator are then introduced.

These methods have been implemented in PFES360. The performance of the algorithms is demonstrated through numerical studies. Comparisons with analytical (flat plate) and experimental (Agard 445.6 and Onera M6) results are presented.

It has been demonstrated that the success of such simulations depends on the turbulence model, the stabilization method and essentially on the mesh quality. Also, it is important to ensure the positivity of the turbulent viscosity in order to avoid convergence problems.

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 PROBLÉMATIQUE	7
1.1 Problématiques associées à l'aspect multiphysique	7
1.2 Problématiques associées à la modélisation de l'écoulement tridimensionnel	8
1.3 Problématiques associées à la modélisation de la turbulence	9
1.4 Problématiques associées au maillage	10
CHAPITRE 2 REVUE DE LITTÉRATURE	12
2.1 Modélisation du couplage fluide–structure : l'aéroélasticité	12
2.2 Modélisation des écoulements turbulents	15
2.2.1 Le modèle de Spalart-Allmaras	18
CHAPITRE 3 MÉTHODOLOGIE ET ALGORITHMIQUE	22
3.1 Résolution séquentielle des problèmes multiphysiques	22
3.2 Résolution parallèle des problèmes multiphysiques	24
3.3 Méthode de décomposition spatio-fonctionnelle	27
3.3.1 Méthode de décomposition de Schwarz	29
3.3.1.1 Méthode de Schwarz multiplicative	31
3.3.1.2 Méthode de Schwarz additive	32
3.4 Algorithme par sous-domaine	32
3.4.1 Module de fluide	34
3.4.2 Module de turbulence	35
3.4.3 Module de structure	36
3.4.4 Module de mouvement de maillage	37
3.5 Résolution des grands systèmes d'équations	38
3.6 Implémentation informatique	39
CHAPITRE 4 RÉOLUTION DES ÉQUATIONS D'EULER ET DE NAVIER-STOKES MOYENNÉES	47
4.1 Équations de Navier-Stokes	48
4.1.1 Forme conservative	48
4.1.2 Forme non conservative	49
4.1.3 Forme conservative écrite en variables conservatives	49

4.1.4	Formulation eulerienne-lagrangienne des équations gouvernantes	50
4.1.5	Forme compacte des équations de Navier-Stokes	50
4.2	Discrétisation spatiale et temporelle	54
4.2.1	Conditions aux limites à l'infini	55
4.2.2	Discrétisation temporelle.....	56
4.2.3	Discrétisation spatiale.....	57
	4.2.3.1 Méthode de stabilisation SUPG standard.....	58
	4.2.3.2 Méthode de capture de choc standard	60
	4.2.3.3 Nouvelles définitions de la matrice τ et l'opérateur de capture de choc	61
4.3	Forme variationnelle stabilisée des équations de Navier-Stokes	64
CHAPITRE 5 RÉSOLUTION DE L'ÉQUATION DE TURBULENCE		66
5.1	Modèle standard de Spalart-Allmaras	67
5.2	Principe de la méthode numérique	68
5.2.1	Forme adimensionnelle de l'équation de Spalart et Allmaras.....	68
5.3	Approche volumes finis - éléments finis	69
5.3.1	Un premier algorithme (Algorithme 6)	69
5.3.2	Un deuxième algorithme (Algorithme 7).....	73
5.4	Approche purement éléments finis	79
5.4.1	Un premier algorithme (Algorithme 8)	79
5.4.2	Un deuxième algorithme (Algorithme 9).....	82
CHAPITRE 6 RÉSULTATS		85
6.1	Validation du couplage Euler-structure : Application à l'étude de la stabilité aéroélastique de l'aile Agard 445.6.....	85
6.2	Validation du module de turbulence isolé : Plaque plane avec une vitesse imposée	87
6.3	Validation du couplage Navier-Stokes-turbulence (RANS) : Plaque plane	92
6.4	Validation du couplage Navier-Stokes-turbulence (RANS) : Onera M6	95
6.4.1	Stratégie de résolution	96
6.5	Étude des performances.....	113
CONCLUSION		117
ANNEXE 1 : PARALLÉLISME		121
BIBLIOGRAPHIE		133

LISTE DES TABLEAUX

	Page
Tableau 2.1 Méthodes de prédiction	12
Tableau 6.1 Maillages de la plaque plane	89
Tableau 6.2 Erreur quadratique relative	90

LISTE DES FIGURES

	Page
Figure 3.1 <i>Approche séquentielle pour la résolution du problème d'aéroélasticité.</i>	23
Figure 3.2 <i>Décomposition géométrique du domaine.</i>	25
Figure 3.3 <i>Le paradigme SPMD : Single Program Multiple Data.</i>	26
Figure 3.4 <i>Approche SPMD en aéroélasticité.</i>	27
Figure 3.5 <i>Le paradigme MPMD : Multiple Program Multiple Data.</i>	28
Figure 3.6 <i>L'approche MPMD-SPMD.</i>	29
Figure 3.7 <i>Domaines et frontières dans la méthode de Schwarz.</i>	30
Figure 3.8 <i>Méthode de Schwarz multiplicative.</i>	31
Figure 3.9 <i>Décomposition fonctionnelle du problème d'aéroélasticité.</i>	33
Figure 3.10 <i>Organigramme du module de fluide.</i>	35
Figure 3.11 <i>Turbulence : organigramme de la première approche.</i>	36
Figure 3.12 <i>Turbulence : organigramme de la deuxième approche.</i>	37
Figure 3.13 <i>Communication entre les domaines voisins.</i>	43
Figure 3.14 <i>Organigramme général du logiciel PFES360.</i>	45
Figure 4.1 <i>Détection des zones d'oscillations et d'instabilités.</i>	64
Figure 4.2 <i>Senseur de variation de pression dans le cas d'une solution convergée.</i>	64
Figure 5.1 <i>Turbulence : organigramme de l'approche VF-EF, première variante.</i>	72
Figure 5.2 <i>Turbulence : organigramme de l'approche VF-EF, deuxième variante.</i>	78
Figure 5.3 <i>Turbulence : organigramme de l'approche purement EF, première variante.</i> ..	81
Figure 5.4 <i>Turbulence : organigramme de l'approche purement EF, deuxième variante.</i>	84

Figure 6.1	<i>Évolution des coordonnées généralisées dans le temps $q = 62\text{lb}/\text{ft}^2$.</i>	86
Figure 6.2	<i>Évolution des coordonnées généralisées dans le temps $q = 63\text{lb}/\text{ft}^2$.</i>	86
Figure 6.3	<i>Indice de flottement Agard 445.6.</i>	87
Figure 6.4	<i>Profil de vitesse imposée à l'entrée de la plaque plane.</i>	88
Figure 6.5	<i>Plaque plane avec vitesse imposée : Conditions aux limites.</i>	89
Figure 6.6	<i>Plaque plane avec une vitesse imposée : χ à différents plans $x = \text{cte}$.</i>	90
Figure 6.7	<i>Plaque plane avec une vitesse imposée : effet de maillage.</i>	91
Figure 6.8	<i>Plaque plane avec une vitesse imposée : erreur en fonction de pas de temps.</i>	92
Figure 6.9	<i>Plaque plane avec vitesse imposée : Convergence du schéma amélioré.</i>	92
Figure 6.10	<i>Plaque plane avec vitesse imposée : Conditions aux limites.</i>	93
Figure 6.11	<i>Profil de vitesse dans le cas d'une plaque plane, couplage fluide-structure.</i>	94
Figure 6.12	<i>Profil de χ dans le cas d'une plaque plane, couplage fluide-structure.</i>	94
Figure 6.13	<i>Erreur relative dans le cas d'une plaque plane, couplage fluide-structure.</i>	95
Figure 6.14	<i>Onera M6 : Conditions aux limites.</i>	96
Figure 6.15	<i>Modèle 3D de l'aile Onera M6.</i>	97
Figure 6.16	<i>Forme du maillage de l'Onera M6.</i>	98
Figure 6.17	<i>Maillage grossier de l'Onera M6.</i>	98
Figure 6.18	<i>Maillage fin de l'Onera M6.</i>	99
Figure 6.19	<i>Onera M6 RANS : Maillage sur le bord de fuite.</i>	99
Figure 6.20	<i>Onera M6 RANS : Viscosité χ au niveau de la section $y/b = 0.2$.</i>	100
Figure 6.21	<i>Onera M6 RANS : Pression et Mach au niveau de la section $y/b = 0.2$.</i>	100
Figure 6.22	<i>Onera M6 RANS : Pression sur l'aile.</i>	101
Figure 6.23	<i>Onera M6 : $-C_p$ à $y/b = 0.20$.</i>	102

Figure 6.24 Onera M6 : $-C_p$ à $y/b = 0.20$.	103
Figure 6.25 Onera M6 : $-C_p$ à $y/b = 0.44$.	103
Figure 6.26 Onera M6 : $-C_p$ à $y/b = 0.65$.	104
Figure 6.27 Onera M6 : $-C_p$ à $y/b = 0.80$.	104
Figure 6.28 Onera M6 : $-C_p$ à $y/b = 0.90$.	105
Figure 6.29 Onera M6 : $-C_p$ à $y/b = 0.95$.	105
Figure 6.30 Onera M6 : $-C_p$ à $y/b = 0.99$.	106
Figure 6.31 Onera M6 : Importance de la méthode SUPG.	107
Figure 6.32 Onera M6 : régime laminaire, $R_e = 10^5$.	108
Figure 6.33 Onera M6 : régime laminaire, $R_e = 10^6$.	109
Figure 6.34 Onera M6 : régime laminaire, $R_e = 10^6$, $\varepsilon_k = 0.05$.	110
Figure 6.35 Onera M6 : régime laminaire avec ζ^e pondéré.	111
Figure 6.36 Onera M6 : régime turbulent avec ζ^e pondéré.	112
Figure 6.37 Onera M6 : Courbes de convergence.	113
Figure 6.38 La machine "beowulf" Granit.	114
Figure 6.39 Speedup de PFES dans le cas de couplage fluide-structure.	114
Figure 6.40 Speedup de PFES360.	115
Figure 6.41 Onera M6 : Temps de calcul réel.	116

LISTE DES ABRÉVIATIONS ET SIGLES

RANS	Équations de Navier-Stokes moyennées (Reynolds-Averaged Navier-Stokes)
LCO	Oscillation(s) de cycle limite (Limit Cycle Oscillation)
Ω	Domaine de calcul
Γ	Frontière du domaine
\mathbf{n}	Vecteur normal unitaire orienté vers l'extérieur
\mathbf{I}	Matrice identité
\mathbf{u}	Vecteur vitesse de composantes u_i
u_i	Composante de la vitesse selon x_i
$\ \mathbf{u}\ $	Norme de la vitesse $\ \mathbf{u}\ = (u_1^2 + u_2^2 + u_3^2)^{\frac{1}{2}}$
\mathbf{f}_v	Vecteur forces volumiques
c	Vitesse du son
ρ	Densité
p	Pression
T	Température
t	Temps
δ	Symbole de Kronecker
λ	Valeurs propres
C_p	Chaleur spécifique à pression constante
C_v	Chaleur spécifique à volume constant
σ	Tenseur des contraintes visqueuses de composantes σ_{ij}
\mathbf{D}	Tenseur de déformations
q	Source de chaleur
r	Sources énergétiques

i	Énergie interne
k	Conductivité thermique
e_t	Énergie totale
E	Énergie totale par unité de volume
\mathbf{R}	Vecteur résidu
Re	Nombre de Reynolds
\mathbf{F}^{adv}	Flux de convection
\mathbf{F}^{diff}	Flux de diffusion
\mathbf{U}	Vecteur des variables conservatives
\mathbf{w}	Vitesse de mouvement de maillage
h	Dimension caractéristique d'un élément
τ	Matrice de stabilisation (dimension de temps)
μ_c	Viscosité artificielle de capture de choc
ς	Senseur de variation de pression
J	Jacobien de la transformation géométrique
\mathbf{W}	Fonctions de pondération
N_k	Fonctions d'interpolation de l'élément de référence
$[\mathbf{K}]$	Matrice de rigidité globale
$[\mathbf{M}]$	Matrice masse globale
ν	Viscosité dynamique
ν_t	Viscosité turbulente
$\tilde{\nu}$	Variable intermédiaire
$\chi \equiv \tilde{\nu}/\nu$	Variable intermédiaire pour le modèle de turbulence
\tilde{S}_e	Terme source calculé dans un élément

d	Distance à la paroi solide
$\tilde{\omega}$	Module de la vorticit� du fluide
$c_{b1}, c_{b2}, \sigma, c_{w2}, c_{w3}, c_{v1}, c_{v2}, k$	Des constantes empiriques
$f_{v1}, f_{v2}, \tilde{f}_{v2}, \tilde{f}_{v3}, f_w, g, r$	Fonctions interm�diaires

Indices :

r	Valeur de r�f�rence
i	Valeur unitaire selon la direction des x
j	Valeur unitaire selon la direction des y
k	Valeur unitaire selon la direction des z
∞	� l'infini
$+$	� la paroi
$,$	D�riv�e par rapport � une variable

Indices sup rieure :

i	Niveau d'it�ration ou nodale
e	�l�mentaire
\star	Variables adimensionnelles

Notation utilis es :

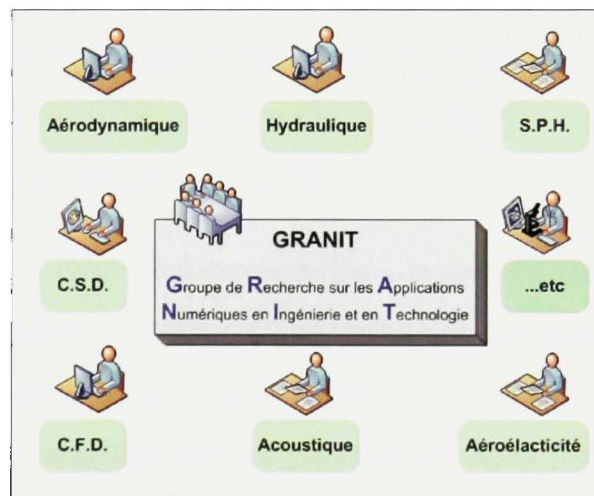
$\{N\}$	Vecteur colonne
$\langle N \rangle$	Vecteur ligne
$[K]$	Matrice de composantes K_{ij}
\otimes	Produit tensoriel $((u \otimes v)_{ij} = u_j v_j)$

Note : Certains symboles sont utilis s de fa on particuli re dans un contexte pr cis ; les d finitions explicites sont alors clairement  nonc es dans le texte.

INTRODUCTION

Présentation de la thèse

Cette thèse a été élaborée au sein du groupe de recherche sur les applications numériques en ingénierie et en technologie (GRANIT). Elle s'insère dans le cadre d'un grand projet de recherche portant sur l'aéroélasticité des ailes d'avions. Le groupe Granit travaille continuellement sur la recherche et le développement des méthodes numériques pour la modélisation et la simulation des problèmes multiphysiques. Ainsi, la présente contribution fait suite aux travaux de recherche de Elkadri (El Kadri El Yamani, 1992; El Kadri El Yamani, 1995), Rebaine (Rebaine, 1997), Azami (Azami, 2001), Bouzouiki (Bouzouiki, 2002), Ben Haj Ali (Ben Haj Ali, 2002), Feng (Feng, 2005) et Roth (Roth, 2007).



Activités de recherche dans le groupe GRANIT.

Dans l'introduction, nous situons la thèse par rapport à l'ensemble des travaux et des écrits dans le domaine. Les objectifs généraux et spécifiques du travail sont alors définis. Le chapitre 1 porte essentiellement sur les sources de motivation qui sont à l'origine de ce travail de recherche. Le chapitre 2 présente une brève et sûrement non exhaustive revue de littérature du domaine de simulation de l'aéroélasticité et des écoulements turbulents tridimensionnels.

Dans le chapitre 3, nous présentons notre méthodologie de résolution du problème global ainsi que les aspects numériques d'implémentation et de résolution. Les équations gouvernantes de chaque domaine physique sont ensuite détaillées dans les chapitre 4 et 5. Finalement, nous illustrons les résultats obtenus et émettons des remarques et des recommandations en guise de conclusion.

L'aéroélasticité

Le domaine de l'aéroélasticité est la branche de la mécanique qui traite des phénomènes résultant de l'interaction entre les forces aérodynamiques d'une part, et les forces massiques et élastiques d'autre part. L'aéroélasticité ¹ peut être définie comme étant un phénomène d'interaction entre les forces aérodynamiques et les réactions élastiques d'une structure. C'est une propriété qu'ont certains corps de reprendre, au moins partiellement, leur forme et leur volume primitifs quand la contrainte aérodynamique qui s'exerçait sur eux cesse d'agir ². L'aéroélasticité peut être considérée aussi comme une branche de la mécanique qui étudie les réactions aérodynamiques résultantes de l'interaction entre les déformations d'une structure élastique et un écoulement fluide (Hodges et Pierce, 2002).

Les phénomènes aéroélastiques sont courants dans la nature ; le mouvement des arbres et des drapeaux sous l'action du vent sont des exemples connus. Ces phénomènes peuvent être à l'origine des déformations statiques et dynamiques et à des instabilités dangereuses. Ce sont, probablement, ces instabilités qui ont provoqué la rupture impressionnante du pont de Tacoma-Narrows le 7 novembre 1940. L'aéroélasticité a des conséquences pratiques dans différents domaines de la technologie. En particulier, les interactions aéroélastiques influencent fortement la dynamique structurale, la stabilité et surtout les performances en vol des avions (Hutin, 1998). Ces phénomènes d'interaction ont également une grande influence sur les turbomachines, les turboréacteurs, les convertisseurs d'énergie éolienne, et dans plusieurs autres domaines. À titre d'exemple, Kim et al (Kim et al., 2000) ont élaboré des techniques d'expé-

¹ Office de la langue française

² Journal officiel de la République française du 22 septembre 2000

rimentation pour la prédiction du phénomène de flottement sur un disque dur acoustiquement excité. Les auteurs ont conclu que le problème de flottement peut survenir à une vitesse de rotation proche de $30.000tr/min$. D'autres travaux ont été élaborés dans le domaine de la conception automobile (Rifai et al., 2000). Aussi, Sathe et al (Sathe et al., 2007) ont utilisé le couplage fluide–structure dans le design des parachutes. Cependant, l'aéronautique reste le domaine le plus concerné par ces études.

Le couplage entre l'écoulement fluide sur une aile aéroélastique et le déplacement de sa structure peut induire des instabilités dynamiques. Tel est le cas du flottement aéroélastique (flutter) qui est défini³ comme suit :

- Vibrations aéroélastiques de haute fréquence, entretenues et parfois divergentes, affectant une structure flexible.
- Vibrations d'une structure aérospatiale, provoquées par le couplage des forces aérodynamiques avec les forces élastiques, inertielles et dissipatives de la structure.
- Vibrations mécaniques et aérodynamiques couplées survenant dans certains cas de vol et affectant une voilure, un empennage ou une gouverne.

Le flottement est caractérisé par une vibration importante de la structure induite par un écoulement fortement non stationnaire. Ces vibrations peuvent engendrer une importante fatigue et même affecter le bon fonctionnement de l'appareil et sa sécurité. Il est alors indispensable de s'assurer de l'absence d'instabilité.

La simulation numérique de l'aéroélasticité turbulente est complexe. D'une part, les codes commerciaux sont chers et jusqu'à date incapables de cerner tous les aspects de ce problème dans le cas général. D'autre part, ce genre d'applications fait appel à de grandes ressources

³ Office québécois de la langue française

de calcul. La taille du système d'équations à résoudre engendre l'épuisement des ressources des calculateurs traditionnels. Pour résoudre ce type de problème, les algorithmes exigent une augmentation significative de mémoire et du temps de calcul (temps CPU). En effet, à l'heure actuelle, cette problématique présente un champ actif de recherche. La vitesse de traitement, l'efficacité des algorithmes, et le traitement parallèle sont essentiels (Robert et al., 1998).

Au cours des dernières années, les codes de calcul sont devenus des outils indispensables dans le processus de développement de plusieurs produits. Ces outils sont devenus capables de simuler l'écoulement turbulent sur des géométries de plus en plus complexes (Peterson et Zing, 2008). Une revue de littérature plus exhaustive du domaine sera abordée ultérieurement.

Notre stratégie de résolution parallèle des problèmes multiphysiques offre une approche efficace en termes de performance et de facilité d'implémentation (Lorin et al., 2007; Soulaïmani et al., 2004; Ben Haj Ali, 2002; Soulaïmani et al., 2002c; Soulaïmani et al., 2001a).

Objectifs de la thèse

L'objectif global de ce projet de recherche est de contribuer à l'étude des phénomènes multiphysiques et en particulier l'aéroélasticité non linéaire et la résolution des écoulements turbulents tridimensionnels. Un code parallèle écrit en langage FORTRAN 90, utilisant la bibliothèque MPI et basé essentiellement sur la méthode des éléments finis est élaboré pour servir comme un banc d'essai pour la recherche et le développement.

Les objectifs généraux de cette thèse sont les suivants :

- Mettre à jour le code de calcul parallèle PFES dans le but d'améliorer sa structure informatique, sa stabilité, ses performances de calcul et surtout sa polyvalence.

Pour atteindre ce but, des objectifs spécifiques ont été fixés. La nouvelle version du code devrait supporter divers problèmes physiques en interaction (Fluide, Structure,

Turbulence, Hydraulique, Acoustique, etc). Le code doit être capable de fonctionner sur différentes plateformes informatiques (type de machine, système d'exploitation..etc). Le code doit aussi tourner en séquentiel ou en parallèle. Une optimisation portant sur l'utilisation de la mémoire, le transfert des données entre les processeurs et le temps de calcul devrait améliorer les performances informatiques du code.

Ainsi, la nouvelle version s'appelle PFES360 pour refléter sa portabilité, sa versatilité et sa capacité de résoudre une multitude de problèmes multiphysiques.

- Développer un module d'analyse aéroélastique basé sur les équations d'Euler.

En particulier, un code fluide permettant la modélisation des écoulements compressibles 3D autour des ailes d'avions et le couplage avec un autre module structure pour la simulation du problème d'aéroélasticité.

- Contribution à l'étude et le développement de méthodes de volumes et éléments finis pour la modélisation des écoulements 3D turbulents.

Le modèle de Spalart-Allmaras est choisi pour sa simplicité et sa précision dans la couche limite. Les équations de Spalart-Allmaras sont discrétisées sur le même maillage distordu utilisé pour la discrétisation des équations de Navier-Stokes. Deux approches de résolution qui assurent la stabilité du schéma numérique et la positivité de la solution sont proposées. Une première, à base d'éléments finis et volumes finis qui a déjà fait l'objet d'une publication (Lorin et al., 2007). La deuxième approche est basée sur une discrétisation purement éléments finis avec un traitement spécial du terme source. Une publication est en cours de préparation. En particulier on vise à :

- Utiliser la méthode des éléments finis dans le but de traiter convenablement les termes visqueux, l'utilisation d'un maillage non structuré purement tétraédrique formé par des éléments anisotropiques.

- Éviter l'emploi de la loi de paroi et des éléments prismatiques afin de simplifier l'étape de maillage.
- Adopter une méthode de résolution adaptée à ce type de maillage.
- Améliorer la stabilisation et la capture de choc pour éviter les instabilités numériques et les oscillations non physiques dues à la dominance de la convection, la présence de chocs et la forte distorsion des éléments.

Ce dernier module de calcul d'écoulement turbulent pourra être couplé avec les autres modules pour étudier les phénomènes aéroélastiques mettant en jeu la turbulence. Cependant, cette étude dépasse le cadre de la présente thèse.

Afin de mettre en évidence la pertinence du présent travail, des simulations numériques et des mesures de performance seront illustrées dans le chapitre 6.

CHAPITRE 1

PROBLÉMATIQUE

Le couplage entre l'écoulement et les mouvements de structures flexibles est un phénomène non-linéaire. Il peut provoquer des instabilités mettant en jeu l'intégrité des structures.

Le domaine de recherche est orienté vers le couplage aéroélastique dans un écoulement, c'est à dire, vers l'étude du comportement des structures dont les mouvements au sein de l'écoulement génèrent des efforts induits. Parmi les domaines les plus concernés, on distingue le génie civil (ponts souples, cheminées), les transports (aéronautique, terrestre) et le secteur de l'énergie (tubes d'échangeurs, éoliennes).

Dans ce chapitre, les problématiques qui sont les sources de motivation de ce travail sont présentées. Ensuite, les difficultés liées à la modélisation de l'écoulement 3D sont évoquées. Enfin, les problèmes de résolution numérique propres à chaque domaine physique sont énumérés.

1.1 Problématiques associées à l'aspect multiphysique

Avec le développement des domaines de recherche en calcul scientifique, des problèmes complexes multiphysiques attirent l'attention. Ces derniers présentent des difficultés, aussi bien au niveau expérimental qu'en modélisation mathématique et numérique. Un problème est dit multiphysique, s'il met en jeu deux ou plusieurs disciplines physiques ou plusieurs phénomènes qui interagissent. Le problème qui nous intéresse en particulier est celui de l'interaction fluide-structure. En effet, ce problème présente des difficultés majeures. D'une part, sa formulation mathématique nécessite la collaboration et l'interaction de plusieurs spécialités. La connaissance pertinente de chaque discipline est indispensable pour bien définir, formuler et résoudre le problème global. D'autre part, la modélisation numérique aboutit généralement

à des systèmes algébriques de grandes tailles. Aussi, les systèmes générés sont souvent très mal conditionnés et par conséquent extrêmement difficiles à résoudre.

La résolution séquentielle du problème d'interaction est fortement limitée par le côté informatique. Malgré les avancées technologiques importantes de ces dernières années, les systèmes informatiques représentent toujours un handicap majeur aussi bien au niveau du stockage qu'au niveau de la puissance de calcul.

Pour pallier ces difficultés mathématiques, numériques et informatiques, une subdivision du problème global en sous problèmes, pourrait s'imposer. Toutefois, la qualité et le coût de la solution restent toujours dépendants du choix de la stratégie de décomposition et de l'algorithme de résolution.

1.2 Problématiques associées à la modélisation de l'écoulement tridimensionnel

La modélisation numérique de l'écoulement du fluide compressible est souvent basée sur les équations d'Euler ou de Navier-Stokes. Mis à part les difficultés analytiques liées à ce problème, des difficultés numériques s'ajoutent. En effet, la stabilisation du schéma numérique et la capture des chocs présentent des contraintes importantes.

Notre stratégie de résolution des problèmes multiphysiques consiste à traiter chaque domaine physique individuellement. Le couplage est assuré par la communication des informations nécessaires à la résolution du problème global à chaque pas de temps. Cependant, un seul maillage est utilisé pour la discrétisation de tous les domaines physiques. La résolution du modèle de turbulence nécessite l'utilisation d'un maillage très fin proche de la paroi solide. Pour limiter sa taille, le maillage est alors plus raffiné dans la direction orthogonale à la paroi solide que dans les autres directions. Des éléments tétraédriques fortement distordus sont alors utilisés. L'apparition de ces éléments engendre une difficulté supplémentaire dans la stabilisation des schémas numériques.

1.3 Problématiques associées à la modélisation de la turbulence

La turbulence se manifeste lorsque le nombre de Reynolds représentatif de l'écoulement devient grand (Spalart et Allmaras, 1994; Wilcox, 1997). Elle est caractérisée par les propriétés suivantes :

- Les variations temporelles et spatiales de la vitesse sont aléatoires.
- Le champ de vitesse est tridimensionnel et rotationnel.
- La turbulence est un phénomène non-linéaire.
- La capacité de mélange est accrue.
- La turbulence a un caractère dissipatif : la présence de nombreuses échelles, qui interagissent entre elles, augmente le taux de dissipation d'énergie.
- La turbulence a un caractère diffusif : elle augmente le taux de mélange et de diffusion d'espèces chimiques et de la température.

La simulation des écoulements turbulents relève principalement de la résolution des équations de la mécanique des fluides. Étant donné la complexité des problèmes envisagés, il est nécessaire de recourir à des techniques permettant leur simulation numérique.

Une manière de décrire les écoulements est l'utilisation des outils statistiques. Les variables physiques sont alors décomposées en une valeur moyenne et une fluctuation turbulente. Le nouveau système d'équations obtenu régit donc le comportement de l'écoulement moyen. Cependant, la non linéarité des équations de conservation, gouvernant l'écoulement du fluide, fait

apparaître des inconnues supplémentaires sous forme de corrélations entre les composantes fluctuantes. La fermeture du système est alors réalisée au moyen d'hypothèses reposant principalement sur une représentation aussi correcte que possible des propriétés caractéristiques de la turbulence. Ces hypothèses forment un modèle de turbulence. Dans ce qui suit, nous adoptons le modèle de Spalart et Allmaras (Spalart et Allmaras, 1994) qui est très utilisé en aérodynamique externe. Le modèle de Spalart et Allmaras est un modèle à une seule équation scalaire qui se base sur une équation de transport de la viscosité turbulente. En pratique, la résolution de cette équation présente plusieurs difficultés numériques qui seront traitées dans le chapitre 5.

1.4 Problématiques associées au maillage

La modélisation numérique des écoulements turbulents nécessite l'utilisation d'un maillage dont le premier point se trouve à une distance très proche de la paroi solide (de l'ordre de $Y^+ \simeq 5$). Dans le cas des maillages structurés, cette contrainte peut être surmontée assez facilement en 2D. En revanche, la génération de ce type de maillage en 3D est complexe. D'autres contraintes, comme l'orthogonalité, compliquent la tâche des mailleurs. Aussi, la subdivision du domaine physique en plusieurs blocs est généralement inévitable.

Les maillages non-structurés sont plus simples à concevoir. Cependant, mailler le domaine physique jusqu'à la paroi solide avec un maillage purement tétraédrique engendre des éléments très aplatis. Plusieurs solutions sont envisageables. Si l'on suppose valable le profil logarithmique de la vitesse dans la couche limite, l'utilisation des lois de paroi est une solution peu coûteuse. Dans ce cas il n'est plus nécessaire de mailler cette zone. Toutefois, l'imposition des conditions aux limites devient plus complexe et la précision de la solution dans le cas des géométries complexes peut être affectée. La deuxième option consiste à l'utilisation des éléments prismatiques proche de la paroi solide. La génération de ce genre de maillage est complexe. En plus, le traitement de différents types d'éléments alourdit la résolution nu-

mérique. La résolution en maillage purement tétraédrique jusqu'à la paroi pose donc un défi important.

La solution adoptée dans le code PFES360 consiste à l'utilisation des éléments anisotropiques. Bien que la génération de ce genre de maillage présentait des contraintes majeures. Récemment, des logiciels commerciaux offrent cette option, en particulier la version 15.11 du logiciel Gridgen. Dans la plupart des cas, cette opération est possible par simple imposition de la hauteur de la première maille et un taux d'agrandissement. Cette méthode offre plusieurs avantages. Premièrement, les éléments tétraédriques linéaires sont déjà implémentés dans PFES. Deuxièmement, un seul type de maille (tétraèdres) est utilisé pour la discrétisation de toutes les équations gouvernantes. Troisièmement, l'intégration de ces équations jusqu'à la paroi solide permet une résolution rigoureuse du problème physique.

CHAPITRE 2

REVUE DE LITTÉRATURE

Dans ce chapitre, une brève et surement non exhaustive revue de littérature du domaine de simulation de l'aéroélasticité et des écoulements turbulents tridimensionnels est présentée.

2.1 Modélisation du couplage fluide–structure : l'aéroélasticité

L'étude de l'aéroélasticité non-linéaire est un domaine de recherche très actif. En pratique, la prédiction et l'analyse de ce phénomène nécessitent la mise en œuvre de plusieurs méthodes expérimentales et numériques complexes. L'industrie aéronautique est attirée par la rapidité et le faible coût (tableau 2.1) qui caractérisent les méthodes numériques. Les ressources informatiques, qui présentent l'obstacle le plus contraignant pour ces méthodes, ne cessent d'augmenter. Par exemple, la résolution des équations de Navier-Stokes moyennées sur un maillage tridimensionnel de plus d'un million de points était de l'utopie il y a deux décades. Actuellement, grâce à l'adoption des stratégies de calcul parallèle sur les calculateurs modernes, ce genre de simulations est devenu accessible (Peterson et Zing, 2008).

Tableau 2.1

Méthodes de prédiction

Méthode	Avantages	Inconvénients
Expérimentation	- Solution physique assurée	- Complexe et coûteuse
Analytique	- Solution directe	- Équations différentielles complexes - Peu de solutions aux problèmes réels
Numérique	- Économie par rapport à l'expérimentation - Accès à toutes les données de l'écoulement - Conditions réelles et idéales	- Modélisation numérique - Maillage de géométries complexes - Temps de développement et d'exécution élevé - Limité par la capacité des calculateurs - La solution dépend de plusieurs facteurs

Les premières observations des phénomènes aéroélastiques ont été constatées dès le début du 20^{ème} siècle avec l'apparition du vol propulsé. Il est probable que les échecs de Samuel Langley en 1903 sont dus à des instabilités de divergence en torsion. Récemment, des chercheurs (Auriti et DeLaurier, 2004) ont constaté que les charges des ailes étaient plus élevées que ce que la structure pourrait soutenir. Durant la même année les frères Wright ont effectué le premier vol propulsé, en partie, grâce à la maîtrise du contrôle latéral de leur avion "flyer" naturellement instable. Les divergences en torsion ont fortement contribué à la prédominance des avions biplan jusqu'au début des années 30. L'apparition des ailes à peau métallique a permis d'assurer une raideur tensorielle mais ce n'était pas suffisant. En effet, des problèmes aéroélastiques ont été découverts. Les modes de flottement dans les queues des avions étaient les plus fréquents. D'ailleurs, le premier cas documenté de ce genre de problème date de 1916 et concerne le bombardier "Handley Page 0/400". Les travaux de Lanchester et Bairstow ont permis de faire les corrections nécessaires sur ces avions. En 1928, Frazer et Duncan ont écrit un document intitulé "The Flutter of Aeroplane wings" (le flottement des ailes d'avion) qui est devenu une référence incontournable depuis cette époque et jusqu'à nos jours. Un autre cas documenté est celui d'un problème d'aéroélasticité statique qui est apparu sur le Bristol Bagshot en 1927. Le problème était analysé suite aux travaux de Roxbee Cox et Pugsley et des mesures préventives étaient appliquées. D'ailleurs ce sont ces deux chercheurs qui furent les premiers à utiliser le terme "aéroélasticité". Cependant, les vraies premières études du phénomène datent de 1947. Ces études ont montré que l'utilisation des flèches arrière permet d'éviter ce phénomène. Grâce à ces constatations et suite aux travaux ultérieurs, les techniques d'analyse des phénomènes d'aéroélasticité linéaire sont devenues des outils indispensables dans l'industrie aéronautique et dans plusieurs autres domaines tels que le génie civil et les turbomachines.

Plusieurs travaux sur l'état de l'art du calcul numérique ont été élaborés (Tucker, 2007). Vue la diversité des champs d'intérêts (fluide, structure, turbulence, stabilisation et capture de choc, maillage et calcul parallèle) et la popularité croissante du domaine d'étude, traduit par le

grand nombre de publications, nous nous sommes contraints de limiter notre exposé aux récents travaux (depuis 2001). En plus, nous nous intéressons spécifiquement aux publications qui traitent les méthodes de simulations des écoulements compressibles turbulents et l'interaction fluide-structure. En particulier, les travaux basés sur les équations de Navier-Stokes moyennées (RANS) et le modèle de Spalart-Allmaras comme modèle de fermeture.

Dans la suite nous survolons les travaux de recherche qui sont à notre avis les plus marquants.

Farhat et al (Farhat et al., 2002) puis Geuzaine et al (Geuzaine et al., 2003) présentent une méthode pour la simulation de l'aéroélasticité non linéaire. Elle est basée sur une discrétisation éléments finis de la formulation ALE (Arbitrary-Lagrangian-Eulerian) soit des équations d'Euler ou des équations Navier-Stokes. Les auteurs estiment que leurs méthodologies de simulation sont suffisamment matures mais pas assez rapides pour être considérées comme un outil de conception. Les auteurs déclarent qu'il est possible de simuler l'écoulement subsonique et trouver cinq points de flottement sur un avion de chasse en moins de dix jours. Des résultats sur la configuration d'un avion F16 sont présentés. Tang et al (Tang et al., 2003) présentent des tests pour quantifier l'effet des non linéarités aérodynamiques sur la détermination des oscillations de cycle limite (LCO). Entre autres, les auteurs concluent que la précision des calculs est très sensible au modèle de turbulence utilisé. Aussi, l'utilisation du modèle de Spalart-Allmaras aboutit à des solutions physiquement raisonnables par rapport à d'autres modèles. Washio et al (Washio et al., 2005) présentent deux méthodes de préconditionnement des systèmes issus du couplage fluide-structure. En 2006, Farhat (Farhat et al., 2006) présente une méthodologie de résolution des problèmes d'interaction fluide-structure. L'auteur met l'accent sur l'importance du schéma de discrétisation temporelle des équations gouvernantes du domaine fluide. Des résultats sur la réponse aéroélastique d'une configuration d'un avion F16 démontrent la stabilité numérique de la méthodologie. Zhang et al (Zhang et al., 2008) ont développé un modèle de simulation des écoulements compressibles visqueux avec mouvement de maillage. La méthode numérique est basée sur la résolution des équations de Navier-Stokes sur un maillage non structuré utilisant une approche de compressibilité artificielle.

Les équations gouvernantes sont écrites suivant une formulation Eulerienne-Lagrangienne et discrétisées suivant un schéma de volumes finis. Zhi et Mavriplis (Yang et Mavriplis, 2007) présentent une technique de résolution des problèmes aéroélastiques. La méthode utilise un couplage fort entre les équations gouvernantes du domaine fluide et du domaine structure. Le schéma de discrétisation est un schéma de volumes finis sur un maillage hybride. Des résultats de prédiction de limite de flottement de l'aile agard 445.6 sont présentés

2.2 Modélisation des écoulements turbulents

La première description connue de la turbulence est due à Léonard de Vinci (Leonardo da Vinci, 1452 - Amboise, 1519). Bien plus tard, à l'aide d'un procédé se basant sur les filtres colorés, O. Reynolds, en 1883, avait analysé ce phénomène en visualisant l'écoulement dans des conduits rectilignes de sections circulaires (Hinze, 1975; Cousteix, 1989; White, 1991; Wilcox, 1997). L'expérience, fréquemment répétée, montre que suffisamment loin de l'entrée de la conduite et à faible nombre de Reynolds, les lignes de courant sont parfaitement parallèles à l'axe de la conduite. Dans ce cas, l'écoulement est stationnaire et il est dit laminaire. À plus grand nombre de Reynolds, des instabilités apparaissent sous forme d'ondes, l'écoulement est devenu alors instationnaire tout en restant parfaitement organisé, il n'est pas encore turbulent. À très grand nombre de Reynolds, l'écoulement devient complètement irrégulier, il est dit turbulent.

La turbulence n'est donc pas une propriété du fluide, mais un régime d'écoulement. On ne peut dire que l'écoulement n'est plus laminaire dès qu'il est instationnaire et on ne peut, non plus, affirmer qu'il est pleinement turbulent dès qu'il n'est plus laminaire. De ce fait, il n'existe pas de définition universelle de la turbulence. Selon Hinze (Hinze, 1975), un écoulement turbulent est un écoulement irrégulier dont les propriétés physiques varient de façon aléatoire dans le temps et dans l'espace.

En écoulement turbulent des fluides compressibles, la vitesse, la pression, la densité, la viscosité, la conductivité thermique et la température fluctuent. En effet, pour un nombre de Mach

élevé, il convient de tenir compte des variations des propriétés du fluide, tels que la masse volumique, le coefficient de viscosité et la conductivité thermique. Les moyennes pondérées par la masse, couramment utilisées dans le cas des fluides compressibles, ont été développées par Favre (Wilcox, 1997).

Avec l'utilisation des moyennes de Reynolds et de Favre, les équations moyennes de Navier-Stokes ont une forme assez simple, en particulier, l'équation de conservation de la masse qui garde sa forme habituelle. Dans l'équation de conservation de la quantité de mouvement, il apparaît des tensions turbulentes $-\overline{\rho u' \otimes u'}$ appelées aussi tensions de Reynolds. De même, l'équation de conservation de l'énergie fait intervenir un terme de diffusion turbulente. Ce dernier terme est interprété comme un flux de chaleur turbulent. Le système d'équations est alors ouvert. La fermeture du système peut être réalisée à l'aide de certains modèles basés sur des hypothèses approchant le plus possible les caractéristiques de la turbulence.

À nos jours, les modèles de turbulence sont d'une grande importance dans plusieurs domaines de la science. À titre d'exemple, Mittal et al (Mittal et al., 2001) ont utilisé la technique de simulation des écoulements turbulents pour l'étude de poulx à travers une sténose artérielle modélisée. Les modèles approchant les caractéristiques de la turbulence sont généralement classés suivant deux critères : l'ordre et le nombre d'équations de fermeture supplémentaires.

- ◇ Modèles du premier ordre basés sur les valeurs moyennes des fonctions de l'écoulement. Ces modèles, appelés aussi modèles à viscosité turbulente, sont basés sur l'hypothèse de Boussinesq qui consiste à modéliser directement les tensions de Reynolds à l'aide d'une viscosité turbulente. Ils sont classifiés par rapport à leurs nombre d'équations de fermetures comme suit :
- Modèles à zéro équation : ils consistent à relier les flux turbulents et les valeurs moyennes sans introduire de nouvelles équations. Parmi ces modèles, nous citons le modèle de Van

Driest (Van Driest, 1956) et la longueur de mélange de Prandtl et Van Karman.

- Modèles à une équation : ils prennent en compte l'historique de la turbulence et une échelle empirique de longueur. Parmi les modèles à une équation, nous citons le modèle de Bradshaw, Ferriss et Atwell (1967), le modèle de Baldwin et Barth (1990) et celui de Spalart et Allmaras (Spalart et Allmaras, 1994).
- Modèles à deux équations : très répandus, ils prennent en compte les variations spatiales de l'agitation turbulente. Parmi ces modèles, nous citons le modèle $k - \varepsilon$ (W P Jones et B E Launder (Jones et Launder, 1972), Launder et Sharma (Launder et Sharma, 1974), Hoffmann (1975), Reynolds (1976)) et le modèle $k - \omega$ (Kolmogorov (Kolmogorov, 1942), Saffman (1970), Launder et Spalding (1972) qui ont développé une version améliorée de celui de Kolmogorov, Wilcox et al (1972-1988), Speziale, Abid et Anderson (Speziale et al., 1992)).

Les modèles du premier ordre à deux équations aboutissent à des résultats satisfaisants pour le cas des écoulements parallèles aux parois solides. En pratique, un grand nombre d'applications, particulièrement en aéroélasticité, sont caractérisées par des zones d'impact fluide-structure. Les modèles à viscosité turbulente surestiment la turbulence dans ces zones. Par exemple, le modèle $k - \varepsilon$ surestime les transferts thermiques dans l'impact d'un jet sur une paroi chauffée. D'où l'intérêt des modèles plus sophistiqués comme le modèle V2F (Durbin, 1995).

- ◊ Modèles du second ordre où l'équation de transport des tensions de Reynolds est résolue tout comme les équations du champ moyen. Ces modèles conviennent aux écoulements avec tourbillons, à forte anisotropie, et permettent de ne pas surestimer l'énergie cinétique dans le cas de forts cisaillements (écoulements avec séparation de région). Nous

citons à titre d'exemples les travaux de Donaldson et Rosenbaum (1968), Daly et Harlow (1970), Reece et Rodi (1975), Lumely (1978), Speziale (1987) et Reynolds (1987).

◇ Les modèles à bas Reynolds.

Lorsque le nombre de Reynolds de la turbulence est inférieur à 100, on procède par les modèles dits à bas Reynolds.

Il existe trois familles de modèles

- Algébrique
- Double couche : l'écoulement est divisé en deux couches. Une formule de viscosité turbulente est associée à chacune de ces couches.
- Multi-équations : c'est une adaptation du modèle $k - \varepsilon$.

2.2.1 Le modèle de Spalart-Allmaras

Le modèle de Spalart-Allmaras est un modèle simple d'implémentation, stable et bien adapté à la modélisation des couches limites dans des gradients de pression adverses et des sillages. Ce modèle a été introduit par P.R. Spalart et S.R. Allmaras (Spalart et Allmaras, 1994). Ils proposent de modéliser la viscosité turbulente en utilisant une équation de transport scalaire composée d'un terme de diffusion non-linéaire, un terme de destruction et un terme de production. Le modèle utilise l'hypothèse de Boussinesq, c'est à dire que le tenseur de Reynolds $-\rho \bar{u}_i \bar{u}_j$ peut s'écrire en fonction de la viscosité turbulente ν_t comme suit : $-\rho \bar{u}_i \bar{u}_j = \nu_t S_{ij}$ où S_{ij} est le tenseur des taux de déformations moyennes. Le modèle contient huit coefficients de fermeture et trois fonctions d'amortissement.

En plus des caractéristiques avancées, notre choix de ce modèle repose aussi sur le grand nombre de travaux de recherche qui ont montré sa validité en aérodynamique externe. Paciorri et al de l'institut des dynamiques des fluides de Von Karman en Belgique (Paciorri et al., 1998) ont montré que dans le cas des écoulements qui impliquent une séparation laminaire ou un rattachement de turbulences, le modèle est en excellent accord avec les données expérimentales.

Ils ont aussi montré que si le maillage utilisé est bien adapté, le modèle donne de bons résultats dans le cas des séparations turbulentes. Roy et Blottner (Roy et Blottner, 2003) ont comparé la précision numérique, la sensibilité, et la validité de plusieurs modèles de turbulence dont le modèle de Spalart-Allmaras dans le cas d'une plaque plane avec friction. Les auteurs ont mis en évidence la précision et la sensibilité du modèle. Chi R. Wang (Chi, 2005) affirme que les valeurs des composantes de cisaillement calculées par un modèle de turbulence, basé sur celui de Spalart et Allmaras, sont très proches des valeurs expérimentales.

En pratique, plusieurs travaux concernant la simulation des écoulements turbulents basés sur la résolution des équations de Navier-stokes moyennées et le modèle de Spalart-Allmaras ont marqué le domaine de recherche. Yong Zhao et Ahmed Forhad (Zhao et Forhad, 2003) proposent une méthode générale pour la simulation des écoulements bidimensionnels sur des structures en mouvement. Tezduyar et Senga (Tezduyar et Senga, 2006) présentent des nouvelles définitions de la matrice τ de la méthode SUPG et de l'opérateur de capture de choc. Le même auteur, (Tezduyar, 2007) publie une étude sur les méthodes de capture des chocs les plus utilisées telles que la méthode SUPG et leurs utilisations pour les écoulements turbulents et les écoulements sur des structures en mouvement. Un travail similaire est présenté par Franco Rispoli et al (Rispoli et al., 2007). Zhixiang Xiao et al (Xiao et al., 2006) utilisent un code de calcul parallèle basé sur une discrétisation volumes finis des équations de Navier-Stokes moyennées et le modèle de turbulence $k-\omega$. Les maillages utilisés sont des maillages hybrides. Le code prédit convenablement les zones de chocs, les vortices et les phénomènes d'interaction entre l'aile et le fuselage. De son côté, Dimitri J. Mavriplis (Mavriplis, 2007) présente une stratégie d'optimisation des profils des ailes d'avions. La méthode est basée sur une résolution tridimensionnelle des équations d'Euler ou de Navier-stokes sur un maillage non structuré. Les équations sont discrétisées en volumes finis. Jakirlic et al (Jakirlic et al., 2007) présentent deux modèles de résolution des équations de Navier-Stokes moyennées (RANS) et du modèle de turbulence de Spalart-Allmaras ou du modèle $k-\omega$ de Wilcox. Les équations gouvernantes sont discrétisées en volumes finis sur un maillage structuré. Les auteurs pré-

sentent des résultats de simulations de l'écoulement sur l'aile Onera M6 et montrent que les méthodes présentées capturent convenablement le choc en λ . Suite à des tests de validation de leur code parallèle basé sur une résolution des équations de Navier-Stokes et du modèle de Spalart-Allmaras, Landmann et al (Landmann et al., 2008) ont soulevé la question de la positivité de la viscosité turbulente ν_t . Les auteurs estiment que l'étude de la positivité du modèle est un champ de recherche très important. Svacek (Svacek, 2008) présente des résultats de simulation d'un écoulement incompressible, visqueux bidimensionnel. Le système d'équations est formé par une discrétisation éléments finis des équations de Navier-Stokes couplées avec les équations décrivant le mouvement de la structure (aile). Aussi, il présente des résultats de simulations trouvés par la résolution des équations de Navier-Stokes moyennées (RANS) avec le modèle de turbulence de Spalart-Allmaras. L'auteur compare les résultats trouvés avec les résultats du code NASTRAN. Peterson et al (Peterson et Zing, 2008) ont présenté un algorithme basé sur la méthode Newton-Krylov pour la simulation des écoulements 3-D compressibles. La méthode de résolution est basée sur une discrétisation en volumes finis des équations de Navier-Stokes et l'équation de Spalart-Allmaras. Des simulations sur l'Onera M6 et le DLR F6 utilisant deux configurations de maillages, hybride et purement tétraédriques, sont en concordance avec les résultats expérimentaux. Dans le cas purement tétraédrique, seul le cas de DLR F6 a été étudié.

D'autres auteurs proposent des améliorations pour adapter le modèle à des cas de configurations bien spécifiques. Aupoix et Spalart (Aupoix et Spalart, 2003) ont introduit deux extensions au modèle pour tenir compte de la rugosité de la paroi solide. Ils ont été indépendamment développés par Boeing et l'ONERA. Les deux supposent une viscosité non nulle au mur et changent la définition de la distance d de sorte que le modèle devient non-local. Bien avant, Lee et Paynter (Lee et Paynter, 1996) ont aussi proposé une modification pour la simulation des écoulements sur des surfaces rugueuses. Rung et al (Rung et al., 2003) suggèrent de transformer la constante C_{b1} en une fonction du taux de vitesse de déformation. Sebastien et al (Deck et al., 2002) ont développé une extension du modèle Spalart-Allmaras adaptée aux flux

3-D compressibles supersoniques. Le modèle a abouti à de bons résultats notamment dans des cas tests complexes comme la simulation des écoulements sur un missile. En effet, leur méthode a la capacité de saisir avec précision l'interaction entre les tourbillons du fuselage et les ailettes. Paciorri et Sabetta (Paciorri et Sabetta, 2003) proposent une correction pour tenir compte des effets de compression. L'analyse des résultats numériques et leurs comparaisons avec les données expérimentales montrent que la correction proposée produit une amélioration significative aux prévisions numériques. Plusieurs travaux sur la simulation des écoulements avec séparation ont été effectués (Spalart et al., 1997; Squires et al., 2002). Lorin et al (Lorin et al., 2007; Lorin et al., 2006; Ben Haj Ali et al., 2004) ont développé une méthode de discrétisation du modèle qui assure la positivité de la viscosité turbulente. Les auteurs ont présenté une analyse mathématique détaillée et des résultats numériques dans le cas d'une plaque plane. Une comparaison avec la solution analytique et des études de convergence ont révélé l'intérêt de la méthode.

Dans le cadre de cette thèse, nous proposons une nouvelle méthode pour la résolution des problèmes d'aéroélasticité et des écoulements turbulents tridimensionnels. Notre approche est basée sur une discrétisation en éléments finis des équations de Navier-Stokes moyennées (RANS). Le modèle de fermeture utilisé est une nouvelle variante, discrétisée en éléments finis, de celui de Spalart et Allmaras. Des nouvelles définitions de la matrice τ de la méthode SUPG ainsi que de l'opérateur de capture de chocs sont utilisées pour la stabilisation de la solution numérique. Cette nouvelle méthode permet l'utilisation d'un maillage purement tétraédrique extrêmement aplati. Nous concluons que la méthode proposée est une première dans le domaine des simulations des écoulements tridimensionnels visqueux par éléments finis.

CHAPITRE 3

MÉTHODOLOGIE ET ALGORITHMIQUE

Dans ce chapitre, la méthodologie, les plus importants détails techniques et les aspects informatiques d'implémentation du programme général PFES360 sont présentés. La méthodologie de résolution est basée sur une approche de décomposition à la fois fonctionnelle et géométrique (Soulaïmani et al., 2004; Soulaïmani et Ben Haj Ali, 2003; Ben Haj Ali, 2002). Pour profiter pleinement de l'approche proposée, le programme (PFES360) est développé selon une approche modulaire distribuée. Dans cette partie, l'algorithme général est présenté et les différents modules sont brièvement exposés avant d'être étudiés en détails dans les chapitres 4 et 5. Dans ce qui suit, l'approche standard (séquentielle) de résolution des problèmes multiphysiques est tout d'abord rappelée. Ensuite, l'approche modulaire (parallèle) est présentée.

3.1 Résolution séquentielle des problèmes multiphysiques

Une approche souvent utilisée pour la résolution des problèmes multiphysiques est l'approche séquentielle. Cette dernière consiste à décomposer le problème global en plusieurs sous-programmes selon leurs aspects physiques. Le programme général est alors une alternance entre les sous-programmes représentant les différents domaines physiques. Dans le cadre de ce travail, nous nous intéressons en particulier aux problèmes d'interaction fluide-structure et à la résolution des écoulements turbulents tridimensionnels. Ce problème met en jeu plusieurs domaines physiques. Un domaine pour le fluide, un autre pour la structure et éventuellement un autre pour la turbulence. La résolution numérique de ce problème nécessite aussi un domaine pour le calcul du mouvement du maillage et une interface de couplage appelée ici 'Matcher'. Le Matcher assure le passage des valeurs nodales entre le maillage fluide et le maillage de la structure qui sont généralement différents. La figure 3.1 donne une illustration globale de ce problème complexe. L'algorithme 1 est essentiellement basé sur une alternance entre le solveur du fluide CFD (Computational Fluid Dynamic) et le solveur de la structure

CSD (Computational Structure Dynamic), intercalé par un appel du Matcher.

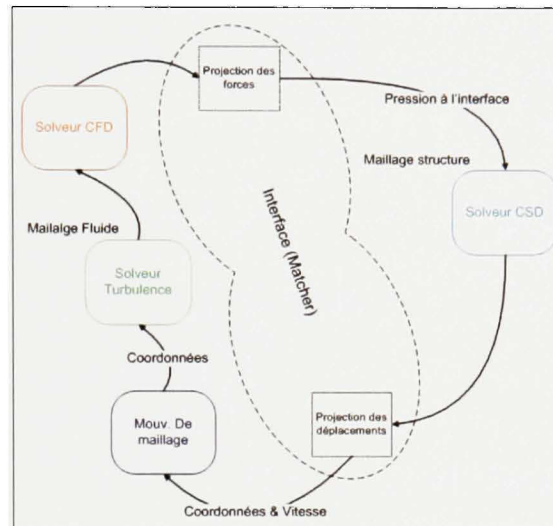


Figure 3.1 *Approche séquentielle pour la résolution du problème d'aéroélasticité.*

Le Matcher assure deux fonctions :

- Projeter les pressions calculées par le fluide sur les nœuds de la structure.
- Mettre à jour les conditions aux limites du fluide en se basant sur les déplacements de la structure.

Dans le cas où le fluide est modélisé par les équations moyennes de Navier-Stokes, l'algorithme peut être résumé comme suit :

Algorithme 1 : Algorithme séquentiel de couplage

Données : U^n, ν_t^n, X^n

Résultat : $U^{n+1}, \nu_t^{n+1}, X^{n+1}$

initialisation. **répéter**

- Calculer la viscosité turbulente ν_t^{n+1} ;
- Calculer le champs de vitesse U^{n+1} ;
- Projeter les forces aérodynamiques sur la structure ;
- Calculer les déplacements sur l'interface ;
- Calculer le mouvement de maillage ;
- Déduire les nouvelles coordonnées X^{n+1} ;

jusqu'à Satisfaction du critère de convergence ;

3.2 Résolution parallèle des problèmes multiphysiques

Dans la littérature, les termes calcul parallèle et calcul distribué sont généralement évoqués dans différents contextes. Les définitions suivantes sont adoptées par C. Leopold (Leopold, 2000).

- En calcul parallèle, une application est partagée en plusieurs tâches qui sont exécutées en même temps.
- En calcul distribué, l'application est partagée en plusieurs tâches qui seront accomplies en utilisant plusieurs ressources.

Dans le cadre de ce travail, cette nuance n'est pas prise en considération. En effet, les aspects rencontrés :

- Utilisation de plusieurs processeurs,
- Processeurs interconnectés,
- Plusieurs activités progressent en même temps et communiquent entre elles pour résoudre un problème global.

sont des caractéristiques communes.

Une première façon de paralléliser l'algorithme séquentiel de la figure 3.1 est de procéder par une pure décomposition géométrique du problème global (Quarteroni et Valli, 1999).

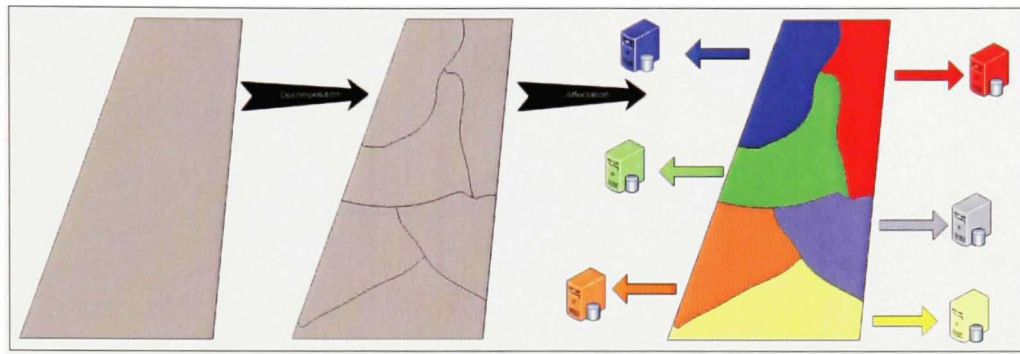


Figure 3.2 *Décomposition géométrique du domaine.*

Les sous-domaines géométriques sont alors affectés aux processeurs qui exécutent la même tâche et en même temps. Cette approche est appelée SPMD (Single Program Multiple Data), figure 3.3.

L'avantage de cette approche est la simplicité de son implémentation. En effet, le programme est très similaire à la version séquentielle. Seulement les fonctions de communication sont ajoutées.

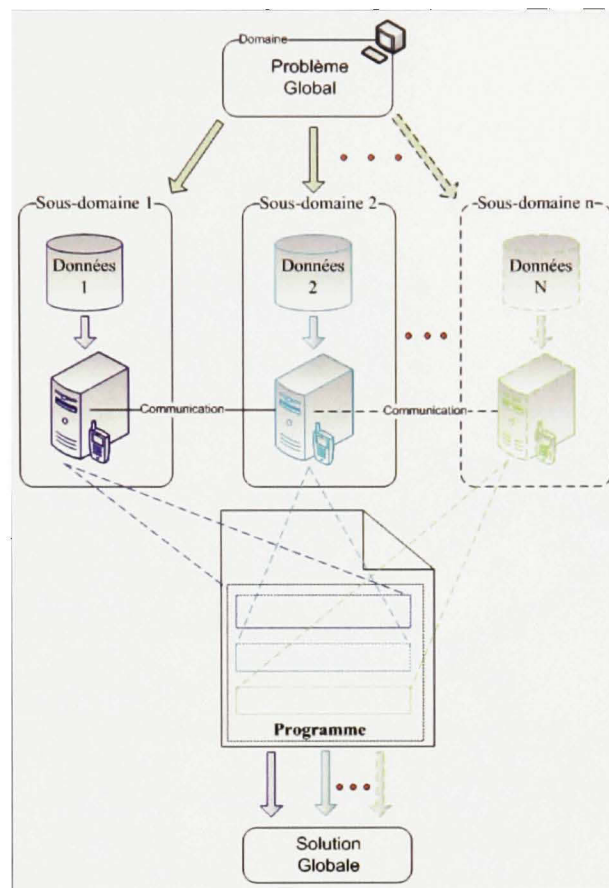


Figure 3.3 *Le paradigme SPMD : Single Program Multiple Data.*

La figure 3.4 illustre l'utilisation de cette approche dans la résolution du problème multiphysique d'interaction fluide-structure ou aéroélasticité turbulente. L'approche SPMD est très utilisée dans le cadre de parallélisation des programmes séquentiels.

Encore une fois, le calcul de viscosité turbulente peut être effectué séparément, ainsi la viscosité turbulente sera calculée dans un module à part et transmise au solveur fluide.

L'algorithme parallèle 3.4 peut être amélioré (optimisé) en utilisant la décomposition fonctionnelle.

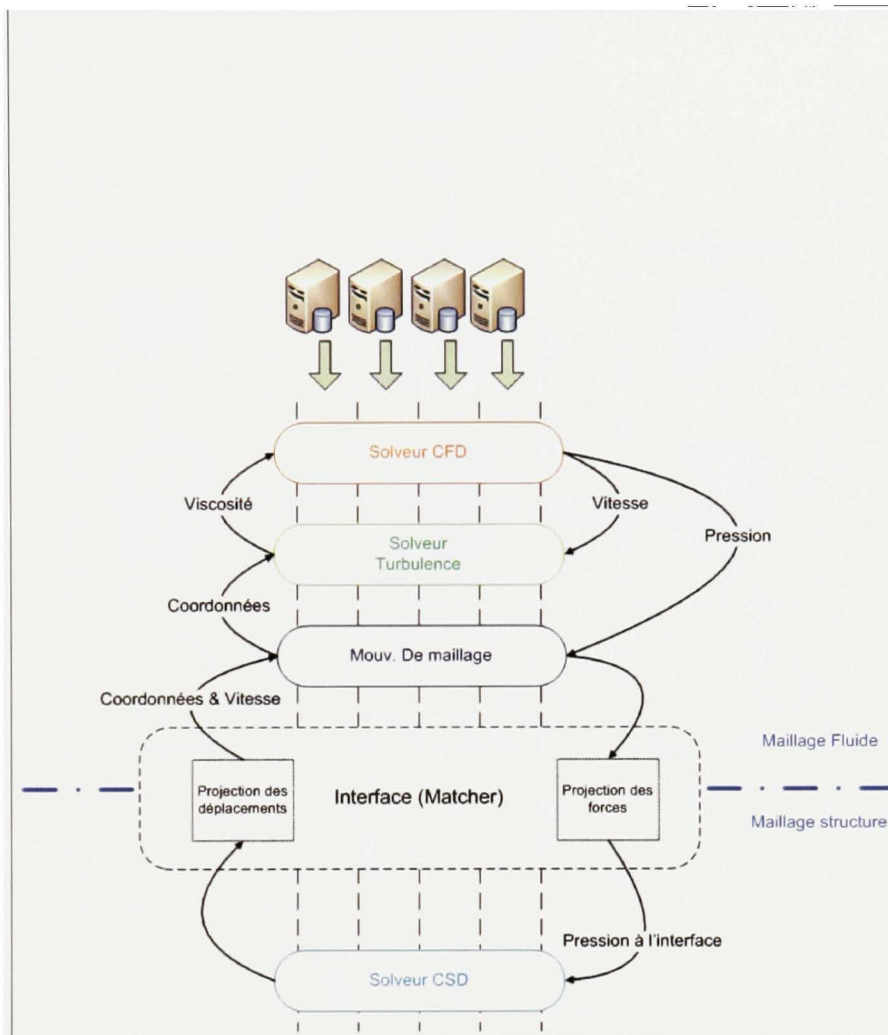


Figure 3.4 *Approche SPMD en aéroélasticité.*

3.3 Méthode de décomposition spatio-fonctionnelle

Une deuxième manière de paralléliser l'algorithme séquentiel (figure 3.1) est de diviser le problème en plusieurs domaines fonctionnels, un domaine pour le fluide, un autre pour la turbulence et un autre pour la structure et le mouvement de maillage. Cette approche est basée sur le paradigme MPMD (Multiple Program Multiple Data) illustré par la figure 3.5

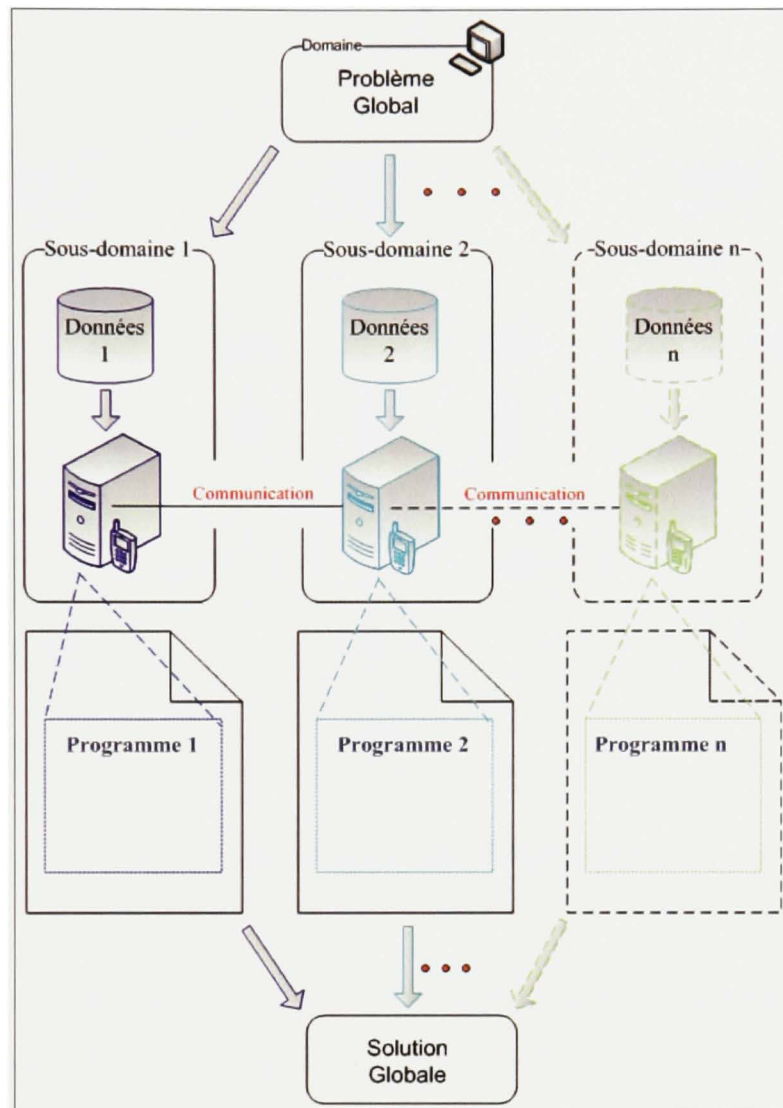


Figure 3.5 *Le paradigme MPMD : Multiple Program Multiple Data.*

Une décomposition fonctionnelle (MPMD) est adoptée en premier. Ensuite, elle est suivie par une décomposition géométrique (SPMD) de chaque domaine physique. L'approche adoptée est illustrée par la figure 3.6.

La communication entre les processeurs de la même famille (traitant le même domaine fonctionnel) est appelée intra-communication. La communication entre les familles est appelée intercommunication.

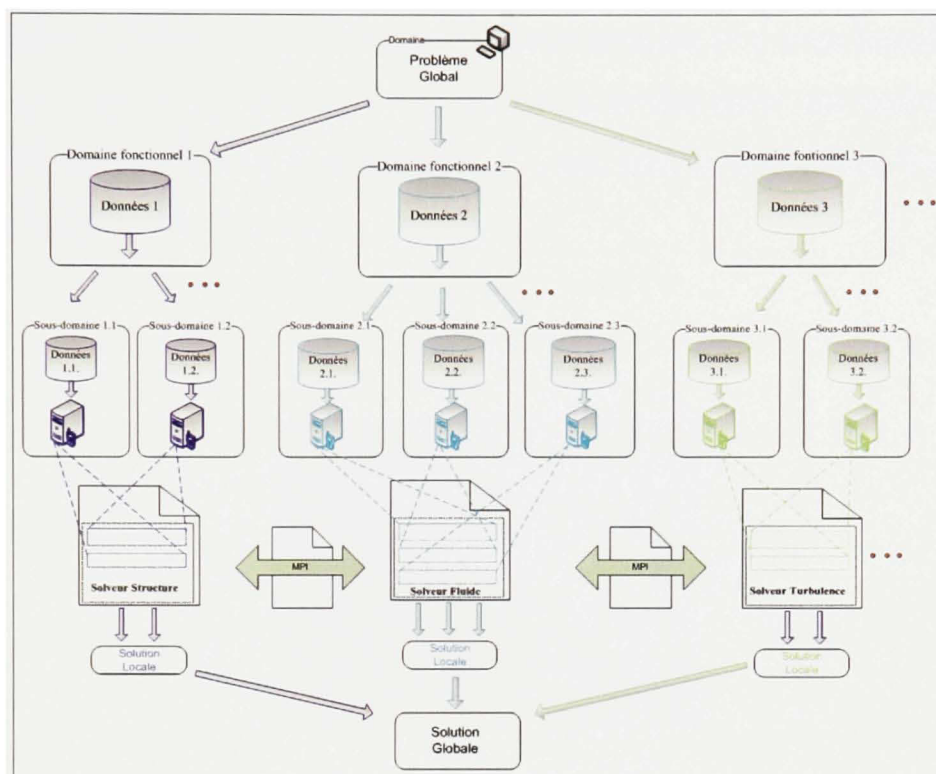


Figure 3.6 *L'approche MPMD-SPMD.*

Plusieurs algorithmes de résolution peuvent être considérés pour résoudre ce problème ainsi divisé (Ben Haj Ali, 2002; Soulaïmani et al., 2001b; Soulaïmani et al., 2002c).

3.3.1 Méthode de décomposition de Schwarz

La méthode de décomposition de domaine utilisée est la méthode de Schwarz développée en 1870 (Quarteroni et Valli, 1999; Lucquin et Pironneau, 1996; Saad, 2003). Le domaine est alors divisé en plusieurs sous-domaines $\{\Omega_1, \Omega_2, \dots, \Omega_n\}$. D'une manière générale, la méthode consiste à parcourir les sous-domaines i et de résoudre le système local en utilisant des conditions aux limites basées sur la solution la plus récente des autres sous-domaines. La solution globale du problème est trouvée par assemblage des solutions locales.

Soit un domaine Ω , la procédure consiste à diviser ce domaine en n sous-domaines Ω_i . Avec i allant de 1 à n . Les frontières internes des différents sous-domaines sont notées par $\Gamma_{i,j}$. $\Gamma_{i,j}$ est la frontière interne à Ω_i avec le domaine Ω_j . $\Gamma_{i,0}$ sont les traces de la frontière du domaine Ω au niveau des sous-domaines Ω_i .

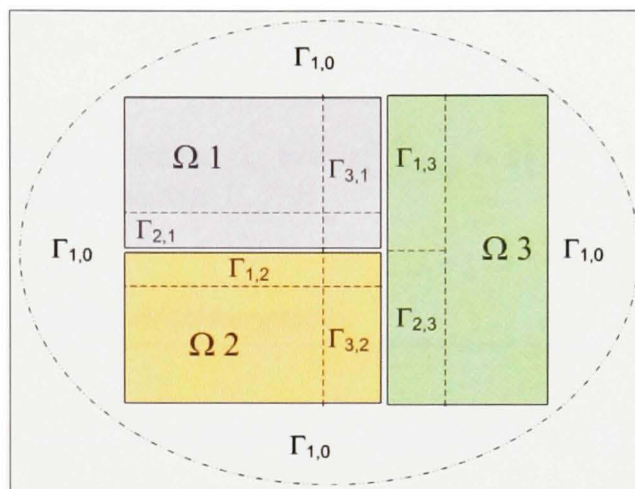


Figure 3.7 *Domaines et frontières dans la méthode de Schwarz.*

Deux variantes de la méthode sont généralement utilisées. Schwarz multiplicative et Schwarz additive.

3.3.1.1 Méthode de Schwarz multiplicative

La méthode de Schwarz multiplicative (Saad, 2003) peut s'écrire comme suit :

Algorithme 2 : Schwarz multiplicative

Données : u_0 et Solution initiale et s le nombre de domaines.

Résultat : u .

initialisation.

répéter

pour $i = 1, s$ **faire**

 Calculer u_i^{k+1} dans le domaine Ω_i , avec $u_i^{k+1}|_{\Gamma_{i,j}} = u_j^k$;

 Mettre à jour u sur la frontière $\Gamma_{i,j} \forall j$;

 ;

fin

jusqu'à Satisfaction du critère de convergence ;

La figure 3.8 illustre le calcul de u dans Ω_1 à l'itération $k + 1$, de la méthode de Schwarz multiplicative.

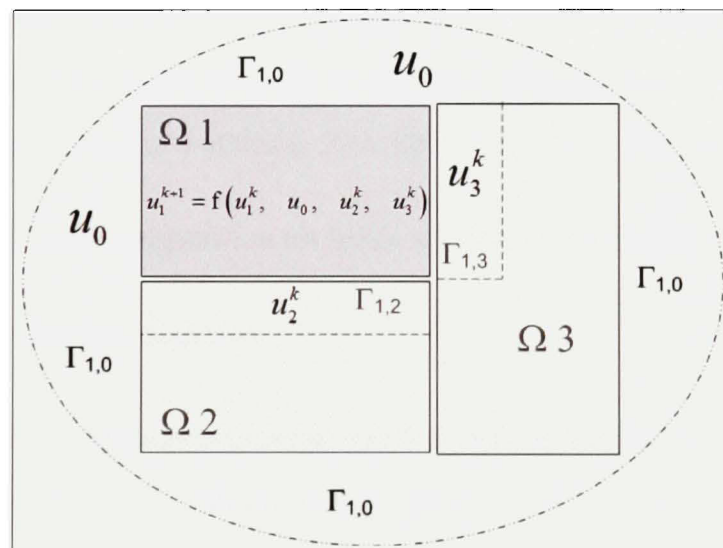


Figure 3.8 Méthode de Schwarz multiplicative.

3.3.1.2 Méthode de Schwarz additive

Cette deuxième variante de l'algorithme de Schwarz est différente de la procédure de Schwarz multiplicative par le fait que la solution n'est mise à jour qu'à la fin de la boucle sur tous les domaines.

La méthode Schwarz additive peut s'écrire comme suit :

Algorithme 3 : Schwarz additive

Données : u_0 et Solution initiale et s le nombre de domaines.

Résultat : u .

initialisation.

répéter

pour $i = 1, s$ **faire**

 Calculer $\delta_i = \mathbf{R}_i^T \mathbf{A}_i^{-1} \mathbf{R}_i (\mathbf{b} - \mathbf{A}u)$;

fin

$u^{k+1} = u^k + \sum_{i=1}^s \delta_i$;

jusqu'à Satisfaction du critère de convergence ;

Où s est le nombre de domaines, \mathbf{R}_i est l'opérateur de restriction associé au sous-domaine i et $\mathbf{A}_i = \mathbf{R}_i \mathbf{A} \mathbf{R}_i^T$ la matrice locale associée au sous-domaine. \mathbf{R}_i est une matrice de dimension $n_i \times n$ qui ne contient que des 0 et des 1. Pour plus de détails voir (Saad, 2003).

Notre méthodologie de décomposition est basée sur cette dernière (Ben Haj Ali, 2002).

3.4 Algorithme par sous-domaine

Le programme PFES360 est un programme général capable de résoudre une grande variété de problèmes physiques et multiphysiques. Pour le cas d'un problème de simulation d'interaction fluide-structure, l'algorithme de résolution est globalement basé sur une approche modulaire. La structure de données est basée sur SPARSKIT (A Basic Tool Kit for Sparse Matrix Computations) et PSPARSLIB (A Portable Library of Parallel Sparse Iterative Solvers) développé par Saad (Saad, 1994; Saad et Malevsky, 1995; Saad et al., 1997) et adaptée aux éléments finis, dans une première version de PFES (Ben Haj Ali, 2002).

Les sections suivantes survolent les différents modules (domaines physiques) implémentés dans PFES360. Les étapes communes à tous les algorithmes sont discutées. Les détails de chaque module sont exposés ultérieurement dans des chapitres dédiés.

La figure 3.9 illustre les différents modules nécessaires à la résolution d'un problème d'interaction fluide structure selon notre méthodologie de décomposition.

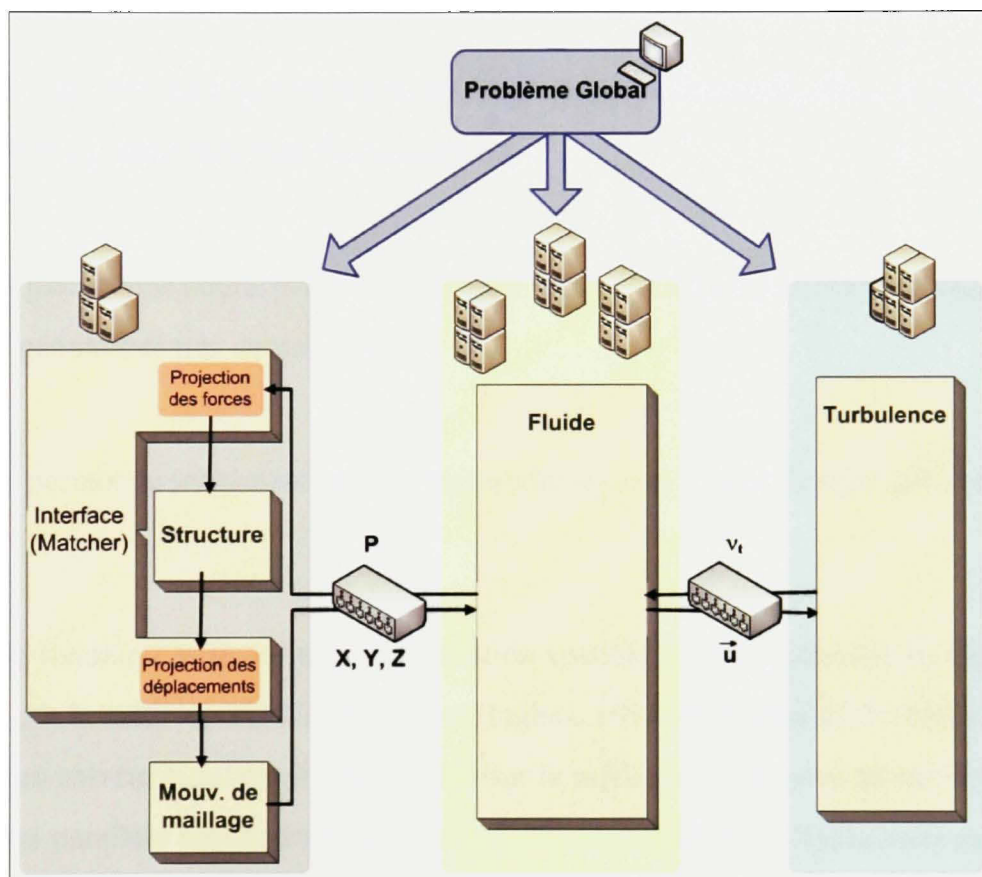


Figure 3.9 *Décomposition fonctionnelle du problème d'aéroélasticité.*

Rappelons qu'en plus de la décomposition fonctionnelle, chaque domaine physique est décomposé géométriquement. Pour plus de détails sur la décomposition géométrique voir (Ben Haj Ali, 2002)

3.4.1 Module de fluide

Grâce à sa puissance et à sa flexibilité, la méthode des éléments finis est largement utilisée en ingénierie. C'est une méthode des résidus pondérés, d'où sa flexibilité d'associer une variété de formulations variationnelles à un modèle différentiel donné. Les facteurs qui ont conduit au succès de cette méthode sont les suivants :

- Elle peut s'adapter facilement à des maillages non structurés, d'où sa puissance pour modéliser des géométries très complexes.
- Elle permet une bonne précision à un coût raisonnable grâce à l'espace d'approximation des géométries très complexes.
- Elle permet un traitement naturel des conditions aux limites de type différentiel.

Le module fluide est basé sur une discrétisation spatiale tridimensionnelle en éléments finis stabilisée par la méthode SUPG (Brooks et Hughes, 1994) (Chapitre 4). La résolution est assurée par un solveur parallèle implicite basé sur la méthode de décomposition de Schwarz et une version parallèle non-linéaire de GMRES (Saad et Schultz, ; Soulaïmani et al., 2002a; Soulaïmani et al., 2001c) dite PGMRES.

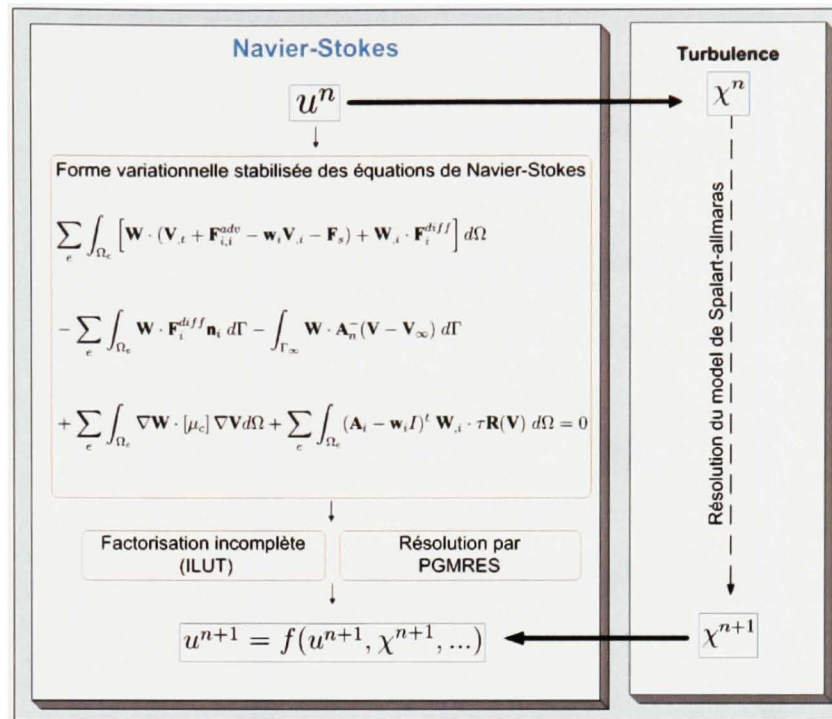


Figure 3.10 Organigramme du module de fluide.

3.4.2 Module de turbulence

Le module de turbulence est proposé sous forme de deux approches. Elles sont détaillées au chapitre 5. Une approche consiste à une discrétisation simultanée en éléments finis et en volumes finis des différents termes de l'équation de transport de Spalart et Allmaras (Spalart et Allmaras, 1994; Lorin et al., 2006; Lorin et al., 2007). La deuxième repose sur une discrétisation éléments finis où le terme source est calculé par une approximation semi-implicite. Les figures 3.11 et 3.12 donnent une illustration simplifiée de ces deux approches.

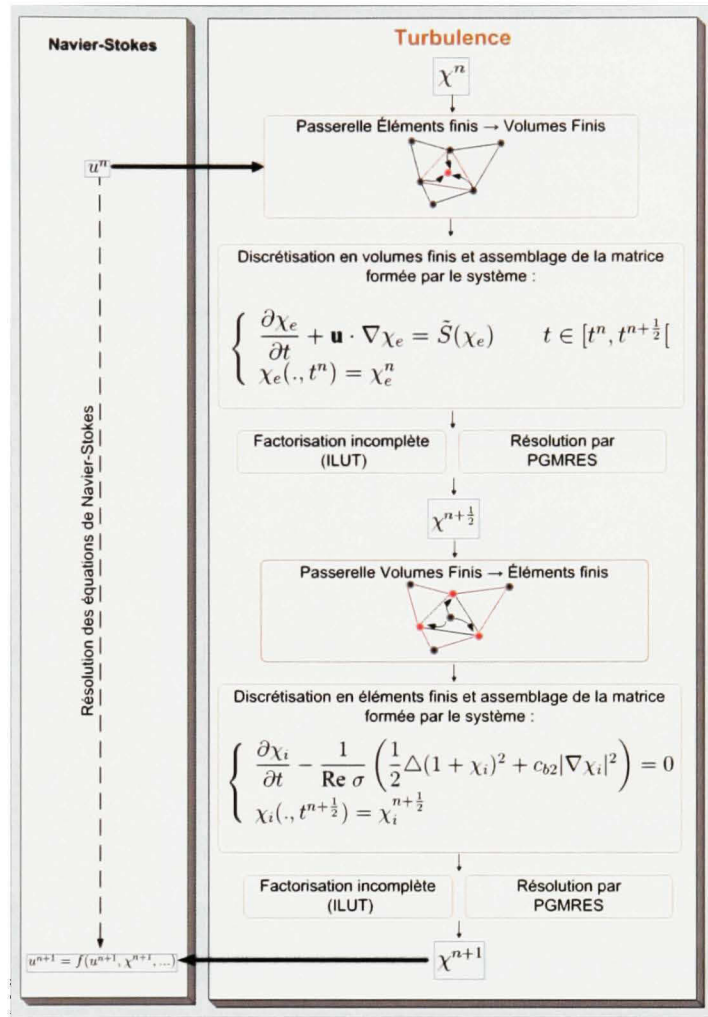


Figure 3.11 *Turbulence : organigramme de la première approche.*

Une étude rigoureuse concernant la modélisation, les méthodes de résolution et d'autres aspects numériques tel que la positivité du schéma numérique sont présentés au chapitre 5.

3.4.3 Module de structure

Le déplacement \mathbf{u}^s d'une structure élastique flexible satisfait les équations d'équilibre dynamiques suivantes :

$$\rho_s \mathbf{u}_{i,tt}^s + \sigma_{ij,j}^s(\mathbf{u}^s) = g_i \text{ dans } \Omega_s \quad (3.1)$$

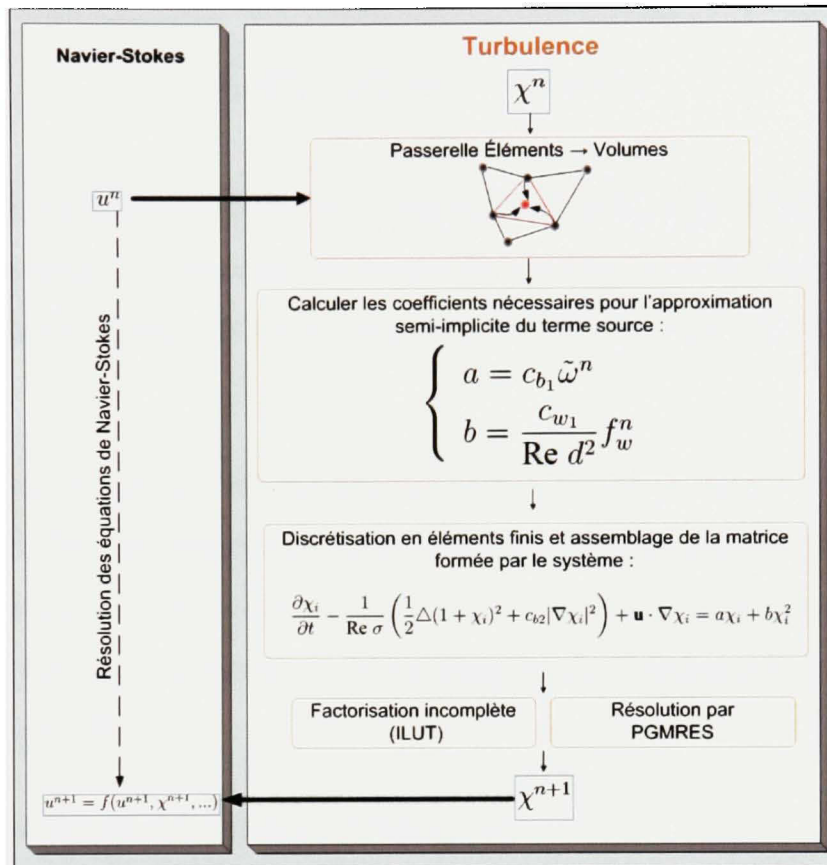


Figure 3.12 *Turbulence : organigramme de la deuxième approche.*

où ($i = 1, 3$), ρ_s est la masse volumique, \mathbf{g} est le vecteur des forces volumiques internes et $\boldsymbol{\sigma}^s$ est le tenseur de Cauchy. Les déplacements de la structure sont supposés petits (Soulaïmani et al., 2002c; Soulaïmani et al., 2001a; Soulaïmani et al., 2002b).

3.4.4 Module de mouvement de maillage

Plusieurs choix peuvent être considérés pour établir l'opérateur de distribution du mouvement du domaine mobile à partir de celui de l'interface fluide-structure. Dans le programme PFES360, le mouvement de maillage est défini par les équations d'élasticité.

Le déplacement \mathbf{u}^m d'un maillage est donné par la relation suivante :

$$\rho_m \mathbf{u}_{i,tt}^m + \boldsymbol{\sigma}_{ij,j}^m(\mathbf{u}^m) = 0 \text{ dans } \Omega_f \quad (3.2)$$

où ($i = 1, 3$), ρ_m est la densité fictive et $\boldsymbol{\sigma}^m$ est le tenseur de Cauchy du milieu élastique fictif formé par le maillage. La vitesse de mouvement de maillage est donnée par $\mathbf{w} = \mathbf{u}_{,t}^m$. Dans le cas visqueux, les conditions aux limites sont $\mathbf{u}^m = \mathbf{u}^s$ sur la frontière fluide–structure et zéro ailleurs.

Au cours du mouvement, il n'est pas toujours garanti que les éléments de maillage gardent une taille acceptable pour des calculs assez précis des écoulements. En effet, les petits éléments peuvent subir de grandes distorsions. Afin d'éviter les volumes négatifs et les grandes distorsions, des techniques adéquates ont été appliquées (Soulaïmani et al., 2002c; Soulaïmani et al., 2001a; Feng, 2005).

3.5 Résolution des grands systèmes d'équations

Dans la résolution des systèmes de grandes tailles tels que les problèmes multiphysiques, l'algorithme standard de la méthode des éléments finis (Ben Haj Ali, 2002) s'avère assez lourd en termes de mémoire et temps de calcul. En effet, dans ces cas, les opérations de factorisation, de produit matriciel et de résolution, sont de plus en plus coûteuses. Pour remédier à ces inconvénients, les modifications suivantes sont apportées à l'algorithme standard :

- Factorisation incomplète ILUT : Les matrices résultantes de la modélisation des problèmes étudiés, sont souvent mal conditionnées (Saad, 2003). De ce fait, leur préconditionnement est inéluctable. Toutefois, l'obtention d'un préconditionneur convenable passe très souvent par une factorisation de la matrice. Une des alternatives envisageables est la méthode LU. Dans notre cas, l'application de la factorisation LU sous sa version standard dite complète, est très coûteuse. La méthode finalement adoptée est une facto-

risation approximative. Il s'agit d'une méthode basée sur une factorisation LU incomplète communément connue sous le nom ILUT (Saad, 2003; Soulaïmani et Ben Haj Ali, 2003).

Algorithme 4 : ILUT : Factorisation LU incomplète

```

for  $i = 1, n$  do
  for  $k = 1, i - 1$  do
     $A_{i,k} = A_{i,k} / A_{k,k}$  ;
    if  $|A_{i,k}|$  is not too small then
      for  $j = k + 1, n$  do
         $A_{i,j} = A_{i,j} - A_{i,k} * A_{k,j}$  ;
      end
    end
  end
  Drop small element in row  $A_{i,*}$  ;
end

```

- Calcul du résidu : un des fardeaux de l'algorithme présenté ci-dessus, est le coût du produit matrice-vecteur. Néanmoins, ce problème est contourné grâce à une méthodologie de résolution basée sur le calcul de résidu élément par élément (Soulaïmani et Ben Haj Ali, 2003; Soulaïmani et al., 2002a; Soulaïmani et al., 2000).
- Parallélisation : en dépit des améliorations apportées par l'adoption de la factorisation incomplète ILUT et le calcul du résidu, la parallélisation de l'algorithme demeure inévitable vu la grande taille des problèmes traités. Outre le gain quantitatif apporté par la subdivision du domaine (temps/mémoire), la parallélisation offre une grande flexibilité de programmation. Mieux encore, cette subdivision offre l'avantage de refléter l'aspect multiphysique des problèmes abordés.

Algorithme 5 : Algorithme de résolution par sous-domaine

```

Boucle sur le pas de temps pour  $I_{pas} = 1, N_{pas}$  faire
| Itération de couplage pour  $Iter = 1, N_{iter}$  faire
| | Boucle sur les éléments pour  $I_{ela} = 1, N_{ELL}$  (Nombre d'éléments local) faire
| | | Calculer la matrice locale  $A_{locale}^e$  ;
| | | Assemblage de  $A_{locale}^e$  dans  $A_{locale}$  ;
| | fin
| | Envoie de la partie  $A_{Int-local}$  vers les voisins ;
| | Application de Schwartz additive avec une seule couche de recouvrement pour
| | l'assemblage définitif de  $A_{locale}$  ;
| | Factorisation incomplète de la matrice  $A_{locale}$  (ILUT) ;
| | si Non linéaire alors
| | | Linéariser
| | fin
| | Résolution avec PGMRES (Produit matrice-vecteur éviter par un calcul de résidu) ;
| | Mise à jour de la solution locale ;
| fin
| Assemblage de la solution globale ;
fin

```

3.6 Implémentation informatique

Le langage utilisé pour l'implémentation informatique de cet algorithme est le Fortran 90. En effet, cette norme permet d'allouer dynamiquement la mémoire. Malheureusement, ce langage n'a pas intégré la possibilité de faire la réallocation dynamique qui est un outil extrêmement utile dans notre cas, car les dimensions des tables ne sont pas toujours connues d'avance. Pour remédier à cet inconvénient, des routines de réallocation ont été implémentées (Ben Haj Ali, 2002).

PFES360 est basé sur la méthode de décomposition spatio-fonctionnelle illustrées par la figure 3.6. La bibliothèque MPI est utilisée pour établir les communications des données entre les processeurs. Les fonctions les plus utilisées de la bibliothèque sont exposées à l'annexe 1.

Le programme PFES360 fait appel à quatre communicateurs :

- 1) Le communicateur MPI_COMM_WORLD est le communicateur par défaut de MPI. Il permet de communiquer des informations entre tous les processeurs.

- 2) Le communicateur `Family` est un sous-communicateur de `MPI_COMM_WORLD`.

`Family` permet la communication entre les processeurs de la même famille (l'intra-communication). On peut le créer en faisant l'appel suivant :

```
CALL MPI_COMM_SPLIT(MPI_COMM_WORLD,myfamily-1, &
                    myproc-1,Family,ierr)
```

où `myfamily` est le numéro de la famille. Il convient de donner un nom à chaque domaine fonctionnel (cela correspond à la notion de groupe selon la terminologie de MPI), cela se fait par l'appel suivant :

```
CALL MPI_COMM_GROUP(Family,FamilyGroup,ierr)
```

Le domaine fonctionnel *FamilyGroup* possède donc le communicateur `Family`.

- 3) Le communicateur `Diplomacy` permet la communication entre les leaders des familles (inter-communication). De la même façon, on le crée par les appels suivants :

```
!---- Connaître le groupe associé à MPI_COMM_WORLD
CALL MPI_COMM_GROUP (MPI_COMM_WORLD,WorldGroup,ierr)
!---- Créer le groupe des leaders
CALL MPI_GROUP_INCL (WorldGroup,nbfamily, &
                    leader,leadersGroup,ierr)
!---- Créer le communicateur entre les leaders
CALL MPI_COMM_CREATE (MPI_COMM_WORLD,leadersGroup, &
                    Diplomacy,ierr)
```

- 4) Les voisins d'un sous domaine sont évidemment les sous-domaines avec lesquels il partage des nœuds. Chaque sous-domaine a donc besoin de communiquer avec ses voisins. Pour éviter les communications point à point, un sous communicateur de `Family` appelé `Neighbors` est créé. Contrairement aux autres communicateurs, le commu-

Le communicateur `Neighbors` est non classique. Ce dernier peut être vu comme une table de communicateurs. Chaque élément de cette table est utilisé par un processeur bien déterminé afin d'envoyer des informations à ses voisins. La complexité de l'implémentation de `Neighbors` vient du fait que les différents groupes qui constituent les voisins ne sont pas indépendants. En effet, un processeur peut appartenir à plusieurs groupes en même temps. De ce fait, il est indispensable de construire un communicateur dédié à chaque processeur.

Pour la création du sous-communicateur `Neighbors` chaque processeur a besoin de connaître la liste de ses voisins et les voisins de chaque processeur de la famille. Pour cela, chaque processeur construit les tables suivantes :

- La liste de ses voisins : `Voisins`

par exemple pour le processeur P2 de la figure 3.13 la liste des voisins est

`Voisins = [1, 2, 4]`

- Le nombre cumulatif des voisins de chaque processeur de la famille : `Pointeur`

Dans le cas de la figure 3.13 :

`Pointeur = [1, 4, 7, 10, 13, 15]`

- La liste de tous les voisins : `Voisinage`

i.e., chaque processeur construit une table `Voisinage` lui permettant de connaître la liste des voisins de n'importe quel processeur de sa famille. Cette opération se réalise (efficacement) par exemple en faisant appel à la routine `MPI_ALLGATHERV` de `MPI`.

```
CALL MPI_ALLGATHERV(voisins,nproc+1,MPI_INTEGER, &
Voisinage,npsize,pointeur-1,MPI_INTEGER,FAMILY,ierr)
```

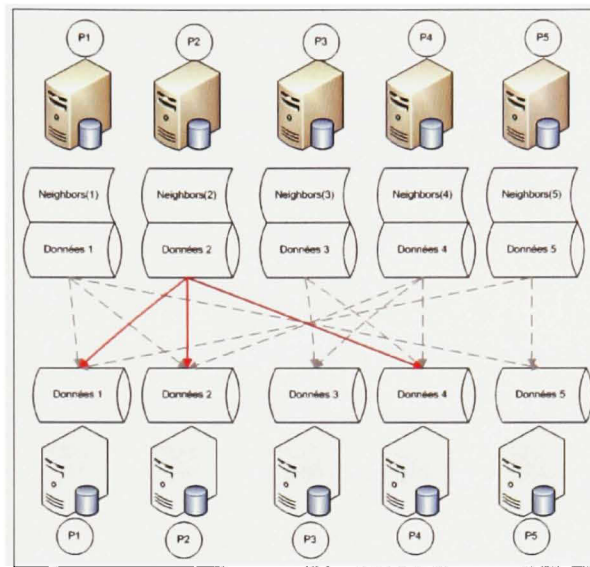


Figure 3.13 *Communication entre les domaines voisins.*

où `nproc` est le nombre de voisins et `npsize` est un vecteur qui contient le nombre de voisins de chaque processeur et qui peut être calculé à partir de `pointeur`. Dans le cas de la figure 3.13 :

`Voisinage = [1, 3, 5, 1, 2, 4, 1, 3, 4, 1, 5]`.

Grâce à cette structure chaque processeur possède toutes les informations nécessaires.

Par exemple, tous les processeurs savent que :

Le processeur P2 possède :

`pointeur(3) - pointeur(2) = 7 - 4 = 3` processeurs voisins.

Ces processeurs sont :

`Voisinage(pointeur(2):pointeur(3)-1) = Voisinage(4:(7-1))`

`Voisinage(4:(7-1)) = Voisinage (4 : 6)`

`Voisinage (4 : 6) = [1, 2, 4]`

On a dit que chaque processeur i possède une liste de voisins. On affecte alors un nom `proc_voisins(i)` à cet ensemble de processeurs (ou sous-groupe) et un communicateur `Neighbors(i)`. Cela se réalise comme suit :

```
DO i=1, numbrother
  !---- Créer les groupes de processeurs voisins
  CALL MPI_GROUP_INCL(FamilyGroup,npsize(i), &
    voisinage(pointeur(i):pointeur(i+1)-1), &
    proc_voisins(i), ierr)
  !---- Créer le communicateur entre les voisins
  CALL MPI_COMM_CREATE (FAMILY, proc_voisins(i), &
    Neighbors(i), ierr)
ENDDO
```

La procédure générale de résolution d'un problème multiphysique comme celui de l'interaction fluide-structure est illustrée à la figure 3.14.

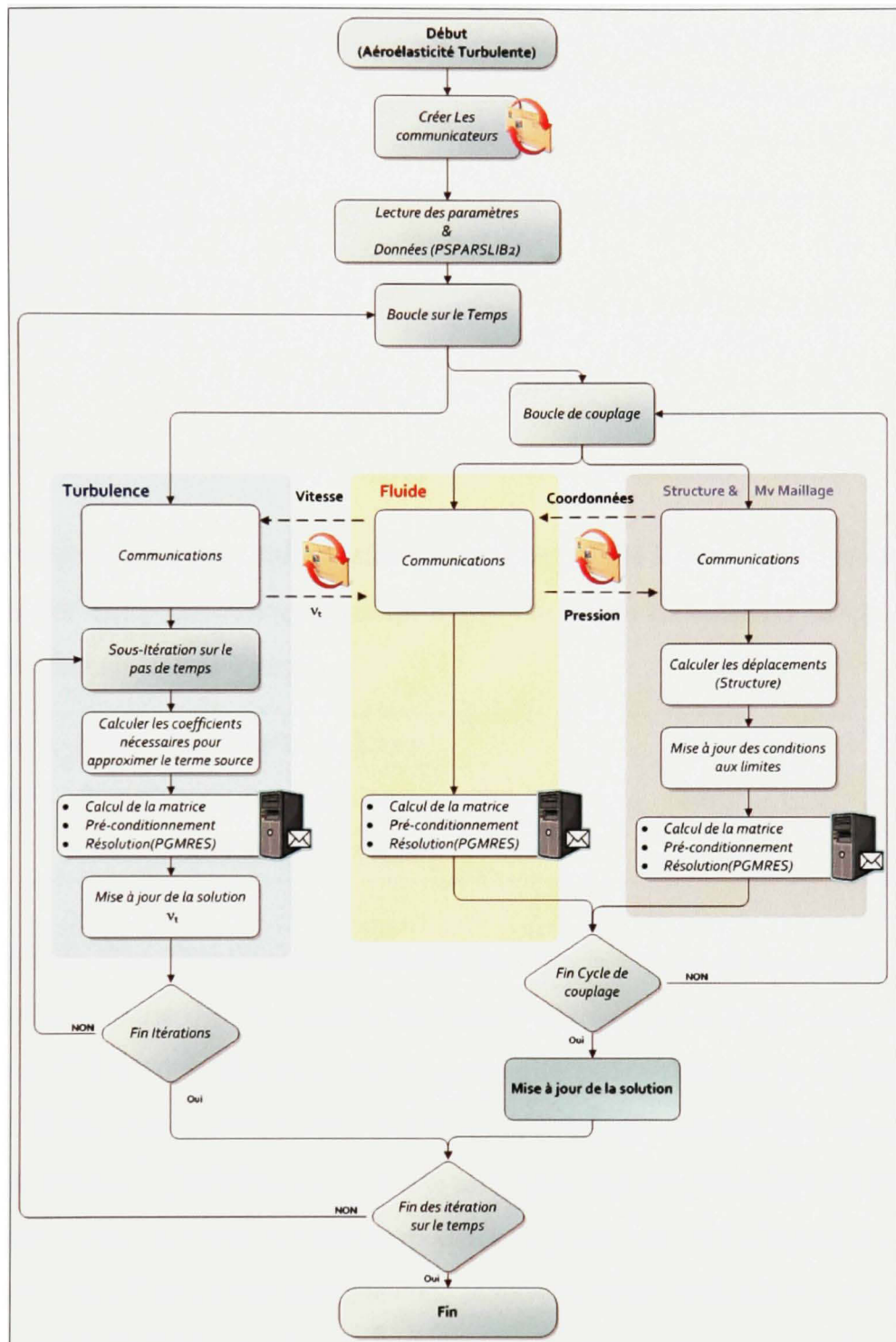


Figure 3.14 Organigramme général du logiciel PFES360.

À titre d'illustration, nous montrons comment le leader du groupe fluide envoie le vecteur des pressions nodales PRESS au leader du groupe structure en utilisant le communicateur Diplomacy pour l'intercommunication.

```

IF (myid==1) THEN
CALL MPI_BARRIER(DIPLOMACY,ierr)
CALL MPI_BCAST(PRESS,NCLI,MPI_DOUBLE_PRECISION, &
               0,DIPLOMACY,ierr)
ENDIF

```

Enfin, nous montrons un exemple d'intra-communication : le leader du groupe structure possède le vecteur VDLF des déplacements qu'il distribue à tous les membres de son domaine en utilisant le communicateur Family.

```

CALL MPI_BARRIER(FAMILY,ierr)
CALL MPI_BCAST(VDLF,3*dimVDLF,MPI_DOUBLE_PRECISION, &
               0,FAMILY,ierr)

```

Ces techniques d'implémentation ont attribué au code PFES360 diverses qualités conceptuelles. Outre la clarté des algorithmes, la rapidité d'exécution et la précision des résultats numériques, PFES360 se distingue par sa modularité caractérisée par la flexibilité et l'extensibilité. D'une part, n'importe quel module peut être facilement modifié ou remplacé. Par exemple, le module de turbulence basé sur le modèle de Spalart-Allmaras peut être remplacé par un autre. Ainsi, une bibliothèque de modules peut être mise à la disposition des utilisateurs. D'autre part, à titre d'extension, le code séquentiel Atlas3D (Soulaïmani et Saad, 1998) a été intégré dans PFES360 comme module d'écoulement à surface libre. Des tests de validation sur un problème de rivière en 3D ont été conduits avec succès.

CHAPITRE 4

RÉSOLUTION DES ÉQUATIONS D'EULER ET DE NAVIER-STOKES MOYENNÉES

Le but est de réaliser un code informatique qui simule l'état d'un fluide compressible. Cet état est caractérisé par les variables physiques dépendantes suivantes :

\mathbf{u} la vitesse ;

ρ la densité ;

p la pression ;

T la température.

Ces variables sont déterminées par la résolution des équations de Navier-Stokes ou d'Euler dans le cas où la viscosité est négligée, d'où son nom "Module fluide". En effet, connaissant les propriétés internes d'un fluide, à savoir viscosité dynamique μ , conductivité thermique k , et chaleurs spécifiques C_p et C_v , les équations de Navier-Stokes permettent de déterminer ces variables physiques en fonction du temps et de l'espace. Ces équations sont :

- Équation de continuité (Conservation de la masse) ;
- Équation de la quantité de mouvement (Loi de Newton) ;
- Équation de la conservation de l'énergie ;
- Équation d'état.

La résolution de ce système d'équations nécessite l'ajout des conditions aux limites et des conditions initiales. Théoriquement, ces équations modélisent tout écoulement, indépendamment de sa nature (laminaire, instationnaire ou turbulent), son état (compressible ou non compressible) ou son régime (subsonique, transonique ou supersonique). En pratique, ces équations non-linéaires engendrent des problèmes mathématiques complexes et des systèmes nu-

mériques de très grandes tailles. La résolution numérique de ce problème présente plusieurs défis mathématiques et numériques.

Dans la suite de ce chapitre, les différentes formes des équations de Navier-Stokes sont rappelées, la discrétisation par la méthode des éléments finis est présentée et finalement les algorithmes de résolution numérique sont détaillés.

4.1 Équations de Navier-Stokes

Les équations de Navier-Stokes peuvent s'écrire sous plusieurs formes. Malgré que le schéma numérique proposé est basé sur une formulation ALE (Arbitrary-Lagrangian-Eulerian) de la forme conservative en variables conservatives (Section 4.1.4), les autres formes sont mentionnées à titre de rappel.

4.1.1 Forme conservative

Les équations de Navier-Stokes peuvent s'écrire suivant la forme conservative suivante :

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0 \\ \frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p - \nabla \cdot \boldsymbol{\sigma} &= \rho \mathbf{f}_v \\ \frac{\partial}{\partial t} \left[\rho \left(i + \frac{1}{2} |\mathbf{u}|^2 \right) \right] + \nabla \cdot \left[\mathbf{u} \rho \left(i + \frac{1}{2} |\mathbf{u}|^2 \right) \right] - \nabla \cdot (\boldsymbol{\sigma} \cdot \mathbf{u}) - \nabla \cdot \mathbf{q} + \nabla \cdot (p \mathbf{u}) &= \rho \mathbf{f}_v \cdot \mathbf{u} + r \end{aligned} \quad (4.1)$$

où i est l'énergie interne ($i = C_v T$), p est la pression (loi des gaz parfaits : $p = (\gamma - 1) \rho i$), q est le flux de chaleur (loi de Fourier : $q = -k_e \nabla T$), \mathbf{f}_v est le vecteur des forces volumiques, r est la source de chaleur et $\boldsymbol{\sigma}$ est le tenseur des contraintes visqueuses qui est relié au tenseur taux de déformation D par la relation suivante (loi de Stokes) :

$$\boldsymbol{\sigma} = \lambda (\nabla \mathbf{u}) \mathbf{I} + 2\mu \mathbf{D}.$$

avec $2\mu + 3\lambda = 0$ (Constantes de Lamé reliées par la loi de Stokes) et $D_{ij} = \frac{1}{2}(\mathbf{u}_{i,j} + \mathbf{u}_{j,i})$.

L'expression détaillée du tenseur des déformations est donnée par :

$$\mathbf{D} = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial u_2}{\partial x_1} + \frac{\partial u_1}{\partial x_2}\right) & \frac{\partial u_2}{\partial x_2} & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial u_3}{\partial x_1} + \frac{\partial u_1}{\partial x_3}\right) & \frac{1}{2}\left(\frac{\partial u_3}{\partial x_2} + \frac{\partial u_2}{\partial x_3}\right) & \frac{\partial u_3}{\partial x_3} \end{bmatrix} \quad (4.2)$$

4.1.2 Forme non conservative

Les équations de Navier-Stokes peuvent aussi s'écrire suivant la forme non conservative :

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \rho \nabla \cdot \mathbf{u} + \mathbf{u} \cdot \nabla \rho &= 0 \\ \rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - \nabla \cdot \boldsymbol{\sigma} &= \rho \mathbf{f}_v \\ \rho \frac{\partial i}{\partial t} + \rho \mathbf{u} \cdot \nabla i + p \nabla \cdot \mathbf{u} - \nabla \mathbf{u} : \boldsymbol{\sigma} - \nabla \cdot \mathbf{q} &= \rho \mathbf{f}_v \cdot \mathbf{u} + r \end{aligned} \quad (4.3)$$

4.1.3 Forme conservative écrite en variables conservatives

Les simulations envisagées consistent à modéliser des écoulements compressibles. Ces problèmes sont généralement caractérisés par la présence d'un choc. Sachant que la vitesse est discontinue à travers le choc, il est indispensable d'écrire le modèle de Navier-Stokes avec les variables conservatives suivantes :

ρ la densité,

$\mathbf{U} = \rho \mathbf{u}$ la vitesse,

$E = \rho e_t$ l'énergie totale par unité de volume.

$e_t = i + \frac{1}{2}|\mathbf{u}|^2$ l'énergie totale.

La forme conservative en variables conservatives s'écrit alors comme suit :

$$\frac{\partial \rho}{\partial t} + \nabla \cdot U = 0$$

$$\frac{\partial U}{\partial t} + \nabla \cdot \left[U \otimes \frac{U}{\rho} \right] + \nabla p - \nabla \cdot \boldsymbol{\sigma} = \rho \mathbf{f}_v \quad (4.4)$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \left[(E + p) \frac{U}{\rho} \right] - \nabla \cdot \left(\boldsymbol{\sigma} \cdot \frac{U}{\rho} \right) - \nabla \cdot q = r + \mathbf{f}_v \cdot U$$

4.1.4 Formulation eulerienne-lagrangienne des équations gouvernantes

Dans le but de faire des simulations d'écoulement autour des structures aéroélastiques, il est important d'écrire les équations gouvernantes dans un repère mobile. En effet, la forme conservative (4.4) décrite dans la section (4.1.3) n'est pas adaptée à la résolution des problèmes à frontières mobiles. Pour surmonter cette restriction, il est indispensable d'introduire la vitesse du mouvement du repère dans la formulation mathématique du problème considéré. La formulation ALE (Arbitrary-Lagrangian-Eulerian) (Soulaïmani et Saad, 1998; Farhat, 1995; Farhat et Lesoinne, 1996) est alors utilisée.

4.1.5 Forme compacte des équations de Navier-Stokes

L'équation de continuité, l'équation de quantité de mouvement et l'équation de l'énergie forment les équations de Navier-Stokes suivant la formulation ALE :

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \nabla \cdot U - \mathbf{w} \nabla \rho &= 0 \\
\frac{\partial U}{\partial t} + \nabla \cdot (U \otimes \frac{U}{\rho}) - \mathbf{w} \nabla U &= \rho \mathbf{f}_v - \nabla p + \nabla \cdot \boldsymbol{\sigma} \\
\frac{\partial E}{\partial t} + \nabla \cdot [(E + p) \frac{U}{\rho}] - \mathbf{w} \cdot \nabla E &= \mathbf{f}_r \cdot U + \nabla \cdot (\boldsymbol{\sigma} \cdot \frac{U}{\rho}) - \nabla \cdot (k_e \nabla T)
\end{aligned} \tag{4.5}$$

où \mathbf{w} est la vitesse de mouvement de maillage.

Les démarches de développement de ces équations suivant la formulation ALE sont détaillées par Feng (Feng, 2005).

Une forme adimensionnelle du système (4.5) peut être trouvée en utilisant les grandeurs de référence (indice r) et les variables adimensionnelles suivantes :

$$\begin{aligned}
\mathbf{u}^* &= \frac{\mathbf{u}}{\mathbf{u}_r} & \rho^* &= \frac{\rho}{\rho_r} & p^* &= \frac{p}{\rho_r |\mathbf{u}_r|^2} \\
i^* &= \frac{i}{|\mathbf{u}_r|^2} & \mu^* &= \frac{\mu}{\mu_r} & \mu^* &= \frac{\mu}{\mu_r} \\
x^* &= \frac{x}{\mathbf{L}_r} & t^* &= \frac{t}{\mathbf{L}_r / |\mathbf{u}_r|} & T^* &= \frac{T}{|\mathbf{u}_r|^2 / C_v}
\end{aligned}$$

Pour alléger la notation, on confond désormais les variables réduites (adimensionnelles) avec les variables conservatives.

La forme adimensionnelle du système (4.5) s'écrit comme suit :

$$\begin{aligned}
 \frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{U} - \mathbf{w} \nabla \rho &= 0 \\
 \frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \left(\mathbf{U} \otimes \frac{\mathbf{U}}{\rho} \right) - \mathbf{w} \nabla \mathbf{U} &= \rho \mathbf{f}_v - \nabla p + \nabla \cdot \boldsymbol{\sigma} \\
 \frac{\partial E}{\partial t} + \nabla \cdot \left[(E + p) \frac{\mathbf{U}}{\rho} \right] - \mathbf{w} \cdot \nabla E &= \mathbf{f}_v \cdot \mathbf{U} + \nabla \cdot \left(\boldsymbol{\sigma} \cdot \frac{\mathbf{U}}{\rho} \right) - \nabla \cdot (k_e \nabla T)
 \end{aligned} \tag{4.6}$$

où

$$\mathbf{U} = \rho \mathbf{u} \Rightarrow \mathbf{u} = \frac{\mathbf{U}}{\rho}$$

$$E = \rho e_t$$

$$\boldsymbol{\sigma}(\mathbf{u}) = \frac{\mu + \mu_t}{\mu_r} \frac{1}{\text{Re}} \left([\nabla \mathbf{u} + (\nabla \mathbf{u})^T] - \frac{2}{3} (\nabla \cdot \mathbf{u}) \mathbf{I} \right)$$

$$k_e = \frac{\gamma}{\text{Re}} \left(\frac{\mu}{\mu_r P_r} + \frac{\mu_t}{\mu_r P_{rt}} \right)$$

P_r et P_{rt} sont respectivement le nombre de Prandtl laminaire et turbulent.

$$\text{Re} = \frac{\rho_r \mathbf{u}_r \mathbf{L}_r}{\mu_r}$$

Le système d'équations (4.6) présente plus d'inconnues que d'équations. Pour le résoudre, nous avons besoin de deux équations supplémentaires. Le système peut être fermé en utilisant

les équations d'états (gaz parfait) suivantes :

$$p = (\gamma - 1)\rho T \quad (4.7)$$

et

$$T = \frac{E}{\rho} - \frac{|\mathbf{U}|^2}{2\rho^2} \quad (4.8)$$

Le système ainsi formé peut être réduit à la forme vectorielle suivante :

$$\mathbf{V}_{,t} + \mathbf{F}_{i,i}^{adv} - \mathbf{w}_i \mathbf{V}_{,i} = \mathbf{F}_{i,i}^{diff} + \mathbf{F}^s \quad (4.9)$$

où

– \mathbf{V} représente le vecteur des variables conservatives ρ , $\rho \mathbf{u}$ et ρe :

$$\mathbf{V} = \begin{pmatrix} \rho \\ \mathbf{U} \\ E \end{pmatrix}$$

– \mathbf{F}_i^{adv} , le vecteur flux de convection,

$$\mathbf{F}_i^{adv} = \begin{pmatrix} \mathbf{U}_i \\ \mathbf{U} \frac{\mathbf{U}_i}{\rho} \\ (E + p) \frac{\mathbf{U}_i}{\rho} \end{pmatrix}$$

- \mathbf{F}_i^{diff} , le vecteur flux de diffusion,

$$\mathbf{F}_i^{diff} = \begin{pmatrix} 0 \\ \sigma_i^* \\ \sigma_i^* \cdot \mathbf{u} - k_e T_{,i} \end{pmatrix}$$

$$\sigma_{i*} = (\sigma_{i1}, \sigma_{i2}, \sigma_{i3})^T$$

- \mathbf{F}^s , le vecteur source.

$$\mathbf{F}^s = \begin{pmatrix} 0 \\ \rho \mathbf{f}_v \\ \rho \mathbf{f}_v \cdot \mathbf{u} \end{pmatrix}$$

4.2 Discrétisation spatiale et temporelle

Pour résoudre numériquement le système (4.9), il est nécessaire de le transformer en un système algébrique. Une discrétisation spatio-temporelle est alors utilisée.

Dans la suite de cette section, la formulation variationnelle de Galerkin des équations de Navier-Stokes (4.9) est discrétisée en éléments finis. Les détails de discrétisation, les techniques de capture de choc ainsi que le traitement des instabilités numériques sont présentés.

En multipliant l'équation (4.9) par une fonction de pondération \mathbf{W} et en intégrant sur le domaine Ω , on obtient la forme variationnelle suivante :

$$\int_{\Omega} \mathbf{W} \cdot (\mathbf{V}_{,t} + \mathbf{F}_{i,i}^{adv} - \mathbf{w}_i \mathbf{V}_{,i} - \mathbf{F}_{i,i}^{diff} - \mathbf{F}^s) d\Omega = 0$$

Le terme de diffusion peut être transformé en utilisant le théorème de Gauss. La forme variationnelle des équations de Navier-Stokes devient :

$$\begin{aligned} & \int_{\Omega} \mathbf{W} \cdot (\mathbf{V}_{,t} + \mathbf{F}_{i,t}^{adv} - \mathbf{w}_i \mathbf{V}_{,i} - \mathbf{F}^s) d\Omega \\ & + \sum_c \int_{\Omega_c} \mathbf{W}_{,i} \cdot \mathbf{F}_i^{diff} d\Omega - \int_{\Gamma} \mathbf{W} \cdot \mathbf{F}_i^{diff} \mathbf{n}_i d\Gamma = 0 \end{aligned} \quad (4.10)$$

\mathbf{n} est le vecteur unitaire normal à Γ_{∞} et dirigé vers l'extérieur. Le domaine de calcul $\Omega \subset \mathbb{R}^3$ est subdivisé en plusieurs sous-domaines Ω_e appelés éléments finis.

4.2.1 Conditions aux limites à l'infini

Soit le Jacobien du flux \mathbf{F}_i :

$$\mathbf{A}_i = \frac{\partial \mathbf{F}_i}{\partial \mathbf{V}}$$

$$\mathbf{A}_n = \sum_{i=1}^3 \mathbf{A}_i n_i$$

La décomposition spectrale de \mathbf{A}_n s'écrit comme suit :

$$\mathbf{A}_n = \mathbf{S} \Lambda \mathbf{S}^{-1}$$

où \mathbf{S} et Λ sont respectivement les matrices des vecteurs propres et des valeurs propres λ_i de \mathbf{A} .

Λ est diagonale alors on définit :

$$|\mathbf{A}_n| = \mathbf{S} |\Lambda| \mathbf{S}^{-1}$$

Soit $|\mathbf{A}_n^-|$ tel que :

$$|\mathbf{A}_n^-| = \mathbf{S} |\Lambda^-| \mathbf{S}^{-1}$$

avec $\lambda_i^- = \min(0, \lambda_i)$ les éléments de Λ^- .

Une manière d'imposer les conditions aux limites à l'infini est de remplacer l'équation 4.10 par :

$$\begin{aligned} & \int_{\Omega} \mathbf{W} \cdot (\mathbf{V}_{,t} + \mathbf{F}_{i,i}^{adv} - \mathbf{w}_i \mathbf{V}_{,i} - \mathbf{F}^s) d\Omega \\ & + \sum_e \int_{\Omega_e} \mathbf{W}_{,i} \cdot \mathbf{F}_i^{diff} d\Omega - \int_{\Gamma} \mathbf{W} \cdot \mathbf{F}_i^{diff} \mathbf{n}_i d\Gamma \\ & + \int_{\Gamma_{\infty}} \mathbf{W} \cdot |\mathbf{A}_n^-| (\mathbf{V} - \mathbf{V}_{\infty}) d\Gamma = 0 \end{aligned} \quad (4.11)$$

Le dernier terme représente un terme de pénalité qui est introduit pour forcer la condition $\mathbf{V}_i = \mathbf{V}_{\infty_i}$ lorsque la i^{ieme} caractéristique est entrante.

4.2.2 Discrétisation temporelle

Les dérivées temporelles sont approchées par un schéma implicite de différences finies d'Euler de premier ordre ou de second ordre (schéma de Gear) suivant :

$$\frac{d\mathbf{V}(t)}{dt} \simeq \frac{3\mathbf{V}(t + \Delta t) - 4\mathbf{V}(t) + \mathbf{V}(t - \Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^2) \quad (4.12)$$

Au premier pas de temps, la valeur de $\mathbf{V}(t - \Delta t)$ est inconnue. On peut alors utiliser un schéma implicite d'Euler de premier ordre. Autrement, on peut poser $\mathbf{V}(t - \Delta t) = \mathbf{V}(t)$ ce qui revient aussi à utiliser un schéma implicite de premier ordre avec un pas de temps de $\frac{2}{3}\Delta t$.

4.2.3 Discrétisation spatiale

Le domaine fluide est discrétisé (en espace géométrique) en éléments finis en utilisant des éléments tétraédriques linéaires. Ce choix est motivé par la simplicité des éléments en question, la complexité des géométries des tests envisagés et surtout l'utilisation d'un maillage non-structuré dans tout le domaine physique et ceci jusqu'à la paroi solide.

La valeur de $\int_{\Omega} f d\Omega$ sur tout le domaine est égale à la somme des valeurs de $\int_{\Omega_e} f d\Omega_e$ sur les éléments.

$$\int_{\Omega} f(x, y, z) d\Omega = \sum_e \int_{\Omega_e} f(x, y, z) d\Omega_e$$

Les éléments qui forment le maillage (domaine géométrique subdivisé en éléments) sont de tailles et de formes différentes. En pratique, il est plus commode de transformer le domaine Ω_e en un domaine (élément) de référence Ω_r . Le vecteur des variables conservatives \mathbf{U} est approché dans l'élément de référence comme suit :

$$\mathbf{V}(i) = \sum_{k=1}^4 N_k(\xi, \eta, \zeta) \mathbf{V}_k(i) = \langle N_1 N_2 N_3 N_4 \rangle \cdot \begin{Bmatrix} \mathbf{V}(i)_1 \\ \mathbf{V}(i)_2 \\ \mathbf{V}(i)_3 \\ \mathbf{V}(i)_4 \end{Bmatrix} \quad (4.13)$$

où $i = (\rho), (\rho \mathbf{u})$ ou (ρe) et N_k sont les fonctions d'interpolation de l'élément de référence :

$$N_1(\xi, \eta, \zeta) = 1 - \xi - \eta - \zeta$$

$$N_2(\xi, \eta, \zeta) = \xi$$

$$N_3(\xi, \eta, \zeta) = \eta$$

$$N_1(\xi, \eta, \zeta) = \zeta$$

Selon la formulation variationnelle de type Galerkin, les fonctions de pondération sont calculées en utilisant les mêmes fonctions d'interpolation N_k .

La dérivée d'une fonction $f(x, y, z)$ par rapport à ξ_i (ξ , η ou ζ) dans l'élément de référence (repère local) est :

$$\frac{\partial f(x, y, z)}{\partial \xi_i} = \sum_{k=1}^4 \frac{\partial N_k}{\partial \xi_i} f_k \quad (4.14)$$

Le calcul de l'intégrale d'une fonction $f(x, y, z)$ dans l'élément de référence s'écrit :

$$\int_{\Omega_e} f(x, y, z) d\Omega_e = \int_{\Omega_r} f(\xi, \eta, \zeta) \det(J) d\Omega_r \quad (4.15)$$

où $\det(J)$ est le déterminant de la matrice Jacobienne suivante :

$$J = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \quad (4.16)$$

Chaque terme de cette matrice peut être calculé en utilisant l'équation (4.14).

4.2.3.1 Méthode de stabilisation SUPG standard

L'utilisation d'une formulation variationnelle faible de type Galerkin engendre des instabilités numériques, surtout quand le flux est dominé par le terme de convection. L'application d'un maillage tétraédrique jusqu'à la frontière solide et le raffinement dans la direction orthogonale à ce dernier (essentiel pour le module de turbulence et le couplage) produisent des éléments

très distordus au niveau de la couche limite (éléments très aplatis). Cette distorsion engendre des difficultés numériques dans la résolution du système établi. Une méthode de stabilisation est alors utilisée pour surmonter ces défis numériques. En plus, en présence de fort gradient comme dans les chocs, des oscillations parasites apparaissent.

La méthode de stabilisation utilisée est une méthode de décentrage amont basée sur une formulation de Petrov-Galerkin (SUPG : Streamline Upwinding Petrov-Galerkin). Cette méthode est devenue très populaire en éléments finis pour la résolution des équations de transport. Elle introduit un terme de perturbation supplémentaire à la formulation standard de Galerkin. Ce terme ajoute de la diffusion artificielle dans le sens de l'écoulement (Mallet, 1985; Brooks et Hughes, 1994; Soulaïmani et al., 1994; Soulaïmani et al., 2001b). La méthode ajoute à la formulation variationnelle (4.9) le terme de stabilisation suivant :

$$\sum_e \int_{\Omega_e} \mathbf{A}_i^t \cdot \mathbf{W}_{,i} \boldsymbol{\tau} \mathbf{R}(\mathbf{V}) d\Omega \quad (4.17)$$

où $\mathbf{R}(\mathbf{V}) = \mathbf{V}_{,t} + \mathbf{F}_{i,i}^{adv} - \mathbf{W}_i \mathbf{V}_{,i} - \mathbf{F}_{i,i}^{diff} - \mathbf{F}_s$ est le vecteur résidu local.

La matrice $\boldsymbol{\tau}$ est une quantité locale qui a la dimension du temps et qui dépend de la géométrie de l'élément. Le choix de cette matrice est crucial (Mallet, 1985; Shakib, 1989). Soulaïmani et al (Soulaïmani et Fortin, 1994; Soulaïmani et al., 1994) proposent une définition en plusieurs dimensions pour la mise en œuvre de cette matrice :

$$\boldsymbol{\tau} = \left(\sum_i \|c_{ij} \mathbf{A}_j\| \right)^{-1} \varsigma(\mathbf{R}_h) \quad (4.18)$$

où $c_{ij} = \frac{\partial \xi_j}{\partial x_i}$ sont les éléments de la matrice de transformation géométrique de l'élément réel à l'élément de référence. Les auteurs définissent $\varsigma(\mathbf{R}_h)$ comme suit :

$$\varsigma(\mathbf{R}_h) = \min \left(\frac{\mathbf{R}_h}{3}, 1 \right) \quad (4.19)$$

où $R_h = \frac{\|\mathbf{u}\| h}{2\mu}$ est le nombre de Reynolds local, μ la viscosité dynamique et h une taille caractéristique de l'élément. Une définition possible de h est $h = Volume^{\frac{1}{3}}$ (El Kadri El Yamani, 1992; Rebaine, 1997; Feng, 2005)

4.2.3.2 Méthode de capture de choc standard

Un autre terme a été ajouté pour permettre de capturer les oscillations dans le cas de forts gradients. En effet, l'ajout d'une dissipation numérique dans les zones de discontinuité permet de renforcer la stabilité du schéma numérique. Plusieurs méthodes sont proposées dans la littérature, en particulier en différences et volumes finis (Fortin, 1991). En éléments finis, d'autres études ont été proposées (Mallet, 1985; Löner et al., 1995; Shakib, 1989). L'introduction d'une viscosité artificielle, qui est fonction du résidu, permet de concentrer cette dernière dans les zones de choc (Soulaïmani et Fortin, 1994) ou d'oscillations parasites.

La formulation variationnelle 4.11 devient :

$$\begin{aligned}
 & \int_{\Omega} \mathbf{W} \cdot (\mathbf{V}_{,t} + \mathbf{F}_{t,t}^{adv} - \mathbf{w}_t \mathbf{V}_{,t} - \mathbf{F}^s) d\Omega \\
 & + \sum_e \int_{\Omega_e} \mathbf{W}_{,i} \cdot \mathbf{F}_i^{diff} d\Omega - \int_{\Gamma} \mathbf{W} \cdot \mathbf{F}_i^{diff} \mathbf{n}_i d\Gamma \\
 & + \int_{\Gamma_{\infty}} \mathbf{W} \cdot |\mathbf{A}_n^-| (\mathbf{V} - \mathbf{V}_{\infty}) d\Gamma + \sum_e \int_{\Omega_e} \mu_c \nabla \mathbf{W} \cdot \nabla \mathbf{V} d\Omega = 0
 \end{aligned} \tag{4.20}$$

Soulaïmani et al (Soulaïmani et Fortin, 1994) proposent la définition suivante de l'équation de viscosité artificielle de capture de choc.

$$\mu_c = \frac{C_k h \text{Min}(\|\boldsymbol{\tau} \mathbf{R}(\mathbf{V})\|, \|\mathbf{u}\|)}{2} \tag{4.21}$$

où $\|\cdot\|$ est la norme euclidienne et C'_k est un paramètre arbitraire à ajuster.

4.2.3.3 Nouvelles définitions de la matrice τ et l'opérateur de capture de choc

Les définitions de la matrice τ et de l'opérateur de choc présentées aux sections 4.2.3.1 et 4.2.3.2 ont été utilisées avec succès pour la simulation des écoulements fluides compressibles non visqueux (les équations d'Euler), laminaires (El Kadri El Yamani, 1992; Soulaïmani et al., 1994) et turbulents avec la loi de paroi (Rebaine et Soulaïmani, 2001). Cependant, dans le cas des simulations des écoulements turbulents par la résolution des équations de Navier-Stokes moyennées, discrétisées sur un maillage purement tétraédrique jusqu'à la paroi solide, ces définitions de la matrice τ et de l'opérateur de capture de choc ne sont plus efficaces. Des simulations de l'écoulement sur l'Onera M6 ont montré que les opérateurs ainsi définis sont incapables de bien estimer la quantité de viscosité artificielle nécessaire pour la stabilisation de la solution. De plus, des problèmes de convergence de la résolution du système non linéaire sont apparus. Ceci est expliqué par le fait que la méthode, basée sur un schéma de décentrage amont de premier ordre, produit une viscosité artificielle de l'ordre de $\frac{\|\mathbf{u}\| h}{2}$.

Cette production de viscosité implique une grande diffusion numérique, en particulier, dans la couche limite. Il n'est pas aisé de produire juste assez de viscosité pour capturer le choc sans trop de diffusion. La définition de la taille d'élément " h " devient encore plus délicate dans le cas des éléments très distordus. Il est alors important de proposer de nouvelles définitions de la matrice τ et de l'opérateur de capture de choc pour mieux contrôler la viscosité artificielle et empêcher ainsi la diffusion de la solution, notamment, dans la couche limite et les zones à fort gradient de pression. Ces nouvelles définitions doivent limiter l'ordre de grandeur de la viscosité artificielle à une valeur inférieure à $\frac{\|\mathbf{u}\| h}{2}$ dans les zones de chocs ou d'oscillations et proche de zéro ailleurs. L'introduction d'un senseur de "variation de pression" dans la formule de la viscosité de capture de choc s'impose. Aussi, il est important de redéfinir la taille caractéristique de l'élément pour l'extension de la méthode aux maillages fortement anisotropiques.

Dans ce qui suit, nous présentons de nouvelles définitions de la matrice τ et de l'opérateur de capture de choc. Un senseur de variation de pression basé sur une adaptation aux éléments finis du schéma de Jameson (Jameson et Mavriplis, 1986; Peterson et Zing, 2008) est appliqué.

La matrice τ est remplacée par une matrice diagonale dont les termes diagonaux sont calculés à partir de la plus grande valeur propre des matrices de convection $\lambda = \|\mathbf{u}\| + c$ et la taille caractéristique $h = h_{\min}^e$ définies comme suit :

$$\tau = \frac{h}{2\lambda} \mathbf{I} \quad (4.22)$$

où \mathbf{I} est la matrice identité, $h = h_{\min}^e = \min(x_i - x_j)$ i et j sont les nœuds de l'élément. Dans l'élément proche de la paroi h est de l'ordre de $y^+ = 1$. En pratique, le coût de calcul de la matrice τ est négligeable comparé à l'ancienne définition qui représentait 30% du temps total dans le calcul du résidu.

Nous définissons une matrice $[\mu_c]$ diagonale de viscosité de capture de choc qui est formulée en fonction du senseur de variation de pression ς^e comme suit :

$$\mu_{c,k}^e = \frac{h\lambda_k}{2} C'_k (\varsigma^e + \varepsilon_k) \quad (4.23)$$

Avec $\lambda_k = \|\mathbf{u}\| + c$ pour l'équation de continuité ($k = 1$), $\lambda_k = \|\mathbf{u}\|$ pour les trois équations de quantité de mouvement ($k = 2, 3, 4$), $\lambda_k = 0$ pour l'équation d'énergie ($k = 5$). C_k et ε_0 sont des coefficients d'ajustement. Les valeurs $C'_1 = \frac{1}{2}$, $C'_k = 1.0$ pour ($k = 2, 3, 4$) et $\varepsilon_k = 0.01$ sont des valeurs recommandées.

Le terme ς^e est le senseur de variation de pression inspiré de l'opérateur de capture de choc de Jameson (Jameson et Mavriplis, 1986). Pour un nœud i appartenant aux éléments ($k = 1, N_{ei}$) et ($j = 1, 3$) les autres nœuds de l'élément k , le senseur de capture de choc est calculé comme

suit :

$$\varsigma^i = \frac{1}{N_{ei}} \sum_{k=1}^{N_{ei}} \left(\sum_{j=1}^3 \frac{|p_i - p_j|}{\frac{p_i + p_j}{2}} \right) \quad (4.24)$$

Pour un élément donné :

$$\varsigma^e = \frac{1}{4} \sum_{i=1}^4 \varsigma^i \quad (4.25)$$

Pour mettre en évidence la pertinence des nouvelles définitions de la matrice τ et de l'opérateur de capture de choc, deux figures illustrant le senseur de variation de pression ς^i sont présentées.

La figure 4.1 superpose les lignes de contours de pression et le senseur de variation de pression ς^i dans le cas d'une solution non convergée (figure à gauche). On remarque une augmentation de la valeur de ς^i dans les zones à haute variation de pression (lignes serrées). Ces zones sont les niches d'oscillations et d'instabilités numériques qui posaient des problèmes de convergence avant la mise en place des nouvelles définitions.

Au cours du processus de résolution (figure à droite), les oscillations non-physiques sont disparues. La viscosité artificielle ajoutée a permis de capturer le choc sans trop de diffusion. Il est évident que les nouvelles définitions ont amélioré la méthode de stabilisation et de capture de choc. Les résultats de cette simulation sont présentés en détails dans le chapitre des résultats.

Remarque : La légende de la figure 4.1 affiche les valeurs de la pression (lignes de contours) et non celles du senseur de variation de pression ς^i . À titre indicatif, les valeurs du senseur sont présentées sur la figure 4.2 dans le cas de la solution complètement convergée.

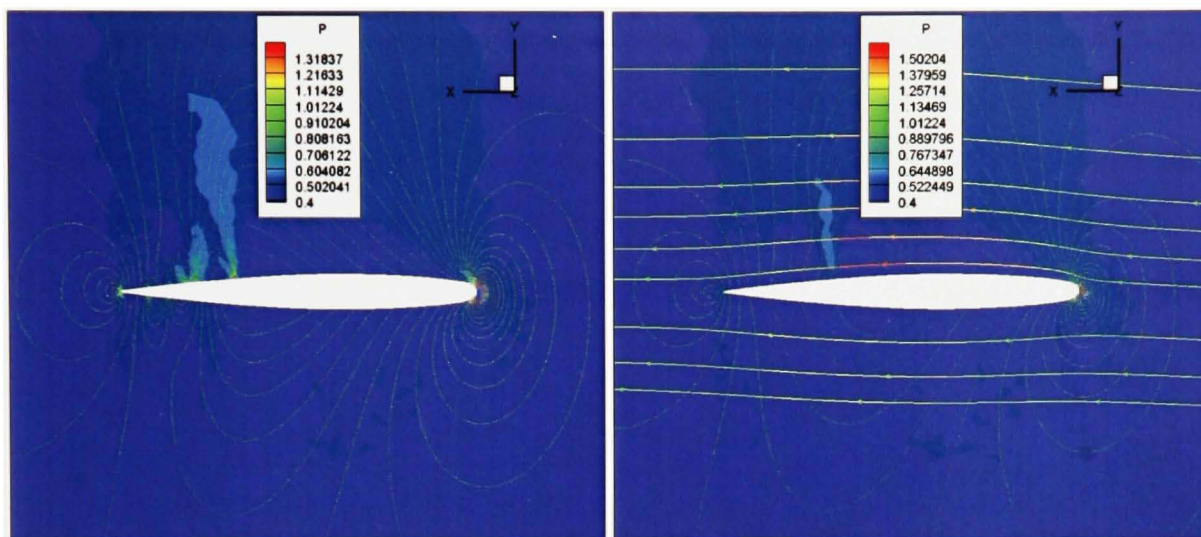


Figure 4.1 *Détection des zones d'oscillations et d'instabilités.*

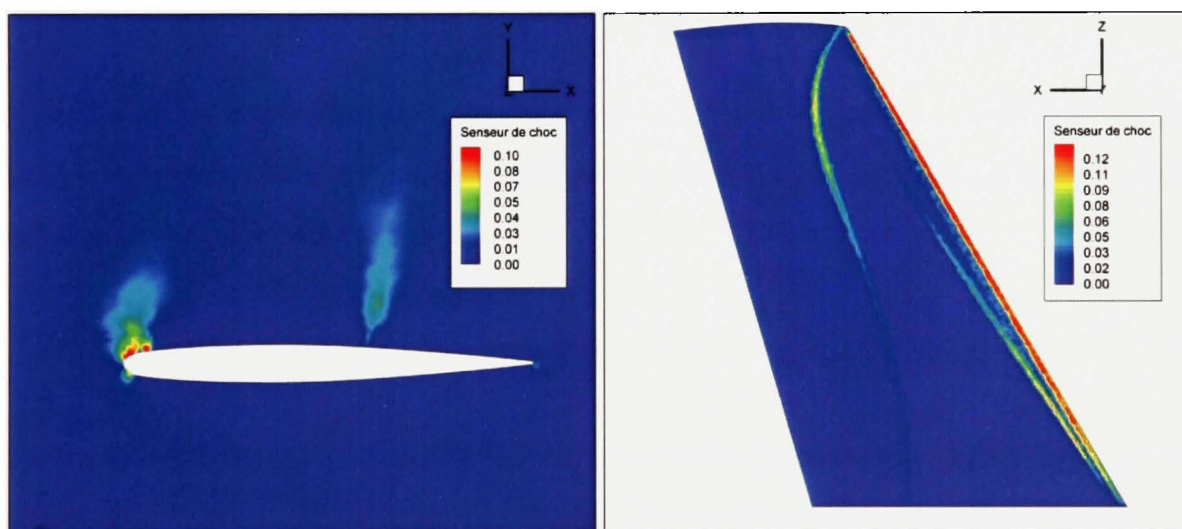


Figure 4.2 *Senseur de variation de pression dans le cas d'une solution convergée.*

4.3 Forme variationnelle stabilisée des équations de Navier-Stokes

Tout au long de ce chapitre, le module de Navier-Stokes a été discuté. Les différentes formes des équations de Navier-Stokes ont été présentées. La forme conservative écrite avec les va-

riables conservatives a été adoptée. Cette dernière a été présentée dans le cas d'un repère mobile utilisant une formulation ALE (eulerienne-lagrangienne). Ensuite, la discrétisation et la stabilisation du modèle numérique ont été discutées. Après ajout de tous les termes annoncés, la formulation variationnelle des équations de Navier-Stokes finale est la suivante :

$$\begin{aligned}
 & \sum_e \int_{\Omega_e} \left[\mathbf{W} \cdot (\mathbf{V}_{,t} + \mathbf{F}_{i,i}^{adv} - \mathbf{w}_i \mathbf{V}_{,i} - \mathbf{F}_s) + \mathbf{W}_{,i} \cdot \mathbf{F}_i^{diff} \right] d\Omega \\
 & - \sum_e \int_{\Omega_e} \mathbf{W} \cdot \mathbf{F}_i^{diff} \mathbf{n}_i d\Gamma - \int_{\Gamma_\infty} \mathbf{W} \cdot \mathbf{A}_n^- (\mathbf{V} - \mathbf{V}_\infty) d\Gamma \\
 & + \sum_e \int_{\Omega_e} \nabla \mathbf{W} \cdot [\mu_c] \nabla \mathbf{V} d\Omega + \sum_e \int_{\Omega_e} (\mathbf{A}_i - \mathbf{w}_i I)^t \mathbf{W}_{,i} \cdot \tau \mathbf{R}(\mathbf{V}) d\Omega = 0
 \end{aligned} \tag{4.26}$$

La résolution de ce modèle, dans le cas d'un fluide non-visqueux compressible, est alors possible en annulant le terme de diffusion et en ajoutant la condition de glissement suivante sur les nœuds de l'interface solide :

$$\mathbf{u} \cdot \mathbf{n} = \mathbf{w} \cdot \mathbf{n} \tag{4.27}$$

où \mathbf{n} est la normale à la surface mobile au nœud considéré et \mathbf{w} la vitesse du nœud sur la frontière mobile (Soulaïmani et Fortin, 1994) calculée avec le schéma de Gear comme suit :

$$\mathbf{w} = \frac{3\mathbf{x}|_{t^{n+1}} - 4\mathbf{x}|_{t^n} + \mathbf{x}|_{t^{n-1}}}{2\Delta t} \tag{4.28}$$

Une fois les transformations géométriques, les dérivés et les intégrations numériques appliquées, les matrices et les vecteurs élémentaires peuvent être assemblés dans un système global. Le problème peut être alors résolu en utilisant les méthodes décrites dans le chapitre 3.

CHAPITRE 5

RÉSOLUTION DE L'ÉQUATION DE TURBULENCE

Le but est de réaliser un module de calcul de la viscosité turbulente facilement intégrable dans les logiciels de simulation des écoulements turbulents. Le module en question, appelé "module de turbulence", doit préserver la positivité de la viscosité turbulente. En outre, il doit offrir une stabilité numérique satisfaisante même en utilisant un maillage tétraédrique fortement distordu dans la couche limite et jusqu'à la paroi solide.

Le module de turbulence est proposé sous forme de deux approches. Une première basée sur une discrétisation simultanée en éléments finis ou en volumes finis des différents termes de l'équation de transport de Spalart et Allmaras (Spalart et Allmaras, 1994). La deuxième approche repose sur une discrétisation purement éléments finis. Pour cette dernière, une approximation semi-implicite du terme source a été introduite pour traiter la raideur du terme source. Cette façon de faire a été inspirée des résultats obtenus avec la première approche.

Rappelons que la décomposition de l'équation et la discrétisation de chacun de ses termes individuellement n'est pas une nouvelle idée en elle même (Ghidaglia et Pascal, 1999; Pascal et Ghidaglia, 2001). Néanmoins, le schéma proposé en est une. Tout d'abord, l'algorithme de décomposition standard a été modifié afin de surmonter la raideur du terme source. Ensuite, la discrétisation de chaque terme de l'équation en éléments finis ou en volumes finis a permis de tester plusieurs variantes et, ainsi, optimiser le programme général en termes de stabilité, d'efficacité et de facilité de mise en œuvre informatique. Enfin, l'intégration de la décomposition de l'équation de Spalart-Allmaras dans un code de calcul parallèle basé sur une décomposition géométrique et fonctionnelle est une première. Cette approche a été l'objet de publications et communications (Lorin et al., 2007; Lorin et al., 2006; Ben Haj Ali et al., 2004)

Dans ce chapitre le modèle de turbulence de Spalart-Allmaras est rappelé. Nous présentons plusieurs méthodes d'implémentation du modèle. Toutes les variantes présentées assurent la

positivité de la viscosité turbulente. Cependant, pour des raisons de simplicité et de performance nous adoptons la dernière variante 5.4.2. Les autres variantes sont exposées pour montrer le raisonnement qui nous a conduit à ce choix.

5.1 Modèle standard de Spalart-Allmaras

L'équation de la viscosité turbulente s'écrit comme suit (Spalart et Allmaras, 1994) :

$$\frac{D\tilde{\nu}}{Dt} - \frac{1}{\sigma} [\nabla \cdot ((\nu + \tilde{\nu}) \nabla \tilde{\nu}) + c_{b2} (\nabla \tilde{\nu})^2] - c_{b1} \tilde{\omega} \tilde{\nu} + c_{w1} f_w(\tilde{\nu}) \left[\frac{\tilde{\nu}}{d} \right]^2 = 0 \quad (5.1)$$

- $\tilde{\nu} = \frac{\nu_t}{f_{v1}}$ où $f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}$ et $\chi = \frac{\tilde{\nu}}{\nu}$,
- d est la distance à la paroi solide,
- ν est la viscosité moléculaire,
- $c_{b1} \tilde{\omega} \tilde{\nu}$ est le terme de production relié à la vorticit  du fluide,
- $\tilde{\omega} = \omega + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}$ où ω est le module de la vorticit  du fluide et $f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}(\chi)}$,
- $\frac{1}{\sigma} [\nabla \cdot ((\nu + \tilde{\nu}) \nabla \tilde{\nu}) + c_{b2} (\nabla \tilde{\nu})^2]$ est le terme de diffusion,
- $c_{w1} f_w(\tilde{\nu}) \left[\frac{\tilde{\nu}}{d} \right]^2$ est le terme de destruction Avec $f_w(g) = g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right)^{1/6}$,
 $g(r(\tilde{\nu})) = r(\tilde{\nu}) + c_{w2}(r(\tilde{\nu})^6 - r(\tilde{\nu}))$ et $r(\tilde{\nu}) = \frac{\tilde{\nu}}{\omega \kappa^2 d^2}$,
- $c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}$,
- $c_{b1} = 0.1355, c_{b2} = 0.622, \sigma = 2/3, c_{w2} = 0.3, c_{w3} = 2$ et $c_{v1} = 7.1$.

Remarque : Spalart a aussi donn  une autre d finition de $\tilde{\omega}$:

$$\tilde{\omega} = \omega \tilde{f}_{v3} + \frac{\tilde{\nu}}{\kappa^2 d^2} \tilde{f}_{v2}$$

avec

$$\tilde{f}_{v2} = \left(1 + \frac{\chi}{c_{v2}}\right)^{-3}, \tilde{f}_{v3} = \left(1 + \chi f_{v1}\right) \left(\frac{1 - \tilde{f}_{v2}}{\chi}\right) \text{ et } c_{v2} = 5$$

5.2 Principe de la méthode numérique

Cette section traite la modification appliquée au modèle standard de Spalart et Allmaras, la décomposition (splitting) des opérateurs différentiels, le traitement du terme source et l'adaptation du schéma numérique au code PFES360.

5.2.1 Forme adimensionnelle de l'équation de Spalart et Allmaras

L'équation de la viscosité turbulente s'écrit :

$$\frac{D\tilde{\nu}}{Dt} - \frac{1}{\sigma} [\nabla \cdot ((\nu + \tilde{\nu}) \nabla \tilde{\nu}) + c_{b2} (\nabla \tilde{\nu})^2] - c_{b1} \tilde{\omega} \tilde{\nu} + c_{w1} f_w(\tilde{\nu}) \left[\frac{\tilde{\nu}}{d} \right]^2 = 0 \quad (5.2)$$

Une forme adimensionnelle du système (5.2) peut être trouvée en utilisant les grandeurs de référence (indice r) suivantes :

$$\begin{aligned} \mathbf{u}^* &= \frac{\mathbf{u}}{\mathbf{u}_r} & \chi &= \frac{\tilde{\nu}}{\nu} \\ x^* &= \frac{x}{L_r} & t^* &= \frac{t}{L_r / |\mathbf{u}_r|} & \text{Re} &= \frac{\mathbf{u}_r L_r}{\nu} \end{aligned}$$

(Pour alléger l'écriture, on confond désormais la notation des variables réduites (\star) avec les variables adimensionnelles).

$$\frac{D\chi}{Dt} - \frac{1}{\text{Re} \sigma} (\nabla \cdot ((1 + \chi) \nabla \chi) + c_{b2} (\nabla \chi)^2) - c_{b1} \tilde{\omega} \chi + \frac{c_{w1}}{\text{Re}} f_w(\chi) \left[\frac{\chi}{d} \right]^2 = 0 \quad (5.3)$$

avec :

- $\frac{D\chi}{Dt} = \frac{\partial \chi}{\partial t} + \mathbf{u} \nabla \chi$;
- $\tilde{\omega} = \omega + \frac{1}{\text{Re}} \frac{\chi}{\kappa^2 d^2} f_{v2}$ ou $\tilde{\omega} = \omega \tilde{f}_{v3} + \frac{\chi}{\text{Re} \kappa^2 d^2} \tilde{f}_{v2}$;
- $f_w(\chi) = g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right)^{1/6}$;
- $g(r) = r + c_{w2}(r^6 - r)$;

- $r(\chi) = \frac{\chi}{\text{Re } \omega \kappa^2 l^2}$

La solution de cette équation est régulière (Ben Haj Ali et al., 2004; Lorin et al., 2007; Lorin et al., 2006) (au moins \mathcal{C}^2). On peut alors utiliser l'égalité $\Delta \chi^2 = \nabla \cdot (\nabla \chi^2) = 2\chi \Delta \chi + 2|\nabla \chi|^2$ pour déduire que : $\frac{1}{2} \Delta (1 + \chi)^2 = (1 + \chi) \Delta \chi + |\nabla \chi|^2 = \nabla \cdot ((1 + \chi) \nabla \chi)$ et ainsi réécrire l'équation (5.3) sous la forme suivante :

$$\boxed{\frac{\partial \chi}{\partial t} = -\mathbf{u} \cdot \nabla \chi + \frac{1}{\text{Re } \sigma} \left(\frac{1}{2} \Delta (1 + \chi)^2 + c_{b2} (\nabla \chi)^2 \right) + c_{b1} \tilde{\omega} \chi - \frac{c_{w1}}{\text{Re}} f_w(\chi) \left[\frac{\chi}{d} \right]^2} \quad (5.4)$$

5.3 Approche volumes finis - éléments finis

La première étape de l'élaboration de ce schéma numérique est la décomposition de l'équation (5.4) en deux parties. Une première partie discrétisée en volumes finis pour calculer la convection et le terme source, et une deuxième partie discrétisée en éléments finis pour la résolution de la diffusion. Les méthodes d'éléments finis sont stables et précises pour les opérateurs elliptiques. Alors que les schémas de volumes finis sont bien adaptés aux opérateurs hyperboliques.

5.3.1 Un premier algorithme (Algorithme 6)

L'équation (5.4) du modèle standard est remplacée par le système d'équations suivant :

$$\left\{ \begin{array}{ll} \frac{\partial \chi}{\partial t} + \mathbf{u} \cdot \nabla \chi = \tilde{S}(\chi) & \text{discrétisé en volumes finis.} \\ \text{et} & \\ \frac{\partial \chi}{\partial t} - \frac{1}{\text{Re } \sigma} \left(\frac{1}{2} \Delta (1 + \chi)^2 + c_{b2} |\nabla \chi|^2 \right) = 0 & \text{discrétisé en éléments finis.} \end{array} \right.$$

où

$$\tilde{S}(\chi) = c_{b1} \tilde{\omega} \chi - \frac{c_{w1}}{\text{Re}} f_w(\chi) \left[\frac{\chi}{d} \right]^2.$$

Notons ici que la résolution de ce système donne une solution approchée de l'équation de Spalart-Allmaras. La démonstration est détaillée dans (Lorin et al., 2007; Lorin et al., 2006; Ben Haj Ali et al., 2004).

L'algorithme 6 et la figure 5.1 illustrent les différentes opérations réalisées par le module de turbulence durant un pas de temps.

Au début du pas de temps, le module de Navier-Stokes communique la vitesse \mathbf{u}^n au module de turbulence. Ensuite, partant d'une solution χ_e^n , la viscosité turbulente au pas de temps n (calculée par projection des valeurs nodales χ_i^n : passerelle EF \rightarrow VF), la résolution du système (5.5) par la méthode des volumes finis génère une solution intermédiaire $\chi_e^{n+\frac{1}{2}}$ à $t^{n+\frac{1}{2}}$.

$$\begin{cases} \frac{\partial \chi_e}{\partial t} + \mathbf{u} \cdot \nabla \chi_e = \tilde{S}(\chi_e) & t \in [t^n, t^{n+\frac{1}{2}}[\\ \chi_e(\cdot, t^n) = \chi_e^n \end{cases} \quad (5.5)$$

La solution $\chi_i^{n+\frac{1}{2}}$ est alors trouvée par projection de $\chi_e^{n+\frac{1}{2}}$ sur les nœuds du maillage (passerelles VF \rightarrow EF) puis injectée dans le système (5.6) (terme de diffusion) pour compléter le calcul de la solution globale χ_i^{n+1} .

$$\begin{cases} \frac{\partial \chi_i}{\partial t} - \frac{1}{\text{Re } \sigma} \left(\frac{1}{2} \Delta (1 + \chi_i)^2 + c_{b2} |\nabla \chi_i|^2 \right) = 0 & t \in [t^{n+\frac{1}{2}}, t^{n+1}[\\ \chi_i(\cdot, t^{n+\frac{1}{2}}) = \chi_i^{n+\frac{1}{2}} \end{cases} \quad (5.6)$$

Finalement, le module de turbulence envoie au module de Navier-Stokes la solution χ_i^{n+1} . Ce dernier calcule alors la nouvelle vitesse $\mathbf{u}^{n+1} = f(\mathbf{u}^{n+1}, \chi_i^{n+1}, \dots)$. Les deux modules passent alors au pas suivant et ainsi de suite.

Algorithme 6 : Turbulence : approche VF-EF, première variante

Données : χ_i^0 aux nœuds, N : Nombre de pas de temps.

Résultat : χ_i^{n+1} .

initialisation : $\chi_i = \chi_i^0$. **pour** $n = 1, N$ **faire**

 Calculer χ_e^n aux centres des éléments par projection de χ_i^n ;

 Recevoir les valeurs de la vitesse \mathbf{u} du module Navier-Stokes ;

En volumes finis ;

 Calculer $\chi_e^{n+\frac{1}{2}}$ utilisant le schéma suivant :

$$\begin{cases} \frac{\partial \chi_e}{\partial t} + \mathbf{u} \cdot \nabla \chi_e = \tilde{S}(\chi_e) & t \in [t^n, t^{n+\frac{1}{2}}[\\ \chi_e(\cdot, t^n) = \chi_e^n \end{cases}$$

 Calculer $\chi_i^{n+\frac{1}{2}}$ par projection de la solution $\chi_e^{n+\frac{1}{2}}$ sur les nœuds du maillage ;

En éléments finis ;

 Mettre à jour les conditions aux limites tenant compte des valeurs $\chi_i^{n+\frac{1}{2}}$;

$\chi_i = \chi_i^{n+\frac{1}{2}}$;

 Calculer χ_i^{n+1} par la résolution du système

$$\begin{cases} \frac{\partial \chi_i}{\partial t} - \frac{1}{\text{Re } \sigma} \left(\frac{1}{2} \Delta (1 + \chi_i)^2 + c_{b2} |\nabla \chi_i|^2 \right) = 0 & t \in [t^{n+\frac{1}{2}}, t^{n+1}[\\ \chi_i(\cdot, t^{n+\frac{1}{2}}) = \chi_i^{n+\frac{1}{2}} \end{cases}$$

 Envoyer les valeurs de χ_i^{n+1} au module Navier-Stokes ;

$\chi_i^n = \chi_i^{n+1}$.

fin

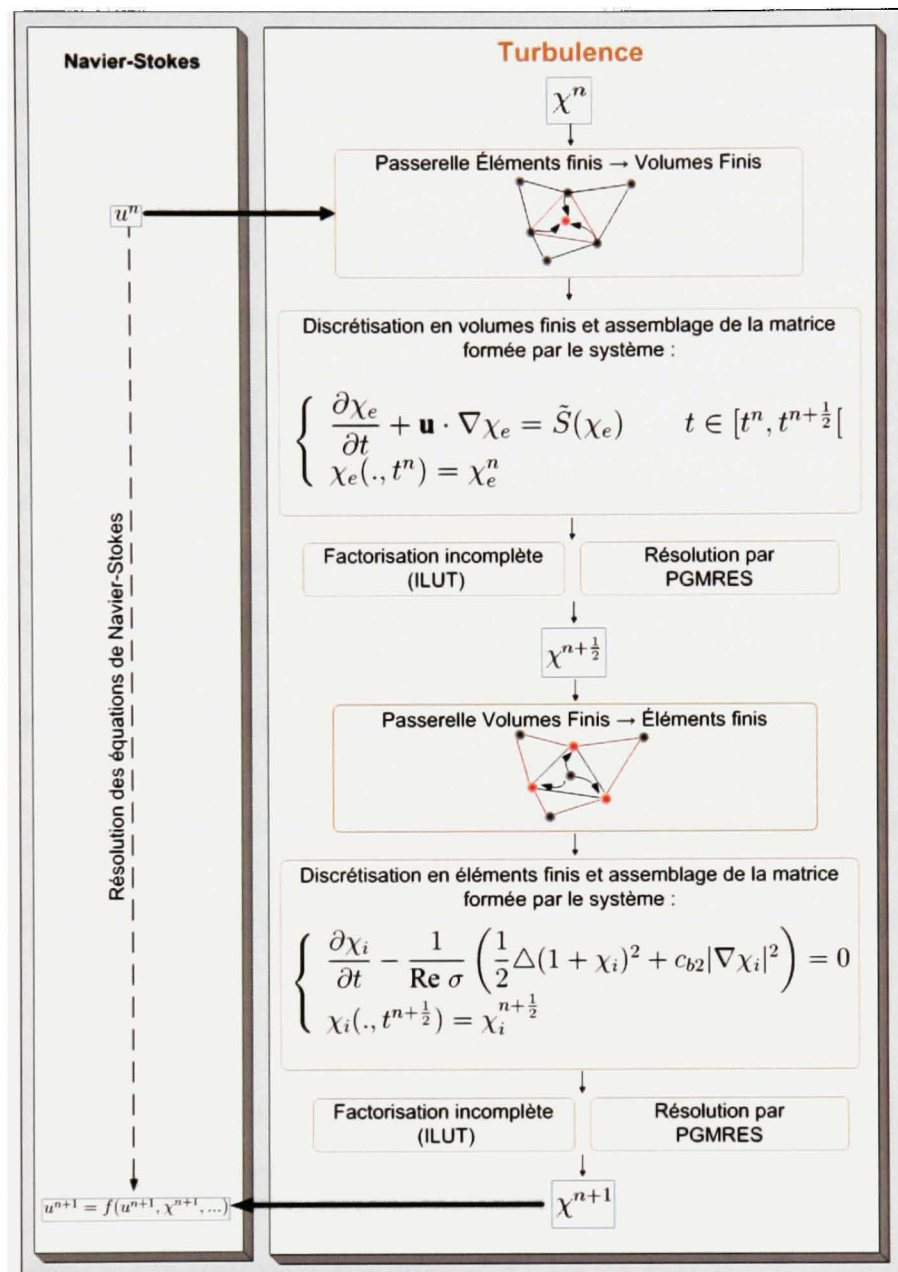


Figure 5.1 *Turbulence : organigramme de l'approche VF-EF, première variante.*

La démonstration mathématique de la décomposition, les différentes méthodes utilisées pour les passerelles et la positivité du schéma numérique sont détaillées dans (Lorin et al., 2007; Lorin et al., 2006; Ben Haj Ali et al., 2004).

Remarque : Il est très important de mettre à jour les conditions aux limites du problème lors de la résolution du terme de diffusion (Éléments finis) pour prendre en considération la solution $\chi^{n+\frac{1}{2}}$ des termes de convection et sources.

Remarque : Le schéma numérique discuté à la section précédente (section 5.3.1) consiste à discrétiser chaque terme de Spalart-Allmaras avec la méthode la plus adaptée. Malgré les avantages numériques offerts par ce schéma, la raideur du terme source pose encore un problème de stabilité. En effet, l'utilisation d'un maillage tétraédrique très fin et très distordu jusqu'à la paroi solide limite énormément la valeur du pas de temps. Cette limitation numérique est causée par la raideur du terme source, plus précisément le terme $\left(\frac{\chi}{d}\right)^2$, et par le fait qu'on a des schémas explicites.

5.3.2 Un deuxième algorithme (Algorithme 7)

L'idée est de séparer le terme source du terme de convection et de le calculer analytiquement.

Le système d'équations (5.5) relatif au terme de convection et au terme source :

$$\begin{cases} \frac{\partial \chi}{\partial t} + \mathbf{u} \cdot \nabla \chi = \tilde{S}(\chi) & t \in [t^n, t^{n+\frac{1}{2}}[\\ \chi(., t^n) = \chi^n \end{cases} \quad (5.7)$$

est alors remplacé par le système d'équations (5.8). En suppose que $\nabla \cdot \mathbf{u} = 0$.

$$\begin{cases} \begin{cases} \frac{\partial \chi}{\partial t} + \nabla(\chi \mathbf{u}) = 0, & [t^n, t^{n^*}[\\ \chi(., t^n) = \chi^n, \end{cases} \\ \begin{cases} \frac{\partial \chi}{\partial t} = \tilde{S}(\chi), & [t^{n^*}, t^{n+1/2}[\\ \chi(., t^{n^*}) = \chi^{n^*}. \end{cases} \end{cases} \quad (5.8)$$

Noter que \mathbf{u} est constant par élément.

Le terme de diffusion est traité de la même manière que l'algorithme précédent (Algorithme 6). Les démonstrations mathématiques sont présentées dans (Lorin et al., 2007; Lorin et al., 2006; Ben Haj Ali et al., 2004) et les étapes de calculs sont présentées dans l'algorithme 7.

Au début du pas de temps, le module de turbulence reçoit la vitesse \mathbf{u}^n . Utilisant la solution actuelle de la viscosité χ_e^n (projection de χ_i^n sur les éléments), une solution intermédiaire χ_e^{n*} est calculée grâce à la résolution du système (5.9) :

$$\begin{cases} \frac{\partial \chi_e}{\partial t} + \nabla (\chi_e \mathbf{u}) = 0, & [t^n, t^{n*}[\\ \chi_e(\cdot, t^n) = \chi_e^n \end{cases} \quad (5.9)$$

La solution χ_e^{n*} est alors injectée dans le système d'équation (5.10) pour calculer la solution $\chi_e^{n+\frac{1}{2}}$.

$$\begin{cases} \frac{\partial \chi_e}{\partial t} = \tilde{S}(\chi_e), & [t^{n*}, t^{n+1/2}[\\ \chi_e(\cdot, t^{n*}) = \chi_e^{n*}. \end{cases} \quad (5.10)$$

La solution $\chi_e^{n+\frac{1}{2}}$ est projetée ensuite sur les nœuds ($\chi_i^{n+\frac{1}{2}}$) du maillage (passerelles VF \rightarrow EF), puis injectée dans le système (5.6) (terme de diffusion) pour compléter le calcul de la solution globale χ_i^{n+1} .

$$\begin{cases} \frac{\partial \chi_i}{\partial t} - \frac{1}{\text{Re } \sigma} \left(\frac{1}{2} \Delta (1 + \chi_i)^2 + c_{b2} |\nabla \chi_i|^2 \right) = 0 & t \in [t^{n+\frac{1}{2}}, t^{n+1}[\\ \chi_i(\cdot, t^{n+\frac{1}{2}}) = \chi_i^{n+\frac{1}{2}} \end{cases} \quad (5.11)$$

Finalement, le module de turbulence envoie au module de Navier-Stokes la solution χ_i^{n+1} . Ce dernier calcule alors la nouvelle vitesse $\mathbf{u}^{n+1} = f(\mathbf{u}^{n+1}, \chi_i^{n+1}, \dots)$. Les deux modules passent au pas suivant et ainsi de suite.

Algorithme 7 : Turbulence : approche VF-EF, deuxième variante

Données : χ_i^0 aux nœuds, N : Nombre de pas de temps.

Résultat : χ_i^{n+1} .

initialisation : $\chi_i = \chi_i^0$. **pour** $n = 1, N$ **faire**

 Calculer χ_e^n aux centres des éléments par projection de χ_i^n ;

 Recevoir les valeurs de la vitesse du module Navier-Stokes ;

En volumes finis ;

 Calculer χ_e^{n*} par la résolution du système

$$\frac{\partial \chi_e}{\partial t} + \nabla (\chi_e \mathbf{u}) = 0 \text{ avec } \chi_e(\cdot, t^n) = \chi_{\text{chapitre}_N S e^n \text{ et } t \in [t^n, t^{n*}[;$$

 Calculer $\chi_e^{n+\frac{1}{2}}$ par la résolution du système

$$\frac{\partial \chi_e}{\partial t} = \tilde{S}(\chi_e) \text{ avec } \chi_e(\cdot, t^{n*}) = \chi_e^{n*} \text{ et } t \in [t^{n*}, t^{n+\frac{1}{2}}[;$$

 Calculer $\chi_i^{n+\frac{1}{2}}$ par projection de la solution $\chi_e^{n+\frac{1}{2}}$ sur les nœuds du maillage ;

En éléments finis ;

 Mettre à jour les conditions aux limites tenant compte des valeurs $\chi_i^{n+\frac{1}{2}}$;

$$\chi_i = \chi_i^{n+\frac{1}{2}} ;$$

 Calculer χ_i^{n+1} par la résolution du système

$$\begin{cases} \frac{\partial \chi_i}{\partial t} - \frac{1}{\text{Re } \sigma} \left(\frac{1}{2} \Delta (1 + \chi_i)^2 + c_{b2} |\nabla \chi_i|^2 \right) = 0 & t \in [t^{n+\frac{1}{2}}, t^{n+1}[\\ \chi_i(\cdot, t^{n+\frac{1}{2}}) = \chi_i^{n+\frac{1}{2}} \end{cases}$$

 Envoyer les valeurs de χ_i^{n+1} au module Navier-Stokes ;

$$\chi_i^n = \chi_i^{n+1}.$$

fin

Une série de simulations a été menée pour étudier la convergence de l'algorithme de calcul de la viscosité turbulente couplé avec le solveur Navier-Stokes. Le cas test est celui de la plaque plane pour un nombre de Reynolds de 10^6 . Le maillage utilisé est un maillage non structuré de 40 points suivant l'axe z et dont le premier point est à $Y^+ \simeq 5$. Le test consiste à varier le pas de temps de calcul de la turbulence et de calculer l'erreur par rapport à la solution donnée par le plus petit pas de temps ($\Delta t_{\min} = \frac{1}{128} \times 10^{-8}$). Ce dernier est considéré comme étant une solution exacte puisque la solution du modèle proposé est une solution asymptotique qui tend vers $\chi = \kappa y^+$. Le pas de temps du solveur fluide est fixé à 1×10^{-5} . Le solveur de la turbulence et celui du fluide s'échangent les valeurs de $\nu_t = \frac{f_{v1}}{\text{Re}} \chi$ et de \mathbf{u} chaque pas de temps fluide (voir chapitre 6).

Notons que pendant un pas du temps fluide, le solveur turbulent avance de plusieurs pas de temps pour arriver au point d'échange au même instant que le solveur Navier-Stokes. Cette opération est nécessaire puisqu'il y a une limitation du pas de temps de la turbulence pour assurer la stabilité et la positivité.

Les résultats de ces simulations sont publiés dans l'article (Lorin et al., 2007). Les auteurs se sont particulièrement concentrés sur la positivité de la solution et le traitement du terme source de l'équation de Spalart et Allmaras. Cependant, de point de vue performance, les schémas présentés à la section 5.3 présentent quelques limitations. En effet, le code de calcul PFES est initialement conçu sur la base d'une structure de données plus adaptée au calcul en éléments finis. L'intégration du module de turbulence, basé sur le schéma numérique présenté auparavant, a significativement alourdi le code en termes de temps de calcul et de mémoire utilisée. En effet, la résolution de la convection en volumes finis a un coût élevé puisqu'elle restreint fortement le pas de temps. Une éventuelle montée en ordre en volumes finis, déjà plus compliquée qu'en éléments finis, est devenue une opération assez laborieuse à cause de la nature de la structure de données.

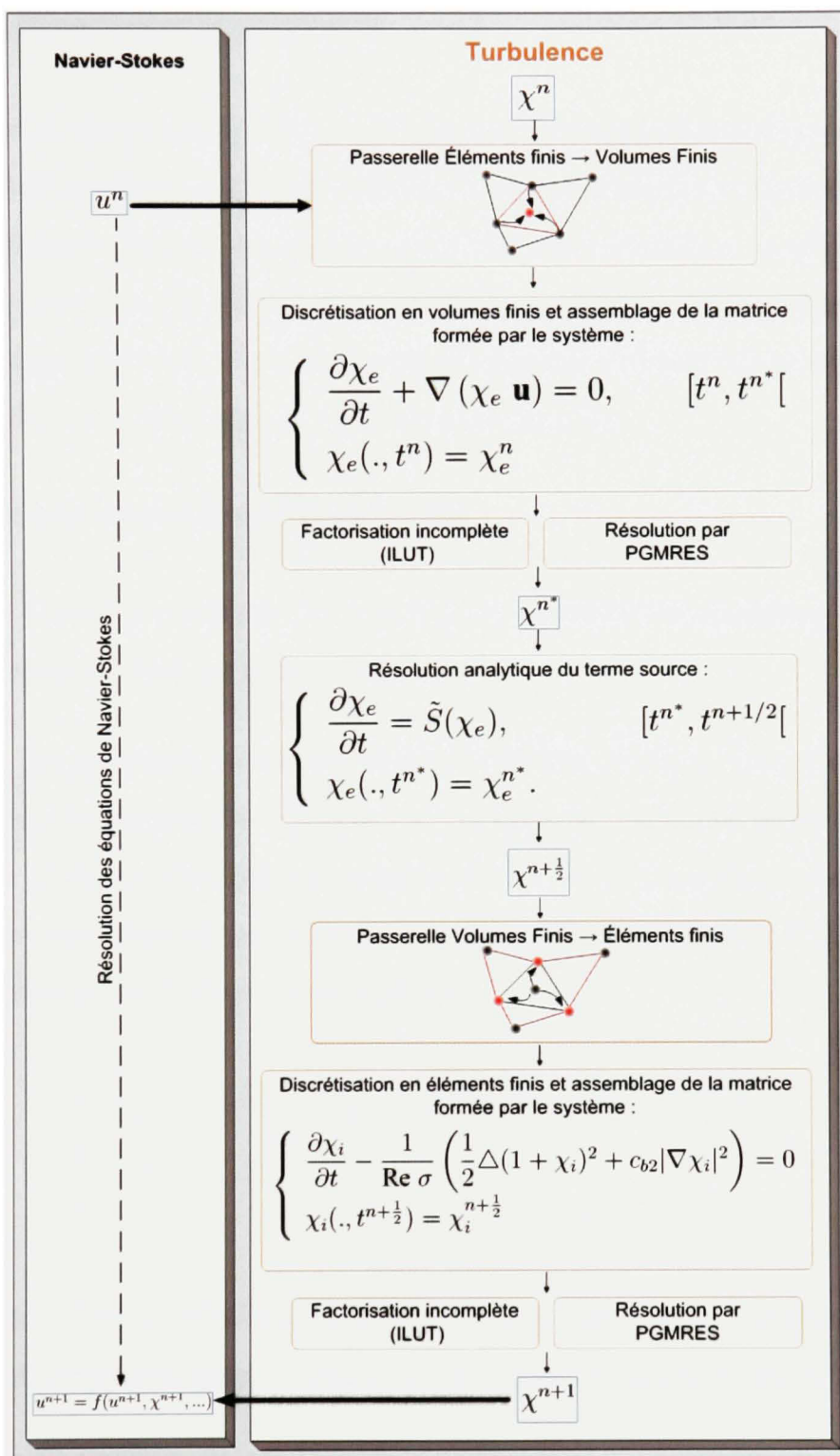


Figure 5.2 *Turbulence : organigramme de l'approche VF-EF, deuxième variante.*

5.4 Approche purement éléments finis

Pour améliorer les performances du code couplé, une autre approche est proposée. Le terme de convection est alors discrétisé en éléments finis, avec une stabilisation de premier ordre, avec le terme de diffusion. Le terme source est calculé par élément utilisant un schéma de volumes finis. Ce changement permet d'alléger significativement le code informatique et d'offrir une aisance théorique dans le cas d'une éventuelle augmentation en ordre du terme de convection. Ces modifications sont présentées dans les sections suivantes.

5.4.1 Un premier algorithme (Algorithme 8)

L'algorithme 8 ainsi que la figure 5.3 à la page 81 illustrent les opérations réalisées par le module de turbulence basé sur cette variante.

Au début du pas de temps, le module de Navier-Stokes envoie la vitesse \mathbf{u} au module de turbulence. Ensuite, partant d'une solution χ_e^n , la viscosité turbulente au pas de temps n (la projection de χ_i^n sur les éléments par la passerelle EF \rightarrow VF), une approximation du terme source (\tilde{S}_e) est alors calculée par la résolution temporelle de l'équation 5.12 par un schéma de différences finies (explicite ou semi-implicite) :

$$\begin{cases} \frac{\partial \chi_e}{\partial t} = c_{b1} \tilde{\omega} \chi_e - \frac{c_{w1}}{\text{Re}} f_w \left(\frac{\chi_e}{d} \right)^2 & t \in [t^n, t^{n+\frac{1}{2}}[\\ \tilde{S}_e(\chi_e) = \frac{\chi_e^{n+\frac{1}{2}} - \chi_e^n}{\Delta t} \end{cases} \quad (5.12)$$

Le terme source ($\tilde{S}_e(\chi_e)$) calculé par élément est ensuite injecté dans le système (5.13) (termes de diffusion et convection) pour compléter le calcul de la solution globale χ_i^{n+1} .

Remarque : Dans le cas de ce dernier schéma numérique, on n'a pas besoin de la passerelle VF \rightarrow EF. En effet, le terme source \tilde{S}_e calculé par élément peut être directement injecté dans

l'équation 5.13 qui est résolue par éléments finis.

$$\begin{cases} \frac{D\chi_i}{Dt} - \frac{1}{\text{Re } \sigma} \left(\frac{1}{2} \Delta (1 + \chi_i)^2 + c_{b2} |\nabla \chi_i|^2 \right) = \tilde{S}_e(\chi_e) & t \in [t^n, t^{n+1}[\\ \chi_i(\cdot, t^n) = \chi_i^n \end{cases} \quad (5.13)$$

Finalement, le module de turbulence envoie la solution χ_i^{n+1} au module de Navier-Stokes. Ce dernier calcule alors la nouvelle vitesse $\mathbf{u}^{n+1} = f(\mathbf{u}^{n+1}, \chi_i^{n+1}, \dots)$. Les deux modules passent alors au pas suivant et ainsi de suite.

Algorithme 8 : Turbulence : approche purement EF, première variante

Données : χ_i^0 aux nœuds, N : Nombre de pas de temps.

Résultat : χ_i^{n+1} .

initialisation : $\chi_i = \chi_i^0$.

pour $n = 1, N$ **faire**

 Calculer χ_e^n aux centres des éléments par projection de χ_i^n ;

 Recevoir les valeurs de la vitesse du module Navier-Stokes ;

En volumes finis ;

 Calculer $\chi_e^{n+\frac{1}{2}}$ utilisant l'un des schémas suivants :

$$\begin{cases} \text{Explicite : } \chi_e^{n+\frac{1}{2}} = \chi_e^n + \left[c_{b1} \tilde{\omega} \chi_e^n - \frac{c_{w1}}{\text{Re}} f_w \left(\frac{\chi_e^n}{d} \right)^2 \right] \Delta t \\ \text{Implicite : } \chi_e^{n+\frac{1}{2}} = \chi_e^{n+\frac{1}{2}} + \left[c_{b1} \tilde{\omega} \chi_e^{n+\frac{1}{2}} - \frac{c_{w1}}{\text{Re}} f_w \left(\frac{\chi_e^{n+\frac{1}{2}}}{d} \right)^2 \right] \Delta t \\ \text{Semi implicite : } \chi_e^{n+\frac{1}{2}} = \chi_e^n + \left[c_{b1} \tilde{\omega} \chi_e^n - \frac{c_{w1}}{\text{Re}} f_w \left(\frac{\chi_e^{n+\frac{1}{2}}}{d} \right)^2 \right] \Delta t \end{cases}$$

 Calculer $\tilde{S}_e(\chi_e) = \frac{\chi_e^{n+\frac{1}{2}} - \chi_e^n}{\Delta t}$;

En éléments finis ;

 Calculer χ_i^{n+1} par la résolution du système complet :

$$\frac{\partial \chi_i}{\partial t} - \frac{1}{\text{Re } \sigma} \left(\frac{1}{2} \Delta (1 + \chi_i)^2 + c_{b2} |\nabla \chi_i|^2 \right) + \mathbf{u} \cdot \nabla \chi_i = \tilde{S}_e(\chi_e) ;$$

$\chi_i^n = \chi_i^{n+1}$;

 Envoyer les valeurs de χ_i^n au module Navier-Stokes.

fin

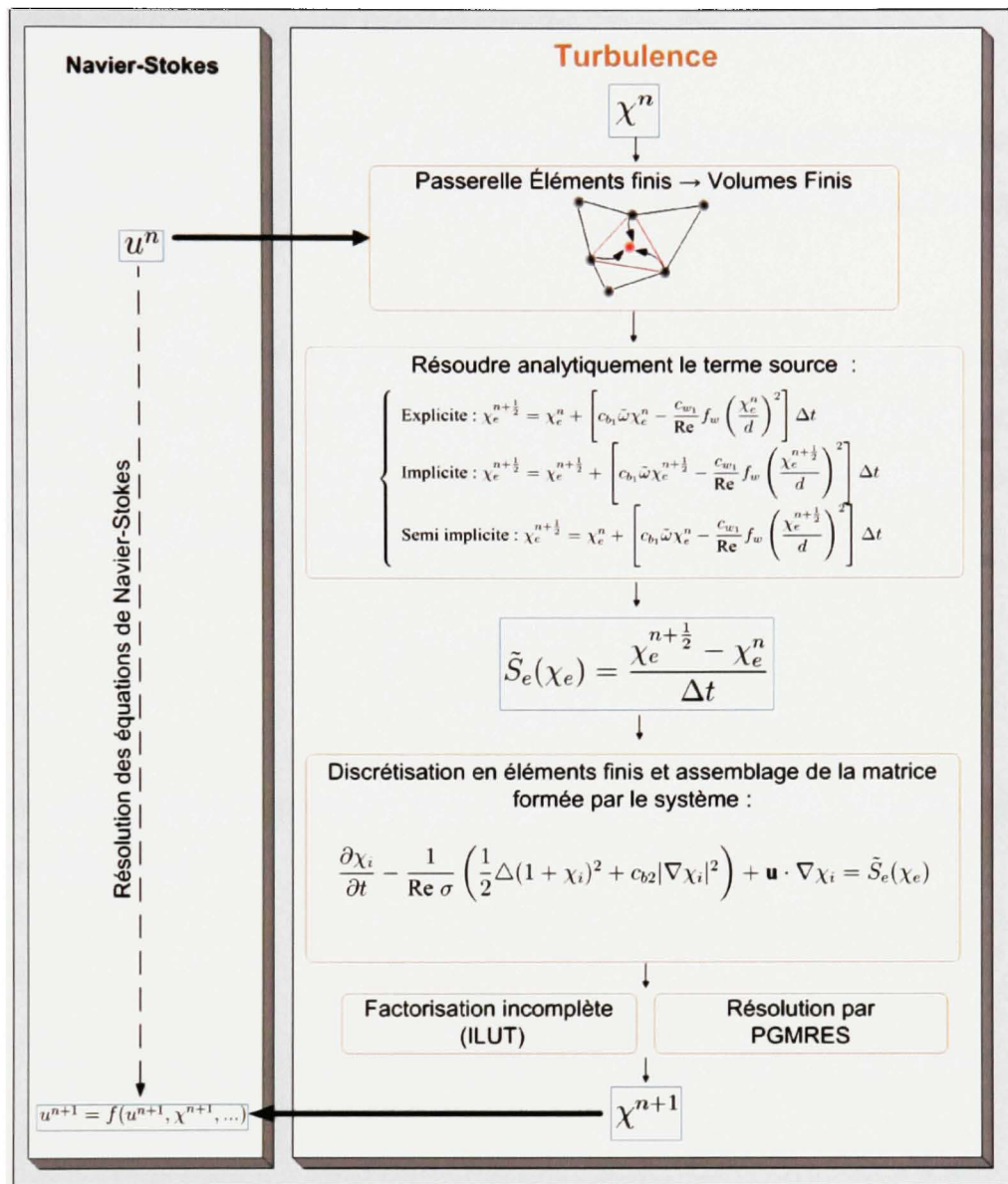


Figure 5.3 *Turbulence : organigramme de l'approche purement EF, première variante.*

5.4.2 Un deuxième algorithme (Algorithme 9)

Ce dernier schéma numérique est très similaire au schéma discuté à la section précédente. La seule distinction réside dans le calcul du terme source. En effet, ce dernier est approché par un schéma semi-implicite.

Au début du pas de temps, le module de fluide envoie la vitesse \mathbf{u} au module de turbulence. Ensuite, utilisant les valeurs χ_e^n , la viscosité turbulente au pas de temps n (la projection de χ_i^n sur les éléments par la passerelle $EF \rightarrow VF$), le module de turbulence envoie au module fluide les paramètres a et b suivants l'équation 5.14.

$$\begin{cases} a = c_{b1} \tilde{\omega}^n \\ b = \frac{c_{w1}}{\text{Re}} f_w^n \end{cases} \quad (5.14)$$

Ce dernier calcule alors la nouvelle vitesse $\mathbf{u}^{n+1} = f(\mathbf{u}^{n+1}, \chi_i^{n+1}, \dots)$ par la résolution de l'équation de transport :

$$\begin{cases} \frac{D\chi_i}{Dt} - \frac{1}{\text{Re} \sigma} \left(\frac{1}{2} \Delta (1 + \chi_i)^2 + c_{b2} |\nabla \chi_i|^2 \right) = a\chi_i - b\chi_i^2 & t \in [t^n, t^{n+1}[\\ \chi_i(\cdot, t^n) = \chi_i^n \end{cases} \quad (5.15)$$

Remarque : la discrétisation par éléments finis des équations 5.13 et 5.15 utilise une formulation standard de Galerkin où un terme de diffusion numérique est ajouté. Ce terme est de la forme :

$$- \frac{\|\mathbf{u}\| h_{\min} C_\chi}{2} \Delta \chi \quad \text{où } C_\chi \text{ est un coefficient } \leq 1 \quad (5.16)$$

Il en résulte un schéma de discrétisation de premier ordre en h_{\min} mais qui maintient, par expérience, la positivité de χ .

Après ces opérations les deux modules passent au pas suivant et ainsi de suite. L'algorithme devient alors :

Algorithme 9 : Turbulence : approche purement EF, deuxième variante

Données : χ_i^0 aux nœuds, N : Nombre de pas de temps.

Résultat : χ_i^{n+1} .

initialisation : $\chi_i = \chi_i^0$.

pour $n = 1, N$ **faire**

 Calculer χ_e^n aux centres des éléments par projection de χ_i^n ;

 Recevoir les valeurs de la vitesse du module Navier-Stokes ;

En volumes finis ;

 Calculer a et b avec :

$$\begin{cases} a = c_{b1} \tilde{\omega}^n \\ b = \frac{c_{w1}}{\text{Re } d^2} f_w^n \end{cases}$$

En éléments finis ;

 Calculer χ_i^{n+1} par la résolution du système complet :

$$\frac{\partial \chi_i}{\partial t} - \frac{1}{\text{Re } \sigma} \left(\frac{1}{2} \Delta (1 + \chi_i)^2 + c_{b2} |\nabla \chi_i|^2 \right) + \mathbf{u} \cdot \nabla \chi_i = a \chi_i - b \chi_i^2 ;$$

$$\chi_i^n = \chi_i^{n+1} ;$$

 Envoyer les valeurs de χ_i^n au module Navier-Stokes.

fin

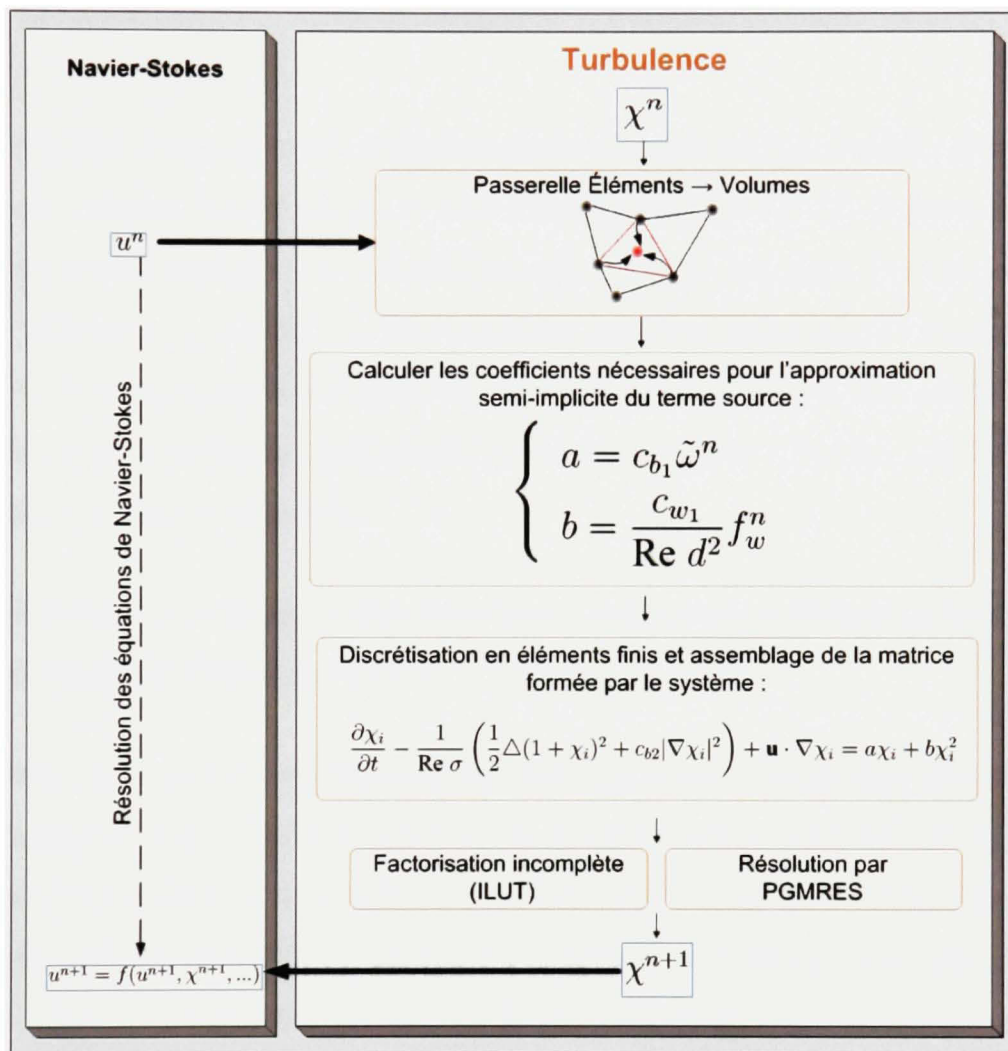


Figure 5.4 *Turbulence : organigramme de l'approche purement EF, deuxième variante.*

CHAPITRE 6

RÉSULTATS

Dans ce chapitre, nous présentons différents résultats numériques pour valider chaque module séparément. Ensuite, nous validons le code PFES360 dans le cas des problèmes multiphysiques où plusieurs modules sont couplés ensemble. La garantie d'une solution positive de la viscosité turbulente est vérifiée. La pertinence des nouvelles définitions de la matrice τ et de l'opérateur de capture de choc est mise en évidence. Les simulations sont conduites sur une grappe de calcul (Cluster) de vingt nœuds (2 x AMD Athlon 64 2.16 Ghz, 3.86 GB de RAM) connectés par un réseau infiniband.

6.1 Validation du couplage Euler–structure : Application à l'étude de la stabilité aéroélastique de l'aile Agard 445.6

Le problème de la stabilité aéroélastique de l'aile Agard 445.6 est un banc d'essai classique bien documenté (Yates, 1987). L'aile possède un profil mince NACA65A004. Le modèle considéré est le modèle 3 de la référence (Yates, 1987). Pour la résolution des équations d'Euler, un maillage de 37965 nœuds et 177042 éléments est utilisé. Le nombre d'équations couplées générées est 388464. Concernant la structure, le maillage utilisé est constitué de 1250 nœuds et 1176 éléments coques quadrilatéraux. Les cinq premiers modes sont obtenus à partir du code commercial ANSYS. Les fréquences propres calculées sont : $9.6Hz$, $39.42Hz$, $49.60Hz$, $96.095Hz$ et $126.30Hz$. Elles sont très proches de celles rapportées dans (Yates, 1987). Pour chaque nombre de Mach, une solution d'écoulement est obtenue en mode rigide. Comme le profil est symétrique, la portance est nulle. La structure est alors perturbée par une force imposée en un point au bout de l'aile durant un pas de temps. La structure est ensuite relâchée pour étudier sa stabilité dynamique. On fait alors varier la pression dynamique de référence jusqu'à l'obtention éventuelle d'un mode instable.

Les figures 6.1 et 6.2 illustrent l'évolution des coordonnées généralisées pour les deux premiers modes. Pour la pression dynamique $q = 62 \text{ lb}/\text{ft}^2$, les résultats montrent une solution neutre. En augmentant la pression à $q = 63 \text{ lb}/\text{ft}^2$, la réponse du premier mode diverge. Selon les résultats de Yates (Yates, 1987) la pression critique à $Mach = 0.96$ est de $61.3 \text{ lb}/\text{ft}^2$. PFES reproduit avec précision les résultats expérimentaux de l'aile Agard 445.6.

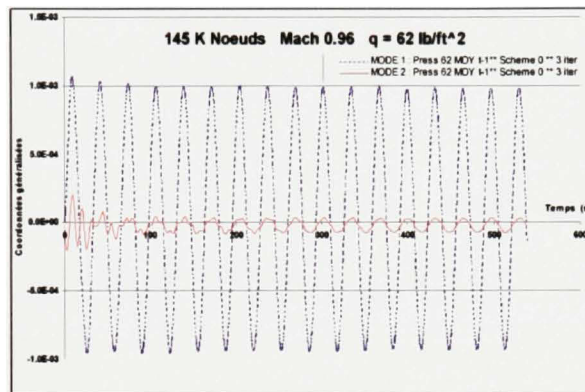


Figure 6.1 *Évolution des coordonnées généralisées dans le temps $q = 62 \text{ lb}/\text{ft}^2$.*

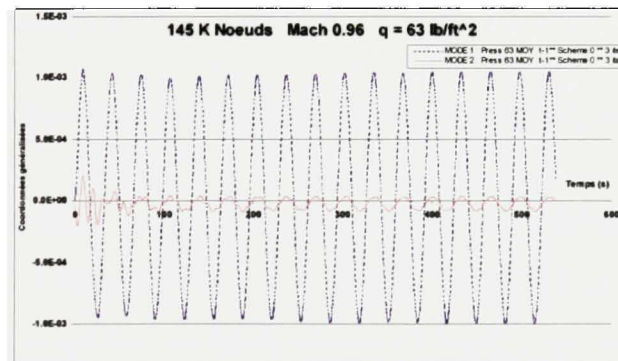


Figure 6.2 *Évolution des coordonnées généralisées dans le temps $q = 63 \text{ lb}/\text{ft}^2$.*

La figure 6.3 présente une comparaison des indices de flottement obtenus par le code PFES360 avec les résultats expérimentaux (Yates, 1987) et avec quelques résultats numériques rapportés

dans la littérature (Soulaïmani et Ben Haj Ali, 2003; Farhat et Lesoinne, 2000; Lee-Rauch et Batina, 1993; Gupta, 1996).

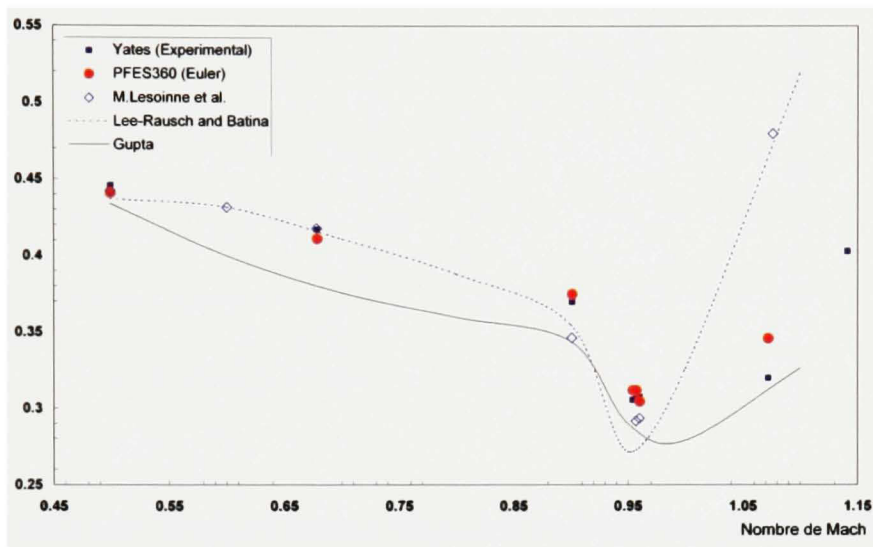


Figure 6.3 *Indice de flottement Agard 445.6.*

D'autres résultats sont publiés dans (Soulaïmani et al., 2004) et par souci de concision ils ne sont pas reproduits dans cette thèse.

6.2 Validation du module de turbulence isolé : Plaque plane avec une vitesse imposée

Un simple cas test est proposé pour la validation du schéma numérique du module de turbulence. Les résultats sont publiés dans (Lorin et al., 2007; Lorin et al., 2006; Ben Haj Ali et al., 2004). Dans ce qui suit, le cas test est défini et quelques résultats sont analysés.

Le but est de valider la consistance du module de turbulence indépendamment des résultats du module fluide. Dans ce cas test, la vitesse du fluide communiquée au module de turbulence est alors imposée. L'algorithme utilisé ici est l'algorithme 6.

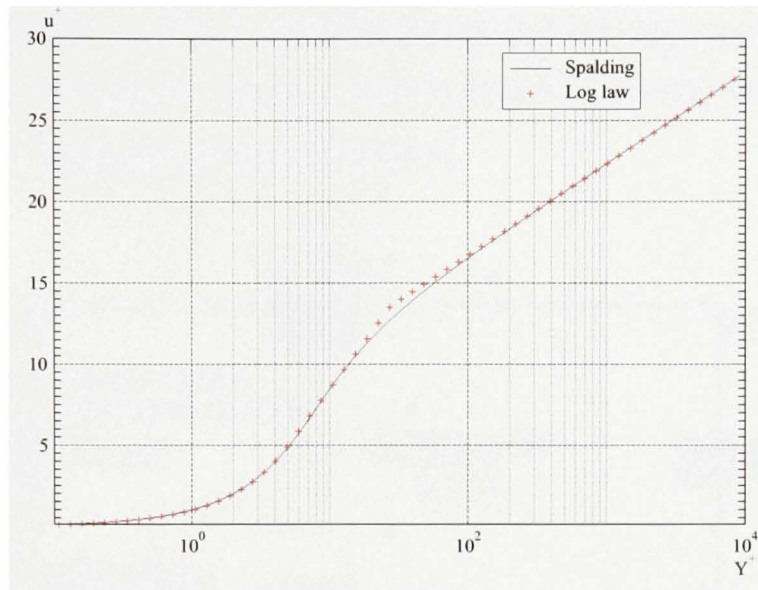


Figure 6.4 *Profil de vitesse imposée à l'entrée de la plaque plane.*

Le cas test consiste à calculer la viscosité turbulente sur une plaque plane sans gradient de pression. La solution de ce problème est connue (Spalart et Allmaras, 1994); la viscosité turbulente χ est linéaire dans la couche limite et sa valeur est égale à κy^+ . Le champ de vitesse u^+ imposé est donné par l'équation de Spalding (figure 6.4). Le domaine physique est un cube de dimension $1.0 \times 0.2 \times \frac{y^+}{\text{Re } u^+} \Big|_{y^+=200}$. Dans la direction verticale z , la direction de y^+ , la hauteur du domaine est choisie pour avoir $y^+ = 200$ à l'infini ($z = z_{\max}$). La vitesse de friction est approximée par la formule suivante :

$$u_\tau \approx \sqrt{\frac{0.0135}{\text{Re}^{1/7}}} \quad (6.1)$$

Les conditions aux limites sont :

- $\chi = \kappa y^+$ à l'entrée
- $\chi = 0$ à la sortie
- $\chi = \kappa y^+$ à $z = z_{\max}$

Le nombre de Reynolds est égale à 10^7 et La vitesse du fluide est imposée à $\mathbf{u} = u_\tau \mathbf{u}^+$. La solution initiale χ_0 est égale à 0 (zéro) à l'intérieur du domaine et elle respecte les conditions aux limites ailleurs. La figure 6.5 illustre ce cas test.

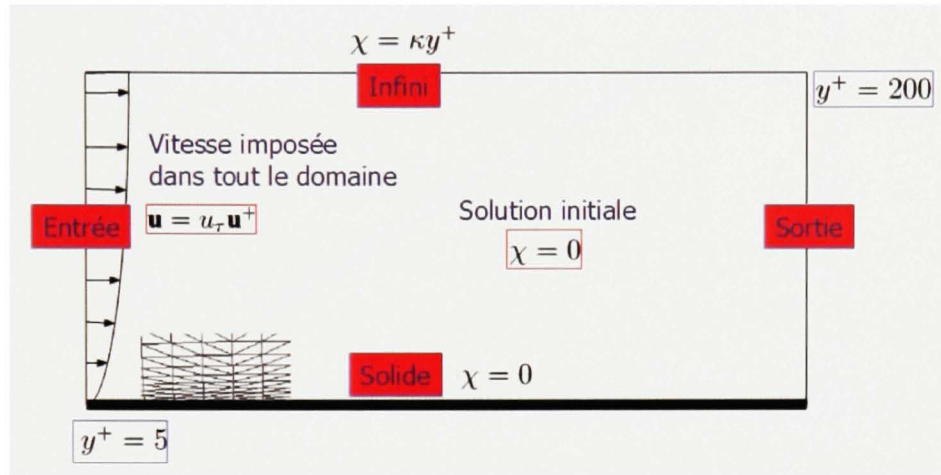


Figure 6.5 *Plaque plane avec vitesse imposée : Conditions aux limites.*

Plusieurs maillages (Tableau 6.1) ont été utilisés pour étudier l'effet du nombre de nœuds dans la direction verticale (y^+) sur le résultat. Le premier point de tout ces maillages se trouve à une distance équivalente à $y^+ = 5$.

Tableau 6.1

Maillages de la plaque plane

Nombre de nœuds dans la direction y^+	Nombre de nœuds total	Nombre d'éléments
20	5460	19600
30	5790	28560
40	10680	38480

Les résultats montrent que le profil de χ est indépendant de x (Figure 6.6) et de y .

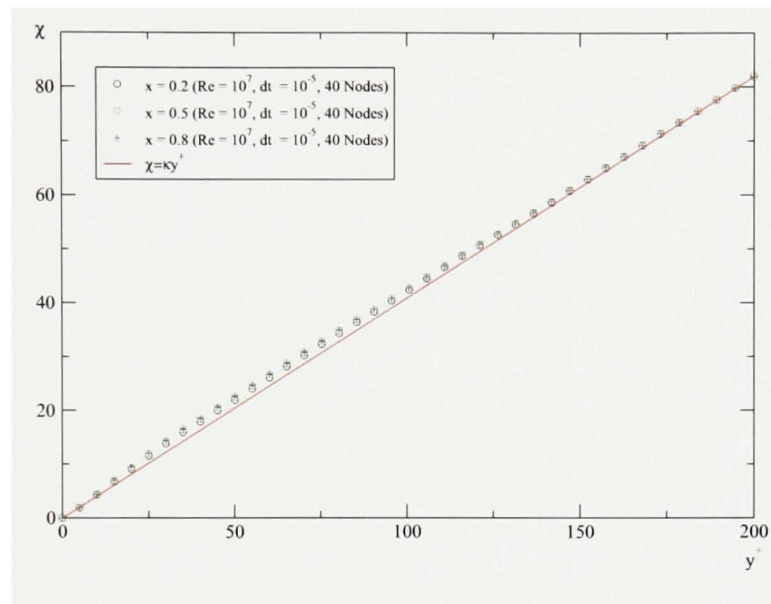


Figure 6.6 *Plaque plane avec une vitesse imposée : χ à différents plans $x = cte$.*

Le tableau 6.2 présente l'erreur relative calculée à l'aide de l'équation (6.2).

Tableau 6.2

Erreur quadratique relative

Nombre de nœuds dans la direction y^+	Erreur quadratique relative
20	0.169 (17%)
30	0.075 (7.5%)
40	0.035 (3.5%)

$$\text{Erreur} = \frac{\sqrt{\sum_i^{N_{ib.noeuds}} (\chi_i^{n+1} - \kappa y_i^+)^2}}{\sqrt{\sum_i^{N_{ib.noeuds}} (\kappa y_i^+)^2}}, \quad (6.2)$$

La figure 6.7 montre que le raffinement de maillage améliore la qualité de la solution qui s'approche de la valeur théorique κy^+ avec le maillage fin.

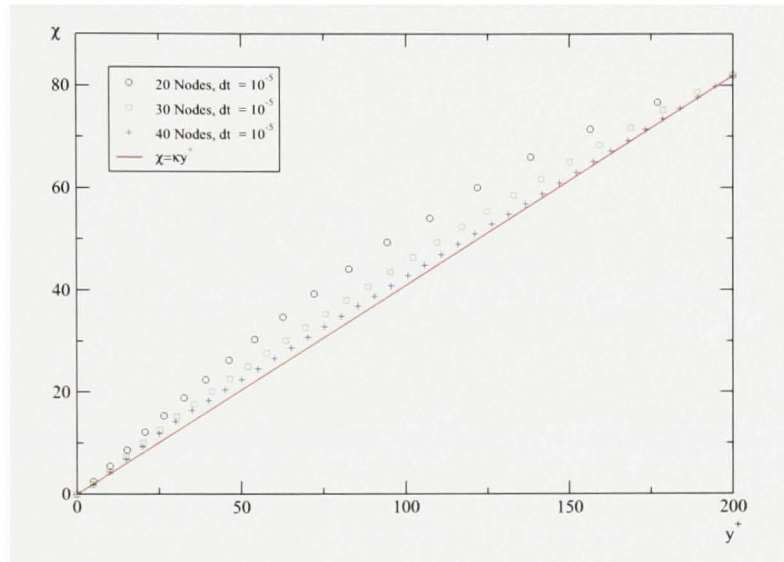


Figure 6.7 *Plaque plane avec une vitesse imposée : effet de maillage.*

La figure 6.8 présente l'erreur relative totale en fonction du pas de temps pour les différentes configurations du maillage.

La figure 6.9 montre la convergence du schéma numérique en fonction du nombre de pas de temps respectant la contrainte (6.3).

$$\frac{\sqrt{\sum_i^{N_{ib.noeuds}} (\chi_i^{n+1} - \chi_i^n)^2}}{\sqrt{\sum_i^{N_{ib.noeuds}} (\chi_i^n)^2}} \leq 10^{-6}. \quad (6.3)$$

Remarque : d'autres tests ont montré l'indépendance du schéma numérique par rapport à la solution initiale.

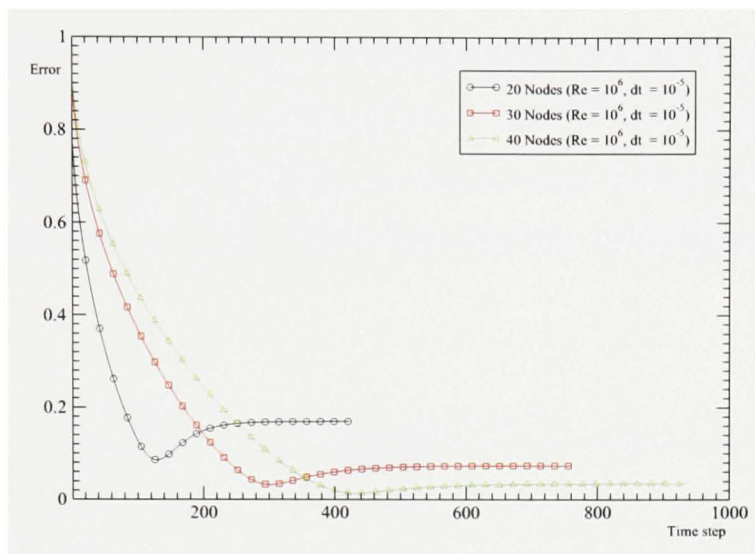


Figure 6.8 *Plaque plane avec une vitesse imposée : erreur en fonction de pas de temps.*

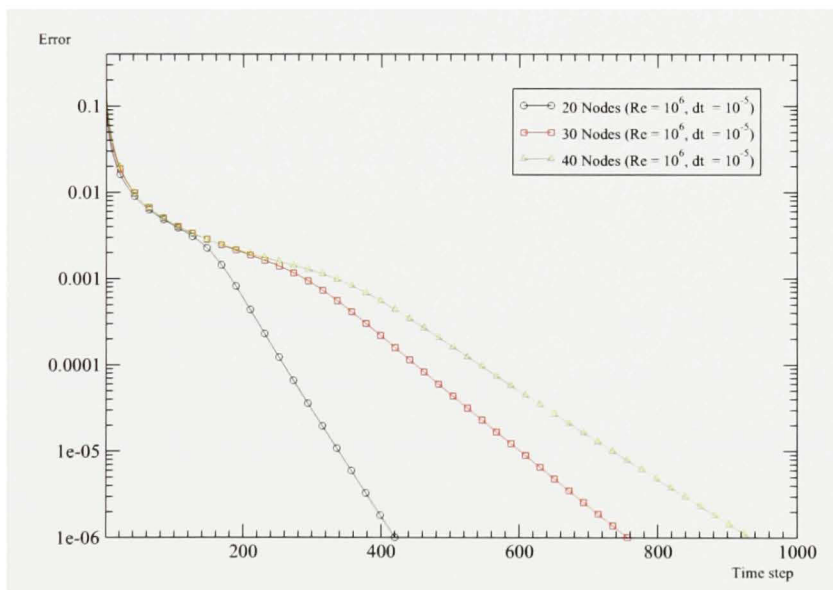


Figure 6.9 *Plaque plane avec vitesse imposée : Convergence du schéma amélioré.*

6.3 Validation du couplage Navier-Stokes–turbulence (RANS) : Plaque plane

Le cas test est similaire à celui présenté à la section 6.2. Cependant, dans le présent test la vitesse est calculée par le module fluide. La résolution de la turbulence est basée sur l'algo-

ritme 8. Le maillage utilisé est constitué de 40 points dans la direction y^+ . Le premier point est à $y^+ = 5$. Le pas de temps fluide est de $\Delta t_{NS} = 10^{-5}$. Le pas de temps turbulence est $\Delta t_{turb} = \frac{\Delta t_{NS}}{2^N}$ avec $N \in \{0, \dots, 8\}$. Pendant un pas de temps fluide le module de turbulence fait 2^N pas de temps. Les conditions aux limites sont illustrées par la figure 6.10.

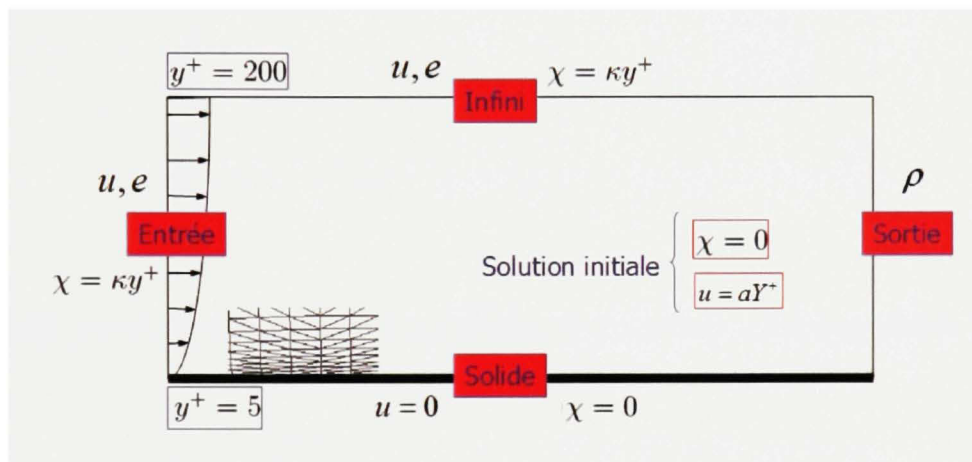


Figure 6.10 *Plaque plane avec vitesse imposée : Conditions aux limites.*

La solution initiale de la viscosité turbulente est fixée à zéro dans tout le domaine et respecte les conditions aux limites sur les frontières. Ce choix de solution initiale a pour but de mettre en évidence la robustesse des algorithmes proposés. La solution initiale de la vitesse est linéaire de 0 à 200.

La figure 6.11 illustre la solution fluide après 6000 itérations. Le graphique présente la vitesse U^+ en fonction de y^+ pour différents pas de temps. On remarque que la vitesse calculée est peu sensible à la variation du pas de temps dans la plage des valeurs présentées.

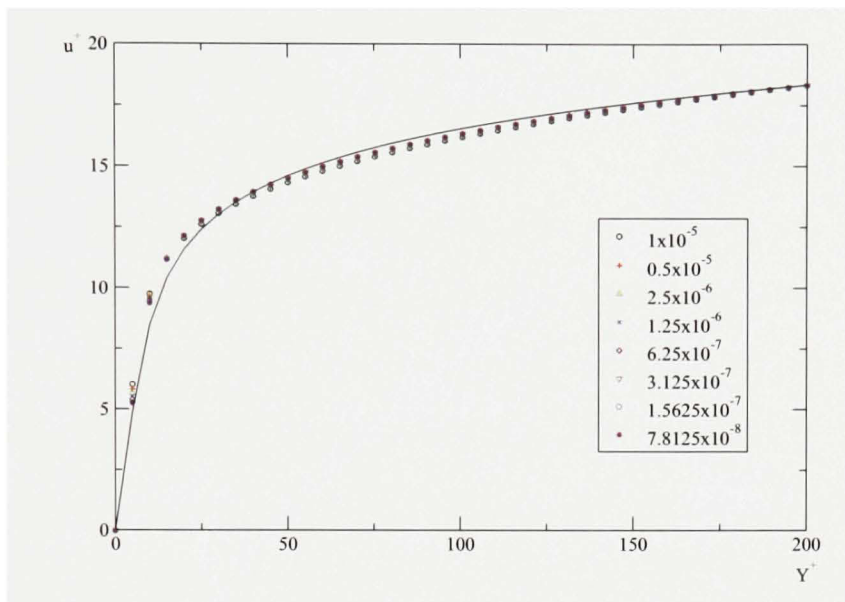


Figure 6.11 *Profil de vitesse dans le cas d'une plaque plane, couplage fluide-structure.*

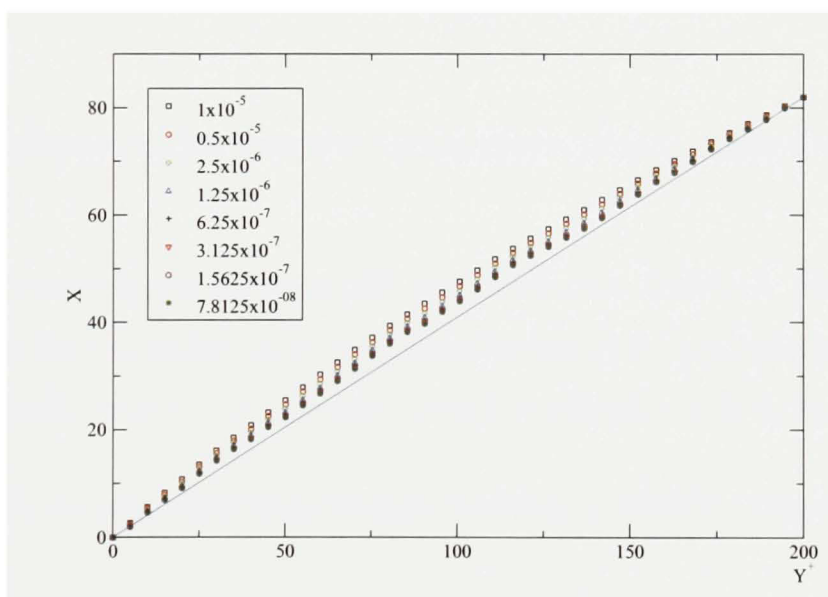


Figure 6.12 *Profil de χ dans le cas d'une plaque plane, couplage fluide-structure.*

La figure 6.12 montre que la résolution du module converge vers une solution acceptable pour des valeurs de pas de temps inférieures à 10^{-5} .

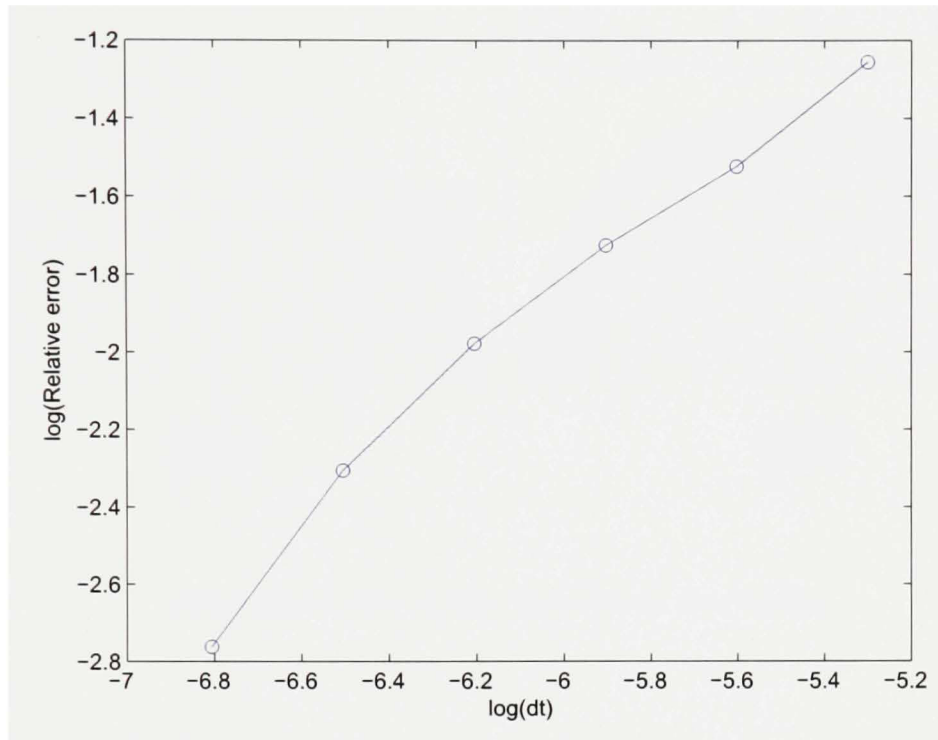


Figure 6.13 *Erreur relative dans le cas d'une plaque plane, couplage fluide–structure.*

La figure 6.13 montre que la méthode offre une convergence d'ordre 1. Nous avons constaté que dans ce cas test, la résolution par la méthode de Galerkin toute seule (sans SUPG ni une viscosité artificielle) n'engendre aucun problème d'oscillations ou de convergence.

6.4 Validation du couplage Navier-Stokes–turbulence (RANS) : Onera M6

Le cas test de l'Onera M6 est bien documenté dans la littérature. Des données expérimentales et numériques sont disponibles (Slater, 2008; Schmitt et Charpin, 1979). Les résultats sont obtenus à un angle d'attaque de 3,06 degrés, à un nombre de Mach de 0,8395 et à un nombre de Reynolds de 11,72 millions calculé sur la base de la corde aérodynamique moyenne (Schmitt et Charpin, 1979). La figure 6.14 expose deux manières d'imposer les conditions aux limites pour ce type de problème.

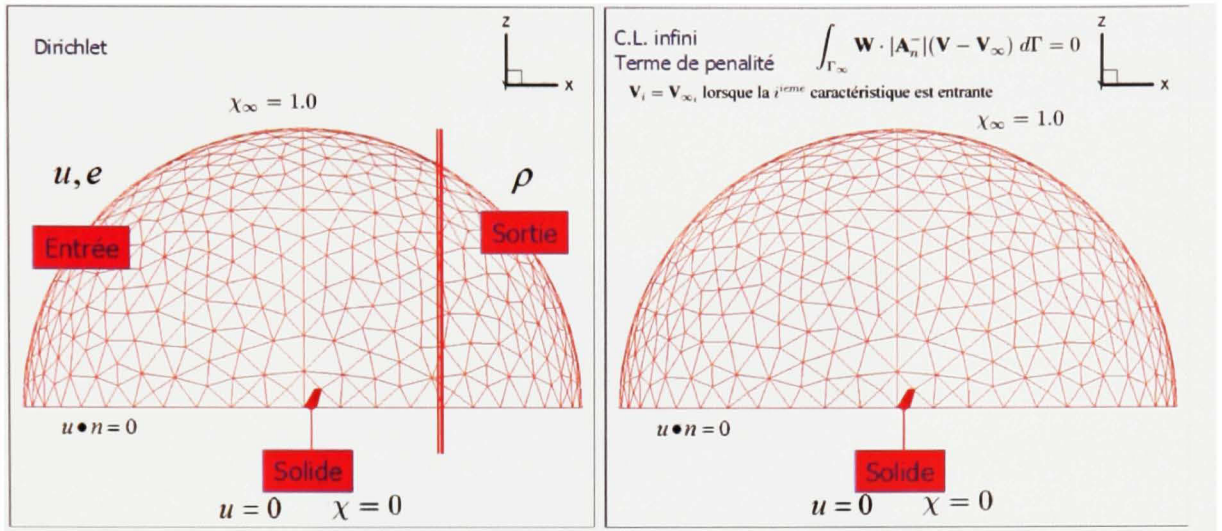


Figure 6.14 *Onera M6 : Conditions aux limites.*

6.4.1 Stratégie de résolution

Nous avons constaté que l'utilisation d'un pas de temps constant dans tout le domaine nous limite à de petits pas de l'ordre de 10^{-6} ce qui devenait un handicap. Grâce à la discrétisation totalement éléments finis (algorithme 9) il a été possible d'utiliser des pas de temps locaux définis élément par élément, donc petits dans la couche limite et grands ailleurs. La formule utilisée est la suivante :

$$\Delta t^e = \frac{h_{\min}}{(\|\mathbf{u}\| + c)^e} CFL \quad (6.4)$$

où h_{\min} est la distance minimale entre les nœuds de l'élément et $(\|\mathbf{u}\| + c)^e$ est la plus grande vitesse caractéristique de l'élément. La boucle sur les pas de temps constitue une procédure itérative pour accélérer la convergence vers une solution développée et en régime permanent. Aussi, il a été possible de garder les mêmes pas de temps pour la turbulence comme le fluide, c'est à dire, les deux modules restent synchronisés à chaque pas de temps. La stratégie de résolution globale consiste à initialiser le champ d'écoulement par les valeurs à l'infini partout et sauf à la paroi. De même, le champ de la viscosité χ est initialisé à $\chi_{\infty} = 1.0$ partout sauf

au solide où $\chi = 0$. Le nombre CFL est au départ égale à 0.5 et augmente tranquillement jusqu'à une valeur maximale de l'ordre de 2.5. Le modèle géométrique (figure 6.15) est adimensionnalisé par rapport à la corde au mur.

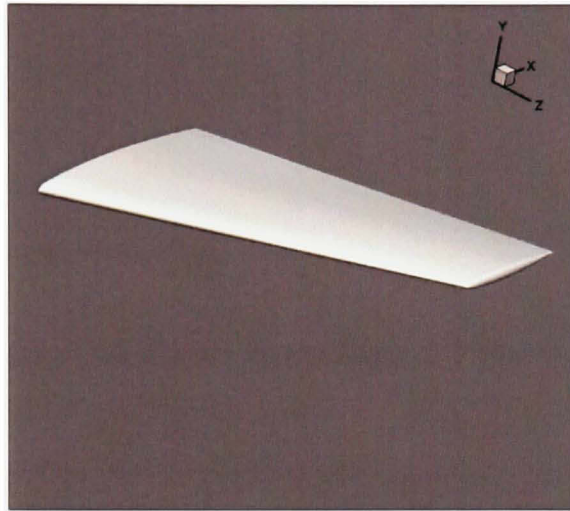


Figure 6.15 *Modèle 3D de l'aile Onera M6.*

Le maillage utilisé est un maillage purement tétraédrique généré dans un volume demi-sphérique de rayon 15 (figure 6.16).

Deux maillages tétraédriques anistropiques sont utilisés. Un premier maillage grossier d'environ 0.6 millions de nœuds et 3.4 millions d'éléments (figure 6.17). Le plus grand rapport entre les dimensions caractéristiques des mailles $\left(\text{Max} \left(\frac{h_{\max}}{h_{\min}} \right) \right)$ est de l'ordre de 2×10^4 .

Un deuxième maillage plus fin d'environ 1.3 millions de nœuds et 7.4 millions d'éléments (figure 6.18). Pour ce deuxième cas, le plus grand rapport entre les dimensions caractéristiques des mailles $\left(\text{Max} \left(\frac{h_{\max}}{h_{\min}} \right) \right)$ est de l'ordre de 10^3 . Dans les deux cas, des éléments anisotropiques (tétraèdres fortement distordus) sont utilisés pour limiter le nombre d'éléments.

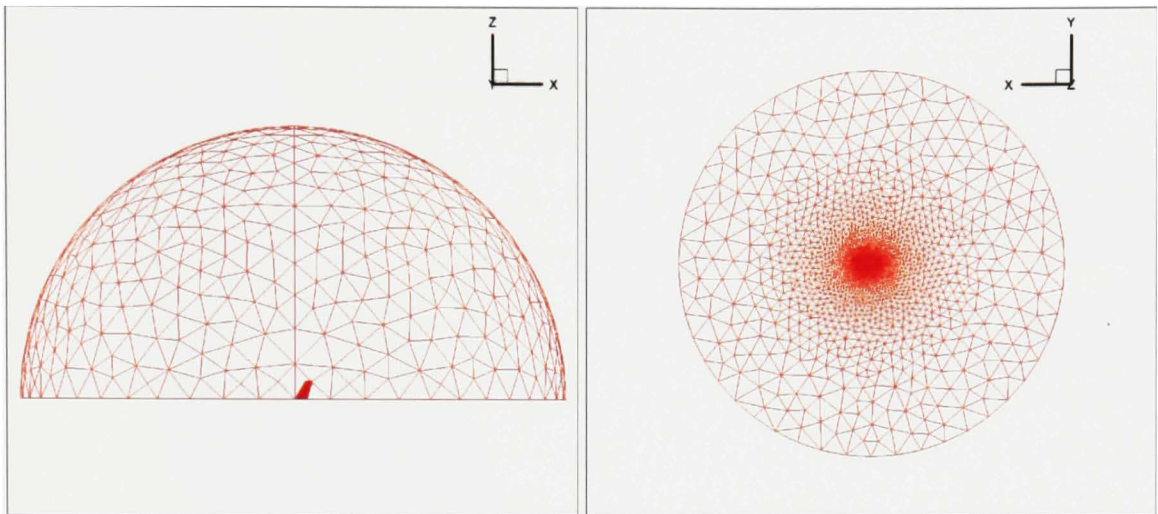


Figure 6.16 *Forme du maillage de l'Onera M6.*

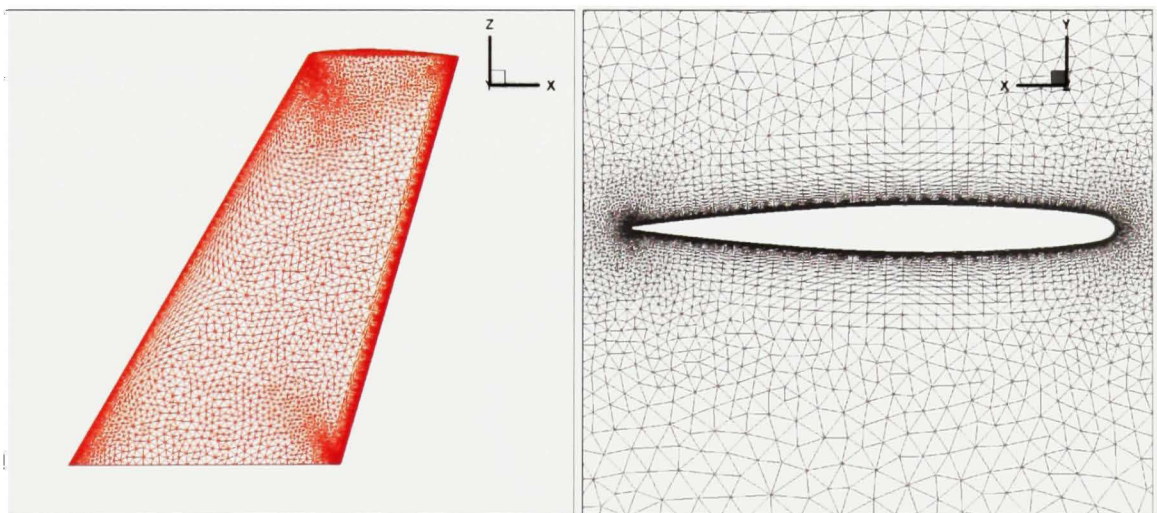


Figure 6.17 *Maillage grossier de l'Onera M6.*

La distance à la paroi solide du premier nœud variée entre $y^+ \simeq 1$ et $y^+ \simeq 12$.

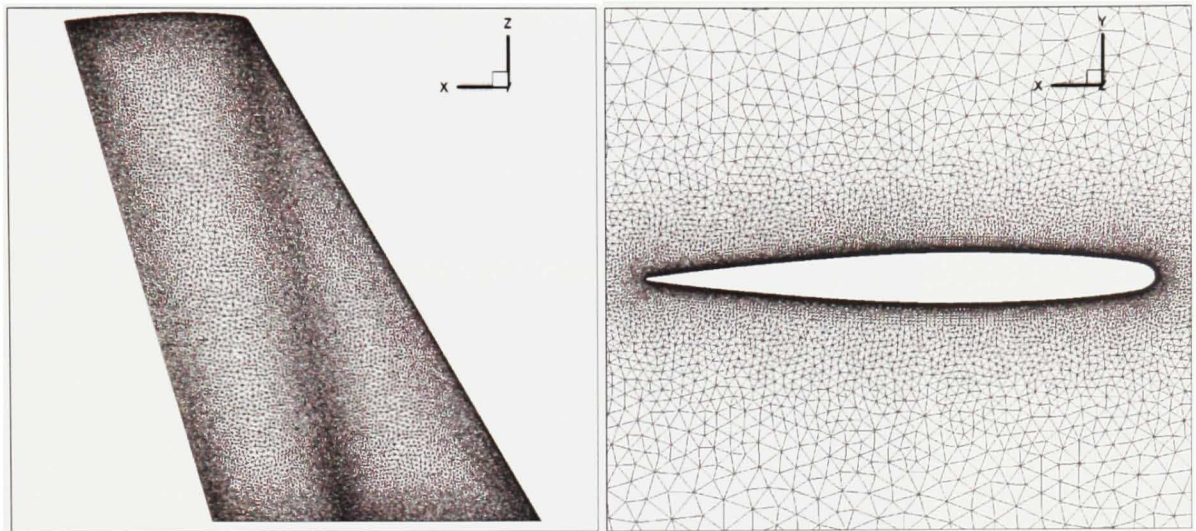


Figure 6.18 *Maillage fin de l'Onera M6.*

La figure 6.19 montre l'arrondissement effectué sur la forme du maillage au bord de fuite de l'aile.

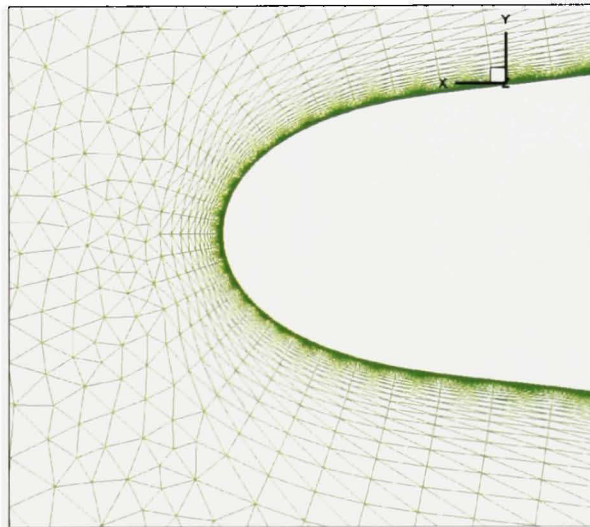


Figure 6.19 *Onera M6 RANS : Maillage sur le bord de fuite.*

La figure 6.20 montre les isovaleurs de χ au niveau de la section 1 ($y/b = 0.2$).

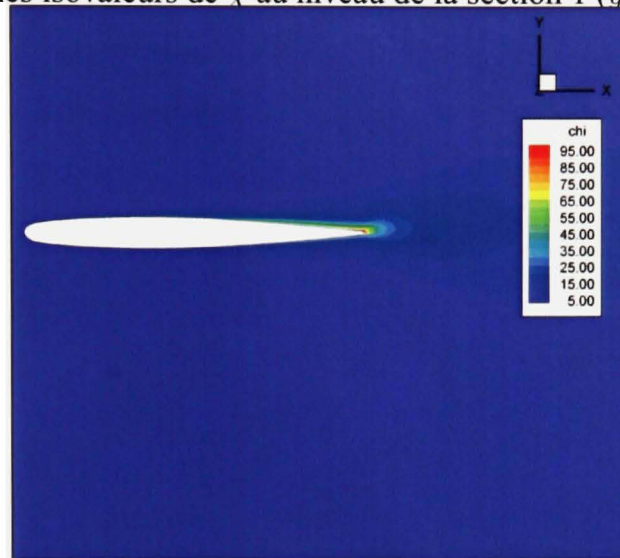


Figure 6.20 *Onera M6 RANS : Viscosité χ au niveau de la section $y/b = 0.2$.*

La figure 6.21 illustre les valeurs de la pression et du nombre de Mach sur la section 1 ($y/b = 0.2$). Elles sont trouvées en utilisant le maillage fin.

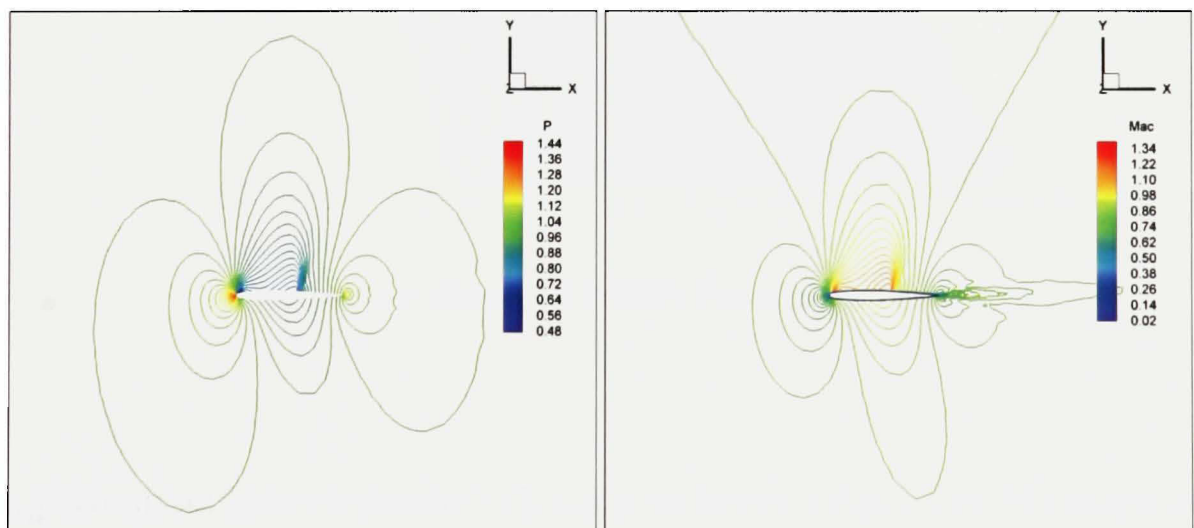


Figure 6.21 *Onera M6 RANS : Pression et Mach au niveau de la section $y/b = 0.2$.*

La figure 6.22 donne une idée qualitative sur la variation de la pression sur la surface de l'aile (maillage fin). Les isovaleurs de pression forment un choc de la forme d'un lambda (λ) qui est la caractéristique de l'aile Onera M6.

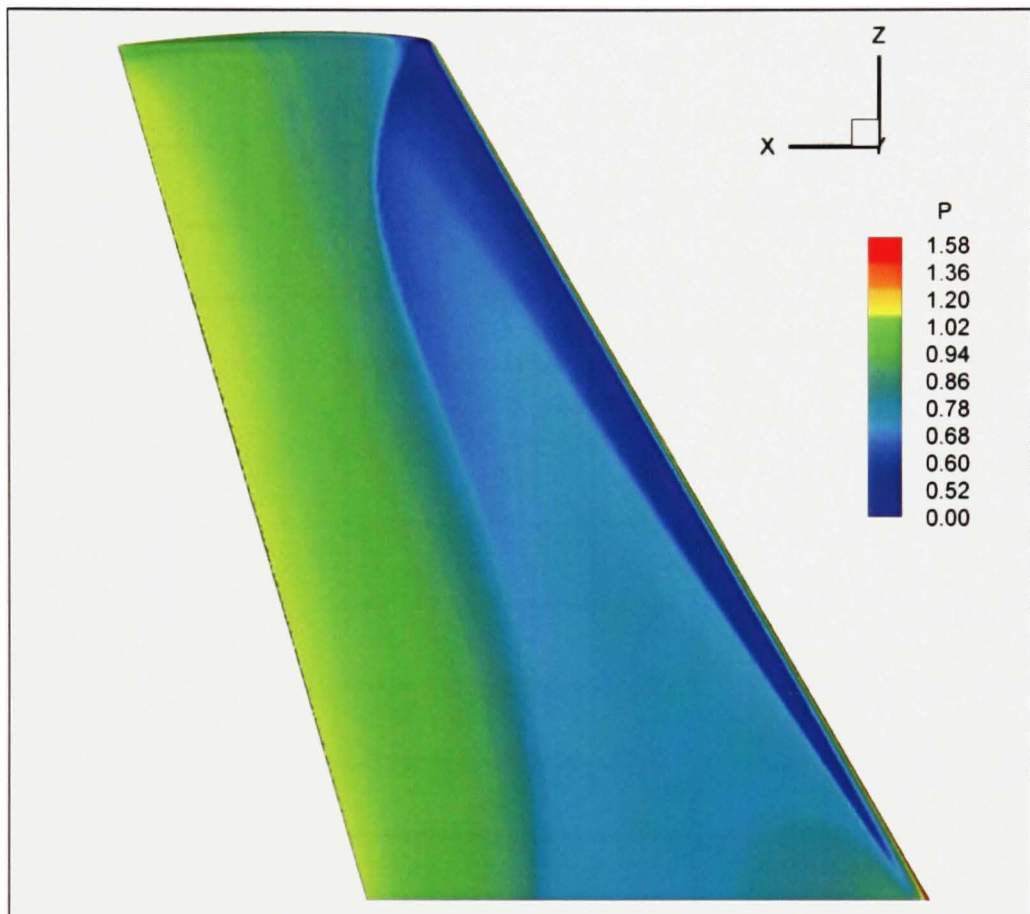


Figure 6.22 *Onera M6 RANS : Pression sur l'aile.*

Les valeurs de coefficient de pression $-C_p$ sont présentées sur la figure 6.23. Une comparaison entre les résultats trouvés avec le maillage fin, le maillage grossier ainsi que les résultats expérimentaux et les résultats obtenus par le code WIND (Schmitt et Charpin, 1979) (figures 6.24 à 6.30) montre une grande dépendance à la qualité du maillage.

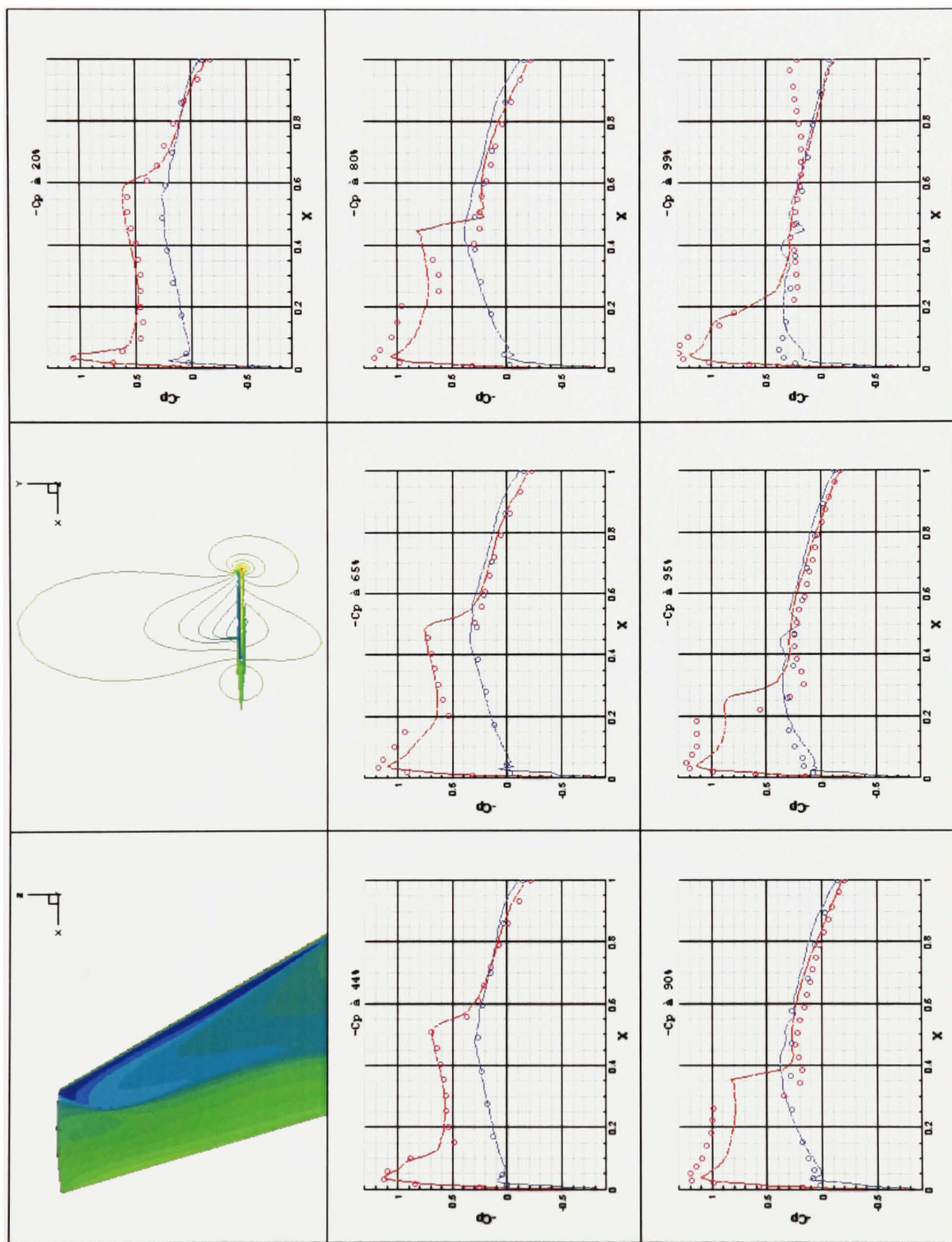


Figure 6.23 *Onera M6* : $-C_p$ à $y/b = 0.20$.

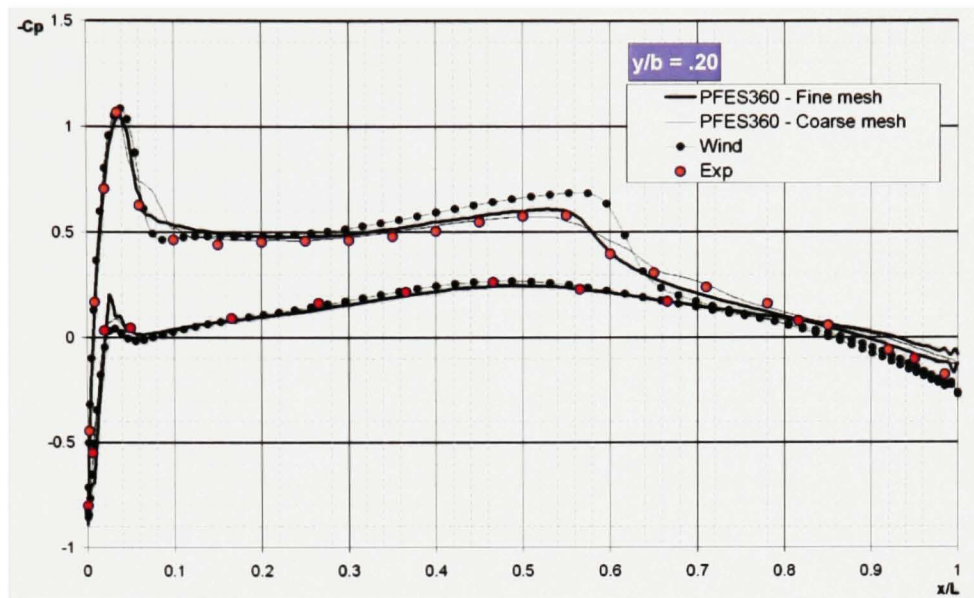


Figure 6.24 *Onera M6* : $-C_p$ à $y/b = 0.20$.

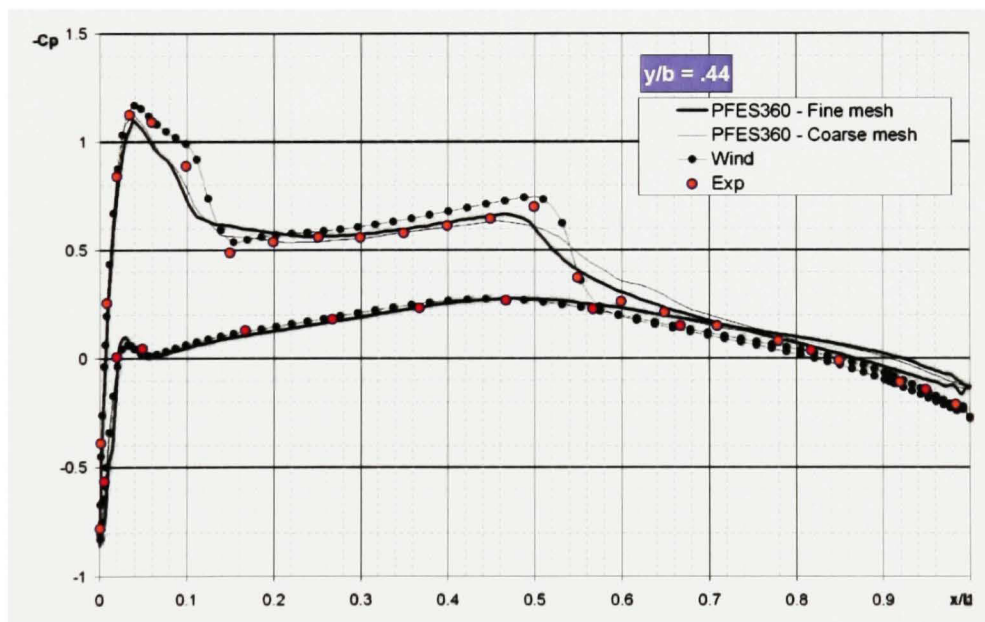


Figure 6.25 *Onera M6* : $-C_p$ à $y/b = 0.44$.

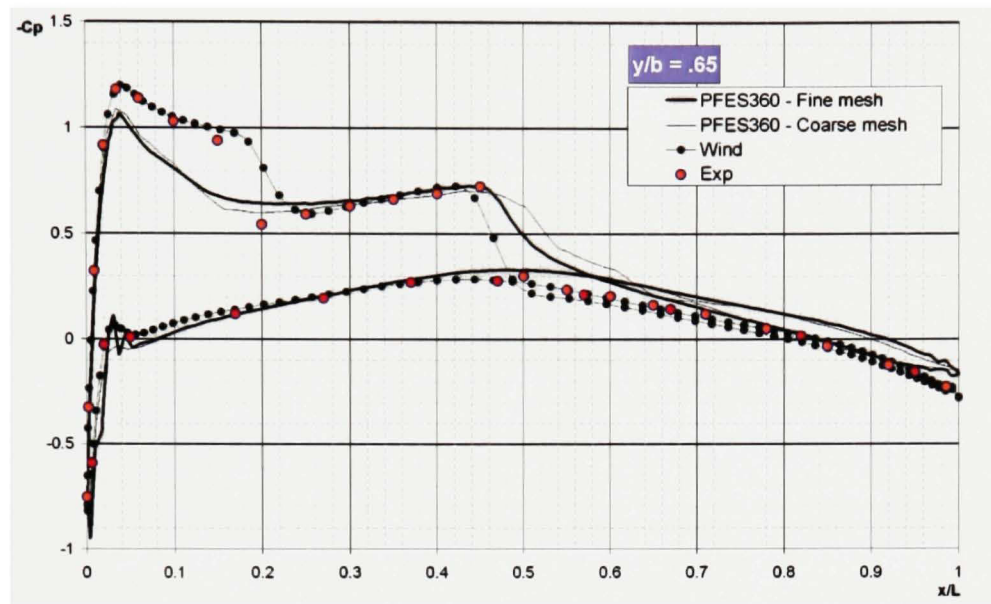


Figure 6.26 *Onera M6* : $-C_p$ à $y/b = 0.65$.

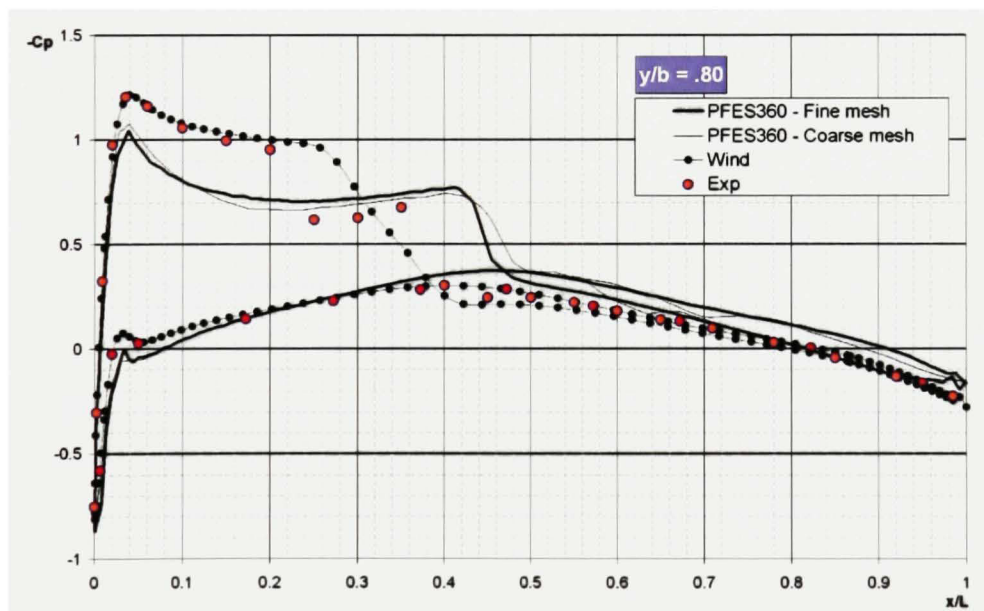


Figure 6.27 *Onera M6* : $-C_p$ à $y/b = 0.80$.

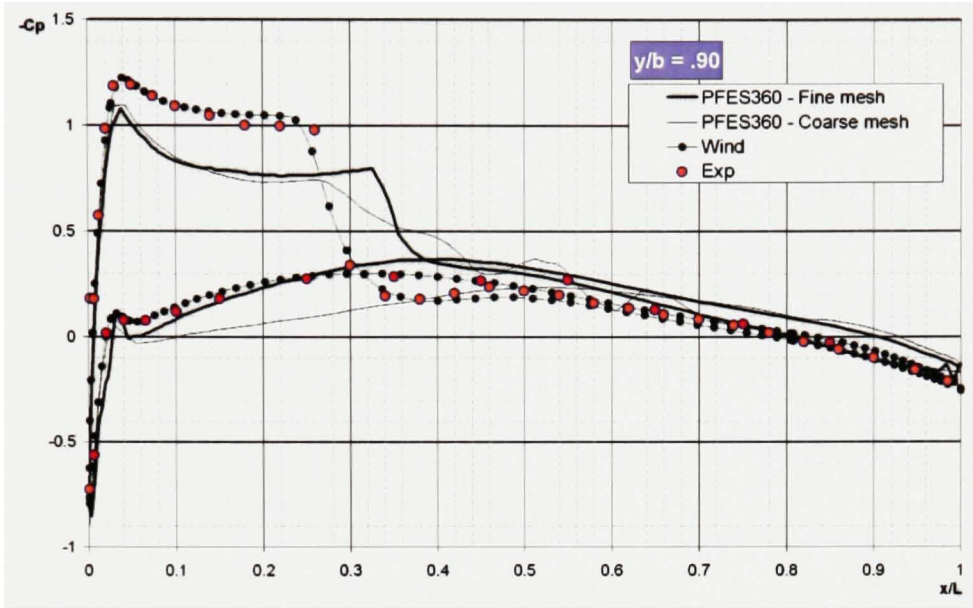


Figure 6.28 *Onera M6* : $-C_p$ à $y/b = 0.90$.

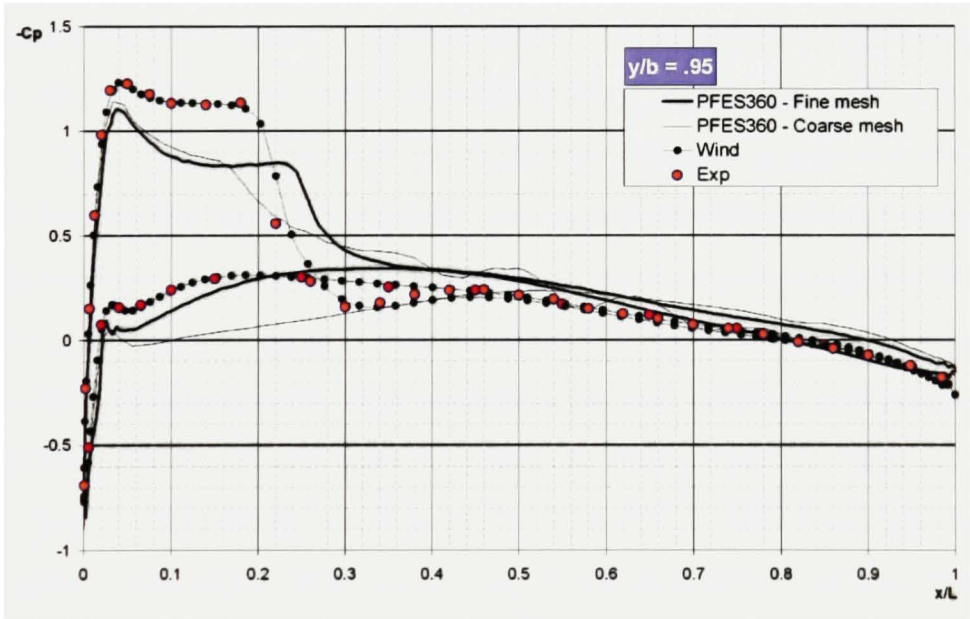


Figure 6.29 *Onera M6* : $-C_p$ à $y/b = 0.95$.

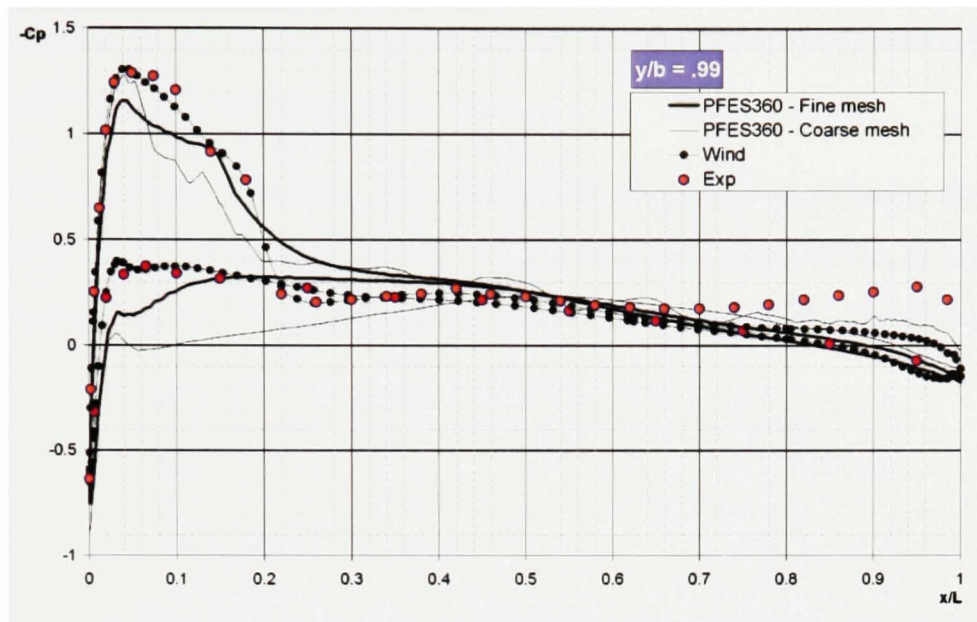


Figure 6.30 *Onera M6* : $-C_p$ à $y/b = 0.99$.

Les résultats montrent une bonne concordance entre les valeurs de coefficient de pression enregistrées par PFES360 et les valeurs expérimentales, en particulier sur les sections $y/b = 0.20$ et $y/b = 0.44$. Proche de la paroi, les résultats sont plus précis que ceux du code Wind et ceci pour les deux maillages. On constate une dépendance de la précision de la solution à la qualité du maillage. Le choc est évidemment mieux capturé avec le maillage fin. En effet, ce dernier est beaucoup plus dense, en particulier, dans la zone des chocs au bord d'attaque et au milieu de l'aile (figure 6.18). Dans ces régions la valeurs de y^+ sont proches de 1. On remarque que le raffinement a été appliqué seulement sur le dessus de l'aile. D'ailleurs, les valeurs de C_p sur le dessous ne sont pas précises. Proche de bout d'aile (tip) les résultats sont moins précis. Cette imprécision peut être expliquée par le fait que la zone de raffinement ne suit pas correctement la position des chocs. Dans ces régions la valeur de y^+ peut être trop élevée. Aussi, on remarque l'existence de quelques oscillations. La solution n'est pas complètement convergée au zéro de la machine. D'ailleurs le résidu calculé par le solveur GMRES stagne autour d'une valeur de 10^{-5} . Une amélioration du préconditionneur devrait

résoudre ce problème. Cependant, dans le cadre de ce travail nous nous sommes intéressés plus particulièrement à l'aspect stabilité.

Une étude plus poussée a été menée pour déterminer la cause de la diffusion de la solution numérique. La viscosité artificielle ajoutée par la méthode de stabilisation, la pertinence du modèle de turbulence et l'effet de la qualité du maillage sont des causes possibles.

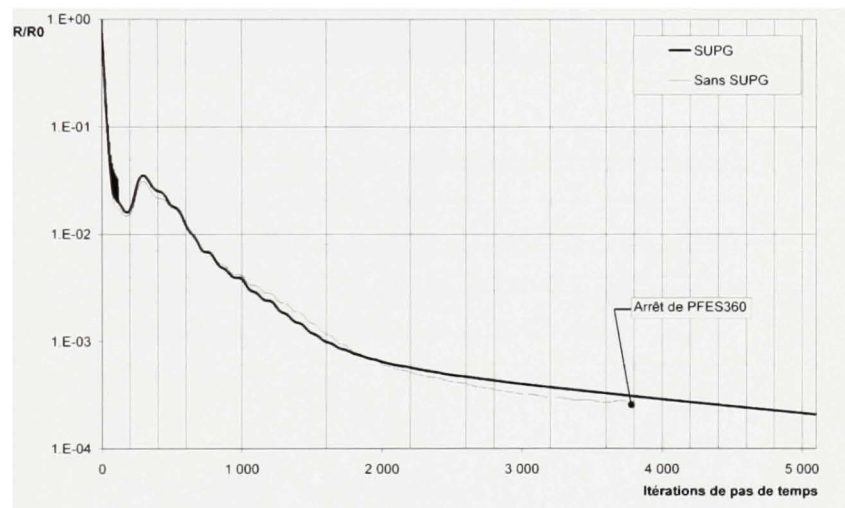


Figure 6.31 Onera M6 : Importance de la méthode SUPG.

La figure 6.31 montre qu'en absence de la méthode SUPG le code génère des pressions négatives à certains points et cause l'arrêt des calculs. La comparaison des pressions calculées avec SUPG ou sans SUPG au pas 3500 montre que la méthode SUPG n'a pas introduit trop de diffusion tout en améliorant le conditionnement du système. Il est à noter aussi qu'avec la stratégie de résolution adoptée, la convergence de l'algorithme GMRES prend de trois à quatre directions par pas de temps. Ce nombre de directions est trois fois plus petit que celui enregistré avec les anciennes définitions. Les figures 6.32 et 6.33 illustrent que la solution en régime laminaire varie selon le nombre de Reynolds. Ces résultats montrent que la viscosité artificielle ajoutée par l'introduction du terme de stabilisation et de capture de choc affecte localement la solution. En effet, Cette influence est clairement localisée au niveau des chocs.

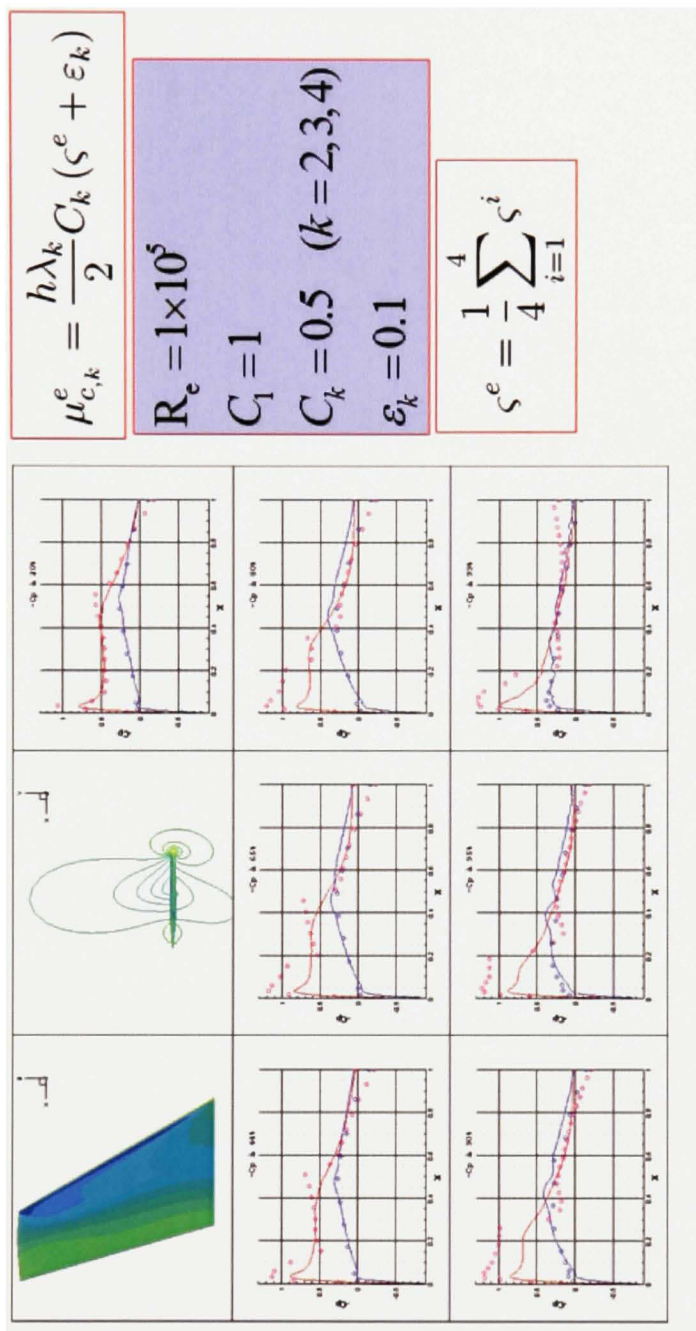


Figure 6.32 *Onera M6 : régime laminaire*, $R_e = 10^5$.

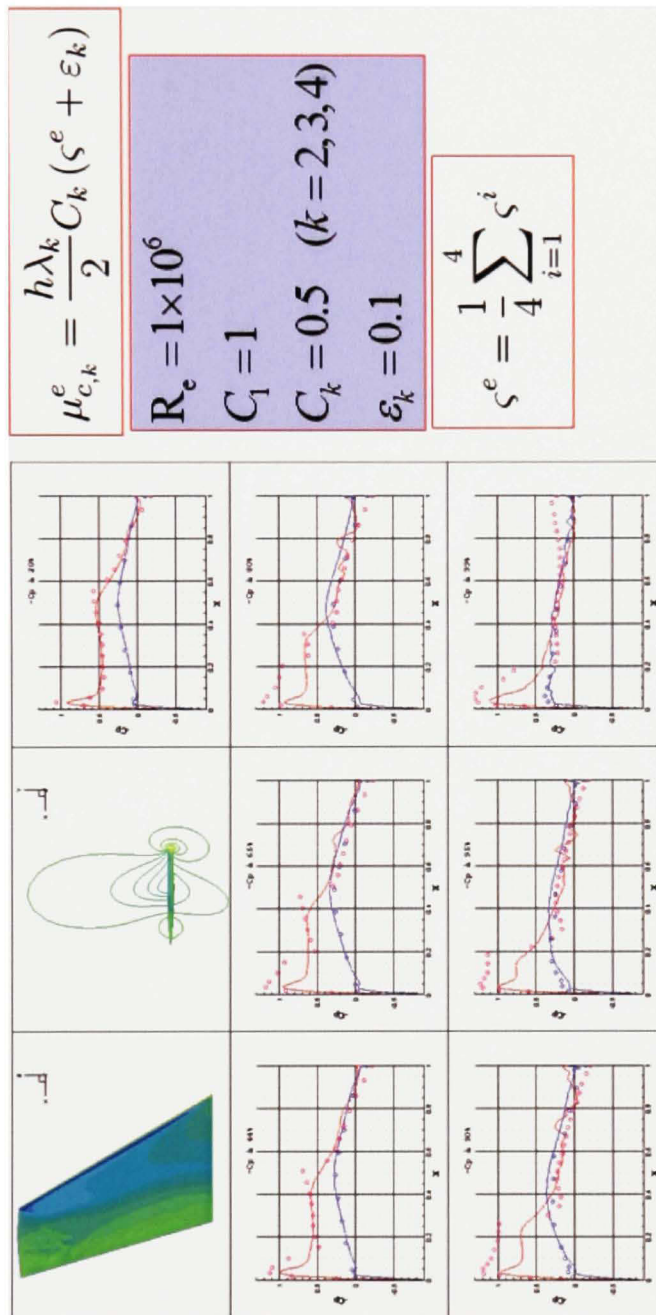


Figure 6.33 *Onera M6 : régime laminaire, $R_e = 10^6$.*

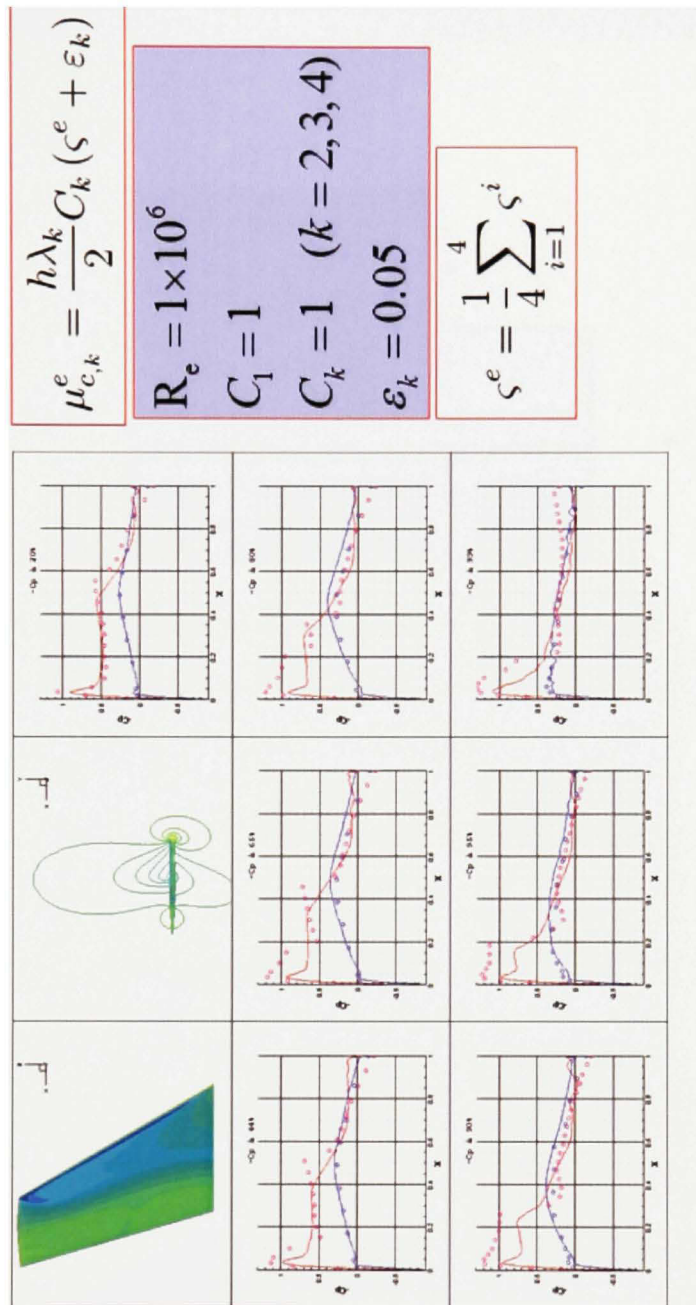


Figure 6.34 *Onera M6 : régime laminaire*, $R_e = 10^6$, $\varepsilon_k = 0.05$.

La figure 6.35 montre que la manière de calculer le senser de pression élémentaire est très importante.

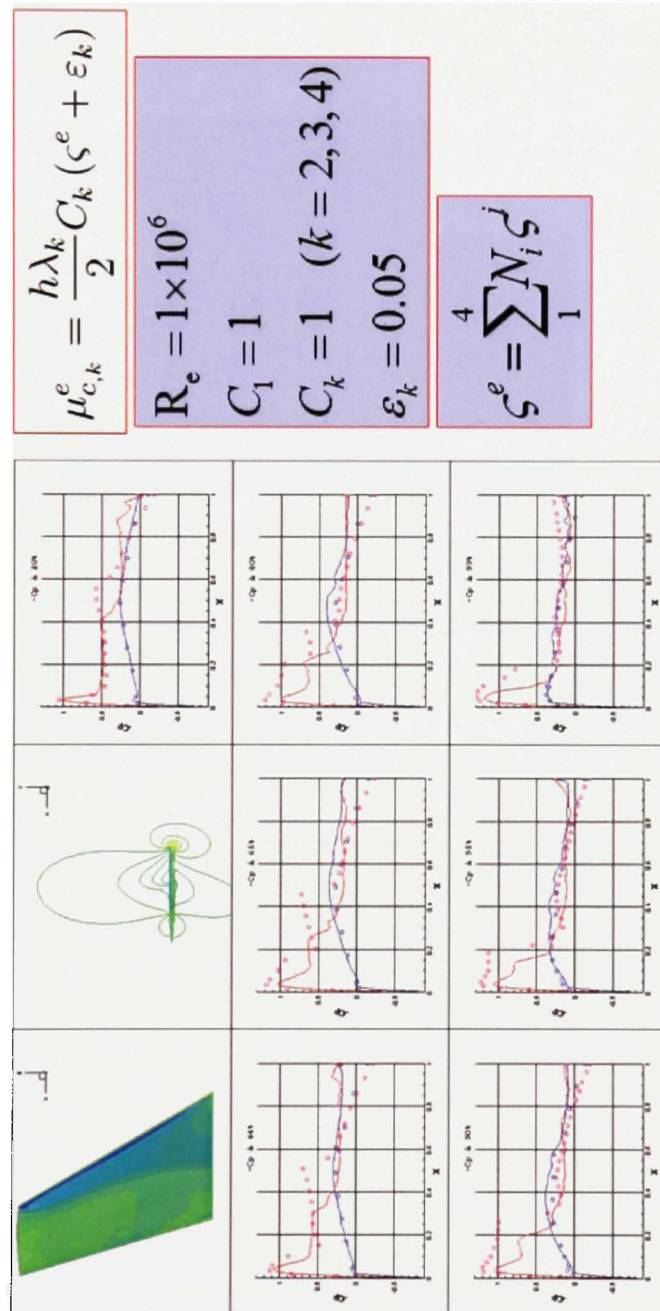


Figure 6.35 *Onera M6 : régime laminaire avec ζ^e pondéré.*

La figure 6.36 illustre l'évolution de la solution une fois le module de turbulence est activé. On remarque que le choc est maintenant bien localisé sans toutefois affecter l'amplitude des courbes de C_p .

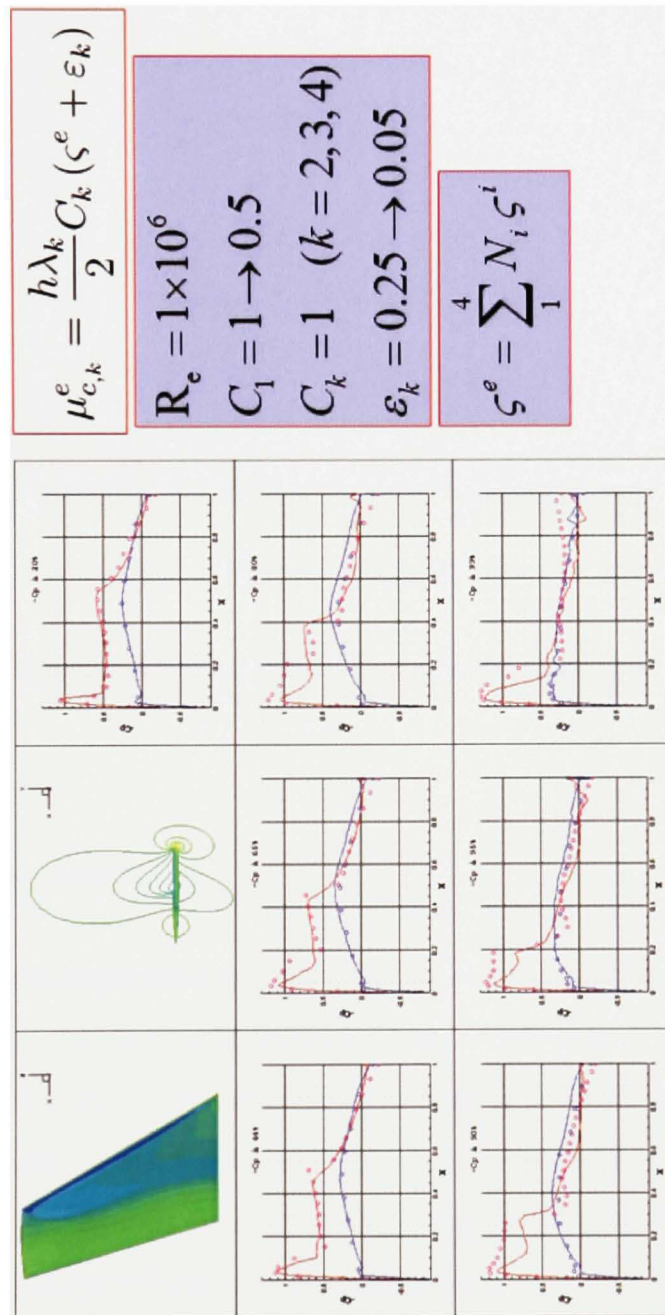


Figure 6.36 *Onera M6 : régime turbulent avec ζ^e pondéré.*

La figure 6.37 montre la courbe de convergence de PFES360 dans le cas de l'Onera M6. Le résidu est réduit à quatre ordres de grandeur en 750 pas de temps et de cinq ordres de grandeurs en 1550 pas.

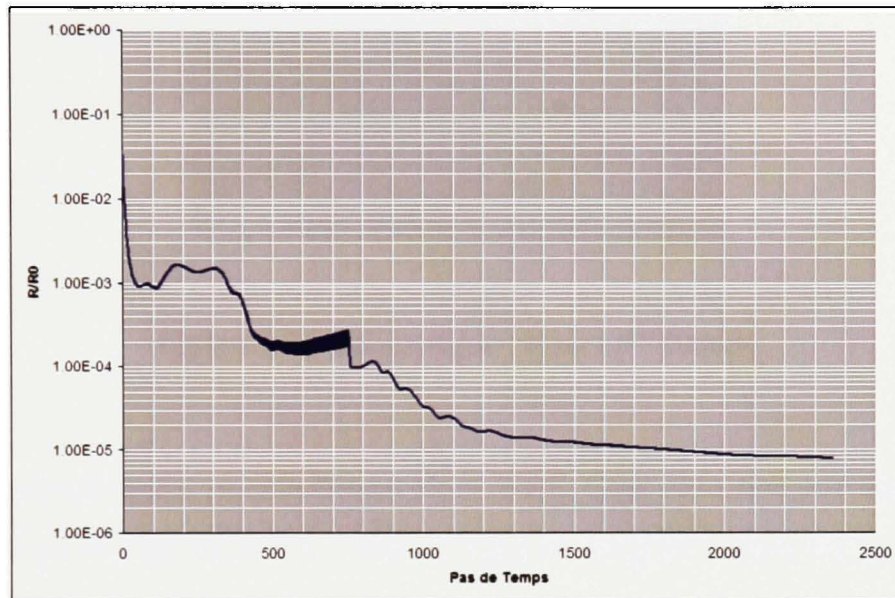


Figure 6.37 Onera M6 : Courbes de convergence.

6.5 Étude des performances

Une étude classique des performances (figure 6.39) a été effectuée dans (Ben Haj Ali, 2002; Soulaïmani et Ben Haj Ali, 2003; Soulaïmani et al., 2004) pour mesurer le "speedup" du code PFES. Cette étude a mis en évidence l'efficacité de la stratégie du parallélisme adoptée. Ces tests ont été conduits sur une machine composée d'un ensemble d'ordinateurs personnels connectés entre eux. Chaque unité est dotée d'un processeur Pentium IV (1700Mhz), 512Ko de mémoire cache et 500Mo (800Mhz) de mémoire vive.

La courbe de "speedup" (figure 6.39) montre une accélération supérieure à l'idéal. Une raison possible de cette accélération superlinéaire est l'effet de la mémoire cache. En effet, avec l'augmentation du nombre d'unités la quantité de données traitée par processeur décroît. Le



Figure 6.38 *La machine “beowulf” Granit.*

temps d'accès aux données essentielles est alors réduit. Cette réduction provoque une accélération supplémentaire qui s'ajoute à l'accélération due au parallélisme. En plus, l'utilisation d'un nombre réduit de processeurs engendre l'échauffement de ces derniers. L'augmentation de la température affecte leurs performances.

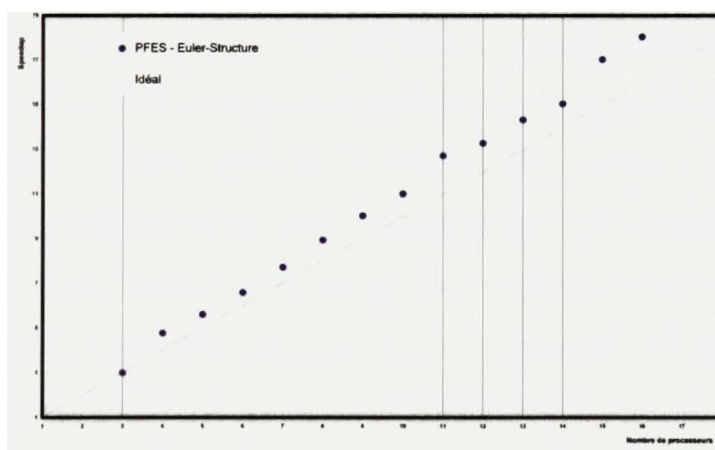


Figure 6.39 *Speedup de PFES dans le cas de couplage fluide–structure.*

Une récente étude de performance a été effectuée pour mesurer l'accélération du code PFES360 pour un plus grand nombre de processeurs. Cette étude concerne le module de fluide (Navier-Stokes) isolé. Le maillage utilisé est le maillage fin. Compte tenu de la mémoire nécessaire

pour cette simulation un nombre minimale de vingt et un (21) processeurs est utilisé. Le Speedup est calculé par rapport au temps de calcul sur 21 processeurs.

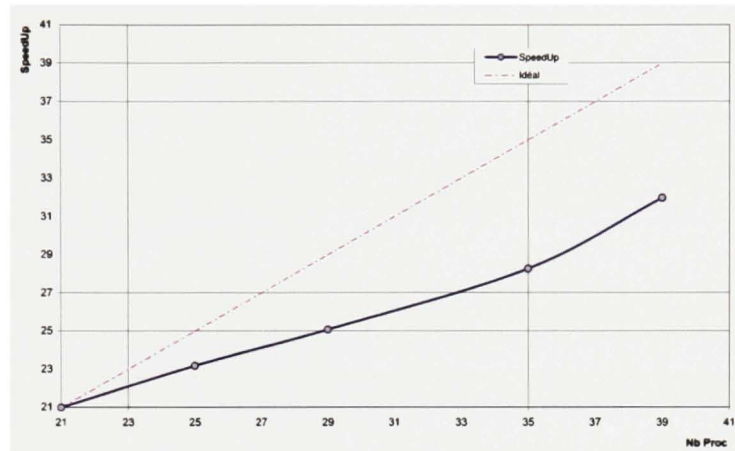


Figure 6.40 *Speedup de PFES360.*

La simulation du problème d'écoulement turbulent sur l'aile Onera M6 demande une grande capacité en mémoire. La résolution de ce problème sur notre machine ne peut se faire séquentiellement. Toutefois, dans le cadre de ce travail nous nous intéressons à un autre aspect d'étude de performance. En fait, l'approche de décomposition spatio-temporelle utilisée dans le code PFES360 exige une attribution d'un nombre de processeurs pour chaque module. Le choix de ce nombre peut affecter dramatiquement les performances du code. Dans la suite, nous présentons une étude de performance basée sur l'optimisation de la distribution des processeurs entre le module de fluide et le module de turbulence. L'étude consiste à utiliser toutes les ressources disponibles, c'est à dire les quarante processeurs, pour résoudre le problème d'écoulement turbulent sur l'aile Onera M6. Durant cette étude le nombre de pas de temps est limité à cent. La simulation est répétée plusieurs fois en variant le nombre de processeurs par module. En plus, cette opération est répétée pour chaque maillage.

La figure 6.41 illustre le temps de calcul réel (cpu + communication) mesuré pour chaque combinaison. Les courbes montrent que le rapport optimal entre le nombre de processeurs

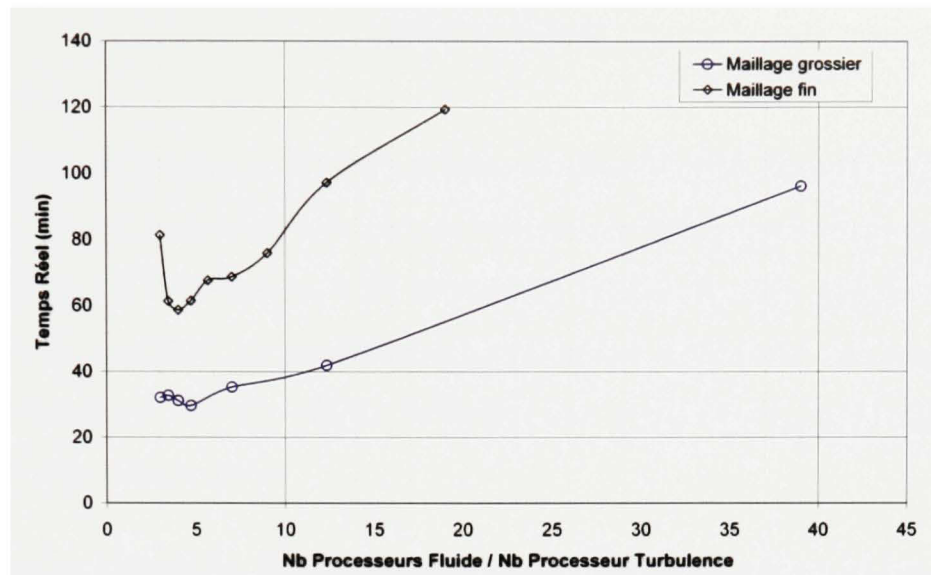


Figure 6.41 Onera M6 : Temps de calcul réel.

attribués au module de fluide et le nombre de processeurs attribués au module de turbulence est de l'ordre de cinq pour les deux maillages.

Notre stratégie d'optimisation consiste à attribuer 4.5 (81%) fois plus de processeurs au module de fluide. Ensuite, à rectifier cette estimation en se basant sur les intervalles mesurés pendant les premiers pas de temps. La valeur de 4.5 (au lieu de 5) est expliquée par le fait que dans PFES360, le module de turbulence a besoin d'un peu plus de temps pour la phase d'approximation du terme source. Le reste de l'algorithme est à peu près similaire, seule la différence entre le nombre d'inconnues par nœuds est considérée. Signalons à ce stade que les performances sont peu sensibles à des petites variations dans le choix du rapport. Aussi, le nombre de processeurs est un entier. Une petite variation dans le choix du rapport aboutit dans la majorité des cas à la même répartition. Ces résultats seront publiés dans (Ben Haj Ali et Soulaïmani, soumis 2008).

CONCLUSION

Un code de calcul parallèle, basé sur l'ancienne version de PFES (Ben Haj Ali, 2002) a été développé. Il permet la simulation de divers problèmes multiphysiques. Écrit en langage Fortran90 et utilisant la bibliothèque MPI, la nouvelle version du code, appelée PFES360, se distingue par sa simplicité, sa portabilité et sa versatilité. Grâce à son approche modulaire basée sur le paradigme MPMD–SPMD, PFES360 offre une extensibilité à l'intégration d'une multitude de nouveaux modules.

Dans le cadre de cette thèse, quatre modules permettant la simulation des écoulements turbulents tridimensionnels et l'interaction fluide–structure ont été utilisés. Ces modules traitent les domaines fluide, structure, mouvement de maillage et turbulence. Plusieurs tests de validations sur des géométries simples et complexes ont été conduits avec succès.

La simulation des écoulements autour des ailes aéroélastiques fait appel à trois modules soit le module fluide, le module structure et le module de mouvement de maillage.

En ce qui concerne le module fluide, la méthode de résolution est basée sur une discrétisation éléments finis des équations d'Euler. Dans ce cas, une stabilisation par la méthode SUPG standard s'est avérée suffisante. Quant au module de structure, les déplacements sont calculés à partir des équations d'équilibre dynamique. Le mouvement de maillage est défini par les équations d'élasticité. Ces deux derniers modules ont été développés dans le cadre d'une autre thèse au sein du groupe Granit (Feng, 2005).

Une étude aéroélastique sur l'aile Agard 445.6 a montré que le code reproduit avec précision les résultats expérimentaux de Yates (Yates, 1987).

Un des objectifs principaux de ce travail de recherche est la simulation des écoulements turbulents tridimensionnels. Afin d'arriver à cette fin, un module de calcul de la viscosité turbulente a été développé. Ce dernier communique avec le module fluide pour la résolution du système

d'équations formé par la discrétisation éléments finis des équations de Navier-Stokes moyennées (RANS).

Le module de turbulence repose sur le modèle de Spalart-Allmaras. Deux approches d'implémentation du modèle sont proposées. Ces approches concernent le traitement de chaque terme de l'équation de transport et en particulier le terme source (Lorin et al., 2007). La méthode adoptée par PFES360 est basée sur une discrétisation purement éléments finis où le terme source est calculé par une approximation semi-implicite. Une formulation standard de Galerkin avec un terme de diffusion numérique ajouté est utilisée. Il en résulte un schéma de discrétisation de premier ordre en h mais qui maintient, par expérience, la positivité de χ . Des tests de validations sur une plaque plane avec une vitesse imposée, et ensuite, avec la vitesse fournie par le module fluide ont montré que la résolution de l'équation de turbulence par la méthode de Galerkin toute seule (sans SUPG ni viscosité artificielle) ne présente aucun problème d'oscillations ou de convergence.

La modélisation numérique des écoulements turbulents nécessite l'emploi d'un maillage dont le premier point se trouve à une distance très proche de la paroi solide. Le recours aux maillages structurés, aux maillages hybrides et aux méthodes d'approximations telles que les lois de paroi a été exclu dès le départ. Notre objectif est d'octroyer au code PFES360 une stabilité numérique lui permettant de résoudre le problème considéré en utilisant un maillage purement tétraédrique extrêmement aplati.

Des simulations préliminaires de l'écoulement sur l'Onera M6 ont montré que la méthode SUPG standard est incapable de bien estimer la quantité de viscosité artificielle nécessaire pour la stabilisation de la solution. De plus, des problèmes de convergence de la résolution du système non linéaire sont apparus. Des nouvelles définitions de la matrice τ et de l'opérateur de capture de choc ont été élaborées afin de surmonter ce problème. L'introduction d'un nouveau senseur de variation de pression, inspiré de l'opérateur de capture de choc de Jameson, a stabilisé la solution dans les zones d'oscillations et de chocs sans toutefois la diffuser.

Les nouvelles définitions et les méthodes développées ont été appliquées au problème d'écoulement turbulent sur l'Onera M6. Les résultats ont montré une bonne concordance avec les valeurs expérimentales. L'utilisation d'un maillage adapté améliore la précision de la solution. Une comparaison entre les résultats trouvés avec l'emploi de la méthode SUPG (dotée de la nouvelle définition de la matrice τ) et les résultats trouvés sans SUPG, montrent que les nouvelles définitions apportent un meilleur conditionnement au système.

Le présent travail nous a permis de constater que la réussite de ce genre de simulations dépend de la méthode de stabilisation, de la pertinence du modèle de turbulence et essentiellement de la qualité du maillage (Rebaine, 1997; Rumsy et Wahls, 2008). En plus, il est important d'assurer la positivité de la viscosité turbulente pour éviter les problèmes de convergence.

La dernière partie de cette thèse a été consacrée aux mesures des performances du code PFES360. Une étude classique reposant sur le calcul de l'accélération "Speedup" a mis en évidence l'efficacité de la stratégie de parallélisation adoptée. Une accélération superlinéaire a été observée. Elle peut être expliquée par l'effet de la mémoire cache. Une deuxième étude nous a permis d'établir une procédure d'optimisation de la distribution des processeurs entre le module de fluide et le module de turbulence. Cette procédure consiste à attribuer 19% des processeurs au module de turbulence et d'ajuster la répartition au cours de la simulation.

Perspectives pour de futures recherches :

À partir de ce travail, des champs de recherches futures peuvent être suggérés.

- Simplifier la procédure d'importation des maillages en intégrant les interfaces de conversion dans PFES360. Ainsi, il sera possible de lire les fichiers de données générées par les codes commerciaux.
- Reproduire les résultats de l'Onera M6 en utilisant un maillage mieux adapté. Une procédure de raffinement dans les zones de chocs, au bord de fuite et au bout de l'aile devra améliorer les résultats.

- Accélérer la procédure de décomposition géométrique en remplaçant la version actuelle de Metis par la version parallèle.
- Améliorer la convergence du solveur GMRES en développant de nouveaux préconditionneurs.
- Automatiser la procédure d'optimisation de la répartition des processeurs.
- Valider le code pour le cas d'une configuration aile-fuselage.
- Coupler les modules de fluide, turbulence, structure et mouvement de maillage pour la résolution d'un problème d'aéroélasticité en régime turbulent.

ANNEXE 1

PARALLÉLISME

Historique¹ de MPI :

MPI (Message Passing Interface), conçue en 1993-94, est une norme définissant une bibliothèque de fonctions, utilisable avec les langages C et Fortran. Elle permet d'exploiter des ordinateurs distants ou multiprocesseur par passage de messages.

MPI a été écrit pour obtenir de bonnes performances aussi bien sur des machines massivement parallèles à mémoire partagée que sur des clusters d'ordinateurs hétérogènes à mémoire distribuée. Il est grandement disponible sur de très nombreux matériels et systèmes d'exploitation.

MPI possède l'avantage par rapport aux plus vieilles bibliothèques de passage de messages d'être grandement portable (car MPI a été implanté sur presque toutes les architectures de mémoires) et rapide (car chaque implantation a été optimisée pour le matériel sur lequel il s'exécute).

MPI-2 une nouvelle version de MPI, est disponible depuis 1997.

La bibliothèque MPI offre plusieurs fonctions qui permettent d'établir la communication entre les processeurs. Les messages échangés sont interprétés et gérés par un environnement qui peut être comparé à la téléphonie ou encore à une messagerie électronique.

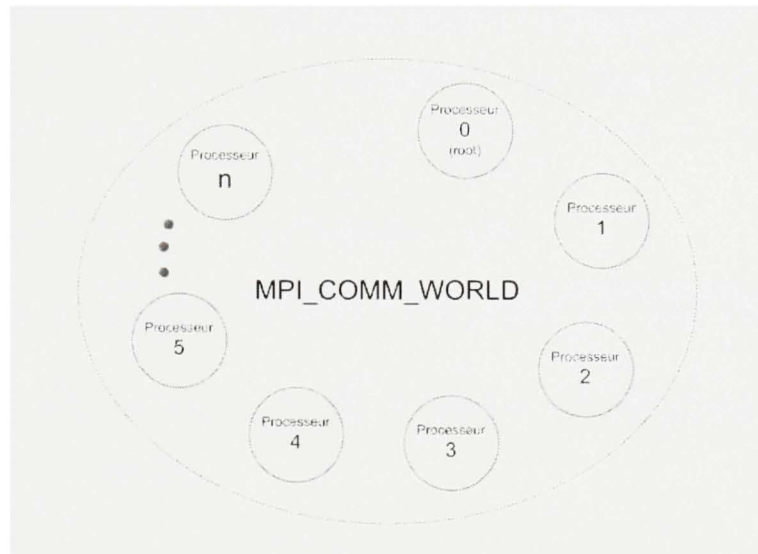
Dans cette section, les fonctions les plus utilisées de la bibliothèque MPI sont illustrées. Rappelons que d'après notre expérience, les fonctions de communication collective sont plus stables que les fonctions de communication point à point.

Ces fonctions peuvent être classées en trois catégories :

1 Fonctions d'environnement

¹Wikipédia

La fonction, `MPI_INIT()` permet d'initialiser l'environnement nécessaire. Entre autres, elle permet la création de `MPI_COMM_WORLD` qui est le communicateur par défaut de MPI (figure 1). Le communicateur est l'environnement dans lequel les processus communiquent entre eux.



Le communicateur `MPI_COMM_WORLD`.

```
integer, intent(out) :: ierr
call MPI_INIT(ierr)
```

Réciproquement, la fonction `MPI_FINALIZE()` désactive cet environnement.

```
integer, intent(out) :: ierr
call MPI_FINALIZE(ierr)
```

La fonction `MPI_COMM_RANK()` permet d'obtenir le rang d'un processus.

```
integer, intent(out) :: rang, ierr
call MPI_COMM_RANK(MPI_COMM_WORLD, rang, ierr)
```

La fonction `MPI_Comm_size()` permet d'obtenir le nombre de processus affectés à un communicateur. Par exemple pour le communicateur `MPI_COMM_WORLD()` :


```
integer, intent(out) :: iproc, ierr
MPI_COMM_SIZE (MPI_COMM_WORLD, iproc, ierr)
```

La fonction `MPI_COMM_SPLIT()` permet de créer un communicateur `NewComm`, basé sur les couleurs, à partir d'un communicateur existant (ex. `MPI_COMM_WORLD`).

```
integer, intent(in) :: color(*), key(*)
integer, intent(out) :: NewComm, ieer
MPI_COMM_SPLIT (MPI_COMM_WORLD, color, key, NewComm, ierr)
```

La fonction `MPI_Comm_group()` Permet d'obtenir la liste des groupes `Groups` présents dans un Communicateur donné.

```
integer, intent(out) :: Groups, ierr
MPI_Comm_group (MPI_COMM_WORLD, Groups, ierr)
```

La fonction `MPI_GROUP_INCL()` permet de créer un groupe `NewGroup` à partir d'un autre.

```
integer, intent(in) :: OldGroup, Nb_Proc, ranks(*)
integer, intent(out) :: NewGroup, ieer
MPI_GROUP_INCL (OldGroup, Nb_Proc, ranks, NewGroup, ieer)
```

La fonction `MPI_COMM_CREATE()` permet de créer un communicateur `NewComm` au sein d'un groupe de processeurs.

```
integer, intent(in) :: OldComm, procs(*)
integer, intent(out) :: NewComm, ieer
MPI_COMM_CREATE (OldComm, procs, NewComm, ierr)
```

2 Barrière de synchronisation

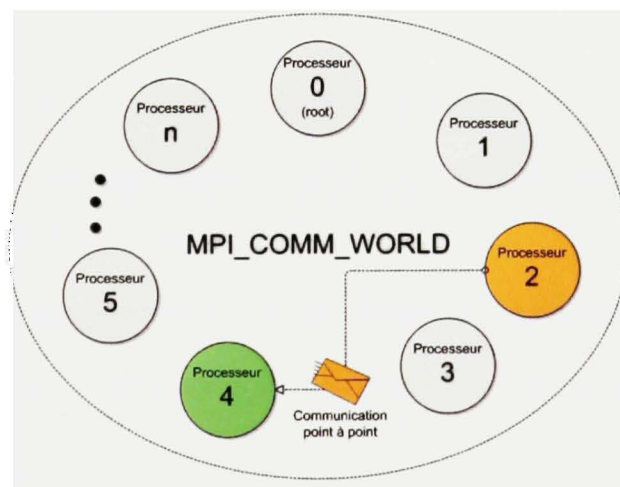
Pour synchroniser des tâches, il est possible de se servir de barrières :

```
integer, intent(out) :: ierr
MPI_BARRIER(MPI_COMM_WORLD, ierr)
```

3 Fonctions de communication point à point

Une communication est dite « point à point » si elle s'effectue entre deux processus, l'un appelé processus émetteur et l'autre processus récepteur (figure 3). l'enveloppe d'un message est constituée de :

- le rang du processus émetteur
- le rang du processus récepteur
- une étiquette de message
- le nom du communicateur



MPI : Communication « point à point ».

Les données envoyées sont typées : MPI_CHAR, MPI_SHORT, MPI_INT, MPI_LONG, MPI_UNSIGNED_CHAR, MPI_FLOAT, MPI_UNSIGNED,

MPI_UNSIGNED_LONG, MPI_UNSIGNED_SHORT, MPI_DOUBLE,
MPI_LONG_DOUBLE.

Plusieurs variantes de communication point à point peuvent être établies, toutefois le principe reste le même ; Un processeur envoie, l'autre reçoit.

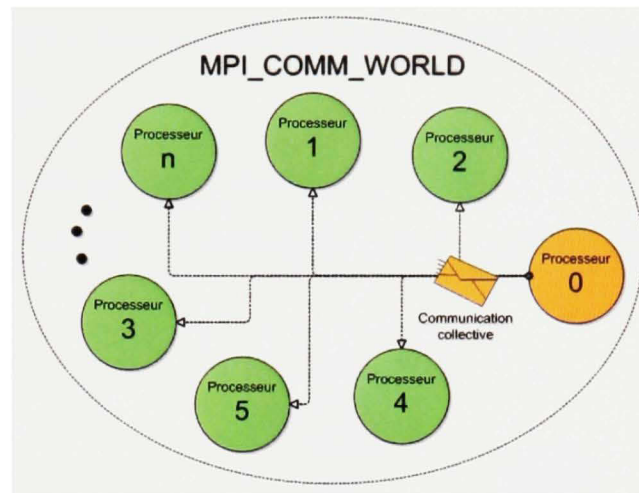
- Synchrones / asynchrones
- Bloquantes / non-bloquantes

Les références (Snir et al., 1996; Gropp et al., 1999) donnent une explication complète de ces fonctions. Cependant, d'après notre expérience, ce schéma de communication n'est pas du tout conseillé. Dans PFES seulement les communications collectives sont utilisées.

4 Fonctions de communication collectives

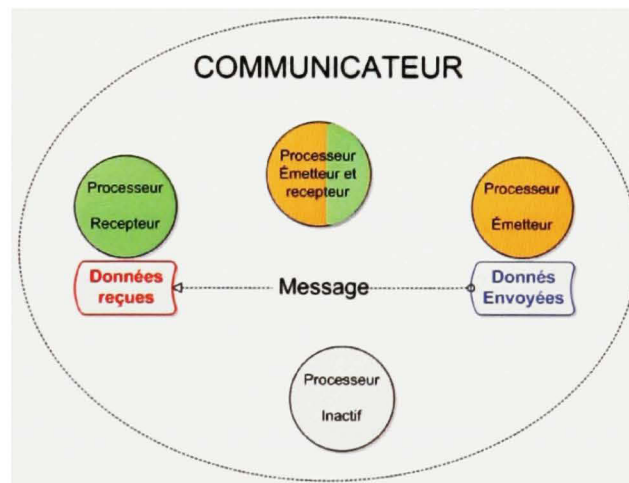
En plus de leur stabilité par rapport aux fonctions de communication point à point, les fonctions de communication collective (figure 4) s'avèrent, parfois, plus intuitives pour :

- Partager la valeur d'une variable connue par seulement un processeur
- Faire une réduction (somme, max...) de données distribuées sur les différents processus et rapatrier cette valeur sur un processus
- Distribuer les valeurs contenues dans un tableau sur un processus vers tous les autres
- Récupérer des données distribuées dans un tableau sur un processus ...



MPI : Communication Collective.

Tout au long de cette partie, le code couleur présenté par la figure 4 sera utilisé pour simplifier l'illustration des différentes fonctions de communication collective.



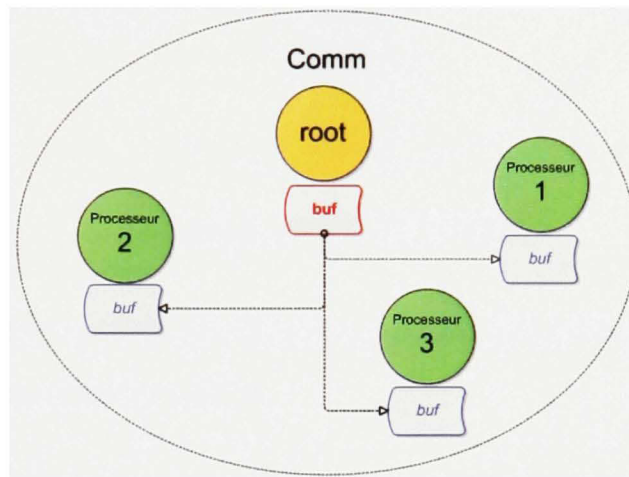
MPI : légende des couleurs et notations.

Les principales fonctions utilisées sont :

- broadcast : pour envoyer une information d'une tâche vers toutes les autres.
- Reduce : une information sur une tâche est réduite vers une autre tâche.
- scatter : répartit un vecteur d'une tâche entre les autres processeurs.
- gather : rassemble les informations de toutes les tâches vers une seule.

La fonction `MPI_BCAST` permet à un processus d'envoyer un message à tous les autres processus du communicateur.

```
integer, intent(out) :: ierr
MPI_BCAST(buf, count, datatype, root, comm, ierr)
```



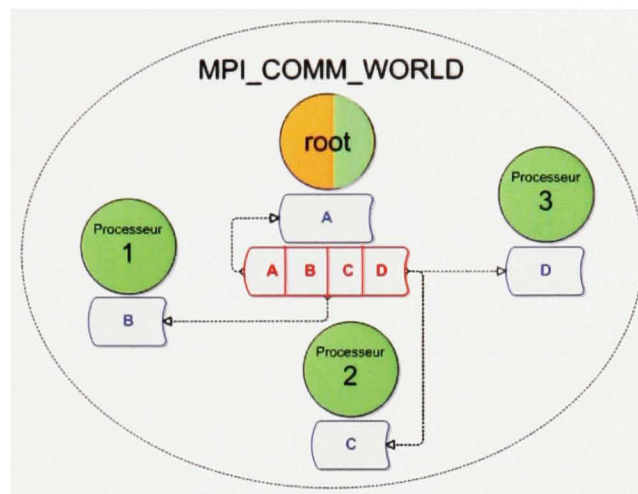
La fonction MPI_BCAST.

La fonction `MPI_SCATTER` permet à un processus de partager un tableau entre tous les autres processus du communicateur

```
integer, intent(out) :: ierr
MPI_SCATTER(sendbuf, sendcount, sendtype, recvbuf, recvcount, &
            recvtype, root, comm, ierr)
```

Dans la plupart des cas, `sendcount=recvcount` et `sendtype=recvtype`.

Cette fonction fait le partage de `sendbuf` (`sendcount` éléments de type `sendtype`) du processus `root` entre les autres tâches. Le processus 0 recevra la première tranche de `sendcount` éléments, le processus 1 recevra la deuxième, etc. Le résultat est stocké dans `recvbuf`.



La fonction MPI_SCATTER.

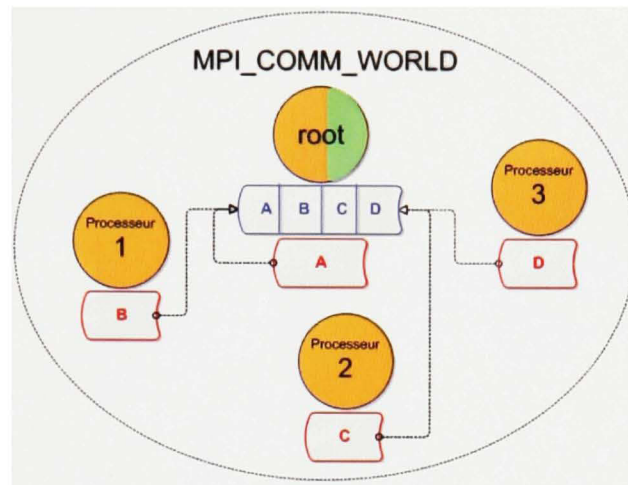
La fonction MPI_GATHER permet de collecter des messages provenant de tous les processus et d'en stocker le contenu dans un tableau.

```
integer, intent(out) :: ier
MPI_GATHER(sendbuf, sendcount, sendtype, recvbuf, &
           recvcount, recvtype, root, comm, ierr)
```

Le processus root reçoit dans recvbuf les tables de données sendbuf, de dimension sendcount éléments chacune, envoyées par chaque tâche. C'est la procédure exactement inverse à MPI_SCATTER. Encore une fois, dans la plupart des cas, sendcount=recvcount et sendtype=recvtype.

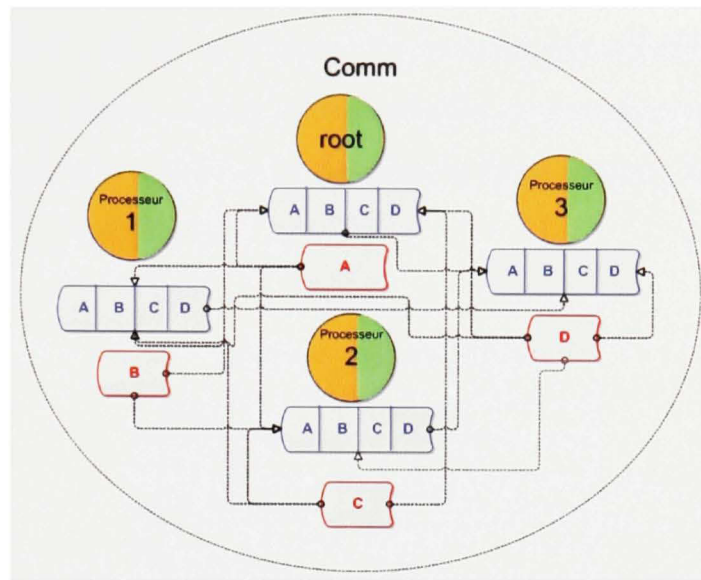
La fonction MPI_ALLGATHER fait la même opération que MPI_GATHER mais le résultat recvbuf est envoyé vers tous les processus.

```
integer, intent(out) :: ier
MPI_ALLGATHER(sendbuf, sendcount, sendtype, recvbuf, &
              recvcount, recvtype, comm, ierr)
```



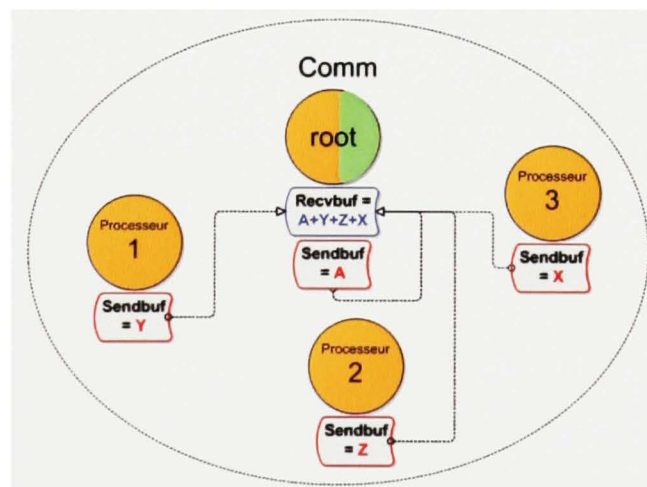
La fonction MPI_GATHER.

Pour communiquer des tables de tailles différentes et pour contrôler la façon dont les données seront stockées dans `recvbuf`, `MPI_GATHERV` et `MPI_SCATTERV` peuvent être utilisées. Pour plus de détails voir les références (Snir et al., 1996; Gropp et al., 1999).



La fonction MPI_Allgather.

Enfin, la fonction MPI_REDUCE permet de collecter des messages provenant de tous les processus et d'effectuer une opération sur leur contenu avant de stocker le résultat de cette opération.



La fonction MPI_Reduce.

```
integer, intent(out) :: ier
MPI_REDUCE(sendbuf, recvbuf, count, datatype, &
           op, root, comm, ierr)
```

La variable `root` est le rang de la tâche qui reçoit la synthèse des données via l'opérateur `op`. `op` peut en principe être n'importe quelle opération commutative et associative. Pour que toutes les tâches reçoivent la synthèse des données, la fonction `MPI_ALLREDUCE`, peut être utilisée. Voici la liste des opérateurs :

Opérations logiques de MPI

Nom	Définition
<code>MPI_MAX</code>	Maximum
<code>MPI_MIN</code>	Minimum
<code>MPI_SUM</code>	Somme
<code>MPI_PROD</code>	Produit
<code>MPI_LAND</code>	.AND. logique
<code>MPI_BAND</code>	.AND. bit à bit
<code>MPI_LOR</code>	.OR. logique
<code>MPI_BOR</code>	.OR. bit à bit
<code>MPI_LXOR</code>	.XOR. logique
<code>MPI_BXOR</code>	.XOR. bit à bit
<code>MPI_MAXLOC</code>	maximum et indice
<code>MPI_MINLOC</code>	minimum et indice

BIBLIOGRAPHIE

- Aupoix, B. et Spalart, P. (2003) : «*Extensions of the Spalart - Allmaras turbulence model to account for wall roughness*», International Journal of Heat and Fluid Flow, tome 24(4) :454 , ISSN 0142-727X.
- Auriti, L. et DeLaurier, J. (2004) : «*Analysis of the flight attempt by samuel langley's "great aerodrome"*», Journal of Aircraft, tome 41(6) :1430 , ISSN 0021-8669.
- Azami, Y. (2001) : «*Realisation d'un environnement de traitement parallèle à partir d'un regroupement d'ordinateurs personnels*», Thèse de maîtrise, École de technologie supérieure, Montréal Canada.
- Ben Haj Ali, A. (2002) : «*Calcul distribué pour des problèmes multiphysiques*», Thèse de maîtrise, École de technologie supérieure, Montréal Canada.
- Ben Haj Ali, A., Lorin, E., et Soulaïmani, A. (2004) : «*Positive scheme for the turbulent Spalart-Allmaras model theory and applications*», European Congress on Computational Methods in Applied Sciences and Engineering.
- Ben Haj Ali, A. et Soulaïmani, A. (soumis 2008) : «*Unstructured finite elements for the rans equations coupled with the Spalart-Allmaras turbulence model*», Computer Methods in Applied Mechanics and Engineering.
- Bouzouiki, M. (2002) : «*Developpement et étude des éléments finis spectraux pour les écoulements rapides*», Thèse de maîtrise, École de technologie supérieure, Montréal Canada.
- Brooks, A. N. et Hughes, T. J. R. (1994) : «*Streamline upwind/petrov-galerkin methods for advection dominated flows*», Transport in Porous Media, tome 16(1) :53, ISSN 0169-3913 (Print) 1573-1634 (Online).
- Chi, R. (2005) : «*Navier-stokes computations with one-equation turbulence model for flows along concave wall surfaces*», Rapport technique.
- Cousteix, J. (1989) : *Turbulence et couche limite*, Cepadues-Editions, 1st édition.

- Deck, S., Duveau, P., D'Espiney, P., et Guillen, P. (2002) : «*Development and application of spalart-allmaras one equation turbulence model to three-dimensional supersonic complex configurations*», Aerospace Science and Technology, tome 6(3) :171 , ISSN 1270-9638.
- Durbin, P. (1995) : «*Separated flow computations with the $k - \varepsilon - v^2$ model*», AIAA Journal, tome 33(4) :659 , ISSN 0001-1452.
- El Kadri El Yamani, N.-E. (1992) : «*Modélisation des écoulements compressibles visques par la méthode des éléments finis*», Thèse de maîtrise, Université de Laval, Québec Canada.
- El Kadri El Yamani, N.-E. (1995) : «*Une méthode d'éléments finis pour la dynamique des gaz et conception orientée objet du code de calcul*», Thèse de doctorat, Université de Laval, Québec Canada.
- Farhat, C. (1995) : «*High performance simulation of coupled nonlinear transient aeroelastic problems*», Special Course on Parallel Computing in CFD (AGARD-R-807), (p. 8 – 1).
- Farhat, C., Geuzaine, P., et Brown, G. (2002) : «*Application of a three-field nonlinear fluid-structure formulation to the prediction of the aeroelastic parameters of an f-16 fighter*», Computers and Fluids, tome 32(1) :3 , ISSN 0045-7930.
- Farhat, C. et Lesoinne, M. (1996) : «*On the accuracy, stability, and performance of the solution of three-dimensional nonlinear transient aeroelastic problems by partitioned procedures*», Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference, tome 1 :629 , ISSN 0273-4508.
- Farhat, C. et Lesoinne, M. (2000) : «*Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems*», Computer Methods in Applied Mechanics and Engineering, tome 182(3-4) :499.
- Farhat, C., van der Zee, K. G., et Geuzaine, P. (2006) : «*Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity*», Computer Methods in Applied Mechanics and Engineering, tome 195(17-18) :1973 , ISSN 0045-7825.
- Feng, Z. (2005) : «*A nonlinear computational aeroelasticity model for aircraft wings*», Thèse de doctorat, École de technologie supérieure, Montréal Québec Canada.

- Fortin, M. (1991) : «*Simulation d'écoulement compressible non-visqueux et visqueux par les méthodes de fractionnement des flux*», Thèse de doctorat, Sherbrooke, Québec Canada.
- Geuzaine, P., Brown, G., Harris, C., et Farhat, C. (2003) : «*Aeroelastic dynamic analysis of a full f-16 configuration for various flight conditions*», AIAA Journal, tome 41(3) :363 , ISSN 0001-1452.
- Ghidaglia, J. M. et Pascal, F. (1999) : «*Passerelles volumes finis - éléments finis*», C.R. Acad. Sci. Paris, tome 328 :711.
- Gropp, W., Lusk, E., et Skjellum, A. (1999) : «*Using MPI, portable parallel programming with the message-passing interface*», MIT Press.
- Gupta, K. (1996) : «*Development of a finite element aeroelastic analysis capability*», Journal of Aircraft, tome 33(5) :995 , ISSN 0021-8669.
- Hinze, J. O. (1975) : *Turbulence*, McGraw Hill College Div, 2nd édition.
- Hodges, H., Dewey et Pierce, G., Alvin (2002) : *Introduction to Structural Dynamics and Aeroelasticity*, ISBN 0-521-80698-4.
- Hutin, P. M. (1998) : «*State of the art and open problems in aeroelasticity*», Office national d'études et de recherches aérospatiales, ISSN 0369-7681.
- Jakirlic, S., Eisfeld, B., Jester-zurker, R., et Kroll, N. (2007) : «*Near-wall, reynolds-stress model calculations of transonic flow configurations relevant to aircraft aerodynamics*», International Journal of Heat and Fluid Flow, tome 28(4) :602 , ISSN 0142-727X.
- Jameson, A. et Mavriplis, D. (1986) : «*Finite volume solution of the two-dimensional euler equations on a regular triangular mesh*», AIAA Journal, tome 24(4) :611.
- Jones, W. P. et Launder, B. E. (1972) : «*The prediction of laminarization with a two-equation model of turbulence*», International Journal of Heat and Mass Transfer, tome 15(2) :301 , ISSN 0017-9310.
- Kim, B. C., Raman, A., et Mote JR., C. D. (2000) : «*Prediction of aeroelastic flutter in a hard disk drive*», Journal of Sound and vibration, tome 238(2) :309.

- Kolmogorov, A. N. (1942) : «*Equations of turbulent motion of an incompressible fluid*», Izv Akad Nauk SSR Ser Phys, tome 1-2(6) :56.
- Landmann, B., Kessler, M., Wagner, S., et Kramer, E. (2008) : «*A parallel, high-order discontinuous galerkin code for laminar and turbulent flows*», Computers and Fluids, tome 37(4) :427 , ISSN 0045-7930.
- Launder, B. E. et Sharma (1974) : «*Application of the energy-dissipation model or turbulence to the calculation of flow near a spinning disc*», Letters in Heat and Mass Transfer, tome 1(2) :131 , ISSN 0094-4548.
- Lee, J. et Paynter, G. (1996) : «*Modified spalart-allmaras one-equation turbulence model for rough wall boundary layers*», Journal of Propulsion and Power, tome 12(4) :809 , ISSN 0748-4658.
- Lee-Rauch, E. M. et Batina, J. T. (1993) : «*Calculation of agard wing 445.6 flutter using navier-stokes aerodynamics*», Rapport technique.
- Leopold, C. (2000) : *Parallel and distributed computing, A survey of models, paradigms, and approaches*, John Wiley and Sons, Inc., 1st édition.
- Löner, R., Morgan, K., et Peraire, J. (1995) : «*A simple extention to multidimensional problems of the artificial viscosity due to lapidus*», tome 1, (p. 141 – 147).
- Lorin, E., Ben Haj Ali, A., et Soulaïmani, A. (2006) : «*An accurate positivity preserving scheme for the Spalart-Allmaras turbulence model. application to aerodynamics*», tome 4, (p. 2270 – 2301), Collection of Technical Papers - 36th AIAA Fluid Dynamics Conference, San Francisco, CA, United States.
- Lorin, E., Ben Haj Ali, A., et Soulaïmani, A. (2007) : «*A positivity preserving finite element-finite volume solver for the Spalart-Allmaras turbulence model*», Computer Methods in Applied Mechanics and Engineering, tome 196(17-20) :2097 , ISSN 0045-7825.
URL <http://dx.doi.org/10.1016/j.cma.2006.10.009>
- Lucquin, B. et Pironneau, O. (1996) : *Introduction au Calcul Scientifique*, Masson, 1st édition.
- Mallet, M. (1985) : «*A finite element method for computational fluid dynamics*», Thèse de doctorat, Stanford University.

- Mavriplis, D. J. (2007) : «*Discrete adjoint-based approach for optimization problems on three-dimensional unstructured meshes*», AIAA Journal, tome 45(4) :740 , ISSN 0001-1452.
- Mittal, R., Simmons, S., et Udaykumar, H. (2001) : «*Application of large-eddy simulation to the study of pulsatile flow in a modeled arterial stenosis*», Journal of Biomechanical Engineering, tome 123(4) :325 , ISSN 0148-0731.
- Paciorri, R., Dieudonnandé, W., Degrez, G., Charbonnier, J.-M., et Deconinck, H. (1998) : «*Exploring the validity of the Spalart-Allmaras turbulence model for hypersonic flows*», tome 35, (p. 121–126), AIAA-1997-2023, Fluid Dynamics Conference, 28th, Snowmass Village, CO.
- Paciorri, R. et Sabetta, F. (2003) : «*Compressibility correction for the spalart-allmaras model in free-shear flows*», Journal of Spacecraft and Rockets, tome 40(3) :326 , ISSN 0022-4650.
- Pascal, F. et Ghidaglia, J. M. (2001) : «*Footbridge between finite volumes and finite elements with applications to CFD*», International Journal for Numerical Methods in Fluids, tome 37(8) :951 , ISSN 0271-2091.
- Peterson, W. et Zing, W. (2008) : «*Three-dimensional aerodynamic computations on unstructured grids using a Newton–Krylov approach*», Computers & Fluids, tome 37 :107 .
- Quarteroni, A. et Valli, A. (1999) : *Domain Decomposition Methods for Partial Differential Equations*, ISBN 0198501781 / 978-0198501787.
- Rebaine, A. (1997) : «*Simulation d'écoulements internes compressibles laminaires et turbulents par une méthode d'éléments finis*», Thèse de doctorat, École de technologie supérieure, Sherbrooke Québec Canada.
- Rebaine, A. et Soulaïmani, A. (2001) : «*Numerical simulation of two-dimensional compressible turbulent flows in ejectors*», Transactions of the Canadian Society for Mechanical Engineering, tome 25(2) :227 , ISSN 0315-8977.
- Rifai, S., Buell, J., Johan, Z., et Hughes, T. (2000) : «*Automotive design applications of fluid flow simulation on parallel computing platforms*», Computer Methods in Applied Mechanics and Engineering, tome 184 :449 .

- Rispoli, F., Corsini, A., et Tezduyar, T. E. (2007) : «*Finite element computation of turbulent flows with the discontinuity-capturing directional dissipation (dcdd)*», Computers and Fluids, tome 36(1) :121 , ISSN 0045-7930.
- Robert, M., Bennett, et John, W. E. (1998) : «*An overview of recent developments in computational aeroelasticity*», 29th AIAA Fluid Dynamics Conference, tome 98-2421.
- Roth, S.-N. (2007) : «*Méthode de calcul numérique rapide en aéroélasticité non-linéaire basée sur l'équation du potentiel complet*», Thèse de maîtrise, École de technologie supérieure, Québec Canada.
- Roy, C. et Blottner, F. (2003) : «*Methodology for turbulence model validation : Application to hypersonic flows*», Journal of Spacecraft and Rockets, tome 40(3) :313 , ISSN 0022-4650.
- Rumsy, C. L. et Wahls, C. L. (2008) : «*Focused assessment of state-of-the-art CFD capabilities for prediction of subsonic fixed wing aircraft aerodynamics*», Rapport technique, NASA Langley Research Center Hampton, VA 23681-2199.
- Rung, T., Bunge, U., Schatz, M., et Thiele, F. (2003) : «*Restatement of the spalart-allmaras eddy-viscosity model in strain-adaptive formulation*», AIAA Journal, tome 41(7) :1396 , ISSN 0001-1452.
- Saad, Y. (1994) : «*SPARSKIT : A basic tool kit for sparse matrix computations*», Rapport technique, Minneapolis, MN 55455.
- Saad, Y. (2003) : *Iterative methods for sparse linear systems*, Society for Industrial and Applied Mathematics, 2st édition, ISBN 0898715342.
- Saad, Y., Kuznetsov, S., Gen-Ching, L., Malevsky, A., et Chapman, A. (1997) : «*Psparslib : a portable library of parallel sparse iterative solvers*», (p. 8), Proceedings Eighth SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, MN, USA.
- Saad, Y. et Malevsky, A. (1995) : «*Psparslib : A portable library of distributed memory sparse iterative solvers*», proceedings of Parallel Computing Technologies (PaCT-95), tome 3-rd International Conference.

- Saad, Y. et Schultz, M. H. () : «*GMRES : A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems*, publisher = SIAM, year = 1986, journal = SIAM Journal on Scientific and Statistical Computing, volume = 7, number = 3, pages = 856-869», .
- Sathe, S., Benney, R., Charles, R., Doucette, E., Milette, J., Senga, M., Stein, K., et Tezduyar, T. (2007) : «*Fluid–structure interaction modeling of complex parachute designs with the space–time finite element techniques*», Computers & Fluids, tome 36 :127 .
- Schmitt, V. et Charpin, F. (1979) : «*Pressure distributions on the onera-m6-wing at transonic mach numbers experimental data base for computer program assessment*», Rapport technique.
- Shakib, F. (1989) : «*Finite element analysis of the compressible euler and navier-stokes equations*», Thèse de doctorat, Stanford University, CA.
- Slater, J. (2008) : «*Cfd verification and validation*», .
URL
[http ://www.grc.nasa.gov/WWW/wind/valid/m6wing/m6wing.html](http://www.grc.nasa.gov/WWW/wind/valid/m6wing/m6wing.html)
- Snir, M., Otto, S., Huss-Lederman, S., Walker, D., et Dongarra, J. (1996) : «*MPI : The complete reference*», .
- Soulaïmani, A. et Ben Haj Ali, A. (2003) : «*Calcul parallèle-distribué pour les problèmes multiphysiques : application à l'aéroélasticité*», Revue Européenne de Mécanique Numérique European Journal of Computational Mechanics (anciennement Revue Européenne des Éléments Finis (Issn : 1250-6559), Hermes Science publications, tome 12(7-8) :793, ISSN 1779-7179.
- Soulaïmani, A., Ben Haj Ali, A., et Feng, Z. (2004) : «*A parallel-distributed approach for multi-physics problems with application to computational nonlinear aeroelasticity*», Canadian Aeronautics and Space Journal, tome 50(4) :221 , ISSN 0008-2821.
- Soulaïmani, A., Ben Salah, N., et Saad, Y. (2002a) : «*Enhanced GMRES Acceleration Techniques for some CFD problems*», International Journal of Computational Fluid Dynamics, tome 16 :1.
- Soulaïmani, A., BenElhaj, A., et Feng, Z. (2002b) : «*A distributed computing-based methodology for computational nonlinear aeroelasticity*», tome 40th Conference, Reno, NV, (p. 14–17), AIAA Aerospace Sciences Meeting and Exhibit.

- Soulaïmani, A., BenElHajAli, A., et Feng, Z. (2002c) : «*Nonlinear computational aeroelasticity : Formulations and solution algorithms*», tome RTO-MP-089, (p. 45–01 45–13), NATO-AVT, Meeting Proceedings, ISBN 92-837-0027-9.
- Soulaïmani, A., El Yamani, N., et C., D. (1994) : «*Une méthode d'éléments finis pour le calcul des écoulements compressibles utilisant les variables conservatives et la méthode supg*, revue européennes des éléments finis», Revue Européennes des Eléments finis, tome 3(2) :211 .
- Soulaïmani, A., Forest, A., Feng, Z., BenElhaj, A., et Azami, Y. (2001a) : «*A distributed computing-based methodology for computational nonlinear aeroelasticity*», tome 48th annual Conference, (p. 123–135), Canadian Aeronautics and Space Institute.
- Soulaïmani, A. et Fortin, M. (1994) : «*Finite element solution of compressible viscous flows using conservative variables*», Computer Methods in Applied Mechanics and Engineering, tome 118(3-4) :319 , ISSN 0045-7825.
- Soulaïmani, A. et Saad, Y. (1998) : «*An arbitrary lagrangian-eulerian finite element method for solving three-dimensional free surface flows*», Computer Methods in Applied Mechanics and Engineering, tome 162(1-4) :79 , ISSN 0045-7825.
- Soulaïmani, A., Saad, Y., et Rebain, A. (2001b) : «*An edge based stabilized finite element method for solving compressible flows : Formulation and parallel implementation*», Computer Methods in Applied Mechanics and Engineering, tome 190(49-50) :6735 , ISSN 0045-7825.
- Soulaïmani, A., Saad, Y., et Rebaine, A. (2000) : «*Parallelization of the edge based stabilized finite element method using PPARSLIB*», Parallel computational fluid dynamics, towards teraflops, optimization and Novel Formulations.
- Soulaïmani, A., Wong, T., Azami, Y., et BenElHajAli, A. (2001c) : «*An object-oriented approach for Pc clusters*», dans «IPDPS '01 : Proceedings of the 15th International Parallel & Distributed Processing Symposium», (p. 178), IEEE Computer Society, Washington, DC, USA, ISBN 0-7695-0990-8.
- Spalart, P. et Allmaras, S. (1994) : «*One-equation turbulence model for aerodynamic flows*», Recherche Aerospatiale, (1) :5 , ISSN 0034-1223.

- Spalart, P., Strelets, W.-H., et Allmaras, S. (1997) : «*Comments on the feasibility of les for wings, and on a hybrid RANS/LES approach*», tome DNS/LES, Aug 408, Greyden Press.
- Speziale, C. G., Abid, R., et Anderson, E. C. (1992) : «*Critical evaluation of two-equation models for near-wall turbulence*», AIAA Journal, tome 30(2) :324 , ISSN 0001-1452.
- Squires, K., Forsyth, J., Morton, S., Blake, D., Serrano, M., Wurtzler, K., Strang, W., et Tomaro, P., R.F. and Spalart (2002) : «*Analysis of full aircraft with massive separation using detached-eddy simulation*», Proceedings of the High Performance Computing Modernization.
- Svacek, P. (2008) : «*Application of finite element method in aeroelasticity*», Journal of Computational and Applied Mathematics, tome 215(2) :586 , ISSN 0377-0427.
- Tang, L., Bartels, R., Chen, P.-C., et Liu, D. (2003) : «*Numerical investigation of transonic limit cycle oscillations of a two-dimensional supercritical wing*», Journal of Fluids and Structures, tome 17(1) :29 , ISSN 0889-9746.
- Tezduyar, T. et Senga, M. (2006) : «*Stabilization and shock-capturing parameters in sup formulation of compressible flows*», Computer Methods in Applied Mechanics and Engineering, tome 195(13-16) :1621 , ISSN 0045-7825.
- Tezduyar, T. E. (2007) : «*Finite elements in fluids : Stabilized formulations and moving boundaries and interfaces*», Computers and Fluids, tome 36(2) :191 , ISSN 0045-7930.
- Tucker, P. G. (2007) : «*Introduction. computational aerodynamics*», Computational aerodynamics, Philosophical Transactions of The Royal Society A, tome 365(1859) :2379, ISSN 1364-503X (Paper) 1471-2962 (Online).
- Van Driest, E. R. (1956) : «*On turbulent flow near a wall*», Journal of the Aeronautical Sciences, tome 23 :1007, ISSN 0001-1452.
- Washio, T., Hisada, T., Watanabe, H., et Tezduyar, T. (2005) : «*A robust preconditioner for fluid-structure interaction problems*», Computer Methods in Applied Mechanics and Engineering, tome 194(39-41) :4027 , ISSN 0045-7825.
- White, F. M. (1991) : *Viscous fluid flow*, McGraw-Hill, 1st édition.

- Wilcox, D. C. (1997) : *Turbulence modeling for CFD*, DCW Industries, 2nd édition.
- Xiao, Z., Chen, H., ZhangY, ., Huang, J., et Fu, S. (2006) : «*Study of delayed-detached eddy simulation with weakly nonlinear turbulence model*», Journal of Aircraft, tome 43(5) :1377 , ISSN 0021-8669.
- Yang, Z. et Mavriplis, D. J. (2007) : «*Higher-order time-integration schemes for aeroelastic applications on unstructured meshes*», AIAA Journal, tome 45(1) :138 , ISSN 0001-1452.
- Yates, J., E. C. (1987) : «*Agard standard aeroelastic configuration for dynamics response, candidat configuration i.-wing 445.6*», NASA TM 100492.
- Zhang, Z., Gil, A., Hassan, O., et Morgan, K. (2008) : «*The simulation of 3d unsteady incompressible flows with moving boundaries on unstructured meshes*», Computers and Fluids, tome 37(5) :620 , ISSN 0045-7930.
- Zhao, Y. et Forhad, A. (2003) : «*A general method for simulation of fluid flows with moving and compliant boundaries on unstructured grids*», Computer Methods in Applied Mechanics and Engineering, tome 192(39-40) :4439 , ISSN 0045-7825.