

UNIVERSITÉ DE MONTRÉAL

VF++ : UN ENVIRONNEMENT ORIENTÉ-OBJET
POUR LA MÉTHODE DES VOLUMES FINIS

JUAN ABANTO FLORIDA
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIAE DOCTOR (Ph.D.)
(GÉNIE MÉCANIQUE)

MARS 1998

©Juan Abanto Florida, 1998.



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

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

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

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

0-612-32981-X

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

VF++ : UN ENVIRONNEMENT ORIENTÉ-OBJET
POUR LA MÉTHODE DES VOLUMES FINIS

présentée par : ABANTO FLORIDA Juan

en vue de l'obtention du diplôme de : Philosophiae Doctor

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

M. PRUD'HOMME Michel, Ph.D., président

M. CAMARERO Ricardo, Ph.D., membre et directeur de recherche

M. REGGIO Marcelo, Ph.D., membre et codirecteur de recherche

M. GHALY Wahid, Ph.D., membre

M. TRÉPANIÉ Jean-Yves, Ph.D., membre

*À mes chers parents et frères,
à mes maîtres.*

«- Fais, Sancho, dit don Quichotte; fais la vérification que je t'ai dite, et ne t'embarrasse pas d'autre chose. Tu ne sais pas un mot de ce que sont les colures, les lignes, les parallèles, les zodiaques, les écliptiques, les pôles, les solstices, les équinoxes, les planètes, les signes, les degrés, les mesures dont se composent la sphère céleste et la sphère terrestre. Si tu connaissais toutes ces choses, ou même une partie, tu verrais clairement combien de parallèles nous avons parcourus, combien de constellations nous laissons derrière nous. Mais, je te répète, tâte-toi, cherche partout, car j'imagine que tu es plus propre et plus net à cette heure qu'une feuille de papier blanc.

Sancho se tâta donc, et, baissant tout doucement la main sous le pli du jarret gauche, il releva la tête, regarda son seigneur, et dit :

- Ou l'expérience est fausse, ou nous ne sommes pas arrivés à l'endroit que dit Votre Grâce, ni même à bien des lieues de là. >

EL INGENIOSO HIDALGO DON QUIJOTE
DE LA MANCHA. MIGUEL DE CERVANTES
SAAVEDRA

Remerciements

Je tiens tout d'abord à remercier le professeur Ricardo Camarero, mon directeur de recherche, pour son soutien, sa patience, sa disponibilité, l'attention qu'il m'a prodiguée tout au long de mes études et pour le soin qu'il a mis dans la correction des versions de cette thèse. Je tiens en particulier à souligner son soutien financier qui m'a permis de mener à terme mes études.

Je remercie mon codirecteur le professeur Marcelo Reggio, pour sa confiance, conseils, discussions et soutien opportuns qui m'ont permis d'entreprendre ces études.

Je veux aussi remercier les professionnels et personnel du Centre de recherche en calcul appliqué (CERCA) pour leur assistance opportune. De façon particulière je remercie Paul Labbé pour ses conseils et sa disponibilité.

Je cite aussi mes collègues étudiants, ou anciens étudiants de l'École polytechnique de Montréal, qui m'ont aidé avec des conseils, encouragements et soutien, dans cette bataille qu'est une thèse de doctorat. Je dis merci beaucoup à Samir Rida pour ses conseils et contributions fructifères dans le contenu de mon travail; à André Hess, Stéphane Thienot, Cristian Ilinca et frères, Serge Eby, Luc Gravel, Roberto Marrano, Xudong Zhang, Golnaz Baradaran, Marie-Isabelle Farinas, Ali Benmeddour et tous les amis qui font une longue liste, pour leur amitié et leur encouragement. Particulièrement, j'exprime ma reconnaissance et gratitude à André Hess pour sa confiance et participation dans la correction minutieuse et finale de ce document.

Je dois remercier messieurs les membres du jury, pour les commentaires intéressants, positifs et judicieux qui m'ont permis d'améliorer le contenu de cette thèse.

Il me faut aussi dire merci à mes pourvoyeurs de fonds : le Centre de recherche en calcul appliqué (CERCA) et le Conseil de recherche en sciences naturelles et en

génie du Canada (CRSNG) dont les subventions ont permis à mes professeurs de me financer.

Je dois, finalement, exprimer mon extrême et éternelle gratitude et reconnaissance à ma famille : mes parents, frères et sœur, représentés par mon frère Tito, son épouse Ana et leurs enfants, pour leurs encouragements, soutien et compréhension qui m'ont été réellement très indispensables.

Résumé

***VF++* : Un environnement orienté-objet pour la méthode des volumes finis**

La méthode des volumes finis, comme technique de résolution des formulations intégrales des équations de convection-diffusion, a un haut degré de liberté du point de vue de son support (géométrique et algébrique) autant que de l'application même. Plusieurs codes scientifiques reposant sur cette méthode et développés jusqu'à ce jour essayent d'accomplir partiellement cette lourde tâche.

Ce code, générique et évolutif, est une tentative d'intégration de cette problématique avec une approche orientée-objet permettant sa réutilisation et offrant un éventail de types de volumes de contrôle pour la solution et un choix plus vaste du côté de la résolution. Les résultats ainsi obtenus permettent un jugement plus objectif dans la prédiction des phénomènes physiques et en même temps confèrent aux volumes de contrôle le centre d'intérêt de la recherche à propos de la méthode des volumes finis.

Ce système est constitué de trois modules : le premier (appelé *vfParser*) est l'interprète du maillage initial, le deuxième (*vfGeo*) est le préprocesseur géométrique et le dernier (*vf*) est le résolveur. Le maillage initial peut être formé d'éléments hybrides.

Les deux premiers modules sont chargés de bâtir le maillage de volumes de contrôle selon la dimension du problème et le stockage ultérieur de la variable (CCCV, ECCV, FCCV ou CCCV) et le troisième consiste en un ensemble d'outils pour la résolution de problèmes. Ce dernier module reproduit deux applications immédiates :

La première utilise le schéma de *Roe* pour la résolution multidimensionnelle des équations d'*Euler* en régime compressible de façon explicite ou implicite, au premier ou au deuxième ordre et pour tout type de volumes de contrôle. La deuxième application utilise la technique SIMPLER d'intégration dans la résolution des équations de *Navier-Stokes* en régime incompressible avec des volumes de contrôle décalés.

Du côté de l'implantation orientée-objet, ce code a été écrit en *C++*. Dans ce contexte des propriétés comme la *data abstraction* et des stratégies comme le *dynamic binding* ont été employées pour assurer l'intégration optimale de la problématique.

Cette étude intègre une discussion multidisciplinaire sur la méthode à partir d'une nouvelle structure de données et sa conclusion principale est basée sur l'oracle : *«Il n'existe que CCCVs*»*.

Abstract

FV++ : A Generic Object-Oriented Environment for Finite Volume Methods

The finite volume method as a discretisation method for convection-diffusion problems has a big degree of freedom from point of view of its support (geometrical and numerical) and its applications. The scientific codes that use this method try to satisfy only one part of this complicated problem.

This code is an effort of integration for the question that involves the method with an object oriented approach. The main advantage of this generic workbench environment is its large degree of freedom of control volumes choice and numerical schemes to solve the proposed equations. This aspect puts into consideration the numerical results which never have been obtained by other codes. Simultaneously, this approach places the control volumes as a center of topological interest for the finite volume researches.

The code has three modules: the first one is the parser recognizer for the interpretation of the primary mesh (called `vfParser`), the second one is an interface based geometrical pre-processor (`vfGeo`) and the last one is the numerical integrator solver called `vf`. The primary mesh can be a hybrid set of grids.

The first two modules build the new topology for any control volume scheme (VCCV, FCCV, ECCV or CCCV) or dimension. The third module reproduces two immediate applications: The first one uses the Roe formulation to solve multidimensional Euler equations (in explicit or implicit, first or second order schemes) for compressible flow. This is done for all kinds of control volumes. The second application uses

the SIMPLER strategy in order to solve the incompressible Navier-Stokes equations using staggered grids.

Its object-oriented implementation concept is based on the C++ programming language. In this context data abstraction properties and dynamic binding strategies are employed to provide solution procedures of optimal computational complexity and integration.

This approach drives us to multidisciplinary discussion about the finite volume method with a new data structures and its principal conclusion is based on the oracle : "There are only one Cell Centered Control Volumes".

Table des matières

Remerciements	v
Résumé	vii
<i>Abstract</i>	ix
Table des matières	xi
Liste des tableaux	xvii
Liste des figures	xix
Liste des notations	xxv
Liste des annexes	xxviii
Liste des codes	xxix
Avant-propos	xxxi
CHAPITRES	
1 Introduction	1
1.1 Le système VF++	1
1.2 La Méthode des volumes finis	3
1.2.1 La conservativité	4
1.2.2 Équations discrètes	5

1.2.3	Distribution des flux	6
1.3	La problématique	8
1.3.1	Le processus numérique de la méthode	8
1.3.2	Supports de la méthode	8
1.3.3	Degrés de liberté de la méthode	9
1.3.4	Le développement des codes de volumes finis	10
1.3.5	Objectif de cette étude	11
1.4	Évolution et conception du système	13
1.4.1	Pourquoi séparer les modules?	13
1.4.2	Influence des autres sources sur VF++	15
1.5	Analyse orientée-objet	16
1.5.1	Représentation du monde réel	16
1.5.2	Choix du langage de programmation	16
1.6	Les avantages du système	17
2	Interfaces	19
2.1	Introduction	19
2.2	Modes de fonctionnement du système	21
2.2.1	Fonctionnement continu	21
2.2.2	Fonctionnement différé	22
3	L'interface parser, vfParser	23
3.1	Introduction	23
3.2	La syntaxe du fichier Vie	23
3.3	Exemple d'un fichier Vie en 2D	25
3.4	Exemple d'un fichier Vie en 3D	25
4	Interface géométrique, vfGeo	29
4.1	Introduction	29
4.2	Topologie du maillage initial	30
4.2.1	Définitions importantes	30

4.2.2	Univers des entités	31
4.2.3	Relations entre entités	33
4.3	Topologie du maillage de volume de contrôle	33
4.3.1	Définitions préliminaires	34
4.3.2	Conditions des <i>éléments</i> et des <i>bords</i>	34
4.3.3	Les volumes de contrôle	35
4.3.4	Les bords de contrôle	36
4.3.5	Relations entre entités	39
4.3.6	Concepts amorphes	40
4.4	Les tâches de <i>vfGeo</i>	40
4.4.1	Interprétation du maillage initial	40
4.4.2	Construction des entités initiales	41
4.4.3	Construction du voisinage	44
4.4.4	Numérotation locale des <i>éléments</i> initiaux	45
4.4.5	Stockage des données et gestion de l'interface	46
4.4.6	Construction des bords de contrôle	48
4.5	Correspondance des <i>codes</i> aux frontières	52
4.6	Validation de la transformation	52
5	Stencils et protocoles	53
5.1	Introduction	53
5.2	Définitions	53
5.2.1	Associations	55
5.2.2	Stencils $\mathcal{S}_f$ par omission	56
5.3	Protocoles	56
5.3.1	PROTOCOLE 1	59
5.3.2	PROTOCOLE 2	60
5.4	<i>Miscellanea</i>	62
5.4.1	Correspondances entre <i>vfGeo</i> et <i>vf</i>	62
5.4.2	Les bords de contrôle décalés	63
5.4.3	Applications	63

6	L'interface résolveur, <i>vf</i>	64
6.1	Introduction	64
6.2	Le <i>kernel</i> de l'interface	66
6.2.1	Création de la nouvelle topologie	66
6.2.2	Création de variables et inconnues abstraites	72
6.2.3	Génération des équations algébriques	75
6.2.4	Gestion de la résolution des équations	77
6.3	Supports opérationnels	77
6.3.1	L'intégration	78
6.3.2	Distribution des flux	81
6.3.3	Fonctions d'interpolation	81
6.3.4	La fonction LSQ générique	82
6.3.5	Mécanismes d'accès aux inconnues	83
6.4	Applications	83
6.4.1	Applications du PROTOCOLE 1	83
6.4.2	Applications du PROTOCOLE 2	86
7	Résultats	94
7.1	Introduction	94
7.2	Résultats de l'application du PROTOCOLE 1	96
7.2.1	Le problème de l'onde réfléchie	97
7.2.2	Écoulements transsoniques autour d'un profil NACA 0012	103
7.2.3	Écoulement à l'intérieur d'un cylindre	116
7.2.4	Provocation d'ondes de choc dans une tuyère convergente-divergente	137
7.2.5	Écoulement supersonique autour d'un profil à double ellipse	151
7.3	Résultats de l'application du PROTOCOLE 2	156
7.3.1	Le problème de <i>Smith</i> et <i>Hutton</i>	157
7.3.2	Le problème de la cavité carrée dynamique	166
7.3.3	Le problème de la marche	173

8 Conclusions et recommandations	180
8.1 Conclusions	180
8.1.1 Sur le système	180
8.1.2 Sur les applications	181
8.2 Recommandations	182
Épilogue	182
RÉFÉRENCES	185
ANNEXES	
A Arbres de dérivation des classes du système	192
A.1 Classes générales	192
A.2 Classes de l'interface parser	192
A.3 Classes de l'interface géométrique	193
A.4 Classes de l'interface résolveur	193
B Représentation BNF de l'interface <i>vfParser</i>	194
B.1 Non terminaux :	194
B.2 Terminaux :	195
C Relations entre les entités topologiques	196
D Numérotation locale des arêtes et des faces des formes géométriques en 3D	198
E Tableaux de construction des bords de contrôle	200
F Variables et inconnues	203
F.1 Les classes <i>VAR</i> et <i>INC</i>	203
F.2 La classe <i>VF_INC_SCALAIRE</i>	203
F.3 Accès aux inconnues	204
F.3.1 Accès à partir d'un élément	204

F.3.2	Accès à partir d'une variable de référence	205
G	Application du schéma de <i>Roe</i> avec le PROTOCOLE 1	207
G.1	Le schéma de <i>Roe</i>	207
G.2	Structure des données principales	207
G.3	Stratégies du calcul des flux	207
G.4	Imposition des conditions aux frontières dans le PROTOCOLE 1	209
G.5	Le programme principal de l'application	209
H	Application des schémas décalés dans la résolution des équations de <i>Navier-Stokes</i> en régime incompressible avec le PROTOCOLE 2	212
H.1	Intégration générique avec le PROTOCOLE 2	212
H.2	Distribution générique	212
H.3	Création de la matrice de convection avec le schéma <i>MWUS</i> pour les polygones	215
H.4	Le programme principal de l'application	215
I	Les outils de <i>VF++</i>	219
I.1	Les fichiers du système <i>VF++</i>	219
I.1.1	Le script <i>VF</i>	219
I.1.2	Le script <i>VFM</i>	220
I.1.3	Le script <i>VF2VU</i>	220
I.1.4	Le programme <i>vf2vu</i>	220
I.1.5	Le programme <i>Grid</i>	220
I.1.6	Le programme <i>Vie</i>	220
I.2	Les répertoires de l'utilisateur	221
J	Les messages d'erreur	222
K	Le format du fichier <i>*.SOL</i>	223

Liste des tableaux

1.1	<i>Liste (non exhaustive) de codes en volumes finis</i>	12
3.1	<i>Syntaxe du vfParser</i>	24
4.1	<i>Types de volumes de contrôle</i>	35
4.2	<i>Pointeurs au voisinage</i>	44
4.3	<i>Codes aux frontières établis par omission</i>	52
5.1	<i>$\mathcal{R}(\tilde{\Phi})$ en 2D, pour les stencils intégrals $\mathcal{S}_f$</i>	54
5.2	<i>$\mathcal{R}(\tilde{\Phi})$ en 3D, pour les stencils intégrals $\mathcal{S}_f$</i>	55
5.3	<i>Gestion d'envoi/réception des mailles de volume de contrôle</i>	62
5.4	<i>Exemple d'envoi et de réception pour un schéma décalé</i>	63
6.1	<i>Transformation de topologies</i>	65
6.2	<i>La matrice rectangulaire du flux, $\mathbf{J}$</i>	81
6.3	<i>Équations vectorielles de Navier-Stokes en régime incompressible</i>	87
7.1	<i>Comparaisons des résultats du Cas-test 7.8 (marche) pour $Re = 800$</i>	179
C.1	<i>Relations initiales entre entités</i>	196
C.2	<i>Relations finales entre entités</i>	197
D.1	<i>Numérotation locale des entités d'un tétraèdre</i>	198
D.2	<i>Numérotation locale des entités d'un prisme</i>	199
D.3	<i>Numérotation locale des entités d'un hexaèdre</i>	199
E.1	<i>Construction des bords internes en 2D</i>	200

E.2	<i>Construction des bords-frontières en 2D</i>	200
E.3	<i>Construction des bords internes en 3D</i>	201
E.4	<i>Construction des bords-frontières en 3D</i>	202
K.1	<i>Format du fichier *.SOL</i>	223

Liste des figures

1.1	<i>Principe de conservativité</i>	5
1.2	<i>Distribution des flux</i>	7
1.3	<i>Pourquoi séparer les modules ?</i>	13
1.4	<i>Mémoire requise par le préprocesseur géométrique et le résolveur dans la version actuelle de VF++ pour le même cas-test illustré dans la figure 1.3</i>	14
1.5	<i>Comportement de VF++ durant l'exécution dans le temps</i>	15
2.1	<i>Interaction des interfaces</i>	20
2.2	<i>Deux modes de fonctionnement du système</i>	21
3.1	<i>Exemples de maillages initiaux hybrides</i>	25
4.1	<i>Fonction de l'interface géométrique</i>	29
4.2	<i>Propriétés des bords de contrôle</i>	38
4.3	<i>Relations principales appliquées à la discrétisation finale</i>	39
4.4	<i>Arbre de dérivation des entités topologiques</i>	42
4.5	<i>Numérotation locale d'une face</i>	45
4.6	<i>Numérotation locale des sommets des formes géométriques en 3D</i>	46
4.7	<i>VCCVs, ECCVs et CCCVs en 2D</i>	49
4.8	<i>Construction des volumes de contrôle en 3D</i>	50
4.9	<i>Bords de contrôle en 3D (élément tétraédrique)</i>	51
5.1	<i>Stencils intégrals et domaine de validité de $\tilde{\Phi}$ en 2D</i>	57
5.2	<i>Esquisses de stencils intégrals en 3D</i>	58

5.3	<i>Exemple d'utilisation des protocoles en 2D</i>	60
5.4	<i>Les protocoles dans VF++ avec des exemples graphiques en 2D</i>	62
6.1	<i>La conception de l'interface résolveur</i>	65
6.2	<i>Arbre de dérivation de la classe FV_POINT</i>	66
6.3	<i>Arbre de dérivation de la classe abstraite VF_BORD</i>	68
6.4	<i>Exemple de stockage des bords pour le PROTOCOLE 2</i>	69
6.5	<i>Les éléments-frontières</i>	71
6.6	<i>Inconnues satellites dans la discrétisation finale des équations</i>	76
6.7	<i>Les bords de contrôle de Saabas</i>	80
6.8	<i>Exemple d'accès aux inconnues</i>	84
6.9	<i>Représentation décalée de l'application en 2D</i>	88
7.1	<i>Domaine de calcul et conditions pour le Cas-test 7.1</i>	97
7.2	<i>Maillages 2D utilisés dans le Cas-test 7.1</i>	99
7.3	<i>Le Maillage initial 7.2 (ou final 7.7)</i>	100
7.4	<i>Distribution du nombre de Mach pour le Cas-test 7.1 avec le Maillage initial 7.1. Convergence de calcul</i>	101
7.5	<i>Visualisation des valeurs calculées du nombre de Mach pour le Cas-test 7.1 avec le Maillage initial 7.2. Convergence de calcul</i>	102
7.6	<i>Maillages 2D utilisés dans le Cas-test 7.2</i>	104
7.7	<i>Le Maillage initial 7.5 (ou final 7.14) utilisé dans le Cas-test 7.2</i>	105
7.8	<i>Lignes iso-Mach pour le Cas-test 7.2 avec le Maillage initial 7.3. $M_\infty = 0.8$, $\alpha = 0^\circ$. Premier ordre. Comparaisons de C_p et convergence de calcul</i>	108
7.9	<i>Lignes iso-Mach pour le Cas-test 7.2 avec le Maillage initial 7.3. $M_\infty = 0.8$, $\alpha = 1.25^\circ$. Premier ordre. Comparaisons de C_p et convergence de calcul</i>	109
7.10	<i>Lignes iso-Mach pour le Cas-test 7.2 avec le Maillage initial 7.3. $M_\infty = 0.95$, $\alpha = 0^\circ$. Premier ordre. Comparaisons de C_p et convergence de calcul</i>	110

7.11	<i>Lignes iso-Mach pour le Cas-test 7.2 avec le Maillage initial 7.3. $M_\infty = 0.8$, $\alpha = 0^\circ$. Deuxième ordre. Comparaisons de C_p et convergence de calcul</i>	111
7.12	<i>Lignes iso-Mach pour le Cas-test 7.2 avec le Maillage initial 7.3. $M_\infty = 0.8$, $\alpha = 1.25^\circ$. Deuxième ordre. Comparaisons de C_p et convergence de calcul</i>	112
7.13	<i>Lignes iso-Mach pour le Cas-test 7.2 avec le Maillage initial 7.3. $M_\infty = 0.95$, $\alpha = 0^\circ$. Deuxième ordre. Comparaisons de C_p et convergence de calcul</i>	113
7.14	<i>Surfaces iso-Mach pour le Cas-test 7.2 avec le Maillage initial 7.5 (hexaèdres). $M_\infty = 0.8$, $\alpha = 0^\circ$. Premier ordre. Convergence de calcul</i>	114
7.15	<i>Le "fish tail". Distribution du nombre de Mach. $M_\infty = 0.95$, $\alpha = 0^\circ$. Maillage initial 7.4</i>	115
7.16	<i>Description géométrique et conditions du problème</i>	116
7.17	<i>Illustration des positions des coupes sur les contours de C_p à la paroi</i>	120
7.18	<i>Résultats du Cas-test 7.3 avec le Maillage initial 7.8. Contours de C_p à la paroi. Coupes et convergence. $\alpha = 5^\circ$</i>	124
7.19	<i>Résultats du Cas-test 7.3 avec le Maillage initial 7.8. Contours de C_p à la paroi. Coupes et convergence. $\alpha = 10^\circ$</i>	125
7.20	<i>Résultats du Cas-test 7.3 avec le Maillage initial 7.8. Contours de C_p à la paroi. Coupes et convergence. $\alpha = 15^\circ$</i>	126
7.21	<i>Cas-test 7.3. Iso-surfaces de C_p à $\alpha = 15^\circ$ avec le Maillage initial 7.8 (tétraèdres)</i>	127
7.22	<i>Résultats du Cas-test 7.3 avec le Maillage initial 7.6 (hexaèdres). Variation du C_p. Coupes. $\alpha = 5^\circ$</i>	128
7.23	<i>Résultats du Cas-test 7.3 avec le Maillage initial 7.6 (hexaèdres). Variation du C_p. Coupes. $\alpha = 10^\circ$</i>	129
7.24	<i>Résultats du Cas-test 7.3 avec le Maillage initial 7.6 (hexaèdres). Variation du C_p. Coupes. $\alpha = 15^\circ$</i>	130
7.25	<i>Résultats du Cas-test 7.3 avec le Maillage initial 7.7 (prismes). Variation du C_p. Coupes. $\alpha = 5^\circ$</i>	131

7.26	<i>Résultats du Cas-test 7.3 avec le Maillage initial 7.7 (prismes). Variation du Cp. Coupes. $\alpha = 10^\circ$</i>	132
7.27	<i>Résultats du Cas-test 7.3 avec le Maillage initial 7.7 (prismes). Variation du Cp. Coupes. $\alpha = 15^\circ$</i>	133
7.28	<i>Résultats du Cas-test 7.3 avec le Maillage initial 7.8. Contours de Mach dans le plan d'incidence. $\alpha = 5^\circ$</i>	134
7.29	<i>Résultats du Cas-test 7.3 avec le Maillage initial 7.8. Contours de Mach dans le plan d'incidence. $\alpha = 10^\circ$</i>	135
7.30	<i>Résultats du Cas-test 7.3 avec le Maillage initial 7.8. Distribution du nombre de Mach dans le plan d'incidence. $\alpha = 15^\circ$</i>	136
7.31	<i>Géométrie du domaine de calcul</i>	138
7.32	<i>Domaine de calcul et les mailles finales, provenant du Maillage initial 7.9, utilisées dans le Cas-test 7.4</i>	140
7.33	<i>Domaine de calcul et les mailles finales, provenant du Maillage initial 7.10, utilisées dans le Cas-test 7.4</i>	141
7.34	<i>Distribution du nombre de Mach pour le Cas-test 7.4 avec le Maillage initial 7.9 (quadrilatères). $\frac{p_2}{p_o} = 0.25$. Convergence de calcul</i>	145
7.35	<i>Distribution du nombre de Mach pour le Cas-test 7.4 avec le Maillage initial 7.9 (quadrilatères). $\frac{p_2}{p_o} = 0.4$. Convergence de calcul</i>	146
7.36	<i>Distribution du nombre de Mach pour le Cas-test 7.4 avec le Maillage initial 7.10 (triangles). $\frac{p_2}{p_o} = 0.25$. Convergence de calcul</i>	147
7.37	<i>Distribution du nombre de Mach pour le Cas-test 7.4 avec le Maillage initial 7.10 (triangles). $\frac{p_2}{p_o} = 0.4$. Convergence de calcul</i>	148
7.38	<i>Visualisation des iso-surfaces du nombre de Mach pour le Cas-test 7.4 avec le Maillage initial 7.11 (tetraèdres). $\frac{p_2}{p_o} = 0.25$. Convergence de calcul</i>	149
7.39	<i>Visualisation des iso-surfaces du nombre de Mach pour le Cas-test 7.4 avec le Maillage initial 7.11 (tetraèdres). $\frac{p_2}{p_o} = 0.4$. Convergence de calcul</i>	150
7.40	<i>Maillages 2D utilisés pour le Cas-test 7.5</i>	152
7.41	<i>Visualisation, constante par volume, de la vitesse horizontale du Cas-test 7.5. $M_\infty = 2$, $\alpha = 20^\circ$. Maillage initial 7.12</i>	153
7.42	<i>Iso-vitesses U du Cas-test 7.5. $M_\infty = 2$, $\alpha = 20^\circ$. Maillage initial 7.12</i>	154

7.43	<i>Convergence de calcul pour le Cas-test 7.5</i>	155
7.44	<i>Géométrie du problème de Smith & Hutton avec des lignes de courant</i>	158
7.45	<i>Domaine de calcul et les maillages 2D initiaux utilisés dans le Cas-test 7.6</i>	159
7.46	<i>Mailles des volumes de contrôle créées à partir du Maillage initial 7.13 et utilisées dans le Cas-test 7.6</i>	160
7.47	<i>Mailles des volumes de contrôle créées à partir du Maillage initial 7.14 et utilisées dans le Cas-test 7.6</i>	160
7.48	<i>Convergence de calcul typique pour le Cas-test 7.6</i>	161
7.49	<i>Cas-test 7.6. $Pe=1E6$. Maillage initial 7.13</i>	162
7.50	<i>Cas-test 7.6. $Pe=1E6$. Maillage initial 7.14</i>	163
7.51	<i>Cas-test 7.6 en 3D. $Pe=1E6$. Schéma FOUS. Maillage final 7.46, VCCV</i>	164
7.52	<i>Domaine de calcul et conditions pour le Cas-test 7.7</i>	166
7.53	<i>Domaine de calcul et les mailles finales utilisées pour le Cas-test 7.7</i>	168
7.54	<i>Cas-test 7.7. Schéma décalé 7.1. MWUS. $Re=1E3$. Maillage initial 7.16</i>	169
7.55	<i>Cas-test 7.7. Schéma décalé 7.2. MWUS. $Re=1E3$. Maillage initial 7.17</i>	170
7.56	<i>Cas-test 7.7. Schéma décalé 7.1. MWUS. $Re=1E3$. Maillage initial 7.17</i>	171
7.57	<i>Coupes pour le Cas-test 7.7. Différents maillages initiaux et schémas décalés. Comparaisons</i>	172
7.58	<i>Domaine de calcul et conditions pour le Cas-test 7.8</i>	174
7.59	<i>Cas-test 7.8. Maillage initial 7.18. Schéma décalé 7.1. MWUS</i>	176
7.60	<i>Cas-test 7.8. Maillage initial 7.19 hybride. Schéma décalé 7.1. MWUS</i>	177
7.61	<i>Cas-test 7.8. Maillage initial 7.18. Schéma décalé 7.2. MWUS</i>	178
8.1	<i>Arbre binaire des volumes de contrôle en 2D, obtenu de la régénération</i>	184
A.1	<i>Classes générales</i>	192
A.2	<i>Classes de vfParser</i>	192
A.3	<i>Classes de vfGeo</i>	193
A.4	<i>Arbres de dérivation de vf</i>	193
F.1	<i>Les classes VAR, INC et leur descendance</i>	203

F.2	<i>Accès à une inconnue à partir d'un élément</i>	204
F.3	<i>Associativité entre variables et inconnues</i>	205
G.1	<i>La variable et inconnue de Roe</i>	208
G.2	<i>L'interface virtuelle ROE et associations</i>	209
H.1	<i>Description de l'application de MWUS sur un pentagone</i>	215
I.1	<i>Répertoires du système</i>	221

Liste des notations

Dynamique des fluides

- $\mathbf{V}$ — Variable vectorielle de la vitesse, $\mathbf{V} = \{u, v, w\}$.
- Φ — Propriété convectante (u, v, w, T, etc).
- $\mathbf{U}$ — Variables conservatives d'Euler, $\mathbf{U} = \{ \rho, \rho u, \rho v, \rho w, \rho E \}$.
- S^Φ — Terme source rapporté à Φ .
- $\mathbf{J}$ — Flux de convection-diffusion de Φ .
- Γ_Φ — Coefficient de diffusion de Φ .
- ∇ — Opérateur d'*Hamilton*.
- α, β — Angles de la direction de la vitesse à l'infini.
- ρ — Densité.
- p — Pression.
- γ — Rapport de chaleurs spécifiques.
- M — Nombre de Mach.
- Pe — Nombre de Peclet.
- Re — Nombre de Reynolds.

Variables et inconnues abstraites

- Φ, Φ — Variables scalaire et vectorielle.
- $\tilde{\Phi}, \tilde{\Phi}$ — *Idem*, de projection.
- Φ_i, Φ_i — Inconnues scalaire et vectorielle habitant dans l' i -ème élément.
- $\tilde{\Phi}_i, \tilde{\Phi}_i$ — *Idem*, de projection habitant dans l' i -ème résidence $\mathfrak{R}$.

Topologie

S, A, F, V, B, E	–	Sommet, arête, face, volume, bord ou élément.
$S, \mathcal{A}, \mathcal{F}, \mathcal{V}, \mathcal{B}, \mathcal{E}$	–	Sous-ensembles de S, A, F, V, B ou E .
$S, \mathbf{A}, \mathbf{F}, \mathbf{V}, \mathbf{B}, \mathbf{E}$	–	L'univers de S, A, F, V, B ou E sur $\mathcal{D}$.
s, a, f, v, b, e	–	Les cardinalités des S, A, F, V, B ou E .
$\mathcal{D}$	–	Domaine géométrique ou de calcul.
D	–	Sous-domaine de calcul, $D \subset \mathcal{D}$.
Ω_Φ	–	Volume de contrôle de la variable ou inconnue Φ .
$\mathfrak{R}$	–	Résidence.
$\mathfrak{R}(\tilde{\Phi})$	–	Résidence de $\tilde{\Phi}$.
$\mathfrak{R}(\Phi)$	–	Résidence de Φ .
S	–	Norme d'un bord.
S_f	–	Stencil intégral.
S_p	–	Stencil projectif.
V	–	Volume d'un élément.
n	–	Normale d'un bord.

Sigles

VCCV	–	<i>Vertex Centered Control Volume.</i>
ECCV	–	<i>Edge Centered Control Volume.</i>
FCCV	–	<i>Face Centered Control Volume.</i>
CCCV	–	<i>Cell Centered Control Volume.</i>
MWUS	–	<i>Mass Weighted Upwind Scheme.</i>
FOUS	–	<i>Flow Oriented Upwind Scheme.</i>
UGEO	–	<i>Upwind géométrique.</i>
SIMPLER	–	<i>Semi-Implicit Method for Pressure-Linked Revised.</i>
BNF	–	<i>Backus Naur Form.</i>
VF ⁺⁺	–	Le code générique.
<i>vfParser</i>	–	Interface parser.
<i>vfGeo</i>	–	Interface géométrique.
<i>vf</i>	–	Interface résolveur.

Synonymes

- bord — Bord de contrôle.
élément — Volume de contrôle.

Indices

- L — Gauche du bord.
 R — Droite du bord.
 ∞ — variables à l'infini.

Environnements numérotés

- Cas – test — Cas-test de validation.
Remarque — Description des remarques.
Définition — Définitions adoptées.
Maillage initial — Maillage de départ.
Maillage final — Maillages finaux obtenus.
Observation — Description d'une observation sur un Cas-test.
Règle générale — Règle générale valide pour tout le système.
Schéma décalé — Type de schéma décalé utilisé.

Liste des annexes

A	Arbres de dérivation des classes du système	192
B	Représentation BNF de l'interface	194
C	Relations entre entités topologiques	196
D	Numérotation locale des arêtes et des faces des formes géométriques en 3D	198
E	Tableaux de construction des bords de contrôle	201
F	Variables et inconnues	203
G	Application du schéma de Roe avec le PROTOCOLE 1	207
H	Application des schémas décalés dans la résolution des équations de <i>Navier-Stokes</i> en régime incompressible avec le PROTOCOLE 2	212
I	Les outils de <i>VF++</i>	219
J	Les messages d'erreur	222
K	Le format du fichier <i>*.SOL</i>	223

Liste des codes

3.1	<i>Exemple du fichier Vie en 2D</i>	26
3.2	<i>Exemple du fichier Vie en 3D</i>	27
3.3	<i>Exemple du fichier Vie en 3D (suite)</i>	28
4.1	<i>Accès au vfParser</i>	41
4.2	<i>La classe abstraite VF_POINT</i>	44
4.3	<i>La classe VF_MESH</i>	47
6.1	<i>La classe FV_ELEMENT</i>	67
6.2	<i>La classe VAR</i>	73
6.3	<i>Les classes VF_TERME et VF_EQUATION</i>	77
6.4	<i>Intégration et distribution avec le PROTOCOLE 1</i>	79
6.5	<i>Intégration dans tout le domaine pour le PROTOCOLE 1</i>	79
6.6	<i>La fonction LSQ</i>	83
F.1	<i>La classe VF_INC_SCALAIRE</i>	204
F.2	<i>Accès à une inconnue à partir d'un élément</i>	205
F.3	<i>Accès à une inconnue à partir d'une variable et de son index</i>	206
F.4	<i>Accès à une inconnue à partir d'une variable et de son élément</i>	206
G.1	<i>Programme principal dans l'intégration explicite</i>	210

H.1	<i>L'intégration générique</i>	213
H.2	<i>La distribution avec le PROTOCOLE 2</i>	214
H.3	<i>Génératrice de la matrice de convection pour le schéma MWUS pour des polygones</i>	216
H.4	<i>Exemple de programme principal dans l'intégration des équations de Navier-Stokes en régime incompressible avec la méthode SIMPLER .</i>	217

Avant-propos

VF++ fait partie d'un ensemble de procédures pour la résolution de problèmes numériques à l'aide de la méthode des volumes finis. Dans ce rapport de fin d'études doctorales le lecteur trouvera la description générale de toutes les démarches suivies dans la conception et la mise en œuvre du code **VF++**.

Suite à l'Introduction générale de ce document, le chapitre 2 est dédié à un survol des interfaces de **VF++** et de leurs interactions.

Les chapitres 3, 4 et 6 correspondent aux définitions et aux descriptions principales de la conception des interfaces du système **VF++**. Finalement le chapitre 7 décrit les résultats préliminaires suivis de leur discussion.

Dans ce document il existe des environnements numérotés relatifs aux concepts, remarques et supports employés. Cela suit strictement les hypothèses et considérations prises lors du design et de la conception du système, ainsi qu'elles sont introduites, dans cet ouvrage, pour faciliter la tâche de rédaction. Ces environnements sont :

1. **<Cas-test>** Description d'un Cas-test pour la vérification du code.
2. **<Définition>** Description des définitions et des concepts employés dans la conception du système; elle a un caractère subjectif.
3. **<Maillage initial>** Description d'un maillage initial.
4. **<Maillage final>** Description d'un maillage final.
5. **<Observation>** Description d'une observation relative à un Cas-test.
6. **<Règle générale>** Description d'une règle générale employée dans l'implantation; elle a un caractère objectif.

7. **<Remarque>** Description de certaines remarques.
8. **<Schéma décalé>** Description d'un maillage décalé.

Quant à la rédaction même, ce document a été écrit avec $\text{\LaTeX 2}_{\epsilon}$ et les environnements VF.sty. La description des types utilisés est comme suit :

- *TYPE*, pour les identificateurs de classes.
- *type ou TYPE*, pour les énoncés dans la description d'un environnement (*voir* la liste ci-dessus).
- *type ou TYPE*, pour certains identificateurs.
- *type ou TYPE*, pour les mots et expressions en anglais, ou les descriptions des figures et des tableaux, ou certains identificateurs importants.
- *type ou TYPE*, pour l'insertion de lignes de code.
- ... , pour substituer les parties non substantielles des codes.

Chapitre 1

Introduction

1.1 Le système $VF++$

L'objectif principal de ce travail est la mise en œuvre d'une machine de résolution de problèmes numériques à l'aide de la méthode des volumes finis en utilisant une nouvelle perspective de structures de données. Cette démarche offre une base au développement ultérieur appuyée sur des concepts plus génériques et d'avant-garde.

Comme élément décisif dans l'évolution du système $VF++$ on trouve la médiation permanente avec l'analyse, l'abstraction et la synthèse de l'information concernant la méthode. Cette approche a abouti à un code robuste, très compact et générique obtenu grâce à ses structures de données orientées-objet utilisées dans tous ses modules. À cet égard $VF++$ est un code qui a évolué dans le conflit entre la performance et la généralisation. Toutefois, ce système bien qu'étant évolutif, est encore loin d'atteindre la généralisation totale.

Sur le plan de l'implantation, il s'agit, premièrement, de préparer les outils pour le passage d'une topologie *primaire* vers une autre *secondaire*. Deuxièmement,

la mise en œuvre, dans le cadre de l'environnement **VF++**, de deux applications spécifiques en Mécanique des fluides.

Le premier aspect vise à la transformation d'un maillage initial vers un maillage final de volumes de contrôle et constitue la clef dans la généralisation de cette proposition.

Le deuxième volet traite de deux applications : La première est axée sur la résolution multidimensionnelle, explicite ou implicite, au premier ou au deuxième ordre de précision, des équations d'*Euler* en régime compressible. Ce développement est générique par rapport au *stockage de la variable* et la dimension du problème permettant un très haut degré de liberté dans le choix de solution. La résolution multidimensionnelle des équations de *Navier-Stokes* en régime incompressible en utilisant des volumes de contrôle décalés et l'intégration avec la méthode SIMPLER constitue la deuxième application.

Sur le plan académique cet environnement a pour rôle d'être un point de départ en fournissant des outils pour la résolution des équations d'*Euler* et/ou *Navier-Stokes* avec des techniques plus avancées de programmation. Une extension de ces techniques est abordée dans [47] et [49], ainsi que le mode d'emploi de **VF++** dans [50].

Dans ce document, après la description des aspects fondamentaux du fonctionnement du système dans le chapitre 2, on présente les structures de données et la conception de chaque composante du système : **vfParser** (chapitre 3), **vfGeo** (chapitre 4) et **vf** (chapitre 6). Dans le chapitre 7, on expose les résultats obtenus des deux applications.

1.2 La Méthode des volumes finis

Les formulations conservatives des équations aux dérivées partielles ont le grand avantage de pouvoir être reformulées de façon intégrale à l'aide du Théorème de la divergence¹. La méthode des volumes finis est, donc, une technique de discrétisation obtenue de cette formulation intégrale.

Les phénomènes de transport d'une propriété quelconque en Mécanique des fluides, Transfert de chaleur, parmi d'autres, sont gouvernés par les principes de conservation de la masse, du mouvement et de l'énergie. Cet ensemble de propriétés peut être représenté par l'intermédiaire d'une seule équation régissant le transport de la propriété scalaire générique Φ et décrite par l'équation différentielle de convection-diffusion instationnaire :

$$\frac{\partial(\rho\Phi)}{\partial\tau} + \nabla \cdot \mathbf{J} = S^\Phi \quad (1.2)$$

avec :

$$\mathbf{J} = \rho \Phi \mathbf{V} - \Gamma_\Phi \nabla \Phi \quad (1.3)$$

Ici :

ρ — la densité.

Φ — la propriété convectante (u , v , w , T , etc).

¹ L'intégrale de volume de la divergence d'un champ vectoriel $\mathbf{F}$ étendu sur un volume Ω est égale à l'intégrale de contour de la projection de ce champ sur la normale traversant la frontière Γ :

$$\int_{\Omega} \operatorname{div} \mathbf{F} \, d\Omega = \oint_{\Gamma} \mathbf{F} \cdot \mathbf{n} \, d\Gamma \quad (1.1)$$

où Γ limite Ω .

- S^Φ – le terme source rapporté à Φ .
- $\mathbf{J}$ – le flux de convection-diffusion de Φ .
- $\mathbf{V}$ – la vitesse.
- Γ_Φ – le coefficient de diffusion de Φ .
- ∇ – l'opérateur d'*Hamilton*.

Les équations (1.2) et (1.3) donnent l'*équation généralisée du transport*:

$$\frac{\partial(\rho\Phi)}{\partial\tau} + \text{div}(\rho\Phi\mathbf{V}) = \text{div}(\Gamma_\Phi \nabla\Phi) + S^\Phi \quad (1.4)$$

Cette équation, englobant plusieurs équations dépendantes de Φ et étant sous la forme divergente, permet l'application du Théorème de la divergence (1.1). Le résultat de cette application est l'*équation généralisée intégrale du transport*:

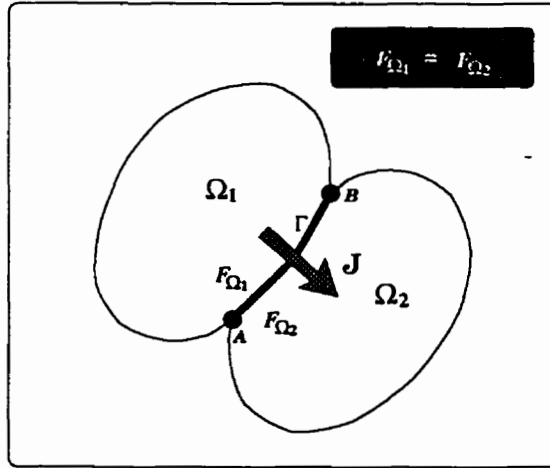
$$\frac{\partial}{\partial\tau} \int_\Omega \rho \Phi \, d\Omega + \oint_\Gamma \mathbf{J} \cdot \mathbf{n} \, d\Gamma = \int_\Omega S^\Phi \, d\Omega \quad (1.5)$$

Remarque 1.1 (Formulation intégrale) L'utilisation du Théorème de la divergence (1.1) est le support fondamental dans le passage vers la forme intégrale de l'équation de convection-diffusion (1.5) utilisée dans la formulation de la méthode.

1.2.1 La conservativité

Le principe de conservativité est la loi fondamentale de la méthode des volumes fins. Selon elle la variation de la propriété Φ dépend du *flux* net $\mathbf{J}$ traversant la frontière Γ (bord de contrôle, voir Définition 4.9) qui enveloppe le volume Ω . Cette propriété dérive directement de l'interprétation physique du Théorème de la divergence (1.1).

Le long d'une frontière (un segment infinitésimal, limité par A et B en 2D,

FIGURE 1.1: *Principe de conservativité*

Cette caractéristique est illustré à la figure 1.1.

1.2.2 Équations discrètes

L'interprétation géométrique de l'équation (1.5) ne pose pas de restrictions quant à la géométrie des frontières d'intégration. Donc, pour un volume de contrôle Ω_i quelconque consistant de m bords, et si on considère la propriété Φ et le terme source S comme constants sur le volume, la discrétisation de l'équation intégrale (1.5) donne :

$$[(\rho\Phi)_i^{n+1} - (\rho\Phi)_i^n] \Omega_i + \sum_{j=1}^m \mathbf{J}_j^{n+k} \mathbf{n}_j \Delta s_j \Delta \tau = (S_i^\Phi)^{n+k} \Omega_i \Delta \tau \quad (1.7)$$

Dans cette équation l'indice k sert à définir le genre de discrétisation temporelle, selon la liste :

- $k = 0$ — pour un schéma explicite.
- $k = \frac{1}{2}$ — pour un schéma semi-implicite.
- $k = 1$ — pour un schéma implicite.

ou une face infinitésimale en 3D), la relation :

$$\int_A^B \mathbf{J} \cdot \mathbf{n} d\Gamma = - \int_B^A \mathbf{J} \cdot \mathbf{n} d\Gamma \quad (1.6)$$

nous ramène à soutenir que, pour un domaine discret donné, les flux *internes* sont toujours annulés, en assurant ainsi la conservation de Φ .

Pour un schéma explicite on obtient :

$$(\rho\Phi)_i^{n+1} = (\rho\Phi)_i^n - \frac{\Delta\tau}{\Omega_i} \sum_{j=1}^m \mathbf{J}_j^n \mathbf{n}_j \Delta s_j + (S_i^\Phi)^n \Delta\tau \quad (1.8)$$

et pour une schéma implicite on obtient l'équation [28] :

$$a_i \Phi_i^{n+1} - a_i^0 \Phi_i^n + \sum_{j=1}^m (a_{ij} \Phi_j^{n+1}) = b_i \quad (1.9)$$

avec les a_i comme coefficients dépendants de la *distribution du flux* et b_i une constante.

Remarque 1.2 (Forme des volumes de contrôle) *L'interprétation géométrique de l'intégrale de volume et de contour dans l'équation (1.5) ne pose aucune restriction quant au choix de la forme du volume de contrôle Ω , pouvant être utilisés des polygones ou des polyèdres. Cela a des conséquences extrêmement positives du côté de la résolution des équations, mais qui fréquemment génère un problème dans la mise en œuvre du support géométrique.*

Remarque 1.3 (Hypothèse) *L'hypothèse lors de l'obtention de l'équation discrète (1.7) de l'équation de convection-diffusion (1.5) a été de considérer la propriété Φ et le terme source S comme constants sur le volume de contrôle.*

1.2.3 Distribution des flux

Une autre étape dans l'évaluation de l'équation (1.7) est la *distribution des flux*. Ce processus se réalise de façon systématique pour tous les volumes de contrôle du domaine. Ladite distribution, pour un volume de contrôle Ω_o et pour 6 autres voisins, partageant les mêmes bords, est illustrée dans la figure 1.2.

Étant donné l'importance de ce processus dans l'intégration des équations (1.5) et (1.6) on y fera appel pendant la conception générale de **VF++** dans les prochains chapitres.

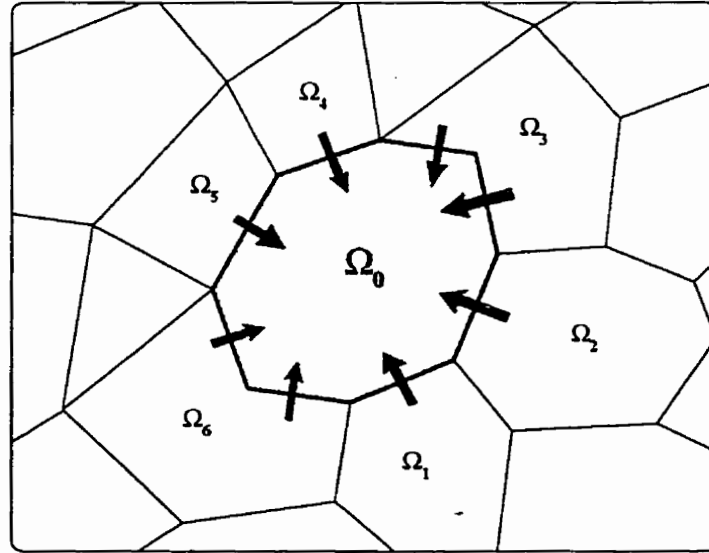


FIGURE 1.2: *Distribution des flux*

Remarque 1.4 (Intégration et distribution) Suite au calcul des flux correspondants et d'autres paramètres de l'équation discrète (1.7), on doit distribuer cette information aux volumes de contrôle voisins, après l'application des conditions aux frontières pertinentes. Ce processus est basé sur le principe de conservativité exprimé par l'équation (1.6).

Remarque 1.5 (Ingéniosité numérique) La méthode des volumes finis est, donc, une méthode de résolution de l'équation discrète (1.7) de façon ingénieuse, naturelle et même intuitive.

1.3 La problématique

1.3.1 *Le processus numérique de la méthode*

Ce processus est divisé en trois étapes consécutives :

1. l'*intégration* sur les bords,
2. la *distribution*,
3. la *résolution*.

L'*intégration* numérique, explicite ou implicite, a rapport au calcul des flux de convection-diffusion des propriétés Φ aux bords. La *distribution* (sous-section 1.2.3) assure le respect du principe de conservativité en ajoutant le flux calculé au volume gauche et droit. Les deux premières étapes (Remarque 1.4) deviennent générales et intrinsèques à la méthode. Le dernier processus suit la stratégie de solution prise pour la résolution de l'équation.

1.3.2 *Supports de la méthode*

Les processus décrits dans la sous-section précédente sont basés sur deux types de supports :

1. un support *géométrique*,
2. un support *algébrique*.

Le premier support est basé sur le maillage de volumes de contrôle limité par les bords. Le deuxième, appuyé sur le premier, permet l'application de la mathématique inhérente à la méthode.

1.3.3 Degrés de liberté de la méthode

La méthode des volumes finis possède un large éventail de degrés de liberté basé principalement selon :

1. la *discrétisation* du domaine,
2. la *formulation*,
3. la *résolution*.

La discrétisation du domaine

Lors de la discrétisation spatiale il faut considérer trois aspects :

1. Suite à la Remarque 1.2 quant à la forme et la dimension des volumes de contrôle, le domaine discret peut être divisé en partitions selon la *résidence de la variable*, par rapport à un maillage initial de polygones et polyèdres. Cette résidence, traitée dans la sous-section 4.3.3, est l'endroit d'intérêt topologique (*sommet, arête, face* ou *volume*) du stockage ultérieur de la variable.
2. En fonction de la formulation et des stratégies de résolution, ce domaine discret peut être divisé en sous-couches de différents types de volumes de contrôle, ce qui permet la résolution des équations avec des maillages *décalés*.
3. Emploi des maillages initiaux *hybrides*² dans le domaine.

La formulation

Du point de vue de la variable primitive, la formulation intégrale (équation (1.5) et Remarque 1.1) peut utiliser deux types de variables :

² Le maillage de volume de contrôle est déjà hybride.

1. Une variable *scalaire* Φ . *Exemple:* dans la résolution de $\nabla^2 \Phi = 0$.
2. Une variable *vectorielle* Φ . *Exemple:* dans la résolution de $\nabla^2 \Phi = 0$.

Ladite formulation et la résolution peuvent demander l'utilisation d'une variable discrète appelée variable *projective*, $\tilde{\Phi}$, lorsqu'il faut faire intervenir un stencil dans les calculs.

La résolution

La résolution de la formulation peut être basée sur l'intégration explicite, semi-implicite ou implicite dans le temps. Le calcul des termes convectifs peut utiliser différents types de schémas *upwind* en fonction de la nature du phénomène.

1.3.4 *Le développement des codes de volumes finis*

Durant la dernière décennie, le développement des codes de volumes finis obéit à des règles du marché et porte sur l'application restreinte des degrés de liberté qu'offre la méthode, discutés dans la sous-section précédente. Les codes sont donc dédiés à la résolution d'un seul type d'équations avec un seul type de schéma ou stockage de la variable et une dimension spécifique du problème. Par conséquent, une autre structure de données, une mise en œuvre plus étroite du support algébrique ou un autre code est nécessaire pour accomplir d'autres exigences. Du point de vue des résultats obtenus et leurs conclusions, ceux-ci deviennent discernables et subjectifs, sans avoir d'autres éléments de comparaison.

En fonction de la structure de données et du langage de développement, la mise à jour et le maintien de ces codes deviennent difficiles dans la compréhension des interfaces qui interagissent. Toutefois, du point de vue des supports de la méthode,

traités dans la sous-section 1.3.2, les codes ainsi conçus ont le grand avantage d'avoir un support algébrique plus riche que son support géométrique.

Une liste incomplète de codes et leurs caractéristiques est donné par le tableau 1.1, où nous avons divisé les codes de volumes finis en deux groupes :

1. Les programmes, programmes de recherche et développement.
2. Les packages, ensemble de systèmes de recherche et développement ou commerciaux.

On peut remarquer que la majorité de ces codes sont classés par la règle restrictive: *«à chaque application un type de maillage initial, un type de volume de contrôle, un type d'équations à résoudre et une technique de résolution»*.

1.3.5 *Objectif de cette étude*

Une solution à cette problématique (soulevée dans la sous-section 1.3.3) a toujours été un défi autant du point de vue des outils de développement que de l'abstraction même dû à la complexité que demande sa intégration et généralisation.

Toutefois, les langages de programmation *orientés-objet* de la dernière génération ouvrent une fenêtre à l'intégration de cette problématique grâce à leurs propriétés permettant la représentation d'un problème avec le recours de l'abstraction.

C'est dans ce contexte que cette étude place ladite intégration comme le centre d'intérêt de cette recherche avec une approche basée sur la programmation orientée-objet, comme technique de résolution.

TABLEAU 1.1: Liste (non exhaustive) de codes en volumes finis

<i>Status</i>	<i>Code</i>	<i>Auteur</i>	<i>Langage</i>	<i>NL^a</i>	<i>TE^b</i>	<i>Stockage</i>	<i>MAI^c</i>	<i>EQNS^d</i>
Programmes	VF	Rida S. [32]	FORTTRAN	12000	3	décalé	Tri ^e	NSi ^f , comb, ...
	VFM	Abanto J.	FORTTRAN	2840	2	décalé	Qua ^g	NSi, CN ^h , ...
	vf	Ilinca, A. [18]	FORTTRAN	8760	3	CCCV	Tetra ⁱ	Eu
	CFS	Zhang X. et al.	FORTTRAN	-	7	CCCV	Tri	EuNSc ^j , ...
	NSC2KE	Bijan M. [8]	FORTTRAN	3130	3	VCCV	Tri	EuNSc
	NaSt3D	Wagner, E. [45]	C	6230	2	décalé	Hexa	NSi, CN
	...	...	...	...	...	...	...	...
	VF++	Abanto J.	C++	17000	0	multiple	multiple	open
Packages	UG	Bastian P. [6]	C	338600	2	multiple	TQT ^k	MS ^l , NSi, ...
	kaskade	Beck, R. et al. [7]	C++	29700	2	-	TT ^m	multiple, ...
	PHYSICA	Bailey, C. et al. [3]	C++	-	1	CCCV	TT	MS, NSi, ...
	STAR-CD	STAR-CD	FORTTRAN	-	10	CCCV	multiple	multiple
	...	...	...	...	...	...	...	...

^a NL - Nombre de lignes.

^b TE - Temps d'exploitation, [années].

^c MAI - Maillage initial.

^d EQNS - Équations.

^e T - Triangles.

^f NSi - Navier-Stokes incompressible.

^g Qua - Quadrilatère.

^h CN - Convection naturelle.

ⁱ Tetra - Tetraèdre.

^j EuNSc - Euler et Navier-Stokes compressible.

^k TQT - Triangles, Quadrilatères et Tetraèdres.

^l MS - Mécanique structurelle.

^m TT - Triangles, Tetraèdres.

1.4 Évolution et conception du système

1.4.1 Pourquoi séparer les modules ?

Les premières versions de *VF++* utilisaient les concepts d'environnements intégrés, où tout le système fonctionnait comme un seul module. Ceci obligeait, à l'intérieur du même code, de supporter des structures de données différentes pour chaque type de volume de contrôle. La mémoire requise pour satisfaire les différents schémas était colossale, surtout pour les ECCVs.

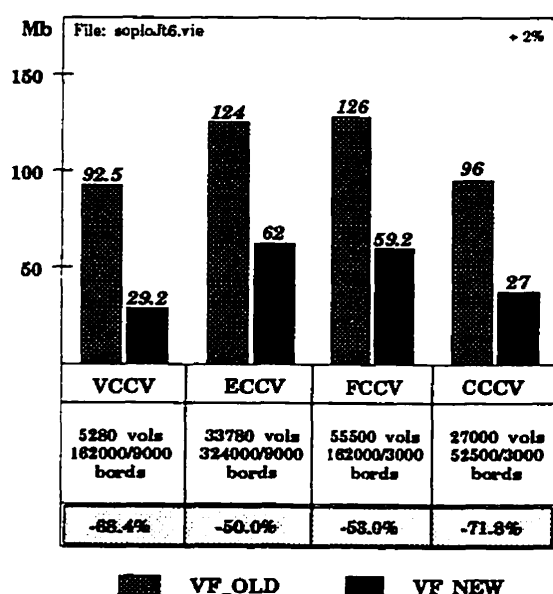


FIGURE 1.3: Pourquoi séparer les modules ?

Après une longue étude sur la performance, la mise à jour et l'évolution future de la méthode des volumes finis, on est arrivé à une conclusion très importante: afin de mieux représenter le problème posé sur le plan théorique de la méthode et afin de bien exploiter la mémoire, il était indispensable de séparer le problème en deux modules:

Le premier appelé *préprocesseur géométrique* générique ayant pour

tâche la génération de mailles de volume de contrôle et, subséquemment, de bords de contrôle, et l'autre le *résolveur*, avec des outils pour la résolution et un ensemble bien filtré d'informations géométriques. Le premier est totalement divorcé du support mathématique qui se retrouve en entier dans le deuxième.

Durant la période de mise à jour un nombre considérable de tests orientés vers l'obtention d'un code performant avec la nouvelle approche ont été accomplis. Parmi ces tests il faut mentionner celui qui est résumé dans la figure 1.3 appliqué à un domaine 3D avec 27,000 tétraèdres pour la résolution explicite des équations d'*Euler* en régime compressible³ avec l'ancienne version (intégrée) et intermédiaire (séparée) du système.

L'état actuel du système *VF++*, quant à la mémoire requise pour le même cas-test, est fidèlement reflété dans la figure 1.4. Peu importe, donc, la mémoire requise pour le préprocesseur géométrique le plus important est de la garder pour la résolution.

Cette prise de décision a joué un autre rôle important, celui de fournir l'information minimale nécessaire à l'utilisateur pour ses applications en employant ce préprocesseur comme une *boîte noire* du système.

Avec cette séparation, le comportement du système *VF++* dans le temps présente une caractéristique très importante illustrée par l'histogramme qualitatif d'usage de la mémoire, figure 1.5, où l'on fait apparaître *vfGeo*⁴ comme étant le préprocesseur géométrique (de très courte durée) et *vf*⁵ comme le résolveur.

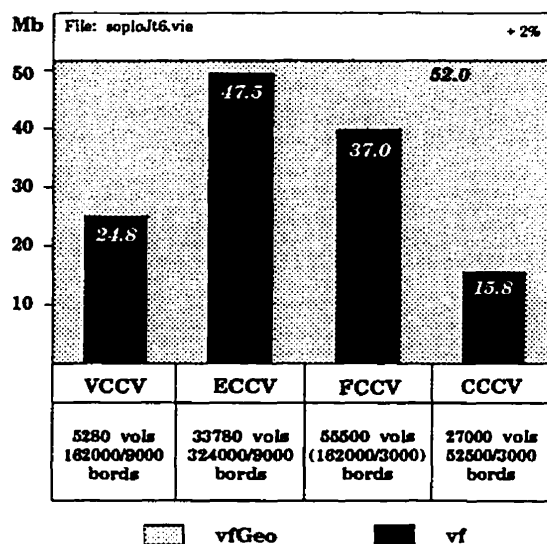


FIGURE 1.4: *Mémoire requise par le préprocesseur géométrique et le résolveur dans la version actuelle de VF++ pour le même cas-test illustré dans la figure 1.3*

³ Stockage vectoriel.

⁴ Voir le chapitre 4.

⁵ Voir le chapitre 6.

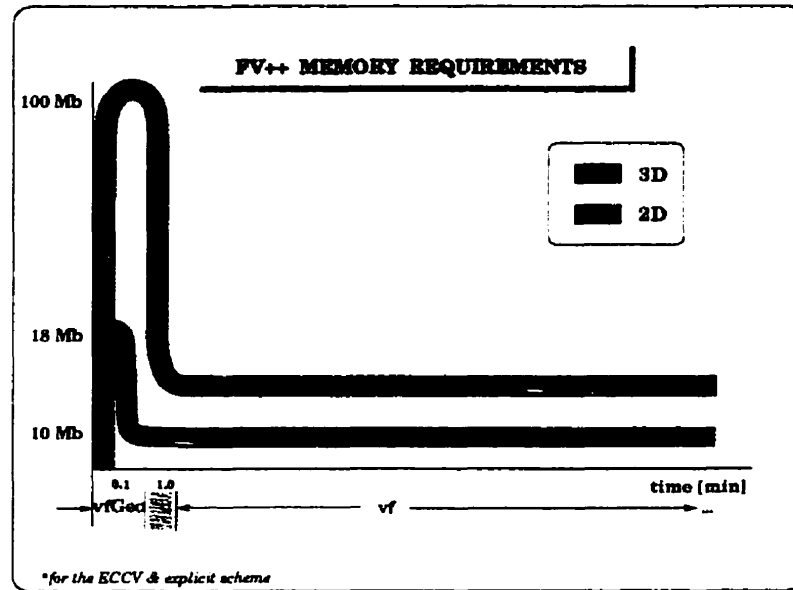


FIGURE 1.5: *Comportement de VF++ durant l'exécution dans le temps*

1.4.2 *Influence des autres sources sur VF++*

La version actuelle de **VF++** est, donc, un code qui est le produit d'une succession de contradictions et qui a évolué entre le déséquilibre de la performance et la généralisation. L'évolution qualitative des autres sources n'a jamais été ignorée dans ce processus. Pendant la période de son évolution, et surtout dans les dernières étapes, il a été enrichi par l'expérience, les méthodes et les structures de données des codes plus avancées ([7, 35, 45, 44, 6], voir le tableau 1.1) du point de vue de leur mise en œuvre.

1.5 Analyse orientée-objet

1.5.1 *Représentation du monde réel*

La philosophie orientée-objet est basée principalement sur le fait qu'à chaque objet abstrait est associé un ensemble d'*attributs* et un ensemble de *comportements*. Donc, il s'agit de la représentation *sui generis* du monde réel à l'aide de la modélisation de ces propriétés et leur interaction [57]. À plus haut niveau cette représentation porte le nom de *data hiding* et *data abstraction* [60, 61] lorsqu'on incorpore ladite association à chaque objet.

Un exemple de cette abstraction est le concept de *point*, qui, dans ce cas il est une entité topologique primaire contenant comme *attribut* deux informations : sa position (coordonnées) et son index (numéro), et comme *comportements* un ensemble de processus accédant aux attributs ou servant à les modifier.

1.5.2 *Choix du langage de programmation*

Le choix du langage de programmation est conditionné dans la mesure où celui-ci est capable de représenter de la meilleure façon ce monde réel en suivant une règle générale : *«de l'abstrait au concret»*. C'est à dire, dans quelle mesure est-il capable de supplanter la problématique de la méthode des volumes finis par un ensemble d'attributs et comportements avec un exercice d'abstraction.

Durant les dernières années les langages de programmation ont essayé d'accomplir cette tâche selon les demandes du marché. Quelques-uns sont à l'origine des langages orientés-objet purs comme *Smalltalk*, d'autres ont réussi, dans un délai très

court, à se placer dans le frêle équilibre de la compétence. D'autres, à l'origine des langages procéduraux et bifurcatifs sont devenus «orientés-objet». Il est difficile, jusqu'à présent, de conjecturer sur des langages de programmation autres que *Smalltalk* qui puissent correctement représenter ce monde réel; pourtant dans cette évolution se trouvent principalement *C++* et *Java* franchissant les difficultés émanant des applications numériques.

C'est dans ce cadre d'évolution linguistique qu'on a choisi un langage de propos général comme *C++* [58, 59] comme outil de développement de *VF++* dû à ses capacités opérationnelles et la croissance de son utilisation dans la mise en œuvre d'autres codes scientifiques⁶. Ce langage fournit des mécanismes profitables grâce à ses propriétés inhérentes aux structures de données orientées-objet: la dérivation des classes, le polymorphisme, l'abstraction et l'encapsulation de l'information. Ces propriétés permettent, entre autres, d'assurer le développement d'un code robuste, générique, réutilisable et compact. Le code de *VF++* démontre que notre choix était le meilleur parmi les autres alternatives.

1.6 Les avantages du système

Parmi les caractéristiques les plus remarquables de *VF++*, il faut mentionner :

- **La dimension du problème:** 2D ou 3D.
- **Le maillage initial:** À la limite hybride : mélange de différent type de polygones (en 2D) ou mélange de tétraèdres, prismes et/ou hexaèdres (en 3D).

⁶ Ce langage, malgré ses faiblesses en face des structures, de surcharge des opérateurs et les problèmes qui en dérivent, parmi d'autres, a été employé avec succès dans les codes scientifiques comme *kaskade* [7], pour les résolveurs matriciels comme *SparseLib++* [56], *LAPACK++* [53], les bibliothèques standards comme *STL* [55], etc.

- **Les types de stockage:** Tous, indexés selon le tableau 4.1, pouvant aussi générer des volumes de contrôle décalés.
- **Deux modules exécutables:** Un pour le préprocesseur géométrique (*vfGeo*) et un autre pour le résolveur (*vf*).
- **Ses structures de données:** orientées-objet réutilisables.
- **Deux PROTOCOLES:** Régissant l'envoi/réception de bords de contrôle et la résolution, (chapitre 5).
- **Les outils du système:** Afin de simplifier la tâche de l'utilisateur (annexe I de la page 219).

Chapitre 2

Interfaces

2.1 Introduction

Le système **VF++** est constitué de trois sous-systèmes ou interfaces :

- vfParser* – l'*interface parser*, l'interprète du maillage initial.
- vfGeo* – l'*interface géométrique*, le préprocesseur géométrique.
- vf* – l'*interface résolveur*, le résolveur.

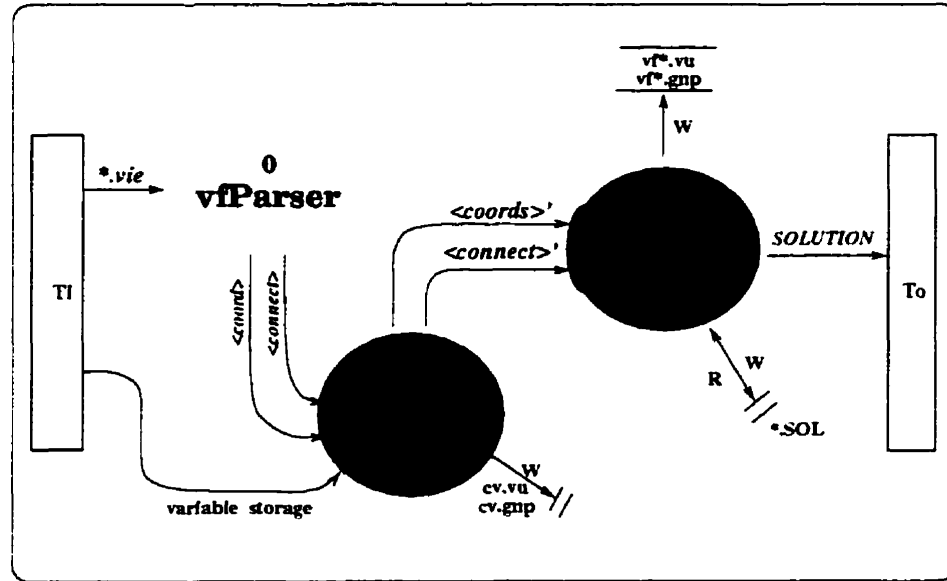
Ces trois interfaces interagissent selon le diagramme des flux de données illustré à la figure 2.1.

D'après cette figure un flux de données stockées dans un fichier en format *Vie*¹ avec la description du maillage initial est lu et interprété par *vfParser*, en suivant les règles de syntaxe de sa grammaire².

Cette information (*coordonnées* et *connectivités*) est utilisée par le deuxième sous-système (l'interface géométrique) qui joue le rôle de préprocesseur géométrique de **VF++**. En stockant les *sommets*, *arêtes*, *faces* et *volumes*, en construisant leur

¹ Voir le chapitre 3.

² Voir la section 3.2.

FIGURE 2.1: *Interaction des interfaces*

voisinage et en adoptant une numérotation locale unique pour chaque *objet*, cette interface bâtit les bords de contrôle, en fonction de la demande du type de volumes de contrôle.

Ces bords de contrôle sont envoyés, en suivant un PROTOCOLE bien défini³, vers le *standard output* du système. Intervient ensuite l'interface résolveur (*vf*), qui lit à partir du *standard input* l'information envoyée par l'interface géométrique. Elle est transformée à l'entrée de ce sous-système (ellipse bleue dans la figure 2.1) en bords de volumes de contrôle avec toute l'information et supports nécessaires pour l'intégration numérique. Toutes les interfaces ont des mécanismes de manipulation d'erreurs lorsque le flux d'information lue est incorrect ou le système de discrétisation géométrique devient non conservatif.

³ Voir le chapitre 5.

2.2 Modes de fonctionnement du système

Il existe deux modes de fonctionnement du système :

- Un fonctionnement continu: lorsque les interfaces *vfGeo* et *vf* fonctionnent séquentiellement.
- Un fonctionnement différé: lorsqu'elles fonctionnent de façon discontinue.

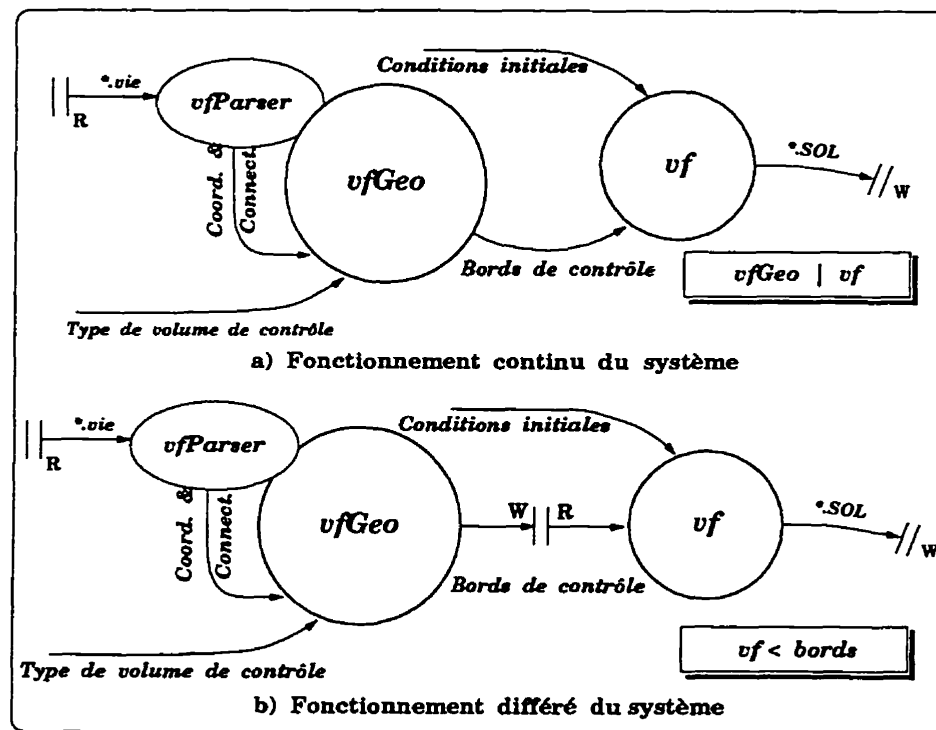


FIGURE 2.2: Deux modes de fonctionnement du système

2.2.1 Fonctionnement continu

Pendant le fonctionnement continu le *standard output* de *vfGeo* est le *standard input* immédiat de *vf* à l'aide d'un tuyau (*pipe*), appliqué dans le *shell* du système d'exploitation; avec cette option il faudra lancer *VF++* (*vfGeo* et *vf*) à chaque fois.

2.2.2 *Fonctionnement différé*

Ce type de fonctionnement devient utile pour des problèmes présentant des grandes discrétisations géométriques, pour lesquelles il est recommandable de ne lancer *vfGeo* qu'une seule fois, en opposition à *vf* qui peut être exécuté plusieurs fois, pour un seul type de problème. Dans ce cas, le *standard output* de *vfGeo*, contenant les bords de contrôle, alloue la mémoire du disque rigide du système d'exploitation.

Quel que soit le mode de fonctionnement de ces modules, en général, *vfGeo* est de courte durée, comparé à *vf*, ce qui libère considérablement de mémoire pour le résolveur (figure 1.5).

Ces techniques sont illustrées par les diagrammes des flux de données à la figure 2.2.

Chapitre 3

L'interface parser, *vfParser*

3.1 Introduction

La fonction du *vfParser* est de bien interpréter un fichier de description du maillage initial. Ce fichier, appelé *Volume input export*, *Vie*, est en format *ASCII* et sa description suit une syntaxe spécifique.

3.2 La syntaxe du fichier *Vie*

La syntaxe d'un fichier *Vie* a trois caractéristiques importantes :

- **La simplicité de sa grammaire:** Si on utilise la notation BNF cette grammaire est décrite dans le panneau montré par le tableau 3.1.
- **Trois langages naturels:** l'anglais, le français et l'espagnol dans la description de *tokens*.
- **Les commentaires:** style *C* ou *C++*. Pouvant accepter aussi commentaires de commentaires.

TABLEAU 3.1: Syntaxe du vfParser

```

Discretizacion [(2|3)] //dimension=2, default
{
  Nodes { < liste_de_coordonnées > } //taille, multiple de la dimension
  [Connect (< forme_géométrique >) [code] //code=0 (default) intérieur
    { < liste_de_connectivités > }]+ /*taille, multiple du nombre de
      sommets de la forme géométrique ... */
  [Transformation //Affectation des coordonnées du domaine...
    {
      [< Escala (Ex[,Ey[,Ez]])> | //Echelle ...
      < Rotation (alpha[,beta])> | //Rotation ...
      < Translation (dx[,dy[,dz]])>]* //Déplacement ...
    }
  ]
}

```

Avec:

```

< forme_géométrique > := < Ligne > | < Triangle > |
                        < Quadrilatere > | < Polygone, ns > |
                        < Tetraedre > | < Prisme > |
                        < Hexaedre > | < Piramide >

```

D'après cette grammaire, la description du maillage initial est décomposée en un ensemble de coordonnées de points et un ensemble de connectivités. Une description plus vaste de cette grammaire est donnée en l'annexe B.

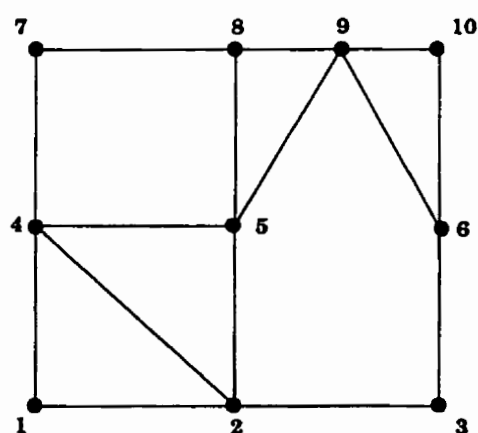
Les classes de cette interface et leurs dépendances sont exposées dans la figure A.2 de la page 192.

3.3 Exemple d'un fichier *Vie* en 2D

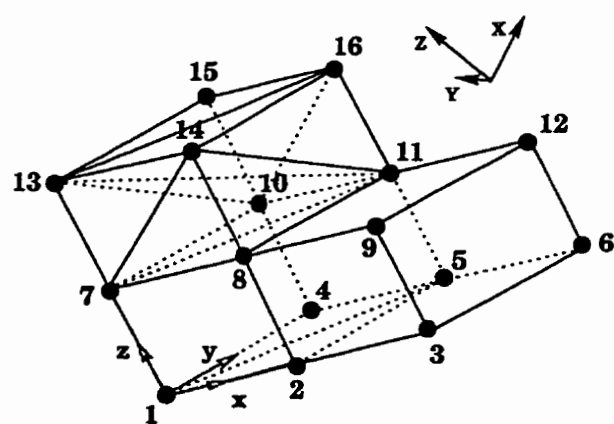
Le contenu d'un fichier en format *Vie* pour le maillage 2D est illustré par la figure 3.1(a) est montré par le code 3.1.

3.4 Exemple d'un fichier *Vie* en 3D

De façon similaire, le contenu du fichier apparenté à la figure 3.1(b) est montré par les codes 3.2 et 3.3.



(a) Maillage Car.vie



(b) Maillage Mixte.vie

FIGURE 3.1: Exemples de maillages initiaux hybrides

Code 3.1 *Exemple du fichier Vie en 2D*

```
//File: Car.vie
Discretisation //dimension=2, par default
{
  Nodes
  {
    0.0 0.0 0.5 0.0 1.0 0.0 0.0 5E-1 0.5 0.5
    1.0 0.5 0.0 1.0 0.5 1.0 0.75 1.0 1.0 1.0
  }
  Connect(Triangle) //Domaine interieur, default code 0
  {
    1 2 4 2 5 4 5 9 8 6 10 9
  }
  Connect(Quadrilatere) //Domaine interieur, default code 0
  {
    4 5 8 7
  }
  Connect(Poligono,5) //Domaine interieur, default code 0
  {
    2 3 6 9 5
  }
  Connect(Line) 5 //Domaine frontiere, code 5
  {
    1 4 4 7
  }
  Connect(Ligne) 6 //Domaine frontiere, code 6
  {
    3 6 6 10
  }
  Connect(Linea) 1 //Domaine frontiere, code 1
  {
    1 2 2 3
  }
  Connect(Line) 3 //Domaine frontiere, code 3
  {
    7 8 8 9 9 10
  }
}
```

Code 3.2 *Exemple du fichier Vie en 3D*

```
//File: Mixte.vie
Discretisation(3)
{
  Noeuds
  {
    0 0 0 1 0 0 0.2e1 0 0
    0 1 0 1 1 0 2 1 0
    0 0 1 1 /* skip this */ 0 1 2 0 1
    0 1 1 1 1 1 2 1 1
    0 0 2 1 0 2 0 1 2
    1 1 2
  }
  Conect(Tetraedre) 0 //Main
  {
    10 16 15 13
    10 11 16 13
    10 11 13 7
    11 14 13 7
    11 8 14 7
    13 16 14 11
  }
  Connect(Triangle) 5
  {
    13 14 16
    15 13 16
  }
  Connect(Cuadrilatero) 6
  {
    6 3 9 12
  }
  Connect(Quadrilateral) 1
  {
    4 5 11 10 5 6 12 11
    2 3 9 8 1 2 8 7
    1 4 10 7 2 3 6 5
    8 9 11 12
  }
  // .. suit
}
```

Code 3.3 *Exemple du fichier Vie en 3D (suite)*

```

// Suite ...
Connect(Triangle) 1
{
    10 11 16 10 16 15
    11 8 14 11 14 16
    7 8 14 7 14 13
    7 10 13 10 13 15
    1 5 4 1 2 5
}
Connect(Prisme) 0 //Main
{
    1 5 4 7 11 10
    1 2 5 7 8 11
}
Connect(Hexaedre) 0 //Main
{
    2 3 6 5 8 9 12 11
}
Transformation
{
    Rotation(60,30) // alpha=60 deg, beta=30 deg
    // Rotation(37,30) // alpha=37 deg, beta=30 deg
    Scale(1.5,1.8) // Ex=1.5,Ey=1.8,Ez=1
    Translation(0.5) // dx=0.5,dy=0,dz=0
}
} //EOF

```

Chapitre 4

Interface géométrique, *vfGeo*

4.1 Introduction

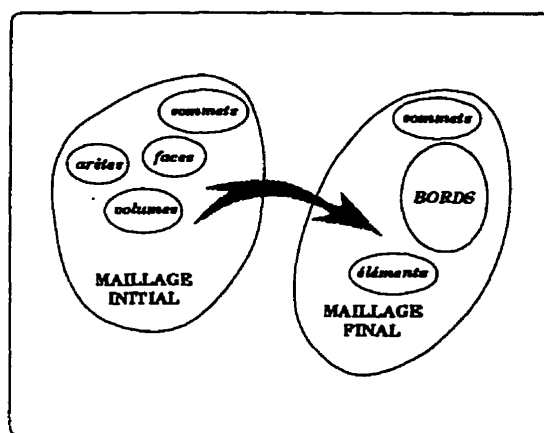


FIGURE 4.1: *Fonction de l'interface géométrique*

L'objectif fonctionnel de *vfGeo* est la génération de bords de contrôle à partir d'un maillage initial, selon la demande du type de volume de contrôle (figure 2.1) associé au stockage ultérieur de la variable. La mission de cette interface est accomplie une fois que cette information géométrique, libre de tout support algébrique, est envoyée vers l'interface résolveur.

La discrétisation initiale lue par *vfParser* est basée sur les éléments de la maille initiale, il s'agit donc, de préparer des outils nécessaires pour le passage de ce

maillage vers un autre final basé sur les bords de volumes de contrôle. Cette transformation, illustrée schématiquement par la figure 4.1, traverse toute une séquence de processus qui sera abordée dans les prochaines sections.

La figure A.3 de la page 193 montre les classes et leurs dépendances utilisées dans cette interface.

4.2 Structure topologique du maillage initial

Le maillage initial, construit par des outils externes à **VF++** et décrit dans un format **Vie** (chapitre 3), consiste en la partition initiale et discrète du domaine géométrique en polygones ou polyèdres.

Le passage d'un type d'information vers un autre exige la décomposition de la discrétisation initiale en entités élémentaires fournissant un support efficace dans ce processus.

4.2.1 Définitions importantes

Pour réussir à élaborer cette transformation, les définitions suivantes sont nécessaires dans l'analyse abstraite de l'information et la création de classes.

Définition 4.1 (*Point*) Entité primitive abstraite et isolée ayant une position dans l'espace.

Définition 4.2 (*Sommet*) Entité topologique délimitante des entités topologiques d'ordres plus élevés¹.

¹ Arête, face et volume.

Définition 4.3 (*Arête*) Entité topologique ayant deux sommets connectés.

Définition 4.4 (*Face*) Entité topologique fermée ayant plus de deux arêtes non-colinéaires et bien connectées. Un polygone.

Définition 4.5 (*Volume*) Entité topologique homéomorphe à une sphère consistant de plus de trois faces non-coplanaires et bien connectés. Un polyèdre.

4.2.2 Univers des entités

La structure topologique abstraite du maillage initial est basée sur l'univers d'ensembles : $(\mathbb{S}, \mathbb{A}, \mathbb{F})$ en 2D ou $(\mathbb{S}, \mathbb{A}, \mathbb{F}, \mathbb{V})$ en 3D.

Avec :

$\mathbb{S}$ — ensemble de s sommets.

$\mathbb{A}$ — ensemble de a arêtes.

$\mathbb{F}$ — ensemble de f faces (polygones fermés).

$\mathbb{V}$ — ensemble de v volumes (polyèdres convexes).

Si on considère que chaque polygone est défini par n sommets, la description de l'univers topologique en 2D, $\forall (n > 2)$, est :

Ensemble de faces $\mathbb{F}$:

$$\mathbb{F} = \{F_1, \dots, F_f\} \quad (4.1)$$

Ensemble de sommets $\mathbb{S}$:

$$\begin{aligned}
 \mathbb{S} &= \{S_1, \dots, S_s\} \\
 &= \underbrace{\bigcup_{F(S_1 S_2 S_3) \in \mathbb{F}} \{S_1, S_2, S_3\}}_{\text{triangles}} \cup \underbrace{\bigcup_{F(S_1 S_2 S_3 S_4) \in \mathbb{F}} \{S_1, S_2, S_3, S_4\}}_{\text{quadrilatères}} \dots \\
 &= \biguplus_{F(S_1 S_2 \dots S_n) \in \mathbb{F}} \{S_1, S_2, \dots, S_n\}
 \end{aligned} \tag{4.2}$$

Ensemble d'arêtes $\mathbb{A}$:

$$\begin{aligned}
 \mathbb{A} &= \{A_1, \dots, A_a\} \\
 &= \underbrace{\bigcup_{F(S_1 S_2 S_3) \in \mathbb{F}} \{\overline{S_1 S_2}, \overline{S_2 S_3}, \overline{S_3 S_1}\}}_{\text{triangles}} \cup \underbrace{\bigcup_{F(S_1 S_2 S_3 S_4) \in \mathbb{F}} \{\overline{S_1 S_2}, \overline{S_2 S_3}, \overline{S_3 S_4}, \overline{S_4 S_1}\}}_{\text{quadrilatères}} \dots \\
 &= \biguplus_{F(S_1 S_2 \dots S_n) \in \mathbb{F}} \{\overline{S_1 S_2}, \overline{S_2 S_3}, \dots, \overline{S_n S_1}\}
 \end{aligned} \tag{4.3}$$

Par analogie, en 3D :

Ensemble de volumes $\mathbb{V}$:

$$\mathbb{V} = \{V_1, \dots, V_v\} \tag{4.4}$$

Ensemble de sommets $\mathbb{S}$:

$$\begin{aligned}
 \mathbb{S} &= \{S_1, \dots, S_s\} \\
 &= \underbrace{\bigcup_{V(S_1 S_2 S_3 S_4) \in \mathbb{V}} \{S_1, S_2, S_3, S_4\}}_{\text{tétraèdres}} \cup \\
 &\quad \underbrace{\bigcup_{V(S_1 S_2 S_3 \dots S_6) \in \mathbb{V}} \{S_1, S_2, S_3, \dots, S_6\}}_{\text{prismes}} \cup \\
 &\quad \underbrace{\bigcup_{V(S_1 S_2 S_3 \dots S_8) \in \mathbb{V}} \{S_1, S_2, S_3, \dots, S_8\}}_{\text{hexaèdres}}
 \end{aligned} \tag{4.5}$$

Ensemble d'arêtes $\mathbf{A}$:

$$\begin{aligned}\mathbf{A} &= \{A_1, \dots, A_a\} \\ &= \underbrace{\bigcup_{V(S_1 S_2 S_3 S_4) \in \mathbf{V}} \{\overline{S_2 S_4}, \overline{S_3 S_4}, \overline{S_1 S_2}, \overline{S_2 S_3}, \overline{S_1 S_3}\}}_{\text{tetraèdres}} \cup \dots\end{aligned}\quad (4.6)$$

Ensemble de faces $\mathbf{F}$:

$$\begin{aligned}\mathbf{F} &= \{F_1, \dots, F_f\} \\ &= \underbrace{\bigcup_{V(S_1 S_2 S_3 S_4) \in \mathbf{V}} \{\overline{S_2 S_4 S_3}, \overline{S_1 S_3 S_4}, \overline{S_1 S_4 S_2}, \overline{S_1 S_2 S_3}\}}_{\text{tetraèdres}} \cup \dots\end{aligned}\quad (4.7)$$

Autrement dit, l'univers $\mathbf{S}$, $\mathbf{A}$, $\mathbf{F}$ et $\mathbf{V}$ est obtenu par décomposition et agrégation des entités élémentaires S , A , F et V à partir des sommets du maillage initial et leurs connectivités.

4.2.3 Relations entre entités

Entre les ensembles $(\mathbf{S}, \mathbf{A}, \mathbf{F}, \mathbf{V})$ il existe des relations définissant la topologie initiale illustrées par le tableau C.1 de la page 196.

4.3 Structure topologique du maillage de volume de contrôle

La structure topologique du maillage final est composée de deux entités: les bords de contrôle, et subséquemment les volumes de contrôle, bâtis à partir du maillage initial.

Cette section traite sur les définitions et construction de ces entités.

4.3.1 Définitions préliminaires

Utilisons deux définitions préliminaires additionnelles dans la description de tout ce document :

Définition 4.6 (*Bord*) Une arête² (en 2D) ou une face³ (en 3D). Donc, l'ensemble de bords $\mathbb{B} = (\mathbb{A} \vee \mathbb{F}) = \{B_1 \cdots B_b\}$.

Définition 4.7 (*Élément*) Une face (en 2D) ou un volume⁴ (en 3D). Donc, l'ensemble d'éléments $\mathbb{E} = (\mathbb{F} \vee \mathbb{V}) = \{E_1 \cdots E_e\}$.

4.3.2 Conditions des éléments et des bords

On peut adapter les conditions de Ciarlet [11], décrites pour les éléments finis triangulaires, à la nouvelle discrétisation en volumes finis pour définir les conditions générales des éléments et bords en 2D ou 3D :

1. La fermeture d'un domaine $\mathcal{D}$ est l'union des *éléments* E sur $\mathbb{E}$:

$$\hat{\mathcal{D}} = \bigcup_{E \in \mathbb{E}} E, \quad \hat{\mathcal{D}} \subseteq \mathcal{D}.$$

2. Chaque *élément* est enfermé par des *bords* et son intérieur est connecté et n'est pas vide :

$$(\check{E} \neq \emptyset).$$

3. Pour chaque paire d'éléments, $(E_i, E_j \in \mathbb{E}, \forall [i \neq j])$, l'expression suivante est valide :

$$\check{E}_i \cap \check{E}_j = \emptyset.$$

² Voir la Définition 4.3.

³ Voir la Définition 4.4.

⁴ Voir la Définition 4.5.

4. Un *bord* B d'un *élément* E_i est soit un sous-ensemble de la frontière $\partial\widehat{D}$, soit le *bord* d'un autre *élément* E_j , ($E_i, E_j \in \mathbb{E}$, $\forall [i \neq j]$).

4.3.3 Les volumes de contrôle

Les entités *sommets*, *arêtes*, *faces* ou *volumes*, définies et obtenues du processus de décomposition (sous-section 4.2.2), sont les supports de référence dans la construction des mailles de volumes de contrôle.

Le *type de volume de contrôle* constitue une entrée dans le diagramme des flux de données de la figure 2.1.

Ces volumes de contrôle peuvent être classifiés en fonction du type de la *Résidence* ultérieur, $\mathfrak{R}$, de l'inconnue⁵. En général cette classification, rapportée au type de stockage et à la dimension spatiale du problème, est résumée et indexée dans le tableau 4.1.

TABEAU 4.1: *Types de volumes de contrôle*

SCHÉMA	INDEX	DESCRIPTION
VCCV	0	Schéma centré sur les <i>sommets</i>
ECCV	1	Schéma centré sur le centre des <i>arêtes</i>
FCCV ⁶	2	Schéma centré sur le centroïde des <i>faces</i>
CCCV	3	Schéma centré sur le centroïde des <i>volumes</i>

Alors, le concept de volume de contrôle devient plus général :

Définition 4.8 (*volumes de contrôle*) Polygones fermés ou polyèdres convexes obtenus de la transformation selon le tableau 4.1. Un nouvel élément⁷.

⁵ Voir la Définition 5.1.

⁶ Devient CCCV en 2D.

⁷ Voir la Définition 4.7.

4.3.4 *Les bords de contrôle*

Dans l'étape du passage de la topologie *initiale* vers la *finale*, le concept de *bord* donné par la Définition 4.6 est enrichi pour une meilleure représentation abstraite d'un bord de contrôle par la Définition suivante :

Définition 4.9 (*Bords de contrôle*) Nouveaux bords possédant les propriétés suivantes :

1. ils divisent discrètement tout le domaine géométrique en volumes de contrôle,
2. ils ont des coordonnées et connectivités,
3. ils ont une orientation dans l'espace,
4. ils pointent sur un volume de contrôle gauche (**L**) et un volume de contrôle droit (**R**), pour les bords internes,
5. ils pointent sur un volume de contrôle, pour les bords-frontières.

La figure 4.2 illustre ces caractéristiques.

La construction de bords dans *vfGeo*, selon le type de volumes de contrôle du tableau 4.1, est appuyée sur des Règles générales valides autant pour les mailles polygonales que polyédriques, ayant pour but la conservation géométrique de la nouvelle discrétisation :

Règle générale 4.1 (*Support de construction des bords*) Tous les bords sont construits en ayant comme support essentiel le centroïde⁸ des éléments⁹.

Règle générale 4.2 (*Nombre de bords internes d'un élément*) Le nombre de bords internes pouvant être construits à l'intérieur d'un élément est égal à une fois le nombre d'arêtes de cet élément. Exception : en 3D ECCV celui-ci est le double.

Règle générale 4.3 (*Numérotation des bords internes d'un élément*) La numérotation des bords internes d'un élément suit la numérotation locale des ses arêtes. Exceptions : en 2D ECCV suit la numérotation locale des sommets, en 3D CCCV suit celle des faces, en 3D ECCV elle n'a pas un ordre établi.

Règle générale 4.4 (*Numérotation des sommets d'un bord*) La numérotation des sommets d'un bord suit l'ordre de son orientation¹⁰. Particulièrement, celle-ci pour les sommets des bords en 3D suit la numérotation d'une face quelconque selon la figure 4.5 par rapport au volume de contrôle gauche.

Règle générale 4.5 (*Nombre de bords dans le domaine*) Le nombre de bords b dans tout le domaine est calculé par une adaptation de l'équation de Descartes :

⁸ Dans *vfGeo*, le centroïde $\vec{O}$ d'un *élément* est déterminé par l'équation :

$$\vec{O} = \frac{\sum_{i=1}^n \vec{S}_i}{n} \quad (4.8)$$

Où n est le nombre de sommets de chaque *élément* et $\vec{S}$ le vecteur position spatiale de chaque sommet.

⁹ Approche plus générique et «conforme» parmi les autres méthodes conçues. Ainsi, si ces *éléments* sont des triangles (des tétraèdres en 3D), les VCCVs et ECCVs (plus FCCV en 3D) peuvent aussi être basés sur le :

- *incentrum*, le centre de la circonférence (sphère) inscrite (intersection des bissectrices en 2D),
- *circuncentrum*, le centre de la circonférence (sphère) circonscrite,
- *orthocentrum*, l'intersection des hauteurs des triangles.

Le deux derniers types de volumes de contrôle causeraient des problèmes inattendus pour des *éléments* élargis et obtus.

¹⁰ Voir les Règles générales 4.6 et 4.7.

$$b = \begin{cases} s + f - 1 & \text{en 2D} \\ a - s + 2 & \text{en 3D} \end{cases} \quad (4.9)$$

où s , a et f sont les nombres de nouveaux sommets, arêtes et faces, respectivement, obtenus de la transformation, selon un type de volume de contrôle du tableau 4.1.

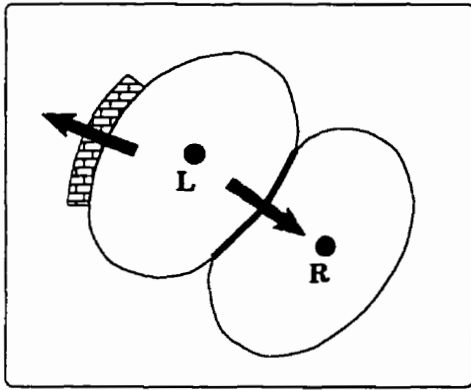


FIGURE 4.2: *Propriétés des bords de contrôle*

L'intégration générique de l'équation (1.5) et le processus de distribution, pour un type de volume de contrôle quelconque, suggèrent l'unification de l'information relative aux bords afin de vérifier le principe de conservativité (sous-section 1.2.1).

Pour réussir dans cette démarche, on établit deux Règles générales suivantes valides en 2D et 3D, reflétées graphiquement par la figure 4.2.

Règle générale 4.6 (*Orientation des bords intérieurs*) L'orientation des normales de tous les bords de contrôle intérieurs est toujours extérieure par rapport au volume de contrôle gauche.

Règle générale 4.7 (*Orientation des bords-frontières*) L'orientation de normales de tous les bords-frontières est toujours extérieure par rapport au domaine intérieur.

Dans la sous-section 4.4.6 on utilisera ces Règles générales afin de construire les bords de contrôle.

4.3.5 Relations entre entités

Un sous-ensemble de relations entre entités définissant les propriétés de la topologie finale, ayant comme supports les ensembles $\mathbb{B}$, $\mathbb{E}$ et même les nouveaux $\mathbb{S}$, peut être représenté abstraitement par le graphe de la figure 4.3 ou par le tableau C.2 de la page 197.

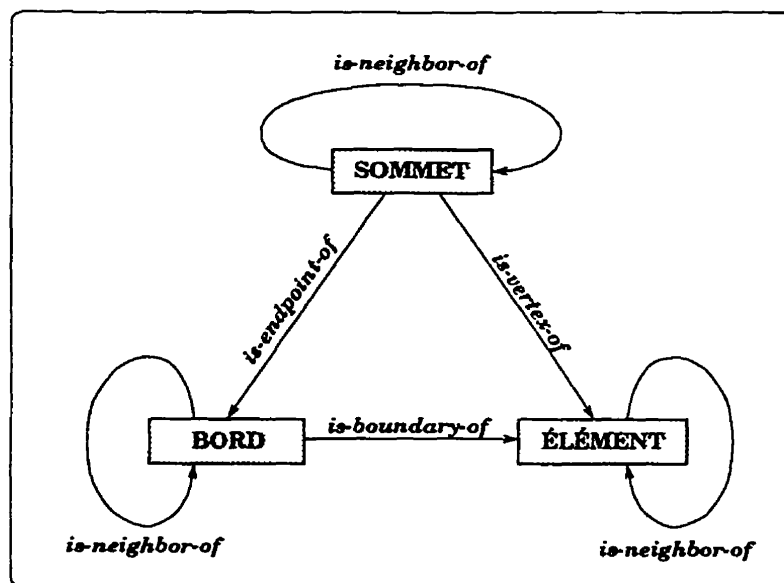


FIGURE 4.3: Relations principales appliquées à la discrétisation finale

Dans ce cadre des relations il faut bien remarquer deux propriétés relationnelles telles que :

“*is – boundary – between*” – sur le sous-ensemble $\mathbb{B} \times \mathbb{E} \times \mathbb{E}$.

“*is – boundary – between*” – sur le sous-ensemble $\mathbb{B} \times \mathbb{E} \times \emptyset$.

qui interprètent la Définition 4.9 et, au même temps, les conditions 3 et 4 de la sous-section 4.3.2.

4.3.6 *Concepts amorphes*

Il est à noter que les concepts d'*éléments* (Définitions 4.7 et 4.8) et des *bords* (Définitions 4.6 et 4.9) sont libérés temporairement de leurs propriétés algébriques, puisqu'ils ne représentent que l'aspect géométrique de la topologie. Ces approches, ayant un concept discernable, seront enrichies par la nature de l'interface résolveur (chapitre 6) où prendra place implicitement le concept final d'élément et de bord de contrôle avec leurs propriétés topologiques au profit de l'analyse numérique de la méthode des volumes finis.

4.4 Les tâches de *vfGeo*

Cette interface, étant le préprocesseur géométrique du système, est chargée de poursuivre les opérations suivantes dans un ordre consécutif :

- L'interprétation du maillage initial. Accès au *vfParser*.
- La construction des listes de sommets (**S**), arêtes (**A**), faces (**F**) et volumes (**V**).
- La construction et la reconnaissance du voisinage.
- La numérotation locale et l'unification de l'information.
- Le stockage des données et tâches de l'interface.
- La création des bords de contrôle selon le tableau 4.1.

4.4.1 *Interprétation du maillage initial*

Ce processus se réalise *at run time* et sert à interpréter le maillage initial. Cette tâche se fait à partir de *vfGeo* vers *vfParser* en passant comme argument

le nom du fichier de la discrétisation initiale *Vie* (voir le chapitre 3), tel qu'indiqué par le code 4.1.

Code 4.1 *Accès au vfParser*

```
...
FV_DISCRETIZATION scene;
FV_PARSER parser(discretizationFile);
parser.parserPrb(scene);      // parses the Vie file

DIM=scene.dim;  // Retrives the dimension and ...
...
```

4.4.2 Construction des entités initiales

La construction d'objets de la topologie initiale fait partie de la décomposition selon la section 4.2, accomplie de façon séquentielle avec le processus précédent, et a pour but la construction de l'univers des entités $\mathbf{S}$, $\mathbf{A}$, $\mathbf{F}$ et $\mathbf{V}$. Considérons un point p à l'intérieur d'une entité sommet S , arête A , face F ou volume V , tel que :

$$p \in \{S, A, F, V\}, \quad \forall [(S \in \mathbf{S}) \vee (A \in \mathbf{A}) \vee (F \in \mathbf{F}) \vee (V \in \mathbf{V})]$$

Si on choisit une position spatiale de p tel que le centre d'une entité, comme étant un endroit d'intérêt topologique¹¹, alors on peut conclure que chaque entité topologique hérite ses propriétés d'une classe de base plus abstraite qu'on appelle *VF_POINT*. Donc, chaque entité S , A , F ou V peut être créée par des classes comme *VF_SOMMET*, *VF_ARETE*, *VF_FACE* et *VOLUME*, dérivées de *VF_POINT* telles qu'illustrées par l'arbre de dérivation¹² de la figure 4.4 (ou la figure A.3 de la page 193).

¹¹ Dans le cas des polygones et polyèdres cet endroit peut être le centroïde, équation (4.8).

¹² Pour une information plus vaste sur cette structure de données regarder [47, 49].

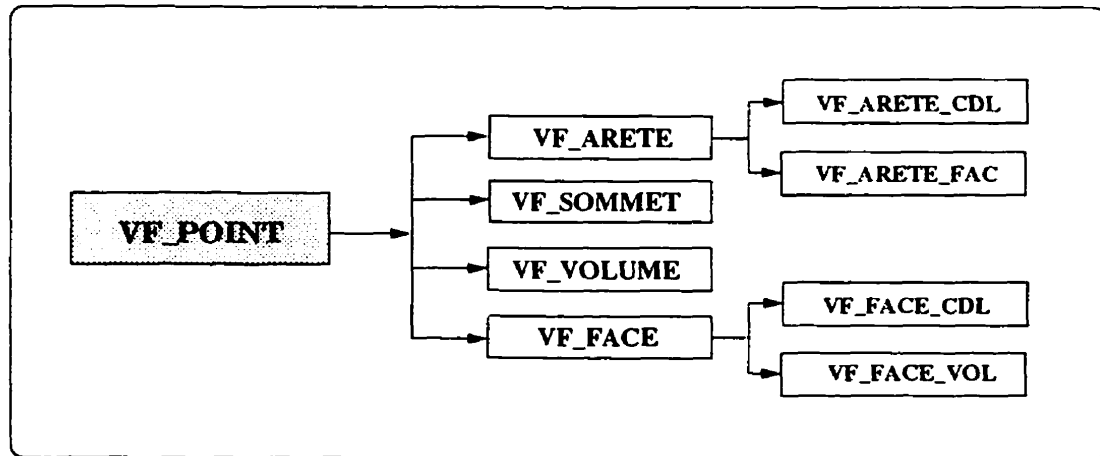


FIGURE 4.4: *Arbre de dérivation des entités topologiques*

La description des classes de cette structure est résumée de la manière suivante :

- **class VF_POINT.** Classe abstraite interprétant un point, Définition 4.1. Chaque objet de cette classe a comme *attributs* les coordonnées de $\{S, A, F, V\}$ et un *index* comme on le constate dans le code 4.2 où la fonction virtuelle *get_type* retourne le type de volume de contrôle associé à une classe dérivée selon les indexations du tableau 4.1.
- **class VF_SOMMET.** Classe dérivant de *VF_POINT*, interprétant un sommet, Définition 4.2, et servant à construire un sommet *S*. L'*attribut* principal de cette classe est une liste de volumes ou faces partageant le même sommet¹³.
- **class VF_ARETE.** *Idem* pour une arête, Définition 4.3. Les *attributs* de cette classe sont les deux sommets qui la définissent et la constellation de faces partageant l'arête créées par sa classe dérivée *VF_ARETE_FAC*.

¹³ Relation "*is-neighbor-of*" avec $F \times S$ et $V \times S$, voir le tableau C.1 de la page 196.

- **class VF_FACE.** *Idem* pour un polygone fermé, Définition 4.4. Cette classe a comme *attributs* une liste de sommets¹⁴ et d'arêtes¹⁵.
- **class VF_VOLUME.** *Idem* pour un polyèdre convexe, Définition 4.5. Cette classe contient la liste de sommets¹⁶, d'arêtes¹⁷ et de faces¹⁸ d'un volume.
- **class VF_ARETE_FAC.** Classe dérivant de VF_ARETE servant de *buffer* dans les processus d'unification des arêtes, de la numérotation locale et de la reconnaissance des faces voisines. Elle a comme *attribut* une liste des faces partageant l'arête¹⁹.
- **class VF_FACE_VOL.** Classe dérivant de VF_VOLUME servant de *buffer* dans les processus d'unification des faces et de la numérotation locale. Elle a comme *attribut* les deux volumes partageant la face²⁰.
- **class VF_ARETE_CDL.** Classe dérivant de VF_ARETE servant à construire une arête se retrouvant à la frontière en 2D. Cette classe a comme attribut additionnel le *code* de cette frontière, attribué dans le fichier de discrétisation initiale *vie* (voir le tableau 3.1 de la page 24).
- **class VF_FACE_CDL.** *Idem* pour une face-frontière, en 3D.

La classe suprême VF_POINT consiste en un ensemble de propriétés dont l'information abrégée est décrite par le code :

¹⁴ Relation "is-vertex-of" avec $S \times F$ (tableau C.1) ou avec $S \times E$ (tableau C.2).

¹⁵ Relation "is-edge-of" avec $A \times F$ (tableau C.1) ou "is-boundary-of" avec $B \times E$ (tableau C.2).

¹⁶ Relation "is-vertex-of" avec $S \times V$ (tableau C.1) ou avec $S \times E$ (tableau C.2).

¹⁷ Relation "is-edge-of" avec $A \times V$ (tableau C.1).

¹⁸ Relation "is-face-of" avec $F \times V$ (tableau C.1) ou "is-boundary-of" avec $B \times E$ (tableau C.2).

¹⁹ Relation "is-neighbor-of" avec $F \times A$, (tableau C.1) en 2D et 3D. Relations "is-boundary-of" dans $B \times E$ ou "is-boundary-between" sur $B \times E \times E$ pour les bords intérieurs en 2D et "is-boundary-between" sur $B \times E \times \emptyset$ pour les bords-frontières en 2D (tableau C.2).

²⁰ *Idem* pour les faces vs volumes.

Code 4.2 *La classe abstraite VF_POINT*

```

class VF_POINT
{
    int num;           //index
    VF_VECTEUR coord;  //coordonnees
public:
    VF_POINT(REAL x1,REAL x2=0,REAL x3=0):coord(x1,x2,x3) {};
    VF_POINT(const VF_VECTEUR& v):coord(v) {};};
    VF_POINT(const VF_POINT& pt):coord(pt.coord) {};}
    VF_POINT(class VF_MESH& m,int* cnc,int nbndsom);
    virtual VF_Type_cv get_type(void) const=0;
    ...
};

```

Les ensembles des sommets (**S**), arêtes (**A**), faces (**F**) et volumes (**V**) construits avec ces classes sont stockés dynamiquement dans une classe *VF_MESH* (voir le code 4.3 de la page 47).

4.4.3 Construction du voisinage

Les relations décrites dans la sous-section 4.2.3 servent, entre autres à la construction et définition du voisinage d'une entité topologique, en suivant les relations principales du tableau C.1.

TABLEAU 4.2: *Pointeurs au voisinage*

↙	sommet	arête	face	volume
sommet	0	1	1	1
arête	0	0	1	1
face	1	1	1	1
volume	1	0	1	1

Dans la version actuelle de *VF++*, chaque objet *sommet*, *arête*, *face* ou *volume* pointe sur un sous-ensemble de ces objets selon le tableau 4.2, où le '1' dénote qu'il est activé et '0' le contraire, dans la direction indiquée par la flèche.

4.4.4 Numérotation locale des éléments initiaux

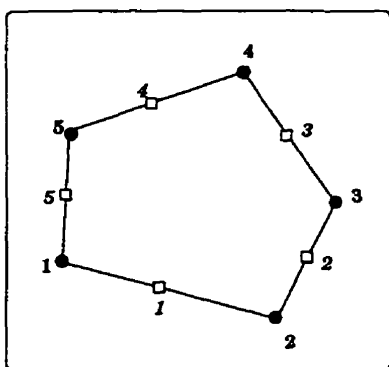


FIGURE 4.5: *Numérotation locale d'une face*

La numérotation locale des entités topologiques initiales joue un rôle important dans l'unification de l'information. En adoptant une seule *règle* de numérotation pour les éléments, *vfGeo* est capable de construire les bords de contrôle et de disposer de l'information sur le voisinage des entités topologiques à l'aide du tableau 4.2.

En 2D et 3D ce processus suit les conventions exposées ci-dessous.

Numérotation locale en 2D

La numérotation locale adoptée pour les polygones est entièrement générique comme on peut le voir sur la figure 4.5. Ici, arêtes et sommets ont une numérotation dans le sens anti-horaire.

Numérotation locale en 3D

En 3D l'unification et la généralisation de la numérotation locale des polyèdres n'est pas évidente comme en 2D dûe au son haut degré de liberté dans l'espace. Étant donné cette restriction le *vfParser* et *vfGeo* n'acceptent que trois types de formes

géométriques 3D : des tétraèdres, des prismes et des hexaèdres ou un mélange des ces derniers.

La numérotation des sommets pour ces trois formes géométriques est montrée en détail sur la figure 4.6. La numérotation des arêtes et faces pour ces formes géométriques est exposée par les tableaux D.1,...,D.3 de la page 199.

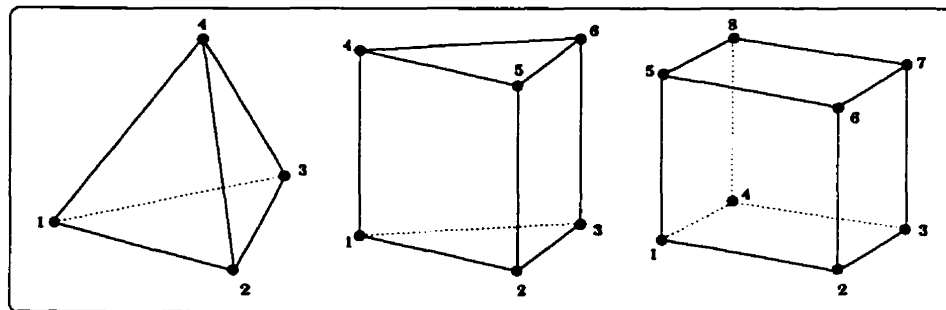


FIGURE 4.6: Numérotation locale des sommets des formes géométriques en 3D

4.4.5 Stockage des données et gestion de l'interface

Les objets, créés par les classes `VF_SOMMET`, `VF_ARETE`, `VF_FACE` et `VF_VOLUME`, sont collectés dynamiquement par la classe `VF_MESH` (voir le code 4.3). Les listes de ces objets et les gestions de *vfGeo* correspondent, respectivement, à l'ensemble d'*attributs* et *comportements* de ladite classe selon la description.

Les *attributs* :

- **sommets**. Correspond à l'ensemble des sommets du domaine, **S**.
- **aretes**. *Idem* pour les arêtes, **A**.
- **faces**. *Idem* pour les faces, **F**.
- **volumes**. *Idem* pour les volumes, **V**.

Code 4.3 *La classe VF_MESH*

```

class VF_MESH
{
    _liste<class VF_SOMMET*> sommets;
    _liste<class VF_FACE*> faces;
    _liste<class VF_ARETE*> aretes;
    _liste<class VF_VOLUME*> volumes;
    ...
public:
    VF_MESH(const char* discr);
    ...
    // THE MAIN METHODS
    void sendBords(int protocol, VF_Type_cv tm, ostream& os);
    void write_vu(const char* fileSol);
    void grid(const VF_Type_cv tcv);
    void skin(const VF_Type_cv tcv);
    ...
};

```

Et les *comportements* plus importants :

- Le constructeur *VF_MESH(const char* discr)* chargé d'effectuer toutes les tâches précédentes incluant le stockage, (sous-sections 4.4.1,...,4.4.5), à partir du nom du fichier *Vie* de la discrétisation initiale (*voir* le chapitre 3) passé comme argument. Par conséquent, ce constructeur a une fonction intégrale dans cette interface.
- La méthode *sendData(int, VF_Type_cv, ostream&)* est la fonction d'envoi des bords de contrôle à l'interface résolveur (*voir* la figure 2.1), en suivant un PROTOCOLE²¹ comme premier argument et le type de volume de contrôle selon le tableau 4.1 comme deuxième argument. Cette information est adressée à une sortie²² comme troisième argument.

²¹ PROTOCOLE 1 ou 2, *voir* le chapitre 5.

²² Le *standard output* par omission.

- La méthode *write_vu(const char* fileSol)* est la fonction d'écriture du fichier pour la visualisation duale²³ d'une solution (comme argument) avec le programme *vu* [51].
- La méthode *grid(const VF_Type_cv tcv)* est la fonction d'écriture du fichier pour la visualisation²⁴ du maillage final, selon le type de volume de contrôle du tableau 4.1 passé comme argument, avec le programme *vu*.
- La méthode *skin(const VF_Type_cv tcv)*, *idem* pour la visualisation de l'enveloppe du maillage final en 3D.

4.4.6 Construction des bords de contrôle

À partir des Règles générales 4.1,...,4.7, *vfGeo* construit les bords de contrôle selon le type de forme géométrique du maillage initial.

La disposition CCCV en 2D ou 3D correspond aux arêtes et faces, respectivement. Pour les autres types de volumes de contrôle on suit des conventions *«con-formes»* :

En 2D, un bord de contrôle type VCCV est une droite partant du centroïde au centre de l'objet arête. Pour le type ECCV le bord est une droite du même centroïde vers le centre de l'objet sommet. Cette construction suit les séquences rapportées par les tableaux E.1 et E.2. La figure 4.7 illustre les volumes de contrôle en 2D obtenus à partir des tableaux E.1 et E.2 et montre l'application de la Définition 4.9 et des Règles générales exposées.

²³ Basée sur la description de [co[co]]volumes de contrôle, créés à partir du maillage initial.

²⁴ Basée sur la description des bords de contrôle, créés à partir du maillage initial.

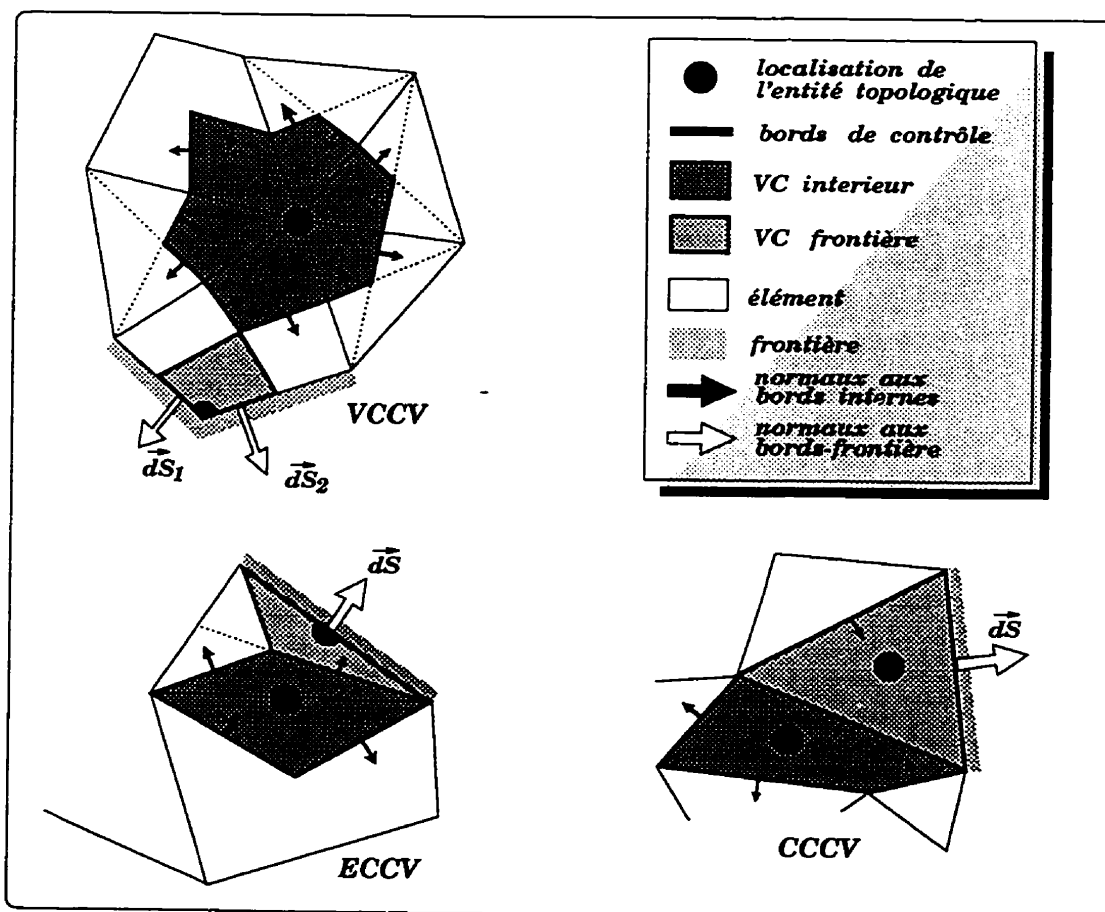


FIGURE 4.7: VCCVs, ECCVs et CCCVs en 2D

En 3D, le même processus est conditionné par le type de polyèdre. Pour réussir dans cette tâche, on emploie des graphes de décision tels qu'illustrés schématiquement par la figure 4.8 en suivant les séquences des tableaux E.3 et E.4 de la page 202. Dans ce parcours les pointeurs du tableau 4.2 deviennent très utiles. Un exemple des bords de contrôle, construit selon cette logique pour un élément tétraédrique, est montré sur la figure 4.9 (les sphères foncées représentent les entités topologiques visibles).

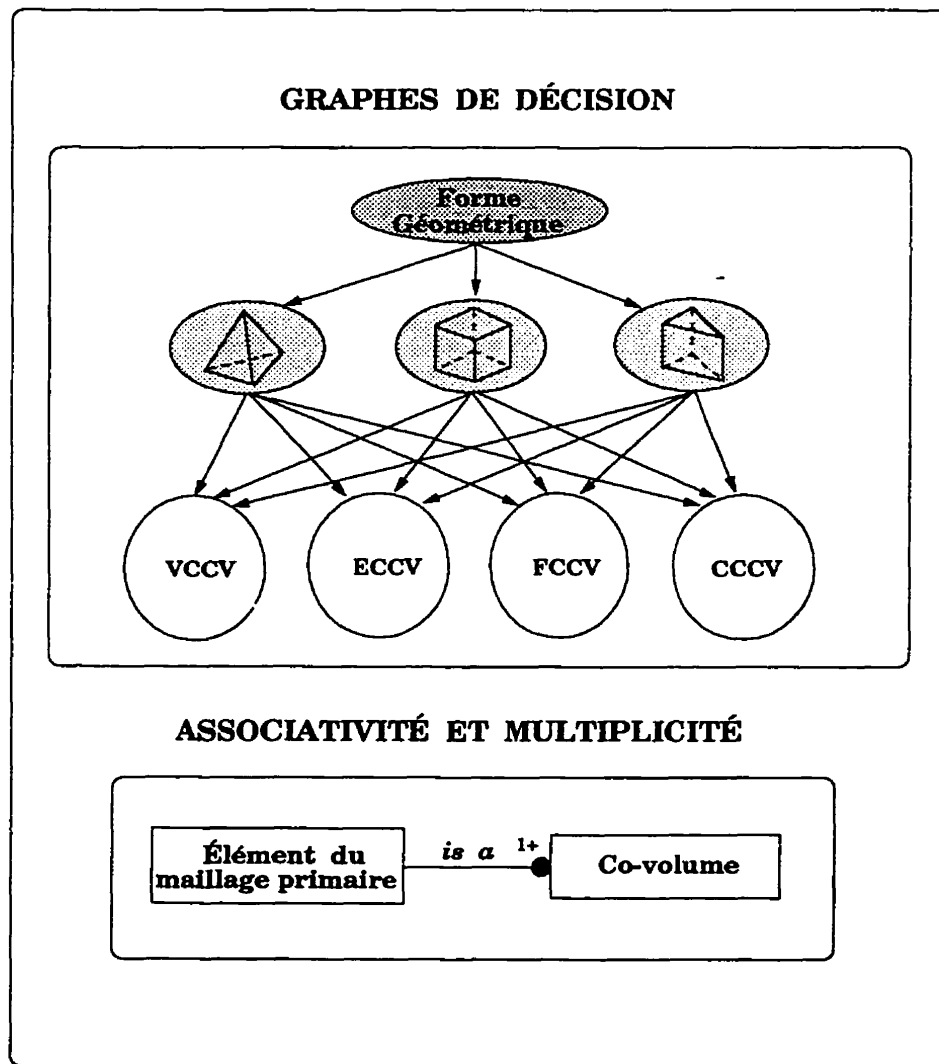


FIGURE 4.8: Construction des volumes de contrôle en 3D

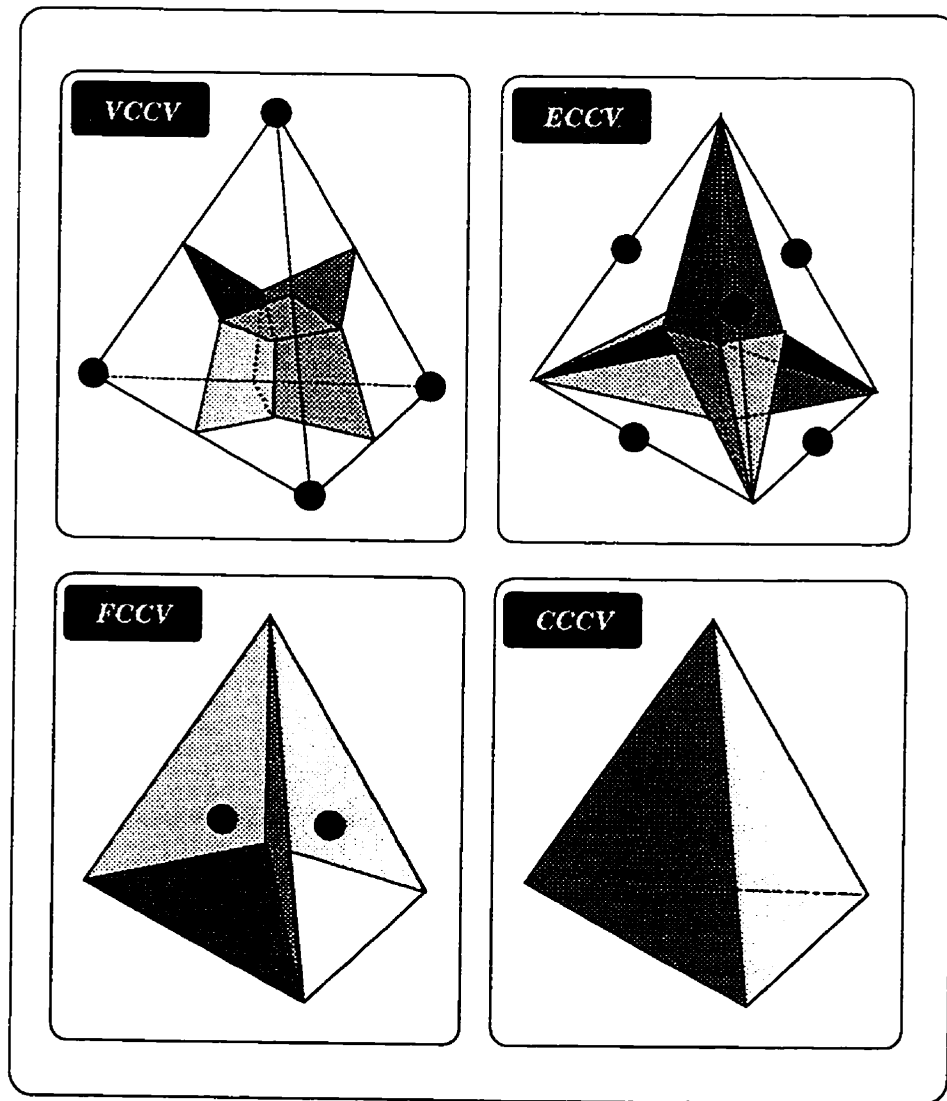


FIGURE 4.9: *Bords de contrôle en 3D (élément tétraédrique)*

4.5 Correspondance des *codes* aux frontières

En général, dans le contexte d'une application spécifique, il y a une correspondance un à un entre les *codes* aux frontières, attribués dans le fichier de description du maillage initial *Vie*, chapitre 3, et l'imposition de conditions aux frontières, choisie par l'utilisateur. Les *codes* définis par omission sont décrits dans le tableau 4.3.

TABLEAU 4.3: *Codes aux frontières établis par omission*

	PAROIS	ENTRÉE	SORTIE
CODES	1-3	5	6

4.6 Validation de la transformation

Pour un type de volume de contrôle donné du tableau 4.1, les éléments et bords doivent être validés. C'est-à-dire, $\forall E \in \{VCCV, ECCV, FCCV, CCCV\}$:

1. **La fermeture du domaine.** Le nouveau domaine discret obtenu, $\mathcal{D}_E$, doit être fermé avec les *codes* qui définissent le type de frontières (section précédente).
2. **La règle des volumes.** L'aire (en 2D) ou le volume (en 3D) du nouveau et de l'ancien domaine doit rester le même : $Volume(\mathcal{D}_E^{new}) = Volume(\mathcal{D}^{old})$.
3. **La règle de la cardinalité.** La cardinalité de l'ensemble des nouveaux éléments E de la discrétisation finale doit rester la même que celle de la discrétisation initiale correspondante, décrite dans la sous-section 4.2.2 : $e = s \vee a \vee f \vee v$.

Ces propriétés, quoique triviales, représentent une bonne approche dans la mise à jour de cette interface.

Chapitre 5

Stencils et protocoles

5.1 Introduction

Ce chapitre a pour but d'établir un pont théorique (section 5.2) et technique (section 5.3) entre les interfaces géométrique (*vfGeo*) et résolveur (*vf*).

5.2 Définitions

En 2D et 3D nous avons les définitions suivantes :

Définition 5.1 (*Résidence et habitat d'une inconnue*¹) La résidence d'une inconnue, $\mathfrak{R}(\Phi)$, est le centroïde de l'élément² ; son habitat est le même élément.

Définition 5.2 (*Stencil*) Ensemble fini de n éléments voisins $\mathcal{E}$, tel que :

$$\mathcal{E} = \cup_{i=1}^n E_i, \quad \forall \mathcal{E} \subset \mathbb{E},$$

servant de support géométrique et algébrique à un interpolant.

¹ Dans ce chapitre les mots «*variable*» et «*inconnue*» sont synonymes.

² Voir la Définition 4.8.

Définition 5.3 (*Types de stencils*) On distingue deux types de stencils :

1. Stencil projectif, S_p . C'est un stencil ne servant qu'à la projection de variables et de valeurs sur un domaine de validité D .
2. Stencil intégral, S_f . C'est un stencil projectif et permet, simultanément, de faire l'intégration numérique³ et la distribution⁴ des flux⁵.

D'après cette Définition, le degré de liberté des stencils projectifs S_p est plus vaste que celui des stencils intégrals.

Définition 5.4 (*Domaine de validité de l'interpolant*) Le Domaine de validité d'un interpolant $D(\tilde{\Phi})$ est une région spécifique $D \subset \mathcal{D}$ composée de tous les éléments d'un stencil.

Définition 5.5 (*Résidence de l'interpolant d'un stencil intégral*) Si B est un sous-ensemble de bords internes B sur D , ($B = \cup B$ et $B \subset \mathbb{B}$), on appelle Résidence de l'interpolant $\mathfrak{R}(\tilde{\Phi})$ d'un stencil S_f , à une position sur B servant de support dans toutes les tâches de ce stencil une seule fois pour chaque bord B .

TABLEAU 5.1: $\mathfrak{R}(\tilde{\Phi})$ en 2D, pour les stencils intégrals S_f

$2D$		Φ		
		VCCV	ECCV	CCCV
$\tilde{\Phi}$	S	0	1	0
	A	1	0	
	F			0

³ Voir l'équation (1.5) de la page 4.

⁴ Voir la sous-section 1.2.3.

⁵ Voir l'équation (1.3).

Après les abstractions dans l'espace, deux conséquences peuvent être tirées de l'interprétation de la Définition 5.5 pour les stencils intégrals :

1. La coexistence de $\mathfrak{R}(\Phi)$ et $\mathfrak{R}(\tilde{\Phi})$ est impossible.
2. La création d'une famille de stencils $\mathcal{S}_f$ admissibles est possible (tableaux⁶ 5.1 en 2D et 5.2 en 3D) pour tous les types de volume de contrôle.

TABLEAU 5.2: $\mathfrak{R}(\tilde{\Phi})$ en 3D, pour les stencils intégrals $\mathcal{S}_f$

<i>3D</i>		$\tilde{\Phi}$			
		VCCV	ECCV	FCCV	CCCV
$\tilde{\Phi}$	S	0	1	0	0
	A	1	0	1	0
	F	0	1	0	
	V				0

5.2.1 Associations

Les tableaux 5.1 et 5.2, de manière plus qualitative, peuvent être classifiés à l'aide de deux associations, par rapport au binôme -Bords de contrôle et Résidence de l'interpolant- :

Association *un à un* : Le sous-ensemble de bords de contrôle $\mathcal{B}$ égal à un seul bord et partageant un seul interpolant $[\mathfrak{R}(\tilde{\Phi}) \text{ sur } (\mathcal{B} = B)]$.

Association *plusieurs à un* : Le sous-ensemble de bords de contrôle $\mathcal{B}$ égal à un nombre fini de bords de contrôle et partageant un seul interpolant $[\mathfrak{R}(\tilde{\Phi}) \text{ sur } (\mathcal{B} \subset \mathbb{B})]$.

⁶ Avec : 0 - pas évident, 1 - possible. Les stencils figurant avec les encadrés ombrés seront mentionnés dans la sous-section 5.2.2 et la section 5.3.

Graphiquement en 2D l'ensemble des stencils $\mathcal{S}_f$ admissibles est illustré par les figures 5.1 a), b), c) et d) (association plusieurs à un) et e) (association un à un).

En 3D, les figures 5.2(a)(i)(ii) montrent les stencils intégrals $\mathcal{S}_f$ avec les interpolants sur l'arête, 5.2(b)(i)(ii), sur la face et 5.2(c)(i)(ii)(iii), sur le centroïde d'un volume. Dû à la complexité de son esquisse, le stencil intégral avec l'interpolant sur le sommet en 3D n'est pas illustré.

5.2.2 Stencils $\mathcal{S}_f$ par omission

La construction des stencils $\mathcal{S}_p$ ou $\mathcal{S}_f$ est supportée par la numérotation locale des éléments initiaux. Dans sa version actuelle, **VF++** n'accepte que les stencils $\mathcal{S}_f$ figurant dans les tableaux 5.1 et 5.2 par les encadrés foncés et sont appelés, dans cette étude, *stencils intégrals par omission*.

L'avantage de ces stencils est la *simplicité* de leur construction et la *commodité* de leur application.

5.3 Protocoles

Peu importe le choix des associations décrites dans la sous-section 5.2.1, la technique d'envoi et de réception des bords de contrôle de **vfGeo** vers **vf** doit considérer trois aspects :

1. le transfert de bords de contrôle selon la Résidence de l'interpolant⁷ $\mathcal{R}(\tilde{\Phi})$,
2. la technique d'intégration et le schéma numérique à employer,
3. le filtrage de l'information arrivant à l'interface résolveur, **vf**.

⁷ Voir la Définition 5.5.

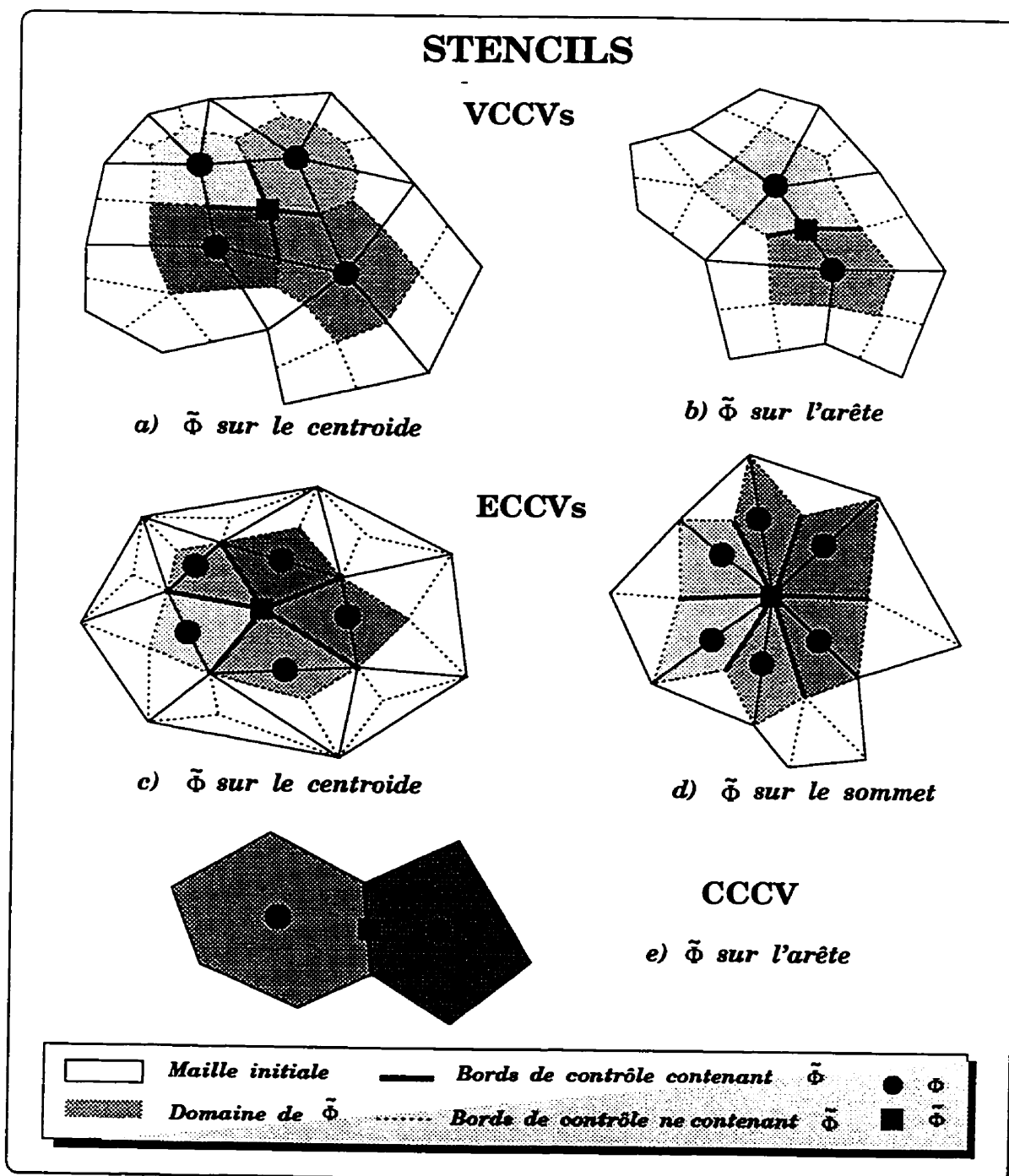
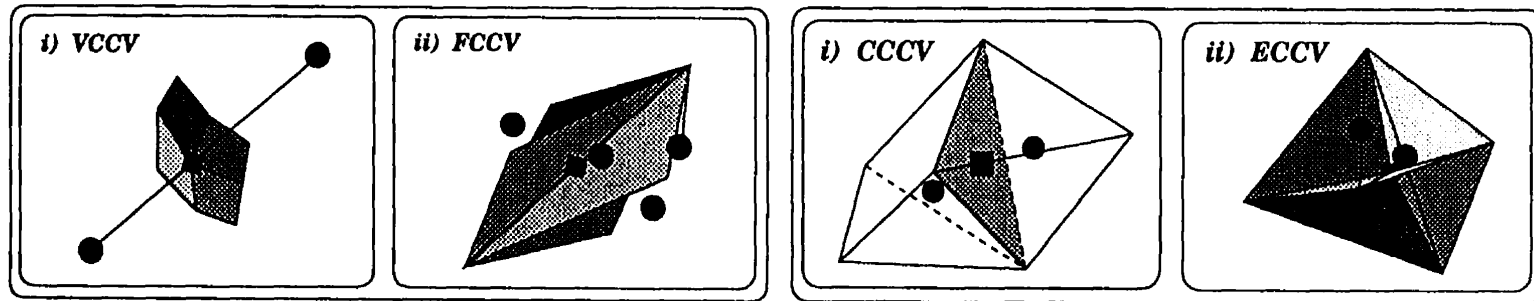
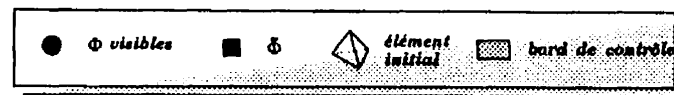
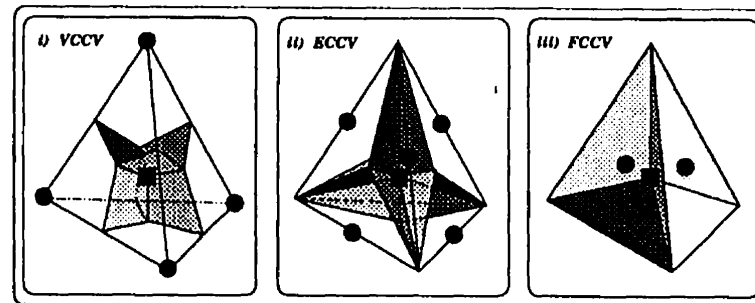


FIGURE 5.1: Stencils intégrals et domaine de validité de $\tilde{\Phi}$ en 2D



(a) Interpolant sur l'arête

(b) Interpolant sur la face



(c) Interpolant sur le centroïde du tétraèdre

FIGURE 5.2: *Esquisses de stencils intégrals en 3D*

Deux PROTOCOLES génériques sont établis afin de gouverner ce processus :

- **PROTOCOLE 1:** Interpolant implicite à chaque bord. Le mécanisme d'envoi et de réception suit la règle: «*un bord pour chaque interpolant*». Association *un à un*⁸.
- **PROTOCOLE 2:** Interpolant explicite situé dans sa résidence. Le mécanisme d'envoi et de réception obéit à la règle: «*plusieurs bords pour chaque interpolant avec un stencil intégral*⁹ S_f ». Association *plusieurs à un*¹⁰.

Observons cela en détail dans les sous-sections suivantes à l'aide de la figure 5.3.

5.3.1 PROTOCOLE 1

Ce type de *protocole* en 2D ou 3D est un *stream* englobant toute l'information nécessaire rapportée à un bord, à savoir¹¹ :

$$\begin{aligned}
 (< Nb_sommets > < Connectivité > < Vol_gauche > < Vol_droit >) | \\
 (< Nb_sommets > < Connectivité > < Volume > < code^{12} >)
 \end{aligned}
 \tag{5.1}$$

Donc, si on considère la figure 5.3 et (5.1), un *stream* pour un bord intérieur et un autre pour un bord-frontière est :

⁸ Voir la sous-section 5.2.1.

⁹ Voir la sous-section 5.2.2.

¹⁰ Voir la sous-section 5.2.1.

¹¹ Ici et dans les exemples les parenthèses et les virgules ne sont que des caractères de séparation et la barre implique la connectivité. Ils ne font pas partie du métalangage. Entre caractères-descripteurs, le caractère de séparation est un espace vide. Ces commentaires sont aussi valides pour le PROTOCOLE 2.

¹² Voir la section 4.5.

- pour VCCV : $\underbrace{2, \overline{a_5 F_{11} s_2 s_{15}}}_{\text{bord intérieur}} \quad \underbrace{2, \overline{a_5 s_2, s_2}}_{\text{bord-frontière}}$
- pour ECCV : $2, \overline{s_2 F_{11} a_3 a_5} \quad 2, \overline{s_0 s_{15}, a_1 5}$
- pour CCCV : $2, \overline{s_0 s_2 F_2 F_{11}} \quad 2, \overline{s_0 s_{15} F_{11} 5}$

La même séquence est répétée pour tous les bords.

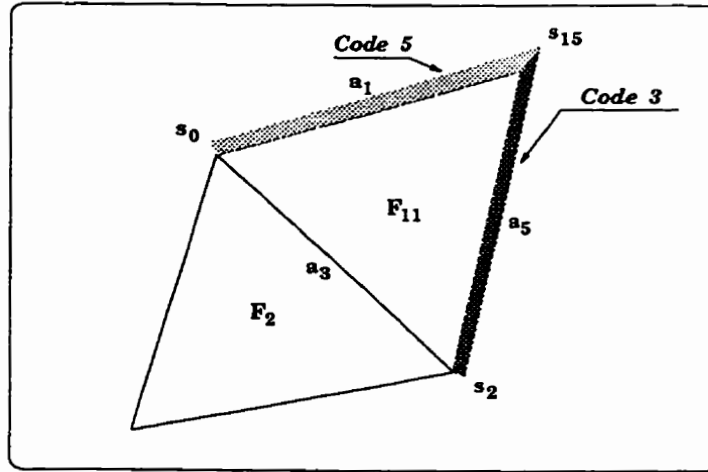


FIGURE 5.3: *Exemple d'utilisation des protocoles en 2D*

5.3.2 PROTOCOLE 2

Ce *protocole* établit l'envoi et la réception d'un ensemble de bords pour chaque interpolant. Le *stream* respectif de ce PROTOCOLE devient plus surchargé puisqu'il porte des bords $\mathcal{B}$ avec un stencil $\mathcal{S}_f$:

$$\begin{aligned}
 & \langle \mathfrak{R}(\tilde{\Phi}) \rangle \langle \mathcal{S}_f \rangle \\
 & \quad (\langle Nb_bords_int \rangle [\langle Connectivité \rangle \langle Vol_gauche \rangle \langle Vol_droit \rangle]^+) \\
 & \quad (\langle Nb_bords_fro \rangle [\langle Connectivité \rangle \langle Volume \rangle \langle code \rangle]^+)
 \end{aligned} \tag{5.2}$$

Donc, le *protocole* (5.2) établit (voir la figure 5.3) :

– pour VCCV :

$$\begin{array}{c}
 \underbrace{F_{11}}_{\mathbb{R}} \underbrace{(3, s_2 s_{15} s_0)}_{s_f} \\
 \underbrace{(3, \overline{a_5 F_{11} s_2 s_{15}}, \overline{a_1 F_{11} s_{15} s_0}, \overline{a_3 F_{11} s_0 s_2})}_{\text{bords intérieurs}} \\
 \underbrace{(4, \overline{s_0 a_1 s_0 5}, \overline{a_1 s_{15} s_{15} 5}, \overline{s_{15} a_5 s_{15} 3}, \overline{a_5 s_2 s_2 3})}_{\text{bords-frontières}}
 \end{array}$$

– pour ECCV :

$$\begin{array}{c}
 F_{11} (3, a_3 a_5 a_1) \\
 (3, \overline{s_2 F_{11} a_3 a_5}, \overline{s_{15} F_{11} a_5 a_1}, \overline{s_0 F_{11} a_1 a_3}) \\
 (2, \overline{s_0 s_{15} a_1 5}, \overline{s_{15} s_2 a_5 3})
 \end{array}$$

– pour CCCV, montrons deux *streams* :

$a_3 (2, F_2 F_{11})$	$a_1 (1, F_{11})$
$(1, \overline{s_0 s_2 F_2 F_{11}})$	(0) pas de bords intérieurs
(0) pas de bords – frontières	$(1, \overline{s_0 s_{15} F_{11} 5})$

Le même métalangage peut être étendu aux autres alternatives données par les tableaux 5.1 et 5.2. Montrons deux *streams* valides pour la figure 5.1(b) :

$a_3 (2, s_0 s_2)$	$a_5 (2, s_2 s_{15})$
$(2, \overline{a_3 F_{11} s_0 s_2}, \overline{a_3 F_2 s_2 s_0})$	$(1, \overline{a_5 F_{11} s_2 s_{15}})$
(0)	$(2, \overline{s_{15} a_5 s_{15} 3}, \overline{a_5 s_2 s_2 3})$

En résumant, les caractéristiques quant au *stream* d'information écoulee sont bien reflétées par la figure 5.4.

En 3D, les PROTOCOLES suivent exactement les mêmes séquences.

PROTOCOLES

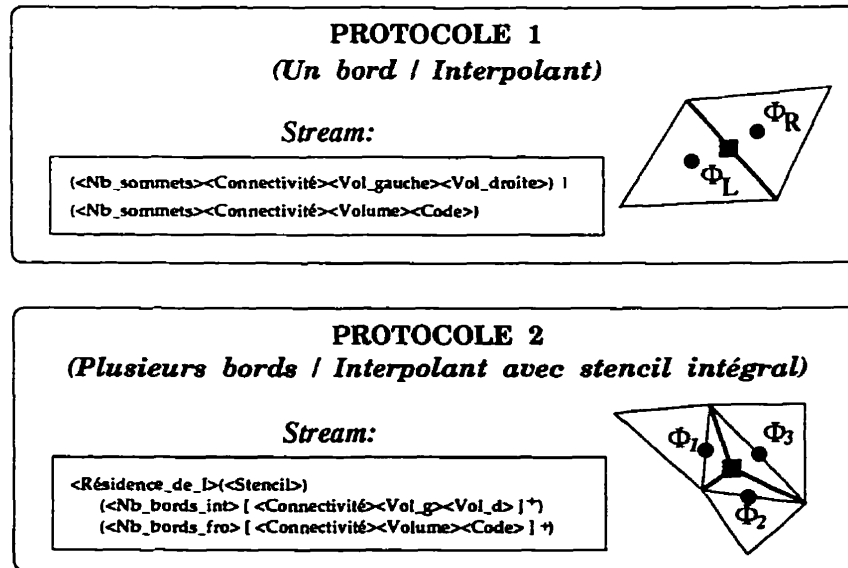


FIGURE 5.4: Les protocoles dans VF++ avec des exemples graphiques en 2D

5.4 Miscellanea

5.4.1 Correspondances entre vfGeo et vf

Les modules *vfGeo* et *vf* fonctionnent ensemble (voir le chapitre 2 et les figures 2.1 et 2.2), donc, à la demande d'un type de volume de contrôle une correspondance un à un est nécessaire entre ce qu'on envoie et ce qu'on reçoit. Le tableau 5.3 illustre ce fonctionnement.

TABLEAU 5.3: Gestion d'envoi/réception des mailles de volume de contrôle

<i>vfGeo</i>	<i>vf</i>
mesh.sendData(protocole,tvc0,cout)	var0.receiveBords(cin)
mesh.sendData(protocole,tvc1,cout)	var1.receiveBords(cin)
...	...

Ici on utilise la méthode d'envoi *sendData* (voir le code 4.3 de la page 47) de la classe *VF_MESH*. La méthode *receiveBords* est traitée dans les annexes G et H.

5.4.2 Les bords de contrôle décalés

L'option décalée n'est habilitée que pour le PROTOCOLE 2 pour $tvc0 \neq tvc1 \neq \dots$ dans le tableau 5.3. Par exemple un maillage décalé quelconque¹³ est obtenu à partir des séquences montrées par le tableau 5.4, en utilisant les indices du tableau 4.1.

TABLEAU 5.4: Exemple d'envoi et de réception pour un schéma décalé

<i>vfGeo</i>	<i>vf</i>
mesh.sendData(2,1,cout)	v.receiveBords(cin)
mesh.sendData(2,2,cout)	p.receiveBords(cin)

5.4.3 Applications

La Définition 5.5 et les tableaux 5.1 et 5.2, nous permettent de concevoir tout un ensemble d'applications selon les PROTOCOLES.

Dans ce contexte, le PROTOCOLE 1 a été utilisé par *VF++* dans le cadre de la résolution des équations d'*Euler* en régime compressible, pour toutes les dispositions de la variable selon le tableau 4.1. Les fondements de cette application sont traités dans la sous-section 6.4.1. Le deuxième PROTOCOLE est appliqué à la résolution de l'équation de convection-diffusion (1.5) dans le cas des équations de *Navier-Stokes* (sous-section 6.4.2).

¹³ ECCV pour les vitesses et CCCV pour la pression.

Chapitre 6

L'interface résolveur, *vf*

6.1 Introduction

Les bords de contrôle provenant de *vfGeo*, guidés selon un type de PROTOCOLE¹, sont exempts des variables et du support mathématique (section 1.4). C'est au rôle de l'interface *vf* de les fournir.

On remarque que cette interface ne reconnaît pas le type de volume de contrôle auquel ces bords appartenaient. Les index du tableau 4.1 deviennent des étiquettes : Tous les volumes de contrôle sont donc des éléments anonymes quant à leur origine.

En utilisant la terminologie du chapitre 4, la *nouvelle topologie* obtenue de la transformation est composée en un univers des volumes de contrôle, ou *éléments*² ($\mathbb{E}$), de bords de contrôle, ou *bords*³ ($\mathbb{B}$) et de *sommets* ($\mathbb{S}$). Ce processus de changement des topologies est schématisé par le tableau 6.1.

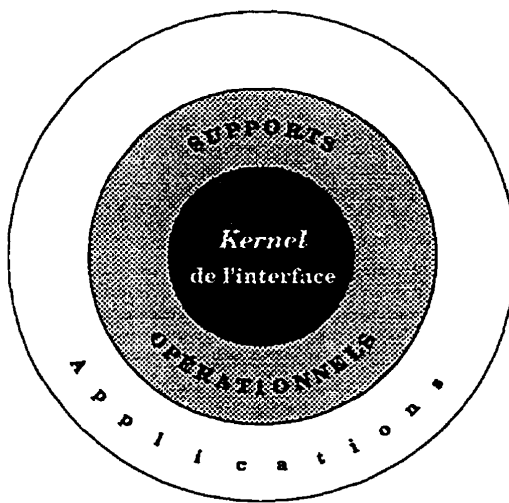
¹ voir le chapitre 5.

² Voir Définition 4.8 de la page 35.

³ Voir Définition 4.9.

TABLEAU 6.1: *Transformation de topologies*

$$\underbrace{\{S, A, F, V\}}_{vfGeo} \Rightarrow \underbrace{\{B, E, [S]\}}_{vf}$$

FIGURE 6.1: *La conception de l'interface résolveur*

L'interface résolveur prétend évoluer vers un environnement enrichi d'outils topologiques et mathématiques avec des structures de données servant à l'application de la méthode des volumes finis et tendant vers sa réutilisation.

Cette tâche difficile exige la création d'un sous-système composé de trois disques hiérarchiques et concentriques du point de vue de l'écoulement et la gestion de l'information (figure 6.1) :

- **le kernel de l'interface**, portant un ensemble de classes de nature stable et générale (information stratégique et de gestion),
- **les supports opérationnels**, portant un ensemble de classes de nature générale dans les processus numériques (information opérationnelle),
- **les applications**, représentées par les implantations de l'utilisateur, (information technique).

Les classes et leurs dérivations utilisées dans la mise en œuvre de cette interface sont exposées à la figure A.4 de la page 193.

6.2 Le *kernel* de l'interface

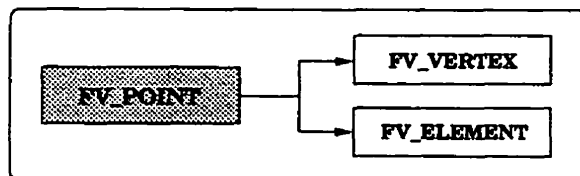
Le *kernel* a comme caractéristique une structure de données essentielle et stable, et les processus en émanant sont divisés en quatre catégories :

- Création de la nouvelle topologie : Bords ($\mathbb{B}$), éléments ($\mathbb{E}$) et sommets ($\mathbb{S}$).
- Création des variables et inconnues abstraites.
- Génération des équations algébriques en implicite.
- Gestion de la résolution des équations en implicite.

6.2.1 *Création de la nouvelle topologie*

Le but de ce processus est la construction des éléments ($\mathbb{E}$), des sommets ($\mathbb{S}$) et des bords ($\mathbb{B}$).

Construction des éléments et des sommets



Les éléments, $\mathbb{E}$, et sommets, $\mathbb{S}$, sont construits avec les classes *FV_ELEMENT* et *FV_VERTEX*, respectivement. Ces deux classes dérivent

FIGURE 6.2: *Arbre de dérivation de la classe FV_POINT*

d'une classe de base *FV_POINT* (figure 6.2) ayant comme *attributs* des co-

ordonnées et un index. Étant donné qu'un élément est le volume de contrôle, il doit

contenir additionnellement un ensemble d'inconnues⁴ et une norme comme *attributs*, tel qu'indiqué dans le code 6.1.

Code 6.1 *La classe FV_ELEMENT*

```
class FV_ELEMENT: public FV_POINT
{
    REAL vol;
    _liste<class INC*> inconnues;
public:
    FV_ELEMENT(REAL x1,REAL x2,REAL x3=0):
        FV_POINT(x1,x2,x3),inconnues(1,8),vol(0) {};
    FV_ELEMENT(const VF_VECTEUR& v):
        FV_POINT(v),inconnues(1,8),vol(0) {};
    class INC* var(INT i) { return inconnues[i]; };
    ...
};
```

Tous les éléments sont stockés dans une liste dynamique associée à une variable.

Construction des bords

La construction des bords, $\mathbb{B}$, est fonction du PROTOCOLE. Un bord B , ($B \in \mathbb{B}$), est bâti à l'aide de la classe abstraite *VF_BORD*. Ce processus se réalise de façon systématique et biunivoque entre les *codes*⁵ établis dans le fichier de discrétisation initiale *Vie*, le type de volume de contrôle du tableau 4.1 et le type de conditions aux frontières à appliquer.

Chaque bord B peut appartenir à l'intérieur ou à la frontière du domaine, $\mathcal{D}$. Ils sont créés par les classes *VF_BORD_INTERIEUR* et *VF_BORD_FRONTIERE*, respectivement, auxquels on doit fournir un ensemble d'*attributs* et *comportements*.

⁴ Voir la sous-section 6.2.2.

⁵ Voir le tableau 4.3.

Par la suite, chaque bord-frontière est classifié en un sous-ensemble de bords selon la nature physique du problème et leur code (section 4.5) :

- **Bords-paroi**: créés par la classe `VF_BORD_FRONTIERE_PAROI` avec les codes par omission 1-3.
- **Bords-entrée**: *Idem*, avec `VF_BORD_FRONTIERE_ENTREE` et le code 5.
- **Bords-sortie**: *Idem*, avec `VF_BORD_FRONTIERE_SORTIE` et le code 6.

Cette classification avec l'attribution des codes aux frontières peut grandir en fonction des demandes de l'utilisateur.

Chaque bord B a les *attributs* :

- *une position*, son centroïde.
- *une normale*, $\vec{n}$ du bord.
- *une norme*, ΔS du bord.
- *deux éléments gauche et droit*, les volumes de contrôle partageant ce bord.

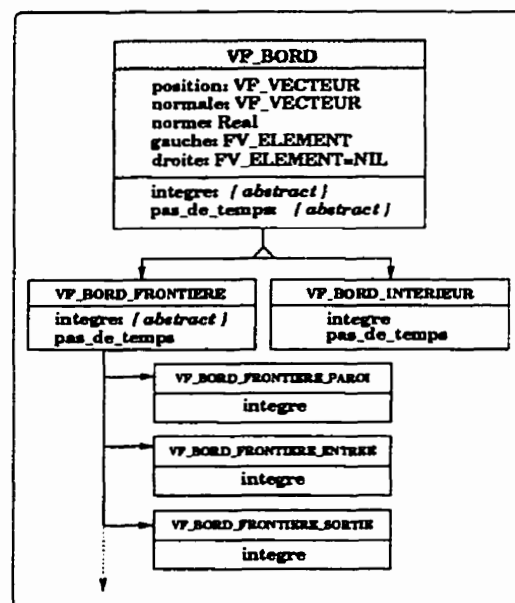


FIGURE 6.3: Arbre de dérivation de la classe abstraite `VF_BORD`

Les *comportements* attachés à un bord sont des fonctions virtuelles liées aux processus qu'ils représentent, comme celui de l'intégration⁶ et de la distribution.

⁶ Dans le cas plus général, ces *comportements* peuvent être associés à un sous-ensemble de bords B , ($B \subset \mathbb{B}$), lorsque, par exemple il faut faire intervenir un stencil intégral $\mathcal{S}_f$ (Définition 5.3).

L'intégration numérique dépend de l'application spécifique⁷ et du type de bord par rapport au domaine de calcul. Par contre, la distribution (sous-section 1.2.3) a un caractère général, comme on le verra dans la sous-section 6.3.2.

L'anatomie principale de la classe abstraite *VF_BORD* est montrée sur la figure 6.3.

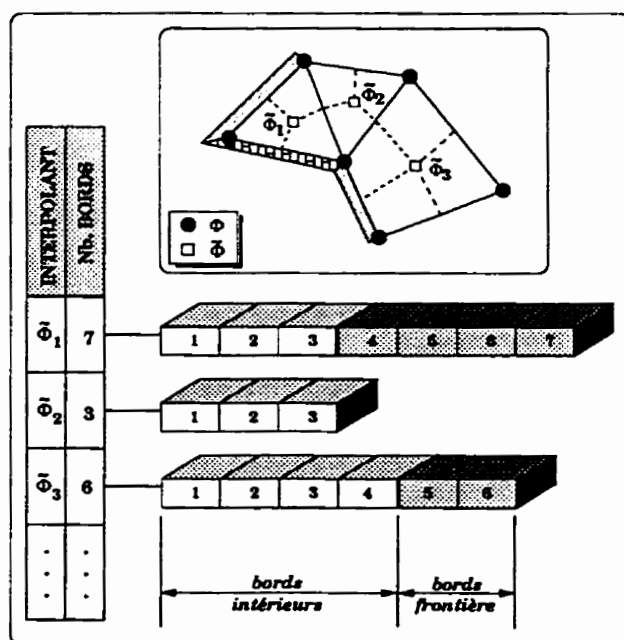


FIGURE 6.4: Exemple de stockage des bords pour le PROTOCOLE 2

L'orientation de chaque bord suit exactement les Règles générales 4.6 et 4.7 de la page 38. Par omission, en vertu de la Règle 4.7, à la frontière du domaine de calcul le volume de contrôle droit est toujours en *NIL*.

Le grand avantage de cette classification des bords est de profiter des propriétés inhérentes à la programmation orientée-objet comme le *polymorphism* et le *dynamic binding* dans le processus d'intégration numérique, en évitant

ainsi des boucles inattendues et coûteuses dans l'implantation.

Les bords de contrôle, $\mathbb{B}$, sont collectés selon le type de PROTOCOLE dans le même ordre établi par celui-ci⁸.

⁷ Voir la sous-section 6.3.1.

⁸ Voir les équations (5.1) et (5.2).

Pour le premier PROTOCOLE ils sont stockés dans une liste vectorielle et dynamique. Dans le cas du PROTOCOLE 2 ils sont collectés vectoriellement par groupes de bords $\mathcal{B}$, ($\mathcal{B} \subset \mathbb{B}$), se trouvant dans le domaine de validité de l'interpolant⁹, $D(\tilde{\Phi})$, car ils font partie intégrante du *stencil intégral*¹⁰ $\mathcal{S}_f$. Ce stockage des bords des domaines D est illustré par la figure 6.4 pour un schéma VCCV en 2D.

Volumes de contrôle décalés

Tout en respectant les correspondances décrites dans la sous-section 5.4.2 l'interface résolveur peut obtenir un système de mailles de volume de contrôle décalés à partir d'un maillage initial, selon la relation :

$$\{ \mathbf{S}, \mathbf{A}, \mathbf{F}, \mathbf{V} \} \Rightarrow \begin{cases} \{ \mathbb{B}, \mathbb{E}, [\mathbf{S}] \}_0 & \text{pour VCCV} \\ \{ \mathbb{B}, \mathbb{E}, [\mathbf{S}] \}_1 & \text{pour ECCV} \\ \{ \mathbb{B}, \mathbb{E}, [\mathbf{S}] \}_2 & \text{pour FCCV} \\ \{ \mathbb{B}, \mathbb{E}, [\mathbf{S}] \}_3 & \text{pour CCCV} \end{cases}$$

Topologie des éléments-frontières

Un élément-frontière peut appartenir à deux catégories de volumes de contrôle, par rapport à la résidence de l'inconnue $\mathfrak{R}(\Phi)$.

La première famille est reliée aux éléments portant une inconnue dans leur centroïde (une disposition CCCV). La deuxième porte une inconnue sur les bords-frontières (toutes les autres dispositions de volumes de contrôle du tableau 4.1). Ces deux dispositions sont illustrées par la figure 6.5(a), avec les normales orientées selon la Règle générale 4.7.

⁹ Voir la Définition 5.4 de la page 54.

¹⁰ Voir la Définition 5.3.

Cette catégorisation et abstraction deviennent importantes lors de l'imposition de conditions aux frontières autant pour les applications liées au premier PROTO-COLE que pour celles du deuxième, puisqu'il s'agit du calcul de flux traversant ladite frontière.

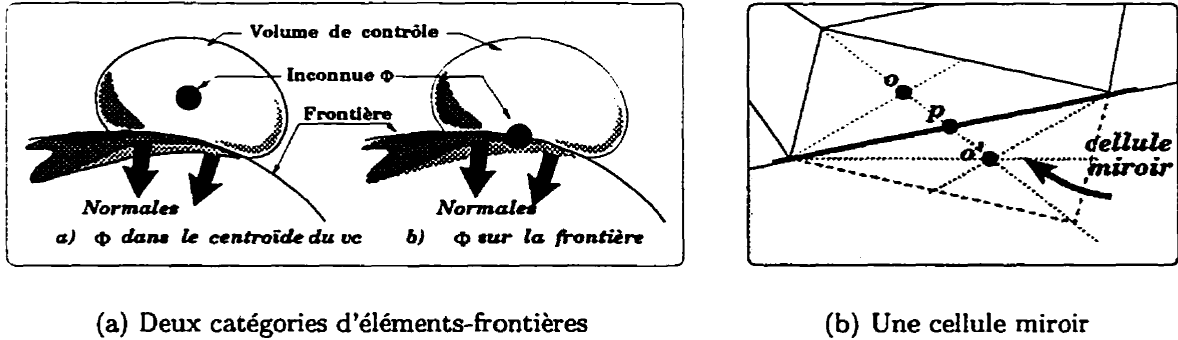


FIGURE 6.5: *Les éléments-frontières*

Dans le cas plus général les deux catégories d'éléments-frontières ont besoin d'un élément-miroir, qui peut contenir deux informations importantes :

1. La valeur-miroir représentative.
2. Les coordonnées-miroir représentatives.

La *valeur-miroir représentative* est toujours la valeur de l'inconnue associée à l'élément-frontière pour les deux catégories d'éléments. Au besoin, les *coordonnées-miroir représentatives* $\vec{o}'$ de la première catégorie d'éléments¹¹ sont déterminées par l'équation vectorielle :

$$\vec{o}' = 2 \cdot \vec{p} - \vec{o}$$

avec $\vec{p}$, le centre des coordonnées du bord-frontière et $\vec{o}$ celui de la résidence de l'inconnue $\mathcal{R}(\Phi)$, tel que montré par la figure 6.5(b).

¹¹ Celles de la deuxième catégorie ne sont pas pertinentes mais elles coïncident avec celles de la résidence de l'inconnue.

6.2.2 *Création de variables et inconnues abstraites*

Une étape importante dans la conception de cette interface est la création de *variables et inconnues* abstraites, comme composantes de son support algébrique.

- Une variable est rapportée au problème, à l'équation à résoudre. Elle est représentée par la classe abstraite *VAR*.
- Une inconnue est associée au volume de contrôle où elle habite¹². Elle est représentée par la classe abstraite *INC*.

À partir de cette approche il existe une *association un à plusieurs* entre une variable et une inconnue. Par ailleurs, la même association est appliquée d'un élément vers une inconnue.

Les *attributs* essentiels de la classe *VAR* sont la liste des volumes de contrôle et leur type (tableau 4.1) selon la demande de l'utilisateur. Les *comportements* comprennent tout un ensemble de fonctions virtuelles permettant la création des inconnues associées, l'accès à ces dernières¹³ ou l'affectation de leurs *attributs*. Ces deux caractéristiques sont reproduites par la structure de données du code 6.2.

Variables et inconnues peuvent être classifiées selon la nature de leur stockage et le support mathématique associé. Cette classification est détaillée dans les lignes qui suivent.

Classification des variables

Pour sa création *vf* classe la variable en trois types :

¹² Voir Définition 5.1.

¹³ Voir la sous-section 6.3.5.

Code 6.2 *La classe VAR*

```

class VAR
{
protected:
    ...
    enum VF_Type_cv habitat;
    _liste<class FV_ELEMENT*> inconnues;
public:
    VAR(const char* name);
    virtual INC* operator [] (INT i) const=0;
    virtual INC* operator () (class FV_ELEMENT* v) const=0;
    virtual INC* new_var(class FV_ELEMENT* v,const INT n=0)=0;
    ...
};

```

- **Variable scalaire:** Associée à un scalaire Φ appartenant à un type volume de contrôle et représentée par la classe *VF_VAR_SCALAIRE*.
- **Variable vectorielle:** Associée à un vecteur $\Phi = \{\Phi_1, \Phi_2, \Phi_3, \dots, \Phi_n\}$ appartenant à un seul type de volume de contrôle¹⁴. Les objets de ce type de variable sont créés par la classe *VF_VAR_VECTORIELLE*.
- **Variable projection:** Représentée par la classe *VF_VAR_REC* et associée à une variable de reconstruction. Ces variables peuvent être classifiées en deux types:
 - **Variable projection scalaire:** $\tilde{\Phi}$ (*Exemple:* Dans la résolution d'un laplacien, on applique $\nabla^2 \tilde{\Phi} = 0$) (classe *VF_VAR_SCA_REC*).
 - **Variable projection vectorielle:** $\tilde{\Phi}$ (*Exemple:* Les schémas *colocated* pour résoudre l'équation (1.7)) (classe *VF_VAR_VEC_REC*).

¹⁴ Schémas *colocated*.

Classification des inconnues

De même que pour les variables, les inconnues ont une classification biunivoque avec les variables auxquelles elles sont associées. Pour un i -ème volume de contrôle, cette classification est :

- **Inconnue scalaire:** Φ_i . Elle a comme *attributs* principaux :
 - *une valeur réelle,*
 - *type de conditions aux frontières,*
 - *une équation,* un objet décrivant l'équation algébrique (1.9).

Un objet de cette catégorie d'inconnues est créé par `VF_INC_SCALAIRE` dont sa structure principale est montrée par le code F.1 de la page 204.

- **Inconnue vectorielle:** $\Phi_i = \{\Phi_1, \Phi_2, \Phi_3, \dots, \Phi_n\}_i$, créée par la classe `VF_INC_VECTORIELLE`. *Attribut* principal : Une liste d'inconnues scalaires.
- **Inconnue projection:** Représentée par la classe `VF_RECONSTRUCTION` et associée à une inconnue de reconstruction qui porte comme *attributs* l'information :
 - *Un stencil*¹⁵ *intégral* S_f *ou projectif* S_p , de taille variable.
 - *Une référence à la variable primitive,* Φ ou $\tilde{\Phi}$.
 - *Un ensemble de bords* B , ($B \subset \mathbb{B}$), de taille variable.
 - *Sa résidence*¹⁶ $\mathcal{R}(\tilde{\Phi})$, sur B .

Les inconnues de projection sont divisées aussi en deux types :

- **Inconnue projection scalaire:** $\tilde{\Phi}_i$, `VF_RECONSTRUCTION_SCA`.
- **Inconnue projection vectorielle:** $\tilde{\Phi}_i$, `VF_RECONSTRUCTION_VEC`.

¹⁵ Voir la Définition 5.2.

¹⁶ Voir la Définition 5.5.

Les inconnues possèdent des mécanismes d'accès aux supports algébriques¹⁷ qui sont partagés avec les objets de la classe *VF_BORD*, selon l'application.

Variables et inconnues dérivent des classes de base *VAR* et *INC*, respectivement. Cette disposition est montrée par l'arbre de dérivation de la figure F.1 de la page 203.

6.2.3 Génération des équations algébriques

Les équations algébriques auxquelles on fait ici référence sont de la même forme que l'équation (1.9), créées après le bilan des flux convectifs-diffusifs et suite au processus de la distribution (sous-section 6.3.2). Ces flux-là sont calculés à l'aide des supports provenant du PROTOCOLE 2, enrichi de l'information de *stencils intégrals*¹⁸ S_f et des bords B .

Rappelons-nous, d'après la sous-section précédente ou le code F.1 de la page 204, que chaque inconnue scalaire Φ contient parmi ses *attributs* un objet décrivant une équation algébrique.

Réécrivons la formulation discrète de l'équation de convection-diffusion (1.7) en termes de coefficients, équation (1.9) :

$$a_0 \cdot \Phi_0 + \sum_{k=1}^m (a_k \cdot \Phi_k) = b_0 \quad (6.1)$$

avec Φ étant toujours une inconnue scalaire, a_i sont les coefficients dépendants de la *distribution des flux* et b_0 une constante.

¹⁷ Voir la sous-section 6.2.3.

¹⁸ Voir la Définition 5.3.

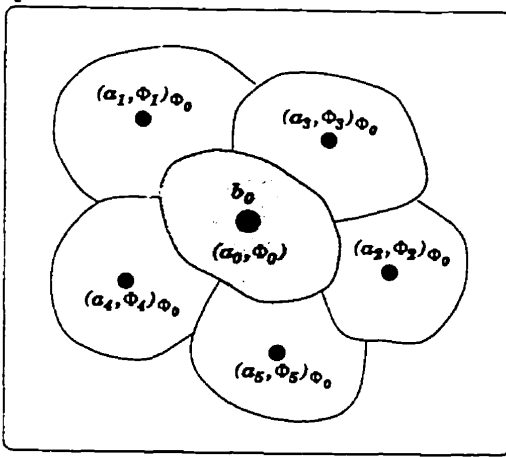


FIGURE 6.6: *Inconnues satellites dans la discrétisation finale des équations*

S'il s'agit d'une inconnue vectorielle Φ , cette génération est propagée à chaque composante scalaire pouvant être modifiée à cause des conditions aux frontières.

La figure 6.6 illustre un exemple abstrait de la connectivité des inconnues liée à l'inconnue Φ_0 créée automatiquement dans le processus d'intégration en 2D. En général, peu importe la dimension du problème et le type de volume de contrôle du tableau 4.1,

pour une inconnue scalaire donnée Φ_0 , le membre gauche de l'équation (6.1) peut être décomposé en une sommation de *termes* comme suit :

$$\overbrace{a_0 \cdot \Phi_0 + a_1 \cdot \Phi_1 + \dots + a_m \cdot \Phi_m}^{\text{ÉQUATION}} = b_0. \quad (6.2)$$

$\text{terme}_0 = (a_0, \Phi_0) \quad \text{terme}_1 = (a_1, \Phi_1) \quad \text{terme}_m = (a_m, \Phi_m)$

À la lumière de cette description, cette *ÉQUATION* est créée dynamiquement avec le recours des deux classes servant à la génération automatique de cette formulation : ce sont les classes *VF_TERME* et *VF_EQUATION*. Sous-entendu, les *attributs* de la première classe doivent être le couple $(a, \Phi)_i$ et ceux de la deuxième une liste dynamique de ces couples et une valeur représentant le terme de droite. La classe *VF_EQUATION* contient les méthodes servant à l'affectation et à l'accès de ces *attributs*¹⁹.

Ces caractéristiques sont montrées par le code 6.3.

¹⁹ Par exemple voir l'instantiation correspondante dans le code H.2 de la page 214.

Code 6.3 *Les classes VF_TERME et VF_EQUATION*

```

class VF_TERME
{
    REAL coefficient;
    class VF_INC_SCALAIRE* inconnue;
public:
    VF_TERME(REAL coef,class VF_INC_SCALAIRE* inc);
    ...
};
class VF_EQUATION
{
    _liste<VF_TERME*> termes;
    REAL terme_droite;
public:
    VF_EQUATION(void);
    ...
};

```

6.2.4 *Gestion de la résolution des équations*

Pour des calculs implicites ou semi-implicites on fait appel aux systèmes de solution, classifiés selon la nature et l'approche de la résolution. Un ensemble de classes sert, entre autres, à la résolution de systèmes d'équations matricielles par blocs ou par éléments avec un stockage en ligne de ciel.

6.3 Supports opérationnels

Ces supports sont un ensemble de processus et outils généraux servant d'intermédiaire entre le *kernel* du sous-système et les applications mêmes (figure 6.1). Pour réussir dans cette entreprise ces supports emploient les objets et structures de données fournis par le *kernel*.

Parmi ces supports on peut mentionner :

- L'intégration de l'équation (1.5).
- Gestion de la distribution des flux.
- Fonctions d'interpolation.
- La fonction **LSQ** générique.
- Mécanismes d'accès.

6.3.1 *L'intégration*

Pour les deux PROTOCOLES la technique d'intégration numérique dans $\mathbf{vf}$, est basée sur un bouclage sur la résidence de l'interpolant²⁰ $\mathcal{R}(\tilde{\Phi})$.

Examinons certaines particularités de ce processus.

Intégration avec le PROTOCOLE 1

L'intégration avec ce type de PROTOCOLE a une caractéristique simple et générale, celle de partager le flux, calculé sur un bord, comme des contributions des flux gauche et droit et une correction. Dans ce contexte l'intégration générique multidimensionnelle pour n'importe quel type de disposition de la variable selon, le tableau 4.1 est possible.

Ainsi par exemple, l'intégration explicite²¹ et la distribution se réalise en deux séquences :

²⁰ Voir la section 5.2.

²¹ Pour l'intégration implicite ce processus est plus directe : on intègre, on forme la matrice globale en la stockant en *ligne du ciel* et finalement on la résout à l'aide des résolveurs matriciels du système (sous-section 6.2.4).

- *Intégration et distribution*²² : boucle sur tous les bords, selon le code 6.4.
- *Application de l'équation (1.8)*²³ : boucle sur tous les éléments, selon le code 6.5.

Code 6.4 *Intégration et distribution avec le PROTOCOLE 1*

Sur les bords intérieurs :

```
...
VF_FLUX flux_d,flux_g,flux_c; // User dependent
...
VF_FLUX flux=0.5*(flux_g+flux_d-flux_c); // Integration
var_g->applique(+S,flux); // Distribution to left
var_d->applique(-S,flux); // Distribution to right
```

Sur les bords-frontières :

```
...
VF_FLUX flux_d,flux_g,flux_c; // User dependent
...
VF_FLUX flux=0.5*(flux_g+flux_d-flux_c); // Integration
var_g->applique(+S,flux); // Distribution to left, right=NIL
```

Code 6.5 *Intégration dans tout le domaine pour le PROTOCOLE 1*

```
void VAR::applique_correction(REAL dt)
{
    INT i;
    forAllVolumes {
        REAL Vol=inconnues[i]->volume();
        (*this)[i]->applique_correction(dt/Vol);
    }
};
```

²² Dans cet exemple, les flux sont calculés à l'aide de la classe *VF_FLUX* selon l'équation (G.1) de la page 207 ou le type d'application (voir la sous-section 6.4.1). La méthode *applique(Real, VF_FLUX)*, des classes *VF_INC_SCALAIRE* ou *VF_INC_VECTORIELLE*, permet la mise à jour du terme $\sum_{j=1}^m J_j^n n_j \Delta s_j$ de l'équation (1.8) et son stockage dans un *buffer* lié à l'inconnue Φ d'intégration. S est la norme du bord (voir la sous-section 6.2.1).

²³ Cette procédure se réalise à l'aide d'une méthode *applique_correction(Real)* de la classe *VAR*.

Intégration avec le PROTOCOLE 2

L'intégration avec ce PROTOCOLE utilise un *stencil intégral par omission* S_f (sous-section 5.2.2), un ensemble de bords $\mathcal{B}$ ($\mathcal{B} \subset \mathbb{B}$) et des interpolants associés.

Par conséquent, ce processus est lié principalement à la création de la *matrice de flux* $\mathbf{J}$. Cette matrice rectangulaire²⁴ est créée lors de l'intégration des termes convectifs et diffusifs de l'équation (1.3). Son ordre, variant dans le cas plus général pour un seul problème, est fonction de la taille du *stencil intégral* S_f et de la liste des bords intérieurs, $\mathcal{B}$ ($\mathcal{B} \subset \mathbb{B}$), associés à chaque interpolant $\tilde{\Phi}$ (voir la sous-section 6.2.2), comme on l'illustre dans le tableau 6.2. Le contenu de la matrice $\mathbf{J}$ dépend du schéma numérique que l'utilisateur implante, et elle est créée en faisant une double boucle comme le montre le code H.1 de la page 213, valide en 2D ou 3D et pour tous les types de volume de contrôle du tableau 4.1, autres que CCCV.

Dans le processus d'intégration une particularité critique est le calcul du paramètre convectif dépendant du schéma *upwind* utilisé. Dans la sous-section 6.4.2 on fera un résumé des applications faites qui utilisent ces schémas et cette structure.

²⁴ Selon la conception des types de volumes de contrôle condensés dans le tableau 4.1 en 2D cette matrice est toujours carrée et égale au nombre d'arêtes ou sommets, car le nombre de termes est égal au nombre de bords intérieurs. En 3D elle est rectangulaire, par exemple pour le cas ECCV avec des hexaèdres, elle est de 24×12 .

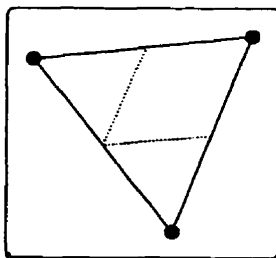


FIGURE 6.7: Les bords de contrôle de Saabas

Cependant, étant donné la Remarque 1.2 de la page 6 et suite à l'analyse du coût de la mémoire requise pour le traitement de ladite matrice et aux difficultés de sa formation, il est légitime de diminuer sa taille, et, par conséquent, d'augmenter la vitesse de calcul, s'il s'agit d'une disposition VCCV. Une telle disposition à considérer dans cette direction est donnée, par exemple, dans la discrétisation du domaine 2D et 3D dans la thèse de Saabas H. [36] (voir la figure 6.7, pour cette discrétisation une matrice de 2×3 sera formée).

6.3.2 Distribution des flux

Dans le cas du PROTOCOLE 1 la distribution devient simple car les volumes de contrôle gauche et droit sont toujours disponibles immédiatement dans le processus d'intégration.

TABLEAU 6.2: La matrice rectangulaire du flux, J

J		Nombre de Termes du Stencil Intégral			
		1	2	3	...
Nombre de Bords Intérieurs	1	$J_{1,1}$	$J_{1,2}$	$J_{1,3}$	...
	2	$J_{2,1}$	$J_{2,2}$	$J_{2,3}$	...
	3	$J_{3,1}$	$J_{3,2}$	$J_{3,3}$	...
	4	$J_{4,1}$	$J_{4,2}$	$J_{4,3}$	...
	.	.	.	.	.
	.	.	.	.	$J_{i,j} = (\rho_i V_i \Phi_{i,j} - \Gamma_i \nabla \Phi_{i,j}) \cdot \mathbf{n}_i \Delta S_i$
	.	.	.	.	.

Pour le deuxième PROTOCOLE la distribution se réalise par l'intermédiaire de la méthode *distribue* de la classe *VF_RECONSTRUCTION_SCA*, selon le code H.2 de la page 214. Dans le cas d'une variable d'intégration vectorielle le processus se propage vers ses variables scalaires. Lors du processus de distribution la mise à jour des coefficients de l'équation (6.1) et l'arrangement du système d'équations matricielles se réalisent de façon automatique.

6.3.3 Fonctions d'interpolation

vf utilise un ensemble de fonctions d'interpolation administrées par la classe *VF_INTERPOLANT* permettant d'obtenir les valeurs de $\tilde{\Phi}(x, y, z)$ ou ses dérivées $(\frac{\partial \tilde{\Phi}}{\partial x}, \frac{\partial \tilde{\Phi}}{\partial y}, \frac{\partial \tilde{\Phi}}{\partial z})$.

Les fonctions d'interpolation par omission dans le système sont :

- **Interpolants linéaires:**

- *en 2D* : $\tilde{\Phi}(x, y) = a + b \cdot x + c \cdot y$

- *en 3D* : $\tilde{\Phi}(x, y, z) = a + b \cdot x + c \cdot y + d \cdot z$

- **Interpolants bi-linéaires:**

- *en 2D* : $\tilde{\Phi}(x, y) = a + b \cdot x + c \cdot y + d \cdot xy$

- *en 3D* : $\tilde{\Phi}(x, y, z) = a + b \cdot x + c \cdot y + d \cdot z + e \cdot xyz$

D'autres interpolants peuvent être définis par l'utilisateur en utilisant la même structure de données.

6.3.4 La fonction *LSQ* générique

Cette fonction, n'appartenant à aucune classe spécifique et servant à appliquer la technique des moindres carrés dans le calcul du gradient $\nabla\Phi$, a un caractère générique quant à la dimension du problème et au type de volume de contrôle du tableau 4.1. Basée fondamentalement sur une approche linéaire du calcul selon [25], cette fonction utilise un *stencil projectif*²⁵ S_p créé par l'utilisateur. Par défaut l'ensemble des éléments faisant partie du *stencil* est composé de tous les éléments connectés par le bord avec l'élément central comme le montre la figure 6.6.

Cette fonction, retournant un vecteur, a comme déclaration les lignes montrées par le code 6.6²⁶.

²⁵ Voir la Définition 5.3.

²⁶ Ici *idx* est l'index de l'inconnue qui habite dans le volume de contrôle (voir les sous-sections 6.2.1 et 6.3.5), *v0* l'élément central et *vols* l'ensemble des éléments satellites.

Code 6.6 *La fonction LSQ*

```
FV_VECTOR
  LSQ(INT idx,class FV_ELEMENT* v0,_liste<class FV_ELEMENT*>& vols);
```

6.3.5 Mécanismes d'accès aux inconnues

vf fournit des mécanismes d'accès aux objets. Parmi ceux-ci on peut nommer les mécanismes accédant aux inconnues à l'aide des opérateurs selon deux techniques :

- L'accès à partir d'un élément.
- L'accès à partir d'une variable primitive de référence.

Les descriptions de ces accès sont traitées dans l'annexe F de la page 203. La figure 6.8 montre le résumé de toutes les formes d'accès à l'inconnue *p* se trouvant à la 1-ème position de la liste d'inconnues du 2-ème élément.

6.4 Applications

Les applications de *vf* font partie du dernier disque dans la conception de l'interface résolveur (figure 6.1). Elles ont un caractère technique étant donné leur liaison avec les autres couches d'où l'information se déplace vers les implantations de l'utilisateur, et profitent des avantages offerts par le *kernel* et le support opérationnel.

6.4.1 Applications du PROTOCOLE 1

On a appliqué ce PROTOCOLE à la résolution multidimensionnelle des équations d'Euler en régime compressible.

ACCÈS AUX INCONNUES					
INC	ÉLÉMENT				
	0	1	2	...	n
0	V_0	V_1	V_2	...	V_n
1	P_0	P_1	P_2	...	P_n
...	...	...	...	...	...
m	T_0	T_1	T_2	...	T_n

$p_2 = \text{ele}_2 \rightarrow \text{var}(1)$

$p_2 = p[2]$

$p_2 = p(\text{ele}_2)$

FIGURE 6.8: Exemple d'accès aux inconnues

Formulation des équations d'Euler

Les équations d'Euler en régime non-permanent et en coordonnées cartésiennes sont décrites par l'équation [21] :

$$\frac{\partial U}{\partial \tau} + \sum_{k=1}^3 \frac{\partial W_k}{\partial x_k} = 0 \quad (6.3)$$

avec :

$$U = (\rho \ \rho u_1 \ \rho u_2 \ \rho u_3 \ \rho E)^T \quad (6.4)$$

et

$$W_k = \begin{pmatrix} \rho u_k \\ \rho u_1 u_k + p \cdot \delta_k^1 \\ \rho u_2 u_k + p \cdot \delta_k^2 \\ \rho u_3 u_k + p \cdot \delta_k^3 \\ (\rho E + p) u_k \end{pmatrix}$$

où δ est le *Kronecker number*.

Cet ensemble d'équations est complété avec l'équation d'état :

$$p = (\gamma - 1)\rho [E - (u_1^2 + u_2^2 + u_3^2)/2]$$

La formulation intégrale de l'équation (6.3), après l'application du Théorème de la divergence (1.1), prend la forme suivante :

$$\frac{\partial}{\partial \tau} \int_{\Omega} \mathbf{U} \cdot d\Omega + \oint_{\Gamma} \mathbf{H} \cdot d\Gamma = 0 \quad (6.5)$$

avec, $\mathbf{H}$ comme étant le flux :

$$\mathbf{H} = \begin{pmatrix} \rho u_n \\ \rho u_1 u_n + p \cdot n_x \\ \rho u_2 u_n + p \cdot n_y \\ \rho u_3 u_n + p \cdot n_z \\ \rho(E + p/\rho)u_n \end{pmatrix}$$

Formulation discrète

La formulation discrète de l'équation (6.5) pour un i -ème élément Ω_i ayant m_i bords est (voir l'équation (1.8)) :

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta \tau}{\Omega_i} \sum_{j=1}^{m_i} \mathbf{H}_{ij}^n \Delta s_{ij} \quad (6.6)$$

Avantages de l'application

Du côté de l'implantation, cette application a le grand avantage d'avoir la variable en cours $\mathbf{U}$ (6.4) comme étant vectorielle par rapport au stockage de chaque variable conservative, permettant ainsi d'utiliser un seul type de volume de contrôle du tableau 4.1. Cette particularité simplifie la mise en œuvre et la mise à jour et dispose des ressources informatiques de mémoire CPU.

Les principaux avantages de cette application sont :

- **La dimension du problème:** 2D ou 3D.
- **Le maillage initial:** provenant du fichier *Vie*, à la limite hybride : mélange de différents types de polygones (en 2D) ou mélange de tétraèdres, prismes et/ou hexaèdres (en 3D).
- **Le type de stockage:** Tous, indexés selon le tableau 4.1.

Caractéristiques numériques

Du point de vue numérique cette application utilise :

- **La discrétisation spatiale:** Le problème de Riemann est résolu par le schéma de Roe [34] dans le traitement de l'*upwind*, équation (G.1) de la page 207.
- **Le type d'intégration:** Explicite ou implicite dans le temps.
- **L'ordre:** 1-er ou 2-ème ordre.

Les détails particuliers, décrits dans l'annexe G de la page 207 et les résultats des cas-test, seront abordés dans la section 7.2.

6.4.2 Applications du *PROTOCOLE 2*

Le *PROTOCOLE 2* a été appliqué à la résolution multidimensionnelle des équations de *Navier-Stokes* en régime incompressible stationnaire avec des mailles décalées et en utilisant la méthode SIMPLER.

Formulation des équations de *Navier-Stokes*

Les équations de *Navier-Stokes* en régime incompressible et stationnaire sont exprimées en utilisant l'équation généralisée de transport (1.2) (section 1.2) :

$$\nabla \cdot \mathbf{J} = S \quad (6.7)$$

avec :

$$\mathbf{J} = \rho \Phi \mathbf{V} - \Gamma_{\Phi} \nabla \Phi \quad (6.8)$$

L'équation de *la continuité* est obtenue en plaçant $\Phi = \text{Const}$ et $S = 0$, et les équations du *mouvement* avec un vecteur $\Phi = \mathbf{V} = \{u, v, w\}$ avec le terme source comme étant un vecteur ∇p et le coefficient de diffusion Γ égal à la viscosité dynamique μ du fluide. Finalement les équations de *Navier-Stokes* en régime incompressible aboutissent aux équations du tableau 6.3.

Les équations discrètes en termes de coefficients des équations de *Navier-Stokes* sont quatre formulations telles que décrites par l'équation (6.1) : une pour chaque composante de la vitesse et une autre pour la pression, produit du couplage de l'équation du mouvement et celle de la continuité.

TABLEAU 6.3: *Équations vectorielles de Navier-Stokes en régime incompressible*

◦ <i>Équation de la continuité :</i> $\nabla \cdot \mathbf{V} = 0$ ◦ <i>Équations du mouvement :</i> $\mathbf{V} \cdot \nabla \mathbf{V} - \nu \nabla^2 \mathbf{V} + \frac{1}{\rho} \nabla p = 0$

Les schémas décalés

Avec les volumes de contrôle décalés, le stockage et le traitement des propriétés scalaires comme la pression et la température sur des volumes de contrôle, autres que ceux du vecteur vitesse, est possible²⁷.

Des exemples de cette représentation en 2D sont montrés par la figure 6.9. En 3D, cette représentation est semblable.

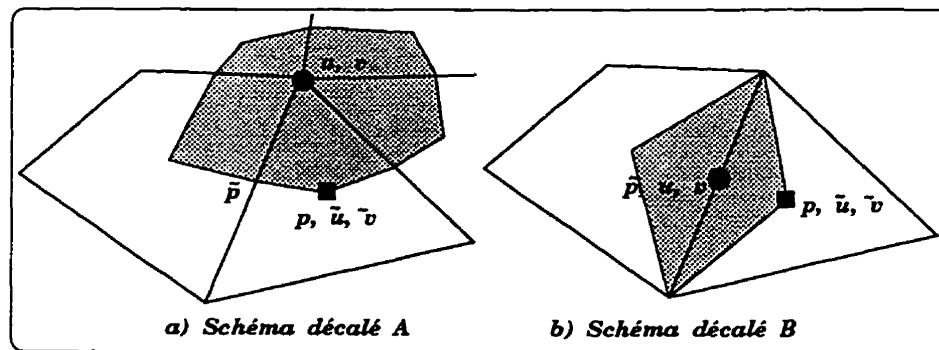


FIGURE 6.9: Représentation décalée de l'application en 2D

Avantages de l'application

Tout en fixant la pression avec un schéma CCCV, cette application porte plus de flexibilité quant à :

- La dimension du problème: 2D ou 3D.
- Les volumes de contrôle décalés: en 2D cette flexibilité est montrée par la figure 6.9.
- Les mailles hybrides: en 2D et 3D.

²⁷ L'approche avec des mailles décalées, avec la pression occupant une disposition CCCV et les vitesses une autre ECCV en 2D a été appliquée dans la résolution des équations de *Navier-Stokes* pour des écoulements turbulents, bien condensées dans le mémoire de maîtrise [32], en obtenant des résultats qu'éliminent l'influence de la pression en damier observée par les schémas *colocated*.

- **Deux schémas:** Pour le calcul des flux convectifs : **MWUS** et **FOUS**.

Schémas *upwind* utilisés

La partie convective de l'équation (6.8) est évaluée à l'aide de deux schémas numériques :

1. **Le schéma décentré pondéré par la masse**, **MWUS**, basé sur l'approche de Schneider [38] et calculé par la méthode **MWUS** appartenant à la classe **VF-VAR-REC**. Ce schéma porte sur la détermination du vecteur $\{\phi\}$ interpolé aux bords $\mathcal{B}$, ($\mathcal{B} \subset \mathbb{B}$) de la propriété convectante associée au vecteur $\{\Phi\}$ se trouvant dans le domaine de validité de l'interpolant (Définition 5.4) $D(\tilde{\Phi})$. Ces valeurs doivent être introduites dans l'équation (6.8) pour le calcul de la partie convective. Donc, pour un domaine 2D ou 3D, la suivante équation matricielle est valide :

$$\{\phi\}_{nb} = \underbrace{[A^{-1}]_{nb \times nb} \cdot [B]_{nb \times nt}}_{[C]_{nb \times nt}} \cdot \{\Phi\}_{nt} \quad (6.9)$$

Ici, $[A]$ est la matrice associée au vecteur $\{\phi\}$, $[B]$ celle associée au vecteur $\{\Phi\}$ et $[C]$ est la *matrice de convection* (rectangulaire). nb et nt sont reliés au nombre de bords $\mathcal{B}$ et au nombre de termes du stencil, respectivement. L'évaluation de la *matrice de convection*²⁸ $[C]$ pour les polygones suit le code H.3 de la page 216.

2. **Le schéma décentré exponentiel**²⁹, **FOUS** basé sur l'approche de Baliga et Patankar [4]. Calculé par la méthode **FOUS** de la classe **VF-VAR-REC**.

²⁸ Dû aux difficultés (complexité et coûts) dans la formation de cette équation matricielle pour un domaine 3D, ce schéma n'a été appliqué qu'à un domaine 2D avec des mailles polygonales et/ou hybrides avec deux types de dispositions du vecteur vitesse (ECCV et VCCV).

²⁹ Le grand avantage de ce schéma est son application à faible coût pour des domaines 2D et 3D. Ses désavantages sont, premièrement : il est appliqué aux faibles nombres de Peclet, deuxièmement :

De même que le schéma précédent, ce schéma porte sur la détermination d'une *matrice de convection* à partir d'une fonction d'interpolation dépendant de l'orientation de l'écoulement. Chaque élément sert de support dans l'interpolation exponentielle de la propriété suivant la direction moyenne de l'écoulement et linéaire dans les autres directions.

Tests préliminaires

Avant d'aborder la résolution des problèmes complexes d'utilisation de ce PROTOCOLE et des schémas utilisés, une étude méthodologique portant autant sur le comportement numérique que sur la vérification de la conservativité avec toutes les dispositions de la variable, a été faite en résolvant des équations moins complexes que celles de *Navier-Stokes*, dans l'ordre :

1. Résolution de l'équation de *Laplace*³⁰.
2. Test de la diffusion numérique³¹.

il n'est fiable que pour les dispositions VCCV (pour les autres dispositions l'extrapolation des valeurs pour $\{\phi\}$ aux bords génère de l'instabilité numérique) et, troisièmement : les éléments des supports doivent être triangulaires en 2D et tétraédriques en 3D.

Avec ces restrictions, on a appliqué dans cette étude ce schéma-là à la résolution des équations de *Navier-Stokes* en 2D et 3D.

³⁰ multidimensionnelle. Le terme source de l'équation (6.7) et le vecteur de vitesses $\mathbf{V}$ de l'équation (6.8) sont égaux à zéro :

$$\nabla^2 \bar{\Phi} = 0 \quad (6.10)$$

Résultat : Les cas-test nous montraient des résultats cohérents en 2D et 3D pour toutes les dispositions de la variable selon le tableau 4.1. Le principe de conservativité discuté dans la sous-section 1.2.1 a été respecté.

³¹ multidimensionnel, ayant pour objectif d'avoir un aperçu de la diffusion numérique en utilisant un schéma *upwind* quelconque et une disposition de la variable, selon le tableau 4.1, avec le test de *Smith & Hutton* [40].

Dans ce cas-test on demande la détermination du champ scalaire Φ à partir d'un champ de vitesses connu tel que décrit dans la sous-section 7.3.1, pour $Pe \rightarrow \infty$.

Résultat : Étant donné l'importance de ce cas-test préliminaire les résultats sont montrés et discutés dans la section 7.3.

3. Calcul des termes convectifs-diffusifs avec un *upwind* géométrique³².

Traitement de la pression

Le couplage des équations de mouvement et de la continuité aboutit à une équation discrète du style de l'équation (6.1) utilisant les valeurs des pseudo-vitesses $\hat{V}$, provenant de l'équation discrète du mouvement. Dans le cas général ces vitesses peuvent occuper différents types de volumes de contrôle. L'équation permettant leur approximation est donnée par³³ :

$$\hat{V} = \left[\underbrace{\frac{\sum_{k=1}^{k=m_u} (a_k^u u_k) + b^u}{a^u}}_{\text{sur } \Omega_u}, \underbrace{\frac{\sum_{k=1}^{k=m_v} (a_k^v v_k) + b^v}{a^v}}_{\text{sur } \Omega_v}, \underbrace{\frac{\sum_{k=1}^{k=m_w} (a_k^w w_k) + b^w}{a^w}}_{\text{sur } \Omega_w} \right] \quad (6.11)$$

La valeur de $\hat{V}$ aux frontières considère les valeurs fixes des vitesses si à celles-ci ont été appliquées des conditions de Dirichlet.

Finalement, dans le cas général, les valeurs de $\hat{V}$ doivent être interpolées aux bords de la cellule de la continuité³⁴.

³² Cas-test portant sur le calcul des équations de *Navier-Stokes* du tableau 6.3 avec un schéma *upwind* qui ne considère que la géométrie de la maille dans le calcul des termes convectif et diffusif de cette équation (approche faible de résolution).

Résultat : La faiblesse du schéma est manifeste, on n'a pas obtenu de bons résultats même pour des faibles nombres de Reynolds. Cette appréciation est valide pour toutes les dimensions et les différents stockages de la variable. Cependant ce cas-test a été très utile dans la mise à jour de la formation de la *matrice de flux J* (voir le tableau 6.2).

³³ Une composante de cette équation est obtenue par le premier argument de la méthode *hat(RealS, RealS)* appartenant à la classe *VF_INC_SCALAIRE*. La deuxième donne la valeur de a^u, a^v ou a^w .

Si les vitesses occupent un seul type de volume de contrôle le vecteur $\hat{V}$ de l'équation (6.11) est obtenue directement par le premier argument de la méthode *hat(FV_VECTORs, FV_VECTORs)* de la classe *VF_INC_VECTORIELLE*. La deuxième argument donne le vecteur (a^u, a^v, a^w) .

³⁴ Ce qui n'est pas nécessaire pour le schéma décalé de la figure 6.9(b).

Imposition des conditions aux frontières dans le PROTOCOLE 2

L'imposition des conditions aux frontières pour les éléments-frontières de la première catégorie exige un traitement spécial. En général ils utilisent une cellule miroir (*voir* la sous-section 6.2.1) avec sa valeur et sa position représentatives.

Si la pression occupe une disposition CCCV, ce qui correspond à cette catégorie d'élément-frontière, et si elle n'a pas de valeurs fixes aux frontières, le responsable de cette imposition est l'équation (6.11), là où on impose indirectement les conditions de *Dirichlet* pour les vitesses lors du couplage vitesse-pression.

Les conditions aux frontières sur les bords de la deuxième catégorie d'élément-frontière est décomposée en deux types :

- **Condition de Dirichlet**

L'application d'une valeur spécifique à l'élément-frontière est exécutée par la méthode *imposeDirichlet(Real)* de la classe *VF_INC_SCALAIRE*, en passant la valeur fixe dans l'argument. S'il s'agit d'une variable vectorielle cette imposition est propagée aux composantes de la variable.

- **Condition de Neumann**

Cette condition, imposée par la méthode *imposeNeumann(Real,Real)* de la classe *VF_INC_SCALAIRE* utilise l'expression linéarisée aux frontières :

$$\frac{\partial \Phi}{\partial n} = q_1 + q_2 \cdot \Phi \quad (6.12)$$

Ici q_1 et q_2 correspondent au premier et deuxième arguments, respectivement, de la méthode décrite.

L'appel aux méthodes lors d'une imposition de conditions aux frontières met à jour automatiquement les coefficients de l'équation (6.1) et arrange le système d'équations matricielles.

Le programme principal de l'application générique

L'intégration semi-implicite des équations de *Navier-Stokes* du tableau 6.3 avec la méthode SIMPLER se fait avec un processus itératif jusqu'à convergence, selon le code H.4 de la page 217. Ce code montre en même temps tous les processus faisant partie de cette interface.

D'autres descriptions concernant cette application sont abordées dans l'annexe H de la page 212 et les résultats des cas-test seront traités dans la section 7.3.

Chapitre 7

Résultats

7.1 Introduction

Dans le but de mettre à l'épreuve la fonctionnalité de *VF++* et la fiabilité de ces interfaces, on montrera les résultats les plus importants obtenus avec les deux PROTOCOLES.

Dans ce contexte les objectifs techniques et scientifiques de cette validation sont ordonnés de la façon suivante :

1. La mise à jour des interfaces et le respect de la loi de conservation décrite dans la sous-section 1.2.1.
2. L'évaluation qualitative (*condition nécessaire*) et quantitative (*condition suffisante*) des résultats ainsi que de la viabilité de l'utilisation des différents types de stockage selon le tableau 4.1. L'étude de la condition suffisante est traitée partiellement.

À cet égard une grande famille de cas-test peuvent être exécutés dû au très haut degré de liberté dans le choix de la solution. On constate que pour un seul type

de problème avec les mêmes conditions, le nombre de cas-test possibles est, alors, calculé par la formule :

$$\text{Nombre de cas test} \approx CGEO \cdot CSOL \cdot (DIM + 1)$$

Où *CGEO* est la constante dépendant de la forme géométrique des éléments initiaux, *CSOL* est une constante dépendant du degré de liberté de la résolution. Le facteur $(DIM + 1)$ considère le choix selon le type de volume de contrôle du tableau 4.1. Autrement dit, par exemple, en 2D si on utilise que des triangles dans le maillage initial ($CGEO = 1$) et un seul type d'implantation numérique ($CSOL = 1$), alors le nombre de solutions à obtenir est égal à 3, pour un seul type de conditions limites.

Étant donné que dans la discrétisation initiale on peut aussi utiliser des mailles hybrides, alors ce nombre devient plus grand. Dans l'état actuel, pour le PROTOCOLE 1 $CSOL = 4$, pour le deuxième PROTOCOLE ce chiffre est encore plus grand.

Dans ce chapitre, on utilise quatre environnements globaux numérotés afin de faciliter la rédaction et la compréhension :

1. **Cas-test.** La déclaration et l'indexation d'un cas-test.
2. **Maillage initial.** La déclaration et l'indexation du maillage initial.
3. **Maillage final.** *Idem*, pour le final.
4. **Observation.** La définition et l'indexation des observations importantes associées au cas-test.

Pour des raisons d'espace on ne montrera que les résultats les plus critiques.

7.2 Résultats de l'application du PROTOCOLE 1

Cette section est dédiée à la présentation et à la discussion des résultats obtenus à l'aide du schéma de *Roe* dans la résolution multidimensionnelle des équations d'*Euler* (6.3) en régime compressible en utilisant toutes les dispositions de la variable vectorielle $\mathbf{U}$ selon le tableau 4.1 avec les cas-test suivants :

Cas-test 7.1 (Le problème de l'onde réfléchie) *Application du schéma de Roe à l'onde réfléchie avec toutes les dispositions de la variable du tableau 4.1 (voir la page 97).*

Cas-test 7.2 (Écoulements autour d'un profil NACA 0012) *Écoulements transsoniques pour la NACA 0012 avec angles d'incidence (voir la page 103).*

Cas-test 7.3 (Écoulement dans un cylindre) *Écoulement interne à $M_\infty = 3.97$ et à fortes incidences (voir la page 116).*

Cas-test 7.4 (Écoulement dans une tuyère convergente-divergente) *Provocation d'ondes de choc à l'intérieur d'une tuyère (voir la page 137).*

Cas-test 7.5 (Écoulement autour d'un profil à double ellipse) *Écoulement supersonique avec angle d'incidence (voir la page 151).*

Tous les calculs explicites ont été réalisés en utilisant le code G.1 de la page 210 du programme principal avec un coefficient de stabilité $CFL \approx (0.7, \dots, 0.9)$. En implicite, pour certains cas, celui-ci oscillait entre 5 et 50.

Les observations décrites ci-dessous ont comme base non seulement les résultats visuels des cas-test qui sont illustrés dans ce rapport mais aussi les résultats avec d'autres mailles de volume de contrôle qui ne sont pas présentés ici.

7.2.1 *Le problème de l'onde réfléchie*

Proposition du problème

Il s'agit de la détermination des variables conservatives $\rho, \rho u, \rho v, \rho w, \rho E$ des équations d'Euler, et, par conséquent, la distribution du nombre de Mach, dans un domaine rectangulaire fixe avec des conditions telles que montrées par la figure 7.1.

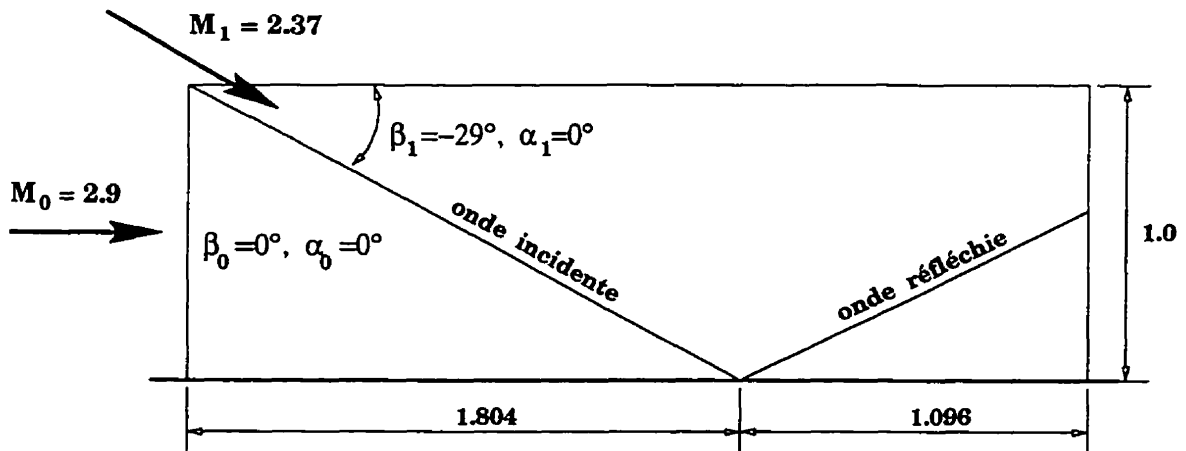


FIGURE 7.1: *Domaine de calcul et conditions pour le Cas-test 7.1*

Maillage initial

Pour la résolution de ce problème on a utilisé deux types de maillages initiaux :

Maillage initial 7.1 ($20 \times 10 \times 2$) *Maillage 2D non structuré, grossier, composé de triangles, tel qu'illustré par la figure 7.2(c).*

Maillage initial 7.2 ($10 \times 6 \times 6 \times 5$) *Maillage 3D non structuré, composé de tétraèdres, tel que montré par la figure 7.3.*

Maillage final

Les maillages finaux obtenus et utilisés dans ce cas-test sont :

Maillage final 7.1 (VCCV) *Obtenu du Maillage initial 7.1, avec 231 éléments et 1200/120 bords, tel que montré par la figure 7.2(a).*

Maillage final 7.2 (ECCV) *Obtenu du Maillage initial 7.1, avec 630 éléments et 1200/60 bords, tel qu'illustré par la figure 7.2(b).*

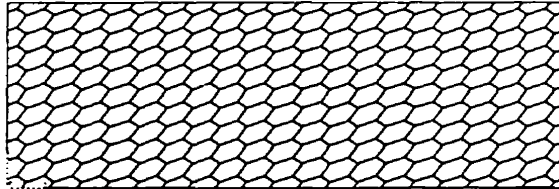
Maillage final 7.3 (CCCV) *Même que le Maillage initial 7.1, avec 400 éléments et 570/60 bords, tel que montré par la figure 7.2(c).*

Maillage final 7.4 (VCCV) *Obtenu du Maillage initial 7.2, avec 539 éléments et 10800/ 1872 bords, (pas de graphique).*

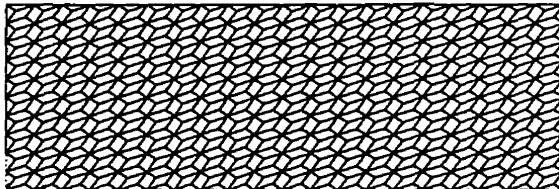
Maillage final 7.5 (ECCV) *Obtenu du Maillage initial 7.2, avec 2650 éléments et 21600/ 1872 bords, (pas de graphique).*

Maillage final 7.6 (FCCV) *Obtenu du Maillage initial 7.2, avec 3912 éléments et 10800/ 624 bords, (pas de graphique).*

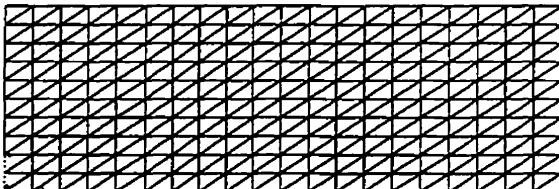
Maillage final 7.7 (CCCV) *Même que le Maillage initial 7.2, avec 1800 éléments et 3288/624 bords, tel que montré par la figure 7.3.*



(a) Le Maillage final 7.1, VCCV



(b) Le Maillage final 7.2, ECCV



(c) Le Maillage initial 7.1 (ou final 7.3)

FIGURE 7.2: *Maillages 2D utilisés dans le Cas-test 7.1*

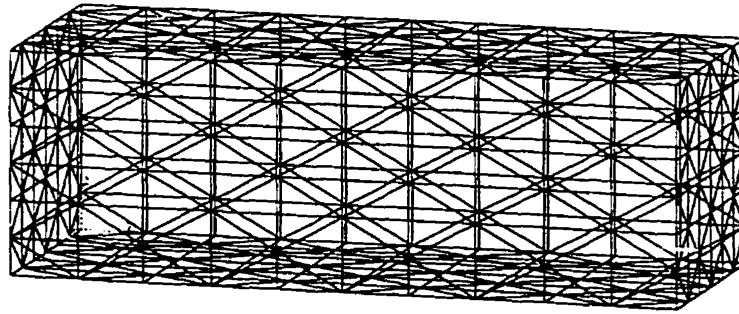


FIGURE 7.3: *Le Maillage initial 7.2 (ou final 7.7)*

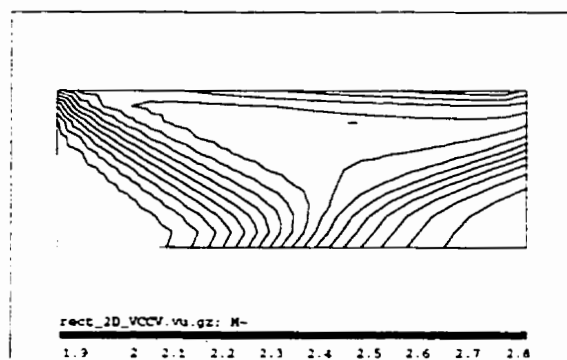
Résultats

La distribution du nombre de Mach en 2D accompagnée de la convergence respective est montrée par la figure 7.4 pour toutes les dispositions de la variable vectorielle U . Les vues isométriques des résultats du nombre de Mach en 3D, avec des valeurs constantes par volume de contrôle, selon les dispositions de la variable, ainsi que l'historique de la convergence, sont montrés par la figure 7.5.

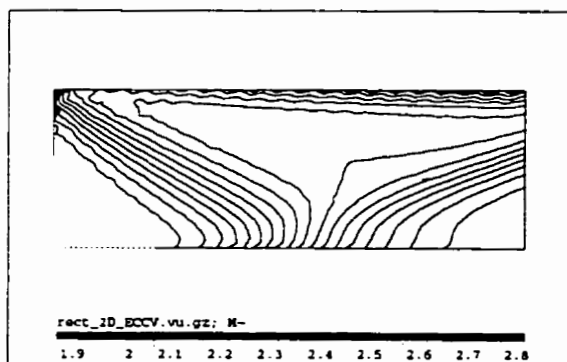
En considérant que les mailles initiales sont grossières, ce cas-test aboutit aux observations suivantes :

Observation 7.1 *Principe de conservation respecté.*

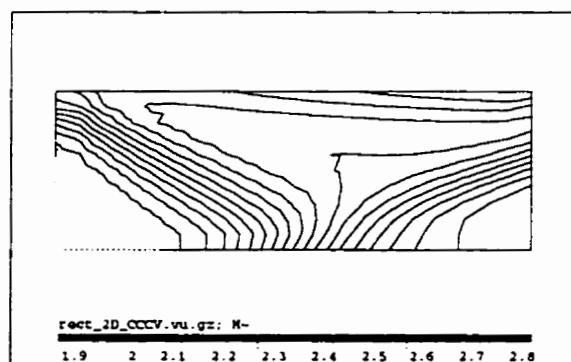
Observation 7.2 *En explicite la convergence est plus rapide pour VCCV, plus lente pour le ECCV en 2D et FCCV en 3D. Celle de CCCV occupe une position intermédiaire. Cette tendance a été observée aussi avec des mailles plus fines.*



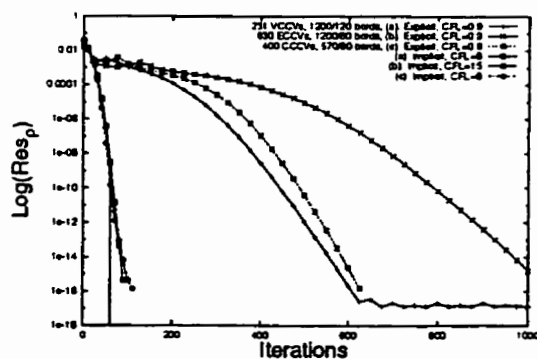
(a) Avec le Maillage final 7.1, VCCV



(b) Avec le Maillage final 7.2, ECCV



(c) Avec le Maillage initial 7.1 (ou final 7.3), CCCV



(d) Convergence

FIGURE 7.4: *Distribution du nombre de Mach pour le Cas-test 7.1 avec le Maillage initial 7.1. Convergence de calcul*

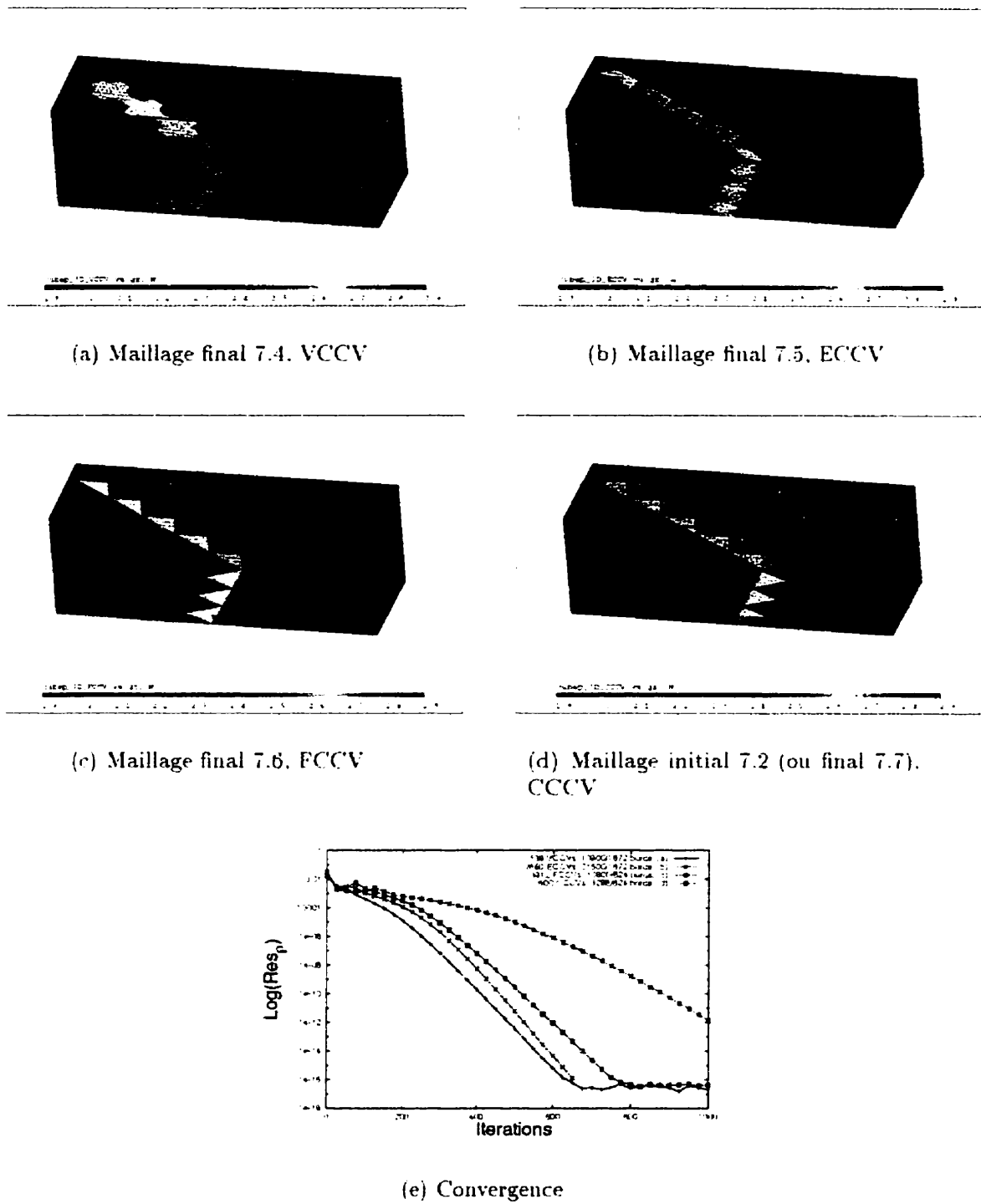


FIGURE 7.5: Visualisation des valeurs calculées du nombre de Mach pour le Cas-test 7.1 avec le Maillage initial 7.2. Convergence de calcul

Observation 7.3 *Résultats très diffusifs du test en 2D à cause de la grossièreté du maillage et sa mauvaise orientation. Malgré cela, il y a une légère amélioration dans la définition des iso-lignes pour le ECCV suivi de CCCV proche du point d'incidence.*

Observation 7.4 *Résultats moins diffusifs du test en 3D à cause de la meilleure orientation des mailles de volume de contrôle pour FCCV et CCCV (voir les figures 7.5(c) et 7.5(d)).*

7.2.2 *Écoulements transsoniques autour d'un profil NACA 0012*

Proposition du problème

Trois Cas-test ont été étudiés :

1. $M_\infty = 0.8$, $\alpha = 0^\circ$. Test de symétrie. Une onde de choc sur l'intrados et l'extrados apparaîtra à approximativement 50% du bord du profil.
2. $M_\infty = 0.8$, $\alpha = 1.25^\circ$. Une onde de choc apparaîtra dans l'extrados (à $\approx 60\%$). Une autre plus faible à $\approx 30\%$ dans l'intrados.
3. $M_\infty = 0.95$, $\alpha = 0^\circ$. Le choc symétrique est placé juste à la queue du profil, la distribution des paramètres dans le domaine en arrière a la forme d'une "fish tail".

Maillage initial

Pour résoudre ce problème on a utilisé trois maillages initiaux :

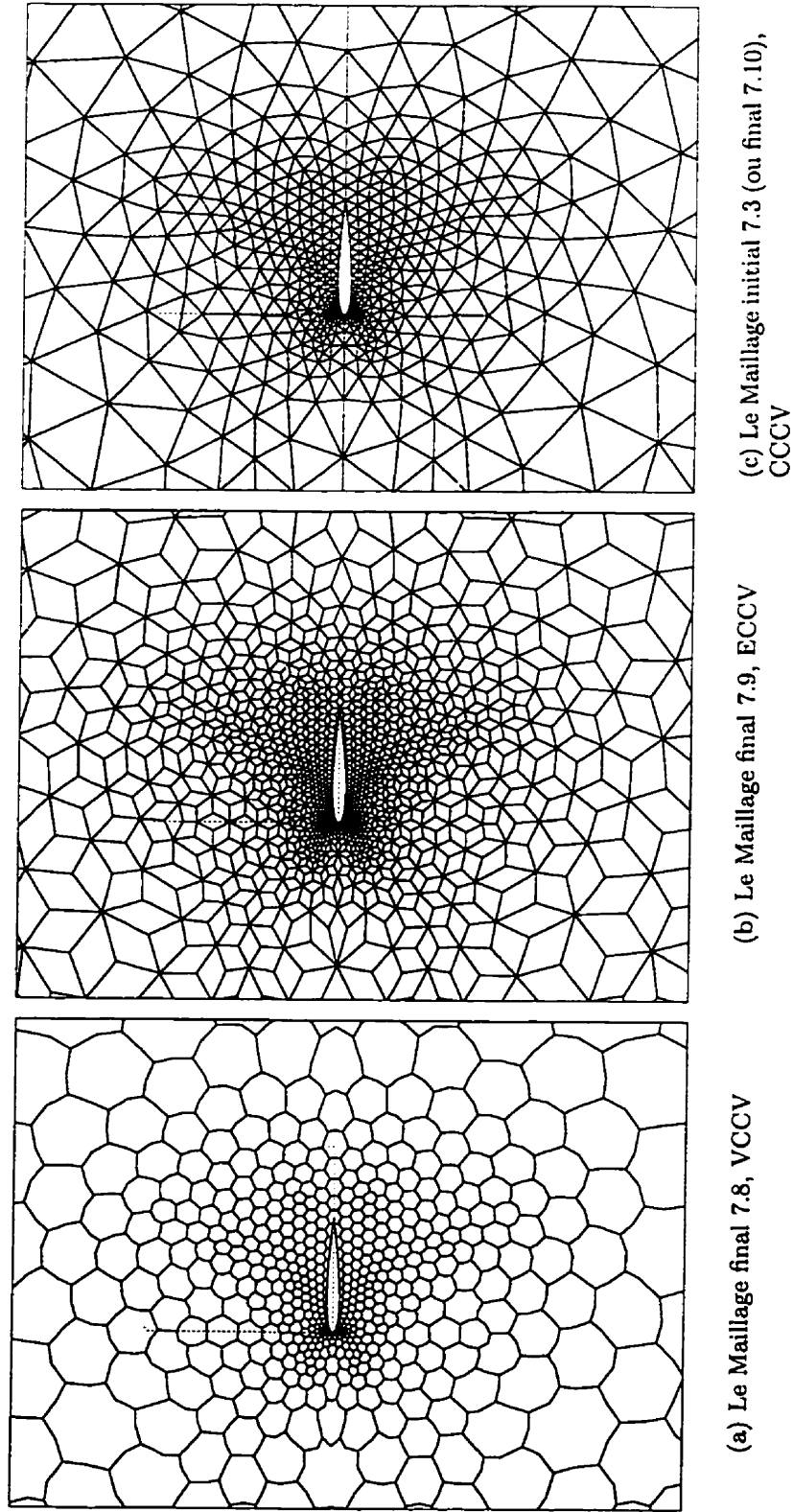


FIGURE 7.6: *Maillages 2D utilisés dans le Cas-test 7.2*

Maillage initial 7.3 () *Maillage 2D non structuré, grossier, emprunté de [8], composé de 1516 triangles, tel qu'illustré par la figure 7.6(c).*

Maillage initial 7.4 () *Maillage 2D non structuré, plus fin, composé de 6191 triangles, (pas de graphique).*

Maillage initial 7.5 ($50 \times 14 \times 6$) *Maillage 3D structuré, composé de mailles hexaédriques, illustré par la figure 7.7.*

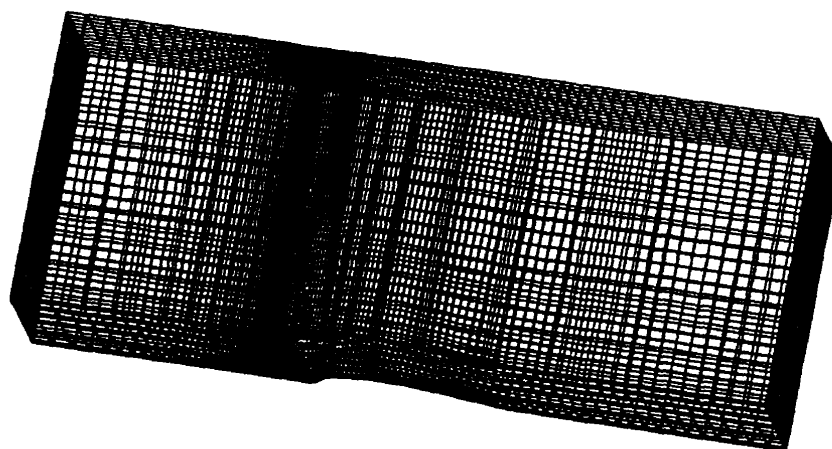


FIGURE 7.7: *Le Maillage initial 7.5 (ou final 7.14) utilisé dans le Cas-test 7.2*

Maillage final

Maillage final 7.8 (VCCV) *Obtenu du Maillage initial 7.3, avec 801 éléments et 4548/172 bords, tel que montré par la figure 7.6(a).*

Maillage final 7.9 (ECCV) *Obtenu du Maillage initial 7.3, avec 2317 éléments et 4548/86 bords, tel qu'illustré par la figure 7.6(b).*

Maillage final 7.10 (CCCV) *Même que le Maillage initial 7.3, avec 1516 éléments et 2231/86 bords, tel que montré par la figure 7.6(c).*

Maillage final 7.11 (VCCV) *Obtenu du Maillage initial 7.5, avec 5355 éléments et 50400/8672 bords, (pas de graphique).*

Maillage final 7.12 (ECCV) *Obtenu du Maillage initial 7.5, avec 14838 éléments et 100800 /8672 bords, (pas de graphique).*

Maillage final 7.13 (FCCV) *Obtenu du Maillage initial 7.5, avec 13684 éléments et 50400/2168 bords, (pas de graphique).*

Maillage final 7.14 (CCCV) *Même que le Maillage initial 7.5, avec 4200 éléments et 11516/2168 bords, tel que montré par la figure 7.7.*

Résultats

Les résultats de cette étude semi-paramétrique en 2D, en utilisant toutes les dispositions de la variable selon le tableau 4.1, pour chaque condition définie dans le paragraphe précédent sont montrés sur les figures 7.8, 7.9 et 7.10 pour le Maillage initial 7.3 et en 3D ceux-ci sont illustrés par la figure 7.14. Tous ces résultats sont accompagnés de courbes comparatives empruntées de [19] ou [8] avec leurs convergences.

Le résidu rapporté à la densité adimensionnelle a atteint difficilement $10^{-6}..10^{-7}$, ce qui est acceptable pour des domaines de calcul très grands.

Les observations suivantes sont une synthèse résumée des tests présentés dans ce rapport qui corroborent celles observées lors de l'utilisation des mailles même plus fines.

Observation 7.5 *Principe de conservation respecté.*

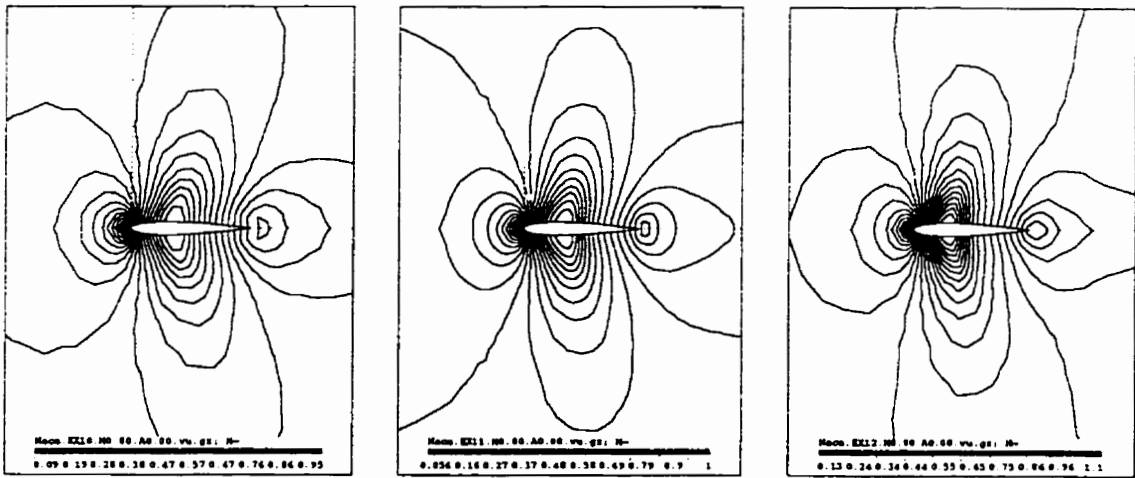
Observation 7.6 *En explicite la convergence est plus rapide pour VCCV, celle pour ECCV et CCCV a presque la même allure en 2D. En 3D les solutions avec ECCV et FCCV sont plus lentes. Au 2-ème ordre des oscillations sont constatées dans la convergence de la solution. Probablement l'origine de ces oscillations est due au limiteur et/ou du calcul des gradients par la méthode des moindres carrés utilisés. En 3D, la convergence est atteinte avec une chute suivie d'une asymptote, mais cette dernière caractéristique n'est pas générale pour les autres tests.*

Observation 7.7 *Résultats très diffusifs pour les solutions VCCV en 2D ou 3D. Moins diffusifs pour ECCV suivis de CCCV et FCCV.*

Observation 7.8 *Résultats symétriques pour la 1-ère et 3-ème condition étant donné la symétrie de la maille.*

Observation 7.9 *Au premier ordre les résultats avec VCCV sont semblables à ceux obtenus par [8] avec le même type de volume de contrôle. Pour les deux premiers tests la meilleure solution est partagée entre CCCV et ECCV. Par contre, pour la deuxième condition, ECCV est moins diffusif au bord d'incidence et de fuite.*

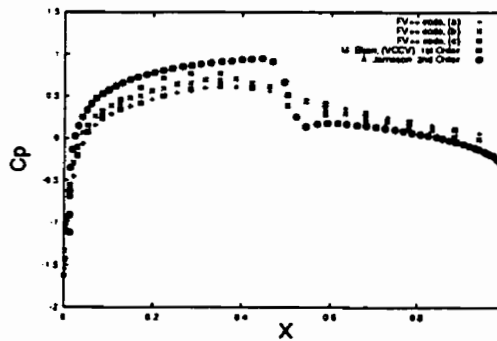
Observation 7.10 *Au deuxième ordre une légère amélioration est aperçue pour toutes les dispositions. Celle-ci est mieux garantie avec la configuration CCCV suivi de ECCV pour la première et deuxième condition. Le ECCV suivi de CCCV ont donné*



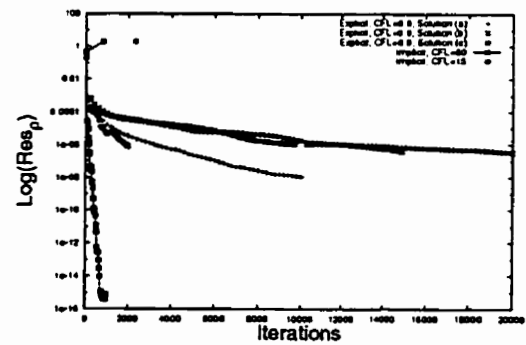
(a) Avec le Maillage final
7.8, VCCV

(b) Avec le Maillage final
7.9, ECCV

(c) Avec le Maillage initial
7.3 (ou final 7.10),
CCCV

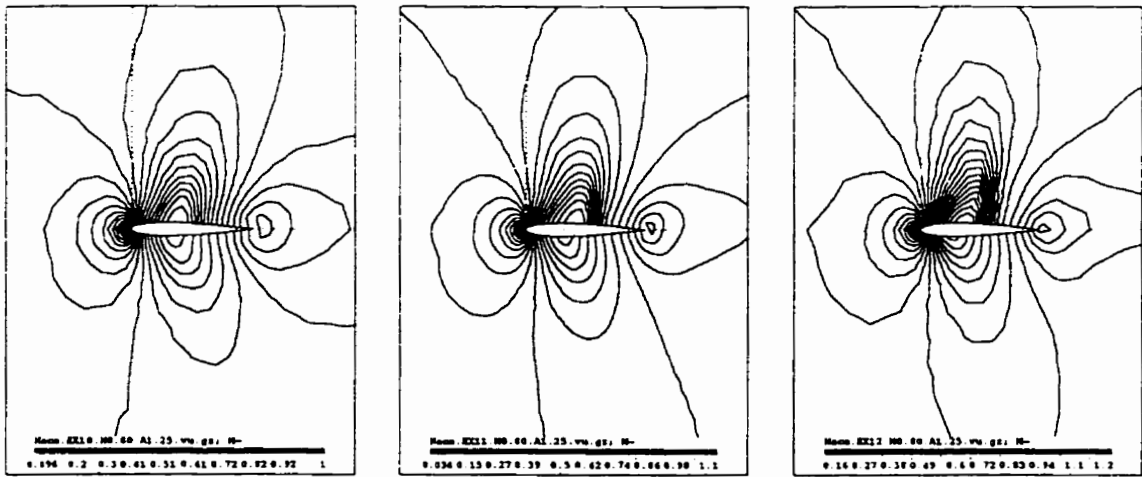


(d) Comparaisons de C_p



(e) Convergence

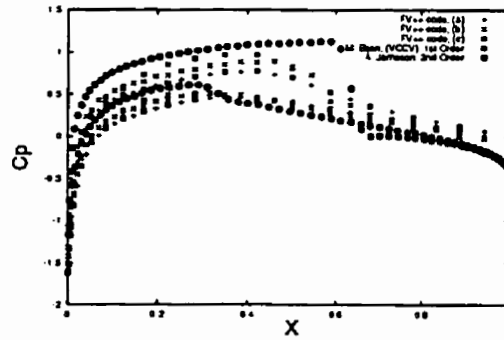
FIGURE 7.8: Lignes iso-Mach pour le Cas-test 7.2 avec le Maillage initial 7.3. $M_\infty = 0.8$, $\alpha = 0^\circ$. Premier ordre. Comparaisons de C_p et convergence de calcul



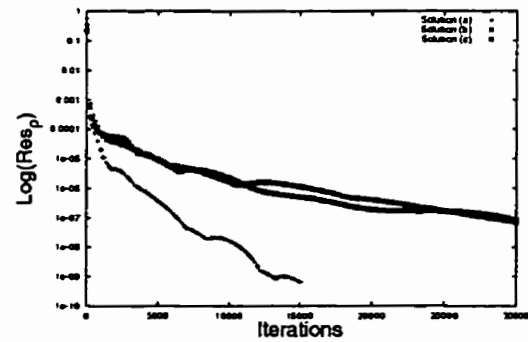
(a) Avec le Maillage final
7.8, VCCV

(b) Avec le Maillage final
7.9, ECCV

(c) Avec le Maillage ini-
tial 7.3 (ou final 7.10),
CCCV

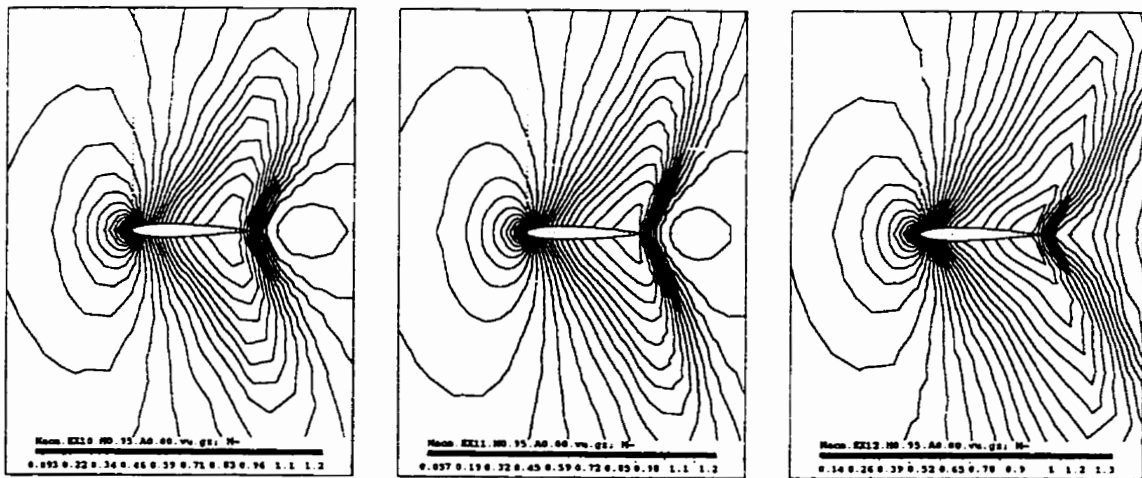


(d) Comparaisons de C_p



(e) Convergence

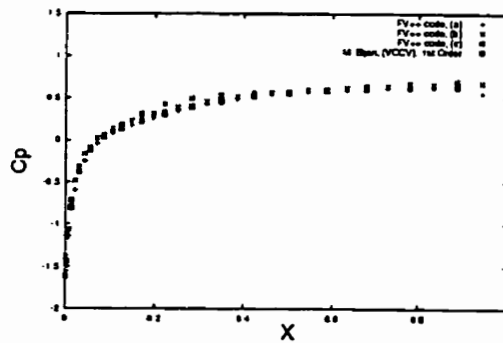
FIGURE 7.9: Lignes iso-Mach pour le Cas-test 7.2 avec le Maillage initial 7.3. $M_\infty = 0.8$, $\alpha = 1.25^\circ$. Premier ordre. Comparaisons de C_p et convergence de calcul



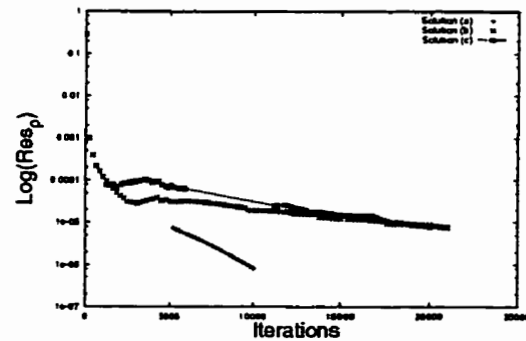
(a) Avec le Maillage final 7.8, VCCV

(b) Avec le Maillage final 7.9, ECCV

(c) Avec le Maillage initial 7.3 (ou final 7.10), CCCV

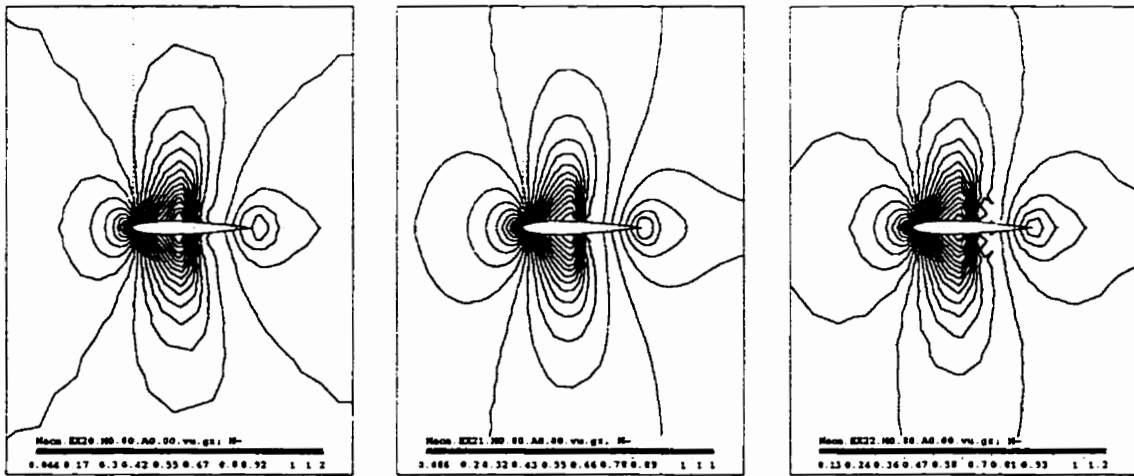


(d) Comparaisons de C_p



(e) Convergence

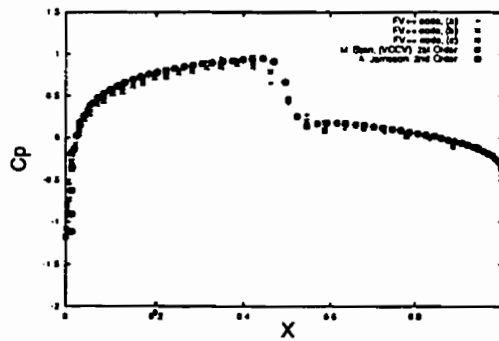
FIGURE 7.10: Lignes iso-Mach pour le Cas-test 7.2 avec le Maillage initial 7.3. $M_\infty = 0.95$, $\alpha = 0^\circ$. Premier ordre. Comparaisons de C_p et convergence de calcul



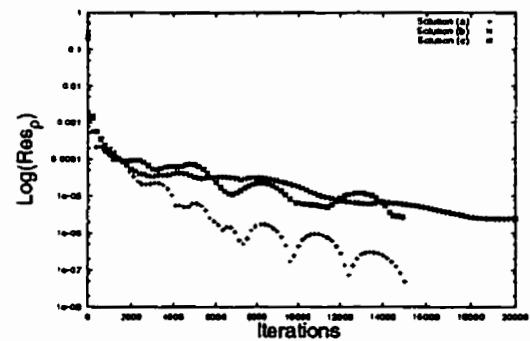
(a) Avec le Maillage final
7.8, VCCV

(b) Avec le Maillage final
7.9, ECCV

(c) Avec le Maillage ini-
tial 7.3 (ou final 7.10),
CCCV

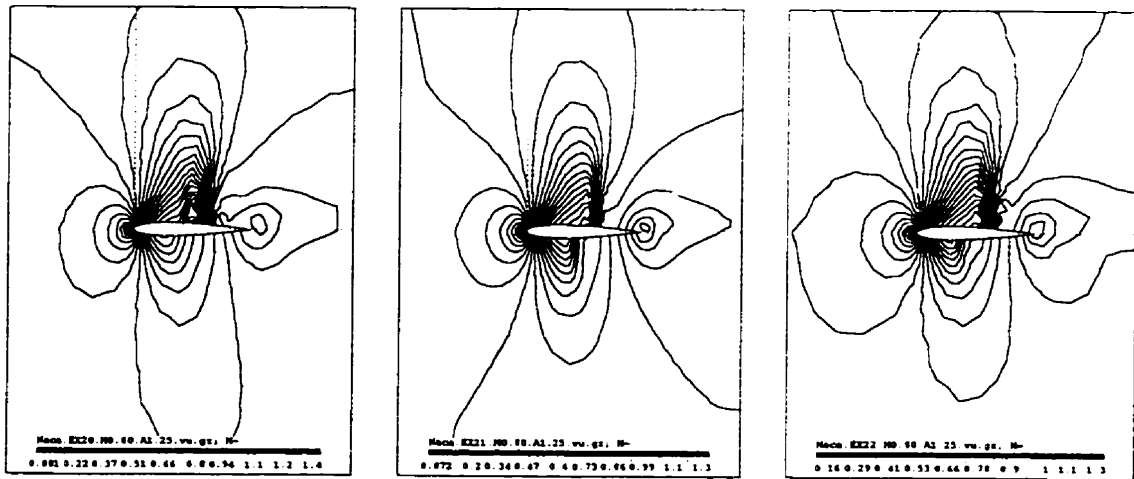


(d) Comparaisons de C_p



(e) Convergence

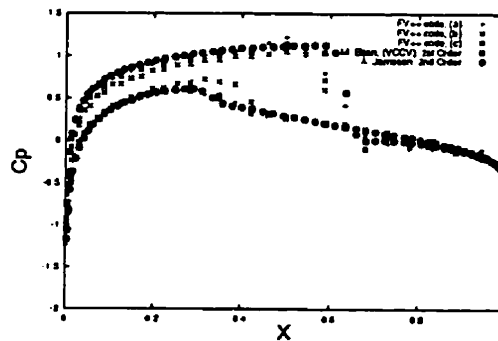
FIGURE 7.11: Lignes iso-Mach pour le Cas-test 7.2 avec le Maillage initial 7.3. $M_\infty = 0.8$, $\alpha = 0^\circ$. Deuxième ordre. Comparaisons de C_p et convergence de calcul



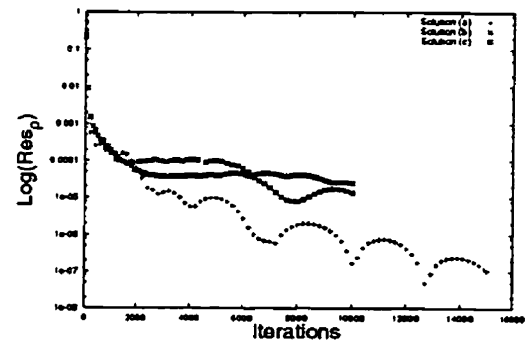
(a) Avec le Maillage final 7.8, VCCV

(b) Avec le Maillage final 7.9, ECCV

(c) Avec le Maillage initial 7.3 (ou final 7.10), CCCV

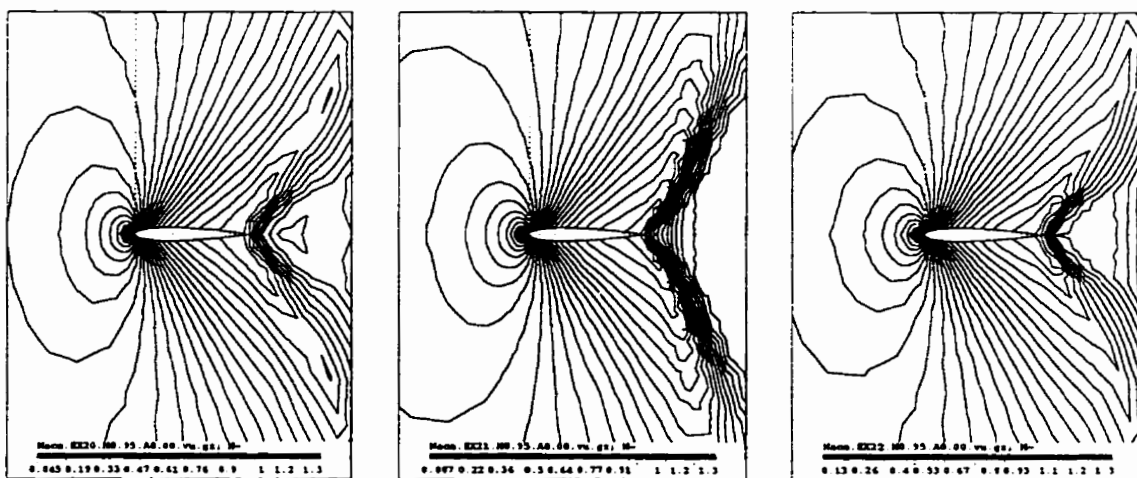


(d) Comparaisons de C_p



(e) Convergence

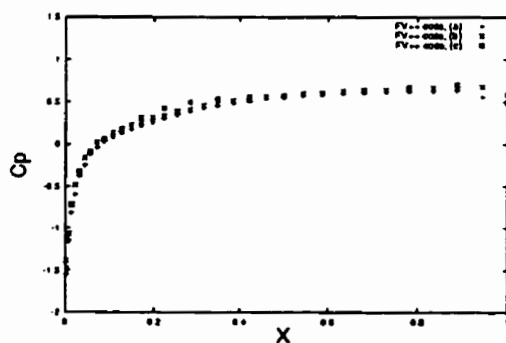
FIGURE 7.12: Lignes iso-Mach pour le Cas-test 7.2 avec le Maillage initial 7.3. $M_\infty = 0.8$, $\alpha = 1.25^\circ$. Deuxième ordre. Comparaisons de C_p et convergence de calcul



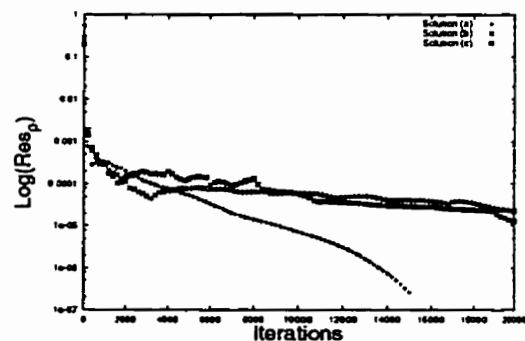
(a) Avec le Maillage final 7.8, VCCV

(b) Avec le Maillage final 7.9, ECCV

(c) Avec le Maillage initial 7.3 (ou final 7.10), CCCV

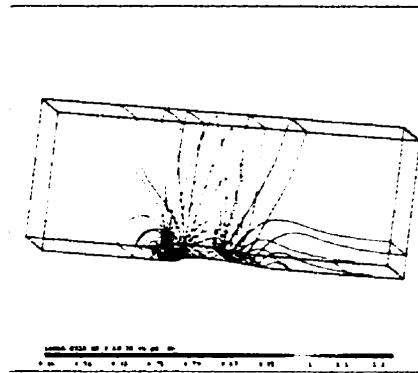


(d) Comparaisons de C_p

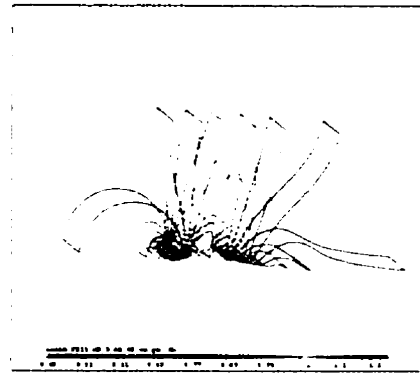


(e) Convergence

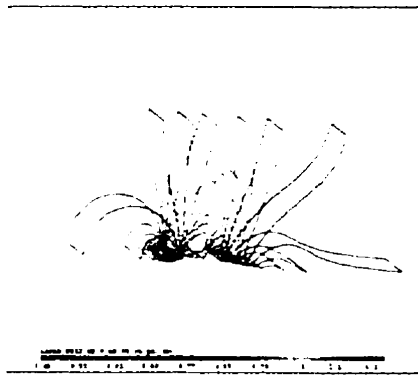
FIGURE 7.13: Lignes iso-Mach pour le Cas-test 7.2 avec le Maillage initial 7.3. $M_\infty = 0.95$, $\alpha = 0^\circ$. Deuxième ordre. Comparaisons de C_p et convergence de calcul



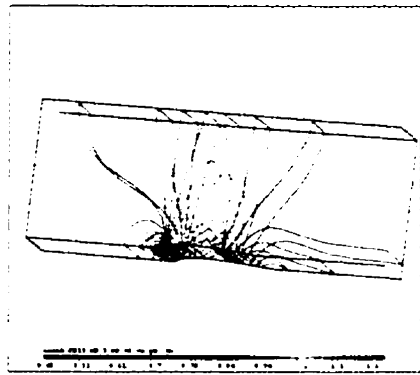
(a) Avec le Maillage final 7.11,
VCCV



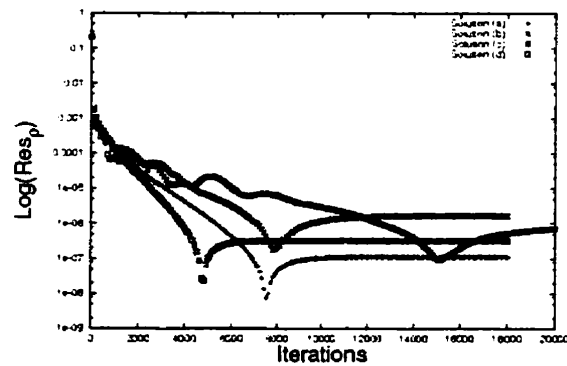
(b) Avec le Maillage final 7.12,
ECCV



(c) Avec le Maillage final 7.12,
FCCV

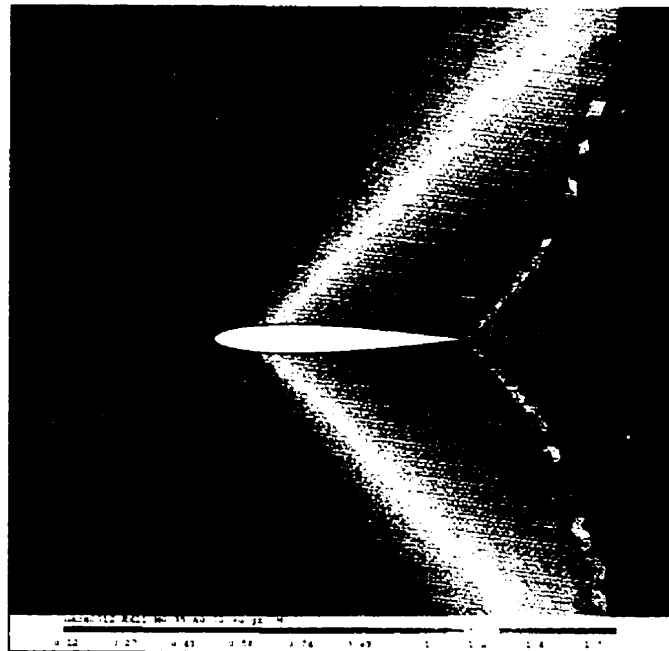


(d) Avec le Maillage initial 7.5
(ou final 7.14), CCCV

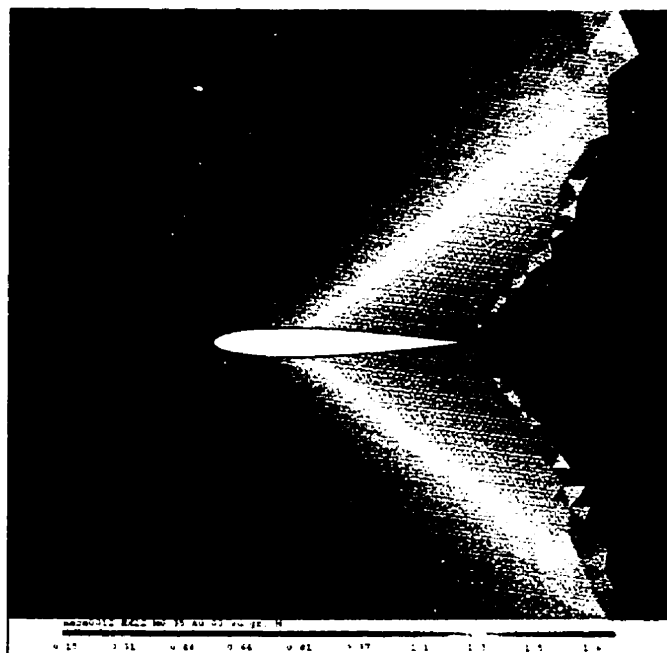


(e) Convergence

FIGURE 7.14: Surfaces iso-Mach pour le Cas-test 7.2 avec le Maillage initial 7.5 (hexaèdres). $M_\infty = 0.8$, $\alpha = 0^\circ$. Premier ordre. Convergence de calcul



(a) 9359 2DECCVs, 18573/145 bords



(b) 6191 2DCCCVs, 9214/145 bords

FIGURE 7.15: *Le "fish tail". Distribution du nombre de Mach. $M_\infty = 0.95, \alpha = 0^\circ$. Maillage initial 7.4*

de meilleurs résultats pour la troisième condition. Cette constatation est observée avec une autre solution avec le Maillage initial 7.4 dans la figure 7.15.

Observation 7.11 Les courbes de comparaison montrent des résultats satisfaisants si on considère que le maillage est grossier.

7.2.3 Écoulement à l'intérieur d'un cylindre

Proposition du problème

Ce cas-test traite de l'étude numérique des phénomènes dans la barrière haute des écoulements supersoniques à fortes incidences¹ et a comme support de validation les données expérimentales *scannées* obtenues par Roe [33] et les études réalisées par Camarero R. [9] à l'aide de la méthode des caractéristiques et celles abordées dans la thèse [18] avec la méthode des volumes finis et le schéma de Roe.

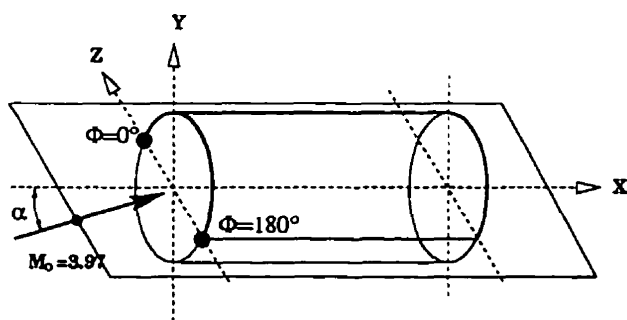


FIGURE 7.16: Description géométrique et conditions du problème

Il s'agit d'un écoulement interne quasi-axial dans un cylindre à haute incidence avec un nombre de $M_\infty = 3.97$. Comme conséquence de cette incidence, l'écoulement interne est décomposé en ondes périodiques de compression et d'expansion de nature tridimensionnelle très complexe,

ayant une directe influence sur la performance et les coefficients de pression et de portance du cylindre. Les études de prédiction numérique de ces phénomènes-là

¹ Ce genre d'écoulements sont d'une importance déterminante dans le processus de simulation de fusées et missiles.

présentent des résultats inattendus dûs aux grands angles d'incidence à l'entrée du cylindre.

La géométrie et les conditions utilisées dans ce cas-test sont montrées par la figure 7.16.

Cette étude paramétrique a été faite pour $\alpha = \{5^\circ, 10^\circ, 15^\circ\}$.

Maillage initial

Ce test a utilisé trois maillages initiaux tridimensionnels :

Maillage initial 7.6 ($40 \times 10 \times 10$) *Maillage 3D structuré, grossier, composé de 4000 hexaèdres, (pas de graphique).*

Maillage initial 7.7 ($70 \times 10 \times 10 \times 2$) *Maillage 3D structuré, moins grossier, composé de 14000 prismes, (pas de graphique).*

Maillage initial 7.8 ($70 \times 10 \times 10 \times 5$) *Maillage 3D non structuré, moins grossier, composé de 35000 tétraèdres, (pas de graphique).*

Maillage final

Maillage final 7.15 (VCCV) *Obtenu du Maillage initial 7.6, avec 4961 éléments et 48000 / 7200 bords, (pas de graphique).*

Maillage final 7.16 (ECCV) *Obtenu du Maillage initial 7.6, avec 13860 éléments et 96000 / 7200 bords, (pas de graphique).*

Maillage final 7.17 (FCCV) *Obtenu du Maillage initial 7.6, avec 12900 éléments et 48000 / 1800 bords, (pas de graphique).*

Maillage final 7.18 (CCCV) Le même que le Maillage initial 7.6, avec 4000 éléments et 11100/1800 bords, (pas de graphique).

Maillage final 7.19 (VCCV) Obtenu du Maillage initial 7.7, avec 8591 éléments et 126000 /12400 bords, (pas de graphique).

Maillage final 7.20 (ECCV) Obtenu du Maillage initial 7.7, avec 31190 éléments et 252000/12400 bords, (pas de graphique).

Maillage final 7.21 (FCCV) Obtenu du Maillage initial 7.7, avec 36600 éléments et 126000/3200 bords, (pas de graphique).

Maillage final 7.22 (CCCV) Même Maillage initial 7.7, avec 14000 éléments et 33400/3200 bords, (pas de graphique).

Maillage final 7.23 (VCCV) Obtenu du Maillage initial 7.8, avec 8591 éléments et 210000 /18000 bords, (pas de graphique).

Maillage final 7.24 (ECCV) Obtenu du Maillage initial 7.8, avec 46590 éléments et 420000/18000 bords, (pas de graphique).

Maillage final 7.25 (FCCV) Obtenu du Maillage initial 7.8, avec 73000 éléments et 210000/6000 bords, (pas de graphique).

Maillage final 7.26 (CCCV) Même que le Maillage initial 7.8, avec 35000 éléments et 67000/6000 bords, (pas de graphique).

Résultats

Tous les calculs ont été réalisés en explicite au premier ordre avec un coefficient de CFL oscillant entre 0.7 et 0.9. Les valeurs plus petites de cette gamme correspondent aux tests avec un angle d'incidence plus grand. Le résidu par rapport à la densité adimensionnelle a atteint le zéro de la machine.

Les résultats, en utilisant toutes les dispositions de la variable selon le tableau 4.1, de cette étude paramétrique en 3D pour chaque condition définie dans le paragraphe précédent sont montrés dans les figures 7.18..7.27 pour les Maillages initiaux 7.6, 7.7 et 7.8. Dans ce contexte les figures 7.18..7.20 montrent la distribution du coefficient de pression C_p (a)..(d) suivi des coupes (e)..(h) et l'historique de la convergence (i) pour les trois angles d'incidence et le Maillage initial 7.8 tétraédrique. La figure 7.21 est une vue polychromatique des iso-surfaces de C_p pour $\alpha = 15^\circ$ dans la développante du cylindre. De même, les figures 7.22..7.27 montrent les coupes des solutions de C_p en utilisant les Maillages initiaux 7.6 et 7.7. Finalement, les distributions du nombre de Mach pour ces conditions sont illustrées par les figures 7.28..figure 7.30. Tous les résultats du coefficient de pression sont comparés avec les données expérimentales [33] sur la surface développée du cylindre en quatre sections critiques à l'entrée, selon $\bar{x} = \sqrt{(M_\infty^2 - 1)}$:

1. **Section I-I** Pour $\bar{x} = 0.25$, $X = 0.96$, premier pic pour la première onde de compression ou d'expansion.
2. **Section II-II** Pour $\bar{x} = 0.50$, $X = 1.92$, *Idem*.
3. **Section III-III** Pour $\bar{x} = 0.75$, $X = 2.88$, *Idem*.
4. **Section IV-IV** Pour $\bar{x} = 2.25$, $X = 8.6445$, second pic pour la même onde.

La figure 7.17 montre ces sections en utilisant une des solutions sur la développante du cylindre.

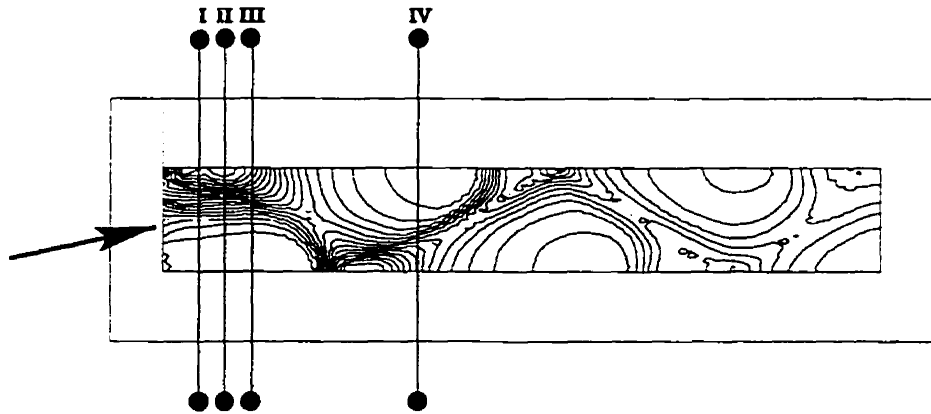


FIGURE 7.17: *Illustration des positions des coupes sur les contours de C_p à la paroi*

Les observations des résultats sur 36 tests avec toutes les mailles de volume de contrôle provenant de tétraèdres, hexaèdres et prismes sont exposées ici.

Observation 7.12 *Principe de conservation respecté.*

Observation 7.13 *La convergence est favorable à VCCV pour des mailles initiales tétraédriques et prismatiques et à CCCV hexaédriques. Elle est défavorable pour FCCV. Celle de ECCV occupe toujours une position intermédiaire.*

Observation 7.14 *Résultats très diffusifs pour les solutions VCCV. Moins diffusifs pour CCCV suivis de FCCV et ECCV.*

Observation 7.15 *En général: Meilleurs résultats pour $\alpha = 5^\circ$, bons pour $\alpha = 10^\circ$, satisfaisants pour $\alpha = 15^\circ$. Les comparaisons faites sur les coupes montrent des résultats acceptables pour CCCV et FCCV (pour les tétraèdres), et ECCV, FCCV suivi de VCCV (pour les prismes et hexaèdres). Résultats exacts pour les dépressions ($[\Phi \rightarrow 180^\circ \wedge \bar{x} = \{0.25, 0.50, 0.75\}] \vee [\Phi \rightarrow 0^\circ \wedge \bar{x} = 2.25]$, onde de compression) et*

en juger au niveau des pics (antipodes de la description précédente). Du point de vue du maillage initial et final ce sont les mailles de volumes de contrôle provenant de tétraèdres qui ont rapporté des meilleurs résultats suivis des ceux obtenus avec des prismes et hexaèdres.

Les CCCV et VCCV se disputent la moins bonne solution pour les hexaèdres.

Toute qualité se détériore lorsqu'on s'éloigne de l'entrée.

Observation 7.16 *Dans les pics, les meilleures solutions sont les CCCV et FCCV pour les volumes de contrôle tétraédriques, et FCCV suivi de ECCV pour les volumes de contrôle hexaédriques et prismatiques. Particulièrement :*

o **Pour $\alpha = 5^\circ$:**

1. À $\bar{x} = 0.25$ tous les résultats sont bons, mais moins bons pour les solutions avec CCCV hexaédrique. Les résultats des mailles tétraédriques sont meilleurs.
2. À $\bar{x} = 0.50$ la qualité de CCCV hexaédrique est améliorée substantiellement.
3. À $\bar{x} = 0.75$ on s'éloigne doucement des données expérimentales : Toutes les solutions avec les dispositions VCCV, ainsi que les solutions avec les mailles prismatiques et hexaédriques, sont un peu au-dessous de celles-ci. Pour ces derniers types de maillages initiaux, le ECCV prend sa place et reflète mieux le phénomène suivis de FCCV et CCCV. Encore ici, le CCCV hexaédrique reprend son problème.
4. À $\bar{x} = 2.25$ aucun type de volume de contrôle, provenant de n'importe quel maillage initial, ne donne pas des résultats coïncidents avec les données.

Par contre les résultats avec les tétraèdres sont les moins mauvais ainsi que ceux avec FCCV pour toutes sortes de mailles initiales.

o Pour $\alpha = 10^\circ$:

1. À $\bar{x} = 0.25$ les résultats maintiennent la même tendance que pour $\alpha = 5^\circ$. Les pires solutions sont obtenues de CCCV hexaédrique, même dans le secteur de dépression. On remarque que dans les données expérimentales une brusque remontée de valeurs C_p est enregistrée² dans l'arc compris entre $78^\circ..82^\circ$.
2. À $\bar{x} = 0.50$ la qualité de CCCV hexaédrique est améliorée mais reste mauvaise en comparaison avec les correspondants ECCV et FCCV. Les résultats de ECCV prismatique montrent une bonne qualité. Les résultats pour les volumes de contrôle tétraédriques sont tous bons sauf pour VCCV.
3. À $\bar{x} = 0.75$ on s'éloigne plus vite des données expérimentales. Les CCCV suivis de FCCV et ECCV tétraédriques (ECCV avec FCCV prismatiques et hexaédriques) sont plus proches de celles-ci. Dans cette section, les VCCV, pour toute forme géométrique, sont les plus éloignés.
4. À $\bar{x} = 2.25$ même que pour $\alpha = 5^\circ$ mais plus intense. Malgré la distance, les CCCV tétraédriques montrent une bonne caractéristique suivis de FCCV et ECCV. Pour les prismes et hexaèdres ce sont les ECCV et FCCV.

o Pour $\alpha = 15^\circ$:

1. À $\bar{x} = 0.25$ les valeurs maximales des données sont atteintes de façon plus satisfaisantes que pour $\alpha = 10^\circ$. Tous les résultats provenant des prismes (figure 7.27) sont meilleurs que ceux avec des tétraèdres ou hexaèdres.

² Phénomène d'hystérésis dont nous parle Roe P. [33] ou une erreur de mesure dans le tunnel aérodynamique?. Si ce secteur était plus grand ou si la section était plus éloignée, il est possible d'accepter une erreur due à la diffusion numérique.

Ceux des hexaèdres sont les moins bons. Pour $\Phi \rightarrow 0^\circ$ les CCCV tétraédriques reflètent ici des valeurs inquiétantes : elles sont plus grandes que les données expérimentales. Par contre, dans cette même région, ces valeurs-là sont beaucoup plus petites pour les CCCV hexaédriques. Pour ces derniers ce sont les ECCV qui conduisent à la meilleure solution.

2. À $\bar{x} = 0.50$ le secteur compris entre $\Phi = 0^\circ..100^\circ$ de façon descendante montre une claire divergence avec les données. Ce sont les volumes de contrôle de tétraèdres qui donnent les meilleurs résultats. Les moins bons sont les hexaèdres. Les ECCV prismatiques se comportent de la meilleure façon parmi toute leur famille de volumes de contrôle.
3. À $\bar{x} = 0.75$ même que pour $\alpha = 10^\circ$ mais plus intense. Les tétraèdres, prismes et hexaèdres échouent, surtout lorsque $\Phi \rightarrow 0^\circ$. Les CCCV suivi de FCCV appartenant aux tétraèdres sont quand même acceptables.
4. À $\bar{x} = 2.25$ il n'a aucune information graphique et numérique annotée dans [33] pour cette section et à cet angle. Toutefois, les tendances du comportement dans ces conditions sont intensifiées à $\bar{x} = 0.75$.

Par conséquent, du point de vue de la géométrie et du stockage, le compromis entre le type de volume de contrôle et leur maillage initial, est crucial dans cette échelle des valeurs.

Observation 7.17 Une explication partielle des phénomènes décrits dans les Observations 7.14..7.16 peut être donnée par la variation du nombre de Mach illustrée par les figures 7.28..7.30 dans une coupe longitudinale du plan d'incidence. L'ordre établi dans ces diagrammes (et ceux des figures 7.18..7.20), coïncidant avec celui du tableau 4.1, est ascendant par rapport à la meilleure solution pour les volumes de contrôle

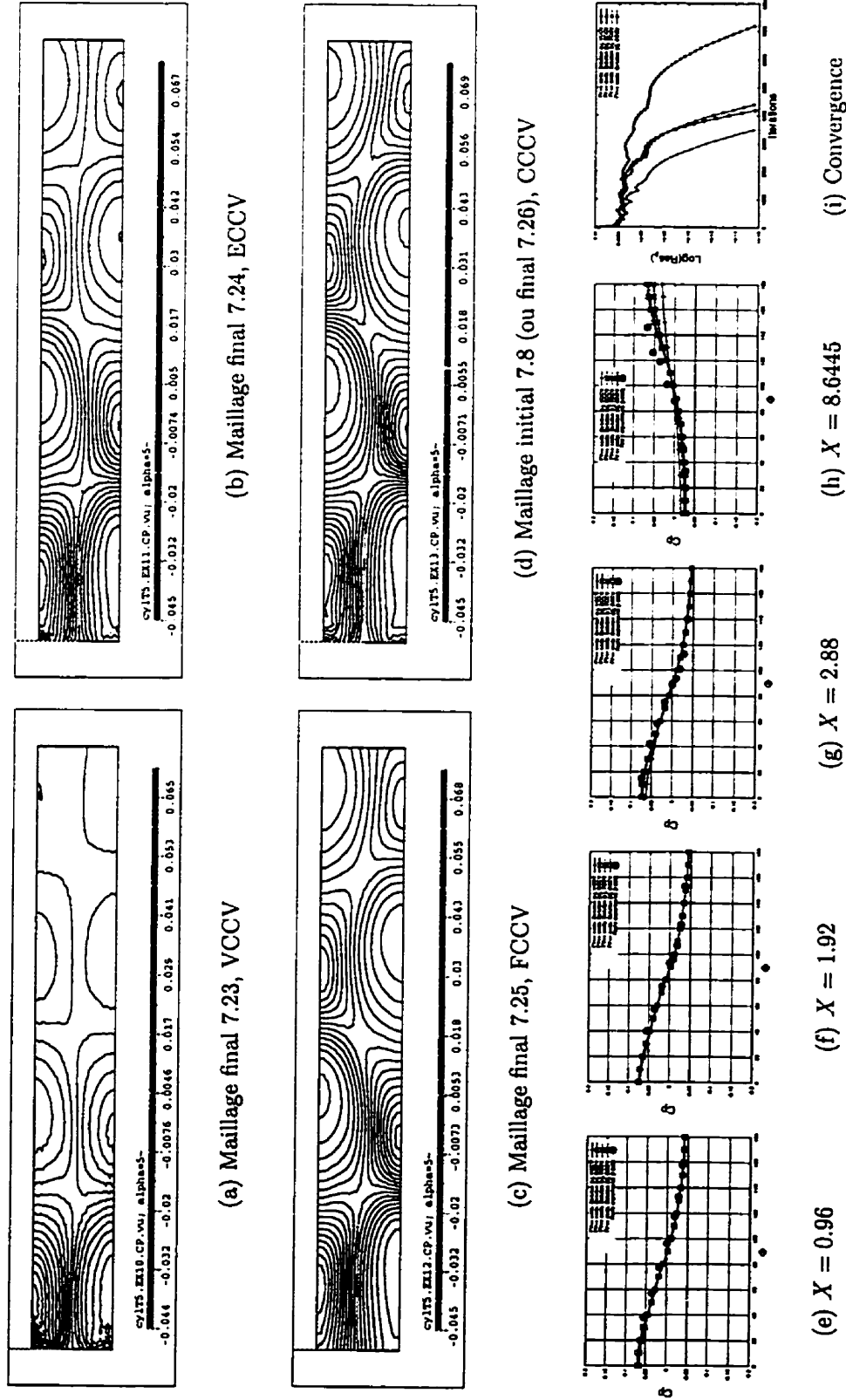
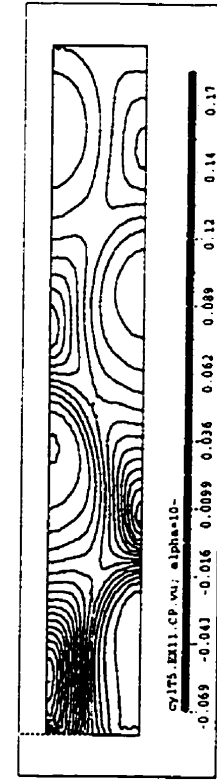
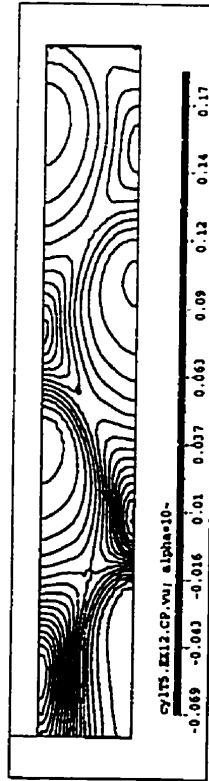


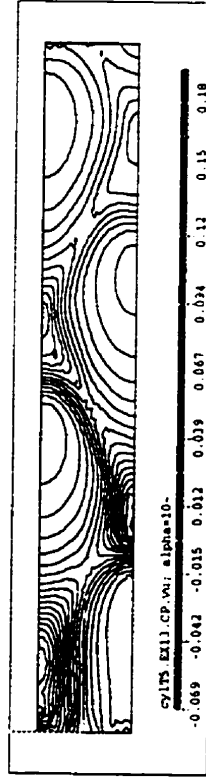
FIGURE 7.18: Résultats du Cas-test 7.9 avec le Maillage initial 7.8. Contours de C_p à la paroi. Coupes et convergence.
 $\alpha = 5^\circ$



(a) Maillage final 7.23, VCCV

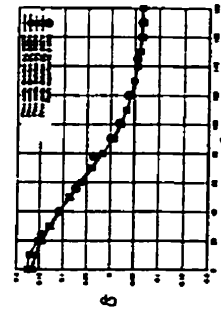
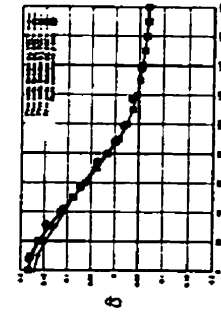
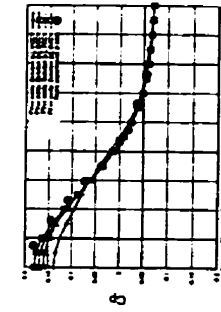
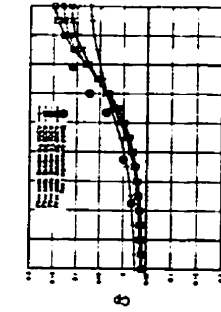
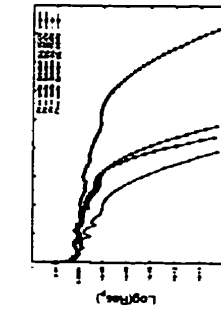


(b) Maillage final 7.24, ECCV



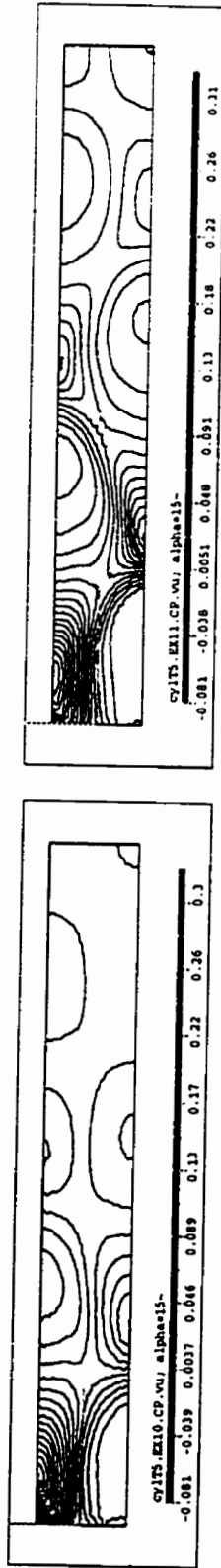
(c) Maillage final 7.25, FCCV

(d) Maillage initial 7.8 (ou final 7.26), CCCV

(e) $X = 0.96$ (f) $X = 1.92$ (g) $X = 2.88$ (h) $X = 8.6445$ 

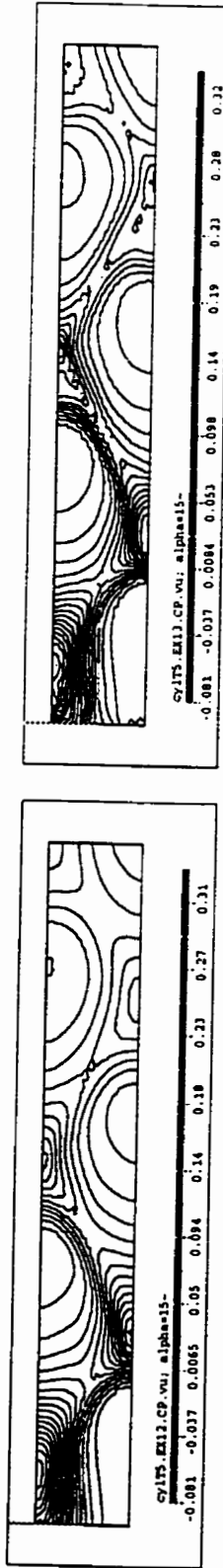
(i) Convergence

FIGURE 7.19: Résultats du Cas-test 7.3 avec le Maillage initial 7.8. Contours de C_p à la paroi. Coupes et convergence.
 $\alpha = 10^\circ$



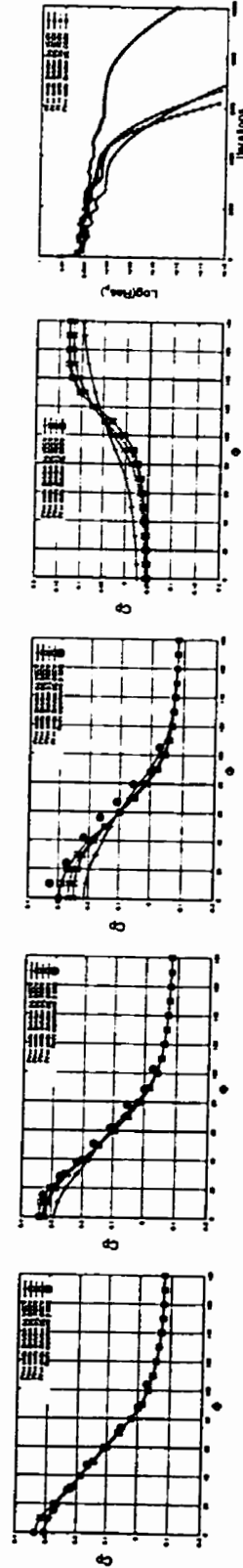
(a) Maillage final 7.23, VCCV

(b) Maillage final 7.24, ECCV



(c) Maillage final 7.25, FCCV

(d) Maillage initial 7.8 (ou final 7.26), CCCV



(e) $X = 0.96$

(f) $X = 1.92$

(g) $X = 2.88$

(h) $X = 8.6445$

(i) Convergence

FIGURE 7.20: Résultats du Cas-test 7.3 avec le Maillage initial 7.8. Contours de C_p à la paroi. Coupes et convergence. $\alpha = 15^\circ$

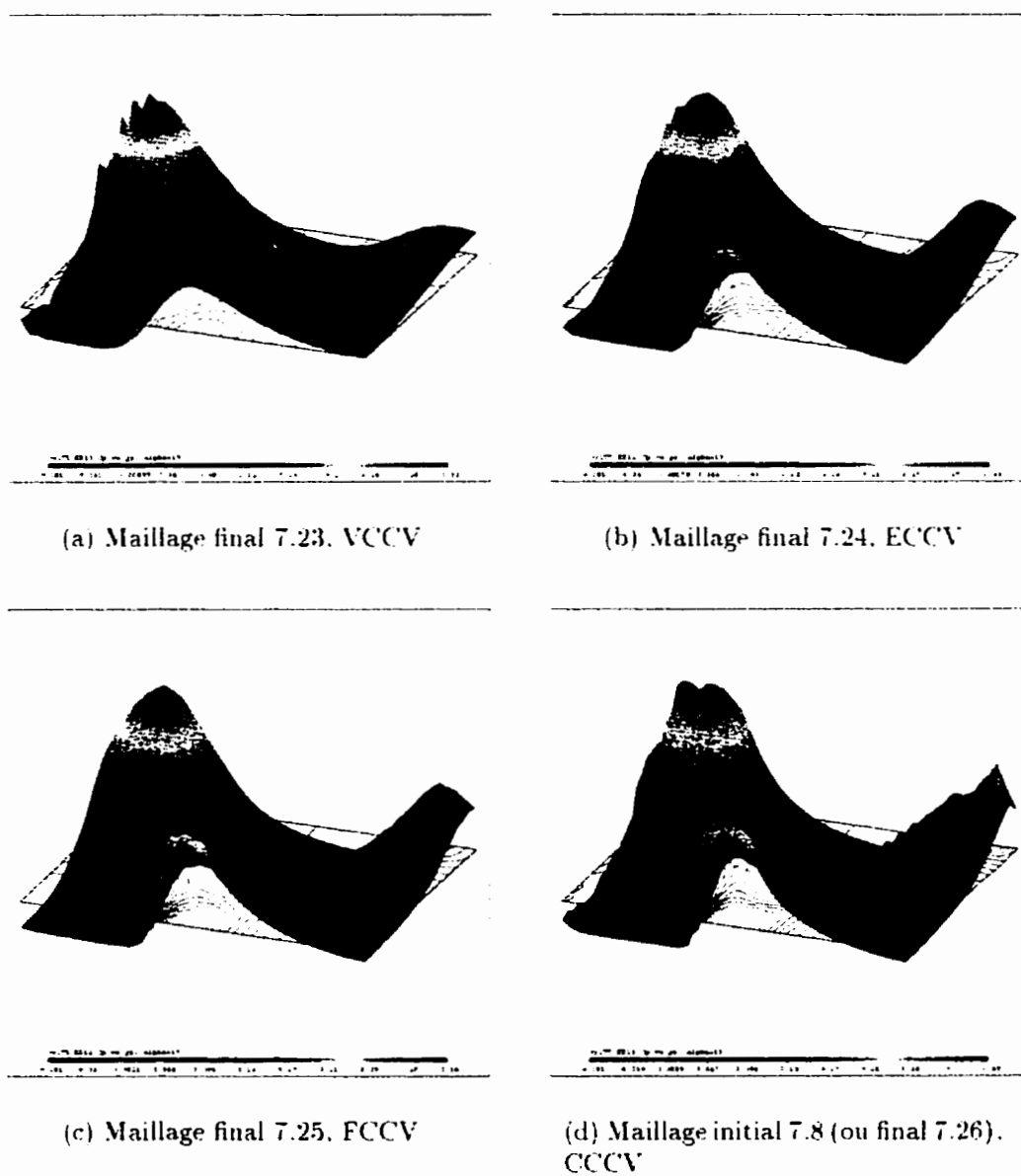


FIGURE 7.21: *Cas-test 7.3. Iso-surfaces de C_p à $\alpha = 15^\circ$ avec le Maillage initial 7.8 (tetraèdres)*

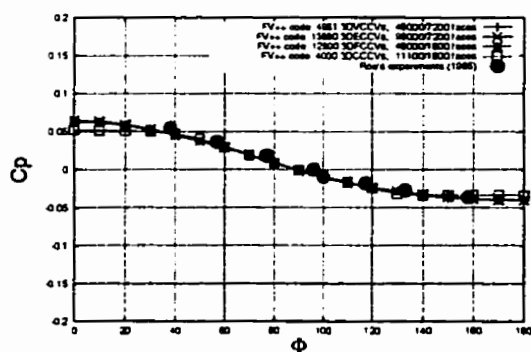
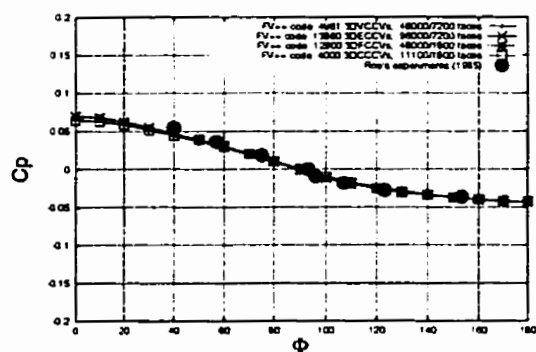
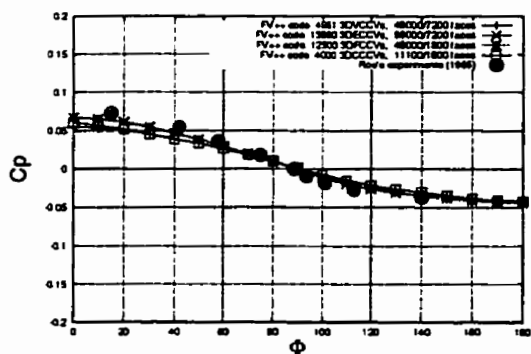
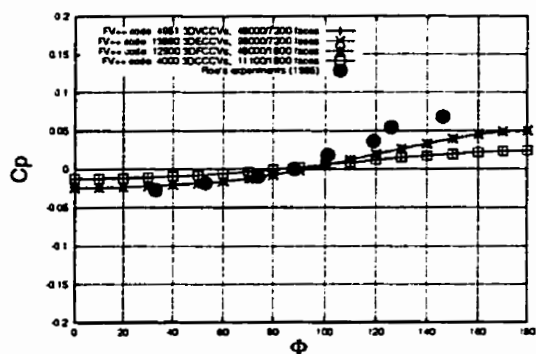
(a) $X = 0.96$ (b) $X = 1.92$ (c) $X = 2.88$ (d) $X = 8.6445$

FIGURE 7.22: Résultats du Cas-test 7.3 avec le Maillage initial 7.6 (hexaèdres). Variation du C_p . Coupes. $\alpha = 5^\circ$

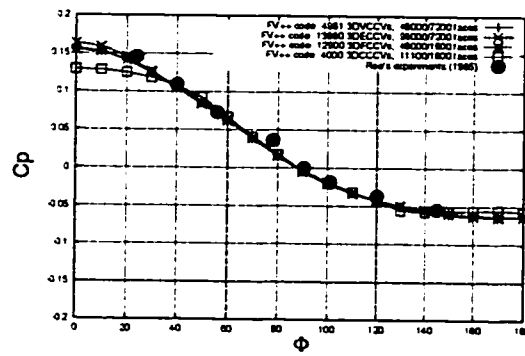
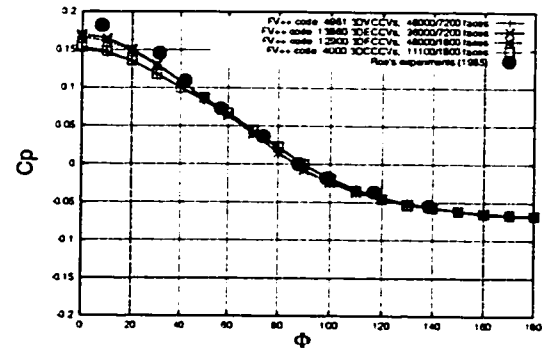
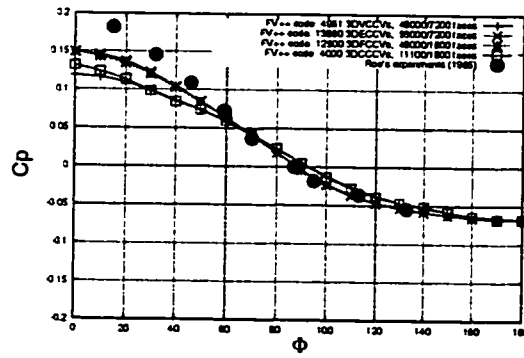
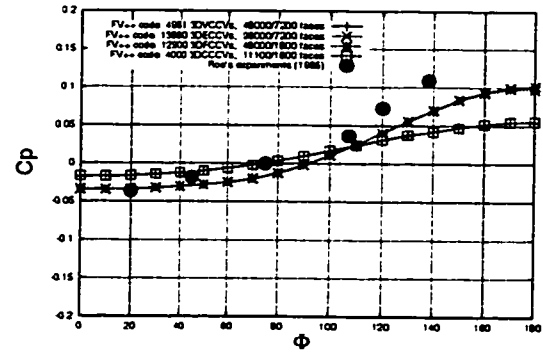
(a) $X = 0.96$ (b) $X = 1.92$ (c) $X = 2.88$ (d) $X = 8.6445$

FIGURE 7.23: Résultats du Cas-test 7.3 avec le Maillage initial 7.6 (hexaèdres). Variation du C_p Coupes. $\alpha = 10^\circ$

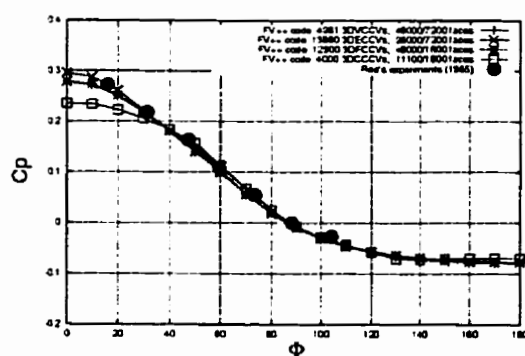
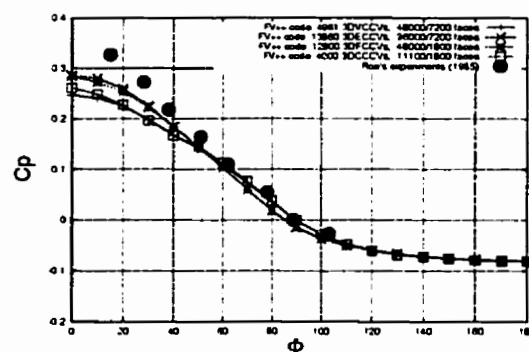
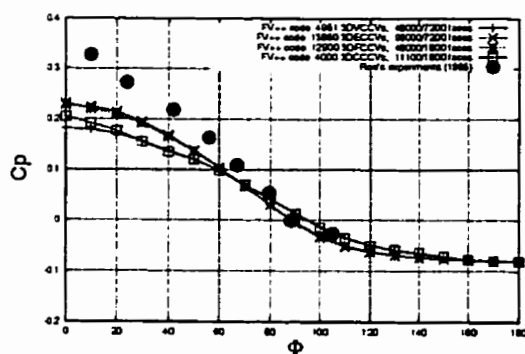
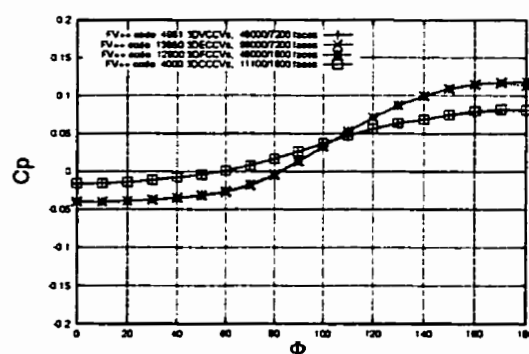
(a) $X = 0.96$ (b) $X = 1.92$ (c) $X = 2.88$ (d) $X = 8.6445$

FIGURE 7.24: Résultats du Cas-test 7.3 avec le Maillage initial 7.6 (hexaèdres). Variation du C_p . Coupes. $\alpha = 15^\circ$

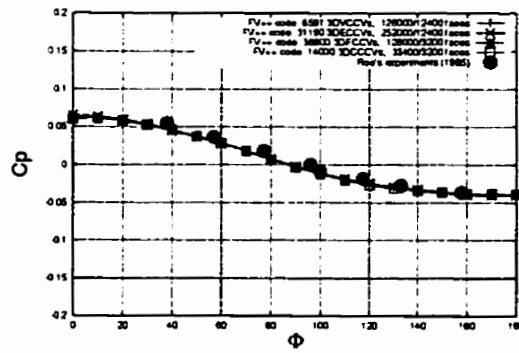
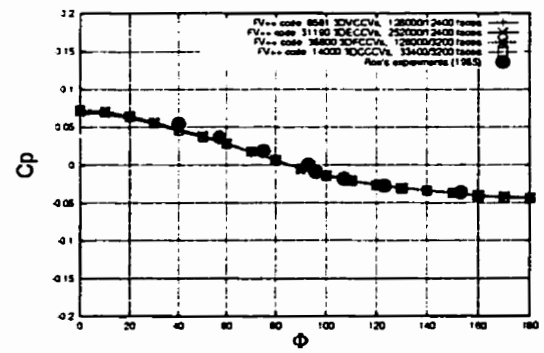
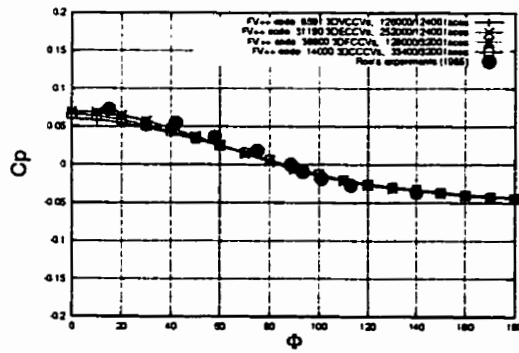
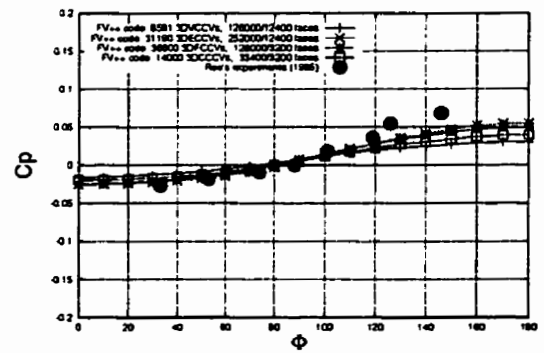
(a) $X = 0.96$ (b) $X = 1.92$ (c) $X = 2.88$ (d) $X = 8.6445$

FIGURE 7.25: Résultats du Cas-test 7.3 avec le Maillage initial 7.7 (prismes). Variation du C_p . Coupes. $\alpha = 5^\circ$

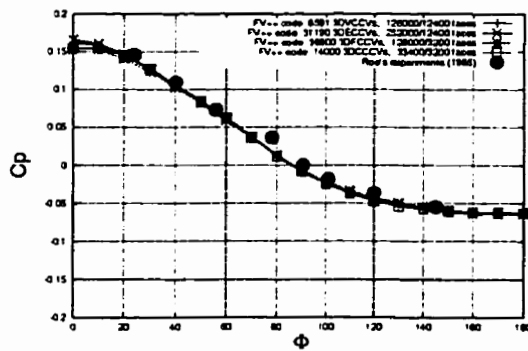
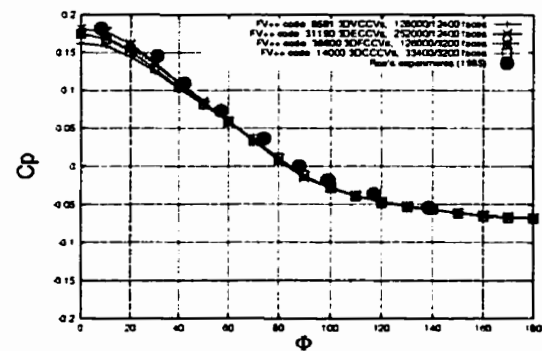
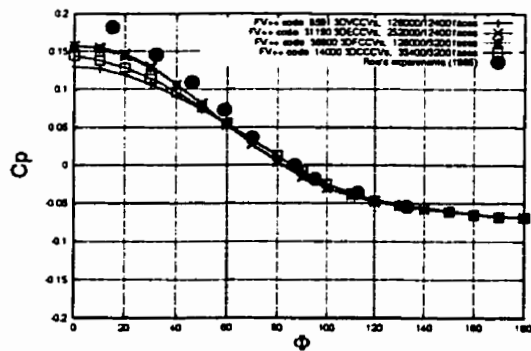
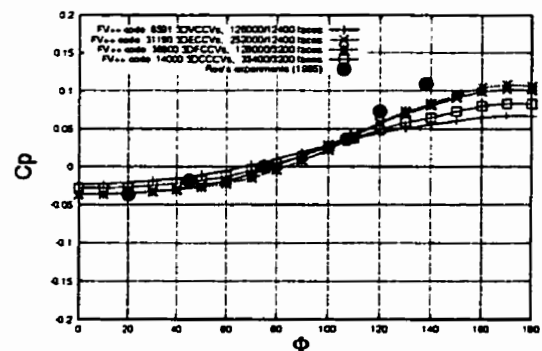
(a) $X = 0.96$ (b) $X = 1.92$ (c) $X = 2.88$ (d) $X = 8.6445$

FIGURE 7.26: Résultats du Cas-test 7.3 avec le Maillage initial 7.7 (prismes). Variation du C_p . Coupes. $\alpha = 10^\circ$

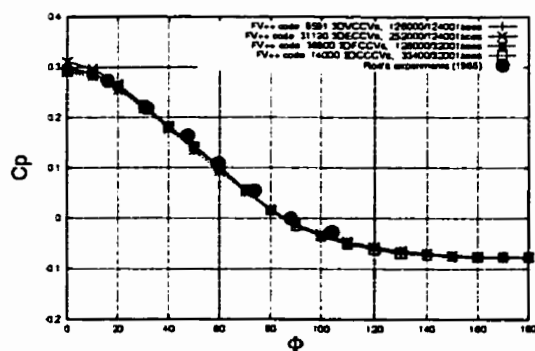
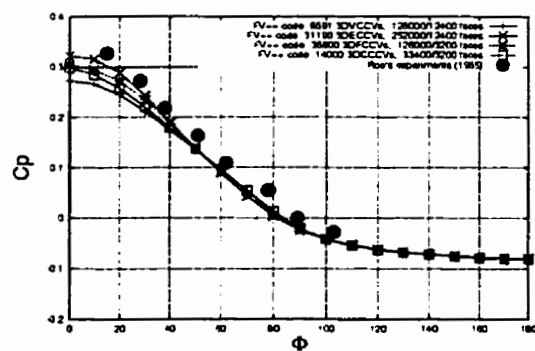
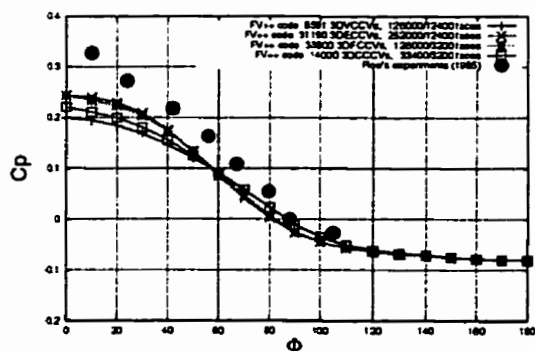
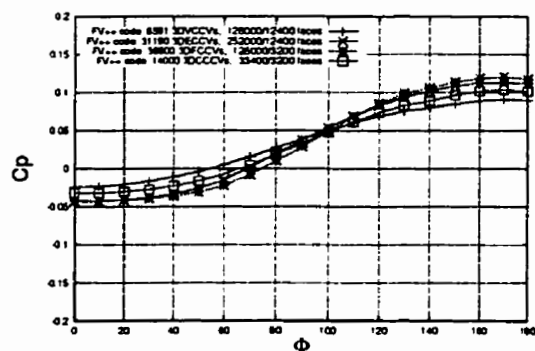
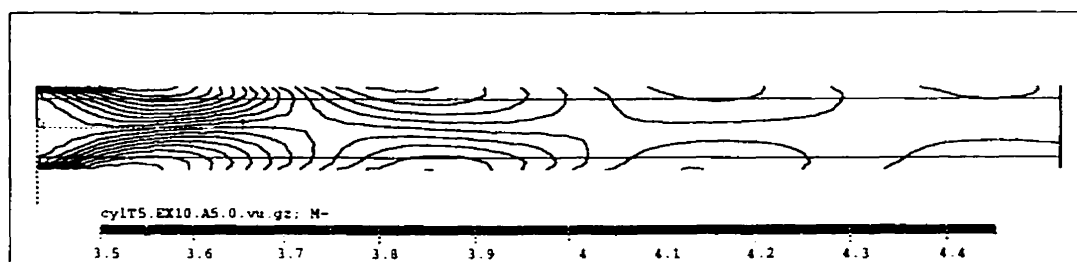
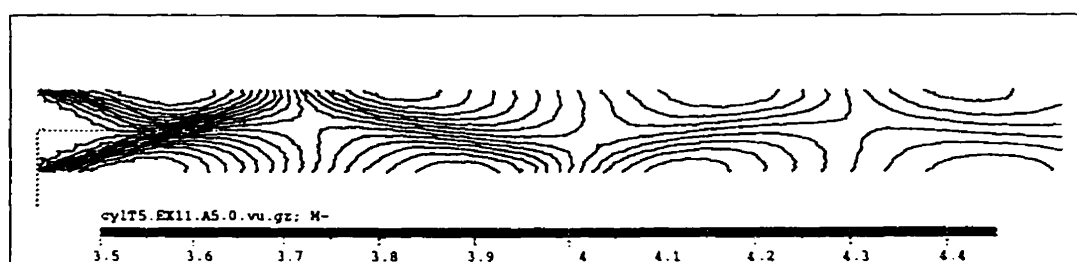
(a) $X = 0.96$ (b) $X = 1.92$ (c) $X = 2.88$ (d) $X = 8.6445$

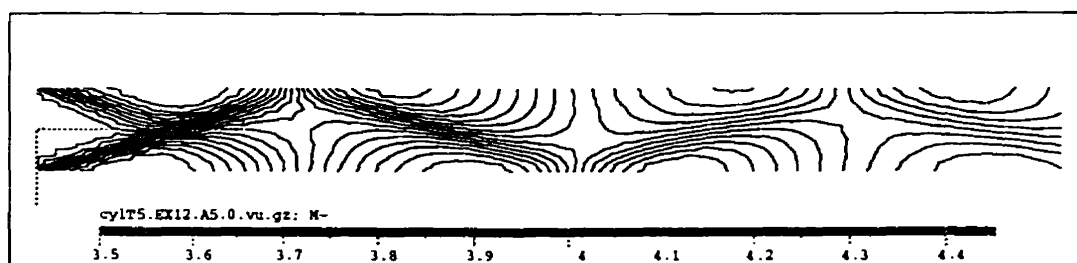
FIGURE 7.27: Résultats du Cas-test 7.3 avec le Maillage initial 7.7 (prismes). Variation du C_p . Coupes. $\alpha = 15^\circ$



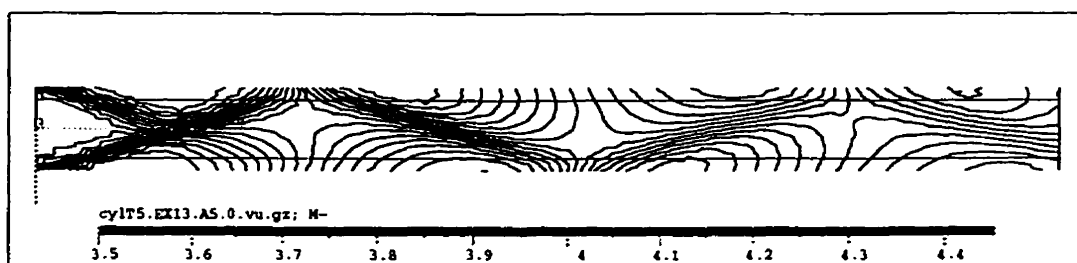
(a) Maillage final 7.23, VCCV



(b) Maillage final 7.24, ECCV

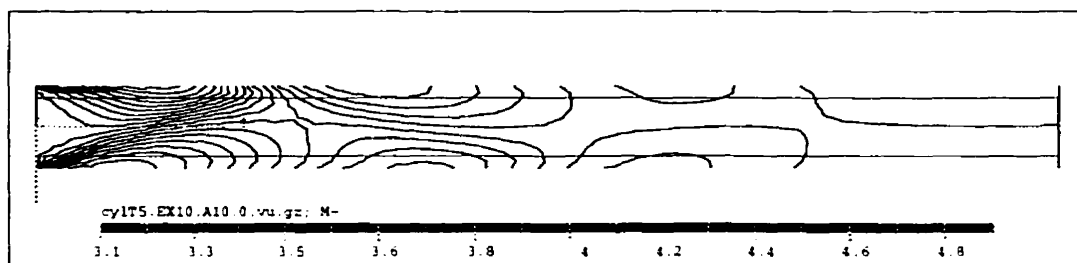


(c) Maillage final 7.25, FCCV

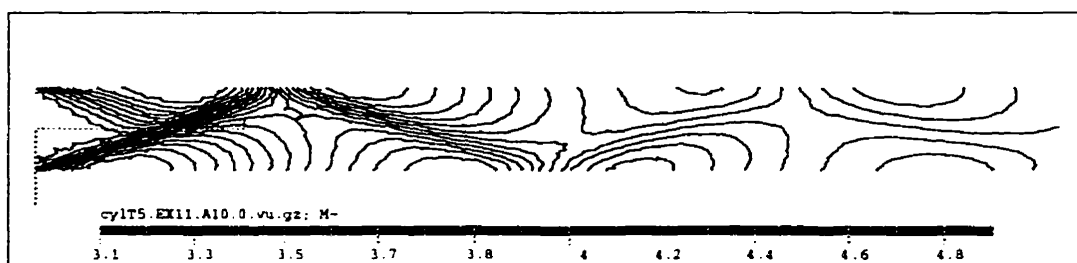


(d) Maillage initial 7.8 (ou final 7.26), CCCV

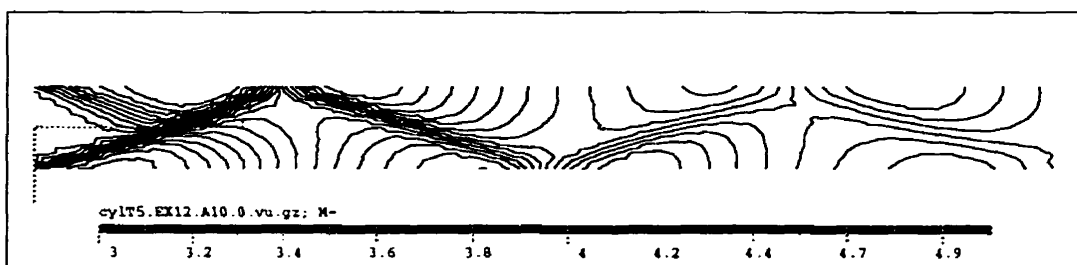
FIGURE 7.28: Résultats du Cas-test 7.3 avec le Maillage initial 7.8. Contours de Mach dans le plan d'incidence. $\alpha = 5^\circ$



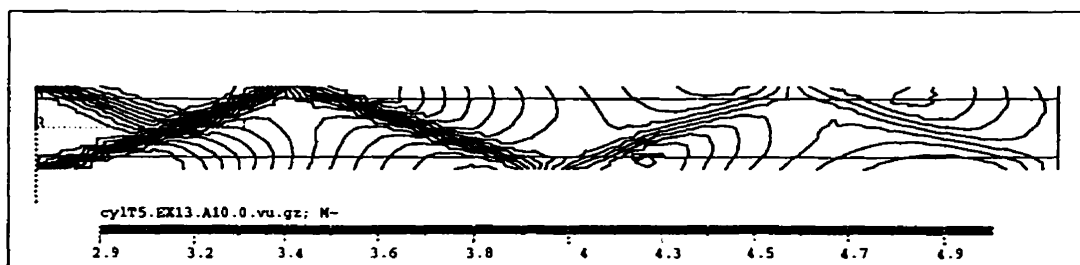
(a) Maillage final 7.23, VCCV



(b) Maillage final 7.24, ECCV

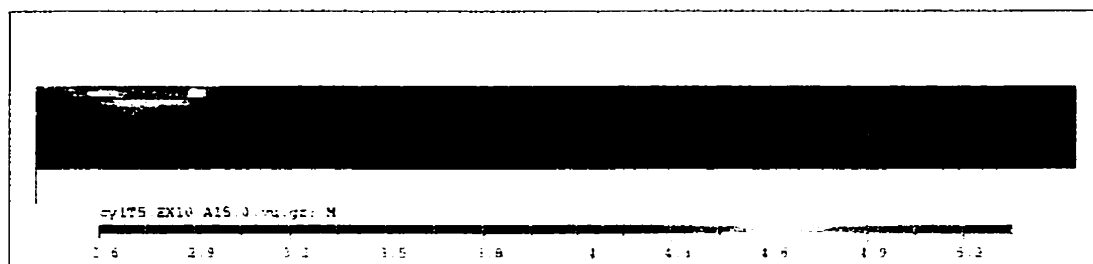


(c) Maillage final 7.25, FCCV

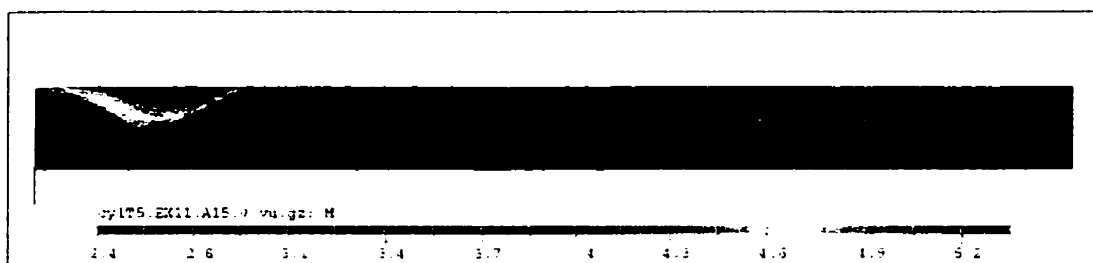


(d) Maillage initial 7.8 (ou final 7.26), CCCV

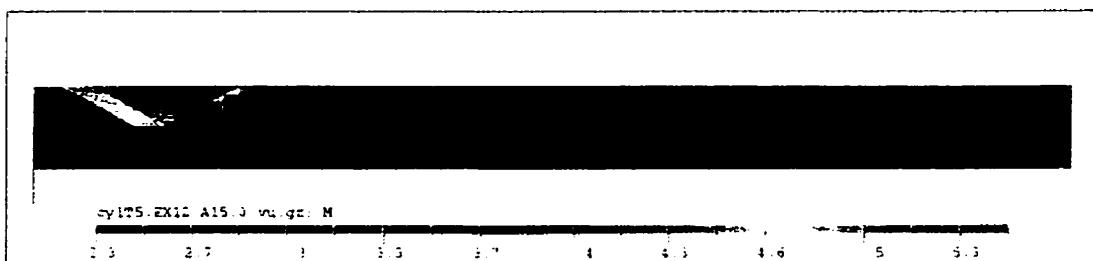
FIGURE 7.29: Résultats du Cas-test 7.3 avec le Maillage initial 7.8. Contours de Mach dans le plan d'incidence. $\alpha = 10^\circ$



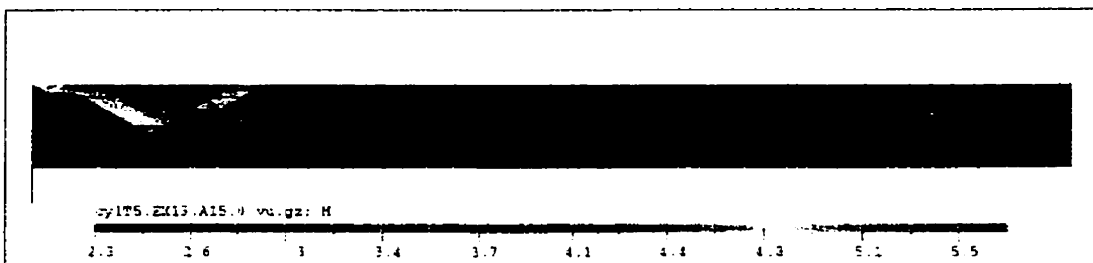
(a) Maillage final 7.23, VCCV



(b) Maillage final 7.24, ECCV



(c) Maillage final 7.25, FCCV



(d) Maillage initial 7.8 (ou final 7.26), CCCV

FIGURE 7.30: Résultats du Cas-test 7.3 avec le Maillage initial 7.8. Distribution du nombre de Mach dans le plan d'incidence. $\alpha = 15^\circ$

des tétraèdres. Ici, on note clairement la bonne définition des champs de distribution dans les zones de compression/expansion pour les CCCV et FCCV.

Les mêmes tendances dans la solution ont été observées avec un maillage initial plus fin de $50 \times 14 \times 14 \times 5$ avec 49000 tétraèdres, et les autres mailles de volume de contrôle qui dérivait de celui-ci.

Pour conclure avec l'analyse de ces résultats, ce sont les CCCV tétraédriques qui ont un grand avantage du point de vue de l'économie de temps de calcul et la précision requise. Les plus économiques et les moins précis sont toujours les VCCV et CCCV hexaédriques.

7.2.4 Provocation d'ondes de choc dans une tuyère convergente-divergente

Introduction

Les ondes de choc surgissant dans la section divergente d'une tuyère sont provoquées en général en conditions hors design par un rapport de pressions entre la sortie et la chambre. Du point de vue aérodynamique, ces phénomènes ont une influence directe sur la performance de la tuyère et, du point de vue structurelle, sur sa vie utile.

Les tuyères sont utilisées dans le domaine de la propulsion, où on profite de l'énergie cinétique générée par les gaz de combustion, dans le design de tunnels de hautes vitesses ou d'éjecteurs, etc.

Les prédictions numériques dans ce domaine de la Physique [46, 10] sont devenues un défi car il ne s'agit pas seulement des prédictions des discontinuités de

nature tridimensionnelle et instable, mais aussi d'avoir un aperçu qualitatif et quantitatif des phénomènes encore plus complexes comme la propagation des ondes hors de la tuyère, l'interaction de la couche limite et des ondes de choc ou les problèmes associés à l'aspiration.

Les ondes de choc apparaissant dans les conditions réelles sont loin d'avoir un comportement prédictible par la théorie unidimensionnelle. La position et concavité de cette surface de brusque variation de l'entropie est fonction de la géométrie et conditions de chambre.

Proposition du problème

On se propose l'étude qualitative³ des paramètres aérothermiques à l'intérieur d'un canal convergent-divergent en considérant un fluide idéal. On résout les équations d'Euler en utilisant le schéma de Roe et tous les types de volume de contrôle du tableau 4.1 provenant des différentes mailles initiales en 2D et 3D avec deux conditions de chambre imposées par les rapports de pression à la sortie :

1. À $\frac{p_s}{p_o} = 0.25$,
2. À $\frac{p_s}{p_o} = 0.40$.

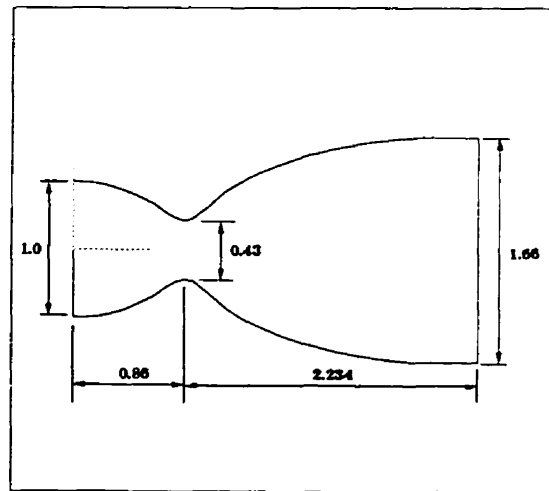


FIGURE 7.31: Géométrie du domaine de calcul

³ La condition nécessaire. L'analyse quantitative devient superflue si la condition nécessaire n'est pas remplie et pour cela il faut avoir des données expérimentales.

Avec ces deux conditions, deux chocs d'intensité différente apparaîtront dans la partie divergente. La première condition provoquera un choc proche de la sortie et, pour la deuxième condition, le choc sera proche du col.

Le profil utilisé, basé sur les suggestions décrites dans [5], a trois sections : une partie convergente sinusoïdale, le col avec un rayon externe calculé et une partie divergente parabolique. La géométrie de ce profil avec les dimensions principales est exposée dans la figure 7.31.

Maillage initial

Dans ce calcul, on a utilisé trois types de mailles initiales :

Maillage initial 7.9 (50×10) *Maillage 2D structuré, très grossier, composé de quadrilatères, tel qu'illustré par la figure 7.32(c).*

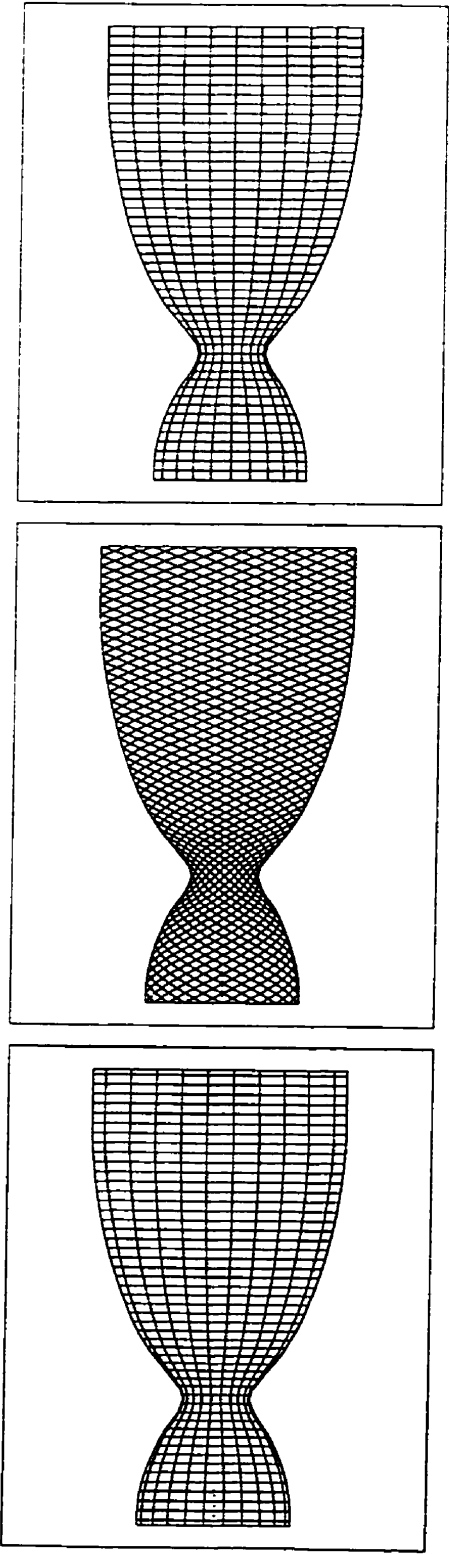
Maillage initial 7.10 ($50 \times 10 \times 2$) *Maillage 2D non structuré, très grossier, composé de triangles, tel qu'illustré par la figure 7.33(c).*

Maillage initial 7.11 ($50 \times 10 \times 6 \times 5$) *Maillage 3D non structuré, composé de tétraèdres, (pas de graphique).*

On remarque que les maillages initiaux 7.10 et 7.11 sont construits à partir du Maillage initial 7.9.

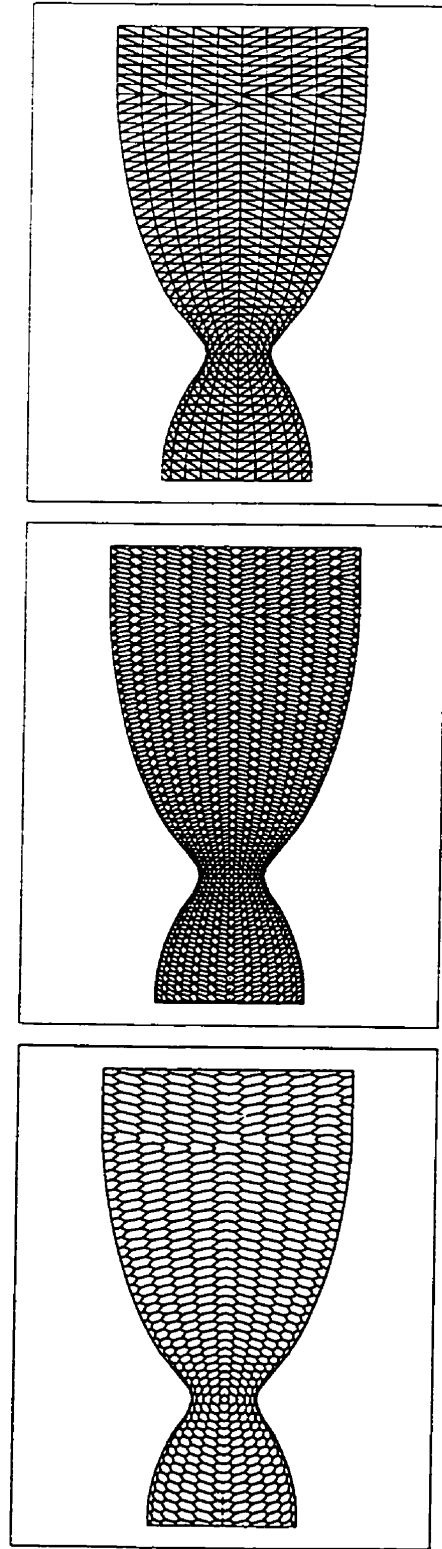
Maillage final

Les maillages finaux utilisés pour ce cas-test sont :



(a) Le Maillage final 7.27, VCCV (b) Le Maillage final 7.28, ECCV (c) Le Maillage initial 7.9 (ou final 7.29)

FIGURE 7.32: *Domaine de calcul et les mailles finales, provenant du Maillage initial 7.9, utilisées dans le Cas-test 7.4*



(a) Le Maillage final 7.30, VCCV

(b) Le Maillage final 7.31, ECCV

(c) Le Maillage initial 7.10 (ou final 7.32)

FIGURE 7.33: *Domaine de calcul et les mailles finales, provenant du Maillage initial 7.10, utilisées dans le Cas-test 7.4*

◦ *en 2D :*

Maillage final 7.27 (VCCV) Obtenu du Maillage initial 7.9, avec 561 éléments et 2000/240 bords, tel que montré par la figure 7.32(a).

Maillage final 7.28 (ECCV) Obtenu du Maillage initial 7.9, avec 1060 éléments et 2000/120 bords, tel qu'illustré par la figure 7.32(b).

Maillage final 7.29 (CCCV) Même que le Maillage initial 7.10, avec 500 éléments et 940/120 bords, tel que montré par la figure 7.32(c).

Maillage final 7.30 (VCCV) Obtenu du Maillage initial 7.10, avec 561 éléments et 3000/240 bords, tel que montré par la figure 7.33(a).

Maillage final 7.31 (ECCV) Obtenu du Maillage initial 7.10, avec 1560 éléments et 3000/120 bords, tel qu'illustré par la figure 7.33(b).

Maillage final 7.32 (CCCV) Même que le Maillage initial 7.10, avec 1000 éléments et 1440/120 bords, tel que montré par la figure 7.33(c).

◦ *en 3D :*

Maillage final 7.33 (VCCV) Obtenu du Maillage initial 7.11, avec 3927 éléments et 90000/10320 bords, (pas de graphique).

Maillage final 7.34 (ECCV) Obtenu du Maillage initial 7.11, avec 20646 éléments et 180000/10320 bords, (pas de graphique).

Maillage final 7.35 (FCCV) Obtenu du Maillage initial 7.11, avec 31720 éléments et 90000/3440 bords, (pas de graphique).

Maillage final 7.36 (CCCV) Même que le Maillage initial 7.11, avec 15000 éléments et 28280/3440 bords, (pas de graphique).

Résultats

Les résultats du cas-test que l'on présente ici ne sont qu'une partie d'une série de tests en 2D et 3D faits avec la même tuyère et les mêmes conditions de chambre avec des mailles initiales moins grossières et diverses.

Tous les calculs ont été faits en explicite au premier ordre avec un $CFL = 0.9..0.95$. Les figures 7.34..7.39 montrent la distribution du nombre de Mach obtenue avec les Mailles finales 7.27.. 7.36 accompagnée de la convergence respective. La meilleure convergence a atteint 10^{-16} , la pire 10^{-3} , comme on peut voir des figures 7.34(d)..7.37(d), 7.38(e) et 7.39(e).

Des résultats de cette étude qualitative, pour les deux conditions de chambre, on tire les observations suivantes :

- ***Pour le Maillage initial 7.9*** (quadrilatères) :

Observation 7.18 *Convergence cyclique variée. Elle est monotonique et rapide pour ECCV atteignant 10^{-6} , plus lente et chaotique pour le VCCV suivi de CCCV.*

Observation 7.19 *Résultats très diffusifs et absurdes pour VCCV et CCCV, acceptables pour ECCV. La position de la discontinuité avec ce dernier schéma est bien définie dans la partie divergente.*

- ***Pour le Maillage initial 7.10*** (triangles) :

Observation 7.20 *Convergence cyclique. Elle est très rapide pour VCCV atteignant 10^{-16} pour la première condition et 10^{-9} pour la deuxième. Monotonique pour ECCV. Très chaotique pour CCCV.*

Observation 7.21 *Résultats très diffusifs et mauvais pour CCCV pour la première condition de chambre et absurdes pour la deuxième. Résultats acceptables pour ECCV et VCCV : La symétrie et la position du choc sont mieux définies.*

- *Pour le Maillage initial 7.11 (tetraèdres). Écoulement plan :*

Observation 7.22 *Convergence cyclique. Elle est plus rapide pour VCCV. atteignant 10^{-16} pour la première condition et 10^{-14} pour la deuxième, suite de ECCV et FCCV. Très chaotique pour CCCV.*

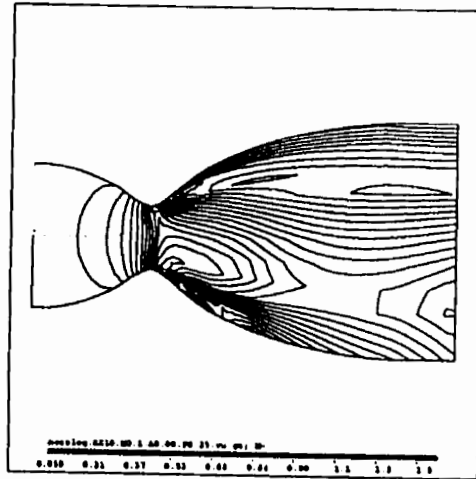
Observation 7.23 *Quant à la diffusion numérique, on observe le même comportement que pour l'Observation 7.21 : Des résultats absurdes pour CCCV pour la deuxième condition. Ici on ajoute la capture mieux définie du choc produite par FCCV pour les deux conditions.*

Il est à noter que parmi les deux conditions de chambre décrites ci-dessus la deuxième est très exigeante dû au fait que la prédiction demandée est assujettie à un rapport de pression proche du rapport critique, forçant la position du choc à se déplacer vers le col. La première condition, pourtant, est mieux assurée même si la convergence est cyclique et chaotique dans certains cas⁴.

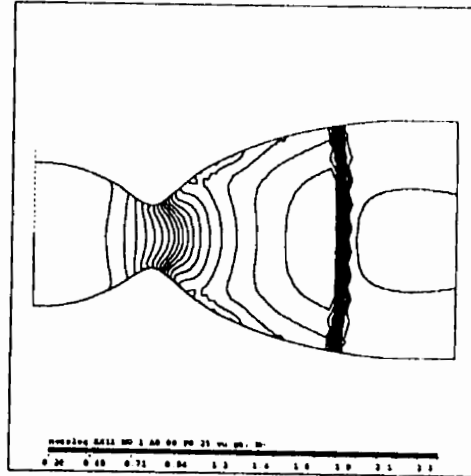
Le but de ce cas-test est de mettre à plat ce type de résultats pour pouvoir porter un jugement objectif quant à l'emploi des types de volumes de contrôle et du schéma numérique pour la résolution de problèmes avec une nature semblable.

Les tests avec ce type de problèmes et les mêmes conditions, mais différentes géométries, comportent des résultats polémiques et contradictoires, ne permettant pas

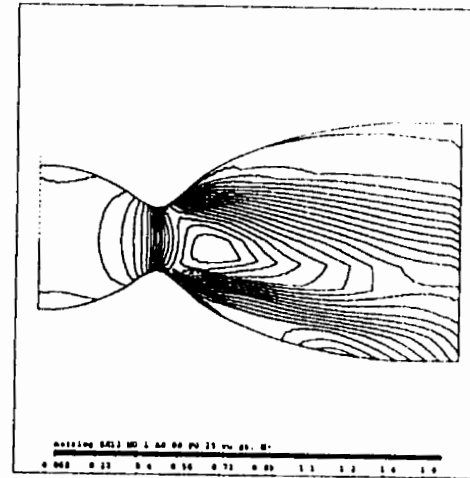
⁴ Souvent, lorsqu'on fait ce genre de calculs on utilise soit le CCCV ou VCCV avec la première condition (mieux assurée), pas avec la deuxième.



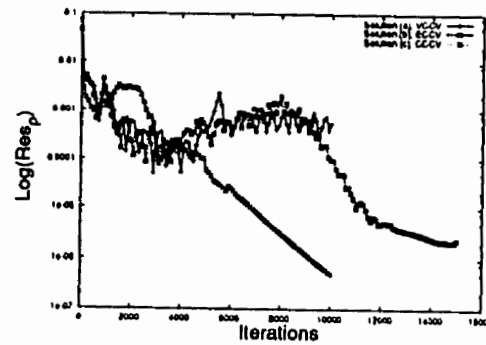
(a) Avec le Maillage final 7.27, VCCV



(b) Avec le Maillage final 7.28, ECCV

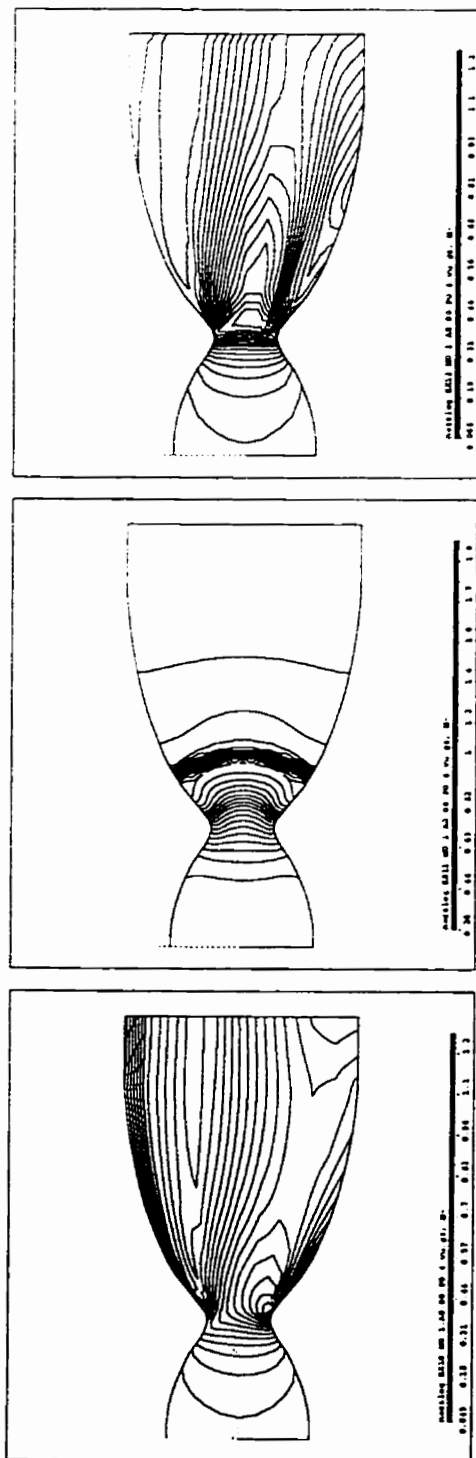


(c) Avec le Maillage initial 7.9 (ou final 7.29), CCCV



(d) Convergence

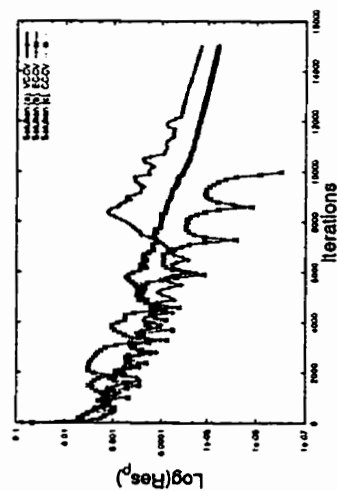
FIGURE 7.34: Distribution du nombre de Mach pour le Cas-test 7.4 avec le Maillage initial 7.9 (quadrilatères). $\frac{p_x}{p_o} = 0.25$. Convergence de calcul



(a) Avec le Maillage final 7.27, VCCV

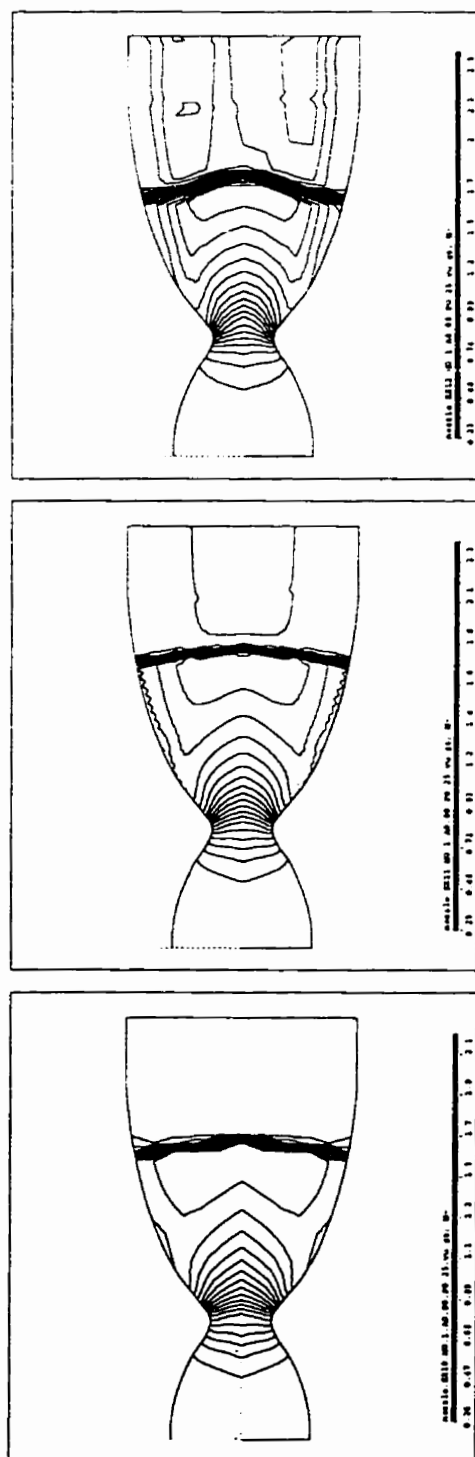
(b) Avec le Maillage final 7.28, ECCV

(c) Avec le Maillage initial 7.9 (ou final 7.29), CCCV



(d) Convergence

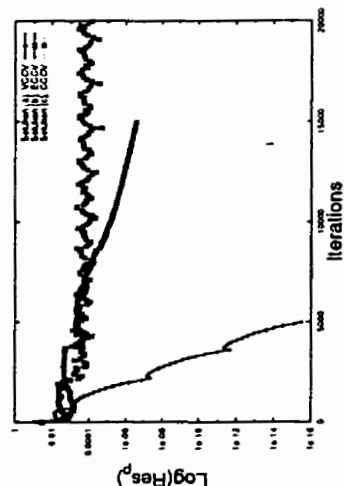
FIGURE 7.35: Distribution du nombre de Mach pour le Cas-test 7.4 avec le Maillage initial 7.9 (quadrilatères). $\frac{P_2}{p_0} = 0.4$. Convergence de calcul



(a) Avec le Maillage final 7.30, VCCV

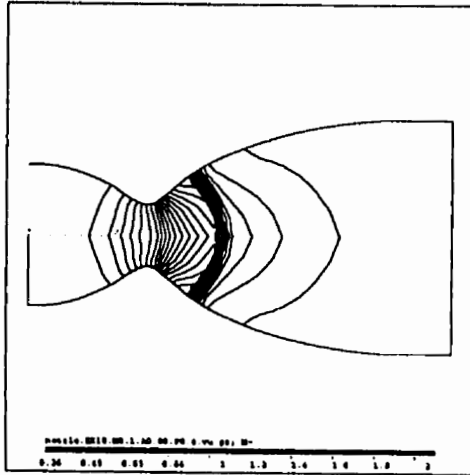
(b) Avec le Maillage final 7.31, ECCV

(c) Avec le Maillage initial 7.10 (ou final 7.32), CCCV

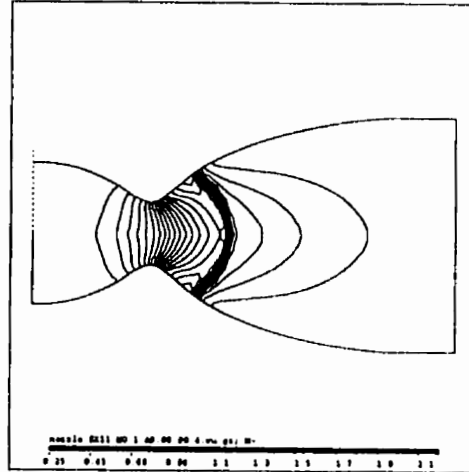


(d) Convergence

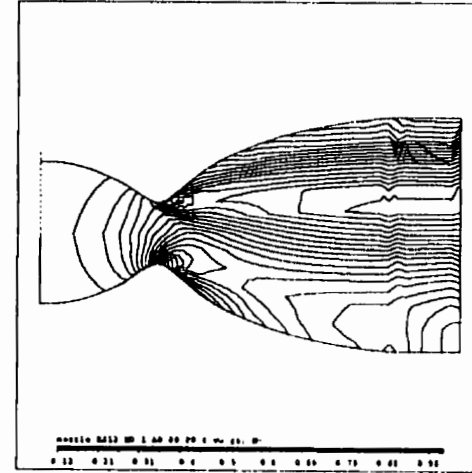
FIGURE 7.36: Distribution du nombre de Mach pour le Cas-test 7.4 avec le Maillage initial 7.10 (triangles). $\frac{p_2}{p_0} = 0.25$. Convergence de calcul



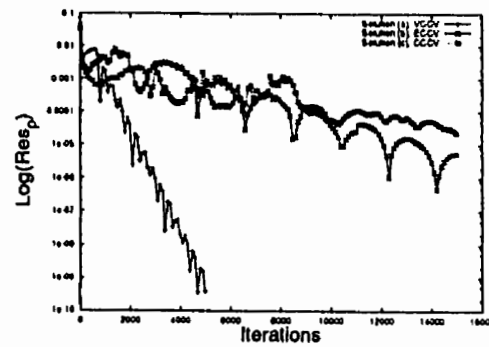
(a) Avec le Maillage final 7.30, VCCV



(b) Avec le Maillage final 7.31, ECCV



(c) Avec le Maillage initial 7.10 (ou final 7.32), CCCV



(d) Convergence

FIGURE 7.37: Distribution du nombre de Mach pour le Cas-test 7.4 avec le Maillage initial 7.10 (triangles). $\frac{p_2}{p_0} = 0.4$. Convergence de calcul

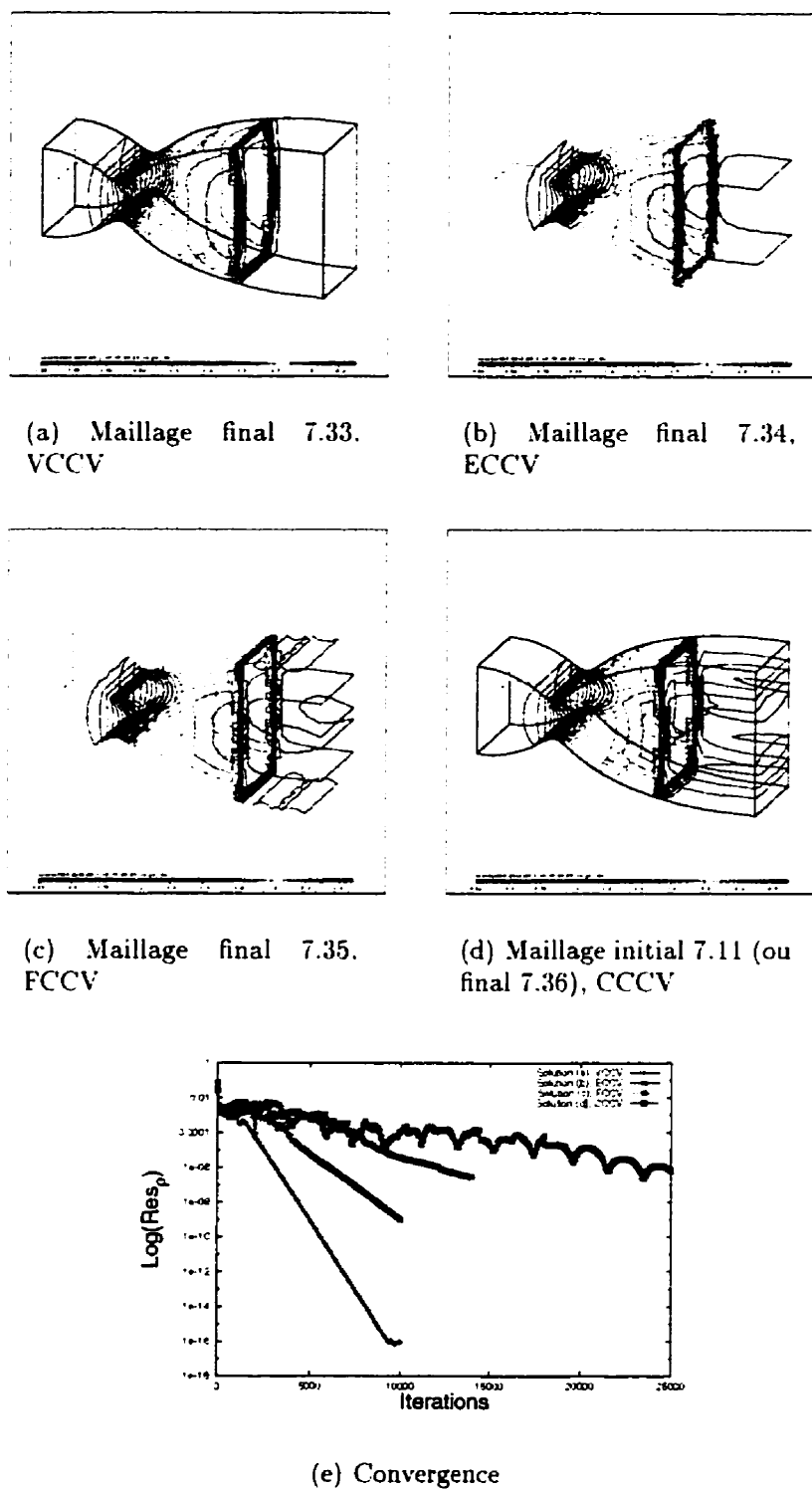
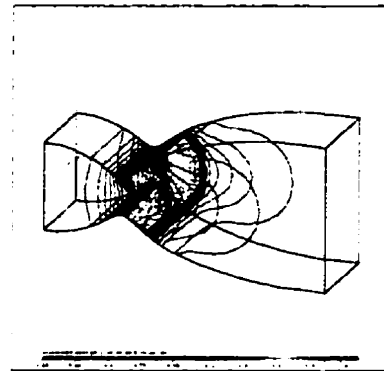
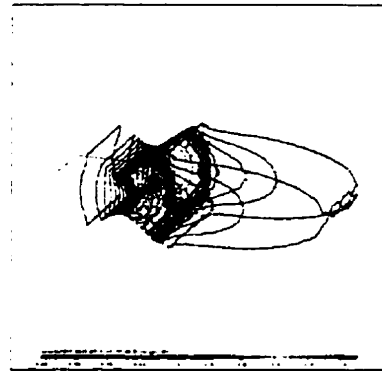


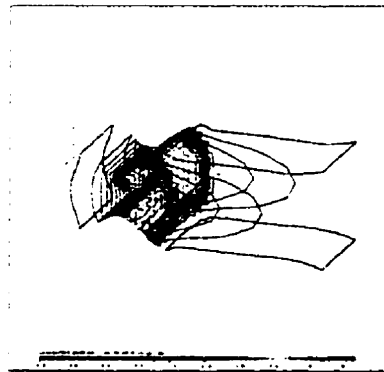
FIGURE 7.38: Visualisation des iso-surfaces du nombre de Mach pour le Cas-test 7.4 avec le Maillage initial 7.11 (tetraèdres). $\frac{p_s}{p_o} = 0.25$. Convergence de calcul



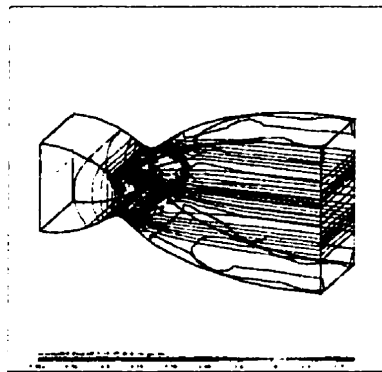
(a) Maillage final 7.33,
VCCV



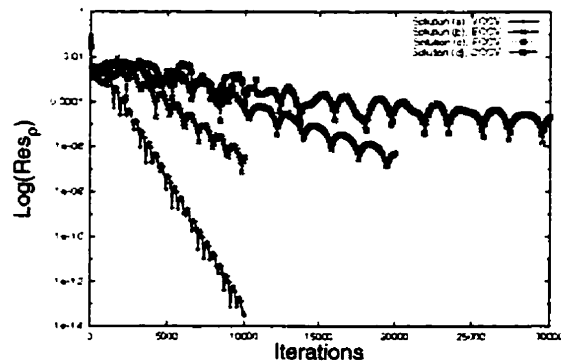
(b) Maillage final 7.34,
ECCV



(c) Maillage final 7.35,
FCCV



(d) Maillage initial 7.11 (ou
final 7.36), CCCV



(e) Convergence

FIGURE 7.39: Visualisation des iso-surfaces du nombre de Mach pour le Cas-test 7.4 avec le Maillage initial 7.11 (tétraèdres). $\frac{p_1}{p_0} = 0.4$. Convergence de calcul

de nous donner une position objective pour une meilleure disposition de la variable. La seule conclusion qu'on peut tirer de ce Cas-test est que le CCCV n'est pas toujours un schéma pouvant nous apporter de bons résultats pour les écoulements internes avec des rapports de pression provoquant des chocs, surtout si ces rapports se rapprochent des valeurs critiques. Autrement dit, ce qui semblait vrai pour les écoulements internes à haute vitesse (Cas-test 7.3) ou d'après les avantages de CCCV avec les écoulements externes (Cas-test 7.2), pour ce problème est tout à fait le contraire.

Le même cas-test a été vérifié avec d'autres codes numériques en 2D (CCCV et VCCV) différents de *VF++* avec les mêmes conditions et schéma. Les résultats et convergences obtenus étaient semblables à ceux-ci.

7.2.5 *Écoulement supersonique autour d'un profil à double ellipse*

Ce Cas-test paramétrique et qualitatif a été fait pour un profil à double ellipse avec différents nombres de Mach et angles d'incidence. Ici on ne présente que celui réalisé à $M_\infty = 2.0$ et un angle d'incidence $\alpha = 20^\circ$.

Maillage initial

Maillage initial 7.12 () Maillage 2D non structuré, composé de 7750 triangles, figure 7.40(c).

Maillage final

Maillage final 7.37 (VCCV) Obtenue du Maillage initial 7.12, avec 3966 éléments et 23250/364 bords, figure 7.40(a).

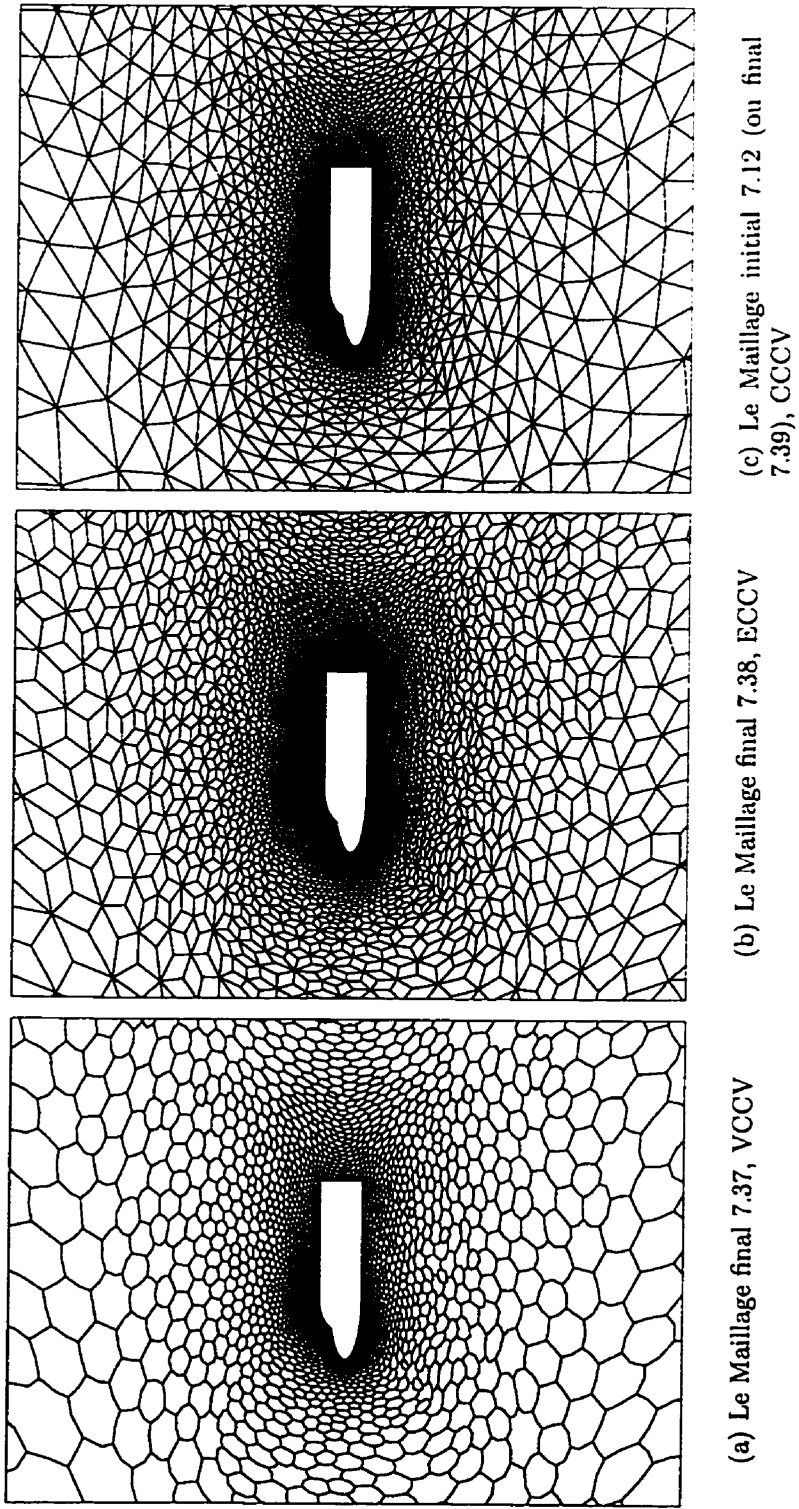
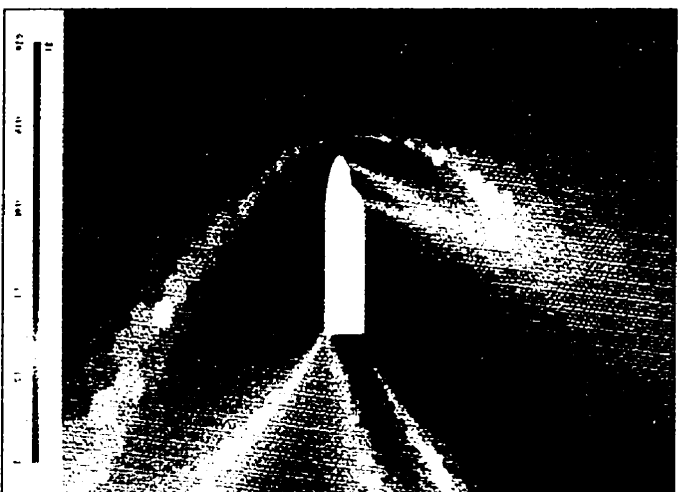


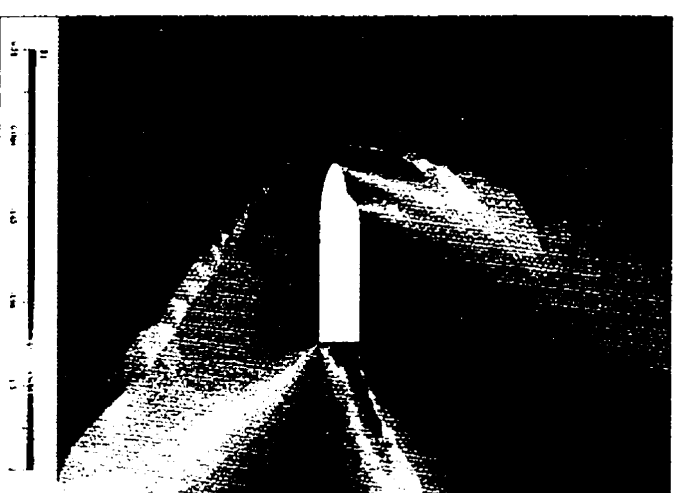
FIGURE 7.40: *Maillages 2D utilisés pour le Cas-test 7.5*



(a) Avec le Maillage final 7.37, VCCV

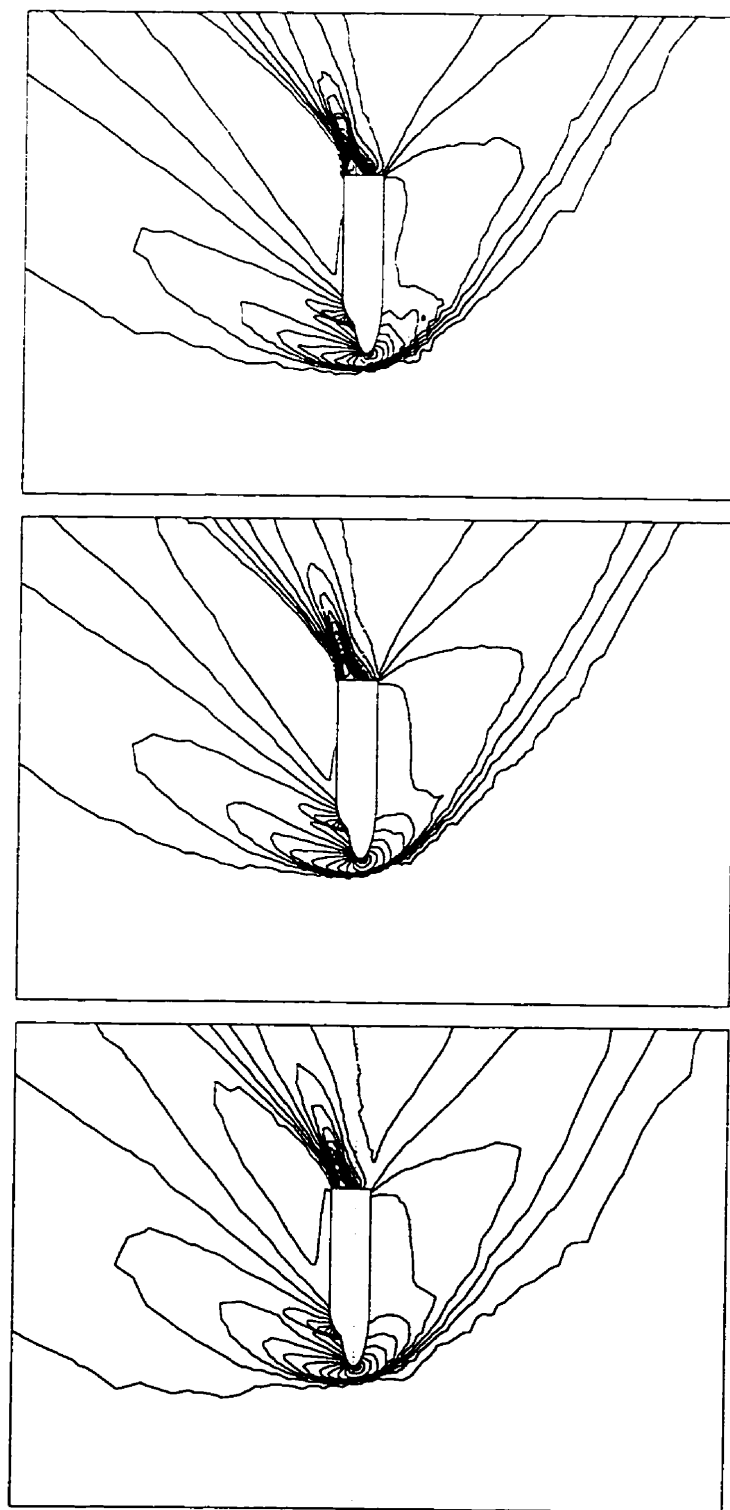


(b) Avec le Maillage final 7.38, ECV



(c) Avec le Maillage initial 7.12 (ou final 7.39), CVCV

FIGURE 7.41: *Visualisation, constante par volume, de la vitesse horizontale du Cus-test 7.5, $M_\infty = 2$, $\alpha = 20^\circ$, Maillage initial 7.12*



(a) Avec le Maillage final 7.37, VCCV (b) Avec le Maillage final 7.38, ECCV (c) Avec le Maillage initial 7.12 (ou final 7.39), CCCV

FIGURE 7.42: Iso-vitesses U du Cas-test 7.5. $M_\infty = 2$, $\alpha = 20^\circ$. Maillage initial 7.12

Maillage final 7.38 (ECCV) Obtenu du Maillage initial 7.12, avec 11716 éléments et 23250/182 bords, figure 7.40(b).

Maillage final 7.39 (CCCV) Obtenu du Maillage initial 7.12, avec 7750 éléments et 11534/182 bords, figure 7.40(c).

Le calcul

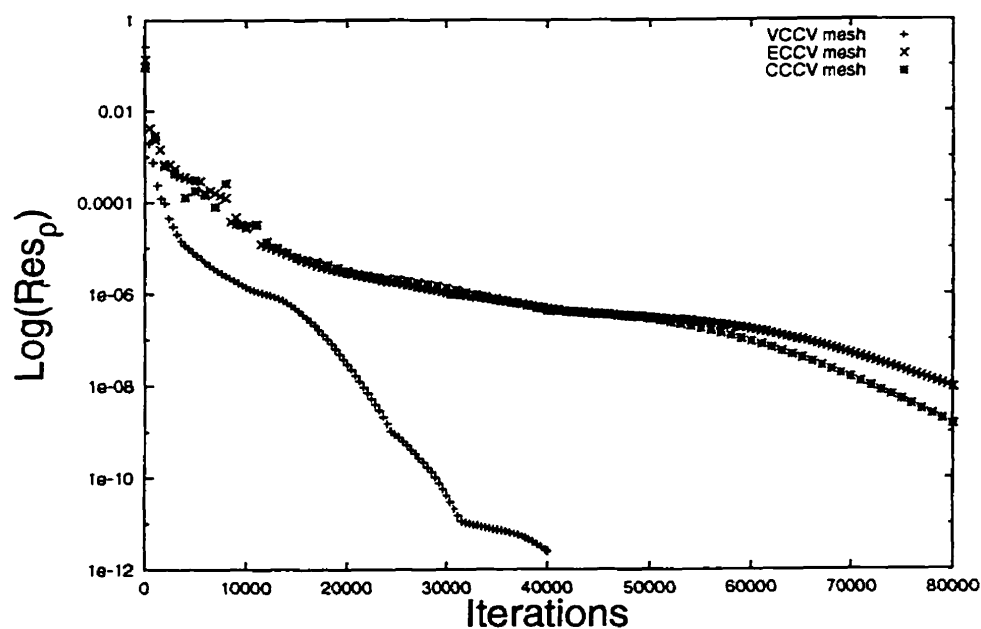


FIGURE 7.43: *Convergence de calcul pour le Cas-test 7.5*

Les résultats de cette étude sont montrés par la figure 7.41 ainsi que leurs convergences par la figure 7.43, obtenus avec un coefficient $CFL = 0.9$.

Observation 7.24 Les résultats avec les trois types de mailles de volume de contrôle sont acceptables. Cependant une meilleure définition dans la capture des ondes de choc est donnée par la solution avec ECCV suivi de CCCV.

7.3 Résultats de l'application du PROTOCOLE 2

Les tests avec ce PROTOCOLE utilisent différentes dispositions de la variable et schémas *upwind*, décrits dans la sous-section 6.4.2, pour le calcul de la composante convective des équations de convection-diffusion.

Cas-test 7.6 (Le problème de Smith et Hutton [40]) *Test préliminaire sur la diffusion numérique des schémas upwind utilisés (voir la page 157).*

Les cas-test suivants visent à la résolution des équations de *Navier-Stokes* en régime incompressible et stationnaire :

Cas-test 7.7 (Test de la cavité carrée) *Test de la cavité carrée dynamique à $Re = 1000$ (voir la page 166).*

Cas-test 7.8 (Test sur la marche) *Test sur une marche à $Re = 800$ (voir la page 173).*

Les Cas-test 7.7 et 7.8 utilisent la technique d'intégration SIMPLER (*voir le code H.4*). Étant donné que dans cette section on fera mention du décalage du maillage final, on utilisera deux types de descripteurs :

Schéma décalé 7.1 (A, VCCV/CCCV) *Les vitesses vectorielles V sont stockées sur les sommets, (VCCV) et le scalaire de la pression sur le centroïde (CCCV) de l'élément, tels qu'illustrés par la figure 6.9(a).*

Schéma décalé 7.2 (B, ECCV/CCCV) *Les vitesses vectorielles V sont stockées sur les arêtes, (ECCV) et le scalaire de la pression sur le centroïde (CCCV) de l'élément, tels qu'illustrés par la figure 6.9(b).*

7.3.1 *Le problème de Smith et Hutton*

Objectif général

Tout en utilisant les schémas FOUS et MWUS, ainsi que différents maillages initiaux, types de volumes de contrôle du tableau 4.1 et dimension du domaine, le but de ce cas-test est, principalement, d'avoir une idée qualitative *a priori* de la dose de diffusion numérique⁵ lors de la résolution des équations de convection-diffusion (1.2) sans termes sources.

Après ce test, la question de l'influence relative des termes sources et l'utilisation d'un schéma *upwind* quelconque lors de la résolution des équations de *Navier-Stokes*, est à reconsidérer.

Proposition du problème

Le domaine de calcul du problème que pose Smith et Hutton dans [40] est illustré par la figure 7.44. L'équation de convection-diffusion à résoudre est obtenue de l'équation (6.7) sans source S :

$$\mathbf{V} \cdot \nabla \Phi = \frac{1}{Pe} \nabla^2 \Phi \quad (7.1)$$

pour un champ solénoïdal et constant de vitesses égal à :

$$\begin{aligned} u &= 2y(1 - x^2) \\ v &= -2x(1 - y^2) \end{aligned}$$

avec les conditions aux frontières :

⁵ Le problème de la fausse diffusion qui apparaît dans ce Cas-test a toujours été le centre d'intérêt de plusieurs études comme [40, 24, 23, 39, 41, 12].

$$\Phi = 1 + th[10(2x + 1)] \quad (7.2)$$

à l'entrée,

$$\Phi = 0 \begin{cases} (x = -1) \wedge (0 \leq y \leq 1) \\ (y = 1) \wedge (-1 \leq x \leq 1) \\ (x = 1) \wedge (0 \leq y \leq 1) \end{cases}$$

et $\frac{\partial \Phi}{\partial y} = 0$ à la sortie.

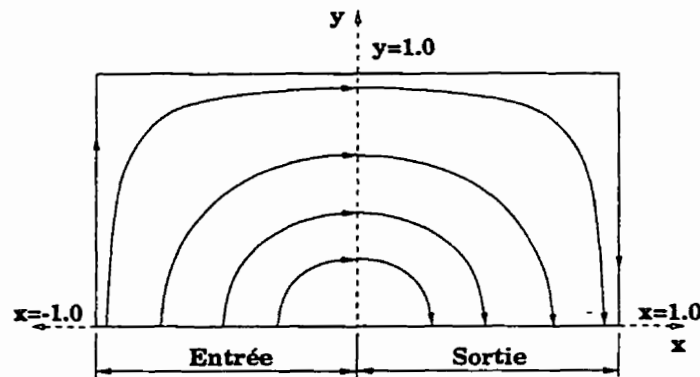


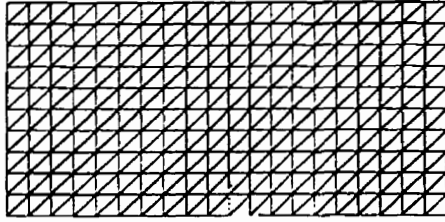
FIGURE 7.44: Géométrie du problème de Smith & Hutton avec des lignes de courant

Ce cas-test se réalise de façon paramétrique pour $Pe =]0, \infty]$. Un comportement important à observer pendant la résolution de l'équation (7.1) est celui lorsque $Pe \Rightarrow \infty$. À ce moment-là la dominance du terme convectif de cette équation doit être reflétée par une parfaite symétrie de la propriété convectante Φ dans le domaine. Une solution devient exacte lorsque à la sortie de ce domaine la propriété Φ varie selon une fonction symétrique à celle de l'équation (7.2).

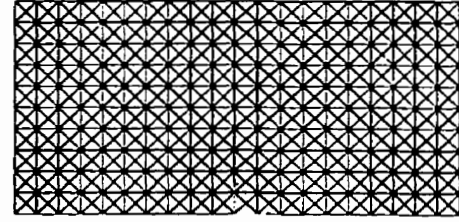
Maillage initial

Pour la résolution de ce problème on a utilisé trois types de maillages initiaux :

Maillage initial 7.13 ($20 \times 10 \times 2$) *Maillage 2D non structuré, grossier, composé de triangles, tel qu'illustré par la figure 7.45(a).*



(a) Le Maillage initial 7.13



(b) Le Maillage initial 7.14

FIGURE 7.45: *Domaine de calcul et les maillages 2D initiaux utilisés dans le Cas-test 7.6*

Maillage initial 7.14 ($20 \times 10 \times 4$) Maillage 2D non structuré, plus fin, composé de triangles, tel que montré par la figure 7.45(b).

Maillage initial 7.15 ($20 \times 10 \times 10 \times 5$) Maillage 3D non structuré, composé de tétraèdres, (pas de graphique).

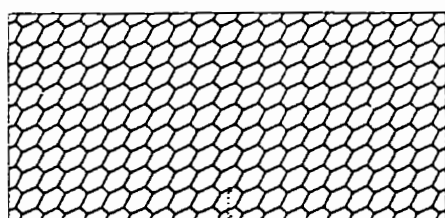
Maillage final

Les maillages finaux utilisés dans ce cas-test sont :

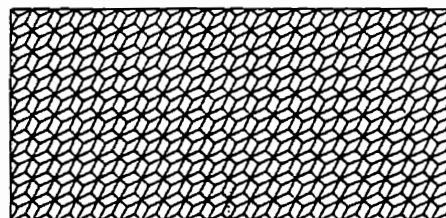
Maillage final 7.40 (VCCV) Obtenu du Maillage initial 7.13, avec 231 éléments et 1200/120 bords, tel que montré par la figure 7.46(a).

Maillage final 7.41 (ECCV) Obtenu du Maillage initial 7.13, avec 630 éléments et 1200/60 bords, tel qu'illustré par la figure 7.46(b).

Maillage final 7.42 (CCCV) Même que le Maillage initial 7.13, avec 400 éléments et 570/60 bords, tel que montré par la figure 7.45(a).

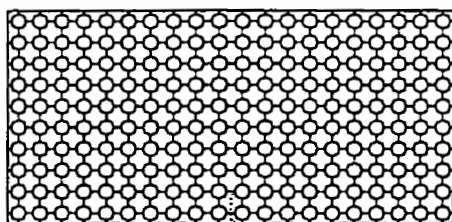


(a) Le Maillage final 7.40, VCCV

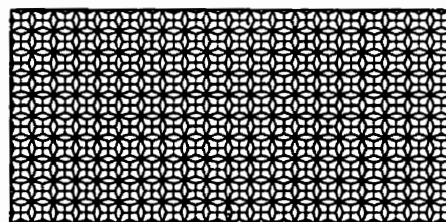


(b) Le Maillage final 7.41, ECCV

FIGURE 7.46: *Mailles des volumes de contrôle créées à partir du Maillage initial 7.13 et utilisées dans le Cas-test 7.6*



(a) Le Maillage final 7.43, VCCV



(b) Le Maillage final 7.44, ECCV

FIGURE 7.47: *Mailles des volumes de contrôle créées à partir du Maillage initial 7.14 et utilisées dans le Cas-test 7.6*

Maillage final 7.43 (VCCV) Obtenu du Maillage initial 7.14, avec 431 éléments et 2400/120 bords, tel qu'illustré par la figure 7.47(a).

Maillage final 7.44 (ECCV) Obtenu du Maillage initial 7.14, avec 1230 éléments et 2400 /60 bords, tel que montré par la figure 7.47(b).

Maillage final 7.45 (CCCV) Même que le Maillage initial 7.14, avec 800 éléments et 1170/60 bords, tel que montré par la figure 7.45(b).

Maillage final 7.46 (VCCV) Obtenu du Maillage initial 7.15, avec 2541 éléments et 60000 /6000 bords, (pas de graphique).

Le calcul

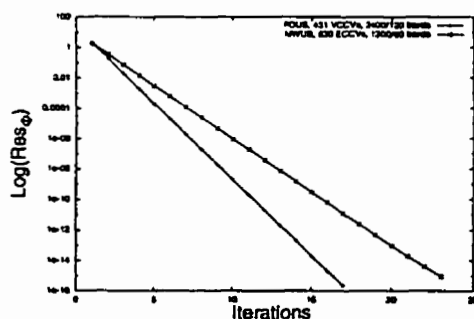
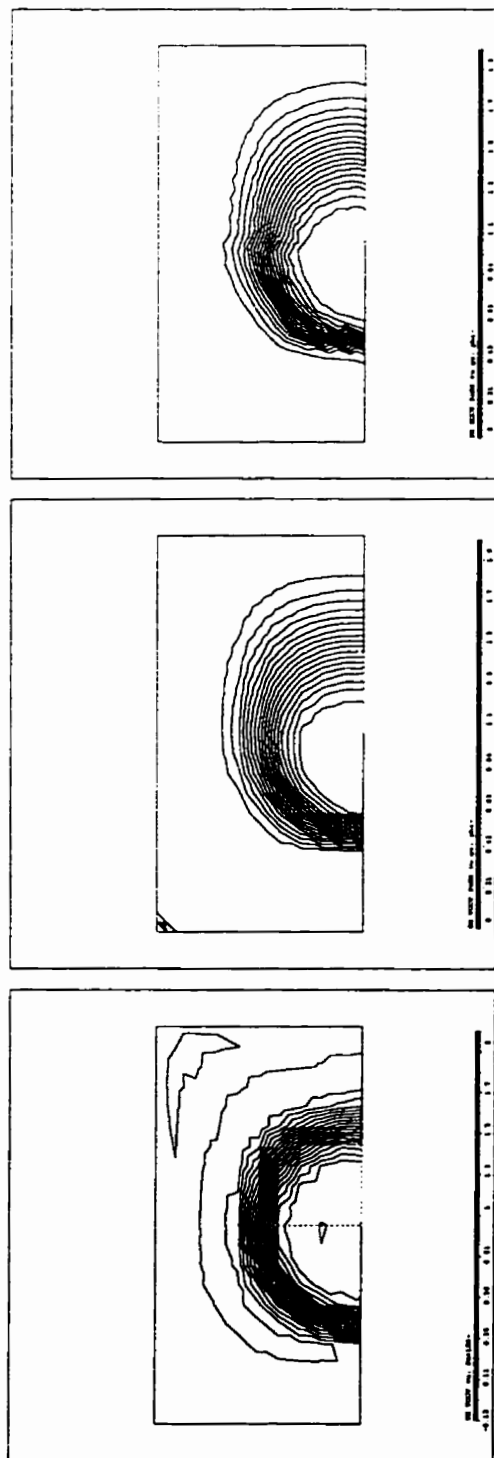


FIGURE 7.48: Convergence de calcul typique pour le Cas-test 7.6

Tous les calculs exposés pour ce cas-test ont été faits de façon implicite. Le système d'équations matricielles ainsi créé a été résolu avec la méthode **SLUSS** de la classe **VF_SYS** après avoir calculé les coefficients de l'équation (6.1), provenant de (7.1), pour un nombre de $Pe = 1E6$. D'autres tests avec des Pe plus faibles ont aussi été réalisés, mais ils ne sont pas présentés dans ce document.

Dans le cadre des résultats obtenus on constate :

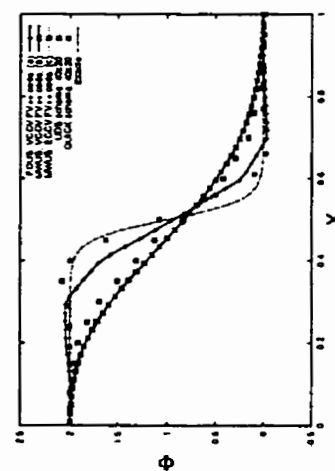
Observation 7.25 Le principe de conservation assuré pour toutes les dispositions de la variable selon le tableau 4.1 et la dimension du problème.



(a) Schéma FOUS. Maillage final 7.40, VCCV

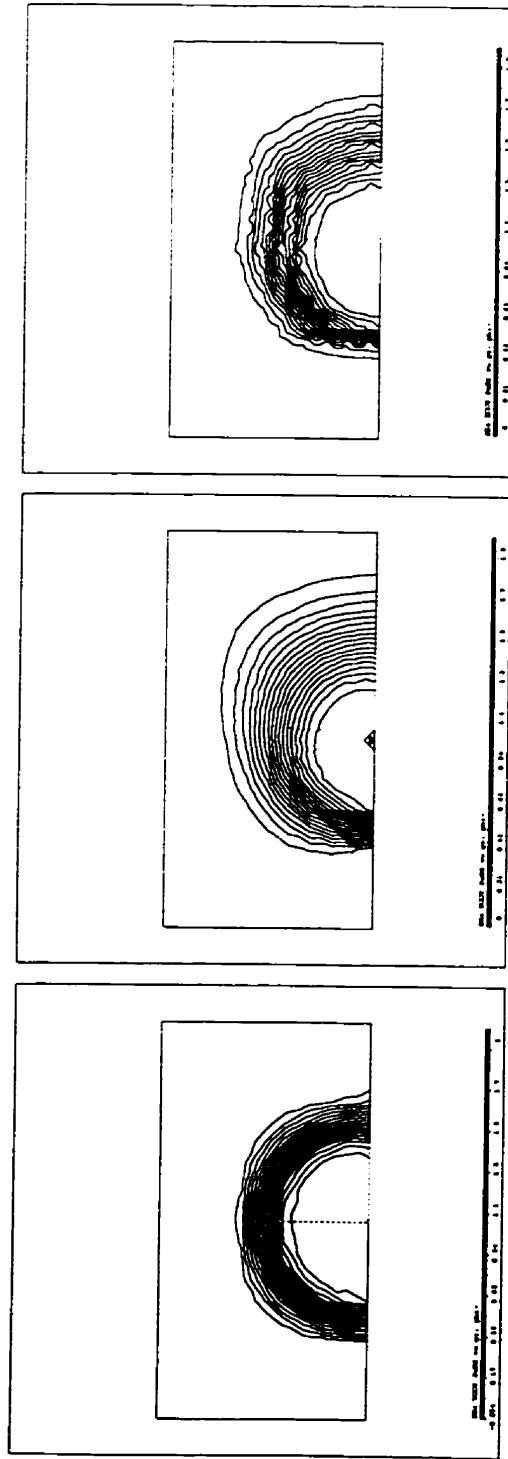
(b) Schéma MWUS. Maillage final 7.40, VCCV

(c) Schéma MWUS. Maillage final 7.41, ECCV

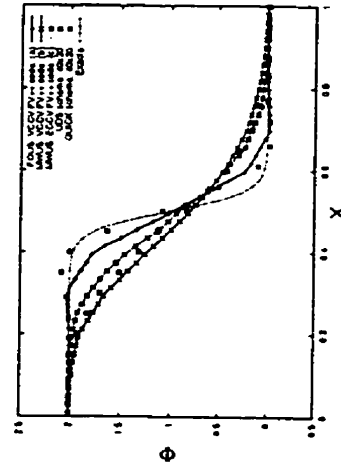


(d) Φ à la sortie, Comparaisons

FIGURE 7.49: *Cas-test 7.6. $Pe=1E6$. Maillage initial 7.13*



(a) Schéma FOUS. Maillage final 7.43, VCCV
(b) Schéma MWUS. Maillage final 7.43, VCCV
(c) Schéma MWUS. Maillage final 7.44, ECCV



(d) Φ à la sortie, Comparaisons

FIGURE 7.50: *Cas-test 7.6. $Pe=1E6$. Maillage initial 7.14*

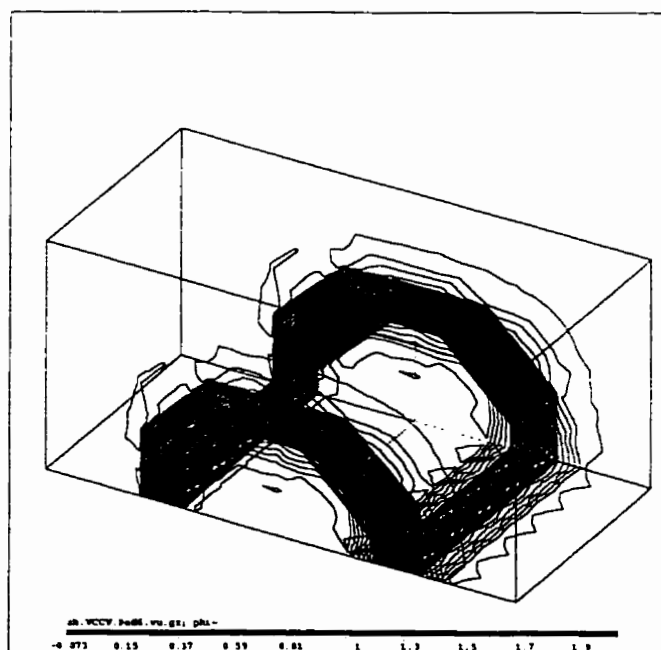
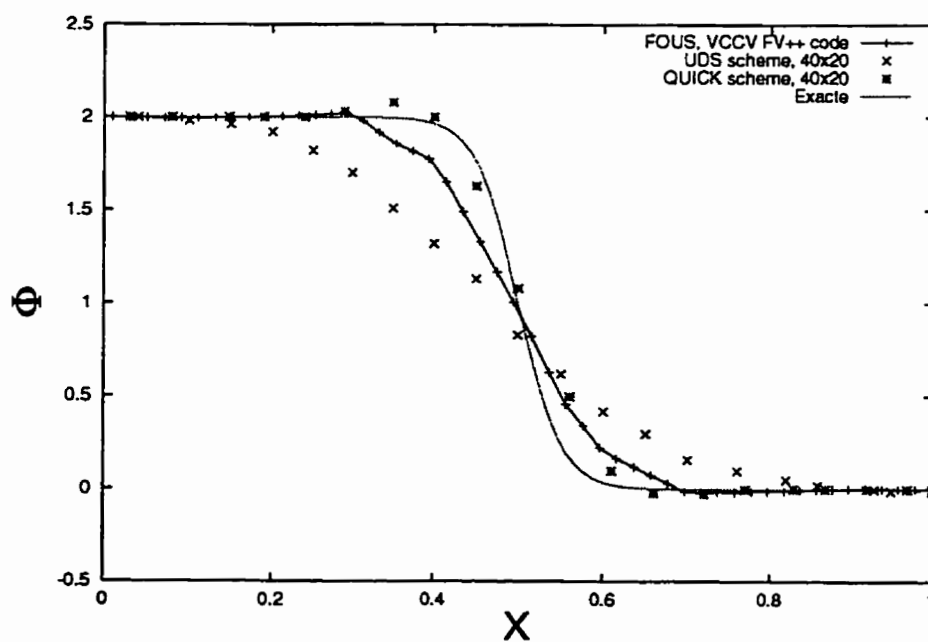
(a) Vue isométrique et iso-lignes de Φ (b) Φ à la sortie. Comparaisons. $z=0.5$

FIGURE 7.51: Cas-test 7.6 en 3D. $Pe=1E6$. Schéma FOUS. Maillage final 7.46, VCCV

Observation 7.26 Il est à noter que toutes les solutions réalisées avec le schéma **FOUS** se sont déroulées plus rapidement que celles du schéma **MWUS** avec des convergences atteignant le zéro de la machine, telles que montrées sur la figure 7.48 pour deux solutions. La convergence des autres solutions a la même allure.

Observation 7.27 Les schémas **MWUS** ou **FOUS** donnent des résultats où l'on constate la présence de la fausse diffusion due à la forme des mailles des volumes de contrôle et à l'ordre de ces schémas.

Observation 7.28 De toute évidence le schéma **FOUS** fonctionne correctement pour toutes les dispositions **VCCV** en 2D (avec des triangles) ou 3D (avec des tétraèdres). Pour toutes les autres dispositions ce schéma-là échoue⁶. Cette constatation correspond au fait que lorsqu'on utilise des stockages autres que **VCCV**, on est en train d'extrapoler les valeurs aux centroïdes des bords, avec la fonction d'interpolation exponentielle proposée par [4].

Observation 7.29 Les meilleurs résultats ont été obtenus avec le schéma **FOUS** pour **VCCV** en 2D (figures 7.49(a) et 7.50(a)) ou 3D (figure 7.51).

Observation 7.30 Le schéma **MWUS** donne des résultats satisfaisants pour les dispositions en 2D **VCCV** (figures 7.49(b) et 7.50(b)) et **ECCV** (figures 7.49(c) et 7.50(c)).

Observation 7.31 Les résultats des valeurs obtenues pour la propriété Φ à la sortie (figures 7.49(d), 7.50(d) et 7.51(b)) sont satisfaisants en comparaison avec la solution exacte et les solutions obtenues avec les schémas **UDS** et **QUICK** empruntées de [24]. Cette observation tient compte des observations précédentes.

⁶ Exemple: Pour **ECCV** avec $Pe > 300$, la solution divergeait.

7.3.2 Le problème de la cavité carrée dynamique

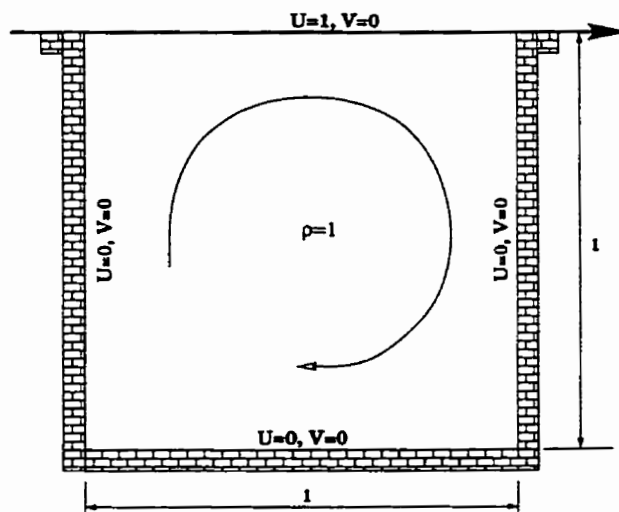


FIGURE 7.52: *Domaine de calcul et conditions pour le Cas-test 7.7*

L'objectif de ce test académique est la validation de **VF++** pour la résolution des équations de *Navier-Stokes* avec un Schéma décalé 7.1 ou 7.2 appliqué à une cavité carrée dynamique, en utilisant le schéma upwind **MWUS**, pour le calcul de la composante convective des équations mentionnées. Tous les calculs sont faits avec $Re = 1000$.

La géométrie et les conditions de ce cas-test sont illustrées par la figure 7.52.

Maillage initial

Les résultats qui seront présentés ici utilisent deux types de maillages initiaux :

Maillage initial 7.16 (40×40) *Maillage 2D structuré, concentré, composé de quadrilatères, voir la figure 7.54(d).*

Maillage initial 7.17 ($30 \times 30 \times 4$) *Maillage 2D non structuré, concentré, composé de triangles, tel qu'illustré par la figure 7.53(c).*

Maillage final

Les maillages finaux obtenus des 7.16 et 7.17 sont :

Maillage final 7.47 (VCCV) *Obtenu du Maillage initial 7.16, avec 1681 éléments et 6400/320 bords, (pas de graphique).*

Maillage final 7.48 (CCCV) *Même que le Maillage initial 7.16, avec 1600 éléments et 3120/160 bords, tel que montré par la figure 7.54(d).*

Maillage final 7.49 (VCCV) *Obtenu du Maillage initial 7.17, avec 1861 éléments et 10800/240 bords, tel que montré par la figure 7.53(a).*

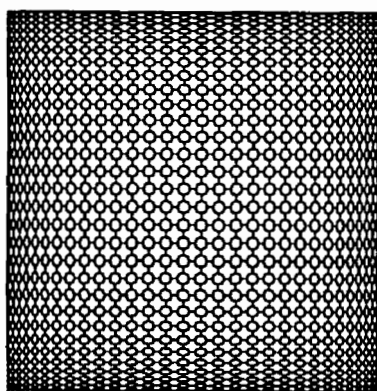
Maillage final 7.50 (ECCV) *Obtenu du Maillage initial 7.17, avec 5460 éléments et 10800/120 bords, tel qu'illustré par la figure 7.53(b).*

Maillage final 7.51 (CCCV) *Même que le Maillage initial 7.17, avec 3600 éléments et 5340/120 bords, tel que montré par la figure 7.53(c).*

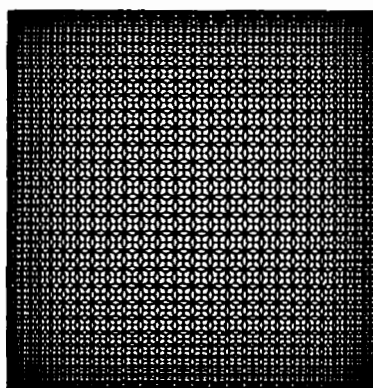
Le calcul

Les résultats de cette étude sont illustrés par la figure 7.54, pour le Maillage initial 7.16 de quadrilatères, et les figures 7.55 et 7.56 pour le Maillage initial 7.17 de triangles. Le critère de convergence utilisé est basé selon le résidu maximal de la variable dans tout le domaine. Le gradient de pression que l'on ajoute aux équations du mouvement a été calculé par interpolation. Les résultats obtenus avec la cavité carrée sont importants dû aux comportements observés lors du calcul.

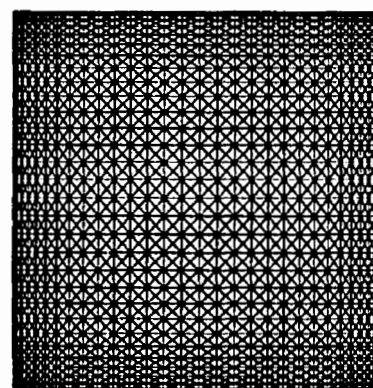
Dans cette étude qualitative et semi-quantitative, on a utilisé l'information des résultats obtenus par Ghia et al. [14] avec un maillage structuré de 129×129 .



(a) Le Maillage final 7.49, VCCV

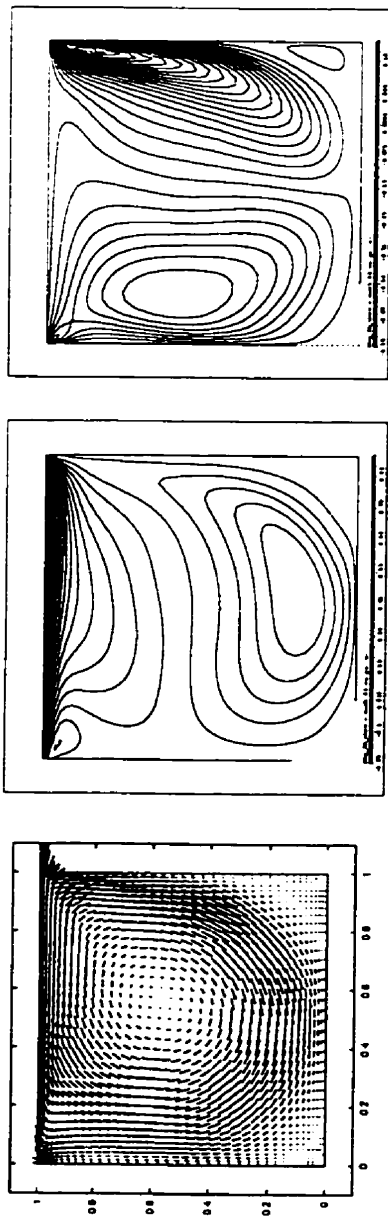


(b) Le Maillage final 7.50, ECCV



(c) Le Maillage initial 7.17 (ou final 7.51)

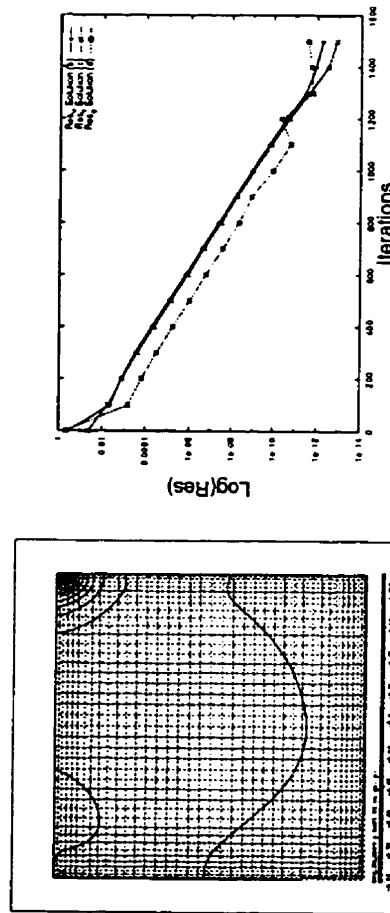
FIGURE 7.53: *Domaine de calcul et les mailles finales utilisées pour le Cas-test 7.7*



(a) Vecteurs

(b) Distribution de la vitesse horizontale. Maillage final 7.47, VCCV

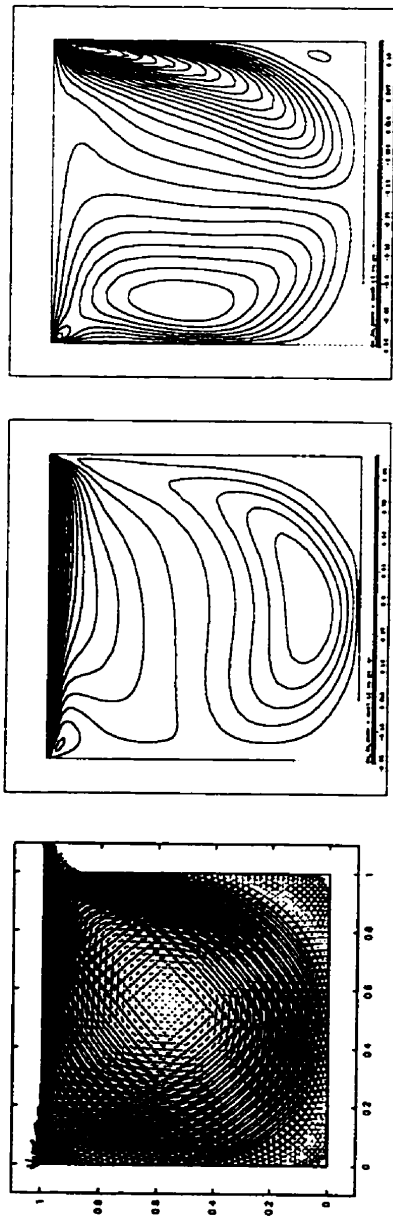
(c) Distribution de la vitesse verticale. Maillage final 7.47, VCCV



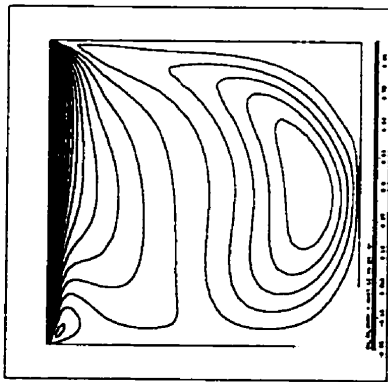
(d) Distribution de la pression. Maillage final 7.48, CCCV

(e) Convergence

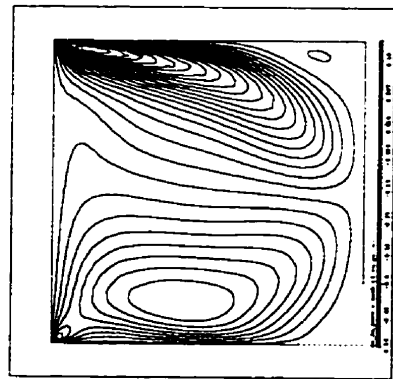
FIGURE 7.54: *Cas-test 7.7. Schéma décalé 7.1. MWUS. $Re=1E3$. Maillage initial 7.16*



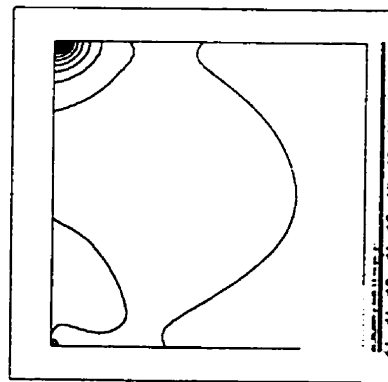
(a) Vecteurs



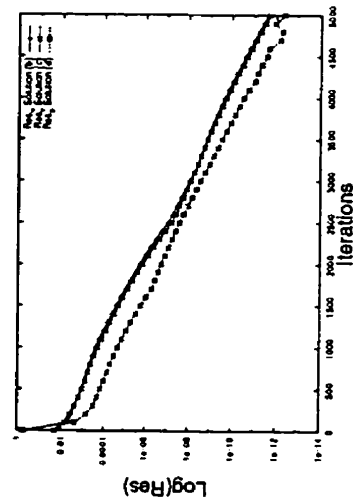
(b) Distribution de la vitesse horizontale. Maillage final 7.50, ECCV



(c) Distribution de la vitesse verticale. Maillage final 7.50, ECCV

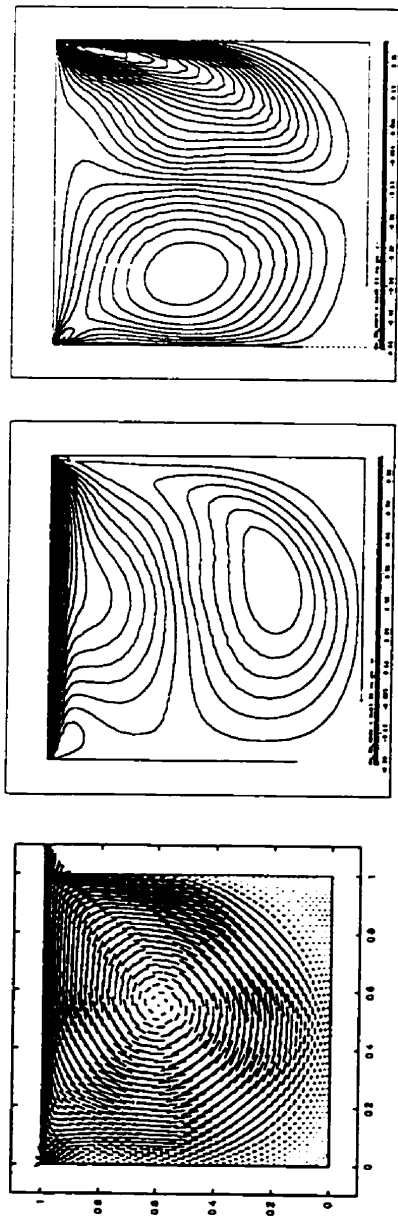


(d) Distribution de la pression. Maillage final 7.51, CCCV



(e) Convergence

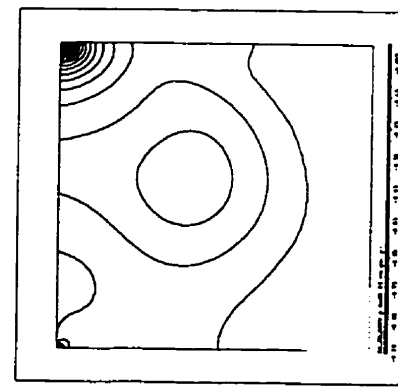
FIGURE 7.55: *Cas-test 7.7. Schéma décalé 7.2. MWUS. $Re=1E3$. Maillage initial 7.17*



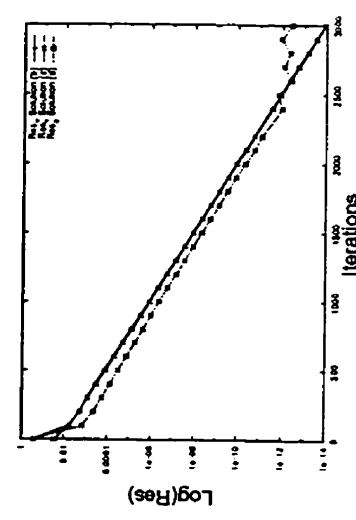
(a) Vecteurs

(b) Distribution de la vitesse horizontale. Maillage final 7.49, VCCV

(c) Distribution de la vitesse verticale. Maillage final 7.49, VCCV



(d) Distribution de la pression. Maillage final 7.51, CCCV



(e) Convergence

FIGURE 7.56: Cas-test 7.7. Schéma décalé 7.1. MWUS. $Re=1E3$. Maillage initial 7.17

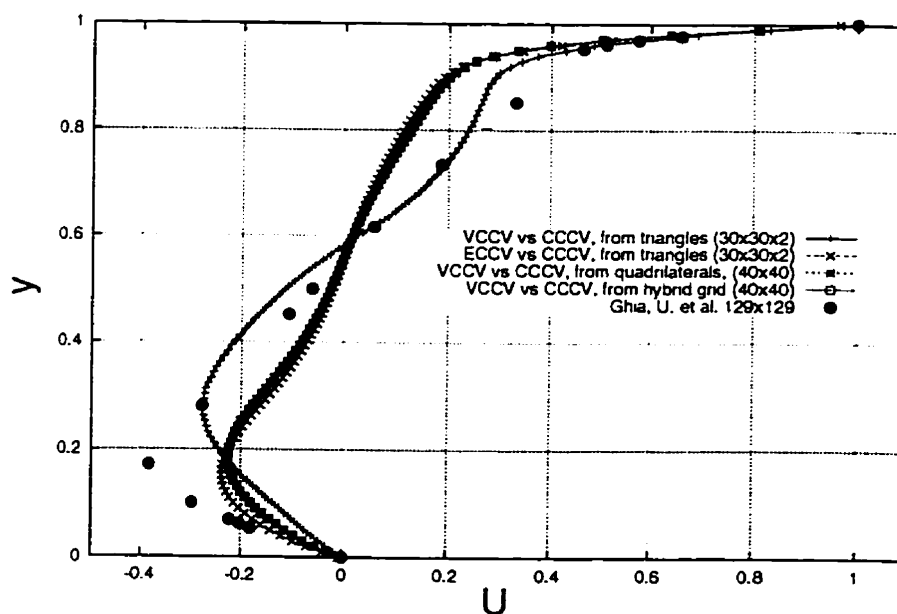
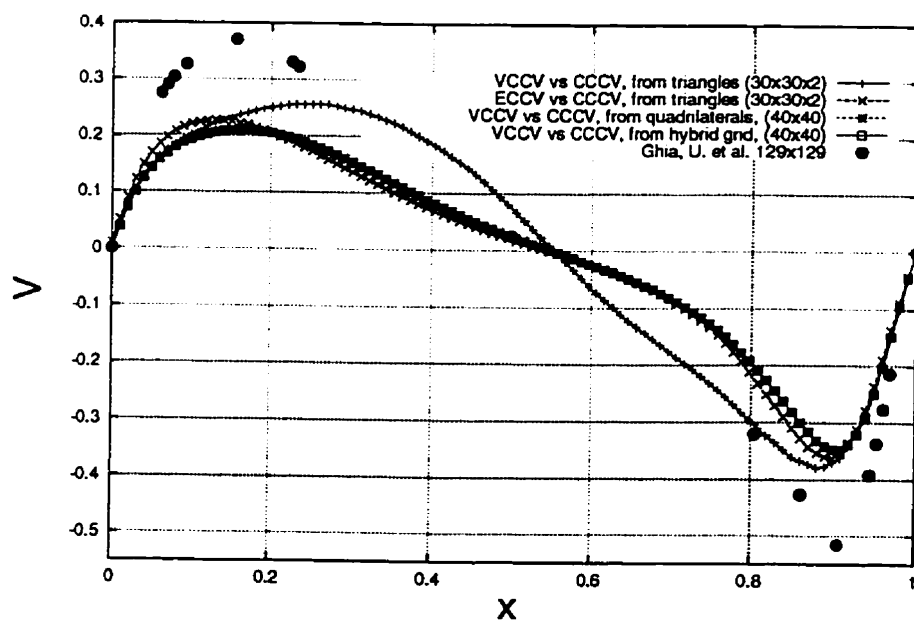
(a) Coupe $X = 0.5$, Vitesse u (b) Coupe $Y = 0.5$, Vitesse v

FIGURE 7.57: Coupes pour le Cas-test 7.7. Différents maillages initiaux et schémas décalés. Comparaisons

Les Observations sur ce Cas-test sont :

Observation 7.32 *Convergence des variables atteignant 10^{-16} après 5000 itérations, pour le Schéma décalé 7.2, et après 1500 pour le Schéma décalé 7.1.*

Observation 7.33 *Résultats diffusifs mais satisfaisants si l'on considère la grossièreté du maillage et la vitesse horizontale à $x = 5$ et verticale à $y = 0.5$.*

Observation 7.34 *Tandis que la solution qui utilise le Schéma décalé 7.1 (sur le Maillage initial 7.17, triangles) devient très éloignée de la solution avec le Schéma décalé 7.2 (sur le même maillage initial), la solution avec le Schéma décalé 7.1 (sur le Maillage initial 7.16, quadrilatères) se rapproche singulièrement du deuxième Schéma décalé avec des triangles. Cette ressemblance peut être aperçue sur les figures 7.54 et 7.55 et les coupes de la figure 7.57.*

Observation 7.35 *En essayant d'expliquer les raisons de ce phénomène et en voulant tester les performances hybrides de ce PROTOCOLE, on a répété la solution avec un maillage hybride consistant de 1390 quadrilatères et 420 triangles, ces derniers disposés dans les coins inférieurs, les résultats ont confirmé le rapprochement constaté par l'Observation 7.34.*

7.3.3 *Le problème de la marche*

Par analogie avec le Cas-test 7.7, ce test vise à la validation qualitative et semi quantitative de **VF++** pour la résolution des équations de *Navier-Stokes* avec un Schéma décalé 7.1 ou 7.2 appliqué à une marche dont la géométrie et les conditions sont illustrées par la figure 7.58, en utilisant le schéma *upwind* MWUS, avec $Re = 800$. Le choix de ce nombre de Reynolds n'est pas arbitraire, car au moins deux zones

de recirculation auront lieu, la première inférieure atteignant X_1 et la deuxième entre X_2 et X_3 .

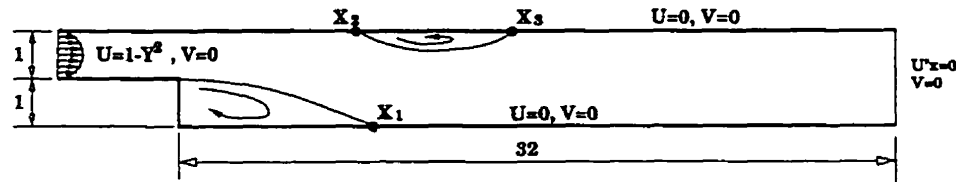


FIGURE 7.58: *Domaine de calcul et conditions pour le Cas-test 7.8*

Maillage initial

Deux maillages initiaux sont utilisés dans ce Cas-test avec les caractéristiques suivantes dans la zone d'expansion :

Maillage initial 7.18 ($64 \times 40 \times 2$) *Maillage 2D non structuré, concentré, composé de triangles, (pas de figure).*

Maillage initial 7.19 () *Maillage 2D hybride, concentré, composé de triangles et quadrilatères, construit à partir du Maillage initial 7.18, illustré partiellement par la figure 7.60(c).*

Maillage final

Les maillages finaux utilisés sont :

Maillage final 7.52 (VCCV) *Obtenu du Maillage initial 7.18, avec 2834 éléments et 16320/452 bords, (pas de figure).*

Maillage final 7.53 (ECCV) *Obtenu du Maillage initial 7.18, avec 8273 éléments et 16320/226 bords, (pas de figure).*

Maillage final 7.54 (CCCV) Même que le Maillage initial 7.18, avec 5440 éléments et 8047/226 bords, (pas de figure).

Maillage final 7.55 (VCCV) Obtenu du Maillage initial 7.19, avec 2834 éléments et 14584/452 bords, (pas de figure).

Maillage final 7.56 (CCCV) Même que le Maillage initial 7.19, avec 4572 éléments et 7179/226 bords, voir la figure 7.60(c).

Le calcul

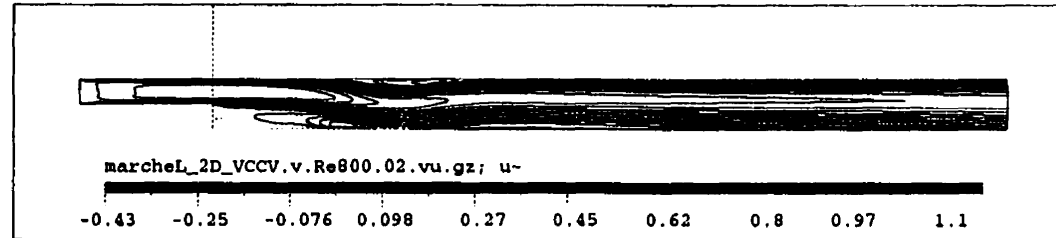
Les résultats de cette étude sont illustrés par les figures 7.59..7.61 et le tableau 7.1. Aucune correction de la pression n'a été faite, le gradient de pression a été calculé par la méthode des moindres carrés.

Les résultats de cette étude semi-quantitative ont été comparés avec les données expérimentales [2] et les résultats obtenus par [37] avec adaptativité et des mailles structurées de 64×64 .

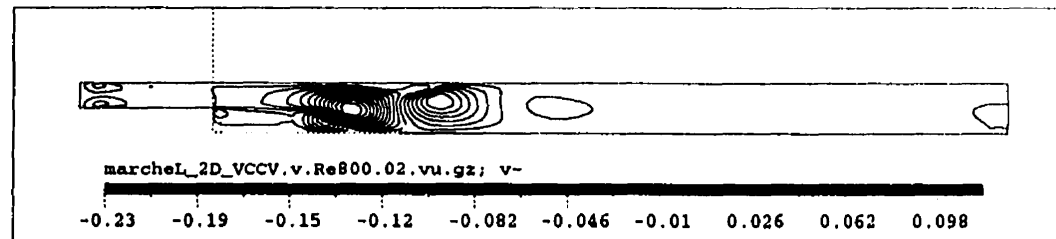
Observations de résultats

Observation 7.36 Convergence très difficile pour le Schéma décalé 7.2, atteignant 10^{-3} , mais bonne pour le Schéma décalé 7.1, atteignant $10^{-8}..10^{-9}$ après la 5000-ème itération.

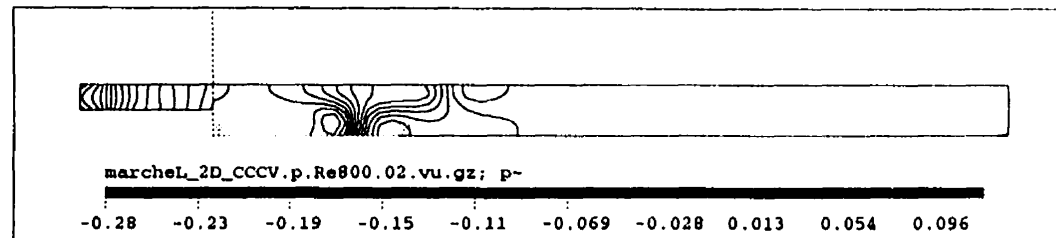
Observation 7.37 Résultats diffusifs mais satisfaisants pour le Schéma décalé 7.2, malgré sa convergence, et mauvais pour le Schéma décalé 7.1 avec le Maillage initial 7.18. Par contre, l'utilisation des mailles hybrides avec le Schéma décalé 7.1 améliore



(a) Distribution de la vitesse horizontale. Maillage final 7.52, VCCV

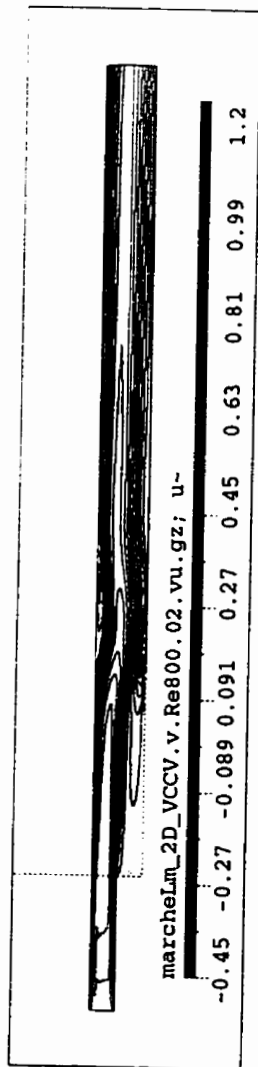


(b) Distribution de la vitesse verticale. Maillage final 7.52, VCCV

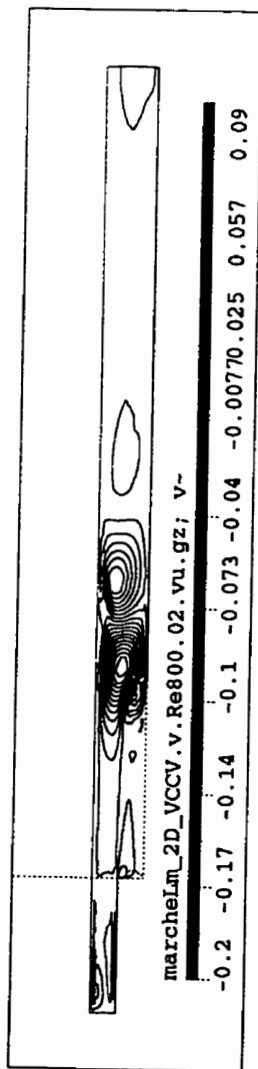


(c) Distribution de la pression. Maillage final 7.54, CCCV

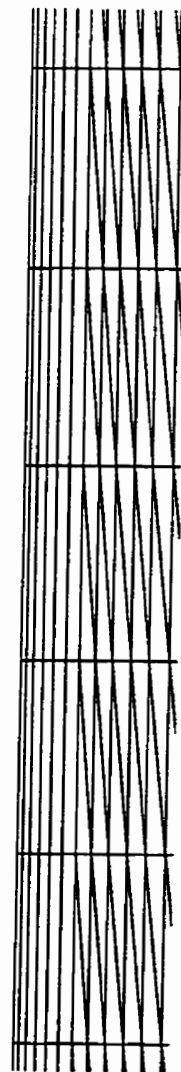
FIGURE 7.59: *Cas-test 7.8. Maillage initial 7.18. Schéma décalé 7.1. MWUS*



(a) Distribution de la vitesse horizontale. Maillage final 7.55, VCCV

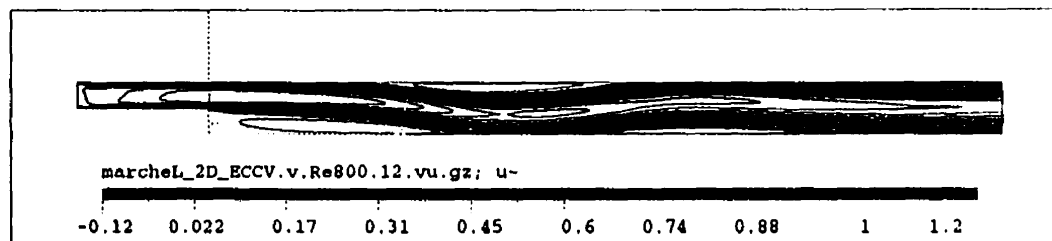


(b) Distribution de la vitesse verticale. Maillage final 7.55, VCCV

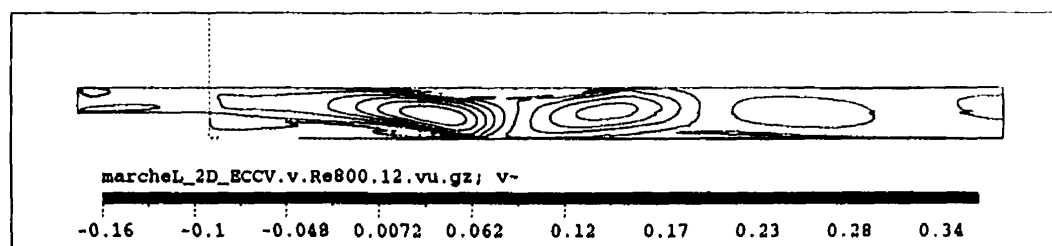


(c) Maillage initial 7.19 (ou final 7.56) CCCV, (détail)

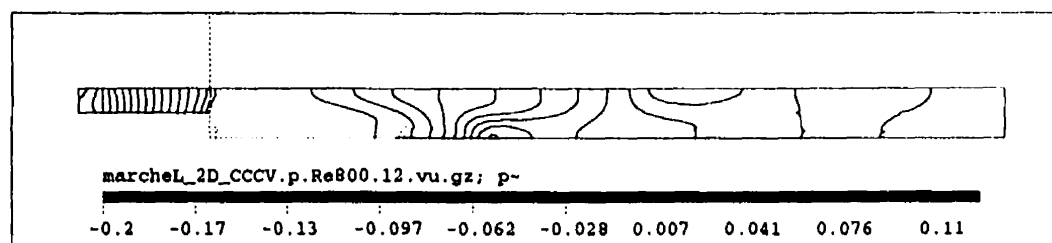
FIGURE 7.60: Cas-test 7.8. Maillage initial 7.19 hybride. Schéma décalé 7.1. MWUS



(a) Distribution de la vitesse horizontale. Maillage final 7.53, ECCV



(b) Distribution de la vitesse verticale. Maillage final 7.53, ECCV



(c) Distribution de la pression. Maillage final 7.54, CCCV

FIGURE 7.61: Cas-test 7.8. Maillage initial 7.18. Schéma décalé 7.2. MWUS

considérablement la qualité de la solution, telle que constatée par l'Observation 7.35 dans le cas de la cavité carrée.

Observation 7.38 Les valeurs des positions du point de rattachement inférieur X_1 et des points de rattachement/détachement supérieurs X_2 et X_3 , comparées avec les données expérimentales [2] et les résultats numériques [37], montrent clairement le problème diffusif des deux schémas décalés. Ici on confirme les résultats visuels des figures 7.59.. 7.61 favorable au Schéma décalé 7.2, pour ce type de maillage initial, mais également acceptable pour le Schéma décalé 7.1 avec des mailles hybrides.

TABLEAU 7.1: Comparaisons des résultats du Cas-test 7.8 (marche) pour $Re = 800$

Source	X_1	X_2	X_3
Expériment [2]	14.2	10.8	19.6
Référence [37]	8.6	10.9	19.5
Schéma décalé 7.1 (triangles)	7.2	4.8	10.1
Schéma décalé 7.1 (hybrides)	9.5	7.4	13.4
Schéma décalé 7.2 (triangles)	9.5	6.5	16.2

Chapitre 8

Conclusions et recommandations

8.1 Conclusions

8.1.1 *Sur le système*

1. Une approche orientée-objet permet la représentation abstraite, compacte et générique de la problématique de la méthode des volumes finis.
2. **VF++** est un code orienté-objet, dont la conception est basée sur la transformation de deux topologies du domaine de calcul, selon l'équation relationnelle :

$$\{S, A, F, V\} \Longrightarrow \{B, E, [S']\}$$

3. Les avantages de celui-ci portent sur un environnement favorable à la recherche au sujet de la méthode des volumes finis pour des applications immédiates, en utilisant des mailles initiales polygonales, polyédriques et pour tout type de disposition de la variable selon le tableau 4.1. Le traitement avec des mailles hybrides et décalées est possible.
4. **VF++** est un code évolutif, robuste et compact dont la généralisation a toujours été en conflit avec sa performance.

5. Les coûts en mémoire de stockage et la vitesse de calcul sont généralement fonction du nombre d'éléments et de bords de contrôle, pour le même cas-test et même domaine de calcul.

8.1.2 *Sur les applications*

1. Les résultats de l'utilisation des différents types de volume de contrôle, conduisent à établir que le *type* de maillage initial, la *forme*, l'*orientation* et la *position* des volumes de contrôle et la *géométrie* du domaine de calcul jouent un rôle déterminant dans la solution d'un problème spécifique. L'expression consacrée «*dépendance du maillage*» est, dans cette étude, catégoriquement remplacée par «*dépendance du volume de contrôle*».
2. Le nombre de volumes de contrôle n'a pas toujours une influence déterminante sur la qualité des résultats. Par exemple, des résultats aussi mauvais ont été obtenus en augmentant le nombre de CCCVs (Cas-test 7.4).
3. Le VCCV dans la résolution des équations d'*Euler* et/ou *Navier-Stokes* donne des solutions très diffusives mais bien convergées.
4. Lors de la résolution des équations d'*Euler* en régime compressible pour les écoulements supersoniques et transsoniques externes, CCCV et ECCV donnent des résultats acceptables. Par contre, la capture du choc est mieux définie par ECCV.
5. Pour les écoulements supersoniques internes, la tendance est contradictoire.
S'il s'agit des écoulements libres à haute vitesse sans rapport de pression à l'intérieur du domaine, CCCV suivi de ECCV, (FCCV en 3D) donne des meilleurs résultats.

- Si, par contre, l'écoulement est gouverné par un rapport de pression à la sortie provoquant des chocs, la qualité de la solution de CCCV diminue dans la capture du choc et si ce rapport se rapproche de la valeur critique, la solution peut devenir instable donnant des résultats absurdes. Dans ce cas, ce sont ECCV suivi de FCCV qui donnent des résultats convergés et plus réalistes.
- 6. Dans la résolution des équations de *Navier-Stokes* en régime incompressible avec des mailles décalées, il manque beaucoup de tests pour donner un jugement objectif. Tout semble dire qu'il ne faut pas négliger la composition du Schéma décalé 7.1 avec des mailles hybrides ou l'utilisation de quadrilatères dans le maillage initial.

8.2 Recommandations

Les étapes à suivre dans l'évolution de *VF++* vers des versions plus génériques suggèrent les études et les mises en œuvre suivantes :

- Descripteur du problème. Paramétrisation et systématisation d'un problème.
- Renumérotation globale performante.
- Changement des structures de données plus profitables.
- Renforcer l'interface résolveur.
- Implémenter des interfaces pour des mailles mobiles.
- Considérer les problèmes posés par l'Épilogue de cette thèse.

Épilogue

On termine cette étude en faisant appel à trois considérations importantes et critiques concernant la topologie finale :

Nouvelles entités topologiques. Selon le chapitre 6, peu importe le type de volumes de contrôle du tableau 4.1 et la dimension du problème, l'interface résolveur ne reconnaît que :

Bords, les bords de contrôle, $\mathbb{B}$,

Éléments, les volumes de contrôle, $\mathbb{E}$,

Sommets, les sommets des éléments (ou des bords), $\mathbb{S}$.

Les CCCVs originals. La nature des dispositions CCCV pose un grand problème dans la généralisation. Spécifiquement, une grande tâche dans le développement des codes génériques pour volumes finis est de rendre unique la catégorisation concernant les éléments-frontières des volumes de contrôle (sous-section 6.2.1).

Régénération du système. Génération infinie d'autres mailles de volumes de contrôle, gérés par le *caractère régénératif du système*. Cela peut être conçu si la sortie de *vfGeo* est aussi son entrée (figure 2.1). Les volumes de contrôle pourront être générés par un arbre de dérivation binaire (en 2D) (figure 8.1) et ternaire (en 3D). À cet égard, le nombre de types de volumes de contrôle est obtenu par récurrence avec N étant le nombre de niveaux :

$$\text{Nombre de types de volumes de contrôle} = \sum_{n=0}^N DIM^n$$

Alors, les acronymes VCCV, ECCV, FCCV, CCCV, ..., EVVFCCV, etc, sont absorbés et disparaissent avec la généralisation.

De cette équation, le maillage initial est obtenu si $N = 0$. Si $N = 1$, on obtient l'ensemble des dispositions données par le tableau 4.1, dont la conception et l'emploi sont l'âme de cette étude.

Autrement dit, la résolution de cet ensemble de questions et paradigmes, offrira un grand saut qualitatif dans l'implantation et la généralisation relative à la méthode des volumes finis. À ce moment-là on envisage les CCCVs ou volumes de contrôle anonymes comme les supports essentiels et uniques de la théorie de la méthode. Tenant compte de ces considérations, on conclut cette étude avec la règle suivante :

«Il n'existe que CCCVs»*

Dans ce contexte, cette étude n'a donné qu'une interprétation partielle à cet oracle.

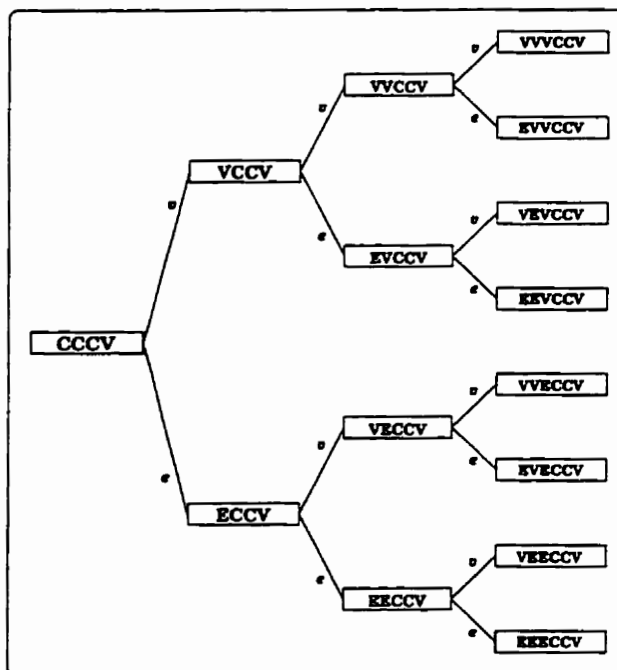


FIGURE 8.1: Arbre binaire des volumes de contrôle en 2D, obtenu de la régénération

Bibliographie

La méthode des volumes finis et bibliographie fondamentale

- [1] AFSHAR M. (1992). *Investigation and Improvement of Some Recent FVM and CVFEM Practices for Two-Dimensional, Incompressible, Viscous Fluid Flow*. Thesis, McGill University.
- [2] ARMALY B. F., DURST F., PEREIRA J.C. et SCHONUNG B. (1983). *Experiment and Theoretical Investigation of Backward-Facing Step Flow*. J. Fluid Mechanics. Vol. 127, pp. 473-496.
- [3] BAILEY C. (1996). *PHYSICA : A Multiphysics Computational Framework and its Application to Casting Simulations*. University of Greenwich, London.
- [4] BALIGA B. R. et PATANKAR S. V. (1980). *A New Finite-Element Formulation for Convection-Diffusion Problems*. Numerical Heat Transfer. Vol 3, pp. 393-409.
- [5] BARRÈRE M. et JAUMOTTE A. (Moskva, 1962). *Raketnie Dvigateli*. Oborongiz.
- [6] BASTIAN P. (November 17, 1995). *UG version 3.1 Manuals : Programming Manual, Reference Manual and Application and Problem Class Documentation*. Universität Stuttgart. UG Group.
- [7] BECK R., ERDMANN B. et ROITZSCH R. (June, 1995). *KASKADE 3.0. An Object-Oriented Adaptive Finite Element Code*. Konrad-Zuse-Zentrum für Informationstechnik Berlin, ZIB. Technical Report TR 95-4.

- [8] BIJAN M. (May, 1994). *Fluid Dynamics Computation with NSC2KE. An User Guide, Release 1.0.* INRIA, RT-0164.
- [9] CAMARERO R. (1973). *Numerical Solutions of Internal and External Hypersonic Flows at High Incidence.* Thesis, McGill University.
- [10] CHEN C. L. et CHAKRAVARTHY S. R. (September, 1994). *Numerical Investigation of Separated Nozzle Flows.* AIAA Journal. Vol. 32, pp. 1836-1843.
- [11] CIARLET P. (1978). *The Finite Element Methods for Elliptic Problems.* Amsterdam, New York, North-Holland.
- [12] DARWISH M. S. (1993). *A New High-Resolution Scheme Based on the Normalized Variable Formulation.* Numerical Heat Transfer. Part B, Vol. 24, pp. 353-371.
- [13] DEHMEL A. (Germany, April 1996). *Numerische Simulation dreidimensionaler Strömungen mit freien Randwerten.* Diplomarbeit, Technische Universität München.
- [14] GHIA U., GHIA K. N. et SHIH C. T. (1982). *High-Re Solution for Incompressible Flow Using Navier-Stokes Equations and a Multigrid Method.* Journal of Computational Physics. Vol. 48, pp. 387-411.
- [15] HOOKEY N. A. (Montreal, 1986). *Evaluation and Enhancements of Some Control Volume-Finite Element Methods for Two-Dimensional Fluid Flow and Heat Transfer.* Thesis, McGill University.
- [16] HOOKEY N. A., BALIGA B. R. et PRAKASH C. (1988). *Evaluation and Enhancements of Some Control-Volume Finite-Element Methods - part 1: Convection Diffusion Problems.* Numerical Heat Transfer. Vol. 14, pp. 255-272.
- [17] HOOKEY N. A. et BALIGA B. R. (1988). *Evaluation and Enhancements of Some Control-Volume Finite-Element Methods - part 2: Fluid Flow Problems.* Numerical Heat Transfer. Vol. 14, pp. 273-293.

- [18] ILINCA A. (1994). *Calcul des écoulements compressibles tridimensionnels sur des maillages en mouvement et adaptatifs*. Thèse, École polytechnique de Montréal.
- [19] JAMESON A., SCHMIDT W. et TURKEY E. (1981). *Numerical Solution for Euler Equations by a Finite Volume Method Using Runge-Kutta Time Stepping Schemes*. AIAA Paper 81-1259.
- [20] KAI-HSIUNG K. et MENG-SING L. (May, 1994). *Grid Adaptation Using Chimera Composite Overlapping Meshes*. AIAA Journal. Vol. 32, No 5, pp. 942-949.
- [21] KOVENIA V. (Novosivirsk, 1990). *Primenienie metoda razsheplenia v zadachah Aerodynamiki*. Izdatel'stvo Nauka.
- [22] LE DAIN MUIR B. (May, 1983). *A Control Volume Finite Element Method for Three-Dimensional Elliptic Fluid Flow and Heat Transfer*. Thesis, McGill University.
- [23] LI Y. et RUDMAN M. (1995). *Assessment of Higher-Order Upwind Schemes incorporating FCT for Convection-Diffusion Problems*. Numerical Heat Transfer. Part B, Vol. 27, pp. 1-21
- [24] LIEN F. S. et LESCHZINER M. A. (1994). *Upstream Monotonic Interpolation for Scalar Transport with Application to Complex Turbulent Flows*. International Journal for Numerical Methods in Fluids. Vol. 19, pp. 527-548
- [25] OLLIVIER-GOOCH C. F. (January, 1996). *A New Class of ENO Schemes Based on Unlimited Data-Dependent Least-Squares Reconstruction*. AIAA 96-0887, Mathematics and Computer Science Division.
- [26] OSHER S. (1984). *Rieman Solvers, the Entropy Condition, and Difference approximation*. SIAM J. Numerical Analysis. Vol. 21, pp. 217-235.
- [27] OSHER S. et SOLOMON F. (1982). *Upwind Schemes for Hyperbolic Systems of Conservation Laws*. Math. Comp.. Vol. 38, pp. 339-377.
- [28] PATANKAR S. V. (Washington DC, 1980). *Numerical Heat Transfer and Fluid Flow*. Hemisphere.

- [29] PRAKASH C. (1986). *An Improved Control Volume Finite-Element Method and Mass Transfer, and for Fluid Flow Using Equal-Order Velocity-Pressure Interpolation*. Numerical Heat Transfer. Vol. 9, pp. 253-276.
- [30] PRAKASH C. (1987). *Examination of the Upwind (Donor-Cell) Formulation in Control Volume Finite-Element Methods For Fluid Flow and Heat Transfer*. Numerical Heat Transfer. Vol. 11, pp. 401-416.
- [31] PRAKASH C. et PATANKAR S. V. (1985). *A Control Volume-Based Finite-Element Method for Solving The Navier-Stokes Equations Using Equal-Order Velocity-Pressure Interpolation*. Numerical Heat Transfer. Vol. 8, pp. 259-280.
- [32] RIDA S. (Juin, 1993). *Schéma de volumes finis pour la simulation numérique d'écoulements turbulents sur des maillages non structurés*. Mémoire de maîtrise, École polytechnique de Montréal.
- [33] ROE P.L. (1965). *An Experimental Investigation of the Flow Through Inclined Circular Tubes at Mach Number of 4.0*. R.A.E. RT. 65110.
- [34] ROE P. L. (1981). *Approximate Riemann Solvers, Parameter Vectors and Difference Schemes*. Journal of Computational Physics. Vol. 43, pp. 357-372.
- [35] RÜDE U. (Germany, 1992). *Data Structures for Multilevel Adaptive Methods and Iterative Solvers*. Institut für Informatik. Technische Universität München.
- [36] SAABAS H. J. (1991). *A Control Volume Finite Element Method for Three-Dimensional, Incompressible, Viscous Fluid Flow*. Thesis, McGill University.
- [37] SAN-YIH L. et TSUEN-MUH W. (1993). *An Adaptive Multigrid Finite-Volume Scheme for Incompressible Navier-Stokes Equations*. International Journal for Numerical Methods in Fluids. Vol. 17, pp. 687-710.
- [38] SCHNEIDER G. E. et RAW M. J. (1986). *A Skewed, Positive Influence Coefficient Upwinding Procedure for Control-Volume-Based Finite-Element Convection Diffusion Computation*. Numerical Heat Transfer. Vol. 9, pp. 1-26.

- [39] SHARIF M. A. (1993). *An Evaluation of the Bounded Directional Transportive Upwind Differencing Scheme for Convection-Diffusion Problems*. Numerical Heat Transfer. Part B, Vol. 23, pp. 201-219
- [40] SMITH R. M. et HUTTON A. G. (1982). *The numerical Treatment of Advection: A Performance Comparison of Current Methods*. Numerical Heat Transfer. Vol. 5, pp. 439-461.
- [41] TZANOS P. C. (1990). *Central Difference-Like Approximation for the Solution of Convection-Diffusion Equation*. Numerical Heat Transfer. Part B, Vol. 17, pp. 97-112
- [42] VANCE R. T. (1996). *Error Analysis of the Finite Volume Element Method for Elliptic and Parabolic Partial Differential Equations*. Thesis. University of Colorado at Denver.
- [43] VAN LEER B. (1982). *Flux Vector Splitting for the Euler Equations*. Lecture Notes in Physics. Vol 170, pp. 507-512.
- [44] VANKEIRSBILCK P. (1991). *Application of Solution Adaptive Meshes and Multigrid Methods for the Computation of Viscous Compressible Flows*. VKI Report of activities.
- [45] WAGNER E. (Germany, Februar 1995). *Numerische Simulation und Visualisierung nitchstationärer Strömungen und thermisher Effekte in inkompressiblen laminaren Medien in drei Dimensionen*. Fortgeschrittenenpraktikum, Technische Universität Mi[u]nchen.
- [46] WANG J. J. et CHEN F. (October, 1996). *On the Start Condition of the Second-Throat Ejector-Diffuser*. Aeronautical Journal.

Analyse, conception et programmation de systèmes

- [47] ABANTO J. (Montréal, 1996). *Le système VF++ v 2.0*: Guide du programmeur. Centre de recherche en calcul appliqué, CERCA.

- [48] ABANTO J. (Montréal, 1995). *Le système VF++ v 2.0: Étapes du développement*. Centre de recherche en calcul appliqué, CERCA.
- [49] ABANTO J. (Montréal, 1997). *Le système VF++ v 2.0: A Quick Class Reference*. Centre de recherche en calcul appliqué, CERCA.
- [50] ABANTO J. et CAMARERO R. (Montréal, 1997). *Le système VF++ v 2.0: Guide de l'utilisateur*. Centre de recherche en calcul appliqué, CERCA.
- [51] OZEL B. (Mars, 1995). *Un environnement configurable pour la visualisation de données*. Thèse, École polytechnique de Montréal.
- [52] DAVIS W. (1983). *Systems Analysis and Design. A Structured Approach*. Addison-Wesley Publishing Company.
- [53] DONGARRA J., POZO R. et WALKER D. (1993). *LAPACK++ : A Design Overview of Object-Oriented Extensions for High Performance Linear Algebra*. Proceedings of Supercomputing '93. IEEE Press, pp. 162-171.
- [54] MARTIN J. (1993). *Principles of Object-Oriented Analysis and Design*. Englewood Cliffs, NJ. PTR Prentice-Hall.
- [55] MUSSER D. et SAINI A. (1996). *STL. Tutorial and Reference Guide. C++ Programming with the STL*. Addison-Wesley Professional Computing Series.
- [56] POZO R. et REMINGTON K. (April, 1996). *SparseLib++ : Sparse Matrix Class Library. Reference Guide*. National Institute of Standards and Technology.
- [57] RUMBAUGH J. (1995). *Object Oriented Modelling and Design*. Prentice-Hall.
- [58] STROUSTRUP B. (MA, 1997). *The C++ Programming Language*. Addison-Wesley, 3rd edition.
- [59] STROUSTRUP B. (MA, 1994). *The Design and Evolution of C++*. Addison-Wesley, Reading.
- [60] STROUSTRUP B. (October 1995). *Why C++ is not just an Object-Oriented Programming Language*. AT&T Bell Laboratories. Addendum to OOPSLA'95 Proceedings. OOPS Messenger. Vol 6 no 4, pp 1-13.

- [61] STROUSTRUP B. (February, 1991). *What is Object-Oriented Programming?*.
Proc. 1st European Software Festival.

Annexe A

Arbres de dérivation des classes du système

Pour une information plus descriptive de l'arbre de dérivation du système se référer à [49].

A.1 Classes générales



FIGURE A.1: *Classes générales*

A.2 Classes de l'interface parser



FIGURE A.2: *Classes de vfParser*

A.3 Classes de l'interface géométrique

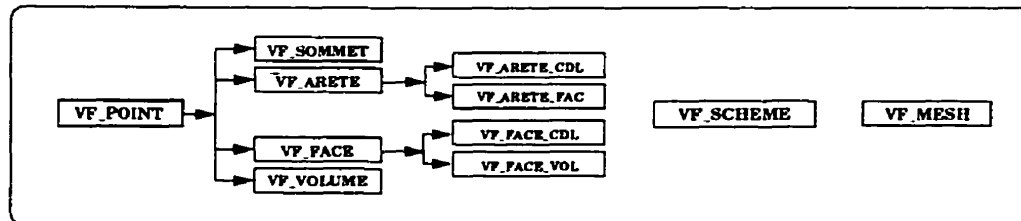


FIGURE A.3: Classes de vfGeo

A.4 Classes de l'interface résolveur

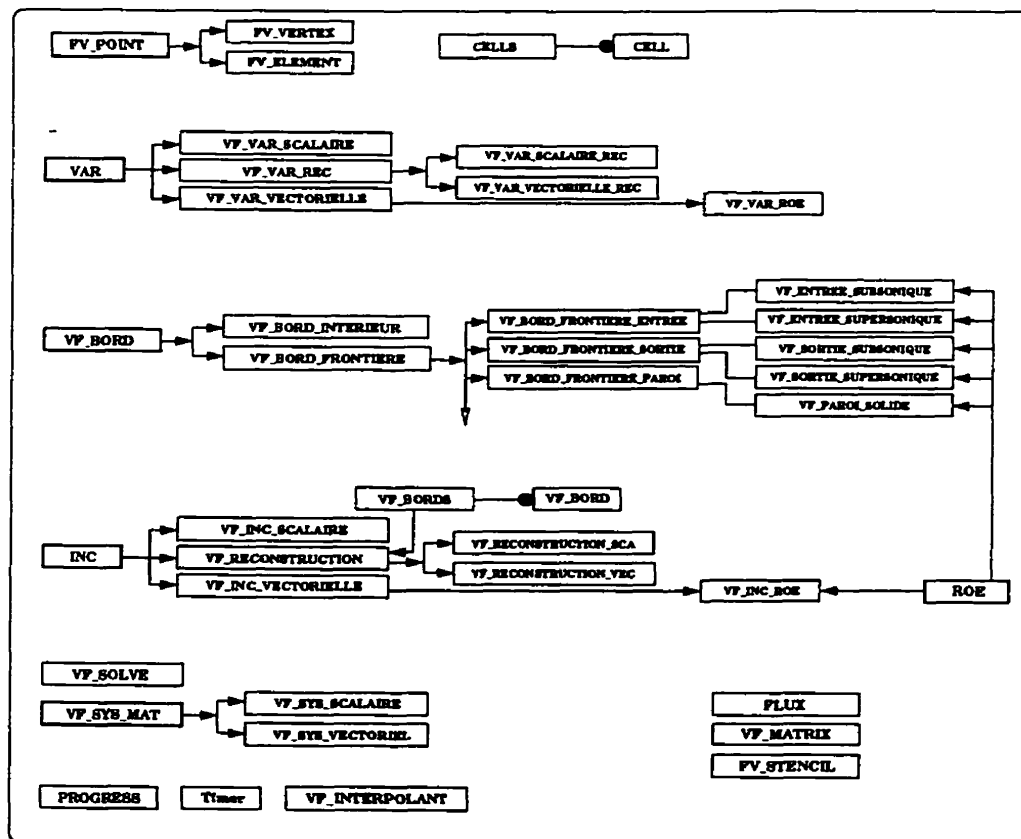


FIGURE A.4: Arbres de dérivation de vf

Annexe B

Représentation BNF de l'interface *vfParser*

B.1 Non terminaux :

```

vfParser ::= <discretization_block><EOF_tok>
discretization_block ::= <discretization_tok>[dimension]
    <lbrace_tok><discretization><rbrace_tok>
discretization ::= <coordinates_block><connectivity_block>
    [<transformation_block>]
coordinates_block ::= <nodes_tok><lbrace_tok><real_set>+<rbrace_tok>
connectivity_block ::= <connectivity>+
connectivity ::= <connect_tok><topo><lbrace_tok><int_set>+<rbrace_tok>
transformation_block ::= <transformation_tok> <lbrace_tok><transformation>+
    <rbrace_tok>
transformation ::= <scale>|<translation>|<rotation>
scale ::= <scale_tok><lparen_tok><real_tok[,real_tok,[real_tok]]><rparen_tok>
translation ::= <translation_tok><lparen_tok><real_tok[,real_tok,[real_tok]]>
    <rparen_tok>
rotation ::= <rotation_tok><lparen_tok><real_tok[,real_tok]><rparen_tok>
dimension ::= <lparen_tok><dimension_tok><rparen_tok>
topo ::= <shape>[code]
real_set ::= <real_tok><dimension_tok>
int_set ::= <integer_tok><dimension_tok>
shape ::= <lparen_tok><shape_tok><rparen_tok>
code ::= <integer_tok>

```

B.2 Terminaux :

EOF_tok ::= <EOF>
discretization_tok ::= <Discretization>|<Discretisation>|<Discretizacion>
nodes_tok ::= <Nodes>|<Noeuds>|<Nodos>
connect_tok ::= <Connect>|<Connect>|<Conect>
transformation_tok ::= <Transformation>|<Transformation>|<Transformacion>
translation_tok ::= <Translation>|<Translation>|<Translacion>
rotation_tok ::= <Rotation>|<Rotation>|<Rotacion>
dimension_tok ::= '1' '2' '3'
shape_tok ::= <<Line>|<Ligne>|<Linea>> |
 <<Triangle>|<Triangle>|<Triangulo>> |
 <<Quadrilateral>|<Quadrilatere>|<Cuadrilatero>> |
 <polygon>|
 <<Tetrahedron>|<Tetraedre>|<Tetraedro>> |
 <<Prism>|<Prisme>|<Prisma>> |
 <<Hexahedron>|<Hexaedre>|<Hexaedro>> |
 <<Pyramid>|<Piramide>|<Piramide>>
polygon ::= <polygon_tok><comma_tok><integer_tok>
polygon_tok ::= <Polygon>|<Polygone>|<Poligono>
real_tok ::= [sign_tok] <ordinary_float>
real_tok ::= <exponential_float>
ordinary_float ::= <digits>+.<digits>+
ordinary_float ::= <digits>+.
ordinary_float ::= <digits>+
exponential_float ::= [sign_tok] <ordinary_float> <E | e> [sign_tok] <integer_tok>
integer_tok ::= <digits>+
digits ::= '0' '1' '2' '3' '4' '5' '6' '7' '8' '9'
sign_tok ::= '+' '-'
lbrace_tok ::= '{'
rbrace_tok ::= '}'
lparen_tok ::= '('
rparen_tok ::= ')'
comma_tok ::= ','

Annexe C

Relations entre les entités topologiques

Des exemples de relations entre les entités topologiques de la discrétisation initiale sont montrées par le tableau C.1, et de la discrétisation finale dans le tableau C.2. Ces deux types de relations sont une représentation de la figure 4.3 et sont basés sur les Définitions 4.1,...,4.7.

TABLEAU C.1: *Relations initiales entre entités*

RELATION	SOUS-ENSEMBLE DE	DESCRIPTION
<i>is-neighbor-of</i>	$F \times A$	faces partageant la même arête
<i>is-neighbor-of</i>	$F \times S$	faces partageant le même sommet
<i>is-neighbor-of</i>	$V \times S$	volumes partageant le même sommet
<i>is-vertex-of</i>	$S \times F$	sommets de la face
<i>is-edge-of</i>	$A \times F$	arêtes de la face
<i>is-vertex-of</i>	$S \times V$	sommets du volume
<i>is-edge-of</i>	$A \times V$	arêtes du volume
<i>is-face-of</i>	$F \times V$	faces du volume

TABLEAU C.2: *Relations finales entre entités*

RELATION	SOUS-ENSEMBLE DE	DESCRIPTION
<i>is-neighbor-of</i>	$S \times S$	sommets connectés par un <i>bord</i>
<i>is-neighbor-of</i>	$B \times B$	<i>bords</i> partageant le même sommet
<i>is-neighbor-of</i>	$E \times E$	<i>éléments</i> partageant le même <i>bord</i>
<i>is-endpoint-of</i>	$S \times B$	sommets des <i>bords</i>
<i>is-boundary-of</i>	$B \times E$	<i>bords</i> des <i>éléments</i>
<i>is-vertex-of</i>	$S \times E$	sommets des <i>éléments</i>
<i>is-boundary-between</i>	$B \times E \times E$	un <i>bord intérieur</i>
<i>is-boundary-between</i>	$B \times E \times \emptyset$	un <i>bord-frontière</i>
<i>has-vertices</i>	$S \times S \times \dots S_n$	un <i>élément</i>
<i>has-vertices</i>	$S \times \dots S_m$	un <i>bord</i>

Annexe D

Numérotation locale des arêtes et des faces des formes géométriques en 3D

Les tétraèdres, prismes, hexaèdres et autres formes géométriques sont des volumes *orientables*. La numérotation locale rapportée à la figure 4.6 est illustrée par les tableaux D.1,...,D.3. Les faces sont orientées et construites en considérant une direction vers l'extérieur par rapport à la forme géométrique selon la règle de Moëbius¹.

TABLEAU D.1: *Numérotation locale des entités d'un tétraèdre*

FACES	Face 1	(S_2, S_3, S_4)
	Face 2	(S_1, S_4, S_3)
	Face 3	(S_1, S_2, S_4)
	Face 4	(S_1, S_3, S_2)
ARÊTES	Arête 1	$(S_2, S_4) \vee (S_4, S_2)$
	Arête 2	$(S_3, S_4) \vee (S_4, S_3)$
	Arête 3	$(S_1, S_4) \vee (S_4, S_1)$
	Arête 4	$(S_1, S_2) \vee (S_2, S_1)$
	Arête 5	$(S_2, S_3) \vee (S_3, S_2)$
	Arête 6	$(S_1, S_3) \vee (S_3, S_1)$

¹ Pendant la numérotation, le parcours de la direction sur une arête limitant deux faces adjacentes doit être dans le sens contraire.

TABLEAU D.2: Numérotation locale des entités d'un prisme

FACES	Face 1	(S_1, S_4, S_5, S_2)
	Face 2	(S_2, S_5, S_6, S_3)
	Face 3	(S_1, S_3, S_6, S_4)
	Face 4	(S_1, S_2, S_3)
	Face 5	(S_4, S_6, S_5)
ARÊTES	Arête 1	$(S_1, S_2) \vee (S_2, S_1)$
	Arête 2	$(S_2, S_3) \vee (S_3, S_2)$
	Arête 3	$(S_3, S_1) \vee (S_1, S_3)$
	Arête 4	$(S_1, S_4) \vee (S_4, S_1)$
	Arête 5	$(S_2, S_5) \vee (S_5, S_2)$
	Arête 6	$(S_3, S_6) \vee (S_6, S_3)$
	Arête 7	$(S_4, S_5) \vee (S_5, S_4)$
	Arête 8	$(S_5, S_6) \vee (S_6, S_5)$
	Arête 9	$(S_6, S_4) \vee (S_4, S_6)$

TABLEAU D.3: Numérotation locale des entités d'un hexaèdre

FACES	Face 1	(S_1, S_4, S_3, S_2)
	Face 2	(S_1, S_2, S_6, S_5)
	Face 3	(S_1, S_5, S_8, S_4)
	Face 4	(S_2, S_3, S_7, S_6)
	Face 5	(S_3, S_4, S_8, S_7)
	Face 6	(S_5, S_6, S_7, S_8)
ARÊTES	Arête 1	$(S_1, S_2) \vee (S_2, S_1)$
	Arête 2	$(S_2, S_3) \vee (S_3, S_2)$
	Arête 3	$(S_3, S_4) \vee (S_4, S_3)$
	Arête 4	$(S_4, S_1) \vee (S_1, S_4)$
	Arête 5	$(S_1, S_5) \vee (S_5, S_2)$
	Arête 6	$(S_2, S_6) \vee (S_6, S_2)$
	Arête 7	$(S_1, S_7) \vee (S_7, S_1)$
	Arête 8	$(S_4, S_8) \vee (S_8, S_4)$
	Arête 9	$(S_5, S_6) \vee (S_6, S_5)$
	Arête 10	$(S_6, S_7) \vee (S_7, S_6)$
	Arête 11	$(S_7, S_8) \vee (S_8, S_7)$
	Arête 12	$(S_8, S_5) \vee (S_5, S_8)$

Annexe E

Tableaux de construction des bords de contrôle

La construction des bords de contrôle en 2D et 3D rapportée à la Définition 4.9, ainsi qu'aux Règles générales 4.1,...,4.7 et à la figure 4.2 suit les indications illustrées par les tableaux E.1,...,E.4¹.

TABLEAU E.1: *Construction des bords internes en 2D*

SCHÉMA	VOLUME	TYPE	SÉQUENCE DE CONSTRUCTION
VCCV	sommet	arête	<i>centroïde</i> $\Rightarrow$ <i>arête</i>
ECCV	arête	arête	<i>centroïde</i> $\Rightarrow$ <i>sommet</i>
CCCV	face	arête	<i>sommet</i> ₁ $\Rightarrow$ <i>sommet</i> ₂

TABLEAU E.2: *Construction des bords-frontières en 2D*

SCHÉMA	VOLUME	TYPE	SÉQUENCE DE CONSTRUCTION
VCCV	sommet	arête	<i>arête</i> $\Rightarrow$ <i>sommet</i> <i>sommet</i> $\Rightarrow$ <i>arête</i>
ECCV	arête	arête	<i>sommet</i> ₁ $\Rightarrow$ <i>sommet</i> ₂
CCCV	face	arête	<i>sommet</i> ₁ $\Rightarrow$ <i>sommet</i> ₂

¹ Dans ces tableaux les mots *«arête»* et *«face»* font référence au centre de l'arête et au centroïde de la face, respectivement.

TABLEAU E.3: *Construction des bords internes en 3D*

SCHÉMA	VOLUME	TYPE	SÉQUENCE DE CONSTRUCTION
VCCV	sommet	face (quadrilatère)	$arête \Rightarrow face_1 \Rightarrow centroïde \Rightarrow face_2$
ECCV	arête	face (triangle)	$sommet \Rightarrow face \Rightarrow centroïde$
FCCV	face	face (triangle)	$sommet_1 \Rightarrow centroïde \Rightarrow sommet_2$
CCCV	volume	face (polygone)	$sommet_1 \Rightarrow sommet_2 \Rightarrow \dots sommet_n$

TABLEAU E.4: *Construction des bords-frontières en 3D*

SCHÉMA	VOLUME	TYPE	SÉQUENCE DE CONSTRUCTION
VCCV	sommet	face (quadrilatère)	$sommet \Rightarrow arête_1 \Rightarrow face \Rightarrow arête_2$
ECCV	arête	face (triangle)	$sommet_1 \Rightarrow face \Rightarrow sommet_2$
FCCV	face	face (polygone)	$sommet_1 \Rightarrow sommet_2 \Rightarrow \dots sommet_n$
CCCV	volume	face (polygone)	$sommet_1 \Rightarrow sommet_2 \Rightarrow \dots sommet_n$

Annexe F

Variables et inconnues

F.1 Les classes *VAR* et *INC*

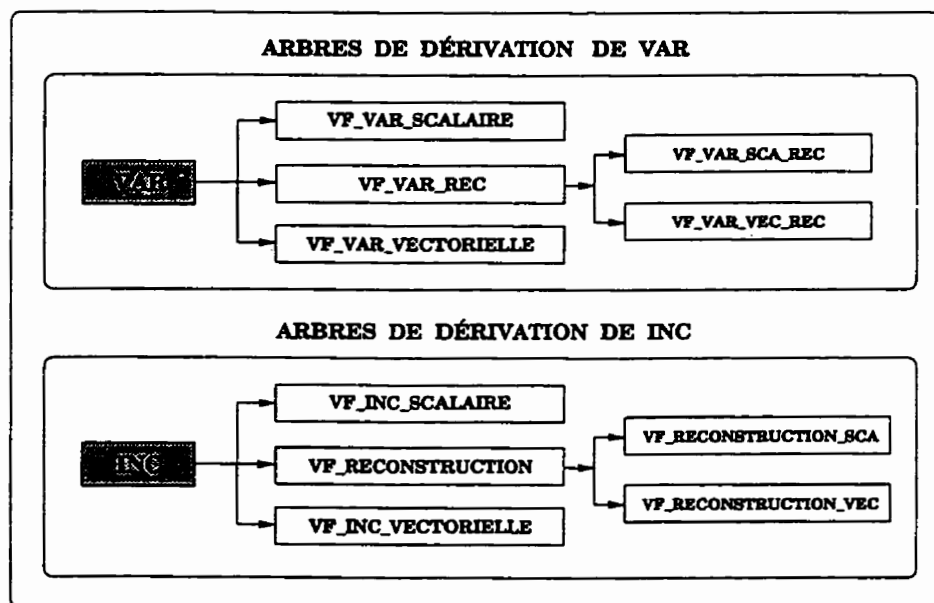


FIGURE F.1: *Les classes VAR, INC et leur descendance*

F.2 La classe *VF_INC_SCALAIRE*

Voir le code F.1 et sa description dans la sous-section 6.2.2.

Code F.1 *La classe VF_INC_SCALAIRE*

```

class VF_INC_SCALAIRE: public INC
{
    ...
    REAL valeur_inc;
    VF_Type_cdl cdl;
public:
    VF_EQUATION eq;
    VF_INC_SCALAIRE()
        :INC(),valeur_inc(0.0),cdl(VF_cdl_interne) {};
    ...
};

```

F.3 Accès aux inconnues

F.3.1 Accès à partir d'un élément

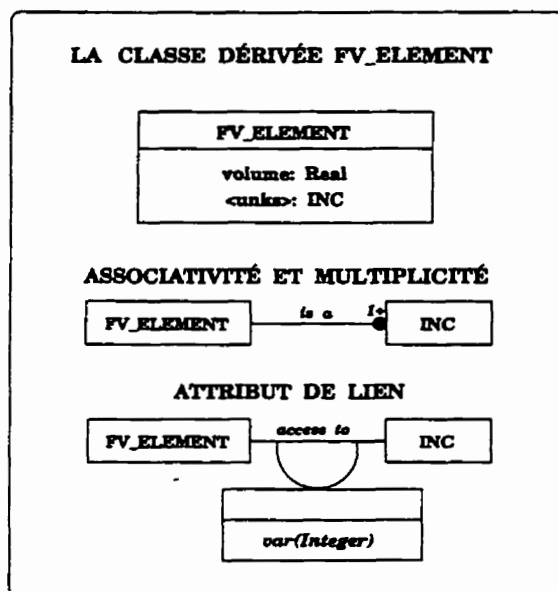


FIGURE F.2: Accès à une inconnue à partir d'un élément

Le stockage d'une inconnue dans l'élément, traité dans la sous-section 6.2.1, respecte le principe de l'unicité de volume *versus* la multiplicité des inconnues stockées dans le volume de contrôle. Cette approche est d'un grand avantage car elle permet l'utilisation d'un seul volume de contrôle, défini dans le *kernel* de *vf*. Celui-ci peut réserver un espace de mémoire pour une ou plusieurs inconnues de façon dynamique. Ces caractéristiques sont illustrées par la figure F.2.

Un élément peut accéder à une inconnue résidant dans une position *i* du tableau d'*inconnues* à l'aide de la méthode *var(INT i)* appartenant à la classe *FV_ELEMENT*¹. Ainsi, une inconnue étant définie comme scalaire peut être récupérée à l'aide de cette méthode. C'est le cas, par exemple, de la récupération des inconnues scalaires gauche *inc.inc_g* et droite *inc.inc_d*, dans le code H.2 et code F.2.

¹ Voir le code 6.1 de la page 67.

Cet opérateur est très fort.

Code F.2 Accès à une inconnue à partir d'un élément

```
VF_INC_SCALAIRE* inc_inc_g=(VF_INC_SCALAIRE*) mg->var(idx);
VF_INC_SCALAIRE* inc_inc_d=(VF_INC_SCALAIRE*) md->var(idx);
```

F.3.2 Accès à partir d'une variable de référence

Entre une variable Φ (ou Φ) et une inconnue Φ_i (ou Φ_i) de la même espèce², il existe une *association un à plusieurs*. L'accès d'une variable primitive à une inconnue se réalise de façon biunivoque. Ce processus est illustré dans la figure F.3.

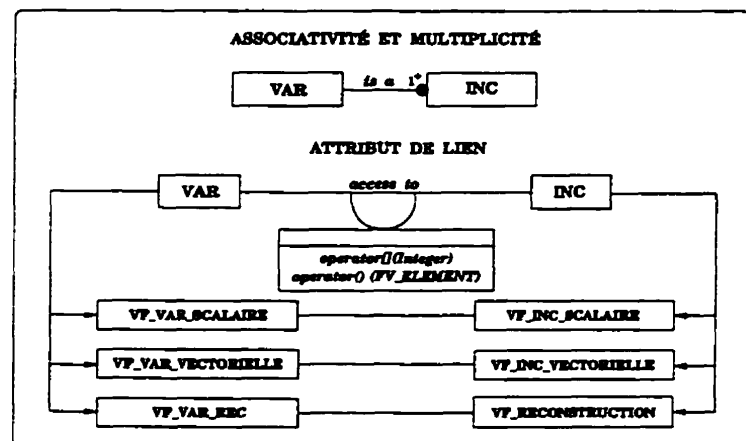


FIGURE F.3: Associativité entre variables et inconnues

Si on est sur un objet référençant une variable Φ (ou Φ), cet accès est concrétisé à l'aide de deux opérateurs :

- **Accès avec l'opérateur $[]$ (int)** : si on connaît la position i dans la liste des éléments de la variable (voir la sous-section 6.2.2) on utilise l'opérateur crochet. *Exemple*: le code 6.5 de la page 79, réécrit dans le code F.3, utilise cet opérateur afin d'accéder à toutes les inconnues Φ_i .
Cet opérateur est fort.
- **Accès avec l'opérateur $()$ (FV_ELEMENT)** : si on connaît l'élément là où l'inconnue habite, on utilise l'opérateur parenthèse. *Exemple*: le code F.4 fera exactement la même chose que le code F.2.

² Voir la sous-section 6.2.2.

Code F.3 *Accès à une inconnue à partir d'une variable et de son index*

```
forAllVolumes
{
    REAL Vol=inconnues[i]->volume();
    // use of [] operator
    (*this)[i]->applique_correction(dt/Vol);
}
```

Cet opérateur est faible.

Code F.4 *Accès à une inconnue à partir d'une variable et de son élément*

```
// use of () operator
VF_INC_SCALAIRE* inc_inc_g=(VF_INC_SCALAIRE*) var(mg);
VF_INC_SCALAIRE* inc_inc_d=(VF_INC_SCALAIRE*) var(md);
```

Annexe G

Application du schéma de *Roe* avec le PROTOCOLE 1

G.1 Le schéma de *Roe*

Le *<Roe's splitter>* dans le calcul du flux convectif $\mathbf{H}$ de l'équation (6.6) permet d'établir sur un bord quelconque :

$$\mathbf{H}(\mathbf{U}_L, \mathbf{U}_R) = \frac{1}{2} [\mathbf{H}(\mathbf{U}_L) + \mathbf{H}(\mathbf{U}_R) - \mathbf{H}_c(\mathbf{U}_L, \mathbf{U}_R)] \quad (\text{G.1})$$

où $\mathbf{H}_c$, est le flux de correction (*voir* implantation dans le code 6.4) qui est fonction du jacobien $\frac{\partial \mathbf{H}}{\partial \mathbf{U}}$ calculé avec les valeurs moyennes de *Roe*[34].

G.2 Structure des données principales

Étant donné le caractère vectoriel du problème, la variable $\mathbf{U}$ est bâtie à l'aide de la classe *VF_VAR_ROE* dérivant de *VF_VAR_VECTORIELLE*. Une inconnue de *Roe* est créée avec une classe *VF_INC_ROE*, dérivant de *VF_INC_VECTORIELLE* et *ROE*. Cette application est illustrée par le diagramme de la figure G.1.

Les mêmes associations et mécanismes d'accès (sous-annexe F.3) sont utilisés.

G.3 Stratégies du calcul des flux

La classe *ROE* de la figure G.1 sert d'interface virtuelle dans le calcul du flux $\mathbf{H}(\mathbf{U})$ de l'équation (G.1) selon deux critères :

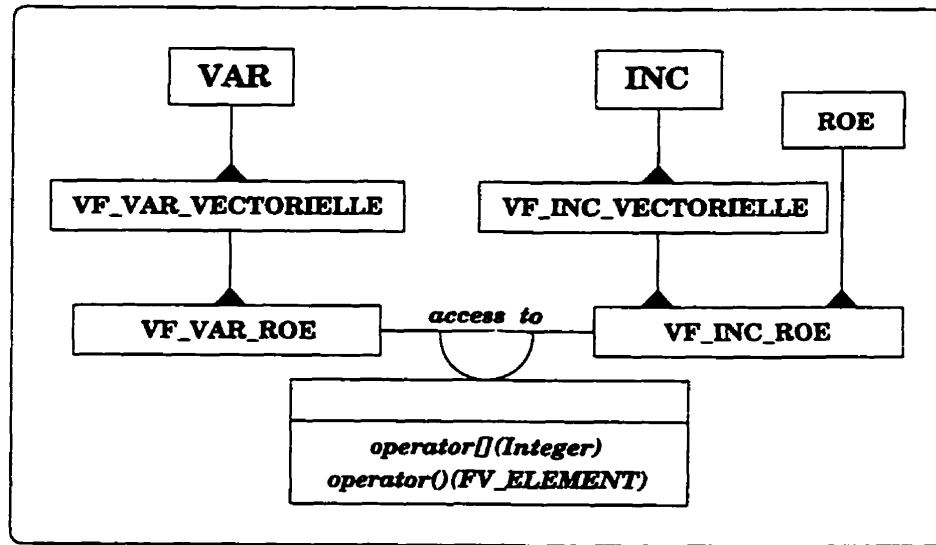


FIGURE G.1: La variable et inconnue de Roe

1. Le calcul de flux $\mathbf{H}(\mathbf{U})$ à l'intérieur du domaine ayant comme support un bord intérieur partageant l'information avec les inconnues gauche et droite¹ en utilisant l'équation (G.1), (*association un à deux*).
2. Le même calcul de flux à la frontière du domaine de calcul ayant comme support un bord-frontière partageant l'information avec l'inconnue gauche², (*association un à un*).

Ce calcul se réalise de façon transparente par la classe *VF_BORD_FRONTIERE* en passant, comme composante droite *R*, un miroir de l'élément-frontière (*voir la sous-section 6.2.1*).

Si la physique du problème exige l'application de conditions aux frontières plus spécifiques, cette imposition se fait tout en conservant l'*association un à un* entre une inconnue et un bord, dans un objet des classes :

- *VF_ENTREE_SUBSONIQUE* servant à imposer les conditions aux frontières subsoniques à l'entrée. Un objet de celle-ci est associé avec un de la classe *VF_BORD_FRONTIERE_ENTREE*.
- *VF_ENTREE_SUPERSONIQUE* *Idem* pour une entrée supersonique.
- *VF_SORTIE_SUBSONIQUE* servant à imposer les conditions subsoniques à la sortie, associé avec un objet de la classe *VF_BORD_FRONTIERE_SORTIE*.
- *VF_SORTIE_SUPERSONIQUE* *Idem* pour une sortie supersonique.
- *VF_PAROI_SOLIDE* Associée avec la classe *VF_BORD_FRONTIERE_PAROI*, où on impose les conditions aux frontières à la paroi.

Cette liste de conditions, de même que le type de bord-frontière peut augmenter en fonction des applications.

¹ Règle générale 4.6.

² Règle générale 4.7.

La disposition des classes de *ROE* et associations sont illustrées par la figure G.2.

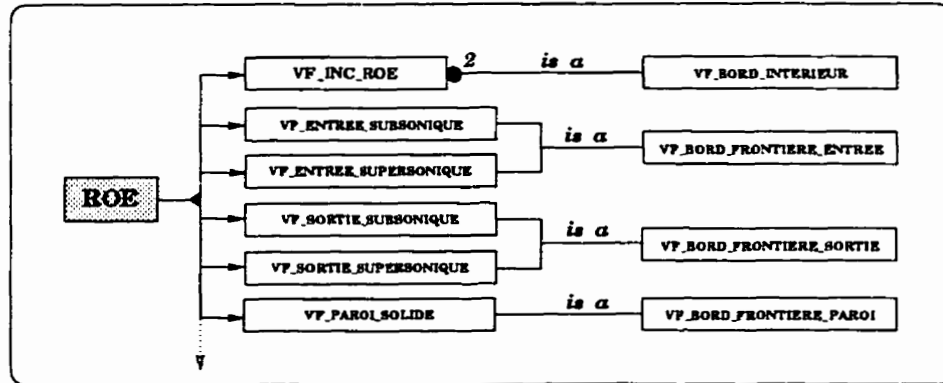


FIGURE G.2: *L'interface virtuelle ROE et associations*

G.4 Imposition des conditions aux frontières dans le PROTOCOLE 1

Cette imposition est gouvernée par les classes dérivant de *ROE* illustrée par la figure G.2 et est basée sur l'approche décrite dans le paragraphe précédent. Du point de vue numérique, elle est basée fondamentalement dans la thèse [18] pour la résolution explicite ou implicite.

G.5 Le programme principal de l'application

L'intégration explicite de l'équation (6.6) se réalise dans un processus itératif jusqu'à la convergence en satisfaisant les conditions de stabilité numérique CFL (*Courant-Friedrichs-Levy*), selon le code G.1. Ce processus a pu être accompli d'une autre manière en appliquant l'intégration *Runge-Kutta* dans le temps selon [19] en respectant le critère de viscosité numérique.

Le programme principal montre tous les processus faisant partie de l'interface résolveur pour cette application. La description du programme principal suit l'ordre :

1. `inFile`, reçoit le nom du fichier contenant les conditions initiales (voir la figure 2.1).
2. `VF_VAR_ROE U("U", varnames, cin, inFile)`, crée un objet variable de *Roe* englobant toutes les variables conservatives. L'argument `cin` permet la réception dans le *standard input* des bords de contrôle amorphes, à l'aide de la méthode *receiveBords* de la classe. Création de bords *B*, volumes de contrôle *E*, sommets *S* et inconnues

Code G.1 *Programme principal dans l'intégration explicite*

```

...
const char* inFile=argv[1];

const char* varnames[]={"rho","rhoH","rhoV","rhoW","rhoE"};

VF_VAR_ROE U("U",varnames,cin,inFile);

Timer time;
while(True)
{
    REAL re=0,res=0;
    REAL dt=U.calculer_temps()*CFL;
    U.initialiser_correction(); // Loop on all elements
    U.integre();                // Loop on all sides
    U.appliquer_correction(-dt); // Loop on all elements
    U.residu(re,res);
    ...
};
...
cout<<"CPU time: "; time.cpu(); cout<<"\n";
cout<<"Total time: "; time.total(); cout<<"\n";
U.print();                // Save the results
...

```

associées à la variable primitive. Attribution de *guess values* aux inconnues. Ce processus correspond à l'ellipse foncée de la figure 2.1.

3. **time**, un objet de la classe support *Timer* permettant d'initialiser le *CPU time*.

Boucle itérative :

- (a) Calcul du pas de temps global $\Delta\tau$ avec la méthode *calcule_temps* en bouclant sur les bords. Le temps calculé est évalué à partir des valeurs moyennes des paramètres de l'élément gauche et droit. La valeur finale est pénalisée avec la valeur la plus petite sur tous les bords $\mathbb{B}$, multipliée par le coefficient *CFL*.
 - (b) Mise à zéro des termes *buffer* de correction correspondant à l'expression droite de l'équation (6.6), $\sum_{j=1}^m \mathbf{H}_j^n$, par la méthode *initialise_correction*.
 - (c) Intégration en bouclant sur tous les bords $\mathbb{B}$, faite par le méthode *integre*. L'intégration sur chaque bord se fait selon le code 6.4 de la page 79.
 - (d) Intégration dans le temps. Boucle sur tous les éléments $\mathbb{E}$, réalisée par la méthode *applique_correction(Real)* selon le code 6.5 et l'équation (G.1).
 - (e) Calcul du critère d'arrêt avec la méthode *residu(Real $\mathcal{E}$, Real $\mathcal{E}$)*. La première valeur de l'argument correspond au résidu basé sur la densité ρ , la deuxième valeur est basée sur la valeur maximale de la correction de toutes les variables conservatives dans le domaine de calcul.
4. Écriture du *CPU time* et *Total time*.
 5. La méthode *print* permet la sortie des résultats vers un fichier **.SOL* (voir la figure 2.1) , selon le format décrit dans l'annexe K servant de source pour la visualisation avec *vu*.

Annexe H

Application des schémas décalés dans la résolution des équations de *Navier-Stokes* en régime incompressible avec le PROTOCOLE 2

H.1 Intégration générique avec le PROTOCOLE 2

L'intégration se réalise selon le code H.1 pour une disposition de la variable autre que CCCV.

H.2 Distribution générique

Le processus de distribution pour le PROTOCOLE 2 suit le code H.2, dans lequel J est la *matrice des flux*. La formation des équations se fait en accédant à la méthode *add_coef* de la classe *VF_EQUATION*, un objet de laquelle, *eq*, appartient à l'inconnue en cours (sous-section 6.2.2).

Code H.1 *L'intégration générique*

```

...
for(INT i=0;i<nbordsInterieurs;i++)
{
    VF_VECTEUR n = normale(i);
    REAL ds = norme(i);
    VF_VECTEUR v = vitesse(i);
    ...
    for(INT j=0;j<nbTermes;j++)
    {
        VF_VECTEUR g;
        REAL phi=0;
        for(INT k=0;k<nt;k++)
        {
            phi += ... //phi(i,j,k)
            for(INT l=0;l<DIM;l++) g += ... //g(i,j,k,l)
        };
        VF_VECTEUR flux=rho*v*phi-Gamma*g;
        J(i,j) = flux.dot(n*ds);
    };
};

```

Code H.2 *La distribution avec le PROTOCOLE 2*

```

void VF_RECONSTRUCTION_SCA::distribue(const VF_MATRIX& J)
{
    INT idx=var_originale.novar();
    const INT nbsc=nsidInt();
    for(INT i=0;i<nbsc;i++)
    {
        FV_ELEMENT* mg=volume(i,0);
        FV_ELEMENT* md=volume(i,1);

        VF_INC_SCALAIRE* inc_inc_g=(VF_INC_SCALAIRE*) mg->var(idx);
        VF_INC_SCALAIRE* inc_inc_d=(VF_INC_SCALAIRE*) md->var(idx);

        for(INT j=0;j<nbtermes;j++)
        {
            FV_ELEMENT* m=stencil[j];

            VF_INC_SCALAIRE* inc_inc=(VF_INC_SCALAIRE*) m->var(idx);

            inc_inc_g->eq.add_coef(+J(i,j),inc_inc);
            inc_inc_d->eq.add_coef(-J(i,j),inc_inc);
        }
    }
};

```

H.3 Création de la matrice de convection avec le schéma MWUS pour les polygones

La *matrice de convection* $[C]$ pour les polygones, rapportée à l'équation (6.9) de la page 89 pour le schéma MWUS, est créée par les lignes de code générique (code H.3) et suit la description de la figure H.1.

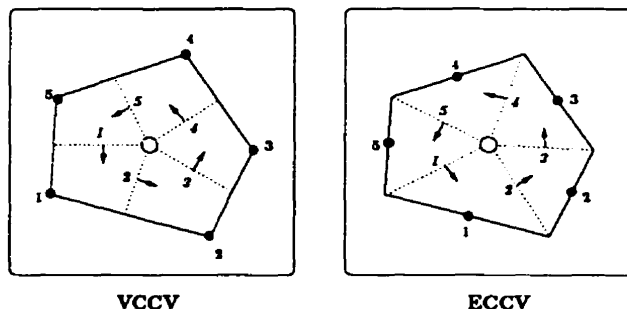


FIGURE H.1: Description de l'application de MWUS sur un pentagone

H.4 Le programme principal de l'application

La description du programme principal (code H.4) suit l'ordre :

1. `VF_VAR_VECTORIELLE v("v",3,vit)`, crée une variable vectorielle pour le vecteur vitesse $V = \{u, v, w\}$.
2. `VF_VAR_VECTORIELLE_REC v_rec("v_rec",v)`, crée une variable projection vectorielle associée à la variable primitive v .
3. `VF_VAR_SCALAIRE p("p")`, crée une variable scalaire pour la pression.
4. `VF_VAR_SCALAIRE_REC p_rec("p_rec",p)`, crée une variable projection scalaire associée à la variable primitive p .
5. `v_rec.receiveBords(cin)`, réception de bords amorphes, provenant du *standard input*, avec un appel à la méthode *receiveBords*. Création de bords B , volumes de contrôle E , sommets S et inconnues associées à la variable primitive. Attribution des *guess values* aux inconnues. Tout ce processus correspond à l'ellipse foncée de la figure 2.1 de la page 20.
6. `p_rec.receiveBords(cin)`, *Idem*, pour les volumes de contrôle de la pression.
7. `VF_SYS_VECTORIEL sysmatv(v)`, préparation des interfaces pour la résolution de v .
8. `VF_SYS_SCALAIRE sysmatp(p)`, *Idem*, pour la pression.

Code H.3 *Génératrice de la matrice de convection pour le schéma MWUS pour des polygones*

```

...
VF_MATRIX A(nbsc,nsc),B{nbsc,nt);
for (i=0;i<nbsc;i++)
{
    INT iapp=(i==nbsc-1)?0:i+1;
    INT iamm=(i==0)?nbsc-1:i-1;
    INT ibmm=(i==0)?nt-1:i-1;

    INT ka=(rhounds[i]>0.0)?iapp:iamm;
    INT kb=(rhounds[i]>0.0)?i:ibmm;

    REAL f=VF_min(VF_max(rhounds[ka]/rhounds[i],zero),one);

    A.set(i,ka,-f);
    A.set(i,i,1);

    B.set(i,kb,1-f);
};
A.inverse();

VF_MATRIX C=A*B;
...

```

Code H.4 *Exemple de programme principal dans l'intégration des équations de Navier-Stokes en régime incompressible avec la méthode SIMPLER*

```

...
const char* vit[]={"u","v","w"};
VF_VAR_VECTORIELLE v("v",3,vit);
VF_VAR_VECTORIELLE_REC v_rec("v_rec",v);
VF_VAR_SCALAIRE p("p");
VF_VAR_SCALAIRE_REC p_rec("p_rec",p);
v_rec.receiveBords(cin);
p_rec.receiveBords(cin);
...
VF_SYS_VECTORIEL sysmatv(v);
VF_SYS_SCALAIRE sysmatp(p);
Timer time;
do
{
    v_rec.FOUS();

    p_rec.Poisson(v);

    sysmatp.SLUSS();

    v_rec.GradientLSQ();

    sysmatv.SLUSS();
    ...
} while(True);

cout<<"CPU time: "; time.cpu(); cout<<"\n";

v.print(); p.print();
...

```

9. `time`, un objet de la classe de support *Timer* permettant d'initialiser le *CPU time*.

Boucle itérative :

- (a) `v_rec.FOUS()`¹ Initialisation et détermination des coefficients de l'équation (6.1) pour les vitesses, en employant le schéma *upwind FOUS*,
- (b) `p_rec.Poisson(v)` Initialisation et détermination des coefficients de l'équation (6.1) pour la pression, en utilisant les pseudo-vitesses, équation (6.11), dans l'équation de la pression,
- (c) `sysmatp.SLUSS()` résolution de la matrice de la pression par la méthode *SLUSS* de la classe *VF_SYS*,
- (d) `v_rec.GradientLSQ()`², calcul du gradient de la pression à l'aide de la technique des moindres carrés et insertion de ces valeurs dans l'équation du mouvement,
- (e) `sysmatv.SLUSS()` résolution de la matrice de la vitesse.

10. Écriture du *CPU time*.

11. Impression des résultats de *u* et *p* dans deux fichiers **.SOL* (voir la figure 2.1), selon le format décrit dans l'annexe K, pour la visualisation avec *vu*.

¹ Ou `v_rec.MWUS()`, si on applique le schéma *MWUS*.

² Ou `v_rec.GradientINT()`, si on applique des fonctions d'interpolation.

Annexe I

Les outils de *VF++*

Les outils de *VF++* servent de support au système pendant l'exploitation. Pour une information plus détaillée sur leur mode d'emploi consultez [50].

I.1 Les fichiers du système *VF++*

Cette version de *VF++* est installée dans le répertoire de domaine public¹ dans `/usr/local/vf/`, contenant les outils et l'information de base du système :

- **Librairies:** Quatre : *vfParser.a*, *vfGeo.a*, *vfSender.a* et *vf.a*.
- **Exécutables:** Six : les *scripts* : *VF*, *VFM*, le *VF2VU* et les programmes : *vf2vu*, *Grid* et *Vie*.
- **Includes:** Les *include files* de *vfGeo*.
- **Data:** Avec exemples de description du maillage initial en format *Vie*.
- **Manuels:** Le guide : "Le système *VF++* v 2.0: A quick class reference" format postscript *VFClass.ps* ([49]).

I.1.1 Le script *VF*

Ce *script* est un exemple pour le *fonctionnement continu*² du système et suit sa règle syntactique, pour le PROTOCOLE 1 :

VF < *vieFile* > < *cvType* >

¹ Pour les usagers du CERCA et de l'École Polytechnique.

² sous-section 2.2.1.

Ici, le $\langle cvType \rangle$ est l'identificateur de type de volume de contrôle, selon le tableau 4.1.

I.1.2 *Le script VFM*

Ce *script* est un exemple pour le *fonctionnement différé*³ du système pour tous les types de stockage dans un étude paramétrique avec sa syntaxe :

VFM $\langle vieFile \rangle$

I.1.3 *Le script VF2VU*

Programme utilisant la solution stockée dans un fichier **.SOL*, provenant de l'interface résolveur (voir la chapitre 2 et la figure 2.1), et la discrétisation initiale correspondante pour écrire un fichier de visualisation en format *vu*.

Le fichier sortant de *VF2VU* est basé sur la visualisation discrète de la solution et le maillage de [co[co]]volumes de contrôle. Aucun décalage de la solution est fait par *VF++*.

I.1.4 *Le programme vf2vu*

Idem, programme appelé par le *script VF2VU*.

I.1.5 *Le programme Grid*

Ce programme sert à écrire un fichier avec le maillage final en format *vu*. Le fichier sortant de *Grid* ne visualise que le maillage final à partir des bords de contrôle. Ce programme écrit automatiquement un fichier dont le nom tient compte du nom du fichier *Vie* et du type de stockage.

Ce programme est très utile pour la visualisation d'un maillage initial venant d'être créé.

³ sous-section 2.2.2.

I.1.6 *Le programme Vie*

Le programme *Vie* sert à écrire un fichier en format *Vie* à partir de la dernière version de *mars*.

Quelque soit l'outil pour créer un fichier en format *Vie*, il faut suivre les règles de syntaxe de la grammaire exposée dans le chapitre 3.

I.2 Les répertoires de l'utilisateur

Les répertoires de *VF++*, créés dans *\$HOME/* de l'utilisateur, contiennent de l'information additionnelle et sont montrées par la figure I.1.

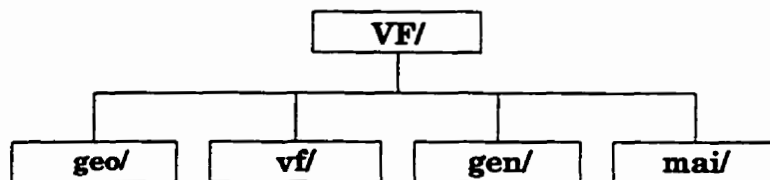


FIGURE I.1: *Répertoires du système*

Ici, les répertoires :

- **geo/** : contient de l'information additionnelle pour l'interface géométrique ;
- **vf/** : *Idem*, pour l'interface résolveur ;
- **gen/** : *Idem*, communes aux interfaces ;
- **mai/** : *Idem*, avec des exemples de maillage initial.

Annexe J

Les messages d'erreur

Le système *VF++* possède de *errors handlers* permettant d'envoyer au *standard error* des erreurs ou *warnings* pendant son exécution. Les erreurs les plus fréquentes se trouvent lors de la création du fichier de la discrétisation initiale *Vie*.

Regardons deux exemples, avec le fichier de description listé dans la section 3.3, appelé *Car.vie* dont sa représentation est illustrée par la figure 3.1(a) :

1. Erreur provenant de *vfParser*. Dans la ligne 16, à la place d'écrire *Connect (Poligono,5)*, l'utilisateur a écrit : *Connect (Poligono)*, à ce moment-là le *vfParser* est chargé d'envoyer le message au *standard error* :

```
Error in file Car.vie, 16:20
Token ',' expected and ')' found instead.
```

Effectivement l'erreur a été détectée à la ligne 16 et à la colonne 20.

2. Erreur provenant de *vfGeo*. Si on laisse sans définition les deux frontières avec le code 5, même si le *vfParser* passe bien la description, la géométrie n'est pas bien fermée. À ce moment-là, le *vfGeo* est chargé d'envoyer des *warnings* au *standard error* :

```
WARNING: Suspicious boundary without code, see:
0x200050a8:1:4
Face(0x20004928,3):(1,2,4)
WARNING: Suspicious boundary without code, see:
0x200054b8:4:7
Face(0x20004c68,4):(4,5,8,7)
```

Ce qui indique qu'il y a deux frontières non-définies (voir la figure 3.1(a)), la première (l'arête 1 4) qui fait partie de la face triangulaire (1,2,4), et l'autre (4 7) du quadrilatère (4,5,8,7).

Annexe K

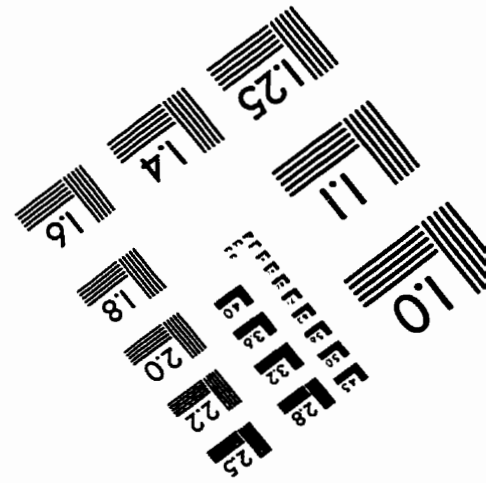
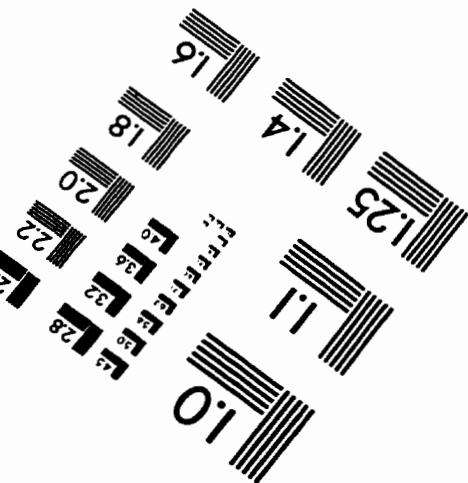
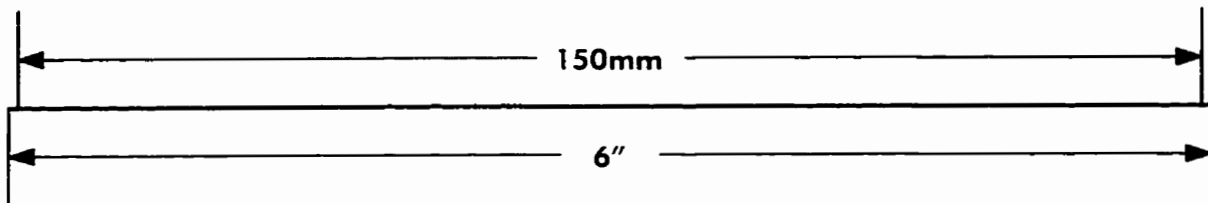
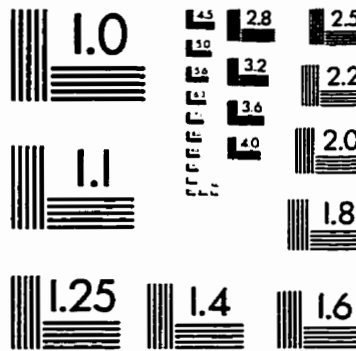
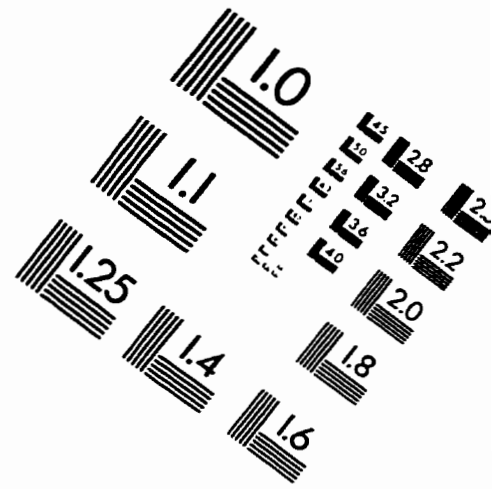
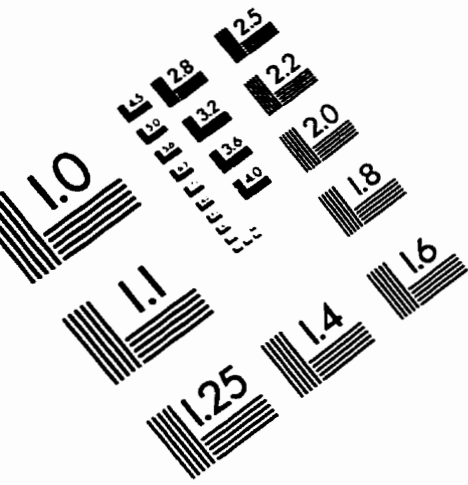
Le format du fichier **.SOL*

Le fichier *ASCII *.SOL* sortant de l'interface résolveur (figure 2.1) a le format montré par le tableau K.1 où $\langle cvType \rangle$ correspond au type de volume de contrôle, selon le tableau 4.1. Les données t et n sont, respectivement, le temps de calcul et le nombre d'itérations atteignant la solution.

TABLEAU K.1: *Format du fichier *.SOL*

$\langle vieFile \rangle$			
$\langle DIM \rangle$	$\langle cvType \rangle$	$\langle nCv \rangle$	$\langle nSca \rangle \langle namesSca \rangle t n$
$volume_1Scalaire_1$	$volume_1Scalaire_2$	...	$volume_1Scalaire_{nSca}$
$volume_2Scalaire_1$	$volume_2Scalaire_2$	...	$volume_2Scalaire_{nSca}$
$volume_3Scalaire_1$	$volume_3Scalaire_2$	...	$volume_3Scalaire_{nSca}$
...			
$volume_{nCv}Scalaire_1$	$volume_{nCv}Scalaire_2$	...	$volume_{nCv}Scalaire_{nSca}$

IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved