

N° d'ordre:

THESE DE DOCTORAT
présentée à
**L'UNIVERSITE DES SCIENCES ET
TECHNOLOGIES DE LILLE**
(U.F.R. de Mathématiques)
pour obtenir le Grade de
DOCTEUR EN MECANIQUE
par



Baléba BALEBA

**Résolution numérique des équations de
Navier-Stokes par une méthode implicite
par zones en vue d'une application en
électrochimie**

Soutenance prévue le 29 Juillet 1997 devant la Commission d'examen

Membres du Jury :

Président : M.

Rapporteurs : M. O. DAUBE, Professeur à l'université d'Evry.

M. D. VANDROMME, Professeur à l'INSA de Rouen.

Membres : M. G. CAIGNAERT, Professeur à l'ENSAM de LILLE.

**M. MONNIER, Ingénieur à l'Institut de Mécanique des Fluides
de Lille (ONERA-IMFL).**

M. L. VANHEE, Ingénieur à la société TDM

**M. STANISLAS, Professeur à l'Ecole Centrale de Lille
(directeur de thèse).**

Table des matières

NOTATIONS PRINCIPALES	8
I Introduction générale	11
1 Position du problème physique	13
2 Position du problème mathématique	17
2.1 Introduction	17
2.2 Cinétique des réactions électrochimiques et surpotentiel de surface	18
2.3 Le flux de matière électrochimique	18
2.3.1 Le cas des solutions fortement diluées	20
2.3.2 Transport des espèces par migration dûe au champ électromagnétique	21
2.4 Transport par convection et équation de Navier-Stokes	22
2.5 Le modèle simplifié	23
2.6 Conclusion	24
3 Modélisation de la turbulence	25
3.1 Introduction	25
3.2 Equations de transport de la turbulence en fluide incompressible	26
3.2.1 Equations pour les grandeurs hydrodynamiques moyennes	26
3.2.2 Equations de transport pour les contraintes de Reynolds	28
3.2.3 Equation de transport pour l'énergie turbulente	29
3.2.4 Equation de transport pour le taux de dissipation turbulente	29
3.3 Le problème de fermeture	30
3.3.1 Fermeture par un modèle $k - \epsilon$ à grand nombre de Reynolds	30
3.3.2 Modèle $k - \epsilon$ à faible Reynolds	32
3.3.3 Equations à résoudre et conditions aux limites	33
3.4 Les équations du modèle électrochimique simplifié et les conditions aux limites	34
3.4.1 Le modèle simplifié	34
3.4.2 Les conditions aux limites	35
3.5 Conclusion	36

4	Le problème de la résolution numérique du modèle d'écoulement	37
4.1	Introduction	37
4.2	La résolution des équations de Navier-Stokes	38
4.2.1	Rappels historiques	38
4.2.2	Méthodes de résolution	39
4.2.3	Architecture des ordinateurs et problèmes informatiques . . .	43
4.3	Démarche choisie	44
II	Discrétisation des équations	47
5	Discrétisation des équations	49
5.1	Introduction	49
5.1.1	Maillage	49
5.1.2	Interpolation de la solution et de ses dérivées	51
5.1.3	La méthode des résidus pondérés	52
5.2	Le problème des termes convectifs	53
5.2.1	Schémas centré et décentré	53
5.2.2	Le schéma hybride	55
5.2.3	Différents types d'interpolation du terme convectif	55
5.2.4	Formulation complète des équations	56
5.3	Conditions aux limites et termes sources du modèle de turbulence . .	57
5.3.1	Traitement des conditions aux limites pour les schémas d'ordre un	57
5.3.2	Termes sources des équations du modèle de turbulence	58
5.4	Conclusion	59
III	Résolution des équations discrétisées.	61
6	Généralités	63
6.1	Introduction	63
6.2	Résolution multidomaines et la méthode alternée de Schwarz	64
6.3	Algorithmes de base	66
6.3.1	Problème linéaire	67
6.3.2	Problèmes non-linéaires et algorithme de Newton-Raphson. . .	68
6.4	Conclusion	69
7	Résolution des équations de Navier-Stokes	71
7.1	Introduction	71
7.2	Linéarisation des équations	71
7.2.1	Déduction des coefficients des équations discrétisées	72
7.2.2	Estimation numérique de la matrice jacobienne	73
7.3	La méthode SIMPLER	74

7.3.1	Equations pour la vitesse	74
7.3.2	Equation pour la pression	75
7.3.3	Correction de vitesse et conservation de la masse	76
7.4	Méthode de résolution par zones implicite.	77
7.4.1	Optimisation de l'inversion pour maillage régulier	77
7.4.2	Les frontières internes et le problème de la pression	80
7.5	Les techniques de lissage étudiées	82
7.5.1	La méthode P.B.C. (Parabolic Block Correction)	83
7.5.2	Une méthode dérivée de l'algorithme SIMPLER	85
7.5.3	Résultat du test de PBC et de notre méthode de lissage	87
7.5.4	La méthode par zones adaptée au sens de l'écoulement	87
7.6	Résolution des équations du modèle turbulent	88
7.7	Inversion des systèmes linéaires	89
7.8	Conclusion	90
8	La méthode multigrille	91
8.1	Introduction	91
8.2	Relaxation sur deux maillages	93
8.3	Discrétisation multigrille	94
8.3.1	Les maillages lâches	95
8.3.2	Projection et Interpolation	95
8.3.3	Equations sur le maillage lâche: le cas non-linéaire	97
8.3.4	Interprétation	100
8.4	Application aux équations de Navier-Stokes.	101
8.5	Le cas des modèles de turbulence.	103
8.6	Conclusion	104
IV	Architecture du code	105
9	Architecture du code	107
9.1	Introduction	107
9.2	La solution TEACHT	108
9.2.1	Présentation	108
9.2.2	Critique du TEACHT	108
9.2.3	conclusion	109
9.3	Présentation générale du code	110
9.3.1	Définition du problème	110
9.3.2	Organisation des calculs	111
9.3.3	Sortie des résultats.	112
9.4	Structure et manipulation des données dans le code	112
9.4.1	Organisation générale des données	112
9.4.2	Détails de la description du domaine	113

9.4.3	Détails des paramètres de linéarisation des équations	113
9.4.4	Accès aux données sur les conditions aux limites	113
9.4.5	Accès aux données du maillage	113
9.4.6	Accès aux champs de valeurs	114
9.5	Description et lectures des données	114
9.5.1	Le fichier de description du domaine	114
9.5.2	Fichier de description du maillage	115
9.5.3	Fichier de description des conditions aux limites	115
9.6	Modules importants	116
9.6.1	Calcul des résidus	116
9.6.2	Linéarisation	116
9.6.3	Résolution des équations de Navier-Stokes par une méthode par zones	116
9.6.4	Résolution des équations de Navier-Stokes par la méthode SIMPLER	117
9.6.5	Algorithme de Thomas pour système mono-variables	118
9.6.6	Projection et Interpolation	118
9.7	Caractéristiques spécifiques du code	119
9.7.1	Le changement de configuration	119
9.7.2	Ajout de variables	119
9.7.3	Ajout d'algorithmes de résolution	119
9.8	Conclusion	120
10	Performances du code	121
10.1	Les écoulements laminaires	121
10.1.1	La cavité.	121
10.1.2	La marche.	122
10.1.3	Effet du nombre de Reynolds.	122
10.1.4	Organisation par sous-domaine et résolution par zone	123
10.1.5	Consommation en place mémoire pour les algorithmes SIM- PLER et par zones	123
10.1.6	La méthode multigrille et la densité de maillage.	124
10.2	Les écoulements turbulents	125
10.2.1	Initialisation multigrille	125
10.2.2	Multigrille complet	126
10.2.3	Conclusion	126
V	Les Résultats.	127
11	Validation des résultats	129
11.1	Écoulements Laminaires	129
11.1.1	Canal plan	129

11.1.2	Cavité entraînée	130
11.1.3	La marche descendante	130
11.2	Les écoulements turbulents	131
11.2.1	Introduction	131
11.2.2	Les écoulements de Couette-Poiseuille	131
11.2.3	L'écoulement de marche descendante	133
11.3	Application à l'électrochimie	134
12	Conclusion et perspectives	135
A	Détails de discrétisation	147
A.1	Schéma hybride	147
A.2	Calcul de dérivées pour le terme de production de turbulence	151
	Figures	152

NOTATIONS PRINCIPALES:

Notations pour le modèle hydrodynamique:

\vec{U}	vecteur vitesse du fluide.
P	pression.
μ	viscosité dynamique.
ν	viscosité cinématique.
ρ	masse volumique.
\vec{g}	accélération de la pesanteur.
\vec{f}	force à distance s'exerçant sur le fluide.
$\vec{\sigma}$	tenseur des contraintes visqueuse s'exerçant sur le fluide.
\overline{U}_i	composante i de la vitesse moyenne du fluide.
u_i	composante i de la vitesse fluctuante du fluide.
$\overline{u_i u_j}$	corrélations croisées de vitesse.
k	énergie turbulente.
ϵ	taux de dissipation de l'énergie turbulente.
ν_t	viscosité cinématique turbulente.
u_τ	vitesse de frottement.
Re_q	nombre de Reynolds basé sur la vitesse débitante.
Re_{max}	nombre de Reynolds basé sur la vitesse maximale.

Notations pour le modèle électrochimie:

\vec{N}_i	vecteur flux des ions de l'espèce i .
\vec{E}	vecteur champ électrique.
\vec{B}	vecteur champ magnétique.
\vec{J}	vecteur densité de courant.
ρ_e	densité de charge.
C_i	concentration en ions de l'espèce i .
μ_i	potentiel électrochimique de l'espèce i .
z_i	charge de l'ion d'espèce i .
F	constante de Faraday.
R	constante des gaz parfait.
Φ	potentiel électrostatique de l'électrolyte.
T	température.
i_0	courant d'échange en l'absence de surpotentiel de réaction.
D_i	coefficient de diffusion laminaire des ions zinc dans l'électrolyte.

D_{ti}	coefficient de diffusion turbulent des ions zinc dans l'électrolyte.
\bar{v}_i	vitesse relative de déplacement des ions de l'espèce i dans l'électrolyte.
S_c	nombre de Schmidt laminaire.
S_t	nombre de Schmidt turbulent.

Notations pour le modèle numérique:

$x(i)$	Abscisse d'un point du maillage non décalé.
$y(j)$	ordonnée d'un point du maillage non décalé .
$xu(i)$	Abscisse d'un point du maillage décalé.
$yv(j)$	ordonnée d'un point du maillage décalé .
A_w^ϕ	coefficient de $\phi(i-1, j)$ dans l'équation discrétisée de ϕ .
$A_W^{\phi_1\phi_2}$	coefficient de $\phi_2(i-1, j)$ dans la version linéarisée de l'équation pour ϕ_1 .
$Res^\phi(i, j)$	résidu de l'équation discrétisée de ϕ .
L	opérateur différentiel.
f	second membre du problème aux dérivés partielles.
ϕ	inconnue du problème à résoudre.
L_h	version discrétisée de L .
f_h	version discrétisée de f .
ϕ_h	version discrétisée de ϕ .
$\delta\phi_h$	correction calculé pour ϕ_h .
Z	$[i_0, i_1] \times [j_0, j_1]$ zone délimité pour une résolution implicite.
$N_I = i_1 - i_0 + 1$	
$N_J = j_1 - j_0 + 1$	
N_{eq}	nombre d'équations à résoudre.
\dot{M}	débit.
$\tau_H = Rh_{sH}$	résidu multigrille.

Partie I

Introduction générale

Chapitre 1

Position du problème physique

L'industrie moderne utilise des systèmes de production de plus en plus complexes. Dans la majorité des cas, leur fonctionnement dépend d'un grand nombre de variables. Les relations souvent non-linéaires entre ces différentes variables, compliquent la recherche de points de fonctionnement optimaux. Mis à part ce problème, l'évolution de ces systèmes pour augmenter leur capacité de production nécessite l'examen d'un certain nombre de configurations en vue de retenir la meilleure. En troisième lieu, la formation du personnel pour le pilotage de ces systèmes nécessite de leur faire manipuler certaines variables afin de mieux les préparer à réagir en situation réelle.

Les trois points que nous venons d'évoquer supposent des interventions risquées, parfois lourdes, dans tous les cas coûteuses sur des systèmes en fonctionnement. Même si l'on peut difficilement se passer d'interventions directes sur le système réel, tout procédé permettant d'en limiter le coût est le bienvenu pour l'industriel. La conception de maquette à petite échelle a constitué avant l'avènement des ordinateurs la solution de choix pour les problèmes de coût évoqués. Une autre approche utilisée jusqu'ici consiste à rechercher des solutions analytiques à des modèles théoriques plus ou moins simples. Ce qui n'est pas d'une grande utilité pour les modèles non-linéaires, ou pour des exigences de précision de plus en plus importantes.

L'avènement des ordinateurs et le développement de techniques d'analyse numérique ont permis de réduire considérablement la classe des problèmes pour lesquels on a pas de solution approchées exploitable. La diminution des coûts des calculateurs et l'amélioration des interfaces de communication avec ces derniers, permet de construire des simulateurs de plus en plus réalistes pour un nombre croissant de systèmes: ceux pour lesquels un modèle théorique est connu. Cette évolution a permis de réduire considérablement les situations où la construction de maquette est nécessaire. Le partenaire industriel de cette étude est la société Beautor S. A. Cette entreprise qui est au départ une aciérie dotée de ligne de laminage à chaud, s'est reconvertie progressivement dans le laminage à froid. En 1966 elle devient la filiale du groupe Belge Cockerill-Sambre. En 1974 elle s'équipe d'une ligne d'électrozingage qui la place dès 1976 au premier rang des producteurs européens de tôles électrozinguées monoface. En 1977 elle cesse ses dernières activités d'aciérie pour se consacrer es-

sentiellement à l'électrodéposition et au laminage à froid, l'essentiel de son marché étant l'automobile.

L'électrodéposition en continu est une technique assez récente de dépôt surfacique d'un revêtement pour protéger un matériau vulnérable des attaques électrochimiques naturelles.

Les lignes d'électrozingage de Beator sont constituées de différentes parties: Une première comporte des dérouleuses, une cisaille, une soudeuse et un accumulateur permettant l'alimentation continue de la ligne en tôle. Une deuxième partie permet la préparation de la tôle pour la phase de dépôt. Cette dernière passe ensuite dans les cellules pour le dépôt proprement dit. Une dernière partie permet le post-traitement pour améliorer la tenue au processus de peinture entre autre.

La figure 1.1 représente une des cellules de forme radiale utilisées actuellement chez Beator. Les deux électrodes de forme cylindrique sont séparées par un espace d'une épaisseur de 8 mm. Cet espace forme un canal hydrodynamique dans lequel s'écoule une solution constituée par un mélange d'acide sulfurique et de sulfate de zinc. La tôle à protéger s'enroule sur le cylindre interne pour former la cathode. Le fluide est injecté par l'une des extrémités du canal, et s'écoule dans le sens inverse du déplacement de la tôle. Une différence de potentiel appliquée aux électrodes assure le dépôt du zinc. La circulation du fluide permet la régénération de l'électrolyte et donc le maintien d'un rythme régulier pour le dépôt. Cette dernière technique a l'avantage de permettre la mise en place d'un processus continu pour la protection de tôle présentée sous forme de bobines comme c'est le cas chez Beator. La continuité du processus suppose un écoulement permanent dans le canal, permettant ainsi d'assurer un flux constant de matière à déposer près de la zone où s'effectue le processus.

Comme on vient de le voir, le fluide joue le rôle fondamental de média de transport. Des études ont montré que ce rôle est encore mieux joué par des écoulements turbulents. Le fluide, qui est en fait une solution acide, s'écoule dans un canal de forme radiale. Si on fait un certain nombre d'approximations (voir [64]), on obtient un écoulement dans un canal plan, qui reste une représentation très réaliste de l'écoulement dans la cellule. Les écoulements dans un canal plan avec une paroi mobile forment la famille des écoulements de Couette-Poiseuille. Les écoulements de Couette-Poiseuille sont une combinaison de deux types d'écoulements connus: l'écoulement de Poiseuille pur et l'écoulement de Couette pur. La présence simultanée d'un gradient de pression et d'une paroi mobile engendre une dissymétrie du champ moyen et du champ fluctuant de ces écoulements. Malgré leur complexité, ils présentent l'avantage, sur un plan fondamental, d'isoler certains phénomènes turbulents. Par ailleurs ils sont des représentations simplifiées mais réalistes, de situations rencontrées dans l'industrie. Notamment dans les problèmes de lubrification mais surtout dans les cellules d'électrodéposition, d'où leur intérêt pour notre travail.

Ces écoulements ont fait l'objet de nombreuses études sur le plan expérimental et de plus en plus par simulation numérique. En ce qui concerne notre laboratoire, quatre thèses ont été consacrées, en grande partie, à ces écoulements.

La première, celle de Monnier [43] s'est focalisée sur l'étude de différentes modélisations turbulentes pour des écoulements monodimensionnels.

Celle de L. Vanhée [64] a permis de poursuivre l'étude des modèles, mais en plus elle a été l'occasion d'un examen plus global du processus d'électrodéposition sur une maquette de cellule de type Beautor. Ce travail s'est achevé par une redéfinition des cellules Beautor avec un changement majeur dans le mode d'injection du fluide dans les cellules. Toute la ligne d'électrozingage de Beautor a depuis été rééquipée avec ces nouvelles cellules qui fonctionnent aujourd'hui de manière satisfaisante.

C'est avec K. Corenflos [12] qu'a commencée l'étude expérimentale proprement dite des écoulements de Couette-Poiseuille, avec la mise en place d'un banc de manipulation. Les résultats des mesures effectuées permettent la validation des résultats de simulations numériques directes. Avant ce travail, les principales données expérimentales utilisées pour les validations de modèles provenaient des travaux d'El Telbany et Reynolds. Ces résultats ont commencé à susciter quelques réserves suite au travail de Corenflos.

La collecte de données commencée par Corenflos à Reynolds 5000 a été poursuivie par A. Deldicque [15] à Reynolds 10000. Celui-ci a confirmé les réserves signalées sur les mesures d'El Telbany et Reynolds [61]. Ces deux dernières études expérimentales ont traité systématiquement les cas de Poiseuille pur, de type Couette et de type intermédiaire dont nous expliquerons les différences par la suite.

L'objectif ultime de notre partenaire industriel est de disposer d'un code permettant le calcul d'écoulements tridimensionnel turbulents dans des géométries relativement complexes. Mais aussi la résolution d'un modèle électrochimique robuste, de manière à disposer d'un outil pratique, permettant des expérimentations numériques réalistes, sur les cellules en exploitation.

En examinant cet objectif, on relève les difficultés suivantes: L'hydrodynamique turbulente et l'électrochimie sont deux domaines de recherche importants. La réunion de modèles venant des deux domaines donne un système à nombre de variable élevé. La prise en compte du caractère tridimensionnel des écoulements ne fait qu'aggraver ce problème.

La présente étude ne constitue qu'une étape intermédiaire vers cet objectif. Elle est basée sur les simplifications suivantes: On traitera d'écoulements bidimensionnels. Le modèle d'électrodéposition sera simplifié et découplé du modèle hydrodynamique pour ce qui concerne sa résolution.

1. On se limitera à des modèles de turbulence à deux équations de transport. Malgré ces simplifications, le modèle résultant comporte un nombre important de variables.
2. De plus, les phénomènes turbulents survenant à proximité immédiate des parois, le recours habituel aux fonctions de parois n'est pas possible, ce qui

nécessite de tester des méthodes de résolution robustes pour un nombre de point important.

Chapitre 2

Position du problème mathématique

2.1 Introduction

L'essentiel du processus qui nous intéresse se situe au voisinage de la tôle qui joue le rôle de cathode. Le dépôt de zinc se fait par une réaction de réduction du cation Z_n^{2+} . Les cations sont prélevés dans la solution au voisinage de la tôle et les électrons sont fournis par un générateur de tension.

L'alimentation du voisinage de la cathode en ions se fait par un processus dont le but est de compenser les variations de potentiel électrochimique créés par la réaction. Cette compensation s'effectue par un flux d'ions qui a trois origines distinctes:

1. La migration provoquée par le champ électrique. Champ électrique créée par un saut de potentiel à l'interface métal-solution.
2. La diffusion créée par un gradient de concentration
3. Un processus complémentaire de transport par convection qui intervient pour assurer un renouvellement plus rapide de la solution près de l'électrode.

Si les flux de diffusion et de migration sont décrit par des modèles d'électrochimie, le flux de convection nécessite la connaissance du champ de vitesse moyen de la solution. Cette vitesse moyenne est normalement composée des vitesses des espèces constituant la solution. Le modèle qui décrit l'évolution de ce champ de vitesse est constitué par les équations de Navier Stokes.

2.2 Cinétique des réactions électrochimiques et surpotentiel de surface

Les réactions chimiques permettant le dépôt de zinc se font à l'interface électrode-fluide. La principale de ces réactions s'écrit :



Ces réactions créent un courant électrique dont nous considérons la composante normale i_n . La relation qui relie la concentration de l'espèce électro-active, le potentiel et le courant s'écrit sous la forme générale suivante :

$$i_n = i_0 \left\{ \exp\left(\frac{\alpha_a F}{RT} \eta_s\right) - \exp\left(\frac{\alpha_c F}{RT} \eta_s\right) \right\} \quad (2.2)$$

1. $\eta_s = V - \Phi$ surpotentiel de surface, V imposé et Φ potentiel dans le fluide adjacent au matériau. L'évolution de la réaction dépendant du potentiel d'activation qui dépend du surpotentiel de surface.
2. α_a et α_c sont des paramètres qui caractérisent la cinétique de la réaction à l'anode et à la cathode.
3. $i_0 = nF k_c^{(1-\beta)} k_a^\beta \prod_i C_i^{(q_i + \beta s_i)}$ est le courant à l'équilibre du système. Il dépend de la composition de la solution au voisinage de l'électrode. β est un coefficient de symétrie qui représente la répartition du potentiel V entre les réaction anodique et cathodiques. p_i et q_i représentent les ordres des réactions anodique et cathodiques respectivement. k_a et k_c sont les constantes de réaction, et $s_i = p_i - q_i$.

La consommation d'ions due à la réaction qui obéit à l'équation 2.2 crée une baisse de concentration que compense le transport par diffusion et par migration.

2.3 Le flux de matière électrochimique

Nous venons de présenter le modèle décrivant la cinétique de la réaction à la cathode. L'un des paramètres dont dépend ce processus est la concentration de l'espèce à déposer C_i . Le modèle général décrivant l'évolution de ce paramètre est donné par l'équation suivante.

$$C_{i,t} = R_i - \nabla \cdot \vec{N}_i \quad (2.3)$$

1. C_i concentration de l'espèce i .
2. R_i taux de création ou de consommation par une réaction par exemple, de l'espèce i .

3. \vec{N}_i flux de l'espèce i .

Cette équation, écrite pour une espèce i , exprime le fait que la variation de concentration en un point en fonction du temps est égale au taux de création de l'espèce, diminué de la variation de flux dans l'élément de fluide. La création de l'espèce dans un volume résultera essentiellement d'une réaction chimique entre les différents constituants du milieu. Le deuxième terme du second membre est le plus important en ce qui concerne les phénomènes de transport.

$$\vec{N}_i = \vec{v}_i C_i \quad (2.4)$$

Ou \vec{v}_i représente la vitesse moyenne des ions de l'espèce i . Le modèle le plus général qui décrit la relation entre ces vitesses et le potentiel électrochimique μ_i est le suivant:

$$C_i \vec{\nabla} \cdot \mu_i = \sum_j K_{ij} (\vec{v}_j - \vec{v}_i) \quad (2.5)$$

ou \vec{v}_j est la vitesse moyenne de l'espèce j .

$$K_{ij} = RT \frac{C_i C_j}{C_T D_{ij}}$$

est le coefficient d'interaction entre les espèces i et j . D_{ij} est un coefficient de diffusion décrivant l'interaction entre l'espèce i et l'espèce j , et $C_T = \sum_i C_i$. Le terme de gauche représente une densité volumique de force, force qui dérive du potentiel électrochimique. Le terme de droite de l'équation 2.5 représente la résultante des interactions avec les autres espèces en présence dans le fluide. Les équations 2.5 forment un système linéaire dont la résolution donne le flux en fonction des différents gradients de potentiel électrochimique. Pour introduire le potentiel électrostatique Φ , qui caractérise l'état électrique de la solution, on considère une espèce de référence n . Son potentiel électrochimique s'écrit:

$$\mu_n = RT \ln(C_n) + z_n F \Phi \quad (2.6)$$

Les potentiels électrochimiques des autres espèces sont alors définis par:

$$\mu_i = RT \ln(C_i) + z_i F \Phi + RT \left(\ln f_i - \frac{z_i}{z_n} \ln f_n \right) \quad (2.7)$$

f_i est le coefficient d'activité molaire pour l'espèce i . z_i est le nombre de charges de l'espèce i . F est la constante de Faraday, R la constante universelle des gaz et T la température.

2.3.1 Le cas des solutions fortement diluées

La solution utilisée dans les cellules de type Beautor est diluée. L'électrolyte inerte dans ce cas est constitué par l'acide sulfurique. Le but de cette dilution est d'augmenter la conductivité de la solution afin d'en limiter la résistance ohmique et donc la consommation d'énergie. Ce type de solution permet de faire des approximations importantes dans les équations du modèle électrochimique. Pour une telle solution, les coefficients d'interaction avec le fluide K_{ij} se réduisent à l'interaction des espèces minoritaires avec le solvant. Dans ce cas pour une espèce minoritaire i , la relation 2.5 se réduit à:

$$C_i \vec{\nabla} \cdot \mu_i = \frac{RTC_0}{C_0 D_{0i}} (C_i \vec{v}_0 - C_i \vec{v}_i) \quad (2.8)$$

Les termes d'indice 0 correspondant au solvant. Pour le flux défini par 2.4, \vec{N}_i est donné par la relation:

$$\vec{N}_i = C_i \vec{v}_0 - \frac{D_{0i} C_i}{RT} \vec{\nabla} \cdot \mu_i \quad (2.9)$$

Le potentiel électrochimique se réduit à:

$$\mu_i = RT \ln(C_i) + z_i F \Phi \quad (2.10)$$

On en déduit pour le flux total, l'expression:

$$\vec{N}_i = C_i \vec{U} - D_i \vec{\nabla} C_i - \frac{z_i F}{RT} D_i C_i \vec{\nabla} \Phi \quad (2.11)$$

où $D_i = D_{0i}$ et $\vec{U} = \vec{v}_0$.

On retrouve les trois composantes du transport des ions:

1. Le transport convectif, assuré par la vitesse du fluide.
2. Le transport par diffusion, résulte des différences de concentration en des points voisins du fluide.
3. Le transport par migration dû aux forces électriques qui s'exercent sur les particules chargées que sont les ions.

Lorsque cela est nécessaire, le potentiel Φ est déterminé par le modèle électromagnétique de Maxwell.

2.3.2 Transport des espèces par migration dûe au champ électromagnétique

L'importance du terme de migration dépend essentiellement de la valeur du champ électrique. Ce dernier dépend de la conductivité de la solution. Pour les métaux, les charges créées dans le matériau sont immédiatement déplacées en surface. Le champ reste nul à l'intérieur du métal. Dans le cas d'un électrolyte, plus on augmente sa conductivité, plus il se comporte comme un métal. Le terme de migration sera donc plus faible dans les solutions diluées pour les espèces minoritaires. Dans le cas contraire, les équations de Maxwell permettent de modéliser le comportement électromagnétique de la cellule. Les résultats de cette modélisation sont utilisés pour obtenir le champ du potentiel intervenant dans les termes de migration des équations de transport pour les C_i . On en déduit également les forces électromagnétiques \vec{f}_{em} pour les équations du modèle hydrodynamique que l'on présentera plus loin.

Les champs électromagnétiques découlent de l'existence d'une densité de charges ρ_e et de courant électrique \vec{J} dans un milieu. La connaissance du champ électrique \vec{E} et du champ magnétique \vec{B} permettent de décrire le comportement électromagnétique d'un milieu. On définit également les vecteurs déplacement électrique \vec{D} et excitation magnétique \vec{H} . Du point de vue électromagnétique, un milieu est caractérisé par sa polarisation \vec{P} et par son aimantation \vec{I} . Un milieu polarisé est un milieu dans lequel il y a peu de charges libres. Les milieux aimantés ne permettent pas la circulation de courant d'intensité mesurables. Les micro-courants qui y circulent créent un champ magnétique permanent. Le milieu est également défini par sa permittivité diélectrique ϵ et sa perméabilité magnétique μ . La loi de comportement générale du milieu s'écrit:

$$\vec{D} = \vec{E} + \vec{P} \quad (2.12)$$

$$\vec{H} = \mu\vec{B} - \vec{I} \quad (2.13)$$

Les équations permettant de déterminer ces différentes variables sont les suivantes:
Conservation de la charge électrique:

$$\vec{\nabla} \cdot \vec{E} = \rho_e \quad (2.14)$$

Conservation de la densité de courant:

$$\vec{\nabla} \times \vec{H} - \frac{\partial \vec{D}}{\partial t} = \vec{J} \quad (2.15)$$

On peut également définir un potentiel électrique scalaire ϕ et un potentiel magnétique vecteur \vec{A} . Ces deux nouvelles variables simplifient parfois la détermination des champs. Les équations qui les relient à \vec{B} et \vec{E} sont:

Equation du potentiel électrique:

$$\vec{\nabla} \Phi = -\left(\frac{\partial \vec{A}}{\partial t} + \vec{E}\right) \quad (2.16)$$

Equation du potentiel magnétique:

$$\vec{B} = \vec{\nabla} \times \vec{A} \quad (2.17)$$

On a en plus la condition de jauge:

$$\vec{\nabla} \cdot \vec{A} + \epsilon\mu \frac{\partial \Phi}{\partial t} = 0 \quad (2.18)$$

Les électrolytes qui nous intéressent ne sont ni polarisés ni magnétisés. Au contraire les équations pour les C_i permettent de connaître à tout moment la densité de charge électrique ρ_e et la densité de courant \vec{J} .

Les forces \vec{f}_{em} qui s'exercent sur un élément de fluide sont données par:

$$\vec{f}_{em} = \rho_e \vec{E} + \vec{J} \times \vec{B} \quad (2.19)$$

avec $\rho_e = \sum_i z_i C_i F$ et $\vec{J} = \sum_i z_i C_i F \vec{v}_i$

2.4 Transport par convection et équation de Navier-Stokes

Le modèle décrivant les écoulements de fluides les plus souvent rencontrés est donné par les équations de Navier-Stokes. Ce modèle décrit la variation dans le temps de la quantité de mouvement d'un élément de fluide soumis à des contraintes internes et externes. Ces équations découlent en fait de la loi fondamentale de la dynamique. Elles s'écrivent:

$$(\rho \vec{U})_t + \rho \vec{U} \cdot \vec{\nabla} \vec{U} = \vec{f} + \vec{\nabla} \cdot (\vec{\sigma} - P) \quad (2.20)$$

Le fluide est soumis à des forces extérieures \vec{f} . Ce sont en général des forces de gravité $\rho \vec{g}$ ou \vec{g} représente l'accélération de la pesanteur. Dans le cas de fluides électrochimiques, il faut prendre en considération les forces électromagnétiques \vec{f}_{em} qui s'exercent sur un élément de fluide par l'intermédiaire des particules chargées qui y sont contenues. Nous verrons cependant plus loin que l'hypothèse d'électronéutralité, qui est assez bien respectée dans la majorité des systèmes électrochimiques, permet de négliger l'effet de ces forces [44]. Le fluide est également soumis à des forces internes qui sont dues au différence de quantité de mouvement en deux points voisins. Ce sont les contraintes diffusives qui sont données pour un fluide Newtonien par la formule:

$$\vec{\sigma} = [\mu(\vec{\nabla} \vec{U} + \vec{\nabla} \vec{U}^t) + (\zeta - \frac{2\mu}{3}) I \vec{\nabla} \cdot \vec{U}] \quad (2.21)$$

Ces contraintes dépendent de la nature du fluide qui est représentée ici par les coefficients de viscosité μ et ζ . Le dernier facteur intervenant dans la mise en mouvement d'un fluide est lié aux gradients de pression qui règnent en son sein. Dans le cas

d'un fluide isovolume, les équations de conservation de la masse et de Navier-Stokes s'écrivent:

$$\vec{\nabla} \cdot \vec{U} = 0 \quad (2.22)$$

pour la conservation de la masse et

$$(\rho \vec{U})_{,t} + \rho \vec{U} \cdot \vec{\nabla} \vec{U} = -\vec{\nabla} P + \vec{\nabla} \mu \cdot \vec{\nabla} \vec{U} + \rho \vec{g} + \vec{f}_{em} \quad (2.23)$$

pour la quantité de mouvement:

2.5 Le modèle simplifié

Nous avons vu plus haut les simplifications qu'entraîne l'hypothèse de solution diluée. Les autres simplifications proviennent de l'hypothèse d'électroneutralité. Cette hypothèse repose sur l'importance des forces nécessaires pour maintenir une densité de charge non nulle. Le résultat en est que, les seuls points où l'on peut trouver une densité de charge sont situés à proximité immédiate de l'interface métal fluide, à quelques rayons atomiques. De telles distances ne sont pas compatibles d'un point de vue dimensionnel avec les domaines de validité des différents modèles que nous avons décrit. C'est la raison pour laquelle cette zone qu'on appelle la double couche électrique est souvent modélisée de manière à décrire ses effets macroscopiques sur le comportement de la cellule. Cette double couche est généralement représentée sous la forme d'un condensateur dont la capacité varie entre autre avec l'état électrostatique du métal. Par conséquent son effet se fait surtout sentir dans les phases transitoires de charge et de décharge. L'électroneutralité permet d'éliminer les forces électromagnétiques du modèle hydrodynamique.

$$\vec{f}_{em} = 0 \quad (2.24)$$

Pour cette étude, nous avons également choisi, pour des raisons de temps de nous arrêter au calcul du courant limite de diffusion. Ce qui permet de retirer toute la partie électromagnétique du modèle. Sinon il faut revenir à l'équation 2.2 pour déterminer les conditions aux limites pour les concentrations. Dans notre cas, on calcule la concentration pour une seule espèce, le zinc. Pour cela on considère une situation où la valeur du courant ne dépend que de l'apport de zinc à l'interface. Il n'y a aucune accumulation de zinc dans cette zone. On peut alors, avec une bonne approximation prendre une concentration nulle comme condition à la limite. Les équations que nous allons résoudre sont donc les suivantes:

Hydrodynamique:

$$\vec{\nabla} \cdot \vec{U} = 0 \quad (2.25)$$

pour la conservation de la masse,

$$(\rho \vec{U})_{,t} + \vec{U} \cdot \vec{\nabla} \vec{U} = -\vec{\nabla} P + \vec{\nabla} \mu \cdot \vec{\nabla} \vec{U} \quad (2.26)$$

pour la quantité de mouvement et

$$C_{i,t} + \vec{U} \cdot \vec{\nabla} C_i = \vec{\nabla} D_i \cdot \vec{\nabla} C_i \quad (2.27)$$

pour l'électrochimie.

Cette dernière est une équation de transport de type convection-diffusion.

En examinant ce modèle, on constate que les variables hydrodynamiques interviennent dans les équations de l'électrochimie alors que l'inverse n'est pas vrai. On pourra donc calculer le champ de vitesse sans se soucier de la partie électrochimique. On résoudra par la suite des équations de transport pour des scalaires passifs que sont les concentrations.

2.6 Conclusion

Le modèle simplifié présenté ci-dessus n'est valable que pour des écoulements laminaires. La prochaine étape consiste à déterminer un modèle mieux adapté à la description des écoulements industriels, qui sont en général fortement turbulents.

Chapitre 3

Modélisation de la turbulence

3.1 Introduction

Deux obstacles importants limitent les possibilités de résolution des équations aux dérivées partielles par ordinateur. Le premier est lié à la nature continue de ces équations, et le second à la limitation intrinsèque des ordinateurs pour la représentation des nombres réels. Le premier obstacle est levé par le théorème de Shannon qui stipule que, si une fonction continue possède un spectre borné, alors il est possible de la représenter par un nombre fini de valeurs sans perte d'information. La représentation en virgule flottante permet de résoudre le second problème de manière tout à fait acceptable. Tout ceci a permis de trouver des solutions satisfaisantes pour une grande classe de problèmes, y compris en Mécanique des Fluides pour la résolution des équations de Navier-Stokes.

Mais, comme le montre également le théorème de Shannon, le coût d'échantillonnage peut devenir prohibitif si la bande spectrale de la fonction à discrétiser devient importante. C'est le cas des écoulements industriels qui sont pour la plupart turbulents. Pour tous ces écoulements, une résolution directe satisfaisante des équations de Navier Stokes n'est pas envisageable. La solution dans ce cas consiste en général à réduire la bande spectrale des fonctions à discrétiser tout en limitant le plus possible les pertes d'information dues à la mise en oeuvre de ce processus. Les fonctions résultantes sont mieux adaptées au niveau de ressource disponible pour leur détermination.

Il est clair que les limitations signalées par le théorème de Shannon ne sont pas seulement valables pour les écoulements turbulents. Ces derniers ont cependant la particularité d'exhiber des comportements spécifiques difficiles à extrapoler à partir d'une simple résolution du modèle laminaire. La turbulence est essentiellement instationnaire et tridimensionnelle. Ces écoulements sont fortement diffusifs ce qui est un avantage si on veut uniformiser la distribution d'une quantité transportable telle que la température ou un colorant, et surtout des concentrations d'ions en électrodéposition. Ils sont également le siège de phénomènes de dissipation qui

transforment l'énergie cinétique en énergie interne, ce qui peut être un inconvénient pour des solides en déplacement dans un fluide, comme c'est le cas pour les engins aéronautiques et aquatiques.

Il existe plusieurs manières d'obtenir une solution exploitable pour des écoulements turbulents. La première consiste, pour des écoulements à faible nombre de Reynolds, à réaliser une simulation numérique directe (donc instationnaire et tridimensionnelle). Les ressources nécessaires sont importantes, notamment pour le stockage et le post-traitement des résultats. Cette solution est pour l'instant utilisée surtout pour les besoins d'études fondamentales des phénomènes turbulents.

La deuxième méthode procède par séparation des fonctions turbulentes en deux parties, par un filtrage des équations tenant compte de la précision du maillage. La première partie est constituée par les composantes fondamentales (bien représentées par le maillage) de la fonction exacte. Le reste constituant la seconde partie. La première partie est déterminées par simulation comme précédemment, alors que la seconde l'est par modélisation. Cette méthode porte le nom de simulation des grosses structures. Malgré un coût nettement plus faible que celui de la première méthode, cette dernière n'est toujours pas utilisable pour les besoins les plus courants.

La troisième méthode est également la plus ancienne. Elle procède par traitement statistique des fonctions à déterminer. Elle est nettement moins coûteuse que les deux premières, mais entraîne également de plus grandes pertes d'informations. On aboutit ainsi à un modèle stationnaire. Les effort de modélisation pour limiter ces pertes d'informations sont également plus importants.

Dans la mesure où nous cherchons à construire un code d'usage courant, cette dernière solution est celle que nous avons retenu. Dans ce qui va suivre, nous allons donner plus de détails sur le traitement statistique et ses conséquences sur les équations à résoudre.

3.2 Equations de transport de la turbulence en fluide incompressible

3.2.1 Equations pour les grandeurs hydrodynamiques moyennes

Dans l'approche statistique, on admet que chaque variable peut être décomposée en une partie basse fréquence et une partie fluctuante:

$$U_i(x_i, t) = \bar{U}_i(x_i) + u_i(x_i, t) \quad (3.1)$$

$$P(x_i, t) = \bar{P}(x_i) + p(x_i, t) \quad (3.2)$$

La partie moyenne est obtenue par intégration sur une durée T très grande par rapport à la période des fluctuations turbulentes.

3.2. EQUATIONS DE TRANSPORT DE LA TURBULENCE EN FLUIDE INCOMPRESSIBL

$$\bar{U}_i(x_i) = \frac{1}{T} \int_0^T U_i(x_i, \tau) d\tau \quad (3.3)$$

$$\bar{P}(x_i) = \frac{1}{T} \int_0^T P(x_i, \tau) d\tau \quad (3.4)$$

O. Reynolds a le premier proposé cette décomposition. Elle permet de séparer chaque variable dépendante en une partie moyenne à faibles variations temporelles, et une partie fluctuante, avec les conventions suivantes:

$$\bar{u}_i(x_i, t) = 0 \quad (3.5)$$

$$\bar{p}(x_i, t) = 0 \quad (3.6)$$

Cette décomposition est substituée dans le système d'équation 2.26. Ensuite on moyenne les expressions obtenues en supposant que les opérations de moyenne et les autres opérateurs différentiels sont permutables. On néglige enfin tout ce qui est moyenne de la partie fluctuante pour toutes les variables, pour obtenir les équations moyennées suivantes:

$$\frac{\partial \bar{U}_i}{\partial x_i} = 0 \quad (3.7)$$

$$\rho \bar{U}_j \frac{\partial \bar{U}_i}{\partial x_j} = - \frac{\partial \bar{P}}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \frac{\partial \bar{U}_i}{\partial x_j} - \rho \overline{u_i u_j} \right] + f_i \quad \text{pour: } i = 1, 2, 3 \quad (3.8)$$

(1) (2) (3) (4) (5)

Certains auteurs considèrent que les valeurs moyennes sont faiblement instationnaires dans une gamme de fréquences différente de celle de la turbulence.

Les différents termes de l'équation de quantité de mouvement représentent:

- (1): la convection
- (2): le terme source dû au gradient de pression
- (3): la diffusion moléculaire
- (4): la diffusion turbulente
- (5): le terme source dû aux forces à distance (pesanteur)

Les équations pour les vitesses fluctuantes servent pour la détermination des équations pour les moments d'ordre supérieurs. Elles s'obtiennent en faisant la différence entre les équations originelles et leur version moyennées. On obtient:

$$\bar{U}_j u_{i,j} = -u_j \bar{U}_{i,j} - u_j u_{i,j} - \frac{1}{\rho} \bar{p}, i + \nu (u_{i,j} + u_{j,i})_{,j} + \overline{u_i u_{j,j}} \quad (3.9)$$

3.2.2 Equations de transport pour les contraintes de Reynolds

Pour obtenir les équations de transport pour les contraintes de Reynolds $\overline{u_i u_j}$, on multiplie l'équation 3.9 par u_i avant d'en prendre la moyenne. On obtient:

$$\begin{aligned} \overline{U_k} \frac{\partial \overline{u_i u_j}}{\partial x_k} = & - \left[\overline{u_j u_k} \frac{\partial \overline{U_i}}{\partial x_k} + \overline{u_i u_k} \frac{\partial \overline{U_j}}{\partial x_k} \right] \\ (1) & \qquad \qquad \qquad (2) \\ & + \frac{\partial}{\partial x_k} \left[\overline{-u_i u_j u_k} - \frac{p}{\rho} (\delta_{ik} u_j + \delta_{jk} u_i) + \nu \frac{\partial u_i u_j}{\partial x_k} \right] \\ & \qquad \qquad \qquad (3) \qquad \qquad \qquad (4) \qquad \qquad \qquad (5) \\ & + \frac{p}{\rho} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - 2\nu \frac{\partial u_i}{\partial x_k} \frac{\partial u_j}{\partial x_k} \\ & \qquad \qquad \qquad (6) \qquad \qquad \qquad (7) \end{aligned} \tag{3.10}$$

L'interprétation physique suivante peut être donnée aux différents termes:

- (1): convection de la corrélation par l'écoulement moyen
- (2): production par interaction entre le champ turbulent et l'écoulement moyen
- (3): diffusion turbulente due aux fluctuations de vitesse
- (4): diffusion turbulente due aux fluctuations de pression
- (5): diffusion visqueuse
- (6): redistribution de l'énergie cinétique turbulente entre les corrélations vitesse-vitesse (retour à l'isotropie pour les corrélations normales)
- (7): dissipation visqueuse

Le terme de dissipation visqueuse est généralement représenté sous la forme tensorielle suivante:

$$\epsilon_{ij} = 2\nu \frac{\partial u_j}{\partial x_k} \frac{\partial u_i}{\partial x_k} \tag{3.11}$$

ϵ_{ij} est appelé taux de dissipation de la corrélation $\overline{u_i u_j}$.

3.2. EQUATIONS DE TRANSPORT DE LA TURBULENCE EN FLUIDE INCOMPRESSIBL

3.2.3 Equation de transport pour l'énergie turbulente

L'équation de transport pour l'énergie turbulente s'obtient en faisant la somme des équations de Reynolds pour les termes $\overline{u_i u_i}$. On obtient:

$$\overline{U}_k \frac{\partial k}{\partial x_k} = - \underbrace{\left[\overline{u_i u_k} \frac{\partial \overline{U}_i}{\partial x_k} \right]}_{(2)} + \frac{\partial}{\partial x_k} \left[\underbrace{-\overline{u_i u_i u_k}}_{(3)} - \underbrace{\frac{\overline{p}}{\rho} u_k}_{(4)} - \underbrace{\nu \frac{\partial k}{\partial x_k}}_{(5)} \right] - \underbrace{\nu \frac{\partial u_i}{\partial x_k} \frac{\partial u_i}{\partial x_k}}_{(6)} \quad (3.12)$$

L'interprétation physique suivante peut être donnée aux différents termes:

- (1): convection de la corrélation par l'écoulement moyen
- (2): production d'énergie par l'écoulement moyen. Il exprime le transfert d'énergie de l'écoulement moyen vers la partie turbulente.
- (3): diffusion turbulente due aux fluctuations de vitesse
- (4): diffusion turbulente due aux fluctuations de pression
- (5): diffusion visqueuse
- (6): dissipation isotrope de l'énergie turbulente, notée ϵ

3.2.4 Equation de transport pour le taux de dissipation turbulente

Une dernière équation de transport est généralement utilisée dans les différents modèles de turbulence. Elle se déduit également des équations de Navier-Stokes. Il s'agit de l'équation de transport pour le taux de dissipation isotrope ϵ :

$$\begin{aligned} \overline{U}_k \frac{\partial \epsilon}{\partial x_k} = & - \underbrace{2\nu \left[\frac{\partial u_i}{\partial x_j} \frac{\partial u_k}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \frac{\partial u_j}{\partial x_k} \right] \frac{\partial \overline{U}_i}{\partial x_k}}_{(2)} - \underbrace{2\nu u_k \frac{\partial u_i}{\partial x_j} \frac{\partial^2 \overline{U}_i}{\partial x_j \partial x_k}}_{(2)} \\ & - \underbrace{2\nu \left[\frac{\partial u_i}{\partial x_k} \frac{\partial u_i}{\partial x_j} \frac{\partial u_k}{\partial x_j} \right]}_{(3)} - \frac{\partial}{\partial x_k} \left[\underbrace{\overline{u_k \epsilon'}}_{(4)} + \underbrace{2 \frac{\nu}{\rho} \frac{\partial p}{\partial x_j} \frac{\partial u_k}{\partial x_j}}_{(5)} - \underbrace{\nu \frac{\partial \epsilon}{\partial x_k}}_{(6)} \right] \quad (3.13) \\ & - \underbrace{2\nu^2 \left(\frac{\partial^2 u_i}{\partial x_k \partial x_k} \right)^2}_{(7)} \end{aligned}$$

La signification des différents termes est la suivante:

- (1): convection par l'écoulement moyen
- (2): production par l'écoulement moyen
- (3): production par le mouvement turbulent
- (4): diffusion turbulente due aux fluctuations de vitesse
- (5): diffusion turbulente due aux fluctuations de pression
- (6): diffusion moléculaire
- (7): dissipation visqueuse de la dissipation

3.3 Le problème de fermeture

Le traitement statistique a fait apparaître de nouveaux termes dans les équations de Navier-Stokes. Il s'agit des corrélations doubles qui forment le tenseur de Reynolds. Résoudre le problème de fermeture consiste à trouver des relations permettant de relier ces nouveaux termes aux grandeurs moyennes. Comme nous l'avons signalé en introduction, des travaux ont été fait au sein du laboratoire sur le calcul des écoulements turbulents. Différentes modélisations ont été testées. Parmi elles, un modèle $k - \epsilon$ à bas Reynolds comportant deux équations de transport a été retenu pour la qualité de ses prédictions. C'est le modèle à bas Reynolds de Chien [11]. Nous avons adopté ce modèle pour cette première phase du développement du code en raison de sa simplicité par rapport aux modèles aux tensions de Reynolds. Une fois le code mis au point, d'autres modèles pourront être implémentés sans difficulté.

Dans cette section, nous allons présenter le modèle $k - \epsilon$ haut Reynolds standart, ainsi que la variante bas Reynolds de Chien [11].

3.3.1 Fermeture par un modèle $k - \epsilon$ à grand nombre de Reynolds

Le modèle $k - \epsilon$ est basé sur l'hypothèse de Boussinesq. Cette hypothèse relie les contraintes de Reynolds aux gradients de vitesse moyenne par l'intermédiaire du concept de viscosité turbulente:

$$-\overline{u_i \cdot u_j} = \nu_t \cdot \left(\frac{\partial \overline{U}_i}{\partial x_j} + \frac{\partial \overline{U}_j}{\partial x_i} \right) - \frac{2}{3} k \delta_{i,j} \quad (3.14)$$

Le terme $-\frac{2}{3} k \delta_{i,j}$ est ajouté pour assurer l'égalité des deux membres par contraction des indices i et j . Ce modèle remplace une inconnue tensorielle par deux inconnues scalaires. Le modèle $k-\epsilon$ a recours à deux équations de transport pour déterminer ces deux inconnues. La détermination de la viscosité turbulente suppose le calcul

d'une longueur et d'une vitesse caractéristiques des structures dissipatives. Elle est donnée par la formule:

$$\nu_t = \frac{C_\mu \cdot k^2}{\epsilon} \quad (3.15)$$

k l'énergie turbulente et ϵ le taux de dissipation sont déterminés à l'aide de deux équations de transport.

Equation de transport pour k

Pour déterminer cette équation, on modélise l'équation (3.12).

(1): n'est pas à modéliser.

(2): Le terme de production de l'énergie turbulente est explicité à l'aide de 3.14:

$$P = \overline{u_i u_j} \frac{\partial \overline{U}_i}{\partial x_j} = \nu_t \cdot \frac{\partial \overline{U}_i}{\partial x_j} \cdot \left(\frac{\partial \overline{U}_i}{\partial x_j} + \frac{\partial \overline{U}_j}{\partial x_i} \right)$$

(3) et (4): la diffusion turbulente due aux fluctuations de vitesse et de pression est modélisé à l'aide d'une hypothèse de type gradient:

$$\frac{\partial}{\partial x_k} \left(\overline{u_i u_i u_k} + \frac{\overline{p}}{\rho} u_k \right) = - \frac{\partial}{\partial x_k} \left(\frac{\nu_t}{\sigma_k} \frac{\partial k}{\partial x_k} \right)$$

(5): la diffusion visqueuse est explicite.

(6): la dissipation isotrope de l'énergie turbulente, est notée:

$$\epsilon = \nu \overline{\frac{\partial u_i}{\partial x_k} \frac{\partial u_i}{\partial x_k}}$$

On obtient finalement l'équation suivante:

$$\frac{\partial (\overline{U}_j \cdot k)}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \cdot \frac{\partial k}{\partial x_j} \right] + S_k \quad (3.16)$$

avec:

$$S_k = P - \epsilon \quad (3.17)$$

et

$$\nu_t = \frac{C_\nu \cdot k^2}{\epsilon} \quad (3.18)$$

Equation de transport pour ϵ

Au lieu d'être déduite de l'équation 3.13, l'équation pour ϵ est déduite par analogie avec l'équation pour l'énergie turbulente.

$$\frac{\partial(\bar{U}_j \cdot \epsilon)}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_\epsilon} \right) \cdot \frac{\partial \epsilon}{\partial x_j} \right] + S_\epsilon \quad (3.19)$$

avec:

$$S_\epsilon = C_{\epsilon 1} \frac{\epsilon}{k} P - \frac{\epsilon}{k} C_{\epsilon 2} \epsilon \quad (3.20)$$

3.3.2 Modèle $k - \epsilon$ à faible Reynolds

Le modèle $k - \epsilon$ précédent est basé sur l'hypothèse d'un nombre de Reynolds turbulent local $R_t = k^2/\nu\epsilon$ grand. Les écoulements turbulents comportent en général des zones à faible Reynolds. L'une des solutions pour traiter ces zones situées près des interfaces, consiste à utiliser des fonctions de parois. Ces fonctions sont construites à partir de données expérimentales. Elles permettent d'éviter un maillage trop serré dans les zones correspondantes. Il en résulte une économie substantielle en place mémoire.

Le principal défaut de ces fonctions est qu'elles ne tiennent pas compte de certaines situations spécifiques. Ces comportements proviennent par exemple de faibles gradients de vitesse à une des interfaces de l'écoulement.

L'autre solution consiste à modifier le modèle précédent afin de prendre en compte les spécificités locales des écoulements de paroi notamment.

Le nouveau modèle prend alors la forme suivante:

$$\frac{\partial(\bar{U}_j \cdot k)}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \cdot \frac{\partial k}{\partial x_j} \right] + P - D - \tilde{\epsilon} \quad (3.21)$$

$$\frac{\partial(\bar{U}_j \cdot \tilde{\epsilon})}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_\epsilon} \right) \cdot \frac{\partial \tilde{\epsilon}}{\partial x_j} \right] + C_{\tilde{\epsilon} 1} \frac{\tilde{\epsilon} P}{k} - C_{\tilde{\epsilon} 2} f \frac{\tilde{\epsilon} \epsilon}{k} + E \quad (3.22)$$

où $\epsilon = \tilde{\epsilon} + D$. Cette relation est retenue pour prendre en compte le fait que ϵ n'est pas nul à la paroi. Sa valeur à la paroi dépend de l'énergie turbulente k . Ce changement de variable permet de spécifier des conditions aux limites simples.

J. C. Monnier [43] a étudié ces modèles, on pourra donc se référer à sa thèse pour plus de détails sur les différents versions de modèles $k - \epsilon$ à bas Reynolds. Les différents termes pour le modèle de Chien sont donnés pour le cas d'une paroi normale à la direction y :

$$D = \frac{2\nu k}{y^2} \quad (3.23)$$

$$E = -\frac{2\nu k \exp(-0.5y^+)}{y^2} \quad (3.24)$$

avec

$$y^+ = \frac{u_\tau y}{\nu} \quad (3.25)$$

et

$$\nu_t = C_\mu \cdot f_\mu \frac{k^2}{\epsilon} \quad (3.26)$$

avec les valeurs suivantes pour les constantes du modèle:

$$\begin{aligned} C_\mu &= 0.09 \\ C_{\epsilon 1} &= 1.35 \\ C_{\epsilon 2} &= 1.80 \\ \sigma_k &= 1.0 \\ \sigma_\epsilon &= 1.3 \\ f_1 &= 1.0 \\ f_2 &= 1.0 - 0.22 \cdot \exp(k^2/\nu_t \epsilon) \\ f_\mu &= 1.0 - \exp(C_3 u_\tau y_p / \nu) \end{aligned} \quad (3.27)$$

3.3.3 Equations à résoudre et conditions aux limites

Equation de continuité:

$$\bar{U}_{i,i} = 0 \quad (3.28)$$

Equations de quantité de mouvement:

$$\bar{U}_j \bar{U}_{i,j} = ((\nu + \nu_t) \bar{U}_{i,j})_{,j} + S_{U_i} \quad (3.29)$$

S_{U_i} est le terme source de gradient de pression.

Equations du modèle turbulent:

$$\bar{U}_j \phi_{,j} = \left(\left(\nu + \frac{\nu_t}{\sigma_\phi} \right) \phi_{,j} \right)_{,j} + S_\phi \quad (3.30)$$

Où ϕ est soit l'énergie turbulent k soit le taux de dissipation ϵ .

On note Γ la frontière du domaine, et $X \in \Gamma$ le vecteur position d'un point de cette frontière.

Pour les équations de Navier-Stokes, les conditions aux limites sont de trois types:

1. La condition de sortie. Cette condition sera surtout utilisée pour marquer le caractère établi de l'écoulement à la traversée d'une frontière. On écrit en général pour les composantes de vitesse une condition de Neumann:

$$\frac{\partial U_i(X)}{\partial n} = 0$$

où n représente la direction normale à la frontière. Pour la variable de pression, on a une condition de Dirichlet:

$$p(X) = p_0$$

2. Les conditions à la paroi. Pour les composantes de vitesse, on a des conditions de Dirichlet:

$$U_i(X) = U_{i0}$$

Pour la pression, le système de maillage que nous avons adopté ne nécessite pas de conditions aux limites explicites. Cependant lors des procédures d'interpolation de la méthode multigrille que nous allons présenter plus loin, nous avons adopté une condition de type Neumann. Ce type de condition est également utilisé pour les maillages non décalés. On a alors:

$$\frac{\partial p(X)}{\partial n} = 0$$

où n représente la direction normale à la paroi.

Cette condition est aussi connue sous le nom de condition numérique pour la pression (voir [14]).

3. Les conditions pour un fluide entrant. Les conditions sont les mêmes que pour une paroi. La différence réside au niveau du traitement des vitesses tangentielles lors de l'implémentation.

3.4 Les équations du modèle électrochimique simplifié et les conditions aux limites

3.4.1 Le modèle simplifié

Pour écrire le modèle turbulent de la partie électrochimique, on applique aux équations (2.27) le même traitement statistique qu'aux équations d'hydrodynamiques.

Considérons l'espèce n . La décomposition de la vitesse est donnée en 3.1. La concentration C_n se décompose en:

3.4. LES ÉQUATIONS DU MODÈLE ÉLECTROCHIMIQUE SIMPLIFIÉ ET LES CONDITIONS

$$C_n(x_i, t) = \bar{C}_n(x_i) + c_n(x_i, t) \quad (3.31)$$

$$(3.32)$$

On reporte cette décomposition dans (2.27), puis on moyenne l'équation.

On obtient alors:

$$\bar{C}_{n,t} + \bar{U}_j \frac{\partial \bar{C}_n}{\partial x_j} = D_n(\bar{C}_{n,jj}) - (\overline{c_n u_j})_{,j} \quad (3.33)$$

Comme pour le modèle hydrodynamique, on a introduit des termes supplémentaires $\overline{c_n u_j}$.

Ces termes sont modélisés de la manière suivante:

$$-\overline{c_n u_i} = D_{nt}(\bar{C}_n)_{,i} \quad (3.34)$$

D_{nt} est le coefficient de diffusion turbulent pour l'ion n . On a en plus les relations suivantes:

$$\nu = D_n S_c \quad \nu_t = D_{nt} S_t \quad (3.35)$$

Où S_c est le nombre de Schmidt laminaire et S_t un nombre de Schmidt turbulent qui vaut un.

L'équation du modèle stationnaire turbulent s'écrit:

$$\bar{U}_j \bar{C}_{n,j} = \left(D_n + \frac{\nu_t}{S_t} \right) \bar{C}_{n,j,j} \quad (3.36)$$

3.4.2 Les conditions aux limites

Les conditions aux limites pour les équations de l'électrochimie sont en général plus complexes. En effet elles dépendent du caractère actif ou non de la paroi. Pour simplifier, dans un premier temps, nous considérons le système au courant limite de diffusion. Dans ce cas, tous les ions qui parviennent à la paroi réagissent immédiatement. On peut donc considérer que pour l'espèce à déposer, la concentration à la cathode est nulle.

1. Les conditions pour un fluide entrant sont des conditions de Dirichlet classiques:

$$C_i(X) = C_{i0}$$

Elles spécifient la concentration en ion du fluide régénéré.

2. Pour toutes les parois inactives (pas de réaction), on écrit une condition de Neumann.

$$\frac{\partial C_i(X)}{\partial n} = 0$$

où n représente la direction normale à la frontière.

3. Pour une paroi active (siège de réactions chimiques) on applique des conditions de Dirichlet:

$$C_i(X) = 0$$

3.5 Conclusion

Avec les équations données ci-dessus, nous disposons d'un modèle simplifié pour le comportement hydrodynamique et électrochimique d'une cellule. Ce modèle permet d'étudier le courant limite de diffusion pour des configurations simples. Il comporte six inconnues et six équations qui peuvent être séparées en deux groupes: Les équations 3.28, 3.29 et (3.30) forment le premier groupe. Elle décrivent la partie hydrodynamique du processus. Le deuxième groupe est uniquement constitué par l'équation de transport 3.36 pour le zinc. Les équations du premier groupe sont couplées et fortement non linéaires. La concentration du zinc n'apparaît dans aucune de équations du premier groupe. On peut donc séparer le calcul de ces deux groupes d'inconnus, à condition de commencer la résolution par le premier groupe. Les modèles de turbulence à bas Reynolds nécessitent un maillage serré près de la parois. Ils sont donc plus coûteux que les modèles à haut Reynolds.

La recherche d'une précision suffisante aboutit à une gestion très serrée des ressources machines disponibles. Ceci se fait par l'utilisation de divers algorithmes que nous allons étudier à partir du prochain chapitre.

Chapitre 4

Le problème de la résolution numérique du modèle d'écoulement

4.1 Introduction

Les équations de Navier-Stokes sont non-linéaires. Leur inversion passe par une linéarisation et donc un processus itératif impliquant la résolution de problèmes linéaires successifs. Des équations 3.28, 3.29, 3.30, 3.36 formant le modèle, seule l'équation de continuité n'est pas une équation de transport. Toutes les autres nécessitent pour leur discrétisation, un traitement particulier des termes convectifs. Les trois équations du modèle hydrodynamique de base sont fortement couplées notamment à travers la pression et l'équation de continuité. Ce couplage rend difficile toute procédure de découplage qui aurait pour objectif une résolution séparée des trois équations.

Si elle ne comporte pas de terme convectif, l'équation de continuité pose un problème d'une autre nature par l'intermédiaire de la variable de pression. En effet pour un fluide isovolume, la pression est connue à une constante près. Dans le cas d'un problème de Neumann ou dans celui d'un problème mixte, cette constante est fixée aux frontières de Neumann. Dans le cas d'un problème de Dirichlet, cette constante est indéterminée. Ce qui peut poser des difficultés lors de la résolution. Il faut cependant noter que même pour les problèmes de la première catégorie, le choix d'un algorithme de résolution couplé peut faire réapparaître une indétermination.

Les termes sources des équations du modèle de turbulence, par leur nature, rendent la résolution des équations délicate. Le principal problème est qu'il est difficile de prédire les propriétés locales de la solution comme on pourrait le faire en considérant l'ordre des termes différentiels pour une équation sans terme source. Les algorithmes qui ont besoin de cette information pour leur mise en oeuvre sont difficilement adaptables pour de telles équations.

4.2 La résolution des équations de Navier-Stokes

4.2.1 Rappels historiques

La première résolution numérique connue des équations pour un fluide visqueux date de 1933 par une méthode de relaxation. Peu avant, en 1910, Richardson [53] réalisa une expérience de résolution numérique du Laplacien par la même méthode. La méthode de relaxation consiste à construire un processus itératif. Dans ce processus, les points de maillage sont pris à tour de rôle. Et pour chaque point, on calcule pour l'inconnue une correction proportionnelle au résidu de son équation. L'intérêt d'une telle méthode réside dans la simplicité des opérations unitaires. Cette simplicité diminue le risque d'erreur de calculs lorsque ceux-ci sont effectués manuellement.

Le problème du Laplacien a reçu beaucoup d'attention dans l'histoire de l'analyse mathématique. Il constitue jusqu'à présent un problème modèle pour le test d'algorithme de résolution de problèmes linéaires. A l'époque où les méthodes de relaxation sont apparues, les seuls calculateurs existant étaient des machines électromécaniques. La majeure partie du travail se faisait par des groupes de personnes payées au laplacien résolu. En 1946, Southwell [58] développe une méthode de relaxation de résidus plus efficace. Elle consiste à traiter à chaque tour, le point pour lequel le résidu est le plus important en valeur absolue. Les premiers calculs à grande échelle sont effectués par cet auteur en 1955 pour le calcul de l'écoulement autour d'un cylindre pour un nombre de Reynolds de 1000. Ces travaux purent s'effectuer grâce au financement accordé par une compagnie privée à l'Imperial College de Londres. Entre-temps la deuxième guerre mondiale eu lieu. Pendant ce conflit, le calcul numérique a connu deux avancées importantes, dues en partie au même homme: John Von Neumann. Ce dernier fut à l'origine de l'architecture des ordinateurs dont beaucoup de constructeurs informatiques continuent de s'inspirer aujourd'hui. Il est surtout à l'origine des premiers travaux sur l'étude de la stabilité des schémas numériques au Laboratoire de Los Alamos en 1944 [66]. Il développa notamment une méthode systématique d'analyse des systèmes linéaires. Cette méthode permet de déduire des critères de stabilité pour les schémas numériques.

En 1966 Thoman et Szewczyk [62] réussirent à calculer un écoulement avec un nombre de Reynolds de 1.000.000 en utilisant un schémas aux différences décentrées. Jusque-là, les différentes expériences de calcul avaient buté sur une limite supérieure pour le nombre de Reynolds. Limite au-delà de laquelle des instabilités apparaissent dans la solution. C'était la naissance du schéma upwind du premier ordre. Son adoption vient notamment du fait que les résultats obtenus sont physiquement réalistes.

Les premières méthodes de résolution passaient par la détermination d'une équation de Poisson pour la pression. Le but ici était d'éviter une résolution directe de l'équation de continuité. L'apparition de l'opérateur de Laplace permettait ensuite d'utiliser des méthodes générales.

La méthode MAC de Harlow et Welch (1965) [32] est considérée comme l'une des

premières tentative de résolution des équations initiales. Sa principale innovation se situe au niveau de la disposition des variables sur le maillage. Les variables de vitesse sont décalées par rapport à la variable de pression. Cette disposition facilite la discrétisation de l'équation de continuité. La méthode utilise toujours une équation de Poisson pour la pression.

En 1972 Patankar et Spalding [46] s'affranchissent de l'équation pour la pression. Leur algorithme passe par une discrétisation des équations initiales. Une procédure de découplage appelée SIMPLE permet ensuite de déterminer séparément les différentes variables. Il est à noter que cette procédure permet de construire une équation pour la variable de pression. Equation qui a la forme d'une équation de Poisson discrète. L'avantage de cette transformation purement algébrique est qu'elle simplifie la spécification des conditions aux limites.

Jusqu'au début des années 80, les principales innovations ont consisté à améliorer la procédure de découplage algébrique des variables. Les premières résolutions couplées sont attribuées à Zedan et Schneider (1983) [55], ici le système linéaire est résolu par une procédure itérative SIP (Strongly Implicite Procedure) qui est du type factorisation LU approchée [59]. Constatant la robustesse de ces méthodes par rapport aux précédentes, les recherches se sont concentrées sur l'optimisation des coûts en temps de calcul et en place mémoire, qui sont la principale contrepartie de leur efficacité. Vanka [65] (1985) a étendu la méthode et l'a appliquée à la résolution d'écoulements turbulents.

4.2.2 Méthodes de résolution

Les ordinateurs ont permis l'émergence et le développement de l'analyse numérique comme un autre moyen de résolution des équations aux dérivées partielles. Plus qu'une révolution en elle-même, l'informatique a apporté un support et des outils permettant une manipulation plus simple d'objets mathématiques de plus en plus complexe.

Pour la résolution des équations aux dérivées partielles, l'objectif est la détermination de fonctions d'une ou plusieurs variables réelles. La première tâche consiste à se donner une représentation discrète (numérique) de la fonction inconnue. Il est ensuite important de fournir une méthode de reconstruction permettant d'obtenir une fonction continue à partir de la version échantillonnée. Cette version reconstruite doit être aussi proche que possible de la fonction d'origine. L'étape suivante consiste à construire les équations qui vont permettre la détermination des inconnues numériques. La dernière étape fournit les algorithmes pour la résolution de ces équations.

Pour la résolution des équations aux dérivées partielles, les méthodes numériques se divisent en trois catégories: La première catégorie comprend les méthodes de différences finies, de volumes finis et d'éléments finis. Les méthodes spectrales constituent la deuxième, les méthodes lagrangiennes la dernière.

Les méthodes de la première catégorie ont pour base d'approximation des fonctions localisées dans l'espace physique (polynômes). Ces fonctions ont l'avantage de bien représenter les forts gradients. Elles s'adaptent également bien aux géométries complexes, et facilitent la prise en compte des conditions aux limites.

Les méthodes spectrales s'appuient sur des fonctions bien localisées dans l'espace spectral et globale dans l'espace physique. Ces fonctions sont en général les fonctions propres des opérateurs différentiels constituant le système à résoudre. Ces méthodes sont plus précises pour un même nombre de degrés de liberté que les méthodes de la première catégorie. Elles ont cependant deux défauts principaux: Elles imposent un maillage uniforme, même pour les zones où la fonction est régulière. Avec un faible nombre de degrés de liberté, elles représentent très mal les fonctions irrégulières. Pour un aperçu des méthodes lagrangiennes, on se référera à Koren[68].

Dans la pratique certaines méthodes combinent les avantages des trois catégories décrites. Cependant, pour les codes du commerce, la méthode des éléments finis s'impose de plus en plus comme la référence. La méthode dispose d'un cadre théorique bien structuré [72]. De plus son application dans différents domaines de la physique (électromagnétisme, fluide, matériaux), permet d'envisager plus facilement l'étude de systèmes englobant tous ces phénomènes en même temps, et ce pour les géométries les plus complexes.

Sur un plan purement informatique, l'adoption de cette méthode permet déjà le partage de code avec d'autres applications, c'est notamment le cas en ce qui concerne les codes de maillage. Les structures de données qui constituent l'une des parties les plus délicates de la conception d'un code vont également vers une forme de standardisation. Une fois ces problèmes réglés, il reste celui des algorithmes d'inversion. C'est donc souvent que nous ferons référence à cette méthode par la suite. D'autant plus qu'elle apparaît aussi comme une généralisation de méthodes telles que celles des différences finies.

A présent nous allons décrire brièvement les trois principales étapes de la résolution des modèles.

Le maillage

Les ordinateurs sont conçus pour la manipulation de grandeurs discrètes. Les équations aux dérivées partielles étant des modèles continus, leur discrétisation est un préalable à toute méthode faisant appel aux ordinateurs pour la résolution. Cette discrétisation comprend une phase de maillage de l'espace-temps suivie de la discrétisation des variables inconnues du problème.

Le type de maillage dépend du type de domaine à traiter. Ici nous considérons trois types de domaines de calcul. Le premier est constitué par les domaines simples, rectangulaires, cylindriques, sphériques etc. Le second est constitué par une réunion des précédents. Le dernier niveau est constitué par les géométries qui ne peuvent être

classées dans aucune des catégories précédentes. La plus grande partie des problèmes sont en fait dans cette dernière catégorie. L'utilisation de géométries simples peu cependant donner des indications utiles pour les problèmes complexes.

Les techniques de maillage forment un domaine important de l'analyse numérique. Ce domaine développe ses propres techniques et de plus en plus de logiciels lui sont entièrement dédiés. On peut en voir une liste à la fin de l'ouvrage de P.L. George [27]. Des formats de données ont également été définis. Ces formats facilitent la communication avec les codes de calcul et les codes de CAO, permettant ainsi de séparer les problèmes de maillage et de résolution des équations.

Lorsque le domaine est quelconque, le maillage non structuré constitue la seule solution. Pour ces maillages, mis à part la méthode des éléments finis, toutes les autres méthodes sont plus complexes à mettre en oeuvre. La méthode des volumes finis utilise de plus en plus ce type de maillage.

Lorsque le domaine est constitué d'une réunion de domaines simples, on peut procéder de deux manières différentes: Dans la première, on considère un domaine en bloc. Dans ce cas on peut effectuer un maillage non structuré. L'autre solution consiste à englober le domaine dans un domaine simple, puis à effectuer un maillage structuré si les sous-domaines s'y prêtent. Ensuite, tous les points de maillage extérieurs au domaine réel sont désactivés lors des procédures de discrétisation (figure 4.1). Cette méthode n'est plus efficace lorsque le nombre de points à désactiver devient important par rapport au nombre de points actifs. Une autre approche consiste à choisir une organisation en sous-domaines (figure 4.2 et 4.3). Dans ce cas l'occupation mémoire est optimale. Par contre les procédures de raccordement entre les maillages de deux sous domaines voisins demande des soins plus ou moins importants suivant le type de sous-domaine.

Dans la présente étude, la géométrie du domaine d'écoulement sera dans un premier temps modélisée sous la forme d'un rectangle. Ce domaine ne présente aucune difficulté ni pour le maillage ni pour les autres opérations. Cependant, pour donner un peu plus de généralité à notre code, nous allons considérer pour l'architecture, des domaines constitué de réunions de rectangles. L'extension à des domaines permettant un maillage structuré orthogonal pourra se faire par la suite. Pour cette phase de notre travail, la procédure de maillage va donc se réduire à une opération de maillage de segments de droites.

L'interpolation

Le maillage du domaine produit une partition de ce dernier en un ensemble de domaines élémentaires. Le but est de trouver une quantité de paramètres la plus petite possible pour représenter une fonction à valeur continue dans le domaine. Pour arriver à cet objectif, les domaines élémentaires sont répartis en tenant compte de la structure locale de la solution. La discrétisation des inconnues (des fonctions)

se fait en choisissant pour chaque domaine élémentaire un ensemble de nombre pour représenter la solution. L'ensemble des nombres pour tous les degrés de liberté forment la représentation discrète de la fonction.

Cette représentation, adaptée pour les manipulations par les ordinateurs ne l'est pas pour tous les usages. La première limitation concerne l'interprétation. L'aspect visuel d'une représentation graphique continue est plus agréable que sa version discrète. La deuxième limitation concerne les opérateurs différentiels qui forment les équations du modèle à résoudre. Ces opérateurs agissent sur des fonctions continues et non sur des fonctions quantifiées. Pour ces deux raisons au moins, l'interpolation est une opération indispensable pour le processus de résolution. L'interpolation permet d'obtenir une représentation explicite de la fonction inconnue à partir d'une part des degrés de liberté précédemment définis, et d'autre part de fonctions usuelles des variables d'espace et de temps dans le cas de problèmes instationnaires. On obtient ainsi une représentation de la fonction sur laquelle on peut étendre les opérations usuelles de différenciation et d'intégration des fonctions utilisées.

L'interpolation des fonctions appartient à une branche importante des mathématiques: celles de l'approximation des fonctions. Parmi les méthodes d'interpolation on peut citer celles qui utilisent les fonctions sinusoidales ou les fonctions splines dont la conception assistée par ordinateur de formes géométriques a favorisé l'essor. En fait la majorité des méthodes utilisent les fonctions polynômiales comme fonctions d'interpolation. Elles sont simples à manipuler et se prêtent bien à l'interpolation par morceaux, propre aux maillages non-uniformes.

Pour des données comportant de fortes variations, les fonctions polynômiales ont le défaut de produire des oscillations sur la fonction interpolée. Nous reviendrons sur ce problème dans le chapitre réservé à la discrétisation des équations.

La détermination des équations discrètes

Après la phase d'interpolation, on a une approximation continue de la fonction inconnue avec beaucoup moins de paramètres que la fonction originale. Les inconnues sont représentés par les degrés de liberté définis dans la partie précédente. Les équations à résoudre peuvent maintenant s'exprimer à partir de ces degrés de liberté et des variables continues d'espace et de temps. La détermination des équations discrètes va donc consister à éliminer les variables continues d'espace et de temps de manière à obtenir des équations algébriques.

Cette dernière étape se fait le plus souvent à l'aide d'une procédure d'intégration. La forme la plus générale de cette opération est donnée par la méthode des résidus pondérés (Galerkin) dont on peut trouver une présentation dans une collection publiée par le CEA sur l'analyse numérique [20]. La méthode de Galerkin est utilisée aussi bien par la méthode des éléments finis que par les méthodes spectrales et les méthodes lagrangiennes. Pour plus d'information on pourra consulter l'ouvrage de Koren [68].

La résolution des équations discrètes.

La discrétisation du modèle aboutit à un système de plusieurs équations à plusieurs inconnues linéaires ou non-linéaires selon le modèle original. Les équations sont toutes couplées les unes aux autres et devraient être résolues simultanément. En fait, elles sont rarement résolues de cette manière. Soit parce que le système est non-linéaire, soit parce que le nombre d'équations est trop important. Les méthodes utilisées vont des plus simples pour les algorithmes explicites aux plus lourdes pour les méthodes globales.

La conception d'un algorithme consiste en général à décomposer le problème en une suite de sous-problèmes dont la résolution séparées donne la solution au problème initial. La plus simple consiste à traiter chaque équation discrète comme un problème indépendant des autres. C'est la méthode explicite. Lorsque le modèle initial comporte plusieurs variables, il est parfois judicieux de regrouper les variables discrètes de même nature comme c'est le cas dans l'algorithme SIMPLE de Patankar [45]. On a enfin les regroupement par zones [7], [65]. Ici le critère de découplage choisi est "géographique". Les points sont regroupé par zones comme ont le verra plus loin. Toutes ces méthodes de découplage ont l'avantage de donner des algorithmes peu onéreux en place mémoire. Leur convergence est par contre souvent très lente.

D'autres algorithmes ont pour objectif d'accélérer les méthodes décrites plus haut: Ce sont les méthodes de conditionnement de type gradient conjugué ou encore les méthodes des sous-espaces de Krilov [68]. Une dernière catégorie de méthodes de résolution est constituée par les méthodes multigrilles. Ces méthodes utilisent plusieurs maillages du même domaine pour accélérer la convergence des méthodes explicites.

4.2.3 Architecture des ordinateurs et problèmes informatiques

Jusqu'au milieu des années cinquante, la majeure partie des calculs se sont fait manuellement. Les premiers ordinateurs d'usage général sont apparus en 1940. Les besoins les plus importants venaient du déchiffrement des codes pendant la guerre et des recherches sur la bombe atomique. Ensuite, les besoins de l'administration, des industries du secteur tertiaire ont soutenu l'industrie informatique naissante. Plus tard la conquête spatiale, l'aéronautique et les universités ont adopté ces nouvelles machines. Des entreprises comme Cray research, Cyber, et d'autres plus récentes se sont orientées vers la construction de superordinateurs pour les besoins en calcul intensifs. Au début des années 80, les limites des possibilités d'intégration des composants électroniques commençant à se faire sentir, les architectures vectorielles sont apparues. Vers la fin des années 90 les besoins toujours plus importants sont de plus en plus traités avec des machines à architecture parallèle.

A côté de cette évolution des machines, les langages et les environnements de pro-

grammation ont beaucoup changés. Le langage de programmation scientifique le plus répandu est le fortran. Cependant le calcul scientifique s'est beaucoup vulgarisé comme outil de travail dans la conception de nouveaux produits. Les entreprises commerciales ont choisi d'autres langages en fonction des problèmes de développement de codes. C'est ainsi que le C et le Pascal ont bénéficié d'efforts de standardisation ainsi que de structures de données plus riches, dues aux exigences des autres domaines du logiciel.

L'industrie informatique s'est beaucoup développée. Le coût de la puissance de calcul brute a baissé d'une manière extrêmement rapide. L'industrie du logiciel a alors pris le pas sur celle du matériel. L'une des raisons de cette situation est que la conception de logiciels tirant le meilleur parti de machine en perpétuelle évolution est devenue une tâche ardue. S'adapter à un environnement économique de plus en plus concurrentiel en essayant de tirer le meilleur parti des nouvelles machines comporte des risques qui sont de mieux en mieux perçus par les différents acteurs.

En ce qui concerne le calcul scientifique, le portage d'un patrimoine logiciel de plus en plus lourd d'une machine à une autre, pose des problèmes cruciaux. En effet le passage des architectures scalaires aux architectures vectorielles a demandé de gros efforts de réécriture des codes. Les nouvelles architectures parallèles vont demander des efforts encore plus importants. Mis à part l'évolution de la puissance qui pose également des problèmes, chaque changement d'architecture, oblige les programmeurs à changer de modèle de machine. Les langages évoluent et les anciens réflexes de programmation peuvent devenir une barrière infranchissable vers les nouveaux besoins. L'importance croissante du parc des micro-ordinateurs et leur sous utilisation ouvre de nouvelles pistes. En effet un réseau de plusieurs petites machines peut développer une puissance de calcul considérable dans le cadre d'une utilisation en machine parallèle virtuelle. Des codes existent déjà pour gérer l'échange de données dans ce type de configuration.

4.3 Démarche choisie

Le but de notre travail est de construire un code de calcul capable de traiter sans modifications des géométries composées de réunions de domaines rectangulaires. Ce, pour la résolution des équations de Navier-Stokes et d'autres équations de transport de scalaire avec termes source. Le code permet la résolution des équations moyennées avec un modèle de turbulence. Il permet également la résolution d'équations de transport pour les concentrations d'espèces ioniques. Dans le cadre d'une utilisation pour tester des modèles de turbulence, le code doit permettre une extension facile pour un nombre d'équations plus grand. Et enfin le plus important, le code doit être optimal du point de vue de sa consommation mémoire et temps de calcul.

Le choix du schéma hybride pour la mise au point du code n'est pas définitif. Le schéma QUICK a également été étudié et son implantation ne devrait pas poser

de problème majeur. En ce qui concerne la robustesse, nous avons opté pour une résolution couplée des équations de l'hydrodynamique. Le but est de contrôler de près la conservation de la masse qui est un point important pour les écoulements incompressibles.

Pour le type de domaine à traiter (réunion de domaines rectangulaires), nous avons choisi une organisation en sous domaines. Tout domaine peut alors être représenté sous la forme d'une liste de rectangles reliés les uns aux autres. On peut alors développer des modules généraux pour le traitement d'un rectangle. Les modules d'inversion des systèmes linéaires ont été conçus pour tirer un maximum de profit du caractère structuré du maillage.

Nous avons choisi de transférer dans un premier temps le code SIMPLER en langage C. Par la suite, la recherche d'algorithmes d'inversion rapide nous a amené à étudier deux autres méthodes: La méthode par zones qui convient parfaitement aux géométries rectangulaires, et la méthode multigrille qui a prouvé son efficacité pour les écoulements laminaires.

La programmation des ordinateurs a commencé avec le langage binaire. Ensuite, on est passé au langage assembleur. Le fortran est l'un des premiers langages de haut niveau à avoir été conçu pour un usage général. Jusqu'à présent, ce langage reste le plus utilisé dans la communauté scientifique. Plusieurs langages ont été développés par la suite, parmi lesquels les langages C et Pascal. Le langage C a été conçu pour implémenter le système d'exploitation UNIX encore très utilisé. L'évolution des langages a surtout été dictée par la complexité croissante des projets, complexité elle-même possible grâce à la baisse du coût de la puissance des ordinateurs. Lorsque nous avons commencé ce projet, le langage C possédait à nos yeux la richesse nécessaire à l'implémentation d'un code de complexité moyenne. Aujourd'hui le fortran 90 possède des qualités similaires à celle du langage C. Mais l'extension du C par une version objet appelé C++ ouvre déjà de nouvelles perspectives. Nous sommes convaincus qu'après avoir résolu de nombreux problèmes dans d'autres domaines informatiques, les nouveaux langages de programmation peuvent apporter beaucoup au monde scientifique. Ces apports concernent aussi bien la structuration des codes que leur maintenance et leur évolution.

Partie II

Discrétisation des équations



Chapitre 5

Discrétisation des équations

5.1 Introduction

Soit à résoudre le problème de Dirichlet suivant:

$$L(\phi(X)) = f(X) \tag{5.1}$$

avec,

$$\phi(X) = \varphi(X) \quad X \in \Gamma \tag{5.2}$$

avec $X = (x, y) \in \Omega$, cet ensemble désignant pour nous le domaine de calcul et Γ frontière Ω .

Dans ce chapitre nous allons étudier le processus de discrétisation du problème à savoir la définition du maillage et l'obtention des équations discrètes.

5.1.1 Maillage

Le domaine global est représenté sous la forme d'une réunion de sous-domaines rectangulaires. Le maillage décalé que nous avons implémenté est décrit sur la figure 5.1. On peut voir sur ce schéma, la disposition des inconnues u , v et p , ainsi que les différents maillages x , xu , y et yv . Dans le cas d'une marche descendante par exemple (figure 4.3), chacun des rectangles est maillé séparément. Les maillages sont de type rectangulaire non uniforme. Pour contrôler la progression du pas, on choisit un coefficient de répartition α . Pour un segment de longueur L , le maillage se fait de la manière suivante: Supposons que l'on veuille mailler le segment au moyen de N pas de maillage en serrant au début. Alors le i^{ieme} pas est donné par:

$$\Delta x(i) = \frac{\alpha^{N-i+1}}{S} \cdot L$$
$$S = \sum_{k=1}^{k=N} \alpha^k \tag{5.3}$$

Les autres possibilités de répartition se déduisent facilement de celle-ci. Etant donné que nous avons choisi d'utiliser un maillage décalé, dans chaque direction x ou y on doit calculer deux maillages correspondant l'un aux variables non décalées et l'autre aux variables décalées. On obtient les quatre séries de valeurs suivantes:

x et xu puis y et yv . Chacun de ces vecteurs est numéroté de la manière suivante:

$$\begin{aligned}
 x(0) &= -\frac{\Delta x(1)}{2} \\
 xu(1) &= 0 \\
 xu(i+1) &= xu(i) + \Delta x(i) \\
 x(i) &= 0.5(xu(i+1) + xu(i))
 \end{aligned} \tag{5.4}$$

$$\begin{aligned}
 y(0) &= -\frac{\Delta y(1)}{2} \\
 yv(1) &= 0 \\
 yv(j+1) &= yv(j) + \Delta y(j) \\
 y(j) &= 0.5(yv(j+1) + yv(j))
 \end{aligned} \tag{5.5}$$

On distingue alors trois types de variables: la composante $u(xu, y)$ de la vitesse du fluide suivant x , la composante $v(x, yv)$ suivant y et la variable de pression $p(x, y)$ ainsi que toutes les autres variables scalaires $k(x, y)$, $\epsilon(x, y)$, $C_i(x, y)$. Ces trois types de variables correspondent à trois types d'éléments différents (voir figure 5.1.): $\Omega_e(i, j)$, $\Omega_e^u(i, j)$ et $\Omega_e^v(i, j)$ centrés respectivement aux points $(x(i), y(j))$, $(xu(i), y(j))$ et $(x(i), yv(j))$.

Il est ensuite nécessaire d'effectuer le raccordement des maillages pour les rectangles voisins. Pour cela, il faut assurer le raccordement des pas de maillage autour de la frontière entre sous-domaines (figure 4.3).

Nous allons donc adopter les notations suivantes:

On considère un domaine rectangulaire divisé en N_x pas dans la direction x et N_y pas suivant y .

$\Omega_e(i, j)$ est un rectangle élémentaire de côté $(\Delta x_i, \Delta y_j)$.

$$\begin{aligned}
 X_{i,j} &= (x_i, y_j) \\
 X_j &= (X_{i,j})_{1 \leq i \leq N_x} \\
 X_i &= (X_{i,j})_{1 \leq j \leq N_y} \\
 X_h &= (X_i)_{1 \leq i \leq N_x} \\
 X_h &= (X_j)_{1 \leq j \leq N_y}
 \end{aligned} \tag{5.6}$$

Une fois le maillage connu, la fonction ϕ , qui peut être n'importe laquelle des inconnues du modèle, sera numérisée pour chaque point $X(i, j)$, les échantillons ainsi choisis $\phi_{i,j}$ sont appelés degrés de liberté.

$$\begin{aligned}
 \phi_j &= (\phi_{i,j})_{1 \leq i \leq N_x} \\
 \phi_i &= (\phi_{i,j})_{1 \leq j \leq N_y} \\
 \phi_h &= (\phi_i)_{1 \leq i \leq N_x} \\
 \phi_h &= (\phi_j)_{1 \leq j \leq N_y}
 \end{aligned} \tag{5.7}$$

ϕ_h forme l'ensemble des degrés de libertés. Evidemment dans le cas où l'on a plusieurs fonctions inconnues, il faut spécifier les degrés de liberté pour chacune des fonctions. ϕ_h est une représentation échantillonnée de ϕ , celle à partir de laquelle on peut reconstruire par interpolation une version continue approchée. Pour la suite il faut numéroter les variables du maillage décalé. Le but est de regrouper pour un point (i,j) un degré de liberté par variable. Le choix standard est le suivant:

$$\begin{aligned}
 u(i, j) &= u(xu(i), y(j)) \\
 v(i, j) &= v(x(i), yv(j)) \\
 p(i, j) &= p(x(i), y(j))
 \end{aligned} \tag{5.8}$$

Beaucoup d'autres choix sont possibles, et il est parfois nécessaire pour la résolution des équations d'adopter d'autres types de numérotation comme on le verra un peu plus loin.

5.1.2 Interpolation de la solution et de ses dérivées

L'objectif de l'interpolation est de donner un procédé pour calculer $\phi(X)$ à partir de ϕ_h . Les polynômes sont les plus utilisés pour cela. L'évolution de la fonction au voisinage d'un point est représentée par un polynôme dont les coefficients sont déterminés à partir des degrés de liberté voisins. Plus le degré du polynôme sera fort, plus il intègrera d'échantillons voisins. Les termes différentiels sont obtenus par simple dérivation de ce polynôme. D'une manière générale, on construit un ensemble de fonctions $N_{i,j}^\phi(X)$, dont le support est $\Omega_\epsilon^\phi(i, j)$.

La fonction interpolée peut alors s'exprimer sous la forme:

$$\phi(X) = \sum_{(i,j)} \phi(i, j) N_{i,j}^\phi(X) \tag{5.9}$$

La méthode des volumes finis utilise souvent plusieurs formules d'interpolation suivant le terme à traiter dans l'équation. Dans notre cas les termes convectifs sont traités différemment des termes diffusifs et du terme source, nous y reviendrons après cette introduction.

5.1.3 La méthode des résidus pondérés

Les équations à résoudre peuvent être mise sous la forme générale suivante:

$$\frac{\partial(u\phi)}{\partial x} + \frac{\partial(v\phi)}{\partial y} = \frac{\partial}{\partial x} \left(\frac{\Gamma \cdot \partial\phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\Gamma \cdot \partial\phi}{\partial y} \right) + S(\phi) \quad (5.10)$$

Le résidu de cette équation au point $X(x, y)$ s'écrit:

$$Res(\phi(X)) = \frac{\partial(u\phi)}{\partial x} + \frac{\partial(v\phi)}{\partial y} - \frac{\partial}{\partial x} \left(\frac{\Gamma \cdot \partial\phi}{\partial x} \right) - \frac{\partial}{\partial y} \left(\frac{\Gamma \cdot \partial\phi}{\partial y} \right) - S(\phi) \quad (5.11)$$

Résoudre l'équation 5.10 revient à déterminer la fonction $\phi(X)$ pour laquelle le résidu est nul pour tout point X . Ceci est en fait la définition de la solution classique. La méthode des éléments finis et, de plus en plus, les autres méthodes déterminent la solution à l'aide d'une formulation faible du problème:

$$\int_{\Omega_e^\phi(i,j)} \omega_{i,j}^\phi(X) \cdot Res(\phi(X)) \cdot d\Omega = 0 \quad (5.12)$$

avec $\omega_{i,j}^\phi \in Test(\Omega)$

On peut revenir au problème classique en choisissant par exemple pour $Test(\Omega)$ l'ensemble des fonctions de Dirac sur Ω .

Pour la méthode des volumes finis on a:

$$\begin{aligned} \omega_{i,j}^\phi(X) &= 1 \quad \text{si } X \in \Omega_e^\phi(i,j) \\ \omega_{i,j}^\phi(X) &= 0 \quad \text{sinon} \end{aligned} \quad (5.13)$$

ou $\phi = u, v$ ou p avec $(i, j) \in [1, N_x] \times [1, N_y]$.

Avec ces définitions, l'équation 5.10 devient pour une cellule élémentaire (figure 5.2):

$$\begin{aligned} \Delta y_p (u_e \phi_e - u_w \phi_w) + \Delta x_p (u_n \phi_n - u_s \phi_s) &= \Delta y_p \left(\Gamma \cdot \frac{\partial\phi}{\partial x} \Big|_e - \Gamma \cdot \frac{\partial\phi}{\partial x} \Big|_w \right) + \\ \Delta x_p \left(\Gamma \cdot \frac{\partial\phi}{\partial y} \Big|_n - \Gamma \cdot \frac{\partial\phi}{\partial y} \Big|_s \right) &+ \int_{\Omega_e^\phi} S(\phi) d\Omega \end{aligned} \quad (5.14)$$

A ce stade, on a pratiquement toutes les équations du modèle discret. Il suffit d'utiliser les formules d'interpolation précédentes pour déterminer les termes de l'équation 5.14 qui ne correspondent pas directement aux valeurs fixées par la discrétisation. Sur la figure 5.2, les lettres majuscules correspondent à des degrés de liberté. Les lettres minuscules correspondent aux positions pour lesquelles l'interpolation est nécessaire.

5.2 Le problème des termes convectifs

5.2.1 Schémas centré et décentré

L'approximation des termes convectifs dans les équations de transport a fait l'objet d'importants travaux. En effet, dès les premières tentatives de résolution de ce type d'équations, on a constaté que les deux schémas linéaires disponibles (différences centrées et décentrées), avaient chacun leurs défauts [45]. En fait chacun de ces deux schémas était approprié pour des cas de figure particuliers.

Nous allons illustrer les problèmes posés par la discrétisation du terme convectif. Pour cela on considère l'équation:

$$\frac{d(u\phi)}{dx} = \frac{d}{dx} \left(\frac{\Gamma \cdot d\phi}{dx} \right) \quad (5.15)$$

$x \in [0, L]$ et pour les conditions aux limites $\phi(0) = \phi_0$ et $\phi(L) = \phi_L$

On intègre l'équation sur le volume de contrôle de la variable dépendante pour obtenir la formulation volume finis de l'équation à résoudre.

$$(u\phi)_e - (u\phi)_w = \left(\frac{\Gamma \cdot \partial\phi}{\partial x} \right)_e - \left(\frac{\Gamma \cdot \partial\phi}{\partial x} \right)_w \quad (5.16)$$

A présent, nous allons définir deux formules d'interpolation pour $x \in [x_i, x_{i+1}]$, une pour une interpolation linéaire correspondant au maillage centré, et l'autre pour un maillage décalé vers l'amont. La formule générale est:

$$\phi(x) = N_i(x)\phi_i + N_{i+1}(x)\phi_{i+1} \quad (5.17)$$

Avec pour le schéma centré:

$$N_i(x) = \frac{x_{i+1} - x}{x_{i+1} - x_i}$$

et

$$N_{i+1} = \frac{x_{i+1} - x}{x_{i+1} - x_i}$$

et pour le schéma décalé vers l'amont:

$$N_i(x) = 1$$

et

$$N_{i+1} = 0$$

Si on applique la méthode d'interpolation centrée (linéaire) pour approximer les termes de l'équation 5.16 sur un maillage uniforme, $x_{i+1} - x_i = h$ avec $0 \leq i \leq n$ et $(n+1)h = L$, on obtient l'équation suivante:

$$\frac{u(\phi_{i+1} - \phi_{i-1})}{2} = \Gamma \left(\frac{\phi_{i+1} - \phi_i}{h} - \frac{\phi_i - \phi_{i-1}}{h} \right) \quad (5.18)$$

la solution exacte de ce système d'équation est:

$$\phi_i = A + BC^i$$

avec

$$A = \phi_0 - B, \quad B = -\frac{\phi_L - \phi_0}{1 - C^{n+1}}, \quad C = \frac{1 + P_{eh}}{1 - P_{eh}}$$

et

$$P_{eh} = \frac{uh}{2\Gamma}$$

le nombre de Peclet local.

Pour des valeurs de $P_{eh} > 1$ la solution oscille. Oscillations qui n'ont aucune origine physique. Comme on peut le voir sur la solution, à L et u fixés, le niveau des oscillations dépend du gradient $\Delta\phi = \phi_L - \phi_0$. On voit également qu'une réduction du pas de maillage h permettra toujours de réduire l'amplitude des oscillations. On retrouve ici le problème de la fréquence d'échantillonnage et du spectre du signal. A L et u fixés, le spectre de $\phi(x)$ dépend directement de $\Delta\phi$. Suivant que cette valeur est forte ou faible, le spectre sera large ou étroit. Nécessitant ainsi une fréquence d'échantillonnage $1/h$ plus ou moins grande, comme le spécifie le Théorème de Shannon.

Il se pose un autre type de problème même pour les faibles valeurs de $\Delta\phi$. Cette fois, la difficulté est liée à la valeur de u la vitesse de convection. Lorsque u augmente, le système linéaire est mal conditionné. On peut voir en se reportant à l'équation 5.18 que le coefficient de ϕ_i peut devenir plus faible que les termes non-diagonaux. Les méthodes d'inversion directes fournissent une solution en ayant recours au pivotage partiel ou total. Les méthodes itératives sans préconditionnement échouent. Mais une fois de plus on peut éliminer ce problème en réduisant le pas de maillage h . On voit que la précision de la solution et les impératifs liés à la méthode de résolution retenue, imposent des conditions différentes sur la valeur de h .

De ce qui vient d'être dit, on déduit que la nécessité de trouver de meilleures formulations pour les termes convectifs est directement liée aux limitations en ressources mémoire et en temps de calcul.

A présent, considérons un schéma décalé vers l'amont pour l'évaluation du terme convectif et le schéma centré pour les termes diffusifs, on obtient l'équation suivante:

$$u(\phi_i - \phi_{i-1}) = \Gamma\left(\frac{\phi_{i+1} - \phi_i}{h} - \frac{\phi_i - \phi_{i-1}}{h}\right) \quad (5.19)$$

la solution exacte de ce système d'équation est :

$$\phi_i = A + BC^i$$

avec

$$A = \phi_0 - B, \quad B = -\frac{\phi_L - \phi_0}{1 - C^{n+1}}, \quad C = 1 + 2P_{eh}$$

Cette fois-ci la solution est stable tant que P_{eh} est positif. Cependant, ce schéma est d'ordre un alors que le précédent est d'ordre deux.

5.2.2 Le schéma hybride

L'une des caractéristiques des phénomènes de transport est que les zones de forts gradients sont souvent localisées. C'est le cas par exemple des ondes de choc. Pour ce type d'écoulement, l'application brute du théorème de l'échantillonnage est inutilement coûteuse. En effet, pour les zones où la fonction est relativement lisse, il y a une équivalence entre la précision p de la méthode de reconstruction et le pas d'échantillonnage h . La programmation d'un schéma de reconstruction plus précis, permet de diminuer le nombre de variable à manipuler. Ce qui est intéressant pour les ressources mémoire, mais aussi dans certains cas pour le temps de calcul. L'objectif principal dans l'amélioration de l'estimation du terme convectif est de permettre l'utilisation de schémas de grande précision même pour des écoulements comportant de forts gradients. Pour cela, la démarche générale consiste à incorporer un "contrôleur" au schéma de base. Ce contrôleur a pour fonction de permettre une détection automatique et un traitement adéquat des zones de forts gradients. Le schéma hybride est une combinaison du schéma centré et du schéma décalé. Son application au problème modèle donne:

$$A_p^\phi \phi_P - A_w^\phi \phi_W - A_e^\phi \phi_E = 0 \quad (5.20)$$

avec $A_p^\phi = A_w^\phi + A_e^\phi$ et

$$\begin{aligned} A_w^\phi &= D_w \text{Max}(0, (1 - \frac{|P_w|}{2})) + \text{Max}(0, F_w) \\ D_w &= \frac{\Gamma}{h} \quad F_w = u \quad P_w = \frac{F_w}{D_w} \end{aligned} \quad (5.21)$$

$$\begin{aligned} A_e^\phi &= D_e \text{Max}(0, (1 - \frac{|P_e|}{2})) + \text{Max}(0, -F_e) \\ D_e &= \frac{\Gamma}{h} \quad F_e = u \quad P_e = \frac{-F_e}{D_e} \end{aligned} \quad (5.22)$$

La présence d'un interrupteur est formalisée ici par l'existence de la fonction Max . Ces formules condensées regroupent à la fois le traitement des termes convectifs et diffusifs. Le paramètre de Peclet P_e contrôle l'influence respective de ces deux termes dans l'équation discrétisée.

5.2.3 Différents types d'interpolation du terme convectif

De nombreux schémas ont depuis amélioré l'approximation des termes convectifs. Le choix n'est plus tranché entre la diffusion et les oscillations numériques. Leonard a proposé le schéma QUICK [37] qui est un schéma décentré d'ordre deux. La technique des limiteurs de flux pour les schémas TVD [35], [14], permet d'utiliser des

approximations de précision arbitrairement élevée sans la contrepartie des oscillations numériques. Les schémas de la famille ENO [35] sont également de plus en plus appréciés. Leur stratégie consiste brièvement à adapter l'ordre d'interpolation à la régularité de la fonction. Cette régularité étant mesurée à l'aide de différences divisées. Ici, le concept de contrôleur est plus général. En effet il se base sur la structure locale de la solution et non pas sur le sens de la convection.

Ici nous allons donner à titre d'exemple la formule d'interpolation pour le schéma décentré d'ordre deux "QUICK". Les notations sont celle de Zhu et Rodi [71] (figure 5.5).

$$\phi_w = U_w^+ [(1 + c_1)\phi_W - c_1\phi_{WW}] + U_w^- [(1 + c_3)\phi_P - c_3\phi_E] \quad (5.23)$$

Le sens du décentrement est contrôlé par le signe de la composante de la vitesse suivant x .

$$c_1 = \frac{\Delta x_W}{\Delta x_W + \Delta x_{WW}}, \quad c_2 = \frac{\Delta x_W}{\Delta x_W + \Delta x_P}, \quad c_3 = \frac{\Delta x_P}{\Delta x_P + \Delta x_E} \quad (5.24)$$

sont les facteurs d'interpolation. et

$$U_w^+ = 0.5(1 + |U_w|/U_w), \quad U_w^- = 1 - U_w^+ \quad (U_w \neq 0) \quad (5.25)$$

permet de calculer le sens du décentrement pour la face w .

Les termes de diffusion peuvent être estimés par un schéma de différences centrées sans formule d'interpolation explicite.

5.2.4 Formulation complète des équations

De toutes les équation du modèle l'équation de continuité est la plus simple à discrétiser. Grace au maillage décalé, on obtient l'équation suivante qui est la plus précise que l'on puisse avoir dans la mesure où aucune interpolation n'est nécessaire:

$$\Delta y_p(u_E - u_P) + \Delta x_p(v_N - v_P) = 0 \quad (5.26)$$

Toutes les autres équations sont des équations de transport. Elles sont traitées suivant la formule appliquée pour le problème modèle décrit plus haut avec le schéma hybride. En effet l'équation 5.14 peut être séparé en terme intégrés suivant x et en termes suivant y . On peut alors appliquer les traitement du problème modèle à chacun de ces deux groupes de termes. On obtient les équations suivantes pour lesquelles les coefficients sont explicités en annexe A.1.

Les équations de transport pour u et v :

$$A_p^u u_P - (A_s^u u_S + A_n^u u_N + A_w^u u_W + A_e^u u_E) + \Delta y_p(p_P - p_W) = 0 \quad (5.27)$$

5.3. CONDITIONS AUX LIMITES ET TERMES SOURCES DU MODÈLE DE TURBULENC

$$A_p^v v_P - (A_s^v v_S + A_n^v v_N + A_w^v v_W + A_e^v v_E) + \Delta x_p (p_P - p_S) = 0 \quad (5.28)$$

Les équations de transport pour k et ϵ :

$$A_p^k k_P - (A_s^k k_S + A_n^k k_N + A_w^k k_W + A_e^k k_E) + S_k = 0 \quad (5.29)$$

$$A_p^\epsilon \epsilon_P - (A_s^\epsilon \epsilon_S + A_n^\epsilon \epsilon_N + A_w^\epsilon \epsilon_W + A_e^\epsilon \epsilon_E) + S_\epsilon = 0 \quad (5.30)$$

L'équation de transport pour C_{Z_n} :

$$A_p^{C_{Z_n}} C_{Z_n P} - (A_s^{C_{Z_n}} C_{Z_n S} + A_n^{C_{Z_n}} C_{Z_n N} + A_w^{C_{Z_n}} C_{Z_n W} + A_e^{C_{Z_n}} C_{Z_n E}) + S_{C_{Z_n}} = 0 \quad (5.31)$$

5.3 Conditions aux limites et termes sources du modèle de turbulence

5.3.1 Traitement des conditions aux limites pour les schémas d'ordre un

Nous nous bornons ici aux schémas d'ordre un. Le traitement des schémas d'ordre supérieur dépend de leur ordre et donc du nombre de points voisins impliqués dans la discrétisation des équations.

La condition de Dirichlet

La condition miroir (figure 5.3): Toutes les conditions de Dirichlet sont traitées de la même manière. Dans un premier temps, nous avons implémenté la condition classique de miroir qui permet d'ajouter un point adjacent à la frontière de telle sorte que le gradient soit correctement évalué.

Dans le cas de la figure 5.3, considérons une inconnue ϕ qui peut être k , ϵ etc... Spécifier la condition miroir pour cette variable consiste à calculer $\phi(i, 0)$ de la manière suivante:

$$\phi(i, 0) = 2\phi_p - \phi(i, 1) \quad (5.32)$$

Cette formule permet une spécification correcte de ϕ_p par interpolation linéaire. Elle permet d'utiliser directement $\Delta y(1)$ au lieu de Δy_s pour le calcul du résidu. Ce procédé fonctionne correctement tant que la frontière solide ne présente pas de coin. Dans le cas contraire, on a un problème comme on peut le voir sur la figure 5.4. En effet les deux points $P1$ et $P2$ ont la même image par leur paroi respective. Pour cette raison nous avons fait le choix d'utiliser directement les valeurs spécifiées à la paroi en modifiant le calcul des termes de diffusion.

La condition de Neumann

Pour spécifier les conditions de Neumann, revenons aux équations de transport de 5.27 à 5.31. Considérons une frontière située à l'est. La condition de Neumann pour un écoulement établi s'écrit:

$$\phi_p = \phi_e$$

Cette équation équivaut à écrire:

$$A_e = 0$$

Il faut noter que dans le cas où le gradient est constant mais pas nul, cette méthode donne un résultat faux.

5.3.2 Termes sources des équations du modèle de turbulence

Les termes source, par leur importance dans les équations de la turbulence, posent des problèmes lors de la résolution. La linéarisation upwind a comme avantage de fournir des systèmes linéaires stables. Dans le cas des équations avec termes source dominants, c'est la discrétisation de ces derniers qui assurent la stabilité. Ils nécessitent donc un traitement spécifique. Le choix que nous avons adopté est celui proposé par Chien [11] le concepteur du modèle que nous avons implanté.

Terme source pour l'équation de l'énergie turbulente:

$$S_k = -\Delta x_p \Delta y_p \left(-P_p + \frac{2\nu}{d_p^2} + \frac{\epsilon_p}{k_p} \right) \quad (5.33)$$

Terme source pour l'équation du taux de dissipation de l'énergie turbulente:

$$S_\epsilon = -\Delta x_p \Delta y_p \frac{\epsilon_p}{k_p} \left(-C_{\epsilon 1} P_p + \frac{2\nu k_p \exp\left(\frac{-C_{4u\tau} d_p}{\nu}\right)}{d_p^2} + C_{\epsilon 2} f_p \epsilon_p \right) \quad (5.34)$$

$$P_p = 2.0\nu_t(i, j) \left(\left(\frac{\Delta u}{\Delta x_P} \right)^2 + \left(\frac{\Delta v}{\Delta y_P} \right)^2 + \left(\frac{\Delta u}{\Delta y_P} + \frac{\Delta v}{\Delta x_P} \right)^2 \right) \quad (5.35)$$

Pour le calcul de $u_{\tau p}$, on distingue deux cas:

le modèle non-local la dérivé de u est évaluée à la paroi la plus proche:

$$u_{\tau p} = \sqrt{\nu \left| \frac{\Delta u}{\Delta y_p} \right|} \quad (5.36)$$

le modèle local la dérivée de u est évaluée au point courant (i, j) :

$$u_{\tau p} = \sqrt{(\nu + \nu_t) \left| \frac{\Delta u}{\Delta y_p} \right|} \quad (5.37)$$

Le calcul des dérivées pour le terme de production est détaillé en A.2. Comme on peut le voir le calcul de ces dérivées implique l'évaluation de nombreuses valeurs par interpolation. Si la résolution stable des équations de transport nécessite un traitement spécifique des termes convectifs, on peut se demander quelle conséquence peut avoir cette interpolation simplifiée des différentes dérivées sur la stabilité de la méthode. En effet les dérivées évaluées sont du même ordre que celle qui interviennent dans les termes convectifs.

5.4 Conclusion

Les principaux avantages du schéma hybride sont: sa facilité de mise en oeuvre et sa stabilité lors du processus de résolution. Ce schéma est cependant très critiqué pour des raisons de précision. Les tests que nous avons effectués sur la cavité montrent clairement la mauvaise tenue de ce schéma par rapport au schéma centré par exemple. Leonard et Drummond [38] ont écrit un article intitulé "Pourquoi vous ne devez plus utiliser les schémas dérivés du schéma exponentiel". En ce qui nous concerne, nous persistons pour deux raisons: la première est qu'au départ nous étudions des écoulements quasi monodimensionnel. Ce qui élimine les problèmes liés au non-alignement de l'écoulement avec le maillage. La deuxième raison est que le calcul d'écoulements turbulents avec des modèles à bas Reynolds oblige à resserrer le maillage, et donc à réduire la valeur du nombre de Peclet local. Toutefois, à plus long terme, nous considérons ce schéma comme un point de départ. Une fois que nous disposerons d'un code structurellement stable, l'adjonction d'autres schémas pourra être envisagée.

Partie III

Résolution des équations discrétisées.

Chapitre 6

Généralités

6.1 Introduction

La résolution des équations discrétisées est la partie la plus coûteuse du processus de simulation. Suivant le type de méthode utilisé, le coût peut être une fonction polynômiale ou exponentielle du nombre de degrés de liberté. Les méthodes de résolution évoluent avec les capacités des ordinateurs. Tout a commencé avec les méthodes de relaxation explicites telles que la méthode de Jacobi ou de Gauss-Seidel. Les méthodes de relaxation par ligne sont ensuite apparues avec l'algorithme de Thomas [34]. Elles sont à la base des algorithmes de type SIMPLER apparu dans les années 70. Des méthodes d'inversion des grands matrices creuses sont de plus en plus utilisés pour la résolution des équations de Navier-Stokes. Ces algorithmes ont notamment été développés pour les besoins de la simulation des circuits électroniques. C'est le cas des modules de la librairie de programmes YSMP [16]. Tous les algorithmes précédents, basés sur la manipulation directe des données, peuvent être améliorés en adoptant une organisation astucieuse de ces dernières comme on le verra un peu plus loin.

Différentes stratégies sont mises en oeuvre pour diminuer les coûts. Toutes ces stratégies passent par différentes formes de regroupement des équations selon des critères qui peuvent être de trois types: Le type de la variable par exemple les u_h entre eux et les v_h entre eux. C'est le cas des algorithmes du type SIMPLE. Un autre type de critère est la distance dans la numérotation des noeuds du maillage. Ce critère donne les méthodes par blocs. La méthode implicite par zone est basée sur ce principe. Un troisième mode de regroupement donne lieu aux méthodes multiniveaux. Cette approche est basée sur un critère hiérarchique.

Dans cette partie, nous présentons uniquement les bases théoriques de la méthode de Schwarz. Nous ferons ensuite une présentation sommaire des méthodes de résolutions classiques des problèmes linéaires, puis de celles adaptées aux problèmes non-linéaires.

6.2 Résolution multidomaines et la méthode alternée de Schwarz

On considère un domaine $\Omega \subset \mathfrak{R}^n$ de bord Γ , formé par la réunion de deux sous-domaines Ω_1 et Ω_2 de bords respectifs Γ_1 et Γ_2 . On considère un problème de Dirichlet:

$$\begin{aligned} L(\phi(X)) &= f(X) & X \in \Omega \\ \phi(X) &= \varphi(X) & X \in \Gamma \end{aligned} \quad (6.1)$$

avec φ une fonction à valeur sur Γ . Elle donne les conditions aux limites pour le problème de Dirichlet. La solution de système sera notée $\phi(\varphi)$.

On définit deux problèmes:

$$\begin{aligned} L(\phi_1(X)) &= f(X) & X \in \Omega_1 \\ \phi_1(X) &= \varphi_1(X) & X \in \Gamma_1 \end{aligned} \quad (6.2)$$

et

$$\begin{aligned} L(\phi_2(X)) &= f(X) & X \in \Omega_2 \\ \phi_2(X) &= \varphi_2(X) & X \in \Gamma_2 \end{aligned} \quad (6.3)$$

avec φ_i une fonction à valeur sur Γ_i . Elle donne les conditions aux limites pour le problème i . La solution de ce système sera notée $\phi_i(\varphi_i)$.

Le procédé alterné de Schwarz donne une méthode d'approximation du problème 6.1 à partir des problèmes 6.2 et 6.3. On note:

$$\Gamma_1' = \Gamma_1 \cap \Gamma \text{ et } \Gamma_2' = \Gamma_2 \cap \Gamma$$

de telle sorte que:

$$\Gamma = \Gamma_1' \cup \Gamma_2'$$

Définition 6.1 *Etant donné une fonction f sur Ω , on appelle solution classique de l'équation 6.1 toute fonction ϕ vérifiant*

$$L(\phi(X)) = f(X)$$

pour tout $X \in \Omega$ et

$$\phi(X) = \varphi(X)$$

pour

$$X \in \Gamma$$

6.2. RÉSOLUTION MULTIDOMAINES ET LA MÉTHODE ALTERNÉE DE SCHWARZ65

Etant donné $\varphi \in C^0(\Gamma)$, il existe une solution classique ϕ de 6.1 si et seulement si il existe des solutions classiques ϕ_1 et ϕ_2 aux problèmes 6.2 et 6.3, avec les conditions aux limites suivantes:

$$\begin{aligned} \varphi_i = \varphi \quad \text{sur} \quad \Gamma'_i \quad \text{pour} \quad i = 1, 2 \\ \phi_1 = \phi_2 \quad \text{sur} \quad \Omega_1 \cup \Omega_2 \end{aligned} \quad (6.4)$$

Une telle solution ϕ de 6.1 vérifie
 $\phi = \phi_i$ sur Ω_i , pour $i = 1, 2$.

Etant donné φ_1 on peut lui associer la fonction $T_2\varphi_1$ définie sur Γ_2 par

$$T_2\varphi_1(X) = \begin{pmatrix} \varphi(X) & \text{sur} \quad \Gamma_2' \\ \phi_1(\varphi_1, X) & \text{sur} \quad \Gamma_2 \cap \Omega_1 \end{pmatrix}$$

De même à φ_2 on associe $T_1\varphi_2$ définie sur Γ_1 par

$$T_1\varphi_2(X) = \begin{pmatrix} \varphi(X) & \text{sur} \quad \Gamma_1' \\ \phi_2(\varphi_2, X) & \text{sur} \quad \Gamma_1 \cap \Omega_2 \end{pmatrix}$$

Le rôle de l'opérateur T_1 est de construire les conditions aux limites pour 6.2 à partir de la solution de 6.3.

De la même manière, celui de l'opérateur T_2 est de construire les conditions aux limites pour 6.3 à partir de la solution de 6.2.

Le problème 6.1 admet une solution classique si

$$T_2\varphi_1 = \varphi_2 \quad \text{et} \quad T_1\varphi_2 = \varphi_1$$

En d'autres termes, l'application T définie par:

$$(\varphi_1, \varphi_2) \rightarrow T(\varphi_1, \varphi_2) = (T_1\varphi_2, T_2\varphi_1)$$

admet un point fixe.

D'après l'unicité de la solution du problème de Dirichlet 6.1, s'il existe, ce point fixe (φ_1, φ_2) sera unique et la solution ϕ du problème général sera définie par:

$$\phi = \phi_i(\varphi_i) \quad \text{sur} \quad \Omega_i \quad \text{pour} \quad i = 1, 2$$

On remarque que (φ_1, φ_2) est un point fixe de T ssi :

$$\varphi_2 = T_2\varphi_1 \quad \text{et} \quad \varphi_1 \text{ est point fixe de } S_1 = T_1T_2$$

et

$$\varphi_1 = T_1\varphi_2 \quad \text{et} \quad \varphi_2 \text{ est point fixe de } S_2 = T_2T_1$$

Il existe plusieurs méthodes pour obtenir un point fixe. L'une d'elle consiste à procéder par approximations successives: partant de φ_1^0 , et de φ_2^0 on construit par récurrence

$$\varphi_1^k = T_1 T_2 \varphi_1^{k-1} \text{ et } \varphi_2^k = T_2 T_1 \varphi_2^{k-1}$$

ou plus simplement, si φ_2^0 est donnée, on construit les suites:

$$\varphi_1^k = T_1 \varphi_2^k \text{ et } \varphi_2^k = T_2 \varphi_1^{k-1}$$

On peut également partir de φ_1^0 , dans ce cas on construit les suites:

$$\varphi_2^k = T_2 \varphi_1^k \text{ et } \varphi_1^k = T_1 \varphi_2^{k-1}.$$

Cet algorithme a donné son nom au "procédé alterné de Schwarz": Partant des données sur la frontière d'un des domaines, on construit par résolution sur ce domaine du problème correspondant, les données aux frontières pour l'autre domaine. L'estimation de l'erreur et les conditions de convergence sont détaillées dans [20].

La méthode alternée de Schwarz ainsi présentée permet de traiter un domaine constitué de deux sous-domaines.

Soit à résoudre le problème 6.1 avec $\Omega = \cup_{i=1}^K \Omega_i$

Alors on peut définir une procédure récursive de la manière suivante:

On pose $\Omega_1^1 = \Omega_1$ et $\Omega_2^1 = \overline{\Omega_1^1}$

On obtient deux sous-problèmes. La résolution du premier problème permet de déterminer des conditions aux limites provisoires pour le second problème. Pour la résolution de ce dernier, on décompose Ω_2^1 comme on l'a fait pour Ω .

La mise en oeuvre de cette technique pour plusieurs sous-domaines nécessite une bonne synchronisation de la communication entre les Ω_i , de telle sorte que leur parcours ne contrarie pas la convergence du processus global. Cette synchronisation est encore plus délicate lorsque des zones de recouvrement existent entre les sous-domaines. Les détails de la solution que nous avons adoptée seront donnés dans la partie concernant l'architecture.

6.3 Algorithmes de base

Par algorithmes de base, nous entendons les algorithmes qui manipulent directement les équations par opposition aux stratégies globales comme celle que nous venons de décrire. Nous séparons ces méthodes en deux catégories: Celle destinées aux problèmes linéaires et celle destinées aux problèmes non-linéaires. Donnons quelques définitions: nous noterons L_h la version discrétisée de l'opérateur L de l'équation 6.1. ϕ_h est l'ensemble des degrés de liberté à déterminer pour l'inconnue ϕ . n est l'indice d'itération, et par conséquent ϕ_h^n est l'estimation de la solution à la n^{ieme} itération. f_h est la version discrétisée du second membre. L'équation 6.1 s'écrit, sans les conditions aux limites, sous sa forme discrétisée:

$$L_h(\phi_h) = f_h$$

(6.5)

6.3.1 Problème linéaire

Pour un opérateur L linéaire, le premier membre de l'équation 6.5 s'écrit $L_h(\phi_h) = L_h \cdot \phi_h$. L_h est une matrice et ϕ_h est un vecteur de même que f_h . La solution peut s'obtenir en une itération par un algorithme d'inversion directe tel que l'élimination de Gauss ou une des nombreuses procédures de factorisation [18]. L'autre solution pour les grands systèmes consiste à utiliser une méthode itérative de type Gauss-Seidel.

Même les algorithmes d'inversion directe peuvent être considérés comme des méthodes itératives à une itération. Le principe de l'inversion itérative est de partir d'une estimation initiale, puis de construire une nouvelle estimation à chaque itération. Le processus comporte de fait deux étapes. Pour la première étape, on calcule une correction $\delta\phi_h$ en résolvant le problème approché suivant:

$$K_h \delta\phi_h = (f_h - L_h \cdot \phi_h^n) = -Res_h(\phi_h) \quad (6.6)$$

Dans la deuxième étape, on calcule l'estimation $n + 1$ par:

$$\phi_h^{n+1} = \phi_h^n + \alpha_n \delta\phi_h \quad (6.7)$$

On décrit ainsi un processus itératif pour lequel l'estimation de la solution à l'itération $n + 1$ est donnée par 6.7. K_h est une approximation de L_h choisie de telle sorte que son inversion soit facile. α_n est un paramètre d'optimisation du processus. L'optimisation est souvent indispensable pour certains problèmes, et nous allons maintenant voir pourquoi.

Si on note ϕ_h la solution exacte du système algébrique, l'équation donnant l'évolution de l'erreur $e_h^n = \phi_h^n - \phi_h$ est:

$$e_h^{n+1} = M_h \cdot e_h^n \quad (6.8)$$

soit encore:

$$e_h^{n+1} = (M_h)^n \cdot e_h^0 \quad (6.9)$$

avec $M_h = I - \alpha_n \cdot B_h \cdot L_h$ et $B_h = K_h^{-1}$

Cette dernière équation montre bien le fait que la vitesse de convergence du processus itératif dépend du spectre de l'opérateur M_h . Le cas le plus simple est de prendre $B_h = L_h^{-1}$. Il correspond à l'inversion globale. Dans ce cas, $\alpha_n = 1$ et l'opérateur M_h "d'amplification" de l'erreur est nul et toutes ses valeurs propres également. La solution est obtenue en une itération. Soit λ_h et g_h l'ensemble des valeurs propres et vecteurs propres de M_h . Soit $e^0 = \sum_h \beta_h g_h$ une expression de l'erreur initiale sur la base des vecteurs propres de M_h . L'équation 6.9 s'écrit alors:

$$e_h^{n+1} = \sum_h \beta_h (\lambda_h)^n g_h \quad (6.10)$$

Pour que le processus itératif converge, il faut que $\|\lambda_h\| < 1$. De plus les composantes de l'erreur pour lesquelles la valeur propre sera la plus grande seront éliminées moins vite que les autres. Pour certains choix de K_h , le processus peut stagner alors que l'erreur reste importante. Dans ce cas des algorithmes d'accélération existent. Les deux principaux sont: Les méthodes de projection dans les sous-espaces de Krilov, et les méthodes multigrilles qui feront l'objet du prochain chapitre.

Pour décrire sommairement les méthodes de projection, on va réécrire l'équation 6.9 de la manière suivante:

$$e_h^{n+1} = (I - \alpha_n \cdot B_h \cdot L_h)^n \cdot e_h^0 \quad (6.11)$$

$(I - \alpha_n \cdot B_h \cdot L_h)^n$ est un polynôme matriciel en $A_h = B_h \cdot L_h$ d'où:

$$e_h^{n+1} = \sum_{i=0}^{i=n} a_i \cdot A_h^i \cdot e_h^0 \quad (6.12)$$

Les sous-espaces de Krilov sont engendrés par les vecteurs $(A_h^i \cdot e_h^0)_{i \leq n}$. Le principe des méthodes de type projection est de construire des suites $A_h^i \cdot e_h^0$ telles que la suite $\|A_h^i \cdot e_h^0\|$ soit à décroissance rapide en jouant sur le choix de α_n . Ce paramètre est calculé de manière à diminuer le plus possible le résidu à chaque itération. Les méthodes moins optimisées utilisent un coefficient de relaxation α_n constant à chaque itération. Les plus connues sont la méthode de Jacobi et la méthode de Gauss-Seidel.

6.3.2 Problèmes non-linéaires et algorithme de Newton-Raphson.

L'inversion directe d'opérateurs non-linéaires n'étant pas possible, les algorithmes de résolution de ces problèmes passent par un processus préalable de linéarisation. Les deux méthodes classiques de résolution des problèmes non-linéaires sont la méthode du point fixe et la méthode Newton. La méthode du point fixe est définie par le processus itératif suivant:

$$\phi_h^{n+1} = \bar{L}_h^{-1}(\phi_h^n) \cdot f_h \quad (6.13)$$

Cette méthode est utilisable pour les équations discrétisées qui peuvent être mise sous la forme:

$$L_h(\phi_h) = \bar{L}_h(\phi_h) \cdot \phi_h \quad (6.14)$$

La méthode de Newton est définie par:

$$L_h(\phi_h^n + \delta\phi_h) - L_h(\phi_h^n) = f_h - L_h(\phi_h^n) \quad (6.15)$$

Pour cette dernière méthode, une nouvelle estimation est donnée à chaque itération par:

$$\phi_h^{n+1} = \phi_h^n + J_h(\phi_h^n)^{-1} \cdot (f_h - L_h(\phi_h^n)) \quad (6.16)$$

où $J_h(\phi_h^n)$ est le jacobien de l'opérateur L_h . $J_h(\phi_h^n)$ est donné par:

$$L_h(\phi_h^n + \delta\phi_h) = L_h(\phi_h^n) + J_h(\phi_h^n).\delta\phi_h + O(\|\delta\phi_h\|^2) \quad (6.17)$$

$J_h(\phi_h^n)$ est obtenu par un développement de Taylor de L_h au voisinage de ϕ_h^n . L'équation pour l'obtention de la correction devient:

$$J_h(\phi_h^n).\delta\phi_h = -Res_h(\phi_h^n) \quad (6.18)$$

où: $Res_h(\phi_h^n) = L_h(\phi_h^n) - f_h$ représente le résidu de l'équation non-linéaire. Les corrections calculées n'étant que des approximations, elles peuvent être soit sous-estimées, auquel cas la convergence sera lente, ou sur-estimées avec un risque de divergence. Un coefficient de relaxation permet d'adapter cette correction au processus de convergence. Cette relaxation est différente de celle qu'on a évoquée pour la résolution du problème linéaire.

Ces deux méthodes décomposent un problème non linéaire sans solution directe en une suite de problèmes approchés linéaires.

Si la méthode du point fixe est robuste, elle converge néanmoins lentement. La convergence de la méthode de Newton n'est assurée que localement. Ce qui veut dire que, le point de départ ϕ_h^0 ne doit pas être trop éloigné de la solution.

Différentes variantes de la méthode de Newton se distinguent, par la manière d'approximer $J_h(\phi_h^n)^{-1}$. En effet le calcul et la factorisation du jacobien constituent la partie la plus coûteuse de l'algorithme de résolution. On sait cependant qu'un calcul exact n'est pas nécessaire pour obtenir la convergence. D'un autre côté une approximation trop grossière peut ralentir sinon contrarier le processus. On retrouve le même problème que pour le cas des systèmes linéaires.

Certaines variantes procèdent comme précédemment en optimisant un coefficient de relaxation. D'autres méthodes limitent les coûts en ne recalculant le jacobien que périodiquement, au lieu de le faire à chaque itération. Une procédure intéressante suivant le même principe permet de calculer le Jacobien et le résidu avec des schémas différents. En effet la linéarisation des schémas de discrétisation haute précision peut donner des systèmes mal conditionné. Cette procédure nous l'avons utilisé dans le cadre d'un stage de DEA pour tester le schéma QUICK et un schéma décentré d'ordre trois.

Si on construit l'équation 6.18 en remplaçant $J_h(\phi_h^n)$ par $\bar{L}_h(\phi)$, on obtient une méthode pseudo-Newton. Dans notre étude nous avons utilisé cette méthode pour les équations du modèle de turbulence (à cause des termes sources), et la méthode de Newton pour toutes les autres équations.

6.4 Conclusion

Il existe une multitude d'algorithmes entre l'inversion globale par élimination de Gauss et les inversions itératives explicites de type Gauss-Seidel. Chacun de ces algorithmes présente des avantages et des inconvénients suivant le type de problème

à résoudre. L'efficacité des méthodes peut être améliorée en tenant compte des spécificités du problème. c'est ce que nous allons essayer de faire en appliquant la méthode de Schwarz à la résolution d'écoulements de fluides isovolumes.

Chapitre 7

Résolution des équations de Navier-Stokes

7.1 Introduction

Les premières tentatives de résolution des équations de Navier-Stokes se sont heurtées au problème de place mémoire et de temps de calcul pour l'inversion de systèmes résultant de la linéarisation des équations. Comme toujours dans pareil cas, il a fallu trouver des astuces pour limiter les coûts.

La majorité des méthodes d'approximations procèdent par découplage des variables u , v et p . La séparation des variables produit des systèmes qui ont des dynamiques complètement différentes. Ces différences peuvent créer des problèmes de coordination délicats pour la convergence du processus global. L'approche la plus répandue pour effectuer ce découplage a donné naissance aux algorithmes de la famille SIMPLE dont nous allons présenter la variante SIMPLER.

Ces algorithmes se sont cependant révélés inefficaces pour des maillages denses, ainsi que pour des nombres de Reynolds importants. Pour palier à ces difficultés, les numériciens ont profité de l'augmentation de la puissance des nouveaux ordinateurs pour tester d'autres types d'approches. L'une des idées consiste à revenir sur le découplage des variables. Les méthodes qui en résultent sont de type par zones, comme celle que nous allons présenter dans la section 7.4.

7.2 Linéarisation des équations

La majorité des équations que nous aurons à résoudre sont non-linéaires. Pour les résoudre, nous allons appliquer la méthode de Newton. Pour cela, nous devons linéariser ces équations. Nous avons utilisé dans notre travail deux méthodes de linéarisation: La première permet d'obtenir les coefficients directement à partir du processus de discrétisation des équations. L'autre méthode permet une estimation numérique de la matrice jacobienne.

7.2.1 Dédution des coefficients des équations discrétisées

Pour cette méthode, nous allons simplement reprendre les équations 5.26 à 5.31. Avant d'écrire les équations du système linéaire, nous allons remplacer les lettres p , s , n etc... par les indices (i, j) . Les équations 5.27 à 5.31, peuvent se mettre sous la forme générale:

$$f_{i,j}^\phi(u, v, p, k, \epsilon) = 0 \quad (7.1)$$

avec:

$$f_{i,j}^\phi(u, v, p, k, \epsilon) = A_{p,i,j}^\phi \phi(i, j) - (A_{s,i,j}^\phi \phi(i, j - 1) + A_{s,i,j}^\phi u(i, j + 1) + A_{w,i,j}^\phi \phi(i - 1, j) + A_{e,i,j}^\phi \phi(i + 1, j)) + S_\phi(i, j) \quad (7.2)$$

On peut alors définir $a_{w,i,j}^{\phi_1 \phi_2}$ comme le coefficient de linéarisation de l'équation pour $\phi_1(i, j)$ par rapport à $\phi_2(i - 1, j)$ et $Res_{i,j}^{\phi_1}$ comme le résidu de l'équation pour $\phi_1(i, j)$. Appliqué par exemple à l'équation 5.27, on obtient l'équation de correction:

$$a_{P,i,j}^{uu} \delta u(i, j) + a_{S,i,j}^{uu} \delta u(i, j - 1) + a_{N,i,j}^{uu} u(i, j + 1) + a_{W,i,j}^{uu} \delta u(i - 1, j) + a_{E,i,j}^{uu} \delta u(i + 1, j) + a_{P,i,j}^{up} p(i, j) + a_{W,i,j}^{up} p(i - 1, j) = -Res_{i,j}^u \quad (7.3)$$

Les coefficients sont donnés par:

$$\begin{aligned} a_{P,i,j}^{uu} &= A_{P,i,j}^u \\ a_{S,i,j}^{uu} &= -A_{S,i,j}^u \\ a_{N,i,j}^{uu} &= -A_{N,i,j}^u \\ a_{W,i,j}^{uu} &= -A_{W,i,j}^u \\ a_{E,i,j}^{uu} &= -A_{E,i,j}^u \\ a_{P,i,j}^{up} &= \Delta y v(j) \\ a_{W,i,j}^{up} &= -\Delta y v(j) \end{aligned} \quad (7.4)$$

Avec cette manière de construire le système linéaire, on obtient une méthode pseudo-Newton. Les autres équations sont linéarisées en suivant le même schéma.

7.2.2 Estimation numérique de la matrice jacobienne

Dans cette partie, nous considérons le cas général d'un système de N_{eq} équations implicites non-linéaires à résoudre. Nous allons appeler ϕ_m les inconnues de ce système. L'équation pour la détermination de l'inconnue discrète $\phi_m(i, j)$ peut s'écrire:

$$f_{i,j}^{\phi_m}(\phi_1, \dots, \phi_{N_{eq}}) = 0 \quad (7.5)$$

On peut alors écrire l'équation de correction sous la forme générale:

$$\begin{aligned} \sum_{n=1}^{n=N_{eq}} [a_{P,i,j}^{\phi_m \phi_n} \delta \phi_m(i, j) + a_{S,i,j}^{\phi_m \phi_n} \delta \phi_m(i, j - 1) + a_{N,i,j}^{\phi_m \phi_n} \delta \phi_m(i, j + 1) + \\ a_{W,i,j}^{\phi_m \phi_n} \delta \phi_m(i - 1, j) + a_{E,i,j}^{\phi_m \phi_n} \delta \phi_m(i + 1, j)] \\ = -f_{i,j}^{\phi_m}(\phi_1, \dots, \phi_{N_{eq}}) \end{aligned} \quad (7.6)$$

Les coefficients sont donnés par:

$$\begin{aligned} a_{P,i,j}^{\phi_m \phi_n} &= \frac{f_{i,j}^{\phi_m}(\phi_1, \dots, \phi_n(i, j) + \Delta \phi, \dots, \phi_{N_{eq}}) - f_{i,j}^{\phi_m}(\phi_1, \dots, \phi_{N_{eq}})}{\Delta \phi} \\ a_{S,i,j}^{\phi_m \phi_n} &= \frac{f_{i,j}^{\phi_m}(\phi_1, \dots, \phi_n(i, j - 1) + \Delta \phi, \dots, \phi_{N_{eq}}) - f_{i,j}^{\phi_m}(\phi_1, \dots, \phi_{N_{eq}})}{\Delta \phi} \\ a_{N,i,j}^{\phi_m \phi_n} &= \frac{f_{i,j}^{\phi_m}(\phi_1, \dots, \phi_n(i, j + 1) + \Delta \phi, \dots, \phi_{N_{eq}}) - f_{i,j}^{\phi_m}(\phi_1, \dots, \phi_{N_{eq}})}{\Delta \phi} \\ a_{W,i,j}^{\phi_m \phi_n} &= \frac{f_{i,j}^{\phi_m}(\phi_1, \dots, \phi_n(i - 1, j) + \Delta \phi, \dots, \phi_{N_{eq}}) - f_{i,j}^{\phi_m}(\phi_1, \dots, \phi_{N_{eq}})}{\Delta \phi} \\ a_{E,i,j}^{\phi_m \phi_n} &= \frac{f_{i,j}^{\phi_m}(\phi_1, \dots, \phi_n(i + 1, j) + \Delta \phi, \dots, \phi_{N_{eq}}) - f_{i,j}^{\phi_m}(\phi_1, \dots, \phi_{N_{eq}})}{\Delta \phi} \end{aligned} \quad (7.7)$$

$\Delta \phi$ est un paramètre fixe dont la valeur dépend de la précision de la machine. Plus elle sera faible plus les valeurs calculées seront proches des dérivées partielles dont elles sont une estimation. Cette procédure est nettement plus coûteuse que la précédente, dans la mesure où elle nécessite une réestimation du résidu pour chaque coefficient. Elle permet cependant de prendre en compte plus de possibilités de couplage entre équations.

Ne pas calculer de coefficient pour ϕ_2 lors de la linéarisation de l'équation pour ϕ_1 revient à stipuler qu'une variation de ϕ_2 n'a pas d'influence sur la variation du résidu de l'équation pour ϕ_1 . Si cette approximation peut simplifier le système linéaire à résoudre, elle peut aussi pénaliser la stabilité et la vitesse de convergence.

L'estimation numérique de la matrice jacobienne ne distingue pas les variables ou les termes des équations. La nécessité d'un traitement spécifique des termes sources dans les équations du modèle turbulent fait que cette méthode ne donne pas de bons résultats. Elle a donc été utilisée pour toutes les équations sauf celles de k et ϵ .

Cette méthode a en grande partie inspiré la solution que nous avons adopté pour limiter les problèmes de gestion de place mémoire pour les tableaux de coefficients $a_{Pos,i,j}^{\phi_1\phi_2}$, notamment lors d'un changement du nombre d'équations du système comme nous le verrons dans la partie traitant de la présentation du code.

7.3 La méthode SIMPLER

L'algorithme SIMPLE dont dérive le SIMPLER est apparu à un moment où les ressources informatiques étaient limitées. Ces algorithmes en ont gardé un faible niveau de demande en ressources mémoire, sans commune mesure avec les algorithmes plus récents. La contrepartie est leur faible vitesse de convergence. L'apparition des méthodes d'accélération multigrille a renouvelé l'intérêt déjà important suscité par ces algorithmes depuis leur apparition.

Dans la présente étude, la méthode SIMPLER a été implantée en langage *C* et sert de référence pour le test des méthodes par zones (sauf lorsque sa convergence devient trop lente). Elle est dérivée de la méthode SIMPLE (Semi Implicit Method for Pressure Linked Equation). Cette méthode consiste à réécrire les trois équations discrétisées de Navier Stokes 5.26, 5.27 et 5.28 de manière à obtenir pour chacune des variables une équation explicite permettant de la déterminer. Cet algorithme a été transformé pour calculer une correction selon la méthode de Newton (SIMPLER). La version d'origine détermine directement les nouvelles estimations des inconnues.

7.3.1 Equations pour la vitesse

Les équations pour u et v sont immédiates :

$$\begin{aligned} A_p^u \cdot \delta u_p - \sum_{q \in \text{Vois}} A_q^u \cdot \delta u_q - b^u &= -Res^u(i, j) \\ A_p^v \cdot \delta v_p - \sum_{q \in \text{Vois}} A_q^v \cdot \delta v_q - b^v &= -Res^v(i, j) \end{aligned} \quad (7.8)$$

où les coefficients A^u et A^v sont donnés en annexe A.1. Les termes de pression sont gelés pour former les termes sources des équations de convection:

$$b^u = \Delta y(j) \cdot (\delta p(i-1, j) - \delta p(i, j)) \quad (7.9)$$

$$b^v = \Delta x(i) \cdot (\delta p(i, j-1) - \delta p(i, j)) \quad (7.10)$$

7.3.2 Equation pour la pression

Pour obtenir l'équation pour la pression, la seule solution consiste à passer par les seules variables communes à l'équation de continuité et aux équations de quantité de mouvement. Pour cela, on effectue une inversion approchée.

$$\begin{aligned}\delta u(i, j) &= \delta \tilde{u}(i, j) + B^u(i, j) \cdot (\delta p(i-1, j) - \delta p(i, j)) \\ \delta v(i, j) &= \delta \tilde{v}(i, j) + B^v(i, j) \cdot (\delta p(i, j-1) - \delta p(i, j))\end{aligned}\quad (7.11)$$

Ces équations expriment le fait que la correction $\delta u(i, j), \delta v(i, j)$ est égale à la correction approchées $\delta \tilde{u}(i, j), \delta \tilde{v}(i, j)$:

$$\begin{aligned}\delta \tilde{u}(i, j) &= \left(\sum_{q \in \text{Vois}} A_q^u \cdot \delta u_q - \text{Res}^u(i, j) \right) \cdot \frac{1}{A_p^u} \\ \delta \tilde{v}(i, j) &= \left(\sum_{q \in \text{Vois}} A_q^v \cdot \delta v_q - \text{Res}^v(i, j) \right) \cdot \frac{1}{A_p^v}\end{aligned}\quad (7.12)$$

où les termes de pression ne sont pas pris en compte, plus une correction calculée après détermination du champ de pression.

avec:

$$\begin{aligned}B^u(i, j) &= \frac{\Delta y(j)}{A_p^u} \\ B^v(i, j) &= \frac{\Delta x(i)}{A_p^v}\end{aligned}\quad (7.13)$$

La substitution de 7.11 dans l'équation de continuité:

$$\Delta y(j) \cdot (\delta u(i+1, j) - \delta u(i, j)) + \Delta x(i) \cdot (\delta v(i, j+1) - \delta v(i, j)) = -\text{Res}^p(i, j) \quad (7.14)$$

conduit à une équation pour la pression :

$$A_p^p \cdot \delta p(i, j) - \sum_{q \in \text{Vois}} A_q^p \cdot \delta p_q - b^p = 0 \quad (7.15)$$

avec

$$\begin{aligned}
A_w^p &= B^u(i, j) \cdot \Delta y(j) \\
A_e^p &= B^u(i+1, j) \cdot \Delta y(j) \\
A_s^p &= B^v(i, j) \cdot \Delta x(i) \\
A_n^p &= B^v(i, j+1) \cdot \Delta x(i) \\
A_p^p &= \sum_{q \in \text{Vois}} A_q^p
\end{aligned} \tag{7.16}$$

et

$$b^p = \Delta y(j) \cdot (\delta \tilde{u}(i+1, j) - \delta \tilde{u}(i, j)) + \Delta x(i) \cdot (\delta \tilde{v}(i, j+1) - \delta \tilde{v}(i, j)) - \text{Res}^p(i, j) \tag{7.17}$$

7.3.3 Correction de vitesse et conservation de la masse

La résolution de l'équation 7.15 fournit le champ de pression. A ce stade, un champ de vitesse $\delta u'(i, j)$, $\delta v'(i, j)$ peut être calculé en résolvant les équations 7.8 complètes, à partir du champ de pression qui vient d'être obtenu et des A calculés précédemment.

Ce champ de vitesse vérifie l'équation de quantité de mouvement, mais pas l'équation de continuité. Il est donc nécessaire de le corriger. Dans la méthode SIMPLER, cette correction est effectuée de la manière suivante: On résout:

$$A_p^p \cdot dp(i, j) - \sum_{q \in \text{Vois}} A_q^p \cdot dp_q - b_c^p = 0 \tag{7.18}$$

Pour obtenir $dp(i, j)$.

Les coefficients A^p sont ceux de l'équation 7.15, b_c^p est donné par:

$$b_c^p = \Delta y(j) \cdot (\delta u'(i+1, j) - \delta u'(i, j)) + \Delta x(i) \cdot (\delta v'(i, j+1) - \delta v'(i, j)) + \text{Res}^p(i, j) \tag{7.19}$$

Les formules de correction des vitesses sont les suivantes:

$$\begin{aligned}
\delta u(i, j) &= \delta u'(i, j) + \frac{\Delta y}{A_p^u} \cdot (dp(i-1, j) - dp(i, j)) \\
\delta v(i, j) &= \delta v'(i, j) + \frac{\Delta x}{A_p^v} \cdot (dp(i, j-1) - dp(i, j))
\end{aligned} \tag{7.20}$$

On peut alors procéder à une nouvelle itération.

7.4 Méthode de résolution par zones implicite.

La résolution par zones est une application de la méthode alternée de Schwarz. Cette méthode est constitué de deux étapes:

1. La résolution implicite des équations de l'hydrodynamique. Elle consiste à résoudre pour la correction des inconnues un seul système d'équation linéaire. Contrairement à la méthode SIMPLER qui détermine un système d'équations par inconnues, cet algorithme respecte le couplage initial des variables. Afin de renforcer ce couplage, le système linéaire est en général résolu par une méthode d'inversion directe.
2. Cette inversion directe est coûteuse en place mémoire et en temps de calcul. Le découpage en zones (figure 7.1) du domaine de calcul permet de réduire ce coût qui est fonction de la taille des zones.

La résolution implicite suppose que ces trois équations soient résolues simultanément. Il existe des code généraux pour l'inversion de matrices creuses. Nous avons testé un de ces codes dans le but de l'incorporer au nôtre. Le premier constat que nous avons fait est que ce code est en lui même assez complexe. Pour réaliser notre travail, il aurait fallu maîtriser sa structure de donnée et son architecture. Tous ces faits nous ont amené à concevoir un algorithme d'inversion plus simple pour une intégration plus facile au reste du code. Plus simple et surtout plus facile à optimiser, cette partie consommant la plus grande partie du temps de calcul.

7.4.1 Optimisation de l'inversion pour maillage régulier

Au départ de cette partie, on a une zone délimitée par $(i, j) \in Z = [i_0, i_1] \times [j_0, j_1]$ et une inconnue à déterminer ϕ . Pour cette zone, on a calculé pour tout (i, j) : Les résidus $Res^\phi(i, j)$. Les coefficients $a_{P,i,j}^{\phi\phi}$, $a_{S,i,j}^{\phi\phi}$, $a_{N,i,j}^{\phi\phi}$, $a_{W,i,j}^{\phi\phi}$, $a_{E,i,j}^{\phi\phi}$.

Le but est de déterminer tous les $\delta\phi(i, j)$.

Méthode de base

La méthode la plus évidente est une inversion de Gauss pour la résolution d'un système linéaire de la forme:

$$A_Z \cdot \phi_Z = D_Z \quad (7.21)$$

Pour ce faire, il faut définir une application de numérotation par:

$$\begin{aligned} Num : \quad Z &\longrightarrow [1, N_Z] \\ (i, j) &\longmapsto Num(i, j) = k \end{aligned} \quad (7.22)$$

avec $N_Z = \text{cardinal}(Z)$.

Cette application permet de remplir la matrice A_Z et le second membre D_Z en attribuant une ligne à chaque inconnue $\delta\phi(i, j)$. Et inversement elle permet de faire correspondre chaque $\delta\phi_Z(k)$ à un $\delta\phi(i, j)$ pour $k \in [1, N_Z]$. Cette application ne permet pas la prise en compte du fait que la matrice est creuse. On va voir maintenant comment la modifier pour gagner en place mémoire et en temps de calcul.

Organisation bloc tridiagonale

On va redéfinir Num de la manière suivante:

$$\begin{aligned} Num : \quad Z &\longrightarrow [1, N_Z] \\ (i, j) &\longmapsto Num(i, j) = Bloc(i) + j - j_0 \end{aligned} \quad (7.23)$$

avec

$$\begin{aligned} Bloc : \quad [i_0, i_1] &\longrightarrow \{N_J\}[1, N_I] \\ i &\longmapsto Bloc(i) = N_J(i - i_0) \end{aligned} \quad (7.24)$$

On peut alors définir le $i^{\text{ième}}$ bloc X_i par:

$$X_i = \{(k, l) \in Z, k - i_0 = i\}$$

En observant les équations, le premier constat que l'on peut faire est que X_i n'est en relation qu'avec X_{i+1} et X_{i-1} . Ce qui veut dire que l'on peut écrire des blocs équations de la forme:

$$A_i \cdot X_{i-1} + B_i \cdot X_i + C_i \cdot X_{i+1} = D_i \quad (7.25)$$

Le système 7.21 devient:

$$A_Z = \begin{pmatrix} B_1 & C_1 & 0 & \dots & \dots & \dots & 0 \\ A_2 & \ddots & \ddots & \ddots & & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & A_i & B_i & C_i & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \ddots & \ddots & C_{N_I-1} \\ 0 & \dots & \dots & \dots & 0 & A_{N_I} & B_{N_I} \end{pmatrix}$$

Le vecteur colonne D contient l'ensemble des résidus des équations.

$$D_Z = \begin{pmatrix} D_{i_0} \\ \vdots \\ D_{i_1} \end{pmatrix}$$

En regroupant les points par ligne, on a réussi à obtenir une matrice bloc-tridiagonale. Cette structure permet comme on le verra plus loin de concevoir des algorithmes d'inversion peu coûteux.

Optimisation pour un domaine comportant des obstacles

Lorsque le domaine de calcul comporte des obstacles, l'une des solutions consiste à affecter à tous les points appartenant aux obstacles une équation de correction de type:

$$\delta\phi(i, j) = 0$$

et d'intégrer ces équations dans le reste du système (figure 7.2a). Mais cette solution est très coûteuse pour les cas où la taille des obstacles est comparable à celle du reste du domaine. Pour limiter ces coût, nous avons à nouveau modifié l'application *Num*:

$$\begin{aligned} \text{Num} : \quad Z &\longrightarrow [1, \overline{N}_Z] \\ (i, j) &\longmapsto \text{Num}(i, j) = \text{Bloc}(i - 1) + \text{Num}_i(j); \end{aligned} \quad (7.26)$$

avec

$$\begin{aligned} \text{Num}_i(j) &= 0 \quad (i, j) \in \text{OBS} \\ &= k \quad \text{sinon} \end{aligned} \quad (7.27)$$

où k est le nombre de point actifs de l'intervalle $\{i\} \times [j_0, j]$.

$$\text{Bloc}(i) = \text{Bloc}(i - 1) + k_i \quad (7.28)$$

où k_i est le nombre de point actifs du bloc X_i .

Calculons les tailles $n \times m$ des blocs matrices:

Pour A_i :

$$\begin{aligned} n &= \text{Bloc}(i) - \text{Bloc}(i - 1) \\ m &= \text{Bloc}(i - 1) - \text{Bloc}(i - 2) \end{aligned} \quad (7.29)$$

Pour B_i :

$$\begin{aligned} n &= \text{Bloc}(i) - \text{Bloc}(i - 1) \\ m &= n \end{aligned} \quad (7.30)$$

Pour C_i :

$$\begin{aligned} n &= \text{Bloc}(i) - \text{Bloc}(i - 1) \\ m &= \text{Bloc}(i + 1) - \text{Bloc}(i) \end{aligned} \tag{7.31}$$

Cette dernière amélioration permet de traiter à moindre coût des configurations telles que celles de la figure 7.2a . La matrice correspondante (figure 7.2b) occupe moins de place qu'avec la première numérotation.

Pour un nombre d'inconnues plus grand que un

Tout le travail précédent a consisté à établir une numérotation permettant de regrouper les points du domaine par blocs. A présent nous allons considérer le cas d'un problème où chaque point comporte un nombre Neq équation. C'est à dire pour nous u , v , et p . Tant que ce nombre d'inconnues est le même pour tous les points, les modifications de la numérotation pour l'étendre à chaque inconnue sont simples.

Supposons que l'on veuille retrouver la composante $\delta v(i, j)$ dans le vecteur ϕ_Z . Si on regroupe les inconnues dans l'ordre logique, $\delta v(i, j)$ occupera la position $Neq.Num(i, j)+2$ dans le vecteur ϕ_Z . En général, la position de l'inconnue numérotée ne sera $Neq.Num(i, j) + ne$.

7.4.2 Les frontières internes et le problème de la pression

On sait que la résolution implicite des équations de Navier-Stokes pour un fluide incompressible peut aboutir à l'inversion de systèmes singuliers (voir [7], [65]. Différentes solutions ont été retenues pour palier cette difficulté. Il s'agit notamment de la méthode de pénalité [40], on peut également résoudre le problème en fixant un point de pression dans le domaine de calcul [45] [7], [65]. L'apparition de ce problème dans la procédure d'inversion par zones nous a poussé à l'examiner de plus près.

Le but de cette partie est d'utiliser l'équation de conservation globale de la masse pour voir dans quels cas le système résultant de la linéarisation des équations hydrodynamique est singulier. Pour étudier le problème, considérons la configuration de la figure 7.3. Les équations de continuité de ce système sont données par:

$$\begin{aligned} q(1, 1) &= dy(1)(u(2, 1) - u(1, 1)) + dx(1)(v(1, 2) - v(1, 1)) = 0 \\ q(2, 1) &= dy(1)(u(3, 1) - u(2, 1)) + dx(2)(v(2, 2) - v(2, 1)) = 0 \\ q(1, 2) &= dy(2)(u(2, 2) - u(1, 2)) + dx(1)(v(1, 3) - v(1, 2)) = 0 \\ q(2, 2) &= dy(2)(u(3, 2) - u(2, 2)) + dx(2)(v(2, 3) - v(2, 2)) = 0 \end{aligned} \tag{7.32}$$

avec les équations de correction:

$$\begin{aligned}
dy(1)(\delta u(2,1) - \delta u(1,1)) + dx(1)(\delta v(1,2) - \delta v(1,1)) &= -q(1,1) \\
dy(1)(\delta u(3,1) - \delta u(2,1)) + dx(2)(\delta v(2,2) - \delta v(2,1)) &= -q(2,1) \\
dy(2)(\delta u(2,2) - \delta u(1,2)) + dx(1)(\delta v(1,3) - \delta v(1,2)) &= -q(1,2) \\
dy(2)(\delta u(3,2) - \delta u(2,2)) + dx(2)(\delta v(2,3) - \delta v(2,2)) &= -q(2,2)
\end{aligned} \tag{7.33}$$

Les équations de correction de vitesse sont de deux types:

$$\delta u = 0 \tag{7.34}$$

pour les conditions aux limites de type Dirichlet.

$$f(\delta u) + g(\delta p) = -r \tag{7.35}$$

Pour les points internes et les conditions de type Neuman.

f et g sont des fonctions linéaires et r le résidu.

En faisant la somme des équations 7.33, on construit l'équation:

$$\delta Q = -Q \tag{7.36}$$

avec:

$$\begin{aligned}
Q &= dy(1)(u(3,1) - u(1,1)) + dx(1)(v(1,3) - v(1,1)) \\
&\quad dy(2)(u(3,2) - u(1,2)) + dx(2)(v(2,3) - v(2,1))
\end{aligned} \tag{7.37}$$

et:

$$\begin{aligned}
\delta Q &= dy(1)(\delta u(3,1) - \delta u(1,1)) + dx(1)(\delta v(1,3) - \delta v(1,1)) \\
&\quad dy(2)(\delta u(3,2) - \delta u(1,2)) + dx(2)(\delta v(2,3) - \delta v(2,1))
\end{aligned} \tag{7.38}$$

Dans cette équation, n'apparaissent que les composantes de vitesse situées sur la frontière. Elle équivaut en fait à la conservation globale de la masse. D'autre part, étant donné qu'elle est une combinaison linéaire des équations de continuité, elle peut se substituer à n'importe laquelle d'entre elles. Le but de cette partie est d'utiliser cette équation pour voir dans quel cas le système à résoudre est singulier. Dans la pratique trois cas de figure peuvent se présenter pour les conditions aux frontières du domaine de la figure 7.3 (voir figure 7.1):

1. Domaine rectangulaire complet avec des conditions de Dirichlet sur toutes les frontières.

2. Domaine rectangulaire complet avec condition de Dirichlet partout (fluide entrant en face Ouest), sauf sur la face Est où l'on a des conditions de Neumann.
3. Domaine rectangulaire incomplet avec frontière interne sur la face Est (cas de la décomposition en zone).

Pour tous ces cas nous allons considérer une inversion directe de tout le système.

1. Premier cas:

Les équations de correction pour les inconnues de l'équation 7.36 sont toutes de type 7.34. Le premier membre est donc nul, et le système est singulier. Si le problème est bien posé, le second membre est nul. On attribue généralement une équation de continuité à chaque inconnue de pression. Dans ce cas l'inconnue correspondant à 7.36 pourra prendre une valeur quelconque sans modifier la solution pour le champ de vitesse.

2. Deuxième cas:

Les équations de correction pour les composantes de vitesse sont toutes de type 7.34, sauf celle correspondant à la sortie qui sont du type 7.35. Par conséquent le premier membre de 7.36 est non-nul, il n'y a donc pas de problème de pivot.

3. Troisième cas: Ce cas est intéressant dans la mesure où le rôle du maillage décalé apparaît clairement. En effet pour une zone, la discrétisation par un maillage décalé donne un nombre d'inconnues $(N_I + 1)N_J$ pour la composante de u , $N_I(N_J + 1)$ pour la composante v et $N_I N_J$ pour p . Le regroupement par trois des inconnues a pour conséquence directe la non prise en compte de certains degrés de liberté dans le système à inverser pour une zone. C'est le cas notamment des composantes de vitesse u correspondant à la section $N_I + 1$. Pour ces composantes, on adopte implicitement des équations de correction de type 7.35. Cette situation nous ramène au premier cas de figure. Nous donnons les différentes possibilités de regroupement sur la figure 7.4.

Au regard de cette analyse, nous concluons qu'un seul cas de figure justifie que l'on fixe un point de pression c'est celui où toutes les conditions au limites du domaine sont de type Dirichlet. Dans les autres cas, l'apparition d'une singularité est directement lié à l'algorithme d'inversion mis en oeuvre. Les solutions apportées dans ce cas doivent assurer la consistance du problème résolu avec le problème posé.

7.5 Les techniques de lissage étudiées

Mis à part les problèmes évoqués précédemment pour le traitement de la pression, la méthode par zones pose d'autres problèmes liés à la propagation de l'erreur dans le domaine. La figure 7.5 montre pour le cas de la cavité et, pour différents types de

subdivisions l'aspect de l'erreur après quelques itérations par une méthode par zones. On voit que les faibles tailles de zone donnent une erreur qui reste importante en valeur absolue, mais dont la répartition est lisse dans le domaine. Les tailles de zone plus importantes font apparaître des stries marquant clairement la délimitation des zones. On peut cependant remarquer pour ce dernier cas que le sens de parcours (x ou y) a également une influence sur le résultat. Ces instabilités peuvent dégénérer au point de faire diverger les calculs.

Braaten [7] a signalé différentes solutions pour ce problème. Il a notamment cité les méthodes utilisant une approximation monodimensionnelle pour l'écoulement et une méthode incluant un calcul préalable sur un maillage plus lâche. Dans ce même article, il propose un algorithme basé sur la première méthode, la méthode PBC pour (Parabolic Block Corrected). Nous avons implémenté cette méthode ainsi qu'une autre méthode basée sur une version réduite de l'algorithme SIMPLER.

7.5.1 La méthode P.B.C. (Parabolic Block Correction)

L'algorithme PBC "Parabolic Block Correction" proposée par Braaten [7], consiste à calculer une correction de vitesse pour une section d'écoulement en faisant l'approximation que la correction de pression correspondante est constante sur toute la section. Les corrections de vitesse et de pression sont calculées de manière à satisfaire la conservation de la masse sur la section j_0 (figure 7.6).

La correction pour la ligne j_0 a pour but de modifier $v(i, j_0)$ et $p(i, j_0)$ de sorte que la masse se conserve dans la bande hachurée:

$$\dot{M} = \rho \cdot \sum_{i=1}^{i=N_I} \{ \Delta x(i) \cdot (v(i, j_0) - v(i, j_0 - 1)) \} + \Delta y(j_0) \cdot (u(N_I, j_0) - u(1, j_0)) \quad (7.39)$$

Nous disposons pour cela des N_I équations de quantité de mouvement 5.28.

On pose :

$$\begin{aligned} v(i, j_0) &= v'(i, j_0) + dv(i, j_0) \\ p(i, j_0) &= p'(i, j_0) + dp(i, j_0) \end{aligned} \quad (7.40)$$

On peut alors construire l'équation de correction suivante:

$$\begin{aligned} A_p^v \cdot dv(i, j_0) - A_w^v \cdot dv(i-1, j_0) - A_e^v \cdot dv(i+1, j_0) + \\ \Delta x(i) \cdot dp(i, j_0) &= -Res^v(i, j) \end{aligned} \quad (7.41)$$

qui se met sous la forme:

$$-a(i).dv(i-1, j_0) + b(i).dv(i, j_0) - a(i).dv(i+1, j_0) + d(i).dp(i, j_0) = e(i) \quad (7.42)$$

et l'équation de conservation de la masse devient :

$$\begin{aligned} \dot{M} = \rho \cdot \sum_{i=1}^{i=N_I} \{ \Delta x(i).dv(i, j_0) \} + \\ \sum_{i=1}^{i=N_I} \{ \Delta x(i).(v'(i, j_0 + 1) - v'(i, j_0)) \} + \\ \Delta y(j_0).(u(N_I, j_0) - u(1, j_0)) \end{aligned} \quad (7.43)$$

on pose

$$\begin{aligned} \dot{M}' = \rho \cdot \sum_{i=1}^{i=N_I} \{ \Delta x(i).(v'(i, j_0 + 1) - v'(i, j_0)) \} + \\ \Delta y(j_0).(u'(N_I, j_0) - u'(1, j_0)) \end{aligned} \quad (7.44)$$

On ne dispose que de $N_I + 1$ équations pour déterminer les $2N_I$ inconnues que sont les $dv(i, j_0)$ et les $dp(i, j_0)$, ce qui est insuffisant. C'est ici que l'on fait l'hypothèse selon laquelle l'écoulement se fait par droites parallèles. Dans ce cas la correction $dp(j_0)$ peut être considérées comme uniforme sur toute la ligne.

Soit pour tout i , $dp(i, j_0) = dp(j_0)$

$$-a(i).dv(i-1, j_0) + b(i).dv(i, j_0) - c(i).dv(i+1, j_0) = -d(i).dp(j_0) + e(i) \quad (7.45)$$

$$\dot{M} = \sum_{i=1}^{i=N_I} \{ \Delta x(i).dv(i, j_0) \} + \dot{M}' \quad (7.46)$$

L'équation 7.45 représente un système tridiagonal qui se transforme en:

$$dv(i, j_0) = A(i) + B(i).dp(j_0) \quad (7.47)$$

En reportant dans 7.46, on obtient facilement :

$$dp(j_0) = \frac{\{M - M'\} - \sum_{i=1}^{i=N_I} \{A(i) \cdot \Delta x(i)\}}{\sum_{i=1}^{i=N_I} \{B(i) \cdot \Delta x(i)\}} \quad (7.48)$$

La valeur de $dp(j_0)$ permet alors de calculer $dv(i, j_0)$ à l'aide de 7.47. Les formules complètes sont dans l'article de Braaten et Patankar [7].

On dérive un système similaire pour u dans la direction y . Après chaque itération à l'aide de la méthode par zones, on effectue un balayage pour $j = 1$ à nmy pour corriger $v(i, j)$. Puis de $i = 1$ à N_I pour corriger $u(i, j)$. Les deux corrections de pression s'ajoutant.

7.5.2 Une méthode dérivée de l'algorithme SIMPLER

Lorsque l'utilisation d'une technique de lissage devient nécessaire, son but est de forcer le respect de la conservation de la masse pour le domaine tout entier si cela est possible. L'une des manières de le faire au moindre de coût est de calculer une bonne approximation d'un champ de pression compatible avec la conservation de la masse. La plupart des méthodes de lissage procèdent en reprenant la dernière partie de l'algorithme SIMPLER. Calcul d'une correction de pression et évaluation des formules de correction de vitesse. Les équations pour la pression sont les mêmes. On diminue les coûts en choisissant un algorithme rapide pour la résolution de cette équation qui ressemble beaucoup à celle que l'on obtient après discrétisation du laplacien. La résolution numérique de cet opérateur a fait l'objet d'importantes études à cause de sa présence dans diverses branches des sciences de l'ingénieur. On pourra trouver certains algorithmes dans [18].

La méthode que nous allons présenter ici fait apparaître quelques différences dans l'évaluation des seconds membres des équations par rapport à la méthode SIMPLER.

On reprend les équations discrétisées données par 5.26, 5.27 et 5.28:

On pose:

$$u(i, j) = u'(i, j) + \delta u(i, j) \quad (7.49)$$

$$v(i, j) = v'(i, j) + \delta v(i, j) \quad (7.50)$$

$$p(i, j) = p'(i, j) + \delta p(i, j) \quad (7.51)$$

$$p(i-1, j) = p'(i-1, j) + \delta p(i-1, j) \quad (7.52)$$

$$p(i, j+1) = p'(i, j+1) + \delta p(i, j+1) \quad (7.53)$$

A chaque itération, u' , v' et p' sont donnés par la méthode par zones. δu , δv et δp sont les corrections à déterminer.

Les deux équations de quantité de mouvement donnent:

$$A_p^u \cdot \delta u(i, j) + \Delta y(j) \cdot (\delta p(i, j) - \delta p(i - 1, j)) + Resu(i, j) = 0 \quad (7.54)$$

$$A_p^v \cdot \delta v(i, j) + \Delta x(i) \cdot (\delta p(i, j) - \delta p(i, j - 1)) + Resv(i, j) = 0 \quad (7.55)$$

l'équation de continuité:

$$\Delta y(j) \cdot (\delta u(i + 1, j) - \delta u(i, j)) + \Delta x(i) \cdot (\delta v(i + 1, j) - \delta v(i, j)) + Resc(i, j) = 0 \quad (7.56)$$

Une inversion partielle en $\delta u(i, j)$ et $\delta v(i, j)$ de 7.54 donne:

$$\delta u(i, j) = \frac{-Resu(i, j)}{A_p^u} + \frac{\Delta y(j)}{A_p^u} \cdot (\delta p(i - 1, j) - \delta p(i, j)) \quad (7.57)$$

$$\delta v(i, j) = \frac{-Resv(i, j)}{A_p^v} + \frac{\Delta y(j)}{A_p^v} \cdot (\delta p(i, j) - \delta p(i, j - 1)) \quad (7.58)$$

La substitution de ces valeurs dans 7.56 donne une équation pour la pression similaire à celles obtenues pour les méthodes du type SIMPLE. La différence réside dans le calcul du second membre.

L'équation pour la correction de la pression s'écrit :

$$A_p^p \cdot \delta p(i, j) - \sum_{q \in \text{Vois}} A_q^p \cdot \delta p_q - Resp(i, j) = 0 \quad (7.59)$$

avec:

$$Resp(i, j) = \Delta y(j) \cdot \left\{ \frac{Resu(i + 1, j)}{A_p^u(i + 1, j)} - \frac{Resu(i, j)}{A_p^u(i, j)} \right\} \\ + \Delta x(i) \cdot \left\{ \frac{Resv(i, j + 1)}{A_p^v(i, j + 1)} - \frac{Resv(i, j)}{A_p^v(i, j)} \right\} - Resc(i, j) \quad (7.60)$$

Les A_p sont les mêmes que pour l'équation correspondante de SIMPLER.

La résolution de 7.59 permet de déterminer la correction de pression $\delta p(i, j)$. Les équations 7.57 permettent de calculer $\delta u(i, j)$ et $\delta v(i, j)$ et de corriger le champ de vitesse, terminant ainsi l'itération en cours. La précision avec laquelle l'équation pour la pression est résolue conditionne le niveau de vérification de la conservation globale de la masse. Dans notre cas nous avons utilisé une méthode de balayage par ligne en résolvant des systèmes tridiagonaux à l'aide de l'algorithme de Thomas. Cette méthode peut devenir coûteuse dès que la taille du problème augmente.

7.5.3 Résultat du test de PBC et de notre méthode de lissage

Nous avons testé les deux méthodes précédentes sur deux cas de figures avec pour chacun deux tailles de maillage. Ces deux méthodes sont comparées à une troisième (UVP) qui consiste à résoudre implicitement le système linéaire pour tous les points du domaine. Nos résultats sont conformes à ceux de Braaten [7].

La cavité:

Sur les figures 7.7a et 7.7b, on a représenté les courbes de convergence des trois méthodes en fonction du nombre d'itérations, respectivement pour les nombres de Reynolds $Re = 100$ et $Re = 1000$. On voit que pour $Re = 100$, la méthode PBC est meilleure que les deux autres méthodes, alors que pour $Re = 1000$ la méthode UVP converge avec moins d'itérations que les deux autres méthodes. Pour ce dernier Reynolds, la méthode PBC est fortement pénalisée. Ceci peut s'expliquer par le caractère fortement bidimensionnel de l'écoulement dans ce cas. Le déficit du découpage par zones pour résoudre ce type d'écoulement semble compensé pour notre méthode par le lissage SIMPLER. Ce comportement confirme la bonne tenue de la méthode SIMPLER comparable à UVP, aussi bien pour nos calculs que pour ceux de Braaten [7].

La marche descendante:

Les courbes de convergence pour le cas de la marche sont représentées sur les figures 7.8a et 7.8b. Les méthodes comparées sont les mêmes, les nombres de Reynolds aussi. A bas Reynolds, les trois méthodes ont des performances comparables, avec un léger avantage pour la méthode UVP et un désavantage pour notre méthode. Pour un nombre de Reynolds $Re = 1000$, les différences s'accroissent. La méthode UVP garde l'avantage, alors que notre méthode est de plus en plus pénalisante. La méthode PBC a un bon comportement qui est confirmé par la meilleure performance en temps de calcul des trois méthodes. Les hypothèses d'écoulement monodimensionnel sont ici assez bien vérifiées. L'inefficacité de la méthode SIMPLER pour ce type d'écoulement voir [7] se répercute sur notre méthode de lissage et donne ici les plus mauvaises performances. Ces deux algorithmes ont cependant le défaut à nos yeux d'être composés de deux parties: une méthode principale (la méthode par zones), et une méthode de lissage, sans que cela les rendent plus polyvalents. Nous avons donc essayé de trouver une méthode plus simple à partir de l'expérience que nous avons acquise.

7.5.4 La méthode par zones adaptée au sens de l'écoulement

A la lumière de l'étude de ces deux méthodes de lissage, nous avons abouti à la conclusion que dans certaines situations, le couplage entre la vitesse et la pression peut poser d'importants problèmes. Pour illustrer ceci, nous avons effectué trois calculs par zones dont les résultats sont présentés sur la figure 7.9 et les

numérotations correspondantes sur les figures 7.10. La figure 7.10a représente un écoulement de canal plan avec tous les degrés de libertés à déterminer. La partie hachurée représente la zone sur laquelle s'effectuent les calculs. Sur les figures 7.10b et 7.10c nous avons représenté uniquement les degrés de liberté mis à jour pour chacune des numérotations.

Le premier cas (figure 7.9a) se fait sans aucune précaution, la numérotation (figure 7.10b) des variables aboutit à un système singulier et on voit bien que le fluide n'a pas beaucoup avancé entre l'entrée et la sortie de la zone. Dans le deuxième cas, (figure 7.9b et 7.10b) on a utilisé un coefficient de pénalité pour éviter le problème de pivot nul, en gardant la même numérotation. On voit que dans ce cas le fluide se déplace plus loin, même si la condition provisoire à la sortie de la zone impose une valeur nulle pour la vitesse normale. Le fluide se comprime pour s'adapter à l'impossibilité de sortir.

La figure 7.9c montre finalement les résultats que l'on obtient en changeant la numérotation des variables (figure 7.10c) au sein de la zone active. Cette numérotation a pour conséquence de permettre le calcul de correction pour les composantes normales de la vitesse en sortie. Dans les deux autres cas, ces composantes restaient fixées aux valeurs de l'itération précédente, c'est à dire zéro pour la première itération. Cette numérotation des inconnues réalise une synthèse de la méthode par zones et de la correction PBC. Par conséquent elle en a aussi bien les défauts que les qualités. Il faut noter que cette méthode ne marche que parce que le débit imposé sur la face d'entrée est donné par les conditions aux limites réelles.

En adoptant la numérotation $(u(i+1, j), v(i, j), p(i, j))$ (type IV sur la figure 7.4), on donne à $u(3, 1)$ et $u(3, 2)$ des équations de correction de type 7.35. On évite le pivot nul et le débit est entièrement imposé par la face d'entrée. La qualité des résultats obtenus par cette dernière numérotation nous a permis d'envisager son utilisation sans méthode de lissage. Elle a cependant un handicap, celui de nécessiter la prise en compte d'un changement de direction de l'écoulement d'un sous-domaine à l'autre. Ceci a été prévu dans l'implémentation de notre code.

7.6 Résolution des équations du modèle turbulent

Dans son article sur la résolution implicite des équations de Navier-Stokes, Vanka [65] a signalé son échec dans une tentative de résolution couplée des équations du modèle turbulent. Il a finalement réussi à résoudre le problème en traitant séparément chaque variable, à l'aide d'un algorithme de balayage par ligne. En ce qui nous concerne, nous avons conservé la méthode utilisée dans le TEACHT de Gossman et al. Elle consiste à linéariser les deux équations pour k et ϵ , puis à résoudre le système résultant par l'algorithme de Thomas. Il y a cependant quelques précautions à prendre pour le traitement des termes sources. Ceux-ci, par leur importance dans les équations du modèle de turbulence, posent des problèmes de stabilité pour leur résolution. La discrétisation upwind avec un pseudo algorithme de Newton (voir

6.3.2) a l'avantage de fournir des systèmes linéaires stables. Dans le cas où les termes source dominant, cette solution ne suffit plus, il faut un traitement spécifique pour ces derniers. Nous avons adopté la solution proposée par Chien [11]. Elle consiste à séparer le terme source S_ϕ en deux parties:

$$S_\phi = A_\phi \phi_p + S'_\phi \quad (7.61)$$

Le terme $A_\phi \phi_p$ est additionné au terme diagonal afin d'améliorer le conditionnement du système. La même méthode est utilisée pour la résolution des équations de transport électrochimique. Pour cette équation, on a cependant pu utiliser l'estimation numérique de la matrice jacobienne.

7.7 Inversion des systèmes linéaires

Les algorithmes que nous avons présentés aboutissent soit à un système tridiagonal scalaire, soit à un système bloc tridiagonal. Les premiers systèmes sont résolus par l'algorithme classique de Thomas [34]. Pour les seconds, des codes généraux pour matrices creuses sont habituellement utilisés. Ces codes n'exploitent cependant pas directement les caractéristiques du système. L'algorithme que nous avons mis au point suit les mêmes lignes que l'algorithme tridiagonal de Thomas. La principale différence réside dans le remplacement des opérations arithmétique normales par des opérations sur les matrices ce qui donne l'algorithme suivant, pour $j_0 = 1$:
lire B_1, C_1 et D_1

$$\bar{C}_1 = B_1^{-1} \cdot C_1$$

et

$$\bar{D}_1 = B_1^{-1} \cdot D_1$$

Pour $j = 2$ à $j = j_1$

début

lire A_j, B_j, C_j et D_j

$$\bar{C}_j = (B_j - A_j \cdot \bar{C}_{j-1})^{-1} \cdot C_j$$

$$\bar{D}_j = (B_j - A_j \cdot \bar{C}_{j-1})^{-1} \cdot (D_j - A_j \cdot \bar{D}_{j-1})$$

fin

Pour $j = j_1 - 1$ à $j = 1$
début

$$\phi_j = \overline{C}_j \cdot \phi_{j+1} + \overline{D}_j$$

fin

L'une des caractéristiques de cet algorithme est la possibilité de traiter des systèmes bloc diagonaux avec des blocs de taille variable. Ceci permet d'optimiser les coûts pour des domaines de calcul non rectangulaires. Son extension au cas tridimensionnel ne devrait pas poser de problèmes majeurs.

7.8 Conclusion

L'algorithme SIMPLER qui dérive de la famille des algorithmes SIMPLE de Patankar [46] procède par découplage des trois variables du modèle hydrodynamique. Ce découplage permet un gain important en terme de place mémoire lors de l'inversion des systèmes par l'algorithme de Thomas. Ce découplage rend l'algorithme instable notamment pour le calcul d'écoulement à fort nombre de Reynolds. La méthode implicite par zones que nous avons présentée est plus robuste, à condition d'être accompagnée d'un algorithme de lissage pour le champ de pression et l'équation de continuité. Nous avons aussi montré dans ce travail (performance des algorithmes) qu'une bonne prise en compte des caractéristiques de l'écoulement peut améliorer considérablement les performances du code.

Chapitre 8

La méthode multigrille

8.1 Introduction

La résolution des équations aux dérivées partielles sur ordinateur peut se décomposer en deux étapes principales: l'approximation des équations, et leur résolution. L'approximation est représentée par deux paramètres h et p . Le premier désigne un pas de maillage et le second un ordre d'interpolation pour la reconstruction de la solution. La philosophie des méthodes adaptatives est de considérer que le choix d'une valeur pour chacun de ces paramètres, pour être optimal, doit se faire localement en fonction des propriétés de la fonction à déterminer. On définit donc pour simplifier les notations:

$$E(h, p) = \int_{\Omega} G(X) \cdot \tau(h(X), p(X)) \cdot \delta\Omega \quad (8.1)$$

et

$$W(h, p) = \int_{\Omega} W(h(X), p(X)) \cdot \delta\Omega \quad (8.2)$$

La première fonction représente l'erreur moyenne dans le domaine. Elle est déterminée à partir de l'erreur de troncature τ qui elle est locale. G désigne une pondération qui permet de privilégier certains points de Ω par rapport à d'autres. La deuxième fonction mesure le travail nécessaire pour la résolution des équations approchées.

Un processus de résolution optimal doit permettre de faire le meilleur choix de h et p permettant d'obtenir des variations données de W et de E . Les deux cas possibles sont les suivants: pour le premier, les ressources W sont limitées, alors il faut rechercher h et p pour que ΔE diminue le plus possible. Dans l'autre cas, la valeur maximale de E est fixée. il faut trouver W le plus faible qui permette d'atteindre cet objectif. La mise en oeuvre de cette méthode pose des problèmes de deux ordres: les problèmes informatiques et les problèmes mathématiques. Les problèmes informatiques sont liés à la manipulation de plusieurs valeurs de h c'est à dire plusieurs maillages, et plusieurs valeurs de p c'est à dire plusieurs schémas de discrétisation. Le problème mathématique se pose pour le calcul explicite de

τ l'erreur de troncature. Ces différents problèmes expliquent le fait que les codes adaptatifs soient si peu répandus, ce malgré le bien fondé de leur principe. Les codes les plus courants manipulent un seul h et un seul p à la fois. L'interaction entre la phase d'approximation et celle de résolution est souvent lente (dans la mesure où l'on teste manuellement l'indépendance de la solution par rapport au maillage), voire inexistante. Pour justifier le recouplage et la complexité des codes qui en résulte, il fallait une raison. L'argument type pour promouvoir les méthodes multigrilles consiste à dire que le processus de résolution peut être optimisé au point de rendre le coût de la méthode proportionnel à la taille N du problème. Alors que le coût des meilleurs méthodes traditionnelles monogrilles est $N \log(N)$ (Transformée de Fourier Rapide). Cette perspective est tout à fait intéressante.

Dans un article publié en 1977 et qui sert depuis lors de référence pour la méthode multigrille, A. Brandt [8] se propose de créer ou de recréer une interaction entre le processus de discrétisation, le choix d'un ordre de précision pour l'interpolation, et l'algorithme de résolution. Certains auteurs avant Brandt ont démontré les bénéfices que l'on pouvait tirer de l'utilisation de plusieurs maillages pour la résolution d'un même problème. La technique consiste à utiliser un maillage lâche pour accélérer la convergence sur un maillage plus fin. Il s'agit notamment de Southwell [57] dans les années quarante, et de Fedorenko [22] dans les années soixante. Trois autres auteurs importants ont travaillé dans la même période que Brandt il s'agit de:

1. W. Hackbush [30]
2. U. Trottenberg [30]
3. K. Stüben [60]

La méthode multigrille est plus qu'une simple méthode d'accélération de convergence. C'est une technique ou une philosophie qui permet d'améliorer l'organisation du processus de résolution de manière à le rendre optimal. Sous sa forme complète, telle que proposée par A. Brandt, la méthode multigrille est un processus adaptatif. Les paramètres d'adaptation sont le pas de maillage, qui peut être uniforme, et la précision du schéma de discrétisation. Sous cette forme, même les domaines de forme complexes peuvent être traités en utilisant un maillage uniforme sur chaque niveau de raffinement.

L'amélioration de l'efficacité des codes passe par une complexification des architectures de ces derniers. L'utilisation de maillage non structuré et le découplage rendu nécessaire entre les problèmes de maillage et ceux concernant la construction et la résolution des équations discrètes en donne des exemples. La méthode multigrille n'échappe pas à la règle.

L'implémentation de la méthode se réduit en grande partie à la conception et à la programmation d'une architecture facilitant l'interaction entre les différents éléments du code. La variété de ces éléments résulte de la possibilité de faire varier h et p . A ces deux paramètres, il faut ajouter pour nous le paramètre de calcul par zone ou multi-domaine. La difficulté d'implémentation d'une telle architecture justifie l'affirmation

selon laquelle, la méthode multigrille est facile à implémenter à partir du moment où l'on dispose d'une organisation efficace des différents niveaux de maillage. Nous avons implémenté une méthode multigrille non adaptative basée sur le schéma de discrétisation hybride et sur l'algorithme SIMPLER comme méthode de résolution de base. Cette méthode donne des résultats satisfaisant pour les écoulements laminaires. Malheureusement elle ne converge pas pour les modèles turbulents. Pour introduire la méthode, nous allons considérer le cas de la résolution d'un problème linéaire par utilisation de deux maillages.

8.2 Relaxation sur deux maillages

Dans cette partie nous allons exposer une démarche qui permet de relier la méthode multigrille aux autres méthodes de relaxation présentées plus tôt. Nous montrerons également que la méthode permet un partage des tâches dans le processus de résolution entre les différents niveaux de maillage.

Les algorithmes de résolution peuvent se représenter sous la forme générale de processus prenant en entrée une erreur et donnant en sortie une erreur plus petite si le processus converge. L'équation 6.8 en donne une représentation matricielle.

L'étude des algorithmes est grandement facilitée lorsque l'on connaît les fonctions propres de l'opérateur à M_h . C'est la raison pour laquelle les problèmes modèles comme celui du laplacien sont importants pour l'analyse des méthodes de résolution. Les fonctions propres de cet opérateur sont connues. Ce qui permet de concevoir des expériences simples par lesquelles on arrive à visualiser les propriétés des algorithmes de résolution [68]. C'est ainsi qu'on peut voir que les algorithmes de type Gauss-Seidel sont plus efficaces pour réduire les composantes de l'erreur qui fluctuent rapidement dans le domaine de calcul. On pourrait montrer avec le même type d'expériences que les algorithmes plus coûteux tels que l'inversion directe sont moins sélectifs. Comment exploiter ces différentes remarques pour optimiser un algorithme quelconque de type Gauss-Seidel. Pour cela reprenons l'équation de correction 6.6:

$$K_h \cdot \delta\phi_h = -Res_h(\phi_h^n) \quad (8.3)$$

avec $\phi_h^{n+1} = \phi_h^n + \alpha_n \cdot \delta\phi_h$.

où K_h est une approximation de L_h . Après quelques itérations sur le maillage h , les composantes hautes fréquences de l'erreur sont éliminées. L'erreur restante peut alors être représentée sur un maillage moins dense. Cette nouvelle représentation peut s'obtenir par une procédure de projection quelconque, que nous noterons R_H . On obtient:

$$Res_H(\phi_h^n) = R_H \cdot Res_h(\phi_h^n) \quad (8.4)$$

A ce nouveau résidu correspond une correction δ_H que nous définissons en introduisant l'opérateur d'interpolation P_h :

$$\delta\phi_h = P_h.\delta\phi_H \quad (8.5)$$

La substitution de cette expression dans 8.3 donne:

$$K_h.P_h.\delta\phi_H = -Res_h(\phi_h^n) \quad (8.6)$$

La multiplication de cette dernière équation par R_H donne:

$$R_H.K_h.P_h.\delta\phi_H = -Res_H(\phi_h^n) \quad (8.7)$$

On vient d'obtenir l'équation pour la correction sur un maillage plus lâche. La détermination de $\delta\phi_H$ sera moins lourde du fait de la moins grande densité du maillage. L'interpolation est en général une opération peut coûteuse. On peut donc remonter à $\delta\phi_h$. Cette brève présentation du principe de la relaxation sur deux maillages nous a permis d'introduire les principales opérations de la méthode multigrille à savoir:

1. Un algorithme qui agit sélectivement sur les composantes haute fréquence de l'erreur.
2. Un opérateur de projection pour transférer les composantes lisses de l'erreur vers un maillage plus lâche.
3. Un opérateur d'interpolation pour transférer les corrections vers le maillage d'origine.

Un algorithme multigrille normal utilise en général plus de deux maillages et un algorithme de résolution supplémentaire permettant, d'éliminer les dernières composantes de l'erreur sur le maillage le plus lâche.

8.3 Discrétisation multigrille

Dans la présentation plus détaillée qui va suivre, nous allons essayer de faire ressortir les liens avec les méthodes classiques. Certaines éléments de la méthode ne sont que des adaptations des éléments déjà étudiés. Il s'agit du maillage, de la discrétisation et de la résolution. Certains éléments semblent nouveaux, comme la projection et l'interpolation, alors qu'il n'en est rien. Leur fonction peut d'ailleurs mieux éclairer la logique des méthodes classiques.

8.3.1 Les maillages lâches

Le problème peut être abordé de deux manières différentes:

La première consiste à se servir du maillage fin pour composer les maillages lâches successifs. La technique consiste à agglomérer des éléments du maillage fin pour obtenir ceux du maillage lâche. Cette méthode ne pose pas de problème pour des maillages rectangulaires, car les éléments obtenus sont du même type. Par contre, dans le cas des maillages non-structurés, la situation est différente. Pour un domaine maillé à l'aide de triangles, l'agglomération va donner des éléments de forme aussi bien triangulaire, que quadrilatérale ou octogonale. De tels éléments ne sont évidemment pas adaptés aux algorithmes prévus pour les éléments triangulaires. D.J. Mavrilis propose une solution [41].

La deuxième méthode consiste à choisir d'abord un maillage lâche, puis à construire tous les autres maillages à partir de ce dernier, par subdivision. Cette technique a l'avantage de conserver la forme des éléments. La contrepartie réside dans la difficulté de choix pour le maillage lâche dans le cas de domaines complexes, et les conséquences sur la précision de la représentation du domaine. En effet pour les géométries complexes, le maillage fin est choisi en fonction de cette contrainte de bonne représentation du domaine. Cette méthode convient parfaitement aux maillages rectangulaires, même si elle pose quelques problèmes pour les maillages non-uniformes.

Nous avons choisi la première méthode afin de pouvoir fixer le point le plus proche de la paroi. Ce dernier étant fonction de l'épaisseur de la couche limite. Nous avons également choisi un processus d'agglomération séparé pour chacune des directions x et y (figure 8.1). Ceci afin de faciliter le traitement des écoulements possédant une direction privilégiée. Le regroupement des éléments se fait par deux. Le rapport deux, entre les tailles de maillages successifs dans le cas uniforme semble être optimal d'après [9]. On peut cependant se poser la question quant au respect de ce rapport dans le cas des maillages non structurés, ou même simplement non-uniforme. A l'origine nous avons prévu la possibilité de traiter des rapports différents. Cette possibilité a été partiellement implémentée. Il serait cependant judicieux de ne s'en servir qu'après une étude de son impact sur l'efficacité du code. Efficacité qui dépendra du type de problème à résoudre.

8.3.2 Projection et Interpolation

Considérons deux maillages représentés par les pas uniformes h et H . La projection et l'interpolation sont des opérations de transfert d'un espace de fonction à un autre. En ce qui nous concerne, le transfert de fonctions définies sur Ω_H à celle définies sur Ω_h et vis-versa. En tant que telles, elles doivent posséder des propriétés de conservation par rapport aux fonctions auxquelles elles sont appliquées. Pour l'étude de ces propriétés, on se reportera à l'article de W. Hemker [33].

L'utilisation du maillage décalé pour la discrétisation des équations, nous amène à

prendre séparément en compte les différents types de variables. Nous avons choisi de regrouper d'une part u et v , et d'autre part toutes les autres variables qui sont positionnées au centre des éléments voir figure 8.2 et 8.3. De plus le choix de découpler les directions x et y , nous permet de ne considérer que le cas monodimensionnel. L'opération de projection concerne aussi bien les variables du problème que les résidus calculés sur les maillages fins. Alors que l'interpolation ne concerne que les corrections. La discrétisation par une méthode de volumes finis conduit à distinguer la projection des variables de celle des résidus contrairement au cas des différences finies. La distinction provient du fait que dans le premier cas le résidu est une valeur intégrale sur un élément de volume, alors que dans l'autre cas c'est une valeur locale. Pour présenter les formules de transfert que nous avons utilisé, nous allons adopter les notations suivantes: ϕ est une inconnue et $r\phi$ le résidu de l'équation correspondant à ϕ . s est une variable d'espace qui représente soit la direction x soit la direction y . Les indices i correspondent au maillage fin et les indices k au maillage lâche. Nous avons choisi d'utiliser l'opérateur de projection le plus recommandé, c'est à dire la moyenne pondérée pour les variables centrées, l'injection directe pour les variables décalées et la somme pondérée pour tous les résidus. Les indices i et k sont ceux décrit par les figures 8.2 et 8.3. Pour la projection des variables décalées, on obtient:

$$\phi(k) = \phi(i) \quad (8.8)$$

$$r\phi(k) = (2r\phi(i-1) + r\phi(i-1) + 2r\phi(i+1)) \frac{V'(i)}{V(i)} \quad (8.9)$$

avec

$$V'(i) = \frac{ds(i-1)}{2} + ds(i) + \frac{ds(i+1)}{2}$$

et

$$V(i) = ds(i-1) + ds(i) + ds(i+1)$$

Pour les variables non-décalées:

$$\phi(k) = (ds(i).\phi(i) + ds(i+1).\phi(i+1)) \frac{1}{V(i)} \quad (8.10)$$

$$r\phi(k) = r\phi(i-1) + r\phi(i+1) \quad (8.11)$$

avec cette fois

$$V(i) = ds(i) + ds(i+1)$$

L'opération d'interpolation des corrections est assurée par une interpolation linéaire. A savoir pour les variables décalées:

$$\delta\phi(i-1) = \frac{(s(k) - s(i-1)).\delta\phi(k-1) + (s(i-1) - s(k-1)).\delta\phi(k)}{s(k) - s(k-1)} \quad (8.12)$$

et pour les variables non-décalées:

$$\delta\phi(i) = \frac{(s(k) - s(i))\delta\phi(k - 1) + (s(i) - s(k - 1))\delta\phi(k)}{s(k) - s(k - 1)} \quad (8.13)$$

Ces formules sont appliquées dans chaque direction avec la variable d'espace correspondante x ou y .

Les frontières sont des zones qui requièrent un soin spécifique notamment pour l'opération d'interpolation. Nous allons montrer les risques en considérant le cas de la pression illustré sur la figure 8.5. L'indice j_{pl} est l'indice le plus proche de la paroi pour le maillage lâche et j_{pf} celui du maillage fin. Le maillage décalé ne nécessite pas de conditions aux limites pour la pression sur les parois imperméables. On considère implicitement que cette valeur est nulle et n'intervient pas dans la discrétisation. Par contre, pour calculer lors de l'interpolation les valeurs situées en j_{pf} , on a besoin de celles situées en $j_{pl} - 1$ et qui n'intervenaient pas dans la discrétisation des équations. On se trouve obligé de faire appel à des techniques qui jusque là étaient réservées aux calculs effectués sur un maillage non-décalé. Avec ce problème, on peut se poser la question de savoir s'il ne serait pas plus judicieux d'adopter le maillage non décalé de manière à procéder à un traitement homogène de la pression, en discrétisation et en interpolation des corrections calculées.

8.3.3 Equations sur le maillage lâche: le cas non-linéaire

Approximation de Galerkin et multigrille

La rediscrétisation des équations est rendue nécessaire par le transfert du problème sur le maillage lâche. La technique la plus générale utilisée pour cela dérive de l'approximation de Galerkin que nous avons déjà rencontré en évoquant la méthode des éléments finis. Comme nous allons le voir plus loin, cette méthode est également la plus naturelle.

Dans la partie discrétisation, nous avons explicité l'opération d'interpolation pour la méthode des éléments finis. On partait d'un vecteur (ϕ_h) contenant les degrés de liberté discrets. La multiplication par des fonctions de forme $(N_h(X))$ permettait d'obtenir l'interpolation sous la forme d'une fonction continue. Nous pouvons également en revenant sur la formulation faible,

$$L_h(\phi_h) = \int_{\Omega} N_h(X) \cdot L(N_h(X) \cdot \phi_h) \cdot d\Omega = \int_{\Omega} f(X) \cdot d\Omega \quad (8.14)$$

donner la forme générale de ce qui peut être considéré comme l'opérateur de projection R agissant sur des fonctions ϕ :

$$R(\phi) = \int_{\Omega} N_h(X) \cdot \phi(X) \cdot d\Omega \quad (8.15)$$

Opération qui consiste à multiplier par la fonction test, puis à intégrer sur l'espace Ω . L'opération d'interpolation correspondante consiste simplement à multiplier par

les fonctions de forme. L'équation 8.14 permet d'obtenir directement l'équation discrétisée sur le maillage h .

La méthode multigrille possède également une approximation dite de Galerkin [68] qui s'écrit de la manière suivante:

$$L_H(\phi_H) = R_h.L_h(P_h.\phi_h) \quad (8.16)$$

Cette approximation permet de donner l'opérateur discrétisé sur le maillage lâche en fonction de celui du maillage fin. On a également défini deux opérateurs de transfert R_h pour la projection et P_h pour l'interpolation. Ces deux opérateurs peuvent être pris conjugués l'un de l'autre. Mais ils peuvent également n'avoir aucun rapport direct entre eux. On retrouve là, les formulations de Bubnov-Galerkin et Petrov-Galerkin. Nous disposons désormais de tous les éléments pour décrire la méthode.

Approximations selon l'algorithme MGFAS

Dans cette section nous allons établir des résultats similaires à ceux obtenus pour le cas linéaire en introduction.

Pour cela nous allons revenir à 6.15 pour expliquer le transfert du problème sur le maillage lâche dans le cas non-linéaire. Nous recherchons donc pour notre estimation $\phi_h^{(n)}$ une correction de la forme 8.5. Pour cela, la substitution de 8.5 dans 6.15 donne:

$$L_h(\phi_h^{(n)} + P_h.\delta\phi_H) - L_h(\phi_h^{(n)}) = Res_h(\phi_h^{(n)}) \quad (8.17)$$

On a N_h équations 8.17 et N_H inconnues $\delta\phi_H$. Le système ainsi formé est surdéterminé. A nouveau nous allons multiplier notre système par R_H . On obtient:

$$R_H.L_h(\phi_h^{(n)} + P_h.\delta\phi_H) - R_H.L_h(\phi_h^{(n)}) = R_H.Res_h(\phi_h^{(n)}) \quad (8.18)$$

Pour obtenir un opérateur L_H tel qu'il est donné par l'approximation de Galerkin, nous allons procéder à la factorisation suivante:

$$\phi_h^{(n)} \simeq P_h.R_H\phi_h^{(n)} \quad (8.19)$$

l'égalité est approchée car ces deux opérateurs ne sont pas inverses l'un de l'autre. En effet ces opérateurs travaillent sur des espaces de dimensions différentes. Une fois cette factorisation (approximative) faite, on obtient l'équation suivante:

$$\begin{aligned} R_H.L_h(P_h.(R_H.\phi_h^{(n)} + \delta\phi_H)) - \\ R_H.L_h(P_h.(R_H.\phi_h^{(n)})) &= R_H.(f_h - L_h(\phi_h^{(n)})) \end{aligned} \quad (8.20)$$

on pose:

$$\tilde{\phi}_H = R_H.\phi_h^{(n)} \quad (8.21)$$

8.3. DISCRÉTISATION MULTIGRILLE

99

$$L_H(\phi_H) = R_H \cdot L_h(P_h \cdot \phi_H) \quad (8.22)$$

$$\tau_H = L_H(\tilde{\phi}_H) - R_H \cdot L_h(\phi_h^{(n)}) \quad (8.23)$$

et

$$f_H = R_H \cdot f_h \quad (8.24)$$

8.20 devient:

$$L_H(\tilde{\phi}_H + \delta\phi_H) = \tau_H + f_H \quad (8.25)$$

Nous avons ainsi obtenu les équations qui vont nous permettre de déterminer $\delta\phi_H$. En faisant le changement de variable $\varphi_H = \tilde{\phi}_H + \delta\phi_H$, 8.25 devient:

$$L_H(\varphi_H) = \tau_H + f_H \quad (8.26)$$

qui est un système non-linéaire sur le maillage lâche.

Remarque:

Le passage de L_h à L_H à partir de l'équation 8.17 fait clairement apparaître la nécessité d'une projection à cause de la surdétermination du système approché.

Pour la méthode des éléments finis la surdétermination apparaît si on supprime l'étape d'intégration après multiplication par la fonction test. Etape que nous avons séparée sous la forme d'une opération de projection. Dans ce cas les variables indépendantes d'espace apparaissent sous forme continue dans les équations. La surdétermination apparaît ici sous la forme d'une infinité continue d'équations pour quelques degrés de liberté (voir 8.3.3).

La méthode des différences finies escamote complètement cette étape de projection en calculant directement l'expression discrète des termes différentiels. C'est l'une des raisons pour lesquelles cette méthode est de plus en plus considérée comme peu fiable pour des maillages trop lâches. En effet certains théoriciens comme M. Germano [28] considèrent qu'il existe un processus implicite (donc incontrôlable) de projection, dans le calcul des différences finies.

Algorithme "Full Approximation Storage"

Les opérations sont finalement regroupées pour former l'algorithme suivant:

1. choisir $\phi_h^{(0)}$ estimation initiale de ϕ_h
2. initialiser $n = 0$
3. tant que $\|Res_h(\phi_h^{(n)})\| > \epsilon$ faire
4. $n+1$ itérations avec la méthode définie plus haut (cas non-linéaire).

- calculer $\tilde{\phi}_H$, $Rhs_H(\phi_h^{(n)})$ et f_H à l'aide 8.21, 8.23, 8.24
 - choisir $\varphi_H^{(0)} = \tilde{\phi}_H$ estimation initiale de φ_H
 - résoudre 8.26 avec la méthode directe pour le cas non-linéaire.
 - calculer $\delta\phi_H = \varphi_H - \tilde{\phi}_H$
 - calculer $\delta\phi_h = P_h.\delta\phi_H$
5. $\phi_h^{(n+1)} = \phi_h^{(n)} + \delta\phi_h$
 6. calculer $Res_h(\phi_h^{(n)})$.
 7. fin faire
 8. fin tant que

8.3.4 Interprétation

Dans cette partie, nous allons examiner le rôle de $\tau_H = L_H(P_h\phi_h) - P_hL_h(\phi_h)$ ainsi que l'interprétation que l'on peut en faire. L'autre point traité dans cette partie concerne les différents choix possibles pour l'approximation de L_H .

Initialement, la technique multigrille est une technique d'accélération des méthodes itératives. Dans cette optique, le calcul démarre sur le maillage fin. Il est ensuite transféré aux maillages plus lâches, au fur et à mesure que la convergence ralentit. Lorsqu'on regarde l'équation 8.26, on s'aperçoit qu'elle a exactement la même forme que l'équation initiale, au terme τ_H près. L'interprétation admise aujourd'hui représente ce terme comme une correction permettant de modifier le problème sur Ω_H de telle sorte que sa solution converge vers celle du problème L_h sur Ω_h . On pourrait faire une vague analogie avec les contraintes de Reynolds. La principale différence réside dans le fait que l'on dispose des estimations de ϕ_h sur Ω_h pour calculer τ_H . Alors que la prise en compte des contraintes de Reynolds dans les calculs nécessite qu'elles soient modélisées comme nous l'avons précédemment.

Cette vision a amené à considérer différemment le rôle du maillage fin. A l'origine Ω_h est considéré comme le maillage sur lequel le problème doit être résolu. Son nouveau rôle est de fournir une bonne estimation de τ_H , c'est à dire calculer les termes supplémentaires apparus avec le transfert du problème sur le maillage lâche. L'autre conséquence de cette manière de voir est que les calculs commencent par le maillage le plus lâche. L'interpolation successive des résultats permet ensuite le calcul de τ_H . Ce procédé est connu sous le nom de FMG (Full Multigrid). Il présente l'avantage de permettre une meilleure initialisation pour l'algorithme de Newton-Raphson.

τ_H permet également d'estimer l'erreur de troncature pour L_h . Ceci peut servir soit comme critère d'arrêt, permettant ainsi d'éviter la réduction du résidu en deçà de cette erreur, dans la mesure où cela n'a aucun intérêt. Une autre utilisation de cette estimation, consiste à corriger l'équation sur le maillage fin, cette fois par rapport à

l'équation continue. Ce procédé pour les équations différentielles est connu sous le nom d'extrapolation de Richardson [24].

La dernière application que nous évoquerons concerne les méthodes adaptatives. Une bonne estimation de l'erreur de troncature permet en effet de disposer d'un critère pour le raffinement du maillage.

L'approximation de Galerkin pour l'opérateur L_H convient parfaitement au problèmes linéaires dans la mesure où L_H ne dépend pas de ϕ_H . Les itérations sur les maillages lâches sont complètement découplées du maillage fin. La difficulté pour les problèmes non-linéaires est qu'à chaque itération l'opérateur change. Et, pour le réévaluer, il faut revenir au maillage fin par interpolation avant de repartir au maillage lâche par projection. Pour éviter ces aller-retour, l'algorithme de base consiste à rediscrétiser l'opérateur continu sur le maillage Ω_H . Cette méthode marche si l'évolution spatiale des coefficients de l'opérateur linéarisé reste faible. Dans le cas contraire, ce choix peut fortement contrarier le processus de convergence. Il n'existe pas pour le moment de technique générale pour calculer L_H . Les solutions proposées jusqu'ici procèdent par décomposition de l'opérateur non-linéaire en deux parties. La plus délicate est évaluée une fois pour toutes sur le maillage fin avant projection. Alors que l'autre est réévaluée directement sur le maillage lâche. On crée ici un découplage entre les termes de l'équation, opération qui aura forcément des conséquences sur la vitesse de convergence.

8.4 Application aux équations de Navier-Stokes.

Nous allons dans cette partie examiner les modifications nécessaires au traitement des équations de Navier-stokes, et notamment le traitement de la contrainte de continuité pour les écoulements incompressibles. Pour cela on repart des équations de correction suivantes:

$$a_p^u \cdot \delta u_p^h + \sum_{q \in \text{Vois}} a_q^u \cdot \delta u_q^h + \delta y_p^h \cdot (\delta p_p^h - \delta p_w^h) = -Res_h^u \quad (8.27)$$

$$a_p^v \cdot \delta v_p^h + \sum_{q \in \text{Vois}} a_q^v \cdot \delta v_q^h + \delta x_p^h \cdot (\delta p_p^h - \delta p_s^h) = -Res_h^v \quad (8.28)$$

$$\delta y_p^h \cdot (\delta u_E^h - \delta u_P^h) + \delta x_p^h \cdot (\delta v_N^h - \delta v_P^h) = -Res_h^p \quad (8.29)$$

On définit suivant les notations précédentes:

$$\tau_H^u = Res_H^u - R_H \cdot Res_h^u \quad (8.30)$$

$$\tau_H^v = Res_H^v - R_H \cdot Res_h^v \quad (8.31)$$

$$\tau_H^p = Res_H^p - R_H \cdot Res_h^p \quad (8.32)$$

et

$$Rhs_H^u = \tau_H^u + f_H^u \quad (8.33)$$

$$Rhs_H^v = \tau_H^v + f_H^v \quad (8.34)$$

$$Rhs_H^p = \tau_H^p + f_H^p \quad (8.35)$$

Les f_H^u , f_H^v et f_H^p sont nuls pour le premier maillage lâche, celui qui vient juste après le maillage principal. Lors du passage aux maillages suivants, Rhs_H joue le rôle de f_H . On peut voir les choses d'une autre manière: Supposons que pour le problème initial continue 6.1, f est une fonction uniformément nulle. Le problème défini sur le maillage fin est donné par:

$$L_h(\phi_h) = 0 \quad (8.36)$$

Sur le premier maillage lâche il est donné par:

$$L_H(v_H) = Rhs_H \quad (8.37)$$

On voit apparaître le premier second membre non nul. Pour les autres maillages lâche, f_H sera non nul et proviendra des Rhs_H successifs.

Les équations sur un maillage lâche quelconque sont données par:

$$a_p^u \cdot \delta u_P^H + \sum_{q \in \text{Vois}} a_q^u \cdot \delta u_q^H + \delta y_P^H \cdot (\delta p_P^H - \delta p_W^H) = Rhs_H^u - Res_H^u \quad (8.38)$$

$$a_p^v \cdot \delta v_P^H + \sum_{q \in \text{Vois}} a_q^v \cdot \delta v_q^H + \delta x_P^H \cdot (\delta p_P^H - \delta p_S^H) = Rhs_H^v - Res_H^v \quad (8.39)$$

$$\delta y_P^H \cdot (\delta u_E^H - \delta u_P^H) + \delta x_P^H \cdot (\delta v_N^H - \delta v_P^H) = Rhs_H^p - Res_H^p \quad (8.40)$$

Examinons à présent le cas particulier de l'équation de continuité. L'un des avantages de la méthode multigrille vient de la possibilité de traiter efficacement les contraintes globales comme celle de la continuité pour les fluides isovolumes. En effet, le fait d'imposer un calcul direct sur le maillage le plus lâche permet de traiter ce type de contrainte pour un coût limité. Il faut cependant adapter les procédures de transfert du problème au respect de cette contrainte. Pour cela, il faut annuler le terme de correction.

Le premier aspect de ce problème concerne la projection des résidus. La projection du résidu de l'équation de continuité se fait en prenant la somme des résidus de chacun des volumes élémentaires du maillage fin, (formule 8.11 et figure 8.4).

Le second aspect concerne la projection des composantes de vitesse. Les composantes de vitesse sur les faces Nord et Sud, sont calculées suivant la formule 8.10. Les composantes de vitesse sur les faces Est et Ouest restent inchangées. Le calcul du flux de matière sur cette face à partir de cette composante donne la somme des flux sur les faces correspondantes du maillage fin. A l'arrivée, le résidu projeté est égal au résidu sur le maillage lâche. Ainsi τ_H^p sera toujours nul.

8.5 Le cas des modèles de turbulence.

Dans le domaine de la résolution multigrille des équations de Navier-Stokes couplées à un modèle de turbulence, les références sont encore peu nombreuses. Nous allons en étudier quelques unes dans ce qui va suivre;

Le premier article est de Peter Johansson [36]. Cet auteur a calculé trois configurations:

1. marche descendante (figure 8.6)
2. cavité ventilée 2-d
3. cavité ventilée 3-d

Chaque configuration est traitée dans le cas laminaire et turbulent. Dans le cas laminaire, deux schémas de discrétisation sont étudiés:

1. hybride
2. QUICK couplé à un limiteur de flux de Van-leer

La méthode itérative utilisée est la méthode SIMPLEC avec le schéma F.A.S pour la partie multigrille. Le modèle turbulent est bi-couche. Dans la couche externe on utilise un $k - \epsilon$ fort Reynolds. Près de la paroi, il résout l'équation pour k avec un modèle de longueur de mélange. Pour le transfert multigrille des corrections de k et ϵ , il utilise la méthode proposée par Lien. [39]. Les résultats montrent un gain significatif en laminaire dès que la taille du maillage augmente. Ces résultats sont confirmés dans le cas turbulent où il obtient des facteurs d'accélération supérieurs à 100.

Le second article est de Lien [39]. Les configurations calculées sont:

1. un élargissement continu
2. un rétrécissement en forme de bosse (figure 8.7)
3. une marche descendante. (figure 8.7)

Les différents schémas de discrétisation utilisés sont:

1. hybride
2. QUICK
3. MUSCL

Comme algorithme itératif, il a utilisé la méthode SIMPLE. Il a également testé trois modélisations turbulentes:

1. $k - \epsilon$ forts Reynolds
2. $k - \epsilon$ faibles Reynolds
3. Reynold-stress

Comme transfert multigrille il utilise le schéma F.A.S. en apportant quelques modifications dans le calcul de L_H en ce qui concerne les termes convectifs. Les résultats en laminaire de Lien confirment ceux de Johanson (figure 8.6). Par contre ses résultats en turbulents, s'ils aboutissent également à une amélioration par rapport au calcul monogrille (de l'ordre de 3 à 5 fois), montrent une nette dégradation des performances de la méthode par rapport au laminaire.

Enfin le troisième article est de X.S. Bai [4]. Dans cet article une seule configuration est étudiée, c'est celle d'une chambre de combustion de turbine à gaz avec modélisation de la combustion. La discrétisation est faite par un schéma upwind d'ordre 1, et la résolution par l'algorithme SIMPLE. Le modèle de turbulence est de type $k - \epsilon$ avec des fonctions de parois. Le multigrille est de type F.A.S comme pour les autres. Bai signale la même dégradation des performances entre les cas laminaire et turbulent que Lien. Johanson est donc le seul à obtenir des résultats uniformes dans les deux cas. L'une des explications possibles de cette différence réside peut être dans l'utilisation d'un modèle de longueur de mélange.

8.6 Conclusion

Les méthodes multigrilles sont à classer dans la catégorie des techniques d'accélération pour méthodes itératives. L'expérience a démontré que les performances de ces méthodes dépendent de la densité du maillage. La technique d'accélération consiste à séparer la solution en composantes. On peut alors montrer que les méthodes permettant une détermination optimale de chacune des composantes nécessitent également une adaptation du maillage. Les composantes lentes (basse fréquence) nécessitent une méthode globale sur un maillage lâche. Les composantes fluctuantes (haute fréquence) nécessitent des méthodes moins globales mais des maillages plus fins. La technique multigrille donne des opérateurs pour assurer le transport du problème d'un maillage fin vers un autre plus lâche et inversement le transport de la solution d'un maillage lâche vers un maillage fin.

Selon que la solution sera ou non riche en composantes hautes fréquences, les opérateurs de transport peuvent ou non introduire des distorsions dans la représentation du problème sur le maillage lâche. Le niveau de stabilité de la méthode itérative peut également influencer celui du processus multigrille. La méthode multigrille a un niveau de complexité qui renforce sa robustesse. Cette robustesse peut masquer des erreurs de programmation importantes tout en permettant la convergence des calculs avec une dégradation des performances difficile à évaluer.

Partie IV

Architecture du code



Chapitre 9

Architecture du code

9.1 Introduction

Dans ce travail, nous nous sommes fixé au départ un cahier des charges relativement souple sur certains points importants du problème à résoudre (algorithme de résolution, structuration des données). L'analyse a montré que nous aurions à résoudre un problème à plusieurs inconnues: hydrodynamique, turbulence et électrochimie. Nous savions également que nous aurions à traiter des maillages relativement denses afin de pouvoir décrire avec précision les phénomènes survenant près de la paroi. Nous aurions pu nous contenter de résoudre notre problème sur un domaine rectangulaire simple. Nous avons pensé que le code n'aurait d'intérêt et donc une chance de survie que si nous lui donnions une certaine généralité. Nous avons décidé d'implémenter le traitement de géométries constituées de réunions de rectangles. On montrera un peu plus loin ce qu'il est possible de faire de ce point de vue avec très peu d'efforts supplémentaires.

A l'arrivée, nous avons un code qui dispose en plus des algorithmes de résolution suivant:

1. La méthode SIMPLER.
2. La méthode par zones avec différentes numérotations pour l'hydrodynamique.
3. La méthode par balayage avec algorithme tridiagonal de Thomas.
4. La méthode multigrille pour toutes les inconnues et tous les algorithmes précédents.
Le multigrille peut être utilisé dans sa forme complète ou comme une méthode d'amorçage de calculs délicats (modèles de turbulence).

La mise en oeuvre des solutions logicielles pose des problèmes similaires à ceux rencontrés dans d'autres domaines technologiques. La conception, la réalisation, la maintenance et l'adaptation de ces solutions à un nouveau contexte constituent les principaux points de ce travail. La gestion de ces différentes tâches est rendue particulièrement difficile par l'évolution de l'environnement informatique. Cette évolution

concerne les langages de programmation, les environnements de développement, mais surtout la puissance et l'architecture du matériel.

Une fois conçue, une solution logicielle nécessite de fréquentes adaptations à l'évolution des problèmes physiques à résoudre. Cette adaptation sera plus ou moins facile selon la manière dont le code initial a été architecturé. Il semble acquis aujourd'hui que la décomposition d'un système en blocs aussi indépendant que possible, est une bonne solution pour gérer les différentes tâches que nous avons évoqué plus haut. En effet un traitement composant par composant est alors grandement facilité. C'est donc sur la base de quelques critères liés à l'architecture que nous allons comparer notre solution avec celle du TEACHT qui est à l'origine de notre travail.

9.2 La solution TEACHT

9.2.1 Présentation

Architecture

Le code sur lequel nous avons commencé nos expérimentations est constitué d'un fichier unique. Dans ce fichier, on a le programme principal et des sous-programmes. Le domaine de calcul et le maillage sont décrits dans le programme principal. Les sous-programmes sont séparés en trois catégories permettant respectivement: -le calcul des propriétés du fluide, des résidus et des coefficients de linéarisation - la sortie des résultats. On a donc un seul grand fichier dont le parcours n'est pas aisé.

Structure de données

En fait il n'y a pas de structure de données en tant que telle. Les différents tableaux et les différentes variables sont déclarés en bloc dans le programme principal et transmis aux autres modules.

9.2.2 Critique du TEACHT

Examinons à présent le TEACHT par rapport à quelques modifications de base susceptible d'être faite sur un code de ce type.

Changement de configuration

Pour traiter le type de configuration que nous avons choisi, on a deux solutions en partant du TEACHT: la première consiste à englober le domaine de calcul dans un rectangle, puis à désactiver les points extérieurs. On peut alors utiliser la même structure de données que TEACHT, constituées de tableaux à deux dimensions pour le stockage des différents paramètres de calcul. Par contre, chaque frontière réelle va nécessiter quelques lignes de code pour son traitement. Cette opération devra être multipliée par le nombre de variables. Il faudra également augmenter la partie

traitant du maillage, de manière à permettre une repartition adéquate des points. Il faudrait augmenter le nombre d'indices pour décrire d'éventuels obstacles dans l'écoulement. Dans la deuxième solution, on adopte une représentation par sous-domaines. Alors il faut prévoir plus de frontières à traiter que dans le cas précédent. Il faut également prévoir la multiplication du nombre de tableaux pour représenter les différents sous-domaines. Ce qui va tout de suite alourdir considérablement la partie déclaration des variables.

Ajout de nouvelles variables

L'ajout de nouvelles variables pose les mêmes problèmes que précédemment. Il faut déclarer les tableaux pour les nouvelles variables dans tous les sous-programmes. Le traitement des frontières doit également être modifié.

Test d'autres algorithmes de résolution

Le code repose sur l'algorithme tridiagonal de Thomas pour la résolution de toutes les équations. Les changements nécessaires pour l'implémentation d'une méthode par zones ou d'un algorithme multigrille sont relativement importants. Prenons le cas du calcul par zones. Cette méthode nécessite un traitement partiel des points du domaine. Alors que le traitement des points se fait d'une manière globale dans le TEACHT, aussi bien pour le calcul des résidus, que pour la linéarisation et le traitement des frontières. La mise en oeuvre d'une méthode par zones passe par une plus grande différenciation du traitement des points du domaine et donc l'ajout de modules de bas niveau pour chacune des opérations citées plus haut. Il faut ensuite trouver une méthode d'inversion mieux adaptée à la résolution des systèmes tridiagonaux par blocs, ou à des systèmes plus généraux.

L'adaptation du TEACHT pour le calcul multigrille pose des problèmes apparemment plus simples. Il suffit en effet de multiplier la structure de donnée initiale, autant de fois qu'on a de niveaux de maillage. Or cette multiplication va alourdir le code pour tout changement de taille de tableau par exemple.

9.2.3 conclusion

Il est toujours possible de modifier le TEACHT pour l'adapter à de nouvelle configuration, pour y ajouter des inconnues ou pour tester de nouveau algorithmes. Ce travail est cependant toujours plus simple pour un code dont l'architecture est suffisamment souple. Le TEACHT est un code conçu pour les besoins de l'enseignement, ses qualités en matières d'architecture sont très limitées. Nous avons testé plusieurs algorithmes de résolution dont l'algorithme multigrille, l'un des plus exigeant en terme d'architecture. La nécessité d'homogénéiser la présentation des données pour les différents algorithmes nous a donc amené à concevoir une architecture complètement différente. Dans ce travail, nous avons privilégié une organisation hiérarchique aussi bien pour le code que pour les données. Pour cela nous avons adopté le langage 'C'

dont la richesse autorise des choix difficiles à mettre en oeuvre en fortran77 avant l'arrivée du fortran 90. Il s'agit aussi bien de l'allocation dynamique de la mémoire que du regroupement des données sous forme de structures. D'une manière générale, la richesse d'un langage détermine fortement la qualité des projets dont il participe à la réalisation. De ce point de vue les langages objets ont une génération d'avance sur le C et à fortiori sur le fortran77. Les nouvelles fonctionnalités du fortran90 et fortran95 incluent les concepts de la programmation objet. Si les comités de normalisation vont jusqu'au bout de leur démarche, ce langage a vocation à prendre en compte d'une manière plus précise les besoins spécifiques au calcul scientifique, lui redonnant ainsi une place de choix en tant que langage de haut niveau.

9.3 Présentation générale du code

L'architecture que nous avons implémentée s'appuie sur une structure de donnée hiérarchique qui facilite les opérations de séparation et de regroupement des données à tous les niveaux selon les besoins du type d'opérations à mettre en oeuvre. Le code peut être subdivisé en trois parties (figure 9.1): la première regroupe la lecture des données du problème ainsi que l'initialisation de la structure de données du code. La seconde partie constitue la résolution du problème proprement dite. Le troisième permet la sortie des résultats. Avant de décrire brièvement les différentes parties du code, nous allons d'abord décrire la représentation externe des données du problème. Pour illustrer cette partie, nous avons réalisé une série d'organigrammes représentant aussi bien les parties importantes du code que sa structure hiérarchique. Ces illustrations correspondent aux figures 9.1 à 9.15.

9.3.1 Définition du problème

Dans le but d'assurer la plus grande indépendance du code par rapport au type de configuration à traiter, nous en avons retiré la partie description de configuration ainsi que la partie maillage et spécification des conditions aux limites (figure 9.5).

Le domaine est décrit dans un fichier où l'on retrouve la spécification des différents rectangles qui le constituent.

Le maillage conçu lui aussi à l'extérieur du code est décrit par deux fichiers pour chacune des directions x et y .

Les conditions aux limites sont elles aussi définies à l'extérieur du code. Elles sont décrites dans un fichier principal qui peut selon le cas renvoyer à d'autres fichiers.

Un dernier fichier sert à spécifier les conditions de calcul. Il s'agit de fixer le choix entre différentes options de calcul (Monogrille ou Multigrille), de spécifier les coefficients de relaxation, et les critères d'arrêt. Les paramètres physiques tels que la densité et la viscosité sont également donnés dans ce fichier.

9.3.2 Organisation des calculs

Le code est composé d'une cinquantaine de fichiers écrits en langage 'C'. Le nombre de sous-programme est encore plus important. Le niveau de décomposition du code a été dicté principalement par un souci de lisibilité.

Organisation générale du code

Dans cette partie, nous donnons une description succincte du code. Le but est de faire apparaître les grands ensembles constituant ce dernier. Cette description peut se faire en considérant quatre niveaux.

Le **premier niveau** (figure 9.1) permet d'accomplir les tâches d'initialisation, de lancement des calculs, et de sortie des résultats. Il commence par la lecture des fichiers décrivant le problème, et les options de calcul retenues. On initialise ensuite le maillage le plus fin. Cette phase comprend aussi bien les tâches de mise à jour du maillage, d'allocation mémoire pour les structures et les données, que l'initialisation de certains champs de données. Cette initialisation concerne en grande partie les paramètres permettant de préciser les conditions aux limites aussi bien pour la description du domaine que pour les inconnues à déterminer. On procède à l'initialisation des autres niveaux de maillage dans le cas d'un calcul multigrille, en reprenant pour l'essentiel les mêmes modules que ceux utilisés pour l'initialisation du maillage fin. Après cette initialisation on peut lancer les calculs et sortir les résultats.

Le **second niveau** (figure 9.2) de description du code permet, pour le calcul, le choix entre trois options. La première permet le calcul sur un seul niveau de maillage. La seconde option permet une initialisation multigrille pour le calcul d'écoulements turbulents. La troisième permet de faire un calcul pleinement multigrille.

Le **troisième niveau** (figure 9.3) du code permet le choix entre une procédure de calcul par sous-domaines découplés, et une procédure avec assemblage préalable des équations provenant de plusieurs sous-domaines différents.

Le **quatrième** (figure 9.4) et provisoirement dernier **niveau** permet de choisir un algorithme pour la résolution des différentes équations. La résolution des équations du modèle hydrodynamique peut se faire par deux algorithmes différents. L'algorithme semi-explicite SIMPLER, et l'algorithme implicite par zone. Ce dernier algorithme offre plusieurs variantes sensiblement différentes les unes des autres. Nous avons prévu deux procédures de lissage différentes pour cet algorithme. Nous avons également prévu différentes numérotations possibles pour les inconnues u , v et p . La résolution des équations pour un scalaire quelconque se fait par une procédure de balayage par ligne, Avec pour chaque ligne, l'inversion d'un système tridiagonal par l'algorithme de Thomas de la même manière que pour le code TEACHT.

9.3.3 Sortie des résultats.

La sortie des résultats est de toutes les parties, celle qui dépend le plus de l'environnement informatique et en l'occurrence, des logiciels graphiques disponibles. Les modules de sortie actuels respectent la structure de données en sous-domaines. Cette organisation est d'autant plus intéressante que sa prise en compte est prévue dans certains logiciels graphiques. (C'est le cas notamment de Tecplot).

Notre expérience nous amène à penser que la représentation graphique ne doit pas seulement servir à visualiser les résultats finaux. Les mêmes outils peuvent constituer une aide puissante pour la mise au point du code. Pour cela, il faut que le passage de l'un à l'autre soit aussi facile que possible.

9.4 Structure et manipulation des données dans le code

La conception d'une architecture souple et ouverte à différentes évolutions s'appuie sur un nombre important de modules relativement découplés. Ces modules utilisent les uns et les autres des données différentes. Le but d'une structure de données est d'assurer la logistique au sein du code de telle sorte que chaque module puisse accéder facilement aux données qu'il doit traiter.

9.4.1 Organisation générale des données

Comme nous l'avons annoncé, nous avons adopté une représentation hiérarchique des données afin de faciliter la décomposition du code en modules aussi indépendants que possible.

La structure de base

Cette structure après initialisation contient toutes les informations nécessaires au fonctionnement du code (figure 9.12). Ces informations peuvent être séparées en trois grandes catégories: Les informations sur les différents niveaux de maillage, les informations pour l'inversion des équations et les informations sur le contexte en cours de traitement, à savoir le niveau de maillage, le rectangle, l'inconnue.

Les niveaux de maillage

Chaque structure de ce type contient une description complète du problème du maillage et des conditions aux limites permettant un calcul monogrid (figure 9.13).

Les rectangles

Contient toutes les informations pour le calcul des résidus dans un sous-domaine avec prise en compte des conditions aux limites (figure 9.14).

Les inconnues

Chaque inconnue du problème a deux types de structure qui lui sont dédiées (figure 9.15). La première contient tous les tableaux pour le stockage des valeurs calculées, des résidus, des coefficients du système linéaire etc. Un deuxième type de structure contient toutes les informations sur les conditions aux limites.

9.4.2 Détails de la description du domaine

Pour implémenter le calcul des résidus tel que nous l'avons décrit, nous avons conçu une structure de donnée permettant la description individuelle des points de chaque rectangle. Cette description s'organise sur deux niveaux: Un premier permet de situer les points frontières parmi les points du domaine. Le second niveau permet de dire à quel type de frontière appartient un point donné. Pour cela on a deux types de structure, un premier pour le champ et un second pour les quatres frontières.

9.4.3 Détails des paramètres de linéarisation des équations

Ces paramètres sont propres à chaque inconnue du problème et permettent notamment de gérer les problèmes d'allocation mémoire. Cette organisation permet également l'implémentation de procédure de linéarisation automatique.

9.4.4 Accès aux données sur les conditions aux limites

Lorsque l'on calcule le résidus d'une équation pour un point du maillage, on a besoin de savoir si le point est situé sur la frontière ou s'il est à l'intérieur du domaine. S'il est sur la frontière, on a besoin de savoir de quel type de frontière il s'agit. Chacune de ces informations est accessible à l'aide d'une fonction qui prend en paramètre, le rectangle auquel appartient le point, ainsi que la position (i, j) du point.

9.4.5 Accès aux données du maillage

Le calcul des résidus fait également intervenir le maillage. Dans notre cas, l'implémentation du maillage décalé implique la création de quatre vecteurs de maillage différents: Deux suivant l'abscisse x et xu et deux suivant l'ordonnée y et yv . Chacun de ces vecteurs est accessible par une fonction qui prend en entrée le rectangle correspondant ainsi qu'un entier spécifiant l'un des quatres vecteurs. Cette fonction renvoie un tableau à une dimension dont les éléments sont accessible comme pour un tableau normal.

9.4.6 Accès aux champs de valeurs

Tel que le code est conçu, les seuls champs de valeurs nécessaires pour le calcul des résidus sont constitués des inconnues du problème (u , v etc..) et des inconnues auxiliaires tels que la viscosité turbulente ν_t . Comme pour les maillages, chacune de ces variables est attachée à un entier. Un sous programme permet cette fois de récupérer un tableau à deux entrées auquel on peut affecter un nom arbitraire. Ce sous-programme prend en entrée le rectangle courant ainsi que l'entier correspondant à la variable dont on veut accéder aux valeurs.

9.5 Description et lectures des données

Les modules de lecture sont tous regroupés dans un seul fichier. La logique de cette partie est facile à saisir. La manipulation des données dans cette partie se fait au niveau le plus bas de la structure. C'est la raison pour laquelle nous avons fait le choix de la rendre aussi indépendant que possible des équations à résoudre.

Les fichiers de données sont organisés de telle sorte que la majeure partie des paramètres soient lu de l'extérieur. Ces fichiers sont organisés sur trois niveaux (figure 9.5). Le premier permet le choix d'une configuration. Nous en avons défini quatre pour l'instant: La marche, le canal, la cavité et la dernière représente un écoulement dans un canal avec une valve anti-retour. Chacun d'eux possède un répertoire contenant les données nécessaires au calcul. Le premier niveau est constitué par un fichier `main.dat` contenant une ligne pour l'accès à l'un des répertoires précédent. Pour le niveau deux, chacun de ces répertoires contient un autre fichier `main.dat`. Ce dernier fournit quelques paramètres de calcul ainsi que les noms des autres fichiers contenant notamment le maillage et les conditions aux limites.

A côté du fichier `main.dat`, dans chaque répertoire, il y a un autre fichier important pour la résolution qui contient les paramètres physiques du problème, viscosité cinématique, densité massique, coefficients de diffusion des ions Zn etc. Ce fichier contient les coefficients de relaxation pour les différentes variables, ainsi que les critères d'arrêt. Le nombre de niveaux pour une résolution multigrille est calculé automatiquement en fonction du maillage le plus fin donné dans les fichiers.

9.5.1 Le fichier de description du domaine

Ce fichier contient une liste de paramètres. Ces paramètres décrivent la liste des rectangles qui constituent le domaine d'écoulement. La représentation choisie part du principe que tout domaine constitué de rectangles peut être subdivisé en un tableau de rectangles. On peut voir l'exemple de la marche descendante sur la figure 9.1. L'ensemble des rectangles peut alors être représenté par un tableau à double entrée une nrx suivant x et l'autre nry suivant y . Chaque case de ce tableau est un rectangle qui est soit un obstacle soit un domaine d'écoulement. Cette représentation a été adoptée pour ses avantages dans l'uniformisation du traitement des frontières.

En effet dans ce cas, on peut se concentrer sur la prise en compte de différents cas de figure pour un seul rectangle. Ce traitement peut alors être réutilisé pour chacun des rectangles du domaine. Nous avons pour l'instant constitué une liste de quatre cas de figure applicable à chaque côté d'un rectangle. Deux cas de conditions aux limites de Dirichlet, une pour les parois imperméables et une pour les entrées avec profil de vitesse spécifié non-nul. Le troisième est constitué par les conditions de Neumann. Et le quatrième par les frontières internes. D'autres cas de figures tels que les conditions de symétrie pourront être ajoutés plus tard. Les différents modules traitant des conditions aux limites sont prévus pour reconnaître ces types de frontières. Afin que ces modules puissent différencier les points frontières des autres, nous avons adopté l'organisation suivante: Chaque point de maillage est représenté par un entier qui prend les valeurs 0, 1, 2, 3 ou 4 suivant qu'on est hors frontière, au sud, à l'ouest, au nord ou à l'est. Dans chaque frontière, Chaque point est représenté par un entier qui prend la valeur 0, 1, 2 ou 3 suivant que la frontière est interne, de type Dirichlet entrée de fluide, Dirichlet paroi imperméable ou de type Neumann. Cette organisation permet de différencier individuellement les points et facilite le traitement des conditions aux limites. Les coûts mémoire supplémentaires sont faibles par rapport à la flexibilité apportée au code.

9.5.2 Fichier de description du maillage

La représentation choisie pour le domaine oriente la construction du maillage. Dans la mesure où le domaine est représenté par un tableau à deux entrées, le maillage sera subdivisé en autant de segments qu'il y a d'entrée dans chaque direction. Ce qui donne nrx segments suivant x et nry segments suivant y . Chacun de ces segments est maillé séparément par un programme extérieur qui produit un fichier pour chaque direction. Ces fichiers contiennent le nombre de segments, et pour chaque segment, le nombre et la liste des pas de maillage. La liaison entre rectangles se fait ensuite pour assurer la continuité du maillage au niveau des interfaces.

9.5.3 Fichier de description des conditions aux limites

Ce fichier est prévu pour la spécification de conditions aux limites non nulles dans le cas de Dirichlet. Les valeurs sont alors calculées par un autre code. C'est le cas du code de calcul monodimensionnel permettant de calculer les conditions aux limites pour les écoulements turbulents. On peut également y fixer une valeur constante pour toute une frontière. Dans tous les cas il faut spécifier les coordonnées du rectangle, le côté concerné. Selon le cas, il faudra également donner soit un fichier contenant les valeurs à lire lorsqu'il y en a plusieurs, soit directement la valeur lorsqu'il y en a une seule.

9.6 Modules importants

Le nombre important de sous-programmes nous a amené pour des besoins de clarté à les regrouper en modules. Ce mode de regroupement va nous permettre de décrire les plus importants d'entre-eux à travers leur fonction et les données qu'il manipulent.

9.6.1 Calcul des résidus

Les modules de calcul du résidu occupent deux fichiers par variable. Un premier fichier pour le traitement des points internes (figure 9.6), et un fichier pour le traitement des points situés aux limites du domaine (figure 9.7). Dans le traitement des points internes, les formulations des termes convectif et diffusif et le traitement du terme source sont regroupé dans un même sous programme. Les traitements aux frontières sont subdivisés en cinq sous-programmes: un pour le contrôle, et quatre pour chacune des directions Est, Ouest, Nord, Sud.

9.6.2 Linéarisation

La linéarisation se fait de deux manières (figure 9.8): soit en conservant les coefficients "Ap, Aw, etc..." calculés lors de la discrétisation des équations, soit en calculant un taux de variation du résidu pour un calcul approché des termes du Jacobien selon la méthode de Newton. Ces deux modules sont complétés par des modules de réservation et de libération de place mémoire pour le stockage des coefficients calculés.

9.6.3 Résolution des équations de Navier-Stokes par une méthode par zones

Les modules de résolution par zones comprennent pour l'initialisation, le calcul des limites des différentes zones à partir des pas de zone. Il y a ensuite les modules de parcours des zones. Le rôle de ces derniers modules est de fixer les paramètres propres à chaque zone avant l'appel de modules plus généraux tels que le calcul de résidus, du Jacobien et l'inversion.

Délimitation des zones

Pour chaque maillage on utilise la taille de zone donnée pour calculer les indices délimitant les zones. Pour un rectangle donné, chaque direction peut être subdivisée en plusieurs segments. On obtient ainsi un pavage de rectangle avec ou sans recouvrement. Pour le calcul par zone avec assemblage préalable, les zones calculées pour chaque sous-domaines sont regroupées pour donner les zones du domaine global. Un soin particulier est apporté aux transitions entre zones, afin d'assurer le raccordement des sous domaines et éviter les problèmes de discontinuité liés au découplage.

Numérotation des points

La résolution d'équations en maillage structuré sur un rectangle ne pose pas de problèmes de numérotation majeure. Ce n'est plus le cas lorsqu'il s'agit de traiter des configurations rectangulaires complexes. Le fait d'introduire des obstacles dans le domaine peut rendre une méthode de numérotation simple extrêmement coûteuse. Cette méthode consiste à ne pas faire de différence entre les points du domaine d'écoulement et ceux contenus dans les obstacles. La différenciation des points se fait par une numérotation précise qui est utilisée dans l'algorithme d'inversion par bloc.

Inversion par blocs

L'inversion par blocs est constituée des modules suivants (figure 9.9 et 9.10):

- numérotation des points,
- lecture des coefficients,
- d'inversion de matrices carrées,
- opérations matricielles.
- opérations entre matrices et vecteurs.
- Cette partie est une implémentation de ce qui a été décrit dans le chapitre du calcul par zone (voir 7.7).

Décodage de la solution

Le décodage de la solution pour une résolution par blocs multivariables consiste à répartir les corrections correspondant à chaque variable.

9.6.4 Résolution des équations de Navier-Stokes par la méthode SIMPLER

La structure de données que nous avons adoptée a pour principal avantage de permettre deux types de représentation des inconnues. Le premier type permet de leur affecter des noms suffisamment expressifs pour tout ce qui concerne le calcul des équations discrétisées correspondantes. Le second type de représentation permet de leur affecter un nombre entier comme indice dans une liste d'inconnues. Cette deuxième représentation évite la personnalisation des déclarations de variables et facilite les tâches d'allocation de place mémoire. Ceci, aussi bien pour les différents tableaux de valeurs, résidu etc... que pour les coefficients du système linéaire. Cette représentation des données nous a permis de concevoir un algorithme de d'inversion par zones multivariable utilisable sans perte d'efficacité aussi bien pour les trois équations d'hydrodynamique que pour tout autre ensemble d'équations couplées. Même si l'algorithme SIMPLER a été inspiré par le problème du couplage des équations de l'hydrodynamique, nous considérons qu'il peut être appliqué à un ensemble quelconque d'équations couplées. L'implémentation de l'algorithme SIMPLER

s'est fait en essayant d'étendre le principe du découplage des variables. Les modules conçu l'ont été afin de traiter d'autres découplages que le couplage hydrodynamique. Cette conception a été testée pour le couplage k, ϵ sans succès. Les modules restent cependant tels quel pour d'autres tentatives éventuelles. Les principaux modules sont: l'inversion partielle pour la construction de l'équation pour la pression. Le calcul de résidu partiel pour le résidu de l'équation pour la pression, et le module de substitution inverse utilisé pour la correction des vitesses. Le principe général retenu est la possibilité de séparer des variables par une inversion partielle d'une équation et substitution du résultat dans une autre équation.

L'équation pour la pression

Suivant le principe retenu, la construction de l'équation pour la pression consiste à inverser partiellement l'équation pour u et l'équation pour v . Un même module effectue cette opération pour chacune des deux variables à tour de rôle. Chacune de ces variables est ainsi exprimée en fonction de la variable de pression. Un second module effectue la substitution du résultat précédent. Au lieu d'avoir un module par variable pour ces deux opérations, soit au total quatre modules, nous en avons deux. Ces deux modules peuvent être utilisés dans des cas impliquant d'autres variables sans modification majeure.

Résidus linéaires

La méthode SIMPLER nécessite le calcul de résidus pour le système linéaire à résoudre. On en distingue deux types. Le premier prend en compte tous les coefficients du système linéaire, le second est le résidu partiel d'une équation de quantité de mouvement.

Les corrections vitesse pression

La correction de vitesse se fait par l'opération inverse de celle de la construction de l'équation pour la pression.

9.6.5 Algorithme de Thomas pour système mono-variables

L'algorithme de Thomas est une reprise de l'algorithme classique. La seule modification faite permet le traitement des géométries comportant des obstacles sans avoir à calculer pour les points hors zone d'écoulement.

9.6.6 Projection et Interpolation

Les opérations de transfert sont de trois types: Ce sont les sous-programmes d'application, le calcul de correction et le transfert de résultats. Ces modules fixent les niveaux de maillage, le rectangle à traiter et le champ de données à traiter. Un deuxième

type de sous-programmes est constitué par la projection ligne par ligne ou colonne par colonne suivant le sens de subdivision. Un dernier permet le traitement d'une ligne ou d'une colonne en distinguant les variables du maillage décalé de celle du maillage centré.

9.7 Caractéristiques spécifiques du code

9.7.1 Le changement de configuration

La conséquence de nos choix pour la présentation des données est que le changement de configuration se fait sans recompilation du code. Il suffit de créer un nouveau répertoire et d'adapter les différents fichiers de configuration au nouveau cas d'écoulement. Certains contrôles de cohérence sont fait dans le code. Ils facilitent la vérification de la conformité des fichiers au format lu. Le changement de la taille des tableaux est commandé de l'extérieur par le changement de taille du maillage.

9.7.2 Ajout de variables

Le changement du nombre d'inconnues du problème est plus délicat. Dans un premier temps, il faut construire les fichiers de calcul des résidus sur le modèle que nous avons représenté sur la figure 9.7. Ensuite il faut donner au code le nouveau nombre de variables. L'étape suivante consiste à rajouter les nouvelles fonctions de calcul des résidus dans les fichiers où nous avons regroupé tous les appels de fonction pour le calcul des résidus. En résumé les seules lignes de code à écrire sont pour définir la discrétisation de la nouvelle équation. Il se peut que certains modèles tels que les modèles de turbulence nécessitent des inconnues annexes (u_τ , fonction de paroi etc ...) et des sous-programmes annexes pour les mettre à jour. Nous avons prévu la spécification d'un nombre d'inconnues auxiliaire, et le schéma que nous avons retenu pour l'implémentation du modèle $k-\epsilon$ peut servir de référence pour des modifications même dans la structure de données.

Il est à noter qu'aucune nouvelle déclaration de variable n'est effectuée directement lors de cette modification. Le changement du nombre de variables est directement pris en compte par la structure de données qui le répercute jusque dans la linéarisation des équations.

9.7.3 Ajout d'algorithmes de résolution

Le code que nous avons conçu dispose de plusieurs algorithmes de résolution. La structure de données a beaucoup facilité leur mise en oeuvre. nous pensons donc que de nouveaux algorithmes peuvent s'intégrer facilement au code actuel.

9.8 Conclusion

Le code que nous avons conçu dispose de fonctionnalités de loin plus importantes que celles du TEACHT. Sa complexité est aussi plus importante. Pour en faciliter la maintenance et l'évolution, nous avons parié sur deux principaux concepts de programmation moderne, à savoir le découplage des modules et l'encapsulation des données. La diversité des problèmes que l'on peut résoudre avec ce code est importante. Son potentiel d'évolution est lui aussi très important. Cependant, avant d'aller plus loin, il serait judicieux d'installer la version actuelle dans un environnement de développement susceptible d'en faciliter la manipulation et la documentation. Une interface graphique même sommaire pourrait aussi en faciliter l'utilisation, notamment pour la modification des fichiers de données.

Chapitre 10

Performances du code

Le code offre plusieurs possibilités pour la résolution des équations de l'hydrodynamique, qui n'ont pas pu être testées d'une manière exhaustive faute de temps.

Au niveau le plus bas, la résolution des équations du modèle hydrodynamique peut se faire par deux algorithmes différents. L'algorithme semi-explicite SIMPLER, et l'algorithme implicite par zones. Ce dernier algorithme offre plusieurs variantes sensiblement différentes les unes des autres. Nous lui avons adjoint deux procédures de lissage différentes. Nous avons également prévu différentes numérotations possibles pour les inconnues u , v et p . A un second niveau, nous avons prévu un choix entre une résolution par sous-domaines et une résolution avec assemblage préalable du domaine, couplé à l'un des algorithmes précédents. Et pour finir toutes les équations peuvent être résolues par une méthode multigrille: soit par initialisation multigrille, soit par un multigrille complet. Dans cette étude de performance, nous ne présenterons donc qu'une partie de ces possibilités. Les résultats seront séparés en deux parties: les écoulements laminaires et les écoulements turbulents.

10.1 Les écoulements laminaires

Nous avons testé deux cas de figures. L'écoulement de cavité constitue un cas test intéressant à cause de la recirculation. L'écoulement de marche présente deux caractéristiques différentes: une zone de recirculation et un comportement globalement convectif.

10.1.1 La cavité.

La figure 10.1 présente le domaine de calcul et les conditions aux limites retenues. Dans la première phase de notre travail, nous avons effectué une étude de l'influence de la taille des zones sur les performances de la méthode, à nombre de Reynolds et de points fixe (figure 10.2). Les calculs dans cette partie furent réalisés avec une méthode par zones simple avec coefficient de pénalité pour éviter les problèmes de

pivot nul. Sur cette figure on voit clairement qu'il existe un optimum pour le temps de calcul en fonction de la taille de bande. Deux phénomènes sont ici en compétition. L'équation de continuité requiert une taille de zone aussi grande que possible. Alors que la résolution des non-linéarités est surtout influencé par le nombre d'itération. Sur les figure 10.3 et 10.4, on les résultats de calculs similaires aux précédents en utilisant une méthode de lissage de type SIMPLER. Les résultats sont comparé à ceux obtenus avec l'algorithme SIMPLER. La méthode SIMPLER a des performances tout à fait comparables à celle de notre méthode en temps de calcul. En fait Braaten [7] a montré que pour cet écoulement, la méthode SIMPLER est meilleure que la méthode par zones avec coefficient de pénalité, et ceci a été confirmé dans notre étude.

10.1.2 La marche.

La figure 10.5 présente le domaine et les conditions aux frontières. La figure 10.6 présente des calculs fait avec une méthode par zones avec assemblage préalable et numérotation du type I (figure 7.8b) pour différentes tailles de zones. La méthode de pénalité est utilisée pour éviter le pivot nul. On voit que la méthode par zones est plus performante pour des tailles de zone faibles.

Ces résultats sont confirmés sur la figure 10.7 qui présente la convergence en fonction du temps CPU. Ici les calculs ont été effectué sans coefficient de pénalité et sans méthode de lissage avec une numérotation du type IV (figure 7.8c). Comme on peut le voir cette méthode est de loin plus efficace que la méthode SIMPLER. Sur les courbes de la figure 10.7, on observe deux phases: une première dans laquelle aucune taille de zone ne se détache des autres, et une deuxième phase dans laquelle la taille la plus faible prend l'avantage sur les autres. Sur la figure 10.8, la convergence est présentée en fonction du nombre d'itérations. On observe toujours les deux phases. Et cette fois toutes les tailles de zones se confondent dans la seconde phase qui correspond visiblement à la résolution des non-linéarités.

10.1.3 Effet du nombre de Reynolds.

Les calculs pour cette partie ont été réalisés avec des tailles de maillage relativement modestes afin de ne pas pénaliser la méthode SIMPLER. Pour le cas de la cavité, la taille de zone $z = 8$ a été choisi à cause de sa stabilité sur toute la gamme de nombre de Reynolds étudiées. Pour la marche, la taille la plus faible $z = 2$ n'a posé aucun problème de stabilité. Sur les figures 10.9 et 10.10, on a tracé l'évolution du temps de convergence en fonction du nombre de Reynolds, respectivement pour la cavité et pour la marche. On voit que pour la gamme de Reynolds considérée, le coût diminue quand le nombre de Reynolds augmente. La position relative des deux courbes sur chacune des figures montre que la méthode SIMPLER reste meilleure pour le cas de la cavité, alors que la méthode par zones est nettement meilleure pour l'écoulement de marche descendante. La légère inflexion de la courbe SIMPLER sur

la figure 10.10 est probablement due au poids croissant des non-linéarités pour des Reynolds importants. Alors que pour les faibles Reynolds; la conservation du débit est primordiale.

10.1.4 Organisation par sous-domaine et résolution par zone

Effet de la représentation des données sur le calcul de résidus et l'inversion des équations

Cette partie a pour but d'éclairer les antagonismes qui peuvent exister dans la représentation des données pour les deux opérations importantes que sont la construction des équations et leur résolution. Après plusieurs tentatives, nous avons adopté l'organisation en sous domaines pour la représentation du problème et la construction des équations. Lorsqu'il a fallu passer à la résolution de ces équations, notre premier réflexe fut d'implémenter des algorithmes de calcul par sous-domaines. Pour ces algorithmes, le seul problème était la transmission des informations d'un sous-domaine à l'autre.

La figure 10.11 montre la courbe de convergence d'un calcul par zones appliquée aux sous-domaines. L'autre courbe représente un calcul par zones avec assemblage préalable. Ces deux courbes correspondent à des calculs effectués sur un écoulement de marche descendante. Elles démontrent la nécessité de l'assemblage pour des géométries complexes. Cette procédure d'assemblage est plus complexe à mettre en oeuvre, mais les bénéfices sont importants. On peut par exemple résoudre sans problème un écoulement du type de celui dont la décomposition est présentée sur la figure 10.12a et le maillage sur la figure 10.12b.

10.1.5 Consommation en place mémoire pour les algorithmes SIMPLER et par zones

Nous avons essayé d'estimer le coût relatif des méthodes SIMPLER et par zones en terme d'occupation mémoire. Soit un domaine rectangulaire de dimension nx et ny . La méthode SIMPLER consomme pour la résolution:

$$M_{SIMPLER} = 27nx.ny \quad (10.1)$$

Dans cette formule, nous avons pris en compte la place mémoire occupée par les coefficients des équations discrétisée ainsi que celle occupée par l'équation pour la pression.

La méthode par zones consomme pour une taille de zone de m et ny :

$$M_{Zone} = 23nx.ny + 9m^2ny \quad (10.2)$$

Ici nous avons pris en compte la place mémoire occupée par les coefficients des équations discrétisées ainsi que celle occupée par les blocs hachurés de la figure 7.2b et les seconds membres correspondants.

La différence entre ces deux formules donne:

$$M_{S_Z} = 4nx.ny - 9m^2ny \quad (10.3)$$

Supposons que $nx = ny = n$. on obtient $4n^2 - 9m^2n$.

Pour $m = 2$ les deux méthodes ont le même coût pour $n \leq 9$, au delà la méthode SIMPLER devient plus coûteuse.

Pour $m = 8$, la limite se situe à $n = 154$.

Les valeurs croissent rapidement, mais comme on l'a vu plus haut, les écoulement du type canal plan ou marche descendante se calculent avec m de l'ordre de deux. Dans ce cas, la méthode par zones est moins coûteuse.

10.1.6 La méthode multigrille et la densité de maillage.

Dans la présentation de la méthode, nous avons annoncé que la méthode multigrille doit réorganiser le processus de résolution de telle sorte que le coût en temps de calcul, soit proportionnel au nombre de points. Nous avons tracé vérifié cette assertion pour deux cas d'écoulements.

La cavité

La figure 10.13 présente les résultats pour un écoulement de cavité à Reynolds 1000. On y compare les performance de trois méthodes: La méthode globale, une méthode par zone sur un niveau de maillage, puis une méthode multigrille. Pour la méthode multigrille la subdivision se fait dans une direction à la fois x ou y (voir 8.3.1). Le maillage le plus fin pour la cavité comporte huit niveaux qui correspondent à quatre niveaux dans le cas normal d'une subdivision simultanée suivant x et y . La figure 8.1 montre la méthode de subdivision choisie pour passer d'un niveau de maillage à l'autre. Rappelons que ce choix (unidirectionnel) a été dicté par la nécessité de serrer le maillage dans une direction pour les écoulements de Couette-Poiseuille. Sur la figure 10.13, on voit que la méthode par zones résiste mieux à l'augmentation du nombre de points que la méthode globale. Ces deux méthodes sont cependant surpassées par la méthode multigrille pour laquelle la courbe d'évolution est quasiment linéaire.

La marche descendante

La figure 10.14 concerne un écoulement de marche descendante à Reynolds 1000. Ici on a comparé une méthode par zones monogrille d'une part, et deux méthodes multigrille reposant l'une sur l'accélération de la méthode SIMPLER, et l'autre sur l'accélération d'une méthode par zones. Pour ces dernières méthodes, le nombre de niveaux pour le maillage le plus fin est de dix soit cinq dans le cas standard.

La méthode par zones monogrille résiste moins bien à l'augmentation du nombre de points de maillage. On voit également que l'accélération de la méthode par zones se comporte légèrement mieux que celle de la méthode SIMPLER. Cet avantage est une simple confirmation pour cette configuration de ce que l'on observe déjà dans le cas mono-grille. Les rapports d'accélération sont donnés dans le tableau 10.1. Ces rapports sont à comparer à ceux de la figure 8.7. Dans cette comparaison, il faut prendre en compte les performances de la méthode accélérée. En ce qui concerne Lien, la méthode accélérée est la méthode SIMPLER. Nous avons accéléré une méthode par zones qui est nettement plus performante dans le cas de la marche (facteur 10) que l'algorithme SIMPLER.

Nombre de points (cavité)	Temps CPU monogrille/Temps CPU multigrille
1024	1.3
4096	4.0

Nombre de points (marche)	Temps CPU monogrille/Temps CPU multigrille
640	2.7
2560	3.6
12240	9.8

Tableau 10.1

10.2 Les écoulements turbulents

10.2.1 Initialisation multigrille

Le principal avantage que nous avons tiré des méthodes multigrilles concerne l'amorçage du processus d'itération. En effet comme l'a remarqué K. Hannani [31] et avant lui Monnier [43] et Vanhée [64], le succès du processus de résolution dépend fortement de l'estimation initiale de la solution. Jusqu'ici, le remède proposée consistait à utiliser dans les première phase du processus des modèles ayant une convergence plus facile. Après quelques itérations avec un tel modèle, on passe au modèle à résoudre. Nous proposons une autre solution. Elle consiste à intégrer directement le modèle dans un schéma FMG (Full Multi-Grid). Cette procédure consiste à commencer la détermination de la solution par le maillage le plus lâche. Le calcul sur les maillage lâches semble avoir de bonnes propriétés de stabilité. On remonte alors progressivement jusqu'au maillage le plus fin. L'autre avantage de cette méthode, est qu'elle permet l'utilisation de méthodes peut couteuses. En effet, nous n'avons pas réussi à amorcer une méthode par zones par le modèle $k-l$ sur un maillage suffisamment fin. Les calculs fait par K. Hannani [31] l'ont été par une méthode d'inversion directe avec stockage sur disque. Les deux méthodes de calcul ont donc été comparées pour un nombre de point permettant une inversion directe en mémoire pour un écoulement de Poiseuille pur, soit (24×32) . La méthode multigrille accélère une

méthode par zones, et utilise trois niveaux de maillage. Les résultats sont représentés sur les figures 10.15 et 10.16.

10.2.2 Multigrille complet

Comme nous l'avons dit en fin de présentation de la méthode multigrille, le calcul d'écoulements par cette méthode nécessite des soins particuliers. Les deux principales opérations impliquées dans la méthode à savoir la projection et l'interpolation, sont des opérations approchées. Les conditions fixées pour le choix de ces opérateurs dépendent de l'ordre différentiel de l'équation à résoudre [9].

Plus cet ordre est important, plus les opérateurs de transfert que nous venons de citer doivent être précis. Le problème avec les modèles de turbulence est que la dominance des termes sources que nous avons déjà signalé pour la stabilité de convergence de la résolution, empêche une bonne estimation de cet ordre. Sur la figure 10.17a nous avons représenté le champ de valeurs de l'énergie turbulente pour un écoulement de type Couette. La forme de cette fonction n'est pas compatible avec une équation d'ordre deux.

Pour les opérateurs de transfert que nous avons utilisé et qui sont tous d'ordre deux, nous avons calculé l'erreur résultant d'une projection puis d'une interpolation. Sur le niveau trois (figure 10.17d), l'erreur est de l'ordre de vingt cinq pour cent ce qui est important. Nous n'avons pas étudié l'impact exact de cette erreur sur la stabilité du processus global. Nous pensons cependant qu'elle est certainement à l'origine de l'échec rencontré. La mise en oeuvre d'adaptations particulières consistant à séparer le traitement des différents termes des équations est tout à fait envisageable. Leur manque de généralité risque cependant de compliquer l'utilisation du code. Nous avons donc préféré pour l'instant nous contenter d'une initialisation multigrille dont l'apport est déjà considérable.

10.2.3 Conclusion

Sur la figure 10.15 on voit clairement l'avantage de la présente méthode sur l'ancienne malgré une utilisation partielle du principe de la méthode multigrille. La convergence en fonction du temps de calcul est clairement plus rapide. Sur la figure 10.16 où la convergence est représentées en fonction du nombre d'itération, on observe une évolution quasi parallèle des deux méthodes. Ceci peut permettre de confirmer une idée intuitive selon laquelle, le nombre d'itérations pour un problème non-linéaire, est au moins aussi important que la qualité des itérations. Les itérations de la méthode $k - L$ sont de meilleure qualité à cause de l'inversion globale adoptée pour la résolution des systèmes linéaires de la partie hydrodynamique. Cette qualité est d'ailleurs une garantie de robustesse. Les itérations de type par zones ou SIMPLER sont moins coûteuses mais aussi moins stables. En conclusion l'organisation multigrille permet de renforcer la stabilité de ces méthodes tout en maintenant un coût par itération incomparablement plus bas que celui de la méthode globale.

Partie V
Les Résultats.

Chapitre 11

Validation des résultats

Dans cette partie nous allons comparer les résultats obtenus avec le code développé à ceux d'autres auteurs où à des expériences dans un but de validation. Ce chapitre s'articule en trois parties: la validation du code laminaire, de la partie turbulente et d'une application à l'électrochimie.

11.1 Ecoulements Laminaires

Pour la validation du code en laminaire, nous avons choisi trois cas tests. Pour le premier (écoulement de canal plan établi) la solution exacte est connue. Pour le second cas test (cavité entraînée) il existe de nombreux calculs effectués par différents auteurs. Pour le troisième cas test (la marche descendante) nous avons comparé les longueurs de recollement pour lesquels beaucoup d'auteurs ont également effectué des calculs.

11.1.1 Canal plan

Nous avons calculé un écoulement de canal plan de hauteur $h = 1$ pour la configuration suivante: $-0.5 \leq y \leq 0.5$, $0 \leq x \leq 5$ (figure 11.1). Le canal est décomposé en trois sections comme c'est le cas pour celui utilisé pour les mesures à l'IMFL. Cette configuration permet de spécifier une vitesse non-nulle sur la paroi inférieure de la section centrale pour les écoulement de Couette-Poiseuille. La viscosité dynamique est $\mu = 0.5$. En entrée, on impose un profil de vitesse uniforme $u_0 = 1$, et donc le débit. Le profil dans la partie établie vérifier la relation:

$$u = \frac{1}{2 \cdot \mu} \cdot \frac{dp}{dx} \left(y^2 - \frac{h^2}{4} \right) \quad (11.1)$$

Nous n'avons pas imposé le gradient de pression. Nous avons simplement vérifié que la relation 11.1 était bien satisfaite dans la zone établie de l'écoulement. La figure 11.2 donne le profil de vitesse longitudinale correspondant. L'équation polynômiale

est obtenu par une méthode des moindres carrés. Le gradient de pression calculé du champ établi vaut $\frac{\partial p}{\partial x} = -5.98$ (la valeur exacte étant -6). La valeur maximale de la vitesse est $U_{max} = 1.495$ (la valeur exacte étant 1.5). Les coefficients calculés par regression sont donc en bon accord avec la formule 11.1.

11.1.2 Cavité entraînée

Nous avons ensuite testé le code sur un écoulement de cavité entraînée pour lequel on dispose de nombreux résultats de calculs faits par d'autres auteurs. Ceux que nous avons utilisé pour nos comparaisons ont été effectués par Ghia et al [63] avec un maillage très fin et une méthode de résolution multigrille. Les figures 11.3 et 11.4 donnent le champ de vitesse et les courbes d'iso-pression pour $Re = 1000$. Pour cet écoulement, nous avons également testé l'influence du nombre de points de maillage ainsi que celle de différents schémas de discrétisation sur la qualité de la solution. Les résultats de la figure 11.5 montrent que pour ce cas de calcul, il existe bien une équivalence $h-p$. On peut ainsi améliorer la qualité de la solution en augmentant le nombre de points d'un schéma peu précis, ici le schéma hybride. L'autre solution est de choisir un schéma plus précis, ici le schéma centré d'ordre deux. Cette équivalence est encore mieux perçue sur les courbes de la figure 11.6. Cette figure présente les résultats d'un stage de DEA effectué conjointement à la thèse [17]. Sur cette deuxième figure, on commence cependant à voir des pics numériques près de la paroi supérieure. Leur taille augmente avec l'ordre du schéma. Ceci correspond au problème signalé à propos des termes convectifs et du schéma centré (5.2.1). Sur la figure 11.5, on voit que ce problème disparaît lorsque le nombre de points augmente.

11.1.3 La marche descendante

La marche descendante en laminaire a déjà fait l'objet de nombreuses études. Sur la figure 10.5, nous avons pris $h_m = w = 0.5$, $L = 12$ et $l = 1.5$. Un paramètre de comparaison souvent adopté est la longueur de recollement en aval de la marche $\frac{X_r}{H}$. Les figures 11.7 et 11.8 donnent le champ de vitesse et les courbes d'iso-pression calculés pour $Re = 600$. Le tableau suivant donne les longueurs de recollement calculées ainsi que les résultats obtenus par K. Hannani [31] (méthode des éléments fini et formulation SUPG et GLS). Armaly et al [3] ont effectué une étude expérimentale dont les résultats figurent également sur le tableau ci-dessous.

Nombre de Reynolds	$\frac{X_r}{H}$ présente	$\frac{X_r}{H}$ K. Hannani	$\frac{X_r}{H}$ Armaly et al [3]
400	3.2	4.2	4.2
600	3.9	5.3	5.5
800	3.8	6.0	7.1

Comme on pouvait s'y attendre, l'utilisation du schéma hybride est fortement pénalisante dans ce cas. Les mauvaises prédictions de X_r sont en effet imputables à la vis-

cosité numérique introduite par ce schéma. Mulder et Benucci [14] avaient obtenu $Xr = 4.95$ pour $Re = 800$. Ces auteurs ont utilisé une méthode TVD d'ordre deux. Armaly et al, en plus de l'étude expérimentale, ont fait des calculs par une méthode de volumes finis avec diffusion artificielle leurs prédictions sous-estiment la valeur de $\frac{Xr}{H}$ de 50%.

11.2 Les écoulements turbulents

11.2.1 Introduction

Le but de notre travail concerne en grande partie le calcul d'écoulements turbulents. En effet, nous devons bien prédire l'écoulement dans le canal de la cellule avant d'envisager une étude électrochimique. La validation des codes turbulents est plus difficile que celle des codes laminaires. Il n'existe pas de solution analytique comme dans le cas laminaire. La précision des données expérimentales collectées pour ce type d'écoulements n'est pas garantie. C'est la raison pour laquelle on a de plus en plus recours aux résultats de simulations directes.

Nous avons choisi deux types d'écoulements pour la validation de notre code: les écoulements de Couette-Poiseuille et un écoulement de marche descendante. Les écoulements de Couette-Poiseuille ont été choisis pour leur similitudes avec l'écoulement dans les cellules d'électrozingage. L'écoulement de marche descendante est un peu plus complexe et a déjà fait l'objet d'études en laminaire. La longueur de recollement constitue en outre un paramètre simple et significatif pour les comparaisons. Les résultats dont nous disposons sont d'une part, les données expérimentales obtenues sur les écoulements de Couette-Poiseuille à l'Institut de Mécanique de Fluide de Lille [12] et [15], d'autre part les résultats obtenues sur la marche descendante par Westphal [49].

Pour tous nos calculs, les profils en entrée sont donnés par un modèle de Chien monodimensionnel. Les fonctions d'amortissement du modèle de Chien sont évaluées à l'aide de la vitesse de frottement locale 5.3.2. Les équations du modèle hydrodynamique sont résolues par un algorithme par zones avec assemblage préalable. Les équations du modèle de turbulence sont résolues par linéarisation complète et résolution découplée à l'aide de l'algorithme de Thomas.

11.2.2 Les écoulements de Couette-Poiseuille

Introduction

Les calculs numériques faits sur des expériences réalisées au sein du laboratoire bénéficient d'un avantage important. Ils disposent d'un maximum d'informations sur les conditions expérimentales ce qui facilite l'interprétation des écarts. En effet la comparaison de données produites par des sources différentes n'a d'intérêt que si

on peut expliquer les écarts qui ne manquent jamais d'apparaître. On a alors souvent besoin de plus d'informations sur les conditions de travail des uns et des autres pour comprendre ces écarts. Les premiers calculs faits au sein du laboratoire l'ont été sur des écoulements établis à l'aide de modèles mono-dimensionnels par Monnier [43], Vanhée [64] puis par Deldicque [15]. Les premiers calculs en deux dimensions ont été effectués par K. Hannani [31] à l'aide d'une méthode d'éléments finis.

Les principaux résultats expérimentaux disponibles dans la littérature pour les comparaisons étaient dûs à El-Telbany et Reynolds [61]. A présent le laboratoire dispose d'une importante quantité de données collectées à l'IMFL. Il dispose également d'une base de données de résultats de calcul effectués par simulation directe par Kuroda [1].

Dans les calculs que nous avons effectués, l'écoulement est défini par la géométrie, la vitesse débitante, la vitesse de bande et la viscosité du fluide. La géométrie est celle de la figure 11.1. La vitesse débitante et la viscosité sont utilisées pour calculer des profils de vitesse moyenne, d'énergie et de taux de dissipation turbulents. Ces profils, calculés à l'aide d'un modèle monodimensionnel, sont pris comme conditions d'entrée pour tous les calculs effectués. Ces calculs ayant déjà été faits par K. Hannani [31], nous avons vérifié la concordance de nos résultats avec les siens.

Le principal problème des modèles $k - \epsilon$ à bas Reynolds est lié au comportement des fonctions d'amortissement. Pour plus de détails sur la qualité des différents modèles, on se reportera à Monnier [43], Vanhée [64], Deldicque [15] et K. Hannani [31]. Nous nous contenterons ici de refaire quelques commentaires sur les écarts entre le modèle $k - \epsilon$ de Chien, l'expérience et les simulation directes:

Écoulement de type Poiseuille

Le rapport $\frac{V_h}{U_q}$ qui caractérise l'écoulement de Couette-Poiseuille vaut ici 0.79. Les résultats sont donnés sur les figures 11.9 et 11.10. Le profil de vitesse adimensionné se superpose parfaitement aux profil expérimental. Il en est de même pour le profil de la DNS. Par contre l'accord des profils d'énergie turbulente est moins bon au centre du canal. On observe un pic d'énergie turbulente à la paroi supérieure. Il semble acquit aujourd'hui que les différences observées sont dues à une surestimation par les mesures, problème que l'on attribue à la taille des sondes utilisées. On constate un meilleur accord global de la DNS avec l'expérience aussi bien pour le profil de vitesse que pour le profil d'énergie turbulente. La figure 11.11 donne les profils de vitesse à différentes stations. Ici aussi on a un bon accord avec l'expérience.

Écoulement de type Intermédiaire

Pour cet écoulement, le rapport $\frac{V_h}{U_q}$ vaut 1.224. Les courbes sont données par les figures 11.12 et 11.13. On a toujours une surestimation du pic d'énergie turbulente à la paroi supérieure. Notre profil de vitesse concorde avec celui de l'expérience. La différence entre ces derniers et les résultats de DNS est probablement imputable à

la différence des rapports $\frac{V_h}{U_g}$, avec 0.865 pour la DNS. Les valeurs prédites pour l'énergie turbulente diffèrent des valeurs expérimentales sur toute la hauteur du canal. L'une des causes possibles est l'inadéquation du modèle avec la physique de l'écoulement. En effet, pour l'écoulement de type intermédiaire, le gradient de vitesse est très faible près de la paroi alors que les modèles sont conçu pour les forts gradients. Ici la DNS donne de meilleurs résultats.

Écoulement de type Couette

Pour cet écoulement, le rapport $\frac{V_h}{U_g}$ vaut 1.49. Pour ce cas, nous ne disposons pas de données de simulation directe. Les comparaisons se font uniquement par rapport aux données expérimentales (figure 11.14 et 11.15). Si on constate un bon accord des profils de vitesse, les profils d'énergie turbulente font apparaître des différences significatives près des parois. Les erreurs de mesure peuvent être à nouveau incriminées.

11.2.3 L'écoulement de marche descendante

Le second type écoulement turbulent que nous avons calculé est un écoulement de marche descendante. Le principal intérêt de cet écoulement est la possibilité de comparer la longueur de recollement (un paramètre simple) avec celle mesurée expérimentalement. Les données expérimentales que nous allons utiliser pour cette étude sont celles de Westphal [49]. Sur la figure 10.5, nous avons pris $h_m = 0.066$, $w = 0.1$, $L = 30h_m$ et $l = 3h_m$. Le nombre de Reynolds vaux $R_{emax} = \frac{u_{max}H}{\nu} = 42000$. Cette valeur est basé sur la vitesse maximale à l'entrée du canal et sur la hauteur totale $H = w + h_m$. Le principal problème que nous avons rencontré pour le calcul de cet écoulement concerne le choix du pas de maillage dans chacune des deux directions. Chien propose une étude de stabilité pour son modèle [11]. Cette étude part d'un écoulement dont x est la direction principale. Les conditions de stabilité font apparaître dans ce cas une valeur maximale pour Δx au delà de laquelle la stabilité n'est plus assurée. Ce qui suggère à priori que Δx peut prendre des valeurs arbitrairement petites. Dans le calcul de la marche descendante, il faut pour Δx une valeur suffisamment petite pour une bonne description de l'écoulement au niveau du coin. Nous avons constaté qu'une valeur trop petite rend la convergence difficile voir impossible.

Nous avons trouvé une longueur de recollement $\frac{Xr}{h_m} = 5.75$ cette valeur est nettement moins sous-estimée que dans le cas laminaire, même si elle reste moins bonne que celle prédite par K. Hannani (calculé avec une méthode moins diffusive). Il est vrai que le modèle de turbulence a une influence sur le comportement du schéma dans la mesure où il modifie la viscosité effective du fluide. La figure 11.16 compare le profil longitudinal d'énergie turbulente à $\frac{y}{h_m} = 1.0$, aux résultats de K. Hananni [31] et à l'expérience. Les profils de vitesse dans la zone décollée (figure 11.17a) sont en bon accord avec les données expérimentales. Cet accord se dégrade en aval du point de recollement (figure 11.17b).

11.3 Application à l'électrochimie

Notre travail est directement en rapport avec la simulation d'un processus d'électro-déposition. Pourtant, nous avons passé beaucoup de temps sur la partie hydrodynamique dont nous avons annoncé en introduction le rôle logistique. Malgré le peu de temps qui nous restait, nous avons néanmoins tenu à traiter le problème de l'électro-déposition, même dans le cas le plus simple du courant limite de diffusion dont nous avons décrit le modèle.

Dans cette application, nous avons considéré le canal décrit par la figure 11.1. La paroi mobile (paroi inférieure de la section centrale) du canal est aussi celle où s'effectue le dépôt. Les calculs sont effectués en deux temps: dans un premier temps, on calcule le champ de vitesse et les inconnues du modèle turbulent, ensuite on résout une équation de transport pour C_{Zn} la concentration du zinc.

Le courant limite qui nous intéresse dépend de l'épaisseur de couche limite de diffusion par la formule suivante:

$$J_{lim} = \frac{zFDC_0}{\delta} \quad (11.2)$$

$$\delta = \frac{C_0}{\left. \frac{\partial C_{Zn}}{\partial y} \right|_{y=0}} \quad (11.3)$$

Par conséquent, plus δ est petit, plus J_{lim} est fort, et plus on dispose d'une bonne marge de manoeuvre pour le réglage des cellules.

Nous avons considéré deux cas de vitesse de bande: $V_b = -2 \text{ m/s}$ et $V_b = 0 \text{ m/s}$ pour un nombre de Reynolds $Re_q = \frac{U_q h}{\nu} = 5000$. La figure 11.18, avec $V_b = -2 \text{ m/s}$, présente le cas correspondant au fonctionnement des cellules actuellement installées chez Beautor. La figure 11.19 présente le cas du dépôt dans une cellule carte postale [64]. Dans ces deux cas les tendances sont en accord avec les observations de Vanhée. Pour la carte postale, on observe une phase d'établissement courte dans le sens de l'écoulement, puis une épaisseur constante de la couche limite de diffusion sur l'essentiel de la zone de dépôt. Dans le cas d'un écoulement à contre-courant (figure 11.18), qui correspond à un écoulement de type Poiseuille, la couche limite se développe dans le sens inverse de l'écoulement moyen. On constate là encore que son épaisseur devient très vite constante. Vanhée [64] a montré que cette épaisseur est directement reliée à la vitesse de frottement u_τ sur la paroi où s'effectue le dépôt.

Chapitre 12

Conclusion et perspectives

Le but de notre partenaire industriel est de disposer d'un code de calcul permettant de prédire le comportement des cellules d'électrozingage en fonctionnement. Le processus d'électrodéposition est composé de deux grandes parties: un écoulement dans un canal tridimensionnel cylindrique assurant le transport d'ions jusqu'à la paroi, et le dépôt de métal sur la paroi. Les écoulements sont turbulents et leur bonne description près de la paroi est cruciale pour la prédiction du processus d'électrodéposition proprement dit. Les composantes hydrodynamique turbulente tridimensionnelle et l'électrochimie forment un problème d'une grande complexité.

Cette complexité évidente nous a amené à rechercher des simplifications facilitant la conception d'une solution approchée susceptible d'évoluer vers une solution plus précise dans l'avenir. C'est ainsi que nous avons fait le choix de commencer notre étude par des écoulements bidimensionnels. Ce choix était aussi justifié par les travaux antérieurs faits sur les cas monodimensionnels. Pour la modélisation turbulente, nous avons renoncé à l'utilisation des fonctions de paroi qui constituent une approximation assez peu précise de la solution. Le traitement des parois par un modèle de turbulence à bas Reynolds étant de loin plus coûteuse, nous avons investi dans l'étude d'algorithmes de résolution efficaces.

Le point de départ de notre travail est le code TEACHT. C'est en effet avec ce code que nous nous sommes initié à la résolution numérique des équations de Navier-Stokes. Ce code a joué un rôle très important dans la vulgarisation des méthodes de calcul numérique appliquées aux écoulements de fluides. En tant qu'outil d'investigation industriel, le TEACHT avaient quelques défauts qui ne nous permettaient pas de l'adopter tel quel pour notre travail. L'architecture simple du code ne facilite pas les adaptations à de nouvelles configurations ou l'ajout de nouvelles variables. De plus l'algorithme SIMPLE implémenté pour la résolution du problème hydrodynamique est très peu performant. Le modèle de turbulence implémenté utilise des fonctions de paroi.

Dans le but de remplacer l'algorithme SIMPLE, nous avons étudié la méthode par zones et l'algorithme multigrille pour l'accélération de la convergence. Nous aurions

pu tester des méthodes itératives de type gradient conjugué et bien d'autres encore, mais la diversité des méthodes est telle qu'il est difficile de justifier objectivement un choix par rapport à un autre. Toujours est-il que la méthode par zones comporte pour nous l'avantage d'être parallélisable. En ce qui concerne la méthode multigrille, les facteurs d'accélération qui lui sont prêtés ont suscité notre curiosité et nous n'avons pas été déçu de ce point de vue.

Nous avons commencer par implémenter l'algorithme SIMPLE en C après l'avoir transformé en SIMPLER. La phase suivante a été la mise en oeuvre pour un écoulement de canal plan de la méthode par zones. Avec la méthode par zones et la résolution implicite, nous nous sommes retrouvés confrontés au problème de pivot nul lié au caractère incompressible de l'écoulement. Nos investigations sur cette question ont abouti au constat que dans la majorité des cas où ce problème se pose, l'incompressibilité du fluide est moins à mettre en cause que l'algorithme ou même parfois le maillage (décalé) utilisé. Ce résultats nous a conduit à proposer une implémentation de la méthode par zones qui prend en compte la nature de l'écoulement.

Les premiers tests sur l'écoulement de marche descendante nous ont amenés à considérer le problème de la représentation du domaine pour une prise en compte plus générale des conditions aux limites. Après plusieurs tentatives infructueuses avec une représentation mono-rectangle, nous avons opté pour une représentation en sous-domaines. Après une mise au point de cette représentation sur le cas de la marche descendante, nous avons testé la résolution par sous-domaines sans succès. Nous avons alors implémenté une procédure d'assemblage préalable des équations avant résolution qui s'est avérée efficace, notamment pour la résolution de la marche descendante en turbulent. Cette représentation a été ensuite testée sur une configuration plus complexe en laminaire (figure 10.12a). Le champ de vitesse (figure 12.1) et les courbes d'iso-pression (figure 12.2) que nous obtenons montrent une bonne prise en compte des nombreuses conditions aux limites. Il est à signaler que ces résultats ont été obtenus sans aucune modification du code. Dès la fin de la mise au point de la méthode par zones sur l'écoulement de canal plan, nous avons commencé la transformation de l'architecture pour l'implémentation de la méthode multigrille. Cette partie a été menée en parallèle avec les autres améliorations du code.

Le code que nous avons développé a atteint les principaux objectifs fixés. La prise en compte de géométries décomposables en sous-domaines rectangulaires se fait à partir de fichiers de données externes. Le rajout ou la modification d'équations se réduit aux tâches fondamentales de codage de la version discrétisée des équations. Tout ce qui concerne la déclaration des variables est largement pris en charge par la structure de base du code. Nous avons implémenté une structure de donnée basée sur le traitement par sous-domaines. Le problème hydrodynamique peut être résolu par deux algorithmes de base, SIMPLER et la méthode par zones à laquelle nous avons apporté des améliorations significatives. Les équations de transport mono-variables sont résolues par une méthode de balayage avec inversion par l'algorithme de Thomas, après linéarisation globale. La structure du code est conçue pour supporter une

organisation multigrille complètement découplée des algorithmes de résolution de base.

La résolution des équations de Navier-Stokes reste un problème particulièrement difficile même pour les méthodes numériques. Dans l'évaluation des performances de notre code, nous arrivons à la conclusion que l'algorithme SIMPLER est meilleur que les autres pour un écoulement de cavité. Ce même algorithme est inefficace pour les configurations avec comportement globalement convectif. Pour cette dernière catégorie de problèmes, notre adaptation de la méthode par zones donne de bons résultats. La méthode multigrille dans le cas des écoulements laminaires a tenu ses promesses en terme d'accélération de la convergence. Pour l'instant notre code multigrille ne converge pas en turbulent. Mais nous avons vu que son utilisation comme méthode d'amorçage dans ce cas apporte un bénéfice considérable par rapport à l'utilisation d'un modèle de turbulence plus simple.

En ce qui concerne la validation du code, nous obtenons avec une bonne précision la solution analytique de l'écoulement de canal plan établi. Nos résultats sur la cavité sont conformes à ceux du benchmark [3], et les tests que nous avons effectués avec des schémas de grande précision sont concluants. En ce qui concerne la marche descendante, les mauvais résultats que nous obtenons pour la prédiction de la longueur de recollement nous rappellent que le schéma hybride devrait être abandonné au profit d'un schéma moins diffusif.

En ce qui concerne les écoulements turbulents, nous obtenons globalement un bon accord de nos résultats avec les données expérimentales disponibles ainsi qu'avec les résultats de simulations directes. Les tests limités que nous avons fait en électrochimie donnent des tendances en accord avec celles trouvées par L. Vanhée [64].

L'efficacité des codes de calcul et la facilité de leur utilisation passe visiblement par une certaine complexification de leur architecture. Aujourd'hui, les outils pour gérer cette complexité sont disponibles. Afin d'éviter que le langage fortran ne devienne une limitation pour l'implémentation d'une architecture qui n'était pas encore définie, mais que nous souhaitons robuste, nous avons fait le choix de programmer en C. En contrepartie de sa puissance, ce langage a le défaut de rallonger le temps de mise au point des codes par rapport aux autres langages.

Mis à part la robustesse de l'architecture, la possibilité de réutiliser et même d'échanger certains éléments du code devient une nécessité. Cette réutilisation n'est possible, que s'il y a un minimum de concertation dans la définition des éléments du code susceptible d'être déplacés vers un autre code.

L'un des obstacles à cette concertation réside dans la diversité des méthodes numériques utilisées. Les plus courantes sont les méthodes spectrales, des différences finies, volumes finis, et éléments finis. A ces méthodes de discrétisation, il faut ajouter les méthodes d'interpolation propres à chacune d'elles. Une fois que les équations sont construites, on aborde avec les algorithmes d'inversion, un domaine d'une diversité au moins aussi importante que la précédente.

Si la méthode des éléments finis est un candidat sérieux à une certaine unification, il est difficile d'imaginer son adoption par tous à court terme, ne serait-ce que pour des raisons de formation. Il est cependant possible de commencer à mettre en place les éléments d'une classification des méthodes aussi bien de discrétisation que de résolution. Il peut être important dans cette classification de clarifier les liens à notre avis très importants qui existent entre elles, de manière à tracer des connexions exploitables pour la programmation.

Les projets informatiques disposent de plus en plus d'outils adaptés pour leur gestion. La complexité des projets informatiques est d'abord lié à l'environnement de travail constitué par un nombre important de composants matériel et logiciels en interaction. La traduction d'une solution en langage de programmation a pour conséquence l'écriture de lignes de codes dont la vérification pose très vite des problèmes importants. Les outils sont souvent le produit d'une expérience dans le domaine de la conception des programmes. En tant qu'expérience à transmettre, les justifications de leur rôle sont plus faciles à comprendre après une certaine pratique personnelle. C'est ce qui explique une certaine réticence des programmeurs amateurs à adopter des outils et des méthodes dont le rôle et l'utilisation ne sont pas bien compris. En plus de l'environnement de programmation, la tendance actuelle est à la complexification des produits logiciels et à une exigence accrue sur leur qualité, même pour ceux réalisés par une seule personne. La maîtrise de certains outils de développement est donc en voie de devenir une nécessité pour tous.

Le code que nous avons conçu, pour devenir conforme au besoins de réutilisation évoqués, devrait être transféré dans un environnement de programmation disposant d'outils adaptés pour la gestion des évolutions futures. Cet environnement devra faciliter la gestion des différentes versions afin de toujours permettre la comparaison d'une version stabilisée avec la version en cours de modification. Il devra également faciliter la documentation du code dans le but de bien évaluer les conséquences d'une évolution envisagée. Une fois le code installé dans un environnement de programmation confortable, il sera possible de faire une analyse poussée de l'architecture actuelle par rapport à différentes possibilités d'extension.

En donnant une certaine généralité au code, nous nous retrouvons avec une quantité importante de données à gérer par l'utilisateur. Le nombre de fichiers de configuration peut engendrer des erreurs de manipulation dans la mise en oeuvre du code. Deux points sont à améliorer ici. Le premier doit permettre une vérification poussée de la cohérence des données contenus dans les fichiers, avec diagnostic d'erreurs et suggestion par le programme sur leur origine. Le second point concerne l'adjonction d'une interface graphique comme aide à la génération mais aussi à la modification des fichiers de configuration.

Sur le plan plus numérique, le maillage décalé que nous avons utilisé présente un avantage du point de vue de l'approximation des équations. L'équation de continuité est discrétisée avec précision. De plus il n'est pas nécessaire de spécifier des conditions aux limites pour la variable de pression. Sur le plan de la mise oeuvre,

le problème est autre. La prise en compte du maillage décalé dans le code oblige à distinguer des traitements qui autrement seraient communs à toutes les variables. Cette distinction intervient également dans la structure des données où les différentes variables ne sont pas décrites de la même manière. La mise en oeuvre de la méthode multigrille et la nécessité d'interpoler les inconnues de pression amène à prendre en compte pour cette variable le comportement à la frontière. Ce comportement nécessite la prise en compte de conditions aux limites dite numériques pour la pression remettant ainsi en cause un des avantages du maillage décalé.

En ce qui nous concerne, nous pensons que ces problèmes peuvent être pris en compte dès la discrétisation des équations en adoptant le maillage non-décalé. Les conditions aux limites données dans ce cas seront utilisées pour le multigrille, et l'équation de continuité sera discrétisée en utilisant des formules d'interpolation telles que celle données par De Mulder et Benucci [35]. Il serait judicieux pour cette évolution d'évaluer les conséquences sur notre implémentation de la méthode par zones.

Avec le maillage non-décalé, le passage aux domaines tridimensionnels parallélépipédiques est grandement facilité. Dans cette optique, la capacité mémoire des ordinateurs utilisés devient importante. Elle va déterminer les évolutions à faire sur la méthode par zones afin qu'elle puisse supporter sans être trop pénalisée le coût d'une dimension supplémentaire. Dans l'évolution de l'architecture, il peut être important de maintenir un lien fort avec le code bidimensionnel afin de pouvoir l'utiliser pour le test et la mise au point de fonctionnalités qui leur sont communes.

Avant d'envisager une extension au traitement de domaines plus généraux, il faudra étudier le bénéfice du passage à la méthode des éléments finis en fonction des paramètres suivants: la possibilité d'utiliser l'expérience acquise sur le code volumes finis, évolution du traitement des modèles de turbulences avec la méthode des éléments finis et surtout performances des algorithmes de résolution. Pour ce dernier point, il faudra voir dans quelle mesure l'algorithme par zones développé pour un maillage régulier, peut être étendu à la résolution des équations provenant d'une discrétisation éléments finis non-structuré.

Dans l'immédiat, on peut envisager l'implémentation de modèles de turbulence plus sophistiqués. Les modèles aux tensions de Reynolds donnent des résultats intéressants (voir [15]). De la même manière on peut faire évoluer le modèle d'électrochimie afin de traiter des situations autres que celle du courant limite de diffusion. A plus long terme, on peut envisager l'implémentation de modèles diphasiques prenant en compte le dégagement d'hydrogène à la cathode. Sur le plan du temps de calcul, l'adaptation de nos algorithmes aux machines parallèles peut apporter des gains de temps importants améliorant ainsi l'apport du code en temps qu'outil d'investigation. Sur ce point, le type de machine parallèle disponible déterminera les adaptations les plus appropriées. La méthode par zones est sur ce point la plus facile à adapter. Cependant dans la pratique, les zones peuvent être transformée en sous-domaines attribués aux différents processeurs. Dans ce cas on testera d'abord la bonne transmission des informations en monoprocesseur. La méthode multigrille

pose plus de problèmes pour la parallélisation. Ici il faut choisir entre l'algorithme non-linéaire et l'application du multigrille à la résolution du problème linéaire plus simple.

Bibliographie

- [1] N. Kasagi A. Kuroda and M. Hirata. Direct numerical simulation of turbulent plane couette-poiseuille flows: Effect of mean shear on the wall turbulence structures. *Ninth symposium on turbulent shear flows Kyoto Japan*, pages –, August 1993.
- [2] Alain Autret. Calcul d'écoulements turbulents par la méthode des éléments finis. Master's thesis, Centre d'Etudes Nucleaires de Cadarache, Mars 1986.
- [3] J. C. F. Pereira B. F. Armaly, F. Durst and B. Schonung. Experimental and theoretical investigation of backward facing step flow. *J. Fluid Mech.*, 127: 473, 1983.
- [4] X.S. Bai and L. Fuchs. A multigrid method for the calculation of turbulence and combustion. In P. Wesseling P. W. Hemker, editor, *Multigrid methods IV*, page 131. Birkhauser, July 1993.
- [5] J. Baranger. *Introduction a l'analyse numerique*. Herman Paris, 1 edition, 1977.
- [6] M. Bellanger. *Traitement du signal. Theorie et pratique*. Masson, 4 edition, 1990.
- [7] M. E. Braaten and S. V. Patankar. A block-corrected subdomain solution procedure for recirculating flow calculation. *Numerical Heat Transfert*, 15: 1–20, 1989.
- [8] A. Brandt. Multi-level adaptive solution to boundary-value problems. *Mathematics of computation*, 31: 333, April 1977.
- [9] Achi Brandt. *Guide to Multigrid Development*. The Weizman Institute of Science, Rehovot, Israel 76100. Department of Applied Mathematics.
- [10] Achi Brandt. *Guide to Multigrid Development*. Department of Applied Mathematics The Weizman Institute of Science, 1 edition.
- [11] K.-Y. Chien. Predictions of channel and boundary-layer flows with a low-reynolds-number turbulence model. *AIAA Journal*, 20(1): 2–5, january 1982.

- [12] Kleber Corenflos. Etude experimentale d'écoulements de couette-poiseulle turbulent a faible nombre de reynolds. Master's thesis, Universite des sciences et techniques Lille Flandre Artois, Cite scientifique Villeneuve d'ascq, Mars 1993.
- [13] D. N. de G. Allen and R. V. Southwell. Relaxation methods applied to determine the motion, in two dimensions, of a viscous fluid past a fixed cylinder. *Q. J. Mech Appl. Maths.*, 8: 129-145, - 1955.
- [14] T.F.O. De Mulder and C. Benocci. Wiggle free solution of the incompressible navier-stokes equations over a collocated mesh. *Tech. Note 179*, 179: 1, 1991.
- [15] Amaury Deldicque. Etude de l'influence du nombre de reynolds sur les ecoulement turbulents de couette-poiseulle a l'aide d'une approche experimentale et d'une modelisation aux containtes de reynolds. Master's thesis, Universite des sciences et techniques Lille Flandre Artois, Cite scientifique Villeneuve d'ascq, Novembre 1995.
- [16] S. G. Eisenstat et al. Yale sparse matrix package: The non-symetric code. *Report 114 Yale University*, 114: 1, 1974.
- [17] Samir Ben Youssef Lamiri et Baleba. Etude de différentes formulations en volume finis des termes convectif des équations de navier-stokes. Master's thesis, Universite des sciences et techniques Lille Flandre Artois, Cite scientifique Villeneuve d'ascq, Octobre 1996.
- [18] Gene H-Golub et Gerard A-Meurant. *Resolution numerique des grands systemes lineaires*. CDER-EDF, 1 edition, 1985.
- [19] R. Boite et H. Leich. *Les filtres numeriques. Analyse et synthese des filtres unidimensionnels*. Masson, 3 edition, 1990.
- [20] R. Dautray et J. L. Lions. *Analyse mathématique et calcul numérique pour les sciences et les techniques*, volume 1. Commissariat à l'énergie atomique, 1 edition, 1985.
- [21] R. Dautray et J. L. Lions. *Analyse mathématique et calcul numérique pour les sciences et les techniques*, volume 1. Commissariat à l'énergie atomique, 1 edition, 1985.
- [22] R. P. Fedorenko. A relaxation method for solving elliptic difference equations. *Ž. Vyčisl. Mat. Fiz*, MR 25 #766, 4: 922, 1961.
- [23] R. P. Fedorenko. On the speed of convergence of an iteration process. *Ž. Vyčisl. Mat. Fiz*, MR 31 #6386, 4: 559, 1964.
- [24] William H. Press; William T. Vetterling; Saul A. Teukolski; Brian P. Flannery. *NUMERICAL RECIPES in C The Art of scientific computing*. gc, 2 edition, 1992.

- [25] L. Fuchs. A local mesh-refinement technique for incompressible flows. *computers and fluids*, 14: 69–81, 1986.
- [26] L. Fuchs. Turbulence modelling as a multi-level approach. In P. Wesseling P. W. Hemker, editor, *Multigrid methods IV*, page 35. Birkhauser, July 1993.
- [27] Paul Louis George. *Génération automatique de maillages*. Masson, 1 edition, 1990.
- [28] M. Germano. Turbulence; the filtering approach. *J. Fluid Mech.*, 238: 325, 1992.
- [29] Andrew G. Godfrey, Curtis R. Mitchell, and Robert W. Walters. Practical aspect of spatially high-order accurate methods. *AIAA Journal*, 31: 1634–1642, september 1993.
- [30] W. Hackbush and U. Trottenberg. *Multigrid Methods, Lecture Notes in Mathematics*. Springer, 1 edition, 1982.
- [31] Samak Kazemzadeh Hannani. Calcul d'écoulements laminaires et turbulents par une méthode des éléments finis: Influence de la formulation. Master's thesis, Université des Sciences et Technologies de Lille, juin 1996.
- [32] F. H. Harlow and Welch J. E. Numerical calculation of time dependant viscous incompressible flow of fluid with free surface. *Physics of Fluid*, 8: 2182, 1965.
- [33] P. W. Hemker. On the order of prolongations and restrictions in multigrid procedures. *J. of Computational and Applied Mathematics*, 32: 423, 1990.
- [34] C. Hirsch. *Numerical computation of internal and external flows*, volume 1. 2 edition, 1.
- [35] Von Karman Institute. *High resolution (UPWIND and TVD) methods for the compressible equations*. Von Karman Institute for Fluid Dynamics, 1 edition, 1994.
- [36] P. Johansson and Lars Davidson. A full multigrid method applied to turbulent flow using the simplec algorithm together with a collocation arrangement. In P. Wesseling P. W. Hemker, editor, *Multigrid methods IV*, page 245. Birkhauser, July 1993.
- [37] B. P. Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Comput. Methods Appl. Engrg*, 19: 59–98, 1979.
- [38] B. P. Leonard and J. E. Drummond. Why you should not use hybrid, power-law, related exponential schemes for convective modelling there are much better alternatives. *Int. J. For Num. Meth. in Fluids*, 20: 421–442, - 1995.

- [39] F. S. Lien and M. A. Leschziner. Multigrid acceleration for complex flow including turbulence. In U. Trottenberg W. Hackbusch, editor, *International Series of Numerical Mathematics*, volume 98, page 277. Birkhauser, Marsh 1991.
- [40] P. Marche and C. Beauchemin. On the solution of the navier-stokes equation by the penalty function method. In CIMME Barcelona, editor, *Finite element in fluids new trends and applications*, pages 166–173. Pineridge Press, - 1994.
- [41] D. J. Mavriplis and V. Venkatakrishnan. Agglomeration multigrid for viscous turbulent flows. In ICASE, editor, *25th AIAA Fluid Dynamics Conference*. AIAA, June 1994.
- [42] B. Mohammadi and O. Pironneau. *Analysis of the K-Epsilon Turbulence Model*. Wiley and Masson, 1994.
- [43] J. C. Monnier. Hydrodynamique des écoulements turbulents dans les cellules d'électrodéposition. Master's thesis, Université des sciences et techniques Lille Flandre Artois, Cite scientifique Villeneuve d'ascq, Mai 1989.
- [44] John Newman. *Electrochemical systems*. Prentice-Hall, 1973.
- [45] S. V. PATANKAR. *Numerical Heat Transfert and Fluid Flows*, volume 1. Mc Graw Hill, 1 edition, 1980.
- [46] S. V. Patankar and D. B. Spalding. A calculation procedure for heat, mass and momentum transfer in three dimensional parabolic flow. *Int. J. of Heat and Mass Transfer*, 15: 1787, 1972.
- [47] C. Y. Perng and R. L. Street. A coupled multigrid-domain-splitting technique for simulating imcompressible flows in geometrically complex domains. *Int. J. For Num. Meth. in Fluids*, 13: 269–286, - 1991.
- [48] Valerie PERRIER. *Ondelette et Simulation Numérique*. ONERA, 1 edition, 1991.
- [49] J. P. Joohnston et J. K. Eaton R. V. Westphal. *Experimental study of Flow reattachment in single-sided sudden expansion*. NASA Contractor Reports, 1 edition, 1984.
- [50] G. D. Raithby and G. E. Schneider. Numerical solution of problems in incompressible fluid flow: treatment of velocity-pressure coupling. *Num. Heat Transfer*, 2: 417, 1974.
- [51] Ph. Refregier. *Theorie du signal. Signal, information, fluctuation*. Masson, 1 edition, 1993.

- [52] Arnold Reusken. Multigrid with matrix-dependent transfert operators for convection-diffusion problems. In P. Wesseling P. W. Hemker, editor, *Multigrid methods IV*, page 269. Birkhauser, July 1993.
- [53] L. F. Richardson. The approximate arithmetical solution by finite differencies of physical problems involving differential equations with an application to the stresses in a masonry dam. *The transaction s of the Royal Society of London*, 210: 307-357, 1910.
- [54] R. Schiestel. *Modélisation et simulation des écoulements turbulents*. Hermes, 1993.
- [55] G. E. Schneider and M. Zedan. A modified strongly implicit procedure for the numerical solution of field problems. *Numerical Heat Transfert*, 4: 1-19, - 1981.
- [56] S. Schoemaker. *Computer networks and Simulation*. North Holland, 2 edition, 1982.
- [57] R. V. Southwell. Relaxation methods in engenierring sciences. *Oxford univ. Press*, 1940.
- [58] R. V. Southwell. Relaxation methods in theoritical physics. *Calderon Press Oxford*, 1946.
- [59] H. L. Stone. Iterative solution of implicit approximations of multidimensionnal partial differential equations. *SIAM Journal of Numerical Analysis*, 5: 530-558, - 1968.
- [60] U. Stuben, K.; Trottenberg. *Multigrid methods: Fundamental algorithms, model problem analysis*. Spring, 1 edition, 1982.
- [61] M. M. M. El Telbany and A. J. Reynolds. Turbulence in plane channel flows. *J. F. M.*, 11: 283-318, 1981.
- [62] D. C. Thoman and Szewczyk. Numerical solution of,time dependant two dimensional flow of a viscous flow of a viscous, incompressible over stationary and rotating cylinders. *Tech. Rept. 66-14, heat Transfert and Fluid Mechanics Lab., Dept. of Mechanical Engeneering, University of Notre Dame, Notre Dame, Indiana*, 1966.
- [63] K. N. Ghia U. Ghia and C. T. Shin. High-reynolds solution for icompressible flow using navier-stokes equations and the multigrid method. *J. Comp. Phys.*, 48: 387-411, - 1982.
- [64] Luc Vanhee. Contribution à l'étude du comportement hydrodynamique des cellules d'électrodeposition. Master's thesis, Universite des sciences et techniques Lille Flandre Artois, Cite scientifique Villeneuve d'ascq, Novembre 1991.

- [65] S. P. Vanka. Block-implicit calculation of steady turbulent recirculating flows. *Int J. Heat Mass Transfert*, 15: 1, 1989.
- [66] J. Von Neumann. Proposal an analysis of numerical methods for the treatment of hydrodynamical shock problems. *Nat. Def. and Res. com. Report AM-551*, 1944.
- [67] J. Von Neumann. A method for numerical calculation of hydrodynamic shocks. *J. of applied Physics*, 21: 232–257, 1950.
- [68] C. B. Vreugdenhil and B. Koren. *Numerical Methods for Advection-Diffusion Problems*, volume 45. Vieweg, 2 edition, 1993.
- [69] B. Carnahan; H. A. Luther; J. O. Wilkes. *Applied Numerical methods*. John Wiley and Sons, 1 edition.
- [70] M. Zedan and G. E. Schneider. A strongly implicit simultaneous variable solution procedure for velocity and pressure in fluid flow solution. *AIAA 18 thermophysics conference*, 83: –, 1983.
- [71] J. Zhu and W. Rodi. A low dispersion and bounded convection scheme. *Computer Methods in applied Mechanics and Engineering*, 92: 87–96, - 1991.
- [72] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method*. McGraw-Hill, 4 edition, 1989.

Annexe A

Détails de discrétisation

A.1 Schéma hybride

Les différents coefficients sont donnés par les formules suivantes:

$$\begin{aligned}A_W^\phi &= D_W^\phi \cdot \text{Max}(0, (1 - \frac{|P_W^\phi|}{2})) + \text{Max}(-F_W^\phi, 0) \\A_E^\phi &= D_E^\phi \cdot \text{Max}(0, (1 - \frac{|P_E^\phi|}{2})) + \text{Max}(-F_E^\phi, 0) \\A_N^\phi &= D_N^\phi \cdot \text{Max}(0, (1 - \frac{|P_N^\phi|}{2})) + \text{Max}(-F_N^\phi, 0) \\A_S^\phi &= D_S^\phi \cdot \text{Max}(0, (1 - \frac{|P_S^\phi|}{2})) + \text{Max}(-F_S^\phi, 0) \\A_P^\phi &= A_s^\phi + A_n^\phi + A_w^\phi + A_s^\phi + A_{p\phi}\end{aligned}\tag{A.1}$$

$$\begin{aligned}P_w^\phi &= \frac{F_w^\phi}{D_w^\phi} \\P_n^\phi &= \frac{F_n^\phi}{D_n^\phi} \\P_e^\phi &= \frac{F_e^\phi}{D_e^\phi} \\P_s^\phi &= \frac{F_s^\phi}{D_s^\phi}\end{aligned}\tag{A.2}$$

$$F_e^\phi = \rho \cdot u_w \cdot \Delta y_p \quad D_w^\phi = \Delta y_p \frac{\Gamma_w}{\Delta x_w}$$

$$\begin{aligned}
F_e^\phi &= \rho \cdot u_e \cdot \Delta y_p & D_e^\phi &= \Delta y_p \frac{\Gamma_e}{\Delta x_e} \\
F_s^\phi &= \rho \cdot v_s \cdot \Delta x_p & D_s^\phi &= \Delta x_p \frac{\Gamma_s}{\Delta y_s} \\
F_n^\phi &= \rho \cdot v_n \cdot \Delta x_p & D_n^\phi &= \Delta x_p \frac{\Gamma_n}{\Delta y_n}
\end{aligned} \tag{A.3}$$

Pour la variable u on aura:

Les vitesses de convection:

$$\begin{aligned}
u_w &= 0.5(u(i-1, j) + u(i, j)) & u_e &= 0.5(u(i, j) + u(i+1, j)) \\
v_s &= 0.5(v(i-1, j+1) + v(i, j+1)) & v_n &= 0.5(v(i-1, j) + v(i, j))
\end{aligned} \tag{A.4}$$

Les paramètres de maillage:

$$\begin{aligned}
\Delta x_p &= x(i+1) - x(i) & \Delta y_p &= yv(j+1) - y(j) \\
\Delta x_w &= xu(i) - xu(i-1) & \Delta x_e &= xu(i+1) - xu(i) \\
\Delta y_s &= y(j) - y(j-1) & \Delta y_n &= y(j+1) - y(j)
\end{aligned} \tag{A.5}$$

Les paramètres de diffusion:

$$\begin{aligned}
\Gamma_w &= \mu + \mu_t(i-1, j) \\
\Gamma_e &= \mu + \mu_t(i, j) \\
\Gamma_s &= \mu + 0.25(\mu_t(i-1, j+1) + \mu_t(i, j+1) + \mu_t(i-1, j) + \mu_t(i, j)) \\
\Gamma_n &= \mu + 0.25(\mu_t(i-1, j) + \mu_t(i, j) + \mu_t(i-1, j-1) + \mu_t(i, j-1))
\end{aligned} \tag{A.6}$$

Termes sources:

$$S_u = \Delta y_p (p(i, j) - p(i-1, j)) \tag{A.7}$$

Pour la variable v on aura:

Les vitesses de convection:

$$\begin{aligned}
u_w &= 0.5(u(i, j) + u(i, j - 1)) & u_e &= 0.5(u(i + 1, j) + u(i + 1, j - 1)) \\
v_s &= 0.5(v(i, j) + v(i, j - 1)) & v_n &= 0.5(v(i, j + 1) + v(i, j))
\end{aligned} \tag{A.8}$$

Les paramètres de maillage:

$$\begin{aligned}
\Delta x_p &= xu(i + 1) - xu(i) & \Delta y_p &= y(j) - y(j - 1) \\
\Delta x_w &= x(i) - x(i - 1) & \Delta x_e &= x(i + 1) - x(i) \\
\Delta y_s &= yv(j) - yv(j - 1) & \Delta y_n &= yv(j + 1) - yv(j)
\end{aligned} \tag{A.9}$$

Les paramètre de diffusion:

$$\begin{aligned}
\Gamma_w &= \mu + 0.25(\mu_t(i - 1, j) + \mu_t(i - 1, j - 1) + \mu_t(i, j) + \mu_t(i, j - 1)) \\
\Gamma_e &= \mu + 0.25(\mu_t(i + 1, j) + \mu_t(i + 1, j - 1) + \mu_t(i, j) + \mu_t(i, j - 1)) \\
\Gamma_s &= \mu + \mu_t(i, j) \\
\Gamma_n &= \mu + \mu_t(i, j + 1)
\end{aligned} \tag{A.10}$$

Termes sources:

$$S_v = \Delta x_p(p(i, j) - p(i, j - 1)) \tag{A.11}$$

Pour la variable k on aura:

Les vitesse de convection:

$$\begin{aligned}
u_w &= u(i, j) & u_e &= u(i + 1, j) \\
v_s &= v(i, j) & v_n &= v(i, j + 1)
\end{aligned} \tag{A.12}$$

Les paramètres de maillage:

$$\begin{aligned}
\Delta x_p &= xu(i + 1) - xu(i) & \Delta y_p &= yv(j) - yv(j - 1) \\
\Delta x_w &= x(i) - x(i - 1) & \Delta x_e &= x(i + 1) - x(i) \\
\Delta y_s &= y(j) - y(j - 1) & \Delta y_n &= y(j + 1) - y(j)
\end{aligned} \tag{A.13}$$

Les paramètre de diffusion:

$$\begin{aligned}
 \Gamma_w &= \mu + \frac{0.5}{\sigma_k} (\mu_t(i, j) + \mu_t(i-1, j)) \\
 \Gamma_e &= \mu + \frac{0.5}{\sigma_k} (\mu_t(i+1, j) + \mu_t(i, j)) \\
 \Gamma_s &= \mu + \frac{0.5}{\sigma_k} (\mu_t(i, j) + \mu_t(i, j-1)) \\
 \Gamma_n &= \mu + \frac{0.5}{\sigma_k} (\mu_t(i, j+1) + \mu_t(i, j))
 \end{aligned} \tag{A.14}$$

Termes sources:

$$\begin{aligned}
 S_k &= -\Delta x_p \Delta y_p P_p \\
 A_{pk} &= -\Delta x_p \Delta y_p \left(\frac{2\nu}{d_p^2} + \frac{e_P}{k_P} \right)
 \end{aligned} \tag{A.15}$$

Pour la variable ϵ on aura:

Les vitesse de convection:

Voir A.12

Les paramètres de maillage:

Voir A.13

Les paramètre de diffusion:

$$\begin{aligned}
 \Gamma_w &= \mu + \frac{0.5}{\sigma_\epsilon} (\mu_t(i, j) + \mu_t(i-1, j)) \\
 \Gamma_e &= \mu + \frac{0.5}{\sigma_\epsilon} (\mu_t(i+1, j) + \mu_t(i, j)) \\
 \Gamma_s &= \mu + \frac{0.5}{\sigma_\epsilon} (\mu_t(i, j) + \mu_t(i, j-1)) \\
 \Gamma_n &= \mu + \frac{0.5}{\sigma_\epsilon} (\mu_t(i, j+1) + \mu_t(i, j))
 \end{aligned} \tag{A.16}$$

Termes sources:

A.2. CALCUL DE DÉRIVÉES POUR LE TERME DE PRODUCTION DE TURBULENCE 15

$$\begin{aligned}
 S_\epsilon &= -\Delta x_p \Delta y_p \frac{\epsilon_p}{k_p} C_{\epsilon 1} P_p \\
 A_{p\epsilon} &= -\Delta x_p \Delta y_p \left(\frac{2\nu \exp\left(\frac{-C_4 u_r d_p}{\nu}\right)}{d_p^2} + \frac{\epsilon_p}{k_p} C_{\epsilon 2} f_p \right)
 \end{aligned}
 \tag{A.17}$$

Pour la variable ϵ on aura:

Les vitesses de convection:

Voir A.12

Les paramètres de maillage:

Voir A.13

Les paramètres de diffusion:

$$\begin{aligned}
 \Gamma_w &= D_{C_{2n}} + 0.5(\mu_t(i, j) + \mu_t(i - 1, j)) \\
 \Gamma_e &= D_{C_{2n}} + 0.5(\mu_t(i + 1, j) + \mu_t(i, j)) \\
 \Gamma_s &= D_{C_{2n}} + 0.5(\mu_t(i, j) + \mu_t(i, j - 1)) \\
 \Gamma_n &= D_{C_{2n}} + 0.5(\mu_t(i, j + 1) + \mu_t(i, j))
 \end{aligned}
 \tag{A.18}$$

Termes sources:

$$S_{C_{2n}} = 0
 \tag{A.19}$$

A.2 Calcul de dérivées pour le terme de production de turbulence

$$\begin{aligned}
 \frac{\Delta u}{\Delta x_P} &= \frac{ue - uw}{\Delta x_p} \\
 \frac{\Delta u}{\Delta y_P} &= \frac{un - us}{\Delta y_p} \\
 \frac{\Delta v}{\Delta x_P} &= \frac{ve - vw}{\Delta x_p} \\
 \frac{\Delta v}{\Delta y_P} &= \frac{vn - vs}{\Delta y_p}
 \end{aligned}
 \tag{A.20}$$

$$\begin{aligned}
us &= 0.5(usw + use) \\
vs &= v(i, j) \\
un &= 0.5(unw + une) \\
vn &= v(i, j + 1) \\
vw &= 0.5(vsw + vnw) \\
uw &= u(i, j) \\
ve &= 0.5(vse + vne) \\
ue &= u(i + 1, j)
\end{aligned}
\tag{A.21}$$

$$\begin{aligned}
usw &= 0.5(u(i, j - 1) + u(i, j)) \\
use &= 0.5(u(i + 1, j - 1) + u(i + 1, j)) \\
unw &= 0.5(u(i, j + 1) + u(i, j)) \\
une &= 0.5(u(i + 1, j + 1) + u(i + 1, j)) \\
vsw &= 0.5(v(i - 1, j) + v(i, j)) \\
vnw &= 0.5(v(i - 1, j + 1) + v(i, j + 1)) \\
vse &= 0.5(v(i + 1, j) + v(i, j)) \\
vne &= 0.5(v(i + 1, j + 1) + v(i, j + 1))
\end{aligned}
\tag{A.22}$$

Figures:

- figure 1.1 Schéma de principe d'une cellule d'électrozingage à hydrodynamique contrôlée de type Beautor.
- figure 4.1 Maillage mono-rectangle avec désactivation de points.
- figure 4.2 Décomposition d'un écoulement de marche en quatre rectangles dont un inactif et segments de maillages correspondant.
- figure 4.3 Raccordement de maillage suivant x pour deux rectangles voisins.
- figure 5.1 Description du maillage décalé et positionnement des inconnues
- figure 5.2 Pas de maillage pour la discrétisation d'une équation.
- figure 5.3 Traitement des frontières par la condition miroir.
- figure 5.4 Problèmes posés par la mise en oeuvre de la condition miroir.
- figure 5.5 Pas de maillage pour l'estimation du terme convectif par le schéma QUICK.

A.2. CALCUL DE DÉRIVÉES POUR LE TERME DE PRODUCTION DE TURBULENCE 15

- figure 7.1 Décomposition d'un canal en trois zones
- figure 7.2 Exemple de numérotation des points d'un domaine composite et matrice résultante
- figure 7.3 Disposition des composantes de vitesse pour la conservation de la masse.
- figure 7.4 Différents types de numérotation des inconnues u , v et p pour la résolution par blocs.
- figure 7.5 Influence de la partition du domaine en zones sur la propagation de l'erreur dans le domaine de calcul
- figure 7.6 Disposition des inconnues pour un lissage de type PBC
- figure 7.7 Comparaison des courbes de convergence pour deux méthode de lissage: le cas de la cavité
- figure 7.8 Comparaison des courbes de convergence pour deux méthode de lissage: le cas de la marche descendante
- figure 7.9 Champ de vitesse et champ de pression résultant d'un calcul implicite sur la première zone à l'entrée d'un canal plan
- figure 7.10 Disposition des inconnues prises en compte lors d'un calcul implicite pour une zone
-
- figure 8.1 Construction du maillage lache par agglomération unidirectionnelle des éléments du maillage fin
- figure 8.2 Disposition relative, - maillage fin maillage lache - des inconnues du maillage décalé
- figure 8.3 Disposition relative, - maillage fin maillage lache - des inconnues du maillage non-décalé
- figure 8.4 Correspondance des éléments pour une projection vers maillage lache par sommation
- figure 8.5 Interpolation des variables de pression près de la paroi
- figure 8.6 Résultats obtenus par Johanson [36] pour le calcul d'écoulements turbulents par la méthode multigrille.
- figure 8.7 Résultats obtenus par Lien [39] pour le calcul d'écoulements turbulents par la méthode multigrille.
- figure 8.8 Résultats obtenus par Bai [4] pour le calcul d'écoulements turbulents par la méthode multigrille.
-
- figure 9.1 Organigramme du code Niveau 1.
- figure 9.1bis Initialisation de la structure de donnée.
- figure 9.2 Organigramme du code Niveau 2.
- figure 9.3 Organigramme du code Niveau 3.
- figure 9.4 Organigramme du code Niveau 4.
- figure 9.5 Organisation des fichiers de configuration.

- figure 9.6 Calcul de résidus Niveau 1
- figure 9.7 Calcul de résidus Niveau 2
- figure 9.8 Calcul des coefficients du système linéaire
- figure 9.9 Inversion par blocs du système linéaire Niveau 1
- figure 9.10 Inversion par blocs du système linéaire Niveau 2
- figure 9.11 Calcul du résidu multigrille en vu de la projection du problème sur maillage lache.
- figure 9.12 Structure de base pour les données du code. Niveau 1.
- figure 9.13 Structure contenant les données d'un niveau de maillage. Niveau 2.
- figure 9.14 Structure contenant les données d'un sous-domaine. Niveau 3.
- figure 9.15 Structures de bas niveau: maillage, inconnues du problème et description des points du domaine. Niveau 4.
-
- figure 10.1 Conditions aux limites pour l'écoulement de cavité.
- figure 10.2 Influence de la taille de bande sur trois paramètres de calcul pour la méthode par zone (cavité).
- figure 10.3 Courbes de convergence pour la cavité en fonction du temps CPU.
- figure 10.4 Courbes de convergence pour la cavité en fonction du nombre d'itérations.
- figure 10.5 Conditions aux limites pour l'écoulement de marche descendante.
- figure 10.6 Influence de la taille de bande sur trois paramètres de calcul pour la méthode par zone (marche descendante).
- figure 10.7 Courbes de convergence pour la marche descendante en fonction du temps CPU.
- figure 10.8 Courbes de convergence pour la marche descendante en fonction du nombre d'itérations.
- figure 10.9 Influence du nombre de Reynolds sur la convergence de la méthode par zone (cas de la cavité).
- figure 10.10 Influence du nombre de Reynolds sur la convergence de la méthode par zone (cas de la marche descendante).
- figure 10.11 Comparaison des courbes de convergence en fonction du temps CPU pour un calcul par zone avec assemblage, et un calcul par zone par sous-domaine (trois sous-domaines pour le cas de la marche descendante)
- figure 10.12 Subdivision en sous-domaines et maillage d'un canal avec valve anti-retour.
- figure 10.13 Comparaison méthode par zone méthode monogrille

A.2. CALCUL DE DÉRIVÉES POUR LE TERME DE PRODUCTION DE TURBULENCE¹⁵

- pour l'évolution du temps de calcul en fonction du nombre de point de maillages (la cavité).
- figure 10.14 Comparaison méthode par zone méthode monogrille pour l'évolution du temps de calcul en fonction du nombre de point de maillages (la marche descendante).
- figure 10.15 Comparaison pour le calcul d'écoulements turbulents, d'une initialisation multigrille avec une initialisation par le modèle de turbulence K-L (temps CPU)
- figure 10.16 Comparaison pour le calcul d'écoulements turbulents, d'une initialisation multigrille avec une initialisation par le modèle de turbulence K-L (nombre d'itération)
- figure 10.17 Influence du nombre de niveaux de maillages sur l'erreur de projection interpolation. Cas de l'énergie turbulente dans un écoulement de type Couette.
-
- figure 11.1 Conditions aux limites pour l'écoulement de canal plan. Séparation en trois sections pour le calcul d'écoulements de Couette-Poiseuille
- figure 11.2 Profil de vitesse établi pour l'écoulement de poiseuille pur laminaire (comparaison avec la solution exacte).
- figure 11.3 Champ de vitesse (cavité)
- figure 11.4 Iso-pression (cavité)
- figure 11.5 Comparaison des profile de vitesse $u(y)$ au centre de la cavité avec des résultats de Benchmark.
- figure 11.6 Comparaison des profile de vitesse $u(y)$ au centre de la cavité pour trois schémas de précision.
- figure 11.7 Champ de vitesse (marche descendante).
- figure 11.8 Iso-pression (marche descendante).
- figure 11.9 Comparaison de profils de vitesse pour l'écoulement de type Poiseuille.
- figure 11.10 Comparaison de profils d'énergie turbulente pour l'écoulement de type Poiseuille.
- figure 11.11 Etablissement de la vitesse u pour l'écoulement de type Poiseuille.
- figure 11.12 Comparaison de profils de vitesse pour l'écoulement de type Intermédiaire.
- figure 11.13 Comparaison de profils d'énergie turbulente pour l'écoulement de type Intermédiaire.
- figure 11.14 Comparaison de profils de vitesse pour l'écoulement de type Couette.
- figure 11.15 Comparaison de profils d'énergie turbulente pour l'écoulement de type Couette.

- figure 11.16 Evolution longitudinale de l'énergie turbulente pour l'écoulement de marche descendante.
- figure 11.17 Etablissement de la vitesse u pour l'écoulement de marche descendante.
- figure 11.17bis Etablissement de la vitesse u pour l'écoulement de marche descendante (suite).
- figure 11.18 Représentation de l'épaisseur de la couche de diffusion du zinc en fonction de l'abscisse x pour $V_b = -2$
- figure 11.19 Représentation de l'épaisseur de la couche de diffusion du zinc en fonction de l'abscisse x pour $V_b = 0$
- figure 12.1 Champ de vitesse d'un écoulement de canal plan avec valve anti-retour
- figure 12.2 Champ de vitesse d'un écoulement de canal plan avec valve anti-retour zoom sur la zone centrale
- figure 12.2 Iso-pression d'un écoulement de canal plan avec valve anti-retour



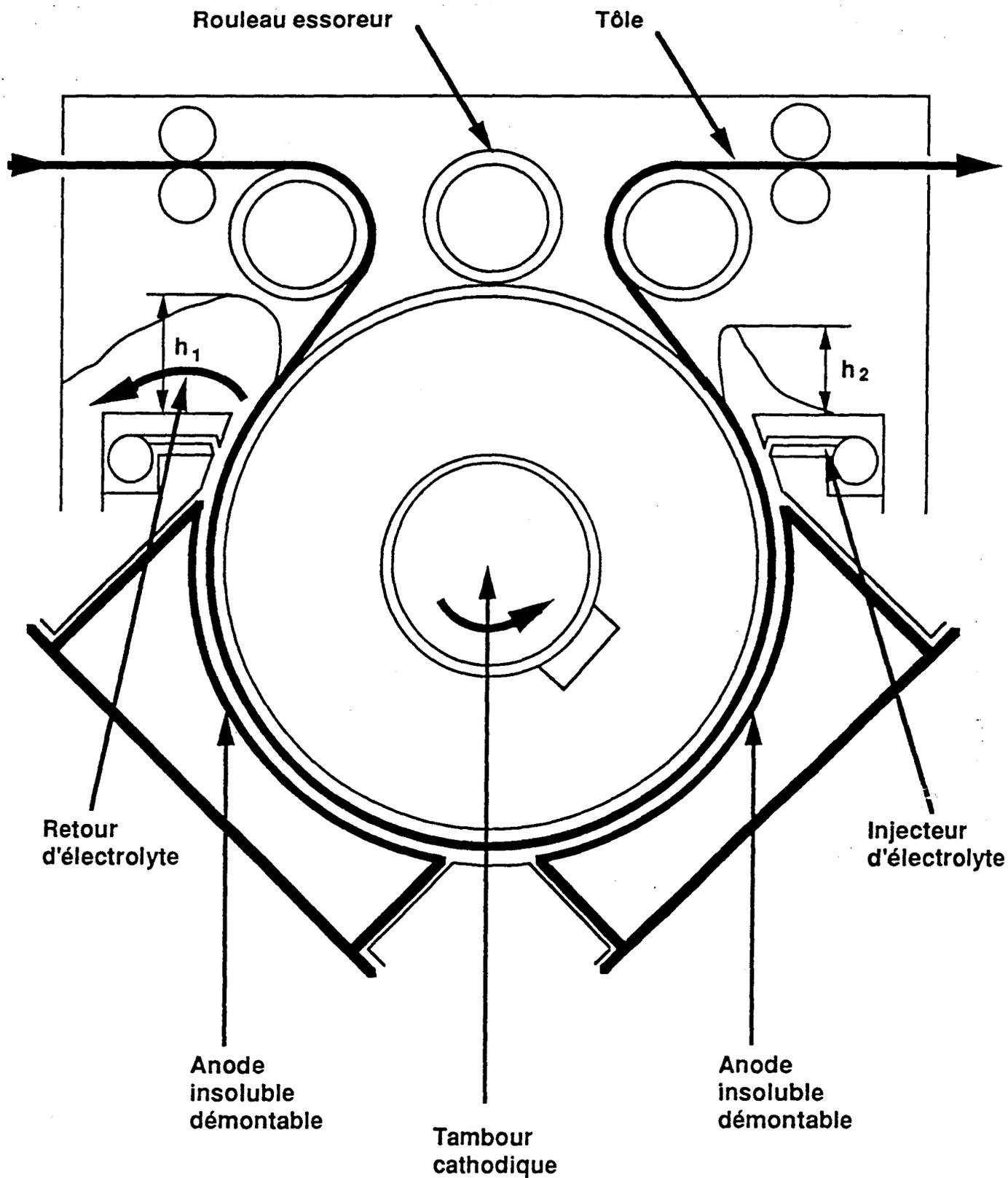


Figure 1.1: Schéma de principe d'une cellule d'électrozingage à hydrodynamique contrôlée de type Beator

○	○	○	○						
○	○	○	○						
○	○	○	○						
○	○	○	○						

figure 4.1: Maillage global avec désactivation de points

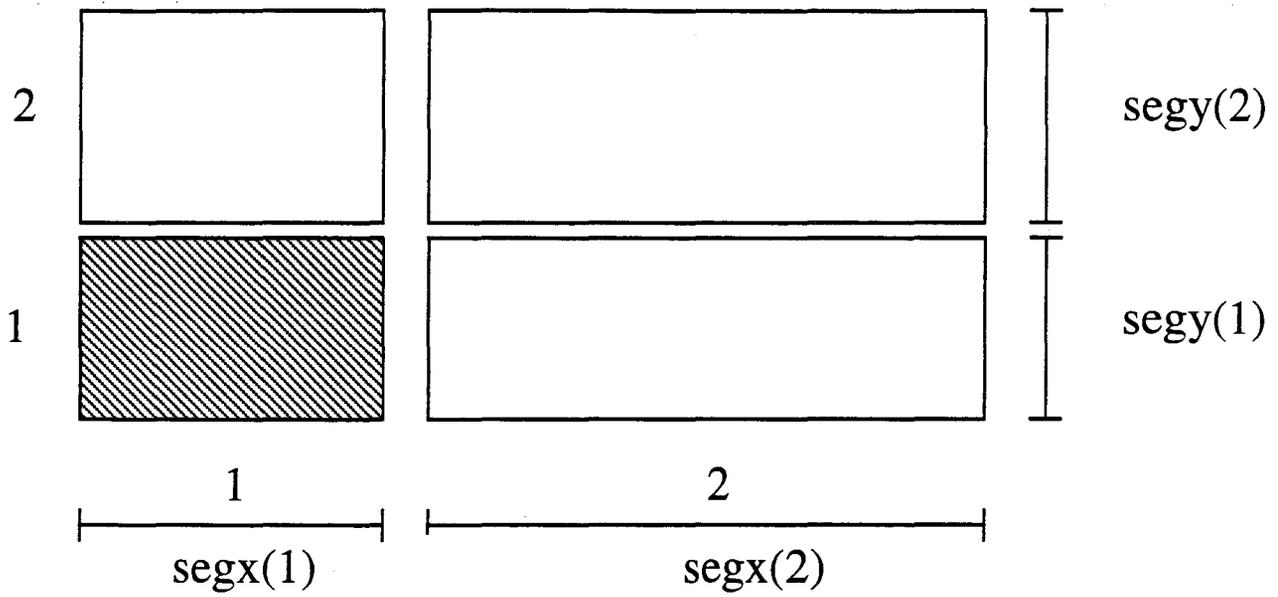


figure 4.2: Décomposition d'un écoulement de marche en quatre rectangles dont un inactif et segments de maillages correspondants

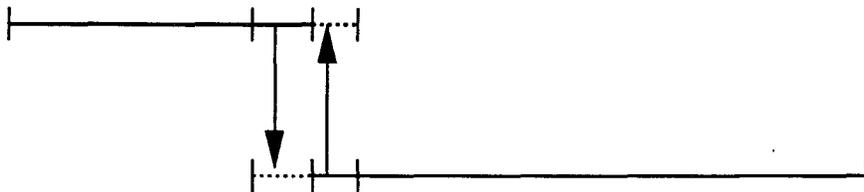


figure 4.3: Raccordement de maillage pour deux rectangles voisins

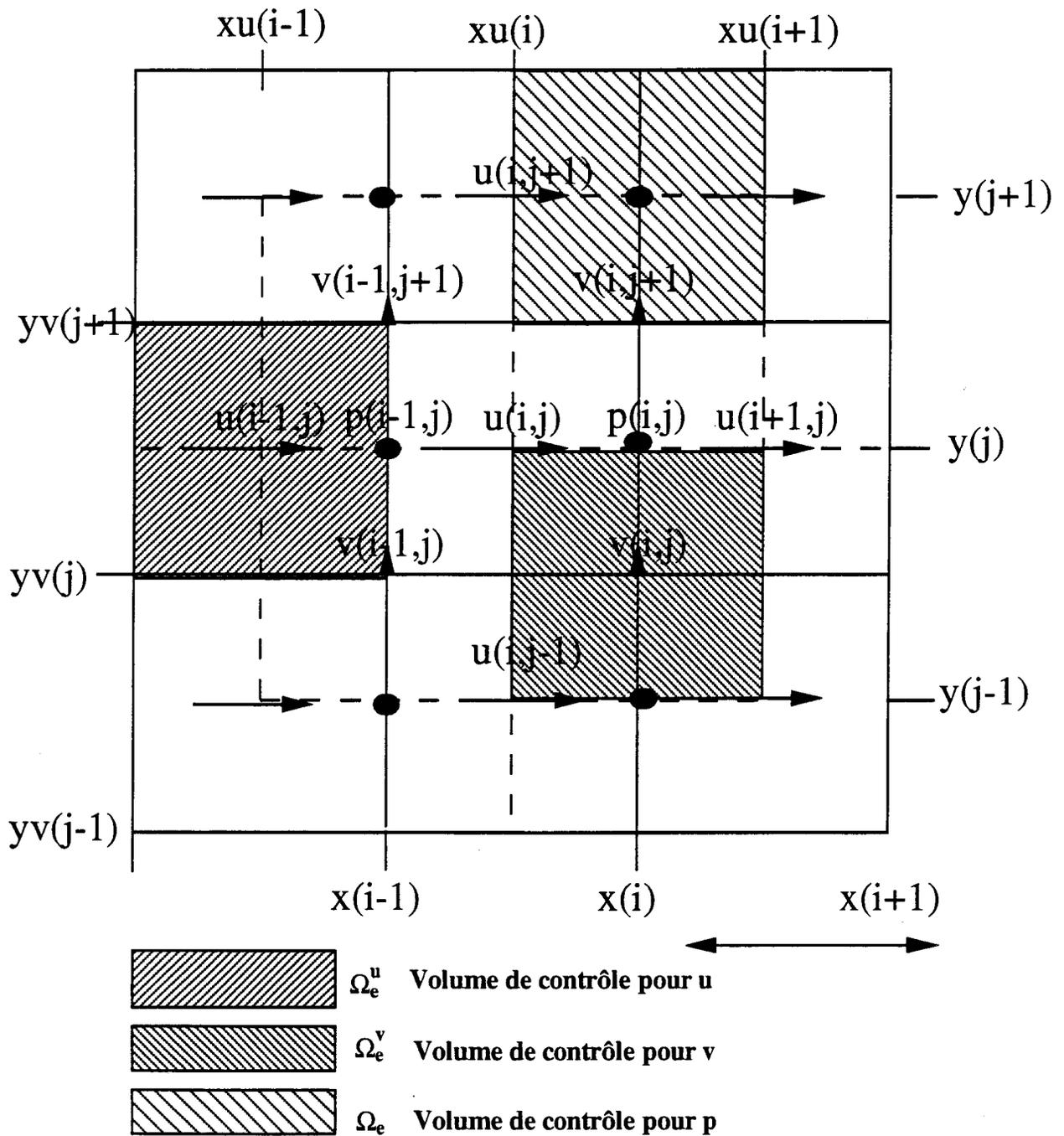


figure 5.1: Description du maillage décalé et positionnement des inconnues.

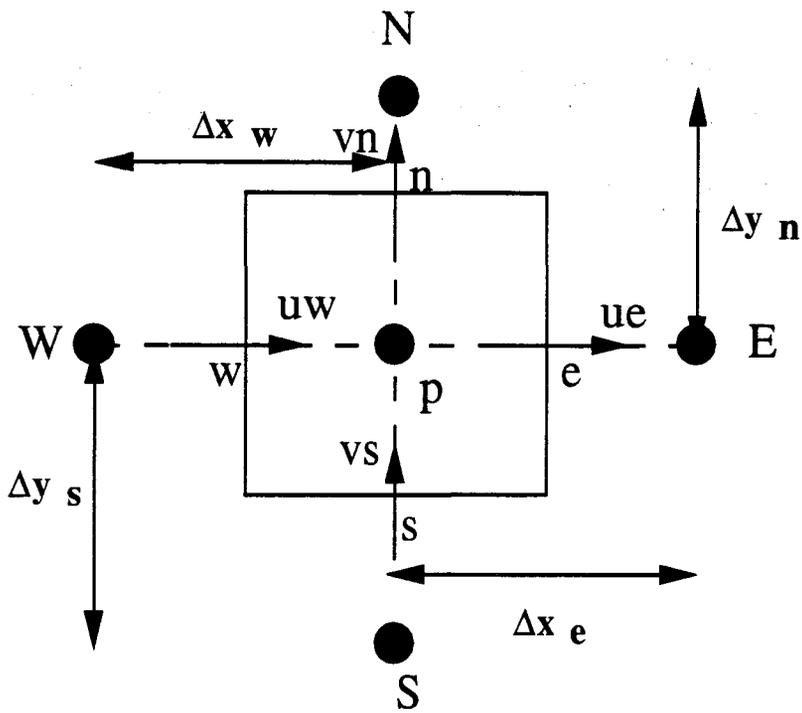


figure 5.2: Pas de maillage pour la discrétisation d'une équation.

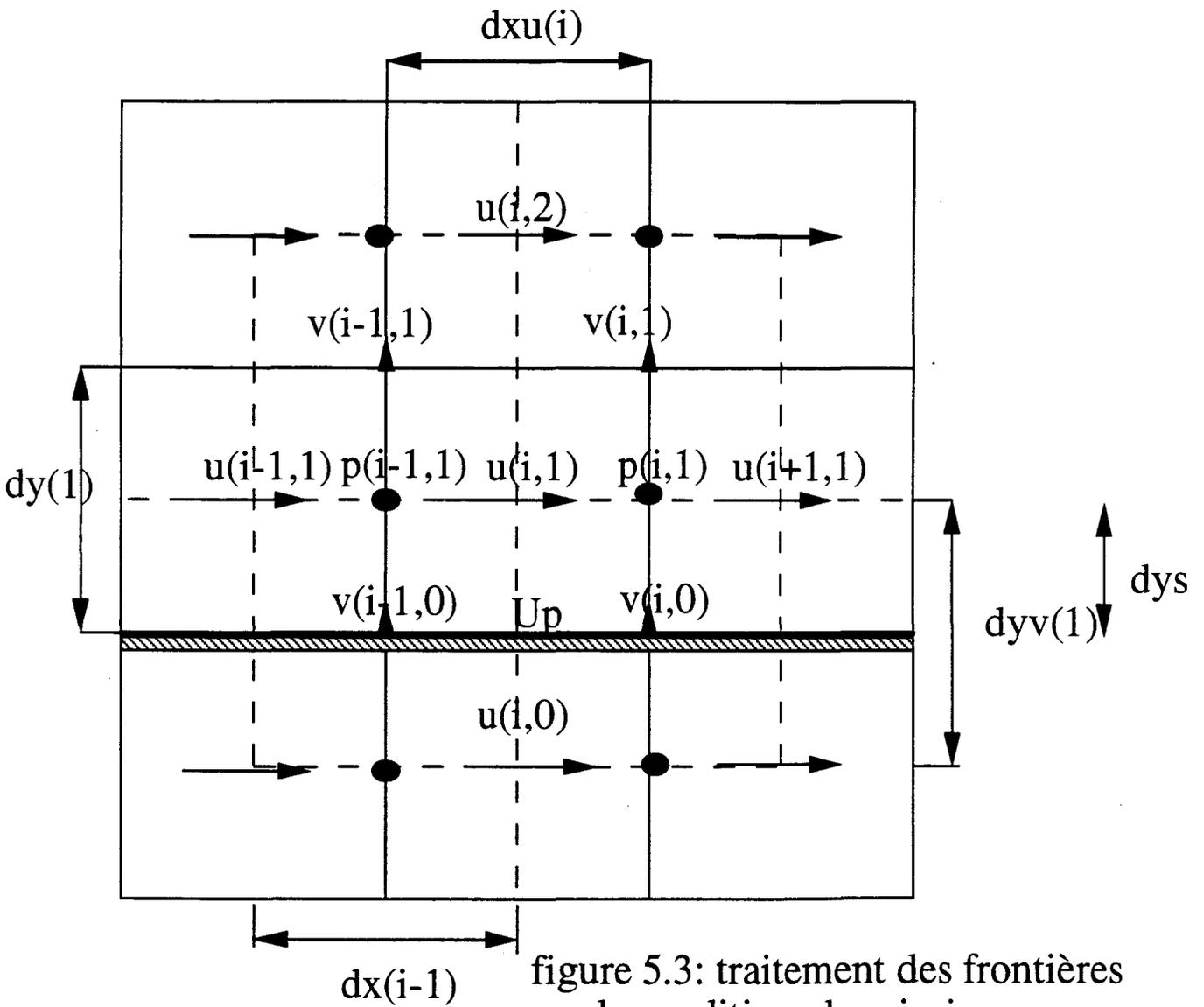


figure 5.3: traitement des frontières par la condition de miroir

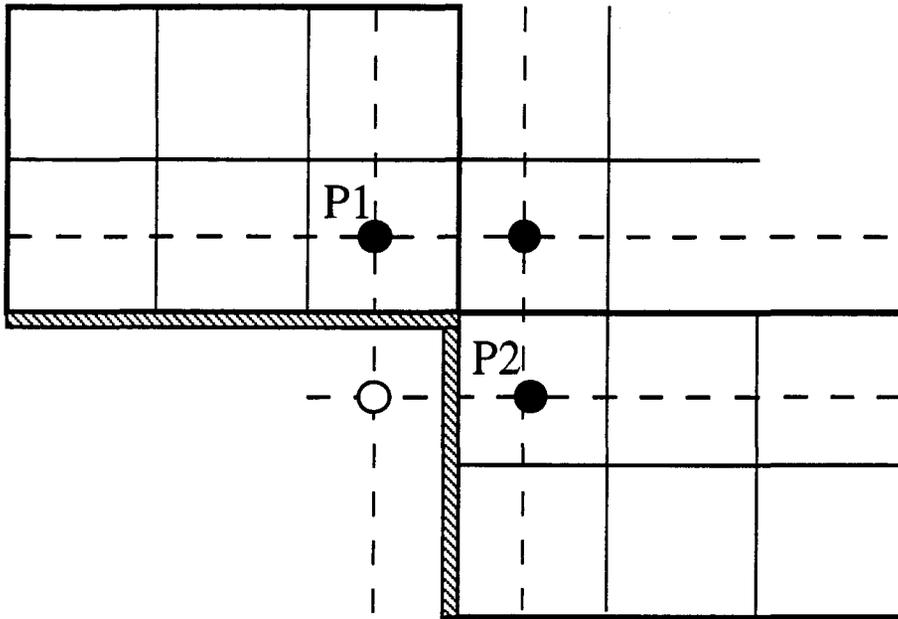


figure 5.4: Problème posé par la condition de miroir

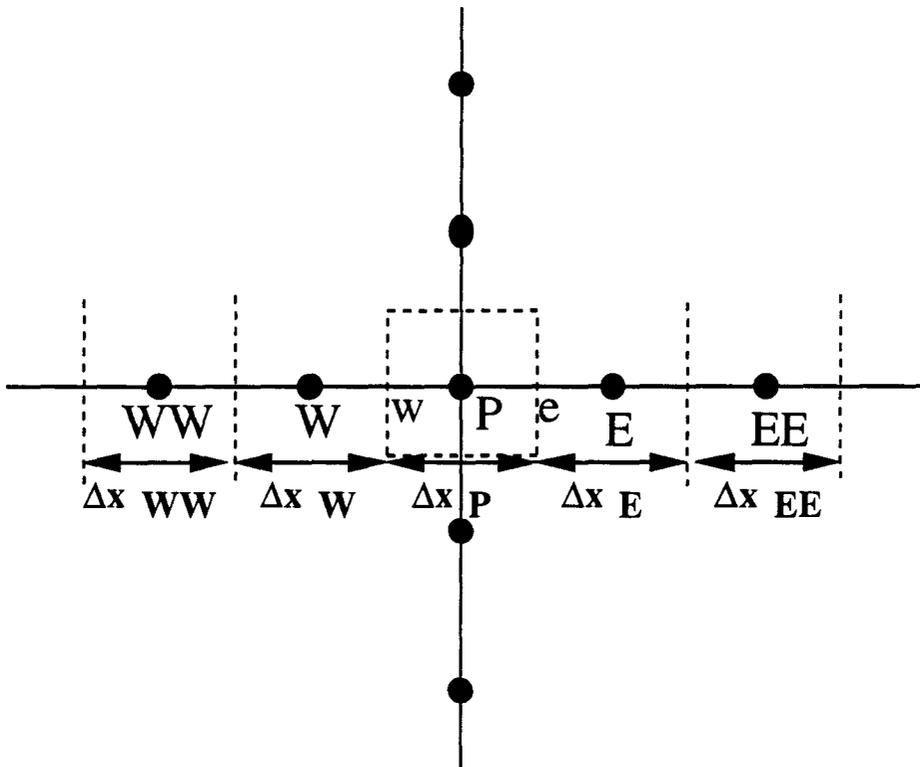
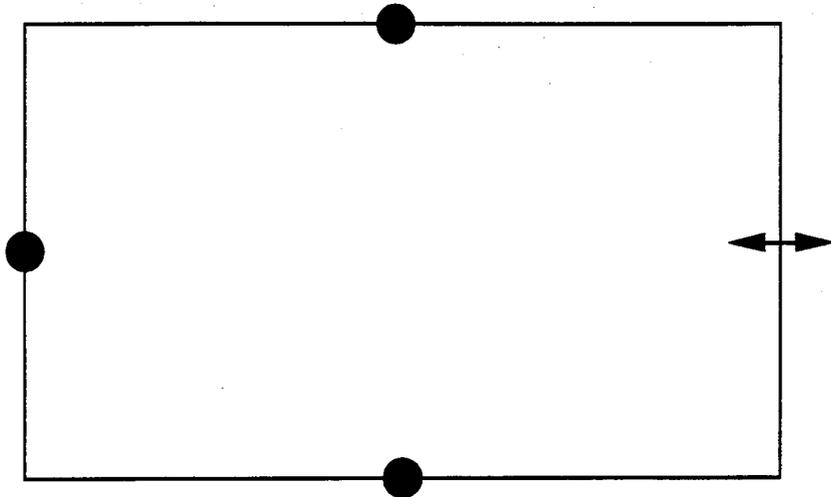
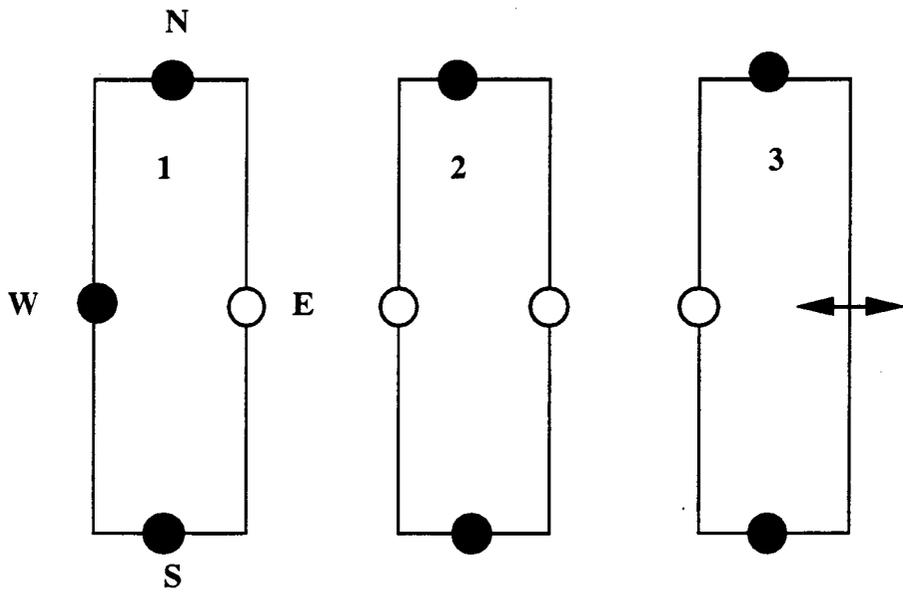


figure 5.5: Pas de maillage pour l'estimation du terme convectif par le schéma QUICK



Ecoulement de Canal



Décomposition en trois zones

● Conditions aux limites de type Dirichlet

○ Conditions aux limites de type pseudo-Dirichlet (limite inter-zone)

↔ Conditions aux limites de type Neumann

figure 7.1: Décomposition en zones d'un écoulement de canal

2	4	8	12
1	3	7	11
0	0	6	10
0	0	5	9

figure 7.2 a

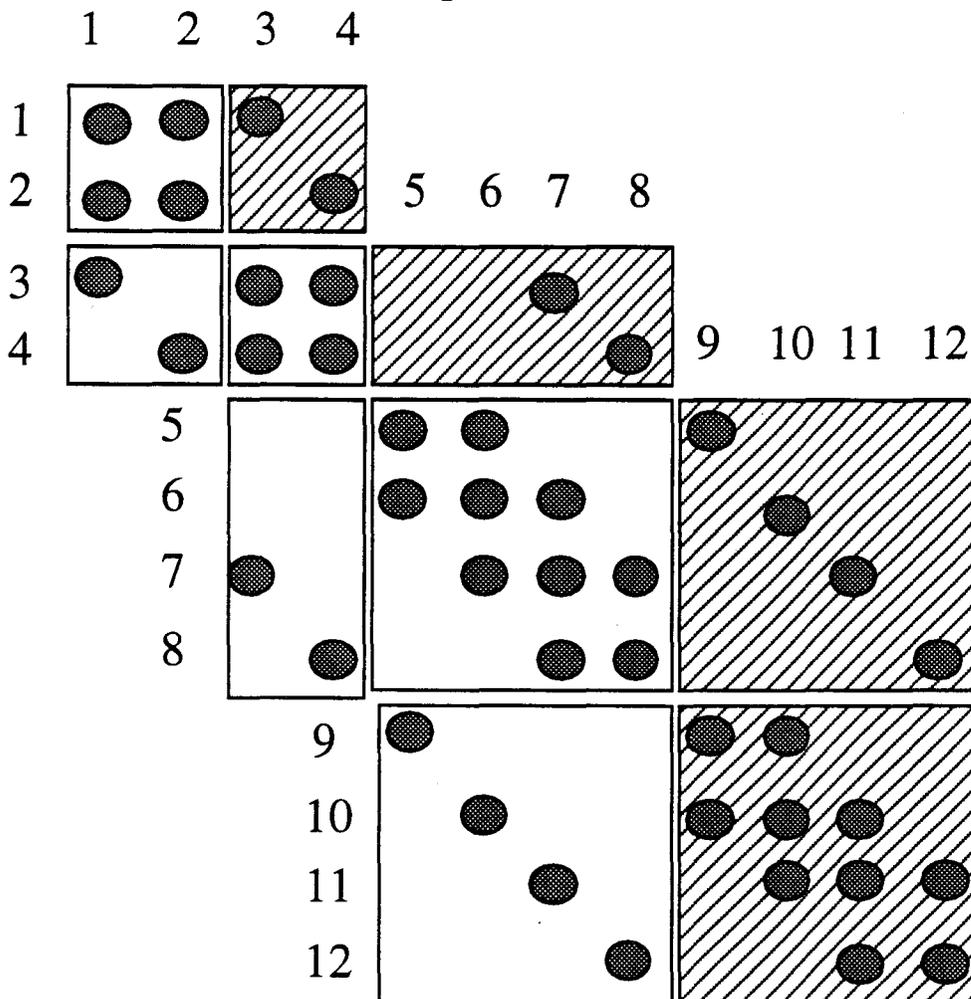


figure 7.2 b

figure 7.2 Exemple de numérotation des points d'un domaine composite et matrice correspondante

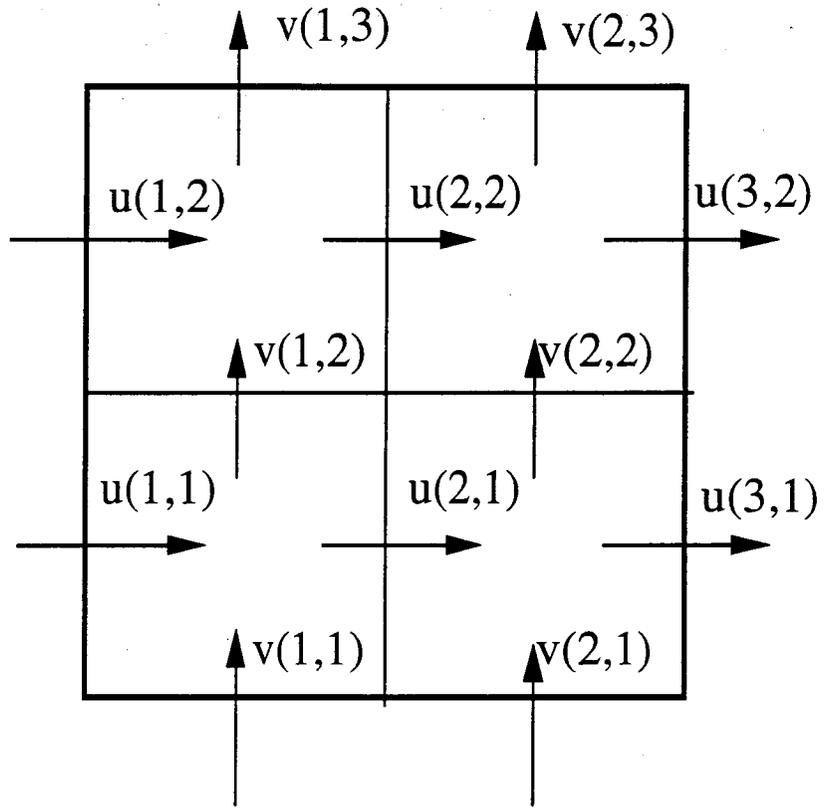


figure 7.3: Disposition des composantes de vitesse pour la conservation de la masse

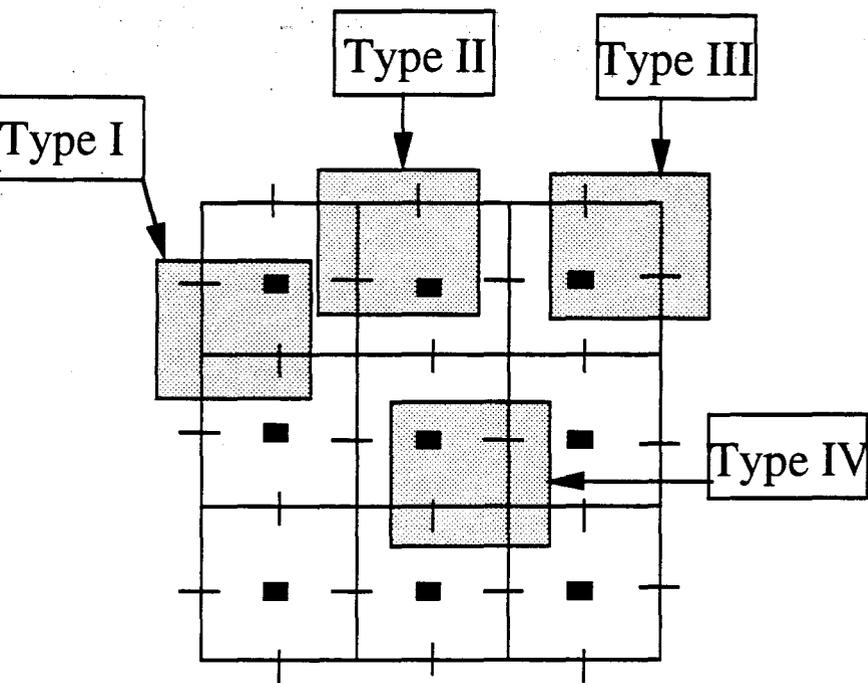
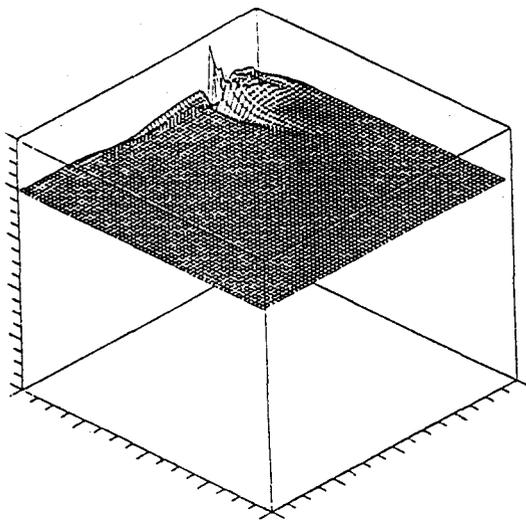
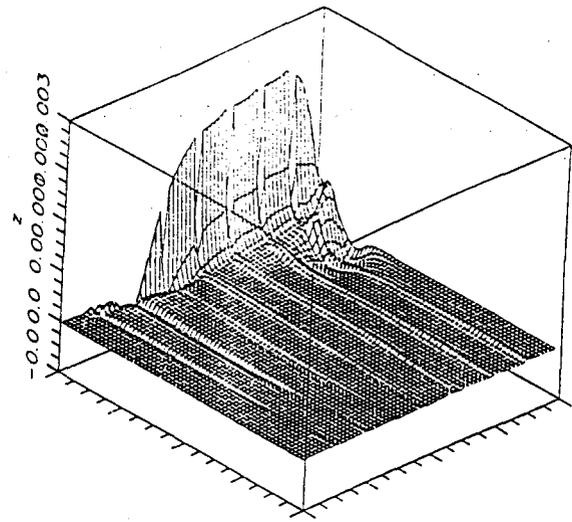


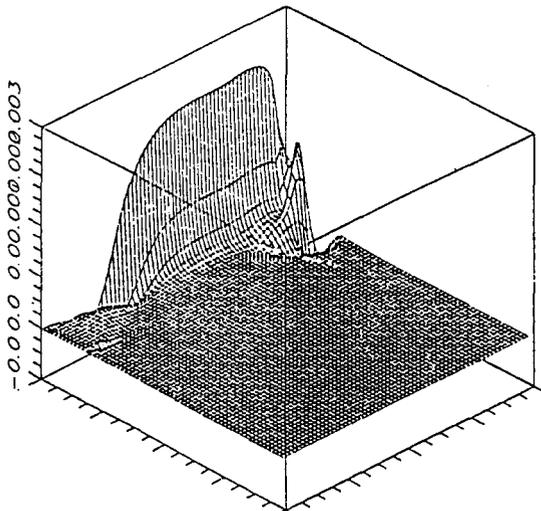
figure 7.4: Différents types de numérotation pour la résolution par bloc



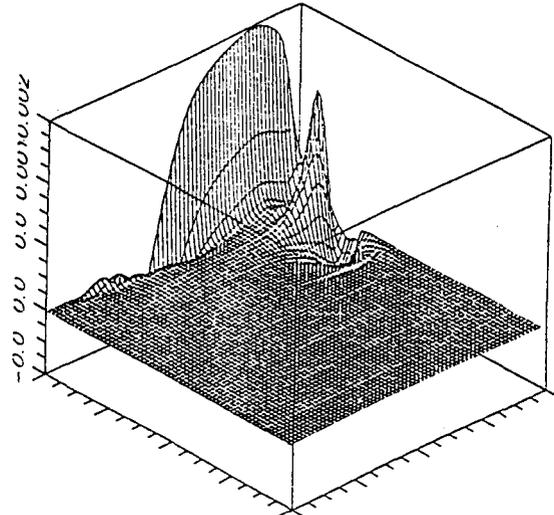
7.5a. 1 bande en x



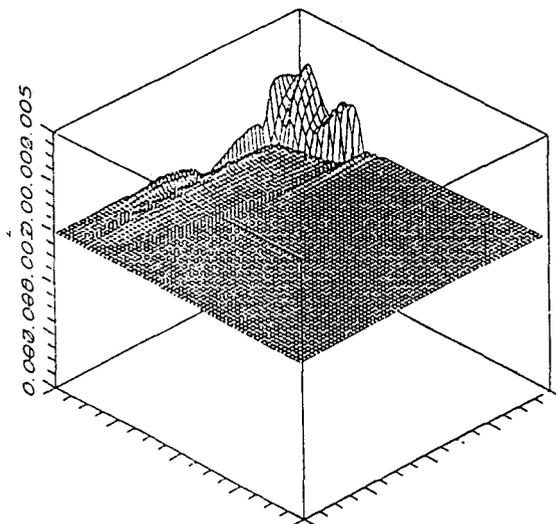
7.5b. 8 bandes en x



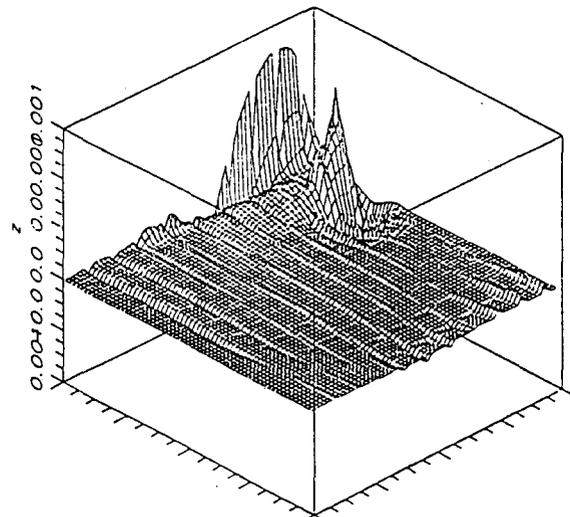
7.5c. 1 bande en y



7.5d. 8 bandes en y



7.5e. explicite



7.5f. blocs 6x6

figure 7.5: influence de la partition du domaine sur la propagation des erreurs

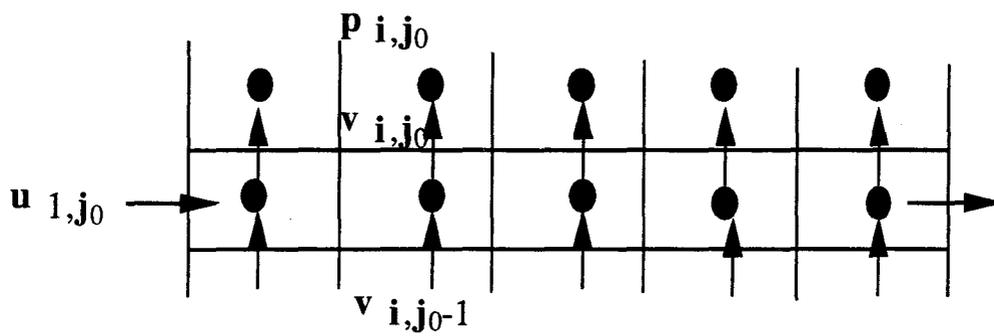


figure 7.6: Lissage PBC

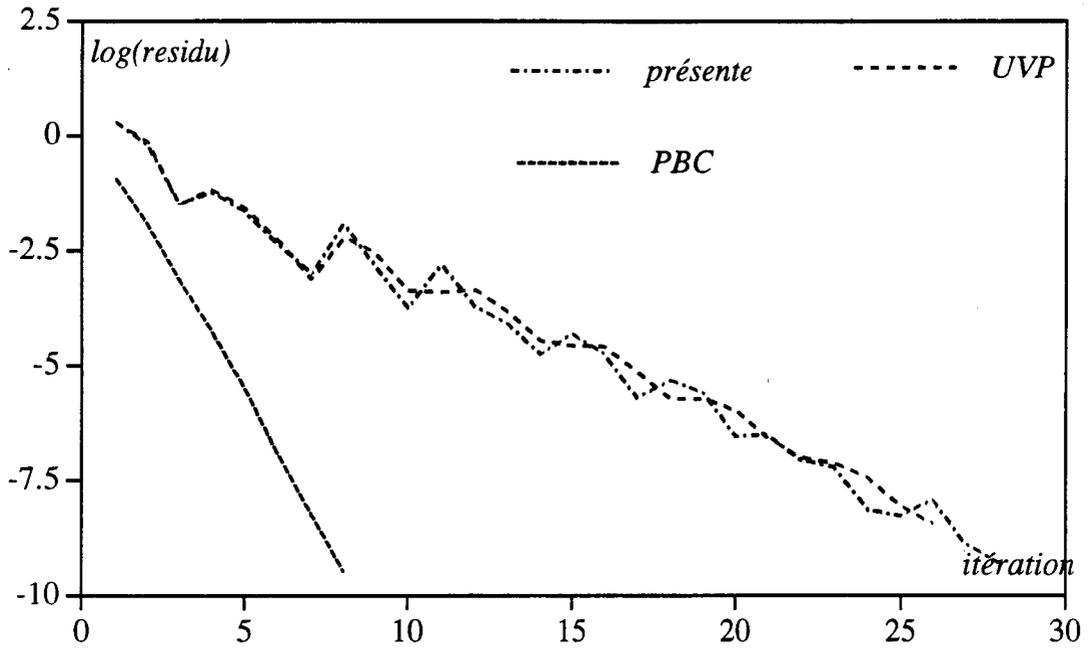


figure 7.7a: Courbes de convergence cavité
 $Re = 100$ maillage 40×40

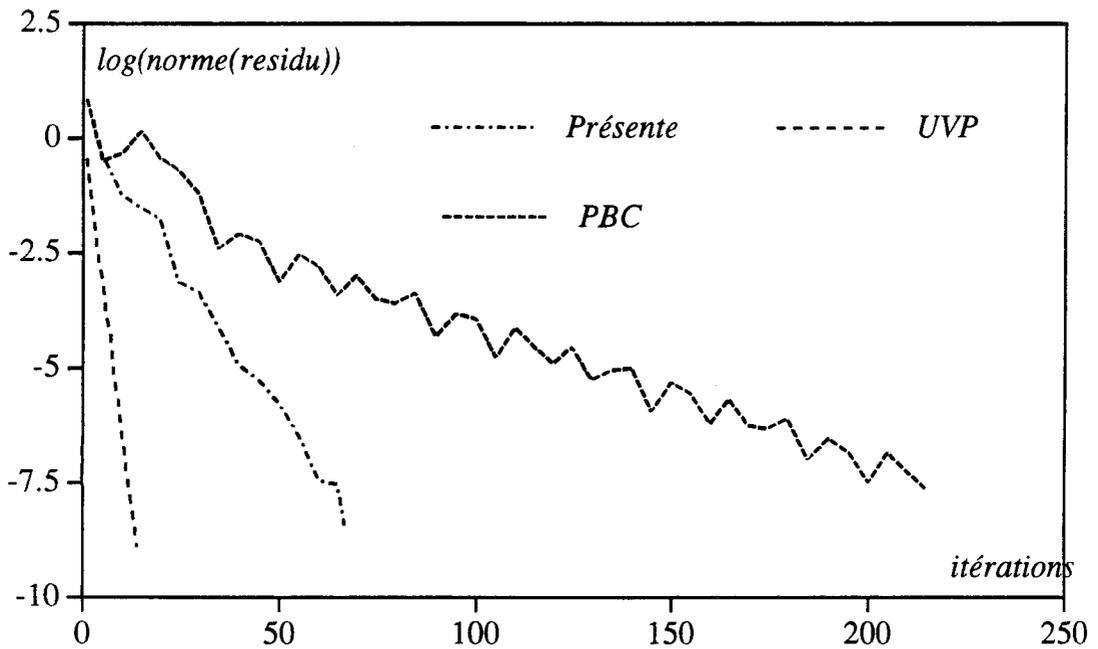
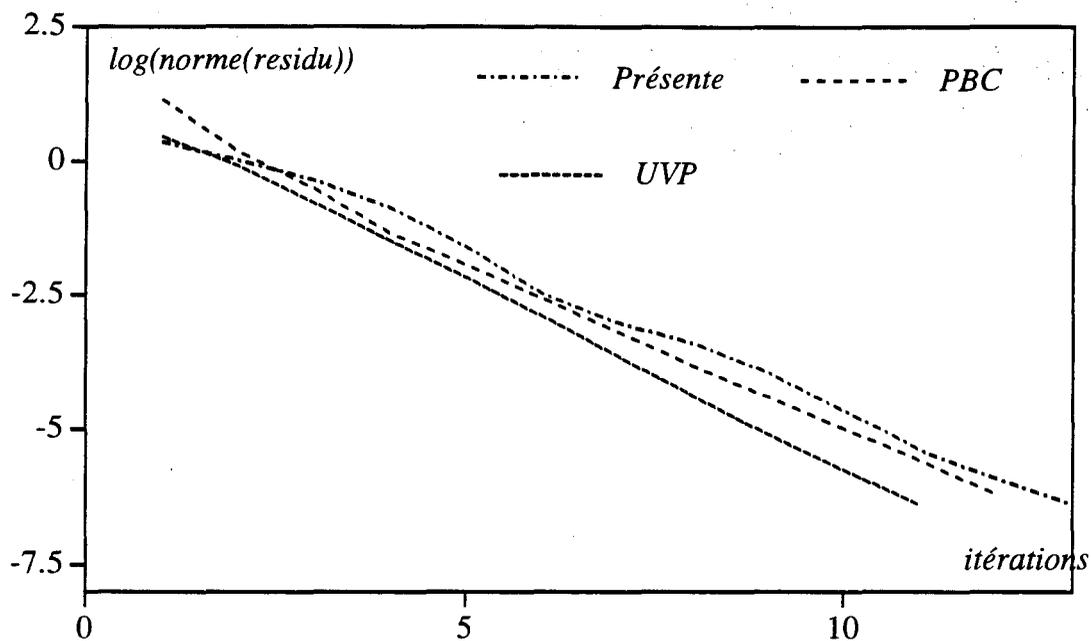
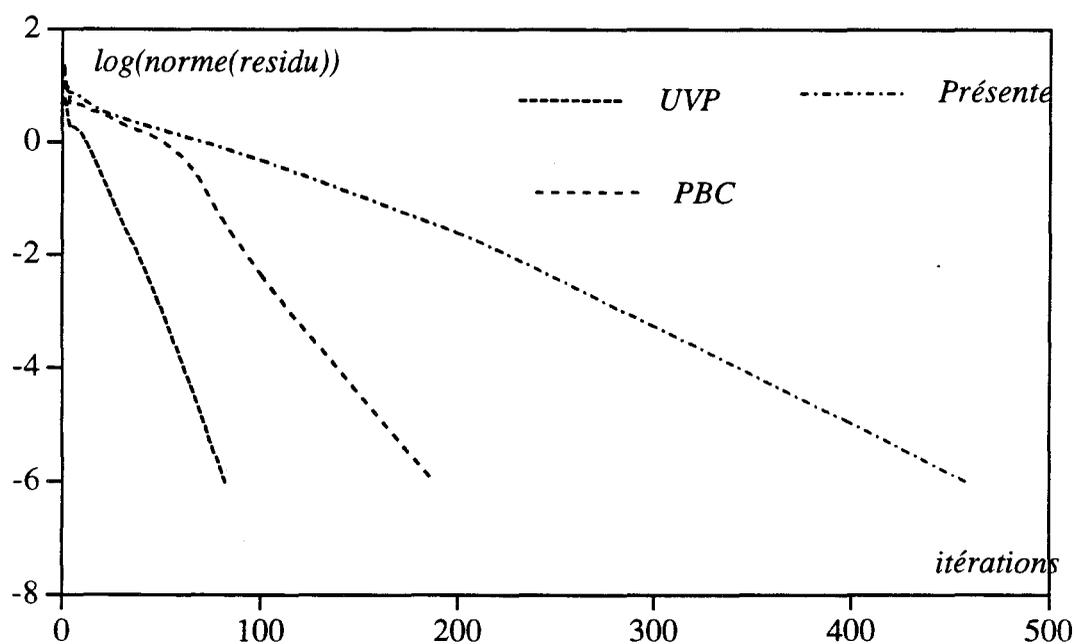


figure 7.7b Courbe de convergence cavité
 $Re = 1000$ maillage 40×40

Figure 7.7: Comparaison de courbes de convergence pour deux méthodes de lissage pour la cavité.



**figure 7.8a: Courbes de convergence pour la marche
Re = 100 maillage 60x32**



**figure 7.8b: Courbes de convergence pour la marche
Re = 1000 maillage 60x32**

Figure 7.8: Comparaison de courbes de convergence pour deux méthodes de lissage pour la marche descendante.

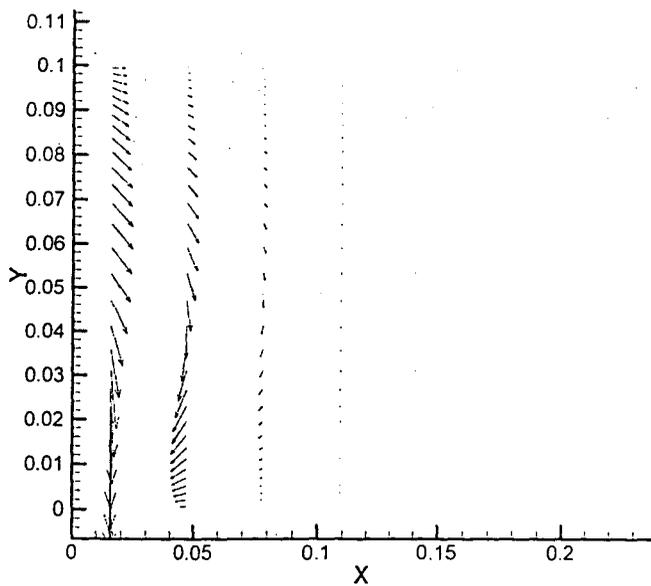


figure 7.9a: Méthode par zones avec numérotation du type I

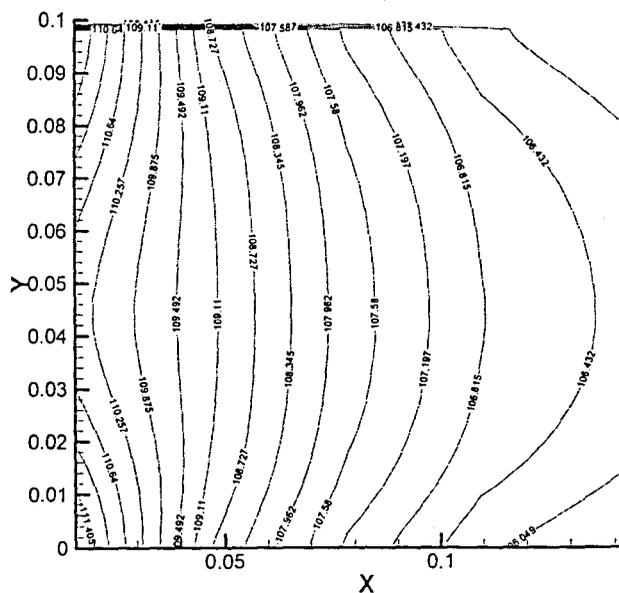
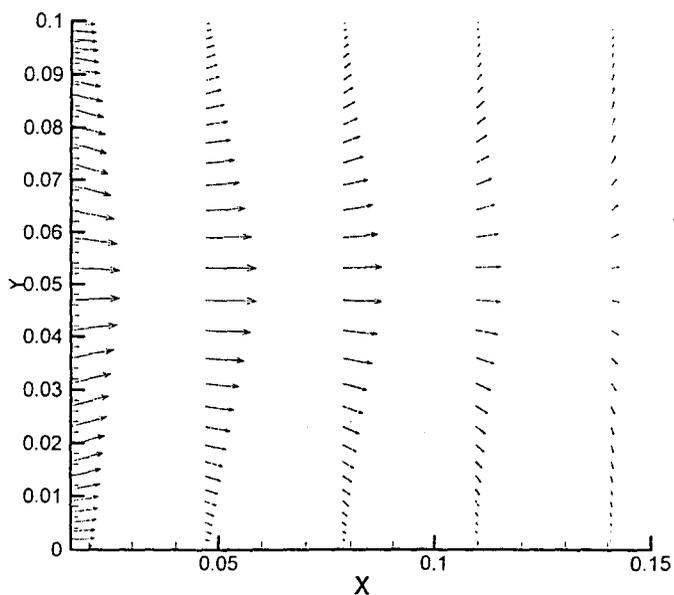


figure 7.9b: Méthode par zones avec numérotation de type I et utilisation d'un coefficient de pénalité

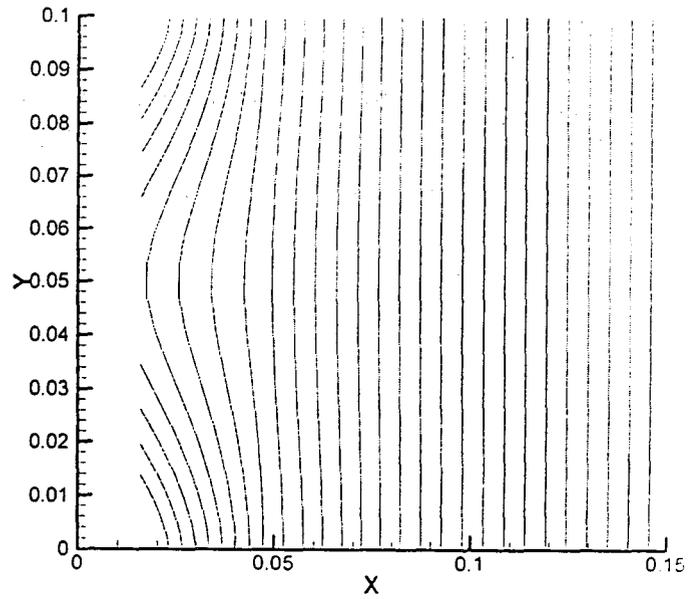
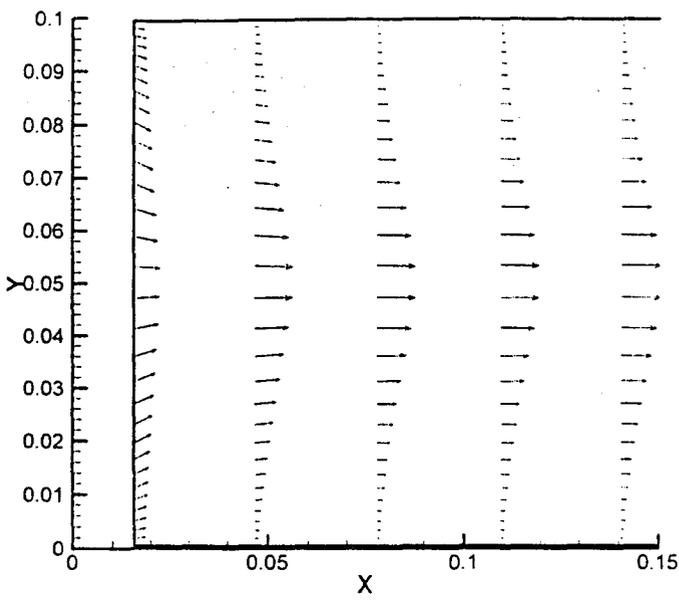


figure 7.9c: Méthode par zones avec numérotation du type IV

figure 7.9: Champ de vitesse et champ de pression résultant du calcul implicite sur une la première zone à l'entrée d'un canal plan

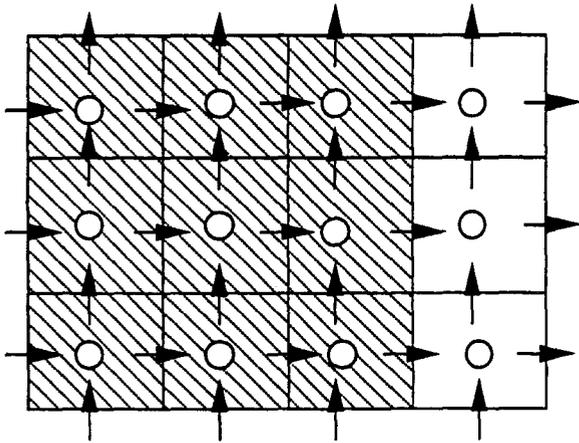


figure 7.10a: Domaine complet

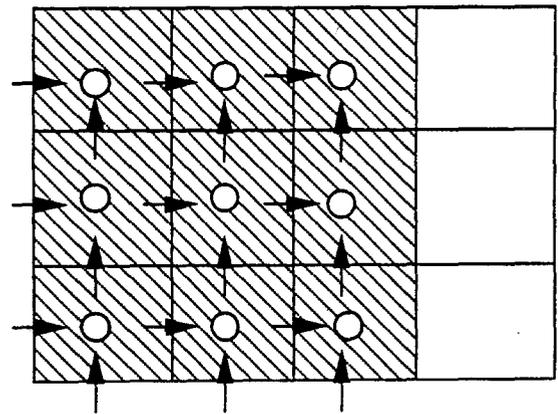


figure 7.10b: zone numérotation I

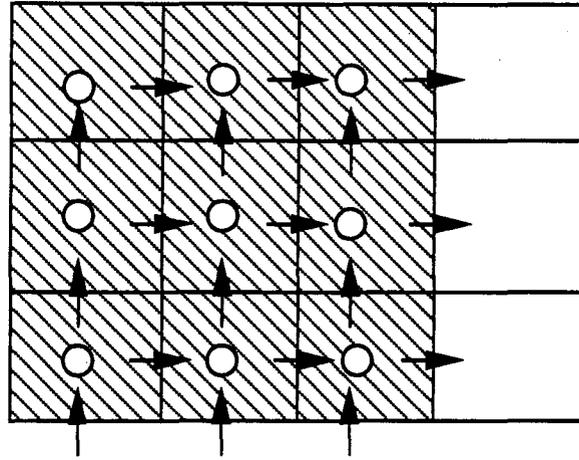
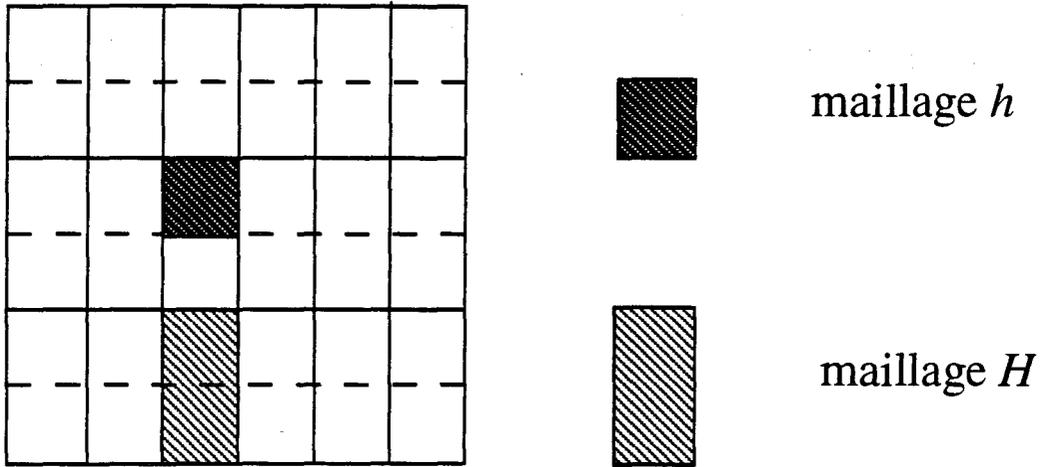
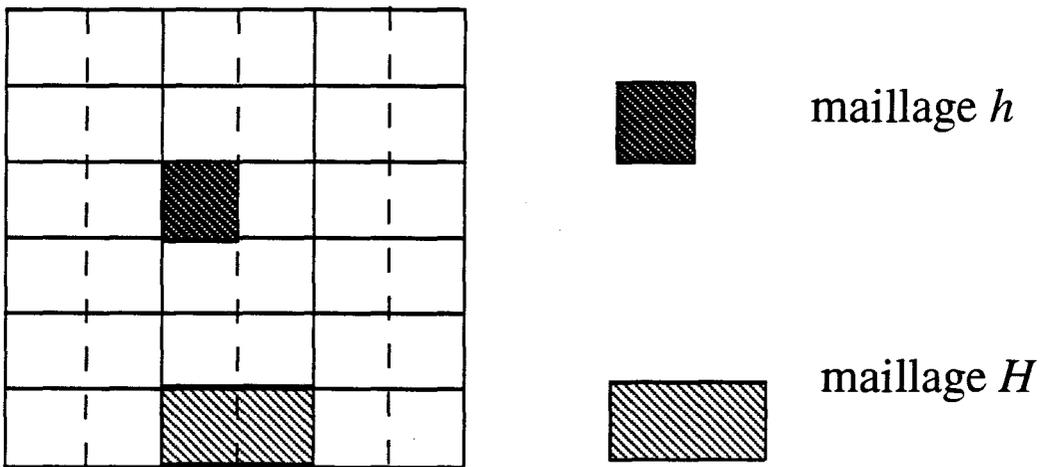


figure 7.10c: Zone avec numérotation du type IV

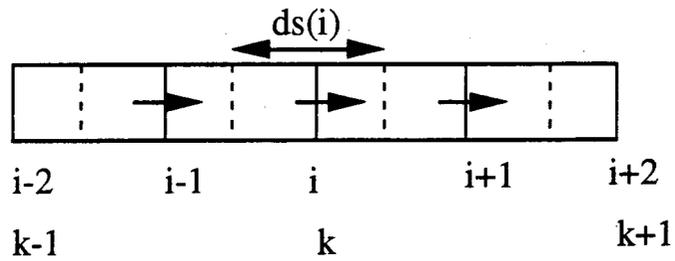


Maillage lâche par regroupement suivant y



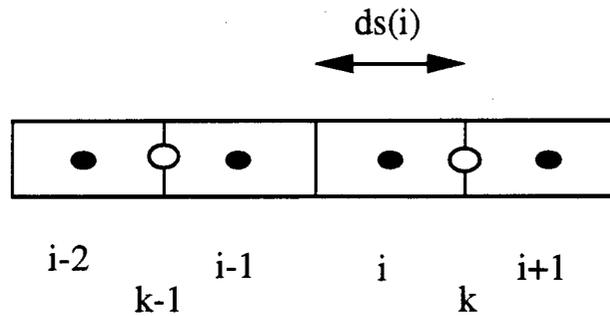
Maillage lâche par regroupement suivant x

figure 8.1: Construction du maillage lâche par agglomération unidirectionnelle.



variables U et V

figure 8.2: Disposition des inconnues décalées
maillage fin (i) maillage lâche (k)



variables P, K, E et CZn

figure 8.3: Disposition des inconnues non-décalées

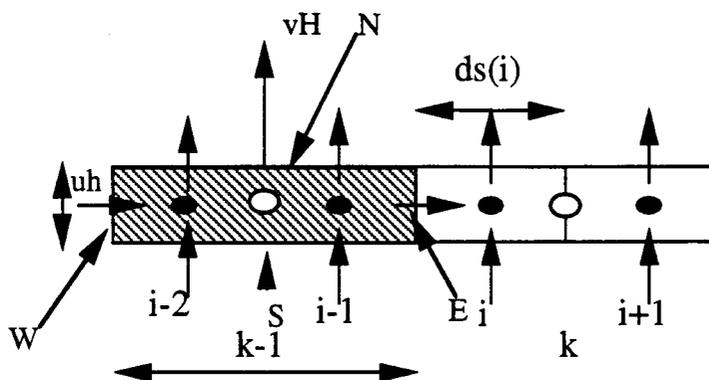


figure 8.4: Projection par sommation

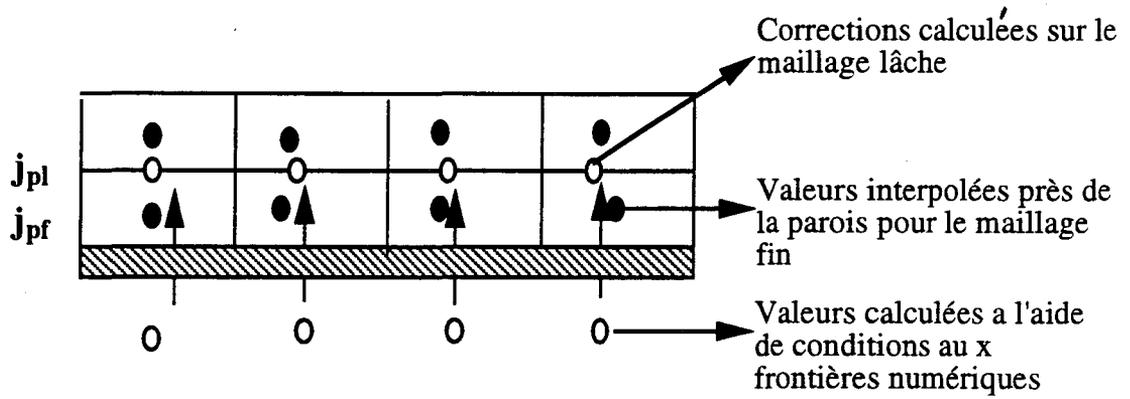


figure 8.5: Interpolation de la variable de pression près de la paroi.



FIGURE 1. 20x10 grid for the backwards facing step

MODEL	LAMINAR		LOW-RE $k - \epsilon$			
RE	100		110 000			
SCHEME	HYBRID		HYBRID		QUICK + VAN-LEER	
	WU	SPEEDUP	WU	SPEEDUP	WU	SPEEDUP
20x10	67	1.0	185	1.0	215	1.0
40x20	58	1.9	108	3.3	130	3.1
80x40	44	6.4	72	10.8	82	∞
160x80	60	15.6	72	23.7	76	∞
320x160	91	37.5	63	100*	73	∞

TABLE 2. Convergence data for backwards facing step

4.1 2D BACKWARDS FACING STEP

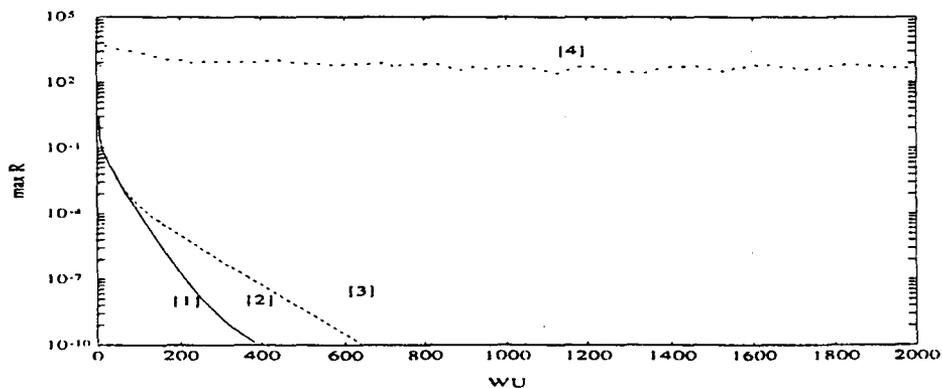


FIGURE 2. Convergence history of the 320x160 calculations for the 2D backwards facing step. [1]: Hybrid,turbulent,5-level FMG. [2]: QUICK,turbulent,5-level FMG. [3]: Hybrid,laminar,5-level FMG. [4]: Hybrid,turbulent,single grid.

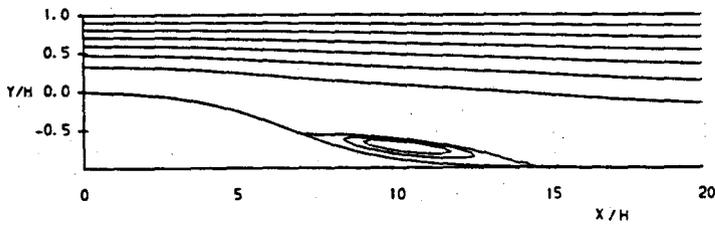
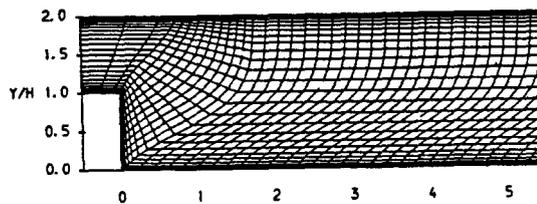


Fig. 4: Case 1: laminar flow in plane constriction

GRID	HYBRID					QUICK			
	FMG-V		FMG-R			FMG-V			
	sg/mg CPU	WU	sg/mg WU	sg/mg CPU	WU	sg/mg WU	sg/mg CPU	WU	sg/mg WU
32x8	1.0	4.8	4.6	1.1	6.0	3.7	1.0	4.8	5.2
64x16	5.9	5.0	16.4	6.5	4.5	18.0	5.7	5.0	17.0
128x32	15.3	2.0	144.0	16.6	4.5	64.0	15.5	2.0	158.5
256x64	64.9	2.0	552.5	70.8	3.5	315.7	63.6	2.0	558.5

GRID	QUICK		MUSCL						
	FMG-R		FMG-V			FMG-R			
	sg/mg CPU	WU	sg/mg WU	sg/mg CPU	WU	sg/mg WU	sg/mg CPU	WU	sg/mg WU
32x8	1.1	6.3	4.0	1.1	4.8	5.4	1.1	6.0	4.3
64x16	6.5	4.5	18.9	6.1	5.0	17.0	7.0	4.5	18.9
128x32	17.9	3.5	90.6	16.7	2.0	158.0	18.2	3.5	90.3
256x64	70.2	3.5	319.1	68.3	2.0	558.5	74.4	3.5	319.1

Table 1: MG performance for Case 1: 2D plane constriction



CV	High-Re $k-\epsilon$				RSM			
	SG	V-CYCLE			SG	V-CYCLE		
	WU	CPU sg/mg	WU	WU sg/mg	WU	CPU sg/mg	WU	WU sg/mg
40x8	135	1.22	71.2	1.90	153	1.74	49.3	3.11
80x16	390	2.60	83.9	4.65	446	2.65	96.5	4.62
120x24	790	3.59	126.1	6.26	808	3.27	149.0	5.42
160x32	1329	5.44	139.3	9.54	1094	3.44	191.0	5.72

Table 4: MG performance for case 4: 2D backward-facing step

Figure 8.7: Résultats tirés de l'article de F.-S. Lien

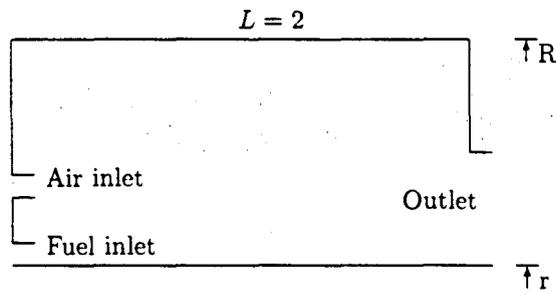


FIGURE 1. A sketch of the model annular gas turbine combustion chamber

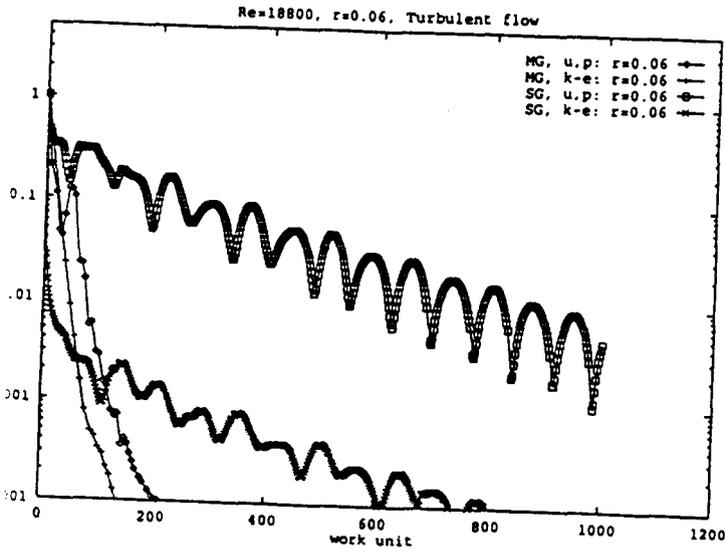


FIGURE 5. Convergence history of *turbulent* flow calculations

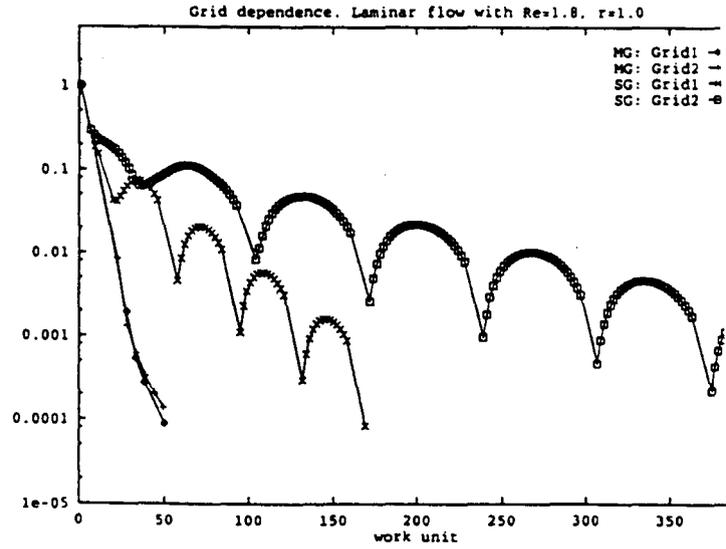


FIGURE 2. Convergence history for Grid 1 and Grid 2

S. Bai and L. Fuchs

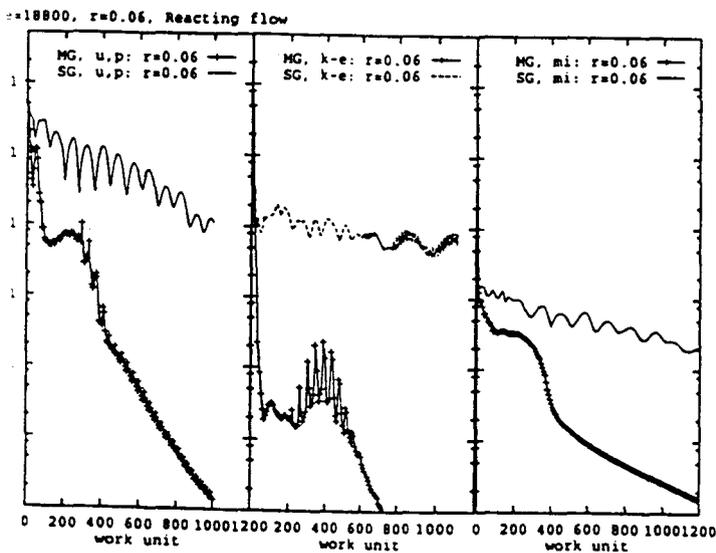


FIGURE 7. Convergence history for *turbulent reacting* flows at $r = 0.06$

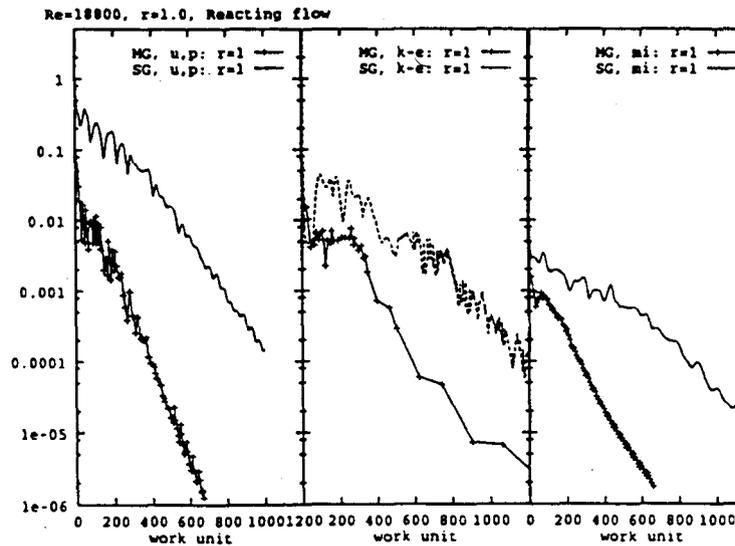


FIGURE 6. Convergence history for *turbulent reacting* flows at $r = 1.0$

Figure 8.8: Résultats tirés de l'article de X.S. Bai

Code

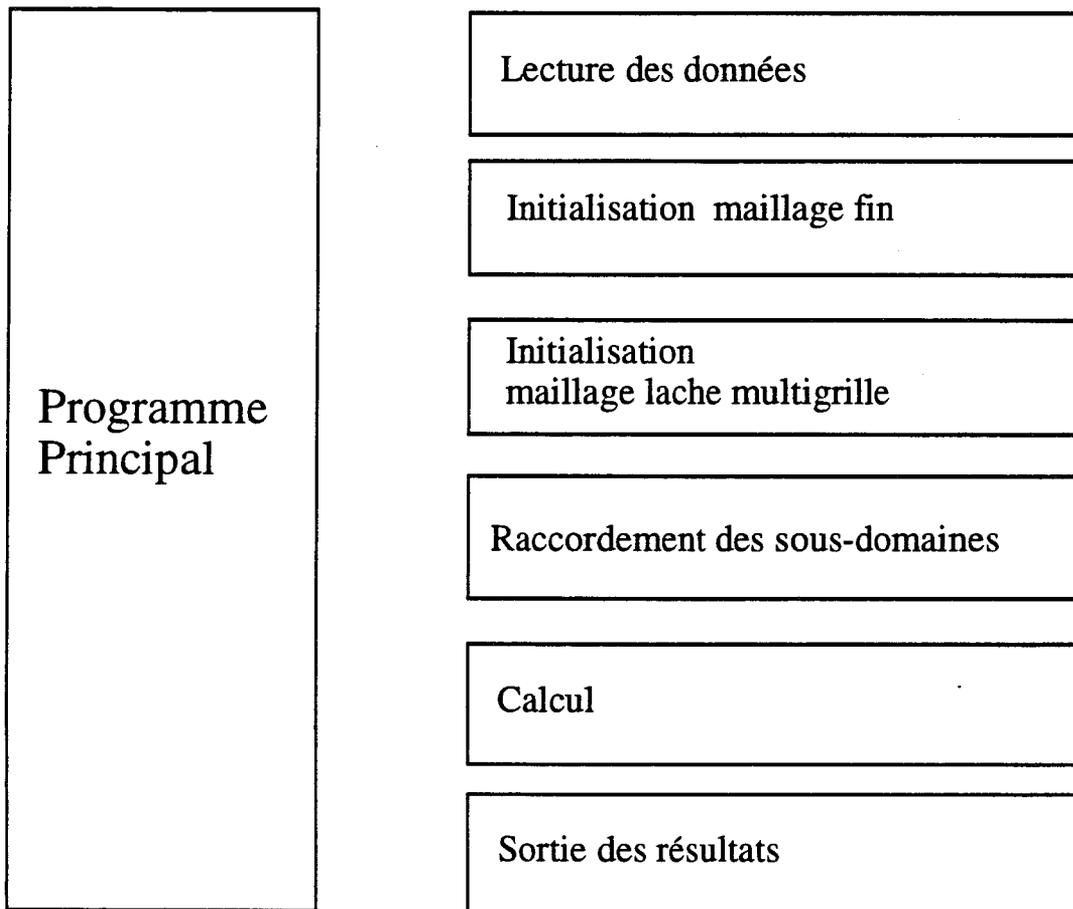


figure 9.1 Organigramme du code premier niveau

Calcul

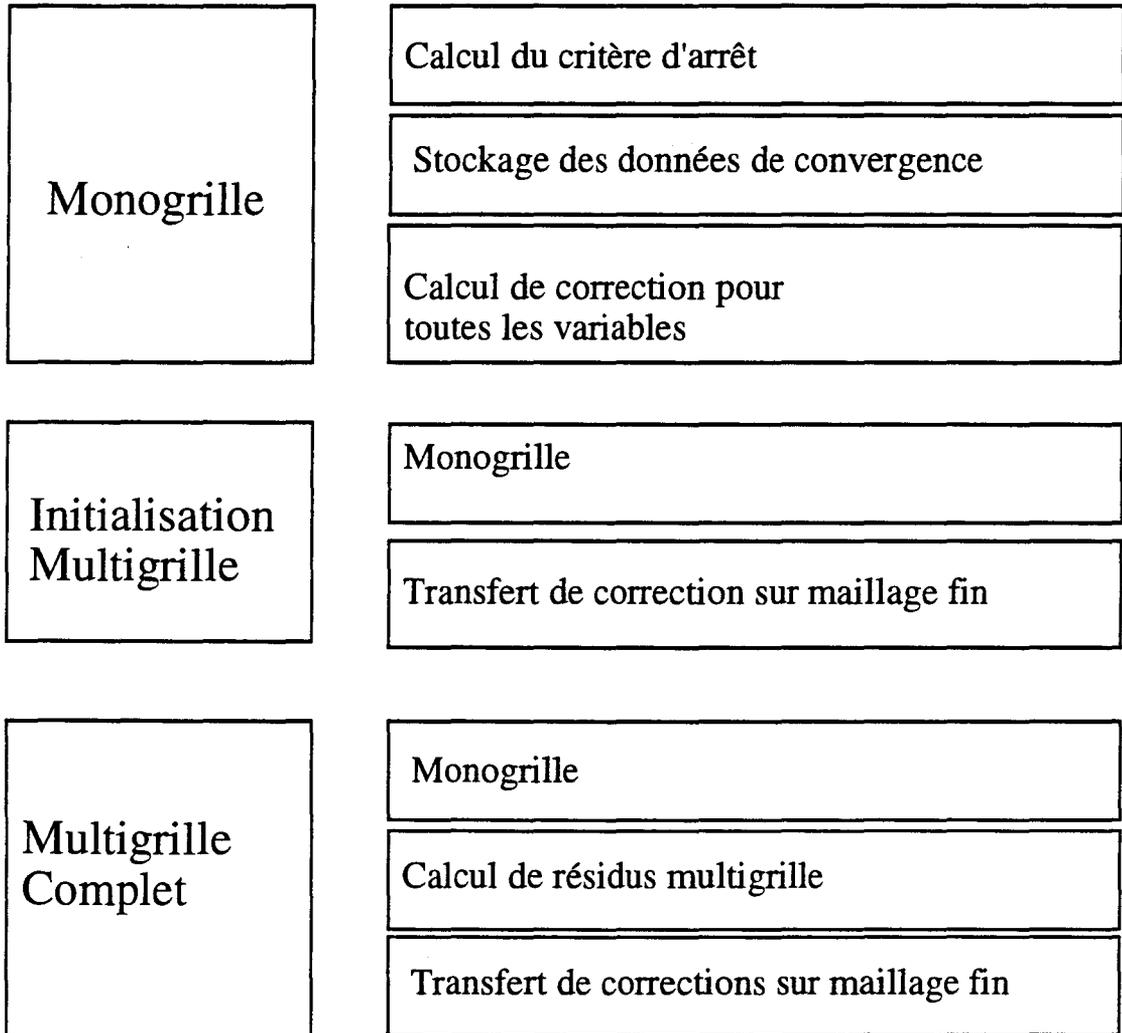


figure 9.2 Organigramme du code deuxième niveau

Calcul de correction pour toutes les variables

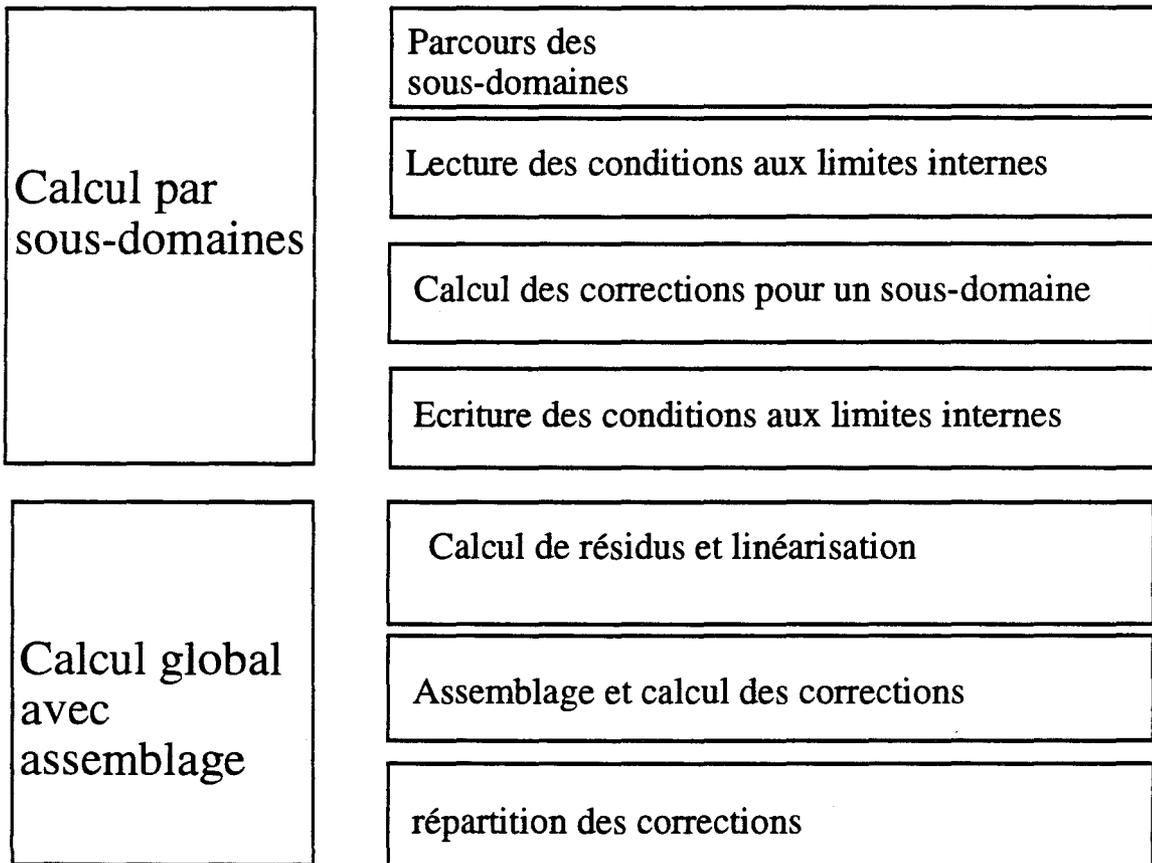


figure 9.3 Organigramme du code troisième niveau

Calcul de correction pour toutes les variables (suite)

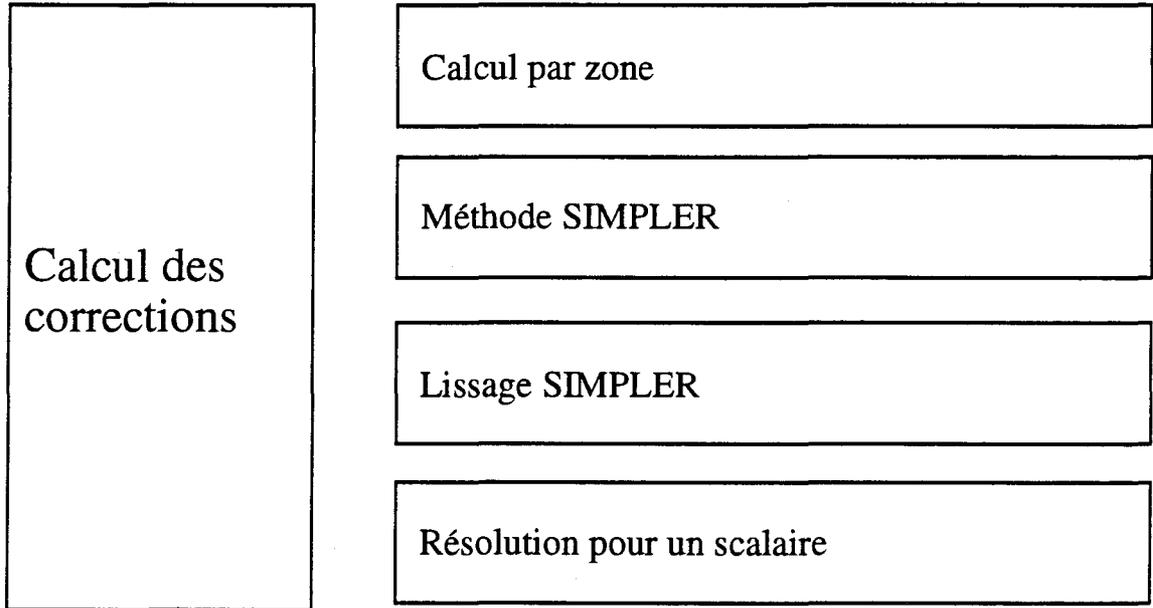


figure 9.4 Organigramme du code quatrième niveau

Fichiers de données

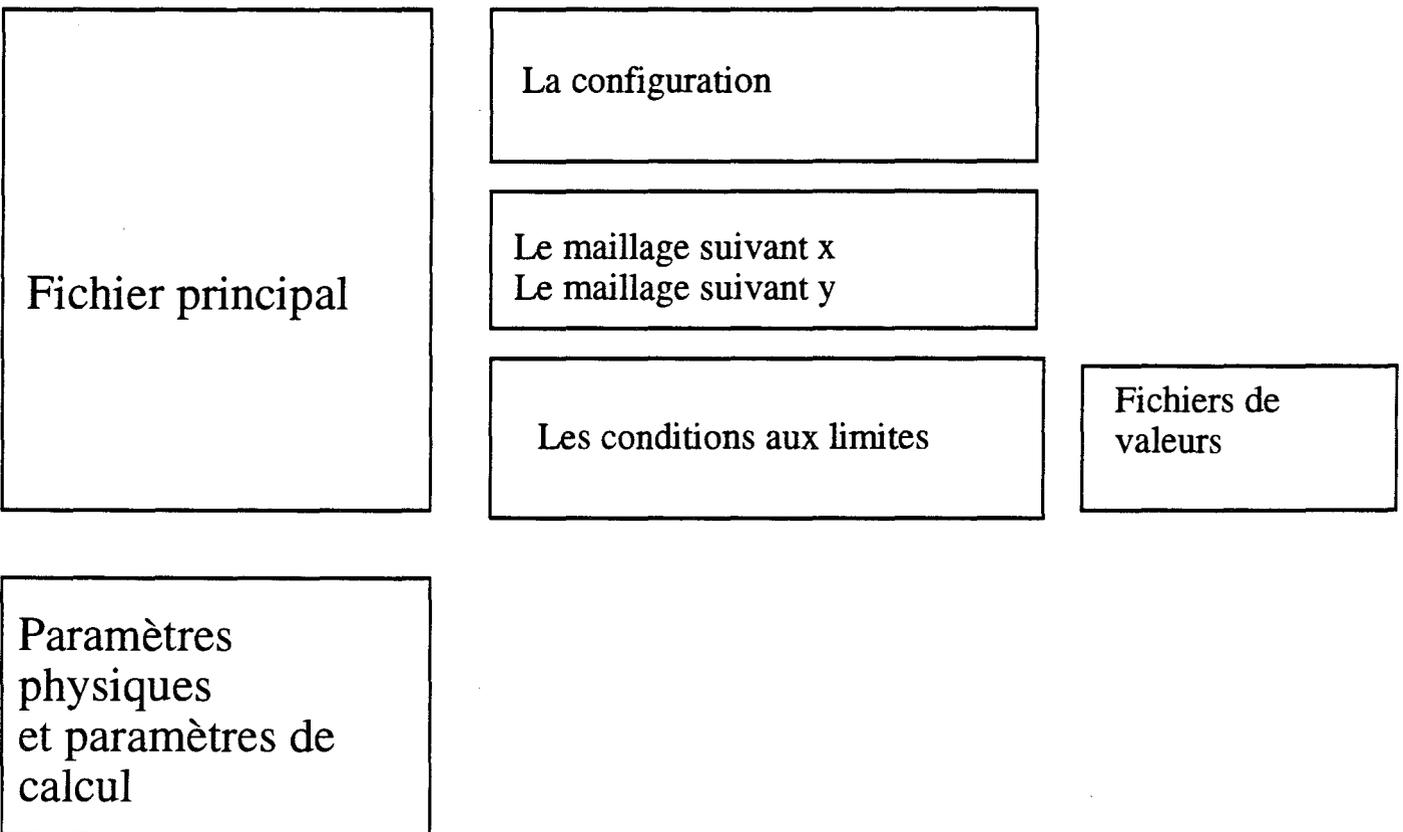


figure 9.5 Organisation des fichier de données pour une configuration de calcul.

Calcul de résidus pour les points d'une zone

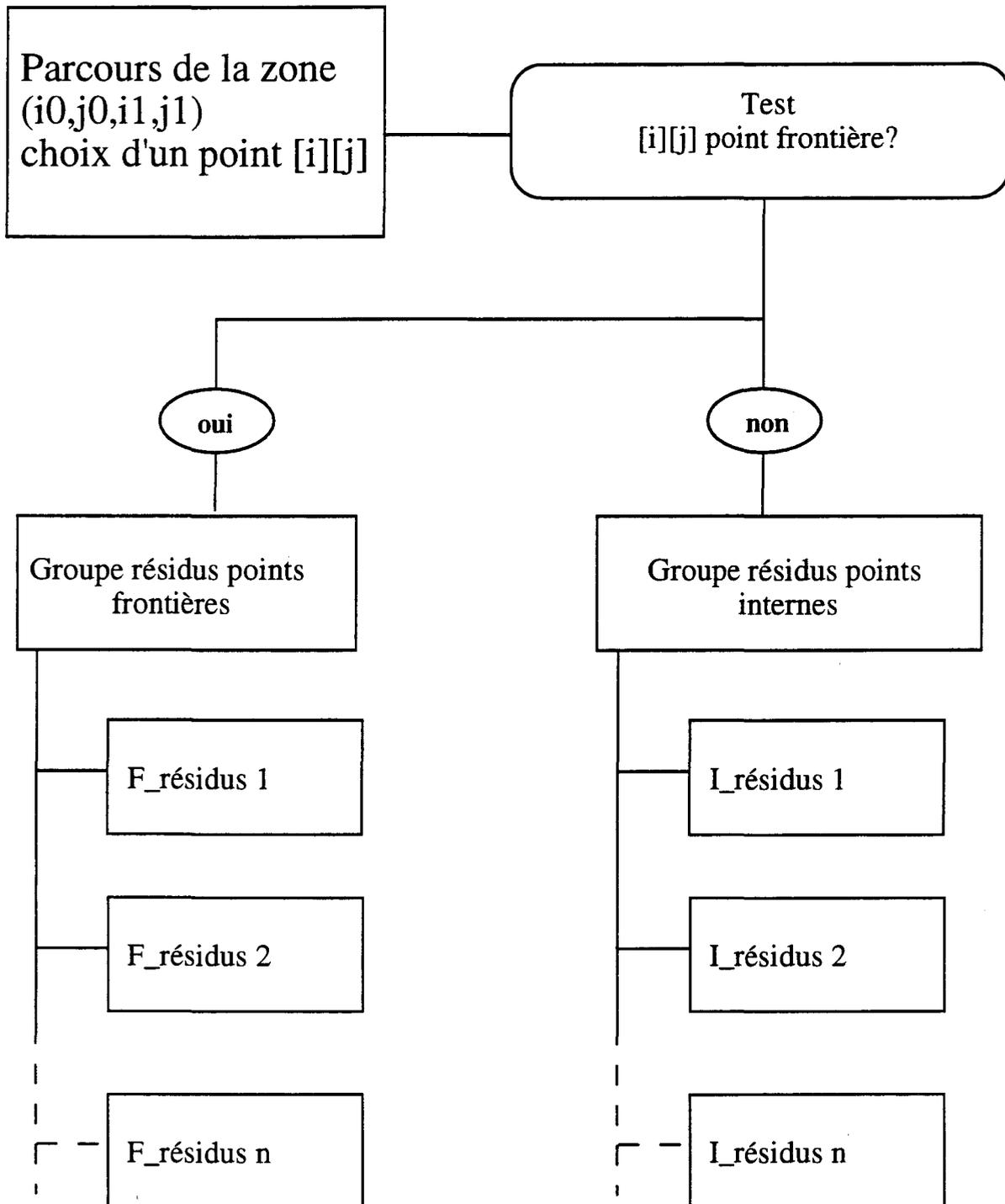


figure 9.6 Calcul du résidu niveau un

Calcul de résidu à un point frontière

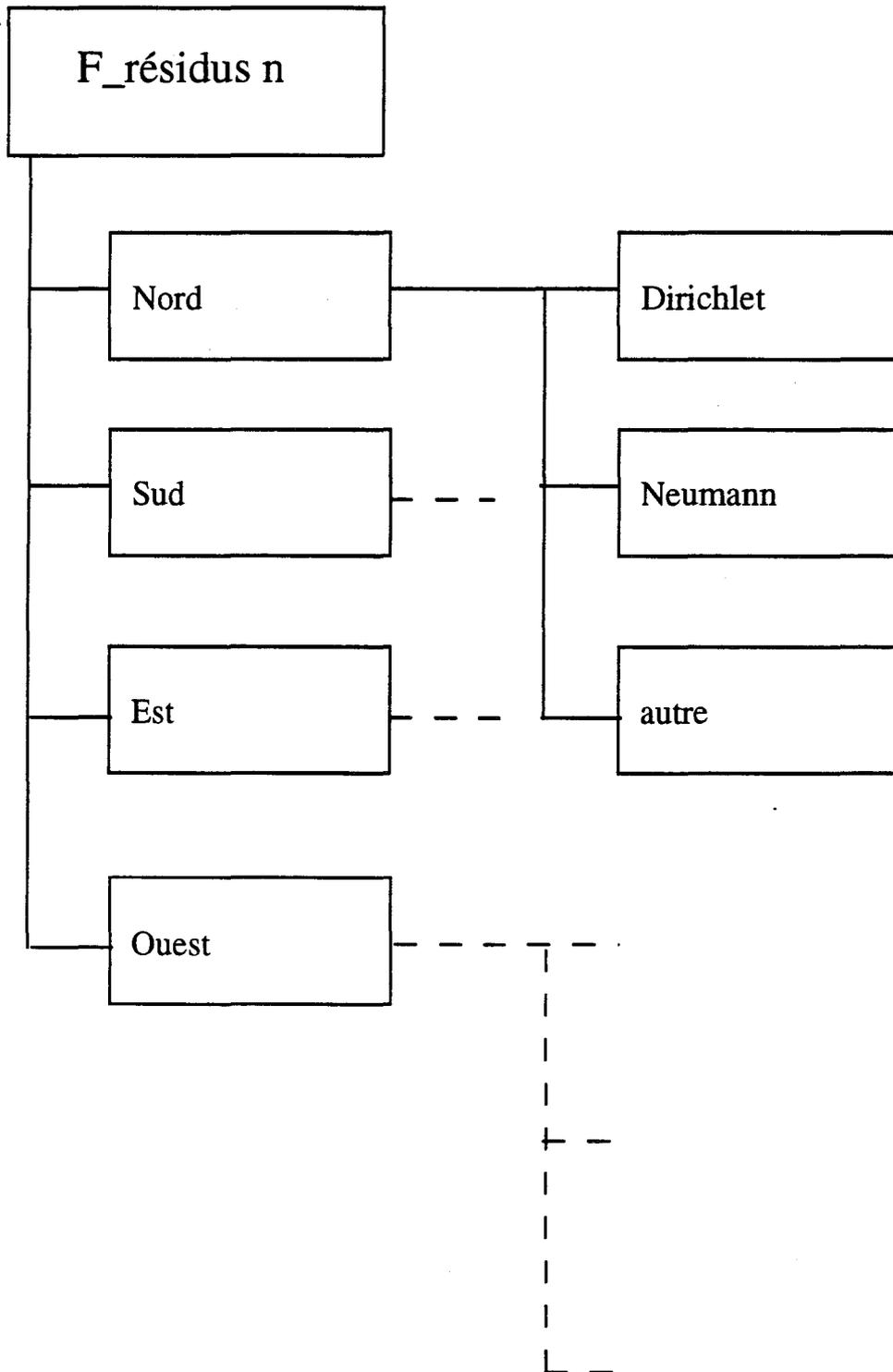


figure 9.7 Calcul du résidu niveau deux (traitement des frontières)

Linéarisation des équations

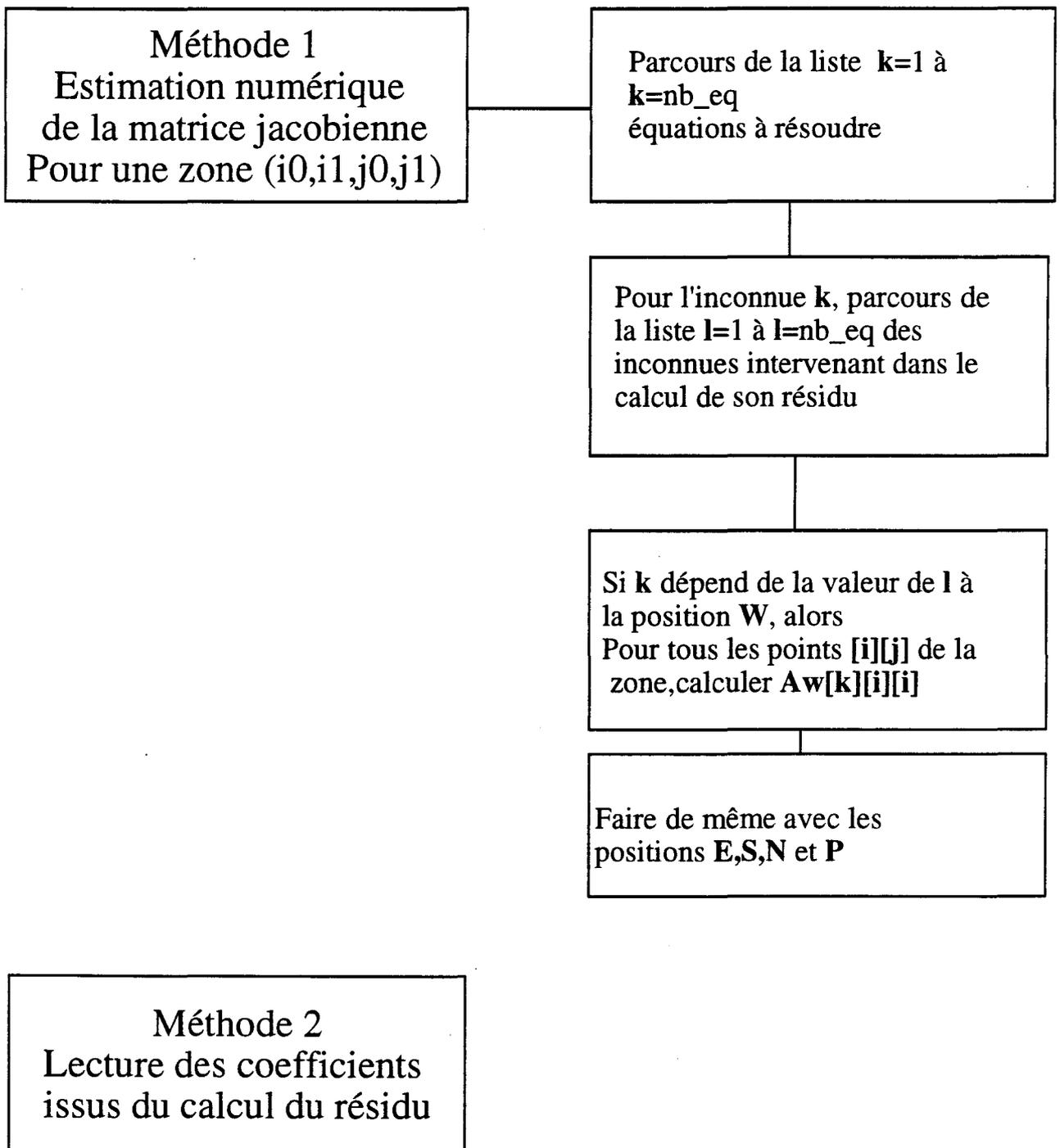


figure 9.8 Calcul des coefficients du système linéaire

Inversion bloc tridiagonale pour une zone (i_0, j_0, i_1, j_1)

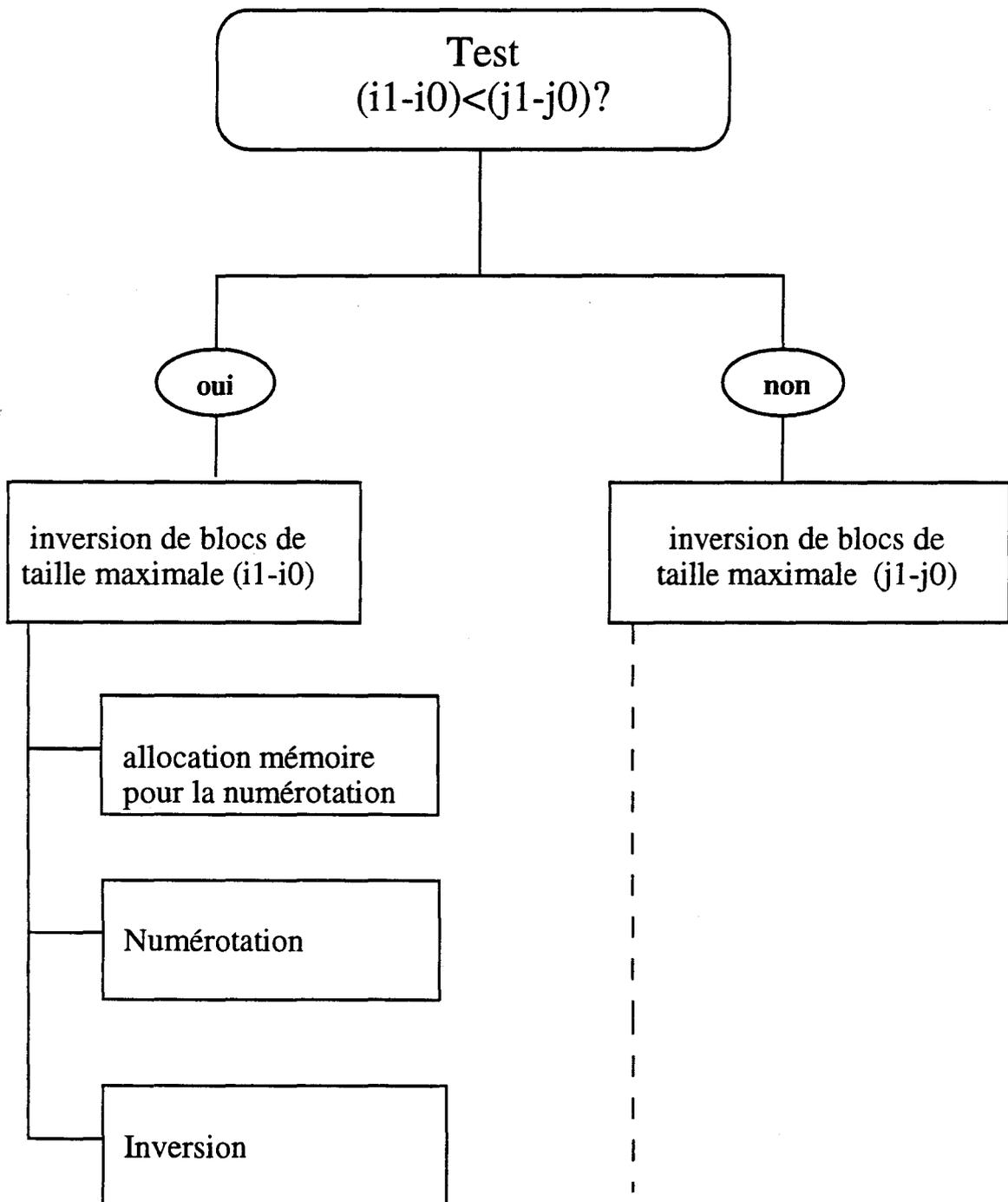


figure 9.9 Inversion par bloc du système linéaire
niveau un

Inversion de blocs de taille maximale (i1-i0)

$$A[j_0] = 0$$

Boucle de $j=j_0$ à $j=j_1$

Allocation mémoire et remplissage des
matrices $A[j]$, $B[j]$, $C[j]$, et $D[j]$

$$\begin{aligned} B[j] &= B[j] - A[j].C[j-1] \\ D[j] &= D[j] - A[j].D[j-1] \\ B[j] &= \text{inverse}(B[j]) \end{aligned}$$

$$\begin{aligned} C[j] &= B[j].C[j] \\ D[j] &= B[j].D[j] \\ \text{libérer place mémoire } &A[j] \text{ et } B[j] \end{aligned}$$

$$X[j_1] = D[j_1]$$

Boucle de
 $j=j_1-1$ à $j=j_0$

$$X[j] = D[j] + C[j].X[j+1]$$

Libérer la place
mémoire pour tous
les $C[j]$ et tous les
 $D[j]$

A partir de $X[j]$
répartir les
corrections suivant
les variables u, v et p

figure 9.10 Inversion par bloc du système
linéaire niveau deux

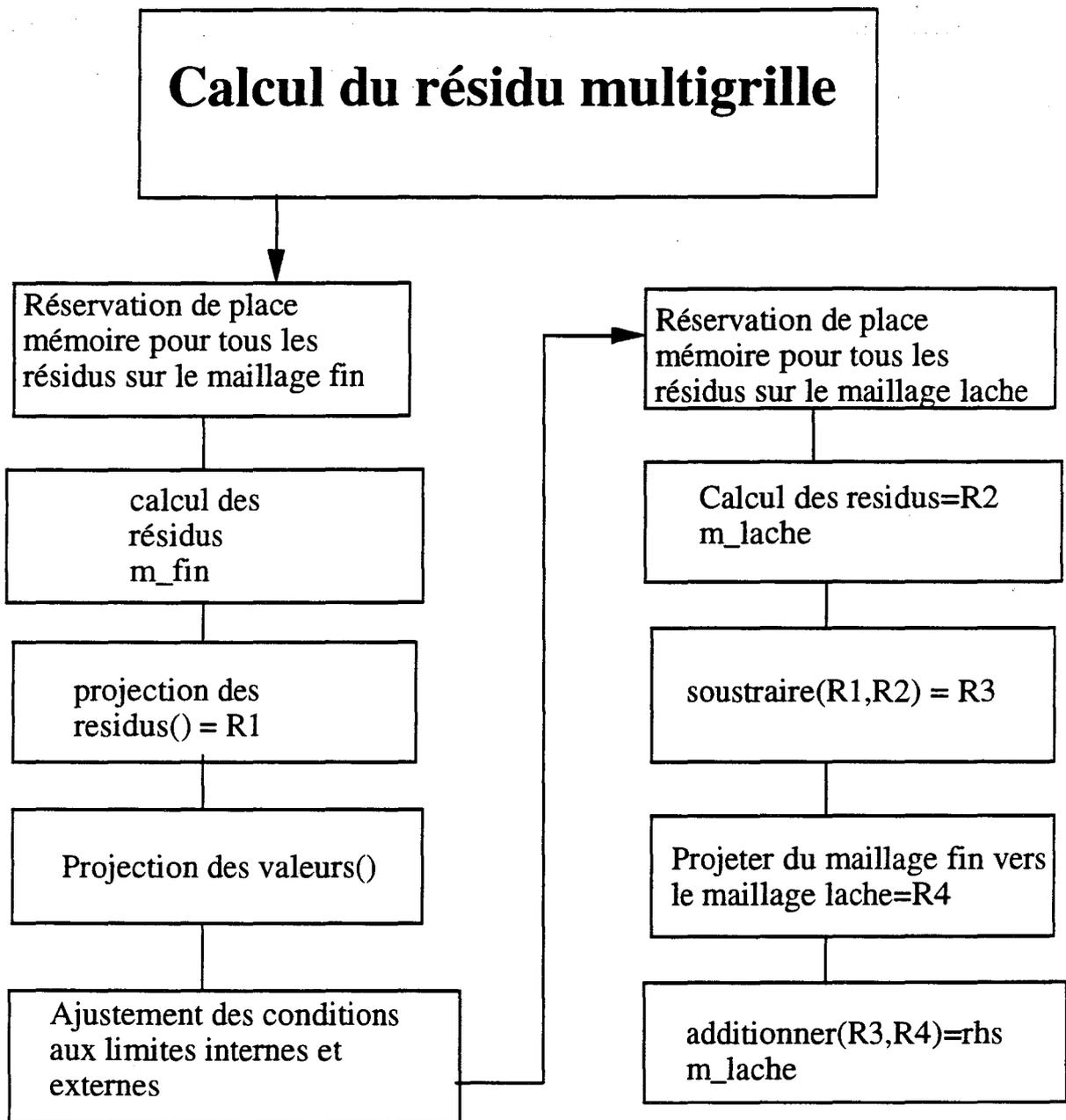


figure 9.11 Calcul du résidu multigrille en vu de la projection du problème sur maillage lâche.

Structure Resol-sys

Structure support pour l'ensemble des données du code

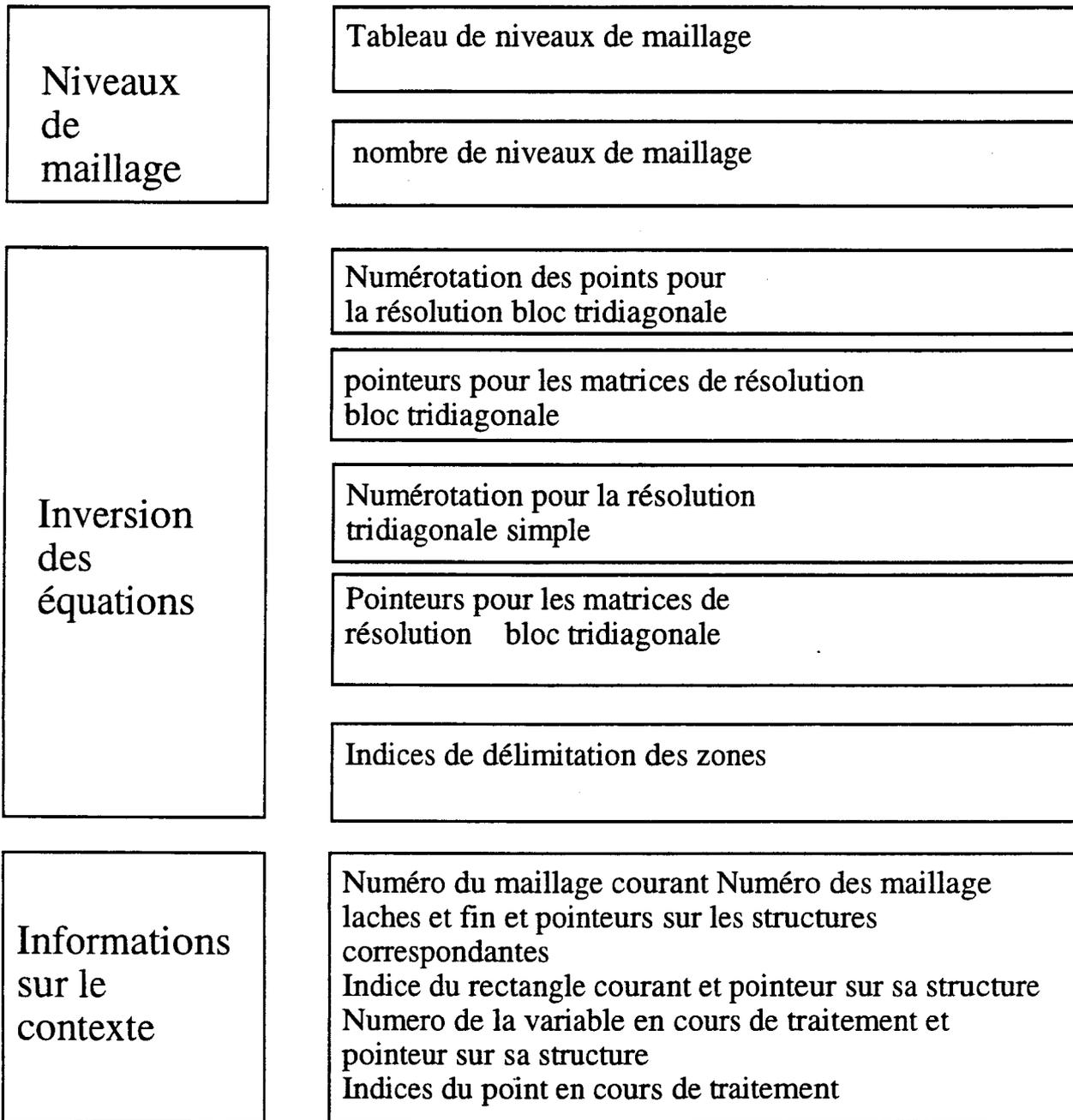


figure 9.12 Structure principale pour les données du code: niveau 1.

Structure Grid

Structure support pour l'ensemble des données d'un niveau de maillage

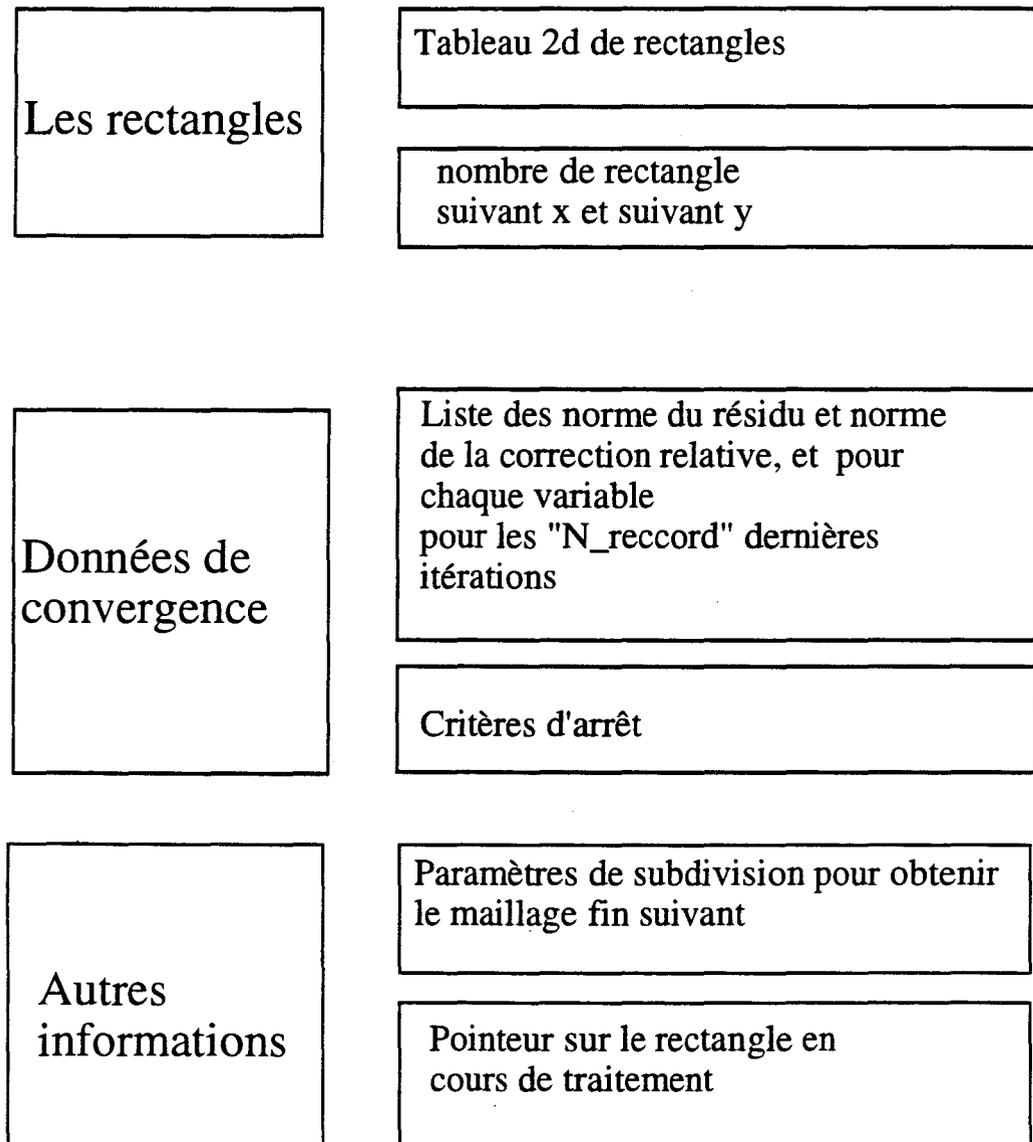


figure 9.13 Structure principale pour les données d'un niveau de maillage : niveau 2.

Structure Square

Structure support pour l'ensemble des données d'un rectangle sous-domaine

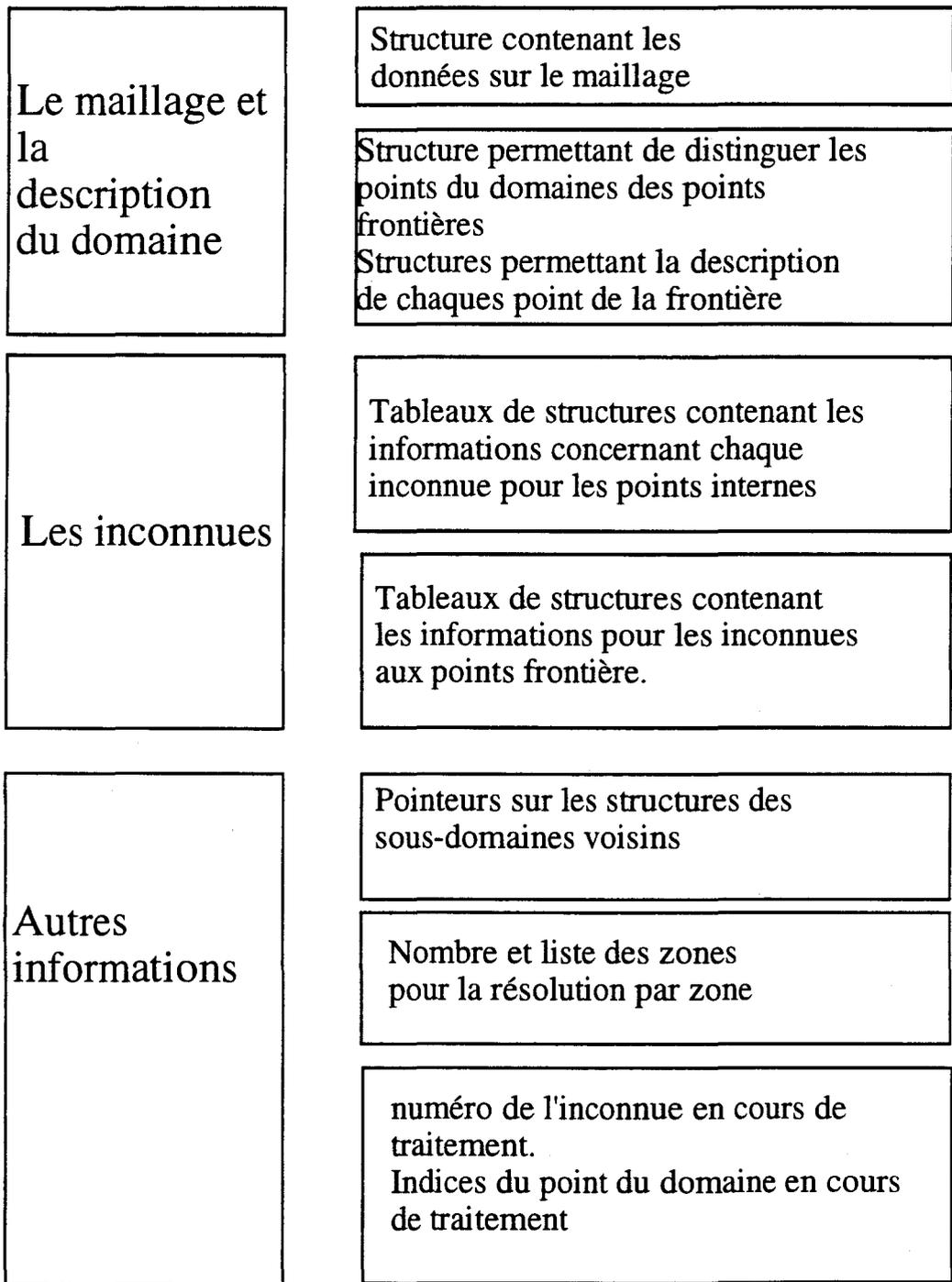


figure 9.14 Structure pour les données d'un sous-domaine: niveau 3.

<p>Le maillage</p>	<p>coordonnées de l'origine et dimensions du rectangle Maillage centre x,y Maillage décalé xu,yv distance à la paroi la plus proche xp,yp</p>
<p>Les inconnues</p>	<p>Structure pour les point domaine liste des tableaux 2d: Valeurs Corrections Résidus sipmple Résidus multigrille Coefficient et résidus linéaire</p> <hr/> <p>Structure pour les points frontière liste des tableaux 1d: Valeurs Corrections</p>
<p>Description des points du domaine</p>	<p>Structures des points du domaine liste des tableaux 2d: Description des points correspondant au maillage décalé suivant x Description des points correspondant au maillage décalé suivant y Description des points correspondant au maillage centré</p> <hr/> <p>Structures des points frontière liste des tableaux 1d: Description des points décallé suivant x Description des points décallé suivant y Description des points centré</p>

figure 9.15 Structures de bas niveau, maillage, inconnues du problème et description des points: niveau 4.

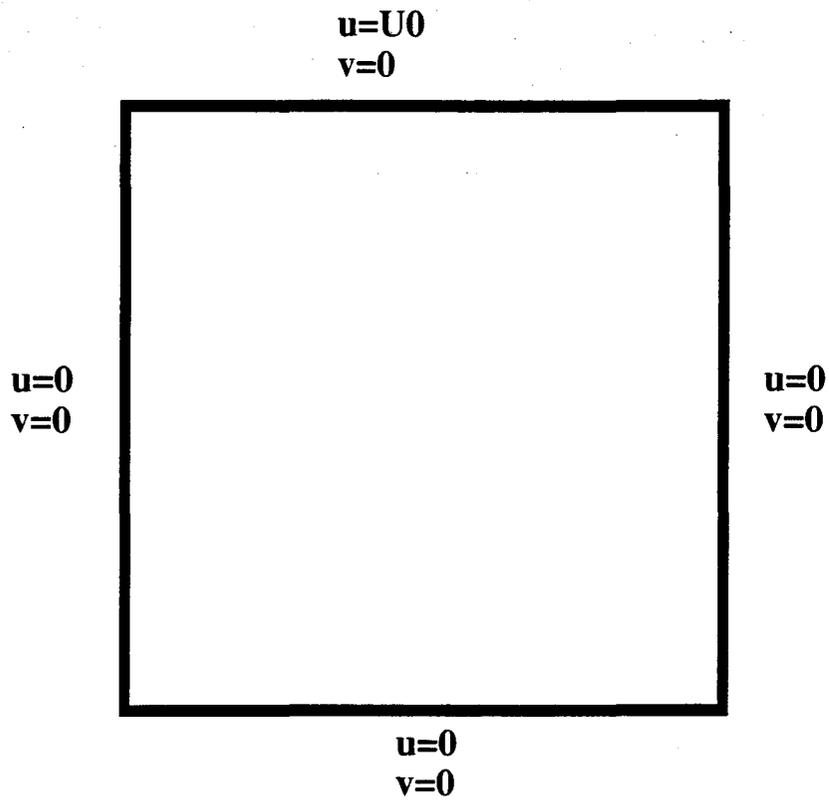
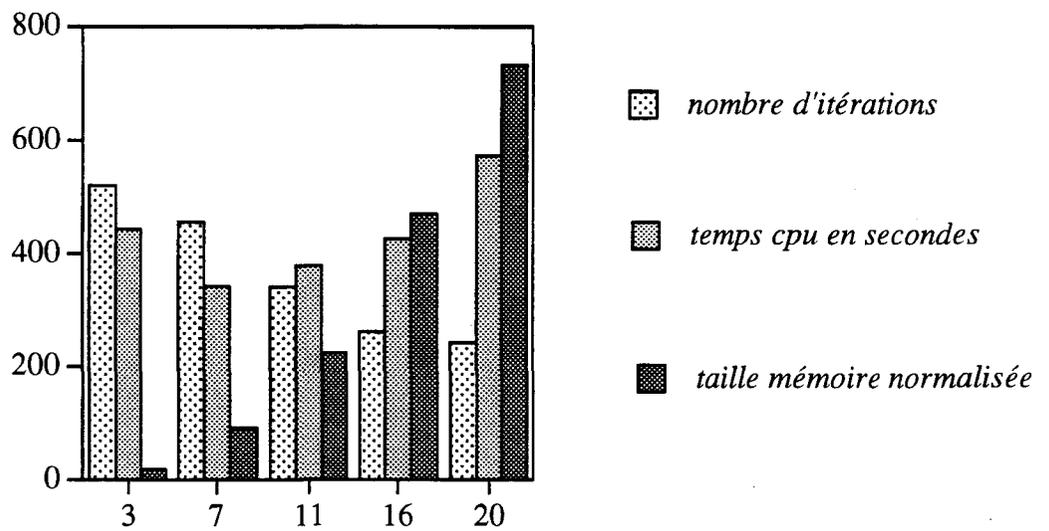


figure 10.1: Conditions aux limites pour l'écoulement de cavité



Cavité entraînée $Re = 1000$ maillage 40×40

figure 10.2: Influence de la taille de bande sur trois paramètres de calcul pour la méthode par zone

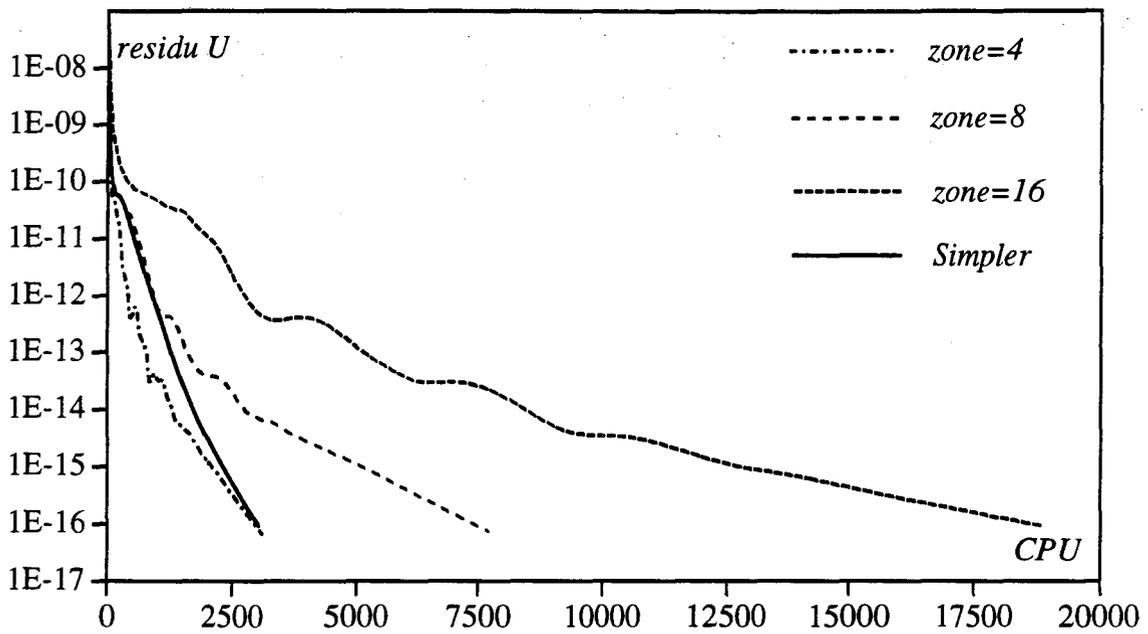


figure 10.3: Courbes de convergence pour la Cavite_64x64 Re=1000 en fonction du temps CPU

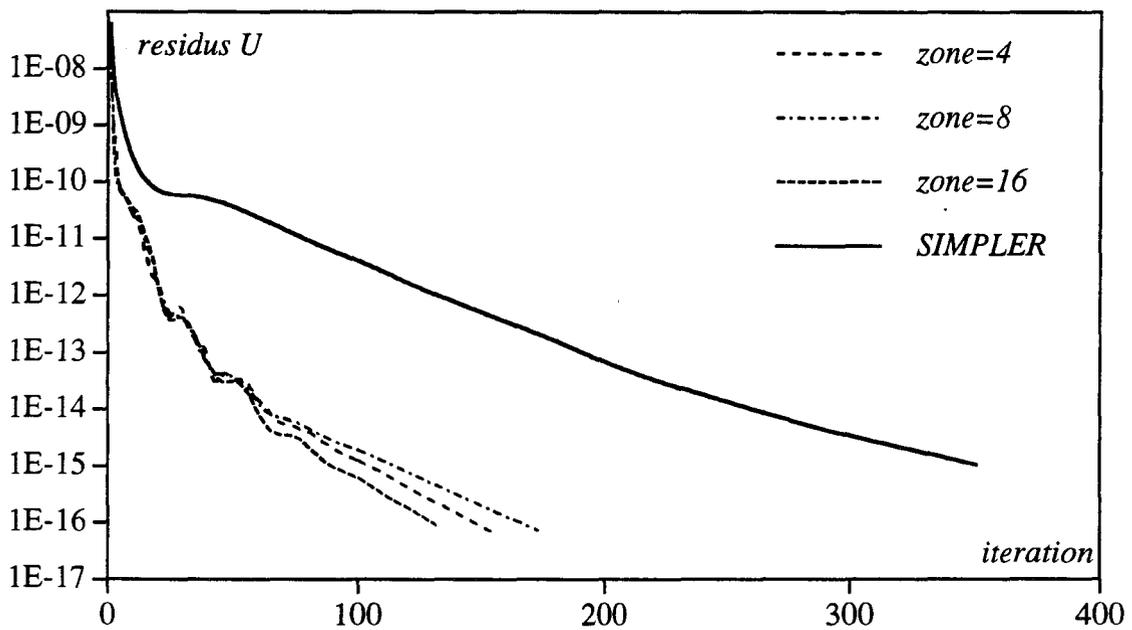


figure 10.4: Courbes de convergence pour la Cavite_64x64 Re=1000 en fonction du nombre d'itérations

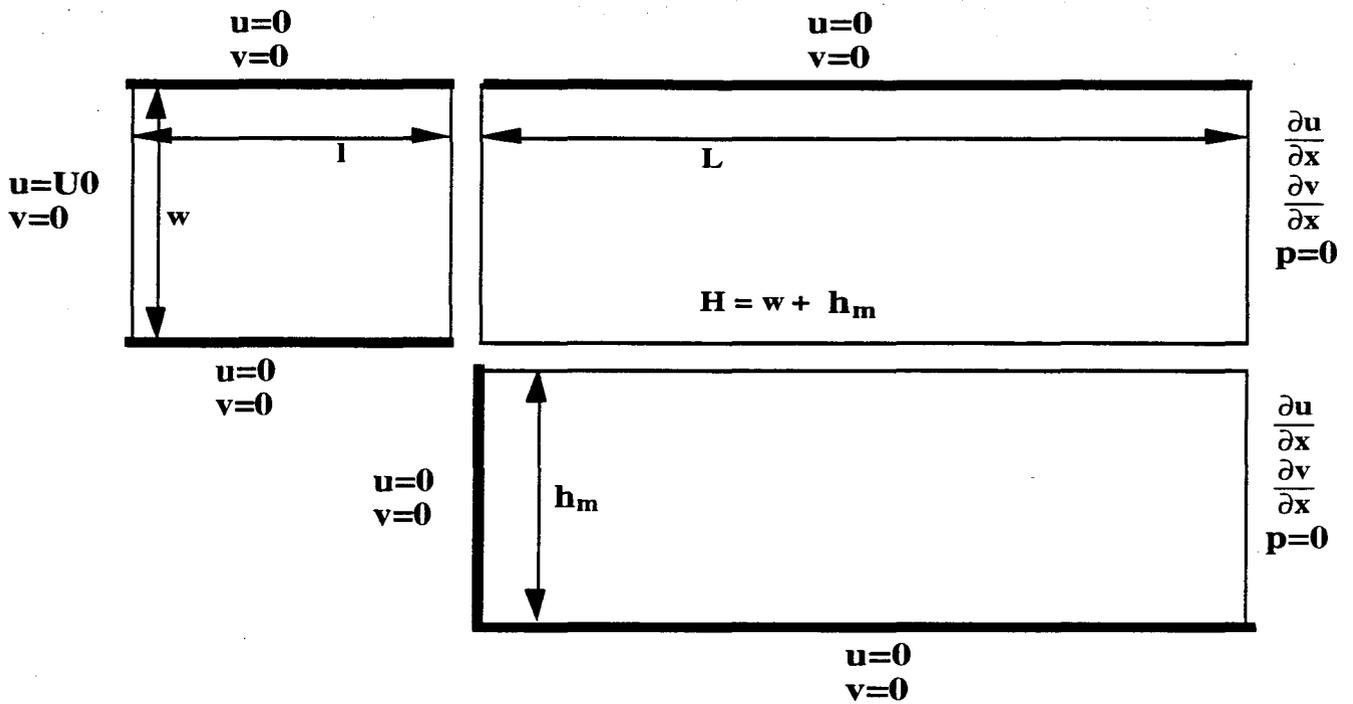


figure 10.5: Conditions aux limites et décomposition en sous-domaines pour la marche

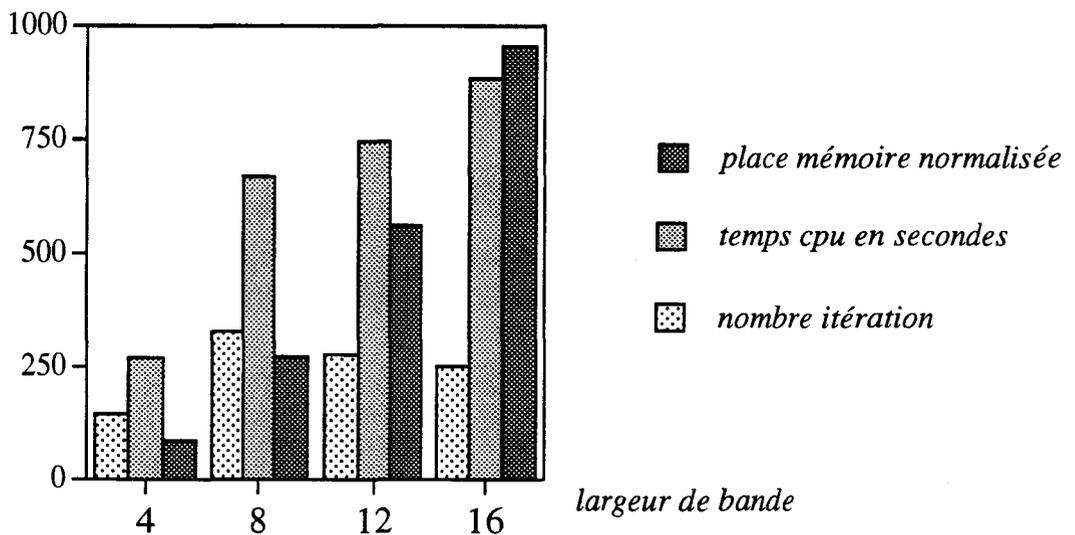


figure 10.6: Influence de la taille de bande sur trois paramètres de calcul pour la méthode par zone

Marche descendante $Re = 1000$ 36×70

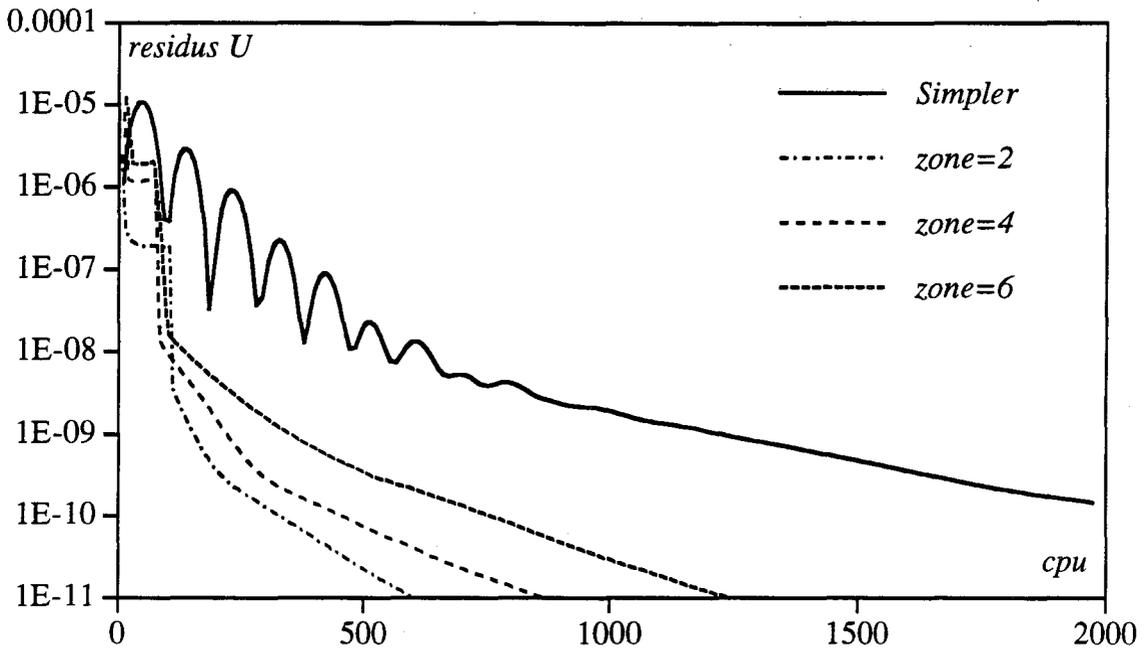


figure 10.7: Marche 48x64 Re=1000 (CPU)

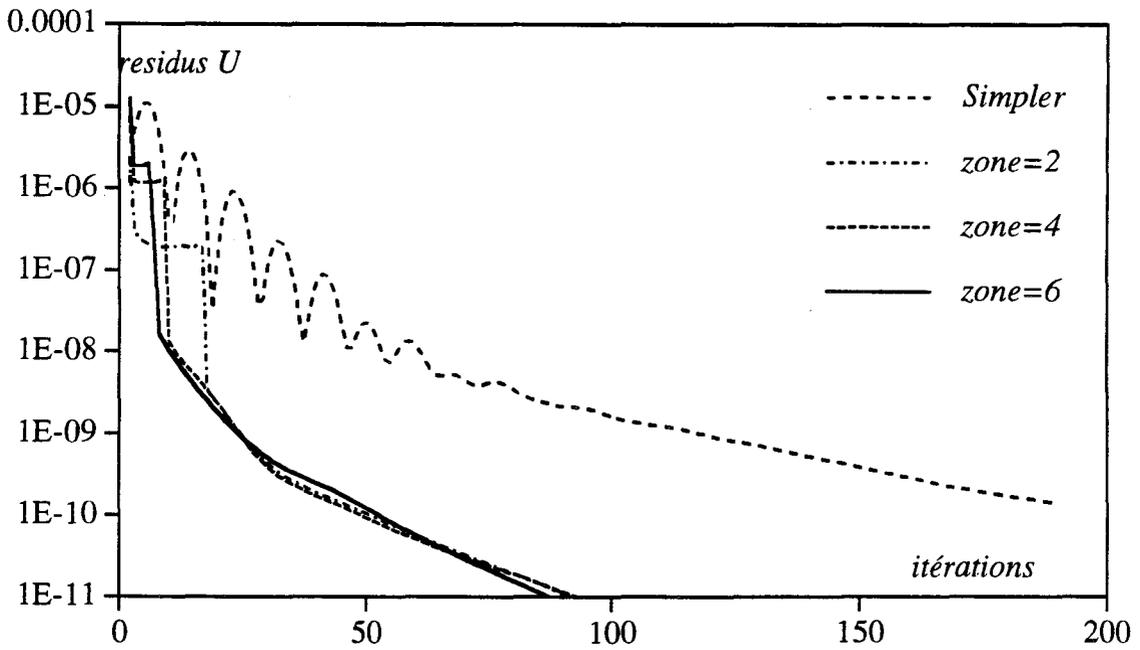


figure 10.8: Marche 48x64 Re=1000 (itérations)

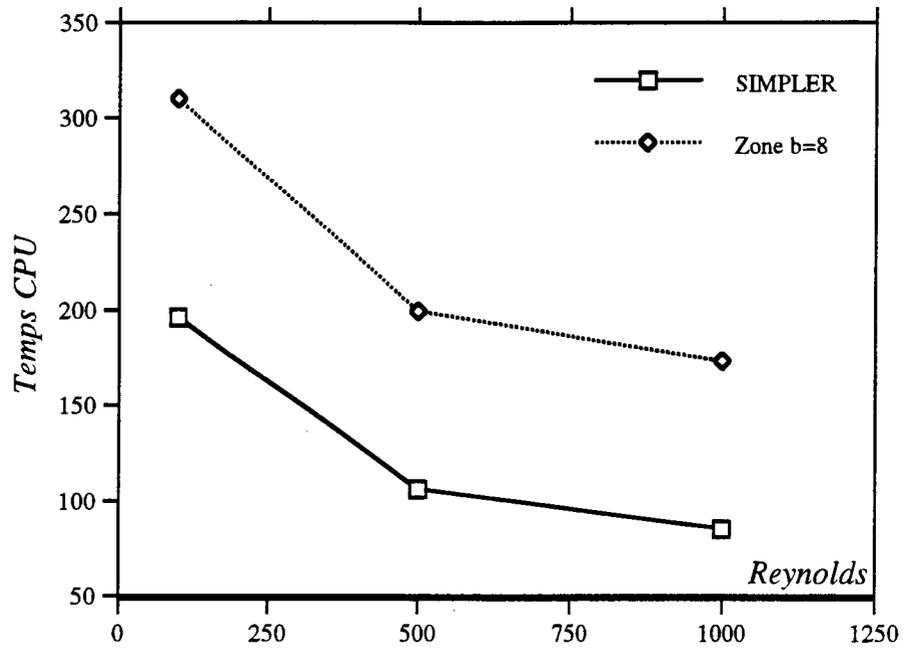


figure 10.9: Cavite 32x32

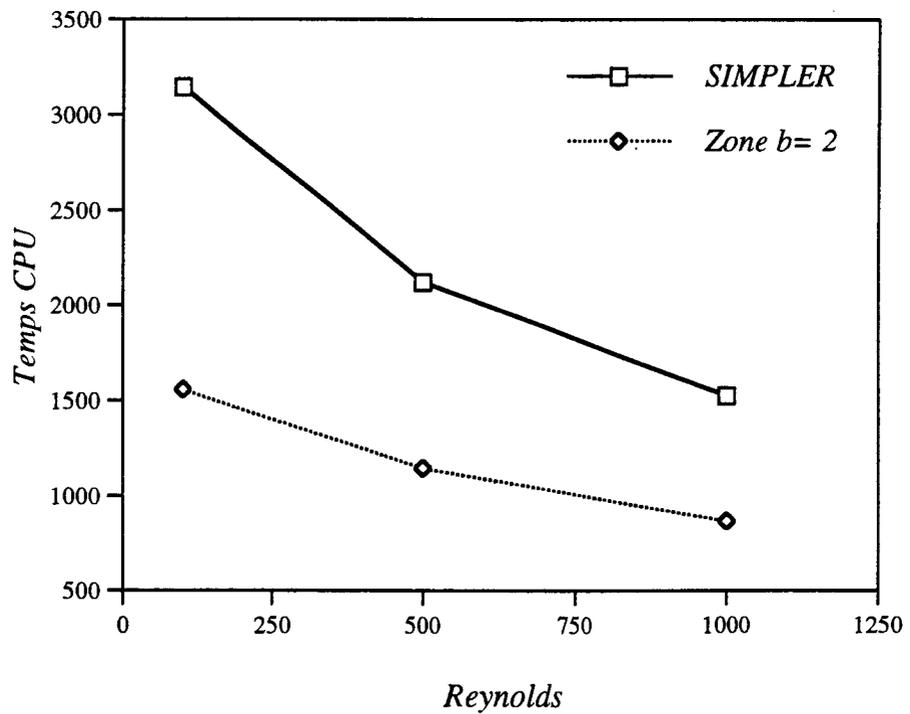


figure 10.10: Marche 16x32, 32x32

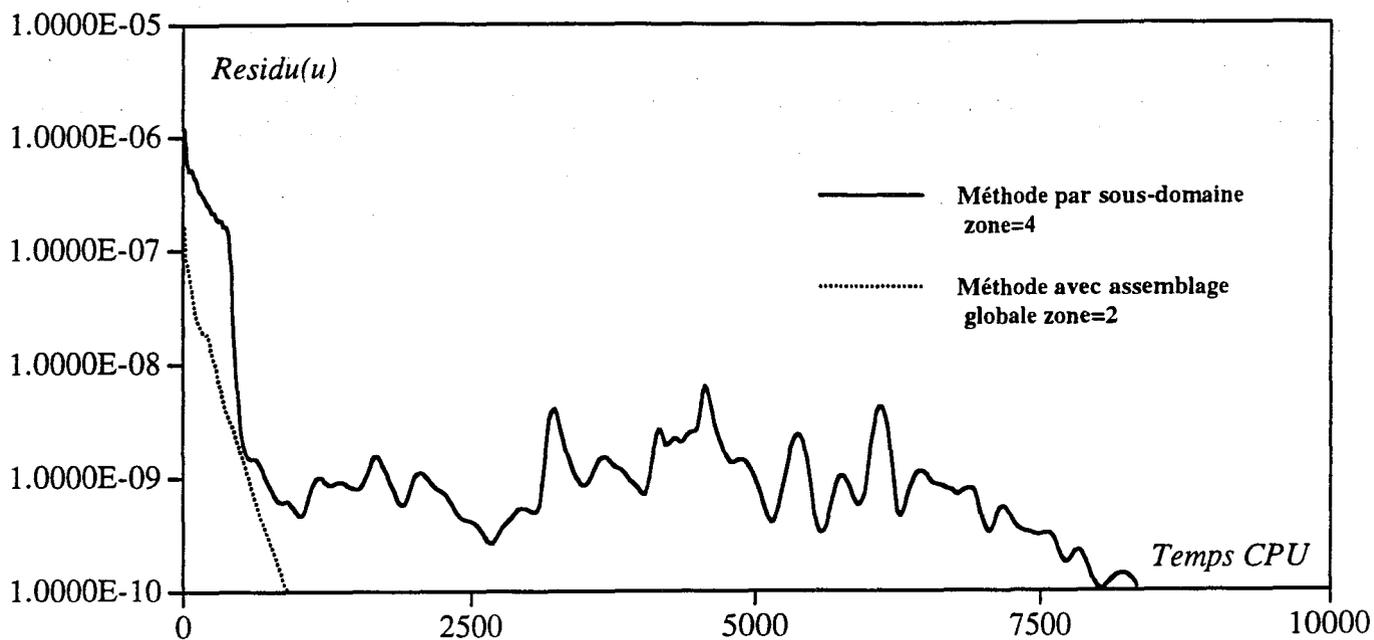


figure 10.11: Marche descendante
Re=1000 maillage 16x32, 32x64

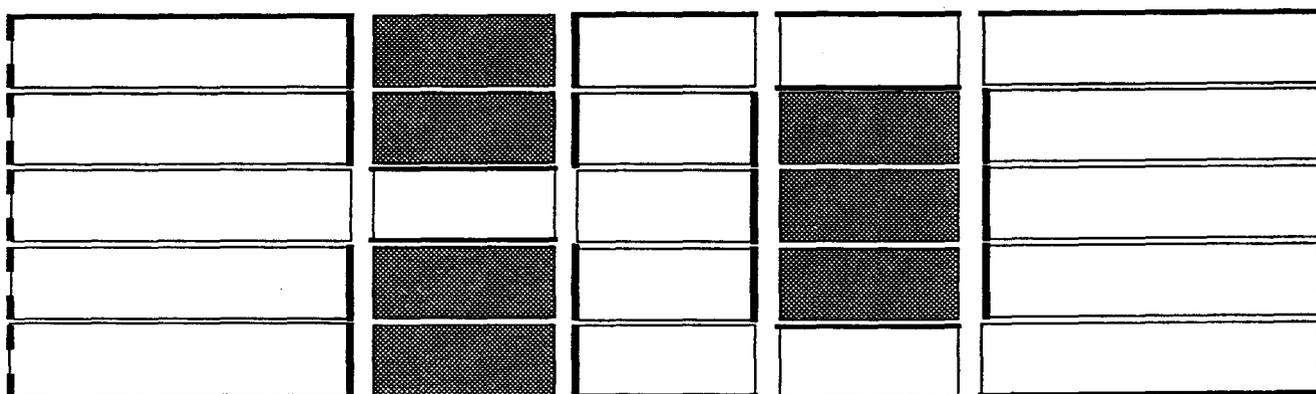


figure 10.12a: Ecoulement dans un canal
avec valve anti-retour: subdivision du
domaine en sous-domaines

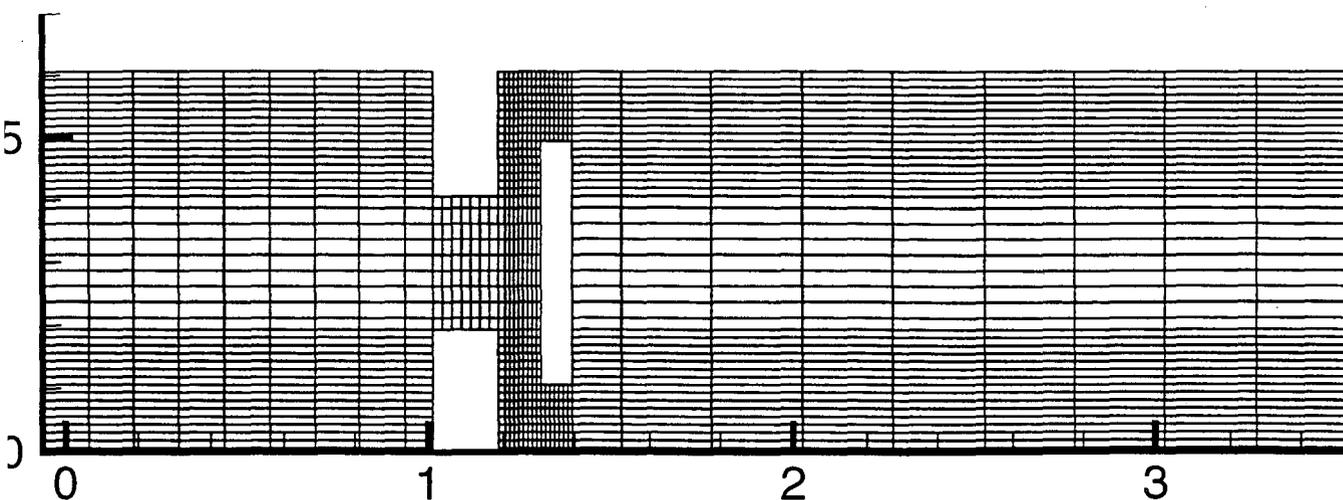


figure 10.12b: Maillage d'un écoulement
dans un canal avec valve anti-retour.

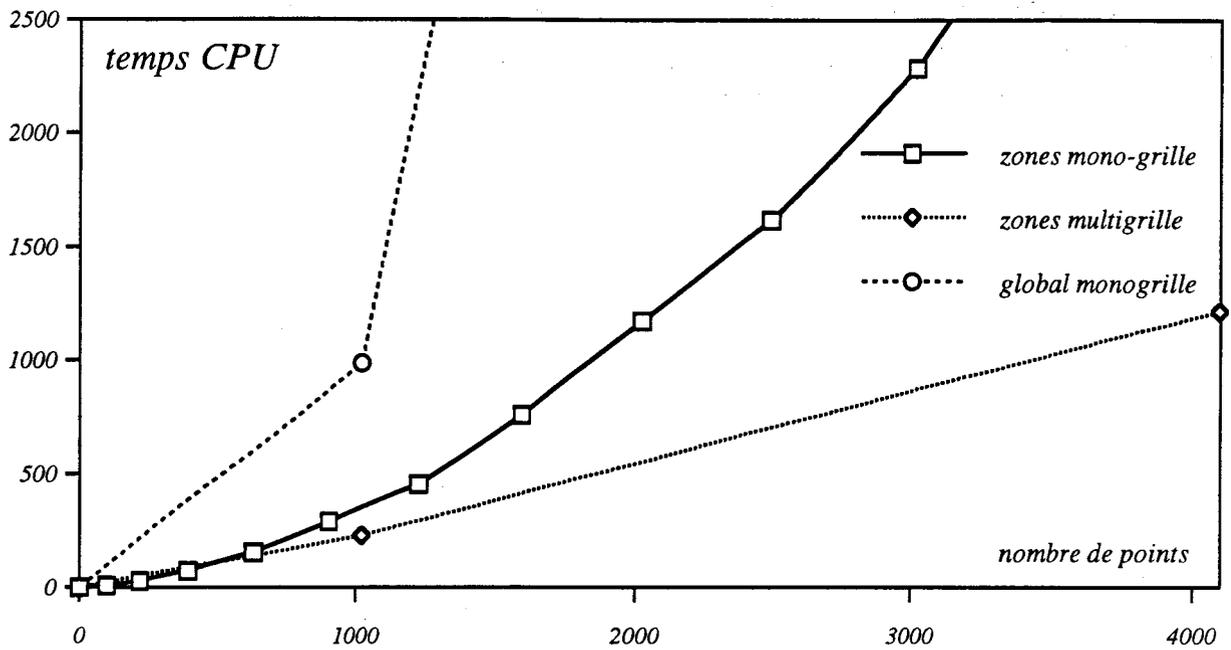


figure 10.13: Cas de la cavité entraînée

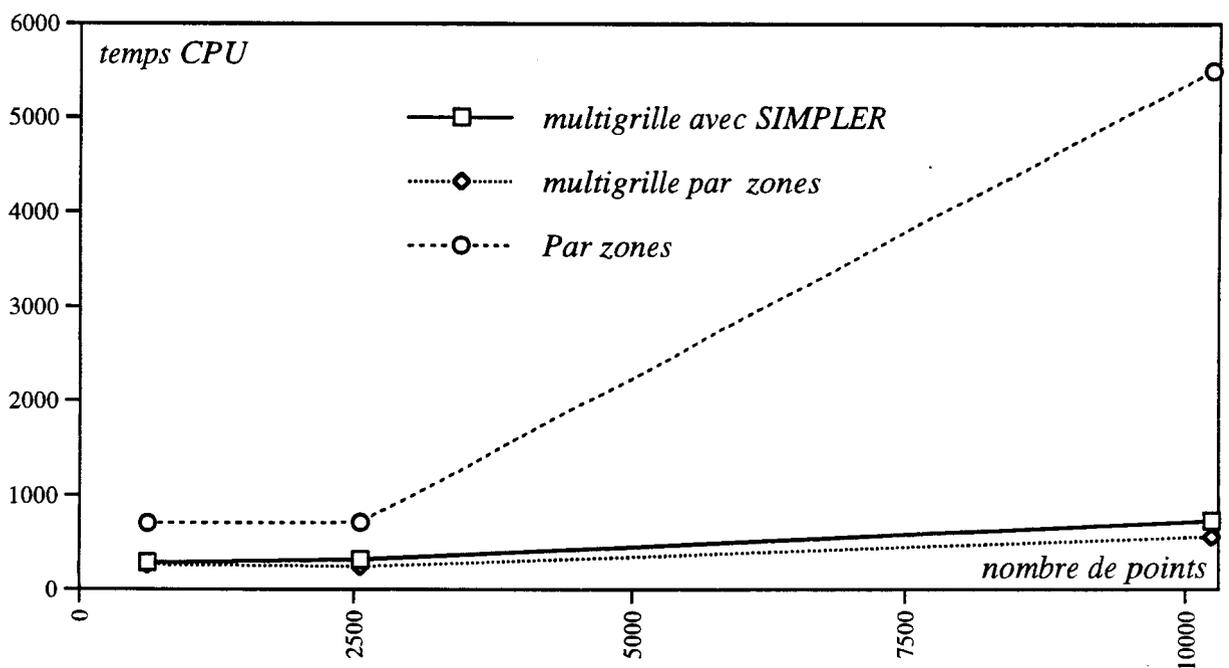


figure 10.14: Cas de la marche descendante

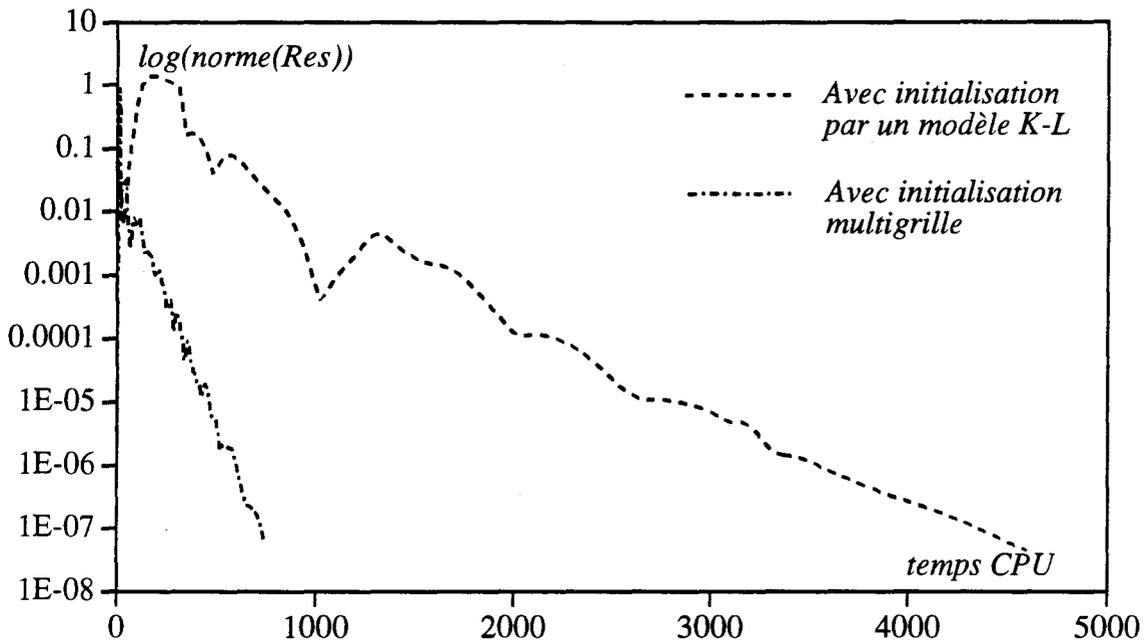


figure 10.15: Temps CPU

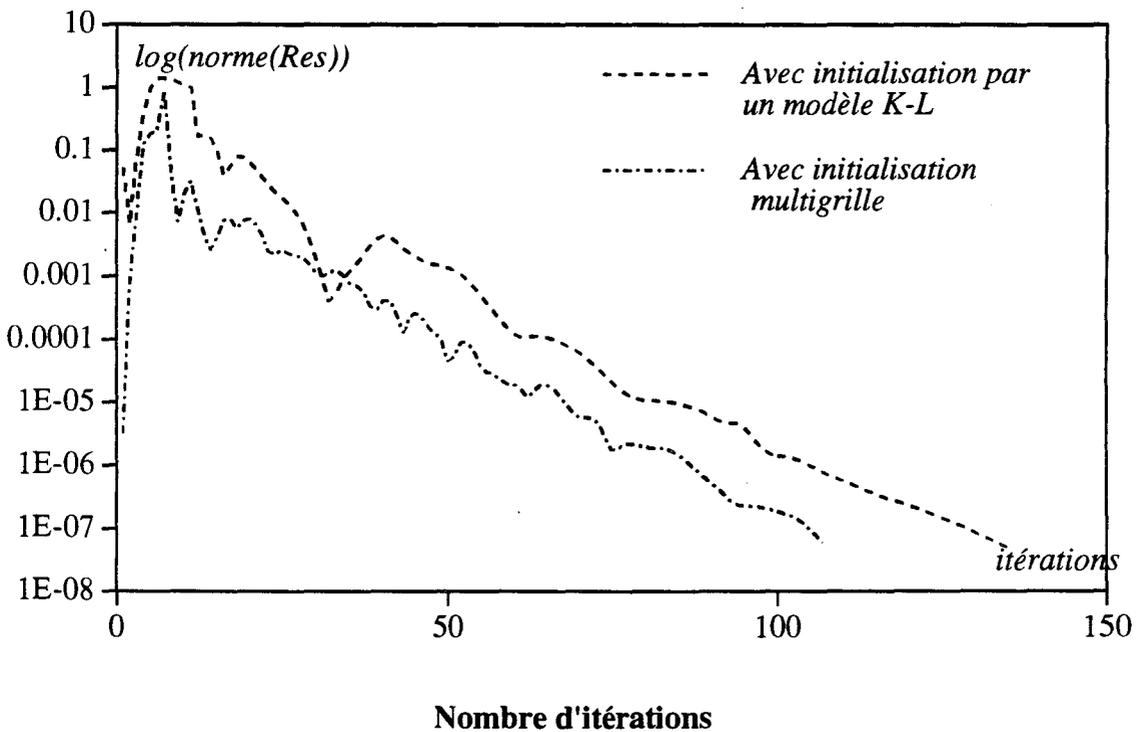
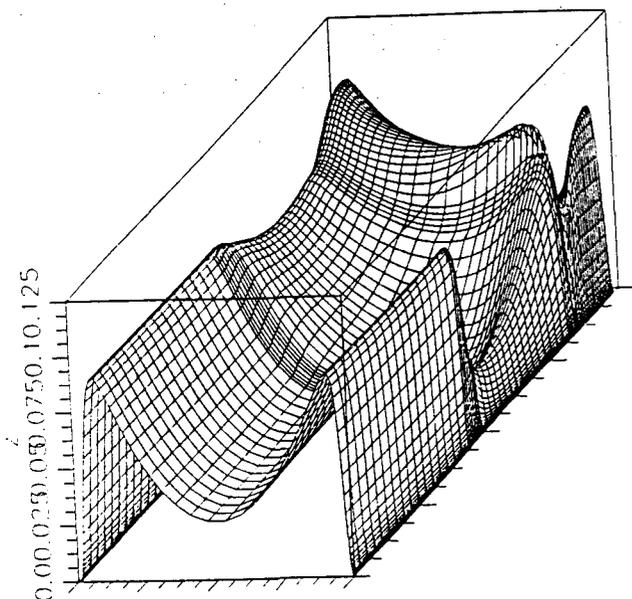
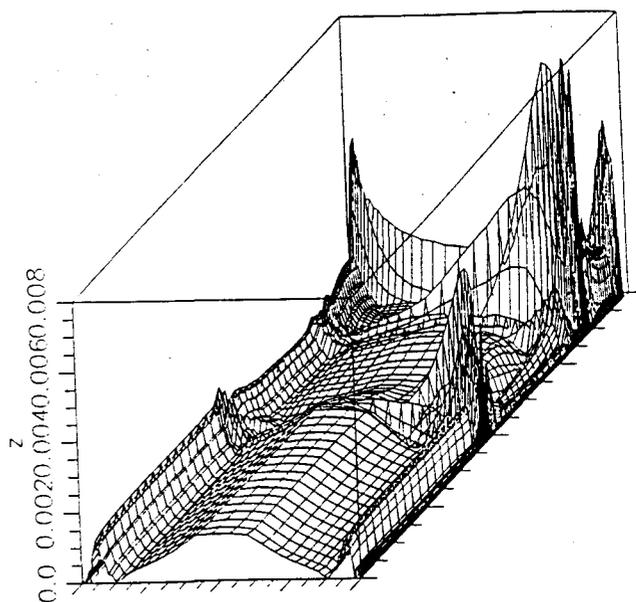


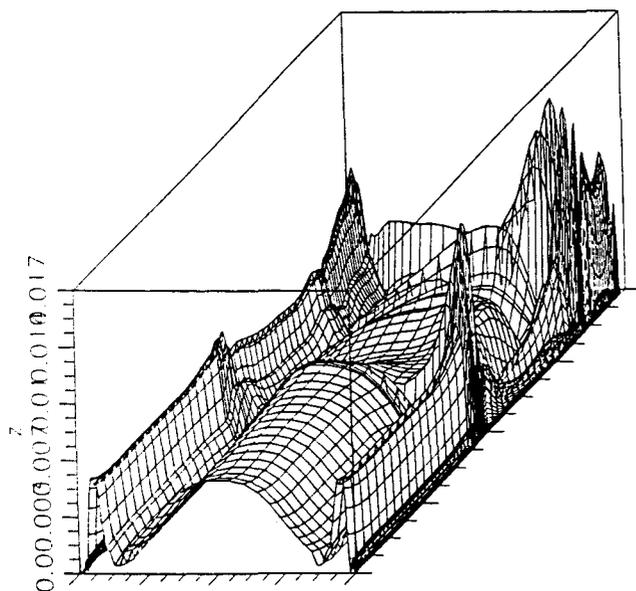
figure 10.16: Comparaison de deux méthodes d'initialisation des calculs turbulents



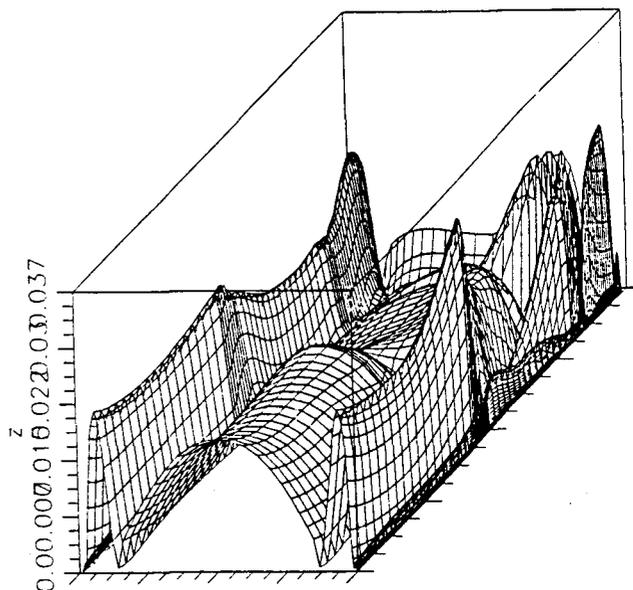
10.17a. fonction initiale sur le maillage fin



10.17b. erreur sur 1 niveau



10.17c. erreur sur 2 niveaux



10.17d. erreur sur 3 niveaux

figure 10.17. influence du nombre de niveaux multigrille sur l'erreur de projection interpolation. Cas de l'énergie turbulente dans un écoulement de canal.

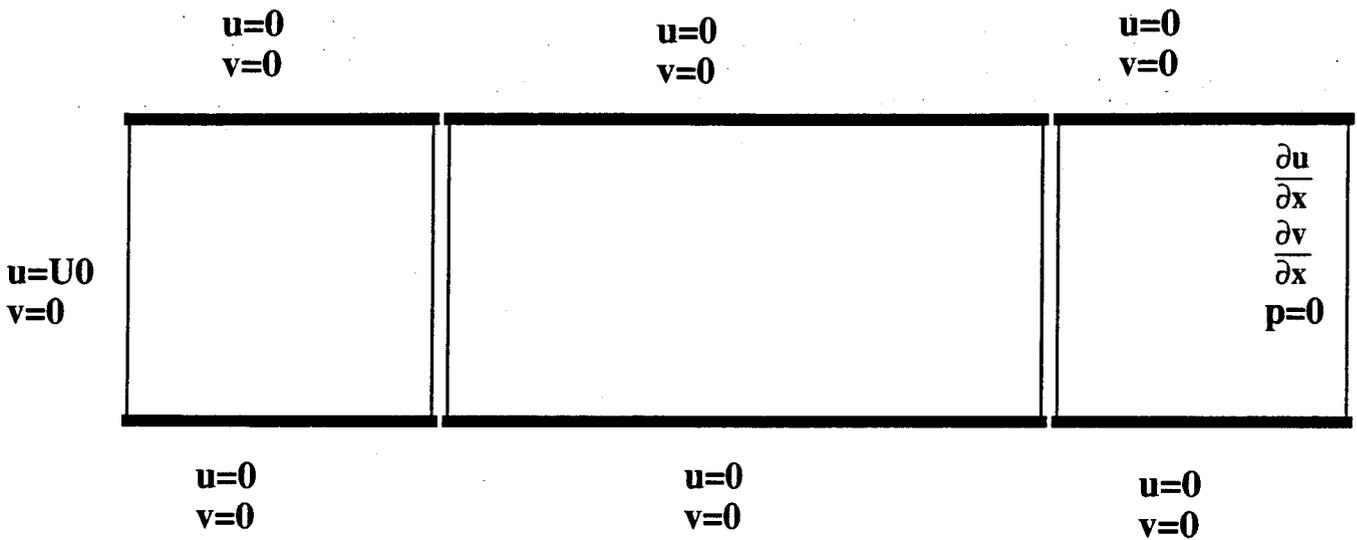


figure11.1: Conditions aux limites pour le canal plan en trois sections

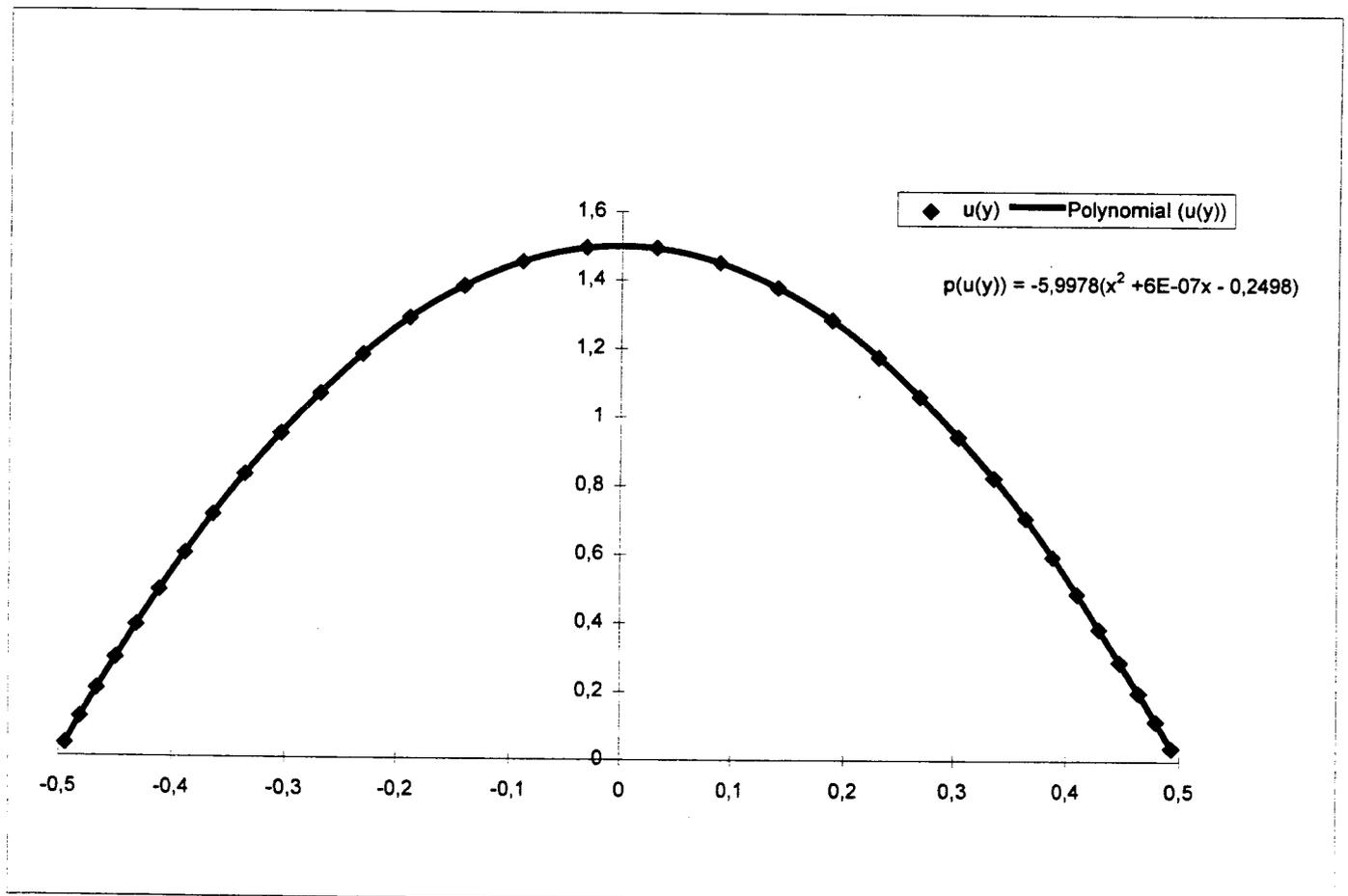


figure11.2: Profil de vitesse établi et polynôme d'interpolation correspondant.

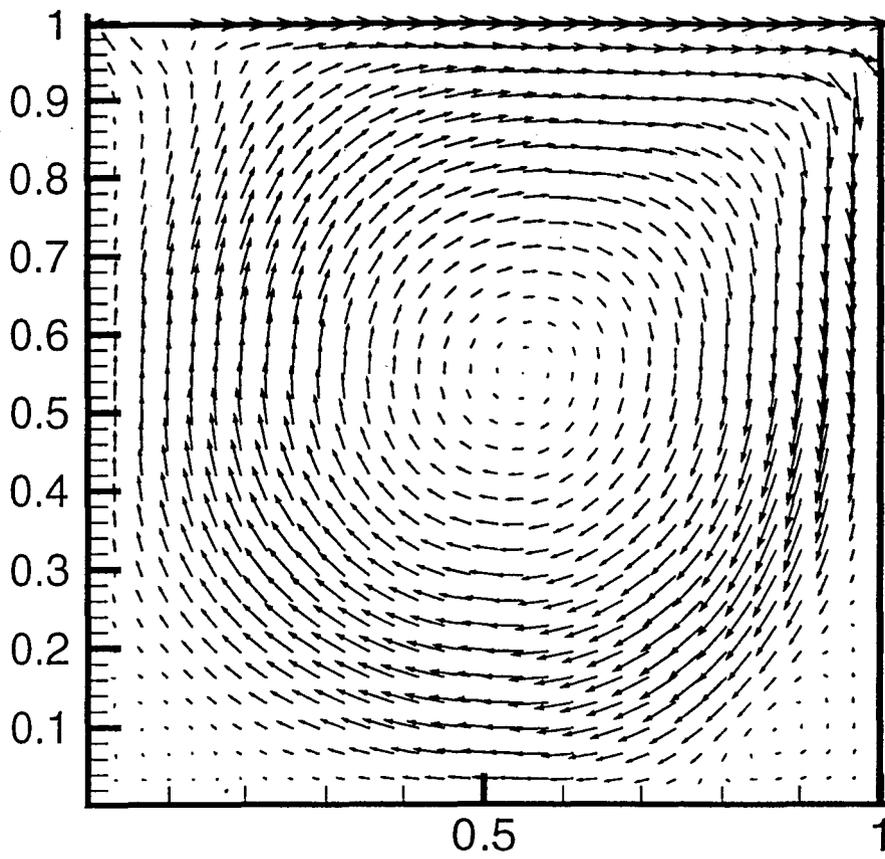


figure11.3: Champ de vitesse pour l'écoulement de cavité

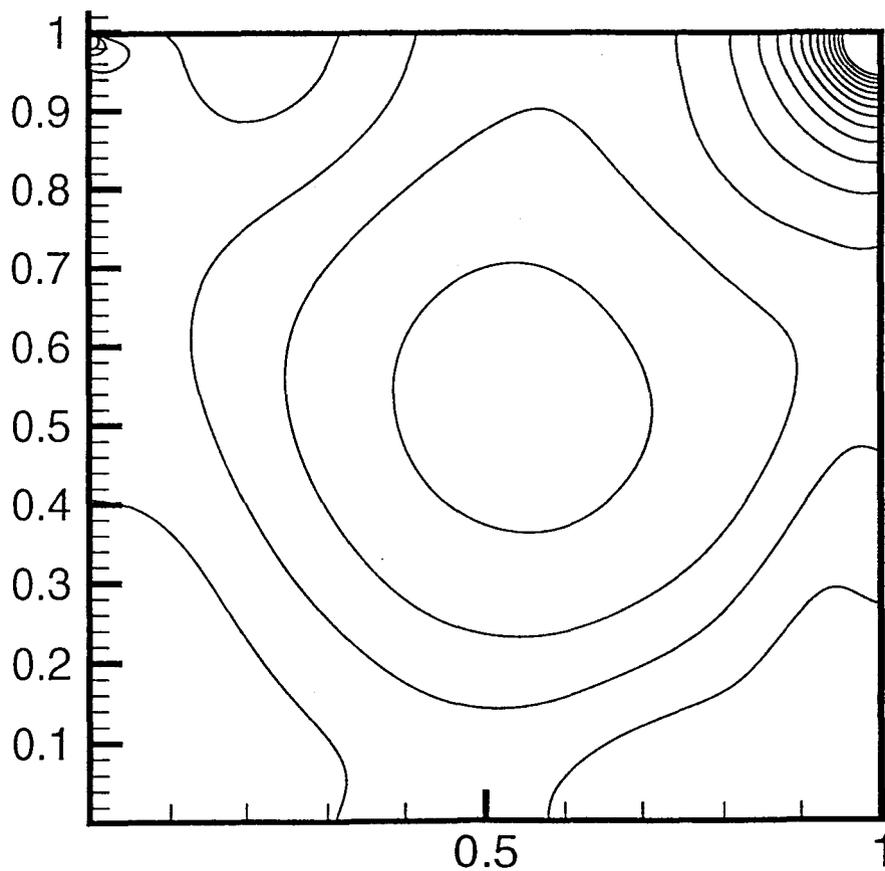


figure11.4: Iso-pression pour l'écoulement de cavité

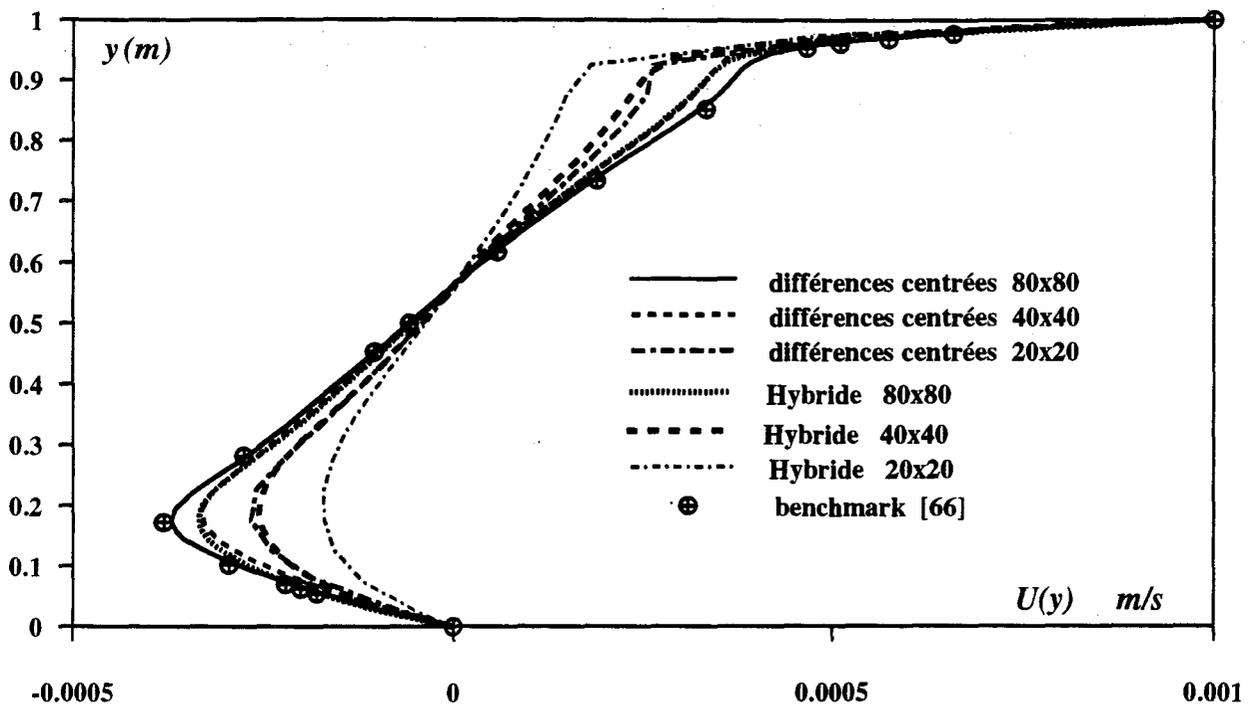


figure 11.5: Comparaison du profil de vitesse pour $x = 0.5 \text{ m}$ pour différentes tailles de maillage et deux schémas de discrétisation à $Re = 1000$

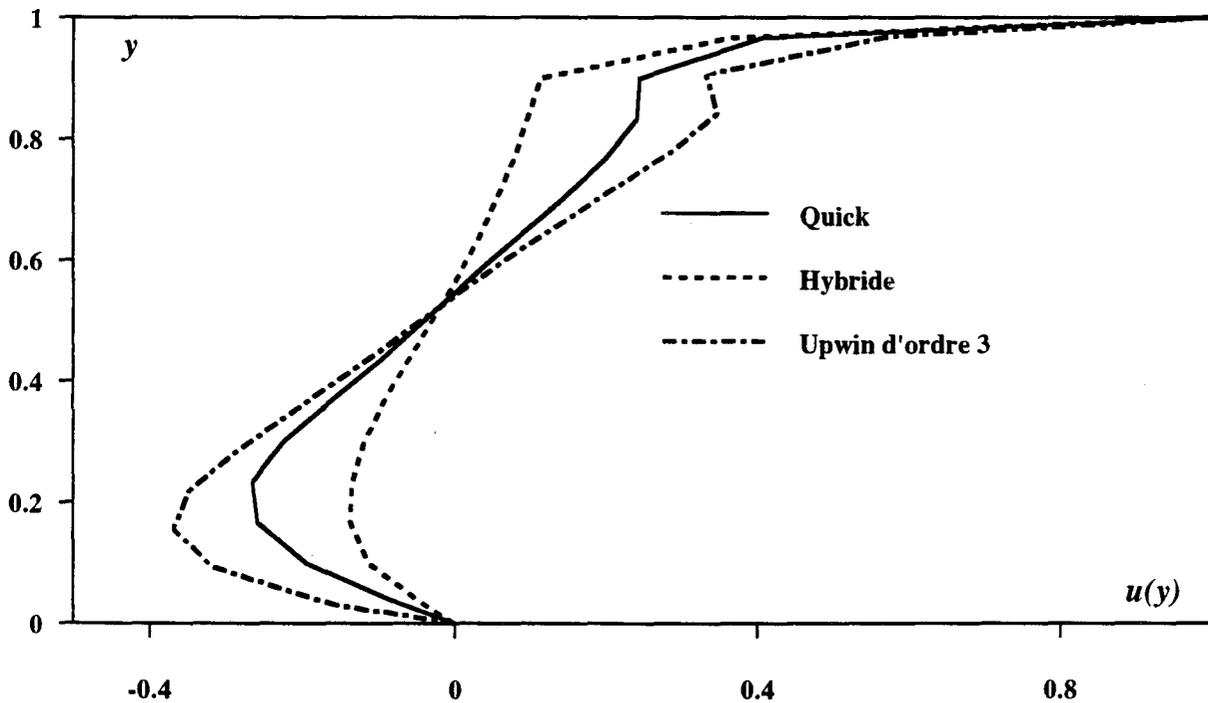


figure 11.6: Comparaison de schémas de précision sur un maillage 16×16 , $Re=1000$

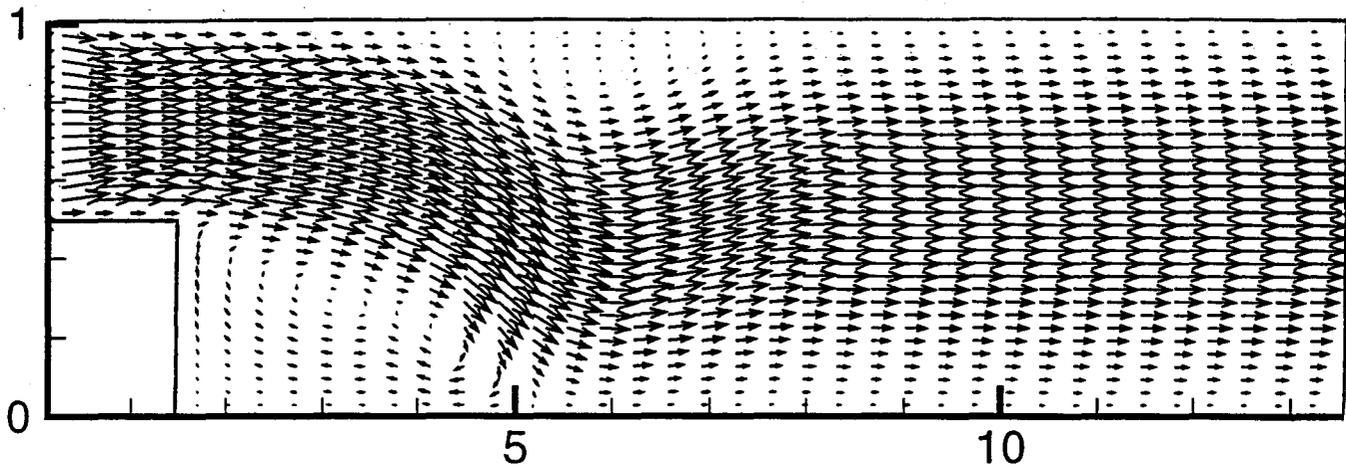


figure11.7: Champ de vitesse pour l'écoulement de marche descendante.

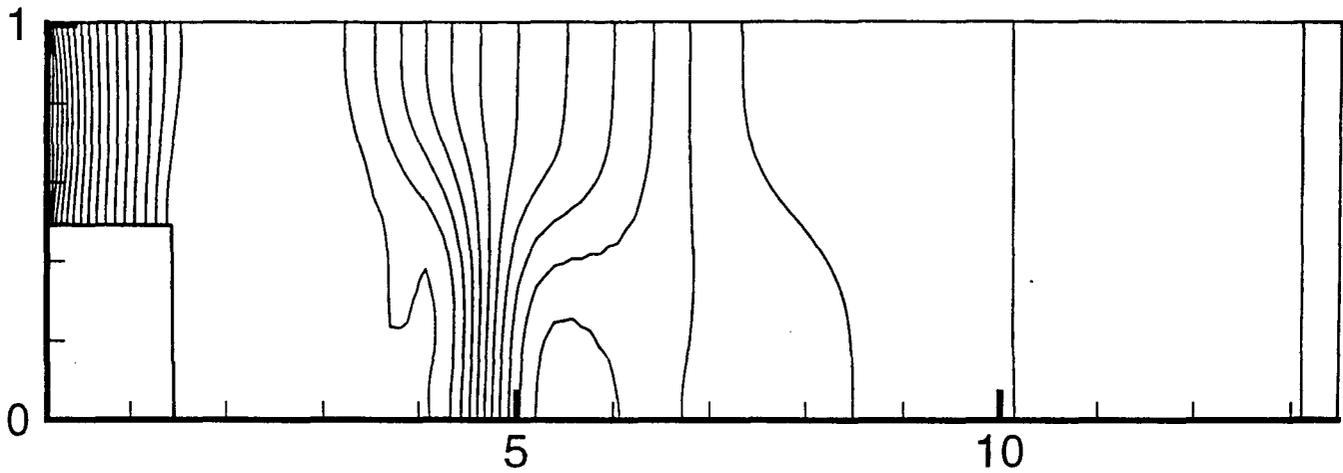


figure11.8: Iso-pression pour l'écoulement de marche descendante.

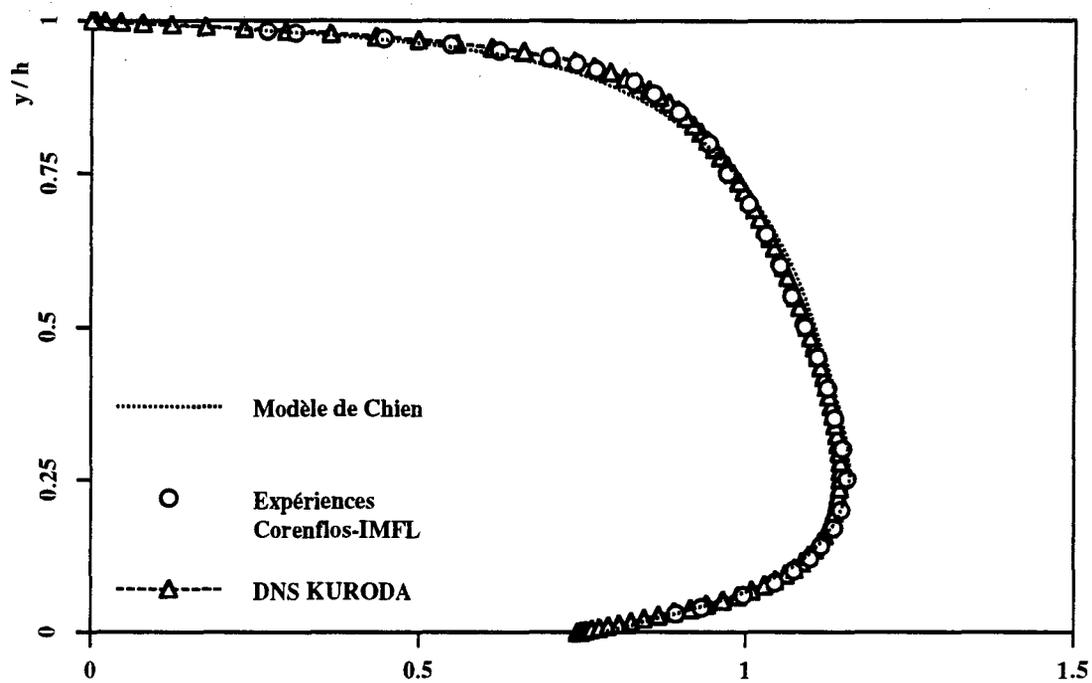


figure 11.9: profil de vitesse

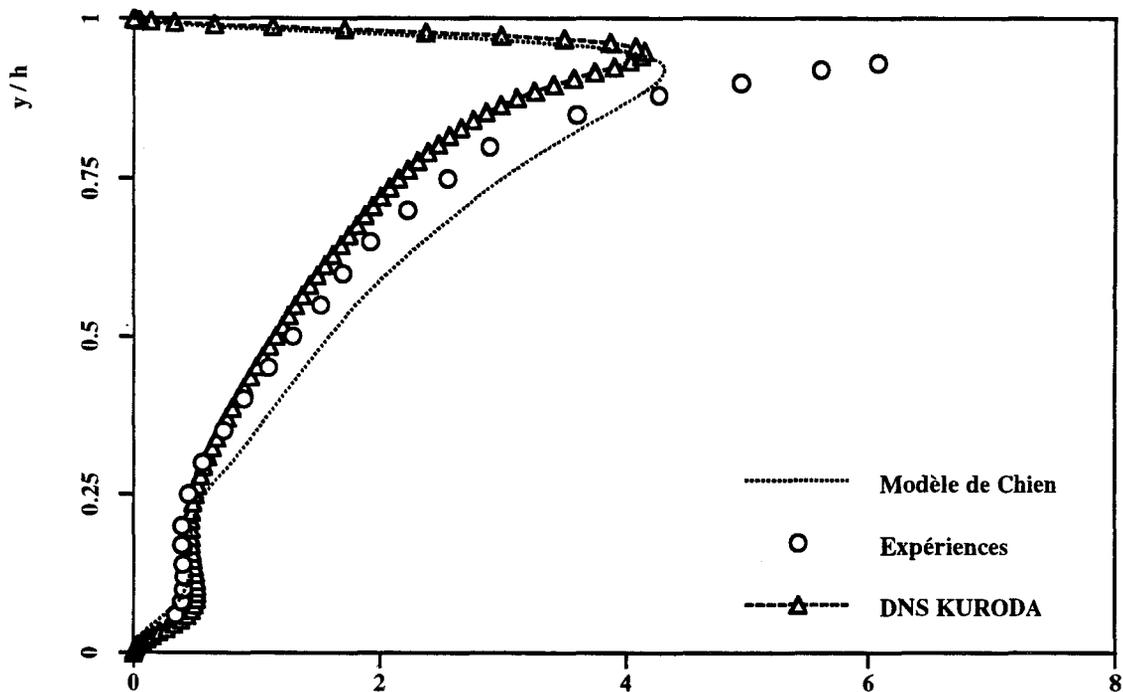


figure 11.10: profil d'énergie turbulente

**Predictions de U et k
pour l'écoulement de type Poiseuille a
Req = 5000**

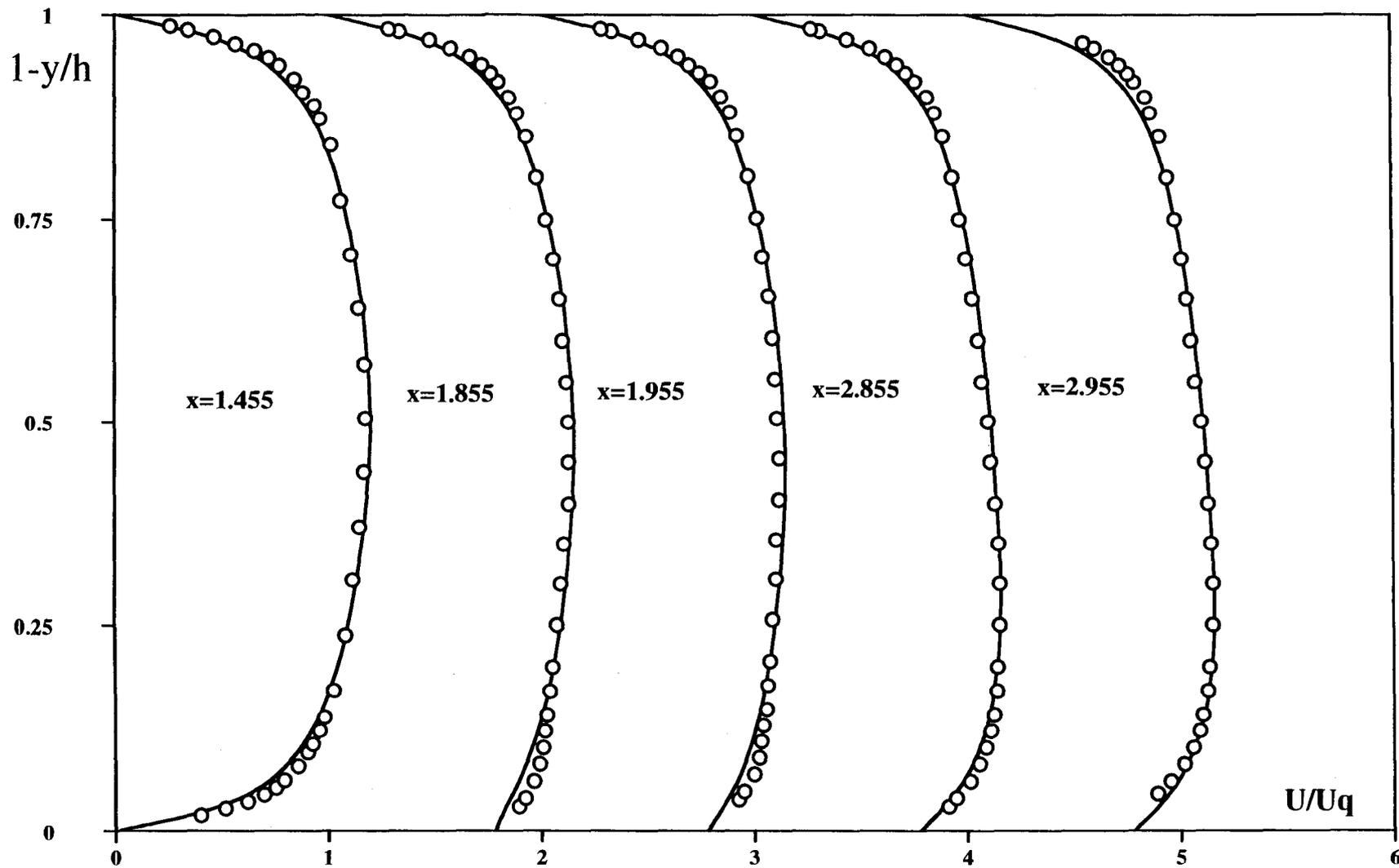


figure 11.11: Etablissement de la vitesse moyenne pour l'écoulement de type Poiseuille
 $Re=5000$

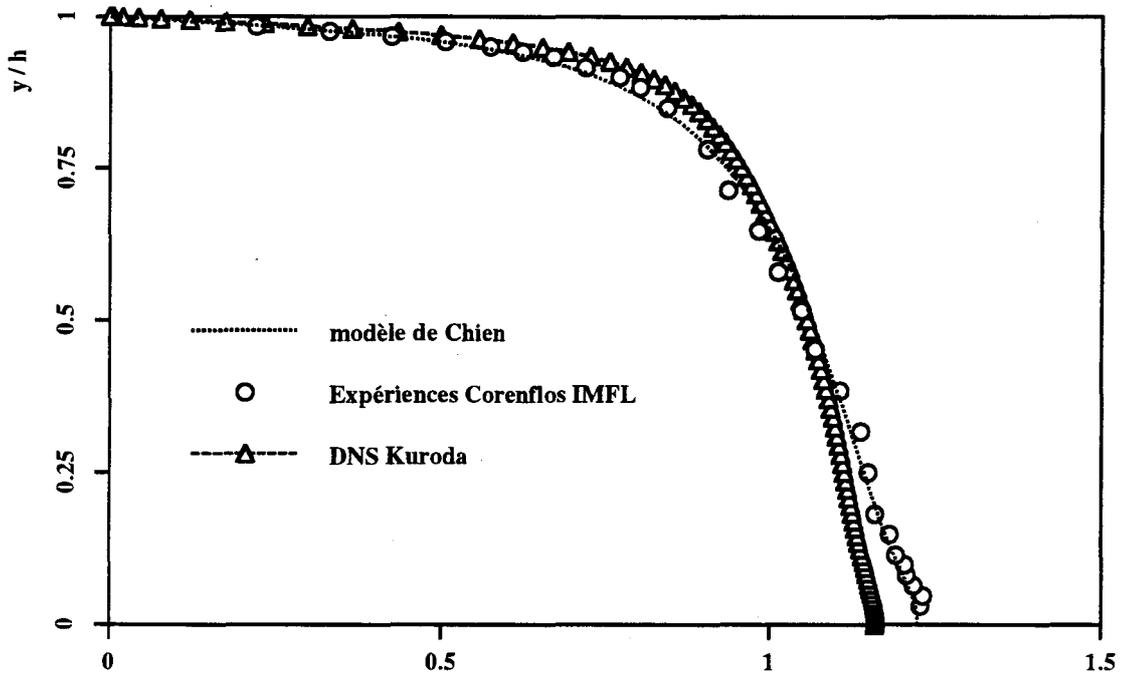


figure 11.12: profil de vitesse

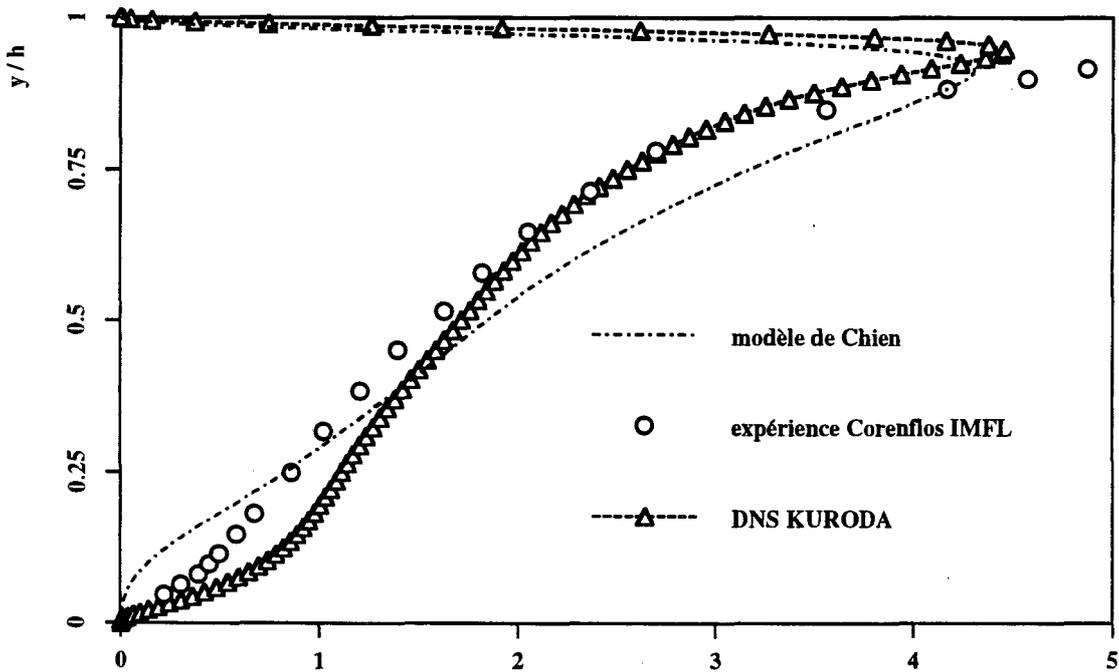


figure 11.13: profil d'énergie turbulente

**Prédiction pour U et k
pour l'écoulement de type intermédiaire à $Re_q = 5000$**

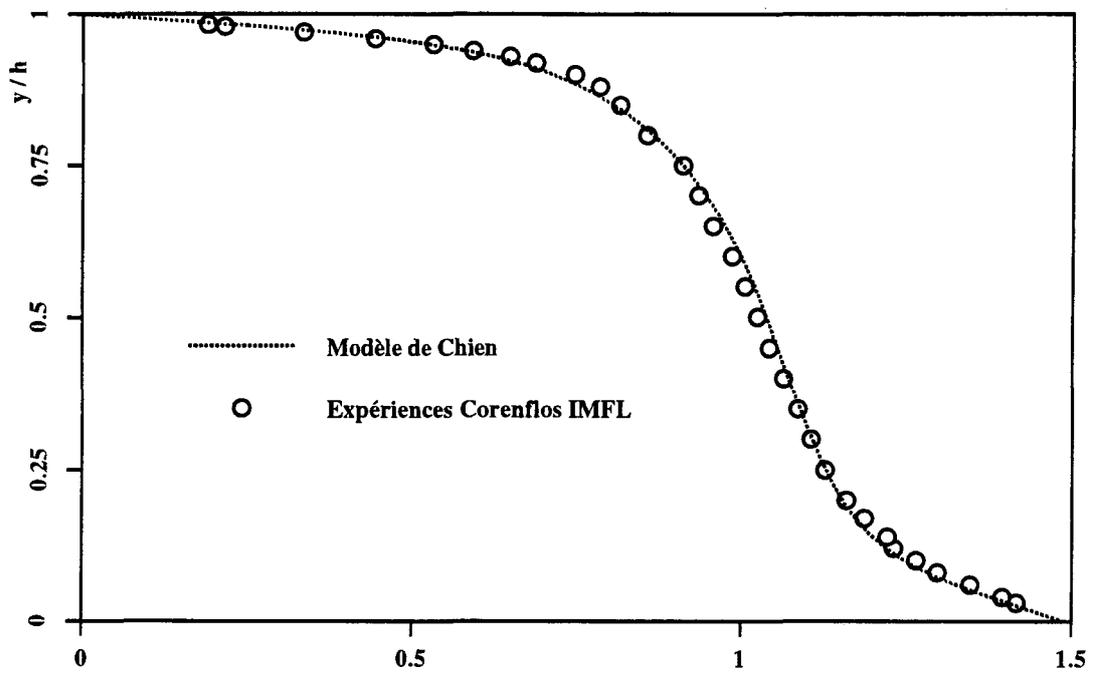


figure 11.14: profil de vitesse

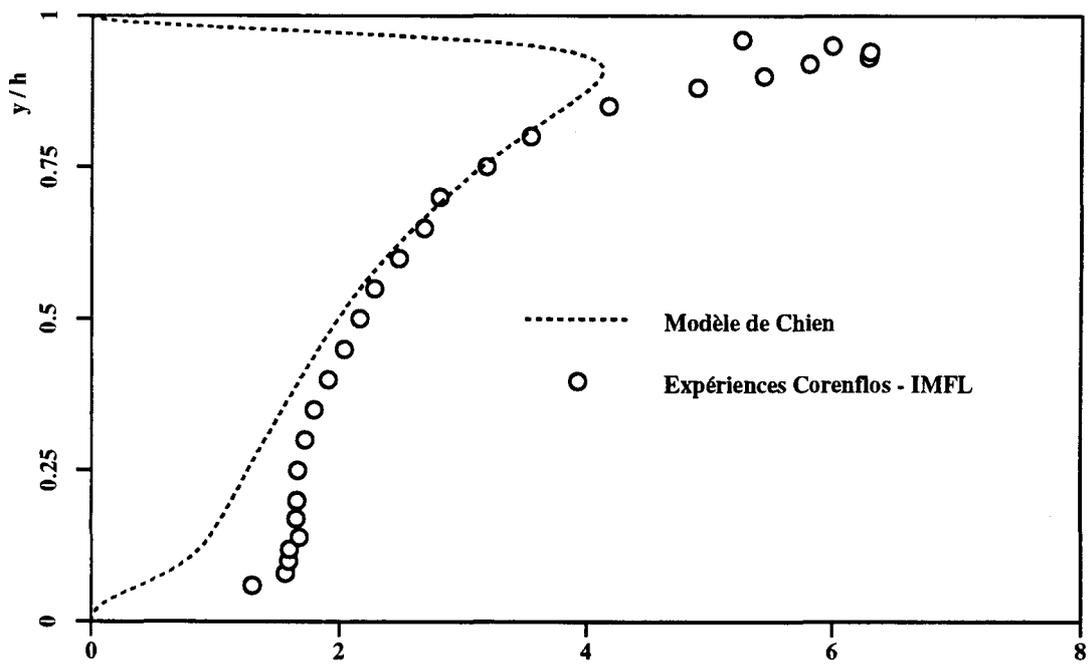


figure 11.15: profil d'énergie turbulente

**Prediction pour U et k
pour l'écoulement de couette a $Re_q = 5000$**

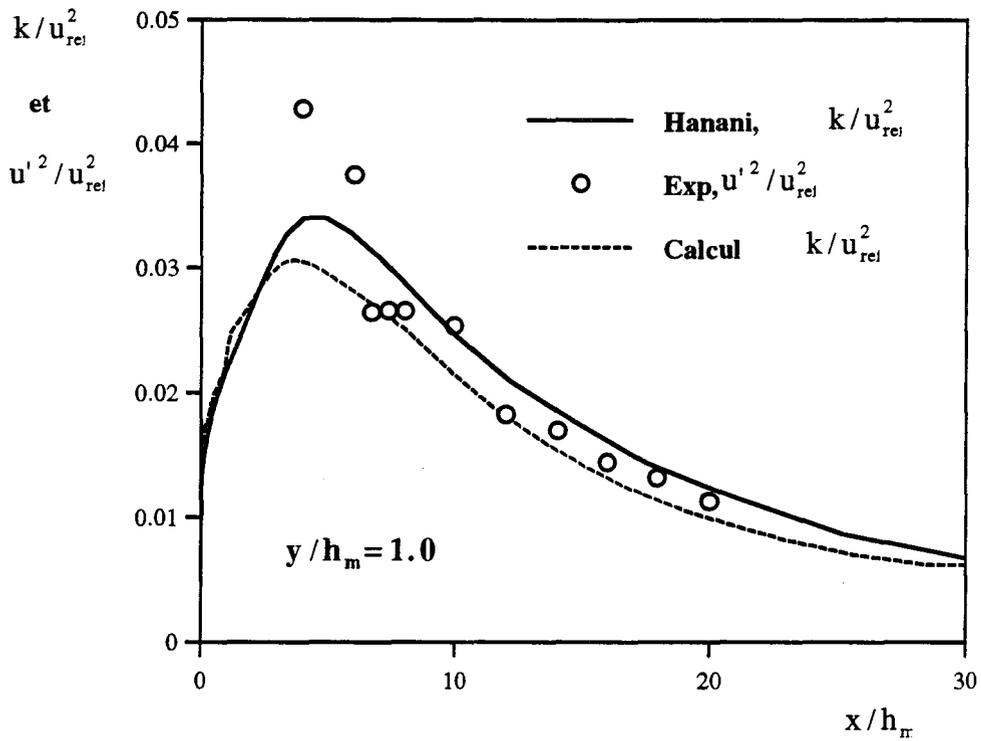


Figure 11.16: Evolution longitudinale d'énergie turbulente et de corrélation normale

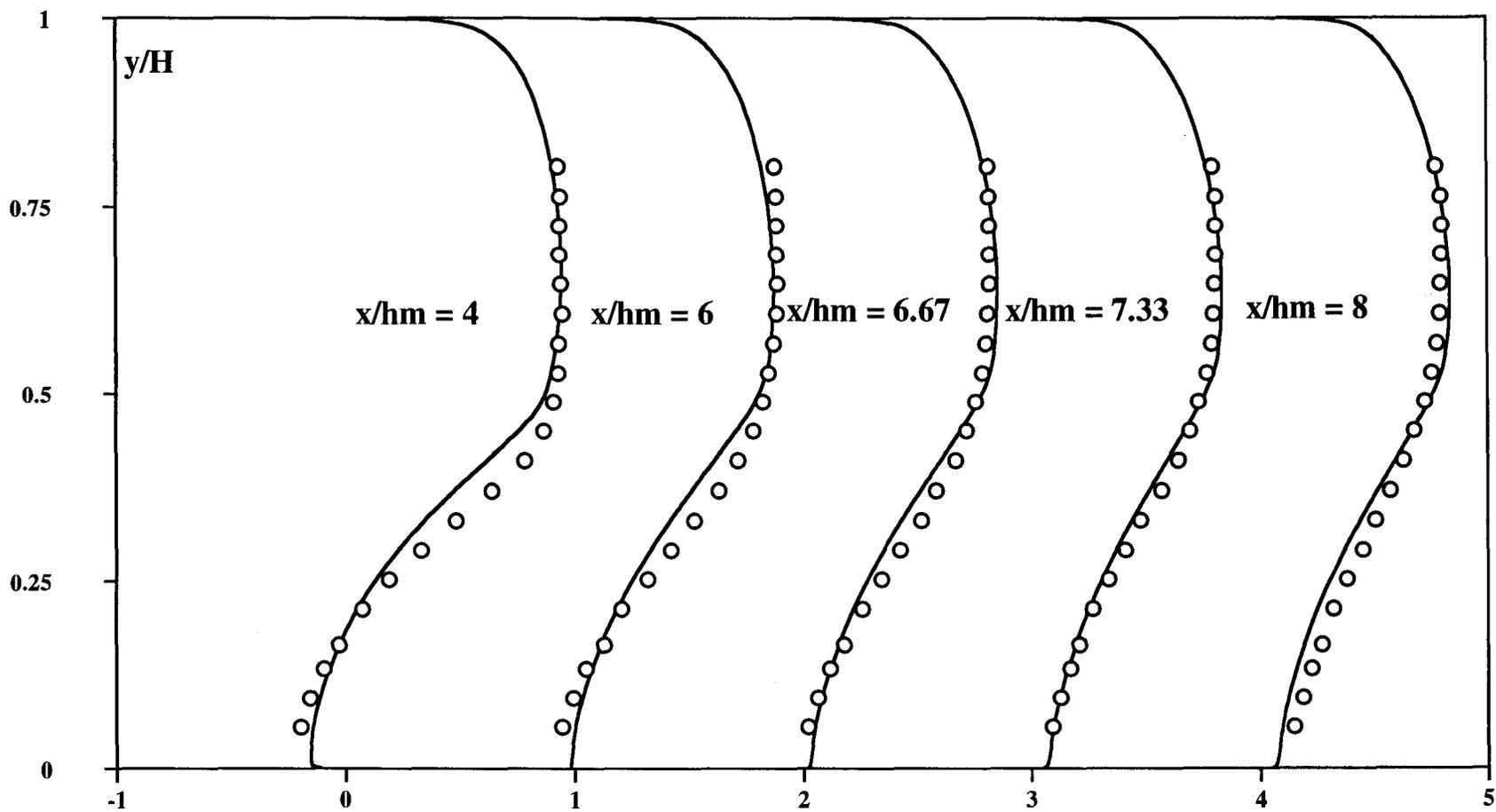


figure 11.17a: Etablissement de la vitesse u pour le cas de la marche

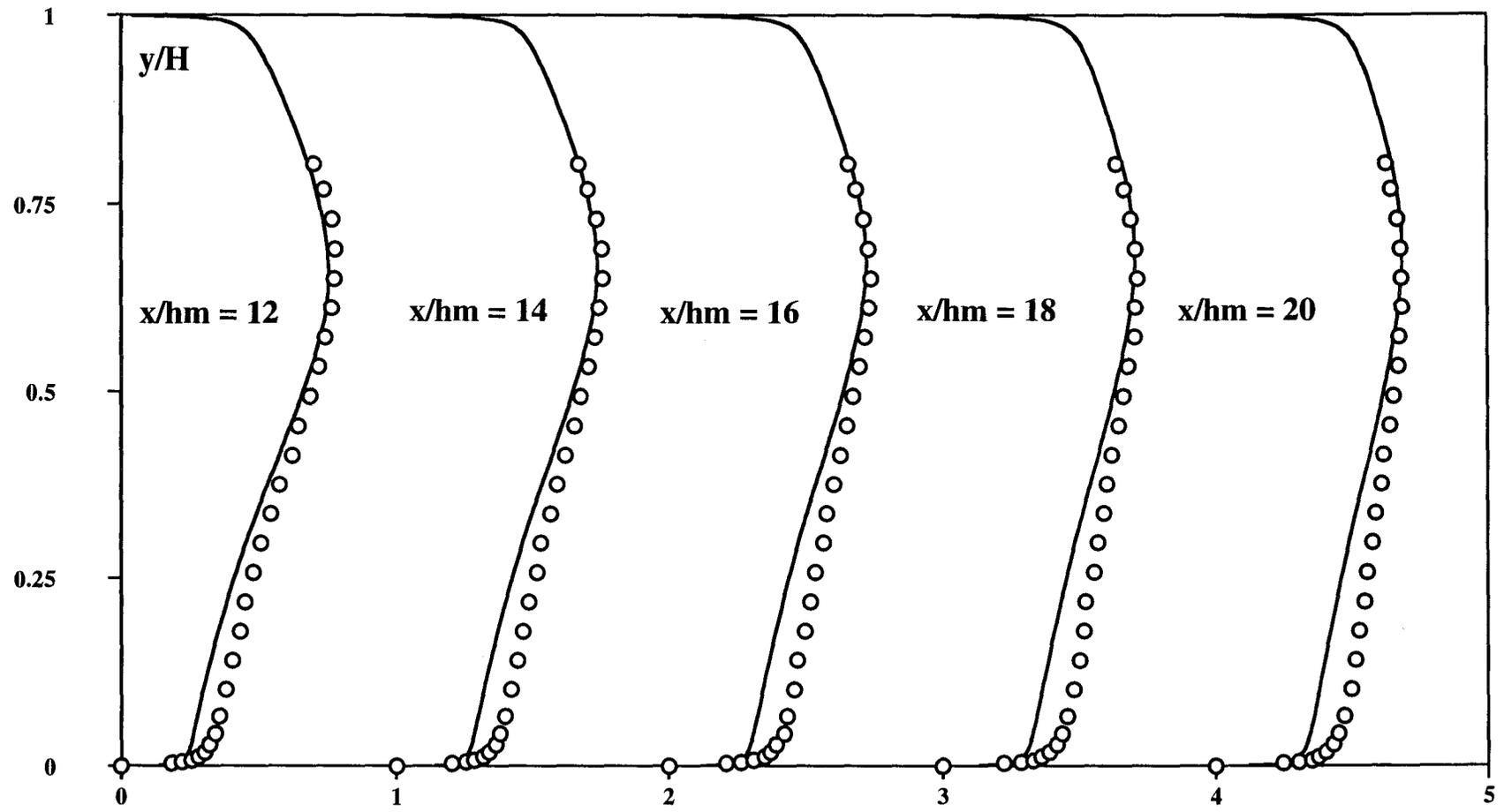


figure 11.17b: Etablissement de la vitesse u dans le cas de marche descendante (suite)

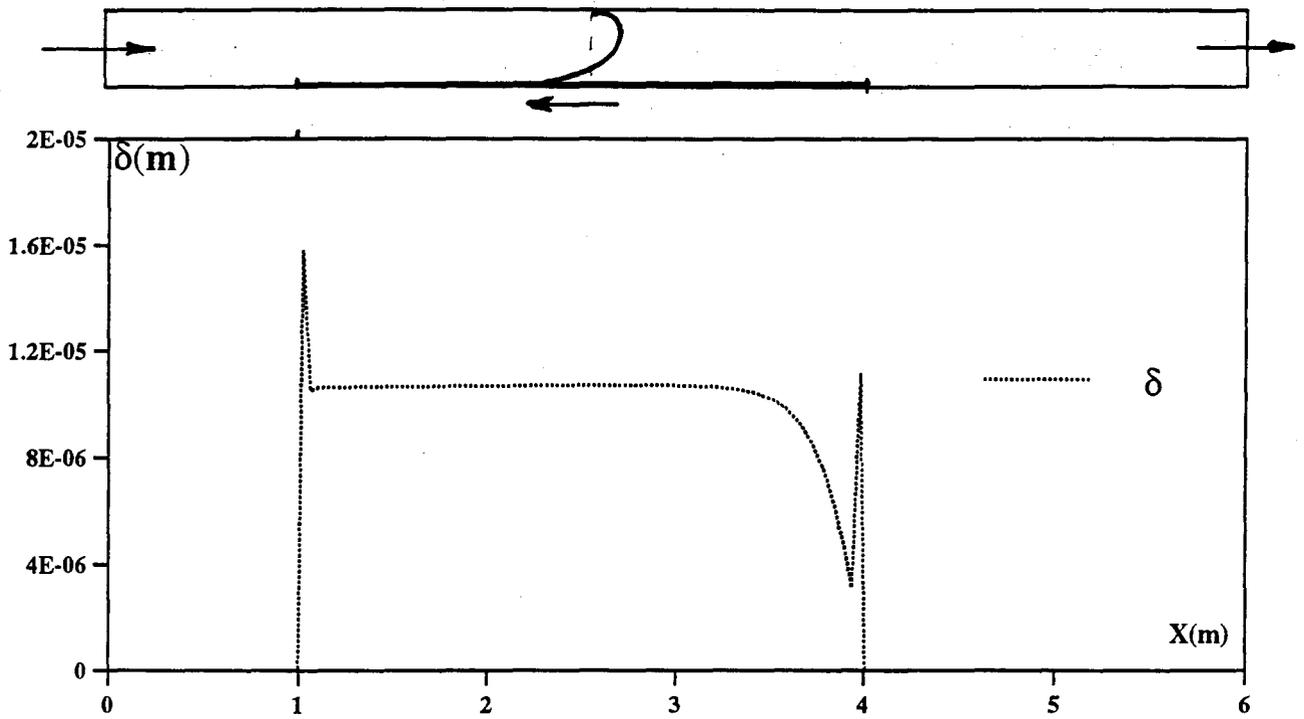


figure 11.18: $V_b = -2$ m/s

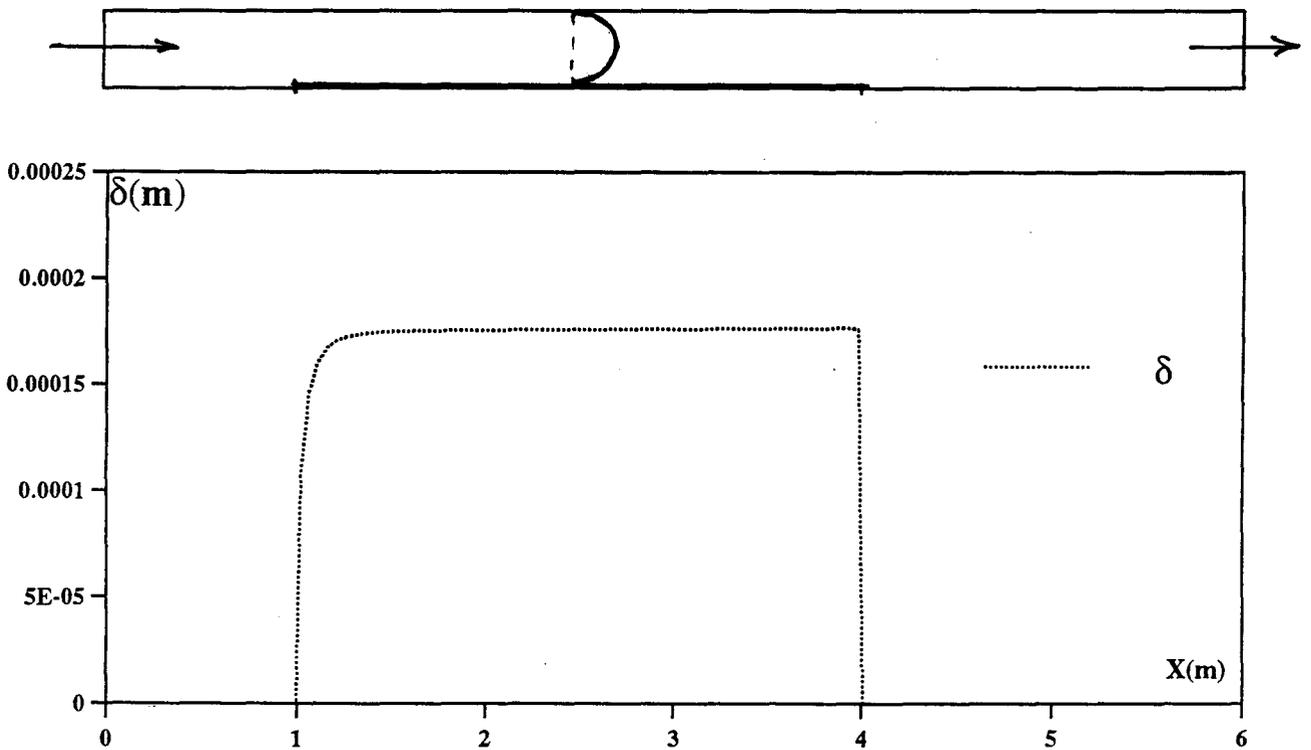


figure 11.19: $V_b = 0$ m/s

Représentation de l'épaisseur de couche limite de diffusion δ en fonction de x

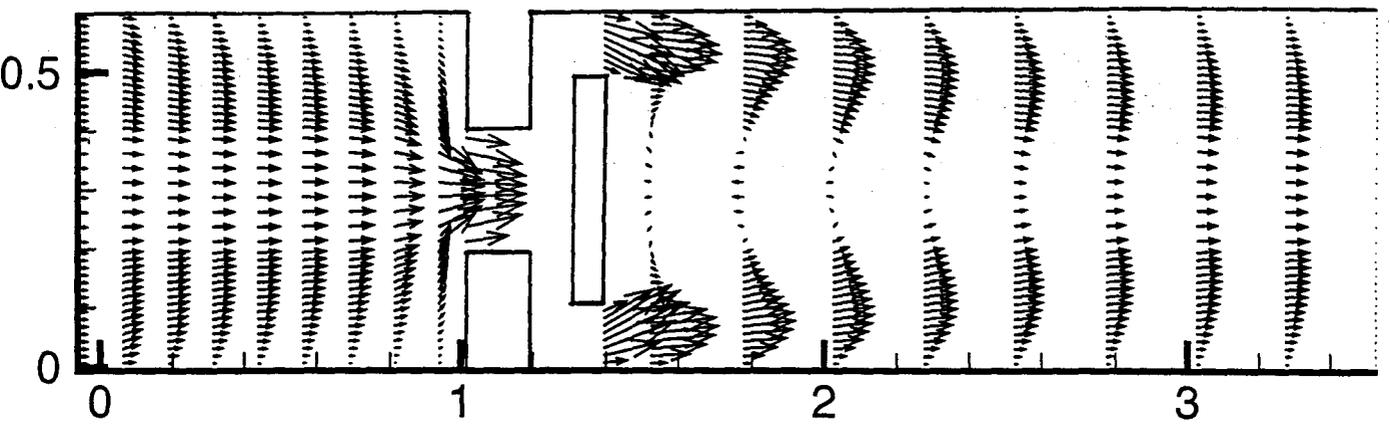


figure12.1: Champ de vitesse pour l'écoulement de canal avec valve anti-retour.

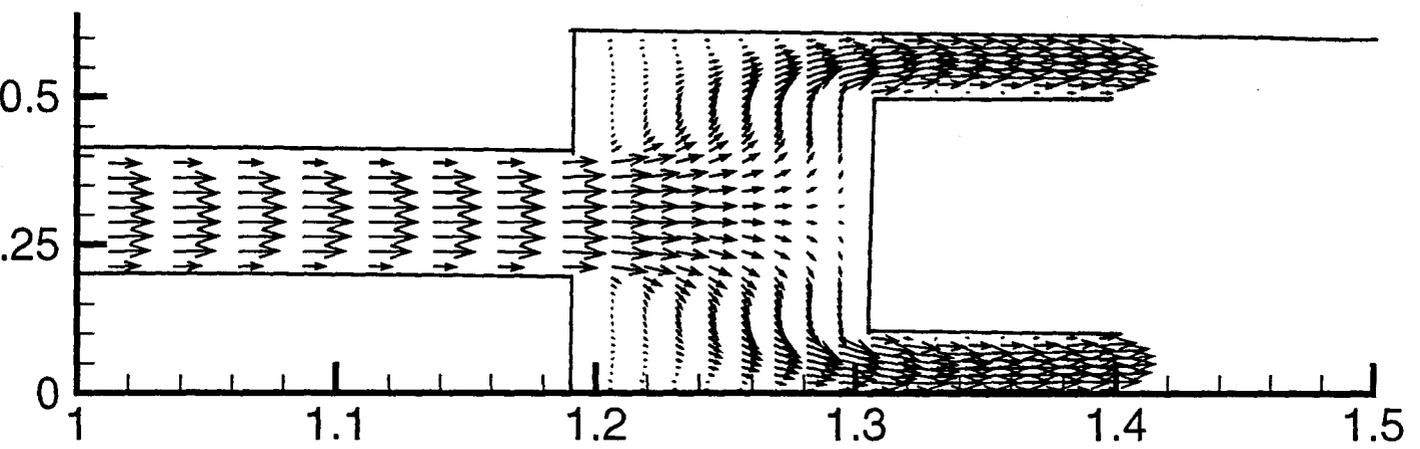


figure12.2: Allongement de la zone centrale pour le champ de vitesse.

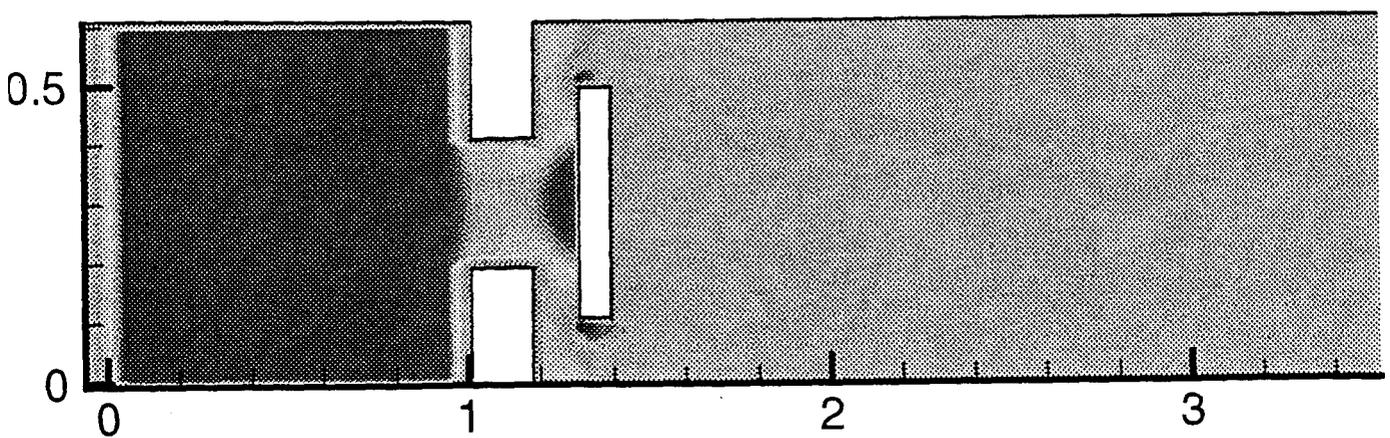


figure12.3: Iso-pression pour l'écoulement de canal plan avec valve anti-retour.