



HAL
open science

Etude de schémas numériques d'ordre élevé pour la simulation de dispersion de polluants dans des géométries complexes

Julien Montagnier

► **To cite this version:**

Julien Montagnier. Etude de schémas numériques d'ordre élevé pour la simulation de dispersion de polluants dans des géométries complexes. Autre [cond-mat.other]. Université Claude Bernard - Lyon I, 2010. Français. NNT : 2010LYO10118 . tel-00842988

HAL Id: tel-00842988

<https://theses.hal.science/tel-00842988>

Submitted on 10 Jul 2013

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

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

N° d'ordre 118-2010

Année 2010

THESE DE L'UNIVERSITE LYON

Délivrée par

L'UNIVERSITE CLAUDE BERNARD LYON 1

ECOLE DOCTORALE : MEGA

LMFA UMR CNRS 5509

DIPLOME DE DOCTORAT

(arrêté du 7 aout 2006)

soutenue publiquement en juillet 2010

par

Julien MONTAGNIER

Etude de schémas numériques d'ordre élevé pour la simulation de
dispersion de polluants dans des géométries complexes



Jury :

LERICHE E., Professeur, Université Jean Monnet, Saint-Etienne (Rapporteur)

VISONNEAU M., Directeur de Recherche, Ecole Centrale de Nantes (Rapporteur)

BUFFAT M., Professeur, Université Claude Bernard, Lyon I (Directeur de thèse)

DERVIEUX A., Directeur de Recherche, INRIA, Sophia Antipolis

KNIKKER R., Maître de Conférence, CETHIL, INSA

BOISSON N., Docteur, Ingénieur d'études INSAVALOR

FERNANDEZ G., Docteur, Ingénieur d'études PARALGO

Je tiens à remercier avant tout Marc Buffat pour avoir dirigé cette thèse. Sans sa patience, sa disponibilité et tous les efforts qu'il a su me consacrer, ce travail n'aurait pas abouti.

Je tiens vivement à remercier Michel Visonneau et Emmanuel Leriche pour avoir accepté d'être rapporteur de cette thèse,

Mes remerciements vont par ailleurs à l'ensemble des membres du jury, et à Alain Dervieux et son équipe pour m'avoir accueilli chaleureusement quelques jours à l'INRIA à Sofia Antipolis.

Enfin je remercie le personnel du LMFA que j'ai cotoyé, en particulier Christophe Pera pour sa disponibilité, toute l'équipe de Modelys, Ariane, Maxime, Jonathan et David, pour les bons moments passés ensemble, et enfin Nicolas Boisson et Gérard Fernandez pour m'avoir offert l'opportunité de faire cette thèse.

Résumé

La prévention des risques industriels nécessite de simuler la dispersion turbulente de polluants. Cependant, les outils majoritairement utilisés à ce jour ne permettent pas de traiter les champs proches dans le cas de géométries complexes, et il est nécessaire d'utiliser les outils de CFD ("Computational Fluid Dynamics") plus adaptés, mais plus coûteux. Afin de simuler les écoulements atmosphériques avec dispersion de polluants, les modèles CFD doivent modéliser correctement d'une part, les effets de flottabilité, et d'autre part les effets de la turbulence. Plusieurs approches existent, notamment dans la prise en compte des effets de flottabilité et la modélisation de la turbulence, et nécessitent des méthodes numériques adaptées aux spécificités mathématiques de chacune d'entre elles, ainsi que des schémas numériques précis pour ne pas polluer la modélisation.

Une formulation d'ordre élevé en volumes finis, sur maillages non structurés, parallélisée, est proposée pour simuler les écoulements atmosphériques avec dispersion de polluants. L'utilisation de schémas d'ordre élevé doit permettre d'une part de réduire le nombre de cellules et diminuer les temps de simulation pour atteindre une précision donnée, et d'autre part de mieux contrôler la viscosité numérique des schémas en vue de simulations LES (Large Eddy Simulation), pour lesquelles la viscosité numérique des schémas peut masquer les effets de la modélisation.

Deux schémas d'ordre élevé ont été étudiés et implémentés dans un solveur 3D Navier Stokes incompressible sur des maillages volumes finis non structurés. Nous avons développé un premier schéma d'ordre élevé, correspondant à un schéma Padé volumes finis, et nous avons étendu le schéma de reconstruction polynomiale de Carpentier (2000) aux écoulements incompressibles. Les propriétés numériques des différents schémas implémentés dans le même code de calcul sont étudiées sur différents cas tests bi-dimensionnels (calcul de flux convectifs et diffusifs sur une solution a-priori, convection d'une tâche gaussienne, décroissance d'un vortex de Taylor et cavité entraînée) et tri-dimensionnel (écoulement autour d'un obstacle cubique). Une attention particulière a été portée à l'étude de la précision et du traitement des conditions limites. L'implémentation proposée du schéma polynomial permet d'approcher, pour un maillage identique, les temps de simulation obtenus avec un schéma décentré classique d'ordre 2, mais avec une précision supérieure. Le schéma compact donne la meilleure précision. En utilisant une méthode de Jacobi sans calcul implicite de la matrice pour calculer le gradient, le temps de simulation devient intéressant uniquement lorsque la précision requise est importante. Une alternative est la résolution du système linéaire par une méthode multigrille algébrique. Cette méthode diminue considérablement le temps de calcul du gradient et le schéma Padé devient performant même pour des maillages grossiers.

Enfin, pour réduire les temps de simulation, la parallélisation des schémas d'ordre élevé est réalisée par une décomposition en sous domaines. L'assemblage des flux s'effectue naturellement et différents solveurs proposés par les bibliothèques PETSC et HYPRE (solveur multigrille algébrique et méthode de Krylov préconditionnée) permettent de résoudre les systèmes linéaires issus de notre problème. Le travail réalisé a consisté à identifier et déterminer les paramètres de résolution qui conduisent aux temps de simulation les plus faibles. Différents tests de speed-up et de scale-up ont permis de déterminer la méthode la plus efficace et ses paramètres optimaux pour la résolution en parallèle des systèmes linéaires issus de notre problème.

Les résultats de ce travail ont fait l'objet d'une communication dans un congrès international "parallel CFD juin 2008" et d'un article soumis à "International Journal for Numerical Methods in Fluids" (Analysis of high-order finite volume schemes for the incompressible Navier Stokes equations)

Table des matières

Introduction	9
1 Modèles physiques	13
1.1 Équations de Navier Stokes et transport d'un polluant	13
1.2 Modélisation des effets de flottabilité	14
1.2.1 Approximation de Boussinesq	14
1.2.2 Approximation anélastique	15
1.2.3 Approximation faible nombre de Mach	16
1.3 Modélisation de la turbulence	17
1.3.1 Modèles RANS	18
1.3.2 Modèles LES	19
1.3.3 Modèles hybrides RANS/LES	20
1.4 Conclusion	20
2 Méthodes numériques	22
2.1 Méthode volumes finis et discrétisations	23
2.1.1 Différences finies	23
2.1.2 Éléments finis	23
2.1.3 Galerkin discontinu	23
2.1.4 Volumes finis	24
2.2 Solveur incompressible	24
2.2.1 Méthodes de projection	25
2.2.2 Méthodes de type SIMPLE	26
2.2.3 Compressibilité artificielle pour les écoulements stationnaires	28
2.3 Solveur compressible pour faible nombre de mach	29
2.3.1 Préconditionnement de Turkel	29
2.3.2 Méthode de projection basée sur l'équation d'énergie par Nerinckx	30
2.4 Conclusion	32
3 Schémas d'ordre élevé	34
3.1 Description des méthodes numériques	36
3.1.0.1 Reconstruction par développement en série de Taylor	37

3.1.0.2	Schéma à reconstruction polynomiale	37
3.1.0.2.1	Condition aux limites	40
3.1.0.3	Schéma Padé	40
3.1.1	Flux convectifs avec une vitesse non uniforme	44
3.2	Validation numérique	45
3.2.1	Calcul du Laplacien et le gradient sur une fonction a priori	45
3.2.2	Advection d'un cône gaussien dans un domaine carré	47
3.2.3	Décroissance visqueuse d'un tourbillon de Taylor	47
3.2.4	Cavité entraînée	48
3.2.4.1	Ordre de Convergence	51
3.2.4.2	Efficacité des différents schémas	52
3.2.5	Écoulement autour d'un obstacle cubique	53
3.2.5.1	Structure de l'écoulement	54
3.2.5.2	Points de séparation et de réattachement	55
3.3	Conclusion	55
4	Parallélisation	60
4.1	Structure de données et assemblage des systèmes linéaires	61
4.1.1	Partitionnement du maillage et structure de données	62
4.1.2	Découpage parallèle du système linéaire	62
4.1.3	Assemblage	63
4.2	Résolution des systèmes linéaires	64
4.2.1	Solveur de Krylov et méthode GMRES	65
4.2.1.1	Amélioration de GMRES	66
4.2.1.2	Parallélisation	68
4.2.2	Solveur multigrille	69
4.2.2.1	Algorithme de construction des niveaux de grilles	71
4.2.2.2	Opérateur d'interpolation	72
4.2.2.3	Procédure de lissage	73
4.2.3	Préconditionnement	73
4.2.3.1	Factorisation incomplète LU	74
4.2.3.1.1	Préconditionneur parallèle de type Schwarz	74
4.3	Évaluation des performances parallèles	76
4.3.1	Équation de Laplace	77
4.3.1.1	Validation de l'implémentation de l'assemblage parallèle	78
4.3.1.2	Méthode multigrille algébrique	78
4.3.1.3	Préconditionnement ILU	80
4.3.1.4	Préconditionnement de Schwartz	80
4.3.1.5	Scale up	84
4.3.2	Écoulement turbulent autour de l'usine de Feyzin	85
4.4	Conclusion	87

Conclusion	88
Annexe 1 : Algorithmes de construction de grilles pour solveur AMG	91
Annexe 2 : Opérateurs d'interpolation pour solveur AMG	95
Annexe 3 : Méthodes de préconditionnement ILU	98

Liste des Figures

2.1	Méthode volumes finis.	25
3.1	Agglomération du stencil.	38
3.2	Notation du maillage.	41
3.3	Maillages.	45
3.4	Résultats pour le calcul du Laplacien et du Gradient	46
3.5	Taux de convergence pour la convection du cone gaussien	47
3.6	Norme L_1 de l'erreur sur la vitesse pour l'écoulement de Taylor.	48
3.7	Ligne de courant et d'iso-vorticité pour la cavité entraînée $Re = 1000$	49
3.8	Comparaison des profils des vitesses horizontale et verticale pour la cavité entraînée	50
3.9	Erreur de la solution pour la cavité entraînée	54
3.10	Géométrie de l'obstacle cubique.	54
3.11	Lignes de courants situées à : $y^* = 0.04, 1/4, 1/2$ et $11/12$ (de haut en bas).	56
3.12	Lignes de courants situées à : $z^* = 0.04, 1/4, 5/2$ et $7/12$ (de haut en bas).	57
3.13	Iso-surface λ_2 obtenue avec le maillage M2 et le schéma d'ordre 2.	58
4.1	Exemple de partitionnement de maillage sur 3 processeurs.	63
4.2	Décomposition du système linéaire selon les processeurs.	63
4.3	Découpage parallèle de la matrice.	69
4.4	Cycle multigrille.	71
4.5	Algorithmes de sélection des grilles et schémas d'interpolation.	72
4.6	Décomposition en domaine avec recouvrement.	75
4.7	Partitionnement d'une boîte cubique non structuré.	77
4.8	Speed up sur une équation de Laplace.	78
4.9	Résultat de scale-up pour l'équation de Laplace.	85
4.10	Maillage du site industriel de Feyzin.	86
4.11	Panache.	86
4.12	Speed up du solveur Navier Stokes sur l'écoulement de Feyzin.	86
3.13	Parallélisation de ILU selon Hysom and Pothén (1999).	101

Liste des Tables

3.1	Puissance des monômes pour le cas 3D.	38
3.2	Puissance des monômes pour le cas 2D.	38
3.3	Coût mémoire pour le schéma polynomiale de degré 3.	40
3.4	Positivité moyenne du schéma polynomial pour les flux visqueux sur différents maillages.	51
3.5	Taux de convergence sur les maillages quadrangulaires pour la cavité entraînée.	51
3.6	Temps d'exécution des différents schémas pour la cavité entraînée.	53
3.7	Maillages utilisés pour l'obstacle cubique.	54
3.8	Distance des points de séparation x_S et recollement x_R à l'obstacle.	57
4.1	Influence du seuil pour l'équation de Laplace, 16 processeurs, 50 000 dof / proc, $\alpha = 0.7$	79
4.2	Influence du seuil pour l'équation de Laplace, 16 processeurs, 50 000 dof / proc, $\alpha = 1$	79
4.3	Comparaison des performances du solveur AMG et LGMRES préconditionné par AMG.	81
4.4	Comparaison des performances du solveur AMG et LGMRES préconditionné par AMG.	82
4.5	Préconditionnement ILU.	83
4.6	Préconditionnement de Schwartz	84
4.7	Nombre d'itérations avec 200 000 degré de liberté / processeur.	85
4.8	Détails du préconditionnement AMG avec 32 processeurs, 200 000 dofs.	85

Liste des algorithmes

1	SIMPLE.	27
2	Nerinckx sans transfert thermique.	32
3	Nerinckx avec transfert thermique.	33
4	Méthode de Jacobi pour schéma Padé.	42
5	Stratégie parallèle.	61
6	Méthode GMRES.	66
7	Méthode GMRES avec préconditionnement à droite.	67
8	Méthode GMRES avec préconditionnement variable.	67
9	Produit matrice vecteur parallèle.	69
10	Algorithme Multigrille.	70
11	Méthode multiplicative de Schwarz avec coloriage des domaines.	75
12	Méthode additive de Schwarz.	76
13	Algorithme de sélection des grilles CLJP	93
14	Algorithme originelle ILU dans sa version IKJ.	98
15	Algorithme ILU(p) dans sa version IKJ.	99
16	Algorithme ILUT.	100
17	Algorithme Parallèle ILU.	102

Liste des symboles

β	Coefficient d'expansion thermique
ϵ	Tau de dissipation
γ	Rapport de la capacité calorifique à pression constante sur la capacité calorifique à volume constant
Γ_{ij}	Face située entre les cellules i et j
λ	Conductivité thermique du fluide
\mathbf{g}	Vecteur gravité
\mathbf{n}_{ij}	Normale unitaire sortante de la face Γ_{ij}
\mathbf{q}	Transfert d'énergie par conduction
\mathbf{u}	Vitesse de l'écoulement
μ	Viscosité du fluide
ν	Viscosité cinématique du fluide
ν_t	Viscosité cinématique turbulente
Ω_{ij}	Tenseur de vorticit�
$\pi(\bar{z})$	Etat moyen de la fonction de Exner
ρ	Masse volumique
ρ_0	Masse volumique de r�f�rence
τ	Tenseur des contraintes
C_i	Cellule i
e	Energie totale du fluide
Fr_∞	Nombre de Froude
H	Taux de production de chaleur par unit� de volume
k	Energie cin�tique turbulente
M	Nombre de Mach

P	Pression du fluide
Pr_∞	Nombre de Prandtl
Re_∞	Nombre de Reynolds
S_{ij}	Surface de la face Γ_{ij}
S_{ij}	Tenseur des contraintes
T	Température du fluide
T_0	Température de référence
V_i	Volume de la cellule i
Y_i	Concentration massique du polluant

Introduction

Les accidents industriels tels que Seveso en 1976 ou plus récemment celui de l'usine AZF à Toulouse ont conduit les autorités européennes à identifier les sites industriels présentant des risques d'accidents majeurs. Dans le cadre de la réglementation en vigueur et notamment de la directive Seveso, les industriels sont amenés à identifier les scénarios possibles d'accidents et évaluer les risques en cas d'incidents industriels. Des simulations de l'évolution temporelle d'une concentration d'un produit toxique ou d'un polluant permet de calculer la dose reçue par une personne présente sur le site ou de cartographier les zones pour lesquelles le risque d'explosion est important dans le cas de rejet de produits explosifs. La réalisation d'études de danger par les industriels permet d'évaluer les conséquences d'un accident et de mettre en place des moyens de prévention ou de conforter les choix techniques (par exemple dans la conception d'un réseau de capteur). La cartographie des zones à risques est un outil de décision permettant de concentrer les efforts des industriels sur les scénarios à risques pour mettre en oeuvre des procédures et améliorations visant à réduire la probabilité qu'un scénario survienne.

Dans cette perspective, de nombreux outils de modélisation permettent de réaliser des études d'impacts et de risques environnementaux à différentes échelles spatio-temporelles, d'une échelle locale (quelques centaines de mètres à quelques kilomètres), à celle d'un continent. A l'échelle locale particulièrement, une multitude de configurations sont évaluées et diffèrent par les conditions météorologiques, la topographie du site, la nature et l'emplacement des sources d'émissions des polluants. Un nombre important de modèles existent, Perkins et al. (2005) en recensent plus de 80, qui peuvent être classés en trois familles selon une complexité croissante : les modèles de type gaussien, les modèles intégraux, et les modèles tridimensionnels CFD (Computational Fluid Dynamics).

Les modèles gaussiens sont basés sur une solution analytique de l'équation d'advection diffusion. Ils supposent donc que la répartition du polluant suit une loi gaussienne dont les écarts-types dépendent de la distance à la source (Briggs (1973), Islitzer and Dumbauld (1963), Turner (1997), Pasquill (1962)) ou du temps écoulé depuis l'émission (appelé aussi temps de transfert) (Doury, 1976, 1984), ainsi que des caractéristiques de la structure de l'atmosphère. Ces modèles s'appuient sur des paramétrisations semi-empiriques développées à partir de campagnes de mesures. On distingue les modèles de panaches stationnaires et les modèles à bouffées instationnaires. L'utilisation de code 1D de type gaussien permet la simulation de dispersion de gaz neutres ou passifs et de masse volumique proche de celle de l'air. Ils fournissent des résultats acceptables dans les situations pour lesquelles ils ont été paramétrés, mais ne sont pas suffisants lorsque les géométries se complexifient, avec la présence de bâtiments ou de reliefs pouvant perturber la dispersion du polluant.

Des modèles de “nouvelles générations” fondés sur une formulation gaussienne ou une formulation intégrale ont été développés pour surmonter les limites des modèles gaussiens, en tenant compte par exemple des bâtiments (Robins and Apsley, 2009) et des reliefs modérés (Carruthers et al., 2009). Les modèles intégraux sont basés sur l’intégration d’équations simplifiées de la mécanique des fluides et permettent de modéliser certains phénomènes qui ne sont pas pris en compte par un modèle gaussien, par exemple les effets dynamiques du rejet pouvant engendrer une turbulence spécifique ou les effets de gravité sur les gaz légers ou lourds (Robins et al., 2009). La simplification des équations se traduit par l’introduction de paramètres représentant les mécanismes non modélisés. Ces paramètres sont calés sur des expérimentations. On retrouve parmi ces modèles, le modèle ADMS (CERC, 2001) et le modèle AERMOD (Cimorelli et al., 2004) recommandé par l’USEPA (United States Environmental Protection Agency).

L’approche la plus générale est la résolution des équations tridimensionnelles gouvernant les écoulements et la dispersion des polluants. Depuis les dernières décennies, les outils CFD (Computational Fluid Dynamics) appliqués aux écoulements atmosphériques constituent un outil de modélisation pour décrire la dispersion en résolvant la dynamique de l’écoulement et les champs de grandeurs scalaires caractéristiques de la dispersion. Particulièrement adaptées aux géométries complexes (bâtiments et topographies accidentées), la résolution des équations de Navier Stokes régissant les écoulements atmosphériques permet d’obtenir un champ de vent turbulent tridimensionnel, à partir duquel est évalué le transport du polluant, par une approche Eulérienne, en résolvant numériquement l’équation d’advection diffusion sur un maillage de l’espace, ou par une approche Lagrangienne basée sur le calcul des trajectoires turbulentes d’un grand nombre de particules. De nombreux codes CFD sont adaptés à la simulation d’écoulements atmosphériques et à la dispersion de polluants. On peut citer entre autres, les codes commerciaux StarCD (Jicha et al., 2000) et Fluent (Sabatino et al., 2008), Flacs (Hanna et al., 2004), CFX (Chu et al., 2005), Fluidyn Panair (Mazzoldi et al., 2008), CFD URBAN (Coirier et al., 2006) ou le code Saturn développé depuis quelques années par EDF R&D (Demaël, 2007).

D’autre part, les outils CFD utilisés actuellement pour l’évaluation des risques sont tous basés sur des méthodes numériques classiques (différences finies, volumes finis) basées sur des schémas numériques au maximum d’ordre 2. La simulation précise de la dispersion de polluants dans des géométries 3D nécessite donc avec ce type de code de très gros maillages de calcul. Malgré l’utilisation des nouvelles architectures de calcul parallèles, l’efficacité de ces codes reste cependant limitée, ce qui conduit à des temps de simulation très importants. La question se pose alors de savoir comment réaliser des calculs précis avec un temps de restitution raisonnable et quelle peut être l’efficacité des schémas d’ordre élevé associés à une parallélisation efficace pour obtenir des simulations précises avec un temps de restitution faible.

D’autre part l’approche CFD nécessite un certain nombre de modélisations, en particulier celle de la turbulence et de la dispersion turbulente. L’approche DNS (Direct Numerical Simulation) permet de simuler les phénomènes de turbulence sans avoir recours à la modélisation, mais la discrétisation doit être capable de capturer la dissipation de l’énergie cinétique qui se produit aux plus petites échelles de l’écoulement appelées échelles de Kolmogoroff. Tennekes and Lumley (1972) montrent que le nombre de points de discrétisation est proportionnel à $Re^{9/4}$ avec Re le nombre de Reynolds représentatif de la turbulence de l’écoulement. La puissance des machines de calcul modernes permettent de réaliser ce type de simulation jusqu’à des nombres de Reynolds de l’ordre de quelques centaines, bien inférieurs

aux nombres de Reynolds des écoulements atmosphériques. Par conséquent, il est nécessaire d'utiliser des modèles de turbulence. Ces modèles peuvent être classés en deux grandes familles suivant la taille des échelles résolues : l'approche RANS (Reynolds Averaged Navier Stokes) généralement utilisée dans le cadre industriel (Chan et al. (2001); Wang and McNamara (2006); Hong et al. (2006)), réalise un filtrage statistique des équations de l'écoulement avec des échelles résolues relativement grandes. Des schémas de discrétisation peu précis (ordre 2 maximum) sont couramment employés avec ce type d'approche. Cependant ces méthodes posent certains problèmes, notamment une dépendance de la solution au maillage près des parois. La seconde approche est la simulation des grandes échelles (LES Large Eddy Simulation). Elle réalise un filtrage spatial et permet de résoudre des échelles plus fines de l'écoulement. Étant plus coûteuse que l'approche RANS, elle est utilisée majoritairement dans le cadre académique. Le développement récent de modèles hybrides RANS/LES (Hanjalic and Kenjeres (2008); Vengadesan and Nithiarasu (2007) par exemple) permet de réduire les temps de simulation et constitue pour les années à venir, une réelle avancée par rapport à l'approche RANS (Paik et al. (2009), Schmidt and Thiele (2002)). Cependant la résolution d'échelles fines sur des maillages fins nécessite d'apporter un soin particulier à la qualité de la discrétisation numérique. L'influence de l'erreur de discrétisation sur les modèles LES est encore mal maîtrisée (Vreman et al., 1996; Noma Park, 2007; Nakayama and Vengadesan, 2002; Park et al., 2004; Ghosal, 1996, 1995; Meinke et al., 2002; katopodes Chow and Moin, 2003). Les schémas disponibles dans la plupart des codes commerciaux sont des schémas au maximum d'ordre 2 et il est probable que les résultats LES obtenus avec ces codes soient fortement influencés par l'erreur numérique provenant des schémas d'ordre 2, particulièrement lorsque les maillages ne sont pas suffisamment fin. Le développement de schémas d'ordre supérieur, parallèles et efficaces pour maillages non structurés, est un enjeu important pour réduire les erreurs dues à la discrétisation, et analyser celles provenant de la modélisation.

Ainsi, dans le but de simuler efficacement la dispersion atmosphérique de polluants sur des sites industriels complexes, une approche volumes finis parallèle, d'ordre élevé, sur maillage non structuré est proposée pour résoudre les équations de Navier Stokes. La méthode proposée implémente des schémas implicites d'ordre élevé pour pouvoir réaliser à terme des simulations LES et hybrides LES/RANS et afin de réduire les temps de simulations, le code est parallélisé selon la technique de décomposition en domaine.

Le premier chapitre décrit les modèles physiques permettant de modéliser les écoulements atmosphériques avec dispersion de polluants. On met ainsi en évidence les différents modèles mathématiques qui nécessitent des méthodes numériques adaptées.

A partir de l'étude de la modélisation des écoulements avec dispersion de polluants, le second chapitre aborde les méthodes numériques permettant de résoudre les différents systèmes d'équations représentant les différents modèles physiques. Les points importants abordés sont le traitement du couplage pression-vitesse et le traitement d'équations de convection-diffusion.

Le troisième chapitre se concentre sur la discrétisation volumes finis en espace. Un schéma Padé d'ordre élevé est proposé et comparé à une reconstruction polynomiale et à schéma décentré classique d'ordre 2. Une attention particulière est portée à l'étude de la précision et du traitement des conditions

limites. Plusieurs cas de validation sont présentés, des cas les plus simples (calcul de gradient et Laplacien sur une solution a priori, convection d'une tache gaussienne), à des cas plus complexes (décroissance d'un tourbillon de Taylor, cavité entraînée, écoulement tridimensionnel autour d'un obstacle cubique).

Enfin la dernière partie présente la parallélisation du code par la technique de décomposition en sous domaines. La résolution des systèmes itératifs issus de l'utilisation de schémas implicites est un point clef de la parallélisation. Des méthodes itératives de type multigrille algébrique, méthodes à espace de Krylov et méthodes avec préconditionnement de type Schwarz, implémentées dans les bibliothèques PETSC et/ou HYPRE sont utilisées pour la résolution de systèmes linéaires de grandes dimensions. L'efficacité parallèle des différentes méthodes est estimée via différents tests (speed-up, scale-up).

Chapitre 1

Modèles physiques

1.1 Équations de Navier Stokes et transport d'un polluant

Les écoulements qui nous intéressent se situent dans la couche limite atmosphérique. L'évolution d'un fluide atmosphérique est gouvernée par les équations de Navier Stokes qui traduisent les principes de conservation de la masse (Eq. 1.1), conservation de la quantité de mouvement (Eq. 1.2) et conservation de l'énergie (Eq. 1.3) :

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (1.1)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) = -\nabla p + \nabla \cdot (\mu \nabla \mathbf{u}) + \rho \mathbf{f} \quad (1.2)$$

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot ((\rho e + p) \mathbf{u}) = \nabla \cdot [\tau \cdot \mathbf{u}] + \rho \mathbf{f} \cdot \mathbf{u} - \nabla \cdot \mathbf{q} \quad (1.3)$$

La quantité \mathbf{f} désigne la résultante des forces massiques s'exerçant sur le fluide. Pour les écoulements étudiés, la force gravitationnelle est la seule force volumique s'exerçant sur le fluide. L'écriture de la conservation de la quantité de mouvement dans le repère lié à la planète tournant à la vitesse angulaire $\Omega = 2\pi/86164 \text{ s}^{-1} = 7.29 \cdot 10^{-5} \text{ s}^{-1}$, fait apparaître lors du changement de repère l'accélération de Coriolis et l'accélération centrifuge. L'accélération de Coriolis est négligée car le nombre de Rossby est largement supérieur à un et l'accélération centrifuge est négligeable devant la force gravitationnelle.

Le transport d'un polluant de concentration massique Y_i est également prédit par une équation de convection-diffusion de la forme :

$$\frac{\partial}{\partial t} (\rho Y_i) + \nabla \cdot (\rho \mathbf{u} Y_i) = \nabla \cdot (\mu_i \nabla Y_i) + S_i$$

avec μ_i le coefficient de diffusion pour le polluant et S_i le terme source représentant la production du polluant ou sa destruction par réaction chimique par exemple.

Pour fermer le système d'équations, une équation d'état reliant la pression, la température et la masse volumique est nécessaire. Dans le cas de l'atmosphère, l'équation d'état d'un gaz parfait (Eq.

1.4) est valable en première approximation, c'est à dire si on néglige les effets de la vapeur d'eau :

$$P = \rho RT \quad (1.4)$$

Si on suppose que l'écoulement est incompressible, les équations deviennent :

$$\nabla \cdot (\mathbf{u}) = 0 \quad (1.5)$$

$$\frac{\partial u}{\partial t} + \nabla \cdot (u\mathbf{u}) = -\frac{1}{\rho} \nabla P + \nabla \cdot (\nu \nabla u) + \rho g \quad (1.6)$$

$$\frac{\partial T}{\partial t} + \nabla \cdot (T\mathbf{u}) = \nabla \cdot (\lambda \nabla T) + q \quad (1.7)$$

et l'équation de conservation de la masse devient une équation de conservation du volume. Cependant, cette approximation est trop forte pour les écoulements considérés et il est nécessaire de garder certains effets liés aux variations de masse volumique et notamment la poussée d'Archimède. Différents modèles permettant de prendre en compte ces effets sont présentés dans ce qui suit.

1.2 Modélisation des effets de flottabilité

La dispersion de polluants dans l'atmosphère fait intervenir des écoulements avec des variations de masse volumique d'origines thermiques. Ces variations sont le moteur de mouvements de convection naturelle, phénomènes pouvant être non négligeables dans les applications considérées. Des modèles d'écoulements dilatables (Boussinesq ou anélastique) ou compressibles permettent de tenir compte de ces effets. Ces modèles sont présentés de manière non exhaustive par ordre de simplification décroissante.

1.2.1 Approximation de Boussinesq

Le modèle le plus simplifié est l'approximation de Boussinesq qui est limitée aux écoulements dont l'échelle de déplacement vertical d'une particule est inférieure à l'échelle de longueur pour la variation de la masse volumique $H_\rho = \left(-\frac{1}{\rho} \frac{d\rho}{dz}\right)^{-1}$. Les variations de la masse volumique ne sont pas prises en compte dans l'équation de continuité, mais uniquement dans le terme de flottabilité selon :

$$\rho - \rho_0 \approx -\rho_0 \beta (T - T_0)$$

où ρ_0 est la masse volumique constante de l'écoulement, T_0 une température de référence et β le coefficient d'expansion thermique :

$$\beta = -\frac{1}{\rho} \left(\frac{\partial \rho}{\partial T} \right)_p$$

L'influence des perturbations de pression sur les termes de flottabilité est négligée et la pression P est décomposée en une pression hydrostatique p_H et une pression dynamique p telle que $P = p_H + p$ et

$\nabla p_H = \rho_0 \mathbf{g}$. Ainsi l'équation de quantité de mouvement devient :

$$\frac{\partial}{\partial t} u + \nabla \cdot (u\mathbf{u}) = -\frac{1}{\rho_0} \nabla p + \nabla \cdot \left(\frac{\mu}{\rho_0} \nabla u \right) + \beta(T - T_0)$$

L'approximation de Boussinesq est valable lorsque $\beta(T - T_0) \ll 1$.

1.2.2 Approximation anélastique

L'approximation anélastique a été introduite par Ogura et al. (1962) dans le but de filtrer les ondes sonores, permettant aux équations d'être intégrées en utilisant un pas de temps beaucoup plus grand que celui nécessaire à l'intégration du système d'équations pour écoulement compressible. Elle se situe entre le modèle de Boussinesq et le modèle compressible, le modèle de Boussinesq décrit dans le paragraphe précédent étant une simplification de l'approximation anélastique. Les équations du modèle anélastique sont dérivées des équations de Navier Stokes en réalisant un développement en série de puissance autour d'un état de référence fixe dans le temps. Cet état de référence correspond pour Ogura and Phillips (1962) à celui d'une atmosphère stratifiée et adiabatique. L'analyse proposée par les auteurs est basée sur une analyse dimensionnelle des mouvements à méso-échelles, et montre que le terme $\frac{\partial \rho}{\partial t}$ est négligeable. L'équation de continuité devient :

$$\nabla \cdot (\bar{\rho} \mathbf{u}) = 0 \tag{1.8}$$

avec $\bar{\rho}(z)$ la masse volumique d'un état de référence adiabatique. Si le modèle est appliqué à des phénomènes de convection humide, l'analyse d'échelle réalisée par Ogura n'est plus valable (Durran, 1989). Différents auteurs (Dutton and Fichtl (1969), Gough (1969), Lipps and Hemler (1982), Wilhelmson (1972)) ont proposé des modèles alternatifs pour lesquels l'état de référence est quelconque. Les hypothèses réalisées par les différents auteurs conduisent à différents systèmes d'équations, mais tous obtiennent une équation pour la conservation de la masse de la forme (Eq. 1.8). Durran (1989) propose de filtrer les ondes acoustiques par une approximation minimale des équations pour un fluide compressible. L'équation de conservation de la masse devient :

$$\nabla \cdot (\bar{\rho} \bar{\theta} \mathbf{u}) = \frac{H}{c_p \bar{\pi}}$$

avec $\bar{\rho}(z)$ et $\bar{\theta}(z)$ respectivement les états moyens de la masse volumique et de la température potentielle, $\bar{\pi}(z)$ l'état moyen de la fonction de Exner, et H le taux de production de chaleur par unité de volume.

Chacun de ces modèles se différencie par des propriétés de conservation d'énergie, de conservation de la masse, et la capacité à représenter correctement les ondes de gravité.

L'approximation anélastique a été appliquée à la modélisation de l'atmosphère terrestre avec par exemple le modèle météo ARPS ou aux applications astrophysiques telles que la modélisation des zones de convection solaire (Lantz and Fan, 1999) ou la convection du manteau terrestre (Schubert et al., 2000; Tackley, 2008).

1.2.3 Approximation faible nombre de Mach

Les effets de flottabilité peuvent également être directement simulés grâce aux équations de Navier Stokes (Eq. 1.1, Eq. 1.2, Eq. 1.3). Mais la nature hyperbolique des équations de Navier Stokes pour les écoulements compressibles permet la présence d'ondes de chocs et autres discontinuités dans l'écoulement et notamment d'ondes acoustiques qui se propagent à la vitesse du son c . Pour les écoulements étudiés, à faible nombre de Mach ($Ma \leq 0.01$), l'efficacité des méthodes explicites utilisées est sévèrement diminuée car la stabilité numérique nécessite l'utilisation de très petits pas de temps (H. Guillard, 1999) pour capturer les ondes acoustiques dont la vitesse est grande comparée à la vitesse convective. D'autre part, l'utilisation de méthodes implicites pour accroître la stabilité n'est pas adaptée car une disparité des valeurs propres du système rend le système à résoudre mal conditionné (S. Roller, 2000).

Trois stratégies différentes peuvent être adoptées pour surmonter ce problème. La première est d'étendre les solveurs compressibles (appelés "density based" par I. Keshtiban (2004)) aux écoulements à faible nombre de Mach. La réduction artificielle de la vitesse de propagation des ondes acoustiques permet d'utiliser des pas de temps plus grands, améliorant l'efficacité des méthodes numériques. Une autre stratégie est d'étendre aux écoulements à faible nombre de Mach les solveurs incompressibles basés sur une correction de pression. Plus de détails sont donnés par la suite concernant ces deux méthodes.

Enfin, la dernière solution est d'approximer les équations de Navier Stokes en utilisant un développement asymptotique. La séparation des échelles permet d'obtenir un système d'équations hyperboliques-elliptiques et peut éliminer les ondes acoustiques en conservant les effets de compressibilité causés par des phénomènes thermiques (forts gradients de température ou distribution non homogène de l'entropie). La décomposition n'est pas unique (Klein, 1995). Par exemple Müller (1999) utilise une approche asymptotique simple échelle et avec une "méthode de perturbation", réécrit les équations avec un développement en série de Taylor des variables en fonction du nombre de Mach :

$$p(\mathbf{x}, t, M) = p_0(\mathbf{x}, t) + Mp_1(\mathbf{x}, t) + M^2p_2(\mathbf{x}, t) + o(M^3)$$

Après adimensionnalisation des équations (voir Müller (1999) pour plus de détails), on obtient les équations de Navier Stokes d'ordre 0 :

$$\begin{aligned} \frac{\partial}{\partial t}\rho_0 + \nabla \cdot (\rho_0 \mathbf{u})_0 &= 0 \\ \frac{\partial \rho_0 \mathbf{u}_0}{\partial t} + \nabla \cdot (\rho_0 \mathbf{u}_0 \mathbf{u}_0) + \nabla p_2 &= G_0 \\ \frac{\gamma}{\gamma - 1} \rho_0 \left(\frac{\partial T}{\partial t} + \mathbf{u}_0 \cdot \nabla T_0 \right) - \frac{dp_0}{dt} &= Q_0 \end{aligned}$$

avec

$$\begin{aligned} Q_0 &= \frac{\gamma}{(\gamma - 1)Re_\infty Pr_\infty} \nabla \cdot (\kappa \nabla T_0 + (\rho q)_0) \\ G_0 &= \frac{1}{Re_\infty} \nabla \cdot \tau_0 - \frac{1}{Fr_\infty^2} \rho_0 \frac{\mathbf{g}}{\|\mathbf{g}\|} \end{aligned}$$

et l'équation d'état d'ordre 0 :

$$p_0(t) = \rho_0(\mathbf{x}, t)T_0(\mathbf{x}, t)$$

Cette méthode découple les ondes acoustiques des équations pour écoulements compressibles en les remplaçant par une forme pseudo acoustique dont la vitesse de propagation est comparable à celle de l'écoulement. La réduction artificielle de la vitesse du son permet d'utiliser des schémas numériques avec des pas de temps plus grands et la pression p_2 des équations pour écoulements à faible nombre de Mach joue le même rôle que la pression pour le système incompressible : elle est déterminée par la contrainte d'incompressibilité et est découplée des variations de masse volumique et de température.

1.3 Modélisation de la turbulence

La prise en compte uniquement des effets de flottabilité dans les simulations n'est pas suffisante. En effet, la turbulence atmosphérique joue également un rôle important dans la dispersion des polluants. Elle est visible dans la dispersion des nuages de fumées émanant par exemple des cheminées d'usines et est capable de diffuser les polluants d'une manière bien plus importante qu'un écoulement laminaire. Son origine dans la couche limite atmosphérique est double.

D'un point de vue dynamique, l'adhérence au sol de l'écoulement par viscosité donne naissance à un écoulement cisailé. Les gradients de vitesses moyennes ainsi créés sont à l'origine d'une production d'énergie cinétique turbulente issue de phénomènes mécaniques.

D'un point de vue thermique, l'instabilité de flottabilité pilote également la turbulence. Il se développe au dessus d'un sol chauffé par le rayonnement solaire, des fluctuations, des zones de convection bien connues des parapentistes. Au contraire si la surface du sol est plus froide que l'air, la conduction thermique refroidit les basses couches de l'atmosphère et inhibe la turbulence d'origine dynamique. Ainsi l'existence des mouvements convectifs dépend de la stratification thermique de l'atmosphère et de la forme du profil de température. On distingue alors trois régimes d'écoulement dans la couche limite atmosphérique qui dépendent du nombre de Richardson de flux R_f , rapport entre la production d'énergie cinétique par les forces d'Archimède et par le cisaillement de la vitesse :

- instable $R_f < 0$: les forces d'Archimède jouent un rôle important en créant de la turbulence.
- neutre $R_f \approx 0$: la turbulence est entièrement engendrée par l'instabilité dynamique du cisaillement de la vitesse moyenne.
- stable $R_f > 0$: les forces d'Archimède jouent un rôle important mais ont tendance à détruire l'énergie cinétique turbulente.

Différentes approches permettent de simuler des écoulements turbulents et sont présentées dans les sous sections suivantes. L'objectif des paragraphes suivants n'est pas de détailler les différents modèles existants, mais de montrer la structure et les caractéristiques communes de tous ces modèles. Le lecteur est invité à se référer à la littérature pour plus d'informations, entre autres Hanjalić and Kenjeres (2008) et Spalart (2000). On distingue principalement trois grandes familles de modèles de turbulence qui sont les modèles RANS (Reynolds Averaged Navier Stokes), les modèles LES (Large Eddy Simulation) et les modèles hybrides (RANS/LES). Ces trois familles de modèles sont présentées dans l'ordre historique suivant lesquelles elles sont apparues et se différencient par la nature des échelles résolues et la taille des échelles modélisées.

1.3.1 Modèles RANS

L'approche RANS (Reynolds Averaged Navier Stokes) décompose chaque variable ϕ en une somme d'une valeur moyenne $\overline{\phi}$ et d'une fluctuation ϕ' :

$$\phi = \overline{\phi} + \phi'$$

La valeur moyenne est définie comme la moyenne arithmétique d'un grand nombre de réalisations, répétées dans les mêmes conditions expérimentales. Ainsi on a la propriété suivante :

$$\overline{\phi'} = 0$$

La moyenne de l'équation de quantité de mouvement fait apparaître τ_{ij} un terme de flux turbulent inconnu d'ordre 1 :

$$\frac{\partial \overline{u}}{\partial t} + \overline{u_j} \frac{\partial \overline{u_i}}{\partial x_j} = -\frac{1}{\overline{\rho}} \frac{\partial \overline{p}}{\partial x_i} + \frac{1}{\overline{\rho}} \frac{\partial \overline{\tau_{ij}}}{\partial x_j} - \underbrace{\frac{\partial \overline{u'_j u'_i}}{\partial x_j}}_{\tau_{ij}}$$

Une équation décrivant l'évolution des termes d'ordre 1 peut être obtenue, mais fait apparaître d'autres termes inconnus d'ordre 2, et ainsi de suite. Il est alors nécessaire de réaliser une fermeture du modèle en modélisant les termes d'ordre les plus élevés en fonction de quantités connues.

La majorité des modèles de turbulence peuvent être classés en deux grandes classes. La première classe est basée sur l'hypothèse de Boussinesq qui relie le tenseur de Reynolds au gradient de la vitesse moyenne :

$$-\overline{\rho u'_i u'_j} = \mu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \left(\rho k + \mu_t \frac{\partial u_k}{\partial x_k} \right) \delta_{ij}$$

ou plus généralement :

$$\overline{u'_j u'_i} = f(\nu_t, \Omega_{ij}, S_{ij})$$

avec S_{ij} le tenseur des contraintes :

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$$

et Ω_{ij} le tenseur de vorticit  :

$$\Omega_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right)$$

Parmi les mod les couramment utilis s, le mod le $k - \epsilon$ (Launder and Spalding, 1972) d'ordre 1 est largement utilis . La viscosit  turbulente est d finie par $\nu_t = C_\mu k^2 / \epsilon$ o  $k = \frac{1}{2} \overline{u'_k u'_k}$ est l' nergie cin tique turbulente et ϵ le tau de dissipation. Ces deux grandeurs sont obtenues   partir d' quations de transport :

$$\frac{\partial k}{\partial t} + \nabla \cdot (k \mathbf{u}) = \nabla \cdot ((\nu + \nu_t) \nabla k) + P_k + G - \epsilon$$

$$\frac{\partial \epsilon}{\partial t} + \nabla \cdot (\epsilon \mathbf{u}) = \nabla \cdot ((\nu + \nu_t) \nabla \epsilon) + \frac{C_{\epsilon 1} P_k + C_{\epsilon 3} G - C_{\epsilon 2} \epsilon}{\tau} + Y$$

avec P et G représentant respectivement la production d'énergie cinétique turbulente par le tenseur moyen des contraintes et la force de gravité et τ est une échelle de temps de la turbulence définie par k/ϵ pour les écoulements à forts nombres de Reynolds.

Néanmoins, Franke et al. (2004) ont mis en évidence une surévaluation du terme de dissipation ϵ , et l'apparition de points de stagnation à proximité des parois d'obstacle, où l'énergie cinétique turbulente et la dissipation sont surévaluées. Ainsi pour compenser les points faibles du modèle originale, de nombreuses variantes ont été proposées (cf. Demaël (2007) et Hanjalic and Kenjeres (2008) pour une liste plus complète). Par exemple, le modèle RNG $k - \epsilon$ (Yakhot et al., 1992) permet de compenser la surestimation du terme de dissipation. Pour contourner l'apparition anormale d'un point de stagnation près des parois, le modèle $k - \Omega$ (Kato and Launder, 1993) modifie le terme de production d'énergie cinétique turbulente $P_k = C_{\mu} \epsilon |S|^2$ par $C_{\mu} \epsilon |S| |\Omega|$ avec $|S| = \frac{k}{\epsilon} (2S_{ij}S_{ij})^{1/2}$ et $|\Omega| = \frac{k}{\epsilon} (2\Omega_{ij}\Omega_{ij})^{1/2}$.

L'ensemble des modèles basés sur l'hypothèse de Boussinesq a été construit sur la base d'une turbulence homogène et isotrope et donnent des résultats médiocres pour certains écoulements avec des effets de rotation et de tourbillonnement. Ainsi la seconde classe de modèles abandonne l'hypothèse isotrope et résout chaque composante du tenseur de Reynolds. Ces modèles sont plus coûteux car ils nécessitent la résolution en trois dimensions d'au moins sept équations (6 composantes pour le tenseur de Reynolds, et la dissipation ϵ).

1.3.2 Modèles LES

La simulation des grandes échelles LES (Large Eddy Simulation) se différencie de l'approche RANS par la nature du filtrage et des échelles résolues. Elle consiste à réaliser un filtrage spatial des variables. Ainsi une variable ϕ se décompose en une partie filtrée grande échelle $\tilde{\phi}$ résolue, et une partie sous-maille ϕ' modélisée. Ainsi la vitesse u s'écrit :

$$u = \tilde{u} + u'$$

avec \tilde{u} la vitesse filtrée et u' la vitesse de sous-maille. Le filtrage de l'équation de quantité de mouvement introduit des corrélations entre les vitesses de sous-maille sous la forme du tenseur des contraintes de sous-maille τ_{ij} :

$$\tau_{ij} = \tilde{u}_i \tilde{u}_j - \widetilde{u_i u_j}$$

Ce tenseur doit être modélisé en fonction des grandeurs résolues pour fermer le système d'équations.

Un des modèles les plus utilisés est celui de Smagorinsky (1963) où le tenseur de sous-maille τ'_{ij} est exprimé par :

$$\tau'_{ij} = 2\nu_s \overline{S_{ij}} = 2\nu_{SM} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right)$$

où $\overline{S_{ij}}$ est le tenseur des contraintes de l'écoulement filtré et ν_{SM} la viscosité de sous-maille modélisée par :

$$\nu_{SM} = l_s^2 \sqrt{2\overline{S_{ij}S_{ij}}}$$

et l_s , l'échelle de longueur de Smagorinsky donnée par :

$$l_s = C_s \Delta$$

avec C_s le coefficient de Smagorinsky et $\Delta = (\Delta x_1 \Delta x_2 \Delta x_3)^{1/3}$ la largeur du filtre. La valeur de la constante C_s n'est pas universelle mais dépend de l'écoulement considéré. Une valeur $C_s = 0.1$ est souvent choisie. Près des parois, Van Driest (1956) propose d'utiliser une fonction d'amortissement pour l'échelle de longueur.

En réalité, la valeur optimale de C_s dépend de l'écoulement et varie spatialement et temporellement. M.Germano et al. (1991) et Lilly (1992) proposent de la déterminer localement et à chaque pas de temps en filtrant une seconde fois les équations de Navier Stokes avec un filtre test plus large que le premier.

1.3.3 Modèles hybrides RANS/LES

Dans les régions lointaines des parois, la densité des cellules nécessaires pour résoudre correctement la dynamique des grandes échelles varie selon $Re^{0.4}$, près des parois la densité des points du maillage varie selon $Re^{1.8}$ (Hanjalic and Kenjeres, 2008). Au contraire, une simulation RANS nécessite un maillage fin à la paroi uniquement dans sa direction normale, le nombre de points nécessaires est ainsi proportionnel à $\ln(Re)$. Une combinaison des approches RANS et LES permet donc de réduire le nombre de cellules requises. La sélection de zones en proche paroi pour lesquelles la simulation utilise l'approche RANS, peut s'effectuer selon une approche zonale avec une interface RANS / LES prédéfinie, ou avec un modèle qui tend vers l'approche RANS en proche paroi et vers l'approche LES lorsqu'on s'en écarte. La seconde approche sans doute la plus connue est la méthode DES (Detached Eddy Simulation) proposée par Spalart et al. (1997) qui utilise le modèle de Spalart and Allmaras (1992) pour l'approche RANS. Depuis, de nombreuses variantes ont été développées (cf. Spalart (2009, 2006) pour plus de détails). Chacun des modèles prend en compte la turbulence sous la forme d'une viscosité turbulente ν_t .

1.4 Conclusion

Ce chapitre, en présentant les deux principaux phénomènes physiques associés à la dispersion atmosphérique (effets de flottabilité et turbulence), a permis de mettre en évidence les caractéristiques communes des différents modèles.

Les différents modèles présentés pour la prise en compte des effets de flottabilité nécessitent la résolution d'équations de convection-diffusion avec des termes sources pour un certain nombre de variables. Ces équations sont non linéaires pour certaines et couplées. Un point important est le couplage à travers la pression : les différents modèles permettant de prendre en compte les effets de flottabilité conduisent à des couplages différents, dont la nature mathématique varie selon le modèle utilisé. Certains systèmes d'équations ont un caractère hyperbolique et permettent la propagation d'ondes dont la vitesse de propagation élevée comparée à celle de l'écoulement est problématique pour les écoulements

étudiés, alors que les autres systèmes ont un caractère elliptique, la pression assurant la continuité de la vitesse. Ainsi des méthodes numériques adaptées aux différents types de systèmes d'équations doivent être utilisées, en particulier les méthodes numériques classiques pour les systèmes hyperboliques ont été développées pour les écoulements avec un nombre de Mach relativement important et ne sont pas efficaces pour les écoulements étudiés. La résolution des différents modèles mathématiques présentés est l'objet du prochain chapitre.

La prise en compte des effets de turbulence s'effectue selon différentes approches. Même si les modèles RANS et LES diffèrent par la nature du filtrage et des échelles résolues, l'influence des échelles non résolues sur les échelles résolues est modélisée pour les deux approches sous la forme d'une viscosité turbulente ν_t (appelée viscosité de sous-maille ν_{SM} pour la LES). Dans le cas de simulations avec modèle de turbulence LES, une attention particulière est nécessaire pour la discrétisation numérique des termes convectifs : pour les écoulements considérés, la viscosité du fluide est faible et l'utilisation de schéma numérique trop dissipatif peut masquer les effets de la viscosité turbulente, en particulier pour les géométries complexes traitées pour lesquelles des maillages très fins sont difficilement réalisables pour des raisons de temps de simulation. Des discrétisations d'ordre 2 sont souvent utilisées et nécessitent l'utilisation d'un filtre de longueur quatre fois plus grande que la taille d'une cellule (Vreman et al., 1996), mais en pratique une seule cellule est utilisée pour le filtre. L'impact de l'erreur de discrétisation sur les simulations LES n'est pas encore très bien compris (Vreman et al., 1996; Noma Park, 2007; Nakayama and Vengadesan, 2002; Park et al., 2004; Ghosal, 1996, 1995; Meinke et al., 2002; katopodes Chow and Moin, 2003) et il est probable que les résultats soit fortement influencés par l'erreur numérique provenant des schémas d'ordre 2. Par conséquent, pour réduire les interférences entre les erreurs numériques et le modèle de sous-maille LES, des schémas numériques plus précis doivent être utilisés. La construction de schémas d'ordre élevé sera abordée ultérieurement.

Chapitre 2

Méthodes numériques

Le chapitre précédent a mis en évidence différents modèles permettant de prendre en compte les écoulements avec dispersion de polluants. Quelque soit le modèle physique (Boussinesq, anélastique, ou compressible), le bilan de quantité de mouvement est exprimé par une équation de convection-diffusion non linéaire, couplée aux équations d'énergie et de conservation de la masse. La dynamique du polluant est également décrite par une équation de transport sur les concentrations Y_i des polluants. Le couplage dépend de la nature du système d'équations représentant le modèle physique. Les équations pour un fluide incompressible forment un système hyperbolique-elliptique avec des vitesses de propagation d'ondes de pression infinies : la pression est vue comme un multiplicateur de Lagrange dont le but est d'assurer instantanément la continuité de la vitesse, alors que pour un fluide compressible, les équations forment un système hyperbolique avec des vitesses de propagation d'ondes de pression finies. L'information se propage suivant des caractéristiques réelles à des vitesses finies, et l'écoulement peut admettre la présence de chocs ou de discontinuités. La résolution numérique de ces équations peut poser certains problèmes pour les écoulements à faible nombre de Mach, comme cela a été décrit dans la section 1.2.3 du chapitre 1.

D'autre part, la viscosité des fluides étudiés est faible et les nombres de Reynolds des écoulements étudiés sont très élevés. Il en résulte une sensibilité des résultats numériques à la qualité de la discrétisation des termes convectifs, en particulier lors de l'utilisation de modèles LES (voir section 1.4 du chapitre 1).

Ainsi, la résolution numérique des équations présentées dans le chapitre 1 présente principalement deux difficultés qui sont le couplage des équations non linéaires et la qualité des schémas numériques utilisés pour réaliser des simulations précises avec à terme l'objectif de réaliser des simulations LES. Ce chapitre aborde le premier point lié au couplage des équations de continuité et de quantité de mouvement. La construction de schémas volumes finis d'ordre élevé sera abordée dans le chapitre suivant.

2.1 Méthode volumes finis et discrétisations

Avant d’approfondir les différentes méthodes numériques adaptées aux différents systèmes d’équations présentés dans le chapitre 1, nous nous intéressons au choix d’une méthode de discrétisation permettant de résoudre des systèmes d’équations aux dérivées partielles. Parmi les méthodes classiques utilisées, on distingue les méthodes locales (différences finies), les méthodes de type Galerkin (éléments finis et Galerkin discontinu) et les méthodes intégrant les équations sur des volumes de contrôle (volumes finis). La méthode choisie doit permettre de traiter des géométries complexes, avec un nombre de points de discrétisation important.

2.1.1 Différences finies

La méthode des différences finies discrétise à chaque point du maillage les équations en remplaçant les dérivées partielles par des approximations en fonctions des valeurs nodales. Cette méthode n’est appliquée généralement qu’aux maillages structurés. Quelques auteurs (Fernandez and Kulas, 2004; Campbell et al., 2002; Erlebacher, 1985) proposent des méthodes différences finies appliquées à des maillages non structurés. Cependant la complexité des géométries étudiées permet difficilement l’utilisation de ce type de méthode.

2.1.2 Éléments finis

Les éléments finis font partis de la classe plus générale des méthodes de Galerkin basées sur une formulation faible ou variationnelle des équations : les équations sont pondérées par des fonctions tests puis intégrées sur les volumes de contrôle. L’intérêt est de pouvoir traiter des géométries complexes en utilisant des volumes de contrôle compatibles avec des maillages non structurés. La solution est cherchée sous la forme d’une combinaison linéaire de fonctions de forme et la méthode de Galerkin consiste à choisir successivement comme fonctions tests dans la formulation faible ces mêmes fonctions de forme. La particularité de la solution pour la méthode éléments finis est d’être continue C^0 sur l’espace discrétisé.

2.1.3 Galerkin discontinu

L’utilisation de la méthode éléments finis crée des instabilités numériques en partie dues aux termes de convection quand on résout des écoulements avec des nombres de Reynolds élevés. La méthode de Galerkin discontinu est une approche hybride éléments finis/volumes finis permettant de remédier à ce problème. Les équations sont multipliées par une fonction test avant d’être intégrées sur un volume de contrôle. D’une manière identique à la formulation volumes finis, les termes convectifs et diffusifs des intégrales de volume sont transformés en intégrales de bord. Les éléments étant discontinus aux interfaces du maillage, le calcul du flux sur une face s’effectue à partir d’un des deux éléments qui partage cette face, avec des possibilités de décentrement (schéma de Roe par exemple). L’intégration volumique du terme temporel fait intervenir tout les degrés de liberté du volume de contrôle et conduit à une matrice de masse non diagonale. Plus d’informations concernant cette méthode sont disponibles dans la littérature (Jan S.Hesthaven, 2008).

2.1.4 Volumes finis

Les équations à résoudre sont des équations de convection-diffusion pouvant s'écrire :

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \mathbf{u} \phi) = \nabla \cdot (\mu_\phi \nabla \phi) + S_\phi \quad (2.1)$$

avec $\phi(\mathbf{x}, t)$ la quantité scalaire transportée par la vitesse \mathbf{u} et diffusée avec le coefficient de diffusion $\mu_\phi \geq 0$. La méthode volumes finis consiste à intégrer l'équation (2.1) sur chaque cellule C_i du maillage (cf. Figure 2.1). L'application du théorème de Gauss permet de transformer les intégrales de volume des termes convectifs et diffusifs en intégrales de surface, ce qui rend la méthode conservative :

$$\dot{\phi}_i = F(\phi) = \sum_{j \in V(i)} F_{ij}^{conv} - \sum_{j \in V(i)} F_{ij}^{viscous} \quad (2.2)$$

La méthode volumes finis est adaptée aux géométries complexes puisque la cellule d'intégration peut être de forme quelconque. Le domaine de calcul est sous divisé en un nombre fini de volumes de contrôle ou cellules. Une question qui se pose au regard de cette méthode est le choix du volume de contrôle et de la localisation des inconnues de vitesse et autres quantités scalaires. L'utilisation d'un arrangement de variables décalés "staggered" est l'approche la plus naturelle, mais n'est pas adaptée au cas de l'utilisation de maillages tridimensionnels non structurés. On préfère dans ce cas utiliser un arrangement "collocated" où toutes les variables sont stockées sur les mêmes noeuds. L'approche usuelle est d'assigner les noeuds de calcul au centre des cellules. Les valeurs calculées représentent alors la moyenne sur une cellule. La méthode qui en résulte correspond dans la littérature à l'approche "cell-centered" (Ferziger and Peric, 2002). Une autre approche répandue est d'assigner les noeuds de calcul aux noeuds des cellules du maillage. Ceci correspond à la méthode "cell-vertex" et à la méthode "mixed finite volume / finite element method" (C.Debiez et al., 1998). Le nombre de volumes de contrôle correspond aux nombres de points du maillage initial, et est inférieur au nombre de volumes dans le cas "cell-centered", mais avec un nombre de faces plus grand. Le traitement des conditions aux limites diffère selon l'approche. Notre choix s'est porté sur l'approche "cell-centered" sur des maillages non structurés ("collocated grid"), qui permet la prise en compte des conditions aux limites à travers l'imposition de flux ou de contraintes.

2.2 Solveur incompressible

Les équations pour écoulements incompressibles sont rappelées ci dessous :

$$\nabla \cdot (\mathbf{u}) = 0 \quad (2.3)$$

$$\frac{\partial u}{\partial t} + \nabla \cdot (u \mathbf{u}) = -\frac{1}{\rho} \nabla p + \nabla \cdot (\nu \nabla u) + \rho g \quad (2.4)$$

Le couplage entre la pression et la vitesse pour les écoulements incompressibles est particulier : la pression est un multiplicateur de Lagrange assurant la contrainte d'incompressibilité et il n'y a donc

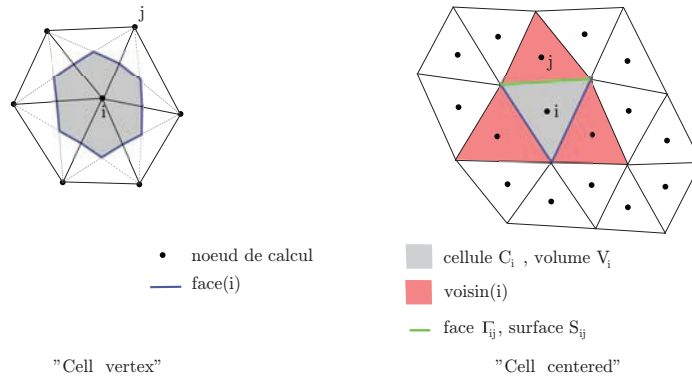


FIGURE 2.1 – Méthode volumes finis.

pas de moyen explicite pour la calculer, elle est déterminée implicitement à partir de l'équation de conservation de la masse $\nabla \cdot (\mathbf{u}) = 0$. Certaines méthodes résolvent simultanément l'équation de quantité de mouvement et celle de la conservation de la masse basée sur la pression. Dans ce cas, le système matriciel à résoudre a une taille importante ($3^2 n_{cell}^2$ en 2D et $4^2 n_{cell}^2$ en 3D). Cette approche nécessite donc un important espace mémoire. Les matrices issues d'un problème de type point-selle, sont généralement très mal conditionnées (Angot, 2007). Pour ces raisons, on s'intéresse aux autres méthodes (nommées dans la littérature "segregated") qui résolvent le système d'équations de Navier Stokes d'une manière découplée. Les sous sections suivantes présentent les principales classes de méthodes permettant de résoudre d'une manière découplée le système d'équations de Navier Stokes pour les écoulements incompressibles. Historiquement, la première méthode développée pour résoudre les équations du modèle Navier Stokes incompressible est la méthode MAC introduite par Harlow (1965). Elle appartient, avec les méthodes de projection de Chorin (1968) (sous section 2.2.1) et la classe de méthodes SIMPLE introduit par Patankar and Spalding (1972a) (sous section 2.2.2) à la classe de méthodes basées sur la pression ("pressure based method"). Ces méthodes résolvent les équations d'une manière découplée et sont caractérisées par l'utilisation d'une équation de Laplace pour le champ de pression à la place de l'équation de conservation de la masse. Le second groupe de méthodes sont les méthodes à compressibilité artificielle (sous section 2.2.3) proposées par Chorin (1967) et provenant des méthodes "density based" développées pour les écoulements compressibles et les systèmes d'équations hyperboliques.

2.2.1 Méthodes de projection

Un ensemble de méthodes regroupées sous le nom de méthodes de projection permettent de découpler l'équation de continuité et l'équation de quantité de mouvement : les méthodes de projection ont été introduites par Chorin (1968) et Témam (1969) pour résoudre les équations du modèle de Navier Stokes incompressible. La méthode est une approche fractionnée dans laquelle une vitesse intermédiaire \tilde{u}^{k+1} est obtenue en résolvant les équations de la quantité de mouvement (Eq. 2.5) sans considérer la contrainte d'incompressibilité. La vitesse est ensuite projetée sur un espace à divergence nulle par résolution d'une équation de Poisson (Eq. 2.6) :

$$\frac{1}{\Delta t}(\tilde{u}^{k+1} - u^k) - \nu \Delta \tilde{u}^{k+1} = 0 \quad (2.5)$$

$$\Delta p^{k+1} = \frac{1}{\Delta t} \nabla \cdot (\tilde{u}^{k+1}) \quad (2.6)$$

Alors que la vitesse peut être calculée avec une précision d'ordre 2 en temps, la pression n'est que d'ordre 1 en temps. Des améliorations ont été apportées par différents auteurs (Liu et al. (2004); Brown et al. (2001); Guermond and Shen (2003) entre autres) afin d'augmenter la précision en temps de la pression et accroître l'efficacité de la méthode. Guermond et al. (2006) présentent une revue des différentes méthodes de projection pour les écoulements incompressibles.

2.2.2 Méthodes de type SIMPLE

Une autre grande classe de méthodes regroupe les méthodes de type "prédiction-correction" conduisant à une équation de correction en pression. L'algorithme SIMPLE (Patankar and Spalding, 1972a) a été proposé dans les années 1970. Depuis, un nombre important de modifications ont été suggérées afin d'améliorer la robustesse et la convergence de la méthode. On peut citer SIMPLER, SIMPLEC (Doormaal and Raithby, 1984), PISO (Issa, 1986), SIMPLEM (Acharya, 1989), CLEAR (Tao, 2004). Ces méthodes sont assez similaires : ce sont des méthodes de point fixe qui calculent la solution non linéaire de manière itérative. Elles sont constituées principalement des trois étapes décrites dans l'Algorithme (1). Dans l'étape de prédiction, on peut remplacer la dérivée temporelle de la vitesse discrétisée à l'ordre 1 (schéma implicite de type Euler) par un schéma implicite à l'ordre 2 (Gear, 1971) faisant intervenir la solution au pas de temps n-1. Le gradient de pression est calculé en utilisant le théorème de Gauss avec la pression de l'itération précédente (k). La vitesse u_{ij}^* et le gradient normal de la vitesse ∇u_{ij}^* sur la face Γ_{ij} sont discrétisés selon le schéma de discrétisation choisi, présenté dans le chapitre 3.

La méthode SIMPLEC est proche de la méthode SIMPLE et diffère uniquement dans la relation de couplage entre correction de pression et correction de flux :

$$u'_{ij} \cdot n_{ij} = \left(\frac{V_i}{a_{ii} + \sum_j a_{ij}} \right)_{\Gamma_{ij}} \nabla p' |_{ij} \cdot n_{ij}$$

Pour les méthodes SIMPLE et SIMPLEC, le champ de vitesse et le flux aux faces obtenus ne vérifient pas l'équation de quantité de mouvement. L'algorithme PISO répète l'étape de correction pour obtenir un champ de vitesse qui vérifie de manière plus proche l'équation de quantité de mouvement.

L'utilisation d'un maillage "collocated", avec les variables de pression et de vitesse situées sur les mêmes noeuds de calcul peut conduire à des oscillations de pression non physiques, couramment appelées dans la littérature "effet damier" ("checkerboard effect"). Ces oscillations de pression n'apparaissent pas dans le cas de l'utilisation d'un maillage avec un arrangement des variables "staggered", pour lequel les noeuds de calcul de la pression et de la vitesse sont décalés. Ce problème de découplage

Algorithme 1 SIMPLE.

Étape de prédiction

Une étape de prédiction de la vitesse u^* à partir de l'équation de quantité de mouvement :

$$\begin{aligned} V_i \frac{u_i^* - u_i^n}{\Delta t} + \sum_{j \in \text{voisin}(i)} u_{ij}^* \mathbf{u}_{ij}^k \cdot \mathbf{n}_{ij} S_{ij} \\ = \frac{V_i}{\rho} \nabla p_i^k + \sum_{j \in \text{voisin}(i)} S_{ij} \nu_{ij} \nabla u_{ij}^* \cdot \mathbf{n}_{ij} \end{aligned}$$

ou sous une forme dans laquelle u_i^* est recherchée sous la forme $u_i^* = \delta u_i + u_i^k$:

$$\frac{\alpha_V}{\alpha_V} \delta u_i + \sum_{j \in V(i)} a_{ij} \delta u_j = - \left(V_i \frac{u_i^* - u_i^n}{\Delta t} + \sum_{j \in \text{voisin}(i)} u_{ij}^k S_{ij} \mathbf{u}_{ij}^* \cdot \mathbf{n}_{ij} S_{ij} + \frac{V_i}{\rho} \nabla p_i^k + \sum_{j \in \text{voisin}(i)} S_{ij} \nu_{ij} \nabla u_{ij}^k \cdot \mathbf{n}_{ij} \right)$$

avec α_V le coefficient de relaxation de la vitesse. Les coefficients a_{ii} et a_{ij} sont ceux assemblés dans la matrice et correspondent aux variations implicites des flux. Le terme de droite représente le résidu du bilan de quantité de mouvement sur la cellule C_i , il tend vers zéro lorsque les équations sont correctement résolues et sa norme sur l'ensemble des cellules du maillage permet de contrôler la convergence et de stopper le processus itératif.

Étape de correction

La vitesse obtenue u^* n'est pas à divergence nulle. Une correction des flux

$\Phi'_{ij} = \int_{\Gamma_{ij}} \mathbf{u}' \cdot \mathbf{n} d\Gamma = u'_{ij} \cdot n_{ij} S_{ij}$ est ajoutée aux flux $\Phi_{ij}^* = \mathbf{u}_{ij}^* \cdot \mathbf{n}_{ij} S_{ij}$ pour obtenir un flux Φ_{ij}^{k+1} à divergence nulle : $\sum_{j \in \text{voisin}(i)} \Phi_{ij}^{k+1} = 0$. Les variations de flux sont exprimées en fonction des variations de pression p' cherchées. Le couplage entre la correction de pression p' et la correction du flux Φ'_{ij} varie selon la méthode utilisée. Pour la méthode SIMPLE, le couplage s'effectue selon la relation :

$$\mathbf{u}'_{ij} \cdot \mathbf{n}_{ij} = \left(\frac{V_i}{a_{ii}} \right)_{\Gamma_{ij}} \nabla p' \Big|_{ij} \cdot \mathbf{n}_{ij} \quad (2.7)$$

et la correction de pression est calculée en une étape. Une fois la correction de pression calculée, la pression est relaxée explicitement :

$$p^{k+1} = p^k + \alpha_P p' \quad (2.8)$$

avec α_P le coefficient de relaxation pour la pression. Les flux aux faces sont corrigés pour être conservatifs :

$$\Phi_{ij}^{k+1} = u_{ij}^{k+1} \cdot n_{ij} S_{ij} = u_{ij}^* \cdot n_{ij} S_{ij} + u'_{ij} \cdot n_{ij} S_{ij} = \Phi_{ij}^* + S_{ij} \left(\frac{1}{a_{ii}} \right)_{ij} \nabla p' \Big|_{ij} \cdot n_{ij} \quad (2.9)$$

et la vitesse moyenne des cellules est corrigée selon la relation :

$$u_i = u_i^* + \frac{V_i}{a_{ii}} \nabla p' \quad (2.10)$$

Résolution des autres équations de transports

On résout les équations supplémentaires décrivant la turbulence, l'évolution d'un polluant ou encore l'équation d'énergie du fluide pour déterminer sa température.

pression-vitesse est résolu par Rhie and Chow (1983) en ajoutant un terme diffusif sur la pression lors du calcul du flux $\Phi_{ij} = \int_{\Gamma_{ij}} \mathbf{u} \cdot \mathbf{n}_{ij} d\Gamma$. Cependant, la méthode ne prend pas en compte la relaxation implicite de la vitesse et le flux dépend du facteur de relaxation. Majumdar (1988) propose une modification de la méthode pour la rendre indépendante du coefficient de relaxation de la vitesse. Les deux méthodes s'avèrent dépendantes du temps de discrétisation, et Choi (1999) modifie la méthode pour la rendre également indépendante du pas en temps. La méthode originale proposée par Rhie and Chow (1983) écrit la vitesse $u_{\Gamma_{ij}} \cdot \mathbf{n}_{ij}$ sur la face Γ_{ij} d'une manière analogue à la vitesse moyenne u_i et u_j sur les cellules C_i et C_j :

$$u_i + \frac{1}{a_{ii}} \sum_{k \in \text{voisin}(i)} a_{ik} u_k = -\frac{V_i}{a_{ii}} \overline{\nabla p}_i$$

$$u_j + \frac{1}{a_{jj}} \sum_{k \in \text{voisin}(j)} a_{jk} u_k = -\frac{V_j}{a_{jj}} \overline{\nabla p}_j$$

La méthode revient à soustraire à la vitesse $u_{\Gamma_{ij}}$ interpolée sur la face Γ_{ij} , la différence entre le gradient de pression sur la face $\left. \frac{\delta p}{\delta n_{ij}} \right|_{\Gamma_{ij}}$, et le gradient de pression $\left. \frac{\delta p}{\delta n_{ij}} \right|_{\Gamma_{ij}}$ interpolé à partir des gradients des cellules i et j , $\left. \frac{\delta p}{\delta n_{ij}} \right|_{C_i}$ et $\left. \frac{\delta p}{\delta n_{ij}} \right|_{C_j}$:

$$u_{ij} \cdot \mathbf{n}_{ij} = \bar{u}_{\Gamma_{ij}}^{HO} \cdot \mathbf{n}_{ij} - \left(\frac{V_i}{a_{ii}} \right) \bigg|_{\Gamma_{ij}} \left(\left. \frac{\delta p}{\delta n_{ij}} \right|_{\Gamma_{ij}} - \frac{\overline{\delta p}}{\delta n_{ij}} \bigg|_{\Gamma_{ij}} \right)$$

avec $\bar{u}_{\Gamma_{ij}}^{HO}$ la vitesse sur la face Γ_{ij} calculée à partir du schéma de discrétisation choisi. Pour un maillage uniforme, les gradients de pression dans la relation précédente s'écrivent :

$$\left. \frac{\delta p}{\delta n_{ij}} \right|_{\Gamma_{ij}} = \frac{p_j - p_i}{\|ij\|} \cdot \mathbf{n}_{ij} \quad , \quad \frac{\overline{\delta p}}{\delta n_{ij}} \bigg|_{\Gamma_{ij}} = \frac{1}{2} (\overline{\nabla p}_i + \overline{\nabla p}_j) \cdot \mathbf{n}_{ij}$$

2.2.3 Compressibilité artificielle pour les écoulements stationnaires

Parmi les autres méthodes permettant de résoudre les équations de Navier Stokes pour écoulements incompressibles, on peut citer la méthode de compressibilité artificielle proposée par (Chorin, 1967) issue des méthodes "density based" adaptées à la résolution des équations de Navier Stokes pour écoulements compressibles. Les méthodes "density based" résolvent les équations d'une manière couplée, en gardant les termes instationnaires. Les écoulements stationnaires sont donc résolus par un avancement en temps. Chorin ajoute une dérivée temporelle de la pression à l'équation de continuité pour introduire une dimension hyperbolique aux équations :

$$\frac{1}{\beta} \frac{\partial p}{\partial t} + \nabla \cdot \mathbf{u} = 0 \tag{2.11}$$

Une approche similaire aux méthodes de prédiction-corrrection peut être utilisée pour résoudre d'une manière découplée le nouveau système d'équations (Eq. 2.11, Eq. 2.4). Mais les méthodes issues de la résolution du système d'équations du modèle Navier Stokes compressible peuvent également être utilisées. Le choix du paramètre de compressibilité artificielle β est un point clef pour la performance

de cette méthode. La valeur de β utilisée peut être déterminée à partir de la pseudo vitesse du son $c = \sqrt{v^2 + \beta}$. Elle est généralement comprise entre 0.1 et 10. Une fois la solution convergée, le terme temporel disparaît et la solution satisfait le système d'équations de Navier Stokes. La technique de double avance en temps ou "dual-time stepping" (Jameson and Alonso, 1991) peut être utilisée pour réaliser des simulations instationnaires. Elle utilise un pseudo pas de temps τ dont la dérivée temporelle devient nulle après convergence. Cependant la méthode de compressibilité artificielle est inefficace pour l'étude de couche limite où les zones de séparation sont très sensibles aux gradients de pression (Ferziger and Peric, 2002)

2.3 Solveur compressible pour faible nombre de mach

Les équations forment un système hyperbolique. L'information se propage suivant des caractéristiques réelles à des vitesses finies. Lorsque le nombre de Mach est très faible, l'échelle de temps associée à la propagation des ondes acoustiques est très faible comparée à l'échelle de temps associée à la vitesse de l'écoulement. La stabilité numérique nécessitant des pas de temps très petits, l'efficacité des méthodes développées pour les écoulements à nombre de Mach modéré est fortement diminuée. Il existe principalement deux approches différentes pour effectuer efficacement des simulations à faible nombre de Mach. La première consiste à étendre les solveurs compressibles "density based" au cas faiblement compressible : Turkel (1987) a initié l'idée d'accélérer la convergence par un préconditionnement en temps ayant pour effet d'augmenter artificiellement l'échelle de temps des ondes acoustiques. Cette méthode est à rapprocher des modèles de développement asymptotique présentés dans le chapitre "Modèles physiques" puisqu'elle uniformise les valeurs propres du système à résoudre. La seconde approche est une extension des solveurs incompressibles (présentés plus haut) pour les écoulements à faible nombre de Mach. Habituellement, les méthodes de projection sont basées sur l'équation de continuité, Nerinckx et al. (2005) proposent une alternative avec une méthode de projection basée sur l'équation d'énergie.

2.3.1 Préconditionnement de Turkel

Turkel (1987) a introduit une famille de préconditionneurs pour les écoulements à faible nombre de Mach permettant d'accélérer la convergence vers une solution stationnaire. Les dérivées temporelles sont modifiées (preconditionnées) pour forcer les valeurs propres du système d'équations à être du même ordre de grandeur lorsque l'écoulement est à faible nombre de Mach. Cette approche est similaire à la méthode de compressibilité artificielle. Les équations modifiées par le préconditionnement sont équivalentes au système d'origine uniquement dans le cas stationnaire. Pour les applications instationnaires, la méthode de double avance en temps (Jameson and Alonso, 1991) est une nouvelle fois utilisée : la dérivée temporelle est traitée comme un terme source et à chaque pas de temps physique, une avance en temps artificiel est réalisée. Les équations sont résolues avec les variables primitives $Q = [\rho, \mathbf{V}, T]^T$ et les termes temporels préconditionnés par la matrice Γ :

$$\Gamma \frac{\partial}{\partial t} \int_V Q dV + \oint [F - G] dA = \int_V H dV$$

avec

$$\Gamma = \begin{bmatrix} \Theta & 0 & 0 & 0 & \rho_T \\ \Theta_u & \rho & 0 & 0 & \rho_T u \\ \Theta_v & 0 & \rho & 0 & \rho_T v \\ \Theta_w & 0 & 0 & \rho & \rho_T w \\ \Theta H - \delta & \rho u & \rho v & \rho w & \rho_T H + \rho C_p \end{bmatrix} \neq \frac{\partial W}{\partial Q}$$

$$\rho_T = \left. \frac{\partial \rho}{\partial T} \right|_p, \quad \rho_P = \left. \frac{\partial \rho}{\partial p} \right|_T$$

$$\Theta = \left(\frac{1}{U_r^2} - \frac{\rho_T}{\rho C_p} \right)$$

$$\alpha = (1 - \beta U_r^2) / 2$$

$$\beta = \left(\rho_P + \frac{\rho_T}{\rho C_p} \right)$$

$\delta = 0$ pour un fluide incompressible et $\delta = 1$ pour un gaz parfait. Les valeurs propres deviennent $u, v, w, u' + c', u' - c'$ avec

$$u = \mathbf{v} \cdot \hat{n}$$

$$u' = u(1 - \alpha)$$

$$c' = \sqrt{\alpha^2 u^2 + U_r^2}$$

Pour le cas faiblement compressible, $U_r \rightarrow 0$ et $\alpha \rightarrow 1/2$.

Similairement, Van Leer et al. (1991) ont développé un préconditionneur symétrique pour les équations d'Euler bi-dimensionnelles. Néanmoins, une limitation à ces méthodes est le manque de robustesse près des points d'arrêt de l'écoulement : Wong et al. (2001) et Darmofal and Schmid (1996) identifient un problème de dissipation artificielle lorsque les vecteurs propres de la solution deviennent presque parallèles et Darmofal and Schmid (1996) démontrent numériquement et théoriquement qu'avec un manque d'orthogonalité entre les vecteurs propres, des faibles perturbations pouvaient être amplifiées de manière significative et altérer la solution stationnaire.

2.3.2 Méthode de projection basée sur l'équation d'énergie par Nerinckx

Contrairement aux méthodes "density based", les méthodes "pressured based" décrites dans la section 2.2 ont été développées initialement pour résoudre les écoulements incompressibles. La première implémentation de ce type de méthode pour les écoulements compressibles est attribuée à Harlow and Amsden (1968). Plus tard, Karki and Patankar (1989) ont étendu la méthode SIMPLE

aux écoulements compressibles. Moukalled and Darwish (2000) et Choi et al. (2003) proposent un algorithme similaire. Dans l'étape de correction, ils écrivent la correction du flux massique sur les faces selon :

$$\rho^n u'_{ij} + \rho'_{ij} u_{ij}^* + \rho'_{ij} u'_{ij}$$

Le terme $\rho' u'_{ij}$ est négligé, la correction de vitesse u'_{ij} est approximée d'une manière similaire à la méthode SIMPLE et la correction de masse volumique est exprimée par :

$$\rho' \approx \left(\frac{\partial \rho}{\partial p} \right) p' = C_\rho p'$$

On obtient ainsi une équation de correction sur la pression avec des termes supplémentaires convectifs et instationnaires par rapport à la méthode incompressible : on ne résout plus un Laplacien comme dans le cas incompressible, mais la convection d'une onde de pression. La présence du terme convectif rend la solution unique, contrairement au Laplacien où la pression est déterminée à une constante près. En plus d'exprimer la variation temporelle de masse volumique $\frac{\partial \rho}{\partial t}$ en fonction d'une variation de pression $\left(\frac{\partial \rho}{\partial p} \right) \frac{\partial p}{\partial t}$ comme le font Moukalled and Darwish (2000) et Choi et al. (2003), Bijl and Wesseling (1998) l'expriment également en fonction de la variation d'enthalpie $\left(\frac{\partial \rho}{\partial h} \right) \frac{\partial h}{\partial t}$. Le terme $\left(\frac{\partial \rho}{\partial h} \right) \frac{\partial h}{\partial t}$ est calculé à partir de l'équation d'énergie. Munz et al. (2003); Roller and Munz (2000) en utilisant une formulation faible nombre de Mach, construisent une méthode avec plusieurs variables pour la pression, pour tenir compte séparément des effets thermodynamiques, des effets acoustiques et des effets incompressibles.

Plus récemment, Nerinckx et al. (2005) proposent de résoudre implicitement les équations d'énergie, de conservation de la masse et de la quantité de mouvement en utilisant une méthode de projection. La projection est réalisée à partir d'un couplage entre l'équation d'énergie et l'équation de conservation de la masse lorsque les transferts de chaleur sont pris en compte (Algorithme 3), ou uniquement à partir de l'équation d'énergie lorsque les transferts thermiques ne sont pas pris en compte (Algorithme 2). Les auteurs montrent que l'équation d'énergie du système d'équations de Navier Stokes compressible est équivalente à $\nabla \cdot \mathbf{u} = 0$ lorsque l'écoulement est adiabatique et le nombre de Mach tend vers zéro.

Une discrétisation implicite volumes finis des équations de Navier Stokes compressible peut alors s'écrire :

$$\begin{aligned} \rho_i^{n+1} - \rho_i^n + \tau \sum_{j \in \text{voisin}(i)} \rho_{ij}^{n+1} \Phi_{ij}^{n+1} &= 0 \\ (\rho u)_i^{n+1} - (\rho u)_i^n + \tau \sum_{j \in \text{voisin}(i)} \Phi_{ij}^{n+1} (\rho u)_{ij}^{n+1} &= \tau \sum_{j \in \text{voisin}(i)} S_{ij} p_{ij}^{n+1} n_{ij} + \tau \sum_{j \in \text{voisin}(i)} S_{ij} \nabla (\nu_{ij} (\rho u)_{ij}^{n+1}) \\ (\rho E)_i^{n+1} - (\rho E)_i^n + \tau \sum_{j \in \text{voisin}(i)} (\rho H)_{ij}^{n+1} \Phi_{ij}^{n+1} &= 0 \end{aligned} \quad (2.12)$$

avec $\tau = \Delta t / V_i$. $\Phi_{ij} = \int_{\Gamma_{ij}} \mathbf{u}_{ij} \cdot \mathbf{n}_{ij} d\Gamma$ représente le flux traversant la face Γ_{ij} . Les exposants n et $n+1$ correspondent au temps t et $t + \Delta t$ avec Δt le pas de temps, V_i le volume de la cellule C_i . ρ , P , u , H et E représentent respectivement la masse volumique, la pression, la vitesse, l'enthalpie totale et l'énergie totale.

Algorithme 2 Nerinckx sans transfert thermique.

- Prédiction de ρ à partir de l'équation de continuité et ρu à partir de l'équation de quantité de mouvement.
 - Une correction de pression est calculée à partir de l'équation d'énergie.
 - Mise à jour de la quantité de mouvement.
 - Calcul de la vitesse à partir de ρ et ρu .
 - Utilisation de l'équation d'état pour calculer la température fonction de ρ et p .
-

A chaque itération non linéaire (k) de l'Algorithme (2), une étape de prédiction (*) permet de calculer la masse volumique et la quantité de mouvement à partir des équations de conservation de la masse et de quantité de mouvement. Pour les flux convectifs Φ_{ij} , la vitesse sur la face Γ_{ij} est calculée en utilisant l'interpolation MIM (Momentum Interpolation Method) proposée par Rhie and Chow (1983) et déjà présentée dans la section précédente 2.2.2 :

$$\begin{aligned} \rho_i^* - \rho_i^n + \tau \sum_{j \in \text{voisin}(i)} \rho_{ij}^* \Phi_{ij}^k &= 0 \\ (\rho u)_i^* - (\rho u)_i^n + \tau \sum_{j \in \text{voisin}(i)} \Phi_{ij}^k (\rho u)_{ij}^* &= \tau \sum_{j \in \text{voisin}(i)} S_{ij} p_{ij}^k n_{ij} + \tau \sum_{j \in \text{voisin}(i)} S_{ij} \nabla(\nu_{ij} (\rho u)_{ij}^*) \end{aligned}$$

Après l'étape de prédiction, une correction de pression p'_i (et éventuellement une correction de température T'_i) est calculée à partir d'une équation basée sur l'équation d'énergie (et éventuellement de l'équation de conservation de la masse). L'équation de correction sur la pression

$$(\rho e)_p^* p'_i + \tau \sum_{j \in \text{voisin}(i)} h_{ij}^* (\rho u)'_{ij} S_{ij} = - \left((\rho E)_i^* - (\rho E)_i^n + \tau \sum_{j \in \text{voisin}(i)} (\rho H)_{ij}^* \Phi_{ij}^* \right) \quad (2.13)$$

est obtenue en remplaçant dans l'équation d'énergie, l'énergie totale et l'enthalpie totale par les expressions (2.14) et (2.15) :

$$(\rho E)^{n+1} = (\rho E)^* + (\rho e)_p^* p' \quad (2.14)$$

$$(\rho H)_i^{n+1} u_{ij}^{n+1} = (\rho H)_i^* u_{ij}^* + h_i^* (\rho u)'_{ij} S_{ij} \quad (2.15)$$

dans lesquelles $\rho u'_{ij}$ est relié à la correction de pression p'_i par l'intermédiaire de l'équation de quantité de mouvement en utilisant le couplage SIMPLE (Patankar and Spalding, 1972b) décrit précédemment.

Chaque étape de l'algorithme nécessite la résolution de systèmes linéaires creux de grandes tailles. Un des points clefs de l'efficacité des méthodes implicites est donc l'utilisation de solveurs linéaires efficaces.

2.4 Conclusion

Le premier chapitre sur les différents modèles physiques utilisés pour la simulation des écoulements avec transport de polluants a mis en évidence des systèmes d'équations avec différents couplages vitesse-

Algorithme 3 Nerinckx avec transfert thermique.

- Prédiction de ρ à partir de l'équation de continuité et ρu à partir de l'équation de quantité de mouvement.
 - Une correction de pression et de température est calculée à partir de l'équation d'énergie et l'équation de continuité.
 - Mise à jour de la densité ρ à partir de l'équation d'état $\rho = f(T, p)$.
 - Mise à jour de la quantité de mouvement.
 - Calcul de la vitesse à partir de ρ et ρu .
 - Utilisation de l'équation d'état pour calculer la température fonction de ρ et p .
-

pression qui nécessitent d'utiliser des méthodes numériques adaptées. Dans ce chapitre, nous avons présenté les principales méthodes permettant de résoudre le système d'équation de Navier Stokes pour un fluide incompressible ou compressible à faible nombre de Mach. Pour les écoulements incompressibles associés à un système d'équations elliptiques avec contrainte, les méthodes classiques de projection découplent les équations et une équation de Laplace sur la pression remplace l'équation de conservation de la masse. Pour les écoulements compressibles, les solveurs compressibles sont inefficaces à faible nombre de Mach. Nerinckx et al. (2005) proposent une méthode de projection basée sur l'équation d'énergie. Cette méthode est une alternative aux méthodes de projection basées sur l'équation de continuité et au préconditionnement en temps de Turkel (1987).

Ainsi de nombreuses méthodes (méthode de projection basée sur l'équation d'énergie, ou sur la continuité, et méthode de type SIMPLE) nécessitent de résoudre une équation de Laplace pour la détermination de la pression ce qui implique l'utilisation d'un solveur linéaire efficace pour une matrice symétrique, creuse et de grande dimension. Ce solveur doit être parallélisable pour pouvoir utiliser les nouvelles architectures des calculateurs actuels. La résolution de l'équation de quantité de mouvement a été décrite de manière implicite, mais sa résolution explicite est possible. Toutefois, un critère de stabilité pour les termes diffusifs (Wesseling (1995)), et un critère CFL pour les termes convectifs (Courant et al. (1967)) limitent la stabilité et le pas de temps utilisable. Il est alors possible d'envisager une implicitation uniquement des termes visqueux pour s'affranchir des conditions de stabilité sur le terme diffusif, auquel cas, la symétrie du système linéaire à résoudre permet d'utiliser des solveurs adaptés et efficaces (Aubry et al., 2008; Nicolaidis, 1987; MacLachlan et al., 2008)). Cependant l'implicitation des termes convectifs permet de s'affranchir du critère CFL et d'accroître la robustesse des méthodes numériques. En effet le pas de temps va être contraint par la cellule de plus grand CFL ($v\Delta t/\Delta x$), et l'utilisation de géométries complexes avec une source de rejet de polluant discrétisée finement et située dans une zone avec des fortes vitesses peut être problématique. Dans ce cas, l'utilisation d'un solveur pour matrices non symétriques est requis.

Dans la suite, la méthode SIMPLE pour résoudre les écoulements incompressibles a été choisie. Elle offre l'avantage d'avoir des systèmes linéaires à résoudre de tailles identiques. La résolution de ces systèmes linéaires est abordée dans le chapitre 4. Le chapitre qui suit (chapitre 3) étudie les schémas numériques d'ordre élevé, en particulier leur efficacité pour réduire le nombre de cellules du maillage et obtenir des simulations précises.

Chapitre 3

Schémas d'ordre élevé

L'obtention de résultats précis nécessite l'utilisation de schémas numériques d'ordre élevé. Un des objectifs est de ne pas masquer les effets de la modélisation de façon générale par la diffusion numérique des schémas. En particulier, la prise en compte de la turbulence par des modèles LES nécessite d'utiliser des schémas numériques précis pour ne pas masquer les effets de la viscosité de sous-maille ν_{SM} par une viscosité numérique trop importante (voir section 1.4 du chapitre 2). Les méthodes d'ordre élevé ont également le potentiel de réduire les coûts des simulations numériques, en réduisant le nombre de cellules de calcul nécessaires pour atteindre la précision désirée. L'efficacité numérique pour des applications nécessitant d'importantes ressources de calcul telles que la simulation des grandes échelles (LES) ou la simulation des échelles détachées (DES) d'écoulements turbulents dans des domaines d'applications variés peut être améliorée.

Les architectures modernes de calcul parallèle fournissent une augmentation de la puissance de calcul et des ressources mémoires surpassant les ordinateurs mono-processeur. Ainsi la mise en œuvre parallèle des méthodes d'ordre élevé est nécessaire afin de réaliser des simulations avec des temps de calcul abordables.

De nombreuses études concernant les schémas d'ordre élevé pour les équations de conservation hyperboliques et la simulation d'écoulements compressibles ont été réalisées, mais les études sur les équations elliptiques associées à la simulation d'écoulements incompressibles sont peu fréquentes. Pour les problèmes hyperboliques, plusieurs approches ont été proposées pour construire des schémas volumes finis d'ordre élevé, basées sur les schémas compacts (M.Piller and E.Stalio, 2008; Pereira et al., 2001; Lacor et al., 2004; Lele, 1992), la reconstruction polynomiale sur molécules fixes ou de type ENO (Carpentier, 2000; Ivan and Groth, 2007; Mandal et al., 2008; Nejat and Ollivier-Gooch, 2008; Ollivier-Gooch and Altena, 2002; Weller and Weller, 2008), ou des méthodes de type Galerkin discontinu (Bassi and Rebay, 1997; Bassi et al., 2007; Cockburn et al., 2005; Dumbser, 2010; Landmann et al., 2008).

Les schémas ENO sélectionnent pour chaque cellule la molécule conduisant à la reconstruction la plus lisse. L'effort de calcul nécessaire à la reconstruction sur plusieurs molécules est important et une molécule fixe est souvent utilisée à la place dans les cas où la solution ne présente pas de forte discontinuité. Plusieurs auteurs (Carpentier, 2000; Ollivier-Gooch and Altena, 2002; Ivan and Groth, 2007) proposent une reconstruction polynomiale k-exacte, avec la préservation de la valeur moyenne dans la

cellule. La préservation de la valeur moyenne est réalisée, soit par une contrainte dans la procédure des moindres carrés (Nejat and Ollivier-Gooch, 2008), soit imposée par la forme de la fonction polynôme (Carpentier, 2000; Nejat and Ollivier-Gooch, 2008). La monotonie est une autre propriété importante pour ces schémas à molécule fixe, dans les régions de forts gradients avec d'éventuelles discontinuités. Elle est réalisée de différentes manières. Pour préserver la monotonie de la solution, une reconstruction limitée ($k = 1$) peut être utilisée conjointement avec un indicateur de lissage pour détecter les régions où l'ordre de reconstruction doit être réduit (Ivan and Groth, 2007). Une autre méthode proposée par Mandal et al. (2008) consiste à calculer le premier et deuxième ordre du développement en série de Taylor en utilisant une formulation des moindres carrés. Une pondération adéquate, fonction de la solution permet de traiter les forts gradients et de conserver la monotonie. Cette approche à plusieurs étapes permet également de restreindre la molécule de calcul aux voisins les plus proches.

Les schémas compacts sont une alternative à la reconstruction d'ordre élevé. Les schémas compacts sont des schémas d'ordre élevé très répandus avec les formulations différences finies (Hirsh, 1975; Lele, 1992; Shen et al., 2006), mais leur extension à la formulation volumes finis n'est pas triviale, particulièrement pour les maillages non uniformes et non cartésiens. Afin de prendre en compte des maillages irréguliers, M.Piller and E.Stalio (2008) proposent une transformation de coordonnées, entre un maillage curviligne non orthogonale dans l'espace physique et l'espace de calcul. Lacor et al. (2004) introduisent un schéma compact d'ordre 3 développé directement dans l'espace physique en utilisant une relation de Padé multidimensionnelle sur des maillages structurés arbitraires.

Comme le soulignent Ollivier-Gooch and Altena (2002), le développement de schémas de hautes précisions pour les simulations en volumes finis s'est historiquement concentré sur les flux convectifs et le développement de méthodes d'ordre élevé pour les écoulements visqueux a été moins important. La conception de schémas numériques pour les flux visqueux qui sont consistants et positifs pour des maillages complexes pose encore problème : la robustesse du solveur est affectée par la discrétisation des flux visqueux, qui tendent à perdre leur positivité pour les discrétisations d'ordre élevé. L'utilisation d'une approche hybride volumes finis / éléments finis (C.Debiez et al., 1998) permet de calculer les flux visqueux par une formulation éléments finis de Galerkin qui satisfait la positivité. Toutefois, il est difficile de concevoir des schémas pour les flux visqueux pour une formulation volumes finis non structurés avec les variables aux centres des cellules ("cell-centered"), satisfaisant à la fois des propriétés de consistance et de positivité (Munikrishna, 2007). Différentes approches ont été proposées. Par exemple, Coirier (1994) a développé une procédure "diamond path" utilisant les valeurs interpolées aux noeuds du maillage pour obtenir un schéma positif. Ravikumar (2001) calcule les dérivées premières et secondes en utilisant une procédure des moindres carrés. Les dérivées normales aux faces sont interpolées à partir des dérivés aux centres des cellules. Holmes and Connell (1989) proposent une procédure de pondération des moindres carrés qui calcule directement un pseudo Laplacien qui préserve la positivité. D'autres auteurs (comme Ollivier-Gooch and Altena (2002)) calculent les flux visqueux par dérivation du polynôme utilisé pour les flux de convection, conduisant à une erreur de troncature d'ordre $p-1$ pour les flux diffusifs au lieu d'une erreur de troncature d'ordre p pour les flux convectifs. Récemment, Vaassen et al. (2006) ont modifié la procédure proposée par Coirier (1994) pour préserver la consistance du schéma sur des maillages quelconques. Avec l'objectif d'obtenir des méthodes plus robustes lorsque les cellules sont déformées, Traoré et al. (2009) proposent une modification de la méthode "deferred correction" (Khosla and Rubin, 1974).

Dans ce chapitre, un schéma compact d'ordre élevé est proposé et comparé à une reconstruction polynomiale sur stencil fixe proposée par Carpentier (2000) et à un schéma classique décentré d'ordre 2. La section 3.1 présente un résumé détaillé des différentes discrétisations volumes finis. En outre, le traitement des conditions aux limites est discuté. Puis la précision des schémas est évaluée pour différents cas tests : le calcul du gradient et du Laplacien sur une solution a priori (section 3.2.1), l'advection d'une tâche gaussienne (section 3.2.2), un tourbillon de Taylor incompressible et bi-dimensionnel (section 3.2.3), une cavité entraînée (section 3.2.4) et enfin, un écoulement tri-dimensionnel autour d'un obstacle cubique (section 3.2.5). Les résultats de ce chapitre font l'objet d'un article soumis à "International Journal for Numerical Methods in Fluids" (Analysis of high-order finite volume schemes for the incompressible Navier Stokes equations).

3.1 Description des méthodes numériques

Pour présenter les différents schémas numériques utilisés, on considère l'équation de transport de convection-diffusion :

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = \nabla \cdot (\nu \nabla \phi) \quad (3.1)$$

où le scalaire $\phi(\mathbf{x}, t)$ est transporté par la vitesse \mathbf{u} et diffusé avec le coefficient de diffusion $\nu \geq 0$. Comme il a déjà été vu dans la section 2.1.4 du chapitre 2, la formulation volumes finis intègre l'équation de transport (3.1) sur chaque cellule C_i . Le théorème de Gauss permet de transformer les intégrales de volume des termes convectifs et diffusifs en intégrales de surface :

$$\dot{\phi}_i = F(\phi) = - \sum_{j \in \text{face}(i)} F_{ij}^{\text{conv}} + \sum_{j \in \text{face}(i)} F_{ij}^{\text{viscous}} \quad (3.2)$$

où ϕ_i est la valeur moyenne de ϕ sur la cellule C_i :

$$\phi_i = \frac{1}{V_i} \int_{C_i} \phi d\Omega$$

et F_{ij}^{conv} , F_{ij}^{viscous} sont respectivement les flux convectifs et diffusifs à l'interface Γ_{ij} entre la cellule C_i et les cellules voisines C_j :

$$F_{ij}^{\text{conv}} = \int_{\Gamma_{ij}} \phi \mathbf{u} \cdot \mathbf{n} d\Gamma, \quad F_{ij}^{\text{viscous}} = \int_{\Gamma_{ij}} \nu \frac{\partial \phi}{\partial n} d\Gamma$$

Sur l'interface Γ_{ij} , les flux F_{ij} sont calculés en utilisant différents schémas basés sur le développement en série de Taylor, la reconstruction polynomiale ou des schémas compacts. L'intégration temporelle de l'Equation (3.2) est réalisée en utilisant des schémas explicites ou implicites. Pour éviter l'utilisation d'une molécule implicite trop grande, les flux implicites d'ordre élevé sont calculés itérativement en utilisant l'approche classique "deferred correction" (Khosla and Rubin, 1974) avec un schéma implicite décentré d'ordre 1 (UDS) :

$$F_{ij}^{HO} = \underbrace{\Delta F_{ij}^{UDS}}_{implicit} + \underbrace{F_{ij}^{HO}}_{explicit} \quad (3.3)$$

L'expression des flux peut s'écrire :

$$\begin{aligned} F_{ij}^{conv} &= \phi_{ij} \mathbf{u}_{ij} \cdot \mathbf{n}_{ij} \\ F_{ij}^{diff} &= \nabla \phi_{ij} \cdot \mathbf{n}_{ij} S_{ij} \end{aligned}$$

Le calcul des valeurs interpolées ϕ_{ij} et du gradient $\nabla \phi_{ij} \cdot \mathbf{n}_{ij}$ sur Γ_{ij} est présenté dans ce qui suit.

3.1.0.1 Reconstruction par développement en série de Taylor

Les valeurs interpolées sur la face Γ_{ij} sont évaluées par le développement en série de Taylor :

$$\begin{aligned} \phi_{ij}^R &= \phi_i + (\mathbf{x}_{ij} - \mathbf{x}_i)^T \nabla \phi_i + \frac{1}{2} (\mathbf{x}_{ij} - \mathbf{x}_i)^T H_i (\mathbf{x}_{ij} - \mathbf{x}_i) + o(h^3) \\ \phi_{ij}^L &= \phi_j + (\mathbf{x}_{ij} - \mathbf{x}_j)^T \nabla \phi_j + \frac{1}{2} (\mathbf{x}_{ij} - \mathbf{x}_j)^T H_j (\mathbf{x}_{ij} - \mathbf{x}_j) + o(h^3) \end{aligned}$$

Les exposants R et L indiquent que le développement en série de Taylor est réalisé autour de la cellule i ou j . Le schéma classique décentré d'ordre 1 (UDS) utilise uniquement la valeur des cellules i et j . Les méthodes d'ordre plus élevé nécessitent le calcul des dérivées. Une approximation du gradient d'ordre 1 est alors nécessaire pour obtenir un schéma d'ordre 2. Les gradients moyens calculés à partir du théorème de Gauss satisfont cette condition et ϕ_{ij} est alors calculé en utilisant une approche décentrée (ϕ_{ij}^R ou ϕ_{ij}^L) ou une approche centrée $\phi_{ij}^c = \frac{1}{2}(\phi_{ij}^R + \phi_{ij}^L)$. Ces deux schémas utilisant les gradients aux centres des cellules conduisent à une approximation d'ordre 2 pour les maillages uniformes. Sur un maillage non uniforme, l'approximation n'est que d'ordre 1, mais avec une erreur de troncature plus faible que celle obtenue avec le schéma UDS. Un moyen pour augmenter la précision du schéma est d'évaluer les dérivées supplémentaires. Vaassen et al. (2006) proposent de calculer le gradient et les dérivées d'ordre 2 par une minimisation des moindres carrés de la différence entre la valeur au centre ϕ_j et la valeur reconstruite à partir du développement en série de Taylor. Cette méthode donne une approximation d'ordre 2 du gradient et d'ordre 1 pour la matrice Hessienne. La section suivante décrit une méthode plus classique pour construire des schémas d'ordre élevé basée sur une reconstruction polynomiale.

3.1.0.2 Schéma à reconstruction polynomiale

Carpentier (2000) a développé une classe de schémas volumes finis décentrés en utilisant une reconstruction polynomiale sur molécule fixe. La molécule est déterminée itérativement en ajoutant les voisins jusqu'à ce qu'une molécule $S(i)$ de taille suffisante ait été trouvée pour chaque volume de contrôle C_i (cf. Figure 3.1). Près des frontières du maillage, les faces frontières sont incluses dans le stencil. Un polynôme multidimensionnel d'ordre 3 est calculé sur la molécule. Le polynôme est écrit comme une somme de n monômes $b_{i,n}(\mathbf{x})$ de degré ≤ 3 :

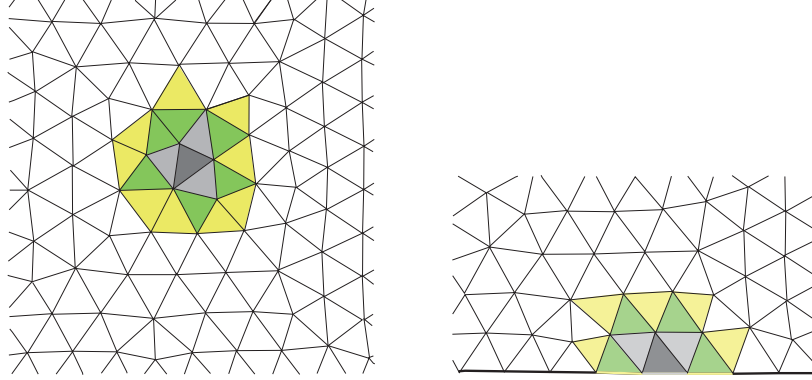


FIGURE 3.1 – Agglomération du stencil.

Degré	1			2						3								
n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
p	1	0	0	2	0	0	1	1	0	3	0	0	2	1	0	0	2	1
q	0	1	0	0	2	0	0	1	1	0	3	0	1	2	2	1	0	0
r	0	0	1	0	0	2	1	0	1	0	0	3	0	0	1	2	1	2

TABLE 3.1 – Puissance des monômes pour le cas 3D.

Degré	1			2			3		
n	1	2	3	4	5	6	7	8	9
p	1	0	2	0	1	3	0	2	1
q	0	1	0	2	1	0	3	1	2

TABLE 3.2 – Puissance des monômes pour le cas 2D.

$$P_i(\mathbf{x}) = \phi_i + \sum_n a_{i,n} (b_{i,n}(\mathbf{x}) - \overline{b_{i,n}})$$

avec

$$b_{i,n}(\mathbf{x}) = (x - x_i)^{p(n)} (y - y_i)^{q(n)} (z - z_i)^{r(n)} \quad \text{avec } p(n) + q(n) + r(n) \leq 3$$

et $\overline{b_{i,n}}$ est la valeur moyenne du monôme $b_{i,n}$ sur la cellule i calculée par intégration numérique (utilisation des points de Gauss).

Les puissances $p(n)$, $q(n)$ et $r(n)$ des monômes pour un polynôme de degré ≤ 3 sont données dans la Table (3.1) pour le cas 3D et la Table (3.2) pour le cas 2D. La forme particulière du polynôme lui permet de vérifier automatiquement la propriété de valeur moyenne sur la cellule C_i :

$$\int_{C_i} P(\mathbf{x}) d\Omega = V_i \overline{\phi_i}$$

où $\overline{\phi_i}$ est la valeur moyenne de la variable sur la cellule, qui est différente à l'ordre 2 de la valeur nodale ϕ_i . Les coefficients $a_{i,n}$ des monômes sont calculés par une minimisation des moindres carrés pondérée de la différence entre la valeur moyenne $\overline{\phi_j}$ et la valeur moyenne du polynôme sur la cellule C_j :

$$\int_{C_j} P_i(\mathbf{x})d\Omega - V_j\overline{\phi_j}$$

L'erreur des moindres carrés s'écrit :

$$H_i = \sum_{j \in S(i)} w_j \left(\int_{C_j} P_i(\mathbf{x})d\Omega - V_j\overline{\phi_j} \right)^2$$

et les poids w_j dépendent de la distance géométrique entre les centres des cellules i et j . Le flux convectif est obtenu par intégration numérique (en utilisant les points de Gauss) du polynôme sur la face Γ_{ij} . Comme les polynômes sont associés à la cellule i , le flux convectif décentré sur Γ_{ij} peut être obtenu en utilisant le polynôme P_i ou P_j . La dérivée normale à la face, requise pour l'évaluation des flux visqueux est obtenue par une moyenne des dérivées normales des polynômes P_i et P_j reconstruits de chaque coté de la face Γ_{ij} . Le problème des moindres carrés est résolu en utilisant une factorisation de Householder QR, et la décomposition en valeurs singulières (SVD) permet de détecter les systèmes des moindres carrés mal conditionnés associés à la dégénérescence du stencil. Dans ce cas, un polynôme d'ordre moins élevé est utilisé.

Pour réduire les temps de calculs, la valeur décentrée sur la face ϕ_{ij}^R (et respectivement ϕ_{ij}^L) est exprimée comme une combinaison linéaire des valeurs $\overline{\phi_k}$ des cellules k du stencil S_i (respectivement S_j). Les stencils et les coefficients obtenus à partir de la méthode des moindres carrés sont calculés lors d'une étape préliminaire et stockés pour une utilisation ultérieure. Cette procédure nécessite un stockage supplémentaire qui est généralement disponible sur les architectures modernes de calculs. La structure de données associée à ces schémas numériques est composé de :

- pour la sauvegarde des molécules S_i , d'un tableau $tab_{molecule}$ d'entiers permettant de sauvegarder les indices des éléments des stencils de chaque cellule et un tableau d'offsets (entier) permet de connaître les indices de début et de fin de la position des éléments du stencil dans le tableau $tab_{stencil}$.

- pour le flux convectif ϕ_{ij}^R (respectivement ϕ_{ij}^L) de la face Γ_{ij} , un tableau de réels tab_{face}^R (respectivement tab_{face}^L) stocke les coefficients de la combinaison linéaire associés aux valeurs $\overline{\phi_k}$ telles que $k \in S_i$ (respectivement $k \in S_j$) et un second tableau d'entier permet de connaître la position des coefficients de la combinaison linéaire dans tab_{face}^R .

Le surcoût mémoire dû au stockage de ces données est donné dans la Table (3.3) pour un maillage 3D tétraédrique de 1 000 000 de cellules, un maillage 2D triangulaire de 1 000 000 de cellules et un maillage 2D structuré de 1 000 0000 de cellules. Le surcoût est important, mais l'utilisation d'un schéma d'ordre élevé doit permettre de diminuer le nombre de cellules du maillage.

De plus, la structure de données est optimisée. En effet, celle présentée précédemment est utilisée pour le calcul des flux sur des faces pour lesquelles le calcul du polynôme ne dépend pas des conditions aux limites. Ce sont les faces situées suffisamment loin des conditions limites. Cette structure de données unique est ainsi utilisée pour calculer les flux de n'importe quelles variables (les trois composantes de la vitesse dans notre cas). Près des frontières du maillage, la reconstruction polynomiale dépend des

	taille du stencil	nombre de faces par cellules	mémoire (Mo)
2D Triangle	9	3	292
2D Map	12	4	484
3D Brick	24	6	1324

TABLE 3.3 – Coût mémoire pour le schéma polynomiale de degré 3.

conditions aux limites et est donc spécifique à chaque variable. Une structure de données similaire à celle présentée est ainsi utilisée pour chaque variable, mais sa taille est généralement largement inférieure à celle utilisée pour les faces internes. La distinction entre les flux des faces internes et les flux des faces affectées par les conditions aux limites évite la redondance d'informations et permet d'optimiser le stockage mémoire.

3.1.0.2.1 Condition aux limites Les conditions aux limites sont imposées en contraignant la reconstruction des moindres carrés sur les faces adjacentes à la frontière. Une condition au limite de type Neumann sur la face Γ_b est imposée par la minimisation de la différence entre le gradient normal et la valeur reconstruite du gradient :

$$-\int_{\Gamma_b} \nabla \phi_b + \int_{\Gamma_b} \nabla P_i(\mathbf{x})$$

et une condition de type Dirichlet, par la minimisation de la différence entre la valeur moyenne sur la face, et la valeur moyenne reconstruite à partir du polynôme :

$$-\int_{\Gamma_b} \phi + \int_{\Gamma_b} P_i(\mathbf{x})$$

Les conditions aux limites de type symétrie sont traitées en utilisant des cellules “ghosts” symétriques.

3.1.0.3 Schéma Padé

Un inconvénient des schémas polynomiaux est qu'ils nécessitent l'utilisation d'une molécule de calcul de taille importante. La taille de la molécule peut être réduit aux plus proches voisins en utilisant des schémas compacts. On considère une reconstruction d'ordre élevé de ϕ_{ij} sur la face $\Gamma_{i,i+1}$, fonction des valeurs moyennes $\{\overline{\phi}_i, \overline{\phi}_{i+1}\}$ et des gradients moyens $\{\overline{\nabla\phi}|_i, \overline{\nabla\phi}|_{i+1}\}$ de part et d'autre de $\Gamma_{i,i+1}$. La relation générale s'écrit :

$$\phi_{ij} = a_0 \overline{\phi}_i + a_1 \overline{\phi}_{i+1} + b_0 \overline{\nabla\phi}|_i \cdot \mathbf{n}_{i,i+1} - b_1 \overline{\nabla\phi}|_{i+1} \cdot \mathbf{n}_{i,i+1} \quad (3.4)$$

où $\mathbf{n}_{i,i+1}$ est la normal unitaire à la face $\Gamma_{i,i+1}$. Pour des raisons de symétrie, la valeur interpolée sur la face opposée $\phi_{i,i-1}$ s'écrit :

$$\phi_{i-\frac{1}{2}} = a_0(h_{i-1}, h_i) \overline{\phi}_{i-1} + a_1(h_{i-1}, h_i) \overline{\phi}_i + b_0(h_{i-1}, h_i) \overline{\nabla\phi}|_{i-1} \cdot \mathbf{n}_{i,i+1} - b_1(h_{i-1}, h_i) \overline{\nabla\phi}|_i \cdot \mathbf{n}_{i,i+1} \quad (3.5)$$

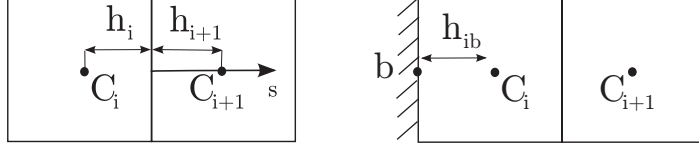


FIGURE 3.2 – Notation du maillage.

h_i, h_{i-1} étant définis sur la Figure (3.2). Le théorème de Gauss permet de relier le gradient moyen $\overline{\nabla\phi}_i$ aux valeurs sur les faces :

$$\overline{\nabla\phi}_i 2h_i = \phi_{i+\frac{1}{2}} - \phi_{i-\frac{1}{2}} \quad (3.6)$$

En substituant les équations (3.4) et (3.5) dans la relation (3.6), on obtient la relation de Padé reliant les valeurs moyennes $\{\overline{\phi}_{i-1}, \overline{\phi}_i, \overline{\phi}_{i+1}\}$ et les gradients moyens $\{\overline{\nabla\phi}|_{i-1}, \overline{\nabla\phi}|_i, \overline{\nabla\phi}|_{i+1}\}$:

$$\begin{aligned} b_0(h_{i-1}, h_i) \overline{\nabla\phi}|_{i-1} + (2h_i - b_0(h_i, h_{i+1}) - b_1(h_{i-1}, h_i)) \overline{\nabla\phi}|_i + b_1(h_i, h_{i+1}) \overline{\nabla\phi}|_{i+1} \\ = -a_0(h_{i-1}, h_i) \overline{\phi}_{i-1} + (a_0(h_i, h_{i+1}) - a_1(h_{i-1}, h_i)) \overline{\phi}_i + a_1(h_i, h_{i+1}) \overline{\phi}_{i+1} \end{aligned} \quad (3.7)$$

Les quatre coefficients $\{a_0, a_1, b_0, b_1\}$ sont tels que la relation (3.7) soit vérifiée par la solution exacte $\phi^e(x)$ jusqu'à l'ordre 4 pour les maillages uniformes et structurés. La solution peut être obtenue en réalisant un développement en série de Taylor de $\phi^e(x)$. La substitution dans le développement en série de Taylor de la valeur moyenne $\overline{\phi^e}|_i$ et de la dérivée moyenne $\overline{\nabla\phi^e}|_i$ permet d'obtenir l'erreur de troncature de la relation (3.7). L'annulation des quatre premiers termes conduit à quatre équations, qui associées à des propriétés de symétrie, permettent de déterminer les quatre coefficients $\{a_0, a_1, b_0, b_1\}$. Une méthode équivalente pour calculer les coefficients a_0, a_1, b_0, b_1 est d'utiliser une interpolation polynomiale d'ordre élevé $P_{i,i+1}(s)$, avec s la distance à la face dans la direction normale $\mathbf{n}_{i,i+1}$. (cf. Figure 3.2). Le polynôme doit vérifier les conditions suivantes :

$$\begin{aligned} \int_{C_i} P_{i,i+1}(s) &= \overline{\phi}_i \\ \int_{C_j} P_{i,i+1}(s) &= \overline{\phi}_j \\ \int_{C_i} \frac{\partial P_{i,i+1}(s)}{\partial s} &= \overline{\nabla\phi}_i \cdot \mathbf{n}_{i,i+1} \\ \int_{C_j} \frac{\partial P_{i,i+1}(s)}{\partial s} &= \overline{\nabla\phi}_{i+1} \cdot \mathbf{n}_{i,i+1} \end{aligned}$$

Ces quatre conditions déterminent un polynôme unique de degré 3. La valeur sur la face et le gradient normal sont calculés à partir de ce polynôme par :

$$\phi_{i,i+1} = P_{i,i+1}(0), \quad \frac{\partial\phi}{\partial n}\Big|_{i,i+1} = \frac{d}{ds} P_{i,i+1}(0)$$

Pour des maillages non uniformes avec des cellules rectangulaires, on obtient la relation suivante :

Algorithme 4 Méthode de Jacobi pour schéma Padé.

1. Calcul d'un premier gradient en utilisant le théorème de Gauss.

2. Tant que $\| \nabla \phi_i^{p-1} - \nabla \phi_i^p \| > \epsilon$

Calcul de la nouvelle valeur ϕ_{ij}^p en utilisant les gradients précédents $\overline{\nabla \phi}_i^{p-1}$ dans l'Equation (3.4).

Mise à jour du gradient $\overline{\nabla \phi}_i^p$ en utilisant le théorème de Gauss et la valeur ϕ_{ij}^p de la face.

$$\phi_{ij} = a_0(h_i, h_j) \overline{\phi}_i + a_1(h_i, h_j) \overline{\phi}_j + b_0(h_i, h_j) \overline{\nabla \phi}_i \cdot \mathbf{n}_{ij} - b_1(h_i, h_j) \overline{\nabla \phi}_j \cdot \mathbf{n}_{ij} \quad (3.8)$$

avec

$$\begin{aligned} a_0(h_i, h_j) &= \frac{h_j^2(h_j + 2h_i)}{h_i^3 + 2h_i h_j^2 + 2h_i^2 h_j + h_j^3}, & a_1(h_i, h_j) &= a_0(h_j, h_i) \\ b_0(h_i, h_j) &= \frac{1}{2} \frac{h_i h_j^2}{h_j^2 + h_i h_j + h_i^2}, & b_1(h_i, h_j) &= -b_0(h_j, h_i) \end{aligned}$$

Pour des maillages structurés et non uniformes, avec un facteur d'expansion r , l'erreur de troncature est :

$$\frac{(1-r^4)(1+r)^2}{15r^2(1+r+r^2)} 0(h^3) \approx \frac{16}{45} (1-r) h^4$$

Une approche similaire est utilisée pour la reconstruction compacte des dérivées normales $\frac{d\phi}{dn_{ij}}$ pour les flux diffusifs. La relation s'écrit :

$$\frac{d\phi_{ij}}{dn_{ij}} = c_0(h_i, h_j) \overline{\phi}_i + c_1(h_i, h_j) \overline{\phi}_j + d_0(h_i, h_j) \overline{\nabla \phi}_i \cdot \mathbf{n}_{ij} + d_1(h_i, h_j) \overline{\nabla \phi}_j \cdot \mathbf{n}_{ij} \quad (3.9)$$

avec

$$\begin{aligned} c_0(h_i, h_j) &= -\frac{6h_i h_j}{h_i^3 + 2h_i h_j^2 + 2h_i^2 h_j + h_j^3}, & c_1(h_i, h_j) &= -c_0(h_i, h_j) \\ d_0(h_i, h_j) &= \frac{h_j^3 - h_i h_j^2 - 3h_j h_i^2}{h_i^3 + 2h_i h_j^2 + 2h_i^2 h_j + h_j^3}, & d_1(h_i, h_j) &= d_0(h_j, h_i) \end{aligned}$$

Le gradient moyen $\overline{\nabla \phi}_i$ est relié à ϕ_{ij} par l'intermédiaire du théorème de Gauss et est solution d'un système tri-diagonale. Sur un maillage uniforme, la relation (3.8) est équivalente à la relation de Padé classique (M.Piller and E.Stalio (2008)) :

$$\frac{1}{4} u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}} + \frac{1}{4} u_{i+\frac{3}{2}} = \frac{3}{4} u_i + \frac{3}{4} u_{i+1} \quad (3.10)$$

Une première méthode pour calculer le gradient moyen consiste à utiliser une méthode de Jacobi sans matrice (Algorithme 4). Cette méthode n'est pas la plus efficace. La résolution du système linéaire formé par les relations de Padé, par une méthode itérative multigrille est préférée et décrite par la suite (paragraphe p52).

Condition aux limites Les conditions aux limites affectent l'erreur de troncature d'ordre 4 du gradient moyen, ainsi que l'ordre 4 du Laplacien. Pour une condition de type Neumann, ou une

condition de périodicité, l'utilisation de cellules "ghosts" symétriques permet de calculer les flux convectifs et diffusives avec une précision identique à celle des cellules intérieures. Pour une condition de type Dirichlet, ϕ_b sur la face frontière Γ_b de la cellule C_i est donné. Sur la face opposée, ϕ_{ij} a une précision d'ordre 4. Le gradient moyen sur la cellule C_i est alors seulement d'ordre 3 et se propage des cellules frontières vers l'intérieur du maillage. Suffisamment loin des frontières, le gradient retrouve une précision d'ordre 4.

La précision de l'opérateur de Laplace sur les cellules adjacentes à la frontière dépend du calcul de la dérivée normale sur la face Γ_b . Une évaluation précise de la dérivée normale sur la face frontière est nécessaire. Pour obtenir une discrétisation consistante, une approximation du second ordre de la dérivée normale est requise, donnant un Laplacien d'ordre 1 sur un maillage non uniforme et d'ordre 2 sur un maillage uniforme. L'utilisation d'une dérivée normale trop simple comme :

$$\frac{\partial \phi}{\partial n} = \frac{(\phi_b - \phi_i)}{h_{ib}}$$

où h_{ib} représente la distance du noeud i à la frontière, conduit à une dérivée normale d'ordre 1 et à un opérateur de Laplace inconsistant. L'utilisation d'un polynôme $P_b(s)$ d'ordre 2 satisfaisant les conditions :

$$\begin{aligned} P_b(0) &= \phi_b \\ \int_{C_i} P_b(s) d\Omega &= V_i \bar{\phi}_i \end{aligned} \quad (3.11)$$

$$\int_{C_i} \frac{d}{ds} P_b(s) d\Omega = V_i \overline{\nabla \phi}_i \cdot \mathbf{n}_{ij} \quad (3.12)$$

donne une dérivée normale d'ordre 2 et un Laplacien moyen d'ordre 1. La dérivée normale s'écrit pour les maillages uniformes :

$$\left. \frac{dP}{ds} \right|_{s=0} = \frac{\partial \phi_{ij}}{\partial n} = \frac{3}{h_{ib}} (\phi_b - \bar{\phi}_i) - 2 \overline{\nabla \phi}_i \cdot \mathbf{n}_{ij} \quad (3.13)$$

Pour améliorer la précision, un polynôme d'ordre 4 $P_b(s)$ peut être utilisé. Il doit satisfaire :

$$\begin{aligned} P_b(0) &= \phi_b \\ \int_{C_i} P_b(s) &= \bar{\phi}_i \quad , \quad \int_{C_{i+1}} P_b(s) = \bar{\phi}_{i+1} \\ \int_{C_i} \frac{\partial P_b(s)}{\partial s} &= \overline{\nabla \phi}_i \cdot \mathbf{n}_b \quad , \quad \int_{C_{i+1}} \frac{\partial P_b(s)}{\partial s} = \overline{\nabla \phi}_{i+1} \cdot \mathbf{n}_b \end{aligned}$$

où la cellule C_{i+1} est pour un maillage cartésien, la seconde cellule distante de la face frontière (cf. Figure 3.2). La dérivée normale devient :

$$\left. \frac{dP}{ds} \right|_{s=0} = \frac{\partial \phi}{\partial n} = A\phi_b + B\phi_i + C\phi_{i+1} + D\overline{\nabla \phi}_i \cdot \mathbf{n}_b + E\overline{\nabla \phi}_{i+1} \cdot \mathbf{n}_b$$

$$\begin{aligned}
A &= -\frac{2h_i^3 + 7h_i h_{i+1}^2 + 3h_i^2 h_{i+1} + 3h_{i+1}^3}{h_i(h_i + h_{i+1})h_{i+1}^2} \\
B &= \frac{3h_{i+1}^2 + 10h_i h_{i+1} + 10h_i^2}{(h_i + h_{i+1})^2 h_i} \\
C &= \frac{h_i^2(5h_{i+1} + 2h_i)}{h_{i+1}^2(h_i + h_{i+1})^2} \\
D &= -\frac{-2(h_{i+1}^3 + 3h_i h_{i+1}^2 + 3h_i^2 h_{i+1} + 2h_i^3)}{(h_i + h_{i+1})h_{i+1}^2} \\
E &= \frac{2h_i^2}{h_{i+1}(h_i + h_{i+1})}
\end{aligned}$$

L'utilisation d'un polynôme $P_b(s)$ d'ordre 4 donne une dérivée normale d'ordre 4 et un Laplacien moyen d'ordre 3 sur la cellule C_i . Une amélioration de la précision de l'opérateur de Laplace au delà du troisième ordre, avec un polynôme $P_b(s)$ d'ordre plus élevé, n'est pas possible car cela dépend également de la précision de l'approximation de la dérivée normale sur les faces intérieures qui est seulement d'ordre 4.

3.1.1 Flux convectifs avec une vitesse non uniforme

Le calcul des termes convectifs dans l'équation de Navier Stokes nécessite une attention particulière comme le soulignent Pereira et al. (2001) pour les maillages cartésiens. L'approximation de $\int_{\Gamma_{ij}} \mathbf{u} \cdot \mathbf{n} u d\Gamma$ par $\bar{u}_{ij} \int_{\Gamma_{ij}} \mathbf{u} \cdot \mathbf{n} d\Gamma$ est uniquement du second ordre, à cause des variations de la vitesse le long de la face. Une information supplémentaire est nécessaire pour prendre en compte les variations de la vitesse le long de la face. La comparaison du développement en série de Taylor de $\int_{\Gamma_{ij}} \mathbf{u} \cdot \mathbf{n} u d\Gamma$ et $\bar{u}_{ij} \int_{\Gamma_{ij}} \mathbf{u} \cdot \mathbf{n} d\Gamma$ autour du centre c_{ij} de la face Γ_{ij} , à l'ordre 4 donne la relation suivante :

$$\int_{\Gamma_{ij}} \mathbf{u} \cdot \mathbf{n} u d\Gamma = u_{ij} \int_{\Gamma_{ij}} \mathbf{u} \cdot \mathbf{n} d\Gamma + (\nabla_{\parallel}(\mathbf{u} \cdot \mathbf{n})) \mathcal{I}(\nabla_{\parallel} u) + D \times o(h^3) + o(h^4)$$

où $\nabla_{\parallel} = \nabla - (\nabla \cdot \mathbf{n})$ est la projection du gradient sur la face et \mathcal{I} la matrice d'inertie de la face. Le coefficient D contient un produit du gradient et de la Hessienne et disparaît pour les faces rectangulaires (maillage 3D). Ainsi pour les schémas d'ordre élevé, les flux convectifs sont approximés par :

$$\int_{\Gamma_{ij}} \mathbf{u} \cdot \mathbf{n} \phi d\Gamma \approx \phi_{ij} \mathbf{u}_{ij} \cdot \mathbf{n}_{ij} S_{ij} + (\nabla_{\parallel} \phi) \mathcal{I}(\nabla_{\parallel}(\mathbf{u} \cdot \mathbf{n})) \quad (3.14)$$

où ϕ_{ij} et \mathbf{u}_{ij} sont les interpolations d'ordre élevé. Une approximation du second ordre des dérivées $\nabla_{\parallel} \phi$, $\nabla_{\parallel}(\mathbf{u} \cdot \mathbf{n})$ est suffisante pour obtenir la précision souhaitée et est donnée par :

$$\nabla_{\parallel} \phi = \frac{\nabla_{\parallel} \phi_i + \nabla_{\parallel} \phi_j}{2}$$

et identiquement pour $\mathbf{u} \cdot \mathbf{n}$.

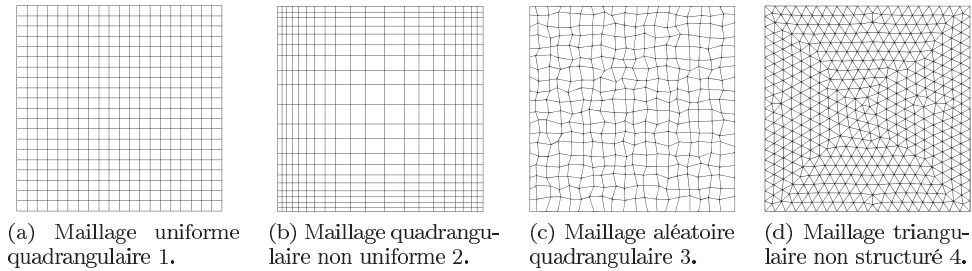


FIGURE 3.3 – Maillages.

3.2 Validation numérique

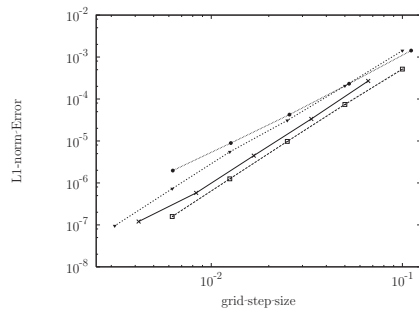
Dans cette section, diverses expériences numériques sont présentées dans le but de vérifier la précision numérique des schémas présentés dans la section 3.1. L'expérimentation est réalisée sur les différents maillages de la Figure (3.3).

3.2.1 Calcul du Laplacien et le gradient sur une fonction a priori

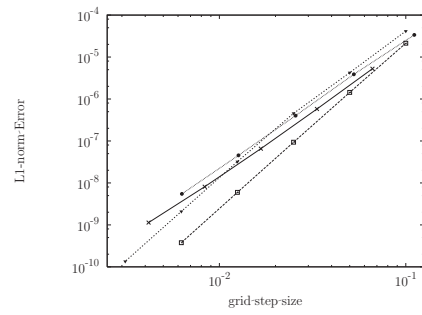
Pour vérifier la précision des schémas, la valeur moyenne $\overline{\phi_i}$ est calculée à partir d'une fonction analytique $\phi(x) = e^x$ et le gradient et le Laplacien sont évalués en utilisant le théorème de Gauss. Les résultats sont comparés avec la valeur moyenne exacte du gradient et du Laplacien. Le gradient et le Laplacien sont calculés sur une séquence de grilles raffinées d'une géométrie carrée unitaire avec une condition de Dirichlet en $x = 0$ et $x = 1$, et une condition de type Neumann en $y = 0$ et $y = 1$.

Les résultats sont présentés pour les trois schémas étudiés, sur les quatre maillages de la Figure (3.3). Le gradient calculé à partir du schéma polynomial a une précision d'ordre 3 ou 4, selon la régularité du maillage (cf. Figure 3.4b). La précision du Laplacien (cf. Figure 3.4a) est inférieure d'un ordre à celle du gradient. Le gradient calculé avec le schéma compact (cf. Figure 3.4d) a une précision d'ordre 4 sur le maillage cartésien uniforme (1), mais donne des résultats inconsistants avec les maillages non cartésiens (3) et (4). Malgré une précision locale théorique d'ordre 3 sur le maillage (3), la précision globale d'ordre 4 des schémas compact et polynomial provient de la variation contrôlée de la taille du maillage et du raffinement uniforme. Ainsi le schéma polynomial et le schéma Padé donnent des meilleurs résultats que le schéma standard : le gradient standard (cf. Figure 3.4f) est d'ordre 2 avec le maillage quadrangulaire uniforme (1) mais seulement d'ordre 1 pour le maillage triangulaire non structuré (4) et le maillage quadrangulaire non uniforme (2). Il est même inconsistant pour le maillage quadrangulaire aléatoire (3). Le Laplacien standard (cf. Figure 3.4e) est d'ordre 2 sur le maillage quadrangulaire uniforme (1), d'ordre 1 sur le maillage quadrangulaire non-uniforme (2) mais inconsistant sur les maillage (3) et (4).

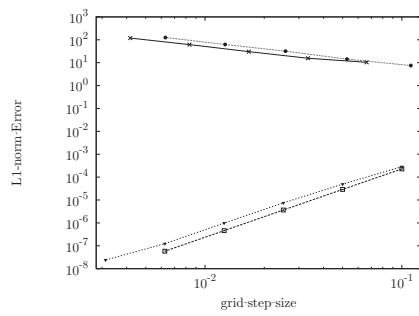
L'utilisation d'une fonction test $\phi(x, y) = \sin(\pi x)\cos(\pi y)$ similaire à celle utilisée par Ollivier-Gooch et Altena (2002) donne un résultat avec une meilleure précision près des frontières, car la fonction test satisfait $\frac{\partial^2 w}{\partial n^2} = 0_b$ à la frontière. Dans ce cas, le gradient calculé avec le schéma compact est d'ordre 4 dans tout le domaine.



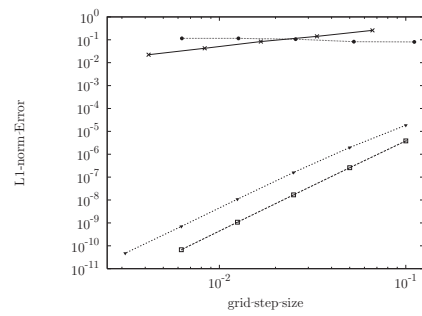
(a) Laplacien polynomial.



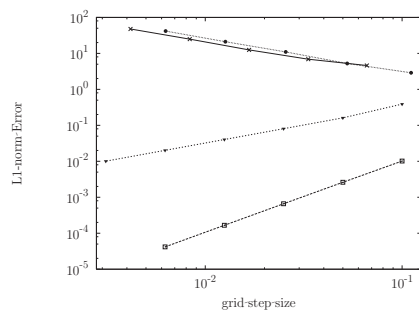
(b) Gradient polynomial.



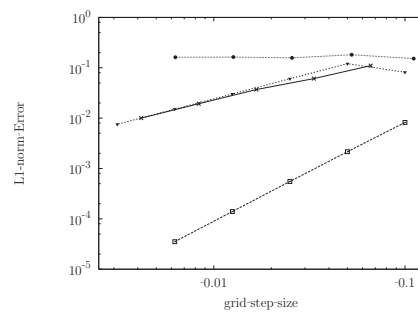
(c) Laplacien compact.



(d) Gradient compact.



(e) Laplacien standard.



(f) Gradient standard.

FIGURE 3.4 – Résultats pour le calcul du Laplacien et du Gradient : \square maillage quadrangulaire uniforme ; ∇ maillage quadrangulaire non uniforme ; \bullet maillage quadrangulaire aléatoire ; \times maillage triangulaire non structuré.

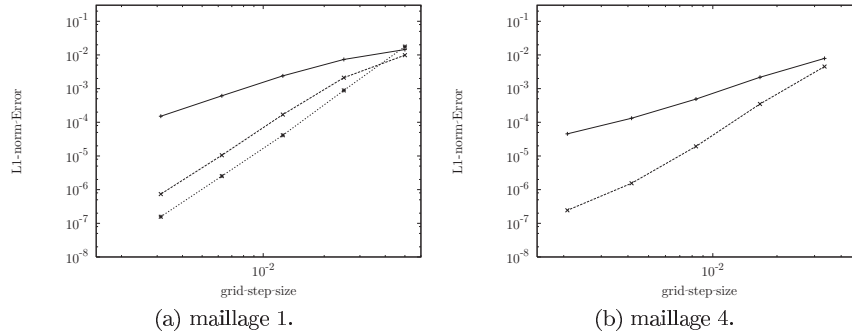


FIGURE 3.5 – Taux de convergence pour la convection du cône gaussien : + décentré d’ordre 2 ; * Padé ; × polynomial.

3.2.2 Advection d’un cône gaussien dans un domaine carré

La précision et le taux de convergence des différents flux numériques convectifs sont évalués sur le test classique d’advection d’une tache gaussienne. Dans la géométrie carrée choisie, de longueur 1.5, l’écoulement est une rotation de corps solide, avec une vitesse angulaire $\omega = 1$ et le champ initial est donné par :

$$\phi(\mathbf{x}, 0) = \exp\left(\frac{\|\mathbf{x} - \mathbf{x}_c\|^2}{\sigma^2}\right)$$

où $\sigma = 0.1$ et $\mathbf{x}_c = [0, -0.25]$. Un schéma Runge Kutta explicite d’ordre 4 est utilisé avec un nombre de Courant fixe, égale à 0.35. Le taux de convergence est affiché sur la Figure (3.5). Comme attendu, les schémas Padé et polynomial ont une précision d’ordre 4 sur un maillage régulier (cf. Figure 3.5a). Le schéma décentré a une précision d’ordre 2 sur les maillages uniforme quadrangulaire (1) et triangulaire non structuré (4). Sur le maillage (4), le schéma polynomial a une précision d’ordre 3. Il est important de noter que l’utilisation de la relation (3.14) est nécessaire pour atteindre une précision d’ordre 3. Enfin, sur un maillage uniforme non structuré (maillage triangulaire composé de triangles équilatéraux), le schéma polynomial retrouve une précision d’ordre 4.

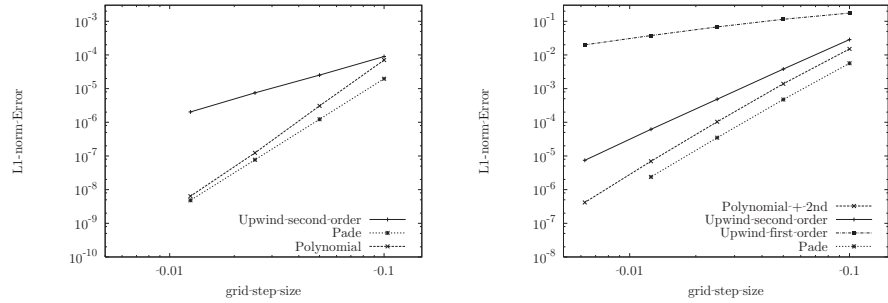
3.2.3 Décroissance visqueuse d’un tourbillon de Taylor

Le problème testé est la décroissance d’un tourbillon de Taylor. Pour cet écoulement, une solution exacte des équations de Navier Stokes pour écoulement incompressible est disponible :

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}(\mathbf{x}, 0)e^{-2\pi^2\nu t}$$

$$\mathbf{u}(\mathbf{x}, 0) = \left(\frac{\partial\psi_0}{\partial y}, -\frac{\partial\psi_0}{\partial x}\right) \quad \text{avec} \quad \psi_0 = \frac{1}{\pi}\sin(\pi x)\sin(\pi y) \quad (3.15)$$

Sur la géométrie unitaire de forme carrée, la condition initiale est donnée par la solution exacte (3.15) et une condition de mur glissant est utilisée pour la simulation. Les solutions numériques sont comparées à la solution exacte pour deux nombres de Reynolds $Re = 10$ et $Re = 10000$. Les simulations sont réalisées



(a) $Re = 10$ sur un maillage quadrangulaire uni- (b) $Re = 10000$ sur un maillage quadrangulaire uniforme.

FIGURE 3.6 – Norme L_1 de l’erreur sur la vitesse pour l’écoulement de Taylor.

en utilisant une intégration implicite en temps d’ordre 2 (Gear, 1971), et les résultats sont analysés à $t = 0.05$ pour $Re = 10$ et $t = 1$ pour $Re = 10000$. A chaque pas de temps, les équations non linéaires de Navier Stokes sont résolues en utilisant la méthode de projection SIMPLE (Patankar and Spalding (1972b)) présentée précédemment, permettant de découpler le calcul des champs de vitesse et de pression. Pour vérifier la précision en espace des schémas, le pas de temps est divisé par quatre à chaque raffinement du maillage. Le solveur Navier Stokes est implémenté dans un code efficace de calcul 3D parallèle (Montagnier et al., 2008) et utilise une méthode algébrique multigrille pour la résolution des systèmes linéaires (Lawrence Livermore National Laboratory, 2006). A chaque pas de temps, la norme des résidus décroît de 9 ordres de grandeurs, et une seule itération du solveur multigrille algébrique est nécessaire pour résoudre les systèmes linéaires.

L’évolution de la norme L_1 de l’erreur $\|u - u_0\|_1$ en fonction de la taille du maillage est présentée dans la Figure (3.6). Pour les écoulements à faible nombre de Reynolds (cf. Figure 3.6a), la précision est donnée par la précision du schéma volumes finis des flux diffusifs : ordre 4 pour le schéma Padé et polynomial, et seulement de l’ordre 2 pour la discrétisation standard. L’utilisation du schéma standard pour les flux visqueux avec des schémas convectifs d’ordre élevé aurait donné une précision d’ordre 2 seulement. Comme le montre la Figure (3.6b), les schémas convectifs d’ordre élevé pour les écoulements à grands nombres de Reynolds ne nécessitent pas l’emploi d’un schéma d’ordre élevé pour les flux diffusifs. Le schéma polynomial d’ordre 4 pour les flux convectifs donne des résultats similaires en utilisant soit la discrétisation standard, soit l’approche polynomiale pour les flux diffusifs.

3.2.4 Cavité entraînée

Le dernier cas test bi-dimensionnel est celui de la cavité entraînée qui a été étudié par de nombreux auteurs pour une large gamme de nombres de Reynolds. Pour le nombre de Reynolds choisi, $Re = 1000$, l’écoulement est laminaire et stationnaire et les résultats provenant d’une méthode spectrale DNS (Botella and Peyret (1998)) sont utilisés comme référence.

L’écoulement est simulé sur le maillage uniforme (1) et le maillage non uniforme (4). Le schéma polynomial pour les flux convectifs est utilisé soit avec la discrétisation standard ou soit avec la discrétisation d’ordre élevé pour les flux visqueux. Les profils des vitesses horizontale et verticale obtenus

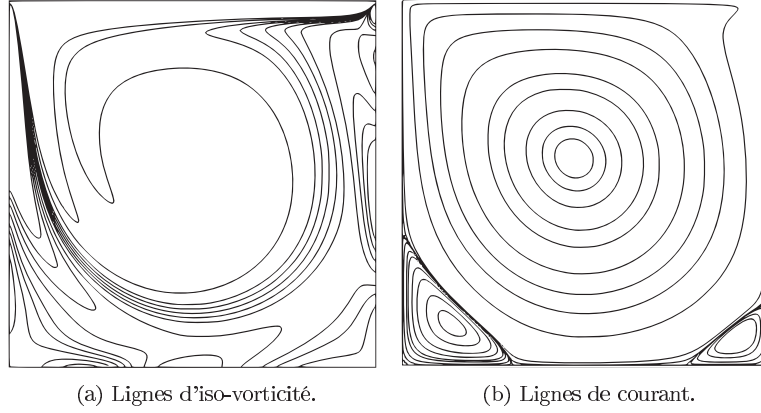


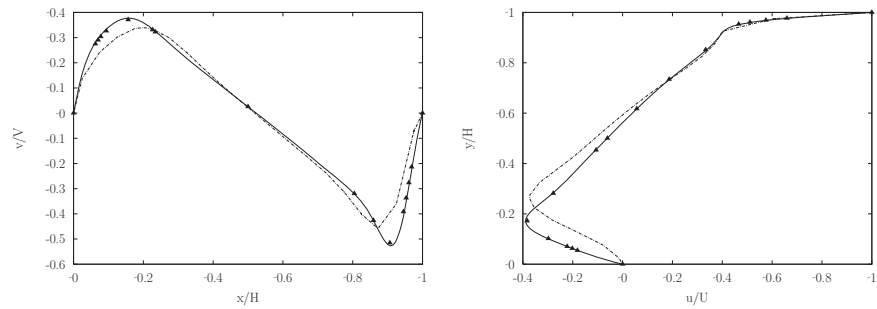
FIGURE 3.7 – Ligne de courant et d'iso-vorticité pour la cavité entraînée $Re = 1000$, obtenues avec le schéma polynomial et un maillage quadrangulaire 160×160 .

par les différentes discrétisations sur le maillage (1) sont comparés sur la Figure (3.8). Les iso-valeurs des fonctions de courant et de vorticité, obtenues avec le schéma polynomial sur la grille fine 160×160 sont affichées sur la Figure (3.7) et les profils des vitesses verticale et horizontale (cf. Figure 3.8) permettent de vérifier la consistance des schémas numériques testés : tout les schémas convergent vers la solution de référence de Botella and Peyret (1998). Les profils des vitesses obtenus avec le schéma polynomial et le schéma Padé sur le maillage le plus grossier 20×20 sont nettement supérieurs à ceux obtenus avec le schéma décentré d'ordre 2. Cependant, le schéma Padé présente des oscillations (“wiggles”) sur ce maillage qui n'est pas assez grossier. Ce problème est récurrent avec les schémas centrés d'ordre élevé. L'utilisation d'un filtre d'ordre élevé (Lele, 1992) est une pratique répandue permettant de filtrer les effets d'aliasing. Enfin, les profils des vitesses obtenus avec un schéma standard pour les termes visqueux (cf. Figure 3.8b) ne présentent pas de différences significatives avec ceux obtenus avec le schéma polynomial pour les flux visqueux (cf. Figure 3.8c). Ainsi la comparaison des profils des vitesses permet de vérifier la consistance des schémas, mais n'est pas suffisante pour une analyse précise de la précision.

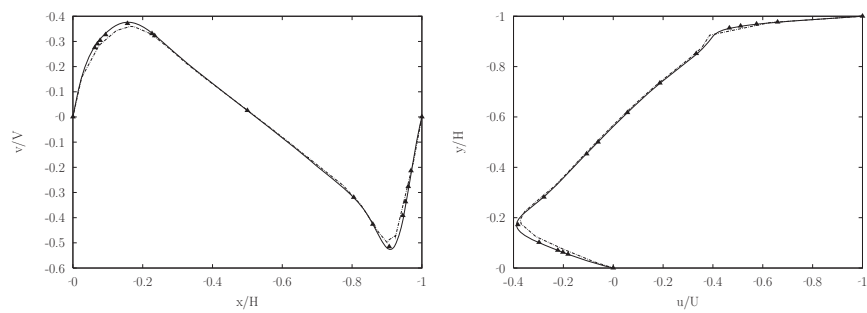
La reconstruction polynomiale pour les flux visqueux montre un manque de robustesse sur certains maillages non structurés. Pour analyser le manque de positivité, propriété cruciale pour les flux visqueux, Coirier (1994) propose de caractériser la positivité en fonction de $\tilde{\alpha}_{min}$ et des coefficients α_i du Laplacien discrétisé $\Delta^h u$:

$$\Delta^h u_i = \sum_{k=0}^N \alpha_k u_k \quad \text{avec} \quad \tilde{\alpha}_{min} = \frac{\min(\alpha_k, 0)}{\sqrt{(\sum_{k=0}^N \frac{\alpha_k^2}{N})}}, \quad k > 0$$

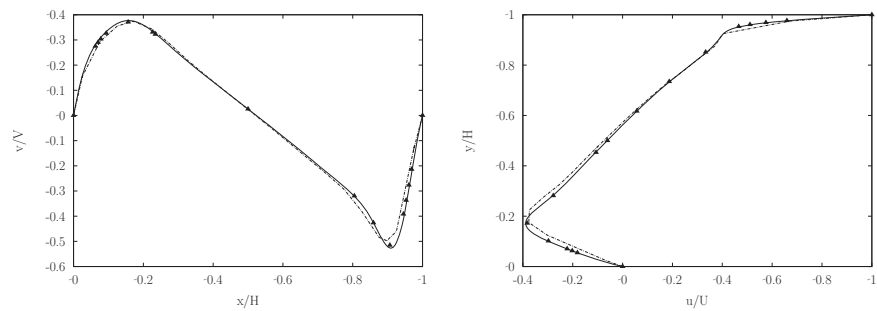
Pour être positif, le schéma doit vérifier $\alpha_0 < 0$ et $\tilde{\alpha}_{min} > 0$. L'analyse de ces coefficients (cf. Table 3.4) pour la reconstruction polynomiale montre que $\alpha_0 < 0$ mais $\tilde{\alpha}_{min} < 0$ avec une valeur absolue maximum de $\tilde{\alpha}_{min}$ pour les maillages non structurés triangulaires. Comme le soulignent Ivan and Groth (2007), la positivité peut être améliorée en utilisant une fonction poids inversement proportionnelle à la distance au carrée. Dans notre cas, l'utilisation d'une fonction poids exponentiel $e^{(-\|x\|/\sigma^2)}$ avec



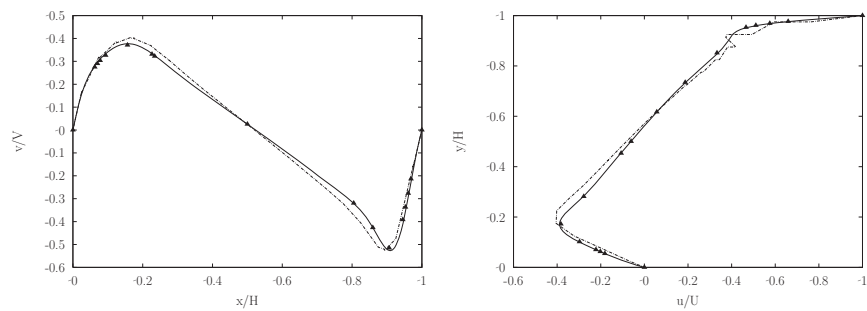
(a) Décentré d'ordre 2.



(b) Schéma polynomiale avec flux visqueux standards.



(c) Schéma polynomial.



(d) Schéma compact.

FIGURE 3.8 – Comparaison des profils des vitesses horizontale (gauche) et verticale (droite) pour la cavité entraînée $Re=1000$: \blacktriangle Spectral DNS Botella and Peyret (1998) ; — maillage quadrangulaire uniforme 160×160 ; --- maillage quadrangulaire uniforme 20×20 .

	3D Tri		Quad	
Poids	$\frac{1}{d^2}$	$\frac{1}{d^3}$	$\frac{1}{d^2}$	$\frac{1}{d^3}$
α_0	-0.292	0.361	-0.330	-0.360
$\tilde{\alpha}_{min}$	-2.53	-1.98	-0.488	-0.414

TABLE 3.4 – Positivité moyenne du schéma polynomial pour les flux visqueux sur différents maillages.

	x-velocity				y-velocity			
Scheme	$R_n\%$	p	σ_p	e_D^*	$R_n\%$	p	σ_p	e_D^*
Ordre 2	76.8	2.33	1.72	$2.25 \cdot 10^{-3}$	73.00	2.05	1.96	$2.23 \cdot 10^{-3}$
EnoS	73.5	1.91	2.06	$9.59 \cdot 10^{-4}$	73.75	1.72	1.70	$1.17 \cdot 10^{-3}$
Eno	69.0	3.84	2.17	$9.52 \cdot 10^{-5}$	58.25	3.32	2.08	$1.16 \cdot 10^{-4}$
Pade	76.0	4.13	1.64	$1.16 \cdot 10^{-4}$	73.25	3.59	1.82	$2.22 \cdot 10^{-4}$

TABLE 3.5 – Taux de convergence sur les maillages quadrangulaires 20×20 , 40×40 , 80×80 (Polynomial S : schéma polynomiale avec flux visqueux standard).

$\sigma = 1.4$ ou inversement proportionnelle à la distance au cube est nécessaire pour obtenir un résultat convergé. Ce résultat est cohérent avec les valeurs de $\tilde{\alpha}_{min}$ observées dans la Table (3.4).

3.2.4.1 Ordre de Convergence

Pour estimer l'ordre de convergence des résultats présentés, les calculs ont été réalisés sur trois maillages uniformément raffinés $\mathcal{M}_1(20 \times 20)$, $\mathcal{M}_2(40 \times 40)$ et $\mathcal{M}_3(80 \times 80)$. L'estimation de l'ordre de convergence est basée sur la théorie d'extrapolation de Richardson (Cadafalch et al., 2002). Lorsque la solution se situe dans la plage de comportement asymptotique, l'erreur locale absolue peut être exprimée par :

$$e(x) = |C_p(x)h^p|$$

L'ordre de précision p est supposé dépendant de la position x et peut être calculé à partir des trois solutions $\phi_1(x)$, $\phi_2(x)$, $\phi_3(x)$ obtenues sur les trois maillages :

$$p(x) = \frac{\ln((\phi_1(x) - \phi_2(x))/(\phi_2(x) - \phi_3(x)))}{\ln(2)}$$

L'ordre de précision peut être déterminé uniquement pour les points de Richardson où la convergence est monotone :

$$(\phi_1^*(x) - \phi_2^*(x))(\phi_2^*(x) - \phi_3^*(x)) > C_0$$

où l'indice * indique que la solution a été normée par la valeur maximale de $\phi(x)$, et C_0 est un faible coefficient positif ($C_0 = 10^{-30}$). Le taux de convergence global p est obtenu par une moyenne sur l'ensemble des cellules de Richardson. Le pourcentage de cellules de Richardson (R_n), l'ordre de convergence global p, l'écart type des valeurs de $p(x)$ par rapport à la valeur moyenne p, et l'erreur de discrétisation e_D sont donnés dans la Table (3.5).

Le schéma compact donne les meilleurs résultats, avec un ordre de convergence supérieur à quatre pour la composante x de la vitesse. L'ordre de convergence pour le schéma polynomial est proche de 4. Le nombre de Reynolds n'étant pas suffisamment élevé, le Laplacien discrétisé avec un schéma standard associé à des flux convectifs discrétisés par le schéma polynomial donne une convergence d'ordre 2 uniquement. Cependant l'erreur de troncature est inférieure à celle obtenue avec un schéma d'ordre 2 pour les flux convectifs.

3.2.4.2 Efficacité des différents schémas

Tout les schémas ont été implémentés dans le même code de calcul, et l'efficacité des différents schémas peut ainsi être analysée et comparée. Comme le précise la section 3.1.0.2, l'implémentation du schéma polynomial a été optimisée en utilisant un ensemble de coefficients pré-calculés. Pour le schéma compact, les valeurs du gradient moyen sont obtenues itérativement par une méthode de Jacobi sans calcul explicite de la matrice.

Les temps d'exécution des différents schémas sont donnés dans les Tables (3.6a) et (3.6b). La Table (3.6a) donne le résultat de référence obtenu avec la discrétisation standard d'ordre 2 : le nombre total d'itérations non linéaires pour le solveur Navier Stokes (*iter*), le temps total CPU pour la simulation (*Total*), le temps d'assemblage moyen pour les équations de quantité de mouvement (*A*), le temps moyen par itération pour le calcul du gradient (*G*), et le temps CPU utilisé pour résoudre le système linéaire (*S*). La Table (3.6b) donne les mêmes informations, relativement aux résultats de référence de la Table (3.6a). Le temps CPU utilisé pour résoudre un système linéaire est presque indépendant du schéma utilisé, à cause de l'utilisation de la méthode "deferred correction". Le pré-calcul des coefficients pour la reconstruction k-exact évite la résolution de problèmes aux moindres carrés à chaque itération. Ainsi le temps d'assemblage (*A*) et le temps pour calculer le gradient sont réduits, et le schéma polynomial est seulement deux fois plus lent que le schéma d'ordre 2.

Cependant, pour le schéma Padé, le calcul itératif du gradient par la méthode de Jacobi sans calcul explicite de la matrice nécessite 100 fois plus de temps que le schéma classique et le nombre d'itérations non linéaires pour obtenir une solution convergée est quatre fois supérieur au schéma standard. Ainsi le schéma Padé est environ 20 fois plus lent que la discrétisation standard.

En utilisant les taux de convergence obtenus précédemment, l'erreur de la solution est tracée en fonction de la taille des cellules du maillage (cf. Figure 3.6a) et en fonction du temps de simulation (cf. Figure 3.6b). Le schéma polynomial est clairement plus efficace que la discrétisation standard, mais l'efficacité du schéma compact est supérieure à la discrétisation standard uniquement pour les maillages fins ($\geq 42 \times 42$). On rappelle que la relative inefficacité du schéma Padé provient de l'utilisation d'une méthode de Jacobi sans matrice, qui est facile à implémenter en parallèle. Sur des maillages cartésiens, avec un solveur séquentiel, des solveurs plus efficaces semblables à ceux utilisés en différences finies peuvent être utilisés pour améliorer l'efficacité du schéma.

Une manière de procéder consiste à résoudre implicitement le système d'équations formé par les relations de Padé pour chaque cellule. A condition d'utiliser un maillage cartésien, les trois composantes du gradient selon les axes x,y,z sont indépendantes et donc solution de trois systèmes d'équations indépendants. Le solveur linéaire multigrille utilisé pour la résolution des trois composantes de la vitesse lors de l'étape de prédiction et la résolution de la pression lors de la correction de pression,

	20x20	40x40	80x80
iter	1045.0	2481.0	6107.0
Total (sec)	3.1	25.2	364.6
A (sec/it)	$2.71 \cdot 10^{-4}$	$9.40 \cdot 10^{-4}$	$6.17 \cdot 10^{-3}$
G (sec/it)	$3.77 \cdot 10^{-5}$	$1.12 \cdot 10^{-4}$	$1.77 \cdot 10^{-3}$
S (sec/it)	$2.07 \cdot 10^{-3}$	$7.11 \cdot 10^{-3}$	$3.35 \cdot 10^{-2}$

(a) Temps de référence pour le schéma décentré d'ordre 2.

	20x20		40x40		80x80	
	k=3	pade	k=3	pade	k=3	pade
iter	1.23	4.25	1.04	4.03	1.01	4.35
Total	2.29	20.37	2.05	21.56	1.55	25.99
A (/it)	4.58	2.53	4.54	3.09	3.00	1.54
G (/it)	4.50	93.26	7.10	115.04	2.74	55.20
S (/it)	1.19	0.96	1.18	1.16	1.08	0.88

(b) Temps relatif par rapport à celui obtenu avec le schéma décentré d'ordre 2.

TABLE 3.6 – Temps d'exécution des différents schémas pour la cavité entraînée.

permet de résoudre également les trois composantes du gradient sans coût mémoire supplémentaire. Dans ce cas, une itération du solveur Navier Stokes nécessite de calculer quatre gradients (vitesses u, v, w et pression), soit 12 systèmes linéaires supplémentaires à résoudre, en plus des quatre existants (étapes de prédiction et correction). La relation de Padé est écrite de manière incrémentale et par conséquent une résolution précise des systèmes linéaires n'est pas nécessaire. Dans le cas d'un maillage uniforme et cartésien, on obtient le système suivant à résoudre :

$$2\nabla' \phi_i + \frac{1}{6} \nabla' \phi_{i+1} - \frac{1}{6} \nabla' \phi_{i-1} = \frac{\phi_{i+1} - \phi_{i-1}}{2h} - \left(2\nabla \phi_i + \frac{1}{6} \nabla \phi_{i+1} - \frac{1}{6} \nabla \phi_{i-1} \right)$$

La diagonale de la matrice est dominante, et les méthodes de Gauss Seidel et Jacobi convergent. Pour le cas du maillage 160×160 , cette approche permet de diviser le temps de simulation par 6, et ainsi de diminuer le temps relatif de simulation du schéma Padé par rapport au schéma d'ordre 2 de 26 à 4.

3.2.5 Écoulement autour d'un obstacle cubique

Le dernier cas étudié est celui d'un écoulement autour d'un obstacle cubique de dimension h^3 , placé à la distance $3h$ d'un canal de hauteur $L_y = 2h$ et de longueur totale $L_x = 10h$. À l'entrée, un profil laminaire de Reynolds moyen $Re = 150$ est imposé. Dans la direction z , de longueur $L_z = 8h$, une condition de périodicité est utilisée, et une condition de paroi non glissante pour les parois inférieures et supérieures du canal. Dijk and de Lange (2006) imposent en sortie de canal une condition mixte pour la pression $(p + L_x \frac{\delta p}{\delta x})_b = 0$. Les auteurs emploient la méthode des différences finies avec des schémas compacts d'ordre 5 et une intégration temporelle d'ordre 4 pour résoudre les équations de Navier Stokes pour écoulement compressible. Dans notre cas, l'écoulement et les quantités observées dépendent

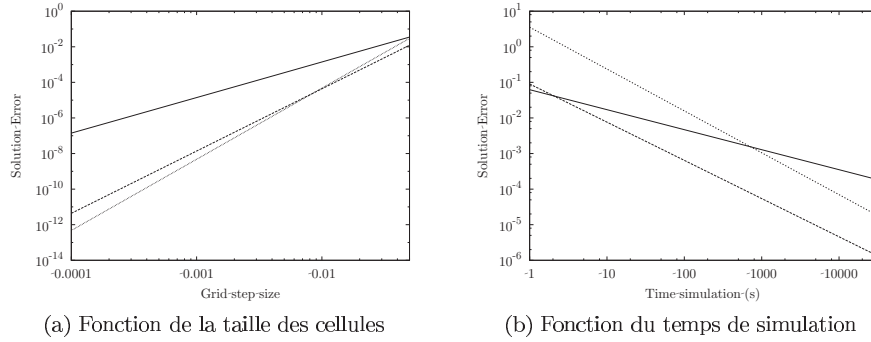


FIGURE 3.9 – Erreur de la solution pour la cavité entraînée. — décentré d’ordre 2 ; - - - polynomial ; ... Compact.

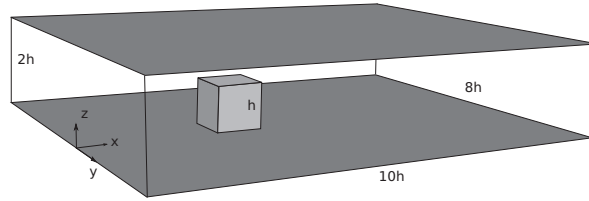


FIGURE 3.10 – Géométrie de l’obstacle cubique.

Maillage	Nombre de cellules
$M1$	137 808
$M2$	1 102 464
$M1_{NU}$	112 757
$M1_{NU}^{R2}$	7 216 448

TABLE 3.7 – Maillages utilisés pour l’obstacle cubique.

fortement de la condition limite avale en particulier pour les schémas d’ordre élevé. Un domaine de longueur $20h$, avec une condition pour la pression $p = 0$ est utilisé à la place. Les simulations sont réalisées avec les schémas Padé, polynomial et décentré d’ordre 2, sur les maillages structurés uniformes $M1$, $M2$ ($M1$ raffiné une fois), et structuré non uniforme $M1_{NU}$ avec des tailles de cellules plus faibles en proche paroi et autour des points de séparation et de recollement de l’écoulement. Le résultat de référence est obtenu avec le schéma Padé sur le maillage $M1_{NU}$ raffiné deux fois ($M1_{NU}^{2R}$). Le nombre de cellules des différents maillages est donné dans la Table (3.7).

3.2.5.1 Structure de l’écoulement

Pour évaluer la structure de l’écoulement, les lignes de courant et les iso-surfaces du critère λ_2 sont présentées dans les Figures (3.12), (3.11), et (3.13). Le critère λ_2 correspond à la seconde valeur propre de la matrice $\Omega^2 + S^2$, avec Ω et S la partie symétrique et anti-symétrique du gradient de vitesse (cf. Jeong.Jinhee and Hussain.Fazle (1995)). Les iso-surfaces λ_2 mettent en évidence différentes structures

qui sont consistantes avec celles obtenues par Dijk and de Lange (2006). Un premier vortex en “fer à cheval” et un second plus petit sont situés à la base amont de l’obstacle. Une autre structure enveloppe l’obstacle, et une structure symétrique émerge des parois latérales de l’obstacle et du vortex en “fer à cheval”, pour s’étendre dans le sillage de l’obstacle.

La Figure (3.12) montre les lignes de courant calculées à partir des composantes u et v de la vitesse, sur le plan xy et la Figure (3.11), les lignes de courant sur un plan xz , calculées à partir des composantes u et w de la vitesse. Les tracés obtenus correspondent globalement à ceux obtenus par Dijk and de Lange (2006).

3.2.5.2 Points de séparation et de réattachement

La structure de l’écoulement présentée précédemment à travers les iso-surfaces λ_2 et les lignes de courants sont calculées à partir de valeurs interpolées et ne permettent pas de montrer la supériorité des schémas d’ordre élevé sur le schéma décentré d’ordre 2. Le profil de vitesse en $y = 0.04$, $z = 0$ présenté par Dijk and de Lange (2006) nécessite également l’interpolation de valeurs. Ainsi, on préfère comparer la position des points de séparation et de réattachement de l’écoulement devant l’obstacle et dans son sillage. Ces points sont définis comme les points les plus en amont et en aval de l’obstacle, situés sur la ligne $y = 0$ et $z = 0$ où $\partial u / \partial y = 0$. Le profil du frottement $\partial u / \partial y$ nécessaire à la détermination de ces points est directement donné par le schéma choisi pour le calcul des flux visqueux et une interpolation polynomiale d’ordre 3 est utilisée pour déterminer avec précision leurs positions.

La Table (3.8) donne la distance entre les points de séparation et de réattachement et les faces amont et aval de l’obstacle (respectivement x_S et x_R). Les valeurs sont comparées à celles obtenues avec le schéma Padé sur le maillage $M1_{NU}$ raffiné deux fois (7.2×10^6 cellules) : $x_S = 1.045$ et $x_R = 2.32$. Le code commercial Fluent, avec le même maillage donne $x_S = 1.043$ et $x_R = 2.326$. Les schémas Padé et Eno permettent donc d’obtenir avec un maillage relativement grossier $M1_{NU}$, des résultats aussi précis que ceux obtenus avec un schéma d’ordre 2 décentré sur un maillage quatre fois plus fin $M1_{NU}^{2R}$. Dans ce cas, la solution avec le schéma Padé sur le maillage $M1_{NU}$ est obtenue sur un processeur en une heure ou 6 minutes avec 16 processeurs et nécessite 38 heures de calcul avec 16 processeurs sur le maillage $M1_{NU}^{2R}$ avec un schéma décentré d’ordre 2.

3.3 Conclusion

Une étude comparative de deux schémas d’ordre élevé a été présentée : une reconstruction polynomiale et un schéma Padé. Les schémas ont été implémentés dans un code de calcul parallèle permettant de résoudre les équations de Naviers Stokes. Différentes expériences numériques ont été présentées.

Des tests de flux sur une solution a priori ont montré que le schéma compact proposé est utilisable uniquement sur les maillages cartésiens, donnant des résultats très médiocres sur un maillage aléatoire de quadrangles et sur les maillages non structurés. Sur les maillages irréguliers, la reconstruction polynomiale est nécessaire pour atteindre des ordres élevés en espace, à condition que le calcul du flux convectif prenne en compte les variations de la vitesse le long de la face (cf. Relation 3.14).

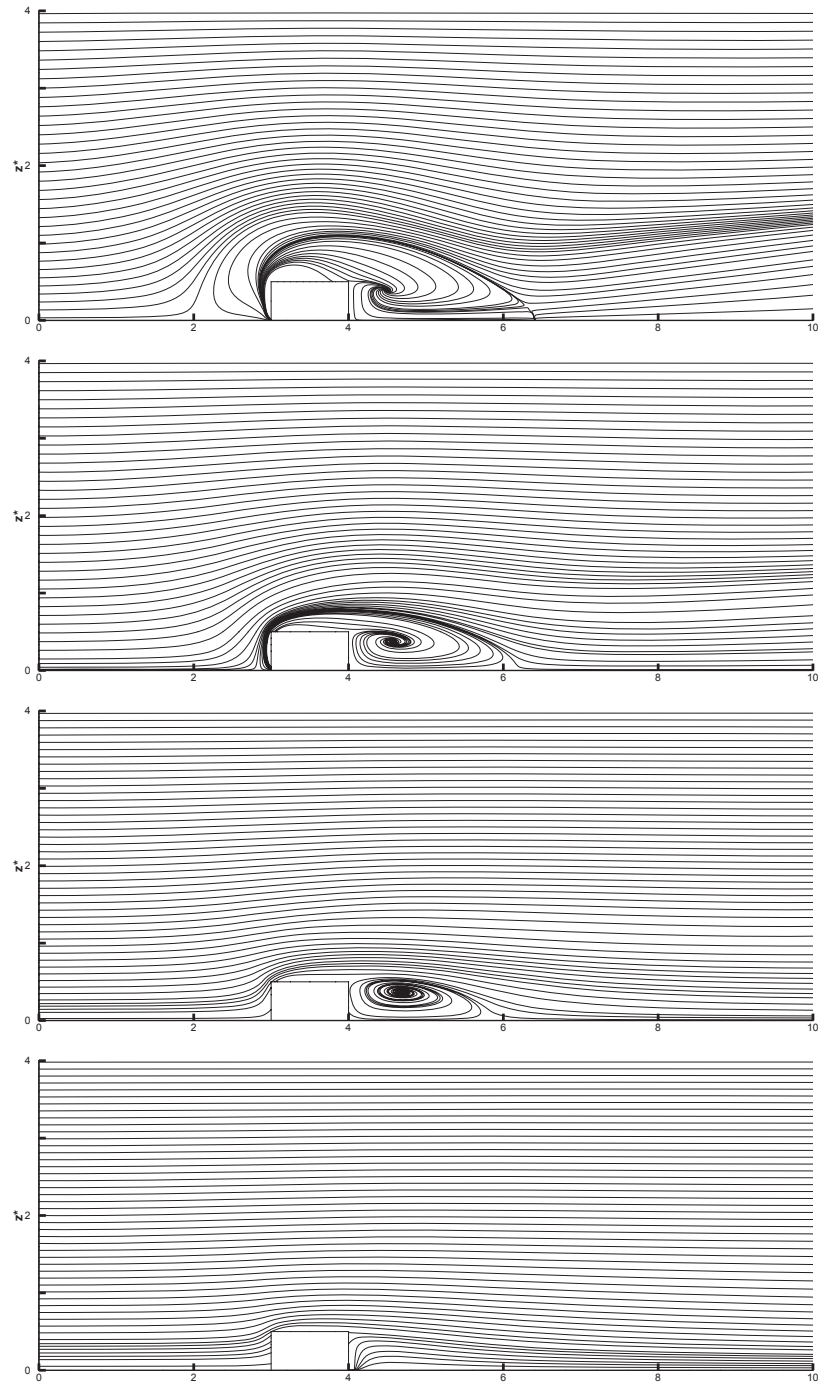


FIGURE 3.11 – Lignes de courants situées à : $y^* = 0.04$, $1/4$, $1/2$ et $11/12$ (de haut en bas).

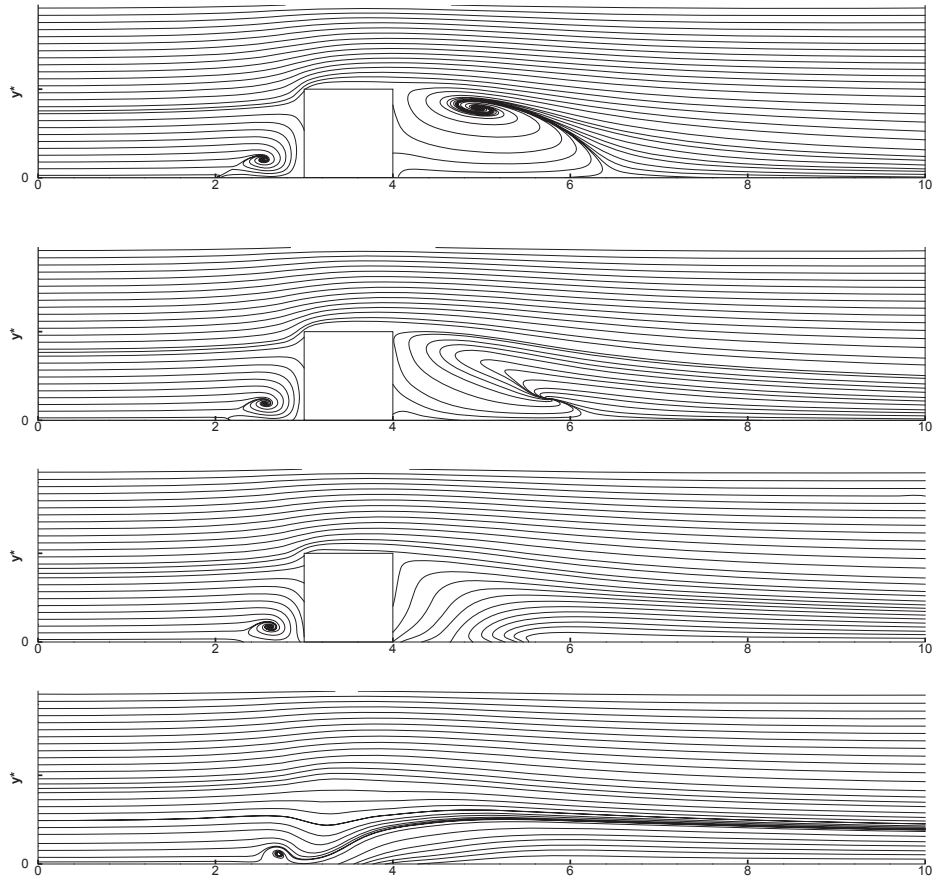


FIGURE 3.12 – Lignes de courants situées à : $z^* = 0.04, 1/4, 5/2$ et $7/12$ (de haut en bas).

Maillage	Δx	erreur $\left(\frac{\Delta x}{2}\right)^3$	décentré ordre 2	Padé	Eno
$M1$	0.167	0.00058	2.677	2.345	2.350
$M2$	0.083	0.00007	2.461	2.307	2.307
$M1_{NU}$	0.178	0.00070	2.408	2.302	2.286
ref ($M1_{NU}^{2R}$)	0.048	1.4×10^{-5}	2.322	2.320	x

(a) x_R

Maillage	Δx	erreur $\left(\frac{\Delta x}{2}\right)^3$	décentré ordre 2	Padé	Eno
$M1$	0.167	0.00058	1.026	0.998	0.997
$M2$	0.083	0.00007	1.053	1.038	1.037
$M1_{NU}$	0.177	0.00114	1.055	1.045	1.043
ref ($M1_{NU}^{2R}$)	0.041	8.6×10^{-6}	1.043	1.045	x

(b) x_S

TABLE 3.8 – Distance des points de séparation x_S et recollement x_R à l'obstacle.

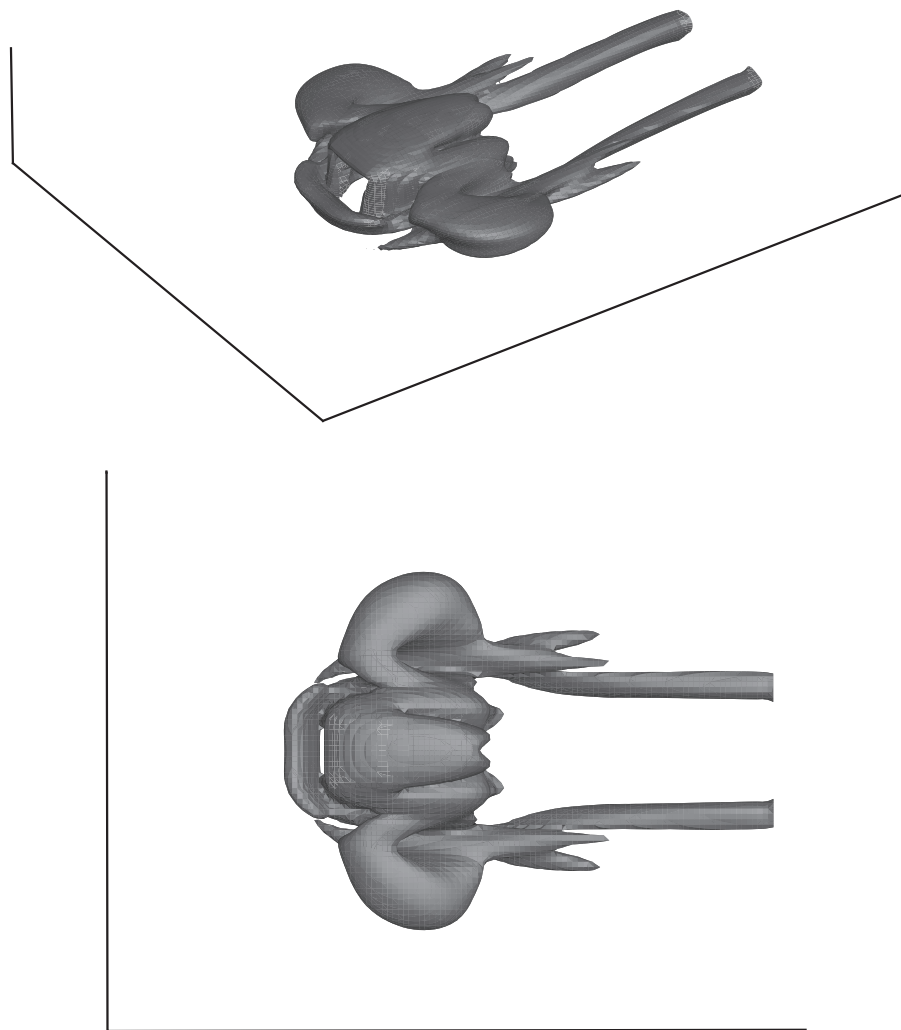


FIGURE 3.13 – Iso-surface λ_2 obtenue avec le maillage M2 et le schéma d'ordre 2.

La simulation numérique de la cavité entraînée à $Re = 1000$ montre un manque de robustesse du solveur provenant des flux visqueux calculés à partir de la reconstruction polynomiale, pour laquelle la propriété de positivité est faible sur les maillages non structurés. L'utilisation de flux visqueux du second ordre améliore la robustesse du solveur au détriment d'ordre de convergence, même si la solution est plus précise que celle obtenue avec des flux convectifs standards. Pour des nombres de Reynolds plus élevés, $Re = 10000$, l'utilisation de flux visqueux standard n'impacte pas l'ordre de convergence, comme le montre l'étude sur le tourbillon de Taylor.

Afin d'implémenter de manière efficace les schémas polynomiaux d'ordre élevé, l'utilisation d'un stencil fixe et d'une fonction de pondération géométrique pour le problème des moindres carrés, a permis de pré-calculer les coefficients de la combinaison linéaire exprimant le flux d'une face en fonction des valeurs moyennes des cellules situés dans le voisinage de cette face. L'implémentation parallèle du schéma Padé en utilisant une méthode itérative de Jacobi sans calcul explicite de la matrice, est moins efficace, même si la précision du schéma Padé est la meilleure de tous les schémas. Cependant, le calcul du gradient par résolution du système linéaire par une méthode multigrille présentée dans la suite, permet de diviser par 6 le temps de simulation globale, ce qui rend le schéma Padé très compétitif : pour le cas de l'obstacle cubique, une simulation sur un maillage uniforme (7.2×10^6 cellules) avec un schéma décentré d'ordre 2, nécessite 38 heures de calcul sur 16 processeurs, alors que la même précision est atteinte avec un schéma Padé, sur un maillage non uniforme de 137 000 cellules, avec un temps de simulation de l'ordre d'une heure sur un processeur ou 6 minutes sur 16 processeurs. Le schéma polynomial est également intéressant : pour une simulation parallèle sur un maillage non structuré, le schéma polynomial est 150 fois plus efficace que le schéma classique d'ordre 2 pour atteindre le même niveau de précision que celui obtenu sur un maillage 80×80 avec un schéma polynomial.

Chapitre 4

Parallélisation

Les temps de simulation pour l'outil CFD sont de loin supérieurs à ceux des outils gaussiens et intégraux. Pourtant, pour certains cas complexes, l'outil CFD est indispensable et pour un maillage et une discrétisation donnés, la parallélisation est alors le seul moyen pour diminuer les temps de calculs. Elle offre un gain largement supérieur à celui que l'on peut attendre de l'augmentation de la vitesse des processeurs (doublement tous les 18 mois selon la loi de Moore). Si les outils CFD, grâce à la parallélisation, commencent à tirer partie des architectures modernes de calcul, leurs efficacités restent limitées. Pour obtenir des méthodes d'ordre élevé efficaces par rapport à des méthodes classiques d'ordre 2, il est nécessaire d'aborder la parallélisation de ces méthodes.

Différentes approches permettent de paralléliser la résolution de systèmes d'équations aux dérivées partielles décrivant le problème à modéliser. Une première approche utilise une résolution temporelle en parallèle (Maday and Turinici, 2007; Guibert, 2009). Cette technique non naturelle décompose le temps en sous intervalles et la solution sur chaque sous intervalle est calculée à condition de pouvoir définir une condition initiale des sous intervalles temps. Cette technique est peu adaptée à la résolution d'équations aux dérivées partielles de type convection-diffusion et s'utilise généralement pour la modélisation de systèmes d'équations différentielles en temps (Guibert, 2009). Une seconde approche plus naturelle pour la résolution d'équations aux dérivées partielles en temps et en espace est la décomposition en sous domaines qui consiste à décomposer le domaine initial en plusieurs sous domaines qui seront distribués aux différents processeurs.

La distribution des données peut être effectuée de différentes manières selon l'architecture de la machine. On distingue les architectures à mémoire partagée des architectures à mémoire distribuée. Les architectures à mémoire partagée possèdent un large bloc de mémoire vive accessible par les différents processeurs. Dans les architectures distribuées, chaque processeur a sa propre mémoire. L'accès à la mémoire d'un autre processeur n'est pas possible et l'échange d'informations entre les processeurs se fait par l'envoi et la réception de messages. Les architectures modernes de calcul sont en réalité des architectures hybrides. Elles sont constituées de noeuds de calcul reliés par un réseau haute performance. Chaque noeud de calcul est un système multi-coeurs qui possède sa propre mémoire (mémoire partagée). L'ensemble des noeuds de calcul forment une architecture à mémoire distribuée. Dans le cadre de cette thèse, la parallélisation est réalisée en mémoire distribuée avec envoi et passage de messages avec l'interface d'échange de messages MPI (Message Passing Interface). La parallélisation

Algorithme 5 Stratégie parallèle.

Lecture maillage par proc0 et partitionnement et distribution des données aux autres processeurs

Création de la structure de données pour les systèmes linéaires

Jusqu'à convergence faire :

Assemblage : calcul de flux et termes sources

Termine l'assemblage

Résolution du système matriciel

Mise à jour des valeurs pour les cellules halos

hybride (utilisation d'openMP entre autres) n'est pas considérée dans le cadre de cette thèse mais constitue une seconde étape pour la parallélisation avec l'avènement des architectures multi-coeurs.

Un algorithme séquentiel itératif, constitué d'au moins deux étapes est utilisé pour résoudre les différents modèles physiques introduits au chapitre 1. Ces deux étapes sont la résolution d'équations de transport non linéaires pour chaque composante de la quantité de mouvement, et la résolution d'une équation de correction pour la détermination de la pression. Les équations étant non linéaires, un processus itératif est nécessaire pour converger vers la solution. A chaque itération, des flux convectifs et diffusifs sont calculés pour chaque face et des termes sources pour chaque cellule du maillage. L'utilisation d'une intégration temporelle implicite conduit à exprimer les flux en fonction de la solution à l'itération k et de la solution à l'itération $k+1$, et nécessite donc la résolution de systèmes linéaires creux de grandes tailles. De la même manière, la correction de pression est solution d'une équation de Laplace.

Ainsi, la décomposition de domaine nécessite dans un premier temps de distribuer les données aux différents processeurs. A condition de mettre en place une structure de données adaptée, l'assemblage se fait naturellement en parallèle. Le partitionnement et la structure de données utilisés sont décrits dans la section 4.1. Le second point important de la parallélisation est la partie résolution des systèmes linéaires. Selon le type de matrice, différentes stratégies peuvent être mises en oeuvre et de nombreuses méthodes existent pour résoudre les systèmes linéaires creux provenant de la discrétisation d'équations aux dérivées partielles. Dans cette thèse, nous avons utilisé les bibliothèques d'algèbre linéaire PETSC et HYPRE qui implémentent de nombreuses méthodes, avec pour chacune d'entre elles, un nombre de paramètres à régler dépendant du problème traité. Ainsi, après une première partie détaillant les bases de ces méthodes, l'influence de ces paramètres sur les performances des différents solveurs est évaluée à travers différents tests de speed-up et scale-up. Cela a permis de déterminer la méthode la plus efficace pour la résolution en parallèle des systèmes linéaires issus de notre problème.

4.1 Structure de données et assemblage des systèmes linéaires

La première étape de la parallélisation concerne le calcul des différents flux et termes sources constituant l'équation discrétisée, ainsi que l'assemblage du système linéaire correspondant. Cependant, une étape préliminaire au calcul est nécessaire pour distribuer les données aux différents processeurs.

4.1.1 Partitionnement du maillage et structure de données

Cette étape préliminaire est nécessaire pour distribuer les données et partitionner le maillage. Dans sa version séquentielle, le maillage est constitué d'un ensemble de faces et de cellules. Ces objets doivent être attribués à un processeur. Ainsi chaque processeur calcule les flux uniquement pour ses faces, et les termes sources uniquement pour ses cellules. L'attribution d'un processeur à une face Γ_{ij} est déduit du partitionnement des cellules selon la règle $proc(\Gamma_{ij}) = proc(i)$ et le partitionnement des cellules est réalisé grâce à la librairie METIS. Les différents algorithmes proposés par METIS construisent le partitionnement à partir du graphe du maillage. Un exemple de partitionnement est donné dans la Figure (4.1).

Le calcul d'un flux sur une face située sur ou proche d'une interface inter-processeurs nécessite de connaître les valeurs de ϕ pour les cellules voisines de la face, certaines de ces cellules n'appartenant pas au processeur qui calcule le flux. Ainsi, de manière à faciliter la manipulation de ces données, chaque processeur possède une partie du maillage au delà des interfaces inter-processeurs. Ce sont les cellules "halos" (ou "ghosts cells", cf. Figure 4.1). L'ensemble des cellules d'un processeur (cellules locales et cellules halos) sont numérotées localement et globalement. La numérotation locale de chaque processeur est indépendante de celle des autres processeurs et commence par les cellules locales puis les cellules halos. Elle est réalisée par l'algorithme inversé de Cuthill-McKee (Cuthill and McKee, 1969) qui améliore l'efficacité des factorisations incomplètes et augmente la rapidité d'exécution par une meilleure utilisation du cache mémoire (Toledo, 1997). Les variables sont stockées sous la forme d'un vecteur et la numérotation locale permet d'accéder aux valeurs associées à chaque cellule. La numérotation globale utilisée pour l'assemblage du système linéaire vérifie :

$$indice\ global(cell) = indice\ local(cell) + \sum_{i=0}^{proc(cell)} n\ cellules\ locales(i)$$

Une troisième numérotation, constante et indépendante du partitionnement et du nombre de processeurs permet de sauvegarder les résultats et de réaliser des reprises de simulation avec un nombre de processeurs différent du nombre de processeurs utilisé lors de la sauvegarde du fichier de reprise. Elle correspond à la numérotation adoptée par le fichier de maillage.

4.1.2 Découpage parallèle du système linéaire

Une fois le partitionnement du maillage réalisé, les librairies PETSC et HYPRE distribuent aux différents processeurs le système linéaire $Ax = b$, formé de la matrice A , du vecteur solution x et du terme de droite b . La matrice d'un système linéaire à résoudre est découpée en bandes horizontales. Pour un système à résoudre à une variable, la hauteur d'une bande correspond au nombre de cellules locales du processeur (cf. Figure 4.2). L'implication des flux utilisant l'approche "deferred correction", le nombre de valeurs non nulles pour la ligne i de la matrice est égale au nombre de voisins directs de la cellule i augmenté de 1. Le nombre de coefficients nuls de la matrice est donc très important, et seulement les coefficients non nuls sont stockés sous le format CSR (Compressed Sparse Row). Le format CSR stocke les valeurs non nulles de chaque ligne de la matrice dans un tableau *val*. Un index de la position du premier élément de chaque ligne est stocké dans un second tableau *row_ptr*. Les

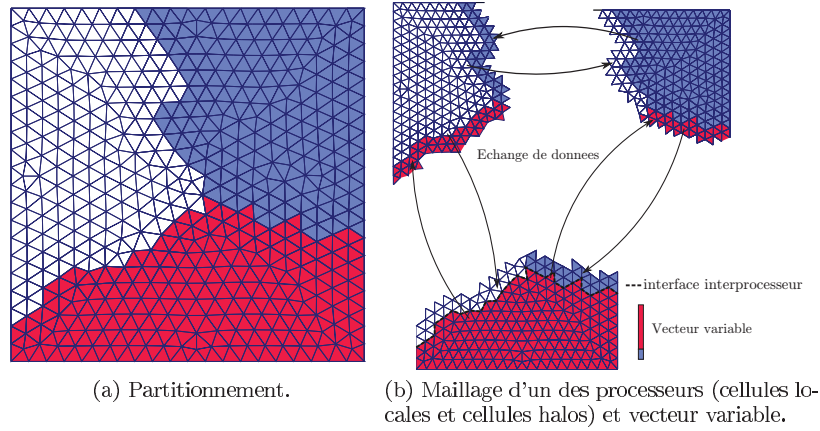


FIGURE 4.1 – Exemple de partitionnement de maillage sur 3 processeurs.

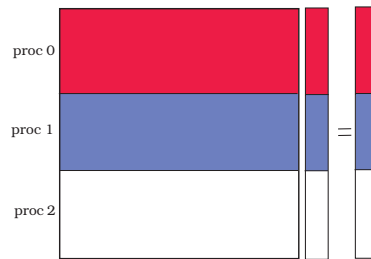


FIGURE 4.2 – Décomposition du système linéaire selon les processeurs.

colonnes de chaque entrée du tableau de valeurs val sont stockées dans un troisième tableau col_ind . Si $val[k]$ stocke la valeur a_{ij} de la matrice, alors $row_ptr[i] < k < row_ptr[i+1]$ et $j = col_ind[k]$.

Le terme de droite b et le vecteur solution x du système $Ax = b$ sont également découpés de la même manière : la taille locale de ces vecteurs correspond au nombre de cellules locales.

4.1.3 Assemblage

Le calcul des flux et des termes sources ne nécessitent aucune communication inter-processeurs, puisque chaque processeur possède toutes les données dont il a besoin (cellules halos et valeurs des variables pour ces cellules). Les valeurs des variables pour les cellules halos sont mises à jour à chaque itération, après résolution des systèmes linéaires. Elles proviennent des processeurs qui possèdent localement ces éléments. Afin de réaliser un échange efficace, un processeur possède les indices locaux des valeurs à envoyer et la liste des indices des valeurs qu'il reçoit des autres processeurs.

Chaque processeur calcule les termes sources uniquement pour ses cellules locales. Le terme source pour une cellule locale d'indice global i_{global} peut s'écrire de la forme $S_i = a\delta\phi_i^{n+1} + a_i$. Le terme a_i est ajouté au vecteur b à la position i_{global} , et le terme a dans la matrice sur la diagonale en (i_{global}, i_{global}) . Les termes sources étant calculés uniquement pour les cellules locales, l'assemblage se fait directement localement et ne nécessite pas de communication inter-processeurs.

Le flux $F_{ij} = \delta F_{ij}(\delta\phi_i, \delta\phi_j) + F_{ij}^{HO}$ se décompose en une partie implicite $\delta F_{ij}(\delta\phi_i, \delta\phi_j)$ et une partie explicite F_{ij}^{HO} . La partie explicite est assemblée dans le membre de droite b à la position i_{global} , et son opposé à la position j_{global} . $\delta F_{ij} = -\delta F_{ji} = a_{ii}\delta\phi_i^{n+1} + a_{ij}\delta\phi_j^{n+1}$ correspond à la variation implicite du flux. Les coefficients a_{ii} , a_{ij} sont assemblés dans la matrice selon :

$$\begin{aligned} A_{ii+} &= a_{ii} \\ A_{ij+} &= a_{ij} \\ A_{ji-} &= a_{ii} \\ A_{jj-} &= a_{ij} \end{aligned}$$

La cellule i appartient au processeur qui calcule le flux et l'assemblage des termes A_{ii} , A_{ij} ne nécessite pas de communication. Par contre, si la face Γ_{ij} appartient à une interface inter-processeurs, les coefficients A_{ji} et A_{jj} sont situés sur la partie de la matrice stockée par le processeur possédant la cellule j . Ces valeurs sont bufferisées par `proc(i)` et sont envoyés à `proc(j)` lorsque l'assemblage est terminé. La bufferisation utilisée par les bibliothèques PETSC et HYPRE permet un gain de temps par rapport à une communication immédiatement après le calcul du flux.

4.2 Résolution des systèmes linéaires

L'utilisation de schémas implicites nécessite la résolution de systèmes linéaires creux de grandes tailles. La résolution d'un système de taille $n = 40000$ était un challenge il y a 40 ans. Aujourd'hui, cette valeur atteint $n = 10^9$ et des systèmes de taille $n = 10^7$ sont utilisés pour les applications courantes. Plusieurs méthodes permettent de résoudre les systèmes linéaires provenant de la discrétisation des équations de type convection-diffusion :

$$Ax = b$$

où A est une matrice creuse non symétrique.

- les méthodes directes basées sur l'élimination de Gauss pour des matrices creuses ("sparse Gaussian elimination") sont robustes mais coûteuses pour les systèmes de grandes tailles. Elles demandent énormément de ressources mémoire pour réaliser les factorisations. Elles sont généralement utilisées pour les systèmes de tailles moyennes correspondant à des problèmes bi-dimensionnels.
- les méthodes itératives adaptées aux systèmes de grandes tailles à matrices creuses, issues de la discrétisation des équations aux dérivées partielles. C'est donc ce type de méthode qui est utilisé dans le cadre de la thèse. La solution au problème est recherchée de manière itérative :

$$x^{k+1} = Mx^k + f$$

Dans un premier temps, nous allons décrire les principes des différentes méthodes itératives utilisées, en mettant en évidence les paramètres importants. Les méthodes itératives utilisées sont classées en deux familles distinctes :

- les méthodes basées sur les espaces de Krylov, présentées dans la section 4.2.1. La section 4.2.3 introduit les techniques de préconditionnement permettant d'accroître l'efficacité des méthodes itératives, en particulier celle des méthodes de Krylov.

- les méthodes multigrilles algébriques, présentées dans la section 4.2.2, pour lesquelles la résolution s'effectue sur plusieurs niveaux de grilles construites à partir des informations de la matrice du système linéaire à résoudre.

Dans une seconde partie, différents tests sont mis en place pour déterminer les méthodes les plus efficaces et leurs paramètres optimaux pour la résolution des systèmes linéaires creux provenant des différentes étapes du solveur Navier Stokes.

4.2.1 Solveur de Krylov et méthode GMRES

Les solveurs de Krylov s'inscrivent dans le cadre plus générale des méthodes de projection. Soit deux sous espaces \mathbb{K} et \mathbb{L} de \mathbb{R}^n . On cherche $x \in \mathbb{K}$ tel que $b - Ax \perp \mathbb{L}$, équivalent à chercher $\delta \in \mathbb{K}$ tel que $r_0 - A\delta \perp \mathbb{L}$ avec $x = x_0 + \delta$ et $r_0 = b - Ax_0$. On distingue deux cas importants pour le choix des sous espaces \mathbb{K} et \mathbb{L} :

- $\mathbb{L} = \mathbb{K}$ forme une classe de méthodes de projection orthogonale (méthode de Galerkin).
- les méthodes de Krylov utilisent un espace de Krylov \mathbb{K} défini par :

$$\mathbb{K}_m(A, r_0) = span(\{r_0, Ar_0, \dots, A^{m-1}r_0\})$$

Les différentes méthodes de Krylov se différencient par le choix de l'espace \mathbb{L} et la manière dont le système est préconditionné. $\mathbb{L} = A\mathbb{K}$ forme une classe de méthodes à résidu minimal : CR, GCR, ORTHOMIN, GMRES, CGNR.

Introduit par Saad and Schultz (1986), l'idée de la méthode GMRES est de chercher la solution dans l'espace engendré par $x_0 + z$, $z \in \mathbb{K}_m$ avec \mathbb{K}_m l'espace de Krylov défini par :

$$\mathbb{K}_m(A, r_0) = span(\{r_0, Ar_0, \dots, A^{m-1}r_0\})$$

Le vecteur z est tel que $\|b - A(x_0 + z)\|_2$ est minimum sur l'espace $\mathbb{K}_m(A, r_0)$. Ceci est équivalent à $r_m \perp A\mathbb{K}_m(A, r_0)$ avec $r_m = b - Ax_m$. $r_0 = b - Ax_0$ est le résidu initial et β sa norme. En pratique, construire les espaces de Krylov revient à construire une base de cet espace. La base $r_0, Ar_0, \dots, A^{m-1}r_0$ n'est pas directement utilisable à cause de la dégénérescence numérique. On construit alors une base orthonormale $V_k = \{v_1, \dots, v_k\}$ par le procédé d'orthogonalisation de Gram-Schmidt (Saad, 2000), appelé procédé d'Arnoldi quand il est appliqué dans le cadre des espaces de Krylov. Le procédé d'orthogonalisation de Gram-Schmidt est sensible aux erreurs d'arrondi et peut être remplacé par le procédé d'orthogonalisation de Gram-Schmidt modifié (Björck, 1967b), moins économe en terme d'occupation mémoire mais plus stable. Dans le cas d'erreurs d'arrondi, l'algorithme modifié permet d'obtenir une base plus orthonormale que celle obtenue avec l'algorithme classique. L'orthogonalisation peut être aussi réalisée par une transformation de Householder, plus précise mais plus coûteuse. La librairie PETSC propose également une méthode avec raffinement itératif (Björck, 1967a)

Le calcul des vecteurs de la base d'Arnoldi nécessite le stockage des nouveaux vecteurs calculés. Pour des questions de place mémoire et de coût de calcul, on limite la taille de l'espace de Krylov à une valeur m . Si la convergence n'est pas atteinte, on redémarre le processus. La dimension maximum

Algorithme 6 Méthode GMRES.

1. Démarrage : choisir x_0 et la dimension m de l'espace de Krylov.

2. Processus d'Arnoldi :

– Calcul de $r_0 = b - Ax_0$ et poser $v_1 = \frac{r_0}{\beta}$, $\beta = \|r_0\|_2$

– Pour $j = 1 \dots m$:

Calcul $\omega = Av_j$

Pour $i = 1 \dots j$:

$$h_{i,j} = (\omega, v_i)$$

$$\omega = \omega - h_{i,j}v_i$$

$$h_{j+1,1} = \|\omega\|_2 \quad , \quad v_{j+1} = \frac{\omega}{h_{j+1,1}}$$

– Définir $V_m = [v_1, \dots, v_m]$ et $\overline{H}_m = \{h_{i,j}\}$

3. Résolution du problème des moindres carrés : $x_m = x_0 + V_m y_m$

avec $y_m = \operatorname{argmin}_y \|\beta e_1 - \overline{H}_m y\|_2$ et $e_1 = [1, 0, \dots, 0]^T$

4. Redémarrage en 1. avec $x_0 = x_m$ si la convergence n'est pas satisfaisante

de l'espace de Krylov est généralement faible devant la taille n du système à résoudre afin de conserver un coût mémoire et un coût de calcul raisonnable.

La construction de la base d'Arnoldi permet de construire la matrice d'Hessenberg $\overline{H}_m = \{h_{i,j}\}$ (cf. Algorithme 6) et la résolution du problème des moindres carrés, $r_m \perp AK(A, r_0)$, est équivalente à déterminer la solution y minimisant la norme de $\beta e_1 - \overline{H}_m y$. La résolution du problème des moindres carrés $y_m = \operatorname{argmin}_y \|\beta e_1 - \overline{H}_m y\|_2$ fait intervenir la matrice d'Hessenberg \overline{H}_m dont la structure est proche d'une matrice triangulaire supérieure. Ainsi une rotation de Givens (Golub and Loan, 1996) par colonne permet d'éliminer $h_{i+1,i}$ et de réaliser rapidement la factorisation QR de \overline{H}_m et de résoudre ainsi efficacement le problème des moindres carrés. Cette approche offre l'avantage d'obtenir à chaque étape le résidu du système, sans coût supplémentaire.

Les méthodes de Krylov peuvent avoir une convergence lente, dépendant du conditionnement de la matrice. Leurs performances peuvent être grandement améliorées par l'utilisation de préconditionneurs présentés dans la section 4.2.3. L'algorithme GMRES avec préconditionnement est rappelé dans l'Algorithme (7).

4.2.1.1 Amélioration de GMRES

Lors d'un redémarrage, GMRES perd les informations de l'espace précédent. L'orthogonalité de la solution aux espaces précédents n'est pas conservée à chaque redémarrage et peut engendrer dans certains cas des problèmes de convergence où les résidus stagnent. Plusieurs auteurs proposent des modifications de GMRES pour remédier à ce problème.

Les méthodes avec déflation enrichissent l'espace de Krylov avec des vecteurs propres approchés provenant des précédents redémarrages de GMRES. Les vecteurs propres choisis sont ceux avec des valeurs propres associées faibles. Eiermann et al. (2000) proposent une revue et analysent des différentes méthodes d'accélération les plus utilisées pour remédier aux problèmes de redémarrage de GMRES. La librairie PETSC implémente deux techniques d'accélération de GMRES : flexible GMRES (Saad, 1993) et LGMRES (Baker et al., 2005) :

Algorithme 7 Méthode GMRES avec préconditionnement à droite.

1. Démarrage : choisir x_0 et la dimension m de l'espace de Krylov.

2. Processus d'Arnoldi :

– Calcul de $r_0 = b - Ax_0$ et poser $v_1 = \frac{r_0}{\beta}$, $\beta = \|r_0\|_2$

– Pour $j = 1 \dots m$:

 Calcul $z_j = M_j^{-1}v_j$

 Calcul $\omega = Az_j$

 Pour $i = 1 \dots j$:

$$h_{i,j} = (\omega, v_i)$$

$$\omega = \omega - h_{i,j}v_i$$

$$h_{j+1,1} = \|\omega\|_2, \quad v_{j+1} = \frac{\omega}{h_{j+1,1}}$$

– Définir $V_m = [v_1, \dots, v_m]$ et $\overline{H}_m = \{h_{i,j}\}$

3. Résolution du problème des moindres carrés : $x_m = x_0 + M^{-1}V_my_m$

avec $y_m = \operatorname{argmin}_y \|\beta e_1 - \overline{H}_m y\|_2$ et $e_1 = [1, 0, \dots, 0]^T$

4. Redémarrage en 1. avec $x_0 = x_m$ si la convergence n'est pas satisfaisante

Algorithme 8 Méthode GMRES avec préconditionnement variable.

1. Démarrage : choisir x_0 et la dimension m de l'espace de Krylov.

2. Processus d'Arnoldi :

– Calcul de $r_0 = b - Ax_0$ et poser $v_1 = \frac{r_0}{\beta}$, $\beta = \|r_0\|_2$

– Pour $j = 1 \dots m$:

 Calcul $z_j = M^{-1}v_j$

 Calcul $\omega = Az_j$

 Pour $i = 1 \dots j$:

$$h_{i,j} = (\omega, v_i)$$

$$\omega = \omega - h_{i,j}v_i$$

$$h_{j+1,1} = \|\omega\|_2, \quad v_{j+1} = \frac{\omega}{h_{j+1,1}}$$

– Définir $Z_m = [z_1, \dots, z_m]$ et $\overline{H}_m = \{h_{i,j}\}$

3. Résolution du problème des moindres carrés : $x_m = x_0 + Z_my_m$

avec $y_m = \operatorname{argmin}_y \|\beta e_1 - \overline{H}_m y\|_2$ et $e_1 = [1, 0, \dots, 0]^T$

4. Redémarrage en 1. avec $x_0 = x_m$ si la convergence n'est pas satisfaisante

Le coût supplémentaire du préconditionnement variable est la sauvegarde des vecteurs $\{z_j\}$, $j = 1..m$

(a) Flexible GMRES : propose la possibilité de changer de préconditionneur à chaque itération d'un cycle GMRES (cf. Algorithme 8). La flexibilité permet d'employer n'importe quelle méthode itérative comme préconditionneur, en particulier les méthodes multigrilles algébriques.

(b) LGMRES : (Baker et al., 2005) proposent à chaque redémarrage d'augmenter l'espace $\mathbb{K}_m(A, r_{i-1})$ avec les k plus récentes approximations de l'erreur $z_j = x_j - x_{j-1}$, pour éviter d'obtenir un vecteur résiduel de la même direction que ceux obtenus aux cycles précédents.

4.2.1.2 Parallélisation

Les étapes de construction de la base et son orthogonalisation sont les plus coûteuses, loin devant la résolution du problème de minimisation qui nécessite la résolution d'un système des moindres carrés $m \times (m + 1)$ avec m faible ($m \approx 50$). Les rotations de Givens appliquées à la matrice d'Hessenberg rendent la résolution du problème des moindres carrés extrêmement efficace et la parallélisation peut être réalisée uniquement sur les étapes de construction de la base et son orthogonalisation. Le calcul de la base est une succession de produits matrice-vecteur et le processus d'orthogonalisation de Gram-Schmidt requiert un ensemble de sous-routines élémentaires qui doivent être parallélisées :

- calculs de produits scalaires.
- calculs vectoriels de type AXPY, AXPBY, AYPX ...

Produit scalaire La parallélisation du calcul de produits scalaires est effectuée en deux étapes : chaque processeur effectue un produit scalaire sur son vecteur local. Une somme de chaque produit scalaire donne le produit scalaire global. La fonction `MPI_Allreduce` permet de combiner chaque contribution avec l'opérateur "somme". Le résultat est redistribué à tous les processeurs. Cependant, l'efficacité parallèle du produit scalaire est réduite lorsque le nombre de processeurs est élevé.

Calcul vectoriel La parallélisation du calcul vectoriel (AXPY, AXPBY, AYPX) est plus triviale. Chaque processeur réalise localement le calcul. Aucune communication n'est nécessaire entre les différents processeurs.

Parallélisation du produit matrice-vecteur $Ax = y$ La librairie PETSC décompose la matrice globale A par ligne, donnant ainsi une matrice rectangulaire A_{loc} . On note B_{loc} le bloc diagonale de A_{loc} , c'est à dire une sous matrice de A_{loc} dont les éléments a_{ij} sont tels que j est local. D'une manière similaire, on note B_{ext} une sous matrice de A_{loc} dont les éléments a_{ij} sont tel que j n'est pas local. La structure de la matrice est illustrée sur la Figure (4.3). Chacune des deux matrices B_{ext} et B_{loc} est stockée sous le format CSR. Le vecteur x se décompose naturellement en $x = \langle x_{loc}, x_{ext} \rangle$, le processeur n'ayant accès qu'à x_{loc} . Lors d'une opération matrice vecteur, le produit matrice-vecteur Ax devient :

$$B_{loc}x_{loc} + B_{ext}x_{ext}$$

L'opération $B_{loc}x_{loc}$ peut être réalisée sans communication, cependant le produit $B_{ext}x_{ext}$ nécessite de connaître une partie de x_{ext} . Afin de diminuer les temps d'attente dus aux transferts de ces données à travers le réseau reliant les noeuds de calcul, l'opération d'échange d'informations commence avant

Algorithme 9 Produit matrice vecteur parallèle.

Envoi aux processeurs voisins des noeuds

Multiplication partie diagonale $y_{loc} = B_{loc}x_{loc}$

Réception des données

Multiplication partie off-diagonale $y_{loc+} = B_{ext}x_{ext}$

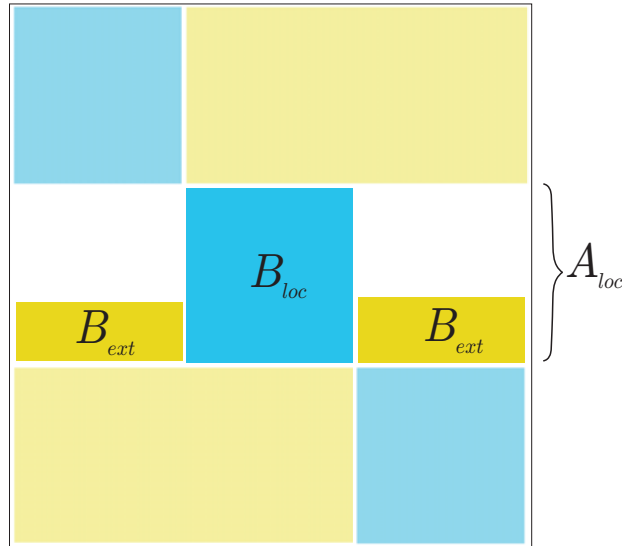


FIGURE 4.3 – Découpage parallèle de la matrice.

la multiplication $B_{loc}x_{loc}$. Cette opération n'est pas bloquante, et le processeur peut réaliser localement la multiplication $B_{loc}x_{loc}$ pendant le transfert des données : il y a recouvrement du transfert de données. Une fois l'opération de transfert terminée, le processeur réalise la multiplication $B_{ext}x_{ext}$ (cf. Algorithme 9). Ce type de parallélisation est également utilisé par Saad (1995) dans la librairie P-SPARSELIB. Le transfert des données nécessite :

- pour chaque processeur de connaître les processeurs adjacents avec lesquels il échange des informations.
- pour chaque processeur adjacent, une liste des noeuds locaux à envoyer et une liste des noeuds à recevoir.

Afin d'améliorer l'efficacité du produit matrice-vecteur, ces informations sont stockées sous formes de listes et de pointeurs.

De nombreuses variantes existent pour la parallélisation de GMRES. La parallélisation peut également être réalisée par la décomposition en domaine via l'algorithme de Schwarz. Cette méthode est présentée dans la partie préconditionnement parallèle (section 4.2.3.1.1).

4.2.2 Solveur multigrille

Les méthodes itératives classiques telles que les méthodes de Jacobi ou Gauss Seidel fournissent un bon opérateur de lissage du résidu du système $Ax = b$ à résoudre. En d'autres termes, ces méthodes réduisent essentiellement les hautes fréquences des résidus du système linéaire. L'idée centrale

des méthodes multigrilles introduites dans les années 1970, est de réduire les erreurs non éliminées par la relaxation à l'aide d'un ensemble de grilles grossières. Les méthodes multigrilles peuvent être séparées en deux catégories : multigrille algébrique ou géométrique. Les différents niveaux de grilles utilisés pour la méthode géométrique sont construits à partir du maillage géométrique alors que les méthodes algébriques les construisent uniquement à partir des informations de la matrice A du système à résoudre. Dans la suite, on ne considère que des méthodes algébriques qui ne nécessitent pas la gestion de maillages imbriqués.

On considère une suite de grilles de plus en plus grossières et on introduit alors un opérateur de restriction $R_H^h : v_h \rightarrow v_H$ permettant de restreindre un vecteur v_h d'une grille raffinée Ω_h à un vecteur v_H d'une grille plus grossière Ω_H , et un opérateur d'interpolation $I_H^h : v_H \rightarrow v_h$ permettant de réaliser l'opération inverse, relié à l'opérateur de restriction par transposition :

$$R_H^h = (I_H^h)^T$$

Les problèmes sur les grilles grossières sont définis en utilisant une approche de Galerkin :

$$A_H = I_H^h A_h R_H^h \quad , \quad f^H = R_H^h f^h$$

La combinaison des opérateurs de restriction, d'interpolation et de relaxation forme un cycle multigrille : le résidu du système linéaire du maillage initial est restreint au maillage le plus grossier pour calculer une correction de la solution, qui est ensuite interpolée sur le maillage grossier, permettant ainsi de réduire les composantes basses fréquences du résidu. Les cycles classiques V et W sont définis récursivement dans l'Algorithme (10). Par exemple, pour quatre niveaux de grilles, la Figure (4.4) détaille l'ordre dans lequel les grilles sont parcourues pour les cycles V et W.

Algorithme 10 Algorithme Multigrille.

Algorithme Multigrille(x_l, b_l, l)

{
 Si ($l = 0$) $x_l = A_l^{-1} b_l$
 Sinon
 {
 ν_1 relaxation de x_l
 Restriction du résidu $r_l = b_l - A_l x_l : r_{l-1} = R_{l-1,l} r_l$
 $x_{l-1} = 0$
 Pour ($j = 0; j < \gamma; j = j+1$) *Algorithme Multigrille* ($x_{l-1}, r_{l-1}, l-1$)
 $x_l = x_l + P_{l,l-1} v_{l-1}$
 ν_2 relaxation de x_l
 }
}

La construction des composants AMG est réalisée dans une étape de préparation, avant l'application du cycle multigrille. Pour décrire complètement la méthode multigrille algébrique, il est nécessaire de définir :

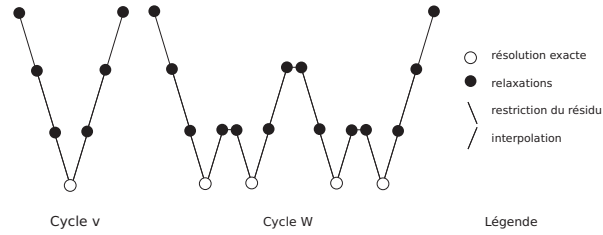


FIGURE 4.4 – Cycle multigrille.

- les algorithmes permettant de construire les différents niveaux des grilles.
- les différents opérateurs du cycle multigrille (opérateur de restriction, opérateur d’interpolation, opérateur de relaxation).

Ces opérateurs sont décrits dans les sections suivantes.

4.2.2.1 Algorithme de construction des niveaux de grilles

Ce paragraphe décrit brièvement les différents algorithmes de construction de grilles existants (appelés “coarsening algorithms” dans la littérature). Les algorithmes de construction des différentes grilles utilisent les informations de la matrice à travers son graphe. L’algorithme de sélection doit permettre de partitionner le graphe de la matrice en points C (“coarse point”) et points F (“refine point”). Un nombre important de points C permet d’avoir une interpolation de meilleure qualité. Cependant un nombre trop élevé de points C augmente les coûts mémoires et le temps d’exécution qui en résulte. Un équilibre doit alors être trouvé entre la sauvegarde, le coût du calcul et la convergence.

Afin de décrire les différents algorithmes, on définit :

- le voisinage d’un point i noté N_i est l’ensemble des points $j \neq i$ qui influence i . Un point i influence un point j si le coefficient de la matrice $a_{ij} \neq 0$. Une dépendance entre deux points i et j est qualifiée de “forte” si le critère suivant est respecté :

$$-a_{ij} \geq \theta \max_{k \neq i} (-a_{ik})$$

avec typiquement $\alpha = 0.25$.

- S_i l’ensemble des points qui influencent fortement i . S_i est un sous ensemble du voisinage N_i :

$$S_i = \{j \neq i : -a_{ij} \geq \theta \max_{k \neq i} (-a_{ik})\}$$

- S_i^T l’ensemble des points qui dépendent du point i .
- pour chaque point $i \in F$, C_i est l’ensemble de points C utilisés pour l’interpolation du point i .

Les algorithmes de construction des niveaux de grilles sont divers et décrits dans l’annexe A1. Ils peuvent être classés en deux catégories :

- les algorithmes classiques basés sur l’heuristique introduit par Ruge Stueben (Falgout, Ruge Stueben, CLJP) conduisent d’une manière générale à des niveaux de grilles denses.
- les algorithmes plus récents (PMIS, HMIS) génèrent des niveaux de grilles moins denses (cf. Figure 4.5).

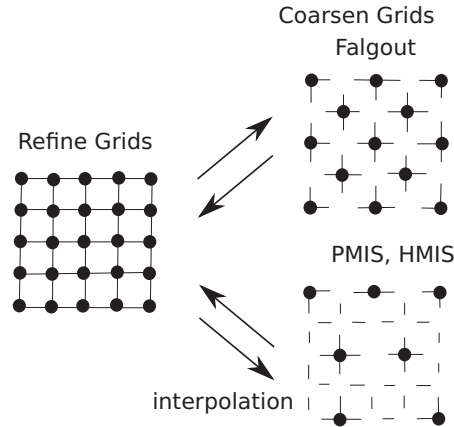


FIGURE 4.5 – Algorithmes de sélection des grilles et schémas d’interpolation.

Les interpolations utilisées avec ces deux familles d’algorithmes doivent tenir compte de la densité des grilles que les algorithmes de sélection des sous maillages génèrent.

4.2.2.2 Opérateur d’interpolation

L’opérateur d’interpolation constitue, avec l’algorithme de construction des grilles, les opérateurs les plus importants de la méthode multigrille. Avant d’introduire les différents schémas d’interpolation, quelques définitions communes à toutes les interpolations AMG doivent être définies. Les points auxquels un point F d’indice i est connecté peuvent être classés en trois sous-ensembles :

- le premier sous ensemble est l’ensemble C_i des points C qui influencent fortement i .
- parmi les points F , on distingue l’ensemble des points F_i^s qui influencent fortement i et l’ensemble F_i^w des points qui sont faiblement connectés à i .

L’erreur d’un point C est connue sur le maillage grossier, et peut être interpolée directement sur le maillage fin. Le schéma d’interpolation doit définir la manière dont est interpolée un point F . D’une manière générale on peut écrire :

$$(I_H^h e)_i = \begin{cases} e_i & \text{si } i \in C \\ \sum_{j \in C_i} \omega_{ij} e_j & \text{si } i \in F \end{cases} \quad (4.1)$$

Le schéma d’interpolation doit définir les poids ω_{ij} pour tous les points F , tels que l’erreur sur le maillage fin vérifie :

$$a_{ii} e_i + \sum_{j \in N_i} a_{ij} e_j \approx 0 \quad (4.2)$$

Les opérateurs d’interpolations sont nombreux et décrits dans l’annexe A2. On distingue les interpolations simples à faibles complexités (interpolation directe, classique), des interpolations plus complexes (interpolation F-F, F-F1, multi-passes) utilisées généralement avec les algorithmes de construction de grilles PMIS et HMIS. En effet, PMIS et HMIS donnent des grilles beaucoup moins denses

que les algorithmes classiques Falgout, Ruge Stueben, ou CLJP qui n'ont donc pas besoin d'utiliser des interpolations complexes (cf. Figure 4.5).

4.2.2.3 Procédure de lissage

L'opérateur de lissage permet d'éviter les erreurs dues aux changements de résolution. Il est appliqué avant l'opérateur de restriction et après avoir appliqué l'opérateur de prolongation. La dénomination "algorithme de lissage" provient de la capacité de l'algorithme à réduire essentiellement les hautes fréquences de l'erreur. Les algorithmes de Gauss-Seidel ou Jacobi sont généralement utilisés pour lisser la solution.

4.2.3 Préconditionnement

La convergence des méthodes itératives dépend du conditionnement du système et du rayon spectral de la matrice. Afin d'accélérer la convergence des solveurs itératifs de type Krylov, le système linéaire $Ax = b$ est remplacé par un système de la forme $M^{-1}Ax = M^{-1}b$ pour un preconditionnement à gauche ou $AM^{-1}y = b$ avec $x = M^{-1}y$ pour un preconditionnement à droite. La matrice M est un bon preconditionneur si elle est proche de l'inverse de A . Les preconditionneurs peuvent être classés en deux catégories :

- preconditionneur implicite ou directe : le preconditionneur est une approximation de la matrice $M \approx A$. Il nécessite la résolution de système linéaire :

$$A = M - N$$

$$x_{k+1} = M^{-1}Nx_k + M^{-1}b$$

- preconditionneur explicite ou inverse : le preconditionneur est une approximation de l'inverse de la matrice $G = M^{-1} \approx A^{-1}$. Un preconditionneur explicite requiert des produits matrice-vecteur et est donc facilement parallélisable. Mais la construction d'un preconditionneur explicite est plus coûteuse qu'un preconditionneur implicite car A^{-1} n'est pas connue. La convergence est généralement meilleure avec un preconditionnement implicite.

La difficulté réside dans le choix d'un preconditionneur efficace. Un certains nombres de preconditionneurs sont présents dans les bibliothèques PETSC et HYPRE. Les procédures de lissage ou schémas de relaxation (Jacobi, Gauss Seidel) peuvent être utilisés comme méthodes de preconditionnement à point fixe. La solution est recherchée de manière itérative par :

$$x^{k+1} = Gx^k + f$$

où G et f ne dépendent pas de l'itération. Dans la relation $x^{k+1} = Gx^k + f$, G est de la forme $I - M^{-1}A$. Le système peut être réécrit :

$$M^{-1}Ax = M^{-1}b$$

Les schémas de relaxation sont ainsi équivalents à une itération de point fixe sur un système preconditionné. A la place de ces preconditionneurs, on préfère utiliser des preconditionneurs plus robustes

tels que les méthodes ILU décrites dans la suite. Les méthodes multigrilles peuvent également être utilisées en tant que préconditionneur du solveur GMRES.

4.2.3.1 Factorisation incomplète LU

La factorisation incomplète LU est largement utilisée comme préconditionnement pour les méthodes itératives. Elle est basée sur la décomposition LU qui permet de décomposer une matrice A en une matrice triangulaire inférieure L et une matrice triangulaire supérieure U . La décomposition s'effectue par une procédure d'élimination de Gauss relativement coûteuse. La factorisation incomplète LU réduit le temps de calcul en appliquant l'algorithme de factorisation LU en limitant le remplissage. Les stratégies d'augmentation du remplissage peuvent être généralement classées en deux catégories :

- ILUT basé sur un seuil
- ILU (l) basé sur la structure.

Ces méthodes sont décrites dans l'annexes A3. Deux stratégies différentes permettent de paralléliser le préconditionnement ILU. La première est le calcul en parallèle de la matrice de préconditionnement. La librairie HYPRE propose deux méthodes similaires utilisant cette stratégie : Euclid (Hysom and Pothén, 1999, 2000) et PILUT (Karypis and Kumar, 1997). Leur description est réalisée dans l'annexe A3. Une autre possibilité décrite dans la section suivante, est l'utilisation de la décomposition de domaine de type Schwarz qui permet de paralléliser l'ensemble "solveur GMRES + préconditionnement ILU" et d'utiliser les algorithmes séquentiels qui ne sont pas parallélisés (ILUT par exemple).

4.2.3.1.1 Préconditionneur parallèle de type Schwarz Un préconditionnement parallèle peut être construit à partir des méthodes de décomposition en sous domaines introduite par Schwarz en 1870. Chaque sous domaine Ω_i est étendu au delà de son interface initiale. L'interface étendue est notée $\Gamma_{i,j}$. La décomposition en domaine est illustrée sur la Figure (4.6). Dans chaque sous domaine, le vecteur des variables peut être décomposé en :

$$z_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

où x_i représente les variables internes et y_i représente les variables aux interfaces du domaine Ω_i . On définit les opérateurs de restriction R_i et de prolongation R_i^T : R_i est une matrice rectangulaire $n \times n_{local}$ qui réduit un vecteur x de Ω dans Ω_i . R_i^T est l'opération inverse. Elle prolonge un vecteur de Ω_i dans Ω . Elle complète de zéro le vecteur pour obtenir un vecteur de dimension n . La matrice

$$A_{\Omega_i} = R_i^T A R_i$$

est une restriction de A au sous domaine Ω_i .

Méthode de Schwarz multiplicative L'algorithme de Schwarz multiplicatif consiste à traiter successivement les sous problèmes Ω_i en prenant des conditions aux limites basées sur les plus récentes solutions des autres domaines. Il est en ce sens équivalent à une procédure de Gauss Seidel par bloc.

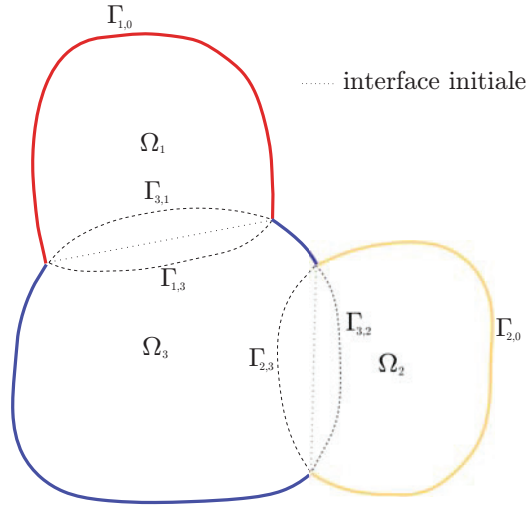


FIGURE 4.6 – Décomposition en domaine avec recouvrement.

Algorithme 11 Méthode multiplicative de Schwarz avec coloriage des domaines.

Jusqu'à convergence faire :

Pour couleur = 1 jusqu'à n_couleur

Si (couleur = couleur(Omega_i))

récupère les données externes y_i

Calculer le résidu local r_i = (b - Ax)_i = b_i - A_i x_i - B_i y_i

Résolution de A_{Omega_i} delta_i = r_i

Mise à jour de la solution x_i = x_i + delta_i

L'algorithme est séquentiel et une fois le système local résolu, le processeur ne travaille plus. La séquentialité peut être rompue par différentes méthodes. Une technique de coloriage des sous domaines permet d'identifier les sous-problèmes qui peuvent être résolue indépendamment (Algorithme 11). Une autre méthode pour remédier à ce temps d'attente est de diviser chaque sous domaine Ω_i en sous domaines et de réaliser le coloriage sur ces sous domaines. Chaque processeur possède alors un sous domaine à résoudre.

Méthode de Schwarz additive à recouvrement (ASM) La méthode de Schwarz additive peut être vue comme une parallélisation de la méthode multiplicative. Chaque domaine est résolu simultanément par tout les processeurs, et les résultats sont ensuite ajoutés uniquement lorsque tout les sous domaines ont été résolues :

$$x^{n+1} = x^n + \sum_i R_i^T A_i^{-1} R_i (b - Ax^n)$$

$$M_{AS}^{-1} = \sum_i R_i^T A_i^{-1} R_i$$

Cette méthode est souvent utilisée comme préconditionneur des méthodes de Krylov. La matrice M_{AS}^{-1} représente ce préconditionneur. Lorsque le recouvrement est nul, l'itération de Schwarz correspond

Algorithme 12 Méthode additive de Schwarz.

- Récupère les données externes y_i
 - Calculer le résidu local $r_i = (b - Ax)_i = b_i - A_i x_i - B_i y_i$
 - Résolution de $A_i \delta_i = r_i$
 - Mise à jour de la solution $x_i = x_i + \delta_i$
-

à un algorithme de Jacobi par bloc. L'Algorithme (12) consiste à restreindre le résidu $r = b - Ax$ pour pouvoir déterminer une correction locale $\delta_i = A^{-1} r_i$. Les systèmes locaux $A_i \delta_i = r_i$ peuvent être résolus de différentes manières : par un solveur directe, ou itérativement avec une méthode de Krylov préconditionnée par exemple. L'algorithme de Schwarz additif nécessite plus d'itérations que l'algorithme de Schwarz multiplicatif, mais présente un meilleur parallélisme.

Méthode de Schwarz additive et restreinte (RAS) L'algorithme RAS a été introduit par Cai and Sarkis (1999). Il requiert généralement moins d'itérations que la méthode ASM pour converger. Dans sa forme restrictive, la matrice de restriction est celle obtenue pour une décomposition sans recouvrement et est notée \tilde{R}_i .

$$x^{n+1} = x^n + \sum_i \tilde{R}_i^T A_i^{-1} R_i (b - Ax^n)$$

$$M_{RAS}^{-1} = \sum_i \tilde{R}_i^T A_i^{-1} R_i$$

Durant les itérations, un domaine Ω_i contribue à la solution uniquement pour ses variables internes : les résidus des cellules ghosts sont utilisés lors du calcul de $\delta_i = A_i^{-1} r_i$, mais les valeurs de δ_i obtenues pour les cellules ghosts ne sont pas utilisées lors de l'addition des solutions sur tout les sous domaines.

4.3 Évaluation des performances parallèles

La section 4.2 a mis en évidence deux grandes familles de méthodes itératives pour la résolution des systèmes linéaires creux :

- les méthodes multigrilles algébriques (AMG). Ces méthodes nécessitent de définir un algorithme de construction de grilles grossières et un opérateur d'interpolation d'une grille grossière à une grille fine. Les algorithmes de construction de grilles grossières peuvent être classés selon la complexité des grilles générées. L'algorithme classique Falgout produit des grilles grossières plus denses que les algorithmes PMIS et HMIS. Le choix de l'opérateur d'interpolation utilisé est réalisé en tenant compte de la complexité des grilles générées : l'algorithme Falgout peut être utilisé avec une interpolation classique, simple, alors que la plus faible complexité des algorithmes PMIS et HMIS nécessite d'utiliser des interpolations plus complexes tels que FF ou FF1.

- les méthodes de Krylov, en particulier la méthode GMRES, dont l'efficacité peut être améliorée grâce à l'utilisation d'un préconditionneur. Les méthodes classiques de résolution (SSOR, Gauss Seidel, Jacobi) peuvent être utilisées comme méthodes de préconditionnement. Dans notre cas, on préfère un préconditionnement ILU plus robuste. Les méthodes AMG sont également utilisées comme préconditionneurs.

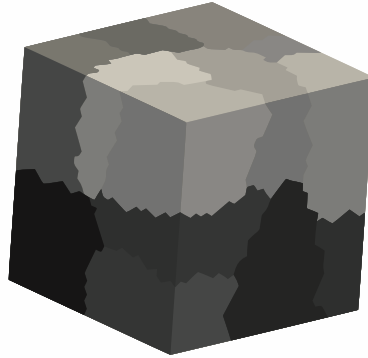


FIGURE 4.7 – Partitionnement d’une boîte cubique non structuré.

L’efficacité de ces méthodes et de leurs parallélisations est évaluée par deux types de tests :

- test de speed-up : un problème de taille globale fixe est résolu sur 1 à n processeurs. Le speed up est défini comme le rapport entre le temps de simulation sur n processeurs et le temps de simulation sur un processeur. Idéalement, le temps de simulation sur n processeurs est n fois plus petit que le temps obtenu sur un processeur.

- test de scale-up : la taille du problème est proportionnelle au nombre de processeurs. La taille du problème local est identique quelque soit le nombre de processeurs, et idéalement le temps de simulation sur n processeurs doit être identique au temps de simulation sur un processeur.

Les résultats ont été obtenus au centre de calcul haute performance P2CHPD de l’université Lyon1. La machine de calcul est une plateforme HPC de milieu de gamme. Chaque noeud est composé de deux quadri-processeurs Intel Xeon avec une vitesse CPU de 2.9 GHz et les noeuds sont connectés par un réseau haute performance Infiniband.

4.3.1 Équation de Laplace

Les différentes étapes utilisées pour la résolution des équations de Navier Stokes couplées nécessitent la résolution de systèmes linéaires de différentes natures. L’étape de prédiction sur la vitesse conduit à la discrétisation d’une équation de convection-diffusion. Lors de cette étape, une relaxation implicite ($\alpha = 0.7$) rend la matrice du système à résoudre à diagonale dominante. Ce n’est pas le cas pour l’étape de correction utilisée lors de la détermination de la pression. Elle conduit à la discrétisation d’un Laplacien sans relaxation implicite.

Ainsi, afin de sélectionner les meilleures méthodes de résolution de systèmes linéaires adaptées à notre problème, des tests de scale-up et speed-up sont réalisés sur une équation de Laplace $\Delta\phi = 1$ avec une condition de Dirichlet sur le bord $\phi_\Gamma = 0$. L’équation de Laplace est relaxée implicitement et les tests de speed-up sont réalisés pour différentes valeurs du paramètre de relaxation ($\alpha = 1$ et $\alpha = 0.7$). Le domaine de calcul est un cube unitaire avec des cellules tétraédriques (cf. Figure 4.7).

4.3.1.1 Validation de l'implémentation de l'assemblage parallèle

La parallélisation de l'assemblage est évaluée sur un maillage de 800 000 cellules. Sur la Figure (4.8), le speed-up est excellent et peu parfois selon la charge des noeuds de calcul être supérieur au speed-up idéal.

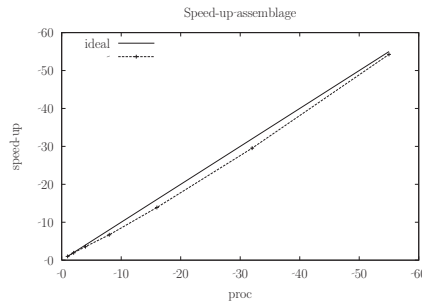


FIGURE 4.8 – Speed up sur une équation de Laplace.

4.3.1.2 Méthode multigrille algébrique

Dans cette section, on étudie l'influence des différents paramètres majeures associés aux méthodes multigrilles algébriques, à savoir, la valeur du seuil θ pour la construction des grilles, le choix de l'algorithme de construction des niveaux de grilles, et l'opérateur d'interpolation. A moins d'être spécifiés, les autres paramètres suivants sont utilisés :

- cycle en V
- méthode de Relaxation : hybride SSOR / Jacobi
- seuil 0.25
- maximum de niveau de grille : 25
- tolérance de la convergence relative 1×10^{-7} , absolue $= 1 \times 10^{-50}$

Influence des algorithmes de construction des niveaux de grilles, de l'interpolation et du seuil θ

Compte tenu des propriétés des algorithmes d'interpolation et de construction des niveaux de grilles présentés dans la section 4.2.2, l'algorithme Falgout associé à l'interpolation classique est comparé aux algorithmes PMIS et HMIS associés alternativement aux interpolations FF et FF1. D'autre part, l'influence du paramètre θ définissant la notion de "liens forts" est étudiée pour les algorithmes Falgout-classical et HMIS-FF. L'équation de Laplace est résolue avec une relaxation implicite $\alpha = 1$ et $\alpha = 0.7$ qui a pour effet d'augmenter la diagonale de la matrice. La complexité de l'opérateur C_{op} , le nombre d'itération (it), de grilles (grilles) et les temps de résolution (t_{solve}) sont donnés pour chacune des deux relaxations dans les Tables (4.1) et (4.2).

Pour une relaxation implicite $\alpha = 0.7$ du Laplacien, l'augmentation du seuil θ permet pour l'algorithme Falgout (cf. Table 4.1a) de diminuer la complexité de l'opérateur et ainsi de réduire le temps de résolution d'un facteur proche de 2, en améliorant le speed-up sur 32 processeurs. Pour l'algorithme HMIS-FF (cf. Table 4.1b) pour lequel la complexité est déjà faible, l'augmentation de θ améliore plus faiblement le temps de résolution, et le speed-up sur 32 processeurs varie peu.

θ	C_{op}	it	grilles	t_{solve}	speed-up(16)	$t_{solve}(32)$	speed-up(32)
0.25	7.39	4	8	2.20	6.98	1.24	12.39
0.50	5.37	5	8	1.96	7.94	0.99	15.80
0.75	3.52	6	7	1.49	7.89	0.80	14.82

(a) Algorithme Falgout-classical.

θ	C_{op}	it	grilles	t_{solve}	speed-up(16)	$t_{solve}(32)$	speed-up(32)
0.25	3.95	6	3	1.53	9.08	0.77	18.18
0.50	3.85	6	4	1.66	7.98	0.78	17.07
0.75	3.03	6	5	1.25	8.90	0.60	18.43

(b) Algorithme HMIS-FF.

TABLE 4.1 – Influence du seuil pour l'équation de Laplace, 16 processeurs, 50 000 dof / proc, $\alpha = 0.7$.

θ	C_{op}	it	grilles	t_{solve}	speed-up(16)	$t_{solve}(32)$	speed-up(32)
0.25	16.05	13	16	12.56	3.99	12.53	4.00
0.50	9.41	18	18	10.68	5.53	9.54	6.19
0.75	4.91	40	18	12.37	5.75	10.20	6.98

(a) Algorithme Falgout-classical.

θ	C_{op}	it	grilles	t_{solve}	speed-up(16)	$t_{solve}(32)$	speed-up(32)
0.25	4.64	13	6	3.87	15.33	2.67	22.21
0.50	×	×	×	×	×	×	×
0.75	×	×	×	×	×	×	×

(b) Algorithme HMIS-FF.

(×) : problème rencontré lors de la construction des grilles

TABLE 4.2 – Influence du seuil pour l'équation de Laplace, 16 processeurs, 50 000 dof / proc, $\alpha = 1$.

Pour une relaxation $\alpha = 1$, le comportement des solveurs multigrilles est différents. La matrice n'est plus à diagonale dominante et le temps nécessaire à la résolution du système linéaire augmente. Ce sont les algorithmes PMIS-FF et HMIS-FF qui donnent les temps de résolution les plus rapides avec $\theta = 0.25$, le temps de résolution étant approximativement deux fois plus grands qu'avec une relaxation $\alpha = 0.7$. En revanche, pour les algorithmes PMIS et HMIS, quelque soit l'interpolation utilisée, les seuils $\theta = 0.5$ et $\theta = 0.75$ conduisent à des problèmes dans la construction des niveaux de grilles.

Comparaison AMG accélérée par GMRES et solveur AMG

Les tests présentés précédemment ont été réalisés avec la librairie HYPRE, et la méthode multigrille a été utilisée en tant que solveur. Une interface entre les librairies PETSC et HYPRE permet d'utiliser les méthodes multigrilles algébriques comme préconditionneur des méthodes itératives proposées par PETSC. Cela permet de comparer le solveur AMG avec les méthodes de Krylov et Richardson préconditionnées par AMG. Les speed-up et temps de résolution sont présentés dans les Tables (4.3) et (4.4). Les valeurs de θ utilisées sont celles déterminées précédemment : $\theta = 0.25$ pour $\alpha = 1$ et $\theta = 0.75$ pour $\alpha = 0.7$.

Pour une relaxation $\alpha = 0.7$, d'une manière générale, le solveur AMG implémenté dans la librairie HYPRE donne les meilleures speed-up et l'itération de Richardson préconditionnée avec AMG, les

meilleurs temps de simulation. Cependant les différences entre les différents solveurs sont inférieures à 20 pour cent. La méthode LGMRES requiert moins d'itérations que le solveur multigrille, mais l'écart est faible et de l'ordre de 20 pour cent.

Lorsque la matrice est plus raide, avec une relaxation $\alpha = 1$, le solveur GMRES avec préconditionnement multigrille permet de diminuer en moyenne d'un facteur deux, les temps de calcul obtenus avec les solveurs multigrilles. Comme pour la relaxation $\alpha = 0.7$, le nombre d'itérations requises pour converger est également le plus faible pour la méthode LGMRES, mais la matrice étant plus raide, l'écart est plus important, de l'ordre de 100 pour cent. Ce résultat est cohérent avec la diminution par deux du temps de calcul par rapport à l'utilisation du solveur sans accélération GMRES.

4.3.1.3 Préconditionnement ILU

Les bibliothèques HYPRE et PETSC fournissent chacune différentes implémentations du préconditionnement incomplet LU, basées soit sur le niveau k de remplissage (méthode ILU(k)), ou soit sur une valeur seuil. Leurs performances sont évaluées sur l'équation de Laplace pour un processeur . Les Tables (4.5a) et (4.5b) donnent pour différentes valeurs de relaxation de l'équation de Laplace $\alpha = 1$ et $\alpha = 0.7$, le nombre d'itérations (it) nécessaires pour atteindre la convergence, et le temps (tps) pour la résolution du système des différentes méthodes testées. Une tolérance relative 10^{-7} est utilisée comme critère d'arrêt du solveur. Pour les deux relaxations étudiées, la préconditionnement basée sur le niveau de remplissage donne les meilleure temps de calcul : l'augmentation du nombre de niveaux permet de diminuer le nombre d'itérations nécessaires. Cependant, l'augmentation du nombre de niveaux accroît le temps nécessaire pour calculer le préconditionneur. La valeur optimale du nombre de niveaux est seulement de un pour $\alpha = 0.7$ et augmente lorsque la relaxation diminue ($\alpha = 1$) car le problème est plus raide et le nombre d'itérations (it) augmente. Sur un processeur, pour une relaxation $\alpha = 0.7$, la méthode ILU(1) est plus efficace que les méthodes multigrilles algébriques, alors que pour $\alpha = 1$, lorsque le système à résoudre est plus raide, le préconditionnement par des méthodes multigrilles est plus rapide.

4.3.1.4 Préconditionnement de Schwarz

On étudie les différentes versions du préconditionnement de Schwarz additif (version standard et restreinte) avec différents niveaux de recouvrement (1,3 et 5). Étant donné les résultats précédents, on limite l'étude du préconditionnement de Schwarz au cas où l'équation de Laplace est relaxée implicitement ($\alpha = 0.7$). Le préconditionnement de Schwarz est utilisé comme accélérateur de la méthode itérative LGMRES préconditionnée localement par ILU(k). Ainsi, la Table (4.6) donne le nombre d'itérations, le temps de résolution et le speed-up des différentes méthodes. Les résultats montrent que la version restreinte du préconditionnement de Schwarz est la plus performante en terme de temps de calcul et de speed-up. L'inefficacité du préconditionnement standard, comparée à la version restreinte provient du nombre d'itérations nécessaires à la résolution qui augmente avec le nombre de processeurs, alors que sur la plage du nombre de processeurs utilisés, pour la version restreinte avec un recouvrement de 3, le nombre d'itérations reste constant à partir de deux processeurs. Lorsque le recouvrement augmente, un nombre plus important d'échanges inter-processeurs diminue le speed-up.

	LGMRES	Richardson	Solveur	LGMRES	Richardson	Solveur
proc	speed-up			temps(s)		
1	1.00	1.00	1.00	7.53	6.82	10.51
2	1.46	1.38	1.92	5.14	4.93	5.46
4	2.44	2.48	2.95	3.09	2.75	3.56
8	3.30	3.39	4.02	2.28	2.01	2.61
16	6.96	7.26	8.59	1.08	0.94	1.22
32	13.36	13.43	16.70	0.56	0.51	0.63

(a) PMIS-FF $\theta = 0.75$.

	LGMRES	Richardson	Solveur	LGMRES	Richardson	Solveur
proc	speed-up			tps(s)		
1	1.00	1.00	1.00	6.93	6.21	10.10
2	1.53	1.66	1.95	4.52	3.75	5.17
4	2.36	2.38	3.01	2.94	2.60	3.35
8	3.15	3.27	3.96	2.20	1.90	2.55
16	6.78	7.00	8.79	1.02	0.89	1.15
32	12.44	11.95	16.11	0.56	0.52	0.63

(b) PMIS-FF1 $\theta = 0.75$.

	LGMRES	Richardson	Solveur	LGMRES	Richardson	Solveur
proc	speed-up			tps(s)		
1	1.00	1.00	1.00	7.72	7.00	11.11
2	1.51	1.24	1.91	5.11	5.64	5.82
4	2.35	2.37	2.93	3.29	2.95	3.79
8	3.21	3.36	4.03	2.40	2.08	2.76
16	6.99	7.03	8.78	1.11	1.00	1.27
32	12.93	13.19	16.53	0.60	0.53	0.67

(c) HMIS-FF $\theta = 0.75$.

	LGMRES	Richardson	Solveur	LGMRES	Richardson	Solveur
proc	speed-up			tps(s)		
1	1.00	1.00	1.00	7.45	6.75	11.56
2	1.55	1.67	2.08	4.81	4.05	5.57
4	2.39	2.41	3.25	3.11	2.80	3.56
8	3.21	3.35	4.36	2.32	2.02	2.65
16	7.22	7.19	9.68	1.03	0.94	1.20
32	13.62	13.34	18.70	0.55	0.51	0.62

(d) HMIS-FF1 $\theta = 0.75$.TABLE 4.3 – Comparaison AMG préconditionneur et AMG solveur : temps de résolution et nombre d'itérations pour relaxation $\alpha = 0.7$.

	LGMRES	Richardson	Solveur	LGMRES	Richardson	Solveur
proc	speed-up			temps(s)		
1	1.00	1.00	1.00	24.15	53.19	57.56
2	2.58	4.81	3.52	9.34	11.06	16.37
4	4.36	8.46	6.57	5.54	6.29	8.76
8	5.79	10.51	8.04	4.17	5.06	7.16
16	9.76	18.16	15.04	2.47	2.93	3.83
32	13.42	25.72	21.40	1.80	2.07	2.69

(a) PMIS-FF $\theta = 0.25$.

	LGMRES	Richardson	Solveur	LGMRES	Richardson	Solveur
proc	speed-up			tps(s)		
1	1.00	1.00	1.00	20.14	45.45	51.24
2	2.21	4.33	3.43	9.11	10.50	14.92
4	4.23	7.75	6.08	4.76	5.86	8.42
8	5.29	9.22	7.29	3.81	4.93	7.03
16	9.30	16.80	12.63	2.16	2.70	4.06
32	12.45	21.44	15.11	1.62	2.12	3.39

(b) PMIS-FF1 $\theta = 0.25$.

	LGMRES	Richardson	Solveur	LGMRES	Richardson	Solveur
proc	speed-up			tps(s)		
1	1.00	1.00	1.00	23.55	45.74	60.95
2	2.33	3.94	4.09	10.09	11.61	14.90
4	3.98	6.90	7.17	5.92	6.63	8.50
8	5.12	8.73	8.44	4.60	5.24	7.22
16	8.84	15.77	15.84	2.66	2.90	3.85
32	12.91	20.58	24.10	1.82	2.22	2.53

(c) HMIS-FF $\theta = 0.25$.

	LGMRES	Richardson	Solveur	LGMRES	Richardson	Solveur
proc	speed-up			tps(s)		
1	1.00	1.00	1.00	19.92	40.81	54.32
2	2.30	4.02	3.94	8.65	10.16	13.80
4	3.97	6.98	6.71	5.02	5.84	8.09
8	5.02	8.52	7.91	3.97	4.79	6.87
16	8.88	15.17	14.03	2.24	2.69	3.87
32	11.49	20.00	16.80	1.73	2.04	3.23

(d) HMIS-FF1 $\theta = 0.25$.TABLE 4.4 – Comparaison AMG préconditionneur et AMG solveur : temps de résolution et nombre d'itérations pour relaxation $\alpha = 1$.

Préconditionneur	librairie	it	tps (s)
ILU(0)	PETSC	8	2.25
ILU(0) avec rcm **	PETSC	7	2.14
ILU(1)	PETSC	5	2.07
ILU(2)	PETSC	4	2.36
euclid 0	HYPRE	8	4.26
euclid 1	HYPRE	5	4.04
euclid 2	HYPRE	4	4.36
euclid 3	HYPRE	4	5.76
ILU pilut	HYPRE	13	12.52
ILUT (0.0001, 0.01-3)*	PETSC	15	4.53
ILUT (0.001, 0.01-3)*	PETSC	15	4.72
ILUT (0.01, 0.01-3)*	PETSC	15	4.54
ILUT (0.01, 0.01, 6)*	PETSC	9	3.98
ILUT (0.1, 0.01, 3)*	PETSC	15	4.82
aucun		34	11.26

(a) $\alpha = 0.7$.

Préconditionneur	librairie	it	tps (s)
ILU(0)	PETSC	212	69.54
ILU(0) avec rcm **	PETSC	168	55.47
ILU(1)	PETSC	113	42.95
ILU(2)	PETSC	77	33.33
euclid 0	HYPRE	212	79.09
euclid 1	HYPRE	113	48.96
euclid 2	HYPRE	77	37.06
euclid 3	HYPRE	62	36.47
ILU pilut	HYPRE	480	222.73
ILUT (0.0001, 0.01-3)*	PETSC	759	204.03
ILUT (0.001, 0.01-3)*	PETSC	759	212.90
ILUT (0.01, 0.01-3)*	PETSC	759	215.73
ILUT (0.01, 0.01, 6)*	PETSC	230	73.64
ILUT (0.1, 0.01, 3)*	PETSC	759	213.76
aucun		1000	252.13

(b) $\alpha = 1$.

(*) : ILUT(i, j, k) avec i : seuil ILUT, j : tolérance pour la permutation des colonnes, k : nombre maximum de valeurs non nulles par ligne.

(**) : rcm, avec re-numérotation Reverse Cuthill-MacKee

TABLE 4.5 – Préconditionnement ILU.

Pour le cas testé, les meilleurs résultats en termes de temps de calcul et de speed-up sont obtenus avec la version restreinte et un recouvrement égale à 3.

Version	standard(1*)			standard(3*)			standard(5*)		
	proc	it	tps(s)	Speed-up	it	tps	Speed-up	it	tps
1	2	5.55	1.00	2	5.55	1.00	2	6.66	1.00
2	5	6.16	0.90	4	5.23	1.06	4	5.47	1.22
4	10	5.50	1.01	10	5.49	1.01	10	6.19	1.08
8	11	3.48	1.59	12	3.92	1.42	13	4.55	1.46
16	12	1.81	3.07	14	2.39	2.32	15	2.71	2.46
32	12	0.94	5.90	14	1.28	4.34	14	1.43	4.66

Version	restreinte(1*)			restreinte(3*)			restreinte(5*)		
	proc	it	tps(s)	Speed-up	it	tps	Speed-up	it	tps
1	2	5.85	1.00	2	4.73	1.00	2	3.06	1.00
2	5	7.22	0.81	3	4.45	1.06	2	2.28	1.34
4	5	2.89	2.02	3	2.24	2.11	2	0.94	3.26
8	6	1.93	3.03	3	1.15	4.11	3	1.18	2.59
16	6	0.94	6.22	3	0.59	8.02	3	0.68	4.50
32	6	0.48	12.19	3	0.38	12.45	3	0.41	7.46

TABLE 4.6 – Préconditionnement de Schwartz : *, recouvrement ; it, nombre total d'itérations ; tps, temps de résolution pour $\alpha = 0.7$.

4.3.1.5 Scale up

Les tests de scale-up explorent différents nombres de degrés de liberté par processeur (dofs), de 12 500 à 400 000. Les méthodes AMG décrites dans le paragraphe 4.2.2 sont comparées au préconditionnement ILU. Dans la Figure (4.9a), on montre les résultats de scale up. Les résultats mettent en évidence la supériorité du préconditionnement AMG sur le préconditionnement parallèle ILU. La scalabilité de ILU est pauvre à cause d'une augmentation du nombre d'itérations avec la taille du problème (cf. Table 4.7). Il est intéressant de noter dans la Table (4.8) que l'algorithme de construction des grilles à faible complexité (PMIS) associé à une interpolation FF1 est plus efficace que l'algorithme de construction de grille Falgout utilisé avec une interpolation classique : la complexité est plus faible et le temps de calcul est réduit. L'algorithme de sélection de grilles HMIS utilisé avec l'interpolation FF ou FF1 donne des résultats similaires. La méthode multigrille avec l'algorithme de construction de grille PMIS et l'interpolation FF1 donne les meilleurs résultats. Sur 64 processeurs, le temps de résolution est 500 fois plus faible que celui obtenu avec le préconditionnement ILU. L'utilisation d'une interpolation classique avec un algorithme de construction de grille à faible complexité (HMIS ou PMIS) conduit à une augmentation du nombre d'itérations à cause de grilles moins denses. La Figure (4.9b) montre l'influence de la taille du problème sur la scalabilité : lorsque le nombre de degrés de liberté par processeur n'est pas suffisant, les temps de communications deviennent importants et se traduisent par une perte de scalabilité.

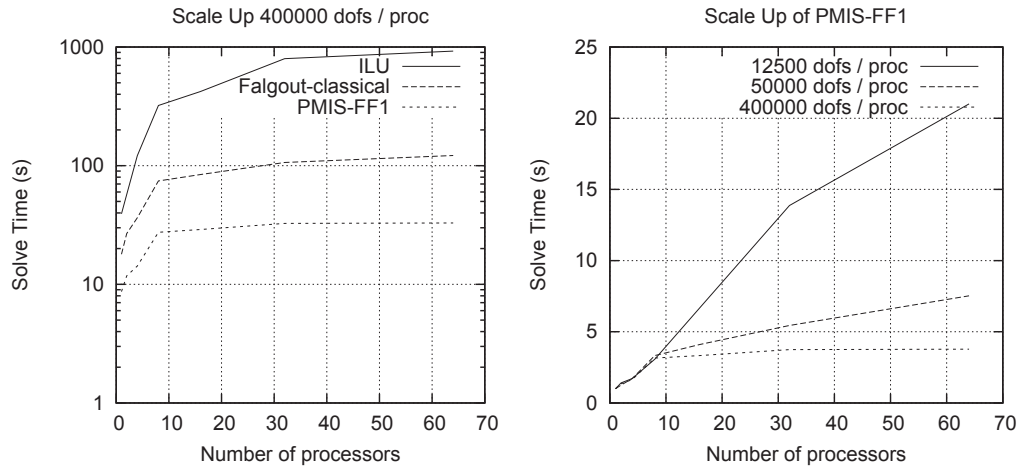


FIGURE 4.9 – Résultat de scale-up pour l'équation de Laplace.

Méthode / processeur	1	2	4	8	16	32
ILU	226	346	368	500	774	955
PMIS -FF1	13	14	14	14	16	16
Falgout classical	11	11	11	12	13	13

TABLE 4.7 – Nombre d'itérations avec 200 000 degré de liberté / processeur.

Méthode	Complexité des grilles	Complexité de l'interpolation	nb it
Falgout classical	2.6	18.5	13
PMIS - FF1	1.42	3.80	16
PMIS - classical	1.42	2.28	37

TABLE 4.8 – Détails du préconditionnement AMG avec 32 processeurs, 200 000 dofs.

4.3.2 Écoulement turbulent autour de l'usine de Feyzin

On s'intéresse à l'écoulement incompressible isotherme autour de la raffinerie de Feyzin. Le modèle standard $k - \epsilon$ (Launder and Spalding, 1972) est utilisé avec des lois de parois (Launder and Spalding, 1974) pour simuler l'écoulement turbulent. Les coefficients de fermeture du modèle sont modifiés selon Dyunkerke (1988) pour prendre en compte la couche limite atmosphérique. Une partie du maillage 3D non structuré utilisé dans la simulation est affichée dans la Figure (4.10). Il contient approximativement 5×10^6 tétraèdres.

Comme l'ont montré les tests de scale-up et speed-up, le préconditionnement multigrille avec l'algorithme de construction de grilles PMIS et l'interpolation FF1 donne les meilleurs résultats en terme de speed up et de temps de calcul pour une équation de Laplace raide. Cette méthode de préconditionnement est alors utilisée pour la simulation parallèle du transport d'un polluant. Le panache du polluant est affiché dans la Figure (4.11). La Figure (4.12) montre que le speed up est excellent avec un maillage de 5×10^6 cellules, mais n'est pas si bon avec 1×10^6 cellules à cause des temps de communication.

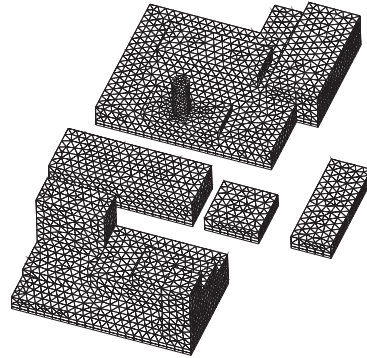


FIGURE 4.10 – Maillage du site industriel de Feyzin.

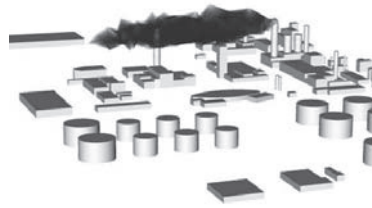


FIGURE 4.11 – Panache.

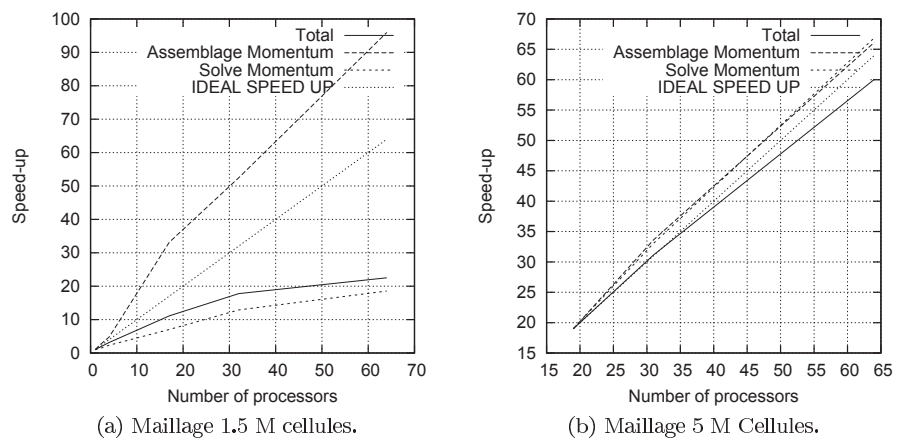


FIGURE 4.12 – Speed up du solveur Navier Stokes sur l'écoulement de Feyzin.

4.4 Conclusion

Les tests de speed-up basés sur la résolution d'une équation de Laplace relaxée implicitement ont permis de mettre en évidence des comportements différents selon le paramètre de relaxation utilisé. Sans relaxation ($\alpha = 1$) et par conséquent en l'absence de diagonale dominante de la matrice, le solveur GMRES préconditionné avec une méthode multigrille composée d'un algorithme de construction de grilles à faible complexité et d'une interpolation plus complexe (PMIS-FF, PMIS-FF1, HMIS-FF, ou HMIS-FF1), donne le temps de résolution le plus faible. En relaxant implicitement, la matrice devient à diagonale dominante et possède de meilleures propriétés numériques pour la convergence des méthodes itératives. L'algorithme GMRES préconditionnée par ILU donne alors les meilleurs temps de résolution lorsque le nombre de processeurs est faible. Pour un nombre plus important de processeurs, au delà de 16, étant donné que le speed-up est d'une manière générale moins satisfaisant que celui obtenu avec les méthodes multigrilles (quelque soit la manière dont est parallélisé la méthode ILU (Schwarz ou Euclid)), les méthodes multigrilles deviennent plus efficaces.

Enfin, le test de scale-up a montré, qu'en présence d'une matrice issue de la discrétisation d'une équation de Laplace non relaxée, si le nombre de degrés de liberté par processeur est suffisant, les méthodes multigrilles algébriques possèdent des meilleures propriétés de scalabilité. Ainsi, pour le cas d'un écoulement autour de l'usine de Feyzin pour lequel le nombre de cellules du maillage est important, le speed-up obtenu sur 64 processeurs est relativement satisfaisant.

Conclusion

Dans le but de simuler efficacement la dispersion atmosphérique de polluants sur des sites industriels complexes, l'outil CFD est indispensable et remplace des outils plus simplistes tels que les modèles gaussiens et les modèles intégraux. Ainsi, avec l'objectif de réaliser efficacement des simulations précises d'écoulements atmosphériques avec transport de polluants, une approche volumes finis parallèle, d'ordre élevé sur maillages non structurés est proposée. Une étude bibliographique non exhaustive a permis de déterminer un problème modèle adapté à la modélisation des écoulements atmosphériques avec transport de polluants. Pour résoudre le système d'équations de ce modèle d'écoulement incompressible, une approche volumes finis de type SIMPLE a été utilisée. Elle découple les équations de quantité de mouvement et de continuité par un algorithme itératif composé d'une étape de prédiction sur la vitesse et d'une étape de projection déterminant la pression à partir d'une équation de Laplace.

Pour acquérir la précision nécessaire lors des simulations, deux schémas d'ordre élevé (schéma Padé et reconstruction polynomiale) ont été étudiés pour le traitement des termes convectifs et diffusifs. Un schéma compact Padé volumes finis basé sur un polynôme mono-dimensionnel de degré 3, a été proposé. Ce schéma requiert la résolution d'un système linéaire à diagonale dominante. Le schéma polynomial proposé par Carpentier (2000) pour des écoulements compressibles a été adapté pour les écoulements incompressibles étudiés. Ce schéma reconstruit par la méthode des moindres carrés un polynôme multi-dimensionnel de degré 3 sur une molécule fixe. Le polynôme et sa dérivée sont alors intégrés sur les faces pour calculer respectivement le flux convectif et le flux diffusif. Pour les deux schémas, la présence d'une vitesse non uniforme dans le terme convectif de l'équation de transport nécessite un traitement particulier pour atteindre une précision supérieure à l'ordre 2.

Différents tests numériques ont montré la supériorité et l'intérêt de ces schémas d'ordre élevé. La précision des termes convectifs est d'ordre 4 sur les maillages structurés uniformes, le schéma Padé donnant l'erreur de troncature la plus faible. Sur un maillage non structuré, la précision atteinte par la reconstruction polynomiale pour les termes convectifs est d'ordre 3 au moins, et 4 si le maillage est uniforme (maillage triangulaire uniforme par exemple). Ainsi, ces schémas permettent d'utiliser, pour atteindre une précision donnée, des maillages beaucoup plus grossiers que ceux nécessaires avec des schémas d'ordre 2. La possibilité d'utiliser des maillages plus grossiers est intéressante pour les applications industrielles dans lesquelles la complexité des géométries étudiées ne permet en général pas de réaliser des maillages particulièrement fins.

Les deux schémas ont été implémentés dans le même code de calcul et leur efficacité a été analysée et comparée. La résolution itérative de la relation de Padé par une méthode de Jacobi sans calcul implicite de la matrice pour la détermination du gradient n'est pas efficace. Cependant, les relations

de Padé forment un système linéaire creux et dans le cas d'un maillage cartésien, l'utilisation du solveur linéaire implicite déjà utilisé dans les étapes de prédiction et correction pour la résolution des équations de Navier Stokes, permet un gain de temps d'un facteur 6 sur les maillages grossiers par rapport à la méthode de Jacobi, sans coût mémoire supplémentaire. Le gain est plus important sur les maillages plus fins. Pour le schéma polynomial, la construction du polynôme et son intégration sur les faces est coûteuse. Le pré-calcul et stockage de coefficients permet un gain de temps conséquent : le temps de simulation est seulement doublé par rapport à celui obtenu avec un schéma d'ordre 2. Ainsi, l'implémentation efficace des schémas, associée à la précision d'ordre élevé permet un gain de temps important. Par exemple, pour le cas d'un écoulement tri-dimensionnel autour d'un obstacle cubique, un maillage 64 fois plus important ($\approx 7\,200\,000$ cellules) a été nécessaire avec le schéma décentré d'ordre 2 pour obtenir une précision identique à celle obtenue avec un maillage grossier ($\approx 110\,000$ cellules) et le schéma Padé. Le schéma Padé a nécessité 52 minutes de calcul sur un processeur et 6 minutes sur 16 processeurs, alors que le schéma d'ordre 2 requiert 38 heures de calcul sur 16 processeurs.

Cependant, certains problèmes subsistent : le schéma Padé est utilisable uniquement sur des maillages cartésiens. Dans le cas de maillages plus complexes, il est alors nécessaire d'utiliser les schémas à reconstruction polynomiale pour atteindre une précision d'ordre élevé. Néanmoins, les coûts mémoires associés à ce type de schéma sont importants et la positivité du Laplacien calculé grâce à l'interpolation polynomiale est faible. Une pondération adéquate de l'erreur des moindres carrés permet d'améliorer la positivité, mais pour certains maillages, le manque de positivité affecte la robustesse du solveur Navier Stokes. Toutefois, pour des nombres de Reynolds élevés, des expérimentations numériques sur la décroissance d'un tourbillon de Taylor ont montré que l'utilisation d'un schéma d'ordre élevé uniquement pour les termes convectifs peut se révéler suffisante.

De manière à résoudre en partie les problèmes de coûts mémoires associés au schéma à reconstruction polynomiale, et afin d'avoir des schémas d'ordre élevé compétitifs face à des méthodes classiques d'ordre 2, la parallélisation du code a été réalisée par la technique de décomposition en sous domaines en espace. L'assemblage des systèmes linéaires s'effectuant naturellement de manière parallèle, notre effort s'est concentré sur la parallélisation de la résolution des systèmes linéaires issus de la discrétisation implicite en utilisant les bibliothèques PETSC et HYPRE, l'objectif étant de sélectionner les solveurs itératifs les plus performants pour la résolution des systèmes linéaires issus de nos problèmes. Les tests de speed-up et scale-up sur une équation de Laplace ont mis en évidence l'influence de la relaxation implicite dans le choix du solveur. La relaxation implicite conduit à une matrice à diagonale dominante qui possède de meilleures propriétés numériques. Dans ce cas, la méthode GMRES préconditionnée par ILU donne les meilleures performances mais les speed-up sont beaucoup plus faibles que ceux obtenus avec les méthodes multigrilles algébriques. En l'absence de relaxation, la matrice perd sa diagonale dominante et le système linéaire à résoudre est plus raide. Dans ce cas, la méthode de Krylov GMRES associée à un préconditionnement multigrille algébrique donne les meilleures performances, en utilisant un algorithme de construction de grilles à faible complexité (PMIS ou HMIS) avec une interpolation complexe (FF ou FF1). Cependant, les tests de scale-up ont montré qu'une augmentation du nombre de processeurs diminue l'efficacité parallèle du solveur car les communications inter-processeurs deviennent non négligeables.

En conclusion, ce travail a permis de montrer que le développement et l'utilisation de schémas d'ordre élevé associés à une mise en oeuvre efficace sur des architectures parallèles, est une voie prometteuse

pour le développement des applications de l'outil CFD pour des problèmes de dispersion de polluants. Néanmoins, certains points nécessitent encore des développements et des études : l'extension des schémas Padé à des maillages non cartésiens en suivant l'approche de Lacor et al. (2004), la diminution du coût mémoire des schémas utilisant la reconstruction polynomiale et l'amélioration de l'efficacité parallèle sur les nouvelles architectures multi-coeurs.

Annexe 1

Algorithmes de construction de grilles pour solveur AMG

Les sous ensembles nécessaires à la description des différents algorithmes ont déjà été définis dans le manuscrit, pour plus de clarté, ils sont rappelés :

- le voisinage d'un point i noté N_i est l'ensemble des points $j \neq i$ qui influencent i . Un point i influence un point j si le coefficient de la matrice $a_{ij} \neq 0$. Une dépendance entre deux points i et j est dite forte si le critère suivant est respecté :

$$-a_{ij} \geq \theta \max_{k \neq i} (-a_{ik})$$

avec typiquement $\alpha = 0.25$.

- S_i l'ensemble des points qui influencent fortement i . S_i est un sous ensemble du voisinage N_i :

$$S_i = \{j \neq i : -a_{ij} \geq \theta \max_{k \neq i} (-a_{ik})\}$$

- S_i^T l'ensemble des points qui dépendent du point i .
- pour chaque point $i \in F$, C_i est l'ensemble de points C utilisés pour l'interpolation du point i .

Algorithme classique de Ruge Stueben (RS)

L'algorithme introduit par J.W. Ruge (1987) est couramment utilisé aujourd'hui. Il est basé sur les deux heuristiques suivantes :

- (C1) pour chaque point $i \in F$, chaque point j de S_i est soit un point C , soit tel que $S_j \cap C_i \neq \emptyset$. C'est à dire, si i est un point F , alors les points influençant i doivent être soit des points C , ou doivent dépendre d'un point C utilisé pour interpoler l'erreur en i . (C1) permet d'assurer une meilleure interpolation.
- (C2) les points C forment un sous ensemble indépendant maximum avec la propriété qu'aucun point C ne dépend d'un autre point C . Si un point C est ajouté au sous ensemble maximal, la

propriété n'est plus vérifiée. Cette heuristique permet de garder une taille de grille grossière aussi faible que possible.

Les deux heuristiques permettent d'assurer la qualité de la grille, en contrôlant sa taille. Il n'est pas toujours possible de respecter les deux heuristiques. L'heuristique (C1) est alors appliquée rigoureusement alors que l'heuristique (C2) est utilisée comme guide. L'algorithme de Ruge Stueben sélectionne les points C en deux passes. Dans un premier temps, on attribue à chaque point i , une valeur λ_i égale au nombre de points qui dépendent du point i :

$$\lambda_i = |S_i^T|$$

Un point avec un maximum local de λ_i est sélectionné pour être le premier point C . Les points j qui dépendent fortement de ce point ($j \in S_i^T$) deviennent des points F . Les points k qui influencent fortement ces nouveaux points F ($k \in S_j$) se voient augmenter leur mesure pour accroître leur chance de devenir un nouveau point C . Ce processus séquentiel est répété tant que tous les points ne sont pas définis. A la fin du processus séquentiel, les sous ensembles C et F peuvent ne pas respecter l'heuristique (C1) : certaines dépendances $F - F$ ne partagent pas de point C . Une seconde passe est alors réalisée : chaque point F est examiné, et si une dépendance $F - F$ ne partage pas un point C commun, un des deux points F est changé en point C . Plus de détail sont disponibles dans la littérature (J.W. Ruge, 1987). La nature séquentielle de l'algorithme fait qu'il n'est pas naturellement parallélisable. L'algorithme suivant permet de remédier à ce problème.

Algorithme CLJP

L'algorithme CLJP (Cleary et al., 1998) évite le processus séquentiel de sélection des points C en ajoutant à la mesure λ_i un nombre aléatoire compris entre 0 et 1 pour la rendre distinctes. Un premier ensemble D de points est sélectionné. Un point i appartient à D si $\lambda_i > \lambda_k$ pour tout $k \in S_i \cap S_i^T$. La matrice S et les mesures λ de certains points du voisinage de i sont modifiés afin de vérifier les deux heuristiques suivantes :

- (H1) comme les valeurs pour un point C ne sont pas interpolées, les voisins qui influencent un point C sont moins éligibles à être potentiellement un point C .
- (H2) si deux points k et j dépendent d'un point C , et si j influence k , alors j est moins éligible à être un point C , puisque k peut être interpolé à partir de ce point C .

Les deux heuristiques sont implémentées selon l'Algorithme (13).

Algorithme 13 Algorithme de sélection des grilles CLJP

La mesure λ_j des points j ($j \in S_i$) qui influencent fortement un point C , d'indice $i \in D$ décroît pour diminuer leur possibilité de devenir un point C :

Pour chaque $i \in D$

 Pour chaque $j \in S_i$

 Décrémente $\lambda_j = \lambda_j - 1$

$S_{ij} = 0$

 Pour chaque $j \in S_i^T$

$S_{ji} = 0$

 Pour chaque $k \in S_j^T$

 Si k dépend de i ($k \in S_i^T$)

 Décrémente $\lambda_j = \lambda_j - 1$

$S_{kj} = 0$

Chaque fois que la décrémentation d'un point j est telle que $\lambda_j < 1$, le point j devient un point F . Lorsque l'heuristique a été appliquée, les points $i \in D$ deviennent des points C et une communication inter-processeurs est nécessaire pour mettre à jour les valeurs de λ pour les éléments ghosts du graphe. Un nouvel ensemble D est sélectionné à partir du graphe S modifié, et le processus se répète jusqu'à la définition de tous les points en points C ou F .

Un des avantages du CLJP est qu'il est entièrement parallèle. L'expérience (Henson and Yang (2000)) montre que l'algorithme a tendance à sélectionner un nombre de points C plus important que nécessaire, ce qui a pour effet d'augmenter la complexité algorithmique de l'opérateur.

Algorithme Falgout

L'algorithme parallèle proposé par Falgout combine l'approche de Ruge Stueben et l'algorithme CLJP. Chaque processeur applique l'algorithme de Ruge Stueben. L'ensemble des points C sélectionnés par l'algorithme exceptés ceux adjacents aux interfaces entre processeurs, sont utilisés comme le premier ensemble indépendant pour l'algorithme CLJP. L'algorithme diffère donc de CLJP uniquement dans la manière dont est choisi l'ensemble des points initiaux.

Algorithme PMIS (Parallel Maximum Independent Set)

Cet algorithme a été introduit afin de réduire la complexité des opérateurs présentés précédemment. Il débute comme CLJP par une mesure distincte, qui est la somme du nombre de points qui influencent i : un nombre aléatoire est ajouté aux mesures pour les rendre distinctes. Une recherche des maximums locaux de λ_i parmi les voisins j non assignés ($j \in S_i^T \cap S_i$), permet de déterminer un ensemble de points C . Les points j influencés par les nouveaux points C sont alors des points F et éliminés du graphe. La procédure se répète jusqu'à la définition de tous les points. L'algorithme PMIS permet d'avoir des connections fortes entre deux points F sans qu'ils aient un point C commun. La parallélisation nécessite uniquement l'échange d'informations aux niveaux des éléments ghosts du graphe. Plus de détail sont disponibles dans Sterck et al. (2006).

Algorithme PMIS Greedy

L'utilisation de l'algorithme PMIS génère une prédominance de connections fortes $F - F$ sans point C commun. Pour y remédier, Butler (2006) propose l'algorithme PMIS Greedy. Il reprend la structure de l'algorithme RS afin d'obtenir des grilles grossières plus structurées et avec moins de connections $F - F$ sans point C commun : une fois qu'un point F est défini dans l'algorithme PMIS, on augmente la valeur λ_j de tous les point j non assignés qui influencent fortement ce point F . L'auteur propose également d'appliquer uniquement l'algorithme PMIS sur les grilles les plus fines afin de tirer parti de la complexité réduite de l'algorithme, et d'utiliser l'algorithme CLJP qui a une meilleure propriété de convergence sur les grilles les plus grossières.

“Agressive coarsening”

Cette méthode reprend la définition de connectivité forte. Un point i dépend fortement d'un point j suivant un chemin de longueur l si il existe une séquence de variables i_0, \dots, i_l avec $i = i_0$ et $i_l = j$ telle que i_k dépend fortement de i_{k+1} . Un point i dépend fortement de j si au moins p chemins de longueur $\leq l$ existent. On définit alors la matrice S^l et la méthode “agressive coarsening” s'applique à n'importe quelle méthode à partir du graphe basé sur S^l . En pratique, $l = 2$ et $p = 1$ ou $p = 2$. L'application d'une méthode à S^2 peut être coûteuse et la méthode procède alors en deux étapes. Dans un premier temps, un algorithme de construction choisi est appliqué afin d'obtenir un ensemble de points C . La seconde étape applique l'algorithme à l'ensemble de points C selon S^2 .

HMIS (Hybrid Maximum Independent Set)

L'algorithme HMIS (Sterck et al. (2006)) est inspiré par l'hybridation de l'algorithme Falgout. La première passe de RS est réalisée pour obtenir un premier sous ensemble indépendant. Tous les points C non situés sur une interface sont utilisés alors par l'algorithme PMIS. Puisque l'algorithme ne requiert pas la vérification de l'heuristique (C1), seul la première passe de CLJP est réalisée.

Annexe 2

Opérateurs d'interpolation pour solveur AMG

L'opérateur d'interpolation constitue, avec l'algorithme de construction des grilles, les opérateurs les plus importants de la méthode multigrille. Avant d'introduire les différents schémas d'interpolation, quelques définitions communes à toutes les interpolations AMG doivent être définies. Les points auxquels un point F d'indice i est connecté peuvent être classés en trois sous-ensembles :

- le premier sous ensemble est l'ensemble C_i des points C qui influencent fortement i .
- parmi les points F , on distingue l'ensemble des points F_i^s qui influencent fortement i et l'ensemble F_i^w des points qui sont faiblement connectés à i .

L'erreur d'un point C est connue sur le maillage grossier, et peut être interpolée directement sur le maillage fin. Le schéma d'interpolation doit définir la manière dont est interpolée un point F . D'une manière générale on peut écrire :

$$(I_H^h e)_i = \begin{cases} e_i & \text{si } i \in C \\ \sum_{j \in C_i} \omega_{ij} e_j & \text{si } i \in F \end{cases} \quad (2.3)$$

Le schéma d'interpolation doit définir les poids ω_{ij} pour tous les points F , tels que l'erreur sur le maillage fin vérifie :

$$a_{ii} e_i + \sum_{j \in N_i} a_{ij} e_j \approx 0 \quad (2.4)$$

Interpolation directe

L'interpolation directe (Stuben, 1999) est l'interpolation la plus simple : si la plupart des points influençant fortement un point F d'indice i sont dans C_i , alors on a :

$$\frac{1}{\sum_{k \in N_i} a_{ik}} \sum_{j \in N_i} a_{ij} e_j \approx \frac{1}{\sum_{k \in C_i} a_{ik}} \sum_{j \in C_i} a_{ij} e_j \quad (2.5)$$

et l'utilisation de l'expression (2.3) dans l'expression(2.4) donne les coefficients d'interpolation :

$$\omega_{ij} = -\frac{a_{ij} \sum_{k \in N_i} a_{ik}}{a_{ii} \sum_{k \in C_i} a_{ik}}$$

Interpolation classique

En utilisant le découpage de l'ensemble N_i des points qui influencent i sur l'expression (2.4), on obtient :

$$a_{ii}e_i \approx -\sum_{j \in C_i} a_{ij}e_j - \sum_{j \in F_i^s} a_{ij}e_j - \sum_{j \in F_i^w} a_{ij}e_j$$

La somme sur l'ensemble F_i^s des points j peut être redistribuée sur la diagonale, mais Briggs et al. (2000) ont montré qu'il est préférable d'approximer l'erreur e_j des points F de l'ensemble F_i^s par une somme pondérée de l'erreur e_k pour $k \in C_i \cap C_j$, plutôt que de redistribuer les termes sur la diagonale. A condition que $C_i \cap C_j \neq \emptyset$, on peut utiliser alors :

$$e_j \approx \frac{\sum_{k \in C_i} a_{jk}e_k}{\sum_{k \in C_i} a_{jk}} \quad (2.6)$$

Ceci ne pose pas de problèmes pour les algorithmes classiques (RS, CLJP, Falgout) qui vérifient l'heuristique (C1) qui assure qu'une connexion forte $F - F$ ait un point C commun. Pour l'algorithme PMIS et HMIS, les connexions $F - F$ sans point C commun sont permises, et dans ce cas les coefficients sont redistribués sur la diagonale. Finalement l'équation s'écrit :

$$(a_{ii} + \sum_{j \in F_i^w} a_{ij})e_i \approx -\sum_{j \in C_i} \left(a_{ij} + \sum_{m \in F_i^s} \frac{a_{im}a_{mj}}{\sum_{k \in C_i} a_{mk}} \right) e_j$$

et les coefficients d'interpolation sont donnés par :

$$\omega_{ij} = -\frac{a_{ij} + \sum_{m \in F_i^s} \left(\frac{a_{im}a_{mj}}{\sum_{k \in C_i} a_{mk}} \right)}{a_{ii} + \sum_{n \in F_i^w} a_{in}}$$

Interpolation F-F

L'interpolation $F - F$ (Sterck and Yang (2004)) étend l'interpolation en utilisant des points C qui ne sont pas connectés à i . L'interpolation classique nécessite que les connexions fortes $F - F$ possèdent un point C commun. L'interpolation $F - F$ se différencie de l'interpolation classique par le traitement d'une connexion forte $F - F$ sans point C commun. Lorsqu'une connexion $F - F$ sans point C commun est rencontrée, l'expression (2.6) est remplacée par :

$$e_j = \sum_{k \in C_j} \frac{a_{jk}e_i}{a_{jk}}$$

On définit alors pour un point F , l'ensemble F_i^{s*} des points F fortement connectés mais sans point C commun, et on redéfinit F_i^w comme l'ensemble des points F fortement connectés à i , mais ayant un point C commun. Pour l'ensemble des points $j \in F_i^{s*}$, l'ensemble C_i est étendu avec C_j pour donner l'ensemble C_i^* . En utilisant une approche identique à celle utilisée pour l'interpolation classique, les coefficients d'interpolation deviennent alors :

$$\omega_{ij} = - \frac{a_{ij} + \sum_{m \in F_i^s \cup F_i^{s*}} \left(\frac{a_{im} a_{mj}}{\sum_{k \in C_i^*} a_{mk}} \right)}{a_{ii} + \sum_{n \in F_i^w} a_{in}}$$

En augmentant le nombre de points C considérés dans l'interpolation, l'interpolation $F - F$ requiert un coût mémoire plus important, et le coût de l'interpolation est augmenté par rapport à une interpolation classique.

Interpolation F-F1

Dans l'algorithme d'interpolation $F - F$, un temps important peut être nécessaire pour prendre en compte les points C distants non directement connectés aux points F . Butler (2006) limite la complexité de l'interpolation $F - F$ en incluant uniquement un point C non connecté à i , lorsque i n'a pas de point C commun avec les points F qui l'influencent.

Interpolation multi-passes

L'interpolation est réalisée en plusieurs passes, en commençant par les points F pour lesquels $C_i^s \neq 0$. Les points F interpolés forment le sous ensemble F^* . Les points F vérifiant $F^* \cap F_i^s \neq 0$ sont interpolés à leurs tours. Certains points peuvent être interpolés avec uniquement un point C . L'algorithme multi-passes est généralement utilisé avec l'algorithme "aggressive coarsening" pour la sélection du maillage.

Annexe 3

Méthodes de préconditionnement ILU

La factorisation incomplète LU est largement utilisée comme préconditionnement pour les méthodes itératives. Elle est basée sur la décomposition LU qui permet de décomposer une matrice A en une matrice triangulaire inférieure L et une matrice triangulaire supérieure U . La décomposition s'effectue par une procédure d'élimination de Gauss relativement coûteuse. La factorisation incomplète LU réduit le temps de calcul en appliquant l'algorithme de factorisation LU en limitant le remplissage à un ensemble d'index S définis par :

$$S = \{(i, j) \mid a_{i,j} \neq 0\}$$

Les entrées des matrices L et U sont nulles pour :

$$l_{i,j} = 0 \quad \text{if } j > i \quad \text{ou} \quad (i, j) \notin S$$

$$u_{i,j} = 0 \quad \text{si } i > j \quad \text{ou} \quad (i, j) \notin S$$

Les entrées possibles non nulles dans L et U sont donc celles qui correspondent aux entrées de A non nulles.

L'algorithme ILU (Algorithme 14) est défini dans sa version IKJ alors que l'algorithme d'élimination de Gauss est souvent présenté dans sa version KIJ. La version KIJ n'est pas adaptée à la structure

Algorithme 14 Algorithme originelle ILU dans sa version IKJ.

Pour $i = 2 \dots n$:

$$u_{i,*} = a_{i,*}$$

Pour $k = 1 \dots i-1$ et si $(i, k) \in S$:

$$\text{Calcul du pivot } l_{i,k} = u_{ik}/u_{kk}$$

Pour $j = k+1 \dots n$ et si $(i, j) \in S$:

$$\text{Calculer } u_{ij} = u_{ij} - l_{ik}u_{kj}$$

Fin boucle k

Fin boucle i

Algorithme 15 Algorithme ILU(p) dans sa version IKJ.

Pour tous les éléments non nul a_{ij} , définir $u_{ij} = a_{ij}$, $lev(u_{ij}) = 0$

Pour $i = 2 \dots N$

Pour $k = 1 \dots i-1$ et si $u_{ik} \neq 0$

$l_{ik} = u_{ik}/u_{kk}$ et $lev(l_{ik}) = lev(u_{i,k})$

$u_{i,*} = u_{i,*} - l_{i,k}u_{k,*}$

Mise à jour du niveau de remplissage selon la Relation (3.7)

Remplacer n'importe quelle valeur de la ligne i avec $lev(u_{ij}) > p$ par 0.

Fin boucle k

Fin Boucle i

de données CSR des matrices creuses. Saad propose une variante nommée Crout (Li et al., 2003) dans laquelle la k^{eme} étape de l'algorithme génère la k^{eme} ligne de U et la k^{eme} colonne de L.

D'autres approches permettent d'augmenter le remplissage par rapport à celui autorisé dans la version présentée dans l'Algorithme (14). Les stratégies d'augmentation du remplissage peuvent être généralement classées en deux catégories :

- ILUT basé sur un seuil
- ILU (l) basé sur la structure.

ILU(k)

L'idée d'utiliser une factorisation ILU plus précise a été proposée par J.A. Meijerink (1976). L'ajout d'une entrée s'effectue selon le critère du niveau de remplissage de cette entrée. Le niveau de remplissage d'un élément est défini récursivement à partir du niveau de remplissage de ces parents dans le processus d'élimination de Gauss. Lorsqu'une entrée dépasse un certain niveau de remplissage, elle est mise à zéro. Le niveau de remplissage d'une entrée f_{ij} nouvellement créée à la position (i,j) par $f_{i,j} = -l_{ik} \times u_{kj}$ est défini par :

$$level(f_{ij}) = level(l_{ik}) + level(u_{kj}) + 1 \quad (3.7)$$

Le niveau 0 correspond aux entrées (i,j) telles que $a_{ij} \neq 0$ et l'Algorithme (14) correspond à ILU(0).

ILU avec seuil : ILUT Saad (1994)

Les préconditionneurs ILU(p) sont basés uniquement sur la structure de la matrice et non les valeurs numériques. Le préconditionnement peut poser problème pour des matrices non M (la factorisation LU d'une matrice M est garantie et calculée de manière stable sans pivot, il en est de même pour la factorisation ILU). Pour les matrices non M, une solution est l'utilisation d'un préconditionneur ILU avec seuil noté ILUT (Saad, 1994). Les entrées sont permises uniquement si elles sont plus grandes qu'une certaine valeur seuil (relativement à la norme de la i ème ligne de L ou U), la valeur diagonale étant toujours conservée. Cet algorithme est décrit dans l'Algorithme (16). Un paramètre supplémentaire l_{fil} permet de contrôler le nombre maximum de valeurs non nulles dans chaque ligne des matrices L et U, qui requièrent alors une capacité de stockage de $2n \times l_{fil}$ valeurs. Ce paramètre

Algorithme 16 Algorithme ILUT.

```
 $\omega = 0$   
Pour  $i = 2 \dots N$   
   $\omega = a_{i,*}$  (copie creuse)  
  Pour  $k = 1 \dots i-1$  et si  $\omega_k \neq 0$   
     $\omega_k = \omega_k / a_{k,k}$   
    Si  $\omega_k \leq \epsilon$  alors  $\omega_k = 0$   
    Si  $\omega_k \neq 0$   
       $\omega = \omega - \omega_k u_{k,*}$   
    Fin si  
  Fin boucle  $k$   
  Pour  $k = 1 \dots N$   
    Si  $\omega_k \leq \epsilon$  alors  $\omega_k = 0$   
  Fin boucle  $k$   
   $l_{i,j} = \omega_j, j = 1 \dots i-1$  (copie creuse)  
   $u_{i,j} = \omega_j, j = i \dots n$  (copie creuse)  
Fin boucle  $i$ 
```

permet de contrôler ainsi les ressources mémoires disponibles. Dans la pratique, le nombre de valeurs effectivement stockées est souvent inférieur à la capacité de stockage $2n \times l_{fil}$. Chen (2004) propose une modification de ILUT en adaptant la valeur l_{fil} pour chaque ligne afin d'utiliser au mieux les ressources mémoires disponibles.

Préconditionnement parallèle ILU

Deux stratégies différentes permettent de paralléliser le preconditionnement ILU :

- la première est le calcul en parallèle de la matrice de preconditionnement.
- la seconde est l'utilisation de la décomposition de domaine de Schwarz qui permet de paralléliser l'ensemble "solveur GMRES + preconditionnement ILU". Cette méthode est présentée dans la section 4.2.3.1.1.

Calcul en parallèle de la matrice de preconditionnement

Hysom and Pothén (1999, 2000) proposent de paralléliser l'algorithme ILU grâce à une numérotation judicieuse des noeuds du graphe de la matrice de manière à pouvoir factoriser un nombre important de lignes localement. La factorisation des lignes restantes s'effectue lorsque le processeur dispose des informations nécessaires provenant des processeurs voisins.

Afin de détailler plus précisément l'algorithme, on définit le graphe $G(A)$ de la matrice, $G(A) = (V, E)$ avec V l'ensemble des noeuds du graphe, qui contient un noeud i pour chaque ligne de la matrice, et E l'ensemble des segments, qui contient un segment (i, j) si $a_{ij} \neq 0$. Un noeud est intérieur si pour l'ensemble des segments (i, j) de V , les noeuds i et j appartiennent au même sous domaine. Si il existe au moins un segment (i, j) tel que les noeuds i et j n'appartiennent pas au même sous domaine, alors le noeud i est un noeud frontière.

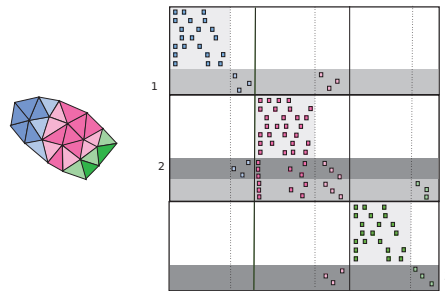


FIGURE 3.13 – Parallélisation de ILU selon Hysom and Pothen (1999).

On définit le graphe réduit $S(A)$ de la matrice A , tel que les noeuds du graphe correspondent aux sous domaines, et un segment (r,s) du graphe existe si les sous domaines r et s sont connectés au moins par un segment dans le graphe $G(A)$.

La première étape de l’algorithme est d’ordonner les noeuds du graphe réduit. Le classement des noeuds du graphe réduit s’effectue avec l’objectif de réduire la longueur des chemins du graphe.

Dans la seconde étape, les noeuds sont ordonnés de façon continue dans chaque sous domaine, en commençant par les noeuds intérieurs puis les noeuds frontières. Puisque la factorisation ILU doit s’effectuer avec un algorithme “ligne par ligne descendant”, la numérotation locale des noeuds intérieurs permet de factoriser les lignes intérieures indépendamment des mises à jour des valeurs dans les lignes appartenant aux autres sous domaines.

Durant la factorisation, la structure creuse de la matrice factorisée peut changer dynamiquement. Lors de la factorisation des noeuds intérieurs, il y a création de nouveaux remplissages qui ajoutent des nouvelles dépendances entre les noeuds des interfaces. Hymson et Pothen imposent aux graphes réduits de la matrice A et de la matrice $F = L + U - I$ d’être identiques. Cette contrainte permet de définir toute les communications nécessaires à la factorisation avant même de la réaliser.

Par exemple, dans la Figure (3.13), un maillage est partitionné en trois sous domaines, correspondant aux trois couleurs (bleu, rose, vert) et les cellules correspondant aux noeuds frontières du graphe sont affichées avec une couleur plus claire. La structure de la matrice A réordonnée est affichée. Chaque processeur peut factoriser les noeuds intérieurs. Cette étape est locale et ne nécessite aucune communication. Ensuite les inconnues correspondant aux noeuds intérieurs sont éliminées des lignes correspondant aux noeuds des interfaces. Le processeur 0 (bleu) peut alors factoriser ses noeuds frontières et le processeur 1 (rose) peut factoriser ses noeuds frontières situés à l’interface avec le processeur 2 (vert). Par contre, il doit attendre que le processeur 1 finisse de factoriser ses noeuds frontières, et le processeur 3 doit attendre que le processeur 2 factorise ses noeuds frontières situés à l’interface avec le processeur 1.

Algorithme 17 Algorithme Parallèle ILU.

1. *Construire les graphes des sous-domaines et renuméroter les noeuds pour réduire*
2. *Ordonner sur chaque processeur les noeuds intérieurs puis les noeuds frontières*
3. *Factoriser les lignes intérieurs.*

Les lignes frontières i qui n'ont pas de voisins avec une numérotation inférieur peuvent être factorisée. Les processeurs qui n'ont pas de voisins dans le graphe des sous domaines peuvent aussi factoriser les noeuds frontières, et vont à l'étape 6.

4. *Reçoit la structure des lignes et le valeurs des lignes frontières de la part des processeurs voisins de rang inférieur.*

5. *Factorise les noeuds frontières.*

6. *Envoi la structure et les valeurs des lignes frontières au processeur voisins ayant un rang supérieur.*

Bibliographie

- Acharya, S. ; Moukalled, F. H. (1989). Improvements to incompressible flow calculation on a nonstaggered curvilinear grid. *Numerical Heat Transfer, Part B : Fundamentals*, 2 :131–152.
- Angot, P. (2007). Vector and scalar penalty-projection methods for incompressible and variable density flows. In *Numerical Flow Models for Controlled Fusion*.
- Aubry, R., Mut, F., Löhner, R., and Cebal, J. R. (2008). Deflated preconditioned conjugate gradient solvers for the pressure-poisson equation. *J. Comput. Phys.*, 227(24) :10196–10208.
- Baker, A. H., Jessup, E. R., and Manteuffel, T. (2005). A technique for accelerating the convergence of restarted gmres. *SIAM J. Matrix Anal. Appl.*, 26(4) :962–98–.
- Bassi, F., Crivellini, A., Pietro, D. A. D., and Rebay, S. (2007). An implicit high-order discontinuous Galerkin method for steady and unsteady incompressible flows. *Comput Fluids*, 36(10) :1529–1546.
- Bassi, F. and Rebay, S. (1997). High-order accurate discontinuous finite element solution of the 2d Euler equations. *J. Comput. Phys.*, 138(2) :251–285.
- Bijl, H. and Wesseling, P. (1998). A unified method for computing incompressible and compressible flows in boundary-fitted coordinates. *J. Comput. Phys.*, 141(2) :153–173.
- Björck, Å. (1967a). Iterative refinement of linear least squares solutions I. *BIT*, 7 :257–278.
- Björck, Å. (1967b). Solving linear least squares problems by Gram-Schmidt orthogonalization. *BIT*, 7 :1–21.
- Botella, . and Peyret, R. (1998). Benchmark spectral results on the lid-driven cavity flow. *Comput Fluids*, 27 :421–433.
- Briggs, G. (1973). Diffusion estimation for small emissions. preliminary repor. Technical report, National Oceanic and Atmospheric Administration, Oak Ridge, Tenn. (USA). Atmospheric Turbulence and Diffusion Lab.
- Briggs, W. L., Henson, V. E., and McCormick, S. F. (2000). *A multigrid tutorial : second edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.
- Brown, D. L., Cortez, R., and Minion, M. L. (2001). Accurate projection methods for the incompressible navier—stokes equations. *J. Comput. Phys.*, 168(2) :464–499.

- Butler, J. S. (2006). *Improving Coarsening and Interpolation for Algebraic Multigrid*. PhD thesis, University of Waterloo.
- Cadafalch, J., Perez-Segarra, C., Consul, R., and Oliva, A. (2002). Verification of finite volume computations on steady state fluid flow and heat transfer. *J. Fluids Engineering*, 124 :11–21.
- Cai, X.-C. and Sarkis, M. (1999). A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.*, 21 :792–797.
- Campbell, J. C., Hyman, J. M., and Shashkov, M. J. (2002). Mimetic finite difference operators for second-order tensors on unstructured grids. *Computers & Mathematics with Applications*, 44(1-2) :157–173.
- Carpentier, R. (2000). High accuracy inviscid scheme on quadrangular structured meshes for compressible flow without viscosity. Technical report, INRIA.
- Carruthers, D. J., Weng, W. S., Dyster, S. J., Singles, R., and Higson, H. (2009). Complex terrain module. Technical report, CERC.
- C. Debiez, Dervieux, A., Mer, K., and Nkonga, B. (1998). Computation of unsteady flows with mixed finite volume/finite element upwind methods. *Int. J. Numer. Method Fluid*, 27 :193–206.
- CERC (2001). Adms user guide. Technical report, Cambridge Environmental Research Consultants.
- Chan, A. T., So, E. S. P., and Samad, S. C. (2001). Strategic guidelines for street canyon geometry to achieve sustainable street air quality. *Atmospheric Environment*, 35(32) :5681 – 5691.
- Chen, T.-Y. (2004). *Computational Science and Its Applications - ICCSA 2004*, chapter ILUTP_Mem : A Space-Efficient Incomplete LU Preconditioner, pages 20–28.
- Choi, H.-I., Lee, D., and Maeng, J.-S. (2003). a node-centered pressure-based method for all speed flows on unstructured grids. *Numerical Heat Transfer, Part B*, 44 :165–185.
- Choi, S. K. (1999). Note on the use of momentum interpolation for unsteady flows. *Numerical Heat Transfer, Part A : Applications*, 36 :545–550.
- Chorin, A. (1967). A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys.*, 2 :12.
- Chorin, A. (1968). The numerical solution of the navier stokes. equation for an incompressible fluid. *Mathematics of Computation*, 22 :745–762.
- Chu, A., Kwok, R., and Yu, K. (2005). Study of pollution dispersion in urban areas using computational fluid dynamics (cfd) and geographic information system (gis). *Environmental Modelling & Software*, 20(3) :273 – 277.
- Cimorelli, A. J., Perry, S. G., Venkatram, A., Weil, J. C., Paine, R. J., Wilson, R. B., Lee, R. F., Peters, W. D., Brode, R. W., , and Paumier, J. O. (2004). Description of model formulation. Technical report, U S. Environmental Protection Agency.

- Cleary, A. J., Falgout, R. D., Henson, V. E., and Jones, J. E. (1998). Coarse-grid selection for parallel algebraic multigrid. Technical report.
- Cockburn, B., Kanschat, G., and Schötzau, D. (2005). The local discontinuous Galerkin method for linearized incompressible fluid flow : a review. *Comput Fluids*, 34(4-5) :491–506.
- Coirier, W. (1994). *An adaptively-refined, Cartesian, cell-based scheme for the Euler and Navier-Stokes equations*. PhD thesis, University of Michigan.
- Coirier, W., Fricker, D., Furmanczyk, M., and Kim, S. (2006). A computational fluid dynamics approach for urban area transport and dispersion modeling. *Env. Fluid Dynamics*, 5 :443–479.
- Courant, R., Friedrichs, K., and Lewy, H. (1967). On the partial difference equations of mathematical physics. *IBM Journal*, pages 215–234.
- Cuthill, E. and McKee, J. (1969). Reducing the bandwidth of sparse symmetric matrices. In *ACM '69 : Proceedings of the 1969 24th national conference*, pages 157–172, New York, NY, USA. ACM.
- Darmofal, D. L. and Schmid, P. J. (1996). The importance of eigenvectors for local preconditioners of the euler equations. *J. Comput. Phys.*, 127 :346–362.
- Demaël, E. (2007). *Modélisation de la dispersion atmosphérique en milieu complexe et incertitudes associées*. PhD thesis, Ecole Nationale des Ponts et Chaussées.
- Dijk, A. V. and de Lange, H. (2006). Compressible laminar flow around a wall-mounted cubic obstacle. *Comput Fluids*.
- Doormaal, J. P. V. and Raithby, G. D. (1984). Enhancements of the simple method for predicting incompressible fluid flows. *Numerical Heat Transfer, Part A : Applications*, 7 :147–163.
- Doury, A. (1976). Une méthode de calcul pratique et générale pour la prévision des pollutions véhiculées par l’atmosphère. Technical report, CEA.
- Doury, A. (1984). Une méthode d’approximation instantanée de la dispersion atmosphérique. Technical report, DAS n 52-CEA/IPSN.
- Dumbser, M. (2010). Arbitrary high orderPNPM schemes on unstructured meshes for the compressible Navier-Stokes equations. *Comput Fluids*, 39(1) :60–76.
- Durrant, D. R. (1989). Improving the anelastic approximation. *Journal of the Atmospheric Sciences*, 46(11) :1453–1461.
- Dutton, J. A. and Fichtl, G. H. (1969). Approximate equations of motion for gases and liquids. *Journal of the Atmospheric Sciences*, 26(2) :241–254.
- Dyunkerke, P. (1988). Application of the E-epsilon turbulence closure model to the neutral and stable atmosphere boundary layer. *J. Atmos. Sci*, 45 :5, 865–880.
- Eiermann, M., Ernst, O. G., and Schneider, O. (2000). Analysis of acceleration strategies for restarted minimal residual methods. *J. Comput. Appl. Math.*, 123(1-2) :261–292.

- Erlebacher, G. (1985). *The Free-Lagrange Method*, chapter Finite difference operators on unstructured triangular meshes, pages 22–53. Springer Berlin.
- Fernandez, F. and Kulas, L. (2004). A simple finite difference approach using unstructured meshes from fem mesh generators. In *Microwaves, Radar and Wireless Communications*, volume 2, pages 585–588.
- Ferziger, J. and Peric, M. (2002). *Computational methods for fluid dynamics*. Springer, 3 edition.
- Franke, J., Hirsch, C., Jensen, A., Krus, H., Schatzmann, M., Westbury, P., Miles, S., Wisse, J., and Wright, N. (2004). Recommendations on the use of cfd in wind engineering. In J.P.A.J., V. K. I., editor, *Proceedings of the International Conference Urban Wind Engineering and Building Aerodynamics*. van Beeck.
- Gear, C. (1971). *Numerical initial value problems for ordinary differential equation*. Prentice-Hall automatic computation series.
- Ghosal, S. (1995). Analysis of discretization errors in les. Technical report, Center for turbulence Research.
- Ghosal, S. (1996). An analysis of numerical errors in large eddy simulations of turbulence. *J. Comput. Phys.*, 125 :187–206.
- Golub and Loan, V. (1996). *Matrix Computations*.
- Gough, D. O. (1969). The anelastic approximation for thermal convection. *Journal of the Atmospheric Sciences*, 26(3) :448–456.
- Guermond, J., Mineev, P., and Shen, J. (2006). An overview of projection methods for incompressible flows. *Comput. Methods Appl. Mech. Engrg*, 195(44-47) :6011 – 6045.
- Guermond, J. L. and Shen, J. (2003). A new class of truly consistent splitting schemes for incompressible flows. *J. Comput. Phys.*, 192(1) :262–276.
- Guibert, D. (2009). *Analyse de méthodes de résolution parallèles d’EDO / EDA raides*. PhD thesis, Université Claude Bernard Lyon 1.
- H. Guillard, C. V. (1999). On the behaviour of upwind schemes in the low mach number limit. *Comput Fluids*, 28 :63–86.
- Hanjalic, K. and Kenjeres, S. (2008). Some developments in turbulence modeling for wind and environmental engineering. *J. Wind Eng. Ind. Aerodyn*, 96 :537–1570.
- Hanna, S. R., Hansen, O. R., and Dharmavaram, S. (2004). Flacs cfd air quality model performance evaluation with kit fox, must, prairie grass, and emu observations. *Atmospheric Environment*, 38(28) :4675 – 4687.
- Harlow, F. H. (1965). Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids*, 8 :2182–2189.

- Harlow, F. H. and Amsden, A. A. (1968). Numerical calculation of almost incompressible flow. *Journal of Computational Physics*, 3(1) :80 – 93.
- Henson, V. E. and Yang, U. M. (2000). Boomeramg : a parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41 :155–177.
- Hirsh, R. S. (1975). Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique. *J. Comput. Phys.*, 19 :90–109.
- Holmes, D. and Connell, S. (1989). Solution of the 2d Navier Stokes equation on unstructured adaptive grids. In *AIIAA 9th CFD conference*, volume Paper 89.
- Hong, H., Ryoza, O., Shinsuke, K., Hong, C., Takeo, T., and Takeaki, W. (2006). Cfd analysis on traffic-induced air pollutant dispersion with non isothermal condition in a complex urban area in winter. *JWE*, (108) :841–844.
- Hysom, D. and Pothen, A. (1999). Efficient parallel computation of ilu(k) preconditioners.
- Hysom, D. and Pothen, A. (2000). Parallel ilu ordering and convergence relationships : Numerical experiments.
- I. Keshtiban, F. Belblidia, M. W. (2004). Compressible flow solvers for low mach number flows - a review. Technical report, Institute of non-Newtonian fluid mechanics, Department of Computer Science, University of Wales, Swansea.
- Islitzer, N. and Dumbauld, K. (1963). Diffusion and transport experiments. pages 117–188.
- Issa, R. (1986). Solution of the implicitly discretised reacting flow equations by operator-splitting. *J. Comput. Phys.*, 62 :40–65.
- Ivan, L. and Groth, C. (2007). High-order central eno finite-volume scheme with adaptive mesh refinement. *AIAA*.
- J.A. Meijerink, H. v. d. V. (1976.). An iterative solution method for linear systems of which the coefficient matrix is a symmetric m-matrix. Technical report, Academisch Computer Centrum, Utrecht,.
- Jameson, A. and Alonso, J. (1991). Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *AIAA*, page 1596.
- Jan S.Hesthaven, T. (2008). *Nodal Discontinuous Galerkin Methods : Algorithms, Analysis, and Applications*. Springer.
- Jeong.Jinhee and Hussain.Fazle (1995). On the identification of a vortex. *J. Fluid Mech*, 285 :69–94.
- Jicha, M., Pospisil, J., and Katolicky, J. (2000). Dispersion of pollutants in street canyon under traffic induced flow and turbulence. *Environmental Monitoring and Assessment*, 65(1) :343–351.
- J.W. Ruge, K. S. (1987). *Algebraic multigrid (AMG)*. Frontiers in Applied Mathematics.

- Karki, K. and Patankar, S. (1989). Pressure based calculation procedure for viscous flows at all speeds in arbitrary configurations. *AIAA J*, 27 :1167–1174.
- Karypis, G. and Kumar, V. (1997). Parallel threshold-based ilu factorization. In *Supercomputing '97 : Proceedings of the 1997 ACM/IEEE conference on Supercomputing (CDROM)*, pages 1–24, New York, NY, USA. ACM.
- Kato, M. and Launder, B. E. (1993). The modeling of turbulent flow around stationary and vibrating square cylinders [c]// proceedings of the 9th kyoto, 1-6.
- katopodes Chow, F. and Moin, P. (2003). A further study of numerical errors in large eddy simulations. *J. Comput. Phys.*, 184 :366–380.
- Khosla, P. and Rubin, S. (1974). A diagonally dominant second-order accurate implicit scheme. *Comput Fluids*, 2 :207–209.
- Klein, R. (1995). Semi-implicit extension of a godunov-type scheme based on low mach number asymptotics i : one-dimensional flow. *J. Comput. Phys.*, 121 :213–237.
- Lacor, C., Smirnov, S., and Baelmans, M. (2004). A finite volume formulation of compact central schemes on arbitrary structured grids. *J. Comput. Phys.*, 535-566 :198.
- Landmann, B., Kessler, M., Wagner, S., and Krämer, E. (2008). A parallel, high-order discontinuous Galerkin code for laminar and turbulent flows. *Comput Fluids*, 37(4) :427–438. Turbulent Flow and Noise Generation.
- Lantz, S. R. and Fan, Y. (1999). Anelastic magnetohydrodynamic equations for modeling solar and stellar convection zones. *The Astrophysical Journal Supplement Series*, 121(1) :247–264.
- Launder, B. and Spalding, D. (1972). *Lectures in mathematical models of turbulence*. Academic Press, London, England,.
- Launder, B. and Spalding, D. (1974). The numerical computation of turbulent flows. *Comput Fluids*, 3 :269–289.
- Lawrence Livermore National Laboratory, L. (2006). *Hypre high performance preconditioner, user's manual*. Center for Applied Scientific Computing.
- Lele, S. K. (1992). Compact finite difference schemes with spectral-like resolution. *J. Comput. Phys.*, 103 :16–42.
- Li, N., Saad, Y., and Chow, E. (2003). Crout versions of ILU for general sparse matrices. *SIAM J. Sci. Comput.*, 25(2) :716–728.
- Lilly, D. (1992). A proposed modification of the germano subgrid-scale closure model. *Phys. Fluids*, 4 :633–635.
- Lipps, F. B. and Hemler, R. S. (1982). A scale analysis of deep moist convection and some related numerical calculations. *Journal of the Atmospheric Sciences*, 39(10) :2192–2210.

- Liu, M., Ren, Y.-X., and Zhang, H. (2004). A class of fully second order accurate projection methods for solving the incompressible navier-stokes equations. *J. Comput. Phys.*, 200(1) :325–346.
- MacLachlan, S. P., Tang, J. M., and Vuik, C. (2008). Fast and robust solvers for pressure-correction in bubbly flow problems. *J. Comput. Phys.*, 227(23) :9742–9761.
- Maday, Y. and Turinici, G. (2007). The parareal in time iterative solver : a further direction to parallel implementation. *Wiley*, 55 :22 :163–175.
- Majumdar, S. (1988). Role of underrelaxation in momentum interpolation for calculations of flows with nonstaggered grids. *Numerical Heat Transfer, Part A : Applications*, 13 :125 – 132.
- Mandal, J., S.Rao, and J.Subramanian (2008). High-resolution finite volume computations using a novel weighted least-squares formulation. *Int. J. Numer. Method Fluid*, 56 :1425–1431.
- Mazzoldi, A., Hill, T., and Colls, J. J. (2008). Cfd and gaussian atmospheric dispersion models : A comparison for leak from carbon dioxide transportation and storage facilities. *Atmospheric Environment*, 42(34) :8046 – 8054.
- Meinke, M., Schroder, W., Krause, E., and Rister, T. (2002). A comparison of second and sixth order methods for large eddy simulations. *Comput Fluids*, 31 :695–718.
- M.Germano, Piomelli, U., Moin, P., and Cabot, W. (1991). A dynamic subgrid scale eddy viscosity model. *Phys. Fluids*, 3 :1760–1765.
- Montagnier, J., Buffat, M., and Guibert, D. (2008). Parallel computation of pollutant dispersion in industrial sites. In Verlag, editor, *Parallel Computational Fluid Dynamics 2008*.
- Moukalled, F. and Darwish, M. (2000). A unified formulation of the segregated class of algorithms for multi-fluid flow at all-speeds. *Numerical Heat Transfer, Part B : Fundamentals*, 37 :103–139.
- M.Piller and E.Stalio (2008). Compact finite volume schemes on boundary-fitted grids. *J. Comput. Phys.*, 227 :4736–4762.
- Müller, B. (1999). Low mach number asymptotics of the navier-stokes equations and numerical implications.
- Munikrishna, N. (2007). *On viscous flux discretization procedures for finite volume and meshless solvers*. PhD thesis, Department of Aerospace Engineering Indian Institute of Science.
- Munz, C. D., Roller, S., Klein, R., and Geratz, K. J. (2003). The extension of incompressible flow solvers to the weakly compressible regime. *Comput Fluids*, 32(2) :173 – 196.
- Nakayama, A. and Vengadesan, S. (2002). On the influence of numerical schemes and subgrid-stress models on large eddy simulation of turbulent flow past a square cylinder. *Int. J. Numer. Method Fluid*, 38 :227–253.
- Nejat, A. and Ollivier-Gooch, C. (2008). A high-order accurate unstructured finite volume Newton-Krylov algorithm for inviscid compressible flows. *J. Comput. Phys.*, 227 :2582–2609.

- Nerinckx, K., Vierendeels, J., and Dick, E. (2005). Mach-uniformity through the coupled pressure and temperature correction algorithm. *J. Comput. Phys.*, 206 :597–623.
- Nicolaidis, R. A. (1987). Deflation of conjugate gradients with applications to boundary value problems. *SIAM J. Numer. Anal.*, 24(2) :355–365.
- Noma Park, K. M. (2007). Analysis of numerical errors in large eddy simulation using statistical closure theory. *J. Comput. Phys.*, 222 :194–216.
- Ogura, Y. and Phillips, N. A. (1962). Scale analysis of deep and shallow convection in the atmosphere. *Journal of the Atmospheric Sciences*, 19(2) :173–179.
- Ollivier-Gooch and Altena, M. V. (2002). A high order accurate unstructured mesh finite volume for the advection-diffusion equation. *J. Comput. Phys.*, 181 :729–752.
- Paik, J., Sotiropoulos, F., and Portè-Agel, F. (2009). Detached eddy simulation of flow around two wall-mounted cubes in tandem. *Int. J. Heat Fluid Flow*, 3 :286–305.
- Park, N., Yoo, J. Y., and Choi, H. (2004). Discretizations errors in large eddy simulation : on the suitability of centered and upwind-biased compact difference schemes. *J. Comput. Phys.*, 198 :580–616.
- Pasquill, F. (1962). *Atmospheric diffusion; the dispersion of windborne material from industrial and other sources*. Van Nostrand London, Princeton, N.J.,.
- Patankar, S. and Spalding, D. (1972a). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. J. Heat Mass Transfer*, 15 :1787.
- Patankar, S. and Spalding, D. (1972b). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. J. Heat Mass Transfer*, 15 :1787.
- Pereira, J. M. C., Kobayashi, M. H., and Pereira, J. C. F. (2001). A fourth-order-accurate finite volume compact method for the incompressible Navier-Stokes solutions. *J. Comput. Phys.*, 167 :217–243.
- Perkins, R., Soulhac, L., Mejean, P., and Rios, I. (2005). Modélisation de la dispersion des émissions atmosphériques d’un site industriel : vers un guide de l’utilisateur. Technical report, LMFA - Ecole Centrale de Lyon.
- Ravikumar, D. (2001). *2D compressible viscous flow computations using acoustic flux vector splitting (AFVS) scheme*. PhD thesis, Department of Aerospace Engineering Indian Institute of Science.
- Rhie, C. M. and Chow, W. L. (1983). Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA*, 21 :1525–1532.
- Robins, A. and Apsley, D. (2009). Modelling of building effects in adms. Technical report, University of Surrey ,National Power and CERC.
- Robins, A., Apsley, D., Carruthers, D., McHugh, C., and Dyster, S. (2009). Plume rise model specification. Technical report, University of Surrey ,National Power and CERC.

- Roller, S. and Munz, C.-D. (2000). A low mach number scheme based on multi-scale asymptotics. *Computing and Visualization in Science*, 3(1) :85–91.
- S. Roller, C. M. (2000). A low mach number scheme based on multi-scale asymptotics. *Comput. and visualization in science* 3, 3 :85–91.
- Saad, Y. (1993). A flexible inner-outer preconditioned gmres algorithm. *SIAM J. Sci. Comput.*, 14(2) :461–469.
- Saad, Y. (1994). Ilut : A dual threshold incomplete lu factorization. *Numer. Linear Algebra Appl.*, 1 :387–402.
- Saad, Y. (2000). *Iterative methods for sparse linear systems*.
- Saad, Y. and Schultz, M. H. (1986). GMRES : a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3) :856–869.
- Saad, Y. A. M. (1995). P-sparslib : a portable library of distributed memory sparse iterative solvers. Technical report, UMSI/95 180, University of Minesota.
- Sabatino, S., Buccolieri, R., Pulvirenti, B., and Britter, R. (2008). Flow and pollutant dispersion in street canyons using fluent and adms-urban. *Environmental Modeling and Assessment*, 13(3) :369–381.
- Schmidt, S. and Thiele, F. (2002). Comparison of numerical methods applied to the flow over wall-mounted cube. *Int. J. Heat Fluid Flow*, 23 :330–339.
- Schubert, G., Turcotte, D., and Olson, P. (2000). *Mantle Convection in the Earth and Planets*. Cambridge University Press.
- Shen, Y., Yang, G., and Gao, Z. (2006). High-resolution finite compact difference schemes for hyperbolic conservation laws. *J. Comput. Phys.*, 216(1) :114–137.
- Smagorinsky, J. (1963). General circulation experiments with the primitive equations. i. the basic experimen. *Mon. Weather Rev.*, 91 :99–152.
- Spalart, P. (2000). Strategies for turbulence modelling and simulations. *Int. J. Numer. Method Fluid*, 21 :252–263.
- Spalart, P., Jou, W.-H., Strelets, M., and Allmaras, S. (1997). Comments on the feasibility of les for wings, and on a hybrid rans/les approach. In *First AFOSR International Conference on DNS/LES*, C. Liu, LA.
- Spalart, P. R. (2006). chapter Topics in Detached-Eddy Simulation, pages 3–12.
- Spalart, P. R. (2009). Detached-eddy simulation. *Annual Review of Fluid Mechanics*, 41 :181–202.
- Spalart, P. R. and Allmaras, S. R. (1992). A one-equation turbulence model for aerodynamic flows. *AIAA*, 92 :5–21.

- Sterck, H. D. and Yang, U. M. (2004). Coarsening and interpolation in algebraic multigrid : a balancing act. In *Copper Mountain Conference on Multigrid Methods*.
- Sterck, H. D., Yang, U. M., and Heys, J. J. (2006). Reducing complexity in parallel algebraic multigrid preconditioners. *SIAM Journal on Matrix Analysis and Applications*, 27(4) :1019–1039.
- Stuben, K. (1999). Algebraic multigrid (amg) : An introduction with applications. Technical report, GMD Report 70.
- Tackley, P. J. (2008). Modelling compressible mantle convection with large viscosity contrasts in a three-dimensional spherical shell using the yin-yang grid. *Physics of the Earth and Planetary Interiors*, 171 :7–18.
- Tao, W. Q.; Qu, Z. G. H. Y. L. (2004). a novel segregated algorithm for incompressible fluid flow and heat transfer problems - clear (coupled and linked equations algorithm revised) part i : Mathematical formulation and solution procedure. *Numerical Heat Transfer, Part B : Fundamentals*, 1 :19–48.
- Témam, R. (1969). Sur l'approximation de la solution des equations de navier-stokes par la méthode des pas fractionnaires (ii). *Archive for Rational Mechanics and Analysis*, 33(5) :377–385.
- Tennekes, H. and Lumley, J. (1972). *A first Course in Turbulence*.
- Toledo, S. (1997). Improving memory-system performance of sparse matrix-vector multiplication. In *IBM Journal of Research and Development*.
- Traoré, P., Ahipo, Y. M., and Louste, C. (2009). A robust and efficient finite volume scheme for the discretization of diffusive flux on extremely skewed meshes in complex geometries. *J. Comput. Phys.*, 228 :5148–5159.
- Turkel, E. (1987). Preconditioned methods for solving the incompressible and low speed compressible equations. *J. Comput. Phys.*, 72 :277–98.
- Turner, D. B. (1997). The long lifetime of the dispersion methods of pasquill in u.s regulatory air modeling. *Journal of Applied Meteorology*, 36 :1016–1020.
- Vaassen, J.-M., D.Vigneron, and J-A.Essers (2006). An implicit high order finite volume scheme for the solution of 3d Navier Stokes equations with new discretization of diffusive terms. *J. of Comp. and Appl. Math.*, 215 issue2 :595–601.
- Van Driest, E. (1956). On the turbulent flow near a wall. *J. Aero. Sc.*, 23 :1007–1011.
- Van Leer, B., Lee, W. T., and Roe, L. (1991). Characteristic time-stepping or local preconditioning of the euler equations. *AIAA journal*,, 91.
- Vengadesan, S. and Nithiarasu, P. (2007). Hybrid les - review and assessment. *Sadhana*, 32 :501–511.
- Vreman, B., Geurts, B., and Kuerten, H. (1996). Comparison of numerical schemes in large eddy simulation of the temporal mixing layer. *Int. J. Numer. Method Fluid*, 22 :297–311.

- Wang, X. and McNamara, K. (2006). Evaluation of cfd simulation using rans turbulence models for building effects on pollutant dispersion. *Environmental Fluid Mechanics*, 6(2) :181–202.
- Weller, H. and Weller, H. (2008). A high-order arbitrarily unstructured finite-volume model of the global atmosphere : Tests solving the shallow-water equations. *Int. J. Numer. Method Fluid*, 56 :1589–1596.
- Wesseling, P. (1995). Von neumann stability conditions for the convection-diffusion equation.
- Wilhelmson, R. and Ogura, Y. (1972). The pressure perturbation and the numerical modeling of a cloud. *Journal of the Atmospheric Sciences*, 29(7) :1295–1307.
- Wong, J. S., Darmofal, D. L., and Peraire, J. (2001). The solution of the compressible euler equations at low mach numbers using a stabilized finite element algorithm. *Computer Methods in Applied Mechanics and Engineering*, 190 :5719–5737.
- Yakhot, V., Orszag, S. A., Thangam, S., Gatski, T. B., and Speziale, C. G. (1992). Development of turbulence models for shear flows by a double expansion technique. *Physics of Fluids A : Fluid Dynamics*, 4(7) :1510–1520.