POLYPUBLIE

Méthodes pour la vérification d'un code de mécanique des fluides

Thomas Reid

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

Date	2007
Type	Mémoire ou thèse / Dissertation or Thesis Mémoire de maîtrise/Master's thesis
URL de PolyPublie PolyPublie URL	https://publications.polymtl.ca/7946/
Conditions d'utilisation Terms of use	© Tous droits réservés / All rights reserved
Référence Citation	Reid, T. (2007). Méthodes pour la vérification d'un code de mécanique des fluides [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7946/

Détails de la thèse
Thesis details

Directeurs de recherche Advisors	Jean-Yves Trépanier
Programme Program	Non spécifié

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal
https://publications.polymtl.ca

UNIVERSITÉ DE MONTRÉAL

MÉTHODES POUR LA VÉRIFICATION D'UN CODE DE MÉCANIQUE DES FLUIDES

THOMAS REID
DÉPARTEMENT DE DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE MÉCANIQUE)
FÉVRIER 2007

© Thomas Reid, 2007.

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-29253-2
Our file Notre référence
ISBN: 978-0-494-29253-2

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

MÉTHODES POUR LA VÉRIFICATION D'UN CODE DE MÉCANIQUE DES FLUIDES

présenté par: REID Thomas

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. PELLETIER Dominique, Ph.D., président

M. TRÉPANIER Jean-Yves, Ph.D., membre et directeur de recherche

M. LAFOREST Marc, Ph.D., membre

Si tu comprends, les choses sont comme elles sont.

Si tu ne comprends pas, les choses sont comme elles sont.

Proverbe zen

REMERCIEMENTS

Je tiens à remercier tous ceux qui m'appuyèrent de près ou de loin durant mes travaux de maîtrise. Je remercie Biothermica Technologies pour le support financier qui me permit d'obtenir mon diplôme de maîtrise, et ce dans un cadre industriel et stimulant. Merci à Simon Painchaud-Ouellet pour son encadrement juste et rigoureux, pour les conseils et les échanges fréquents dont j'ai profité tout au long de mes travaux de recherche. Je tiens aussi à souligner l'apport important sur ma compréhesion de CFDLIB et des algorithmes de CFD de que m'ont apporté les échanges avec Lamine Kalla. Je remercie également Guy Drouin, Louis Béchard et tous les gens de Biothermica qui contribuèrent à enrichir mon expérience et à situer mes travaux dans un contexte plus large. De plus, je remercie Christophe Tribes pour son écoute, son support et ses suggestions, ainsi que pour son aide précieuse durant la phase de rédaction. Merci à Jean-Yves Trépanier, pour sa supervision sérieuse et pour sa disponibilité. Je remercie également Dominique Pelletier pour son aide précieuse au début du projet de vérification de CFDLIB, pour les échanges et les suggestions. Merci à Barnim Janta, Simon Bernard ainsi qu'à tous mes collègues du Luminos, avec qui j'ai eu la chance de travailler et de partager des connaissances durant ma maîtrise. Enfin, je voudrais remercier mon père et ma conjointe pour leur appui et leur support déterminant quant au succès de mes études.

RÉSUMÉ

Ce mémoire porte sur la mise à l'épreuve de deux méthodes de vérification de code : la comparaison avec des solutions d'équations différentielles ordinaires (EDO) reconnues et la méthode de comparaison de codes. Ces méthodes s'appliquent à des codes utilisant des algorithmes d'éléments finis, de volumes finis ou de différences finies. Elles visent à tester un programme de mécanique des fluides avec des problèmes plus complexes que ceux dont on connaît la solution analytique. De plus, elles peuvent être effectuées dans un contexte où la modification du système d'équations par l'ajout de termes sources n'est pas possible.

Pour les deux méthodes, la solution exacte des problèmes étudiés est inconnue. On utilise donc des méthodes d'estimation d'incertitude pour comparer les résultats. Ces estimations sont faites au moyen d'études de convergence de la discrétisation. L'emphase est mise sur deux variantes de la technique du *grid convergence index* (GCI) : la méthode des moindres carrés, qui utilise une courbe de régression sur les résultats obtenus avec différents maillages, et la méthode de Cadafalch et al., qui consiste à appliquer le GCI localement aux noeuds où l'extrapolation de Richardson est applicable. L'utilisation combinée de ces deux variantes permet d'analyser des résultats à la fois sur leur comportement global et local.

L'exemple de comparaison avec une solution d'EDO reconnue est effectué avec le problème du jet laminaire plan. Les résultats numériques sont comparés à la solution de similitude, déterminée analytiquement à partir des équations de la couche limite. Le problème principal de cette approche est que la solution numérique et la solution de similitude s'appliquent à deux systèmes d'équations différents. Les comparaisons de résultats ne répondent donc pas aux besoins d'une étude de vérification rigoureuse.

La méthode de comparaison de codes (MCC) est appliquée sur un écoulement circulatoire incompressible (ECI). Dans un premier temps, le programme utilisé pour obtenir la solution de référence est testé au moyen de la méthode des solutions manufacturées,

avec une solution semblable à celle de l'ECI. Par la suite, le calcul de l'ECI est effectué sur les deux programmes. Les résultats sont comparés en tenant compte des incertitudes sur les calculs numériques.

La MCC s'est révélée plus utile que la comparaison avec des solutions reconnues d'EDO, car elle permet des comparaisons quantitatives des résultats. Toutefois, la MCC est fortement limitée par le comportement du programme de référence.

ABSTRACT

This work presents the evaluation of two methods for code verification : the comparison with benchmark ordinary differential equations solutions and the method of code comparison. These methods are suitable for FEM, FVM or FDM codes. They are meant to test a CFD software on complex flows, which cannot be done using only analytical solutions. Furthermore, the PDE model doesn't have to be modified with source terms, contrary to the method of manufactured solutions.

However, for the two methods, the exact solution of the analyzed problems remains unknown. This makes the uncertainties estimations necessary for the results comparison. For this purpose, we use principally two variants of the grid convergence index (GCI) : the least-square method, which consists in the calculation of a regression curve on the grid convergence results, and the analysis of Cadafalch et al., which focuses on the convergence of the results on each node. The use of these two approaches ensure a complete analysis, both from a local and a global perspective.

The comparison with a benchmark ODE is applied with the laminar jet problem. The numerical results are compared with the similarity solution, which can be found analytically using the boundary-layer equations. However, the method is inadequate for code verification purposes, because the solutions do not come from the same equation system.

The code comparison method is applied to an incompressible rotating flow (IRF). First, a similar flow is verified on a reference program, using the manufactured solutions method. Then, the IRF solution is computed with both codes. Results are then compared, with consideration of the numerical uncertainties of each solution.

The code comparison method has shown to be more convenient for verification studies than the comparison with a benchmark ODE, because it allows quantitative results comparisons. However, this method strongly depends on the behavior of the program used to obtain the reference solution.

TABLE DES MATIÈRES

DÉDICACE iii

REMERCIEMENTS iv

RÉSUMÉ v

ABSTRACT vii

TABLE DES MATIÈRES viii

LISTE DES FIGURES xii

LISTE DES NOTATIONS ET DES SYMBOLES xv

LISTE DES TABLEAUX xviii

INTRODUCTION 1

CHAPITRE 1 DESCRIPTION DU PROJET 2

1.1 Contexte industriel 2

1.1.1 Biothermica Technologies Inc. 2

1.1.2 Objectifs du projet de CFD à Biothermica 3

1.1.3 Objectifs généraux du projet de maîtrise 4

1.2 Revue de littérature 5

1.2.1 Bref historique de la vérification et validation 5

1.2.2 Terminologie 5

1.2.3 Vue d'ensemble 7

1.2.4 Vérification de Code 10

1.3 Présentation du projet de maîtrise 12

1.3.1 Cadre d'application de la méthode présentée 12
1.3.2 Objectifs spécifiques 13
1.3.3 Structure du mémoire 14

CHAPITRE 2 MÉTHODES DE VÉRIFICATION DE L'ALGORITHME NUMÉRIQUE 15
2.1 Solutions exactes et tests de consistance 15
2.2 Solutions Manufacturées 16
2.3 Comparaison à des solutions reconnues d'EDO ou d'EDP 18
2.4 Méthode de comparaison de codes 19

CHAPITRE 3 ESTIMATION D'ERREUR ET QUANTIFICATION D'INCERTITUDE 21
3.1 Théorie des études de convergence 22
3.2 Sources d'erreur 23
3.3 Le *Grid-Convergence Index* 24
3.3.1 Du facteur de sécurité F_s 26
3.3.2 Méthode des moindres carrés (MMC) 26
3.3.3 Méthode de Cadafalch et al. 29

CHAPITRE 4 CONTEXTE D'APPLICATION 32
4.1 Outils de travail 32
4.2 Description des codes utilisés 33
4.2.1 CFDLIB 3.1 33
4.2.2 Fluent 6.1 33
4.3 Modèle étudié 34
4.4 Métriques 35

CHAPITRE 5 APPLICATION DE LA MÉTHODE DE COMPARAISON À DES SOLUTIONS RECONNUES D'EDO 37

5.1 Description du problème de référence 37
5.1.1 Volume de contrôle 37
5.1.2 Conditions frontières 39
5.1.3 Solution de référence 39
5.2 Stratégie de la Vérification 44
5.3 Paramètres de simulation 45
5.3.1 Unités 45
5.3.2 Dimensionnement 45
5.3.3 Domaine 45
5.3.4 Maillage 48
5.3.5 Conditions initiales 48
5.3.6 Conditions frontières 49
5.4 Simulations 50
5.4.1 Matériel 50
5.4.2 Convergence des itérations 50
5.4.3 Statistiques 52
5.4.4 Performance 52
5.5 Résultats et analyse 53
5.5.1 Résultats 53
5.5.2 Convergence de la discrétisation 59
5.5.2.1 Application de la méthode de Cadafalch et al. 59
5.5.2.2 Convergence des Normes L^1 et L^2 67
5.5.3 Analyse 76

CHAPITRE 6 APPLICATION DE LA MÉTHODE DE COMPARAISON DE CODES 77
6.1 Description du problème 77
6.2 Confirmation du calcul par la MMS 79

6.2.1 Convergence des itérations 80
6.2.2 Évaluation d'erreur 81
6.2.3 Analyse des méthodes de Vérification de Calculs utilisées . . . 83
6.2.3.1 Méthode de Cadafalch et al. 83
6.2.3.2 Méthode des moindres carrés 87
6.3 Calcul du problème de référence 89
6.4 Simulations CFDLIB 90
6.4.1 Convergence des itérations 90
6.4.2 Statistiques 90
6.4.3 Performance 91
6.5 Comparaison 92
6.5.1 Comparaison avec la méthode de Cadafalch et al. 92
6.5.2 Comparaison avec la méthode des moindres carrés 102
6.6 Discussion 103

CONCLUSION 107

BIBLIGRAPHIE 110

ANNEXE 117

LISTE DES FIGURES

FIG. 5.1 Jet laminaire plan **(a)** provenant d'un embout avec un profil parabolique et **(b)** provenant d'une origine virtuelle fictive (solution de similitude). Tiré de Schlichting & Gersten (2000). 38

FIG. 5.2 Domaine de calcul 48

FIG. 5.3 Évolution dans le temps des normes L^∞ des dérivées temporelles des variables P, T', U et V. Ces valeurs sont calculées d'après la solution pour $N = 128$. 51

FIG. 5.4 Évolution du temps de calcul en fonction du nombre de cellules. 52

FIG. 5.5 Jet laminaire plan : comparaison entre la solution de référence et la solution obtenue par l'algorithme 2D MAC de CFDLIB. Pour chaque paire de graphiques, la solution de similitude est présentée à gauche, et la solution numérique se trouve à droite. . 54

FIG. 5.6 Distribution de pression (obtenue avec un maillage de 128×128 cellules) 56

FIG. 5.7 Évolution de la quantité de mouvement en Y du jet. 58

FIG. 5.8 Distribution de $((\phi_2^* - \phi_3^*) * (\phi_1^* - \phi_2^*))$ pour les 4 variables à l'étude (N = 128). 60

FIG. 5.9 Ordre de convergence local aur les nœuds de Richardson, pour les quatre variables à l'étude (N = 128). 61

FIG. 5.10 GCI absolus locaux pour les deux composantes de vitesse, d'après la méthode de Cadafalch et al. (2002). 62

FIG. 5.11 Graphique de la différence entre la solution numérique et la solution de similitude pout U et V. 68

FIG. 5.12 Convergence des normes L^1 et L^2 de T, U, et V. La courbe en pointillé est obtenue par la méthode des moindres carrés, de même que la valeur extrapolée, représentée par une droite continue. 74

FIG. 5.13 Graphiques logarithmiques de convergence des normes L^1 et L^2 : $\|T - T_{ER}\|$, $\|U - U_{ER}\|$, et $\|V - V_{ER}\|$. La droite en pointillé est obtenue par la méthode des moindres carrés. La ligne pleine représente la valeur de l'erreur d'arrondi, soit le plus petit écart pouvant être décrit par l'output de CFDLIB. 75

FIG. 6.1 Convergence itérative des résidus de la MMS, pour $N = 72$. . 81

FIG. 6.2 Convergence de la norme L^1 de l'évaluation de l'erreur, $|E|_{L^1}$. . 82

FIG. 6.3 Ordre de convergence observé, obtenu avec les évaluations d'erreur $|E|_{L^1}$, pour différents maillages. 82

FIG. 6.4 Méthode de Cadafalch et al adaptée à la MMS. Les figures de gauche présentent la distribution du GCI sur le domaine Ω_{RC}. Au centre, on retrouve la distribution de l'erreur sur les valeurs du maillage fin, rapportées sur le maillage d'analyse. Les graphiques de droites montrent en bleu les régions où l'erreur évaluée dépasse le GCI. 86

FIG. 6.5 Convergence des variables de la solution manufacturée. 88

FIG. 6.6 Convergence itérative des résidus de solution de référence, pour $N = 72$. 89

FIG. 6.7 Évolution dans le temps des normes L^∞ des dérivées temporelles des variables p, T, u et v. Ces valeurs sont calculées d'après la solution pour $N = 72$. 90

FIG. 6.8 Évolution du temps de calcul en fonction du nombre de cellules effectif. 91

FIG. 6.9 Comparaison des distributions de pression (à gauche) et des GCI y correspondant (à droite). 93

FIG. 6.10 Comparaison des distributions de température (à gauche) et des GCI y correspondant (à droite). 94

FIG. 6.11 Comparaison des distributions de vitesse u (à gauche) et des GCI y correspondant (à droite). 95
FIG. 6.12 Comparaison des distributions de vitesse v (à gauche) et des GCI y correspondant (à droite). 96
FIG. 6.13 Test de chevauchement local. Les régions où il y a chevauchement sont indiquées en vert, et celles où il n'y en a pas, en rouge. 100
FIG. 6.14 Etude du comportement de la pression : comparaison entre la variabilité de ΔP et la somme des GCI(P) des deux programmes. Notes : $\Delta P = P_{CFDLIB} - P_{Fluent}$; $\overline{\Delta P} = 0,736$. 101
FIG. 6.15 Convergence des variables p, T, u et v pour les deux programmes. 104

LISTE DES NOTATIONS ET DES SYMBOLES

A :	aire
α :	constante de diffusion thermique
C :	constante
C :	soution convergée d'un code
D :	résultats expérimentaux
δ :	erreur exacte
δ :	épaisseur d'un jet laminaire
E :	écart
E :	erreur
F :	facteur
f :	fonction de similitude
ϕ :	solution du système d'équation considéré
GCI :	*grid convergence index*
γ :	constante caractéristique de la distribution de la température, pour le problème du jet laminaire plan
H :	hauteur
h :	longueur caractéristique du raffinement de la discrétisation
η :	variable de similitude
J :	quantité de mouvement
L :	longueur
L :	norme
m :	constante
M :	solution exacte d'un modèle
N :	nombre caractéristique du nombre d'éléments de la discrétisation
n :	nombre

ν : viscosité cinématique

Ω : domaine

p : ordre de convergence

p : pression

$p(\mathbf{x})$: ordre de convergence local

Pr : Nombre de Prandtl

ψ : fonction de courant

Q : terme source

R : ratio

r : ratio de raffinement

ρ : masse volumique

S : simulation

$S(\phi_{ER}, C, p)$:mesure de l'écart entre les données (N_i, ϕ_i) et la courbe exponentielle définie par ϕ_{ER}

T : valeur réelle d'une grandeur d'intérêt

T : température

t : temps

θ : température adimensionnelle

U : incertitude

u : vitesse selon x

υ : fonction quelconque de x,y,z et t

v : vitesse selon y

$\mathbf{x}; x, y, z$: coordonées spatiales cartésiennes

X, Y : système de cordonnées alternatif

INDICES

C : convergé

$COMP$:	provenant du calcul numérique
$CONV$:	convergence
dep :	début du domaine considéré
ER :	extrapolation de Richardson
EXP :	provenant de l'expérimentation
emb :	embouchure
GCI :	*grid convergence index*
g :	*grids* (maillages)
∞ :	infini
jet :	centre du jet en x_{dep}
MOD :	modélisation
O :	oscillatoire
OBS :	observé
P :	paramètres utilisés pour la résolution d'un modèle
PAR :	relatif aux paramètres utilisés pour la résolution d'un modèle
R :	Richardson
REF :	référence
S :	simulation
s :	de sécurité
SN :	dû à la résolution numérique

LISTE DES TABLEAUX

TAB. 5.1	Unités utilisées	45
TAB. 5.2	Tableau des paramètres physiques du problème	46
TAB. 5.3	Conditions initiales.	49
TAB. 5.4	Conditions frontières	50
TAB. 5.5	Temps de calcul en fonction du nombre de cellules effectif.	53
TAB. 5.6	Caractéristiques de convergence de la température T', d'après l'analyse proposée par Cadafalch et al. (2002).	64
TAB. 5.7	Caractéristiques de convergence de la vitesse U, d'après l'analyse proposée par Cadafalch et al. (2002).	65
TAB. 5.8	Caractéristiques de convergence de la vitesse V, d'après l'analyse proposée par Cadafalch et al. (2002).	65
TAB. 5.9	Normes L^1 et L^2.	69
TAB. 5.10	Convergence des normes L^1 et L^2 de T, d'après la méthode du GCI Roache (1995).	70
TAB. 5.11	Convergence des normes L^1 et L^2 de U, d'après la méthode du GCI Roache (1995).	71
TAB. 5.12	Convergence des normes L^1 et L^2 de V, d'après la méthode du GCI Roache (1995).	72
TAB. 5.13	Convergence des normes L^1 et L^2, d'après la méthode des moindres carrés Eça & Hoekstra (2002).	73
TAB. 6.1	Tableau des paramètres du problème	78
TAB. 6.2	Caractéristiques de convergence de la pression p, d'après la méthode de Cadafalch et al. (2002).	84
TAB. 6.3	Caractéristiques de convergence de la température T, d'après la méthode de Cadafalch et al. (2002).	85

TAB. 6.4 Caractéristiques de convergence de la Vitesse u, d'après la méthode de Cadafalch et al. (2002). 85

TAB. 6.5 Caractéristiques de convergence de la Vitesse v, d'après la méthode de Cadafalch et al. (2002). 85

TAB. 6.6 Convergence des normes et L^2, d'après la méthode des moindres carrés (Eça & Hoekstra, 2002). 87

TAB. 6.7 Temps de calcul en fonction du nombre de cellules effectif. . . 91

TAB. 6.8 Caractéristiques de convergence de la Pression p, d'après l'analyse proposée par Cadafalch et al. (2002). Note : Les domaines Ω_R, Ω_C et Ω_O ne sont pas identique dans les deux situations. . 97

TAB. 6.9 Caractéristiques de convergence de la température T, d'après l'analyse proposée par Cadafalch et al. (2002). Note : Les domaines Ω_R, Ω_C et Ω_O ne sont pas identique dans les deux situations. 98

TAB. 6.10 Caractéristiques de convergence de la vitesse u, d'après l'analyse proposée par Cadafalch et al. (2002). Note : Les domaines Ω_R, Ω_C et Ω_O ne sont pas identique dans les deux situations. . 98

TAB. 6.11 Caractéristiques de convergence de la vitesse v, d'après l'analyse proposée par Cadafalch et al. (2002). Note : Les domaines Ω_R, Ω_C et Ω_O ne sont pas identique dans les deux situations. . 99

TAB. 6.12 Convergence des normes et L^2, d'après la méthode des moindres carrés (Eça & Hoekstra, 2002). Les valeurs et les GCI présentés sont ceux du maillage $N = 72$. 103

INTRODUCTION

De nos jours, le calcul numérique fait partie intégrante de la recherche et développement dans le domaine de la mécanique des fluides. Des méthodes complexes permettent une compréhension accrue des écoulements d'intérêt dans l'industrie. Cette complexité engendre la nécessité d'accomplir des tests formels, dans le but de s'assurer que les modèles utilisés, les programmes et les solutions qu'on en obtient soient adéquats. Ces tests font partie du domaine de la Vérification et Validation.

Le présent travail s'insère dans la démarche d'implantation du logiciel CFDLIB chez Biothermica Technologies Inc. Ce logiciel de volumes finis est peu couteux à l'achat, mais en contrepartie, il est offert sans aucun support et sans assurance sur la qualité. Le programme est donc présentement en phase de vérification. Cette activité nécessite l'établissement de méthode formelles de vérification d'algorithme et leur mise à l'épreuve.

Le travail présenté porte sur la mise à l'épreuve de deux méthodes de vérification d'algorithme : la comparaison avec des solutions reconnues d'EDO et la méthode de comparaison de codes. De plus, une approche générale d'analyse des résultats est présentée.

Dans un premier temps, le contexte théorique et industriel sera présenté. Puis, les trois chapitres suivants porteront sur trois aspects importants de la méthodologie : la description des méthodes de vérification utilisées, la présentation des méthodes d'estimation d'erreur et d'incertitude et la description du cadre dans lequel les tests ont été effectués. Finalement, les deux derniers chapitres présenteront les applications des méthodes de vérifications choisies.

CHAPITRE 1

DESCRIPTION DU PROJET

Le présent projet de vérification de code fait partie d'une stratégie visant à intégrer la mécanique des fluides assistée par ordinateur aux activités de R&D de Biothermica Technologies Inc. Afin de comprendre les contraintes associées à ce type de projet, et ainsi identifier correctement à quelles situations s'adressent les méthodes à l'étude, il est d'abord nécessaire de présenter le contexte industriel dans lequel ces activités ont lieu. Par la suite, une brève revue de la littérature permettra de situer le rôle joué par la vérification dans le cadre d'activités de modélisation assistée par ordinateur. Une fois ces éléments fondamentaux posés, les objectifs spécifiques du projet de recherche et la structure du mémoire seront détaillés.

1.1 Contexte industriel

1.1.1 Biothermica Technologies Inc.

Biothermica est une entreprise offrant des services de conception et de réalisation de projets dans le domaine du traitement des gaz et des déchets. Sa division rattachée au contrôle de la Pollution de l'air a développé le procédé BIOTOX©, qui permet d'effectuer le traitement d'émissions de composés organiques volatils (COV), de composés organiques condensables (COC) et de soufres réduits totaux (SRT). Ce procédé d'oxydation regénérative est le fruit de plusieurs années de développement, et il est toujours l'objet d'une recherche active. D'ailleurs, un aspect important de cette recherche est la modélisation des phénomènes physiques propres au procédé, notamment les écoulements

en milieu poreux, ainsi que le comportement mécanique et thermique des fluides dans les diverses canalisations et dispositifs utilisés dans le procédé. Les méthodes numériques de modélisation (en particulier, pour la situation présente, celles relevant du domaine de la mécanique des fluides assistée par ordinateur, ou CFD pour *computational fluid dynamics*) sont donc à considérer. En effet, elles permettent l'étude de problèmes complexes là où les méthodes théoriques classiques restent insuffisantes, ou encore permettent d'obtenir certaines conclusions sans avoir à procéder à de coûteuses expérimentations[1].

En outre, d'après les données du site canadien des entreprises et des consommateurs (Industrie Canada, 2006), Biothermica compte une vingtaine d'employés et affiche un chiffre d'affaire de l'ordre de cinq à dix millions de dollars. Or, à cette échelle, il serait trop coûteux d'implanter des logiciels commerciaux de CFD au sein même de l'entreprise. Ainsi, une des alternatives envisagées pour inclure la CFD au sein des activités de recherche est de se tourner vers un code de recherche pour effectuer les simulations numériques. En effet, plusieurs codes ayant été élaborés pour des activités de recherche sont disponibles pour une fraction du prix des codes commerciaux. En contrepartie, ces derniers sont souvent moins performants, peu ou pas documentés, et beaucoup moins largement répandus et utilisés.

1.1.2 Objectifs du projet de CFD à Biothermica

Au printemps 2004, Biothermica fait l'acquisition du solveur CFDLIB 3.1. La distribution comprend un libre accès au code source, des exemples de fichiers d'input, des scripts développés par les utilisateurs, ainsi qu'une base de documentation contenant un guide d'utilisateur et un document portant sur les bases théoriques utilisées ou développées pour son élaboration. Aucun support à l'usager n'est offert, à l'exception de l'accès

[1] La modélisation numérique ne peut toutefois se substituer entièrement à l'expérimentation. Ce point, qui fait appel au concept de validation, sera expliqué plus en détail à la section 1.2

à une liste de courriels d'autres utilisateurs. De plus, aucune information n'est fournie sur l'état de la vérification et de la validation du logiciel. Enfin, la documentation est souvent approximative ou incomplète et peut également contenir des instructions sur des parties non-implantées du code.

Le projet de CFD à Biothermica peut être résumé par les objectifs suivants :

- Évaluation du potentiel de CFDLIB et des ressources nécessaires à sa réalisation.
- Planification de la vérification et de la validation des outils numériques acquis ou développés.
- Vérification des algorithmes numériques (vérification du code).
- Normalisation des méthodes utilisées pour les estimations d'erreur numérique.
- Validation des modèles numériques.
- Acquisition, au sein de l'entreprise, de savoir-faire sur la CFD en général et sur CFD-LIB en particulier (Expertise, documentation).
- Implantation de CFDLIB dans les activités de recherches.

Les études effectuées jusqu'à présent ont eu pour principal objectif la vérification des algorithmes numériques. Elles permirent au passage d'acquérir un savoir-faire sur ce logiciel, sur les méthodes d'analyse et de constituer une documentation de base sur son utilisation, non seulement dans le but de rapporter les résultats obtenus, mais également de servir d'outil de formation de futurs utilisateurs.

1.1.3 Objectifs généraux du projet de maîtrise

Le projet de maîtrise porte sur la vérification de CFDLIB. Il a pour but d'identifier et d'évaluer des méthodes de vérification de code et d'estimation d'erreur numérique, destinées à être appliquées au code "de recherche" qu'est CFDLIB.

Les méthodes doivent être applicables dans les conditions définies à la section 1.3.1.

1.2 Revue de littérature

1.2.1 Bref historique de la vérification et validation

Le développement du calcul assisté par ordinateur a ouvert la porte à de nouveaux moyens de modélisation, si puissants et prometteurs que les gens les plus optimistes prévoyaient la fin des sciences expérimentales au profit des sciences du calcul numérique (Chapman et al., 1975). Tout comme la puissance de calcul des ordinateurs, la recherche sur les algorithmes et sur les outils connexes au calcul numérique (maillage, visualisation, etc.) a connu des avancées importantes. Comme les codes devenaient à la fois plus complexes et plus utilisés (notamment dans des applications à haut risque : énergie nucléaire, environnement, aéronautique, etc.), les questions de caractérisation d'erreur et de contrôle de la qualité des codes ont pris de plus en plus d'importance. Le domaine de la vérification et validation[2] a émergé en réponse à ce besoin.

1.2.2 Terminologie

Dans le contexte de la V&V, les mots "vérification" et "validation" sont des termes techniques qui représentent des concepts distincts, et non des synonymes comme dans leur usage courant. La vérification désigne les démarches qui visent à s'assurer que le modèle numérique implanté dans un code correspond bien à sa description symbolique. Il s'agit d'une démarche qui concerne uniquement la cohérence mathématique entre le code et le modèle, et d'aucune façon sa relation avec le monde réel. Cette dernière question est plutôt du ressort de la validation.

[2]Pour désigner le processus global concernant la qualité des codes de calcul numérique, certains auteurs ont proposé les termes *Certification* (Aeschliman et al., 1995), *Validation science* (Oberkampf & Trucano, 2002) ou encore le symbole V2V (Roache, 2004). L'appelation *Verification and Validation* (V&V) reste toutefois la plus répandue dans la littérature.

La V&V fait appel aux concepts souvent confondus, ou définis de façon équivoque, d'"erreur" et d'"incertitude". Cette confusion a été entretenue par l'opposition entre les définitions en usage dans le jargon des sciences expérimentales (Coleman & Stelle Jr., 1989) et certaines ayant été suggérées pour les sciences du calcul numérique (AIAA, 1998). Toutefois, l'usage pratique des termes le plus courant[3] se rapproche des concepts utilisés dans les sciences expérimentales. Selon ces définitions (Coleman (2003) d'après ISO (1993, 1995)) : (1) une *erreur* est la différence entre le résultat d'une mesure et la vraie valeur du mesurande ; (2) une *incertitude* est une grandeur, associée au résultat d'une mesure, qui caractérise la dispersion des valeurs qui pourraient raisonnablement être attribuées au mesurande. Ces définitions ont été étendues au calcul numérique par Coleman & Stern (1997).

L'appellation "vérification et validation" (ou V&V) désigne trois activités distinctes[4] (ce qui a été souligné notamment dans Roache (2004), Pelletier & Roache (2006)) de quantification d'erreurs et d'incertitudes : la Vérification de Codes, la Vérification de Calculs, et la Validation. Premièrement, la démarche de la Vérification de Codes consiste à évaluer si le code converge bien vers la solution du modèle. Ceci requiert une procédure d'*évaluation* d'erreur, qui se fait à partir d'une solution connue. Deuxièmement, la Vérification de Calculs est une démarche qui utilise l'*estimation* de l'erreur numérique, afin d'associer au résultat d'une simulation un intervalle de confiance. L'estimation d'erreur est faite à partir de deux solutions de précisions différentes, la plupart du temps dans un contexte où la solution exacte est inconnue. Cette estimation de l'erreur – ainsi que la solution numérique elle-même – ne peut avoir de valeur que si le bon comportement du code a été correctement établi au moyen de la Vérification de Code. Troisièmement, la Validation est la démarche qui vise à évaluer dans quelle mesure le résultat d'une simulation se compare à la grandeur correspondante du monde réel. Elle implique la

[3]Usage que l'on retrouve dans des sources telles que Roache (1998b), Eça & Hoekstra (2003), Coleman (2003), Pelletier & Roache (2006).

[4]Le mot "vérification" est utilisé pour décrire deux activités distinctes : la Vérification de Codes et la Vérification de Calculs. Cette nomenclature équivoque est toutefois déjà bien établie.

comparaison des résultats de simulation avec des données expérimentales.

1.2.3 Vue d'ensemble

Une récente contribution au domaine de la V&V (Coleman, 2003) jette la lumière sur l'ensemble des activités de vérification et de validation, et permet d'en mesurer les interactions.

Notons d'abord que le symbole δ sera utilisé pour désigner des erreurs. Aussi, à l'erreur δ, on peut associer l'incertitude U, pouvant être interprétée comme étant l'intervalle $\pm U$ contenant δ dans un pourcentage donné des cas (Coleman, 2003). Par exemple, on définit U_{95} comme remplissant la condition :

$$|\delta| \leq U_{95} \tag{1.1}$$

dans 95% des cas (Pelletier & Roache, 2006).

Coleman (2003) désigne par T la valeur réelle d'une grandeur d'intérêt, telle que définie par un observateur. D'une part, le processus d'expérimentation nous fournit la valeur D qui s'en veut le plus près possible. La différence $D - T$ est l'erreur expérimentale δ_{EXP}.

D'autre part, on peut concevoir un modèle dans le but de représenter la réalité, dont la solution exacte est la grandeur M. L'erreur δ_{MOD}, correspondant à $M - T$, provient de la différence entre le modèle et la réalité.

De plus, même avec la capacité de solutionner correctement un modèle, la solution obtenue dépendra toujours de la précision des paramètres P' utilisés dans la résolution. L'erreur commise par l'utilisation de paramètres ne correspondant pas exactement à ceux de la situation réelle est nommée δ_{PAR}. La grandeur δ_{PAR} est définie comme étant la dif-

férence entre la solution exacte du modèle avec des paramètres inexacts comme input, $M_{P'}$, et la solution exacte du modèle utilisant des paramètres exacts, M.

Enfin, dans le cas du calcul assisté par ordinateur, le modèle est évalué de façon numérique. La solution numérique, nommée S pour son caractère discret, diffère du modèle par la grandeur δ_{SN}, soit $S - M_{P'}$. Ainsi, la différence entre la simulation et la réalité, δ_S, est :

$$\begin{aligned} \delta_S &= S - T && (1.2) \\ &= (S - M_{P'}) + (M_{P'} - M) + (M - T) && (1.3) \\ &= \delta_{SN} + \delta_{PAR} + \delta_{MOD} && (1.4) \end{aligned}$$

L'incertitude sur une simulation peut donc être vue comme[5] :

$$|U_S| = |U_{SN} + U_{PAR} + U_{MOD}| \quad (1.5)$$

Or, il n'existe aucun moyen *a priori* d'évaluer U_{MOD}. On doit donc obligatoirement passer par la comparaison entre les résultats de simulation et ceux de l'expérience. Cette comparaison doit être faite à la lumière des incertitudes qui nous viennent du calcul numérique du modèle et de celles provenant de l'expérimentation.

D'une part, l'incertitude que l'on peut évaluer sur le résultat d'une simulation U_{COMP} est composée de U_{SN} et de U_{PAR}. D'autre part, la valeur réelle T est évaluée par D avec une précision U_{EXP}. Donc, il en découle que la concordance du modèle avec la réalité ne peut être connue avec une précision plus grande que $|U_{COMP} + U_{EXP}|$. Si, toutefois, l'erreur de comparaison E_{OBS}, définie comme étant $E_{OBS} = D - S$, est plus grande que

[5]En autant que les incertitudes ainsi combinées caractérisent des distribution de manière comparable, i.e. on combine des U_{95} avec des U_{95} et non un U_{50} avec des U_{95} (Pelletier & Roache, 2006).

$|U_{COMP} + U_{EXP}|$, il est fort probable que cet écart soit dû à l'imprécision du modèle (à plus forte raison lorsque $|E_{OBS}| \gg |U_{COMP} + U_{EXP}|$).

Ainsi, le "niveau de validation" est donné par le maximum de $\{|E_{OBS}|, |U_{COMP} + U_{EXP}|\}$. Il caractérise la précision maximale avec laquelle on peut affirmer que les résultats du code représentent la réalité.

En résumé, trois estimations d'incertitudes sont nécessaires pour permettre la comparaison avec les résultats expérimentaux (Coleman, 2006) : U_{SN}, U_{PAR} et U_{EXP}. L'étude de l'incertitude U_{SN} inhérente aux méthodes de calcul numérique est l'objet de la Vérification de Calculs. L'incertitude U_{PAR}, due aux paramètres utilisés, est étudiée par le moyen d'analyses de sensibilité (voir notamment Pelletier et al. (2003)). Notons qu'il s'agit dans ce cas d'un *résultat* des calculs numériques. Quant à l'incertitude U_{EXP}, elle peut être appréhendée au moyen de méthodes d'analyse d'incertitude expérimentale répandues et éprouvées.

Pour une plus ample discussion sur cette approche, voir Coleman & Stern (1997), Roache (1998a), Coleman & Stern (1998), Stern et al. (2001), Wilson et al. (2001), Oberkampf (2002), Coleman (2002), Stern & Wilson (2002), Coleman (2003), Stern et al. (2004), Pelletier & Roache (2006).

Finalement, précisons que l'approche globale présentée ici n'est pas encore largement répandue et que plusieurs points restent probablement à débattre. Toutefois, elle permet de comprendre et d'englober un grand nombre de pratiques qui font largement consensus dans le milieu de la V&V. Elle fait partie des efforts entrepris afin d'établir des lignes directrices, des recommandations et éventuellement des standards au sujet de la V&V. À ce propos, il reste encore beaucoup à faire, comme en témoigne la variété des approches, des méthodes, ainsi que l'importance apportée à chaque étape de la V&V (Abanto et al., 2005, Benay et al., 2003, Cadafalch et al., 2002, Eça & Hoekstra, 2004, Flynn & Eisner,

2004, Freitas, 2002, Lacasse et al., 2001, 2004, Pelletier et al., 2003, 2004, Roache, 1994, 1998b, Salari & Knupp, 2000, Wilson et al., 2001).

1.2.4 Vérification de Code

L'objectif de la Vérification de Code est de "prouver", à partir de solutions connues, que le code représente correctement le modèle mathématique dont il doit permettre la résolution. Il s'agit d'une démarche purement mathématique, faite de façon totalement indépendante de l'expérimentation.

Certains auteurs (Oberkampf & Trucano, 2002, Oberkampf et al., 2002) mettent l'emphase sur le point que la Vérification de Code doit se faire en deux étapes : le contrôle de la qualité du logiciel ou SQA (*software quality assesment*), suivi de la vérification de l'algorithme numérique.

La SQA est un sujet vaste, dont l'usage est peu répandu en CFD comparativement à d'autres domaines, probablement en raison des risques jugés moins importants en cas de défaillance. La SQA couvre les sujets de l'analyse statique, dynamique et formelle de codes. Des études montrent que tous les codes, même les plus éprouvés, sont susceptibles de contenir des failles, et qu'une grande proportion d'entre elles peut être révélée par la SQA (Hatton, 1997).

Néanmoins, pour la vérification de codes d'éléments finis, de différences finies ou de volumes finis, l'accent est généralement mis sur la vérification de l'algorithme numérique[6]. Il est important de rappeler que ceci implique à la base le concept d'*évaluation* d'erreur qui nécessite la connaissance d'une solution exacte.

Il existe de nombreux facteurs affectant la précision de la solution numérique d'un mo-

[6]D'après l'appelation d'Oberkampf & Trucano (2002), reprise également dans le document de l'AIAA (1998) : *Numerical algorithm verification*

dèle (Aeschliman et al., 1995, ERCOFTAC, 2000) :

- La méthode de discrétisation des équations (l'EDP, les conditions frontières, etc).
- La convergence de la discrétisation du domaine spatial et/ou temporel.
- La convergence itérative, i.e. le niveau de précision de la méthode itérative de résolution.
- Les erreurs d'arrondis.
- Les erreurs de programmation.
- Les erreurs faites par le ou les usagers.

En pratique, pour une simulation faite sur un code robuste, consistant et sans erreur de programmation, si l'usager l'utilise correctement, la source d'erreur la plus importante viendra de la nature discrète de la modélisation du domaine spatial ou temporel. C'est pourquoi l'activité la plus importante de la Vérification de Code est l'exécution d'études de convergence de la discrétisation. D'une part, ces études nous permettent de vérifier si l'erreur de discrétisation tend vers zéro à mesure que l'on raffine le maillage. D'autre part, le rythme avec lequel cette erreur diminue (l'ordre de convergence) peut nous indiquer si la méthode de discrétisation des équations se comporte tel que prévu.

Tel qu'énoncé plus haut, l'*évalution* d'erreur nécessite la comparaison avec une solution connue. Éprouvée sur une panoplie de codes, la méthode des solutions manufacturées est la méthode la plus puissante pour obtenir des solutions exactes et complexes d'EDP. Toutefois, d'autres méthodes ont été imaginées, ayant toutes comme principe de base la comparaison avec des solutions à haute précision (Oberkampf & Trucano, 2002, AIAA, 1998). La description des méthodes de vérification d'algorithme numérique sera effectuée au chapitre 2.

Enfin, notons que la convergence complète[7] est très difficile à atteindre en pratique. Le processus d'étude de convergence peut toutefois être simplifié par l'utilisation de mé-

[7]Par convergence complète, on n'entend pas ici que l'erreur numérique soit nulle, mais plutôt que la solution se trouve dans la zone asymptotique de convergence. Dans cette zone, le comportement en convergence est monotone et l'évolution de la solution – et de l'erreur – est prévisible.

thodes adaptatives de maillages (Ilinca et al., 1997, Turgeon & Pelletier, 2002, Pelletier et al., 2003, 2004).

1.3 Présentation du projet de maîtrise

1.3.1 Cadre d'application de la méthode présentée

Tel que mentionné à la section 1.1.2, la documentation de CFDLIB est trop fragmentaire pour tenter d'apporter des modifications au cœur du code source sans risquer d'y rajouter des défaillances. D'autre part, la méthode la plus établie pour la vérification d'algorithme est la méthode des solutions manufacturées (MMS), qui demande la modification du système d'EDP par l'ajout de termes sources. Or, comme aucun endroit n'était prévu ni indiqué pour l'ajout de termes sources dans CFDLIB, la MMS ne pouvait être employée pour la vérification de code.

Le présent projet vise donc à évaluer les solutions de remplacement à la MMS pour la vérification de code. La portée des conclusions de ces méthodes pour la vérification d'algorithme est cependant plus limitée qu'avec la MMS. L'idée n'est donc pas de chercher à remplacer la MMS, et nous croyons que dans toute situation de vérification de code où son utilisation est possible, la MMS devrait être préférée aux autres méthodes de comparaison de résultats. Il peut cependant survenir des circonstances faisant en sorte que l'on veuille mettre à l'épreuve un programme en évitant d'utiliser la méthode des solutions manufacturées. On doit donc être conscient des exigences que les méthodes alternatives doivent remplir (dans un objectif de vérification de code), et des limites inhérentes à chaque méthode.

Les méthodes présentées dans le présent mémoire peuvent être appliquées dans les circonstances suivantes :

- Les études sont contraintes à des méthodes non-intrusives (black-box). Le code source doit donc autant que possible rester inchangé.
- Le programme n'est pas conçu de manière à pouvoir ajouter des termes sources dans les EDP résolues.

On utilise dans le présent mémoire le terme *code "de recherche"* pour désigner pareille situation. Cela vise uniquement à alléger le vocabulaire employé pour décrire des situations similaires à celle de la vérification de CFDLIB.

1.3.2 Objectifs spécifiques

Le présent projet de maîtrise a pour but d'orienter les activités de Vérification de Code de CFDLIB et de mettre à l'épreuve les démarches proposées. L'algorithme est testé pour des écoulements 2D à l'état permanent. Les méthodes élaborées serviront éventuellement de base à une vérification plus complète du programme, conçu pour les écoulements instationnaires 2D ou 3D. Considérant les contraintes spécifiques à l'utilisation de CFDLIB (voir section 1.3.1), les méthodes mises à l'épreuve pour la Vérification de Code sont les suivantes :

1. Comparaison à des solutions reconnues d'EDO.
2. Méthode de Comparaison de Codes.

Ces méthodes seront jugées sur la base de leur capacité à fournir des solutions suffisament précises et sur la rigueur des conclusions que l'on peut tirer de leur utilisation. De plus, comme on ne dispose pas de solutions exactes, il est nécessaire d'utiliser des méthodes d'estimation d'erreur et d'incertitude. Ainsi, un sous-objectif est de présenter une stratégie d'analyse des résultats pour l'étude des incertitudes. Cette procédure comprend deux variantes du *grid-convergence index* : la méthode des moindres carrés et la méthode de Cadafalch et al.

Enfin, précisons que les données et les résultats obtenus sont consignés de manière à ce que les études de vérification effectués soient répétables.

1.3.3 Structure du mémoire

Le présent chapitre visait à présenter à quel niveau se situe le projet de maîtrise au sein d'une démarche globale de Vérification et Validation. Les chapitres 2 et 3 présenteront la théorie relative aux méthodes appliquées pour les travaux de Vérification de Code dans le cadre du projet de maîtrise. Le chapitre 4 consiste en une description détaillée du contexte dans lequel est effectuée la mise à l'épreuve des méthodes proposées pour la Vérification de Code "de recherche". Les chapitres 5 et 6 présentent l'application de ces méthodes.

CHAPITRE 2

MÉTHODES DE VÉRIFICATION DE L'ALGORITHME NUMÉRIQUE

La vérification d'algorithme numérique est une démarche d'*évaluation* d'erreur. Celle-ci est faite par comparaison avec des solutions à haute précision (AIAA, 1998, Oberkampf & Trucano, 2002). Ces solutions peuvent provenir directement de la résolution des EDP du modèle étudié. Il peut s'agir de solutions analytiques, ou de solutions reconnues d'EDO ou d'EDP. Toutefois, ces solutions ne portent que sur un nombre restreint de problèmes. De plus, elles nécessitent en général la simplification ou l'approximation (conceptuelle ou numérique) de l'EDP de départ.

D'autre part, au moyen d'une légère modification de l'EDP, il est possible de générer des solutions exactes quelconques dont la complexité est arbitraire. On nomme cette techniques la Méthode des Solutions Manufacturées.

2.1 Solutions exactes et tests de consistance

Les solutions exactes sont des solutions fermées de l'EDP étudiée, obtenues de façon analytique. Les solutions analytiques ont l'avantage de permettre de calculer directement la différence entre la solution numérique et la solution exacte sur tout le domaine, sans source possible d'erreur sur la solution de comparaison. Le désavantage est que les solutions analytiques ne s'appliquent qu'à un nombre restreint de problèmes, lesquels proviennent d'une forme souvent très simplifiée de l'EDP. Certains termes de l'EDP ne se manifestant pas dans la solution, il est possible que des erreurs dans l'évaluation discrète de ces termes ne soient pas relevées par le processus de comparaison.

L'écoulement de Couette, de Poiseuille, de Hagen-Poiseuille, ou l'écoulement axial entre deux cylindres concentriques sont des exemples de solutions exactes. Mentionnons que CFDLIB a d'abord été mis à l'épreuve avec ces écoulements (Reid, 2004). On retrouve également dans la littérature quelques exemples de tests de vérification de code effectués à l'aide de solutions exactes (Abanto et al., 2005, Cadafalch et al., 2002).

Les tests de consistance regroupent un ensemble de contrôles variés : la vérification globale de la conservation de la masse, de la quantité de mouvement ou de l'énergie sur l'ensemble du domaine ; la confirmation du comportement adéquat des conditions frontières implantées ; la répétitivité des résultats, peu importe l'orientation des axes ou l'échelle utilisée ; etc. Ces contrôles font partie des tests de base effectués lors de l'élaboration d'un code et peuvent également être utilisés pour s'assurer que l'on utilise les options du code correctement, ou encore si elles sont correctement implantées.

Les tests de solution exacte et de consistance peuvent être utiles lors de la programmation ou comme première approche à la vérification. Ils sont également pertinents à la formation des usagers.

2.2 Solutions Manufacturées

La Méthode des Solutions Manufacturées (MMS, pour *Method of Manufactured Solutions*) consiste à prendre le problème de résolution de l'EDP à rebours.

Au début de la démarche de MMS, on définit des fonctions analytiques quelconques pour les variables de solution. Par exemple :

$$v = M(x, y, z, t) \tag{2.1}$$

Ensuite, considérons l'opérateur L, qui symbolise le système d'équation du modèle ma-

thématique à résoudre. Aussi complexe que soit le système d'EDP, on peut l'exprimer sous cette forme :

$$L[\phi] = 0 \tag{2.2}$$

où ϕ serait une solution exacte du système. Si on applique l'opérateur L à la solution v, on obtient le système suivant :

$$L[v] = Q(x, y, z, t) \tag{2.3}$$

Le terme Q, sans égard à la complexité du problème et de la solution choisie, peut être déterminé analytiquement. Ceci peut être fait facilement et de façon fiable, au moyen d'un logiciel de manipulation symbolique.

Par la suite, on doit transformer le système implanté dans l'algorithme numérique, de façon à y inclure les termes sources représentés par Q. Encore une fois, cette étape peut être effectuée avec peu de risques, en autant que la manière d'inclure des termes sources dans les calculs du code soit clairement définie. Ici, on a encore recours aux logiciels de manipulation symbolique, qui permettent généralement de traduire des expressions dans les langages de programmation les plus courants.

Précisons que les fonctions v doivent être assez complexes pour faire intervenir tous les termes de l'EDP.

À l'étape de vérification de code, la solution n'est pas tenue d'avoir une signification physique. Il peut même être avantageux de ne pas avoir à se soucier de la physique lors de l'élaboration d'un problème de MMS. En effet, bien souvent, les problèmes réels comportent des configurations de l'écoulement (par exemple, la couche limite) qui font que certains termes ont très peu de poids dans l'EDP.

Soumise à de nombreuses études, la MMS s'est révélée comme étant la méthode la plus

efficace pour détecter les erreurs de programmation affectant la précision de la solution. Elle permet de prouver, "hors de tout doute raisonnable" (Roache, 2004), la convergence de l'algorithme numérique vers la solution exacte. En effet, contrairement aux autres méthodes, même si la MMS est susceptible de faire croire à la présence de failles dans l'algorithme quand les erreurs constatées seraient plutôt dues à l'utilisateur, il est improbable qu'un code erroné passe le test de la MMS. En effet, cela nécessiterait que les erreurs faites lors de l'application de la MMS compensent l'effet des erreurs de programmation.

La littérature, de plus en plus abondante sur le sujet, permet de comprendre le concept et la portée de la MMS, tout en témoignant de son efficacité (Abanto et al., 2005, Roache, 1998b, 2002a,b, 2004, Salari & Knupp, 2000, Pelletier & Roache, 2006).

2.3 Comparaison à des solutions reconnues d'EDO ou d'EDP

La méthode de comparaison à des solutions reconnues d'EDO ou d'EDP a été introduite comme une façon d'obtenir des solutions précises de problèmes plus complexes que ceux pouvant être résolus analytiquement (voir section 2.1). À cet effet, on retrouve dans la littérature (AIAA, 1998, Oberkampf & Trucano, 2002) la suggestion d'utiliser uniquement des solutions confirmées par plusieurs sources indépendantes, idéalement à partir de plusieurs méthodes numériques différentes.

On peut toutefois émettre des réserves quant à la valeur des évaluations d'erreur faites avec cette méthode. En effet, bien que des solutions d'EDO soient bien établies (e.g. la solution de Blasius de la couche limite sur une plaque plane, le jet laminaire), l'équation solutionnée ne correspond pas exactement au modèle implanté dans le code. La problématique liée à l'utilisation d'EDO pour la vérification est traitée en détail au chapitre 5, où un exemple d'application est présenté.

Quant aux solutions reconnues d'EDP, ayant été obtenues par des méthodes numériques, leur véracité repose d'abord et avant tout sur la qualité des codes d'où elles proviennent. Cet enjeu est traité à la section 2.4.

2.4 Méthode de comparaison de codes

La comparaison de codes a longtemps été utilisée comme manière intuitive de vérification de résultats numériques. Toutefois, rarement ont-elles été effectuées sur la base d'un principe formel. La règle, énoncée dans un récent rapport de Trucano et al. (2003), est pourtant simple. Si l'on nomme C_1 la solution convergée du code1 (que l'on veut vérifier), C_{REF} la solution convergée du code de référence (utilisée pour la comparaison), et M, la solution exacte du modèle, le principe de comparaison de code est le suivant :

$$|C_1 - M| \leq |C_1 - C_{REF}| + |C_{REF} - M| \tag{2.4}$$

Ainsi, loin de simplifier le processus de vérification, la méthode de comparaison de codes (MCC) requiert d'abord l'accumulation de preuves de vérification pour le code de référence, avant même de procéder à la comparaison. Trucano et al. (2003) soutiennent que si une telle preuve peut être accumulée pour le code de référence, on devrait pouvoir appliquer la même démarche au code1.

Dans le cas présent, d'après l'état actuel de la littérature, la méthode la plus rigoureuse de vérification de code est la MMS (Roache, 1998b, 2002a,b, 2004, Salari & Knupp, 2000, Pelletier & Roache, 2006). Or, cette méthode ne peut être utilisée pour la vérification d'un code "de recherche", tel qu'il est défini à la section 1.3.1. D'où l'intérêt porté envers la MCC.

Dans un premier temps, les résultats du code de référence seront vérifiés au moyen de

la MMS. Le problème sera choisi de sorte que la solution manufacturée soit semblable à la solution obtenue dans le code de référence, sans inclusion des termes sources. Par la suite, le problème sera analysé dans le code de référence (sans les termes sources), puis dans CFDLIB. Enfin, les résultats seront comparés. Pour ce faire, on doit au préalable estimer les incertitudes sur les résultats de chaque code, puis, au moment de la comparaison, constater s'il y a ou non chevauchement des bandes d'incertitudes.

CHAPITRE 3

ESTIMATION D'ERREUR ET QUANTIFICATION D'INCERTITUDE

Il existe dans la littérature de nombreuses méthodes d'estimation d'erreur qui tombent généralement dans la catégorie des estimateurs *a posteriori* et des estimateurs *a priori*. Les méthodes *a priori* utilisent uniquement les propriétés de l'algorithme numérique (sa façon d'approximer les opérateurs de différentiels et les conditions frontières) pour recueillir l'information nécessaire à l'estimation d'erreur (d'après Oberkampf et al. (2002)). Les méthodes *a posteriori* consistent quant à elles à se servir à la fois des propriétés de l'algorithme *et* de solutions numériques obtenues par cet algorithme afin d'estimer l'erreur. Ces solutions peuvent être obtenues à partir de différents maillages (utilisant chaque fois le même algorithme), ou encore à partir d'algorithmes numériques d'ordres de convergence différents (où un seul maillage peut alors être utilisé).

En pratique, seules les études de convergence du maillage permettent une Vérification de Code rigoureuse, car elles peuvent en principe être menées jusqu'à un niveau de précision arbitraire (Roache, 1998b).

En outre, les études de convergence sont largement utilisées car il s'agit d'une procédure directe, non-intrusive, qui constitue probablement la technique la plus fiable pour estimer quantitativement l'erreur (Ilinca et al., 2000, Oberkampf et al., 2002, Pelletier & Roache, 2006).

En ce sens, le *Grid-Convergence Index* (et les méthodes qui en sont dérivées) utilise la capacité d'estimation d'erreur des études de convergence. On peut par la suite en déduire une estimation quantitative de l'incertitude U_{SN}.

Les analyses des applications présentées utilisent trois variantes du GCI : analyse globale sur un triplet (i.e. trois maillages de résolutions différentes), analyse globale par la méthode des moindres carrés et analyse locale par la méthode de Cadafalch et al. (2002).

3.1 Théorie des études de convergence

Le principe des études de convergence de maillage (ou de discrétisation) est que la précision du résultat numérique augmente avec le raffinement du maillage. Le rythme de cette augmentation –l'ordre de convergence p– est propre à l'algorithme numérique utilisé.

Autrement dit, si on désigne l'erreur comme étant la différence entre la solution discrète $\phi(h)$ et la solution exacte de l'EDP ϕ_{exact}, soit :

$$E = \phi(h) - \phi_{exact} \tag{3.1}$$

alors, pour une méthode numérique d'ordre p, l'erreur E devrait tendre vers zéro de façon asymptotique, proportionnellement au terme h^p (Roache, 1998b). Le terme h est une longueur caractéristique de la finesse de la discrétisation. Ainsi, pour un maillage suffisament fin, on devrait pouvoir observer la relation asymptotique suivante :

$$E \approx C \cdot h^p \tag{3.2}$$

On nomme généralement *zone asymptotique de convergence* la plage de h où cette relation peut être constatée.

Les observations présentées ici s'appliquent à toute méthode de discrétisation consistante, que la solution soit lisse ou non. Une méthode est considérée consistante lorsque les équations aux dérivées partielles, continues, sont équivalentes au système discret lorsque le raffinement est infini.

Note : On peut aussi utiliser une forme alternative à l'équation (3.2), si on veut décrire la discrétisation par un nombre de mailles caractéristique N, au lieu d'utiliser la longueur h. N est alors inversement proportionnel à h. Ainsi, la relation (3.2) devient :

$$E \approx C \cdot N^{-p} \tag{3.3}$$

où la constante de proportionnalité C peut être différente de celle de présentée en (3.2). Cette formulation est plus pratique, car la discrétisation peut ainsi être décrite par des nombres entiers.

3.2 Sources d'erreur

En pratique, pour effectuer une étude de convergence de maillage, on doit faire en sorte que la principale source d'erreur observable soit la convergence de la discrétisation. En général, les quatres sources d'erreur numériques principales, pour un code de FEM, FDM ou FVM, sont les suivantes :

- La convergence de la discrétisation (spatiale ou temporelle)
- La convergence itérative
- Les erreurs d'arrondis
- Les erreurs de programmation

En CFD, les résultats peuvent varier de façon importante en fonction du raffinement du maillage utilisé. Les erreurs d'arrondis, dues à l'utilisation de nombres à virgule flottante, sont donc la plupart du temps inférieures à l'erreur de discrétisation de plusieurs ordres de grandeur.

Toutefois, une plus grande attention doit être apportée à la convergence itérative, qui peut être une source d'erreurs non négligeables. En effet, le processus itératif consiste à résoudre le système d'équations défini avec la méthode de CFD par une suite d'approxi-

mations successives, en se rapprochant peu à peu de la solution exacte[1]. En principe, on peut réduire l'erreur de convergence itérative jusqu'au niveau de l'erreur d'arrondi ; les résultats calculés étant au mieux d'une précision semblable à celle des nombres gardés en mémoire par l'ordinateur. En pratique, l'erreur de convergence itérative est souvent supérieure à l'erreur d'arrondi, principalement parce que l'usager peut limiter le processus itératif pour obtenir des solutions plus rapidement.

Les erreurs de programmation ayant un impact sur la précision du code peuvent être identifiées par le biais d'études de convergence. Salari & Knupp (2000) présentent une étude de MMS pour laquelle des erreurs ont été délibérément introduites dans le code source d'un programme de CFD éprouvé. Toutes les erreurs affectant la précision des solutions ont pu être détectées par des études de convergence.

3.3 Le *Grid-Convergence Index*

Le *Grid-Convergence Index* (GCI) a d'abord été introduit comme un moyen de comparer sur un même pied toutes les études de convergence (Roache, 1994). Il peut être appliqué autant sur des résultats locaux que sur des normes ou des fonctionnelles obtenues sur différents maillages. Le GCI, comme les méthodes apparentées, est basé sur l'extrapolation de Richardson (Roache, 1998b).

L'extrapolation de Richardson (Richardson & Gaunt, 1927) nous fournit un estimé de la fonction exacte. La différence entre la solution du maillage fin et la solution extrapolée nous donne l'estimateur d'erreur E_1.

$$E_1 = \phi_1 - \phi_{ER} \tag{3.4}$$

[1]On parle ici de la solution exacte du système d'équations algébriques qui découle de la discrétisation de l'EDP, et non de la solution exacte de l'EDP.

La grandeur $|E_1|$ peut être perçue comme un estimateur d'incertitude U_{50}, i.e. la solution a autant de chances de se retrouver à l'intérieur qu'à l'extérieur de cette intervalle. Or, avec le GCI, on vise à obtenir un intervalle de confiance d'environ 95%. Pour y parvenir, $|E_1|$ est majorée par un facteur de sécurité F_s.

$$\text{GCI} = F_s \times |E_1| \tag{3.5}$$

En utilisant trois maillages (ou davantage) pour calculer le GCI, on obtient également une mesure de l'ordre de convergence p de l'algorithme. La détermination de l'ordre de convergence observé est considérée comme une étape importante de la Vérification de Code, car elle permet la comparaison avec l'ordre de convergence théorique. Il s'agit d'une manière de vérifier que le code a bien le comportement prévu. C'est pourquoi l'ordre de convergence observé est évalué de façon systématique sur tous les problèmes étudiés.

Notons toutefois que certains facteurs peuvent contribuer à dégrader l'ordre de convergence observé sans que la qualité de l'algorithme soit mise en cause[2] : manque de similarité géométrique des maillages utilisés ; impossibilité technique d'atteindre la zone de convergence asymptotique ; présence de discontinuités ou de singularités, dans la géométrie ou dans la solution ; utilisation de techniques numériques d'interpolation ou de quadrature ; phénomènes apparaissant seulement à partir d'un certain niveau de raffinement (*multi-scale problems* ; voir DeVolder et al. (2002)).

[2]Voir notamment Oberkampf & Trucano (2002), Eça & Hoekstra (2000, 2002), Pelletier & Roache (2006).

3.3.1 Du facteur de sécurité F_s

Le GCI est proposé comme méthode pratique de quantification de l'incertitude U_{95}. En ce sens, son comportement a été observé sur une vaste étendue de problèmes, où les suppositions nécessaires à l'extrapolation de Richardson (i.e., solution lisse, ordre de convergence observé correspondant à l'ordre théorique de la méthode, convergence monotone de la solution) n'étaient pas nécessairement rencontrées.

La valeur à attribuer au facteur de sécurité F_s fait encore l'objet de recherches intensives. Néanmoins, dans l'état actuel de la littérature, il semble que l'utilisation de $F_s = 1.25$, proposée par Roache (1998b) pour le calcul du GCI à partir de trois maillages ou plus, soit encore aujourd'hui la meilleure façon d'obtenir une estimation de l'intervalle U_{95}. Des études majeures effectuées sur le sujet (Eça & Hoekstra, 2002, Cadafalch et al., 2002) semblent d'ailleurs confirmer le bien-fondé et la robustesse de cette recommandation.

3.3.2 Méthode des moindres carrés (MMC)

La méthode des moindres carrés permet d'identifier une courbe de convergence $E = CN^{-p}$ qui, appliquée à toutes les solutions obtenues, minimise la dispersion des résultats (Eça & Hoekstra, 2003).

Tout d'abord, on doit définir la dispersion des données autour de la courbe de conver-

gence. En utilisant la méthode des moindres carrés[3], on a :

$$S(\phi_{ER}, C, p) = \sqrt{\sum_{i=1}^{n_g} \left(\phi_i - (\phi_{ER} + CN_i^{-p})\right)^2} \tag{3.6}$$

où S est une mesure de l'écart entre les données (N_i, ϕ_i) et la courbe exponentielle définie par ϕ_{ER} (correspondant à l'estimé de la fonction exacte obtenu par la généralisation de l'extrapolation de Richardson), par C (constante de proportionalité), et par l'ordre de convergence observé p. Dans la somme, la valeur n_g correspond au nombre de maillages utilisés.

La dispersion minimale est obtenue en posant les dérivées de S par rapport à ϕ_{ER}, C et p égales à zéro. On obtient donc le système suivant :

$$C = \frac{n_g \sum_{i=1}^{n_g} \phi_i N_i^{-p} - \left(\sum_{i=1}^{n_g} \phi_i\right)\left(\sum_{i=1}^{n_g} N_i^{-p}\right)}{n_g \sum_{i=1}^{n_g} N_i^{-2p} - \left(\sum_{i=1}^{n_g} N_i^{-p}\right)\left(\sum_{i=1}^{n_g} N_i^{-p}\right)} \tag{3.7}$$

$$\phi_{ER} = \frac{\sum_{i=1}^{n_g} \phi_i - C \sum_{i=1}^{n_g} N_i^{-p}}{n_g} \tag{3.8}$$

$$\phi_{ER} \sum_{i=1}^{n_g} N_i^{-p} log(N_i) + C \sum_{i=1}^{n_g} N_i^{-2p} log(N_i) - \sum_{i=1}^{n_g} \phi_i N_i^{-p} log(N_i) = 0 \tag{3.9}$$

Ce système est non-linéaire pour p. On procède à sa résolution en effectuant d'abord un balayage de valeurs de p entre 0 et 10, afin de détecter un changement de signe du résidu de l'équation (3.9). On trouve ensuite la racine par la méthode de la sécante.

[3]La méthode des moindres carrés utilisée ici est en réalité l'une des trois variantes proposées par Eça & Hoekstra (2002). En général, pour toutes les versions, les objectifs généraux, comme les limitations, sont les mêmes. La présente version utilise un seul terme exponentiel pour représenter la courbe de convergence (qui mène à l'obtention d'un unique ordre de convergence observé p), alors que les autres versions cherchent plutôt à quantifier la contribution de plusieurs ordres p_j entiers (ce qui est utile notamment pour des méthodes numériques d'ordre mixte, où, par exemple, certains termes seront évalués à l'ordre 1 et d'autres à l'ordre 2).

L'estimation de la fonction exacte ϕ_{ER} peut alors être utilisée pour calculer un estimé de l'erreur sur le maillage fin. À partir de cette estimation, Eça & Hoekstra (2000) affirment que le calcul d'un GCI avec $F_s = 1.25$ "semble être viable et robuste". Ils notent également que pour des problèmes réalistes, la convergence est rarement monotone, et que, si l'ordre de convergence observé est semblable à l'ordre de convergence théorique pour certains échantillons, il peut en être tout autrement avec des échantillons voisins. Pelletier & Roache (2006) soutiennent que cette situation est représentative du calcul assisté par ordinateur en ingénierie.

La MMC a été développée pour effectuer des analyses sur quatre maillages ou plus. Si on l'applique à seulement trois maillages, la courbe pourra alors passer par les trois points et la dispersion des résultats sera toujours nulle ; on retrouvera ainsi l'estimé de la fonction exacte utilisé pour le calcul du GCI sur un triplet. Les deux variantes peuvent être utilisées conjointement, car d'une part, la MMC permet de rendre compte de la dispersion des résultats autour d'une courbe de convergence, mais ne peut caractériser l'approche de la zone asymptotique de convergence ; d'autre part, en présence de données sur un grand nombre de maillages, le calcul du GCI triplet par triplet devient vite laborieux, mais la comparaison de deux triplets consécutifs, par le ratio R_{GCI}, permet d'apprécier l'approche de la zone asymptotique de convergence. Ce ratio est calculé de la manière suivante (d'après Roache (1998b)) :

$$R_{GCI} = \frac{GCI_2}{GCI_1} \left(\frac{N_2}{N_1}\right)^p \tag{3.10}$$

où p est l'ordre de convergence observé sur le maillage le plus fin (Roache, 1995, Slater, 2005). Le ratio R_{GCI} doit tendre vers l'unité à l'approche de la zone asymptotique de convergence, car la valeur de $|E_i| \times N_i^p$ y est alors constante.

3.3.3 Méthode de Cadafalch et al.

La procédure de post-processing de Cadafalch et al. (2002) permet de recueillir un certain nombre d'informations sur les valeurs locales, souvent difficiles à analyser en raison de leur sensibilité au maillage utilisé. Leur démarche comporte six étapes :

1. **Construction du maillage d'analyse.** On commence par rapporter les solutions obtenues pour trois discrétisations différentes à partir d'un même maillage, dit maillage d'analyse (*post-processing grid*). Ce maillage correspond en général au maillage le plus grossier. Pour les problèmes présentés, aucune interpolation n'est nécessaire pour rapporter les solutions sur le maillage d'analyse, car on utilise un raffinement régulier qui ne fait que subdiviser les cellules existantes (*grid doubling* –ou *grid halving*, selon le point de vue).

2. **Classification des résultats locaux.** Une particularité importante de l'approche de Cadalfach et al. est d'effectuer un certain tri des résultats, afin d'utiliser l'extrapolation de Richardson aux seuls nœuds où elle serait valide. Ainsi, les nœuds sont classés en trois catégories :

$$\text{Noeuds de Richardson :} \quad (\phi_2^* - \phi_3^*) * (\phi_1^* - \phi_2^*) \geq C_0 \tag{3.11}$$

$$\text{Noeuds convergés :} \quad |(\phi_2^* - \phi_3^*) * (\phi_1^* - \phi_2^*)| < C_0 \tag{3.12}$$

$$\text{Noeuds oscillatoires :} \quad (\phi_2^* - \phi_3^*) * (\phi_1^* - \phi_2^*) \leq -C_0 \tag{3.13}$$

où l'astérisque (*) signifie que la solution est normalisée par le plus grand ϕ_i en valeur absolue. Quant à la constante C_0, on choisit pour $C_0 = 10^{-8}$ pour les calculs effectués avec CFDLIB et $C_0 = 10^{-18}$ pour les calculs effectués avec Fluent, ce qui correspond à la précision maximale des valeurs de $|(\phi_2^* - \phi_3^*) * (\phi_1^* - \phi_2^*)|$ traîtées avec chacun de ces programmes[4].

[4]Le calcul de la constante C_0 est effectué par le logiciel Tecplot. Or, les données inscrites dans les fichiers d'output générés par CFDLIB pour Tecplot sont du format 1PE12.5 de Fortran (par exemple :

Cette classification demeure intéressante malgré certains points sur lesquels, de l'avis même de Cadafalch et al. (2002), on doit rester critique. En effet, les nœuds de Richardson peuvent ne pas remplir toutes les conditions nécessaires à l'extrapolation du même nom (notamment lorsque la solution se trouve à l'extérieur de la zone de convergence asymptotique) ; de plus, les nœuds dits *convergés* peuvent être en réalité les points d'inflexion d'une solution oscillatoire. Néanmoins, l'approche peut être appliquée à plusieurs triplets, ce qui permet d'extraire les tendances générales de la solution.

Afin de mener à bien une étude de convergence, la solution doit comprendre une bonne proportion de nœuds de Richardson. Notons que les frontières à valeurs fixes (ou autres valeurs imposées dans le domaine s'il y a lieu) ne doivent pas être considérées[5].

3. **Calcul de l'ordre de convergence local.** Le calcul de l'ordre de convergence observé p est effectué sur les nœuds de Richardson. L'utilisation d'un raffinement régulier $r = h_2/h_1 = h_3/h_2$ permet d'utiliser la formule suivante :

$$p(\mathbf{x}) = \left\{ \ln \frac{\phi_2 - \phi_3}{\phi_1 - \phi_2} \right\} / \ln r \tag{3.14}$$

D'après Cadafalch et al. (2002), la dispersion de l'ordre de convergence autour de sa valeur moyenne sur le domaine peut être utilisée pour analyser l'approche de la zone de convergence asymptotique.

4. **Calcul de l'ordre de convergence global.** Cadafalch et al. (2002) suggèrent que l'ordre de convergence global p soit estimé en prenant la moyenne des $p(\mathbf{x})$ sur un domaine constitué des volumes finis contenant les nœuds de Richardson. En pratique,

$1.2345E-04$). Pour ce type de format, la précision relative sur la différence entre deux valeurs est de 10^{-4}. L'output de Fluent traduit dans des fichiers Tecplot a quant à lui une précision de 10^{-9}.

[5]Dans le cas présent, bien que des vitesses et des températures soient imposées aux frontières, leurs valeurs seront altérées lors du déplacement du stockage des variables du centre des cellules vers les nœuds, lors de la mise en forme de l'output. On observera donc tout de même une variation des valeurs aux frontières dans l'output.

cela n'a de sens que si les nœuds de Richardson sont présents en bonne proportion.

5. **Calcul du GCI local.** Connaissant l'ordre de convergence global, on calcule un GCI local sur les nœuds de Richardson dont l'ordre observé $p(\mathbf{x}) > 0$. L'expression utilisée pour le calcul du GCI est la suivante :

$$GCI(\mathbf{x}) = F_s \left| \frac{\phi_1(\mathbf{x}) - \phi_2(\mathbf{x})}{1 - r^p} \right| \tag{3.15}$$

On utilise $F_s = 1.25$, tel que recommandé dans la littérature (Roache, 1998b) pour un GCI calculé avec un ordre de convergence observé. Notons aussi que le GCI peut être normalisé par la valeur de la solution sur le maillage le plus fin. Enfin, Cadafalch et al. (2002) suggèrent de considérer les nœuds convergés comme ayant un GCI nul. Cette extension du concept du GCI implique que certaines de ses propriétés avancées par Roache (1995) ne tiennent plus, étant donné qu'un GCI nul ne fournit aucune information sur la convergence, si ce n'est que l'erreur de discrétisation est devenue égale ou inférieure à l'erreur d'arrondi. En outre, un GCI nul est incompatible avec l'objectif d'obtenir un estimateur d'erreur U_{SN} qui soit un intervalle U_{95}.

6. **Calcul du GCI global.** Un GCI global peut être évalué en prenant la moyenne des GCI locaux, pondérée par la grandeur de cellule. Cette évaluation ne doit pas tenir compte des nœuds oscillatoires.

CHAPITRE 4

CONTEXTE D'APPLICATION

4.1 Outils de travail

Traitement

Comme CFDLIB n'inclut pas de préprocesseur, les fichiers d'input et les maillages utilisés sont générés au moyens de programmes maison, élaborés spécifiquement pour chacun des problèmes. Ces algorithmes sont axés sur la paramétrisation des maillages, de façon à être efficaces pour procéder aux études de convergence.

Résultats et analyses

On utilise le logiciel Tecplot 360 pour la visualisation des résultats obtenus. Ce logiciel permet le calcul des normes des variables de la solution et sert également à effectuer l'analyse de Cadafalch et al (voir section 3.3.3).

Pour les calculs de la méthode des moindres carrés, on utilise le logiciel Scilab 4.1. Les résultats sont inscrits dans des fichiers textes, de manière à être traités par Gnuplot 4.

4.2 Description des codes utilisés

4.2.1 CFDLIB 3.1

CFDLIB est une bibliothèque de programmes de CFD élaborés pour la résolutions d'écoulements multi-champs. Ces programmes utilisent des schémas de volumes finis pour lesquels les variables d'état sont toutes stockées au centre des cellules. Le domaine est discrétisé par des maillages structurés multi-blocs.

Le projet de vérification est axé sur l'algorithme CCMAC, variante *cell-centered* généralisée de la méthode MAC (Harlow & Welch, 1965), qui requiert la résolution d'une équation de Poisson pour la pression à chaque pas de temps. Le couplage des champs de pression et de quantité de mouvement est effectué au moyen d'un schéma TVD. La précision de l'algorithme peut être réglée à l'ordre 1 ou 2. Dans notre cas, on a choisi le schéma d'ordre 2.

Comme l'objectif est de vérifier le code pour des solutions à l'état permanent, le monitoring de la convergence itérative comprend deux volets. D'une part, l'erreur itérative relative de chaque pas de temps sera amenée à l'ordre de grandeur de l'erreur d'arrondi (1E-16) en imposant une tolérance du plus grand résidu de la méthode du gradient conjugué. D'autre part, la convergence des solutions à l'état permanent est contrôlée par la norme L^{∞} de la dérivée temporelle (obtenue par différence centrée), pour quatre variables de la solution.

4.2.2 Fluent 6.1

Le logiciel Fluent 6.1 est un code de volumes-finis qui stocke les variables calculées au centre des cellules. Le code permet d'utiliser des maillages hybrides.

Par défaut, les équations de conservation sont résolues de façon séquentielle, mais elles peuvent également être couplées. Le code permet de choisir parmi plusieurs schémas de discrétisation, autant pour la pression que pour toutes les équations de transport. Le couplage vitesse-pression peut également être fait selon plusieurs schémas de couplage. Pour la présente étude, on utilise les options de résolutions définies par défaut. Ainsi :

- La résolution est faite de façon séquentielle ;
- Les valeurs de pression aux faces sont obtenues par interpolation, en utilisant les coefficients des équations de mouvement ;
- L'algorithme SIMPLE effectue le couplage de la pression et de la vitesse.
- La discrétisation des équations de mouvement et d'énergie utilise le schéma *Upwind* de premier ordre.

En vertu de ces choix, l'erreur de discrétisation sur les variables de la solution doit théoriquement converger à l'ordre 1.

Enfin, le monitoring de la convergence itérative est effectué au moyen des résidus normalisés de chaque équation de transport.

4.3 Modèle étudié

Les problèmes étudiés pour mettre à l'épreuve les méthodes de vérification proposées s'appliquent aux écoulements satisfaisant les hypothèses suivantes : fluide incompressible, à propriétés constantes, à l'état permanent, en régime laminaire, avec un échauffement visqueux négligeable, dans un domaine bidimensionnel. En appliquant ces hypothèses aux équations de conservation sous forme différentielle, on obtient le système :

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{4.1}$$

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left\{\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right\} \quad (4.2)$$

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\left\{\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right\} \quad (4.3)$$

$$u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \alpha\left\{\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right\} \quad (4.4)$$

Le programme CCMAC effectue la résolutions d'équations plus générales, notamment parce qu'il considère le terme de variation dans le temps, et qu'il permet également la résolutions d'écoulements tridimensionnels. La vérification du modèle complet dépasse toutefois les objectifs fixés pour la présente étude.

4.4 Métriques

La vérification est généralement faite sur les variables de la solution, ainsi que sur des fonctionnelles d'intérêt qui dépendent du problème étudié. Dans notre cas, comme aucune application précise n'est définie *a priori* (notamment parce que les applications de CFDLIB dépendront en partie des conclusions de la présente étude), aucune fonctionnelle n'a été identifiée. La vérification a donc été effectuée directement sur les variables du problème, ainsi que sur des normes de ces variables. On peut ainsi avoir un aperçu global de la convergence de la solution, mais également en constater les différences locales.

Les deux types d'analyses, locale et globale, sont jugés essentiels à la consistance d'une étude de vérification. En effet, l'analyse globale permet d'avoir une vue d'ensemble sur la solution. Il est aussi plus pratique de faire des analyses de convergence sur un grand nombre de maillages lorsqu'on travaille avec des normes plutôt qu'avec des distributions. Toutefois, un mauvais comportement local peut être noyé dans la norme d'une variable et ne pas apparaître dans l'analyse globale. C'est pourquoi l'analyse de convergence de

la distribution des variables de la solution est également essentielle.

CHAPITRE 5

APPLICATION DE LA MÉTHODE DE COMPARAISON À DES SOLUTIONS RECONNUES D'EDO

Le problème choisi pour l'application de la méthode de comparaison à des solutions reconnues d'EDO est la solution de similitude pour le jet laminaire plan. Dans ce chapitre, on retrouve d'abord une description détaillée du problème de référence. On passe par la suite à la résolution numérique. Dans un premier temps, la démarche de vérification sera présentée. Ensuite, le choix des paramètres de la simulation seront justifiés. Puis, des explications sur le déroulement des simulations seront fournies. Enfin, les résultats obtenus seront présentés et analysés.

5.1 Description du problème de référence

5.1.1 Volume de contrôle

Le jet laminaire fait partie des écoulements libres pouvant être modélisés avec la théorie de la couche limite. Le jet consiste en un apport de quantité de mouvement J, provenant d'un fluide expulsé d'un canal très étroit. En effet, pour obtenir une solution de similitude, on doit poser l'hypothèse simplificatrice suivante :

$$\delta \ll L \tag{5.1}$$

Ce qui signifie que la solution n'est valable qu'à une certaine distance de l'embouchure de dimension finie (FIG. 5.1a). De plus, l'origine de la solution de similitude se trouve

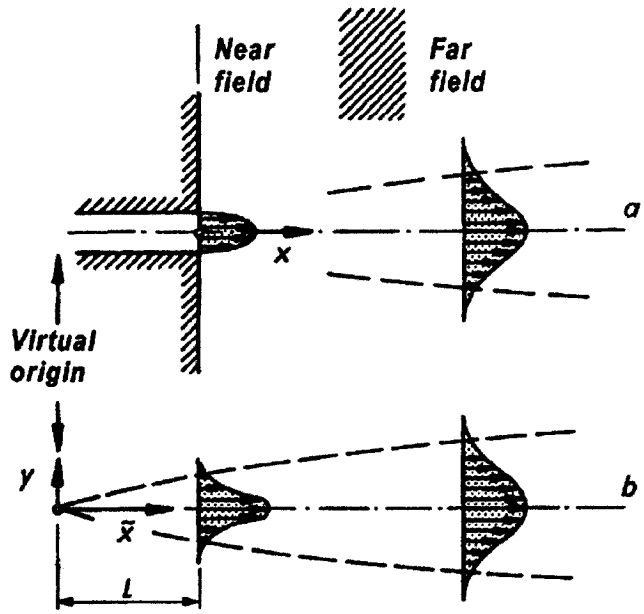


FIG. 5.1 Jet laminaire plan **(a)** provenant d'un embout avec un profil parabolique et **(b)** provenant d'une origine virtuelle fictive (solution de similitude). Tiré de Schlichting & Gersten (2000).

légèrement en aval de l'embouchure de dimension finie (FIG. 5.1b), car la condition (5.1) doit être remplie sur tout le domaine. On constate donc que les conditions d'application de la solution de similitude à l'embouchure du jet ne peuvent être reproduites (numériquement ou expérimentalement), car non seulement l'embouchure serait infiniment petite, mais la vitesse à cet endroit devrait être infinie. Ainsi on obtiendrait une quantité de mouvement de grandeur finie. Pourtant, l'écoulement serait alors turbulent dès son entrée. Pour contourner cette difficulté, on prendra donc un volume de contrôle se situant à une distance suffisament grande de l'embouchure.

5.1.2 Conditions frontières

La solution de similitude a comme conditions frontières une symétrie au centre du jet ($\partial T/\partial y = \partial u/\partial y = v = 0; y = 0$) et une vitesse nulle et une température constante à l'infini ($u = v = 0, T = T_\infty; y \to \infty$).

5.1.3 Solution de référence

À partir du système énoncé aux équations (4.1–4.4), on applique l'argument d'échelle énoncé en (5.1). On obtient alors le système d'équations de la théorie de la couche limite :

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{5.2}$$

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} \approx -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\frac{\partial^2 u}{\partial y^2} \tag{5.3}$$

$$0 \approx \frac{\partial p}{\partial y} \tag{5.4}$$

$$u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} \approx \alpha\frac{\partial^2 T}{\partial y^2} \tag{5.5}$$

Le jet plan permet également d'éliminer la pression, qui prend une valeur constante sur tout le domaine (en raison de l'argument d'échelle et de la constance de la pression à l'infini).

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{5.6}$$

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} \approx \nu\frac{\partial^2 u}{\partial y^2} \tag{5.7}$$

$$u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} \approx \alpha\frac{\partial^2 T}{\partial y^2} \tag{5.8}$$

Ces équations permettent dès lors de faire ressortir une des caractéristiques du jet (Mei,

2002). En effet, en utilisant l'identité $\partial(uv)/\partial y = u\partial v/\partial y + v\partial u/\partial y$, on peut reformuler l'équation de continuité ainsi :

$$v\frac{\partial u}{\partial y} = \frac{\partial(uv)}{\partial y} + u\frac{\partial u}{\partial x} \tag{5.9}$$

On injecte alors ce résultat dans l'équation de quantité de mouvement en x (équation (5.7)), que l'on intègre selon y (en un lieu d'abscisse quelconque). Après réarrangement, on obtient :

$$\int_{-\infty}^{\infty} d(uv) + \frac{d}{dx}\int_{-\infty}^{\infty} u^2 dy = \nu \int_{-\infty}^{\infty} d\left(\frac{du}{dy}\right) \tag{5.10}$$

Puisque la vitesse u est nulle à l'infini, l'équation (5.10) se réduit à

$$\frac{d}{dx}\int_{-\infty}^{\infty} u^2 dy = 0 \tag{5.11}$$

Or, par définition, la quantité de mouvement du jet dans la direction x correspond à :

$$J \equiv \rho \int_{-\infty}^{\infty} u^2 dy \tag{5.12}$$

Ce qui montre que la quantité de mouvement J doit être conservée le long de l'axe x. La quantité de mouvement peut être considérée comme la caractéristique principale du jet. Ce raisonnement est expliqué plus en détail par Mei (2002). En pratique, la quantité de mouvement J peut être reliée à la vitesse moyenne à l'embouchure U_{emb} par l'équation suivante :

$$J = \rho A_{emb} U_{emb}^2 \tag{5.13}$$

où A_{emb} est la grandeur de l'embouchure.

À présent, introduisons les hypothèses de similitude. Ces dernières s'appliquent à la largeur du jet, à la fonction de courant, et à la distribution de température. Historiquement, on a d'abord choisi de supposer une distribution exponentielle de ces variables (dans la direction de l'écoulement). L'hypothèse a été soutenue et largement répandue dans la littérature (notamment par Bickley (1937) et Schlichting (1933)). Soit :

$$\eta = m_\eta \frac{y}{x^{2/3}} \tag{5.14}$$

$$\psi = m_\psi x^{1/3} f(\eta) \tag{5.15}$$

$$T(x,0) = T_\infty + Cx^\gamma \tag{5.16}$$

où m_η et m_ψ sont des constantes arbitraires de mise à l'échelle.

Récemment, la théorie des groupes est venue confirmer l'hypothèse de similitude pour η et pour ψ (voir Mason (2002)). Un résultat semblable pour la température n'est pas encore disponible. Toutefois, certains auteurs (Magyari & Keller, 1999) maintiennent ce point de vue en avançant que toute distribution de température peut être reproduite avec une bonne précision en superposant plusieurs distributions exponentielles. En effet, l'équation d'énergie est linéaire ; on peut donc appliquer le principe de superposition.

$$T(x,0) = T_\infty + \sum_n C_n x^{\gamma_n} \tag{5.17}$$

Dans le cas présent, nous nous contenterons de la distribution donnée par l'équation

(5.16). Pour ce cas, on a démontré (Magyari & Keller, 1999) que la distribution correspondant à la condition adiabatique en $y = 0$ est :

$$\gamma = -1/3 \tag{5.18}$$

Ainsi, à partir de l'équation (5.16) et des conditions frontières, on peut déduire que :

$$T = T_\infty + (T_{jet} - T_\infty)x^{-1/3}\theta(\eta) \tag{5.19}$$

où T_{jet} représente la température en (x_{dep},0).

La solution de similitude fait intervenir la fonction de courant. L'équation de continuité sera donc implicitement respectée. Par définiton :

$$u \equiv \frac{\partial \psi}{\partial y} \qquad v \equiv -\frac{\partial \psi}{\partial x} \tag{5.20}$$

On peut dès lors reformuler l'équation de la quantité de mouvement en x en termes de la fonction de courant. Ainsi :

$$\frac{\partial \psi}{\partial y}\frac{\partial^2 \psi}{\partial x \partial y} + \frac{\partial \psi}{\partial x}\frac{\partial^2 \psi}{\partial y^2} = \nu \frac{\partial^3 \psi}{\partial y^3} \tag{5.21}$$

Lorsqu'on substitue les équations (5.14) et (5.15) dans l'équation (5.21), on trouve la valeur des exposants p et q. Ainsi, l'équation (5.21) peut être réécrite en termes de la fonction de similitude $f(\eta)$.

$$f''' + \frac{m_\psi}{m_\eta}\frac{1}{3\nu}(ff'' + f'f') = 0 \tag{5.22}$$

Par la suite, un choix judicieux des constantes de mise à l'échelle permet d'intégrer l'équation à deux reprises en utilisant les conditions frontières suivantes pour $f(\eta)$:

$$f(0) = f''(0) = \lim_{\eta\to\infty} f'(\eta) = 0 \tag{5.23}$$

On peut ainsi trouver une expression pour $f(\eta)$:

$$f(\eta) = \tanh\eta \tag{5.24}$$

$$\text{avec } m_\eta = \left(\frac{J}{48\rho\nu^2}\right)^{1/3} \text{ et } m_\psi = 6\nu m_\eta$$

Ce qui nous donne les expressions suivantes pour les variables de vitesse :

$$u = m_\eta m_\psi x^{-1/3}\text{sech}^2\eta \tag{5.25}$$

$$v = \frac{1}{3}m_\psi x^{-2/3}\left[2\eta\,\text{sech}^2\eta - \tanh\eta\right] \tag{5.26}$$

Considérons à présent l'équation d'énergie[1]. En substituant les vitesses par les dérivées de la fonction de courant, et en remplaçant la température par la variable adimensionnelle $\theta = (T - T_\infty)\,/\,(T_{jet} - T_\infty)$, on obtient l'équation :

$$\theta'' + 2\text{Pr}\,(\theta' f + f'\theta) = 0 \tag{5.27}$$

[1]On utilise le nombre de Prandtl : $\text{Pr} = \nu/\alpha$.

que l'on peut intégrer deux fois, avec les conditions frontières :

$$\theta'(0) = f(0) = \lim_{\eta \to \infty} \theta(\eta) = 0 \tag{5.28}$$

On obtient donc l'expression de la distribution de température :

$$\theta = \Big(\text{sech}(\eta)\Big)^{2Pr} \tag{5.29}$$

5.2 Stratégie de la Vérification

La solution de similitude au problème du jet laminaire plan est obtenue à la suite d'hypothèses simplificatrices –d'approximations– qui modifient l'équation différentielle initiale. Les résultats ne sont donc pas suffisamment précis pour mener une étude de convergence directe, qui utiliserait la solution de similitude comme solution de référence.

La stratégie consiste donc à obtenir une solution acceptable en utilisant une approche de Vérification des Calculs, en étudiant divers aspects de la convergence de la solution. Puis, les résultats convergés seront comparés à la solution de référence.

Cette méthode permet de comparer les résultats numériques de CFDLIB à des solutions largement documentées dans la littérature. Ainsi, des résultats numériques probants peuvent être comparés à la solution de référence, ce qui peut augmenter le niveau de confiance qu'on leur porte. L'inconvénient est que les différences entre les deux solutions peuvent être attribuées à de multiples causes : les différences entre les modèles (les équations résolues), les géométries, les conditions frontières, etc. De plus, ces divergences sont difficilement quantifiables, même si on effectue les simulations numériques de manière à représenter le plus fidèlement possible la situation décrite par la solution de référence.

5.3 Paramètres de simulation

5.3.1 Unités

On utilise les unités conventionnelles du système international. Le tableau 5.1 indique les unités correspondant aux principales variables à l'étude.

Variables	Unités
x, y	m
t	s
ρ	kg/m^3
p	Pa
T	K
u, v	m/s
ν, α	m^2/s

TAB. 5.1 Unités utilisées

5.3.2 Dimensionnement

Le dimensionnement du problème a été effectué en s'inspirant des valeurs que l'on trouve dans la littérature pour la validation expérimentale de la solution de similitude (Andrade, 1939). Les valeurs sont consignées dans le tableau 5.2.

5.3.3 Domaine

Le domaine de calcul est formé d'une seule zone rectangulaire, de longueur L (dans le sens de l'écoulement du jet) et de largeur H (perpendiculaire au jet), tel qu'illustré à la FIG. 5.2. Notons l'utilisation d'un axe de symétrie afin de réduire le temps de calcul. De plus, on a choisi d'orienter le jet à la verticale, de sorte que le fluide soit entraîné vers le haut. Ainsi, les axes x et y de la solution de similitude sont reliés aux axes X, Y de la

Propriété	Valeur	Description
ρ	998 kg/m^3	Semblable à la densité de l'eau à TPN
U_{emb}	5×10^{-3} m/s	Semblable aux vitesses moyennes à l'embouchure employées lors des expériences de Andrade (1939). Cette valeur sert pour le calcul de la quantité de mouvement du jet (nécessaire au calcul des conditions frontières).
A_{emb}	0.002 m	Dimension de l'embouchure, semblable à celle utilisée dans la littérature Andrade (1939). Utilisé pour le calcul de la quantité de mouvement du jet.
H, L	0.005 m	Voir section 5.3.3
T_∞	273.16 K	Proche de la température normale.
T_{jet}	293.16 K	Choisi pour avoir écart de température relativement faible.
ν	1×10^{-6} m^2/s	Semblable à la viscosité cinématique de l'eau à TPN
α	1.25×10^{-6} m^2/s	Choisi pour avoir $Pr = 0.8$, afin que les couches de cisaillement visqueux soient de même échelle que les couches où il y a gradient thermique.
p_∞	1 Pa	Contraint par CFDLIB

TAB. 5.2 Tableau des paramètres physiques du problème

simulation de la manière suivante :

$$X = -y \tag{5.30}$$

$$Y = x + x_{dep} \tag{5.31}$$

où x_{dep} est la distance entre l'origine virtuelle et l'entrée du domaine de calcul. L'entrée du domaine de calcul coïncide avec le plan perpendiculaire au jet où le débit, Q_{dep}, est égal au débit de référence $U_{emb} \cdot A_{emb}$. Ainsi, on a :

$$x_{dep} = \left(\frac{U_{emb} \cdot A_{emb}}{2m_{\psi}} \right)^3 \tag{5.32}$$

Notons également que pour la discussion qui suit, les variables P, T', U, V font référence au système de coordonnées de la simulation, alors que les variables écrites en minuscules (à l'exception de la température T) seront associées au système d'axes employé lors de la résolution par solution de similitude.

Les dimensions ont dues être ajustées pour que l'écoulement soit semblable à celui décrit par la solution de référence. En effet, des tests ont été effectués avec des domaines de plusieurs tailles différentes. Or, lorsqu'une trop grande largeur était spécifiée, le fluide aspiré de l'infini avait tendance à revenir vers l'embouchure avant d'être entraîné dans l'autre sens par le jet. En d'autres termes, d'après la solution de similitude, la composante u (qui correspond à la vitesse V dans les axes de la simulation) devrait tendre vers 0^+ lorsque y tend vers l'infini. Or, la variable V passait par un minimum négatif au voisinage de la frontière de droite. On a donc choisi un domaine de grandeur intermédiaire : assez large pour permettre de constater l'aspiration du fluide par le jet ; assez étroit pour réduire au maximum la zone où V prend des valeurs négatives, circonscrite à une petite région dans le coin inférieur droit du domaine.

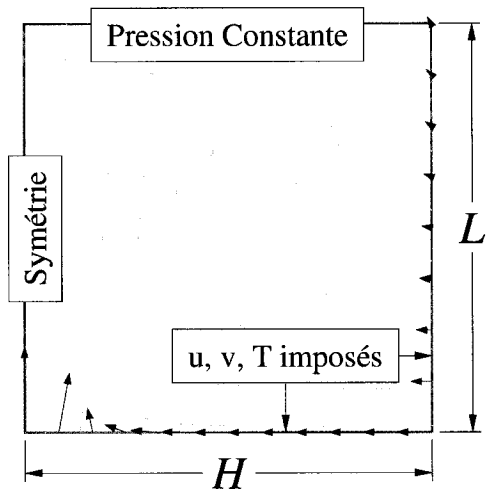


FIG. 5.2 Domaine de calcul

5.3.4 Maillage

Le domaine de calcul est discrétisé au moyen de mailles de dimensions uniformes. En effet, les estimations d'erreur locale ont semblé indiquer que les autres types de concentration testés n'était pas en mesure d'augmenter la précision de façon significative. En outre, le maillage régulier a été utilisé pour simplifier l'analyse de convergence des valeurs locales. En effet, ces maillages permettent (en doublant le raffinement) que les valeurs calculées sur des maillages fins puissent être directement rapportés sur les plus grossiers, sans interpolation.

5.3.5 Conditions initiales

Le problème est initialisé aux conditions du fluide à l'infini. Ces paramètres sont consignés au tableau 5.3.

T	273.16 K
u	0 m/s
ρ	998 kg/m^3

TAB. 5.3 Conditions initiales.

5.3.6 Conditions frontières

Une certaine adaptation des conditions frontières doit être faite à partir des conditions limites de la solution de similitude pour que la CFD reproduise un écoulement semblable à la théorie.

On opte pour apporter une modification à la sous-routine d'imposition des conditions d'*inflow* afin d'imposer comme condition de Dirichlet les valeurs de la solution de similitude sur les frontières sud et est[2]. Cette stratégie permet d'éviter d'avoir recours à une très petite embouchure par laquelle le fluide sortirait à grande vitesse (une telle configuration rendrait l'écoulement turbulent (Schlichting & Gersten, 2000)) ; cela évite aussi d'avoir à utiliser un domaine très long (ce qui augmenterait le temps de calcul). Enfin, comme la solution de similitude a une expression analytique, on peut facilement trouver les valeurs que prennent les vitesses et la température sur les frontières.

Les conditions frontières sont résumées au TAB. 5.4.

[2]En vertu de la solution de similitude, on peut également imposer une pression constante sur la frontière de droite. Toutefois, sur le domaine considéré, le temps de simulation s'en trouve considérablement allongé. En effet, cette configuration provoque la formation de tourbillons en bordure du jet. Ces tourbillons doivent alors être convectés à l'extérieur du domaine, puis la solution doit passer par une phase de stabilisation, avant d'atteindre enfin l'état permanent. L'imposition de la condition d'*inflow* sur la frontière de droite prévient la formation des tourbillons, ce qui accélère l'obtention de l'état permanent.

Néanmoins, on peut toujours éliminer les tourbillons (et conserver la condition de pression à droite) en utilisant un domaine très large et très court ($H \gg L$). Les champs de pressions obtenus sont alors plus continus. Cependant, les temps de calculs plus longs limitent les raffinements.

Une autre raison d'utiliser la condition d'*inflow* sur la frontière de droite est qu'une quantité importante de fluide traversant le domaine provient de cet endroit. Il est donc important de pouvoir en contrôler les caractéristiques (u, v, T), ce que ne permet pas la condition de pression constante, dans la formulation actuelle du programme.

Type	Coté(s)	Valeurs
Pression	haut	$p_\infty = 1$ Pa
Inflow	bas, droite	Solution de similitude
Symétrie (axe)	gauche	N/A

TAB. 5.4 Conditions frontières

5.4 Simulations

5.4.1 Matériel

L'ordinateur utilisé pour les calculs est un PC équipé d'un processeur Pentium IV de 2.4 GHZ. Les calculs ont été lancés alors que le processeur était faiblement occupé.

5.4.2 Convergence des itérations

Au niveau de la convergence des itérations, le critère `ercg` de contrôle de l'erreur sur la méthode du gradient conjugué –fixé au zéro machine– a été atteint pour tous les pas de temps de toutes les simulations. Le pas ayant demandé le plus d'itérations en compte 252 (avec $N = 128$), ce qui signifie que la limite de 999 imposée sur le nombre d'itérations n'a pas été atteinte.

La convergence dans le temps a exigé d'étendre les simulations jusqu'à un temps final d'environ 20 unités. Ce temps d'atteinte de l'état permanent a été déduit des graphiques des dérivées temporelles des variables P, T', U, V (dérivées obtenues par différence centrée). Ces graphes ont également permis de constater que les variations affichées sur les derniers pas de temps n'étaient que des oscillations de la solution.

La figure 5.3 présente un graphe typique des comportements de la convergence temporelle. Ce graphique est obtenu suite à deux opérations : premièrement, on calcule en tout point les dérivées temporelles de toutes les variables (par différence centrée) ; puis, on

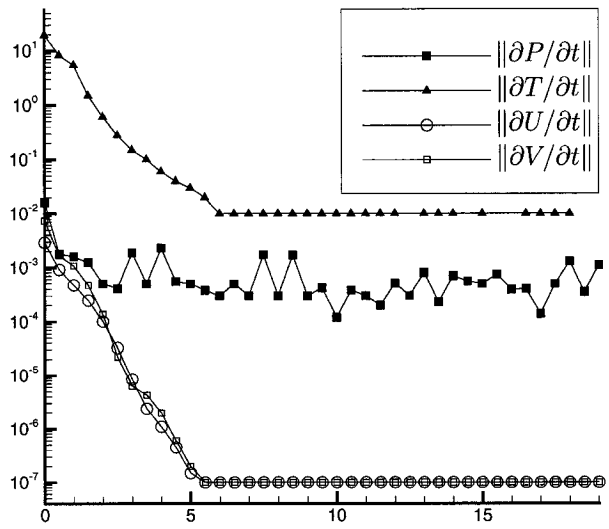


FIG. 5.3 Évolution dans le temps des normes L^{∞} des dérivées temporelles des variables P, T', U et V. Ces valeurs sont calculées d'après la solution pour $N = 128$.

calcule la norme L^{∞} pour chaque variable, à chaque pas de temps. La norme L^{∞} a été choisie car elle permet de mieux capter les petites oscillations qui caractérisent la solution à l'état permanent. En effet, ces oscillations se présentent sous forme de pics très concentrés localement ; ces derniers auraient pu être mal représentés par une norme L^1 ou L^2, par exemple.

On peut remarquer à la FIG. 5.3 que les normes sur $\partial T'/\partial t$, $\partial U/\partial t$ et $\partial V/\partial t$ diminuent jusqu'à atteindre un palier. Ce phénomène se produit lorsque les oscillations temporelles sont environ du même ordre de précision que le plus petit incrément pouvant être appliqué à la variable considérée. De plus, comme on utilise ici une échelle logarithmique, les valeurs nulles n'ont pu être représentées. Cela explique pourquoi il semble manquer des données pour $\partial T'/\partial t$ (comme à $t = 16$). En effet, la dérivée temporelle de T' était nulle en tous points pour ces temps. La norme calculée est donc nulle, et la valeur n'apparaît pas sur le graphique.

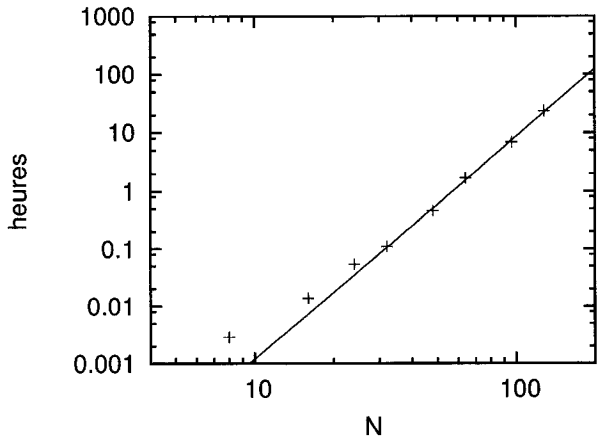


FIG. 5.4 Évolution du temps de calcul en fonction du nombre de cellules.

5.4.3 Statistiques

Pour effectuer l'étude de convergence, 8 simulations ont été retenues en tout, avec des maillages ayant respectivement 8, 16, 24, 32, 48, 64, 96 et 128 cellules dans chaque direction.

5.4.4 Performance

Le temps de calcul est environ proportionnel à N^4. Cela signifie qu'un raffinement faisant diminuer de moitié la taille des cellules aura pour conséquence d'augmenter le temps de calcul d'un facteur de 16. Les temps de calcul sont consignés au TAB. 5.5, et leur évolution est dessinée à la FIG. 5.4.

N	h : min : sec
8	00 : 00 : 10 . 28
16	00 : 00 : 50 . 14
24	00 : 03 : 13 . 67
32	00 : 06 : 38 . 65
48	00 : 27 : 36 . 12
64	01 : 40 : 23 . 97
96	06 : 52 : 06 . 03
128	23 : 11 : 04 . 86

TAB. 5.5 Temps de calcul en fonction du nombre de cellules effectif.

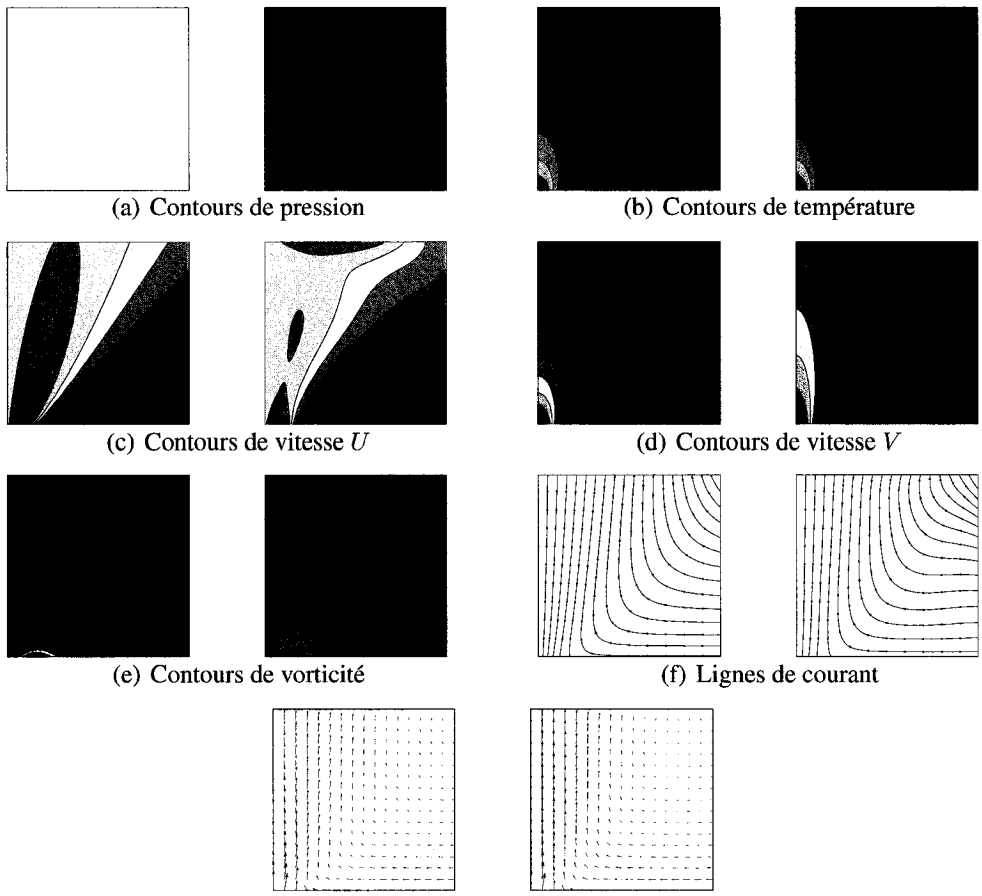
5.5 Résultats et analyse

5.5.1 Résultats

La distribution des quatres inconnues P, T', U et V, celle de la vorticité, de même que les lignes de courant, sont présentées à la FIG. 5.5. Pour chaque variable, le graphique de gauche présente la solution de similitude, et le graphique de droite la solution obtenue numériquement par CFDLIB[3]. Sur l'ensemble des graphiques présentés, on constate que la plupart présentent des différences sur l'allure des contours. En effet, seuls les graphiques de température semblent présenter environ les mêmes résultats. Sur les autres graphiques on remarque des différences, qui se manifestent à différents endroits selon la variable observée.

La pression obtenue par la solution de similitude est constante sur tout le domaine. La solution numérique présente une distribution très différente. Sur la majeure partie du domaine, les gradients de pression sont assez faibles, mais non nuls. Près de la frontière inférieure, vers la gauche du domaine, la pression croît rapidement. Notamment, entre la deuxième et la troisième cellule (à partir du bas), le gradient de

[3]Les graphiques de résultats des FIG. 5.5 et 5.6 correspondent à la solution sur le maillage le plus fin ($N = 128$).



(a) Contours de pression

(b) Contours de température

(c) Contours de vitesse U

(d) Contours de vitesse V

(e) Contours de vorticité

(f) Lignes de courant

(g) Vecteurs de vitesse

FIG. 5.5 Jet laminaire plan : comparaison entre la solution de référence et la solution obtenue par l'algorithme 2D MAC de CFDLIB. Pour chaque paire de graphiques, la solution de similitude est présentée à gauche, et la solution numérique se trouve à droite.

pression selon Y connaît une discontinuité. À l'autre extrémité on observe une isobare qui semble intersecter la frontière supérieure (où l'on a pourtant imposé une pression constante). Il s'agit en réalité d'une discontinuité entre les pressions de part et d'autre des cellules contiguës à cette frontière. Sur la FIG. 5.6 on peut voir la discontinuité, et également constater que la pression est belle et bien constante sur toute la frontière supérieure. Sur la même figure, on voit également comment la pression varie sur les autres faces.

La température est assez semblable pour les deux solutions. Les isothermes sont environs aux mêmes endroits et leurs formes sont, dans l'ensemble, assez proches. On note toutefois que les profils de température calculés par CFDLIB ne sont pas tout à fait similaires, tel qu'on peut le constater d'après l'isotherme se terminant proche du coin supérieur droit du domaine.

La vitesse U, perpendiculaire à la direction du jet, n'a pas la même distribution pour les deux solutions. Bien que les deux distributions de U soient assez proches l'une de l'autre pour que l'allure de l'écoulement soit semblable dans les deux cas, on note des différences importantes, et ce à plusieurs endroits :

1. La solution numérique possède un maximum local situé à l'intérieur du domaine. Ceci est un signe que les profils de l'écoulement décrits ne sont pas similaires.
2. La vitesse U calculée par CFDLIB recommence à croître à l'approche de la frontière supérieure. Encore une fois, cela signifie que les profils ne sont pas similaires.
3. Les lignes d'iso-vitesse obtenues par le calcul numérique et situées dans la région de droite du domaine présentent un point d'inflexion, contrairement à celles obtenues par solution de similitude. Cet effet s'accentue si on élargit le domaine.

   On comprend mieux le phénomène en observant le comportement des lignes

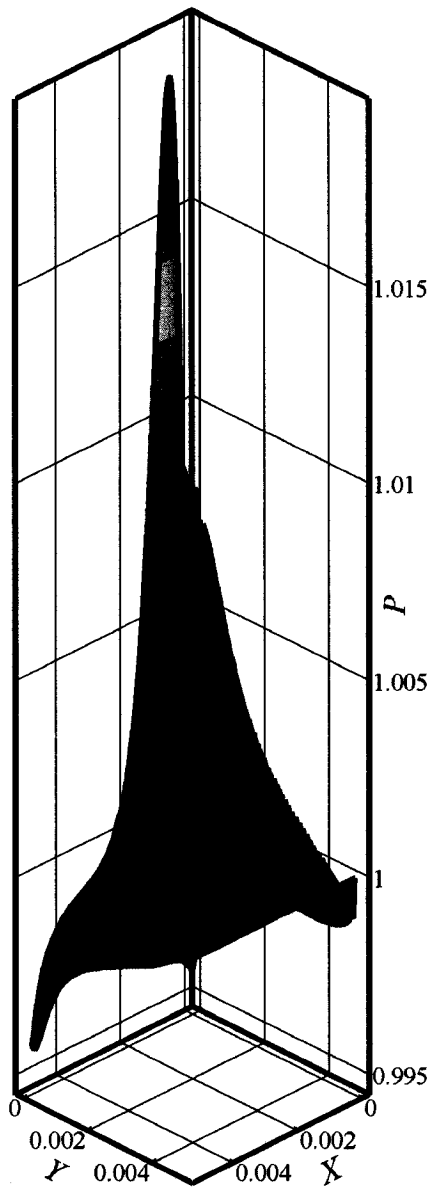


FIG. 5.6 Distribution de pression (obtenue avec un maillage de 128×128 cellules)

de courant. Dans le cas de la similitude, le seul endroit où la vitesse est complètement perpendiculaire au jet se trouve à l'infini (Les lignes de courant y sont parallèle à x). Or, pour la solution numérique, le fluide passe par une région où la vitesse V est négative, après quoi le vecteur vitesse doit redevenir horizontal pour changer de direction. En passant par l'horizontale, la vitesse U est augmentée momentanément, au détriment de V.

La vitesse V se présente comme étant assez semblable dans les deux situations. On note toutefois que les gradients sont plus forts dans la solution numérique (étant donné que les lignes d'iso-vitesse sont plus rapprochées que dans l'autre cas). Dans l'ensemble, les profils décrits numériquement sont assez proches de la similarité. Excluons cependant deux endroits : le côté inférieur droit du domaine, où la composante V prend une valeur légèrement négative ; ainsi qu'à proximité de la frontière supérieure, où certaines lignes de vitesse constante possèdent un point d'inflexion (surtout celles situées les plus à droite).

La vorticité permet de revenir sur les principales différences constatées entre les deux solutions. En effet, bien que la forme de cloche que présentent les lignes de vorticité constante soit assez semblable dans les deux cas, on constate que la distribution de la solution numérique diffère considérablement à l'approche de la frontière supérieure, ainsi que dans le coin inférieur droit. Les lieux où ces différences surviennent ont déjà été relevés par observation des graphiques de vitesse.

Dans l'ensemble, les deux solutions présentent les mêmes caractéristiques physiques. Cela vient du fait que les deux systèmes d'équation ont la capacité de décrire correctement les mêmes phénomènes. Cependant, en raison de leur formulation différente, on peut s'attendre à ce que leur solutions exactes ne soient pas identiques. Ceci pourrait expliquer pourquoi l'on observe des distributions légèrement différentes.

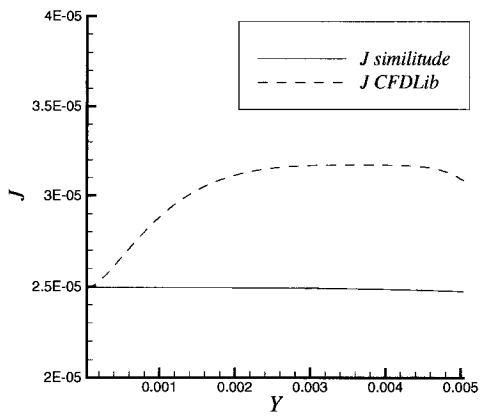


FIG. 5.7 Évolution de la quantité de mouvement en Y du jet.

La quantitité de mouvement J

La section 5.1.3 a mis en lumière la propriété principale du jet : la quantité de mouvement dans la direction du jet, qui est conservée peu importe la distance à l'orifice. On peut constater à la FIG. 5.7 que la quantité de mouvement en Y pour la solution de similitude sur le domaine de calcul est pratiquement constante, du moins pour les régions proches de l'embouchure. Plus loin en aval, la quantité de mouvement sur le domaine diminue légèrement car une plus grande partie de la quantité de mouvement du jet se trouve à avoir été diffusée à l'extérieur du domaine de calcul.

La situation est différente pour la solution numérique. Les valeurs de V sont plus grandes au centre du jet, ce qui, pour un même débit, nous donnera une quantité de mouvement plus grande. Ainsi, on remarque une stabilisation de la quantité de mouvement, vers le milieu du domaine, à une valeur plus élevée que pour la solution de similitude. D'autre part, comme l'écoulement à $Y = 0$ est le même que pour la solution de similitude, la quantité de mouvement J y est également identique.

Ainsi, on peut constater que la caractéristique de constance de la quantité de mouvement,

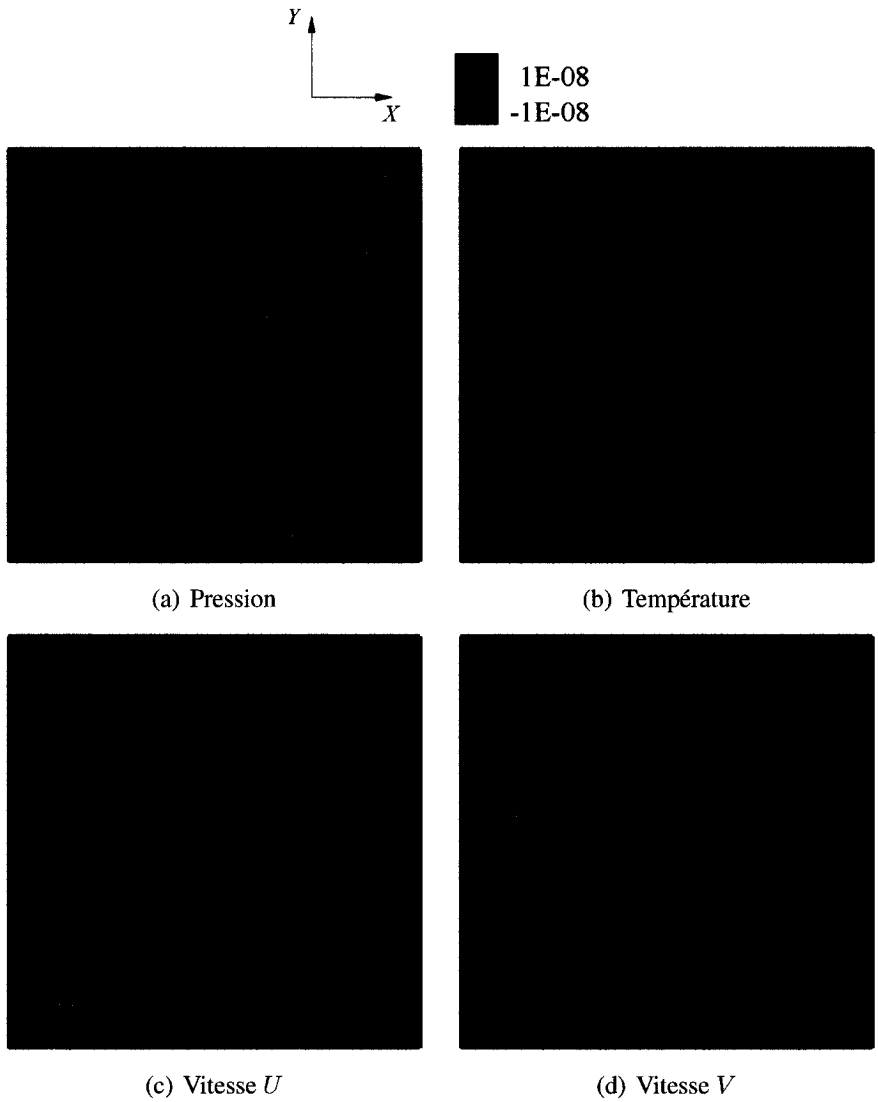
mise en évidence par la solution de similitude, se manifeste pour la solution numérique uniquement sur la partie centrale du domaine.

5.5.2 Convergence de la discrétisation

5.5.2.1 Application de la méthode de Cadafalch et al.

Une étude de convergence locale, couvrant l'ensemble du domaine de calcul, permet d'identifier des configurations ou des phénomènes avec lesquels le code éprouve des difficultés. Les résultats présentés sur les FIG. 5.8 et 5.9 proviennent du triplet 32–64–128. L'étude de convergence des valeurs locales a toutefois également été effectuée avec les triplets 16–32–64 et 24–48–96.

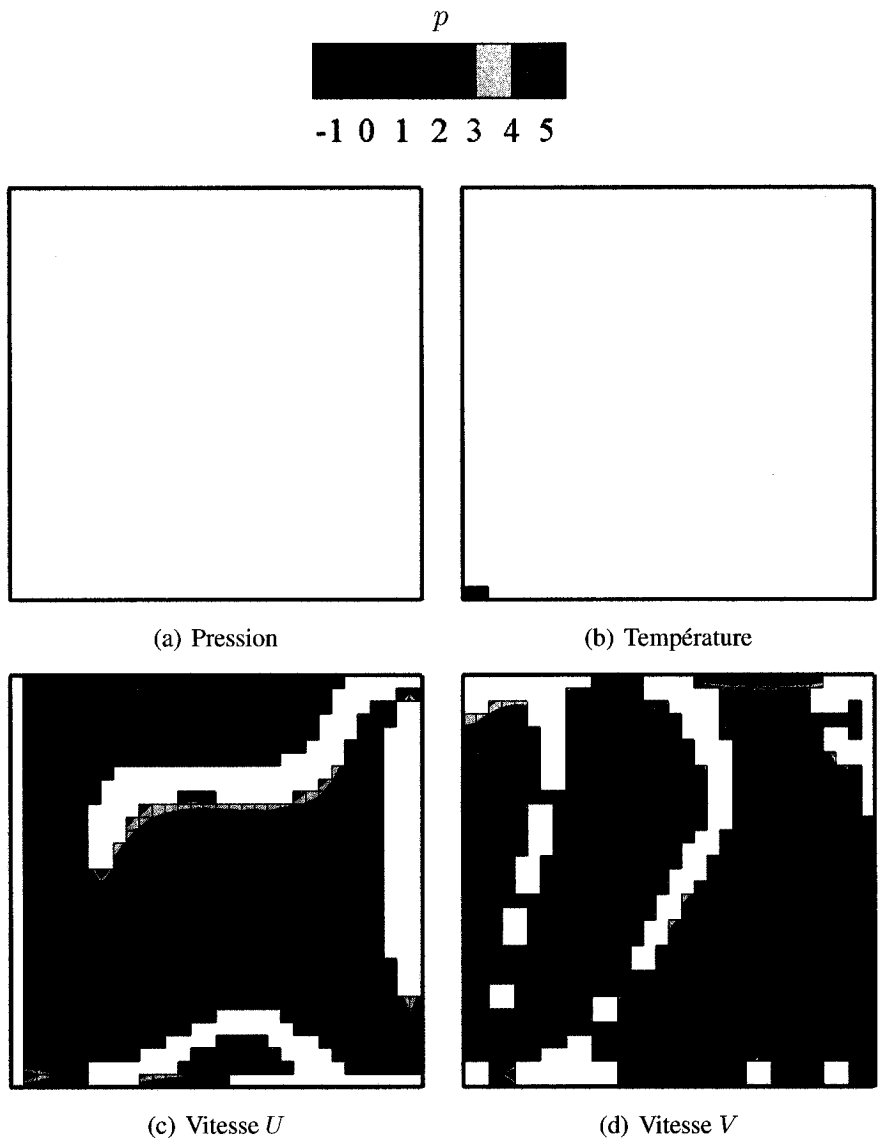
Tel que décrit à la section 3.3.3, on commence par observer le comportement de la convergence des valeurs locales au moyen du produit $((\phi_2^* - \phi_3^*) * (\phi_1^* - \phi_2^*))$. Sa distribution est présentée à la FIG. 5.8. Puis on poursuit avec l'ordre de convergence local (voir FIG. 5.9), lequel est intégré sur le sous-domaine Ω_R, qui exclue du domaine de calcul Ω les cellules comprennant des nœuds convergés ou des nœuds oscillatoires. On obtient ainsi l'ordre de convergence global. Sur le même domaine Ω_R, on calcule également une mesure de la dispersion de $p(\mathbf{x})$, en prenant la norme $\|p(\mathbf{x}) - \overline{p}\|_{L^2(\Omega_R)}$, notée σ_p. On utilise ensuite cet ordre de convergence global ($\|p(\mathbf{x})\|_{L^1(\Omega_R)}$, noté $\overline{p}$) pour calculer le GCI local. Le GCI local pour U et pour V est présenté à la FIG. 5.10. Sur cette figure, le domaine représenté correspond à l'ensemble des cellules ne comprenant aucun nœud oscillatoire ($\Omega_{RC} = \Omega_R \cup \Omega_C$). C'est d'ailleurs ce domaine qui est utilisé pour le calcul du GCI global ($\|GCI\|_{L^1(\Omega_{RC})}$, noté $\overline{GCI}$). Le sommaire de l'étude de convergence provenant de la méthodologie de Cadafalch et al. (2002) est présenté dans les TAB. 5.6, 5.7 et 5.8.



(a) Pression (b) Température

(c) Vitesse U (d) Vitesse V

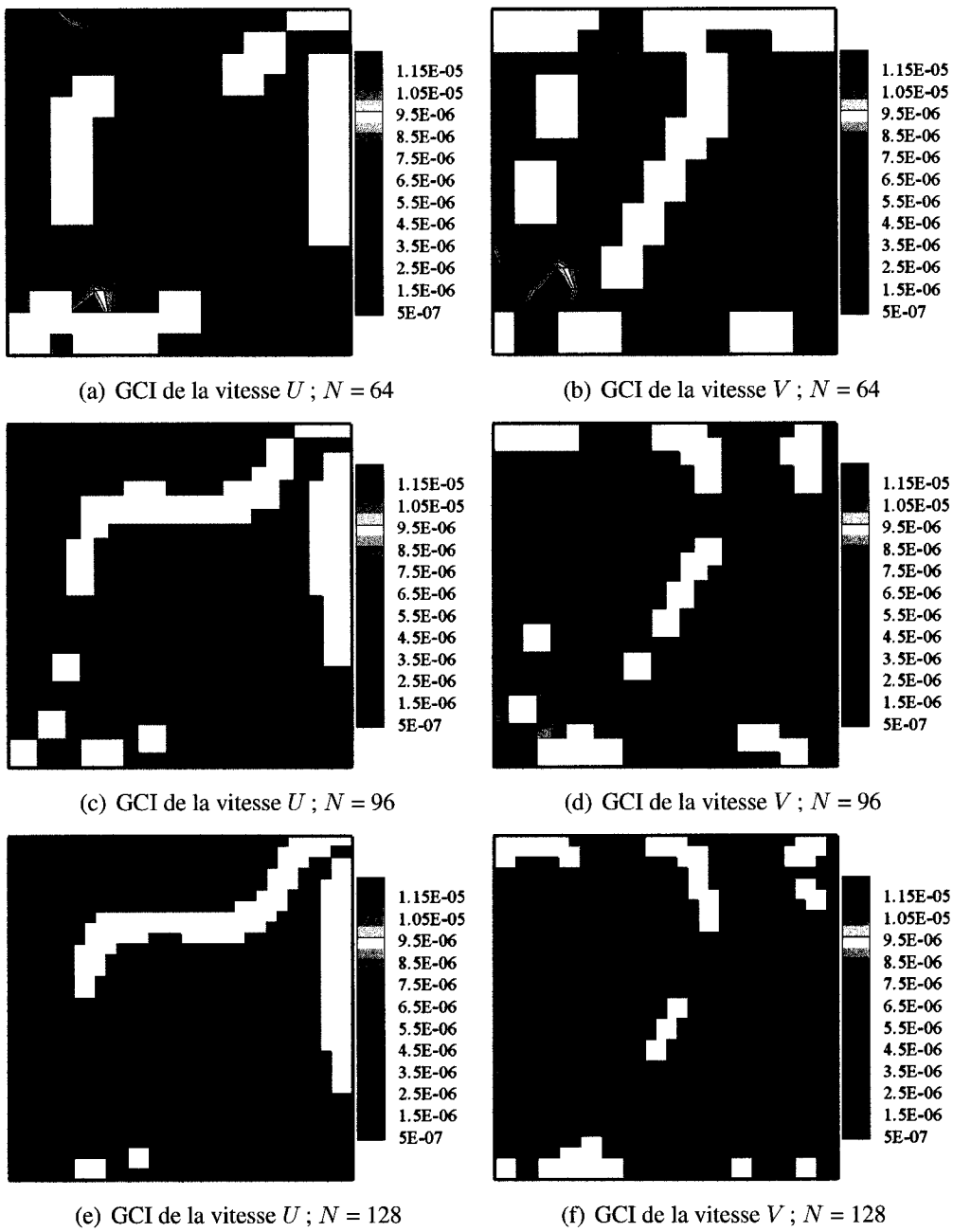
FIG. 5.8 Distribution de $((\phi_2^* - \phi_3^*) * (\phi_1^* - \phi_2^*))$ pour les 4 variables à l'étude (N = 128).



(a) Pression (b) Température

(c) Vitesse U (d) Vitesse V

FIG. 5.9 Ordre de convergence local aur les nœuds de Richardson, pour les quatre variables à l'étude ($N = 128$).



(a) GCI de la vitesse U ; $N = 64$

(b) GCI de la vitesse V ; $N = 64$

(c) GCI de la vitesse U ; $N = 96$

(d) GCI de la vitesse V ; $N = 96$

(e) GCI de la vitesse U ; $N = 128$

(f) GCI de la vitesse V ; $N = 128$

FIG. 5.10 GCI absolus locaux pour les deux composantes de vitesse, d'après la méthode de Cadafalch et al. (2002).

On remarque alors que :

1. **La pression** est convergée presque partout sur le domaine. On voit aussi que les discontinuités relevées à la section 5.5.1 nuisent à la convergence près de l'entrée et de la sortie du domaine. On y trouve des nœuds oscillatoires, mais également des nœuds de Richardson. Ces derniers ne sont toutefois pas regroupés de manière à former des cellules. Les ordres de convergence observés sur ces nœuds de Richardson sont proches de l'unité, et relativement peu dispersés (compris entre -1 et 3.5). Malgré tout, comme le sous-domaine Ω_R n'existe pas pour la variable P, on ne peut continuer d'avantage l'analyse de convergence locale. Ainsi, pour la pression, on peut difficilement se prononcer sur la qualité de la convergence, principalement en raison de la parité entre l'erreur de discrétisation et l'erreur d'arrondi.

2. **La température** est également convergée sur la quasi-totalité du domaine. On peut en déduire que l'erreur de discrétisation y serait de l'ordre de l'erreur d'arrondi, ou plus petite. Cela nous empêche à nouveau d'effectuer l'extrapolation de Richardson en suffisament de points pour que les valeurs trouvées soient représentatives de ce qui se passe sur tout le domaine. Il ne reste qu'une petite région à l'embouchure, où les variations de températures (de maillage en maillage) sont assez grandes pour évaluer un ordre de convergence observé. Ce dernier est proche de l'ordre de convergence théorique, et ce pour les trois triplets utilisés. Bien que cette cohérence avec la théorie ne peut être assumée comme étant aussi bonne sur tout le domaine, on peut dire qu'aucun mauvais comportement en convergence n'est observé. Cela ne garantit pas la consistance du calcul, mais elle ne peut non plus être exclue.

   En outre, les GCI calculés ne doivent pas être utilisés comme estimateurs U_{95} de l'incertitude du calcul numérique, car leur valeur est imposée à zéro sur les nœuds convergés, et que ces nœuds couvrent la très grande majorité du domaine. L'arrondissement des valeurs pour la création des fichiers de sortie est donc la source

Triplet	16–32–64	24–48–96	32–64–128
$\|T\|_{L^1(\Omega_{RC})}$	277.38	277.38	277.38
$\overline{GCI}$	8.1 E-4	1.8 E-4	5.0 E-5
$\overline{p}$	2.12	2.19	2.21
σ_p	0.39	0.36	0.09
Région de Richardson (Ω_R)	3.5 %	0.9 %	0.2 %
Région convergée (Ω_C)	96.5 %	99.1 %	99.8 %
Région oscillatoire (Ω_O)	0.0 %	0.0 %	0.0 %

TAB. 5.6 Caractéristiques de convergence de la température T', d'après l'analyse proposée par Cadafalch et al. (2002).

d'erreur dominante. Ainsi, l'incertitude U_{SN} devrait être légèrement supérieure à l'erreur d'arrondi, dont la valeur est de 10^{-2} car le format pour l'output ne contient que cinq chiffres significatifs.

3. La vitesse U est la variable qui, toute proportion gardée, subit les variations les plus marquées lors des raffinements. Ainsi, puisque les raffinements produisent une variation notable sur une grande partie des nœuds, la région Ω_R couvre la grande majorité du domaine. Sur la figure 5.8, on trouve seulement deux bandes de cellules oscillatoires entourées de quelques nœuds convergés. Pour les trois triplets utilisés, ces configurations sont présentes. De plus, on note une légère augmentation de la proportion de nœuds de Richardson et de nœuds convergés lors des raffinements, au détriment des nœuds oscillatoires.

Sur la région Ω_R, les ordres de convergence locaux observés sont en général très près de 2 et sont relativement peu dispersés. De plus, la région Ω_R est très étendue, alors l'ordre de convergence moyen observé est représentatif du comportement de la solution en convergence.

La distribution du GCI local montre que les régions présentant les plus grandes incertitudes sont à l'entrée ainsi qu'en sortie. On voit aussi à la FIG. 5.10 que le GCI présente une allure semblable peu importe le triplet utilisé. Enfin, on remarque que les deux plus grandes régions oscillatoires de Ω_O se trouvent aux endroits où l'incertitude – ou la variation présentée lors du raffinement – est la plus faible. On a

Triplet	16–32–64	24–48–96	32–64–128
$\|U\|_{L^1(\Omega_{RC})}$	6.105 E-4	6.077 E-4	6.067 E-4
$\overline{GCI}$	2.7 E-6	1.4 E-6	8.9 E-7
$\overline{p}$	1.75	1.80	1.77
σ_p	0.81	0.73	0.77
Région de Richardson (Ω_R)	67.2 %	72.4 %	76.3 %
Région convergée (Ω_C)	8.6 %	7.8 %	8.3 %
Région oscillatoire (Ω_O)	24.2 %	19.8 %	15.4 %

TAB. 5.7 Caractéristiques de convergence de la vitesse U, d'après l'analyse proposée par Cadafalch et al. (2002).

Triplet	16-32-64	24–48–96	32–64–128
$\|V\|_{L^1(\Omega_{RC})}$	1.6773 E-3	1.6776 E-3	1.6778 E-3
$\overline{GCI}$	2.4 E-6	1.1 E-6	6.4 E-7
$\overline{p}$	1.97	2.00	1.92
σ_p	0.94	0.66	0.60
Région de Richardson (Ω_R)	70.3 %	80.4 %	81.8 %
Région convergée (Ω_C)	1.2 %	3.8 %	7.6 %
Région oscillatoire (Ω_O)	28.5 %	15.8 %	10.6 %

TAB. 5.8 Caractéristiques de convergence de la vitesse V, d'après l'analyse proposée par Cadafalch et al. (2002).

tout lieu de croire que ces oscillations se produisent surtout à l'intersection de la solution numérique et de la solution exacte[4]. Il serait donc normal que la solution numérique soit très sensible à la discrétisation en ces endroits, et donc que l'on observe ces oscillations.

4. La vitesse V varie peu lors des raffinements, mais suffisament pour pouvoir y extraire des caractéristiques sur sa convergence. En effet, la plus grande partie du domaine est couverte de nœuds de Richardson. Les nœuds oscillatoires et les noeus convergés, que l'on retrouve en partie à l'entrée et à la sortie, sont également présents le long des intersections entre la solution numérique et la solution extrapolée.

L'ordre de convergence observé est près de l'ordre de convergence théorique sur

[4]En l'absence de solution exacte, ou de preuves suffisantes que le code converge vers la solution exacte, cette affirmation reste incertaine. Toutefois, on peut montrer que la solution numérique, peu importe le maillage, croise la solution extrapolée dans ces régions.

la quasi totalité du domaine Ω_R. Ceci est compatible avec le fait que la solution pour V varie doucement et ne connaît pas de discontinuités. Ainsi, on constate que le $\overline{p}$ observé est très près de 2 et que la dispersion est faible. De plus, la dispersion tend à diminuer plus le triplet de calcul utilise des maillages fins.

Enfin, la distribution du GCI connaît son maximum à l'entrée, où les gradients sont les plus forts. Le GCI diminue par la suite plus on avance dans le domaine. Sa distribution se divise alors en trois bandes (signe des croisements entre la solution numérique et la solution extrapolée). Quant au GCI moyen sur Ω_{RC}, il est assez représentatif de l'incertitude sur l'ensemble du domaine, parce que la plupart des cellules sont formées de nœuds de Richardson, et que le domaine contient peu de nœuds convergés.

Dans l'ensemble, on obtient donc des résultats qui suggèrent le bon comportement du code, pour les valeurs qui varient suffisament lors des raffinement. Pour U et V, on obtient, à l'aide de la distribution du GCI, un estimé de la distribution de l'incertitude sur le domaine.

Cet estimé peut maintenant être confronté à la différence entre la solution numérique et la solution de similitude, proposée plus tôt comme estimateur d'erreur. On constate que les distributions présentées à la FIG. 5.11 diffèrent grandement des distributions de l'incertitude sur U et V de la FIG. 5.10. Et non seulement les lieux des crêtes et des creux des distributions diffèrent, mais on constate également que les échelles en jeu sont à des ordres de grandeur de différence. Comme l'ordre de grandeur de la différence entre les solutions des deux modèles est beaucoup plus grand que le GCI, et que la solution numérique présente un bon comportement en convergence (ce qui aurait permis d'utiliser le GCI comme bande d'incertitude, si le code avait été vérifié auparavant), on doit remettre en question la méthode de vérification. D'autant plus que l'on ne doit pas s'appuyer sur la solution numérique pour évaluer la fonction exacte, et que l'on ne connaît pas l'écart entre la solution de similitude et la solution exacte de l'EDP. Dans ces condi-

tions, la comparaison avec le modèle simplifié ne peut fournir de preuves permettant de conclure au bon fonctionnement du code.

Ces conclusions seront discutées plus en détail à la section 5.5.3.

5.5.2.2 Convergence des Normes L^1 et L^2

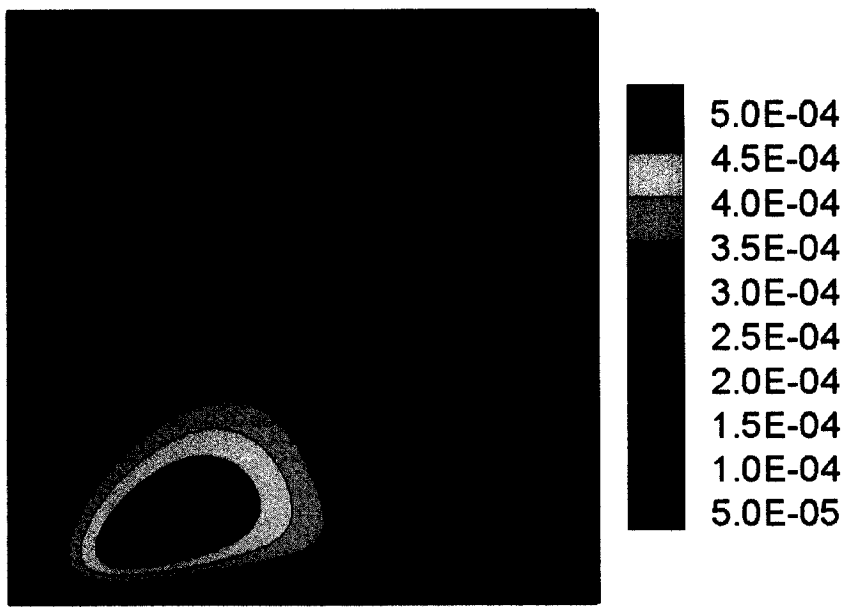
La méthode du GCI, telle que présentée par Roache (1998b), ainsi que la méthode des moindres carrés (Eça & Hoekstra, 2002), conviennent bien aux études de convergence de fonctionnelles ou de valeurs globales. Les valeurs globales choisies pour la présente étude de convergence sont les normes L^1 et L^2.

Pour fins de comparaisons, les normes L^1 et L^2 des variables T', U et V de la solution de similitude ont été calculées sur un domaine correspondant au domaine de calcul. Ces valeurs sont présentées au TAB. 5.9, à la suite des normes calculées sur les solutions numériques.

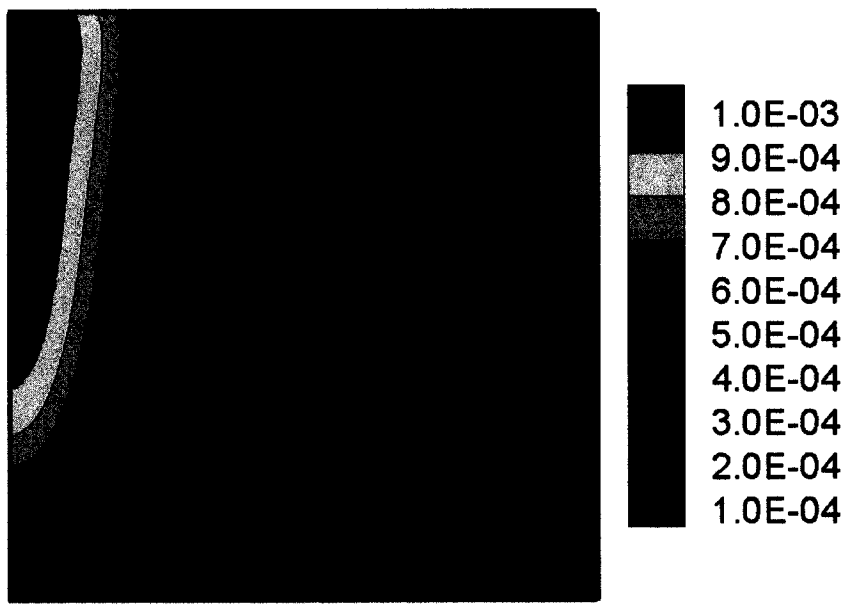
Une inspection rapide du TAB. 5.9 permet déjà de constater que les solutions numériques ne convergent pas vers la solution de similitude. Ce constat se retrouve dans les résultats des deux méthodes présentées ci-bas.

Méthode du GCI utilisant des triplets de solutions

La méthode du GCI est appliquée à tous les triplets formés par des maillages successifs. Les autres combinaisons (par exemple, le triplet 8 – 24 – 64) ont été ignorées. Les TAB. 5.10, 5.11 et 5.12 présentent les résultats obtenus pour tous les triplets menant à un ordre de convergence observé compris entre 0 et 10. Aucun résultat n'est présenté pour la pression car aucun triplet ne correspond à ce critère, ceci étant dû à une variation trop



(a) $U_{CFD} - U_{similitude}$



(b) $V_{CFD} - V_{similitude}$

FIG. 5.11 Graphique de la différence entre la solution numérique et la solution de similitude pout U et V.

N	$\|P\|$	$\|T\|$	$\|U\|$	$\|V\|$
(Norme L^1)				
8	1.0004	277.25	6.3211 E-04	1.6618 E-03
16	1.0001	277.36	6.1961 E-04	1.6739 E-03
24	1.0001	277.37	6.1322 E-04	1.6763 E-03
32	1.0001	277.38	6.1007 E-04	1.6770 E-03
48	1.0001	277.38	6.0771 E-04	1.6775 E-03
64	1.0001	277.38	6.0681 E-04	1.6777 E-03
96	0.99957	277.38	6.0609 E-04	1.6778 E-03
128	1.0001	277.38	6.0581 E-04	1.6778 E-03
Similitude	1.0000	278.15	5.5467 E-04	1.6645 E-03
(Norme L^2)				
8	1.0004	277.27	9.5433 E-04	2.3897 E-03
16	1.0001	277.38	9.2774 E-04	2.4600 E-03
24	1.0001	277.40	9.2021 E-04	2.4660 E-03
32	1.0001	277.40	9.1699 E-04	2.4664 E-03
48	1.0001	277.40	9.1467 E-04	2.4668 E-03
64	1.0001	277.40	9.1379 E-04	2.4667 E-03
96	0.99957	277.40	9.1315 E-04	2.4667 E-03
128	1.0001	277.40	9.1291 E-04	2.4668 E-03
Similitude	1.0000	278.17	8.3331 E-04	2.2342 E-03

TAB. 5.9 Normes L^1 et L^2.

Triplet	p	$\|T\|$	GCI	R_{GCI}
(Norme L^1)				
8 - 16 - 24	3.16	277.37	0.0048	N/A
(Norme L^2)				
8 - 16 - 24	2.03	277.40	0.02	N/A

TAB. 5.10 Convergence des normes L^1 et L^2 de T, d'après la méthode du GCI Roache (1995).

faible pour être observée.

La température connaît un problème semblable pour la majorité des triplets. Pour la norme L^2, seul le premier triplet $8-16-24$ présente une variation pour chaque maillage. La situation est semblable pour la norme L^1, à l'exception du deuxième triplet, où seule la dernière décimale varie d'une seule unité, ce qui correspond à un ordre de convergence observé nul. Pour le triplet $8-16-24$, l'ordre de convergence en température est relativement proche de l'ordre de convergence théorique. Ainsi, dans les limites de ce que l'on peut mesurer, le comportement du code est bon. Le GCI est quant à lui proche de l'erreur d'arrondi. En l'absence d'un deuxième triplet avec lequel on puisse obtenir des résultats, on ne peut se prononcer sur l'approche de la zone asymptotique de convergence pour la température.

La vitesse U subit des variations notables lors des raffinements. Pour les deux normes à l'étude, on voit que l'ordre de convergence observé croît lors des raffinements, pour se stabiliser avec les derniers triplets ; à environ 1.7 pour la norme L^1, et autour de 1.9 pour la norme L^2. Parallèlement, on voit la valeur de R_{GCI} s'approcher de 1. Enfin, on peut voir que la valeur du GCI décroit à mesure que le maillage est raffiné.

Avec la vitesse V, on voit l'erreur de discrétisation diminuer lors des raffinements et rejoindre l'ordre de grandeur de l'erreur d'arrondi. Ainsi, la totalité des triplets ne peuvent être utilisés. Aussi, bien que l'on en constate la convergence, l'analyse est perturbée en raison de l'importance de l'erreur d'arrondi dans la solution présentée.

Triplet	p	$\|\|U\|\|$	GCI	R_{GCI}
(Norme L^1)				
8 - 16 - 24	0.24	6.1322 E-04	7.7 E-05	N/A
16 - 24 - 32	1.04	6.1007 E-04	1.1 E-05	5.0457
24 - 32 - 48	1.86	6.0771 E-04	2.6 E-06	2.0243
32 - 48 - 64	1.76	6.0681 E-04	1.7 E-06	0.9280
48 - 64 - 96	1.66	6.0609 E-04	9.4 E-07	0.9286
64 - 96 -128	1.71	6.0581 E-04	5.5 E-07	1.0380
(Norme L^2)				
8 - 16 - 24	1.28	9.2021 E-04	1.4 E-05	N/A
16 - 24 - 32	1.44	9.1699 E-04	7.8 E-06	1.1644
24 - 32 - 48	1.97	9.1467 E-04	2.4 E-06	1.4887
32 - 48 - 64	1.78	9.1379 E-04	1.6 E-06	0.8600
48 - 64 - 96	1.95	9.1315 E-04	6.7 E-07	1.1251
64 - 96 -128	1.93	9.1292 E-04	3.9 E-07	0.9856

TAB. 5.11 Convergence des normes L^1 et L^2 de U, d'après la méthode du GCI Roache (1995).

Les résultats en convergence pour V présentent une certaine dispersion. L'ordre de convergence observé oscille autour de 2 (du moins, c'est le cas pour la norme L^1 ; il est difficile de se prononcer pour ce qui est de la norme L^2, en raison du petit nombre de résultats utilisables). Si on cherche la zone asymptotique en observant plutôt le R_{GCI}, on ne peut constater de tendance dans son évolution. Pourtant dans la mesure où les résultats sont perturbés par l'erreur d'arrondi, l'évolution de V en convergence semble cohérente.

Dans l'ensemble, tout porte à croire que la solution numérique se rapproche de la solution exacte des EDP. Cependant, les normes des variables de la solution de similitude sont très différentes de celles obtenues numériquement, celles-là étant très loin à l'extérieur des intervalles décrits par le GCI. Sans connaître la différence entre la solution de similitude et la solution exacte, il est impossible de tirer les conclusions recherchées pour la vérification.

Triplet	p	$\|V\|$	GCI	R_{GCI}
(Norme L^1)				
8 - 16 - 24	1.89	1.6763 E-03	2.6e-06	N/A
16 - 24 - 32	2.50	1.6770 E-03	8.3e-07	1.5338
24 - 32 - 48	2.00	1.6775 E-03	5.0e-07	0.7370
32 - 48 - 64	1.63	1.6777 E-03	4.2e-07	0.7478
48 - 64 - 96	3.08	1.6778 E-03	5.0e-08	2.3888
(Norme L^2)				
8 - 16 - 24	3.26	2.4660 E-03	2.7e-06	N/A
16 - 24 - 32	6.45	2.4664 E-03	9.3e-08	4.5945
24 - 32 - 48	1.00	2.4668 E-03	1.0e-06	0.0619

TAB. 5.12 Convergence des normes L^1 et L^2 de V, d'après la méthode du GCI Roache (1995).

Méthode des Moindres carrés

La méthode des moindres carrés a été appliquée avec la limitation suivante : pour permettre la convergence de l'algorithme isolant p, C et ϕ_{ER}, l'ordre de convergence observé devait nécessairement être compris entre 0 et 10. Or, ce critère a été rempli pour les normes L^1 et L^2 de toutes les variables, à l'exception de la pression P, dont l'erreur d'arrondi était trop importante pour pouvoir appliquer cette approche. D'autre part, les courbes de convergence de chaque norme des autres variables ont pu être obtenues en utilisant les résultats de tous les maillages.

Au niveau de la convergence, on observe essentiellement les mêmes phénomènes tant pour les normes L^1 que L^2. Malgré tout, dans l'ensemble, on observe des ordres de convergence légèrement plus élevés pour la norme L^2. En contrepartie, on remarque que la courbe de convergence de cette norme croiserait le seuil de l'erreur d'arrondi plus rapidement.

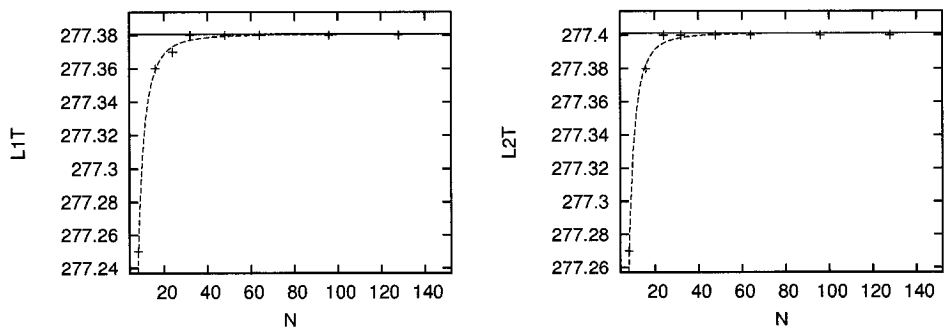
Comme les solutions fournies par CFDLIB ont seulement cinq chiffres significatifs, les normes calculées à partir de ces solutions sont également présentées avec un maximum de cinq chiffres significatifs. Aussi, les incertitudes évaluées avec la méthode des

Norme	CFDLIB	p
$\|T\|_{L_1}$	277.38 ± 0.01	2.61
$\|U\|_{L_1}$	6.058 E-4 ± 3.5 E-6	0.95
$\|V\|_{L_1}$	1.6778 E-3 ± 1 E-7	2.04
$\|T\|_{L_2}$	277.40 ± 0.01	2.82
$\|U\|_{L_2}$	9.1292 E-4 ± 1.3 E-6	1.47
$\|V\|_{L_2}$	2.4668 E-3 ± 1 E-7	3.54

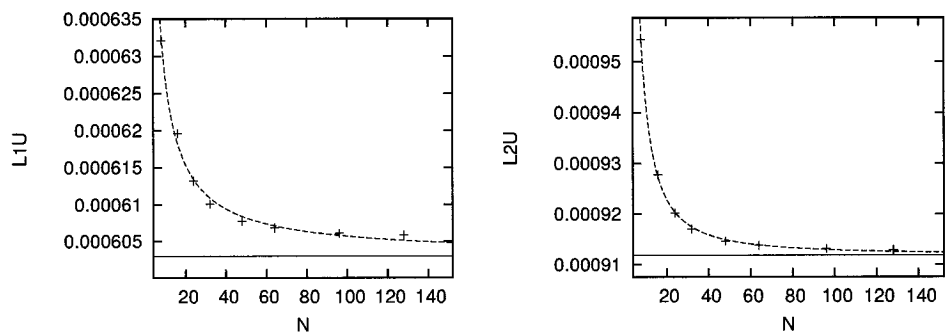
TAB. 5.13 Convergence des normes L^1 et L^2, d'après la méthode des moindres carrés Eça & Hoekstra (2002).

moindres carrés ont été ajustées d'après l'erreur d'arrondi. Ainsi, pour la température, bien que les résultats de la méthode des moindres carrés suggéraient une erreur de discrétisation de l'ordre de 1×10^{-3}, l'incertitude a été ramenée à la valeur de l'erreur d'arrondi, soit 1×10^{-2}. Pour la vitesse V, l'erreur de discrétisation estimée est de l'ordre de l'erreur d'arrondi. On peut d'ailleurs le constater sur la FIG. 5.13c, où les derniers points autour de la courbe de convergence sont bloqués au niveau de l'erreur d'arrondi. En fait, seules les normes de la vitesse U varient suffisament pour que l'erreur d'arrondi ne vienne brouiller l'étude de convergence. Par comparaison, $\|U\|$ est environ trois fois plus faible que $\|V\|$, ce qui en fait une variable plus sensible ; son erreur de discrétisation absolue serait d'ailleurs plus grande que celle de $\|V\|$ d'un ordre de grandeur.

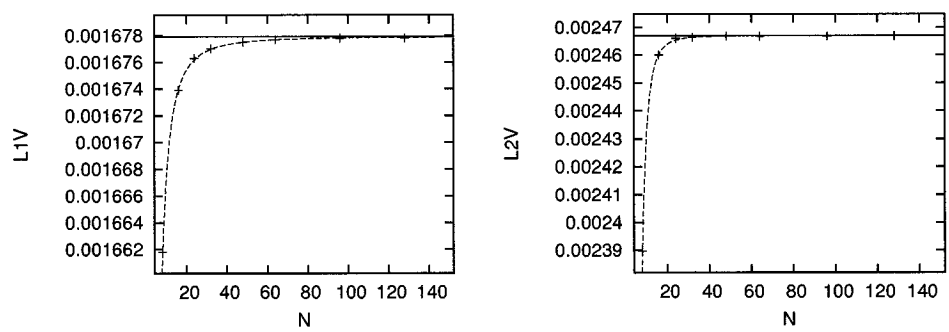
Enfin, les comparaisons avec la solution de similitude mènent aux mêmes constats que pour les autres méthodes : les solutions sont très différentes, suffisament pour que les valeurs obtenues pour la solution de similitude soient loin à l'extérieur des bornes d'incertitudes estimées pour la solution numérique. L'absence d'incertitude pour la solution de similitude nous empêche de pousser plus loin les conclusions. En résumé, bien que le code n'ait pas présenté de comportement erratique, on ne peut considérer la comparaison avec la solution de similitude comme une preuve quantitativement valable, tel que le requiert le processus de vérification.



(a) Convergence de $\|T\|$

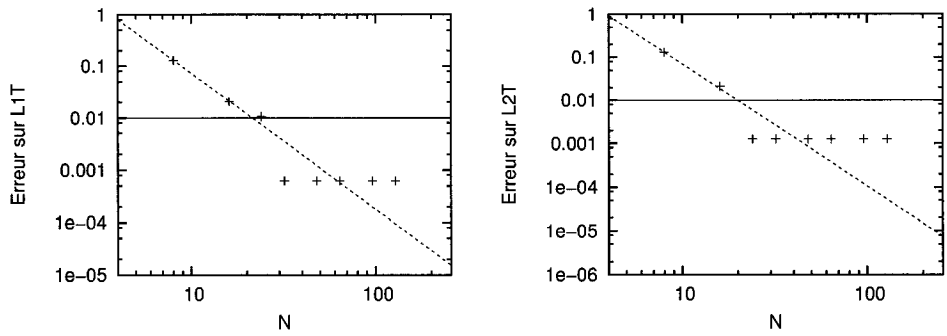


(b) Convergence de $\|U\|$

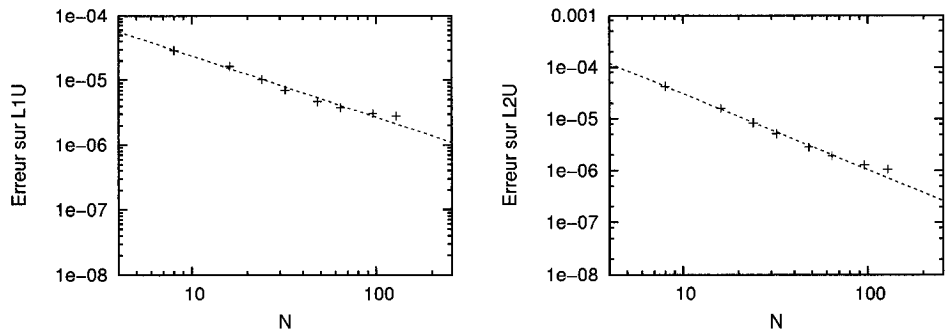


(c) Convergence de $\|V\|$

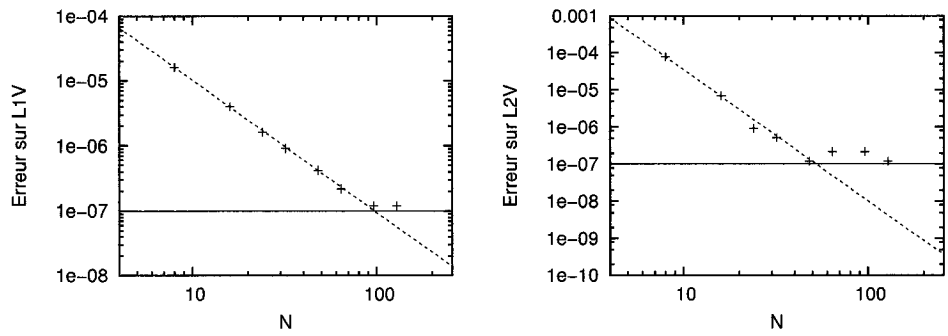
FIG. 5.12 Convergence des normes L^1 et L^2 de T, U, et V. La courbe en pointillé est obtenue par la méthode des moindres carrés, de même que la valeur extrapolée, représentée par une droite continue.



(a) Convergence de $\|T - T_{ER}\|$



(b) Convergence de $\|U - U_{ER}\|$



(c) Convergence de $\|V - V_{ER}\|$

FIG. 5.13 Graphiques logarithmiques de convergence des normes L^1 et L^2 : $\|T - T_{ER}\|$, $\|U - U_{ER}\|$, et $\|V - V_{ER}\|$. La droite en pointillé est obtenue par la méthode des moindres carrés. La ligne pleine représente la valeur de l'erreur d'arrondi, soit le plus petit écart pouvant être décrit par l'output de CFDLIB.

5.5.3 Analyse

L'objectif d'une étude de vérification de code est d'amener une preuve quantitative témoignant que le code résout correctement – ou non – les équations aux dérivées partielles pour lesquelles il a été conçu. Cette démarche repose sur le raisonnement suivant : si la solution donnée par le code est proche de la solution de similitude, et que la solution de similitude est proche de la solution de l'EDP, alors la solution donnée par le code est proche de la solution de l'EDP. Le raisonnement est semblable à celui utilisé par la méthode de la comparaison de codes, tel que formalisé par Trucano et al. (2003). On peut l'écrire ainsi :

$$\|\phi_{CFD} - \phi_{EDP}\| \leq \|\phi_{CFD} - \phi_{similitude}\| + \|\phi_{similitude} - \phi_{EDP}\| \tag{5.33}$$

Bien que la solution de similitude utilisée pour modéliser le jet laminaire soit largement répandue dans la littérature, on peut difficilement évaluer quantitativement le terme $\|\phi_{similitude} - \phi_{EDP}\|$. Pour ce faire, on aurait besoin d'une évaluation de ϕ_{EDP} plus précise que celle que nous donne $\phi_{similitude}$. Or, les équations de type couche limite ne sont alors d'aucun recours, puisqu'elles introduisent des différences avec l'EDP qui ne sont pas quantifiables sans connaître ϕ_{EDP}.

Ainsi, la comparaison des résultats nous fournit uniquement la différence entre la solution des équations de couche limite et les résultats obtenus par CFDLIB. Ces solutions ne peuvent être liées quantitativement à l'EDP de départ. On ne peut donc tirer de conclusion quantitative sur la valeur du code, ce qui a pour conséquence qu'on ne peut considérer cette activité comparative comme une activité de vérification au sens strict. Bien que l'étude n'ait relevé aucun mauvais comportement du code, celle-ci ne constitue pas une preuve telle que requise par les activités de vérification de code.

CHAPITRE 6

APPLICATION DE LA MÉTHODE DE COMPARAISON DE CODES

Le problème choisi pour l'application de la méthode de comparaison de code (MCC) est celui d'un écoulement circulatoire incompressible (ECI) dans une cavité carrée. Ce problème a été suggéré pour la vérification de codes commerciaux par Abanto et al. (2005). Dans leur article, ces derniers traitent l'ECI au moyen de la méthode des solutions manufacturées (MMS).

Dans le cadre de la MCC, on doit procéder essentiellement en quatre étapes. D'abord, le bon comportement du code de référence, Fluent 6.1, ainsi que celui des techniques d'estimation d'erreur, doivent être vérifiés. Ceci est effectué au moyen de la MMS. Ensuite, on procède au calcul du même problème, mais cette fois-ci sans termes sources, toujours avec le code de référence. Puis, on soumet le problème au code à vérifier : CFDLIB 3.1. Enfin, les résultats obtenus au moyen des deux codes sont analysés et comparés, au moyen de techniques de Vérification de Calculs.

6.1 Description du problème

Le problème présenté par Abanto et al. (2005) est une étude utilisant la MMS. Afin de pouvoir l'appliquer à la MCC, on s'assure d'abord que la solution manufacturée soit représentative de celle qui serait obtenue sans les termes sources. Ainsi, dans le cas présent, on choisit un nombre de Reynolds[1] assez petit ($Re = 100$), afin que la solution maintienne une configuration semblable si l'on retire les termes sources des EDP.

[1]Notez que la longueur caractéristique utilisée pour le calcul du nombre de Reynolds correspond à la moitié de la longueur des cotés de la cavité. De plus, la vitesse de référence U_{ref}, utilisée pour le calcul du Re, est la vitesse maximale que l'on retrouve sur les frontières.

Propriété	Valeur
ρ	998 kg/m^3
U_{ref}	0.04 m/s
H, L	0.005 m
T_1	273.16 m/s
T_2	293.16 m/s
ν	1×10^{-6} m^2/s
α	1.25×10^{-6} m^2/s
p_0	0 Pa

TAB. 6.1 Tableau des paramètres du problème

D'autre part, le problème d'Abanto et al. (2005) est présenté sous forme adimensionnelle. On devra donc l'adapter car CFDLIB fonctionne au moyen de paramètres dimensionnels. Il sera aussi plus convenable d'utiliser une formulation semblable pour les cas effectués avec le code de référence. Les paramètres dimensionnels utilisés sont présentés au tableau 6.1.

Enfin, le problème original ne comporte pas de solution pour l'équation d'énergie. Une solution non-triviale pour la température a donc été imaginée. Notons aussi que, parmi les cas proposés par Abanto et al. (2005), seul le cas de la distribution uniforme de la pression a été considéré, celle-ci étant plus proche des solutions attendues pour les EDP sans termes sources.

Ainsi, l'étude portera sur les fonctions suivantes :

$$p = 0 \tag{6.1}$$

$$u = -\frac{4(-2y+H)x(-x+L)U_{ref}}{HL^2} \tag{6.2}$$

$$v = \frac{4(-2x+H)y(-y+L)U_{ref}}{HL^2} \tag{6.3}$$

$$T = \frac{T_1+T_2}{2} + \frac{(T_2-T_1)}{4}\left(\frac{2y}{H} - 1 + (\frac{2x}{L} - 1)^3\right) \tag{6.4}$$

Le domaine considéré est un rectangle allant de 0 à L dans la direction x, et de 0 à H dans la direction y.

Dans le cas de la MMS, les EDP discrètes seront modifiées par les termes sources adéquats pour que leur solution exacte corresponde à ces fonctions. De plus, dans les cas de comparaison effectués sur Fluent et CFDLIB sans termes sources, ces fonctions seront utilisées comme conditions frontières de Dirichlet (du moins pour ce qui est de la température et des vitesses).

Finalement, précisons que tous les maillages utilisés sont structurés et réguliers, et ce à toutes les étapes de la méthode de comparaison de code. Bien que cette approche ne soit pas optimale au niveau du temps de calcul, elle permet néanmoins de créer facilement des maillages similaires.

6.2 Confirmation du calcul par la MMS

Le système d'EDP résolu par le code de référence est le suivant :

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = Q_{CONT} \tag{6.5}$$

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial x} - \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = Q_u \tag{6.6}$$

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial y} - \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) = Q_v \tag{6.7}$$

$$u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} - \alpha\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) = Q_T \tag{6.8}$$

En substituant les fonctions 6.1 à 6.4 dans les équations 6.5 à 6.8, et en posant $H = L$, on obtient les termes sources :

$$
\begin{aligned}
Q_{CONT} &= 0 && (6.9)\\
Q_u &= 8U_{ref}(8x^3U_{ref}y^2 - 12x^2U_{ref}y^2L + 4xU_{ref}y^2L^2 \\
&\quad -8x^3U_{ref}yL + 12x^2U_{ref}yL^2 - 4xU_{ref}yL^3 \\
&\quad +4x^3U_{ref}L^2 - 6x^2U_{ref}L^3 + 2xU_{ref}L^4 + 2\nu L^3y - \nu L^4)/L^6 && (6.10)\\
Q_v &= 8U_{ref}(8y^3U_{ref}x^2 - 12x^2U_{ref}y^2L - 8y^3U_{ref}xL \\
&\quad +12xU_{ref}y^2L^2 + 4x^2U_{ref}yL^2 - 4xU_{ref}yL^3 + 4y^3U_{ref}L^2 \\
&\quad -6y^2U_{ref}L^3 + 2yU_{ref}L^4 - 2\nu L^3x + \nu L^4)/L^6 && (6.11)\\
Q_T &= 2(-T_2 + T_1)(2x - L)(12x^3U_{ref}y - 18x^2U_{ref}yL + 6xU_{ref}yL^2 \\
&\quad -6x^3U_{ref}L + 9x^2U_{ref}L^2 - 3xU_{ref}L^3 - y^2U_{ref}L^2 \\
&\quad +yU_{ref}L^3 + 3\alpha L^3)/L^6 && (6.12)
\end{aligned}
$$

Les manipulations algébriques sont effectuées avec Maple 10, qui génère le code C inséré dans les *user defined functions* de Fluent.

6.2.1 Convergence des itérations

Pour toutes les simulations utilisées, le système d'équations discrètes est résolu jusqu'à sa convergence complète. Nos essais montrent que le nombre d'itérations requis pour y arriver est proportionnel à N^2. De plus, tous les graphiques de convergence itérative observés sont similaires. On peut donc paramétrer le nombre d'itérations dans les fichiers d'input du problème, ce qui permet d'effectuer efficacement des simulations sur plusieurs maillages.

La figure 6.1 présente un exemple de graphique de convergence itérative.

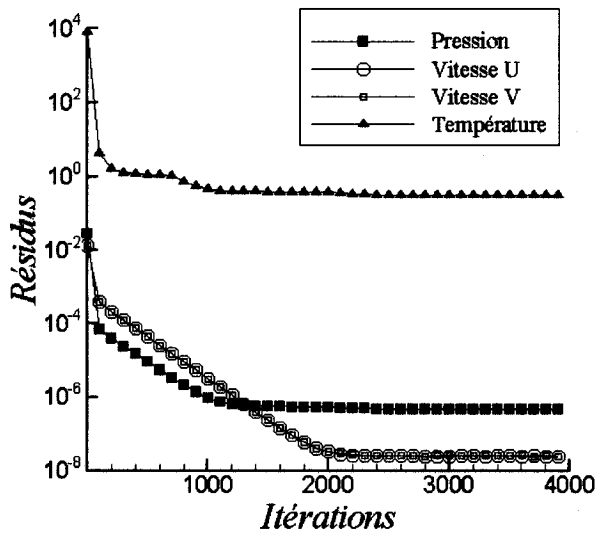


FIG. 6.1 Convergence itérative des résidus de la MMS, pour $N = 72$.

6.2.2 Évaluation d'erreur

Les évaluations d'erreur indiquent que Fluent converge bien vers la solution exacte. En effet, la figure 6.2 présente une convergence monotone de la norme L^1 de l'erreur, conformément à la relation 3.3.

La figure 6.3 montre que l'ordre de convergence observé $p_{obs} \approx 0.8$ et qu'il est à peu près stable tout au long du raffinement, ce qui est légèrement inférieur à l'ordre de convergence théorique $p_{th} = 1$. Cependant, on se serait attendu à un ordre de convergence observé un peu plus élevé. En effet, plusieurs facteurs peuvent affecter l'ordre de convergence sans que l'intégrité du code n'ait à être remise en question. On retient donc de cette étude de convergence les observations suivantes :

1. La solution numérique converge vers la solution exacte.
2. La solution numérique converge de façon monotone.
3. L'ordre de convergence de toutes les variables est semblable et est environ de 0,8.

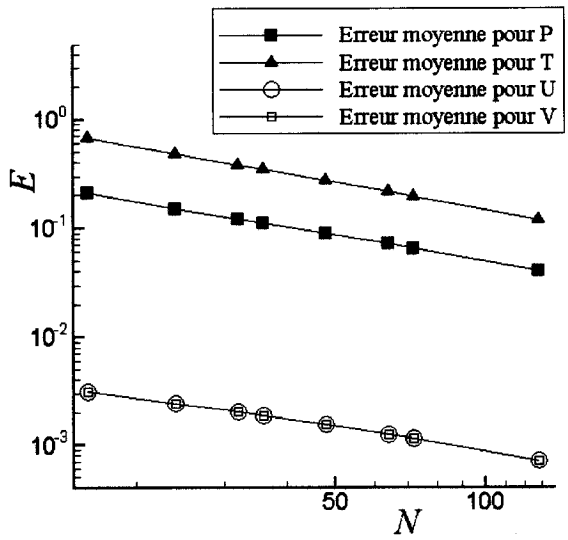


FIG. 6.2 Convergence de la norme L^1 de l'évaluation de l'erreur, $|E|_{L^1}$.

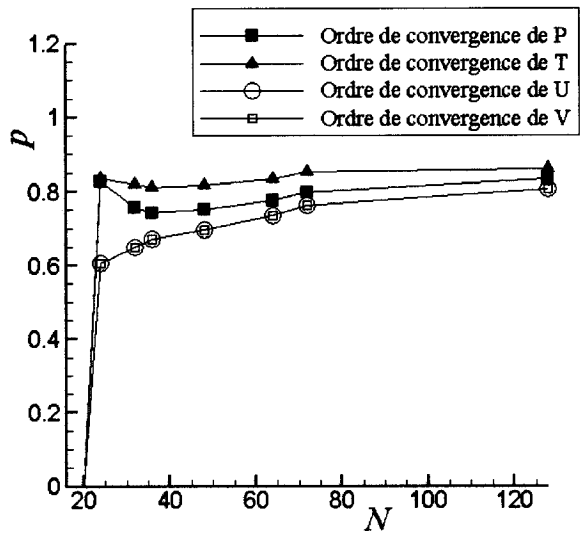


FIG. 6.3 Ordre de convergence observé, obtenu avec les évaluations d'erreur $|E|_{L^1}$, pour différents maillages.

6.2.3 Analyse des méthodes de Vérification de Calculs utilisées

La MMS nous fournit la distribution de l'erreur exacte. On peut donc mettre à l'épreuve les méthodes d'estimation d'incertitude que nous utiliserons pour la comparaison de codes.

6.2.3.1 Méthode de Cadafalch et al.

Les tableaux 6.2 à 6.5 présentent les résultats obtenus avec la méthode de Cadafalch et al. (2002).

On voit d'abord qu'une bonne proportion de la solution, pour chaque variable, est constituée de nœuds de Richardson. On s'attend donc à ce que l'ordre de convergence observé moyen soit proche de l'ordre de convergence observé avec les évaluations d'erreur. En effet, bien qu'on constate une variation assez importante sur le domaine, l'ordre de convergence observé moyen évalué est toutefois toujours assez proche de 0,8.

De plus, on constate que le GCI semble être un bon estimateur de l'incertitude pour T, u et v. En effet, on voit que la région où l'erreur est inférieure au GCI, pour la température et les composantes de vitesse, couvre environ 80 à 90 % du domaine Ω_R. Ces valeurs sont proches de l'objectif de 95%, surtout si l'on considère que les cellules ne contenant ne serait-ce qu'un seul nœud oscillatoire ou convergé ont été exclues de cette région, et que les variables ont toutes été transférées aux nœuds par interpolation lors de la création de l'output. Cette interpolation peut introduire du bruit dans les données pour l'étude de convergence.

On note aussi que le GCI sous-évalue l'incertitude pour la pression, puisqu'il est supérieur à l'erreur seulement dans 43% des cas.

Triplet	18–36–72
$\|P\|_{L^2(\Omega_{RC})}$	0.082
$\|\text{GCI}(P)\|_{L^2(\Omega_{RC})}$	0.075
$\overline{p}$	0.94
σ_p	0.41
Région de Richardson (Ω_R)	94 %
Région convergée (Ω_C)	0 %
Région oscillatoire (Ω_O)	6 %
Taux de réussite du GCI local (sur Ω_R)	43 %

TAB. 6.2 Caractéristiques de convergence de la pression p, d'après la méthode de Cadafalch et al. (2002).

La figure 6.4 nous montre que, pour toutes les variables, le GCI est représentatif de la distribution de l'erreur ; on constate que les distributions ont la même allure. Pour la température et les composantes de vitesse, on constate également que les valeurs de GCI sont légèrement plus grandes que celles de l'erreur, et ce sur la très grande partie du domaine. Ce n'est pas le cas pour la pression, le GCI étant systématiquement inférieur à l'erreur dans une région assez grande, au centre du domaine.

Le cas de la pression est particulier : n'étant imposée nulle part dans la définition du problème, le programme n'a donc pas de point de référence en pression. De plus, le système d'EDP résolu ne contient que des gradients de pression. Ainsi, la solution exacte prescrit une pression constante, mais dans l'état actuel de la définition du problème, on n'a pas suffisament d'information pour donner une valeur à cette constante. Peu importe vers quelle valeur tend la pression, l'EDP sera correctement résolue si la pression est constante sur tout le domaine.

Triplet	18–36–72
$\|T\|_{L^2(\Omega_{RC})}$	283.15
$\|\text{GCI}(T)\|_{L^2(\Omega_{RC})}$	0.34
$\overline{p}$	0.76
σ_p	0.36
Région de Richardson (Ω_R)	78 %
Région convergée (Ω_C)	10 %
Région oscillatoire (Ω_O)	12 %
Taux de réussite du GCI local (sur Ω_R)	82 %

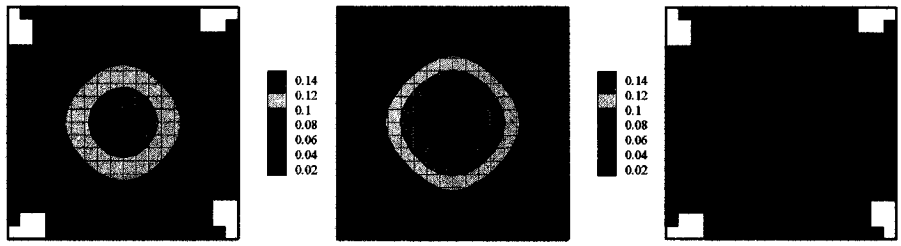
TAB. 6.3 Caractéristiques de convergence de la température T, d'après la méthode de Cadafalch et al. (2002).

Triplet	18–36–72
$\|U\|_{L^2(\Omega_{RC})}$	0.0164
$\|\text{GCI}(U)\|_{L^2(\Omega_{RC})}$	0.0028
$\overline{p}$	0.52
σ_p	0.67
Région de Richardson (Ω_R)	81 %
Région convergée (Ω_C)	10 %
Région oscillatoire (Ω_O)	9 %
Taux de réussite du GCI local (sur Ω_R)	86 %

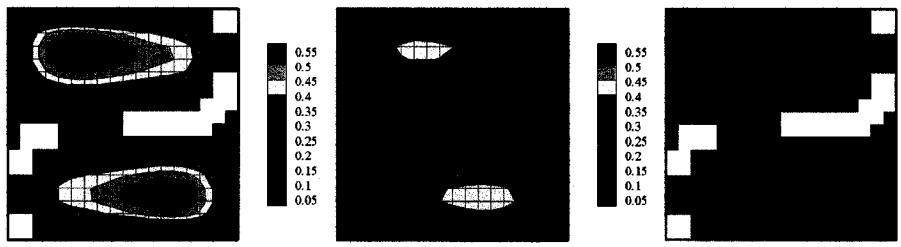
TAB. 6.4 Caractéristiques de convergence de la Vitesse u, d'après la méthode de Cadafalch et al. (2002).

Triplet	18–36–72
$\|V\|_{L^2(\Omega_{RC})}$	0.0163
$\|\text{GCI}(V)\|_{L^2(\Omega_{RC})}$	0.0028
$\overline{p}$	0.52
σ_p	0.67
Région de Richardson (Ω_R)	83 %
Région convergée (Ω_C)	10 %
Région oscillatoire (Ω_O)	7 %
Taux de réussite du GCI local (sur Ω_R)	87 %

TAB. 6.5 Caractéristiques de convergence de la Vitesse v, d'après la méthode de Cadafalch et al. (2002).



(a) Pression



(b) Température

(c) Vitesse u

(d) Vitesse v

FIG. 6.4 Méthode de Cadafalch et al adaptée à la MMS. Les figures de gauche présentent la distribution du GCI sur le domaine Ω_{RC}. Au centre, on retrouve la distribution de l'erreur sur les valeurs du maillage fin, rapportées sur le maillage d'analyse. Les graphiques de droites montrent en bleu les régions où l'erreur évaluée dépasse le GCI.

Norme	MMS Valeur ($N = 72$) $\pm$ GCI	p	Valeur exacte
$\|P\|_{L_2}$	0.079 ± 0.094	0.75	0
$\|T\|_{L_2}$	–	–	283.1810205...
$\|U\|_{L_2}$	0.0156 ± 0.0058	0.23	0.01686548085...
$\|V\|_{L_2}$	0.0156 ± 0.0058	0.23	0.01686548085...

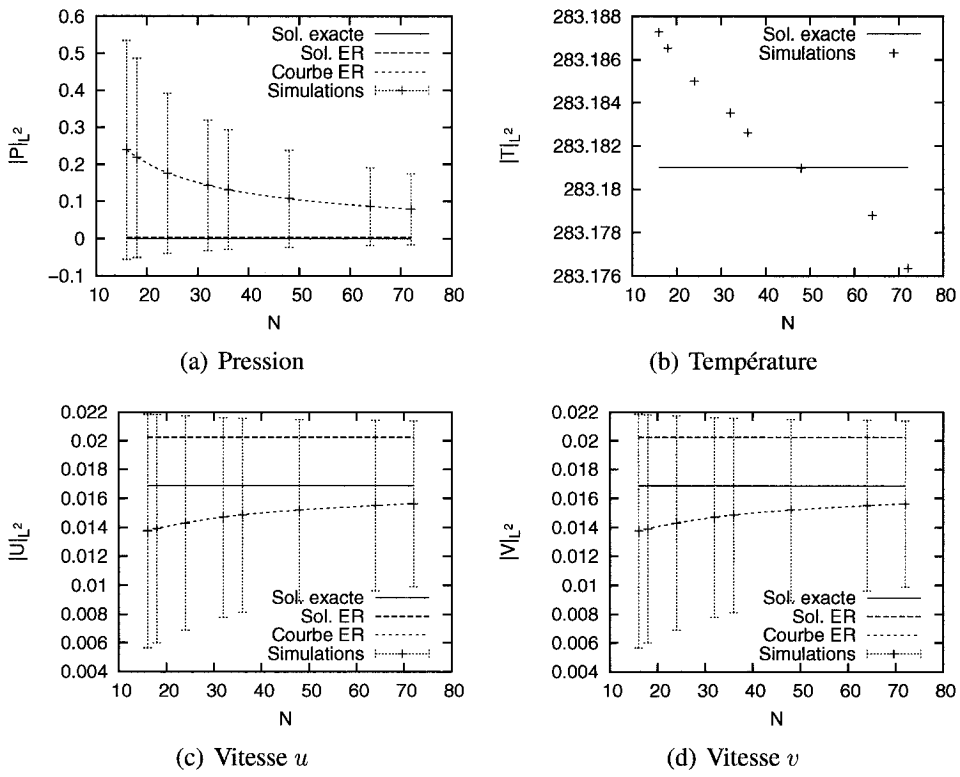
TAB. 6.6 Convergence des normes et L^2, d'après la méthode des moindres carrés (Eça & Hoekstra, 2002).

6.2.3.2 Méthode des moindres carrés

La méthode des moindres carrés (MMC) est mise à l'épreuve au moyen de la norme L^2 de chaque variable. Pour fins de comparaison, la valeur exacte des normes L^2 a également été calculée de façon analytique. Les résultats de la méthode des moindres carrés sont présentés au tableau 6.6 et à la figure 6.5.

D'abord, on voit que la solution exacte se situe dans l'intervalle d'incertitude obtenu par la MMC, pour la pression et les composantes de vitesse. De plus, tandis que l'ordre de convergence de la pression est pratiquement le même que celui obtenu avec les évaluations d'erreur, celui des vitesses est très petit. Cependant, notons que le mauvais comportement local du GCI pour la pression n'apparaît pas avec la MMC sur la norme L^2. Ainsi, pour le problème de comparaison, on devra traiter avec précaution les résultats obtenus avec la MMC pour la pression.

On constate également que la norme L^2 de la température diverge lorsqu'on tente un calcul de MMC avec les résultats de toutes les simulations effectuées. Pourtant, on sait d'après la figure 6.2 que la température converge de façon monotone, comme toutes les autres variables. On peut donc constater la divergence avec la MMC sans pour autant que l'erreur exacte ne diverge.



(a) Pression (b) Température

(c) Vitesse u (d) Vitesse v

FIG. 6.5 Convergence des variables de la solution manufacturée.

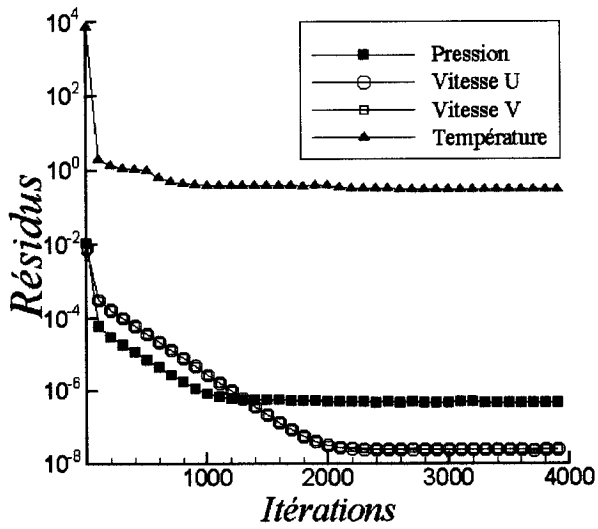


FIG. 6.6 Convergence itérative des résidus de solution de référence, pour $N = 72$.

6.3 Calcul du problème de référence

À présent que le programme de référence a été évalué au moyen de la MMS, on peut procéder à la solution du problème sans les termes sources. Dans le cas présent, on observe à peu près le même comportement de la part du programme de référence : les résidus convergent de façon similaire (tab. 6.6) ; les estimations d'incertitudes présentent des distributions semblables (fig. 6.9 à 6.12), sauf pour la température, dont la distribution diffère de façon plus marquée de celle de la solution manufacturée ; les ordres de convergence observés avec la MMC sont légèrement plus élevés pour la pression et les vitesses (tab. 6.12), tandis que la norme L^2 de la température diverge toujours (fig. 6.15).

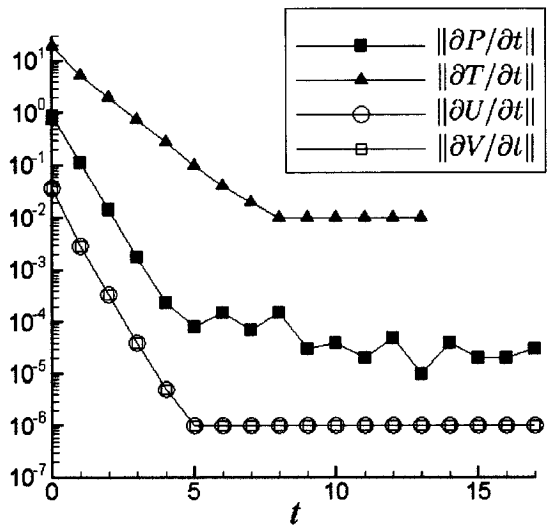


FIG. 6.7 Évolution dans le temps des normes L^{∞} des dérivées temporelles des variables p, T, u et v. Ces valeurs sont calculées d'après la solution pour $N = 72$.

6.4 Simulations CFDLIB

6.4.1 Convergence des itérations

La convergence itérative a nécessité d'utiliser un temps de simulation d'environ 20 secondes. L'évolution de la convergence itérative est présentée à la figure 6.7.

6.4.2 Statistiques

Pour effectuer l'étude de convergence, 8 simulations ont été retenues en tout, avec des maillages ayant respectivement 16, 18, 24, 32, 36, 48, 64 et 72 cellules dans chaque direction principale.

N	h : min : sec
16	00 : 00 : 34.00
18	00 : 00 : 46.30
24	00 : 01 : 14.27
32	00 : 02 : 50.28
36	00 : 05 : 41.26
48	00 : 13 : 24.88
64	00 : 57 : 06.98
72	01 : 27 : 05.21

TAB. 6.7 Temps de calcul en fonction du nombre de cellules effectif.

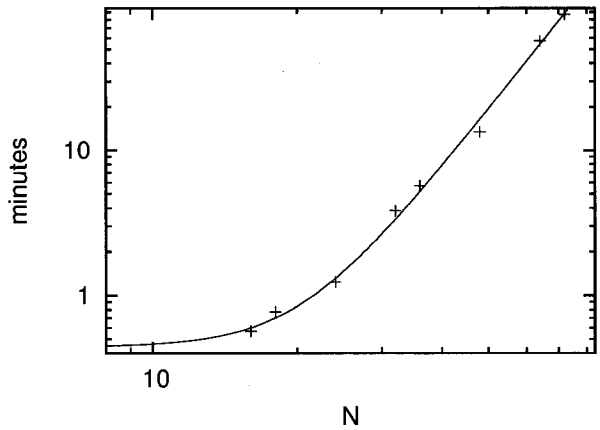


FIG. 6.8 Évolution du temps de calcul en fonction du nombre de cellules effectif.

6.4.3 Performance

Le temps de calcul est environ proportionnel à $N^{4,2}$, ce qui signifie que doubler le nombre de cellules dans chaque direction fait augmenter le temps de calcul d'un facteur 18, tel que le confirment les temps de simulation consignés au tableau 6.7.

6.5 Comparaison

Le critère de base pour la vérification, lorsqu'on utilise la méthode de comparaison de codes, est le chevauchement des zones d'incertitudes. Si le code de référence a été correctement vérifié, la solution exacte devrait la plupart du temps se situer à l'intérieur des zones d'incertitudes estimées. De plus, si le code à vérifier converge effectivement vers la solution exacte, cette dernière devrait également se trouver (dans la très grande majorité des cas) à l'intérieur des zones d'incertitudes calculées avec les simulation sur le code à vérifier. Ainsi, on devrait constater le chevauchement des barres d'incertitudes dans une proportion importante des cas. Toutefois, cette proportion n'est pas quantifiée, car cela nécessiterait de poser certaines hypothèses sur la dispersion stochastique des erreurs.

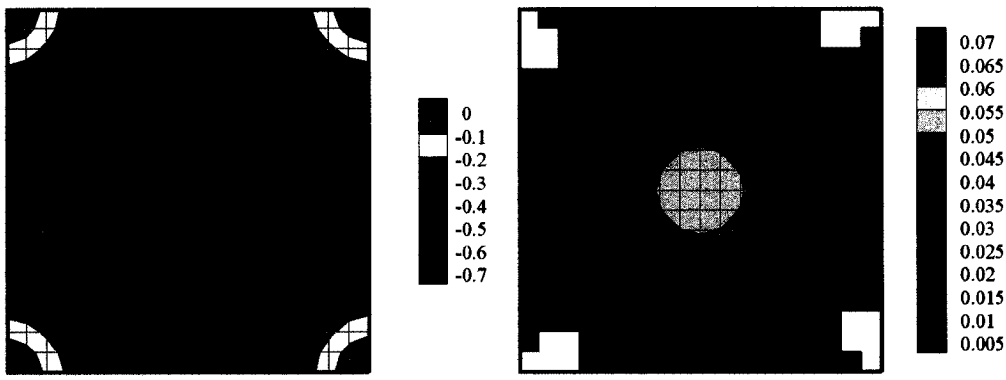
De façon générale, on peut dire qu'il y a chevauchement lorsque la somme des incertitudes des deux résultats est plus grande que l'écart entre les résultats. Ainsi, pour la présente étude de comparaison de codes, il y a chevauchement si :

$$|\phi_{CFDLIB} - \phi_{Fluent}| < |GCI_{CFDLIB} + GCI_{Fluent}| \tag{6.13}$$

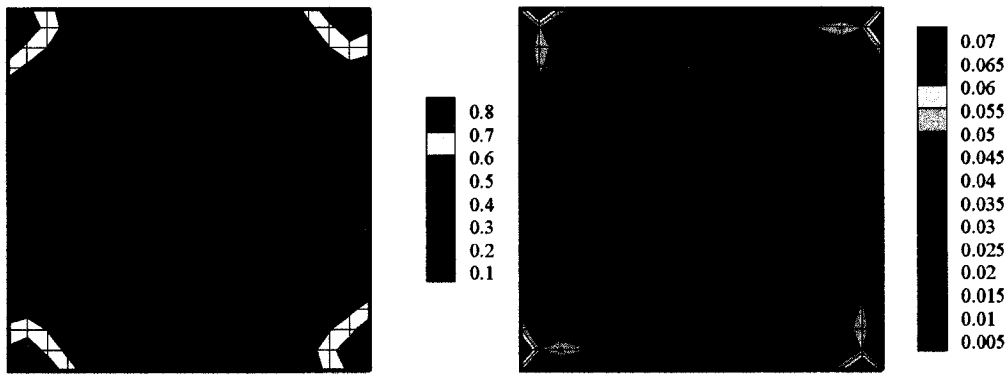
6.5.1 Comparaison avec la méthode de Cadafalch et al.

Tout d'abord, observons les solutions obtenues. Les figures 6.10, 6.11 et 6.12 nous montrent que les distributions de T, de u et de v sont semblables. De plus, les distributions des estimations d'incertitude (GCI) pour ces trois variables sont semblables, bien que les GCI locaux Fluent soient plus grands que ceux de CFDLIB.

Il n'est pas nécessaire que les distributions du GCI correspondent. Toutefois, si c'est le cas, cela nous indique que les domaines Ω_R, Ω_C et Ω_O devraient être assez semblables,

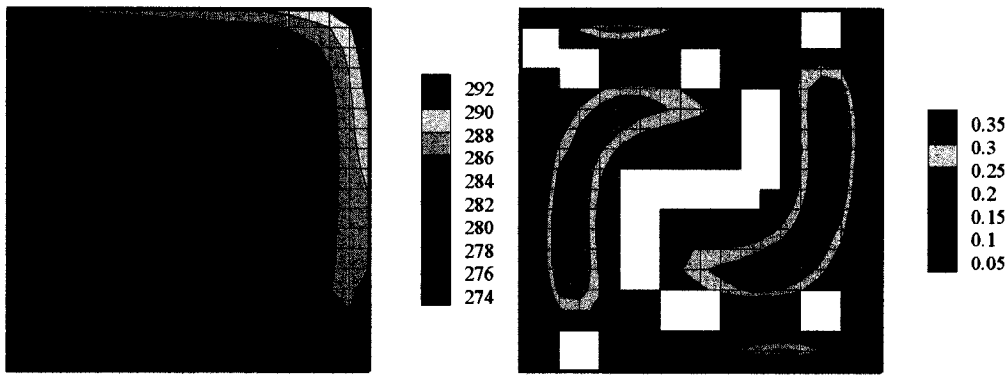


(a) Solution avec Fluent

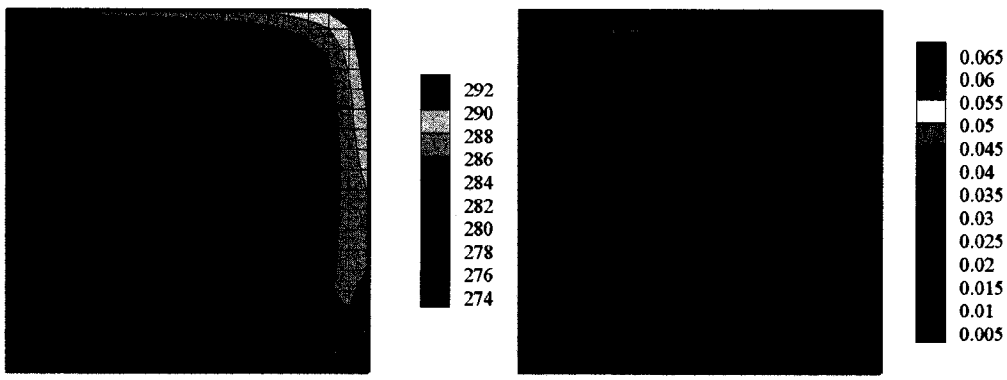


(b) Solution avec CFDLIB

FIG. 6.9 Comparaison des distributions de pression (à gauche) et des GCI y correspondant (à droite).

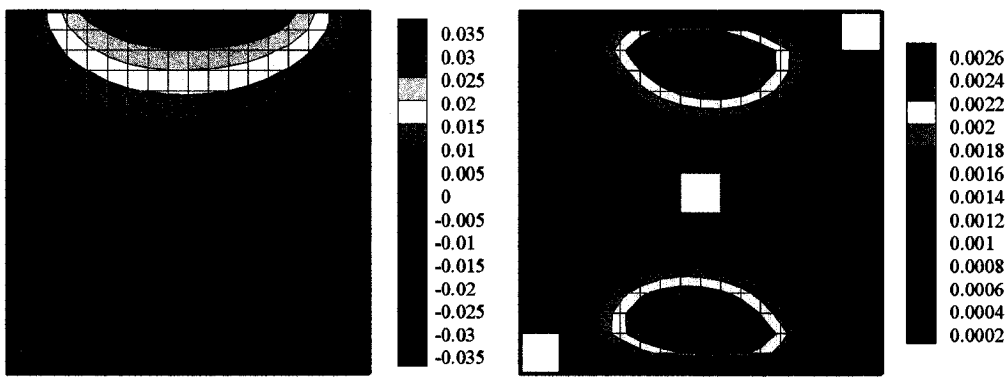


(a) Solution avec Fluent

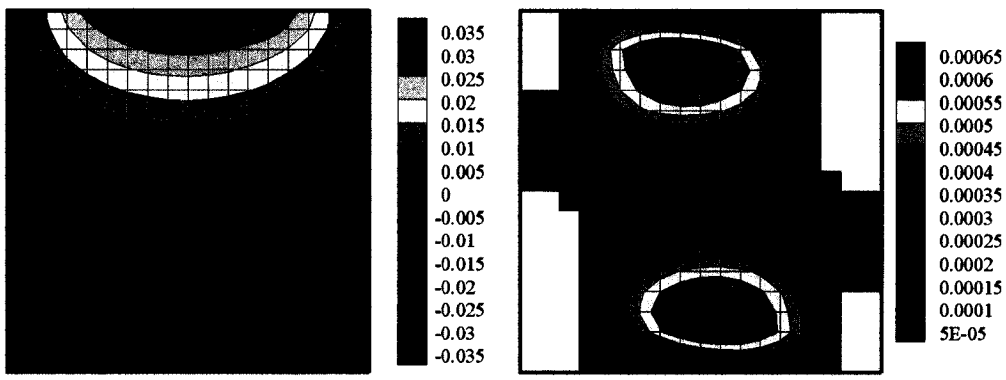


(b) Solution avec CFDLIB

FIG. 6.10 Comparaison des distributions de température (à gauche) et des GCI y correspondant (à droite).

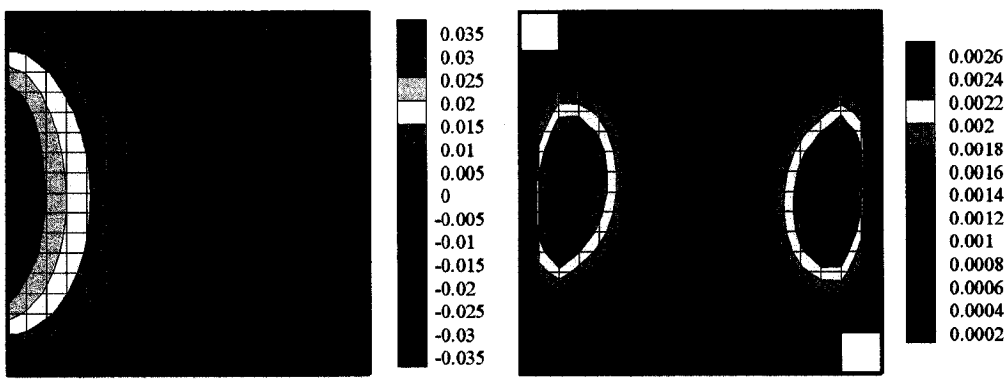


(a) Solution avec Fluent

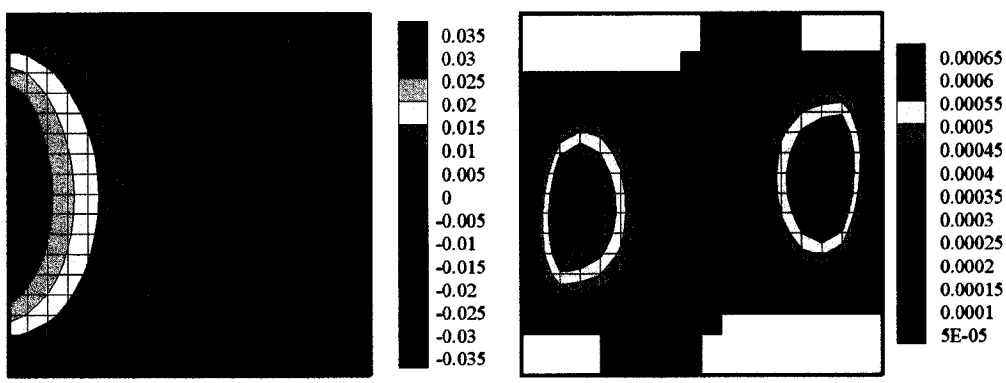


(b) Solution avec CFDLIB

FIG. 6.11 Comparaison des distributions de vitesse u (à gauche) et des GCI y correspondant (à droite).



(a) Solution avec Fluent



(b) Solution avec CFDLIB

FIG. 6.12 Comparaison des distributions de vitesse v (à gauche) et des GCI y correspondant (à droite).

Programme	Fluent	CFDLIB
$\|P\|_{L^2(\Omega_{RC})}$	0.568	0.323
$\|\text{GCI}(P)\|_{L^2(\Omega_{RC})}$	0.033	0.027
Écart	0.25	
U_{CONV}	0.06	
$\overline{p}$	1.31	1.07
σ_p	0.64	0.34
Région de Richardson (Ω_R)	94 %	99 %
Région convergée (Ω_C)	0 %	0 %
Région oscillatoire (Ω_O)	6 %	1 %

TAB. 6.8 Caractéristiques de convergence de la Pression p, d'après l'analyse proposée par Cadafalch et al. (2002). Note : Les domaines Ω_R, Ω_C et Ω_O ne sont pas identique dans les deux situations.

et donc que des comparaisons pourront être faites sur des grandeurs provenant environ des mêmes endroits.

On doit garder cette considération en tête lorsqu'on étudie les résultats des tableaux 6.8 à 6.11. En effet, ces tableaux présentent la valeur de normes L^2 obtenues sur des domaines différents selon le programme utilisé. Les valeurs présentées pour les normes de variables ne sont donc pas tenues d'être identiques, et les écarts observés peuvent n'être en réalité qu'une mesure de la différence entre les portions utilisées pour le calcul des normes. Également, on doit se rappeler que la norme sur le GCI est calculée en considérant un domaine qui inclut les régions convergées Ω_C, où le GCI local est supposé nul (Cadafalch et al., 2002). La norme du GCI ainsi obtenue ne pourra donc être considérée comme une incertitude sur le calcul numérique car elle ne prend pas en compte l'incertitude due aux erreurs d'arrondis. En conséquence, il appert que les résultats des tableaux 6.8 à 6.11 ne permettent pas une comparaison quantitative rigoureuse, comme le voudrait la théorie de la vérification.

Ainsi, on préfère effectuer des comparaison locales, sur l'intersection des domaines Ω_R de chaque programme. Sur ces régions, on sait que la composante d'erreur dominante devrait être l'erreur de discrétisation. On peut donc procéder à la somme des incertitudes

Programme	Fluent	CFDLIB
$\|T\|_{L^2(\Omega_{RC})}$	283.16	283.16
$\|\text{GCI}(T)\|_{L^2(\Omega_{RC})}$	0.21	0.02
Écart	0.00	
U_{CONV}	0.23	
$\overline{p}$	0.81	1.78
σ_p	0.89	0.36
Région de Richardson (Ω_R)	69 %	50 %
Région convergée (Ω_C)	13 %	50 %
Région oscillatoire (Ω_O)	19 %	0 %

TAB. 6.9 Caractéristiques de convergence de la température T, d'après l'analyse proposée par Cadafalch et al. (2002). Note : Les domaines Ω_R, Ω_C et Ω_O ne sont pas identique dans les deux situations.

Programme	Fluent	CFDLIB
$\|U\|_{L^2(\Omega_{RC})}$	0.0143	0.01486
$\|\text{GCI}(U)\|_{L^2(\Omega_{RC})}$	0.0014	0.00036
Écart	0.0006	
U_{CONV}	0.0018	
$\overline{p}$	0.69	1.35
σ_p	0.54	0.59
Région de Richardson (Ω_R)	86 %	75 %
Région convergée (Ω_C)	10 %	4 %
Région oscillatoire (Ω_O)	4 %	21 %

TAB. 6.10 Caractéristiques de convergence de la vitesse u, d'après l'analyse proposée par Cadafalch et al. (2002). Note : Les domaines Ω_R, Ω_C et Ω_O ne sont pas identique dans les deux situations.

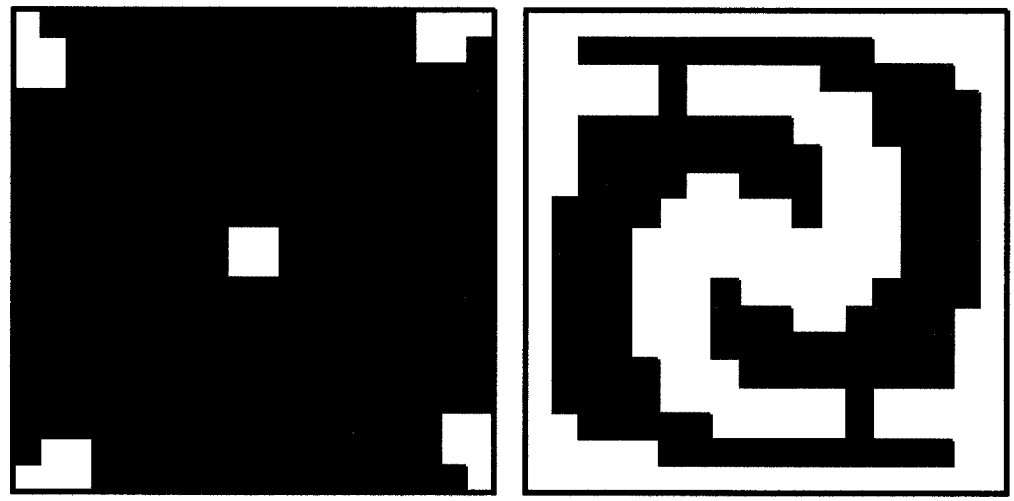
Programme	Fluent	CFDLIB
$\|V\|_{L^2(\Omega_{RC})}$	0.0143	0.01486
$\|\text{GCI}(V)\|_{L^2(\Omega_{RC})}$	0.0014	0.00036
Écart	0.0006	
U_{CONV}	0.0018	
$\overline{p}$	0.69	1.35
σ_p	0.59	0.59
Région de Richardson (Ω_R)	88 %	75 %
Région convergée (Ω_C)	10 %	4 %
Région oscillatoire (Ω_O)	2 %	21 %

TAB. 6.11 Caractéristiques de convergence de la vitesse v, d'après l'analyse proposée par Cadafalch et al. (2002). Note : Les domaines Ω_R, Ω_C et Ω_O ne sont pas identique dans les deux situations.

(GCI) des deux programmes, et comparer ce résultat à l'écart entre les solutions (voir éq. 6.13). La figure 6.13 présente ces résultats. On y voit que pour la température, la vitesse u et la vitesse v, il y a chevauchement des bandes d'incertitudes des valeurs locales, et ce, sur la totalité des domaines $((\Omega_R)_{Fluent} \cap (\Omega_R)_{CFDLIB})$ de chaque variable, soit sur une très grande partie du domaine. Cela constitue jusqu'à présent la preuve quantitative la plus convainquante du bon comportement de CFDLIB.

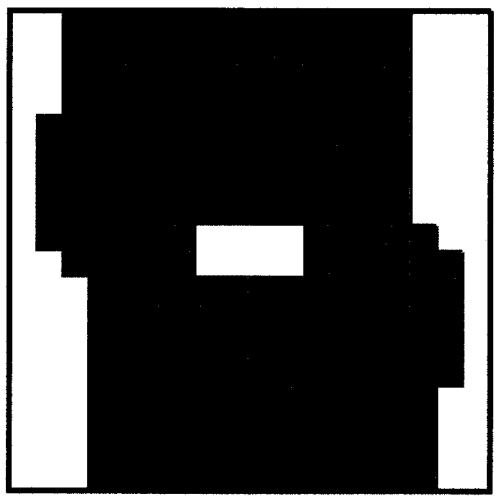
Par contre, la pression (fig. 6.9) prend des valeurs très différentes pour les deux programmes : négatives dans le cas des résultats Fluent, positives pour CFDLIB. Aussi, les distributions d'incertitudes pour la pression n'ont ni la même allure, ni la même amplitude. Le chevauchement des bandes d'incertitudes des valeurs locales de pression n'est constaté nulle part sur l'intersection des domaines Ω_R obtenus avec les deux programmes.

Cependant, une inspection plus en profondeur des graphiques de pression permet de constater que les gradients de pression calculés par les deux programmes semblent être approximativement les mêmes, les pressions ne différant que par une valeur constante. Considérant que la pression ne figure pas explicitement dans le système d'EDP résolu par les codes –seulement le gradient de pression– et que la pression n'est imposée nulle

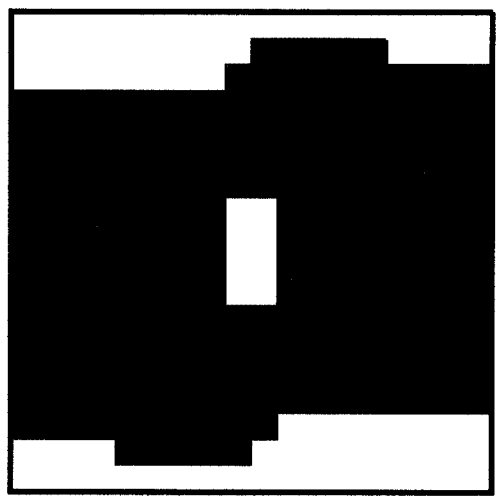


(a) Pression

(b) Température

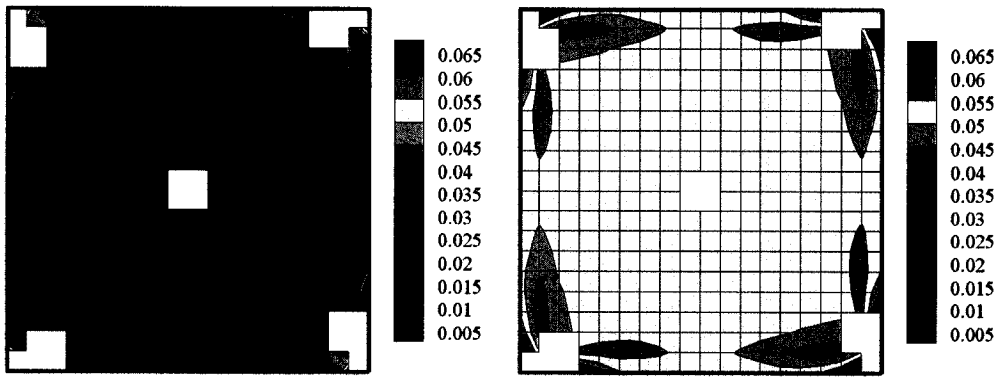


(c) Vitesse u



(d) Vitesse v

FIG. 6.13 Test de chevauchement local. Les régions où il y a chevauchement sont indiquées en vert, et celles où il n'y en a pas, en rouge.



(a) $\Delta P - \overline{\Delta P}$ (b) GCI$(P)_{CFDLIB}$+GCI$(P)_{Fluent}$

FIG. 6.14 Etude du comportement de la pression : comparaison entre la variabilité de ΔP et la somme des GCI(P) des deux programmes. Notes : $\Delta P = P_{CFDLIB} - P_{Fluent}$; $\overline{\Delta P} = 0,736$.

part dans le domaine, il est possible que le système d'EDP soit correctement résolu par les deux codes, les solutions en pression étant les mêmes à une constante près.

La figure 6.14(a) montre la variation absolue de la différence des pressions des deux codes, ΔP, autour de sa valeur moyenne, $\overline{\Delta P}$. En comparaison, la somme des incertitudes des deux codes est présentée à la figure 6.14(b). On peut voir que sur la quasi totalité de l'intersection des domaines Ω_{RC} des deux solutions, l'incertitude du calcul numérique sur la comparaison est plus grande que la variabilité de $\Delta P - \overline{\Delta P}$. En résumé, il se peut que les deux programmes convergent, à une constante près, vers la même distribution de pression.

Ainsi, en regard des résultats obtenus, on ne peut conclure au mauvais comportement de CFDLIB pour la variable de pression. Les études futures de vérification devraient toutefois veiller à ce que la définition du problème comporte l'imposition de valeurs de pression à certains endroits du domaine (en conditions frontières, en valeurs imposées dans le domaine, ou encore en conditions initiales si l'on veut vérifier des simulations en régime instationnaire). Si la pression ne peut être imposée nulle part comme para-

mètre, on devrait alors songer à considérer les gradients de pression lors des études de convergence.

6.5.2 Comparaison avec la méthode des moindres carrés

Les résultats obtenus avec la méthode des moindres carrés sont présentés au tableau 6.12 et à la figure 6.15. On peut y constater que les courbes de convergence calculées représentent assez fidèlement l'évolution des valeurs des normes ; la dispersion des données autour des courbes de convergence est toujours très faible. Cela peut s'expliquer principalement par la relative simplicité du problème et par la similitude des maillages utilisés.

Ainsi, pour les deux programmes, des courbes de convergence ont pu être identifiées pour toutes les variables, à l'exception de la température obtenue avec Fluent. Pour ce dernier cas, on obtient essentiellement le même comportement que celui observé avec la MMS ; la norme L^2 semble diverger mais les résultats pour la convergence locale semblent néanmoins acceptables (voir section précédente). Notons également que les résultats varient très peu.

On doit également être prudents avec les résultats obtenus par CFDLIB pour la température. En effet, la variation observée a lieu à un niveau inférieur à celui de l'erreur d'arrondi. D'ailleurs, le GCI obtenu avec la méthode des moindres carrés (tab. 6.12) est cinq fois plus petit que l'erreur d'arrondi qui survient à la création des fichiers d'output (où les données sont inscrites au format Fortran `1PE12.5`). Ainsi, les résultats de la figure 6.15(b) ne permettent pas de vérifier quantitativement le comportement de CFDLIB en température.

Pour la pression, bien que l'on observe la convergence pour les deux codes, il subsiste un écart entre les deux normes, et cet écart est bien supérieur à la somme des incertitudes.

Norme	Fluent Valeur ± GCI	p	CFDLIB Valeur ± GCI	p
$\|P\|_{L_2}$	0.558 ± 0.030	1.24	0.323 ± 0.024	1.18
$\|T\|_{L_2}$	–	–	283.16 ± 0.0002	1.67
$\|U\|_{L_2}$	0.0141 ± 0.0021	0.41	0.0148 ± 0.0003	1.26
$\|V\|_{L_2}$	0.0141 ± 0.0021	0.41	0.0148 ± 0.0003	1.26

TAB. 6.12 Convergence des normes et L^2, d'après la méthode des moindres carrés (Eça & Hoekstra, 2002). Les valeurs et les GCI présentés sont ceux du maillage $N = 72$.

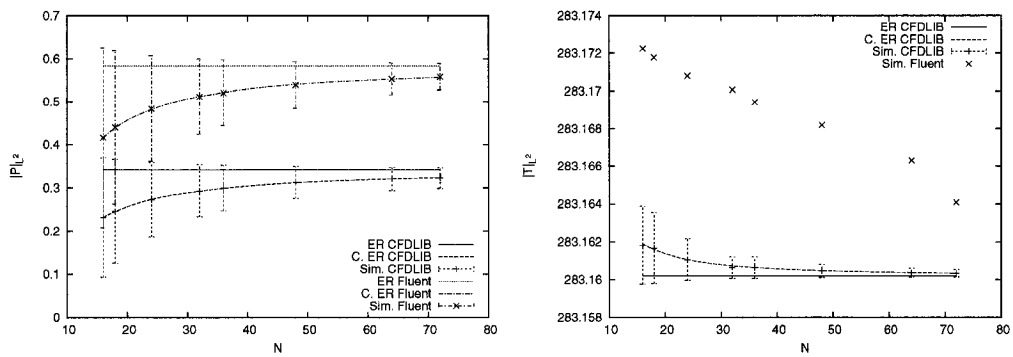
Une cause possible de cet écart est que les deux codes convergeraient, à une constante près, vers la même distribution. Cette problématique relève davantage d'une analyse des distributions, ce qui est présenté à la section 6.5.1.

Enfin, les deux composantes de vitesse montrent, dans chaque programme respectif, un comportement semblable en convergence. De plus, les normes de ces variables présentent d'importantes variations lors des raffinements. Dans ce cas-ci, on peut supposer que l'erreur de discrétisation serait la plus importante source d'erreur. On peut voir que la totalité des bandes d'incertitude pour les vitesses calculées avec CFDLIB sont à l'intérieur de celles provenant des calculs sur Fluent. On peut considérer ces observations comme un signe qu'en pratique, CFDLIB se comporte correctement pour le calcul des vitesses.

6.6 Discussion

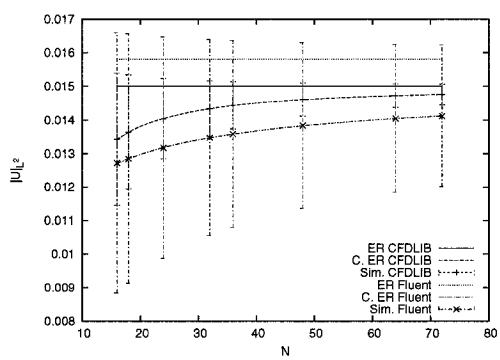
En l'absence de la possibilité de tester un programme de CFD avec la MMS, il peut être utile de procéder à des comparaisons de codes, afin de pouvoir mettre le programme à l'épreuve avec des problèmes complexes. On doit toutefois être conscient des limites inhérentes à la MCC, et être prêt à accepter un certain niveau de risque.

Tout d'abord, rappelons que la comparaison de code n'a de sens que si l'on a accumulé

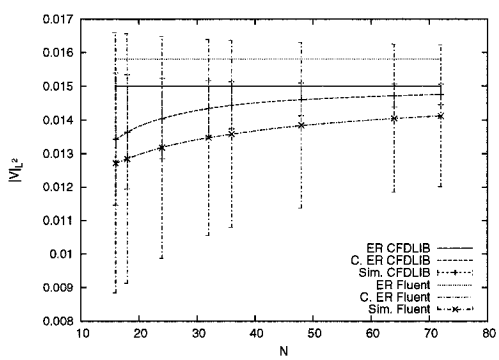


(a) Pression

(b) Température



(c) Vitesse u



(d) Vitesse v

FIG. 6.15 Convergence des variables p, T, u et v pour les deux programmes.

suffisament de preuves au sujet du bon fonctionnement du programme de référence. Cela peut être effectué de façon efficace pour certains cas particuliers, où le code de référence (comme dans le cas présent) répondrait bien aux tests de MMS. Les incertitudes peuvent alors être estimées avec confiance. En pratique, il peut être difficile de trouver de tels cas, même en utilisant des codes commerciaux éprouvés. Une grande quantité de tests doit donc être entreprise, avec tantôt différents problèmes, différents domaines, différents maillages, différents types de conditions frontières, etc. Par exemple, on peut vouloir utiliser la MCC pour contribuer à la vérification d'un programme pour une situation particulière. Cependant, il est possible que le programme choisi comme code de référence ne puisse pas fournir de preuves probantes de son bon fonctionnement. Il peut également produire des résultats avec un niveau d'incertitude inacceptable. Pour ces situations, l'étude de comparaison de code ne pourrait alors être effectuée sur des bases scientifiquement solides. De plus, ce constat ne viendrait qu'après de potentiellement longs et couteux essais.

Ensuite, pour que la comparaison puisse être faite, l'autre code (le code1) doit également être en mesure de fournir des solutions. Par exemple, même si l'on trouve un cas où le code de référence est approprié, il est possible que le code1 ne soit pas assez robuste pour fournir une solution dans la même situation.

Ainsi, on doit d'abord accumuler des preuves permettant d'affirmer que des simulations des deux programmes peuvent être utilisées pour fins de comparaisons. Puis, au moment de la comparaison, on doit garder en tête certains points concernant les incertitudes en situation de comparaison de code. Premièrement, la nature du test de comparaison, où l'on compare l'écart à la somme des incertitudes, fait en sorte qu'il est plus facile de constater le chevauchement des bandes d'incertitudes plus ces dernières sont étendues. Deuxièmement, le niveau de confiance que l'on acquiert pour le code1 est proportionel à celui que l'on peut accorder au code de référence, en regard des preuves disponibles sur son bon comportement et des estimations d'incertitude qu'il nous fournit.

Tel que souligné par Trucano et al. (2003), la MCC ne devrait pas être utilisée seule pour accomplir la vérification de l'algorithme numérique d'un programme. Au mieux, elle peut faire partie d'un plan plus large d'activités de vérifications.

Dans le cas de CFDLIB, des résultats concluants ont déjà été obtenus pour des tests d'évaluation d'erreur sur des solutions analytiques (Reid, 2004). La MCC permet à présent d'observer le comportement du code sur des situations plus complexes que celles pouvant être résolues analytiquement. De plus, contrairement à la comparaison à des EDO reconnues, la comparaison des résultats peut être faite sur une base quantitative.

Cependant, aussi précis que soient les calculs numériques effectués sur le code de référence, la MCC n'est pas une démarche d'*évaluation* d'erreur, au sens où on l'entend généralement en vérification. Comme les évaluations d'erreur sont le cœur des activités de vérification de code, on peut se questionner sur la pertinence de classer la MCC (ou toute autre activité impliquant la comparaison avec des résultats numériques d'EDP, sans connaîssance de la solution exacte) comme une activité de *vérification* de code.

CONCLUSION

Dans le but d'implanter le logiciel CFDLIB à Biothermica, le processus de vérification et validation vise à s'assurer de la qualité des résultats obtenus par le code. Cela permet de prendre en compte le risque associé à son utilisation. Le présent travail consistait tout d'abord à proposer une procédure d'analyse des résultats qui rende compte de l'incertitude due au calcul numérique. De plus, cette procédure devait être appliquée à deux méthodes de vérification de code. Ces dernières devaient être mises à l'épreuve dans le cadre de la vérification de CFDLIB.

Ainsi, une stratégie d'analyse des solutions et des incertitudes a été sélectionnée, implémentée et appliquée aux études de vérification d'algorithmes. Elle comprend l'utilisation de la méthode des moindres carrés et de la méthode de Cadafalch et al. Cette stratégie permet de faire des comparaisons quantitatives des résultats, en tenant compte des incertitudes liées au calcul numérique. Elle permet d'appréhender le comportement global et les particularités locales des solutions. En ce sens, elle permet une analyse plus complète des résultats. En outre, elle pourrait facilement être appliquée ultérieurement à la vérification de calculs. Elle contribue donc à l'indentification du niveau de risque associé aux simulations.

Également, le présent projet permet de mieux comprendre les avenues disponibles pour la vérification de CFDLIB. Ainsi, deux méthodes de vérification d'algorithme ont été mises à l'épreuve : la comparaison à des solutions reconnues d'EDO et la comparaison de codes. Les écoulements considérés font appel à une plus grande partie des termes du modèle que ceux dont on peut connaître l'expression analytique. Toutefois, aucune des deux méthodes ne permet une vérification aussi complète que la méthode des solutions manufacturées, écartée en raison des ressources nécessaires à son implantation.

La comparaison à des solutions reconnues d'EDO s'est révélée inadéquate en regard des

objectifs de la vérification de code. En effet, les approximations utilisées pour obtenir la solution de référence induisent des différences dans le modèle qui sont difficilement quantifiables. Ainsi, les comparaisons ne permettent pas d'identifier si les résultats de CFDLIB convergent vers la solution des équations du modèle.

La méthode de comparaison de codes permet quant à elle des comparaisons quantitatives des résultats. De plus, elle peut être employée pour étudier des problèmes complexes. Toutefois, la méthode est limitée par la qualité des résultats du programme de référence. Ainsi, les écoulements pour lesquels le programme de référence montrera un mauvais comportement ne pourront être étudiés avec la MCC. De plus, la MCC est une méthode lourde qui nécessite que l'on obtienne un bon comportement en convergence pour trois solutions : le test de MMS, le problème dans le code de référence et le problème dans le code à vérifier.

L'application de la MCC sur l'écoulement circulatoire incompressible a permis de constater la concordance des résultats des deux codes. De plus, l'application de la MMS a permis de vérifier la convergence de la solution du programme de référence vers la solution du modèle (modifié par les termes sources). La solution de CFDLIB devrait donc converger vers une fonction qui se situe, par rapport à la solution du modèle, à l'intérieur de l'incertitude cumulative associée aux calculs numériques des deux programmes. Enfin, les prochaines applications pourraient utiliser un algorithme de comparaison d'ordre supérieur, pour tenter de réduire l'incertitude sur la solution de référence et sur la comparaison. En effet, les solutions de CFDLIB sur les maillages les plus grossiers étant déjà à l'intérieur des bandes d'incertitudes estimées sur les solutions Fluent les plus précises, rien ne peut nous indiquer que les solutions de CFDLIB se rapprochent réellement de la solution exacte du modèle lors des raffinements. On peut uniquement affirmer que la convergence de CFDLIB est cohérente avec les résultats que l'on attendrait d'un programme ayant un bon comportement en convergence. Bref, il est nécessaire d'utiliser les solutions les plus précises possibles.

Finalement, le travail effectué sur la MCC et sur l'analyse des résultats pourra être utile à des études de vérification futures, sur des modèles plus élaborés (3D, turbulence, etc.) et des géométries plus complexes. Toutefois, les temps de calculs limitent l'utilisation de CFDLIB à des écoulements relativement simples. Pour améliorer l'efficacité du programme et pour pouvoir aborder des problèmes plus ambitieux, on pourrait considérer l'utilisation de techniques adaptatives de génération de maillage, basées sur des estimations des erreurs locales. D'autre part, la MCC pourrait également être utile pour atteindre certains objectifs ne faisant pas partie de la vérification et validation du code. Par exemple, tel que mentionné par Trucano et al. (2003), la MCC pourrait être utilisée pour la calibration de modèles, pour mettre en évidence une anomalie et aider à déboguer le programme, ou encore pour contribuer à l'estimation de l'incertitude épistémique (due au manque de connaissances), par l'utilisation de plusieurs codes d'une manière analogue à l'utilisation de plusieurs sites d'expérimentation.

BIBLIOGRAPHIE

Abanto, J., Pelletier, D., Garon, A., Trépanier, J.-Y., & Reggio, M. (2005). Verification of some commercial CFD codes on atypical cfd problems. *43rd AIAA Aerospace Sciences Meeting and Axhibit*. Reno, NV.

Aeschliman, D. P., Oberkampf, W. L., & Blottner, F. G. (1995). A proposed methodology for computational fluid dynamics code verification, calibration, and validation. Sandia Report SAND–95-1189C, Sandia National Labs., Sandia National Labs., Albuquerque, NM.

AIAA (1998). AIAA guide for the verification and validation of computational fluid dynamics simulations. guide G-077-1998, American Institute of Aeronautics and Astronautics.

Andrade, E. N. d. C. (1939). The velocity distribution in a liquid-into-liquid jet. part 2: the plane jet. *Proceedings of the Physical Society*, *51*(5), 784–793.

Benay, R., Chanetz, B., & Délery, J. (2003). Code verification/validation with respect to experimental data banks. *Aerospace science and technology*, *7*(4), 239–262.

Bickley, W. B. (1937). The plane jet. *The Philosophical Magazine*, *23*, 727–31.

Cadafalch, J., Perez-Segarra, C. D., Cònsul, R., & Oliva, A. (2002). Verification of finite volume computations on steady-state fluid flow and heat transfer. *Journal of Fluids Engineering*, *124*(1), 11–21.

Chapman, D. R., Mark, H., & Pirtle, M. W. (1975). Computers vs. wind tunnels for aerodynamics flow simulations. *Aerospace America*, *13*(4), 22–35.

Coleman, H. W. (2002). Closure to "discussion: 'comprehensive approach to verification and validation of cfd simulations–part 1: Methodology and procedures' ". *Journal of Fluids Engineering*, *124*(3), 810.

Coleman, H. W. (2003). Some observations on uncertainties and the verification and validation of a simulation. *Journal of Fluids Engineering*, *125*(4), 733–735.

Coleman, H. W. (2006). An approach to verification and validation in computational fluid dynamics and heat transfer using experimental uncertainty analysis concepts. Consulté en déc. 2006, tiré de `https://www.dmso.mil/public/library/projects/vva/found_04/coleman_an_approach_to_vv_in_computational_etc.pdf`.

Coleman, H. W. & Stelle Jr., W. G. (1989). *Experimentation and uncertainty analysis for engineers*. Wiley-Interscience.

Coleman, H. W. & Stern, F. (1997). Uncertainties and CFD code validation. *Journal of Fluids Engineering*, *119*(4), 795–803.

Coleman, H. W. & Stern, F. (1998). author's closure. *Journal of Fluids Engineering*, *120*, 635–636.

DeVolder, B., Glimm, J., Grove, J. W., Kang, Y., Lee, Y., Pao, K., et al. (2002). Uncertainty quantification for multiscale simulations. *Journal of Fluids Engineering*, *124*(11), 29–41.

ERCOFTAC (2000). Best practice guidelines. Rapport technique, European research community on flow, turbulence and combustion.

Eça, L. & Hoekstra, M. (2000). An evaluation of verification procedures for CFD applications. IST Report D72-7, Insituto Superior Tecnico, Lisbon.

Eça, L. & Hoekstra, M. (2002). An evaluation of verification procedures for CFD applications. *24th Symposium on Naval Hydrodynamics*. Fukuoka, Japon.

Eça, L. & Hoekstra, M. (2003). An example of uncertainty estimation in the calculation of a 2-d turbulent flow. *MARNET-CFD Final Workshop*. Haslar, UK.

Eça, L. & Hoekstra, M. (éditeurs) (2004). *Workshop on CFD Uncertainty Analysis*, Instituto superior técnicio.

Flynn, M. R. & Eisner, A. D. (2004). Verification and validation studies of the time-averaged velocity field in the very near-wake of a finite elliptical cylinder. *Fluid Dynamics Research*, *34*(4), 273–288.

Freitas, C. J. (2002). The issue of numerical uncertainty. *Applied Mathematical Modelling*, *26*(2), 237–.

Harlow, F. H. & Welch, J. E. (1965). Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, *8*(12), 2182–2189.

Hatton, L. (1997). The T experiments: Errors in scientific software. *IEEE Computational Science and Engineering*, *4*(2), 27–38.

Ilinca, C., Zhang, X. D., Trépanier, J.-Y., & Camarero, R. (2000). A comparison of three error estimation techniques for finite-volume solutions of compressible flows. *Computer methods in applied mechanics and engineering*, *189*(4), 1277–1294.

Ilinca, F., Pelletier, D., & Garon, A. (1997). An adaptive finite element method for a two-equation turbulence model in wall-bounded flows. *International journal for numerical methods in fluids*, *24*(1), 101–120.

Industrie Canada (2006). Biothermica technologies inc. - profil complet. Consulté le 21 novembre 2006, Consulté en nov. 2006, tiré de `http://strategis.ic.gc.ca/app/ccc/search/navigate.do?language=fre\&portal=1\&subPortal=\&estblmntNo=123456132856\&profile=completeProfile`.

ISO (1993). *International vocabulary of basic and general terms in metrology*. International Organisation for Standardization, Genève, Suisse, 2^{e} édition.

ISO (1995). *Guide to the expression of uncertainty in measurement.* International Organisation for Standardization, Genève, Suisse.

Lacasse, D., Pelletier, D., & Turgeon, E. (2001). Prediction of turbulent separated flow in a turnaround duct using wall functions and adaptivity. *International Journal of Computational Fluid Dynamics, 15*(10), 209–225.

Lacasse, D., Turgeon, E., & Pelletier, D. (2004). On the judicious use of the $k-\epsilon$ model, wall functions and adaptivity. *International Journal of Thermal sciences, 43*(10), 925–938.

Magyari, E. & Keller, B. (1999). Heat transfer characteristics of the separation boundary flow induced by a continuous stretching surface. *Journal of Physics D: Applied Physics, 32*(22), 2876–81.

Mason, D. P. (2002). Group invariant solution and conservation law for a free laminar two-dimensional jet. *Journal of Nonlinear Mathematical Physics, 9*(supplément 2), 92–101.

Mei, C. C. (2002). Two dimensional laminar jet. [En ligne], Consulté en déc. 2005, tiré de `http://web.mit.edu/fluids-modules/www/highspeed_flows/3-4-lamjet.pdf`.

Oberkampf, W. L. (2002). Discussion: "Comprehensive approach to verification and validation of cfd simulations–part 1: Methodology and procedures". *Journal of Fluids Engineering, 124*(3), 809–810.

Oberkampf, W. L. & Trucano, T. G. (2002). Verification and validation in computational fluid dynamics. *Progress in Aerospace Sciences, 38*(3), 209–272.

Oberkampf, W. L., Trucano, T. G., & Hirsch, C. (2002). Verification, validation, and predictive capability in computational engineering and physics. *Applied Mechanics Reviews, 57*(5), 345–384.

Pelletier, D. & Roache, P. J. (2006). Chapitre 13. *Handbook of numerical heat transfer*, (pp. 417–442). John Wiley & sons, Inc., 2^{e} édition.

Pelletier, D., Turgeon, E., Lacasse, D., & Borggaard, J. (2003). Adaptivity, sensitivity, and uncertainty: Toward standards of good practice in computational fluid dynamics. *AIAA Journal*, *41*(10), 1925–1933.

Pelletier, D., Turgeon, E., & Tremblay, D. (2004). Verifiaction and validation of impining round jet simulations using an adaptive fem. *International Journal for numerical methods in fluids*, *44*(7), 737–763.

Reid, T. (2004). Vérification d'un code de mécanique des fluides 3d. Rapport de projet de fin d'études en aéronautique, École Polytechnique de Montréal & Biothermica Technologies Inc.

Richardson, L. F. & Gaunt, J. A. (1927). The deferred approach to the limit. *Transaction of the Royal Society of London, Series A: Mathematical and Physical Sciences*, *226*, 299–361.

Roache, P. J. (1994). Perspective: A method for uniform reporting of grid refinement studies. *Journal of Fluid Engineering, Trans. ASME*, *116*, 405–413.

Roache, P. J. (1995). Verification of codes and calculations. *The 26^{t}h AIAA Fluid Dynamics Conference*, AIAA 95–2224. June 19–22, San Diego, USA.

Roache, P. J. (1998a). Discussion. *Journal of Fluids Engineering*, *120*, 635.

Roache, P. J. (1998b). *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers, Albuquerque, New Mexico, USA.

Roache, P. J. (2002a). Code verification by the method of manufactured solutions. *Journal of Fluids Engineering*, *124*(1), 4–10.

Roache, P. J. (2002b). Recent contributions to verification and validation methodology, keynote address. *Proceedings of th 5th World Congress on Computational Mechanics*. Vienna, Austria: Minisymposium on Verification and Validation II.

Roache, P. J. (2004). Building PDE codes to be verifiable and validatable. *Computing in Science & Engineering*, *6*(4), 30–38.

Salari, K. & Knupp, P. (2000). Code verification by the method of manufactured solutions. Sandia Report SAND2000-1444, Sandia National Laboratories, Albuquerque, New Mexico.

Schlichting, H. (1933). Laminare strahlausbildung. *Zeitschrift für Angewandte Mathematik und Mechanik*, *13*, 260–263.

Schlichting, H. & Gersten, K. (2000). *Boundary Layer Theory*. Springer-Verlag Berlin.

Slater, J. W. (2005). Examining spatial (grid) convergence. [En ligne], Consulté en 19 mai 2006, tiré de `http://www.grc.nasa.gov/WWW/wind/valid/tutorial/spatconv.html`.

Stern, F., Wilson, R., & Shao, J. (2004). Quantitative V&V of cfd simulations and certification of CFD codes with examples. *Proceedings of CHT-04, ICHMT International Symposium on advances in Computational heat transfer*. Norway.

Stern, F. & Wilson, R. V. (2002). Closure to "Discussion: 'Comprehensive approach to verification and validation of cfd simulations–part 1: Methodology and procedures' ". *Journal of Fluids Engineering*, *124*(3), 810–811.

Stern, F., Wilson, R. V., Coleman, H. W., & Paterson, E. G. (2001). Comprehensive approach to verification and validation of CFD simulations–part 1: Methodology and procedures. *Journal of Fluids Engineering*, *123*(4), 793–802.

Trucano, T. G., Pilch, M., & Oberkampf, W. L. (2003). On the role of code comparaisons in verification and validation. Sandia Report SAND2003-2752, Sandia National Laboratories, Albuquerque, New Mexico.

Turgeon, E. & Pelletier, D. (2002). Verification and validation in CFD using an adaptive finite-element method. *Journal aéronautique et spatial du Canada*, *48*(4), 260–3.

Wilson, R. V., Stern, F., Coleman, H. W., & Paterson, E. G. (2001). Comprehensive approach to verification and validation of CFD simulations–part 2: Application for rans simulation of a cargo/container ship. *Journal of Fluids Engineering*, *123*(4), 803–810.

ANNEXE I

GLOSSAIRE

Cas Variante d'un problème donné, obtenue en définissant la valeur des paramètres du problème. Ex. : *Écoulement turbulent sur une marche descendante,* $Re = 1000$*;* $h/D = 0.15$.

CFD Acronyme de *Computational fluid dynamics*. Mécanique des fluides assistée par ordinateur.

Dynamique Se dit d'un test ou d'une série de tests ayant pour but de révéler les failles d'un programme au moyen de son exécution.

EDO Équations différentielles ordinaires.

EDP Équations aux dérivées partielles.

Erratique Dont la manifestation, l'importance et les conséquences sont irrégulières.

Erreur Différence entre le résultat d'une mesure et la vraie valeur du mesurande (Coleman (2003) d'après ISO (1993, 1995)).

Incertitude Paramètre, associé au résultat d'une mesure, qui caractérise la dispersion des valeurs qui pourraient raisonnablement être attribuées au mesurande (Coleman (2003) d'après ISO (1993, 1995)).

Modèle *Représentation d'un système physique ou d'un procédé, dont le rôle est d'améliorer notre capacité à comprendre, prédire ou contrôler son comportement* (AIAA, 1998).

Problème Situation d'intérêt décrite de manière générale par un certain nombre de paramètres d'influence. Ex. : *Écoulement turbulent sur une marche descendante.*

Statique Se dit d'un test ou d'une série de tests ayant pour but de révéler les failles d'un programme sans nécessiter son exécution.

Triplet Ensemble de trois maillages de concentration différente, choisis pour être utilisés dans une étude de convergence.