

# THÈSE

Présentée à

## L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DE SCIENCES PHYSIQUES ET DE L'INGÉNIEUR

par **Stéphanie DELAGE SANTACREU**

POUR OBTENIR LE GRADE DE

**DOCTEUR**

SPÉCIALITÉ : **Mécanique**

\*\*\*\*\*

### **Méthode de raffinement de maillage adaptatif hybride pour le suivi de fronts dans des écoulements incompressibles**

\*\*\*\*\*

Soutenue le : 24 juin 2006

Après avis de :

**Mme** Christine Bernardi, Directrice de recherche CNRS, laboratoire J. L. Lions,  
**M** Philippe Angot, Professeur, Université de Provence, LATP,

**Rapporteur**  
**Rapporteur**

Devant la commission d'examen formée de :

**M** Pierre Fabrie, Professeur, Université Bordeaux I, laboratoire MAB  
**M** Jean-Paul Caltagirone, Professeur, Université Bordeaux I, laboratoire TREFLE,  
**M** Stéphane Vincent, Maître de conférence, ENSCPB, laboratoire TREFLE,  
**Mme** Christine Bernardi, directrice de recherche CNRS, laboratoire J. L. Lions,  
**M** Philippe Angot, Professeur, Université de Provence, LATP,  
**M** Jean-Luc Estivalezes, Ingénieur de recherche, ONERA, Toulouse

**Président**  
**Directeur de thèse**  
**CoDirecteur de thèse**  
**Rapporteur**  
**Rapporteur**  
**Examineur**





---

## REMERCIEMENTS

Cette thèse a été réalisée au sein du laboratoire TrEFIE (Transfert Écoulement Fluides Énergétique) dans l'équipe *MFEN* (Mécanique des Fluides et Énergétique Numérique), site ENSCPB, à l'Université Bordeaux I.

Je remercie tout d'abord les directeurs du laboratoire qui se sont succédés, les professeurs Eric Arquis et Jean-Rodolphe Puiggali.

Ma plus profonde gratitude s'adresse au professeur Jean-Paul Caltagirone pour m'avoir donné la possibilité de faire cette thèse (merci pour avoir tenté un deuxième sujet et financement après l'échec du premier) et de m'avoir encadrée en me laissant une grande liberté d'action.

Je tiens à remercier vivement Stéphane Vincent tant pour son encadrement que pour sa motivation sans limites, son acharnement et son optimisme au quotidien.

Je suis reconnaissante à la directrice de recherche Christine Bernardi et au professeur Philippe Angot d'avoir accepté d'être les rapporteurs de cette thèse et de s'être déplacé un samedi pour participer à la soutenance. Les remarques et conseils de Christine Bernardi ainsi que les commentaires de Philippe Angot m'ont permis d'améliorer sensiblement ce document.

Je remercie également le professeur Pierre Fabrie et Jean-Luc Estivalèzes de m'avoir fait l'honneur de participer à ce jury de thèse, surtout un samedi.

J'exprime ma gratitude au professeur Mejdî Azaiez, pour ses conseils avisés, son aide pour élucider le secret du Lagrangien Augmenté dans Aquilon et son soutien. Je remercie profondément mon doudoux, Guillaume Galliéro pour son soutien tout au long de cette thèse, son aide précieuse, son oreille attentive et sa patience en fin de thèse (SOS femme martyrisée par Aquilon)!

Je remercie chaleureusement les membres du laboratoire, Stéphane G, l'amateur de cannelés, pour sa disponibilité et sa gentillesse, Sylvie pour sa bonne humeur et son enthousiasme.

siasme, un second souffle pour Bernard (merci pour Elanion!), Cédric le gentil caliméro qui a osé se frotter à l'*AMR*!, Etienne le petit basque sympa qui a partagé mon bureau, Sylvain le blagueur et amateur de grimpeur, Hamza le capoeiraman dont le réveil avant 9h est fatal, Delphine la serviable et gentille miss Lascaux, la dynamique Aurélie (merci pour Grangé!), Pierre le nouveau papa du labo (encore félicitations!), Marie-Paule l'as de l'économie, Jean le complexe. Et aussi les anciens : Greg le Groslandman et grimpeur doué exilé au Canada (merci pour l'initiation à l'escalade), Nirina pour sa gentillesse à toute épreuve, Claude reconverti dans la pêche en Nouvelle Calédonie et mon ex-collègue de bureau Bobo le surfeur et maître Yoga! Et les nouveaux et expatriés : Aurélie, Erwann, Guillaume, Jérôme et Mathilde (phénomène du labo 2006 d'après JPC). Je remercie également Géraldine pour m'avoir soutenue pendant ces trois ans (Bonne chance pour la deuxième grossesse!).

Je tiens à remercier toute ma famille, oncles et tantes, cousin et cousines et plus particulièrement ma maman et ma sœur Sabrina pour m'avoir soutenue, encouragée et m'avoir permis d'arriver jusque là. Mes pensées vont aussi à mes grands-parents, décédés, qui ont toujours cru en moi et m'ont permis d'avancer.

# Table des matières

<b>Nomenclature</b>	<b>1</b>
<b>1 Partie théorique</b>	<b>7</b>
1.1 Modélisation des équations de conservation . . . . .	7
1.1.1 Equations de Navier-Stokes . . . . .	7
1.1.2 Equation de conservation de l'énergie . . . . .	12
1.1.3 Equation de transport d'espèce . . . . .	13
1.2 Méthodes numériques . . . . .	13
1.2.1 Méthode des volumes finis . . . . .	13
1.2.2 Discrétisation et résolution de l'équation de l'énergie et de transport d'espèces . . . . .	15
1.2.3 Discrétisation et résolution de l'équation de Navier-Stokes . . . . .	17
1.2.4 Discrétisation et résolution du transport d'interface . . . . .	21
1.2.5 Traitement des limites . . . . .	23
1.3 Mise en place de l' <i>AMR</i> . . . . .	26
1.3.1 Stratégies de maillage existantes . . . . .	28
1.3.2 La méthode One-Cell Local Multigrid <i>OCLM</i> . . . . .	32
<b>2 Améliorations apportées à la méthode <i>OCLM</i></b>	<b>39</b>
2.1 Principe de connexion . . . . .	39
2.1.1 Traitement des nœuds scalaires . . . . .	40
2.1.2 Traitement des nœuds vitesses . . . . .	44
2.1.3 Traitement des nœuds par équation . . . . .	50
2.2 Gestion dynamique de l' <i>AMR</i> . . . . .	53
2.3 Opérateurs de prolongement . . . . .	60
2.3.1 Interpolations non conservatives . . . . .	60

2.3.2	Interpolations conservatives pour les nœuds scalaires . . . . .	68
2.4	Opérateurs de restriction . . . . .	71
2.5	Traitement des nœuds fins aux limites physiques . . . . .	73
2.6	Contribution de l' <i>AMR</i> dans un solveur implicite . . . . .	76
2.7	<i>AMR</i> et lagrangien augmenté . . . . .	78
<b>3</b>	<b>Résultats sur les équations scalaires</b>	<b>79</b>
3.1	Etude d'une tache cisailée par un tourbillon dans une cavité <i>2D</i> en mono- phasique . . . . .	79
3.1.1	Influence du type d'interpolation appliqué à la concentration massique	80
3.1.2	Capacité de l' <i>AMR</i> à réduire la diffusion numérique. . . . .	83
3.1.3	Choix du critère de raffinement . . . . .	86
3.1.4	Calcul du champ de vitesse . . . . .	86
3.1.5	Ordre de convergence de la méthode <i>AMR</i> . . . . .	91
3.1.6	Performances en temps CPU et mémoire de la méthode de raffinement	98
3.2	Etude d'une tache cisailée par un tourbillon dans une cavité <i>2D</i> en diphasique	107
3.2.1	Couplage <i>AMR</i> et <i>VOF - CIAM</i> . . . . .	107
3.2.2	Couplage <i>AMR</i> et <i>VOF - TVD</i> . . . . .	112
3.3	Etude d'un cas de thermique en <i>2D</i> : cas de Whitaker . . . . .	114
3.3.1	Traitement des limites scalaires . . . . .	115
3.3.2	Méthode <i>AMR</i> et précision des solutions . . . . .	117
3.3.3	Choix des opérateurs de restriction et de prolongement . . . . .	122
3.4	Tache cisailée dans un écoulement monophasique par un tourbillon dans une cavité <i>3D</i> . . . . .	130
3.4.1	Tache cisailée autour de l'axe <i>y</i> . . . . .	133
3.4.2	Tache cisailée autour de l'axe <i>z</i> . . . . .	137
3.4.3	Tache cisailée autour de l'axe <i>x</i> . . . . .	138
3.5	Tache cisailée par un champ tournant périodique dans un écoulement di- phasique <i>3D</i> . . . . .	140
<b>4</b>	<b>Résultats sur les équations de Navier-Stokes</b>	<b>143</b>
4.1	Résultats préliminaires sur un écoulement de Poiseuille 2D . . . . .	143
4.1.1	Influence du degré du polynôme d'interpolation . . . . .	145
4.1.2	Influence du nombre de niveaux de raffinement . . . . .	146
4.1.3	Influence du coefficient de raffinement . . . . .	147

---

4.2	Tourbillon de Green-Taylor $2D$ . . . . .	149
4.2.1	Interpolations non conservatives et condition d'incompressibilité . .	150
4.2.2	Précision du champ de vitesse . . . . .	151
4.2.3	Performances de l' <i>AMR</i> en taille mémoire et temps <i>CPU</i> . . . . .	153
4.3	Ruptures de barrage en $2D$ . . . . .	157
4.3.1	Rupture de barrage sur fond sec . . . . .	157
4.3.2	Rupture de barrage sur fond mouillé . . . . .	162
<b>Annexes</b>		<b>173</b>
<b>A Solutions analytiques</b>		<b>173</b>
A.1	Écoulement de Poiseuille $2D$ . . . . .	173
A.2	Conduction $2D$ dans un rectangle dont les bords sont chauffés à des températures différentes . . . . .	174
A.2.1	Cas où $T_f$ est constante par morceaux . . . . .	174
A.2.2	Cas où $T_f$ est de classe $C^1$ . . . . .	176
<b>B Interpolation non conservative et éléments finis</b>		<b>179</b>
<b>C Interpolation conservative pour les nœuds vitesses</b>		<b>189</b>
C.1	Vitesses horizontales . . . . .	189
<b>Bibliographie</b>		<b>193</b>



# Nomenclature

## Variables de la mécanique des fluides diphasique

### Éléments physiques

$c_p$	: capacité calorifique massique à pression constante ...	$W.m^{-2}.K^{-1}$
$p$	: pression .....	$Pa$
$t$	: temps .....	$s$
$E_c$	: énergie cinétique .....	$J$
$E_i$	: énergie interne .....	$J$
$F_{TS}$	: force de tension superficielle .....	$N$
$\vec{g}$	: accélération de gravité .....	$m.s^{-2}$
$\vec{V}$	=: vecteur vitesse .....	$m.s^{-1}$
$(u, w, v)$		
$\vec{V}_{int}$	: vecteur vitesse .....	$m.s^{-1}$
$S_p$	: concentration massique .....	$kg.m^{-3}$
$T$	: température .....	$K$
$C$	: fonction de phase, fraction volumique ou fonction couleur .....	
$D$	: coefficient de diffusion moléculaire .....	$m^2.s^{-1}$
$Re$	: nombre de Reynolds .....	$\frac{\rho L u_0}{\mu}$
$\mu$	: viscosité dynamique .....	$kg.m^{-1}.s^{-1}$
$\mu_T$	: viscosité dynamique turbulente .....	$kg.m^{-1}.s^{-1}$
$\kappa$	: courbure de l'interface .....	$m^{-1}$
$\rho$	: masse volumique .....	$kg.m^{-3}$
$\Phi$	: inconnue générique .....	
$\lambda$	: coefficient de conduction thermique .....	$W.m^{-1}.K^{-1}$

### Éléments mathématiques

$\Omega$	:	domaine d'étude .....	
$\partial\Omega$	:	Frontière du domaine d'étude .....	
$x, y, z$	:	coordonnées de $\mathfrak{R}^3$ .....	$m$
$d\Sigma$	:	élément de surface .....	$m^2$
$d\Omega$	:	élément de volume .....	$m^3$
$\vec{n}$	:	vecteur unitaire normal à $d\Sigma$ et de direction sortante à $d\Omega$ .....	

### Éléments numériques

$A$	:	matrice des coefficients de discrétisation .....	
$B$	:	second membre .....	
$G$	:	maillage monogrille .....	
$G_L$	:	$L^{ime}$ niveau de raffinement .....	
$i, j, k$	:	indices de discrétisation .....	
$NT$	:	nombre d'itérations en temps .....	
$N$	:	nombre de nœuds d'un maillage .....	
$RAF$	:	coefficient de raffinement .....	
$\Delta x, \Delta z, \Delta y$	:	pas d'espace suivant x, z, y .....	$m$
$\Delta x_L, \Delta z_L, \Delta y_L$	:	pas d'espace de niveau $G_L$ suivant x, z, y .....	$m$
$\Delta t$	:	pas de temps .....	$s$



# Introduction

Une des problématiques actuelles en mécanique des fluides est la description des écoulements incompressibles multiphasiques et multi-échelles. La simulation numérique de tels écoulements est très difficile à réaliser de part leur complexité et leurs besoins en ressources informatiques.

Plusieurs voies sont envisageables pour tenter d'apporter des solutions. Celles-ci peuvent bien entendu être couplées. Parmi elles, on peut citer la simplification des modèles (comme l'utilisation de la LES en turbulence [SAGAUT 06]), l'amélioration des méthodes et des schémas numériques, l'optimisation des ressources nécessaires en adoptant des stratégies de maillage (multigrille dans [TROTTEBERG 01], raffinement de maillage adaptatif développé par [BERGER 84]), ou la répartition des ressources demandées sur plusieurs processeurs (parallèle : voir [DONGARRA 03]).

On s'intéresse plus particulièrement dans cette thèse au suivi d'interface dans des écoulements incompressibles diphasiques et au suivi d'espèces inertes faiblement diffusives dans des écoulements incompressibles monophasiques. Ces écoulements font apparaître localement des structures d'échelles différentes.

Une des difficultés apparaissant lors du suivi d'espèces faiblement diffusives est l'introduction de diffusion numérique inhérente aux schémas utilisés, masquant la diffusion physique (moléculaire et turbulente). Une solution est d'utiliser un maillage très fin, de façon à ce que la diffusion physique se propage sur une distance plus grande que la diffusion numérique. Cependant, le nombre de points à résoudre devient très vite prohibitif en 3D, et il est alors impossible de traiter des cas réalistes compte tenu de la taille mémoire et du temps de simulation nécessaires. Il est donc indispensable de mettre en place une stratégie de maillage pour optimiser les deux facteurs limitants : taille mémoire et temps de simulation.

Une méthode de raffinement de maillage adaptatif semble donc adaptée au type de problèmes étudiés dans la mesure où cette approche consiste à augmenter localement la

densité de points d'un maillage donné, dans les zones où il est indispensable de calculer la solution avec plus de précision que dans le reste du domaine. C'est l'approche qu'on a choisi d'utiliser.

[BERGER 84] ont été les premiers à introduire le concept de raffinement de maillage pour la résolution des équations différentielles partielles hyperboliques en 1984. [BERGER 89] ont ensuite proposé une version adaptative du raffinement de maillage en  $2D$  pour les lois de conservation et ont montré l'efficacité de l'*AMR* pour la dynamique des gaz en 1989. Celle-ci a été étendue au  $3D$  deux ans après par [BELL 91]. L'approche *AMR* a ensuite été adaptée à des domaines variés de la physique. On peut citer le suivi de multi-espèces dans un milieu poreux pour déterminer les interactions entre différentes échelles ([TRANGENSTEIN 02]), le suivi de formation et propagation des microstructures lors du phénomène de solidification ([PROVATAS 99]), le traquage des instabilités rencontrées dans des problème d'interaction laser-plasma ([DORR 02]), ou encore l'étude de l'interaction entre une onde de choc et un nuage de particules dans le milieu interstellaire en astrophysique ( cf [KLEIN 94]). La méthode *AMR* a aussi été étendue aux écoulements de fluides incompressibles diphasiques et à surface libre pour décrire les interfaces le plus précisément possible (voir [MINION 97], [SUSSMAN 98], [SUSSMAN 99], [HAY 05], [LOSASSO 06]...). Au laboratoire *TREFLE*, en s'inspirant des méthodes *AMR* et d'une méthode de multigrille adaptatif développé par [KHADRA 95] et [KHADRA 96], [VINCENT 99] a implémenté une méthode de raffinement de maillage adaptée au maillage décalé ( méthode *OCLM*, i. e. One Cell Local Multigrid).

Au cours de cette thèse, on a amélioré et optimisé la méthode *OCLM* développée par [VINCENT 00]. Ce travail a consisté tout d'abord à élargir le type d'interpolations non conservatives (passage du  $Q_1$  préexistant à l'interpolation  $Q_2$  et binaire) pour traiter certains nœuds, puis à implémenter la méthode pour éviter les décalages de temps entre les nœuds discrétisés et ceux interpolés. On a ensuite effectué un travail de connexion pour que chaque niveau de raffinement soit considéré, du point de vue solveur, comme une entité à part entière. On s'est enfin intéressé à l'amélioration des critères de raffinement et à l'optimisation de la méthode *OCLM*.

Dans le premier chapitre, on explicitera les équations utilisées ainsi que les méthodes numériques associées pour les résoudre. On détaillera aussi les stratégies de maillage existantes et les raisons qui nous ont poussées à améliorer la méthode *OCLM*.

Dans le deuxième chapitre, on détaille la technique utilisée, pour des écoulements incompressibles. La mise en place de cette méthode dans le code *Aquilon*, développé au

laboratoire *Trefle*, se décompose en trois parties : La première concerne la structure même du maillage *AMR*, qui est constitué d'un emboîtement strict des niveaux de raffinement; chaque niveau est constitué d'un ensemble d'entités, les cellules *AMR*. Il a donc fallu connecter les nœuds d'un même niveau de raffinement (c'est-à-dire connecter ces cellules), imposer des conditions aux limites de chaque niveau fin sous forme d'interpolations, mais aussi connecter les niveaux entre eux afin que le maillage *AMR* constitue une seule entité. La deuxième partie consiste à impliciter la méthode *AMR*, c'est-à-dire à introduire la contribution des nœuds traités par interpolation dans le solveur implicite. La résolution de l'ensemble des nœuds du maillage localement raffiné en une seule étape permet de conserver une bonne précision de la solution. La troisième partie est consacrée à la recherche de critères physiques de raffinement-déraffinement de maillage les plus adaptés au problème traité.

Le troisième chapitre est consacré à la validation de la méthode *AMR* en deux et trois dimensions d'espace pour les équations de conservation d'énergie et de transport d'espèces inertes, sur des cas académiques : fluide chauffé par une extrémité en  $2D$ , tache cisailée autour d'un axe de rotation en  $2D$  et  $3D$ . On montre l'efficacité de cette approche pour réduire la diffusion numérique des schémas utilisés ainsi que ses performances en termes de gains en taille mémoire et en temps *CPU* en  $2D$  et  $3D$ .

Dans le quatrième chapitre, la méthode est validée pour les équations de Navier-Stokes en  $2D$  sur des cas académiques, comme un écoulement de Poiseuille et le tourbillon de Green-Taylor. Une confrontation avec des résultats expérimentaux a été menée sur des cas de rupture de barrage.



# Chapitre 1

## Partie théorique

### 1.1 Modélisation des équations de conservation

On s'intéresse à des écoulements multiphasiques avec transport d'espèces inertes. Ceci se traduit par le couplage des équations de conservation de la masse, de la quantité de mouvement et de la concentration massique. On s'intéressera aussi à l'équation de conservation d'énergie.

#### 1.1.1 Equations de Navier-Stokes

On considère un domaine  $\Omega$  de frontières  $\partial\Omega$  dans lequel évolue un fluide newtonien. Dans le cas où deux fluides évoluent dans le domaine  $\Omega$ , on note  $\Omega_1$  (respectivement  $\Omega_2$ ) le domaine occupé par le *fluide*<sub>1</sub> (respectivement *fluide*<sub>2</sub>).

La zone séparant les deux milieux porte un nom différent suivant la nature des deux milieux. On appelle interface, la surface qui limite deux phases condensées et surface libre celle qui sépare une phase condensée d'un gaz. Pour éviter d'alourdir le texte, on utilisera le mot interface pour désigner indifféremment l'interface ou la surface libre. Celle-ci est considérée comme une surface de contact de mesure nulle à l'échelle du milieu continu.

La frontière  $\Gamma$ , représentant l'interface, délimite  $\Omega_1$  et  $\Omega_2$ .

Soit un écoulement incompressible, la conservation de la masse se réduit à l'équation :

$$\vec{\nabla} \cdot \vec{V} = 0 \tag{1.1}$$

L'équation de conservation de la quantité de mouvement ou principe fondamental de la dynamique s'écrit :

$$\rho \left( \frac{\partial \vec{V}}{\partial t} + (\vec{V} \cdot \vec{\nabla}) \vec{V} \right) = \rho \vec{g} - \vec{\nabla} p + \vec{\nabla} \cdot (\mu (\vec{\nabla} \vec{V} + \vec{\nabla} \vec{V}^T)) \quad (1.2)$$

$\rho \frac{\partial \vec{V}}{\partial t}$  représente la variation temporelle,  $\rho (\vec{V} \cdot \vec{\nabla}) \vec{V}$  le terme d'advection ou d'inertie,  $\rho \vec{g}$  la force de gravité,  $\vec{\nabla} p$  les forces de pression et  $\vec{\nabla} \cdot (\mu (\vec{\nabla} \vec{V} + \vec{\nabla} \vec{V}^T))$  les forces dues au tenseur des contraintes de viscosité. Le système (1.1)-(1.2), appelé équations de Navier-Stokes, permet de décrire l'écoulement d'un fluide newtonien incompressible (cas monophasique).

### Écoulement diphasique

Pour un écoulement diphasique incompressible, c'est-à-dire faisant intervenir deux fluides séparés par une interface, les équations de Navier-Stokes restent vérifiées dans chacun des fluides  $fluide_1$  et  $fluide_2$ . On indicera par 1 (respectivement 2) les caractéristiques du  $fluide_1$  (respectivement  $fluide_2$ ). L'écoulement peut être décrit par les équations de Navier-Stokes dans chacune des phases, ce qui donne :

$$\begin{aligned} \vec{\nabla} \cdot \vec{V}_1 &= 0 \quad \text{dans } \Omega_1 \\ \vec{\nabla} \cdot \vec{V}_2 &= 0 \quad \text{dans } \Omega_2 \\ \rho_1 \left( \frac{\partial \vec{V}_1}{\partial t} + (\vec{V}_1 \cdot \vec{\nabla}) \vec{V}_1 \right) &= \rho_1 \vec{g} - \vec{\nabla} p_1 + \vec{\nabla} \cdot (\mu_1 (\vec{\nabla} \vec{V}_1 + \vec{\nabla} \vec{V}_1^T)) \quad \text{dans } \Omega_1 \\ \rho_2 \left( \frac{\partial \vec{V}_2}{\partial t} + (\vec{V}_2 \cdot \vec{\nabla}) \vec{V}_2 \right) &= \rho_2 \vec{g} - \vec{\nabla} p_2 + \vec{\nabla} \cdot (\mu_2 (\vec{\nabla} \vec{V}_2 + \vec{\nabla} \vec{V}_2^T)) \quad \text{dans } \Omega_2 \end{aligned} \quad (1.3)$$

On remarque que le système (1.3) permet de prendre en compte l'évolution de chaque fluide, mais de manière séparée. Or ces deux fluides étant en contact via l'interface, ils évoluent de façon couplée. L'ajout de conditions à l'interface, traduisant la continuité du flux de masse et du flux de quantité de mouvement, permet de connecter les deux phases. On suppose que l'interface se déplace à une vitesse  $\vec{V}_{int}$ .

Dans le cas de deux fluides non miscibles, la continuité du flux de masse donne la relation suivante :

$$[\rho (\vec{V} - \vec{V}_{int})] \cdot \vec{n} = 0 \quad (1.4)$$

Dans le cas où il n'y a pas de changement de phase, l'interface se déplace perpendiculairement à elle-même et donc :

$$\vec{V}_{int} = \vec{V}_1 \cdot \vec{n} = \vec{V}_2 \cdot \vec{n}$$

. Soit :

$$[\vec{V}] \cdot \vec{n} = 0 \quad (1.5)$$

La continuité du flux de quantité de mouvement s'obtient en appliquant le principe fondamental de la dynamique sur un petit volume de contrôle entourant l'interface.

Celle-ci est soumise à des forces volumiques (gravité), surfaciques (pression, contraintes de viscosités et interfaciales). La tension interfaciales,  $\gamma$ , traduit la cohésion de l'interface et la force associée,  $\vec{\Sigma}_{int}$ , a pour expression

$$\vec{\Sigma}_{int} = 2\gamma\kappa\vec{n} + \vec{\nabla}_s\gamma$$

avec  $\kappa = -\frac{1}{2}\vec{\nabla}_s\vec{n}$ , le rayon de courbure moyen et  $\vec{\nabla}_s$  l'opérateur gradient restreint à la surface de l'interface.

L'épaisseur de l'interface étant très faible, les forces associées aux forces volumiques sont négligeables devant celles associées aux forces surfaciques et linéiques. On suppose de plus que le *fluide*<sub>1</sub> ne glisse pas par rapport au *fluide*<sub>2</sub>, ce qui annule la composant tangentielle (le long de la surface de l'interface) de la tension interfaciale. La conservation de la quantité de mouvement à travers un petit volume de contrôle autour de l'interface donne donc la relation de saut :

$$[-p\vec{n} + \mu(\vec{\nabla}(\vec{V} - \vec{V}_{int}) + \vec{\nabla}(\vec{V} - \vec{V}_{int})^T) \cdot \vec{n}] = 2\gamma\kappa\vec{n} \quad (1.6)$$

La résolution des équations du système (1.3), (1.4) et (1.6) permet de décrire un écoulement incompressibles de deux fluides non miscibles (voir [DELHAYE 74]).

### Modèle 1-fluide

La formulation 1-fluide d'un écoulement diphasique consiste à modéliser l'ensemble  $\{\text{fluide}_1, \text{fluide}_2\}$  par un seul fluide 'hybride'. Celle-ci a été introduite par [KATAOKA 86] en 1986 et permet de résoudre des problèmes diphasiques en utilisant des outils numériques qui ont été développés pour des cas monophasiques.

Soit  $\chi^k$  l'indicateur de phase  $k$ ,  $k \in [1, 2]$ , défini par  $\chi^k = 1$  dans la phase  $k$  et  $\chi^k = 0$

sinon. Notons  $\delta_s$  le Dirac centré sur l'interface de surface  $s$ ,  $\vec{n}^k$  la normale sortante à la phase  $k$  et  $\vec{n}$  la normale à l'interface. L'indicateur de phase vérifie les propriétés suivantes :

$$\begin{aligned}\sum_{k=1}^2 \chi^k &= 1 \\ \chi^k \chi^j &= \delta_{kj} \chi^k \\ \vec{\nabla} \chi^k &= -\vec{n}^k \delta_s \\ \frac{\partial \chi^k}{\partial t} &= \vec{V}_{int} \cdot \vec{n}^k \delta_s\end{aligned}$$

La variable 1-fluide  $\Phi$  est définie comme la somme des phases  $k$  pondérées par l'indicateur de phase  $\chi^k$ , i.e. :

$$\Phi = \sum_{k=1}^2 \chi^k \Phi^k.$$

Ainsi la conservation de la masse pour la variable 1-fluide devient, après sommation sur toutes les phases,

$$\begin{aligned}\vec{\nabla} \cdot \left( \sum_{k=1}^2 \chi^k \vec{\nabla} \vec{V}^k \right) &= \sum_{k=1}^2 \vec{\nabla} \vec{V}^k \cdot \vec{\nabla} \chi^k \\ &= - \sum_{k=1}^2 \vec{\nabla} \vec{V}^k \cdot \vec{n}^k \delta_s \\ &= -[\vec{V}] \cdot \vec{n}\end{aligned}\tag{1.7}$$

En utilisant la relation de saut (1.5) l'équation (1.7) se réduit à :

$$\vec{\nabla} \cdot \vec{V} = 0\tag{1.8}$$

Après sommation sur toutes les phases de l'équation de conservation de mouvement et utilisation de la relation de saut associée (1.6), on obtient :

$$\begin{aligned}\frac{\partial \rho \vec{V}}{\partial t} + (\vec{V} \cdot \vec{\nabla})(\rho \vec{V}) &= \rho \vec{g} - \vec{\nabla} p + \vec{\nabla} \cdot (\mu (\vec{\nabla} \vec{V} + \vec{\nabla} \vec{V}^T)) + \vec{\nabla}_s \cdot (\gamma \vec{n}^k \delta_s \vec{n}^k) \\ &= \rho \vec{g} - \vec{\nabla} p + \vec{\nabla} \cdot (\mu (\vec{\nabla} \vec{V} + \vec{\nabla} \vec{V}^T)) + 2\gamma \kappa \delta_s \vec{n}\end{aligned}\tag{1.9}$$

Ainsi, par rapport à l'équation (1.2), l'équation (1.9) fait apparaître un terme supplémentaire, liée aux forces interfaciales,  $2\gamma \kappa \delta_s \vec{n}$ .



### Écoulement turbulent

On adopte l'approche de la simulation des grandes échelles (*LES*) pour modéliser un écoulement turbulent. Celle-ci considère que l'intégralité de l'agitation turbulente cesse d'être aléatoire (se référer à [CHASSAING 00]). Ainsi, les structures supérieures à la taille d'une maille caractéristique sont calculées explicitement tandis que les structures de taille inférieure à cette maille caractéristique (structures de sous maille) sont modélisées.

Dans le cas d'un écoulement turbulent, la vitesse instantanée  $\vec{V}$  se décompose comme la somme de la vitesse moyenne de l'écoulement  $\overline{\vec{V}}$  et des fluctuations de vitesses  $\vec{v}$ . En introduisant le concept de viscosité turbulente  $\mu_t$ , les équations de Navier-Stokes peuvent alors se mettre sous la forme (voir [CHASSAING 00]) :

$$\begin{aligned}\overline{\nabla} \cdot \vec{V} &= 0 \\ \rho \left( \frac{\partial \vec{V}}{\partial t} + (\vec{V} \cdot \overline{\nabla}) \vec{V} \right) &= \rho \vec{g} - \overline{\nabla} p + \overline{\nabla} \cdot ((\mu + \mu_t)(\overline{\nabla} \vec{V} + \overline{\nabla} \vec{V}^T))\end{aligned}\quad (1.10)$$

La modélisation de la viscosité turbulente  $\mu_t$ , introduite par *Smagorinsky* en 1963 ([SMAGORINSKY 63]), consiste à écrire :

$$\mu_t = \rho C_s^2 \bar{\Delta}^2 (\bar{S}_{ij} \bar{S}_{ij})^{0.5}$$

avec  $C_s$  la constante du modèle de *Smagorinsky*,  $\bar{\Delta} = (\Delta x \Delta y \Delta z)^{\frac{1}{3}}$  le filtre et  $\bar{S}_{ij}$  le tenseur des déformations.

Le système à résoudre devient :

$$\begin{aligned}\overline{\nabla} \cdot \vec{V} &= 0 \\ \rho \left( \frac{\partial \vec{V}}{\partial t} + (\vec{V} \cdot \overline{\nabla}) \vec{V} \right) &= \rho \vec{g} - \overline{\nabla} p + \overline{\nabla} \cdot ((\mu + \mu_t)(\overline{\nabla} \vec{V} + \overline{\nabla} \vec{V}^T))\end{aligned}\quad (1.11)$$

$$\frac{\partial C}{\partial t} + \vec{V} \cdot \overline{\nabla} (C) = 0 \quad (1.12)$$

et prend la forme globale :

$$\begin{aligned}\int_{\Omega} \overline{\nabla} \cdot \vec{V} d\Omega &= 0 \\ \int_{\Omega} \left( \rho \left( \frac{\partial \vec{V}}{\partial t} + (\vec{V} \cdot \overline{\nabla}) \vec{V} \right) \right) d\Omega &= \int_{\Omega} \left( \rho \vec{g} - \overline{\nabla} p + \overline{\nabla} \cdot ((\mu + \mu_t)(\overline{\nabla} \vec{V} + \overline{\nabla} \vec{V}^T)) \right) d\Omega \\ \int_{\Omega} \left( \frac{\partial C}{\partial t} + \vec{V} \cdot \overline{\nabla} (C) \right) d\Omega &= 0\end{aligned}\quad (1.13)$$

### 1.1.2 Equation de conservation de l'énergie

Cette équation s'établit en utilisant le premier principe de la thermodynamique. Ce dernier stipule que la variation d'énergie interne  $E_i$  et cinétique  $E_c$  d'un système fermé entre deux états (en équilibre thermodynamique) successifs 1 et 2 est due aux échanges d'énergies mécanique (sous forme de travail  $\delta W_{1-2}$ ) et thermique ( $\delta Q_{1-2}$ ). Le premier principe s'écrit alors :

$$\frac{D}{Dt}(E_i + E_c) = \frac{D}{Dt}(\delta W_{1-2}) + \frac{D}{Dt}(\delta Q_{1-2}) \quad (1.14)$$

La puissance mécanique provient des forces volumiques  $\vec{F}$  et surfaciques (tenseur des contraintes  $\vec{\bar{\Sigma}} = -\vec{\nabla}p + \vec{\bar{\tau}}$ , avec  $\vec{\bar{\tau}} = (\mu(\vec{\nabla}\vec{V} + \vec{\nabla}\vec{V}^T))$ ).

Dans notre cas, la puissance thermique échangée avec l'extérieur se réduit aux transferts de chaleur de type conductif. Ceux-ci suivent la loi de Fourier  $\vec{q} = -\lambda\vec{\nabla}T$ , avec  $\vec{q}$  le vecteur densité de flux de chaleur.

Par ailleurs, dans un fluide incompressible, l'énergie interne s'écrit :

$$E_i = \int_{\Omega} \rho c_v T d\Omega$$

où  $c_v$  est la capacité calorifique massique à volume constant. Vu que pour un fluide incompressible, les capacités calorifiques massiques à pression constante ( $c_p$ ) et à volume constant ( $c_v$ ) sont équivalentes, on considère  $c = c_p \sim c_v$  la capacité calorifique massique du fluide incompressible.

En combinant l'équation (1.14) et celle de conservation de l'énergie cinétique (dédue de l'équation (1.2)), on obtient l'équation de conservation de l'énergie :

$$\int_{\Omega} (\rho c \frac{\partial T}{\partial t} + \vec{V} \cdot \vec{\nabla} T) d\Omega = \int_{\Omega} \vec{\nabla} \cdot (\lambda \vec{\nabla} T) d\Omega + \int_{\Omega} \vec{\nabla} \cdot (\vec{\bar{\tau}} \cdot \vec{\nabla} \vec{V}) d\Omega \quad (1.15)$$

Sa forme locale est :

$$\rho c \left( \frac{\partial T}{\partial t} + \vec{V} \cdot \vec{\nabla} T \right) = \vec{\nabla} \cdot (\lambda \vec{\nabla} T) + \vec{\nabla} \cdot (\vec{\bar{\tau}} \cdot \vec{\nabla} \vec{V}) \quad (1.16)$$

La prise en compte du changement de phase fusion/solidification se traduit par l'ajout d'un terme source  $S = -\rho L_{fs} \frac{\partial FRS}{\partial t}$  qui modélise l'énergie latente du changement de phase, où  $L_{fs}$  est la chaleur latente lors de la fusion (ou solidification) et  $FRS$  la fraction solide.

### 1.1.3 Equation de transport d'espèce

Cette équation traduit la conservation de la concentration massique de l'espèce considérée  $S_p$ , soit :

$$\frac{\partial S_p}{\partial t} + \vec{V} \cdot \vec{\nabla}(S_p) = \vec{\nabla} \cdot (D \vec{\nabla}(S_p)) \quad (1.17)$$

Sa forme globale est :

$$\int_{\Omega} \left( \frac{\partial S_p}{\partial t} + \vec{V} \cdot \vec{\nabla}(S_p) \right) d\Omega = \int_{\Omega} \vec{\nabla} \cdot (D \vec{\nabla}(S_p)) d\Omega \quad (1.18)$$

## 1.2 Méthodes numériques

### 1.2.1 Méthode des volumes finis

Les équations (1.16), (1.18) et (1.2) sont constituées d'un terme de variation temporelle ( $\frac{\partial \Phi}{\partial t}$ ), d'un terme d'advection ( $\vec{V} \cdot \vec{\nabla} \Phi$ ) où  $\vec{V}$  est la vitesse à laquelle est advectée la quantité  $\Phi$ , d'un terme de diffusion ( $\vec{\nabla} \cdot (K \vec{\nabla} \Phi) + \vec{\nabla} \Phi^T$ ) si  $\Phi$  est une des composantes de la vitesses,  $\vec{\nabla} \cdot (K \vec{\nabla} \Phi)$  sinon) et de termes sources  $S$ , où  $\Phi$  est l'inconnue représentant  $\vec{V}$ ,  $T$ ,  $C$  ou  $S_p$  et  $K$  un coefficient de diffusion. Un bilan de flux sur la variable scalaire  $\Phi$  donne l'équation :

$$\frac{\partial \Phi}{\partial t} + (\vec{V} \cdot \vec{\nabla}) \Phi = \vec{\nabla} \cdot (K \vec{\nabla} \Phi) + S \quad (1.19)$$

On peut écrire ces équations sous la forme globale :

$$\int_{\Omega} \left( \frac{\partial \Phi}{\partial t} + (\vec{V} \cdot \vec{\nabla}) \Phi \right) d\Omega = \int_{\Omega} \vec{\nabla} \cdot (K \vec{\nabla} \Phi) d\Omega + \int_{\Omega} S d\Omega \quad (1.20)$$

puis sous sa forme conservative :

$$\int_{\Omega} \left( \frac{\partial \Phi}{\partial t} + \vec{\nabla} \cdot (\Phi \vec{V}) \right) d\Omega = \int_{\Omega} \vec{\nabla} \cdot (K \vec{\nabla} \Phi) d\Omega + \int_{\Omega} S d\Omega \quad (1.21)$$

On précise que l'équation (1.21) n'est autre qu'un bilan de flux  $\Phi$  dans le domaine  $\Omega$ . Celle-ci est discrétisée à partir de la méthode des volumes finis comme indiqué dans [PATANKAR 80].

Elle se décompose en trois étapes :

1. On définit les volumes de contrôle sur lesquels on va faire des bilans de flux  $\Phi$ . Prenons un maillage cartésien à  $N$  nœuds. On définit pour chaque nœud  $i \in [1, N]$  un volume de contrôle  $V_i$  centré en  $i$  de telle façon que  $\bigcup_{i=1}^N V_i = \bar{\Omega}$  et  $V_i \cap_{i \neq j} V_j = \text{face}_{ij}$ .  $\text{face}_{ij}$  est la face *contigüe* aux deux volumes de contrôle  $V_i$  et  $V_j$ .

2. Un bilan de flux  $\Phi$  est fait dans chaque volume de contrôle  $V_i$ ,  $i \in [1, N]$

$$\int_{V_i} \left( \frac{\partial \Phi}{\partial t} + \vec{\nabla} \cdot (\Phi \vec{V}) \right) dV_i = \int_{V_i} \vec{\nabla} \cdot (K \vec{\nabla}(\Phi)) dV_i + \int_{V_i} S dV_i \quad (1.22)$$

On remarque que si on somme (1.22) sur  $i \in [1, N]$ , on retrouve l'équation (1.21).

3. L'équation (1.22) doit être discrétisée sachant que  $\Phi$  est connue aux nœuds de discrétisation  $i \in [1, N]$ . Les flux se calculant sur les faces  $face_{ij}$ ,  $(i, j) \in [1, N] \times [1, N]$ , il est nécessaire d'interpoler  $\Phi$ . La nature de l'interpolation dépend du terme de l'équation (1.22) (cf [PATANKAR 80])

Le raisonnement est identique quand la variable  $\Phi$  est vectorielle.

L'utilisation des volumes finis sur un maillage unique, collocalisé pour toutes les variables, donne de bons résultats lorsqu'on veut résoudre l'équation de l'énergie ou celle du transport d'espèce. Il en est autrement pour l'équation de Navier-Stokes. [PATANKAR 80] explique que l'utilisation des volumes finis dans ces conditions ne permet pas de traiter le terme du gradient de pression dans l'équation (1.2) correctement. En effet, la variation de pression prise en compte ne se fait pas entre deux nœuds successifs mais d'un nœud sur deux. En utilisant l'exemple 1D d'une pression en *zigzag*, [PATANKAR 80] montre que le terme  $\frac{dp}{dx}$  est considéré comme nul et donc que la pression est uniforme, ce qui est faux. [PATANKAR 80] propose alors comme solution l'utilisation d'un maillage décalé introduit par *Harlow* quand il a présenté la méthode *MAC* (Marker And Cells) [HARLOW 65]. Celui-ci est constitué d'une grille de pression (scalaire) et d'une grille par composante de vitesse (voir figure 1.1).

Si on considère un écoulement à deux fluides (*fluide*<sub>1</sub> de viscosité  $\mu_1$ , *fluide*<sub>2</sub> de viscosité  $\mu_2$ ), la détermination des termes de cisaillement dans le tenseur des viscosités nécessite la connaissance de la viscosité au centre des mailles de la grille de pression. La viscosité étant connue sur les nœuds de pression, il suffit de l'interpoler au centre des mailles. Cependant, la valeur obtenue est une combinaison linéaire de  $\mu_1$  et  $\mu_2$ , ce qui n'est pas très satisfaisant. Une grille de viscosité a donc été ajoutée aux autres grilles (voir figure 1.2). Les nœuds de la grille de viscosité se situent aux centres des mailles de la grille de pression. Ceci permet grâce à la fraction volumique de mettre en ces nœuds la valeur de la viscosité du fluide correspondant.

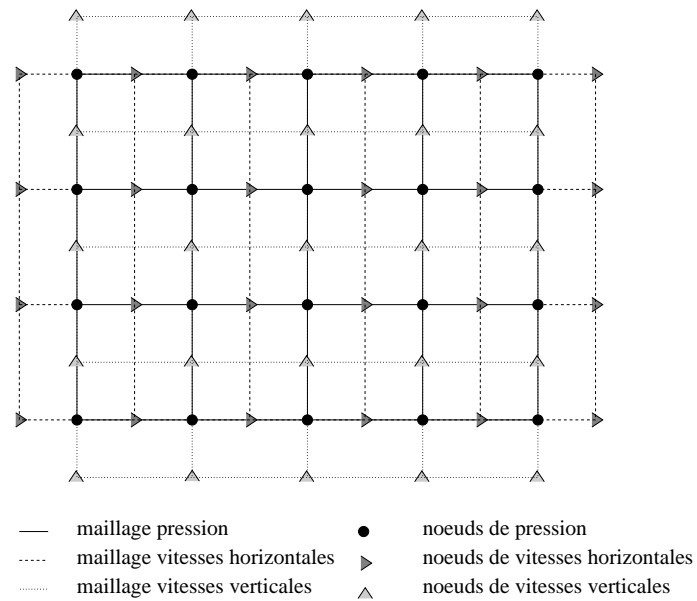


Figure 1.1 : Maillage décalé de type MAC.

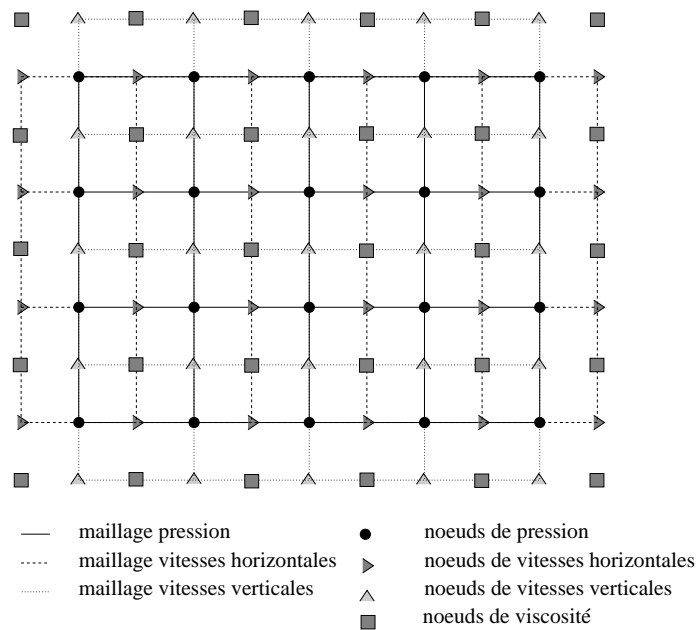


Figure 1.2 : Maillage décalé de type MAC auquel on a rajouté une grille de viscosités.

### 1.2.2 Discrétisation et résolution de l'équation de l'énergie et de transport d'espèces

On considère l'équation (1.19), où  $\Phi$  représente la température ou la concentration massique. Sa discrétisation implicite en temps s'écrit :

$$\frac{\Phi^{n+1} - \Phi^n}{\Delta t} + (\vec{V}^{n+1} \cdot \vec{\nabla}) \Phi^{n+1} + Lim_{\Phi}^{n+1} = \vec{\nabla} \cdot (K^n \vec{\nabla} \Phi^{n+1}) \quad (1.23)$$

où  $Lim_{\Phi}$  est le terme de pénalisation qui permet de prendre en compte les limites (voir la sous-section 1.2.5). En ce qui concerne le terme d'advection des équations scalaires (1.16) et (1.18), l'utilisation de schémas implicites d'ordre élevé tel *Lax – Wendroff* ou centré donne des solutions  $\Phi$  précises lorsque celle-ci est régulière. Par contre, quand la solution  $\Phi$  devient irrégulière, la discrétisation du terme d'advection avec des schémas implicites d'ordre élevé est à proscrire car ils provoquent la formation d'oscillations qui se développent. La solution finit alors par diverger [HIRSCH 90] (cf figure 1.3 droite).

Les schémas implicites d'ordre peu élevé, comme le schéma *UPWIND*, par exemple,

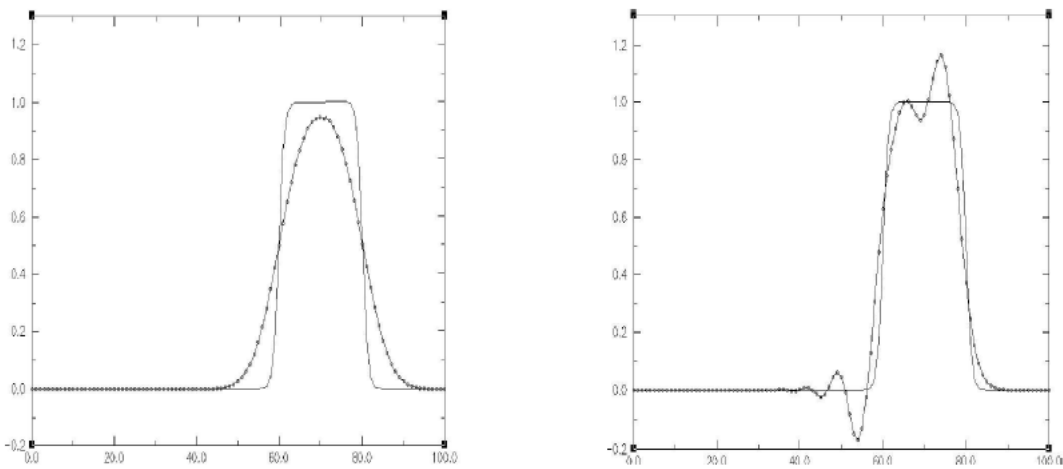


Figure 1.3 : Propagation d'une fonction  $C(x) = (0.5 + 0.5tanh(80 * (x - 0.1)))$  .  
 Comparaison entre le schéma *Upwind* à gauche et *Lax-Wendroff* à droite.

permettent d'éviter les oscillations, mais ils sont très diffusifs (cf figure 1.3 gauche). Donc la qualité de la solution est fortement dégradée [HIRSCH 90]. Une idée est donc d'implémenter un schéma *WENO* (Weighted Essentially Non-Oscillatory) [LIU 94] ou *TVD* (Total Variation Diminishing) [LEVEQUE 92], qui permet d'obtenir un ordre de schéma satisfaisant tout en évitant les oscillations en présence de champ discontinu. Ce sont des schémas explicites car la version implicite de ces schémas demande une *CFL* (*Courant – Friedrich – Levy*) aussi restrictive que la version explicite. Cette *CFL* assure la stabilité de ces schémas et consiste à imposer que l'inconnue  $\Phi$  parcourt pendant le temps  $\Delta t$  une distance inférieure ou égale à la longueur d'une demi-maille.

Le terme de diffusion des équations scalaires (1.16) et (1.18) est traité avec un schéma implicite centré tant que le coefficient de diffusion moléculaire (respectivement le coefficient de conduction) de l'équation (1.18) (respectivement (1.16)) est supérieur à une valeur cri-

tique. Plus précisément, si on écrit l'équation (1.16) ou (1.18), sous la forme d'un système linéaire à résoudre, on obtient  $A^{n+1}\Phi^{n+1} = B^n$  où  $B^n$  est le second membre à l'itération  $n$ ,  $\Phi^{n+1}$  l'inconnue (température ou concentration massique) à l'itération  $n + 1$  et  $A^{n+1}$  la matrice de discrétisation à l'itération  $n + 1$ . Les termes diagonaux de  $A^{n+1}$  sont de l'ordre de  $\frac{1}{\Delta t}$  tandis que ceux extra-diagonaux sont de l'ordre de  $\frac{K}{\Delta x^2}$ . Si  $K \ll \frac{\Delta x^2}{\Delta t}$ , alors le solveur implicite devient inefficace. Donc si  $K \ll \frac{\Delta x^2}{\Delta t}$ , on utilisera plutôt un schéma explicite centré pour résoudre le terme de diffusion des équations (1.16) et (1.18). Dans le cas d'une résolution implicite de ce terme de diffusion, le solveur utilisé est l'algorithme itératif du *Bi - CGStabII* (Bi-Conjugate Gradient Stabilized) avec un préconditionnement de *JACOBI* (cf [GUSTAFSSON 78] et [VORST 92]).

### 1.2.3 Discrétisation et résolution de l'équation de Navier-Stokes

Dans cette partie, on va décrire brièvement la discrétisation des équations de Navier-Stokes exprimées en vitesse-pression. On rappelle qu'il s'agit de calculer à chaque instant le couple (vitesse, pression) vérifiant à la fois la conservation de la quantité de mouvement et la contrainte d'incompressibilité.

Avant de présenter l'algorithme de calcul de la pression et de la vitesse, nous allons présenter la discrétisation spatio-temporelle.

#### Discrétisation temporelle

L'intégration temporelle est réalisée de manière classique via un schéma d'ordre 1. Un schéma implicite centré est utilisé pour le terme d'inertie [PATANKAR 80]. Concernant les termes visqueux, un schéma implicite centré d'ordre 2 est utilisé. Le choix de ces schémas implicites est valable pour un champ de vitesse continu sous la contrainte d'incompressibilité. Il n'y a donc pas de *CFL* à satisfaire.

Une fois discrétisé en temps et à l'ordre 1, le problème continu devient : connaissant

$\vec{V}^n, \vec{V}^{n-1}$  et  $p^n$ , chercher  $\vec{V}^{n+1}$  et  $p^{n+1}$  solutions de

$$\begin{aligned} \vec{\nabla} \cdot \vec{V}^{n+1} &= 0 \\ \rho^n \left( \frac{\vec{V}^{n+1}}{\Delta t} + (\vec{V}^n \cdot \vec{\nabla}) \vec{V}^{n+1} \right) + \overrightarrow{Lim}_{\vec{V}}^{n+1} &= \rho^n \vec{g} \\ -\vec{\nabla} p^{n+1} + \vec{\nabla} \cdot (\mu^n (\vec{\nabla} \vec{V}^{n+1} + \vec{\nabla}^T \vec{V}^{n+1})) + \rho^n \left( \frac{\vec{V}^n}{\Delta t} \right) \end{aligned} \quad (1.24)$$

où  $\overrightarrow{Lim}_{\vec{V}}$  est un tenseur d'ordre un lié à la méthode de pénalisation pour traiter les conditions aux limites (voir sous section 1.2.5).

On note que la pression et la vitesse sont couplées, ce qui rend le coût de résolution du système algébrique final prohibitif, notamment quand il s'agira de simuler des écoulements instationnaires et complexes. Pour contourner cette difficulté, plusieurs algorithmes sont proposés dans la littérature (cf [PEYRET 83]) afin de réaliser un découplage, le plus consistant et précis possible. On se contente par la suite de donner un résumé de l'algorithme utilisé dans le cadre de ce travail.

### Méthode du lagrangien augmenté

La méthode dite de lagrangien augmenté fait partie de la famille des algorithmes mélangeant à la fois la minimisation sous contrainte et l'accélération de convergence via un terme du type pénalisation.

La prise en compte de la contrainte est assurée par le biais d'un multiplicateur de Lagrange (la pression). Le découplage de la vitesse et de la pression est implicite en vitesse et explicite en pression ([FORTIN 82]).

Notons :

$$\begin{aligned} L^2(\Omega) &= \{f \text{ tel que } \int_{\Omega} |f|^2 d\Omega < \infty\} \\ H_0^1(\Omega) &= \{f \in L^2(\Omega), \frac{\partial f}{\partial x_i} \in L^2(\Omega), \quad i \in [1, 3], \quad f|_{\partial\Omega} = 0\} \end{aligned}$$

Plus précisément, on définit une fonctionnelle  $J(\vec{V})$  pour tout  $\vec{V} \in H_0^1(\Omega)$ , issue de la formulation faible de l'équation de conservation de la quantité de mouvement. Il s'agit de minimiser cette fonctionnelle sous la contrainte  $M = \{\vec{V} \in H_0^1(\Omega) \text{ tel que } \vec{\nabla} \cdot \vec{V} = 0\}$ . Cela correspond à trouver  $\vec{U} \in M$  tel que :

$$J(\vec{U}) = \min_{\vec{V} \in M} (J(\vec{V})) \quad (1.25)$$



On transforme ce problème de minimisation sous contrainte en un problème de minimisation sans contrainte, en introduisant la pression  $q$ , comme un multiplicateur de Lagrange. On définit le Lagrangien comme :

$$\begin{aligned} L : M \times L^2 &\rightarrow \mathfrak{R} \\ (\vec{V}, q) &\mapsto L(\vec{V}, q) = J(\vec{V}) - \int_{\Omega} q \vec{\nabla} \cdot \vec{V} d\Omega \end{aligned} \quad (1.26)$$

Le problème de minimisation (1.25) consiste à trouver le point selle  $(\vec{U}, p) \in M \times L^2$  du lagrangien (1.26), c'est-à-dire trouver le couple  $(\vec{V}, q) \in M \times L^2$  vérifiant :

$$\begin{aligned} L(\vec{U}, p) &= \min_{\vec{V} \in M} (\max_{q \in L^2} (L(\vec{V}, q))) \\ &= \max_{q \in L^2} (\min_{\vec{V} \in M} (L(\vec{V}, q))) \end{aligned} \quad (1.27)$$

Afin d'améliorer les propriétés de convergence (cf [FORTIN 82]), on introduit un terme de pénalisation faisant apparaître la contrainte, soit :  $\frac{dr}{2} |\vec{\nabla} \cdot \vec{V}|^2$ ,  $\vec{V} \in H_0^1(\Omega)$ ,  $dr \in \mathfrak{R}$ .

Notons le lagrangien augmenté :

$$\begin{aligned} L_r : M \times L^2 &\rightarrow \mathfrak{R} \\ (\vec{V}, q) &\mapsto L(\vec{V}, q) = J(\vec{V}) - \int_{\Omega} q \cdot \vec{\nabla} \cdot \vec{V} d\Omega + \int_{\Omega} \frac{dr}{2} |\vec{\nabla} \cdot \vec{V}|^2 d\Omega \end{aligned} \quad (1.28)$$

Le problème de minimisation (1.26) consiste alors à trouver le point selle  $(\vec{U}, p) \in M \times L^2$  du lagrangien (1.28), c'est-à-dire trouver le couple  $(\vec{V}, q) \in M \times L^2$  vérifiant :

$$\begin{aligned} L_r(\vec{U}, p) &= \min_{\vec{V} \in M} (\max_{q \in L^2} (L_r(\vec{V}, q))) \\ &= \max_{q \in L^2} (\min_{\vec{V} \in M} (L_r(\vec{V}, q))) \end{aligned} \quad (1.29)$$

En revenant à la formulation forte du problème, on admet que le point selle est aussi solution du problème (ceci a été montré pour les équations de Stokes et est admis pour les équations de Navier-Stokes) :

$$\rho \left( \frac{\partial \vec{V}}{\partial t} + (\vec{V} \cdot \vec{\nabla}) \vec{V} \right) = \rho \vec{g} - \vec{\nabla} p + \vec{\nabla} \cdot (\mu (\vec{\nabla} \vec{V} + \vec{\nabla} \vec{V}^T)) - dr \vec{\nabla} (\vec{\nabla} \cdot \vec{V}) \quad (1.30)$$

$$\vec{\nabla} \cdot \vec{V} = 0 \quad (1.31)$$

La discrétisation implicite du système (1.30)-(1.31) s'écrit :

$$\begin{aligned} \rho^n \left( \frac{\vec{V}^{n+1}}{\Delta t} + (\vec{V}^n \cdot \vec{\nabla}) \vec{V}^{n+1} \right) + \overline{Lim}_{\vec{V}}^{n+1} &= \rho^n \vec{g} - \vec{\nabla} p^{n+1} \\ + \vec{\nabla} \cdot (\mu^n (\vec{\nabla} \vec{V}^{n+1} + \vec{\nabla}^T \vec{V}^{n+1})) & \\ + \rho^n \left( \frac{\vec{V}^n}{\Delta t} \right) + dr \vec{\nabla} (\vec{\nabla} \cdot \vec{V}^{n+1}) & \end{aligned} \quad (1.32)$$

$$\vec{\nabla} \cdot \vec{V}^{n+1} = 0 \quad (1.33)$$

On remarque que la seule différence entre le système (1.25) et (1.32)-(1.33) est l'ajout du terme de pénalisation. En observant le système (1.32)-(1.33), on voit qu'il n'y a pas d'équation explicite permettant le calcul de la pression. Toute la difficulté réside donc dans la détermination de la pression à partir du système (1.32)-(1.33). Notons le système (1.32)-(1.33) sous la forme :

$$A \vec{V}^{n+1} + G p^{n+1} = F \quad (1.34)$$

$$D \vec{V}^{n+1} = 0 \quad (1.35)$$

A étant inversible, l'équation (1.34) donne :

$$\vec{V}^{n+1} + A^{-1} G p^{n+1} = A^{-1} F \quad (1.36)$$

En multipliant à gauche les termes de l'équation (1.34) par  $D$ , on obtient :

$$D \vec{V}^{n+1} + D A^{-1} G p^{n+1} = D A^{-1} F \quad (1.37)$$

or  $D \vec{V}^{n+1} = 0$ , donc on peut calculer la pression en résolvant le système (on reconnaîtra l'opérateur d'*Uzawa* :  $DA^{-1}G$ ):

$$D A^{-1} G p^{n+1} = D A^{-1} F \quad (1.38)$$

Le système algébrique issu de (1.38) peut être résolu par différentes méthodes itératives. Soient  $p_0^{n+1} = p^n$  et  $m$  l'itération en cours de l'algorithme itératif. On choisit la méthode itérative de Richardson qui consiste à déterminer  $p^{n+1}$  de la façon suivante :

$$p_{m+1}^{n+1} = p_m^{n+1} - dp (D A^{-1} F - D A^{-1} G p_m^{n+1}) \quad (1.39)$$

or  $DA^{-1}F - DA^{-1}Gp_m^{n+1} = D\vec{V}_m^{n+1}$ , donc :

$$p_{m+1}^{n+1} = p_m^{n+1} - dpD\vec{V}_{m+1}^{n+1} \quad (1.40)$$

Précisons que la matrice  $A$  étant non symétrique et mal conditionnée, la résolution du système algébrique (1.34) est réalisée grâce à l'algorithme *BiCGStabII* introduit par [VORST 92] préconditionné par une factorisation *MILU* ([GUSTAFSSON 78]).

La résolution du système (1.34)-(1.35) est insérée dans une boucle en itérations de lagrangien augmenté  $k$ , qui permet de faire diminuer la valeur moyenne de la divergence  $div_{moyen}$ . Ces itérations cessent dès que  $div_{moyen}$  a atteint une valeur suffisamment faible. Notons  $k_{max}$ , le nombre d'itérations en lagrangien augmenté qu'il faut effectuer pour atteindre cette valeur, l'algorithme du lagrangien augmenté s'écrit :

1. Initialisation :  $\vec{V}^{k=0} = \vec{V}^n$  et  $p^{k=0} = p^n$

2. Itérations : pour  $k$  allant de 0 à  $k_{max} - 1$

$$\begin{aligned} \rho^n \left( \frac{\vec{V}^{k+1}}{\Delta t} + (\vec{V}^k \cdot \vec{\nabla}) \vec{V}^{k+1} \right) + \overline{Lim}_{\vec{V}}^{k+1} &= \rho^n \vec{g} - \vec{\nabla} p^k \\ + \vec{\nabla} \cdot (\mu^n (\vec{\nabla} \vec{V}^{k+1} + \vec{\nabla}^T \vec{V}^{k+1})) & \\ + \rho^n \left( \frac{\vec{V}^k}{\Delta t} \right) + dr \vec{\nabla} (\vec{\nabla} \cdot \vec{V}^{k+1}) & \end{aligned} \quad (1.41)$$

$$p^{k+1} = p^k - dp \vec{\nabla} \cdot \vec{V}^{k+1} \quad (1.42)$$

3. Couple solution :  $\vec{V}^{n+1} = \vec{V}^{k=k_{max}}$  et  $p^{k=k_{max}} = p^{n+1}$

On remarquera que la résolution explicite de la pression, obtenue avec l'équation (1.40), nous évite d'imposer des conditions aux limites pour celle-ci. L'évaluation des paramètres  $dp$  et  $dr$  est délicate car les équations utilisées ne sont pas adimensionnées. Des travaux récents ont été effectués par [VINCENT 04] pour les déterminer de façon automatique.

### 1.2.4 Discrétisation et résolution du transport d'interface

Pour un écoulement diphasique, le modèle  $1 - fluide$  est utilisé pour la résolution des équations de Navier-Stokes. Le transport de l'interface est réalisé par la méthode *VOF* (Volume Of Fluids) de [HIRT 81].

Cette méthode consiste à repérer chaque fluide  $i \in [1, 2]$  par une fonction de phase, appelée encore fonction couleur  $C_i$ , propre à chaque fluide  $i$ ,  $i \in [1, 2]$ , définie par,

$$\begin{aligned} C_i(M, t) &= 1, \forall M \in \text{fluide}_i \\ C_i(M, t) &= 0, \forall M \text{ sinon.} \end{aligned} \quad (1.43)$$

Lorsque l'interface coupe une maille, les deux fluides sont présent dans la maille. La fonction de phase prend dans ce cas une valeur strictement comprise entre 0 et 1. Elle permet ainsi de représenter la fraction de  $\text{fluide}_i$  dans chaque maille. Le transport de l'interface est régi par l'équation d'advection :

$$\frac{\partial C_i}{\partial t} + \vec{V} \cdot \vec{\nabla}(C_i) = 0 \quad (1.44)$$

La masse volumique et la viscosité sont directement recalculées à partir de la valeur de la fonction couleur.

Dans le contexte *VOF*, on peut résoudre l'équation (1.44) de plusieurs façons :

- la première méthode consiste à la discrétiser en utilisant un schéma *TVD* ou *WENO* de préférence (se référer à la sous section 1.2.1 et à [HIRSCH 90], [LEVEQUE 90], et [VINCENT 99]). Celle-ci introduit de la diffusion numérique sur deux à trois mailles autour de l'interface. On ne connaît donc la position de l'interface qu'approximativement.
- la deuxième méthode, appelée *VOF – CIAM*, développée par [YOUNGS 82] et utilisé par [ZALESKI ] par exemple, consiste à utiliser l'équation (1.44) pour reconstruire géométriquement l'interface. Ceci permet de connaître l'interface de manière très précise. Cependant, cette méthode a tendance à fractionner l'interface lorsque celle-ci est trop fine ou fortement cisailée.

La prise en compte des tensions superficielles se fait par l'ajout de la force de tension superficielle  $F_{TS}$  comme terme source de l'équation (1.2) (cf [BRACKBILL 92]) :

$$F_{TS} = -\gamma \nabla C_i \nabla \cdot \left( \frac{\nabla C_i}{\|\nabla C_i\|} \right) \quad (1.45)$$

avec  $\gamma$  le coefficient de tension superficielle.

### 1.2.5 Traitement des limites

L'imposition des conditions aux limites se fait via une méthode de pénalisation qui a été développée au laboratoire *TREFLE* par [ANGOT 89] et [KHADRA 94]. Celle-ci s'applique aux différents types de conditions aux limites : Dirichlet, Neumann. On considère  $\Phi$  l'inconnue scalaire représentant la température ou la concentration massique et  $\vec{V} = (u, w)$  l'inconnue vitesse. Posons  $\Phi_\infty(M, t)$  (respectivement  $\vec{V}_\infty(M, t)$ ) la valeur de  $\Phi$  (respectivement  $\vec{V}$ ) en un point  $M$  quelconque à l'extérieur du domaine  $\Omega$  de frontière  $\partial\Omega$ . La méthode consiste à exprimer les conditions aux limites sous la forme d'un flux surfacique :

$$-\left(\frac{\partial\Phi}{\partial n}\right)_{\partial\Omega} = Bi_\Phi(\Phi_{\partial\Omega} - \Phi_\infty) \quad (1.46)$$

$$-\left(\frac{\partial\vec{V}}{\partial n}\right)_{\partial\Omega} = Bi_{\vec{V}}(\vec{V}_{\partial\Omega} - \vec{V}_\infty) \quad (1.47)$$

$Bi_\Phi$  est un réel et  $Bi_{\vec{V}}$  est un tenseur d'ordre deux diagonal ( $Bi_u, Bi_w$ ) qui sont soit nul, soit infini.

- Si  $Bi_\Phi = 0$  alors  $(\frac{\partial\Phi}{\partial n})_{\partial\Omega} = 0$ . C'est donc une condition de Neumann qui est imposée.
- Si  $Bi_\Phi \rightarrow \infty$  alors  $\Phi_{\partial\Omega} = \Phi_\infty$ . C'est donc une condition de Dirichlet qui est imposée.
- Si  $Bi_{\vec{V}} = 0$  alors  $(\frac{\partial\vec{V}}{\partial n})_{\partial\Omega} = 0$ . C'est donc une condition de Neumann qui est imposée.
- Si  $Bi_{\vec{V}} \rightarrow \infty$  alors  $\vec{V}_{\partial\Omega} = \vec{V}_\infty$ . C'est donc une condition de Dirichlet qui est imposée.

Numériquement, cette approche se traduit par l'introduction du terme  $Lim_\Phi = Bi_\Phi(f_\Phi - \Phi_\infty)$ , où  $f_\Phi$  est une fonction discrète de  $\Phi$ , dans l'équation

$$\frac{\partial\Phi}{\partial t} + (\vec{V} \cdot \vec{\nabla})\Phi = \vec{\nabla} \cdot (K \vec{\nabla} \Phi) \quad (1.48)$$

ce qui donne l'équation suivante à résoudre pour les scalaires :

$$\frac{\partial\Phi}{\partial t} + (\vec{V} \cdot \vec{\nabla})\Phi + Lim_\Phi = \vec{\nabla} \cdot (K \vec{\nabla} \Phi) \quad (1.49)$$

$Bi_\Phi$  est un réel indiquant l'appartenance du point  $M$  à l'intérieur du domaine d'étude ou aux limites et le type de condition à imposer. Celui-ci vérifie :

- .  $Bi_\Phi = 0$  si  $M$  appartient à l'intérieur du domaine  $\Omega$ ,

- .  $Bi_\Phi = 10^{40}$  si  $M \in \partial\Omega$  pour une condition de Dirichlet.
- .  $Bi_\Phi = 10^{40}$  si  $M \in \partial\Omega$  pour une condition de Neumann. Plus précisément, le voisin du nœud  $M$  de situant à l'extérieur du domaine  $\Omega$ , est remplacé par son symétrique par rapport à la limite.

Quant à la fonction  $f_\Phi$ , elle permet d'appliquer la condition aux limites souhaitée.

Par exemple, pour une condition de Dirichlet, on veut imposer  $\Phi_{\partial\Omega} = \Phi_\infty$ . On a donc  $\forall M \in \partial\Omega, f_\Phi(M) = \Phi(M)$ . Ainsi  $Lim_\Phi(M) = 10^{40}(\Phi(M) - \Phi_\infty)$ .

De même que pour les scalaires, le traitement des limites par la méthode de pénalisation pour les vitesses se traduit numériquement par l'insertion du vecteur  $\overrightarrow{Lim_{\vec{V}}} = (Bi_u(f_u - u_\infty), Bi_w(f_w - w_\infty))$ , où  $(f_u, f_w)$  est une fonction discrète de  $(u, w)$ , dans l'équation de conservation de la quantité de mouvement, soit :

$$\rho\left(\frac{\partial \vec{V}}{\partial t} + (\vec{V} \cdot \vec{\nabla}) \vec{V}\right) + \overrightarrow{Lim_{\vec{V}}} = \rho \vec{g} - \vec{\nabla} p + \vec{\nabla} \cdot (\mu + (\vec{\nabla} \vec{V} + \vec{\nabla} \vec{V}^T)) \quad (1.50)$$

$Bi_u$  (respectivement  $Bi_w$ ) est un réel indiquant l'appartenance du point  $M_u$  (respectivement  $M_w$ ) à l'intérieur du domaine d'étude ou sur ses limites et le type de condition imposée. Chacun vérifie :

- .  $Bi_u = 0$  si  $M_u$  appartient à l'intérieur du domaine  $\Omega$ .
- .  $Bi_u = 10^{40}$  si  $M_u \in \partial\Omega$  pour une condition de Dirichlet.
- .  $Bi_w = 10^{40}$  si  $M_w \in \partial\Omega$  pour une condition de Dirichlet.
- . Pour une condition de Neumann où  $M_j \in \partial\Omega, j \in \{u, w\}$ , on impose  $Bi_j = 10^{40}$  à la composante de vitesse normale à la limite. On impose  $Bi_j = 0$  à la composante de vitesse tangentielle à la limite et on applique la même relation de symétrie que pour le cas scalaire.

On rappelle qu'on travaille sur un maillage décalé de type *MAC*. La grille des nœuds de pression coïncide avec le domaine  $\Omega$ . Les autres grilles étant décalées, les vitesses appartenant aux limites de leur grille et étant normales à  $\partial\Omega$  sont à l'extérieur du domaine  $\Omega$ . Quant aux vitesses appartenant aux limites de leur grille et étant tangentielles à  $\partial\Omega$  elles appartiennent à cette limite. La fonction  $(f_u, f_w)$  permet d'appliquer la condition aux limites souhaitée sur  $\partial\Omega$ .

On détaille le cas d'une condition aux limites de Dirichlet. Soit  $(u_\infty, w_\infty)$  fixé, on impose  $(u(M_u), w(M_w)) = (u_\infty, w_\infty)$ . On se référera aux figures 1.4 et 1.5 pour les notations.

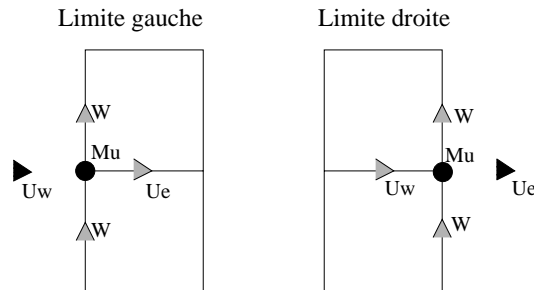


Figure 1.4 : *Traitement de la condition aux limites de Dirichlet pour la vitesse u, pour les limites gauche et droite.*

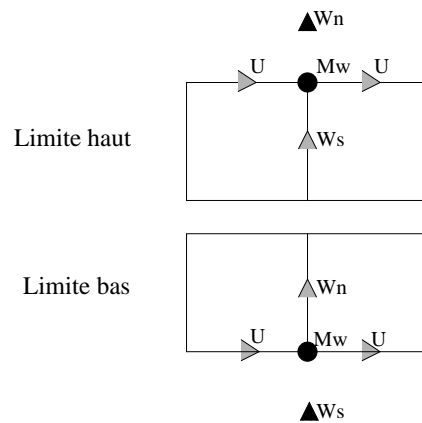


Figure 1.5 : *Traitement de la condition aux limites de Dirichlet pour la vitesse w, pour les limites bas et haut.*

– Si on est sur une limite gauche ou droite, alors :

$$\begin{aligned} f_u(M_u) &= \frac{1}{2}(u_W + u_E) \\ f_w(M_w) &= w(M_w) \\ Bi_u &= 10^{40} \\ Bi_w &= 10^{40} \end{aligned}$$

- Si on est sur une limite bas ou haut, alors :

$$\begin{aligned} f_u(M_u) &= u(M_u) \\ f_w(M_w) &= \frac{1}{2}(w_N + w_S) \\ Bi_u &= 10^{40} \\ Bi_w &= 10^{40} \end{aligned}$$

### 1.3 Mise en place de l'AMR

On rappelle qu'on s'intéresse à des écoulements diphasique et monophasique avec transport d'espèces inertes faiblement diffusives. La précision des solutions est essentielle pour analyser ce type d'écoulement. Il est donc indispensable d'utiliser des maillages très fins.

Dans le cas d'un écoulement diphasique, le transport de l'interface peut se faire grâce à une méthode *VOF – CIAM*. On rappelle que celle-ci permet de connaître la position de l'interface de façon très précise. Cependant, lorsque l'interface est fortement cisailée, cette méthode a tendance à la fractionner. On peut éviter cette difficulté en utilisant un maillage très fin.

Dans le cas d'un écoulement monophasique avec transport d'espèce faiblement diffusives, la méthode *VOF – CIAM* est à proscrire puisqu'elle ne peut pas traiter le terme diffusif de l'équation de transport. On est donc amené à résoudre l'équation (1.18) avec un schéma *TVD*. Vu que ce schéma diffuse sur deux à trois mailles, il faut travailler sur un maillage assez fin de façon à ce que la zone de diffusion physique soit supérieure strictement à ces deux mailles.

Dans les deux cas, se pose le problème de la faisabilité compte tenu des moyens informatiques qui sont à notre disposition. Leurs capacités en taille mémoire et notre durée d'attente de résultats acceptable étant de dimension finie, il est indispensable de les optimiser. On peut citer quelques méthodes possibles telles la parallélisation des codes de calculs, l'utilisation de méthodes de raffinement de maillage adaptatif (*AMR*), ou un couplage de ces deux méthodes.

Le parallélisme rend possible la résolution d'équations sur des maillages très fins car la taille mémoire nécessaire est distribuée sur plusieurs processeurs. Ainsi le temps de calcul (temps *CPU*) devient très intéressant par rapport à un calcul en monoprocesseur.

Si on traite des cas où l'utilisation d'un maillage très fin n'est réellement utile que dans



certaines zones du domaine, il est dommage de le raffiner en entier. C'est pour ce type de problème que l'utilisation d'une méthode *AMR* devient décisive. En effet, on ne raffine que les zones qui nécessitent un traitement plus fin, et ceci de manière adaptative dans l'espace et le temps. Ainsi, des cas qui dépassent les capacités d'une machine monoprocesseur standard à cause d'un nombre de nœuds à résoudre trop important, peuvent être réalisés en utilisant une méthode *AMR*. Pour certains cas où cette approche ne suffit plus, l'utilisation du parallélisme en plus de celle-ci serait une solution appropriée.

On prend l'exemple d'un écoulement turbulent dans le quel se trouvent un gros et un petit tourbillon. La méthode classique consiste à prendre un maillage assez fin pour pouvoir décrire précisément les deux tourbillons, ce qui est très coûteux en taille mémoire et en temps *CPU* (cf figure 1.6 gauche). Par ailleurs le gros tourbillon aurait pu être décrit de manière précise avec beaucoup moins de points, ce qui aurait permis d'utiliser moins de mémoire. En effet, le maillage de la figure 1.6 droite est suffisant pour décrire la grosse structure avec la précision souhaitée, mais il ne peut appréhender la petite de façon précise. L'idée de la méthode *AMR* consiste à utiliser un maillage qui permettent de décrire au mieux les deux structures tout en utilisant un maillage adapté aux différentes zones du domaine et en temps pour suivre le petit tourbillon (cf figure 1.7).

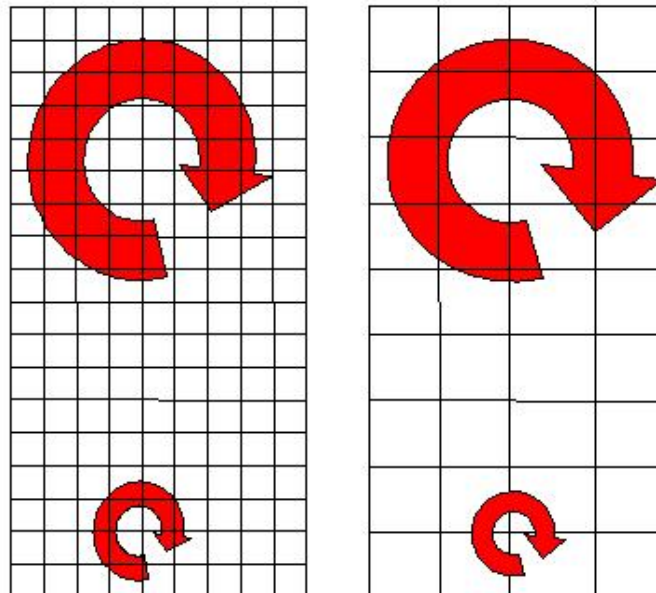


Figure 1.6 : *Résolution sur un maillage classique.*

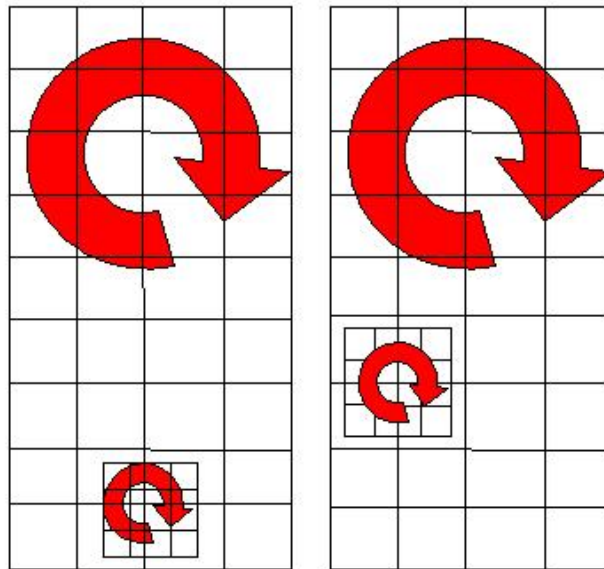


Figure 1.7 : *Résolution sur maillage adaptatif en espace et en temps.*

### 1.3.1 Stratégies de maillage existantes

Plusieurs méthodes de raffinement de maillage ont été développées pour capter des phénomènes locaux qui apparaissent de façon quelconque dans le temps. Cependant, chaque méthode est efficace pour traiter un problème spécifique. Il convient donc de choisir la méthode la mieux adaptée au problème qui nous intéresse.

- Méthode  $p$  : Le maillage reste fixe ainsi que le nombre de nœuds du maillage. La précision de la solution est obtenue en augmentant l'ordre du schéma dans les zones de plus forte erreur. Celle-ci a été développée par [BASU 83] pour des solutions continues dans des écoulements 1-fluide. Son utilisation reste difficile pour des solutions discontinues à cause de l'oscillation des schémas d'ordre élevé (cf [BAKER 97]). Cette méthode reste très utilisée dans le cadre des éléments finis.
- Méthode  $r$  : L'ordre du schéma et le nombre de nœuds du maillage restent fixes. Il suffit de déplacer les nœuds du maillage dans les zones qui requièrent d'être raffinées pour obtenir la précision souhaitée. Cette méthode est performante dans

le cas d'écoulements non visqueux compressibles sur des maillages non structurés (cf [PALMERIO 88] et [PALMERIO 94]). Cette méthode rendant le maillage fortement non-orthogonal, les termes visqueux ne sont pas traités correctement (voir [PIRZADEH 99]). La figure 1.8 montre la déformation dynamique du maillage pour suivre l'évolution des champs de température et de vitesse pour le problème de *Stefan* ([TENCHEV 05]).

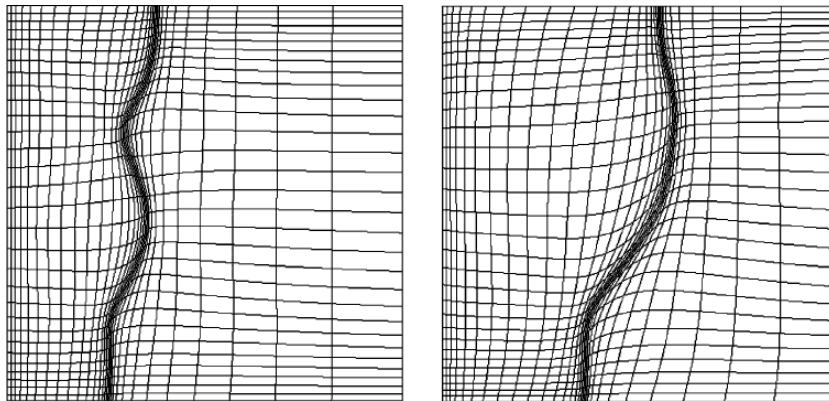


Figure 1.8 : *Raffinement de maillage par la méthode r*

- Méthode h : C'est actuellement la méthode la plus répandue dans la communauté scientifique. Elle consiste à fixer l'ordre du schéma tandis que le maillage et le nombre de nœuds varient au cours du temps. Des nœuds sont ajoutés dans les zones de plus forte erreur pour améliorer la précision de la solution. De la même façon, des nœuds sont enlevés des régions où le maillage est considéré trop fin. La figure 1.9 gauche représente un maillage initial ainsi que la zone à raffiner. La technique consiste à subdiviser chaque maille appartenant à la zone de raffinement (maille mère) en quatre mailles (mailles filles) (voir figure 1.9 droite). Ainsi de nouveaux nœuds apparaissent au centre des arêtes des mailles mères. cela correspond à de l'enrichissement de maillage. Comme le montre cette figure, cette méthode transforme le maillage initialement structuré en un maillage non structuré. Elle a été mise au point par [BERGER 89] pour des écoulements incompressibles. Elle a ensuite été utilisée, par exemple, par [HAY 04] pour des écoulements compressibles turbulents puis pour des écoulements diphasique (cf [HAY 05]).

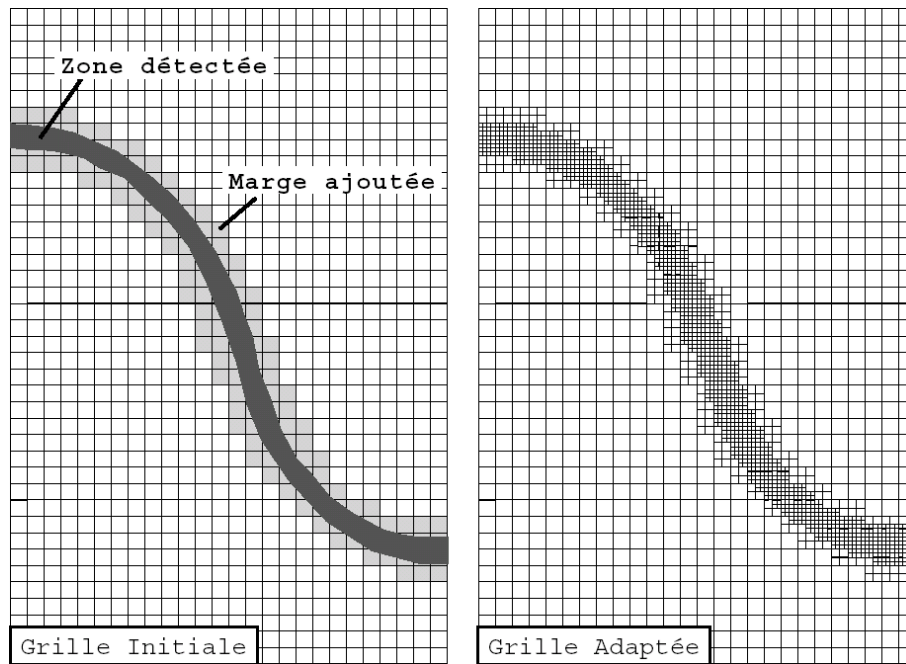


Figure 1.9 : *Raffinement de maillage par la méthode  $h$*  ([HAY 04])

- Méthode *OCLM* : Elle a été développée par [VINCENT 99] qui s’est inspiré des travaux de [BERGER 89]. L’objectif était de résoudre des écoulements incompressibles diphasiques avec un maillage structuré. C’est dans cette optique que la méthode *OCLM* a été conçue. Elle peut être considérée comme la combinaison de la méthode  $h$  et d’une méthode multigrille classique (voir [LASCAUX 00] et [TROTTEBERG 01]). L’idée consiste à raffiner le volume de contrôle d’un nœud  $P$  appartenant au maillage grossier et non pas les mailles grossières dont un des sommets est  $P$  (voir la figure 1.10). Le raffinement du maillage grossier génère une grille fine déconnectée de la grille grossière. Les connexions entre ces deux grilles se font uniquement par l’intermédiaire d’opérateur de prolongement et de restriction, comme on peut le rencontrer dans les méthodes multigrilles classiques. La figure 1.11 montre à la fois la conservation d’un maillage structuré par niveau et sa structure par niveau.

Des couplage entre les différentes méthodes *AMR* ont été mises en place pour bénéficier des avantages de chacune. Par exemple, [GUO 86] a présenté une  $h-p$  méthode. Parmi les personnes qui utilisent cette méthode, on peut citer [ODEN 89], [DEMKOWICZ 89] ou [RACHOWICZ 89]. Un couplage entre la méthode  $h$  et  $r$  a aussi été testée par [HABASHI 98],

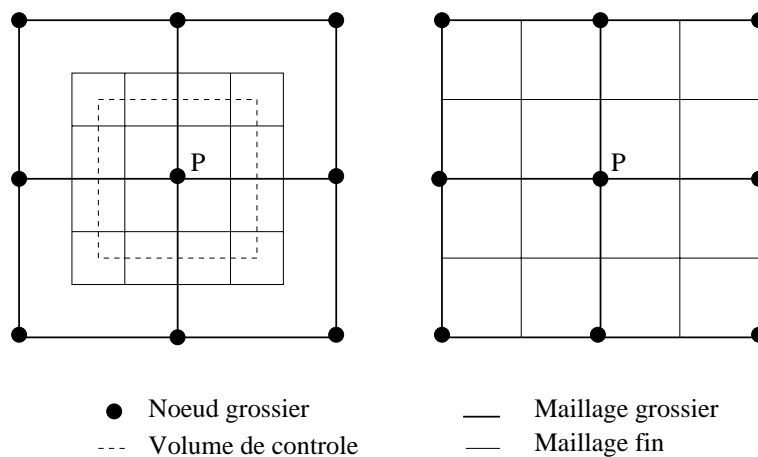


Figure 1.10 : *Raffinement de maillage par la méthode OCLM à gauche et par une méthode AMR classique à droite.*

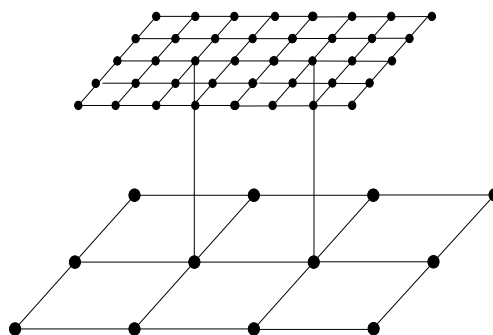


Figure 1.11 : *Raffinement de maillage par la méthode OCLM*

[AIT-ALI-YAHA 00], [PIERCE 03] entre autre. Elle consiste à adapter le maillage en deux étapes : dans la première on applique un raffinement de type  $p$ , puis dans la deuxième un raffinement de type  $h$ .

Les méthodes  $h$  et  $OCLM$  sont adaptées pour le type de problème que l'on souhaite traiter. Souhaitant garder un maillage localement structuré, nous devons abandonner la méthode  $h$  au profit de la méthode  $OCLM$ . Nous allons donc la détailler dans la sous section suivante.

Ayant précisé le type de raffinement choisi, il convient d'explicitier la manière de générer le maillage adaptatif. Deux techniques sont possibles :

Soit on décide d'adapter le maillage de façon statique, c'est-à-dire qu'on va utiliser successivement le code de calcul et un mailleur. Dans la première étape, une simulation est faite

pour déterminer la taille maximale des mailles qui permet d'atteindre la précision désirée. Dans la deuxième étape, le mailleur tient compte de cette contrainte (taille des mailles à ne pas dépasser) pour générer un nouveau maillage. On applique successivement les deux étapes jusqu'à obtention de la solution finale. Elle permet d'avoir un maillage de très bonne qualité, cependant elle est pénible d'utilisation et grosse consommatrice en temps (à chaque début de simulation, il faut recalculer les métriques,...) (cf [VENDITTI 02]) ou [HETU 92], par exemple).

Soit on utilise une adaptation dynamique de maillage, i.e. le maillage est généré par le code lui-même. Lors de la simulation, la vérification de certains critères permet d'adapter la taille des mailles de façon à atteindre la précision souhaitée. Cette méthode est connue sous le nom de méthode *AMR* (Adaptive Mesh Refinement). Elle génère des maillages de moins bonne qualité qu'avec l'utilisation d'un mailleur mais contrôle la taille des mailles de façon plus précise. De plus, elle permet de réduire l'intervention de l'utilisateur et le nombre de calculs lié au passage d'informations code-mailleur ([BERGER 84], [BERGER 89], [PEMBER 95]).

On choisira donc d'utiliser une méthode *AMR*.

### 1.3.2 La méthode One-Cell Local Multigrid *OCLM*

Cette approche consiste à augmenter la densité des points à proximité d'une interface par exemple, afin de mieux la localiser, ce qui est indispensable pour diminuer la diffusion numérique des schémas qui filtrent la turbulence et déforment les interfaces. On entend par diminution de la diffusion numérique, le fait que la distance sur laquelle il y a cette diffusion diminue quand on réduit le pas d'espace du maillage. Les schémas utilisés ne réduisent pas cette diffusion qui se prolonge sur deux à trois mailles quelquesoit le maillage. Plus précisément, la méthode *OCLM* consiste à générer des grilles fines à partir d'une grille grossière (cf figure 1.14) à proximité de l'interface lorsque celle-ci subit de fortes déformations. Ces grilles fines sont générées lorsque des critères de raffinement sont vérifiés. On en cite deux à titre indicatif : soit, on calcule l'erreur d'approximation [HAY 04], soit on utilise la variation de variables physiques (gradient, seuil,...). On a choisi le second critère qui paraît plus adapté aux problèmes physiques étudiés dans ce travail. Lorsqu'un nœud scalaire  $P_{m,l-1}$  du niveau  $G_{L-1}$  vérifie ce critère, le volume de contrôle autour de ce point est raffiné (cf figure 2.40). La grille fine ainsi créée sera appelée cellule *AMR*. Le niveau  $G_L$  est alors construit. En maillage décalé, on utilise un découpage impair pour

assurer une connexion conforme entre les cellules d'un même niveau fin. Le coefficient de raffinement  $RAF$ , indique le rapport entre le pas d'espace de niveau  $G_{L-1}$  et celui de niveau  $G_L$ . Avoir un pas d'espace de niveau  $G_L$  trois fois plus petit que celui de niveau  $G_{L-1}$ , revient à imposer  $RAF = 3$ . La figure 1.12 montre à gauche une cellule AMR avec  $RAF = 3$  et à droite avec  $RAF = 5$ .

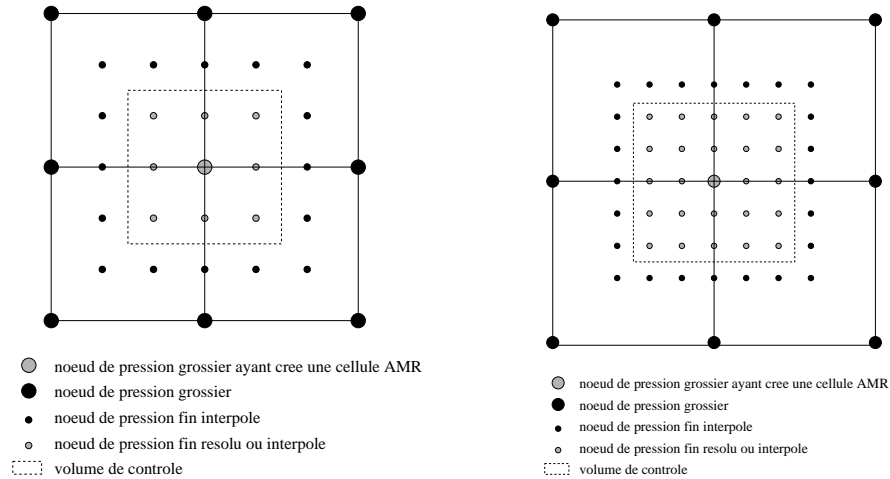


Figure 1.12 : *coefficient de raffinement 3 à gauche et 5 à droite*

Seuls les nœuds intérieurs aux cellules de niveau  $G_L$  peuvent à leur tour créer une cellule de niveau  $G_{L+1}$ . Il en résulte une structure emboîtée des niveaux, ie  $G_L, G_{L+1}, G_{L+2}, \dots$ . On crée des niveaux de plus en plus fin jusqu'à obtenir la précision requise ( voir figure 1.13).

On note aussi la structure en arbre des niveaux fins sur la figure 1.15. Cette structure signifie que chaque nœud  $LP$ , de niveau  $G_L$  a des fils appartenant au niveau  $G_{L+1}$ ,  $LP$  a un père de niveau  $G_{L-1}$ .

On précise que pour déterminer l'inconnu pression  $P_m$  d'un niveau  $G_L$  à partir des équations de discrétisation, nous avons besoin des informations contenues dans les nœuds voisins de pression de même niveau ( $P_e, P_w, P_s, P_n$ ), vitesse de même niveau ( $U_e, U_w, V_s, V_n$ ) et viscosité de même niveau ( $M_{ne}, M_{nw}, M_{se}, M_{sw}$ ) comme le montre la figure 1.16. Les inconnues vitesses  $U_m$  et  $V_m$  sont déterminées de la même façon ( figures 1.17 et 1.18). Considérons une cellule AMR de niveau  $G_L$ . Les nœuds de pression de cette cellule sont créés dès que le critère de raffinement est vérifié par le nœud de pression de niveau  $G_{L-1}$ . Les nœuds vitesses fins et viscosités fines sont ensuite créés comme voisins fins des

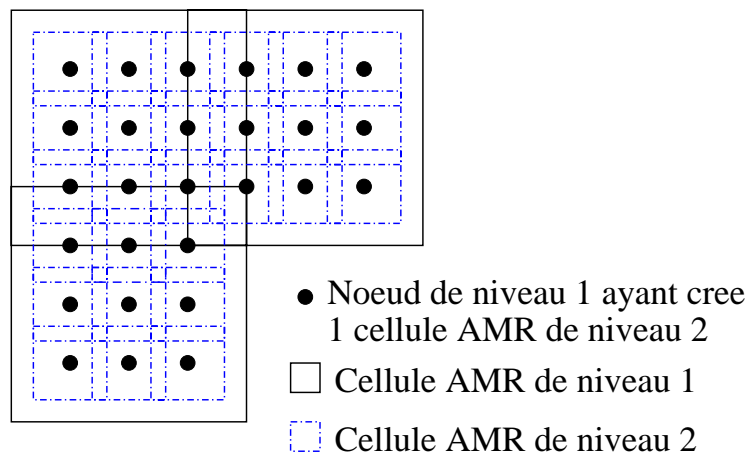


Figure 1.13 : Construction du niveau 2 à partir du niveau 1.

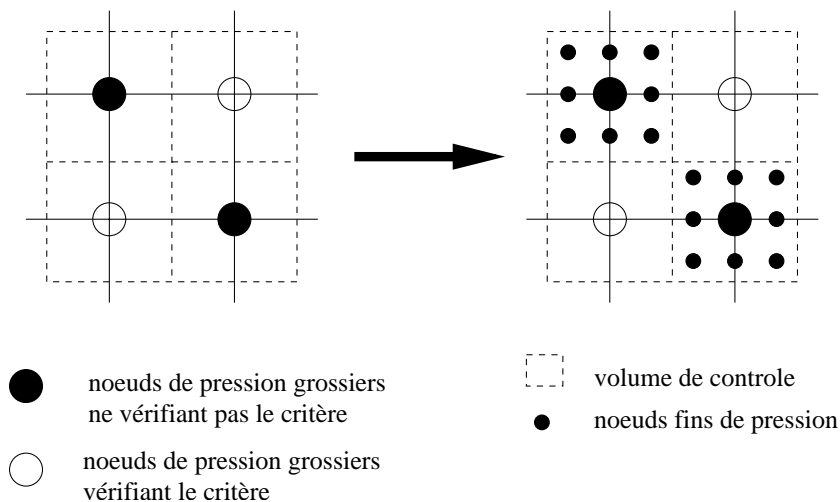


Figure 1.14 : création d'un niveau de grille fin

noeuds de pression de la cellule. La figure 1.19 montre le traitement des différents noeuds de pression et de vitesse. Les noeuds aux limites de la cellule sont interpolés (interpolation  $Q_1$ ), tandis que ceux à l'intérieur sont calculés à partir des équations de conservation discrétisées. Le choix du traitement des noeuds fins se fait conformément aux schémas de discrétisation : les noeuds pression et vitesses représentés en noir sur la figure 1.19 ont au moins un voisin manquant, ce qui nous conduit à calculer la solution en ces noeuds grâce à une procédure d'interpolation.

Soit  $t_0$  le temps initial où les grandeurs physiques et autres paramètres sont initialisés. La phase d'initialisation de l'AMR se fait soit en interpolant la solution du niveau grossier



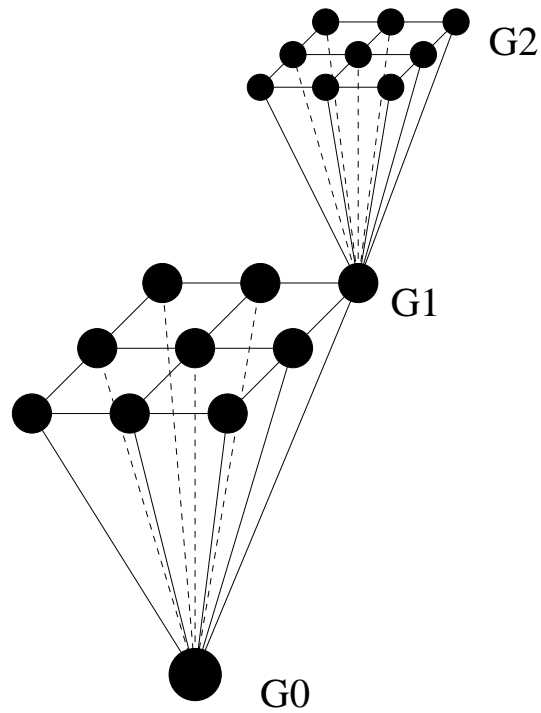


Figure 1.15 : Liens de parenté entre des nœuds de différents niveaux AMR pour un coefficient de raffinement  $RAF = 3$ .

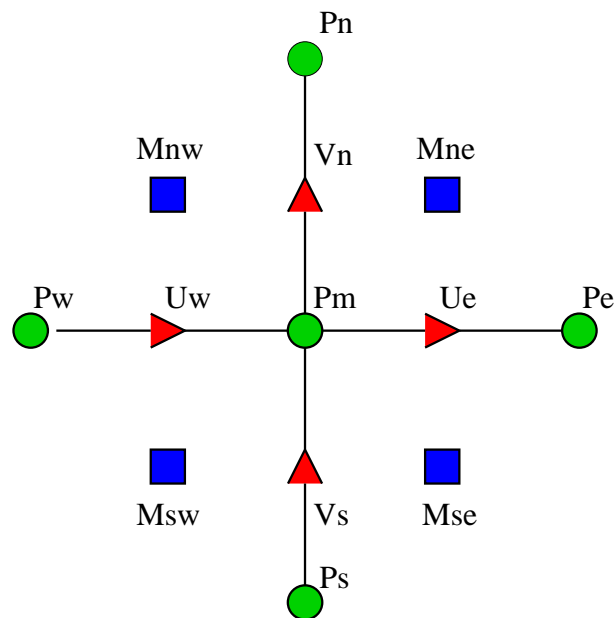


Figure 1.16 : Schéma de discrétisation pour les nœuds de pression

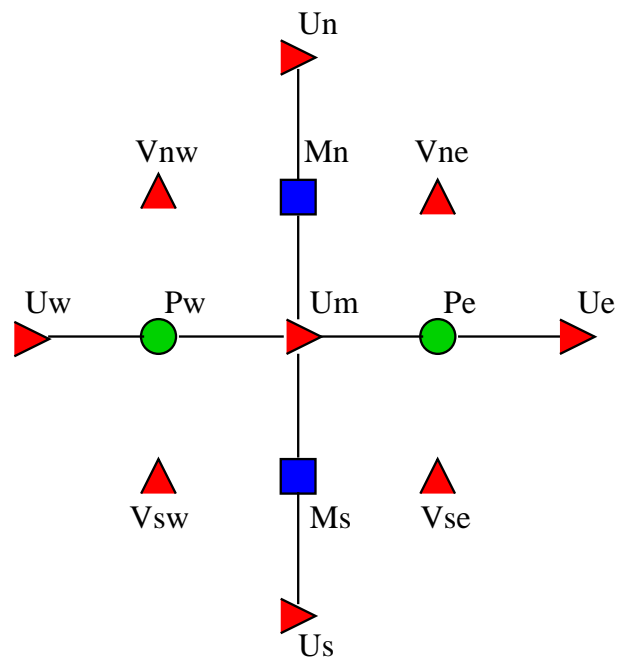


Figure 1.17 : Schéma de discrétisation pour les nœuds de vitesses horizontales

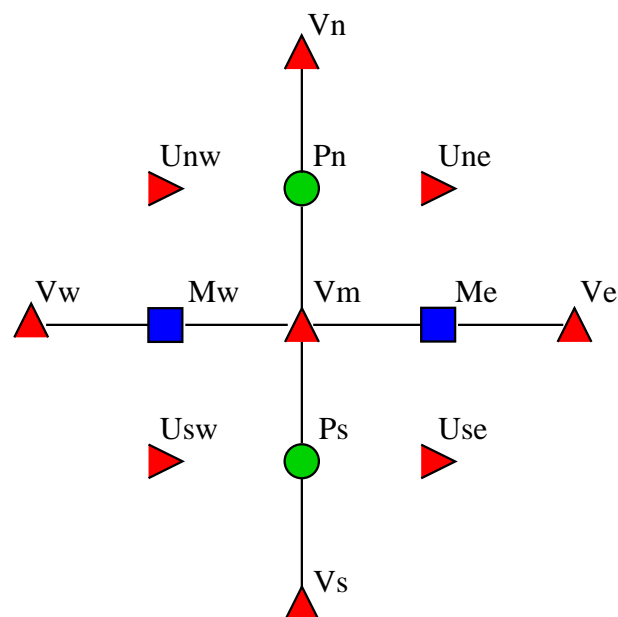


Figure 1.18 : Schéma de discrétisation pour les nœuds de vitesses verticales

sur le niveau fin quand les cellules *AMR* sont créées à partir d'un temps  $t > t_0$ ; soit de la même façon que les autres éléments du maillage grossier  $G_0$  quand les cellules *AMR* sont

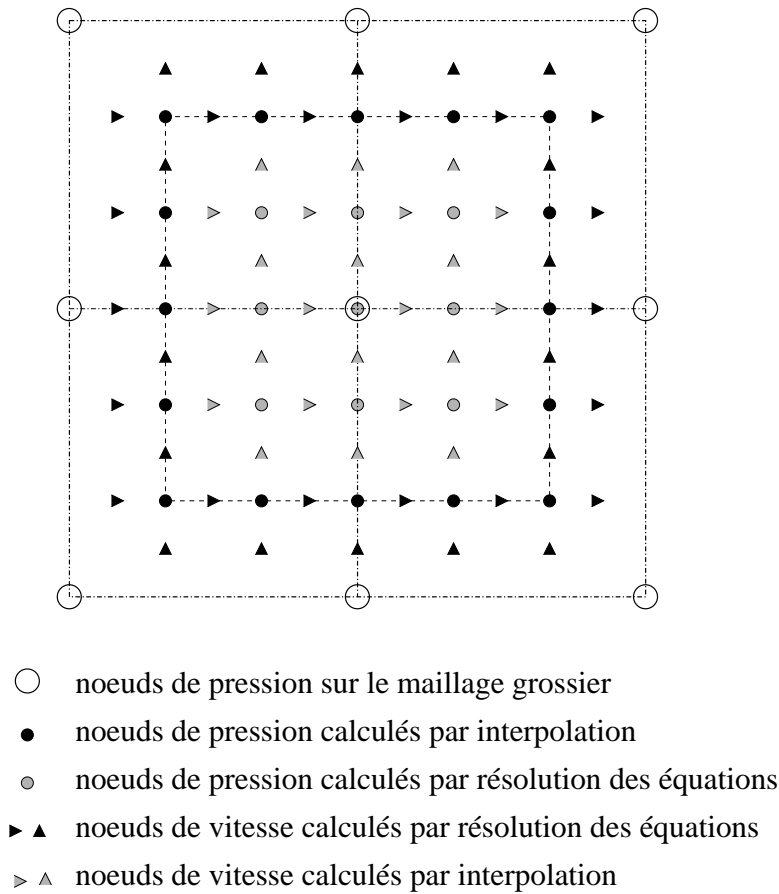


Figure 1.19 : détermination de l'inconnue sur les nœuds fins

créées à partir d'un temps  $t = t_0$ .

La solution obtenue sur un niveau fin  $G_L$  est restreinte sur un niveau grossier  $G_{L-1}$  par la méthode Full Weighting Control Volume (FWCV) [HACKBUSH 84]. Cette méthode est utilisée pour la fonction de phase dans un écoulement diphasique car elle assure la conservation de la masse. L'injection conservative consiste à enrichir la solution d'un nœud père en lui appliquant la formule (2.7).

Cependant, la méthode OCLM montre des faiblesses. Une d'elles correspond au découplage de la procédure d'interpolation et du solveur implicite. Quand on cherche la solution  $\Phi_l^{n+1}$  sur le niveau  $G_L$  au temps  $t^{n+1} = (n+1)\delta t$ , la première étape consiste à calculer la solution sur les nœuds des limites des cellules AMR de niveau  $G_L$  à partir de la solution connue au temps  $t^n = (n)\delta t$ , sur les nœuds de niveau  $G_L$ , en utilisant une procédure d'interpolation. Donc la solution  $\Phi_l^{n+1}$  sur les nœuds des limites des cellules AMR est une

combinaison linéaire de la solution  $\Phi_{l-1}^n$  au temps  $t^n = (n)\delta t$ . Ceci provoque une première erreur car la solution  $\Phi_l^{n+1}$  en ces nœuds de limites ne représente pas réellement la solution au temps  $t^{n+1}$ , mais plutôt au temps  $t^n$ .

Dans la deuxième étape, on utilise un solveur implicite pour trouver la solution  $\Phi_l^{n+1}$  sur les nœuds intérieurs aux cellules *AMR*. En considérant les schémas de discrétisation des figures 1.16, 1.17 et 1.18, on se rend compte que la plupart de ces nœuds intérieurs ont au moins un voisin qui est un nœud interpolé. Donc l'erreur qui a été commise sur ces nœuds interpolés se propage sur ceux à l'intérieur des cellules *AMR*. Une première correction à apporter à cette méthode est de rendre l'interpolation implicite dans le but de déterminer  $\Phi_l^{n+1}$  sur les nœuds intérieurs et aux limites des cellules *AMR* en une seule étape. Donc on interpole la solution  $\Phi_l^{n+1}$  à partir de la solution  $\Phi_{l-1}^{n+1}$  et non plus  $\Phi_{l-1}^n$  (l'implication de l'interpolation est détaillée dans la section 2.6).

La méthode *AMR* ainsi développée devient plus efficace que la méthode *OCLM*. Elle représente réellement la solution  $\Phi_l^{n+1}$  au temps  $t^{n+1}$ . Cependant la résolution des équations discrétisées donne des solutions plus précises que l'interpolation. Comme on interpole tous les nœuds appartenant aux limites des cellules *AMR*, on se retrouve avec un grand nombre de nœuds à interpoler par rapport à ceux résolus par les équations de discrétisation. La deuxième amélioration consiste donc à connecter ces cellules entre-elles. Ceci permet le passage d'informations de cellule en cellule. On verra par la suite comment cela permet de réduire considérablement le nombre de nœuds à interpoler et donc d'obtenir une bonne précision de la solution  $\Phi_l^{n+1}$ .

## Chapitre 2

# Améliorations apportées à la méthode *OCLM*

### 2.1 Principe de connexion

Dans la méthode One-Cell Local Multigrid (*OCLM*) développée par [VINCENT 00], les cellules *AMR* d'un même niveau  $G_L$  sont indépendantes les unes des autres. Ainsi les informations contenues dans chacune d'elle proviennent d'une résolution locale et des informations du niveau plus grossier  $G_{L-1}$ . En aucun cas il n'y a passage d'information d'une cellule *AMR* à l'autre. Le nombre de points interpolés a donc une action significative sur la précision de la solution. Comme cela a été expliqué dans la première partie, une des améliorations à apporter à la méthode *OCLM* est la mise en place d'une technique de connexion entre cellules d'un même niveau de raffinement.

On rappelle que les nœuds d'une cellule *AMR* sont traités différemment selon leur position dans la cellule : les nœuds aux limites sont résolus par interpolation tandis que ceux à l'intérieur sont résolus par discrétisation des équations de conservation. L'objectif est de minimiser le nombre de nœuds interpolés.

On expose ci-dessous la technique de connexion pour les nœuds scalaires et ceux de vitesses. Globalement, la démarche est la même, cependant quelques nuances apparaissent concernant le choix des nœuds à résoudre. On décide donc de détailler le principe pour les nœuds scalaires et de se restreindre à l'explication des nuances de la technique pour les nœuds vitesses. Les principes décrits dans les parties suivantes sont généralisables à trois dimensions d'espace et à plusieurs niveaux fins. Le principe de parcours des nœuds est valable pour tout type de maillages curvilignes structurés orthogonaux.

### 2.1.1 Traitement des nœuds scalaires

On prend le cas de deux nœuds voisins d'un même niveau  $G_{L-1}$ . Lorsque chacun d'eux génère une cellule  $AMR$  de niveau  $G_L$ , il y a recouvrement (voir figure 2.1). Les nœuds appartenant aux limites des cellules  $AMR$  et étant dans la zone de recouvrement ne sont plus résolus. Tandis que ceux appartenant à l'intérieur (strict) des cellules  $AMR$  et étant dans la zone de recouvrement sont connectés à des nœuds appartenant à l'intérieur (strict) de la cellule voisine. Prenons le cas d'un nœud  $P_M^G$  appartenant à l'intérieur (strict) de la cellule  $AMR$  de gauche et étant dans la zone de recouvrement (se référer à la figure 2.1). Pour simplifier la démarche, on suppose que dans le schéma de discrétisation, il est nécessaire de connaître l'inconnue aux nœuds voisins Ouest  $P_W^G$ , Est  $P_E^G$ , Nord  $P_N^G$  et Sud  $P_S^G$ , pour déterminer l'inconnue  $P_M^G$ . En utilisant la figure 2.1, on observe que le nœud  $P_E^G$  appartient à la fois aux limites de la cellule  $AMR$  gauche et à la zone de recouvrement, il n'est donc pas résolu. La figure 2.1 montre qu'un autre nœud appartenant à l'intérieur de la cellule de droite et étant dans la zone de recouvrement,  $P_E^D$ , a les mêmes coordonnées que le nœud  $P_E^G$ .  $P_E^D$  devient le nouveau voisin Est de  $P_M^G$ . On suit une démarche similaire pour tous les autres nœuds appartenant à l'intérieur (strict) des cellules  $AMR$  et à la zone de recouvrement. On connecte ainsi les deux cellules  $AMR$ . Seuls les nœuds appartenant aux limites des cellules  $AMR$  et n'étant pas dans la zone de recouvrement sont interpolés.

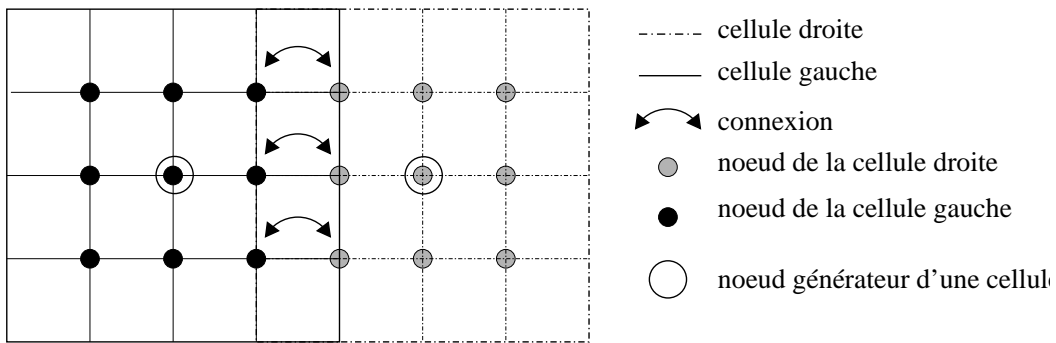


Figure 2.1 : Connexion des nœuds scalaires de deux cellules  $AMR$ .

On généralise le cas à plusieurs cellules  $AMR$  d'un même niveau  $G_L$  sur la figure 2.2. Chaque cellule conserve son identité, à savoir son numéro de cellule, les numéros des nœuds fins, sa position, les pas d'espaces associés à chaque nœud, ... On peut ainsi gérer le raffinement et le déraffinement de manière adaptative en temps et en espace. L'originalité

de cette technique réside dans le fait que cette succession de cellules est traitée comme un unique bloc lors de la résolution. Le passage d'informations entre cellules d'un même niveau est donc établi.

On note aussi que le nombre de nœuds de pression interpolés devient très petit devant le nombre de nœuds de pression résolus, ce qui permet d'avoir une meilleure précision de la solution.

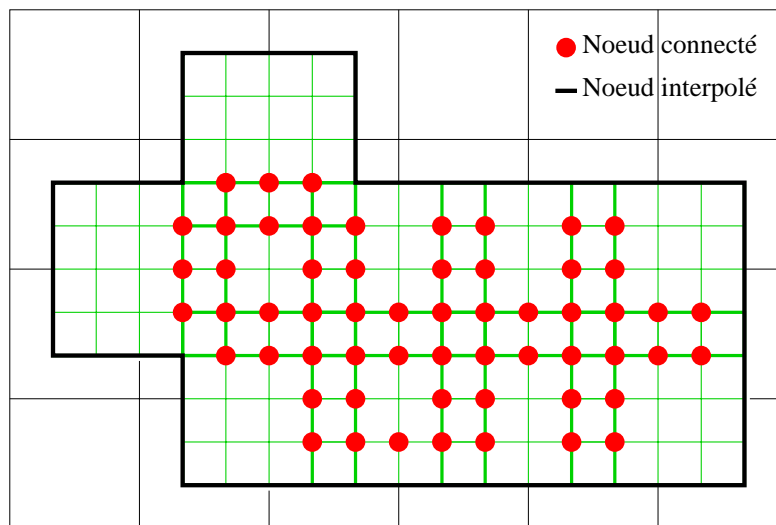


Figure 2.2 : Connexion de plusieurs cellules AMR.

On ajoute que seuls les nœuds appartenant à l'intérieur (strict) de chaque cellule  $AMR$  de niveau  $G_L$  et n'ayant aucun voisin interpolé, peuvent générer à leur tour une cellule de niveau  $G_{L+1}$  (cf figure 2.3). La première condition assure l'unicité des cellules  $AMR$  : en effet, si on suppose qu'un des nœuds de la figure 2.1, appartenant à la limite de la cellule  $AMR$  gauche et étant dans la zone de recouvrement génère une cellule, il existe un nœud appartenant à l'intérieur (strict) de la cellule droite qui a les mêmes coordonnées que le premier nœud. Ces deux nœuds étant superposés, si chacun d'eux crée une cellule  $AMR$  de niveau supérieur, on se retrouve alors avec deux cellules identiques l'une superposée à l'autre. C'est pour cela que nous interdisons la création de cellules de niveau  $G_{L+1}$  à partir d'un nœud appartenant à la limite d'une cellule  $AMR$  de niveau  $G_L$ . La deuxième condition permet d'assurer l'emboîtement strict des niveaux fins : par exemple, le niveau  $G_{L+1}$  est donc strictement emboîté dans le niveau  $G_L$ .

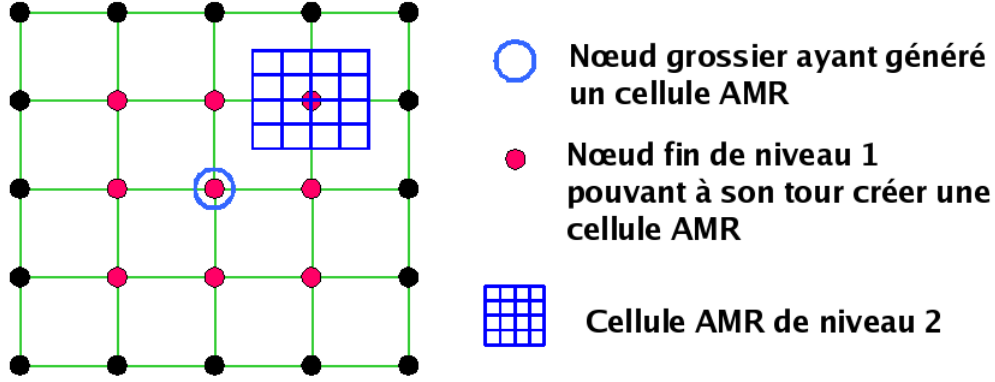


Figure 2.3 : Création d'un deuxième niveau d'AMR.

Techniquement, on stocke dans un tableau *BLIM* tous les nœuds scalaires appartenant aux limites de chaque cellule *AMR*. C'est assez simple à réaliser puisque chacune d'entre-elles a conservé son identité. Il en est de même pour les nœuds scalaires eux-mêmes. Ainsi, le numéro d'un nœud scalaire fin dans la numérotation globale est  $LTP = KKT I + (IC - 1)DIMTT + I + K(RAF + 2) + 1$ , où  $KKT I$  est le nombre de nœuds scalaires sur  $G_0$ ,  $IC$  le numéro de cellule courant,  $DIMTT$  le nombre de nœuds scalaires fin par cellule *AMR*,  $RAF$  le coefficient de raffinement,  $I \in \{0, RAF + 1\}$  l'indice de parcours horizontal des nœuds et  $K \in \{0, RAF + 1\}$  l'indice de parcours vertical.

Ainsi les nœuds aux limites de chaque cellule *AMR* de niveau  $G_L$  sont :

- $P_{G,I,K}^L$  tel que  $\{I = 0, K \in \{1, RAF\}\}$  pour la limite gauche.
- $P_{D,I,K}^L$  tel que  $\{I = RAF + 1, K \in \{1, RAF\}\}$  pour la limite droite.
- $P_{B,I,K}^L$  tel que  $\{K = 0, I \in \{1, RAF\}\}$  pour la limite inférieure.
- $P_{H,I,K}^L$  tel que  $\{K = RAF + 1, I \in \{1, RAF\}\}$  pour la limite supérieure.
- $P_{BG,I,K}^L$  tel que  $\{K = 0, I = 0\}$  pour le coin bas gauche.
- $P_{BD,I,K}^L$  tel que  $\{K = 0, I = RAF + 1\}$  pour le coin bas droit.
- $P_{HG,I,K}^L$  tel que  $\{K = RAF + 1, I = 0\}$  pour le coin haut gauche.
- $P_{HD,I,K}^L$  tel que  $\{K = RAF + 1, I = RAF + 1\}$  pour le coin haut droit.

Une fois que nous connaissons la liste des nœuds fins du niveau  $G_L$  appartenant aux limites de chaque cellule *AMR*, il faut déterminer ceux qui sont sur l'interface  $G_{L-1}/G_L$ .



Pour cela, on utilise la structure en arbre de la méthode de raffinement (cf figure 1.15). On considère une cellule  $AMR$  de niveau  $G_L$  générée par un nœud de pression grossier  $P_m^{L-1}$ . Pour les nœuds de pression fins  $P_{D,I,K}^L$  (resp :  $P_{G,I,K}^L$ ,  $P_{B,I,K}^L$ ,  $P_{H,I,K}^L$ ), on regarde si le nœud  $P_m^{L-1}$  possède un voisin Est  $P_e^{L-1}$  (resp : West  $P_w^{L-1}$ , Sud  $P_s^{L-1}$ , Nord  $P_n^{L-1}$ ) ayant généré une cellule  $AMR$ . Si c'est le cas, les deux cellules  $AMR$  sont connectées, sinon ces nœuds  $P_{D,I,K}^L$  (resp :  $P_{G,I,K}^L$ ,  $P_{B,I,K}^L$ ,  $P_{H,I,K}^L$ ) appartiennent à la limite droite (resp : gauche, inférieure, supérieure) de l'interface  $G_{L-1}/G_L$ .

Les coins sont des nœuds particuliers car ils appartiennent à deux limites. Il convient donc de regarder si le nœud  $P_m^{L-1}$  possède deux voisins ayant généré une cellule  $AMR$ . Soit  $P_{HD}^L$  le coin haut droit. Si aucun des voisins  $P_e^{L-1}$  et  $P_n^{L-1}$  de  $P_m^{L-1}$  n'a généré une cellule, alors  $P_{HD}^L$  reste un coin de l'interface  $G_{L-1}/G_L$ . Si seul  $P_n^{L-1}$  a généré une cellule  $AMR$ , alors  $P_{HD}^L$  devient un nœud de la limite droite de l'interface  $G_{L-1}/G_L$ . La détection des seuls voisins Est, Ouest, Sud et Nord suffit d'après le schéma de discrétisation des nœuds de pression 2.16 et 1.16. On rappelle que l'absence d'un seul voisin scalaire, vitesse ou viscosité du nœud qui est en cours de traitement provoque son interpolation. Plusieurs cas de figure sont possibles. La cellule  $AMR$  traitée peut avoir entre zéro et huit cellules voisines. On note que quand celle-ci a un nombre de cellules voisines supérieur ou égal à 4, aucun des nœuds fins appartenant à ses limites ne sont interpolés. Lorsque la cellule  $AMR$  traitée possède une voisine (respectivement deux, trois voisines), plusieurs combinaisons (position des cellules voisines par rapport à celle traitée) sont possibles. Seuls quelques cas sont représentés ci-dessous pour illustration (cf figures 2.4, 2.5 et 2.6). Les nœuds scalaires fins de la cellule traitée sont représentés en gris quand il faut les interpoler. Le cas avec quatre voisins ne figure pas parmi les schémas suivants car aucun nœud scalaire fin n'est interpolé.

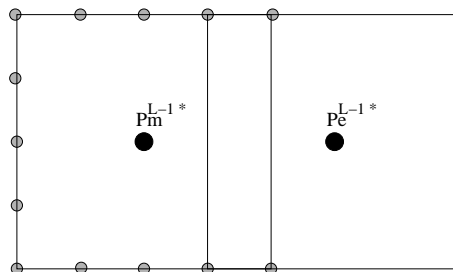


Figure 2.4 : Une seule voisine a généré une cellule  $AMR$  (Nœuds interpolés en gris).

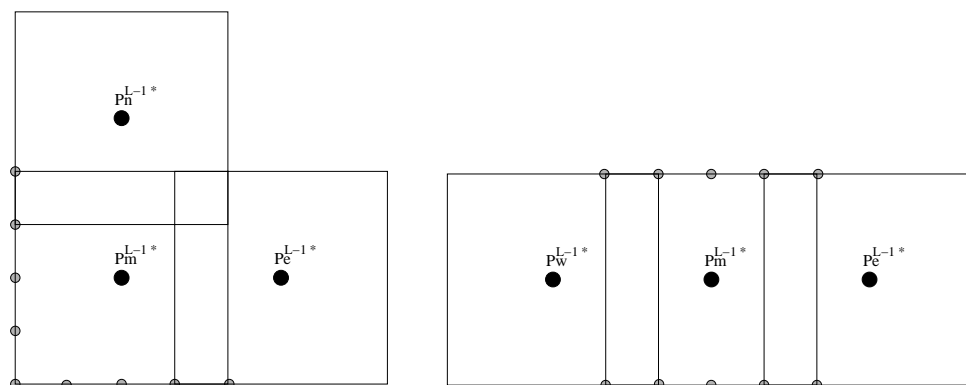


Figure 2.5 : Deux voisines ont généré une cellule *AMR* (Nœuds interpolés représentés en gris).

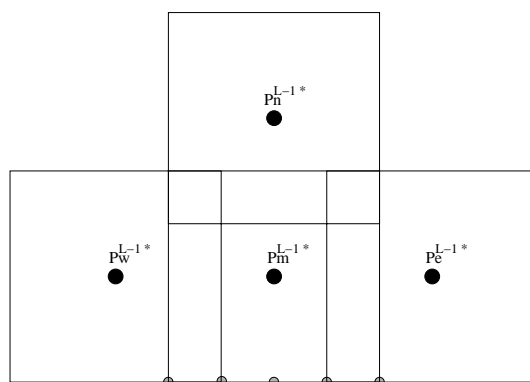


Figure 2.6 : Trois voisines ont généré une cellule *AMR* (Nœuds interpolés en gris).

### 2.1.2 Traitement des nœuds vitesses

Le raisonnement pour les nœuds vitesses fins est un peu différent de celui fait pour les nœuds fins scalaires. Ceci est dû au fait qu'une cellule de niveau  $G_L$  est créée à partir d'un nœud de pression  $P_m^{L-1}$ , du niveau  $G_{L-1}$ .

On prend le cas d'un nœud scalaire de niveau  $G_{L-1}$ ,  $P_m^{L-1}$  ayant généré une cellule *AMR* de niveau  $G_L$ . On suppose que son voisin Est,  $P_E^{L-1}$ , en a lui aussi créé une. On a vu dans la sous-section précédente qu'il y a une zone de recouvrement pour les nœuds scalaires (cf figure 2.1). Étant donné que chaque cellule *AMR* a la structure d'un maillage décalé de type *MAC*, il existe aussi une zone de recouvrement pour les nœuds vitesses selon l'axe  $x$  (voir figure 2.7) et selon l'axe  $z$  (voir figure 2.8). La démarche pour les nœuds vitesses est similaire à celle des nœuds scalaires. On souhaite que dans la zone de recouvrement, un

seul des nœuds superposés soit résolu. On rappelle qu'un nœud appartenant à l'intérieur (strict) d'une cellule *AMR* est toujours prioritaire devant un nœud de limite de la cellule *AMR* voisine, superposé au premier (voir figures 2.7 et 2.8). Pour les nœuds vitesses  $u$  de la figure 2.7, il existe aussi dans la zone de recouvrement, des nœuds vitesses superposés, qui appartiennent tous les deux à l'intérieur strict d'une cellule *AMR* différente. Il faut donc choisir de façon arbitraire lequel des deux on va résoudre. On note  $M_1 = \{\text{nœuds appartenant à la fois à l'intérieur strict d'une cellule } AMR, \text{ à la zone de recouvrement et ayant un voisin appartenant à la limite de la même cellule et à la zone de recouvrement}\}$ . La figure 2.7 montre que les nœuds vitesses de la cellule gauche appartenant à  $M_1$  sont superposés à ceux de la cellule de droite appartenant à  $M_1$ . On choisit de résoudre ceux de la cellule de gauche.

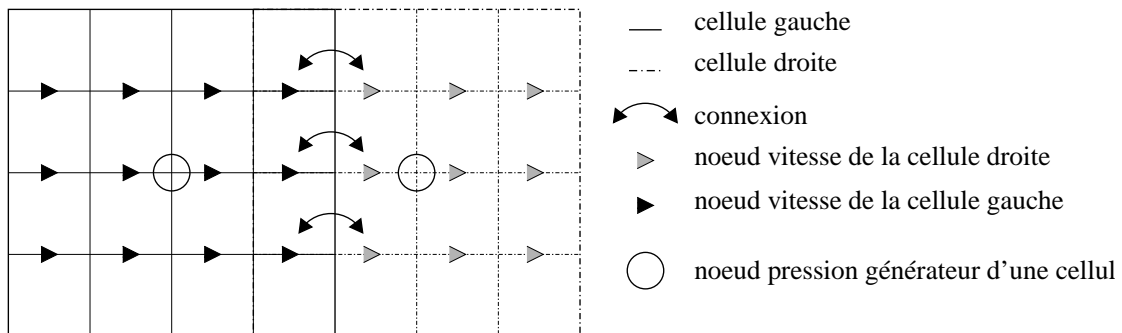


Figure 2.7 : Connexion des nœuds vitesses selon l'axe  $x$  de deux cellules *AMR* générées par un nœud scalaire et son voisin Est. Nœuds interpolés représentés en gris.

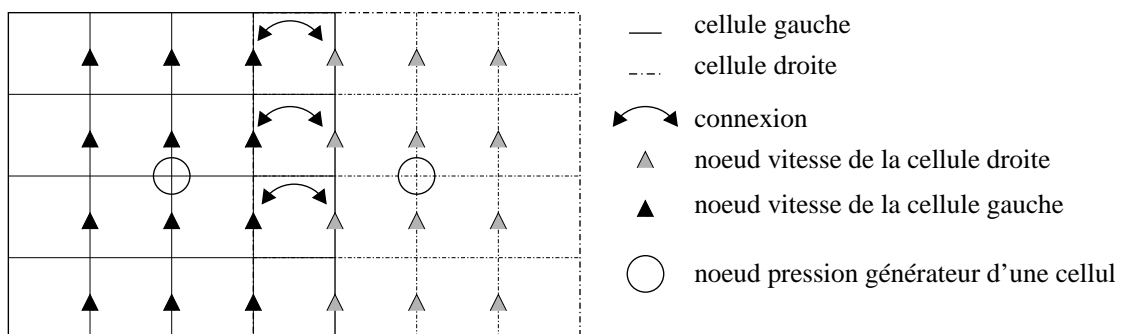


Figure 2.8 : Connexion des nœuds vitesses selon l'axe  $z$  de deux cellules *AMR* générées par un nœud scalaire et son voisin Est. Nœuds interpolés représentés en gris.

On considère maintenant le cas d'un nœud scalaire de niveau  $G_{L-1}$ ,  $P_m^{L-1}$  ayant généré

une cellule de niveau  $G_L$ . Son voisin Sud,  $P_S^{L-1}$ , a lui aussi créé une cellule *AMR*. On note sur les figures 2.9 et 2.10 une zone de recouvrement. Les nœuds appartenant à la fois à l'intérieur (strict) d'une cellule *AMR* et à la zone de recouvrement sont toujours prioritaires devant ceux superposés aux premiers et appartenant aux limites de la cellule *AMR* voisine. De même que précédemment, on observe pour les nœuds vitesses  $w$  de la figure (2.10), qu'il existe aussi dans la zone de recouvrement, des nœuds vitesses superposés, qui appartiennent tous les deux à l'intérieur strict d'une cellule *AMR* différente. En adoptant les mêmes notations et en suivant un raisonnement identique, on choisit de résoudre les nœuds de la cellule du bas appartenant à  $M_1$ . La conservation d'identité de chaque nœud

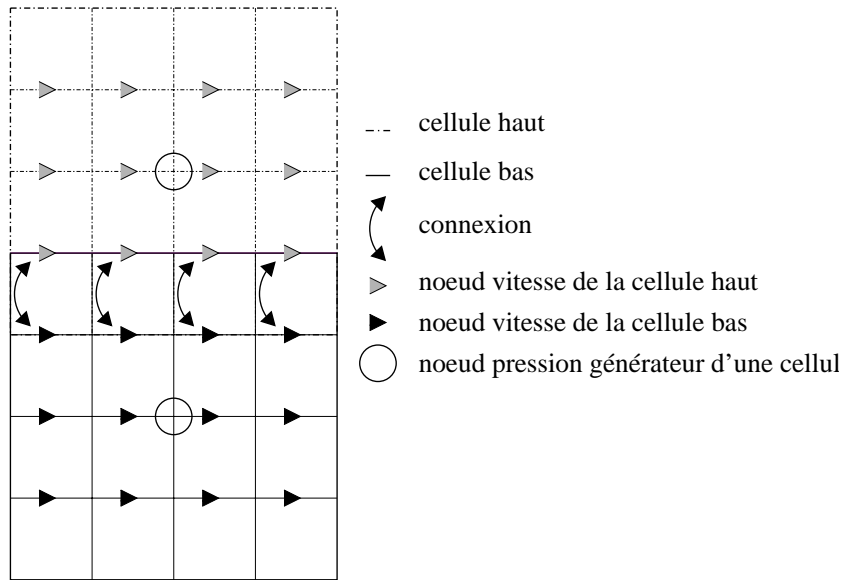


Figure 2.9 : Connexion des nœuds vitesses selon l'axe  $x$  de deux cellules *AMR* générées par un nœud scalaire et son voisin Sud.

fin d'une cellule *AMR* permet de connaître le numéro global des nœuds vitesses horizontales  $U$  et verticales  $V$ . Sachant qu'on parcourt pour chaque cellule *AMR* les vitesses  $U$  puis  $V$ , le numéro global est :  $LVP1 = KKVI + (IC - 1)DIMTV + I + K(RAF + 3) + 1$  pour  $U$  et  $LVP2 = KKVI + (IC - 1)DIMTV + I + K(RAF + 2) + 1 + (RAF + 3)(RAF + 2)$  pour  $V$ . Avec  $KKVI$  le nombre de nœuds vitesses sur  $G_0$ ,  $IC$  le numéro de cellule courant,  $DIMTV$  le nombre de nœuds vitesses fins par cellule *AMR*,  $RAF$  le coefficient de raffinement,  $I \in \{0, RAF + 2\}$  pour  $U$  (resp  $I \in \{0, RAF + 1\}$  pour  $V$ ) l'indice de parcours horizontal des nœuds et  $K \in \{0, RAF + 1\}$  pour  $U$  (resp  $K \in \{0, RAF + 2\}$  pour  $V$ ) l'indice de parcours vertical des nœuds. Les nœuds vitesses appartenant aux limites des

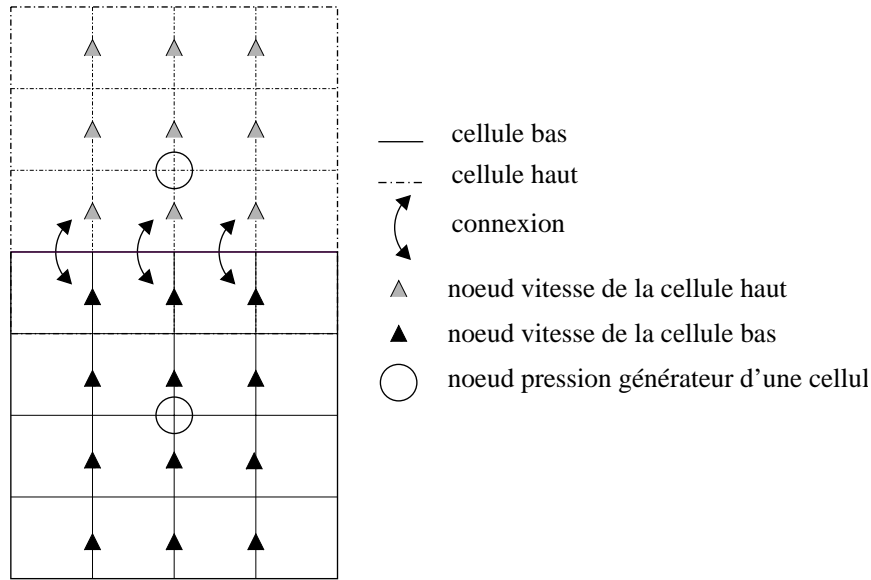


Figure 2.10 : Connexion des nœuds vitesses selon l'axe  $z$  de deux cellules AMR générées par un nœud scalaire et son voisin Sud.

cellules AMR de niveau  $G_L$  sont pour  $U$  :

- $U_{G,I,K}^L$  tel que  $\{I = 0, K \in \{1, RAF\}\}$  pour la limite gauche.
- $U_{D,I,K}^L$  tel que  $\{I = RAF + 2, K \in \{1, RAF\}\}$  pour la limite droite.
- $U_{B,I,K}^L$  tel que  $\{K = 0, I \in \{1, RAF + 1\}\}$  pour la limite inférieure.
- $U_{H,I,K}^L$  tel que  $\{K = RAF + 1, I \in \{1, RAF + 1\}\}$  pour la limite supérieure.
- $U_{BG,I,K}^L$  tel que  $\{K = 0, I = 0\}$  pour le coin bas gauche.
- $U_{BD,I,K}^L$  tel que  $\{K = 0, I = RAF + 2\}$  pour le coin bas droit.
- $U_{HG,I,K}^L$  tel que  $\{K = RAF + 1, I = 0\}$  pour le coin haut gauche.
- $U_{HD,I,K}^L$  tel que  $\{K = RAF + 1, I = RAF + 2\}$  pour le coin haut droit.

et pour  $V$  :

- $V_{G,I,K}^L$  tel que  $\{I = 0, K \in \{1, RAF + 1\}\}$  pour la limite gauche.
- $V_{D,I,K}^L$  tel que  $\{I = RAF + 1, K \in \{1, RAF + 1\}\}$  pour la limite droite.

- $V_{B,I,K}^L$  tel que  $\{K = 0, I \in \{1, RAF\}\}$  pour la limite inférieure.
- $V_{H,I,K}^L$  tel que  $\{K = RAF + 2, I \in \{1, RAF\}\}$  pour la limite supérieure.
- $V_{BG,I,K}^L$  tel que  $\{K = 0, I = 0\}$  pour le coin bas gauche.
- $V_{BD,I,K}^L$  tel que  $\{K = 0, I = RAF + 1\}$  pour le coin bas droit.
- $V_{HG,I,K}^L$  tel que  $\{K = RAF + 2, I = 0\}$  pour le coin haut gauche.
- $V_{HD,I,K}^L$  tel que  $\{K = RAF + 2, I = RAF + 1\}$  pour le coin haut droit.

Une fois la liste de ces nœuds établie, on doit déterminer ceux qui appartiennent à l'interface  $G_{L-1}/G_L$ . Celle-ci est déterminée à partir de la liste des nœuds de pression appartenant à l'interface  $G_{L-1}/G_L$ . Soit  $P_{m,i}^{L-1*}$   $i \in \{1, KKTI\}$  l'ensemble des nœuds de pression grossiers ayant généré une cellule *AMR* dont au moins une limite est incluse dans l'interface  $G_{L-1}/G_L$ . Considérons  $P_{m,i}^{L*}$   $i \in \{1, DIMTT\}$  l'ensemble des nœuds créés par un nœud  $P_m^{L-1*}$ . Lorsque  $P_{m,i}^{L*}$  ne sont pas des coins, la recherche des quatre voisins grossiers de  $P_m^{L-1*}$  suffit pour savoir si on doit interpoler les nœuds vitesses fins des limites de chaque cellule *AMR*. Pour les nœuds de pression se trouvant aux coins d'une cellule *AMR*, ce n'est plus suffisant ( cf schémas de discrétisation des nœuds de vitesse 2.17 et 2.18), il faut aussi tester les voisins obliques. Ainsi, pour le coin bas gauche (resp droit) on teste le voisin  $P_{sw,i}^{L-1*}$  (resp  $P_{se,i}^{L-1*}$ ). Pour le coin haut gauche (resp droit), on teste le voisin  $P_{nw,i}^{L-1*}$  (resp  $P_{ne,i}^{L-1*}$ ).

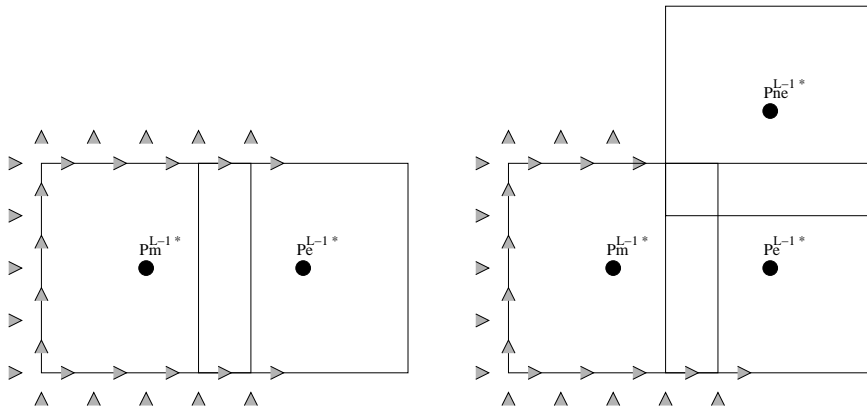


Figure 2.11 : Un seul voisin Est a généré une cellule *AMR*. Aucun voisin oblique (gauche), un voisin oblique (droite). Nœuds interpolés représentés en gris.

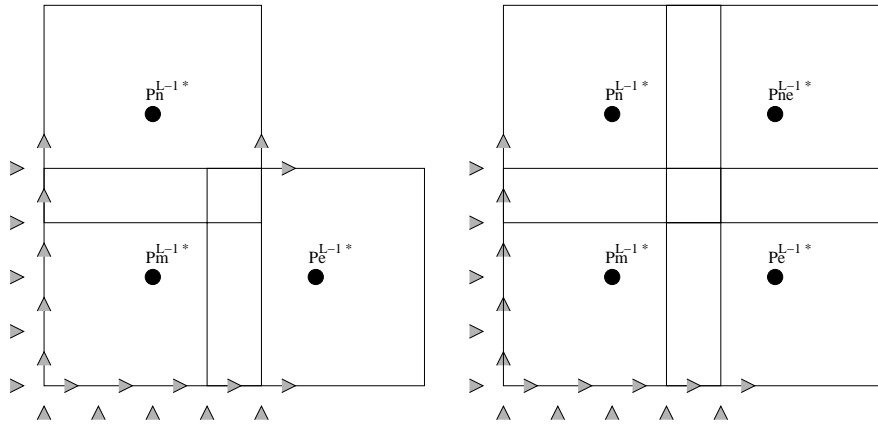


Figure 2.12 : Deux voisins ont généré une cellule AMR. Aucun voisin oblique (gauche), un voisin oblique (droite). Nœuds interpolés représentés en gris.

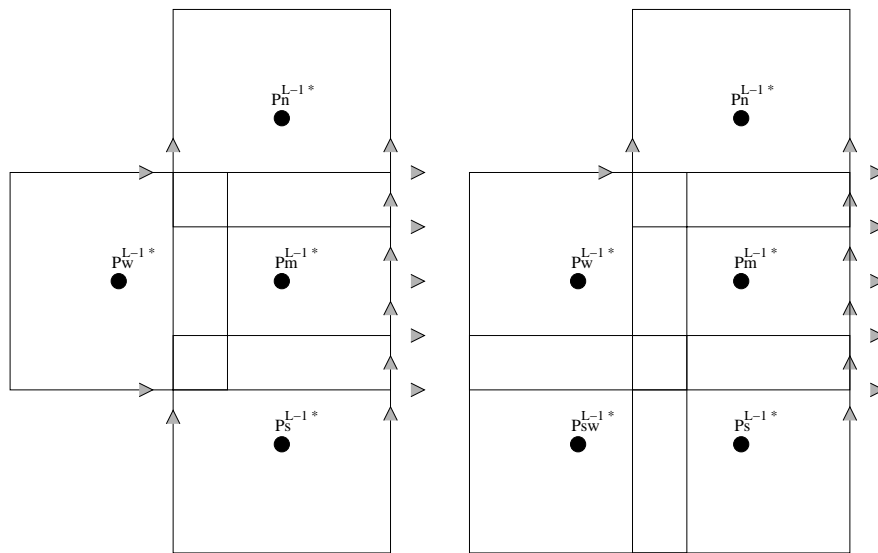


Figure 2.13 : Trois voisins ont généré une cellule AMR. Aucun voisin oblique (gauche), un voisin oblique (droite). Nœuds interpolés représentés en gris.

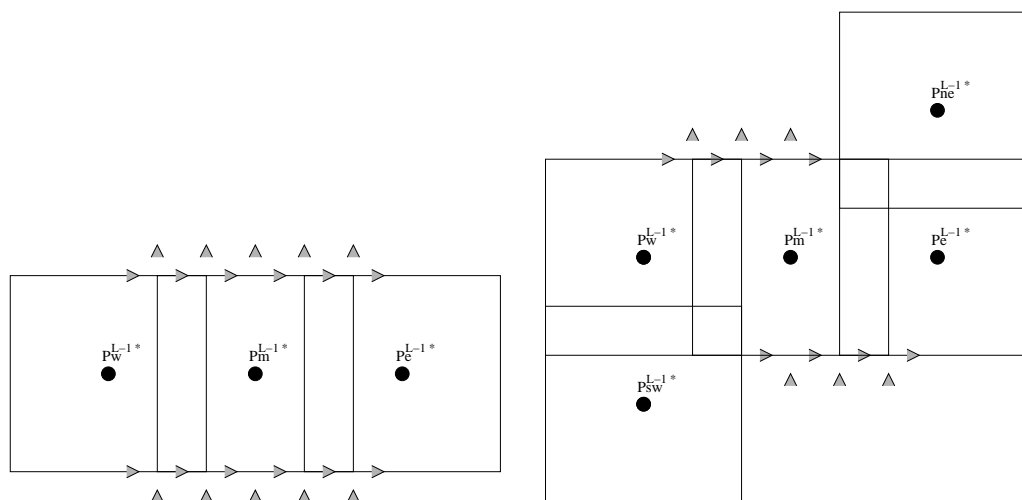


Figure 2.14 : Deux voisins ont généré une cellule AMR. Aucun voisin oblique (gauche), deux voisins obliques (droite). Nœuds interpolés représentés en gris.

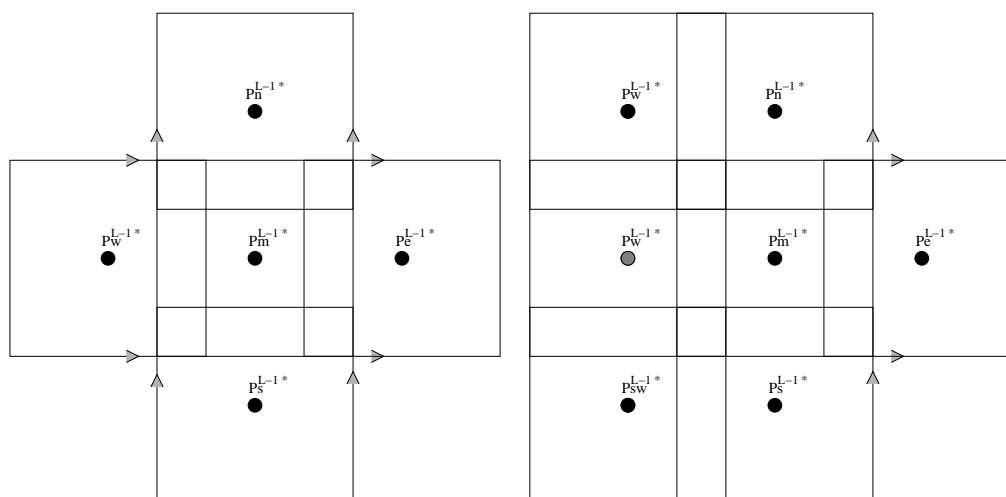


Figure 2.15 : Quatre voisins ont généré une cellule AMR. Aucun voisin oblique (gauche), deux voisins obliques (droite). Nœuds interpolés représentés en gris.

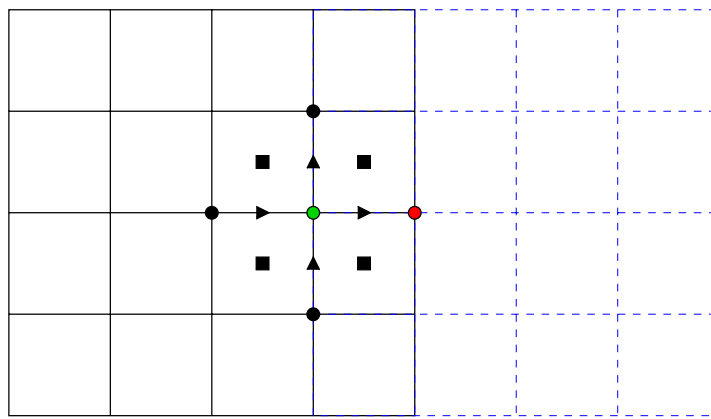
### 2.1.3 Traitement des nœuds par équation

dans les deux sous-sections qui précèdent, on a décrit le principe de connexion entre nœuds appartenant à la même grille (grille de pression, grille de vitesse). On décide maintenant de décrire le principe de connexion en suivant les schémas de discrétisation utilisés pour les équations scalaires (conservation de l'énergie, transport d'espèces inertes) et de l'équation de conservation de la quantité de mouvement.



### Équation scalaire

On rappelle qu'il s'agit d'une équation d'advection-diffusion où l'inconnue  $\Phi$  se situe sur la grille de pression. Le calcul de la solution sur un nœud  $P_M$  (représenté en vert sur la figure 2.16), nécessite la connaissance d'informations provenant des quatre nœuds scalaires voisins et des quatre nœuds vitesses voisins. La figure 2.16 montre que seul le voisin scalaire est de  $P_M$  appartient à la limite de la cellule *AMR* de gauche. Il faut donc connecter au nœud  $P_M$  un voisin scalaire appartenant à l'intérieur de la cellule *AMR* de droite.

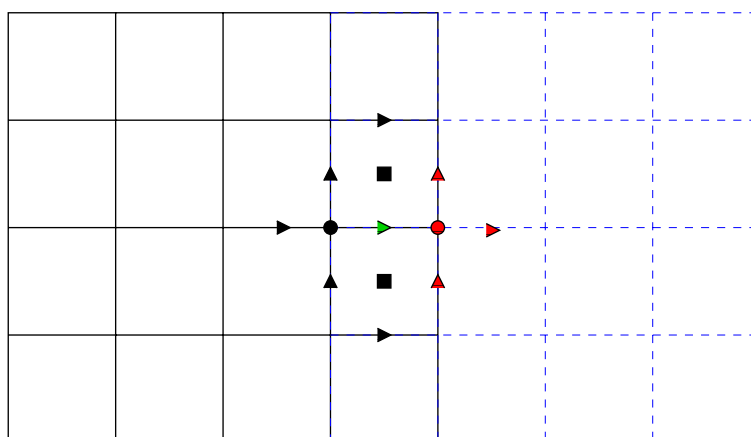


- nœud de pression de la cellule1 à connecter à la cellule2
- ▶ nœuds de la cellule1 intervenant dans la discrétisation du nœud de pression ●
- ▶ nœuds de la cellule2 nécessaires a la discrétisation du nœud de pression ●

Figure 2.16 : Détermination des voisins d'un nœud de pression par connexion en suivant le schéma de discrétisation.

### Équation de conservation de la quantité de mouvement

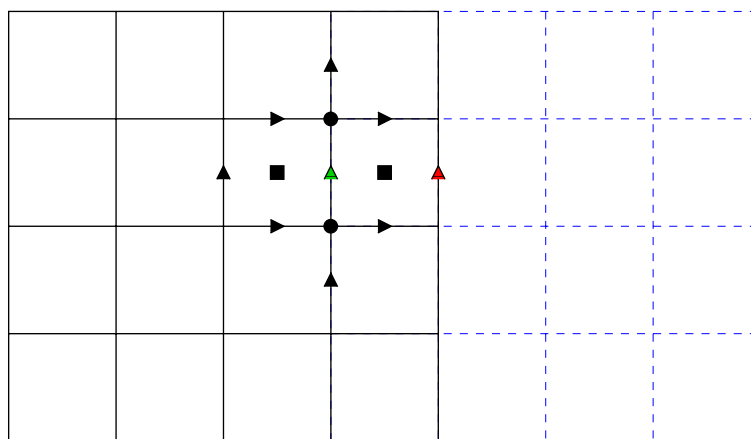
On indique ici le principe de connexion pour les nœuds vitesses  $u$ . On s'appuie sur la figure 2.17. Le calcul de la vitesse  $u$  sur le nœud  $U_M$  (représenté en vert sur la figure 2.17), nécessite, d'après le schéma de discrétisation utilisé, la connaissance d'informations provenant des huit nœuds voisins vitesses ( $u, w$ ), deux nœuds voisins sur la grille de pression et deux nœuds voisins sur la grille des viscosités. En utilisant le même code couleur que pour le cas des équations scalaires, la figure 2.17 montre que la vitesse indiquée en vert doit être connectée à trois nœuds vitesses et un nœud de pression appartenant à la cellule *AMR* de droite. La démarche est la même pour les nœuds vitesses  $w$  (voir figure



- ▶ noeud vitesse de la cellule1 à connecter à la cellule2
- ▶ noeuds de la cellule1 intervenant dans la discrétisation du noeud vitesse ▶
- ▶ noeuds de la cellule2 nécessaires à la discrétisation du noeud vitesse ▶

Figure 2.17 : Détermination des voisins d'un noeud de vitesse horizontale par connexion en suivant le schéma de discrétisation.

2.18).



- ▲ noeud vitesse vertical sur la limite de la cellule1
- ▶ noeuds de la cellule1 intervenant dans la discrétisation du noeud vitesse ▲
- ▶ noeuds de la cellule2 nécessaires a la discrétisation du noeud vitesse ▲

Figure 2.18 : Détermination des voisins d'un noeud de vitesse verticale par connexion en suivant le schéma de discrétisation.

## 2.2 Gestion dynamique de l'AMR

La construction des niveaux fins et la résolution sur les nœuds fins ont été décrites respectivement dans les sections 1.3.2 et 2.1. Reste à savoir quand et comment ces cellules *AMR* vont être créées ou détruites pour construire des niveaux de raffinement qui ont des géométries dynamiques les mieux adaptées au cas traité.

La gestion du raffinement-déraffinement se fait au travers de l'introduction de critères de raffinement-déraffinement. Le critère de raffinement consiste à générer une cellule de niveau  $G_L$  lorsqu'un nœud de niveau inférieur  $G_{L-1}$  vérifie ce critère. Quand l'écoulement est instationnaire, il se peut que des cellules *AMR* deviennent obsolètes. Il convient donc de les détruire. C'est pourquoi, un critère de déraffinement, cette fois, est défini. Comme son nom l'indique, celui-ci commande la destruction d'une cellule de niveau  $G_L$  si ces nœuds fins vérifient le critère et s'ils ne sont pas générateurs de cellules de niveau  $G_{L+1}$ . Plusieurs types de critères de raffinement-déraffinement existent dans la littérature. En général, des estimations d'erreurs a posteriori sont utilisées pour gérer le raffinement-déraffinement ([BERGER 89]), [HAY 04]). Récemment, ont été introduits des critères de raffinement-déraffinement faisant intervenir la taille mémoire. Il s'agit de trouver le meilleur compromis entre la précision souhaitée et le coût maximum accepté (taille mémoire). Ce type de critère est utilisé entre autre par [BELLENGER 05].

On a décidé d'utiliser ici des critères de raffinement-déraffinement physiques, i. e., basés sur les caractéristiques des fluides et des écoulements étudiés (se référer à [VINCENT 00]). Vu qu'on s'intéresse soit au suivi d'interfaces dans un écoulement diphasique, soit au suivi d'une espèce inerte dans un écoulement monophasique, ces critères sont basés sur la fonction de phase  $C$  (cas diphasique) et sur la concentration de l'espèce  $S_p$  (cas monophasique). On note  $\Phi$  l'inconnue représentant  $C$  ou  $S_p$ .

Plusieurs critères de raffinement sont mis en place :

- Le premier, appelé *criter<sub>seuil</sub>* est basé sur les valeurs de  $\Phi$ . Il regroupe un seuil minimum *seuil<sub>min</sub>* et un seuil maximum *seuil<sub>max</sub>*. Si un nœud  $P$  de niveau  $G_L$  a une valeur  $\Phi(P) \in [\textit{seuil}_{min}, \textit{seuil}_{max}]$  alors le nœud  $P$  génère une cellule *AMR* de niveau  $G_{L+1}$ .
- On définit un deuxième critère de raffinement *criter<sub>grad</sub>* : pour chaque nœud  $P$  de niveau  $G_L$ , on calcule les variations de  $\Phi$  relativement à la distance dans quatre

directions d'espace (horizontale, verticale, première et deuxième bissectrice).

Conformément à la figure 2.19, on calcule les variations :

$$\left| \frac{\Phi(S) - \Phi(P)}{dist(S - P)} \right|, \left| \frac{\Phi(P) - \Phi(N)}{dist(P - N)} \right|, \left| \frac{\Phi(E) - \Phi(P)}{dist(E - P)} \right|, \left| \frac{\Phi(P) - \Phi(W)}{dist(P - W)} \right|,$$

$$\left| \frac{\Phi(P) - \Phi(NW)}{dist(P - NW)} \right|, \left| \frac{\Phi(SE) - \Phi(P)}{dist(SE - P)} \right|, \left| \frac{\Phi(P) - \Phi(NE)}{dist(P - NE)} \right|, \left| \frac{\Phi(SW) - \Phi(P)}{dist(SW - P)} \right|$$

On choisit un seuil minimum pour les variations de  $\Phi$  relativement à la distance,  $seuil_{grad}$ .

Si  $\text{Max}(\text{variations calculées}) \geq seuil_{grad}$ , alors le nœud  $P$  génère une cellule  $AMR$  de niveau  $G_{L+1}$ .

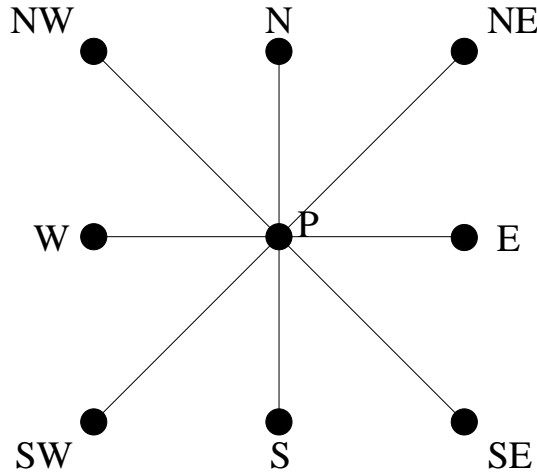


Figure 2.19 : Voisins du nœud  $P$  utilisés pour le critère de raffinement basé sur les variations de  $\Phi$  relativement à la distance.

- Le troisième critère de raffinement  $criter_{gradlarg}$  est défini de la façon suivante : l'idée consiste à utiliser le deuxième critère  $criter_{grad}$  en remplaçant les calculs des variations de  $\Phi$  relatives à la distance entre voisins directs ( $N, S, E, SE, SW, NE, NW$ ) d'un nœud  $P$  par des calculs de variations relatives aux distances entre voisins de second degré ( $SS, NN, EE, WW, WSS, ENN, WWSS, EENN, EES, WWN, WWNN, EESS, WNN, ESS$ ). La position des nœuds utilisés pour le critère est explicitée sur la figure 2.20. Donc pour chaque nœud  $P$ , on raffine si

$$\text{Max} \left( \left| \frac{\Phi(SS) - \Phi(P)}{SS - P} \right|, \left| \frac{\Phi(P) - \Phi(NN)}{P - NN} \right|, \left| \frac{\Phi(EE) - \Phi(P)}{EE - P} \right|, \left| \frac{\Phi(P) - \Phi(WW)}{P - WW} \right|, \right.$$

$$\begin{aligned}
 & \left| \frac{\Phi(EEN) - \Phi(P)}{EEN - P} \right|, \left| \frac{\Phi(P) - \Phi(WWS)}{P - WWS} \right|, \left| \frac{\Phi(EENN) - \Phi(P)}{EENN - P} \right|, \\
 & \left| \frac{\Phi(P) - \Phi(WWSS)}{P - WWSS} \right|, \left| \frac{\Phi(EEN) - \Phi(P)}{EEN - P} \right|, \left| \frac{\Phi(P) - \Phi(WSS)}{P - WSS} \right|, \\
 & \left| \frac{\Phi(ESS) - \Phi(P)}{ESS - P} \right|, \left| \frac{\Phi(P) - \Phi(WNN)}{P - WNN} \right|, \left| \frac{\Phi(EESS) - \Phi(P)}{EESS - P} \right|, \\
 & \left| \frac{\Phi(P) - \Phi(WWNN)}{P - WWNN} \right|, \left| \frac{\Phi(EES) - \Phi(P)}{EES - P} \right|, \left| \frac{\Phi(P) - \Phi(WWN)}{P - WWN} \right| \geq \text{seuil}_{grad}
 \end{aligned}$$

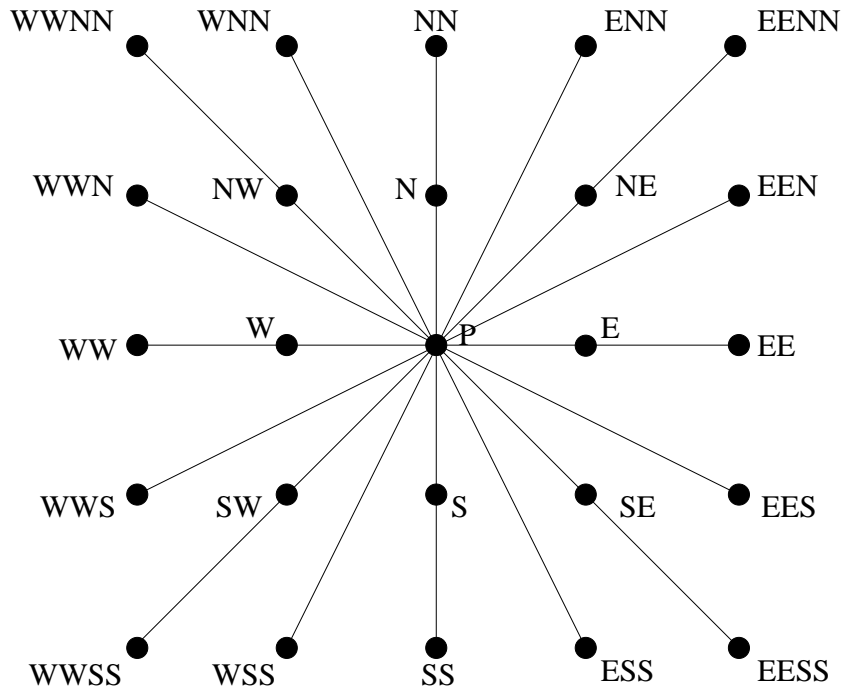


Figure 2.20 : Voisins du nœud P utilisés pour le critère de raffinement  $criter_{gradlarg}$  basé sur les variations de  $\Phi$  relativement à la distance.

Le premier critère, basé sur la valeur de  $\Phi$ ,  $criter_{seuil}$  est adapté pour des profils de concentration dont la variation locale de la valeur est faible. Supposons qu'on utilise ce critère pour suivre un profil de concentration faiblement ou pas diffusif (ou une interface en diphasique), c'est-à-dire un front très raide.

- Soit le maillage est assez fin, de façon à ce que la largeur du front  $L_{int}$  vérifie  $L_{int} > \sqrt{ndim} * \text{Min}(\Delta x, \Delta z, \Delta y)$ , où  $ndim$  est la dimension de l'espace,  $\Delta x$  (respectivement  $\Delta z, \Delta y$ ) le pas d'espace suivant  $x$  (respectivement  $z, y$ ). Dans ce cas, les nœuds disposés sur le front ont des valeurs qui varient faiblement localement.

Il y a donc assez de nœuds sur le front vérifiant le critère. Il y a donc création de cellules *AMR* au niveau du front. Ceci reste néanmoins insatisfaisant car le raffinement de ses extrémités est impossible du fait des seuils minimum et maximum qui ont été imposés. Les extrémités sont prises en compte si  $seuil_{min} = 0g.m^{-3}$  et  $seuil_{max} = 1g.m^{-3}$ . Ceci sous-entend la possibilité de raffiner l'ensemble du domaine. Dans ce cas l'*AMR* perd tout son intérêt car elle devient plus coûteuse que l'utilisation d'un maillage fin équivalent sans *AMR*.

- Soit le maillage est trop grossier, de telle manière que  $L_{int} < \sqrt{ndim} * Min(\Delta x, \Delta z, \Delta y)$ . Le front possède alors très peu ou pas de points, ce qui provoque un raffinement inexistant ou insuffisant au niveau de celui-ci.

Pour plus de clarté, on considère un domaine  $\Omega = [0, 1]$ , 1D et une fonction créneau  $\Phi(x) = 1$  si  $x \in [0.2, 0.4]$ ,  $\Phi(x) = 0$  sinon au temps initial.  $\Phi$  se déplace à la vitesse  $a$  vérifiant la *CFL* sur le niveau le plus fin  $a < \frac{\Delta x_{fin}}{\Delta t}$ . On utilise comme critère  $crit_{seuil}$ . On impose  $seuil_{min} = 0.0001$  et  $seuil_{max} = 1$ . La figure 2.21 représente la fonction  $\Phi$  sur un maillage grossier ( $L_{int} < \sqrt{(ndim)} * Min(\Delta x, \Delta z, \Delta y)$ ), au temps initial. On note que seuls les nœuds de couleur gris vérifient le critère. Donc seuls ceux-ci peuvent créer des cellules *AMR*, ce qui est insuffisant pour suivre le front de concentration.

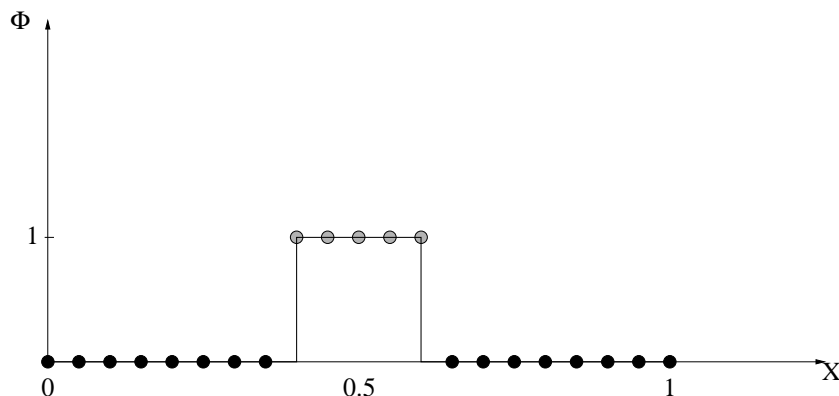


Figure 2.21 : Profil de concentration  $\Phi$  au temps initial et détection des nœuds créant une cellule *AMR*.

Étant donné le profil de concentration, il est inutile de raffiner les zones loin du front où la concentration est constante. On peut donc imposer  $seuil_{min} = 0.0001$  et  $seuil_{max} = 0.99$ . Si on se réfère toujours à la figure 2.21, on réalise qu'aucun nœud ne vérifie le critère.

Lorsque le front a légèrement diffusé, le nombre de nœuds ayant généré une cellule *AMR* reste encore insuffisant (cf figure 2.22), et ce jusqu'à ce que le maillage vérifie  $L_{int} > \sqrt{(ndim)} * \text{Min}(\Delta x, \Delta z, \Delta y)$ . La figure 2.23 montre que dans ces conditions et avec  $seuil_{min} = 0.0001g.m^{-3}$  et  $seuil_{max} = 1g.m^{-3}$ , la zone du front est raffinée sauf à ses extrémités dont la valeur est proche de zéro. Donc à moins de raffiner partout, ce critère est inadapté pour suivre des profils de concentration faiblement diffusif. Ce critère peut cependant être utile à ce type de problème s'il est nécessaire de raffiner la zone où la concentration vaut  $1g.m^{-3}$ , ou plus généralement une zone où la concentration massique a une valeur constante.

Contrairement au critère  $criter_{seuil}$ , le deuxième critère,  $criter_{grad}$ , basé sur la variation

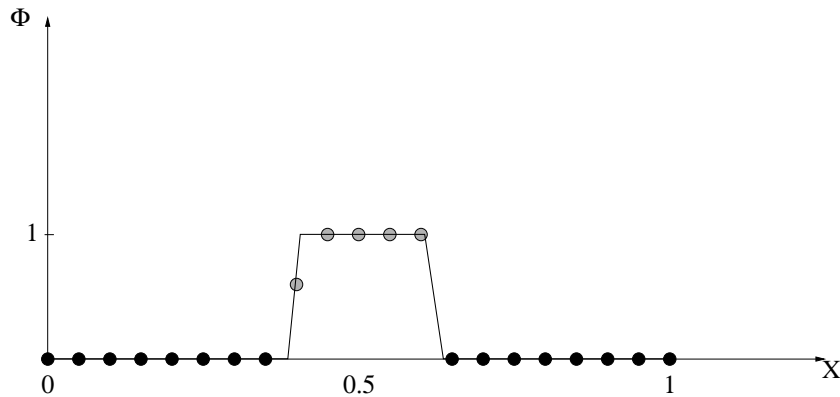


Figure 2.22 : Profil de concentration  $\Phi$  à un temps quelconque et détection des nœuds créant une cellule *AMR*.

de  $\Phi$  relativement à la distance semble mieux adapté au suivi de profil de concentration faiblement ou pas diffusif. En effet, la création de cellules *AMR* est liée à la variation locale de concentration, et ce quelque soit sa valeur au nœud  $P$  considéré. Ceci permet donc de prendre en compte la totalité du front, extrémités comprises.

Reprenons l'exemple de la fonction crête  $\Phi$  dans le domaine  $\Omega = [0, 1]$ . La figure 2.24 représente  $\Phi$  au temps initial sur un maillage quelconque. On impose une valeur de  $seuil_{gradlarg} > 0$  correctement choisie. Les nœuds coloriés en gris sont ceux qui vérifient le critère. On conclut dans ce cas que la zone du front est entièrement raffinée. Ceci montre que ce critère est tout à fait adapté à ce genre de problème.

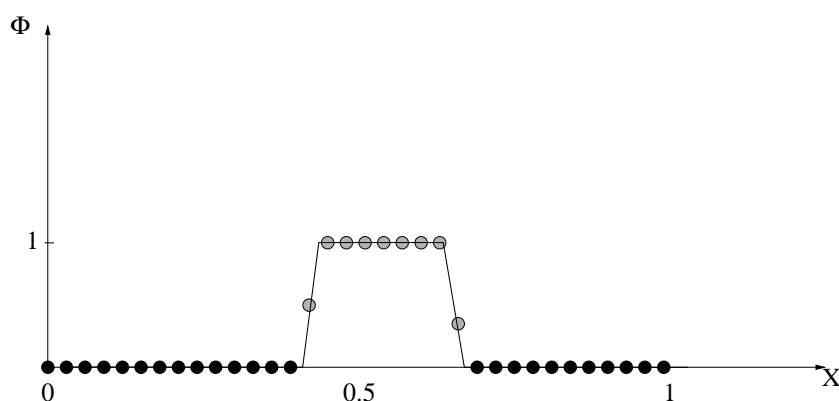


Figure 2.23 : Profil de concentration  $\Phi$  à un temps quelconque et détection des nœuds créant une cellule AMR.

On peut remarquer que dans la zone où la fonction  $\Phi$  vaut 1 uniformément, aucune cellule ne peut être créée. Donc si pour une raison ou pour une autre, il est nécessaire de raffiner cette zone, il est faut utiliser le deuxième critère combiné au premier (avec  $seuil_{max} = 1$ ).

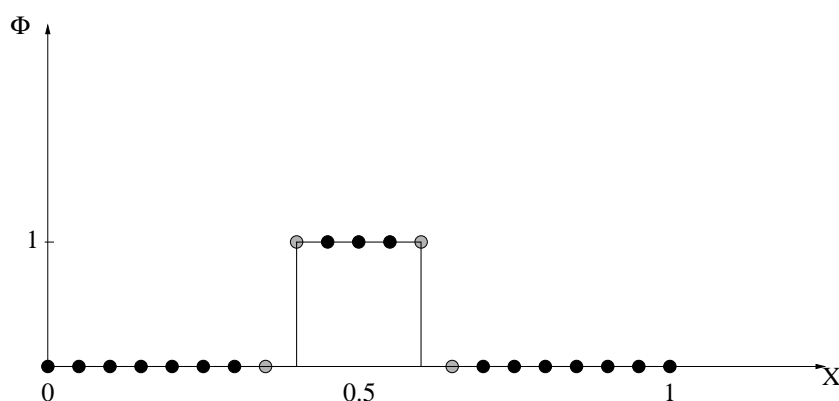


Figure 2.24 : Profil de concentration  $\Phi$  au temps initial et détection des nœuds créant une cellule AMR.

Quant au troisième critère, il fonctionne de la même façon que le deuxième. Étant donné qu'il compare des variations de  $\Phi$  relatives à la distance sur des longueurs plus grandes (voisins du second degré), le raffinement prend en compte le front dans sa totalité ainsi qu'une épaisseur de mailles fines supplémentaire.

Reprenons l'exemple de la fonction créneau  $\Phi$  dans le domaine  $\Omega = [0, 1]$ . La figure 2.25 représente  $\Phi$  au temps initial sur un maillage quelconque. On impose une valeur de



$seuil_{gradlarg} > 0$  correctement choisie. Les nœuds coloriés en gris sont ceux qui vérifient le critère. De même que pour le deuxième critère, on conclue que ce critère est tout à fait adapté au suivi de profil de  $\Phi$  faiblement ou pas diffusif.

Nous allons détailler dans ce qui suit dans quel cas il est plus intéressant de choisir le

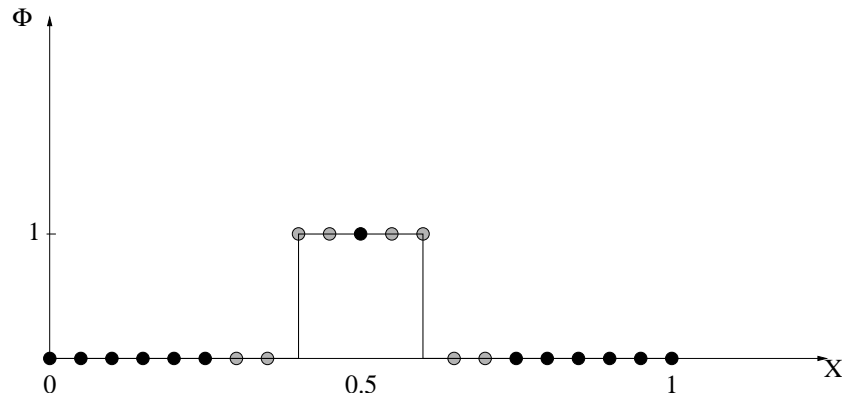


Figure 2.25 : *Profil de concentration  $\Phi$  au temps initial et détection des nœuds créant une cellule AMR.*

deuxième ou le troisième critère.

Le choix et le paramétrage des critères de raffinement ainsi que le respect de la  $CFL$  sur le niveau le plus fin, quand des schémas explicites sont utilisés ( $TVD$ ,  $WENO$ ), permettent une gestion spatiale et temporelle du raffinement satisfaisante. Un autre paramètre important est l'initialisation des cellules  $AMR$ . Celle-ci se fait à la première itération en temps et lorsque de nouvelles cellules  $AMR$  sont créées au cours du temps.

Lors de la première itération  $NT = 1$ , tous les niveaux  $AMR$  sont initialisés de la même façon que le niveau grossier, ce qui suppose qu'on connaît la forme de la solution initialement (cercle, sphere, ...).

Lorsqu'une cellule  $AMR$  est créée à une itération  $NT > 1$ , deux types d'initialisations sont possibles,  $Q_1$  et binaire. L'interpolation  $Q_1$  donne des résultats satisfaisants sur le transport d'espèce faiblement diffusive quand les nouvelles cellules  $AMR$  sont générées à plusieurs mailles du front. En effet, si un des nœuds servant à l'interpolation est trop proche du front, il apparaît une diffusion numérique supplémentaire qui dégrade la solution. Donc pour limiter la diffusion du front due à l'interpolation  $Q_1$ , il est nécessaire de raffiner sur une plus grande largeur de bande de part et d'autre du front de  $\Phi$ .

Il est donc indispensable d'utiliser le troisième critère de raffinement avec une initialisation de type  $Q_1$ .

Dans un soucis d'optimisation du temps *CPU* et de la taille mémoire, il est plus intéressant de combiner le deuxième critère de raffinement avec une interpolation binaire (voir sous section 2.3.1) car celle-ci limite fortement la diffusion numérique. Cette initialisation n'est valable que pour des fronts très raides.

En faisant une analogie avec un écoulement diphasique, l'interface peut être considérée comme un front raide. On peut alors utiliser le deuxième critère avec une interpolation binaire sur la fonction de phase pour suivre l'interface.

## 2.3 Opérateurs de prolongement

### 2.3.1 Interpolations non conservatives

On a vu dans la sous-section 1.3.2 que les nœuds de niveau  $G_L$  appartenant à l'interface  $G_L - G_{L-1}$  sont déterminés à partir d'une interpolation. Cette procédure de prolongement fait intervenir des nœuds de niveau  $G_{L-1}$  (cf sous-section B). On détaille dans cette sous-section l'emplacement des cellules de référence par rapport aux cellules *AMR* pour l'interpolation non conservative. Il se peut que certains nœuds de la cellule de référence n'existent pas à l'intérieur du domaine d'étude et qu'on soit contraint de la décaler pour que tous les nœuds de la cellule de référence appartiennent au domaine d'étude. C'est ce qu'on détaille dans cette sous-section.

Soit  $\Omega$  le domaine étudié. On note  $\dot{\Omega}$  l'intérieur de ce domaine et  $\partial\Omega$  ses frontières. On construit un premier maillage grossier  $G_0$  (resp :  $G_{V_0}$ ) sur  $\Omega$  pour les nœuds de pression (resp : de vitesses). Soit  $\dot{G}_0$  (resp :  $\dot{G}_{V_0}$ ), l'ensemble des nœuds de pression  $P_m^0$  (resp : de vitesse  $\vec{V}_m^0$ ) du maillage  $G_0$  (resp :  $G_{V_0}$ ) appartenant à  $\dot{\Omega}$ .

Soit  $\partial G_0$  (resp :  $\partial G_{V_0}$ ), l'ensemble des nœuds grossiers de pression  $P_{lim}^0$  (resp : de vitesse  $\vec{V}_{lim}^0$ ) du maillage  $G_0$  (resp :  $G_{V_0}$ ) appartenant à  $\partial\Omega$ . Les nœuds  $P_m^0$  (resp :  $\vec{V}_m^0$ ) sont traités par résolution des équations discrétisées. Quant aux nœuds  $P_{lim}^0$  (resp :  $\vec{V}_{lim}^0$ ), on leur impose des conditions aux limites de type Dirichlet ou Neumann par une méthode de pénalisation. Étant donné qu'on travaille sur un maillage décalé *MAC*, les nœuds de vitesses  $\vec{V}_{lim}^0$  sont à l'extérieur du domaine physique. Il faut donc ramener les conditions aux limites sur les bords  $\partial G_0$ .

Pour les niveaux plus fins  $G_L$  (resp :  $G_{V_l}$ ),  $l > 0$ , on distingue trois types de nœuds de pression (resp : de vitesses) : les premiers, notés  $P_m^L$  (resp :  $\vec{V}_m^L$ ), appartiennent à l'inté-

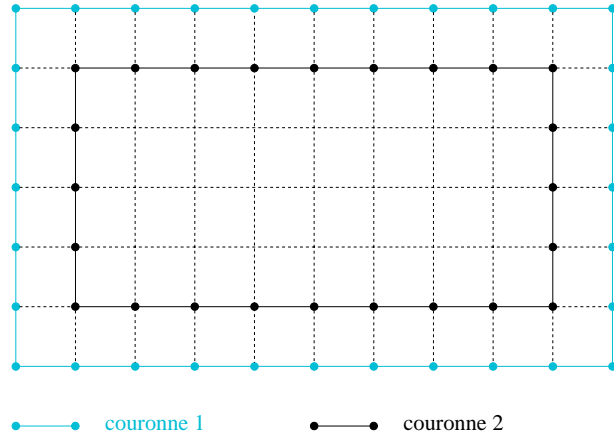


Figure 2.26 : grille des nœuds de pression

rieur du niveau  $G_L$ ; les deuxièmes, notés  $P_{i/l-1}^L$  (resp:  $\vec{V}_{i/l-1}^L$ ), appartiennent à l'interface  $G_L/G_{L-1}$ . Les troisièmes, notés  $P_{lim}^L$  (resp:  $\vec{V}_{lim}^L$ ), sont sur les limites du domaine physique. On a donné dans la section dédiée à la méthode *OCLM*, le traitement des nœuds  $P_m^L$  (resp:  $\vec{V}_m^L$ ) (résolution des équations ou interpolation) et  $P_{i/l-1}^L$  (resp:  $\vec{V}_{i/l-1}^L$ ) (interpolation). L'interpolation de ces nœuds fins de niveau  $G_L$  se fait différemment suivant la position du nœud de niveau  $G_{L-1}$ , générateur de la cellule *AMR*. Trois cas sont possibles : le nœud générateur est sur le bord du domaine  $\Omega$  (couronne 1), à une maille du domaine (couronne 2) ou à plus de deux mailles du bord du domaine. Ces cas sont représentés sur la figure 2.26. Quant aux nœuds  $P_{lim}^L$  (resp:  $\vec{V}_{lim}^L$ ), leur traitement sera expliqué dans ce qui suit.

### Nœuds générateurs à l'intérieur d'un niveau

On représente ci-après la position des nœuds appartenant à  $\Sigma$  (défini dans l'annexe B) par rapport aux nœuds d'une cellule *AMR*, en 2D puis 3D.

#### Interpolation binaire

Cette interpolation a été mise en place uniquement pour traiter les nœuds scalaires. Celle-ci s'applique plus particulièrement à la concentration massique  $S_p$  et à la fonction de phase  $C$ . Soit  $\Phi$  l'inconnue représentant  $S_p$  ou  $C$ ,  $P^0$  un nœud ayant généré une cellule *AMR*. Pour tout nœud  $P$  appartenant à cette cellule, l'interpolation binaire consiste à imposer :  $\Phi(P) = 1$  si  $\Phi(P^0) \geq 0.5$ ,  $\Phi(P) = 0$  si  $\Phi(P^0) < 0.5$ .

#### Interpolation Q1

Concernant les nœuds fins de pression, la cellule de référence pour le  $Q_1$  couvre un quart

de la cellule *AMR*. Il faut donc utiliser une cellule de référence différente pour chaque quart de cellule *AMR* (figure 2.27).

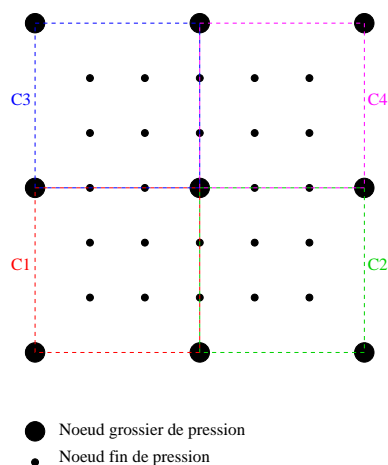


Figure 2.27 : *Interpolation  $Q_1$  sur les nœuds de pression*

Les nœuds fins de vitesses se répartissent sur six cellules de référence (voir figure 2.28). Donc suivant la position des nœuds fins, on interpolera à partir d'une des cellules de références.

Pour ce type d'interpolation, aucun traitement spécial est nécessaire.

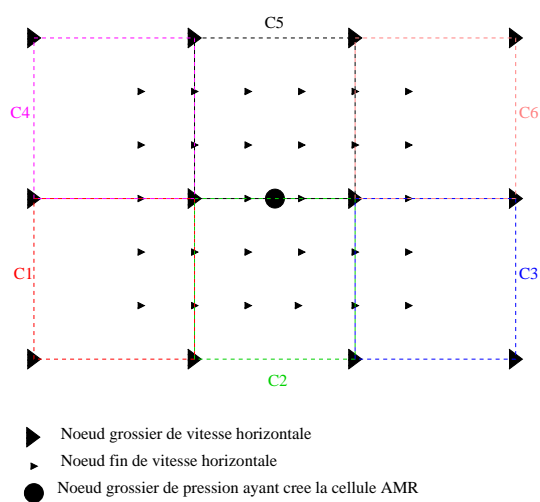


Figure 2.28 : *Interpolation  $Q_1$  sur les nœuds de vitesse horizontale*

### Interpolation $Q_2$

Pour interpoler les nœuds fins de pression, une seule cellule de référence est nécessaire (cf figure 2.29 pour le 2D et figure 2.30 pour le 3D).

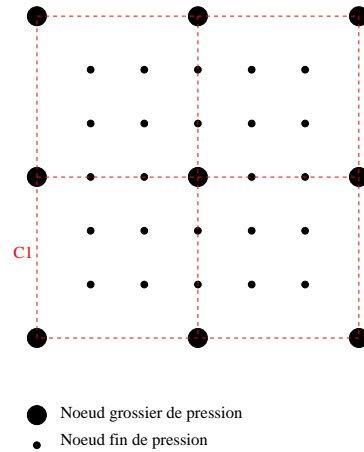


Figure 2.29 : *Interpolation  $Q_2$  sur les nœuds de pression*

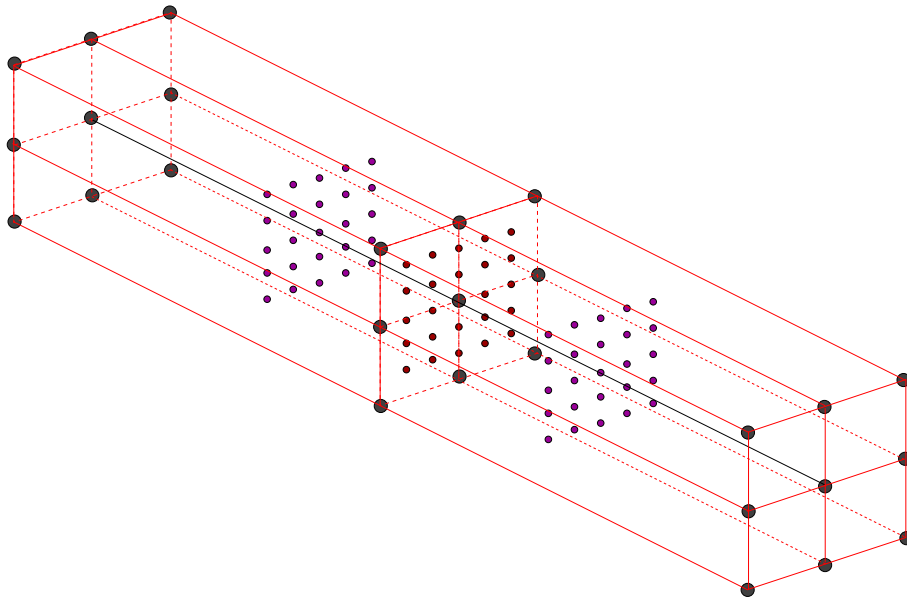


Figure 2.30 : *Interpolation  $Q_2$  sur les nœuds de pression en 3D*

Par contre, pour les nœuds fins de vitesse, une cellule de référence ne peut recouvrir la totalité de la cellule *AMR*. On a en effet besoin de 2 cellules de références (cf figure

2.31). Pour des raisons de symétrie on a choisi d'interpoler les nœuds vitesses fins en gris par la cellule de référence gauche et ceux en bleu par la cellule de référence à droite.

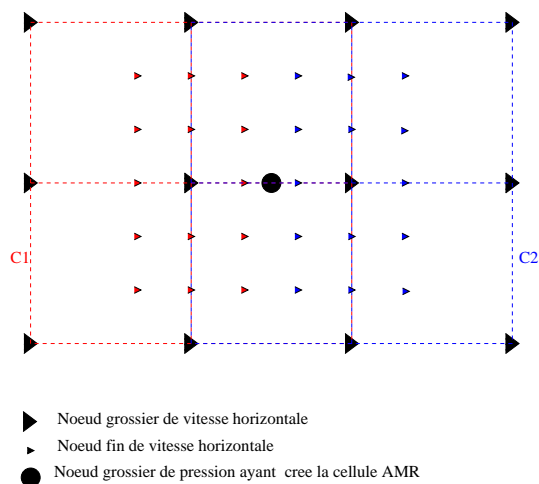


Figure 2.31 : *Interpolation  $Q_2$  sur les nœuds de vitesse horizontale*

### Interpolation $Q_3$

Pour interpoler les nœuds fins de pression comme de vitesses, un seul élément de référence est nécessaire puisque celui-ci couvre l'ensemble des nœuds à interpoler. La figure 2.32 (resp : 2.34) montre l'élément de référence pour la grille de pression en 2D (resp : 3D). Celui pour la grille de vitesse horizontale en 2D est représenté sur la figure 2.33.

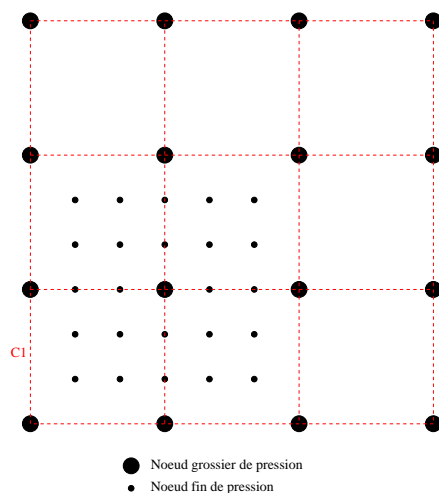
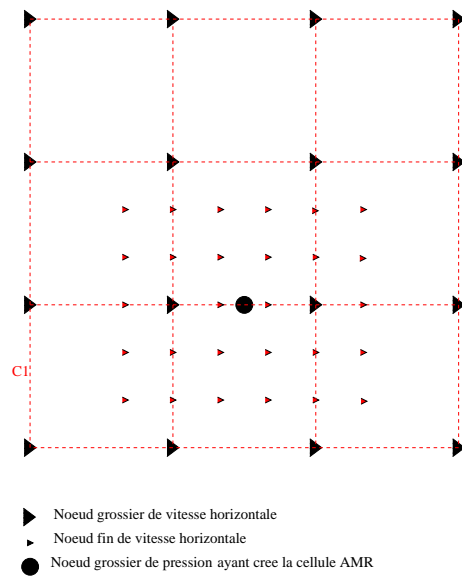
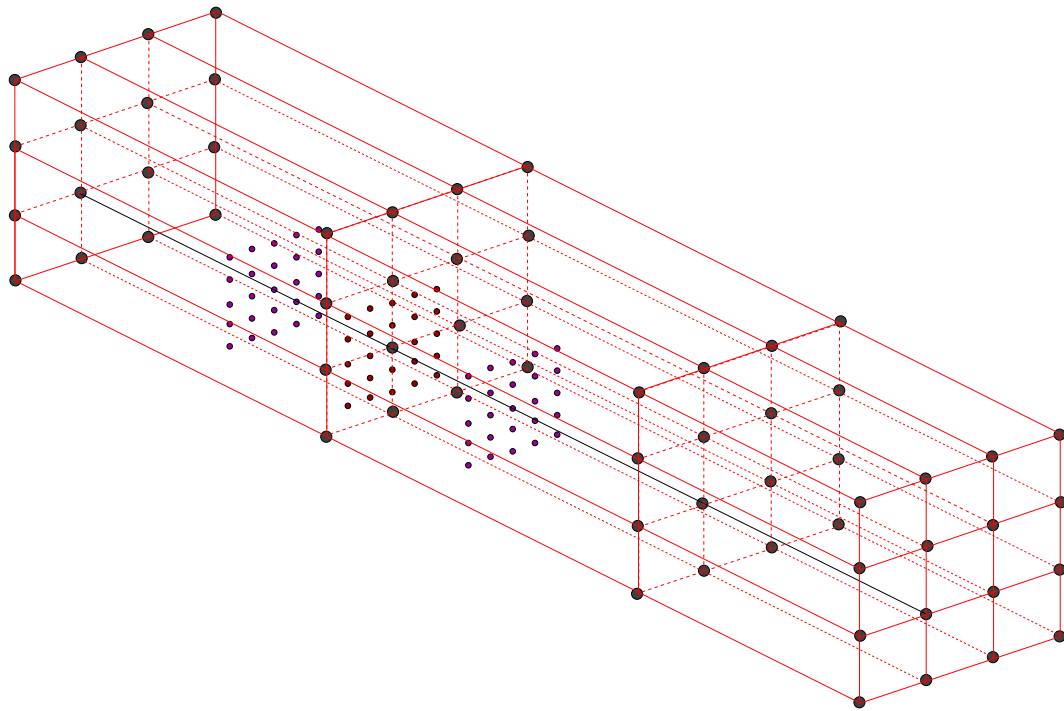


Figure 2.32 : *Interpolation  $Q_3$  sur les nœuds de pression*

Figure 2.33 : *Interpolation  $Q_3$  sur les nœuds de vitesse horizontale*Figure 2.34 : *Interpolation  $Q_3$  sur les nœuds de pression en 3D*

### Nœuds générateurs sur la couronne 2 d'un niveau

Ce qui a été présenté dans la sous section 2.3.1 reste valable tant que tous les nœuds de  $\Sigma$  sont dans le domaine  $\Omega$ . Cette sous section concerne uniquement le type d'interpolation

$Q_3$ . Les figures 2.35 et 2.36 montrent que pour ce type d'interpolation, certains nœuds de  $\Sigma$  (défini dans la section B) n'existent pas (ils sont représentés à l'extérieur du domaine  $\Omega$  sur les figures 2.35 et 2.36). Il convient alors de décentrer l'élément de référence, comme le montrent les figures 2.35 et 2.36, pour que tous les nœuds scalaires de l'élément de référence soient dans le domaine en 2D. Le raisonnement est le même pour les nœuds vitesses et en 3D.

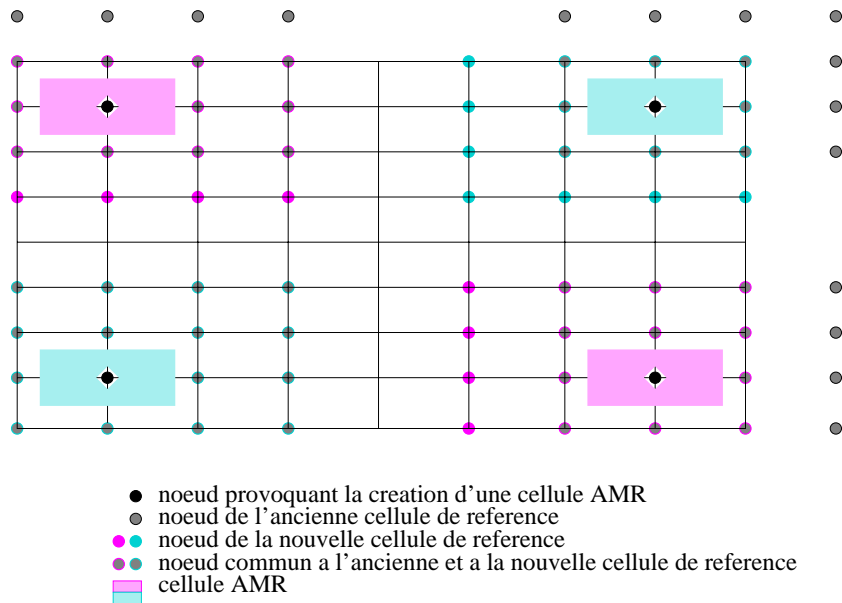


Figure 2.35 : Élément de référence pour les nœuds de pression sur les coins de la couronne 2 pour l'interpolation  $Q_3$

### Nœuds générateurs sur la couronne 1 d'un niveau

On s'intéresse dans cette section aux nœuds situés sur l'interface des niveaux  $G_L/G_{L-1}$ ,  $l \geq 1$ . Donc, seules les interpolations de type  $Q_2$  et  $Q_3$  sont concernées. Les figures 2.37 et 2.38 montrent que pour l'interpolation  $Q_3$  certains nœuds scalaires de  $\Sigma$  n'existent pas dans le domaine  $\Omega$ . Comme pour les nœuds générateurs sur la couronne 2, il convient de décentrer l'élément de référence de façon à ce que tous les nœuds de  $\Sigma$  existent dans le domaine. Il suffit de suivre la même démarche pour les nœuds de vitesses, l'interpolation  $Q_2$  et en 3D.



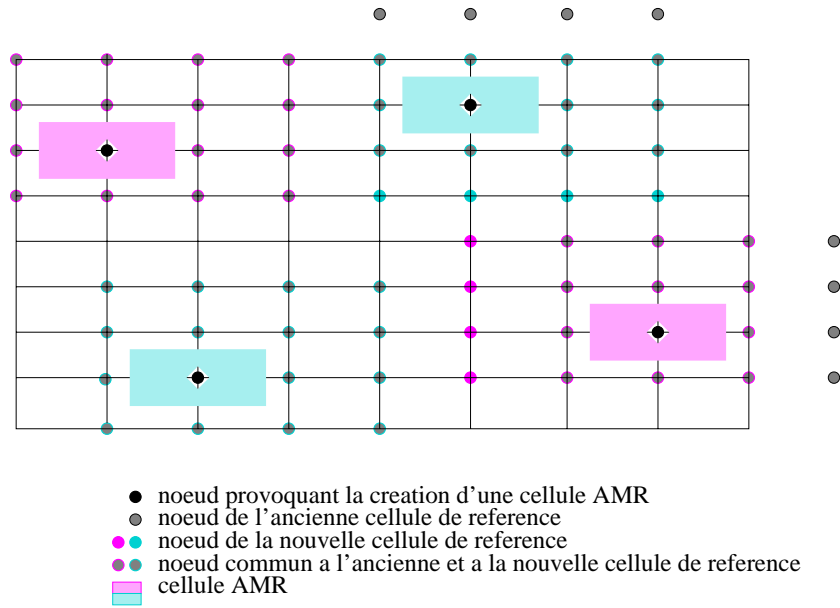


Figure 2.36 : Élément de référence pour les nœuds de pression sur le reste de la couronne 2 pour l'interpolation  $Q_3$

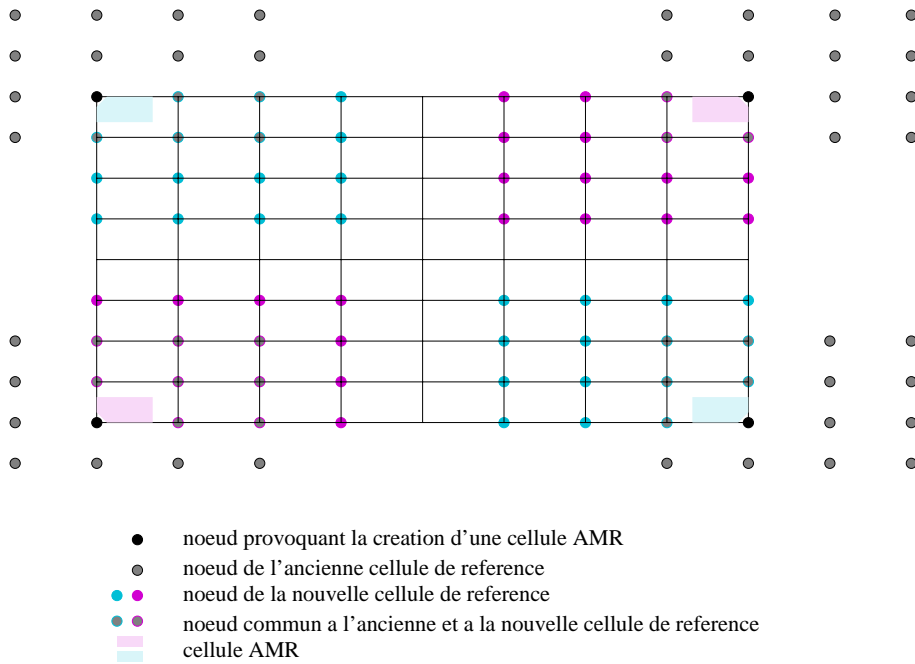


Figure 2.37 : Élément de référence pour les nœuds de pression sur les coins de la couronne 1 pour l'interpolation  $Q_3$

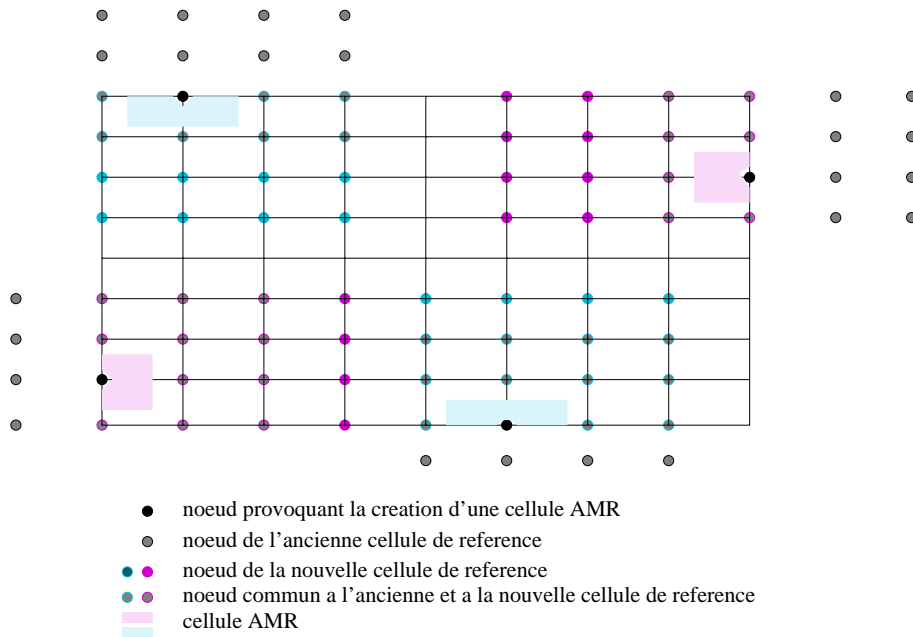


Figure 2.38 : Élément de référence pour les nœuds de pression sur le reste de la couronne 1 pour l'interpolation  $Q_3$

### 2.3.2 Interpolations conservatives pour les nœuds scalaires

La méthode *FIC* (Flux Interface Correction) a été introduite par [ANGOT 92] dans le cas d'une méthode de raffinement de maillage de type multigrille, pour corriger le flux à l'interface de deux niveaux  $G_{L-1}/G_L$ . Son principe est de satisfaire une relation de continuité des flux, au sens faible, associés à la solution entre deux niveaux  $G_{L-1}/G_L$ , sur les volumes de contrôle de niveau  $G_L$ , dont au moins une des faces appartient à l'interface  $G_{L-1}/G_L$ . Ceci permet de raccorder les sous domaines emboîtés de manière conservative. En s'inspirant de ce principe et en suivant la démarche de l'interpolation conservative décrite dans [TROTTEBERG 01], on détermine la solution sur les nœuds de niveau  $G_L$ , dont au moins une des faces appartient à l'interface  $G_{L-1}/G_L$ , en faisant un bilan de flux composites. En effet, certains flux font intervenir des nœuds de même niveau  $G_L$ , tandis que d'autres font intervenir des nœuds de niveau  $G_{L-1}$ . Ceci dépend directement de la position de chaque face d'un volume de contrôle de niveau  $G_L$ . Si cette face appartient à l'interface des deux niveaux, le flux la traversant fait intervenir des nœuds de niveaux  $G_{L-1}$ . Sinon, le flux traversant la face fait intervenir des nœuds de niveau  $G_L$ . On détaille la démarche en s'appuyant sur la figure 2.39. Dans la suite du manuscrit, on parlera de la méthode *FIC* pour désigner ce type d'interpolation conservative sur la

grille de pression.

Celle-ci représente le maillage d'un niveau  $G_{L-1}$  (en trait plein). Les volumes de contrôle

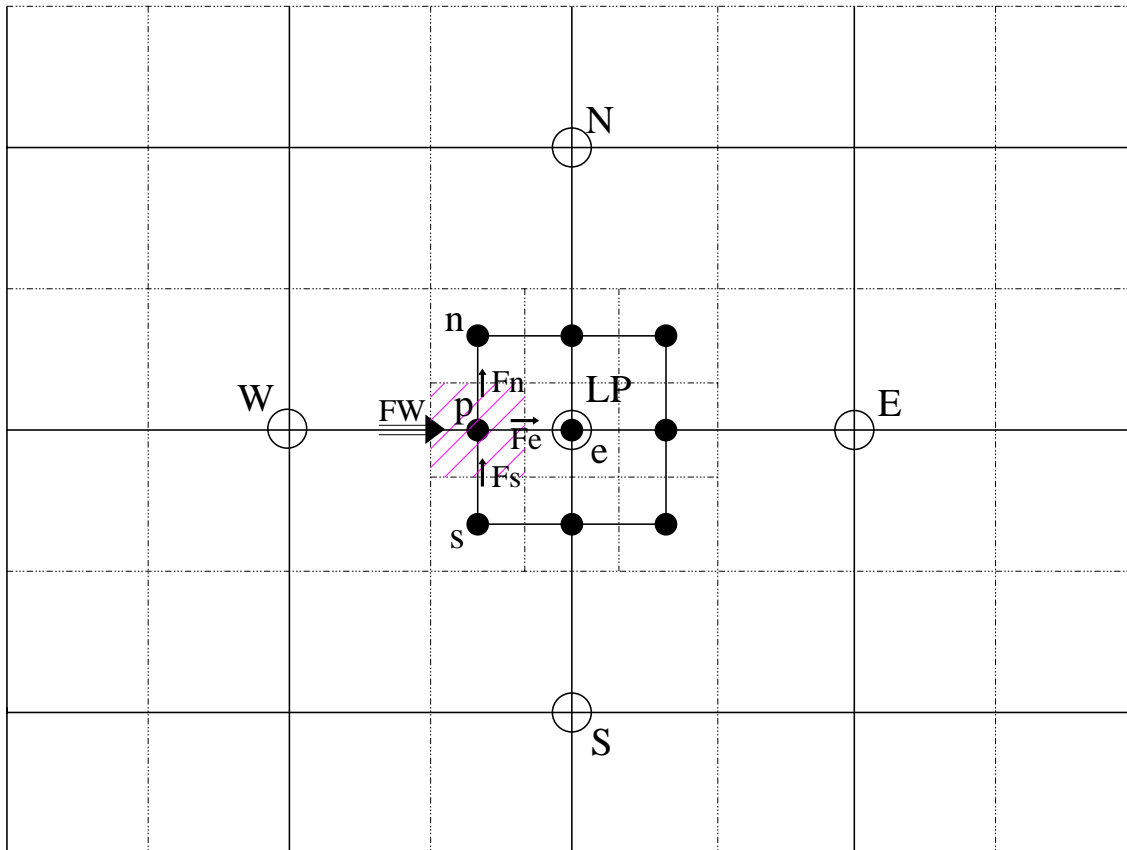


Figure 2.39 : Détermination d'un nœud scalaire à l'interface  $G_0 - G_1$  par une méthode FIC.

des nœuds de niveau  $G_{L-1}$ , centrés en chacun d'eux, sont représentés en pointillé. Une cellule AMR de niveau  $G_L$  a été créée, à partir du nœud  $LP \in G_{L-1}$ , avec un coefficient de raffinement  $RAF = 3$ . Le maillage de la cellule AMR est représenté en trait plein et les volumes de contrôle associés sont en pointillé. Les nœuds appartenant à la limite de la cellule (nœuds interpolés sur la figure 1.19) ne sont plus résolus puisque les nœuds appartenant à l'intérieur de la cellule sont résolus à partir de bilans de flux qui ne font pas intervenir les nœuds de limite de cellule. On remarque que les faces des volumes de contrôle des nœuds de niveau  $G_L$ , appartenant à l'interface des deux niveaux, sont *contigües* à celles des faces des volumes de contrôle des nœuds  $W$ ,  $E$ ,  $S$  et  $N$  de niveau

$G_{L-1}$ . On considère l'équation :

$$\frac{\partial \Phi}{\partial t} + (\vec{V} \cdot \vec{\nabla}) \Phi = \vec{\nabla} \cdot (K \vec{\nabla} \Phi) \quad (2.1)$$

La discrétisation temporelle implicite de l'équation (2.1) donne :

$$\frac{\Phi^{k+1} - \Phi^k}{\Delta t} + (\vec{V} \cdot \vec{\nabla}) \Phi^{k+1} = \vec{\nabla} \cdot (K \vec{\nabla} \Phi^{k+1}) \quad (2.2)$$

Elle s'écrit sous la forme conservative :

$$\frac{\Phi^{k+1} - \Phi^k}{\Delta t} + \vec{\nabla} \cdot (\Phi^{k+1} \vec{V}) = \vec{\nabla} \cdot (K \vec{\nabla} \Phi^{k+1}) \quad (2.3)$$

En appliquant la méthode des volumes finis sur le volume de contrôle  $\Omega_L^p$  associé au nœud  $p \in G_L$  à l'équation (2.3), on obtient :

$$\int_{\Omega_L^p} (\Phi^{k+1} - \Phi^k) d\Omega_L^p + \Delta t \int_{\Omega_L^p} \vec{\nabla} \cdot (\Phi^{k+1} \vec{V}) d\Omega_L^p - \Delta t \int_{\Omega_L^p} \vec{\nabla} \cdot (K \vec{\nabla} \Phi^{k+1}) d\Omega_L^p = 0 \quad (2.4)$$

Soit  $\Sigma_{N,L}^p$  (respectivement  $\Sigma_{S,L}^p$ ,  $\Sigma_{W,L}^p$ ,  $\Sigma_{E,L}^p$ ) la face Nord (respectivement Sud, Ouest, Est) de  $\Omega_L^p$  et  $\Sigma$  une face générique. Pour simplifier le calcul, on suppose qu'on travaille à pas d'espace constant. Ceux de niveau  $G_L$  sont notés  $\Delta x_L$  et  $\Delta z_L$  respectivement selon l'axe  $x$  et  $z$ . On note  $\vec{n}$  la normale sortante à une face  $\Sigma$  et  $(u_\Sigma^L, w_\Sigma^L)$  le vecteur vitesse au centre de la face  $\Sigma$  appartenant au niveau  $G_L$ . D'après la formule de Green, on obtient :

$$\begin{aligned} \frac{1}{\Delta t} (\Phi_p^{k+1} - \Phi_p^k) \Delta x_L \Delta z_L + \int_{\Sigma_{S,L}^p} \Phi^{k+1} \vec{V} \cdot \vec{n} d\Sigma_{S,L}^p + \int_{\Sigma_{N,L}^p} \Phi^{k+1} \vec{V} \cdot \vec{n} d\Sigma_{N,L}^p \\ + \int_{\Sigma_{W,L}^p} \Phi^{k+1} \vec{V} \cdot \vec{n} d\Sigma_{W,L}^p + \int_{\Sigma_{E,L}^p} \Phi^{k+1} \vec{V} \cdot \vec{n} d\Sigma_{E,L}^p \\ - \int_{\Sigma_{S,L}^p} K \overrightarrow{\Phi^{k+1}} \cdot \vec{n} d\Sigma_{S,L}^p - \int_{\Sigma_{N,L}^p} K \overrightarrow{\Phi^{k+1}} \cdot \vec{n} d\Sigma_{N,L}^p \\ - \int_{\Sigma_{W,L}^p} K \overrightarrow{\Phi^{k+1}} \cdot \vec{n} d\Sigma_{W,L}^p - \int_{\Sigma_{E,L}^p} K \overrightarrow{\Phi^{k+1}} \cdot \vec{n} d\Sigma_{E,L}^p = 0 \quad (2.5) \end{aligned}$$

Soit  $F_I = + \int_{\Sigma_{I,L}^p} \Phi^{k+1} \vec{V} \cdot \vec{n} d\Sigma_{I,L}^p - \int_{\Sigma_{I,L}^p} K \overrightarrow{\Phi^{k+1}} \cdot \vec{n} d\Sigma_{I,L}^p$ , le flux traversant la face  $\Sigma_I$ ,  $I \in \{N, S, E, W\}$ . L'équation (2.5) se réécrit sous la forme :

$$\frac{1}{\Delta t} (\Phi_p^{k+1} - \Phi_p^k) \Delta x_L \Delta z_L + F_E + F_W + F_N + F_S = 0 \quad (2.6)$$

- Discrétisation du flux à travers la face Ouest :  $F_W$

$$F_W = - \frac{\Phi_W^{k+1} + \Phi_{LP}^{k+1}}{2} u_{\Sigma_{W,L}^{L-1}} \frac{\Delta z_{L-1}}{3} + K \frac{\Phi_{LP}^{k+1} - \Phi_W^{k+1}}{\Delta x_{L-1}} \frac{\Delta z_{L-1}}{3}$$

- Discrétisation du flux à travers la face Est :  $F_E$

$$F_E = \frac{\Phi_e^{k+1} + \Phi_p^{k+1}}{2} w_{\Sigma_{E,L}^p}^L \Delta z_L - K \frac{\Phi_e^{k+1} - \Phi_p^{k+1}}{\Delta x_L} \Delta z_L$$

- Discrétisation du flux à travers la face Nord :  $F_N$

$$F_N = \frac{\Phi_n^{k+1} + \Phi_p^{k+1}}{2} w_{\Sigma_{N,L}^p}^L \Delta x_L - K \frac{\Phi_n^{k+1} - \Phi_p^{k+1}}{\Delta z_L} \Delta x_L$$

- Discrétisation du flux à travers la face Sud :  $F_S$

$$F_S = -\frac{\Phi_p^{k+1} + \Phi_s^{k+1}}{2} w_{\Sigma_{S,L}^p}^L \Delta x_L + K \frac{\Phi_p^{k+1} - \Phi_s^{k+1}}{\Delta z_L} \Delta x_L$$

Après avoir injecté les formes discrétisées des flux dans l'équation 2.6 et après avoir regroupé les contributions des termes  $\Phi_n^{k+1}$ ,  $\Phi_s^{k+1}$ ,  $\Phi_e^{k+1}$ ,  $\Phi_W^{k+1}$ ,  $\Phi_{LP}^{k+1}$  et  $\Phi_p^{k+1}$ , on remplit la matrice de discrétisation et le second membre. On peut ainsi déterminer  $\Phi_p^{k+1}$  à l'aide d'un solveur implicite.

## 2.4 Opérateurs de restriction

Il existe deux types d'opérateurs de restriction : l'injection conservative Full Weighted Control Volume (*FWCV*) introduite par [HACKBUSH 84] et l'injection directe.

Notons  $LP$  un nœud de niveau  $G_L$  ayant généré une cellule *AMR* (qui est constitué de  $(RAF + 2)(RAF + 2)$  fils).

- L'injection *FWCV* consiste à approcher l'inconnue  $\Phi_{L-1}$  du niveau  $G_{L-1}$  par une solution plus précise  $\Phi_L$  du niveau  $G_L$ , définie comme suit (cf aussi figure 2.40) :

$$\int_{V_{father}} \Phi_{L-1} dvol \sim \sum_{i=1}^{raf^{dim}} \int_{V_{son(i)}} \Phi_L dvol. \quad (2.7)$$

- Dans l'injection directe, on injecte au nœud  $LP$  la valeur de son fils de mêmes coordonnées.

On cherche à savoir quelle est l'injection la plus adaptée au type de problème traité. L'efficacité de l'injection *FWCV* a déjà été montrée pour les problèmes à très fort gradient (suivi de fronts ou d'interface) car elle assure la conservation de la masse. On se demande si l'injection directe pourrait être utilisable dans le cas de gradients d'intensité plus faible.

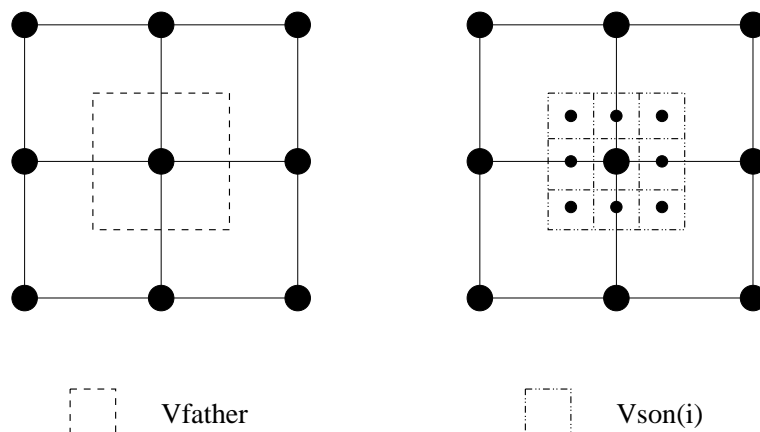


Figure 2.40 : Procédure d'injection en 2D, pour un coefficient de raffinement  $raf = 3$ .

En effet, l'injection directe assure l'obtention d'une solution précise, mais assure-t-elle la conservation de la quantité  $\int_{\Omega} \Phi d\Omega$ ?

On suppose que  $\Phi$  est connue en  $2N$  points d'un maillage (1D) de pas d'espace  $\Delta x$ . On considère un maillage grossier de pas d'espace  $2\Delta x$  contenant  $N$  points, de tel façon que chacun de ces nœuds coïncide avec un nœud du maillage fin. La solution étant connue en chaque point du maillage fin, on l'injecte sur les nœuds du maillage grossier qui coïncident avec ceux du maillage fin. Ainsi, la solution sur les nœuds du maillage grossier a la même précision que sur le maillage fin. Cependant, la solution entre deux nœuds du maillage grossier est approchée par une droite, alors que la solution analytique ne l'est pas forcément. Ainsi, l'aire sous la courbe numérique est différente de celle sous la courbe analytique. La figure 2.41 illustre ce problème sur un maillage grossier à trois points. Pour

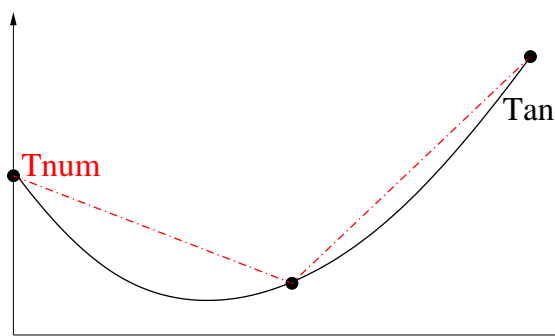


Figure 2.41 : Comparaison de deux courbes identiques en des points connus et d'aires différentes.

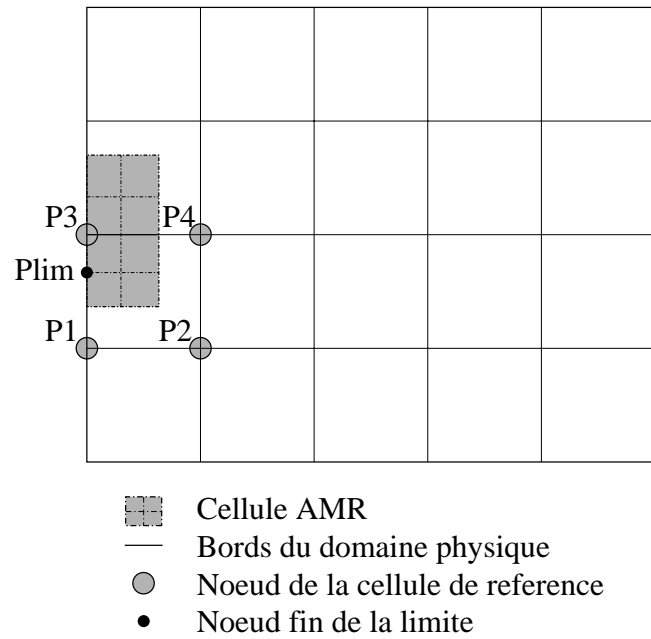
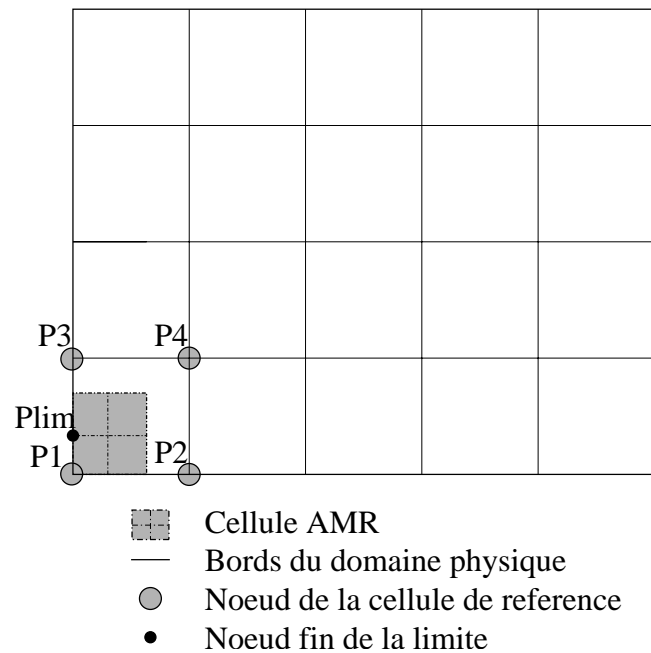
essayer de répondre à cette question, on teste les différents opérateurs de restriction sur le cas de Whitaker (cf sous-section 3.3.3).

## 2.5 Traitement des nœuds fins aux limites physiques

Soit  $N_{l,lim}$  le nombre de nœuds de niveau  $G_L$  sur les limites physiques du domaine. Lorsqu'un nœud  $P_{lim,k}^{l-1} \in G_{L-1}$ ,  $k \in [1, \dots, N_{l,lim}]$ , crée une cellule  $AMR$ , certains nœuds fins de cette cellule de niveau  $G_L$  sont à l'extérieur du domaine d'étude. Seuls les nœuds  $P_K^L = P_m^L \cup P_{lim}^L$  sont traités. Parmi les nœuds  $P_K^L$ , seuls ceux vérifiant la condition évoquée dans la sous section 2.1.1 sont susceptibles de créer une cellule  $AMR$  à leur tour. La démarche étant la même pour les nœuds de pression et ceux de vitesses, on montre le raisonnement à suivre uniquement pour les nœuds scalaires. Prenons l'ensemble des nœuds de pression  $P_{lim}^L$ , de niveau  $G_L$  appartenant à  $\partial\Omega$ . Trois méthodes ont été appliquées pour traiter ce type de nœuds.

La première consiste à interpoler  $P_{lim}^L$  à partir des nœuds de l'élément de référence de niveau  $G_{L-1}$ . Elle nécessite donc très peu de développements informatiques supplémentaires. Prenons par exemple une interpolation de type  $Q_1$ . Suivant la figure 2.42  $P_1^{l-1}$ ,  $P_2^{l-1}$ ,  $P_3^{l-1}$  et  $P_4^{l-1}$  constituent l'élément de référence.  $P_{lim}^L \in [P_1^{l-1} P_3^{l-1}]$ . Donc les pondérations associées aux nœuds  $P_1^{l-1}$  et  $P_3^{l-1}$  sont prépondérantes devant celles associées à  $P_2^{l-1}$  ou  $P_4^{l-1}$ . Ainsi cette méthode est très satisfaisante pour des conditions aux limites de type Dirichlet. Elle reste toutefois très contraignante puisqu'il faut imposer une seule valeur constante à l'ensemble des bords  $\partial\Omega$ . La création d'une discontinuité sur  $\partial\Omega$  rend la méthode défaillante. Prenons par exemple un domaine  $\Omega$  rectangulaire. On impose sur la paroi gauche une valeur  $C_0$ , et sur les autres parois une valeur  $C_1 \neq C_0$ . Supposons, comme l'indique la figure 2.43 qu'une cellule  $AMR$  ait été créée dans le coin bas gauche (i.e., au voisinage de la discontinuité), le nœud  $P_1^{l-1}$  a donc pour valeur  $C_0$  tandis que le nœud  $P_3^{l-1}$  a pour valeur imposée  $C_1$ . Le nœud  $P_{lim}^L$  appartenant à la paroi de gauche acquiert comme valeur interpolée une combinaison linéaire des nœuds  $P_1^{l-1}$  et  $P_3^{l-1}$ , ce qui biaise la valeur de la condition aux limites qu'on aurait voulu imposer à  $P_{lim}^L$  (figure 2.43).

On décide donc d'utiliser une seconde méthode moins contraignante concernant les conditions aux limites de type Dirichlet. Celle-ci est plus naturelle mais impose un développement informatique plus important que la première. Elle consiste à imposer aux

Figure 2.42 : Traitement des nœuds de pression fins appartenant aux limites de  $\Omega$ Figure 2.43 : Traitement des nœuds de pression fins appartenant aux coins limites de  $\Omega$ 

nœuds  $P_{lim}^L$  de niveau  $G_L$  les mêmes conditions aux limites que celles imposées sur le niveau  $G_{L-1}$ . Il faut tout d'abord repérer à quelle limite appartient  $P_{lim}^L$  pour imposer la



bonne condition aux limites. Pour cela, il suffit de déterminer le(s) voisin(s) manquant de son père  $P_{pre}^{L-1} \in \partial\Omega$ . La démarche est quelque peu différente suivant que le père appartienne au niveau grossier  $G_0$  ou  $G_{L-1}$ ,  $L > 1$ . Pour les nœuds  $P_{pre}^{L-1}$ ,  $L > 1$ , les nœuds à l'extérieur du domaine  $\Omega$  existent mais ne sont pas traités. Un tableau indique pour chacun d'eux s'il appartient au domaine (true) ou s'il est à l'extérieur (false). Pour les nœuds  $P_{pre}^0$ , seuls les nœuds appartenant à  $\Omega$  existent (ceci est propre au code de calcul *AQUILON*). Donc pour savoir si  $P_{pre}^0 \in \partial G_0$ , il suffit de voir s'il possède au moins deux fois le même voisin.

Cette méthode reste efficace tant que les conditions aux limites restent de type Dirichlet (valeurs imposées sur  $\partial G$ ). En effet, celle-ci montre des défaillances pour des conditions aux limites de type Neumann (ou adiabatique) i.e.,  $\frac{\partial}{\partial \mathbf{n}}$ , où  $\mathbf{n}$  est la normale sortante à  $\Omega$ . Celles-ci sont en partie dues à la structure du code. A l'intérieur du domaine  $\Omega$ , ils ont effectivement le même nombre de voisins. Ce n'est pas le cas pour les nœuds des limites  $\partial G_L, l \in [1, N]$ . La structure du code imposant que tous les nœuds aient le même nombre de voisins, on désigne les voisins manquant par une symétrie centrale par rapport au nœud étudié (appelé effet miroir). Supposons, par exemple, que le voisin Ouest n'existe pas, alors le voisin Est (existant) est désigné comme étant aussi le voisin Ouest du nœud en question.

Cette structure pose des problèmes dans la mesure où un nœud  $P_{lim}$  appartenant à la fois à la limite du domaine physique et à l'interface  $G_L/G_{L-1}$  est discrétisé et non interpolé. Du fait des effets miroirs, les voisins de  $P_{lim}$  intervenant dans la discrétisation sont en partie des voisins désignés. Ceci entraîne une mauvaise estimation de la valeur en  $P_{lim}$ . Imaginons un nœud  $P_{lim}$  sur la limite gauche de  $\Omega$  et à l'interface  $G_1/G_0$ . Supposons que son voisin Sud n'existe pas. On attribue alors au voisin Nord, le voisin Sud. On fait de même avec le voisin Ouest. Si le flux à travers la limite gauche est constant le long de cette limite, on obtient une bonne estimation de la valeur en  $P_{lim}$ . Sinon, elle est erronée car la valeur prise par le voisin Sud n'est autre que celle prise par le voisin Nord. Elle est donc elle-même erronée. Donc la valeur calculée en  $P_{lim}$  par discrétisation devient fausse. (figure à faire)

On décide donc d'utiliser une troisième méthode qui est une combinaison entre la première et la deuxième. Il s'agit d'interpoler les nœuds  $P_{lim}$  appartenant à la fois à la limite du domaine physique et à l'interface  $G_L/G_{L-1}$  et d'appliquer la deuxième méthode aux autres.

## 2.6 Contribution de l'*AMR* dans un solveur implicite

On a montré dans la sous section 1.2.1 la nécessité d'utiliser un solveur implicite en fonction des paramètres des différentes équations à résoudre et de la régularité des solutions. On rappelle que lorsqu'on choisit une méthode itérative implicite, les équations discrétisées peuvent se mettre sous la forme d'un système linéaire à résoudre :  $A^{n+1}\Phi^{n+1} = B^n$  où  $B^n$  est le second membre à l'itération  $n$ ,  $\Phi^{n+1}$  l'inconnue à l'itération  $n+1$  et  $A^{n+1}$  la matrice à inverser à l'itération  $n+1$ .

D'autre part, on a évoqué dans la sous section 1.3.2 les limites de la méthode *OCLM*, liées au découplage de la procédure d'interpolation et du solveur implicite.

Pour contrer cette difficulté, on procède à l'implication de l'interpolation. Cette méthode consiste à insérer les contributions de cette dernière dans la matrice  $A^{n+1}$  à inverser.

Lorsque la méthode *AMR* est utilisée, l'inconnue  $X$  rassemble les nœuds des différents niveaux  $G_0$  à  $G_{N_{max}}$ , avec  $N_{max}$  le nombre de niveaux *AMR*. Soit  $KKT_0$  le nombre de nœuds sur le maillage grossier  $G_0$ ,  $KKT_j$ , le nombre de nœuds sur le niveau  $G_j$  et  $KKT$  le nombre de nœuds total.  $\Phi$  se décompose de la façon suivante :  $\Phi = (\dots X_j^{niv})^t$ ,  $j \in [1, KKT_{niv}]$ ,  $niv \in [1, N_{max}]$ . La matrice  $A$  d'élément  $(A_{j,i})$ ,  $j \in [1, KKT]$ ,  $i \in [1, KKT]$ , est construite en conséquence.

Il faut ensuite s'occuper des lignes de la matrice correspondant aux nœuds à interpoler. D'après la section B, l'inconnue est approchée par un polynôme d'interpolation. Si  $X_j^{niv}$ ,  $j \in [1, KKT_{niv}]$ , représente  $X^{niv}$  au nœud  $a_j$ ,

$$X_j^{niv} = \sum_{i=1}^N X_i^{niv-1} f_i(a_j).$$

$N$  est le nombre de nœuds nécessaires à l'interpolation. Soit  $f_i(a_j)$ ,  $(i,j) \in [1, N]$ , les nouveaux coefficients  $(A_{j,i})$  de la matrice  $A$  là où l'interpolation remplace la discrétisation des équations. Notons  $AA_{j,i}$  ces éléments. On obtient alors

$$X_j^{niv} - \sum_{i=1, i \neq j}^N X_i^{niv-1} AA_{j,i} \sim 0 \quad (2.8)$$

On note  $BB$  le nouveau second membre correspondant à  $B$  auquel on a remplacé les lignes correspondant à des nœuds interpolés par 0. Il vient alors :

$$\sum_{i=1}^N AA_{j,i} X_i^{niv-1} = BB_j \quad (2.9)$$

On note  $AA^{n+1}$  la nouvelle matrice à inverser,  $BB^n$  le nouveau second membre, chacun contenant la contribution de l'interpolation et celle des équations discrétisées. le nouveau système à résoudre est donc  $AA^{n+1}X^{n+1} = BB^n$ .

On a représenté sur la figure 2.44 un exemple de système initial à résoudre, i.e. avant l'implicitation de l'interpolation. Les sous matrices  $A_0$ ,  $A_1$ ,  $A_2$  et  $A_3$  représentent respectivement les coefficients de la matrice  $A$  sur les niveaux  $G_0$ ,  $G_1$ ,  $G_2$  et  $G_3$ . En réalité, la matrice est assemblée dans l'ordre de création des cellules  $AMR$ , quelquesoit leur niveau  $AMR$  d'appartenance. Les mêmes notations sont adoptées pour l'inconnue  $X$  et le second membre  $B$ . La figure 2.45 illustre de manière simplifiée la démarche suivie pour implicititer l'interpolation.

Partant de la matrice  $A$  représentée sur la figure 2.44, on remplace les lignes de chaque niveau correspondant à des nœuds à interpoler par les coefficients d'interpolation. Ceux-ci sont représentés dans la nouvelle matrice de la figure 2.45 par des lignes noires.

On remarque aussi que pour chaque niveau  $G_L$ , la matrice initiale  $A$  fait intervenir uniquement des nœuds de même niveau. Par contre, dans la matrice finale  $AA$ , les coefficients d'interpolation d'un nœud de niveau  $G_L$  font intervenir des nœuds de niveau  $G_{L-1}$ . Cela se traduit par un prolongement des lignes noires d'un niveau  $G_L$  sur le niveau inférieur  $G_{L-1}$ . La contribution de l'interpolation dans le second membre se traduit aussi par l'insertion de lignes noires. Remarquons que ces termes sont nuls (cf équation 2.8).

Cette démarche permet donc de calculer en une seule étape les nœuds déterminés à partir des coefficients provenant de la discrétisation des équations et ceux provenant de l'interpolation. On lève ainsi un des problèmes rencontrés dans la méthode *OCLM*.

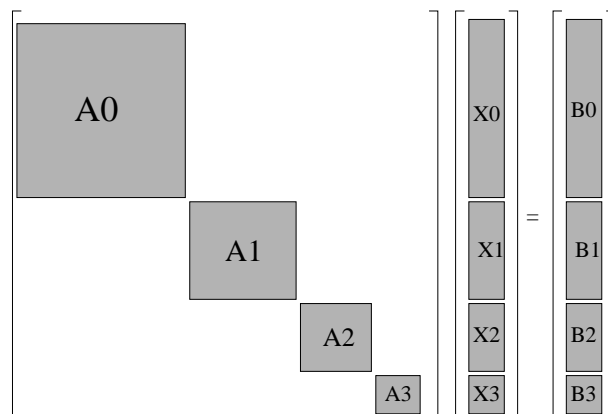
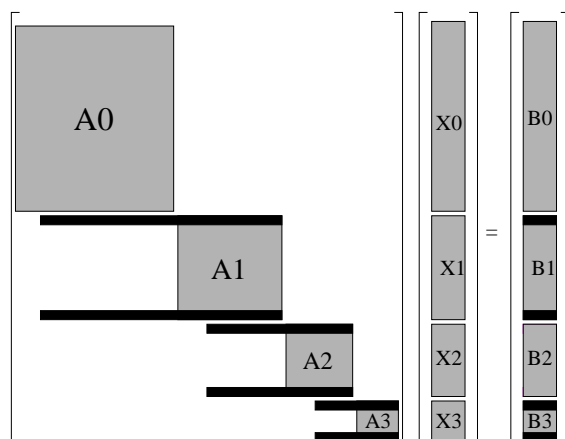


Figure 2.44 : *Système initial à résoudre.*

Figure 2.45 : *Système final à résoudre.*

## 2.7 *AMR* et lagrangien augmenté

On a expliqué dans la sous-section 1.2.3 que le découplage de la vitesse-pression pour résoudre les équations de Navier-Stokes se fait par l'utilisation d'une méthode de minimisation sous la contrainte d'incompressibilité, appelée lagrangien augmenté. On explique ici l'adaptation de cette méthode à l'approche *AMR*.

L'implication de la méthode *AMR* nécessite, comme on l'a détaillé dans la section 2.6, un travail sur la matrice de discrétisation. En particulier, les coefficients de discrétisation correspondant aux nœuds vitesses appartenant à l'interface  $G_L - G_{L-1}$  sont remplacés par les coefficients d'interpolation. Sachant qu'on travaille sur un maillage décalé, la divergence du champ de vitesses est calculée sur les nœuds de la grille de pression, centrés sur les quatre nœuds vitesses (intervenant dans le calcul de la divergence) qui les entourent. Seuls les nœuds appartenant à l'intérieur (strict) des cellules sont résolus par la méthode du lagrangien augmenté (cf figure 1.19). Les autres nœuds sont interpolés. On rappelle que l'interpolation consiste à approcher les vitesses  $\vec{V}_L \in G_L$  par des polynômes faisant intervenir des nœuds vitesses  $\vec{V}_{L-1} \in G_{L-1}$ , ce qui revient en quelque sorte à imposer la vitesse sur ces nœuds de niveau  $G_L$ . Cependant certains nœuds de vitesse ont au moins un voisin interpolé intervenant dans l'équation de Navier-Stokes discrétisée. Ceux-ci sont donc fortement influencés par l'interpolation. Cette dernière n'étant pas à divergence nulle, la minimisation des nœuds intérieurs (strictement) à la cellule *AMR* s'effectue correctement jusqu'à un certain seuil (à cause de l'interpolation qui impose une vitesse et ne vérifie pas la contrainte d'incompressibilité).

## Chapitre 3

# Résultats sur les équations scalaires

On cherche à montrer l'efficacité de la méthode *AMR* à réduire la diffusion numérique pour pouvoir suivre des interfaces (dans les écoulements diphasiques) et les fronts de concentration (dans les écoulements monophasiques) et à optimiser les coûts en temps CPU et espace mémoire.

### 3.1 Etude d'une tache cisailée par un tourbillon dans une cavité $2D$ en monophasique

On évalue les performances de l'*AMR* sur la résolution de l'équation de transport d'espèces en deux dimensions d'espace :

$$\frac{\partial S_p}{\partial t} + \vec{V} \cdot \vec{\nabla}(S_p) = \vec{\nabla} \cdot (D \vec{\nabla}(S_p))$$

avec  $S_p$  la concentration massique,  $\vec{V}(x, z) = (u(x, z), w(x, z))$  le champ de vitesse et  $D$  le coefficient de diffusion moléculaire. On considère un domaine carré d'arête  $0.1m$  dans lequel on place un disque de concentration massique  $1g.m^{-3}$  submergé dans un fluide de concentration massique nulle (figure 3.1 gauche).

On impose un champ de vitesse tournant d'intensité moyenne  $V_{moyen} = 0.1m.s^{-1}$  et de forme analytique (voir aussi figure 3.1 droite):

$$\begin{aligned} u(x, z) &= -V_{moyen} \cos\left(\frac{\pi(x - x_0)}{L}\right) \sin\left(\frac{\pi(z - z_0)}{L}\right) \\ w(x, z) &= V_{moyen} \sin\left(\frac{\pi(x - x_0)}{L}\right) \cos\left(\frac{\pi(z - z_0)}{L}\right) \end{aligned} \quad (3.1)$$

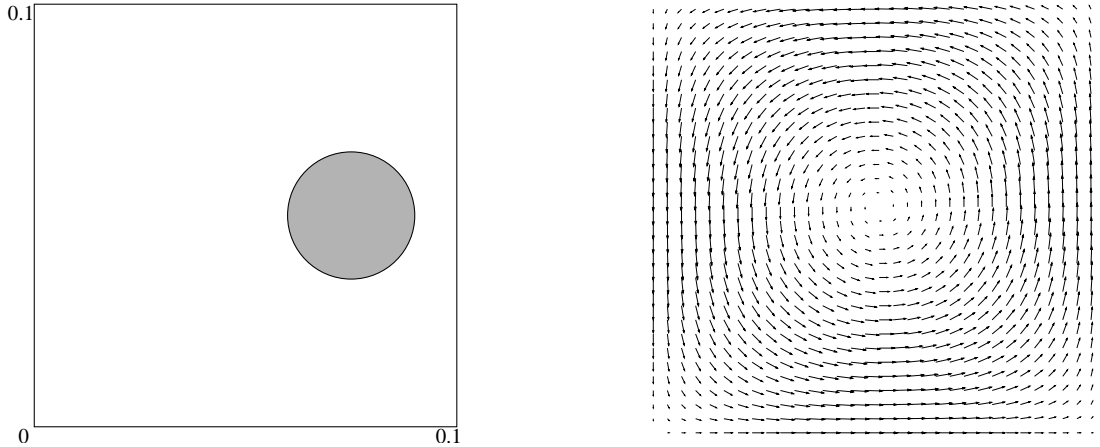


Figure 3.1 : Répartition initiale de la concentration massique (gauche). Champ de vitesses imposé (droite).

$(x_0 = 0.05m, z_0 = 0.05m)$  sont les coordonnées du centre du domaine. On précise que le disque a initialement un rayon  $R$  de  $0.015m$  et un centre  $(x = 0.075m, z = 0.05m)$ . Dans cette section, la partie diffusion de l'équation de transport est traitée de manière implicite.

### 3.1.1 Influence du type d'interpolation appliqué à la concentration massique

On cherche à savoir quel effet peut avoir l'interpolation sur la qualité de la solution. On teste alors différents types d'interpolations  $Q_1$ ,  $Q_2$  et  $Q_3$  sur la concentration massique. On calcule son évolution sur un maillage grossier  $G_0 = 16 * 16$  avec deux niveaux de raffinement, soit un maillage fin équivalent sans *AMR*,  $G = 144 * 144$ . On choisit un pas de temps égal à  $\Delta t = 0.001s$ , une durée de simulation de  $0.1s$  et un coefficient de diffusion  $D = 10^{-8}m^2.s^{-1}$ . La vitesse analytique est imposée sur les nœuds vitesses des niveaux  $G_0$  à  $G_2$ . On choisit comme critère de raffinement-déraffinement un critère basé sur le gradient de concentration  $S_p$ .

On observe sur la figure 3.2 la formation d'oscillations qui se développent. Ce comportement est semblable à celui connu sous le nom de phénomène de *Gibbs*. Ces oscillations ont une amplitude d'autant plus forte que le degré du polynôme d'interpolation est élevé. En effet, pour l'interpolation  $Q_2$ ,  $-0.06g.m^{-3} \leq S_p \leq 1.003g.m^{-3}$  tandis que pour l'interpolation  $Q_3$ ,  $-0.107g.m^{-3} \leq S_p \leq 1.006g.m^{-3}$ . On constate que l'interpolation  $Q_1$  n'introduit

pas d'oscillations ( $\min(S_p) = 0g.m^{-3}$  et  $\max(S_p) = 1g.m^{-3}$ ). On choisit d'utiliser des polynômes d'interpolation de plus petit degré possible lorsque les solutions sont discontinues.

On vient de montrer que l'interpolation  $Q_1$  permet d'éviter les oscillations. Considé-

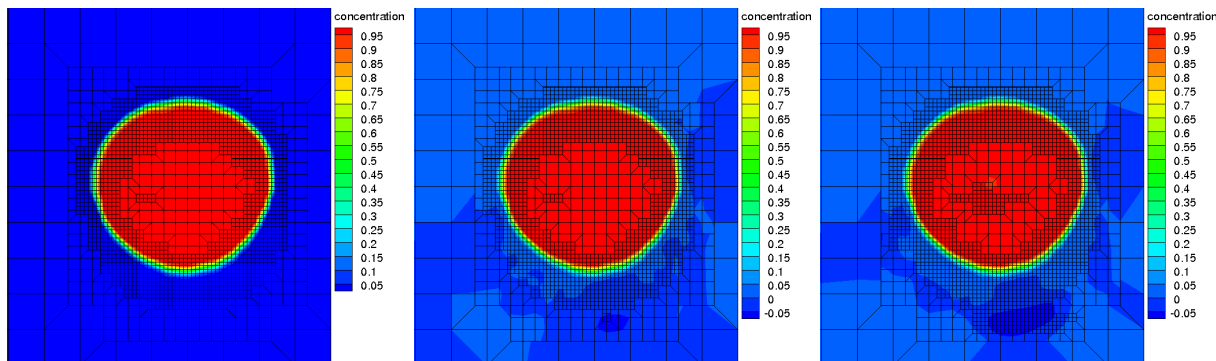


Figure 3.2 : Concentration massique interpolée sur les limites des niveaux de raffinement par une interpolation  $Q_1$  (gauche),  $Q_2$  (centre) et  $Q_3$  (droite) à  $t = 0.1s$ . Vitesse analytique imposée sur tous les nœuds vitesses.

rons un nœud  $LP_{lim}$  de limite de niveau fin,  $G_1$ , devant être interpolé. Si les nœuds  $(a_1, a_2, a_3, a_4) \in G_0$  intervenant dans l'interpolation  $Q_1$  (cf sous-section 2.3.1) ont tous une valeur nulle, il en sera de même pour le nœud  $LP_{lim}$ . Si il existe au moins un nœud  $a_i$ ,  $i \in [1, 4]$  tel que  $S_p(a_i) > 0$  (i.e. le nœud  $LP$  est trop proche du front), alors  $S_p(LP_{lim}) > 0$ . Or sur un maillage fin équivalent, sans  $AMR$ , le nœud correspondant à  $LP_{lim}$  a une valeur nulle. Donc l'interpolation  $Q_1$  peut introduire une légère diffusion s'il existe au moins un nœud  $LP_{lim}$  trop proche du front. La figure 3.3 compare l'iso-concentration  $S_p = 0.5g.m^{-3}$  obtenue en interpolant la concentration massique avec une interpolation  $Q_1$  sur les limites des niveaux de raffinement à celle obtenue avec une interpolation binaire de la concentration massique sur les limites des niveaux de raffinement. Dans les deux cas, un critère de gradient de concentration a été choisi. L'iso-concentration avec  $AMR$  et interpolation binaire, est confondue avec celle sans  $AMR$ . Par contre, on observe une dégradation de cette iso-concentration lorsque l'interpolation  $Q_1$  est utilisée (voir tête et queue de la tache de la figure 3.3). Le maillage correspondant est représenté sur la figure 3.4.

Pour palier à ce problème, on peut élargir le niveau  $G_1$ , de façon à ce que les nœuds à interpoler soient assez éloignés du front de concentration massique. Étant donné qu'on cherche à optimiser le temps de calcul et la taille mémoire, la proposition qui vient d'être faite ne semble pas être un bon compromis.

Pour éviter la diffusion introduite par l'interpolation  $Q_1$  et pour minimiser le nombre de nœuds à résoudre, une alternative consiste à interpoler les nœuds  $LP_{lim}$  de limite de niveau fin avec une interpolation binaire.

Notons  $LPG0 \in G_0$ , le père de  $LP \in G_1$ ,  $S_{pmax} = \max_{x \in G_0}(S_p(x))$  et  $S_{pmin} = \min_{x \in G_0}(S_p(x))$ . Cette stratégie consiste à imposer  $S_p(LP_{lim}) = S_{pmax}$  si  $S_p(LPG0) \geq 0.5(S_{pmin} + S_{pmax})$ , et  $S_p(LP_{lim}) = S_{pmin}$  si  $S_p(LPG0) < 0.5(S_{pmin} + S_{pmax})$ .

On va donc privilégier une interpolation binaire de la concentration massique lorsque l'espèce est faiblement diffusive.

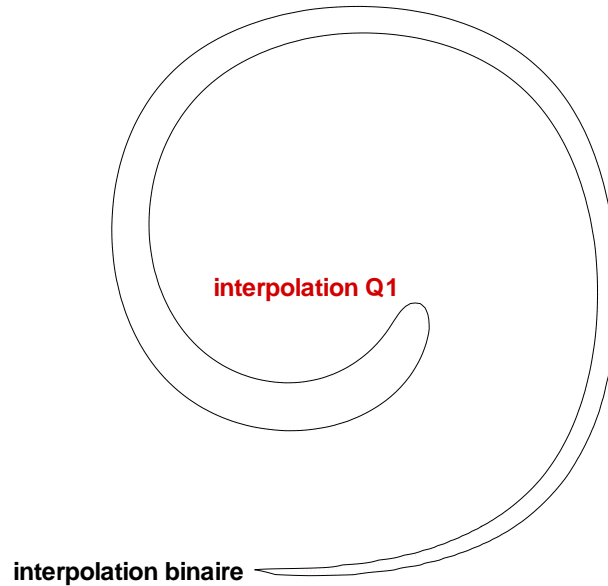


Figure 3.3 : Isovaleur  $S_p = 0.5g.m-3$  obtenue par interpolation  $Q_1$  (en rouge) et binaire (en noir) de  $S_p$ , sur  $G_2$ .  $t = 8s$ ,  $G_0 = 20 * 20$ , vitesse analytique imposée sur tous les nœuds vitesses.



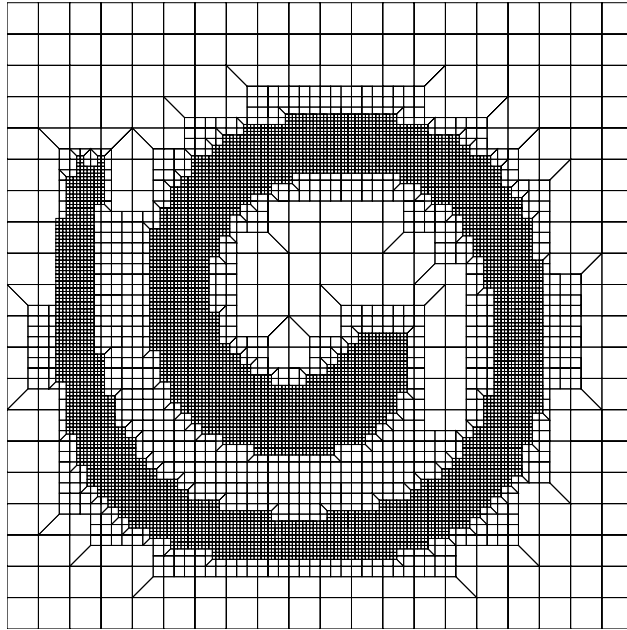


Figure 3.4 : Maillage AMR utilisé pour  $S_p$ .  $t = 8s$ ,  $G_0 = 20 * 20$ .

### 3.1.2 Capacité de l'AMR à réduire la diffusion numérique.

L'objectif de cette sous section est de montrer l'efficacité de la méthode pour réduire la diffusion numérique.

On considère un maillage grossier  $G_0 = 16 * 16$  avec deux niveaux de raffinement, un pas de temps  $\Delta t = 0.001s$ , un temps total  $t = 2s$  et un coefficient de diffusion  $D = 10^{-8}m^2.s^{-1}$ . On résout l'équation de transport sur les nœuds du niveau  $G_0$  et sur ceux à l'intérieur des autres niveaux de raffinement  $G_1$  et  $G_2$ . Les nœuds aux limites des deux niveaux de raffinement sont interpolés en binaire. Le champ de vitesse analytique est imposé sur  $G_0$  et sur la totalité des niveaux fins.

On a représenté sur la figure 3.5, différentes masses du disque cisillé pour estimer la diffusion numérique ( $M_1, M_2, M_3$ ) mais aussi vérifier la conservation de la masse ( $M_0, M$ ).  $M_0$  est la masse théorique calculée à partir du volume ( $\Pi R^2$ ) et de la concentration massique initiale  $1g.m^{-3}$ , soit  $M_0 = 7.07 \cdot 10^{-4}g$ .  $M$  est la masse numérique calculée à partir de l'équation discrétisée.  $M_1$  (resp  $M_2, M_3$ ) représente la masse pour laquelle la concentration massique est supérieure à  $0.9g.m^{-3}$  (resp  $0.95g.m^{-3}, 0.99g.m^{-3}$ ).

On a rassemblé les valeurs des masses  $M_1, M_2, M_3$  dans le tableau 3.1, au temps  $t = 2s$ .

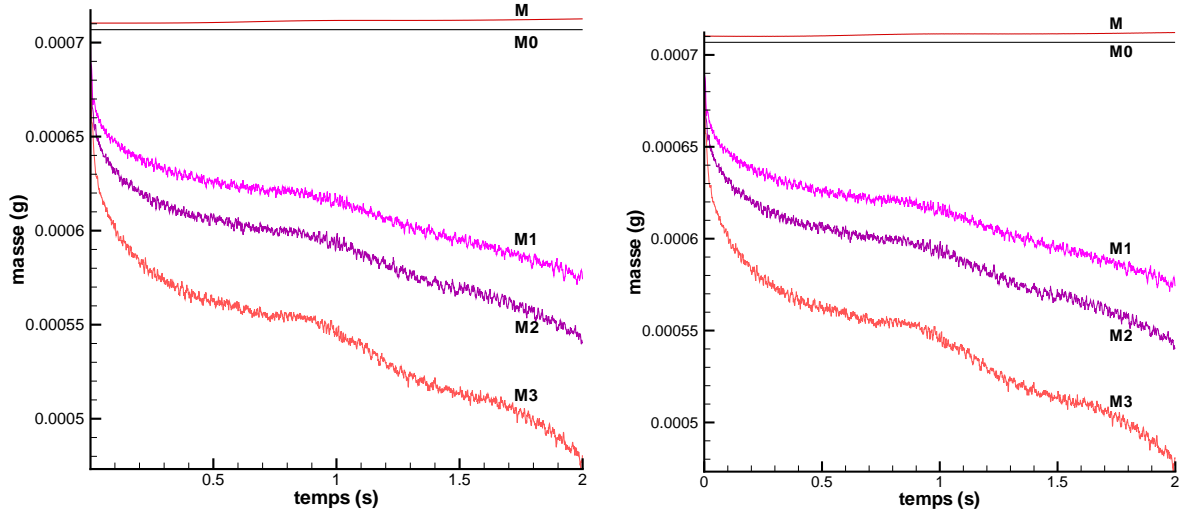


Figure 3.5 : *Evolution de la masse, avec  $G_0 = 16 * 16$  et deux niveaux de raffinement pour la figure de gauche et avec  $G = 144 * 144$  sans AMR pour la figure de droite. Le champ de vitesse est analytique. Interpolation binaire de  $S_p$  sur limites des niveaux de raffinement.*

Dans la première colonne, on utilise un maillage grossier  $G_0 = 16 * 16$  avec deux niveaux de raffinement et dans la deuxième colonne, un maillage unique fin  $G = 144 * 144$ .

Pour déterminer si la masse est conservée, on calcule l'écart relatif  $e_{rel}$  de  $M$  à  $M_0$ , soit  $e_{rel} = \frac{|M - M_0|}{M_0}$ . En utilisant les résultats du tableau 3.1, on évalue cet écart à 0.8%. Ceci montre une bonne conservation de la masse. Cette variation de masse est simplement due au fait que  $M_0$  est calculée à partir de la formule  $\Pi r^2$  alors que la masse est calculée numériquement et donc de manière discontinue (les arcs de cercle sont approché par des segments).

Il est à noter que si on utilise un maillage unique  $G = 16 * 16$  sans AMR, les masses  $M_1$ ,  $M_2$  et  $M_3$  ont totalement disparu au temps  $t = 2s$ , ce qui signifie que la tache cisailée a été complètement diffusée (cf figure 3.6). En calculant l'écart relatif  $e_{rel}$  de  $M$  à  $M_0$ , on obtient 15%. Ceci montre simplement que l'utilisation de la méthode AMR permet de réduire la diffusion numérique de la même façon que ce qu'on pourrait obtenir sur un maillage fin équivalent sans AMR. Sachant que l'évolution temporelle des différentes masses est quasiment la même avec et sans AMR, on souhaite savoir si la répartition de la concentration massique reste la même en utilisant les deux méthodes.

On compare donc la solution obtenue sur un maillage fin  $G = 144 * 144$ ,  $S_{psamr}$ , sans

masse	AMR, vitesse analytique	sans AMR, vitesse analytique
$M$	0.000712591	0.000712161
$M_1$	0.000578019	0.000578019
$M_2$	0.000540958	0.000540958
$M_3$	0.000478011	0.000478011

Tableau 3.1 :  $M$ ,  $M_1$ ,  $M_2$  et  $M_3$  à  $t = 0.8s$  sur  $G_0 = 16 * 16$  avec deux niveaux d'AMR (colonne 1) puis  $G = 144 * 144$  sans AMR (colonne 2). Vitesse analytique imposée sur tous les nœuds.

AMR à celle obtenue via deux niveaux fins,  $S_{pamr}$ . Pour cela, on calcule l'erreur  $\epsilon$  en norme  $L^2$  entre ces deux solutions en considérant  $S_{psamr}$  comme solution de référence (cf définition de l'erreur (3.3)). On trouve  $\epsilon = 6.1672 \cdot 10^{-5}$ . On peut donc affirmer que la répartition de la concentration est la même à  $10^{-4}$  près avec et sans AMR. Ceci, montre que la méthode est satisfaisante, non seulement parce qu'elle permet de réduire la diffusion numérique mais aussi parce qu'elle ne dégrade pas la solution.

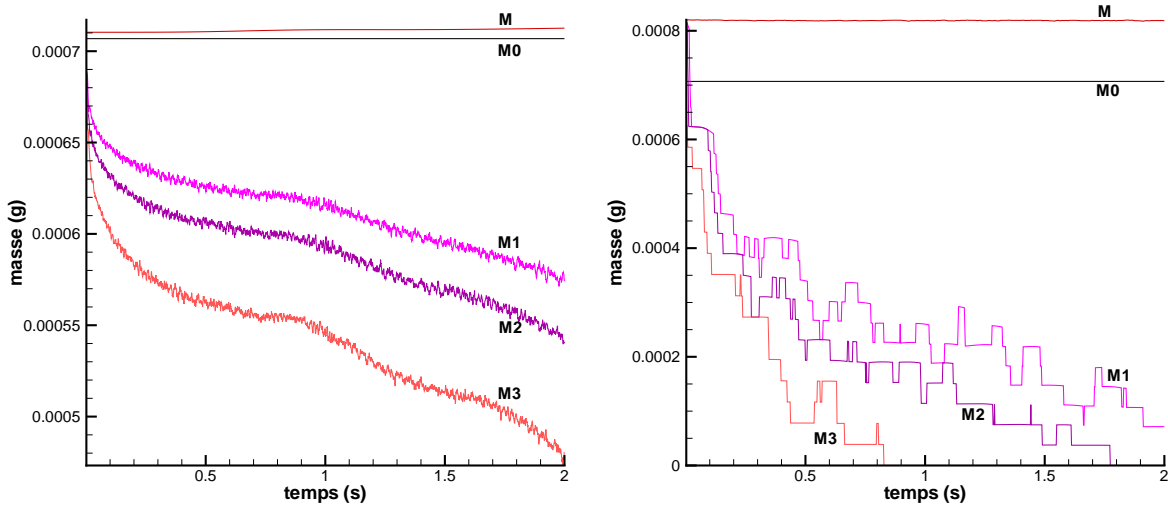


Figure 3.6 : Evolution de la masse, avec  $G_0 = 16 * 16$  et deux niveaux d'AMR pour la figure de gauche et avec  $G = 16 * 16$  sans AMR pour la figure de droite. Le champ de vitesse est analytique. Interpolation binaire de  $S_p$  sur limites des niveaux de raffinement.

### 3.1.3 Choix du critère de raffinement

Suite aux explications fournies dans la sous-section 2.2, nous avons choisi un critère de raffinement-déraffinement de type gradient de concentration.

On a essayé de faire une analogie avec un mur semi-infini en thermique pour évaluer le seuil de raffinement  $seuil_{grad}$  (défini dans la sous-section 2.2), cependant, les effets d'inertie et le fort cisaillement de la tache ne permettent pas de définir une valeur de seuil satisfaisante, à partir de la seule diffusion.

Par expérience, on a estimé qu'une variation de  $0.07g.m^{-3}$ , par maille, de concentration massique  $S_p$  sur un maillage  $G_0 = 16 * 16$  permettait un recouvrement parfait du front de concentration par les cellules  $AMR$ , durant le temps de simulation. Cette variation donne au seuil la valeur  $seuil_{grad} = 11.9$ . Elle est utilisée pour générer des cellules sur les différents niveaux.

Il suffit de prendre une valeur du seuil de déraffinement légèrement inférieur à  $seuil_{grad}$ , soit  $seuil_{grad-deraf} = 11$ .

On remarque que, à condition qu'on ait bien choisi  $seuil_{grad}$ , l'utilisation de l'interpolation binaire pour la concentration massique n'introduit aucune diffusion et permet même de la contenir à l'intérieur des niveaux de raffinement. Une sous-estimation de ce seuil n'est donc pas gênante car elle n'entraînera aucune augmentation du nombre de cellules  $AMR$  créées (cette remarque n'est bien sûr pas valable pour un seuil nul).

### 3.1.4 Calcul du champ de vitesse

Dans la sous section précédente, le champ de vitesses analytique a été imposé sur tous les niveaux de raffinement. Dans la plupart des écoulements, le champ de vitesses n'est pas connu, il faut donc le déterminer en résolvant les équations de Navier-Stokes. Est-il suffisant de calculer le champ de vitesses sur le niveau grossier et de l'interpoler sur les niveaux de raffinement? Cette interpolation n'étant pas à divergence nulle, il est important d'évaluer son effet sur la dynamique de la tache cisillée. On va donc se mettre dans les mêmes conditions que dans la sous section précédente, mis à part que la vitesse est maintenant interpolée sur les deux niveaux fins avec du  $Q_1$  dans le cas raffiné. On trace l'évolution des différentes masses  $M$ ,  $M_1$ ,  $M_2$  et  $M_3$  jusqu'au temps  $t = 2s$  sur la figure 3.7 dans les cas avec et sans  $AMR$ . On rassemble les résultats obtenus avec et sans  $AMR$  dans le tableau 3.2 au temps  $t = 2s$ : On indique dans la première colonne les valeurs des masses avec  $AMR$  et dans la deuxième colonne les valeurs des masses sans méthode de

raffinement. On a reporté dans la troisième colonne les écarts relatifs  $e_{rel}$  de masse obtenus avec les deux méthodes. On précise que ces calculs ont été effectués en considérant les valeurs des masses obtenues sans *AMR* comme référence.

On remarque que ces écarts sont très faibles pour  $M_1$ ,  $M_2$  et  $M_3$ , donc la méthode de raffinement de maillage donne une évolution des différentes masses assez proche de la méthode classique. Par contre, l'écart relatif entre  $M$  et  $M_0$  atteint 4%. Ce qui est plus inquiétant, est l'augmentation de masse  $M$ , assez marquée (de l'ordre de 4% cf figure gauche 3.7) lorsque la vitesse est interpolée sur les niveaux fins.

Qu'en est-il de la répartition de la concentration massique? Sachant que les résultats sur la masse  $M$  sont peu convaincants, on s'attend à avoir une répartition de la concentration très moyenne. Pour en savoir plus, on calcule l'erreur  $\epsilon$  entre la solution avec et sans *AMR*. Cette fois-ci,  $\epsilon = 0.11$  (voir aussi la figure 3.8) contre  $\epsilon = 6.1672 \cdot 10^{-5}$  lorsque le champ de vitesses analytique est imposé sur les niveaux fins. Ce résultat montre donc que la méthode *AMR* a su réduire la diffusion numérique efficacement, cependant la répartition de la concentration est très sensible à la dynamique de l'écoulement. Il est donc impératif d'avoir un champ de vitesses assez précis sur les niveaux fins.

Remarque : Si on regarde la figure 3.8 de plus près, on observe que la tache cisailée avec la méthode de raffinement est en retard par rapport à celle sans *AMR*. Le champ analytique étant sinusoidal, l'interpolation linéaire a tendance à sous-estimer son amplitude au voisinage de ses extrema. La tache évoluant dans la zone de fort cisaillement, la sous-estimation du champ de vitesses expliquerait le retard observé sur la figure 3.8.

masse	<i>AMR</i> , vitesse $Q_1$	sans <i>AMR</i> , vitesse analytique	écart relatif en %
$M$	0.000738269	0.000712161	4.43
$M_1$	0.000576199	0.000578019	0.315
$M_2$	0.000544176	0.000540958	0.6
$M_3$	0.000486193	0.000478011	1.7

Tableau 3.2 :  $M$ ,  $M_1$ ,  $M_2$  et  $M_3$  à  $t = 2s$  sur  $G_0 = 16 * 16$  avec deux niveaux fins (colonne 1) puis  $G = 144 * 144$  sans *AMR* (colonne 2). Vitesse interpolée sur les niveaux fins et analytique ailleurs.

Deux possibilités peuvent être alors envisagées pour améliorer la précision du champ

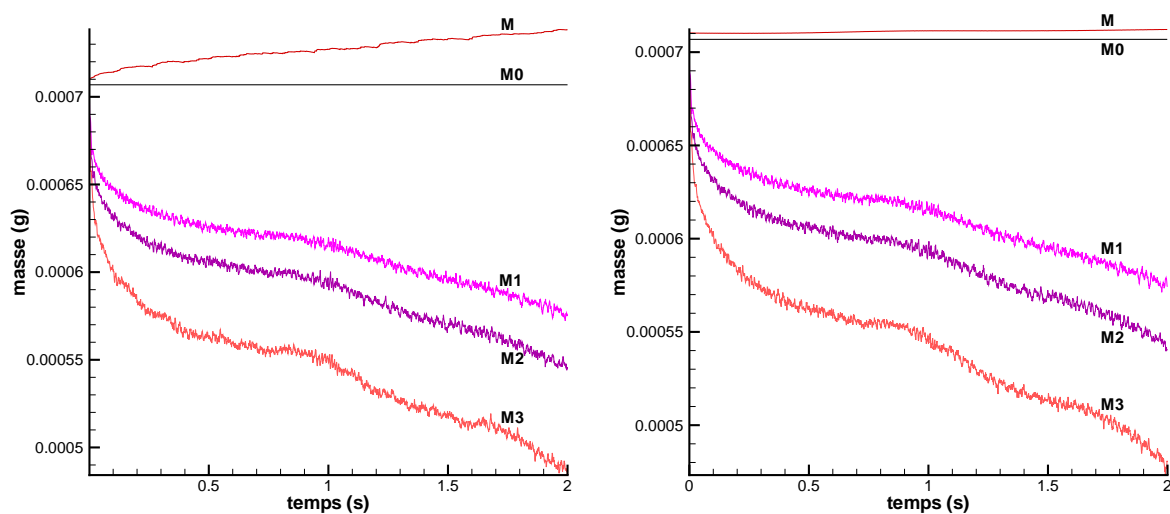


Figure 3.7 : *Evolution de la masse, avec  $G_0 = 16 * 16$  et deux niveaux de raffinement pour la figure de gauche et avec  $G = 144 * 144$  sans AMR pour la figure de droite. Vitesse interpolée sur les niveaux fins avec du  $Q_1$  et analytique ailleurs. Interpolation binaire de  $S_p$  pour les nœuds de limites des niveaux de raffinement.*

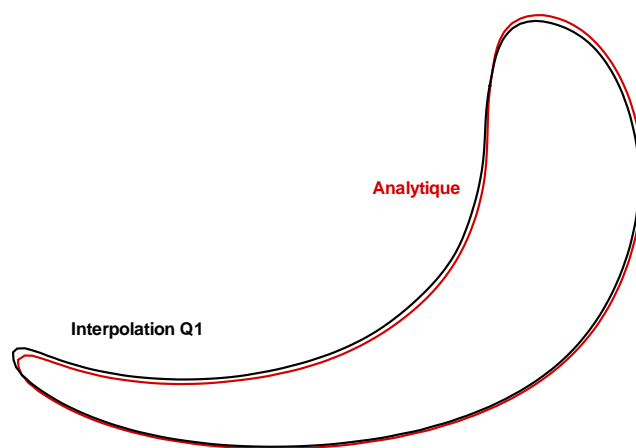


Figure 3.8 : *Concentration massique sur le second niveau de raffinement, avec un champ de vitesse analytique (rouge, trait continu) et interpolé (noir, trait discontinu). Maillage  $G_0 = 16 * 16$ ,  $t = 2s$ . Interpolation binaire de  $S_p$  sur limites des niveaux de raffinement.*

de vitesses : la première consiste à résoudre les équations de Navier-Stokes à l'intérieur des niveaux fins et interpoler leurs limites dans le but de réduire l'effet de l'interpolation. La deuxième méthode consiste à se placer sur un maillage grossier  $G_0$  assez fin de façon à ce que le champ de vitesses ait déjà la précision souhaitée et de l'interpoler sur les niveaux fins.

Nous allons voir si la deuxième méthode donne des résultats satisfaisants. On considère un maillage  $G_0 = 32 * 32$  avec deux niveaux fins. Son maillage équivalent fin sans *AMR* est  $G = 288 * 288$ .

La figure 3.9 illustre les évolutions temporelles des masses  $M$ ,  $M_1$ ,  $M_2$  et  $M_3$  avec et sans *AMR* jusqu'au temps  $t = 2s$ . Comme on l'a vu pour le cas raffiné avec  $G_0 = 16 * 16$ , on retrouve une évolution des masses  $M_1$ ,  $M_2$  et  $M_3$  similaires avec et sans *AMR* avec des écarts relatifs très faibles entre 0.1% et 0.2% (cf tableau 3.3). On observe cette fois-ci que l'évolution de la masse  $M$  avec la méthode *AMR* est très proche de celle sans cette méthode. Ceci montre qu'en partant d'un maillage grossier  $G_0$  plus fin, la seule interpolation du champ de vitesses et la résolution du transport d'espèce sur les niveaux fins permettent d'obtenir une évolution de la masse  $M$  très satisfaisante.

D'ailleurs, le tableau 3.3 vient conforter la démarche suivie puisque l'écart relatif entre  $M$  et  $M_0$  est de 0.1% pour le cas avec raffinement et 0.2% pour celui sans. Donc dans les deux cas, la masse est bien conservée. Quant à la répartition de la concentration massique, on calcule une erreur  $\epsilon = 3.8345 \cdot 10^{-2}$  (cf figure 3.10), donc la dynamique de la tache cisailée a été mieux traitée en partant d'un maillage  $G_0 = 32 * 32$  au lieu de  $G_0 = 16 * 16$ .

On peut donc conclure que la méthode *AMR* donne de bons résultats concernant la réduction de la diffusion numérique et la précision de la répartition de la concentration massique lorsque la vitesse est interpolée sur les niveaux fins et l'équation de transport résolue sur ces mêmes niveaux. Il est à noter que cette stratégie (vitesse interpolée, transport résolu) est intéressante si et seulement si on utilise un maillage grossier sur lequel la précision souhaitée du champ de vitesse a été atteinte et que celle souhaitée pour la concentration massique ne l'a pas encore été. Cependant, pour des écoulements réalistes cette approche a de fortes chances d'être inutilisable. On en déduit que pour des écoulements réalistes, il faut résoudre les équations de Navier-Stokes sur tous les niveaux pour avoir un champ de vitesses le plus précis possible. Reste à déterminer l'influence de l'interpolation non conservative (utilisée pour résoudre les nœuds appartenant aux limites de niveaux de raffinement) sur l'intérieur des niveaux fins. En effet, les gradients de vitesse se situant dans l'ensemble du domaine, l'interpolation non conservative risque d'avoir une

influence non négligeable sur le champ de vitesses des niveaux fins. Pour l'appréhender, on fait une analogie avec un cas de thermique (cas de Whitaker, section 3.3).

masse	AMR, vitesse $Q_1$	sans AMR, vitesse analytique	écart relatif en %
$M$	0.00071077	0.00071056	0.003
$M_1$	0.000637201	0.000636302	0.1413
$M_2$	0.000621058	0.000620178	0.1419
$M_3$	0.000585361	0.000584021	0.2294

Tableau 3.3 :  $M$ ,  $M_1$ ,  $M_2$  et  $M_3$  à  $t = 0.8s$  sur  $G_0 = 16 * 16$  avec deux niveaux de raffinement (colonne 1) puis  $G = 144 * 144$  sans AMR (colonne 2). Vitesse interpolée sur les niveaux fins et analytique ailleurs.

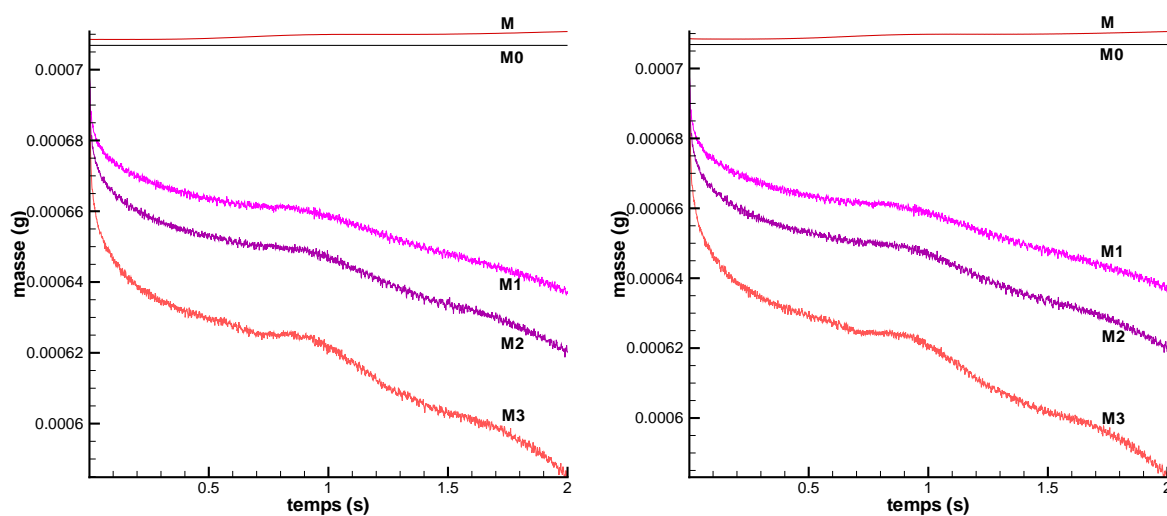


Figure 3.9 : Evolution de la masse, avec  $G_0 = 32 * 32$  et deux niveaux de raffinement (gauche) et avec  $G = 288 * 288$  sans AMR (droite). Le champ de vitesse est interpolé avec du  $Q_1$  sur les niveaux fins et analytique ailleurs. Interpolation binaire de  $S_p$  sur limites des niveaux de raffinement.



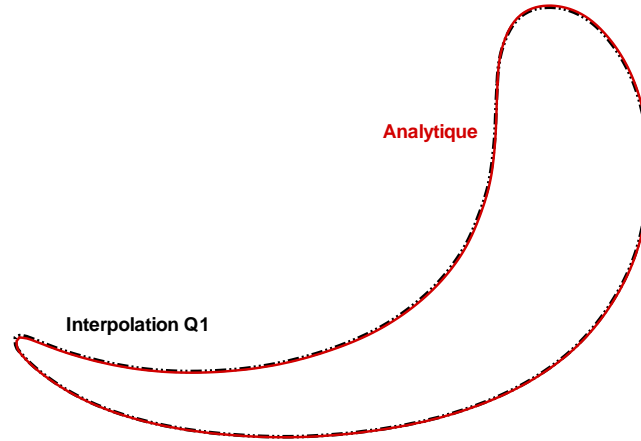


Figure 3.10 : Concentration massique sur le niveau  $G_2$ , avec un champ de vitesse analytique (rouge, trait continu) et interpolé (noir, trait discontinu). Maillage  $G_0 = 32 * 32$ ,  $t = 0.8s$ . Interpolation binaire de  $S_p$  sur limites des niveaux de raffinement.

### 3.1.5 Ordre de convergence de la méthode AMR

On veut savoir ici comment agit cette méthode sur l'ordre de convergence des schémas. On effectue donc dans un premier temps une étude de convergence en maillage de la tache cisailée sans AMR pour deux coefficients de diffusion  $D = 10^{-9}m^2.s^{-1}$  et  $D = 10^{-8}m^2.s^{-1}$ .

Soient  $L_x$  (respectivement  $L_z$ ) la longueur du domaine suivant  $x$  (respectivement  $z$ ),  $N_x$  (respectivement  $N_z$ ) le nombre de nœuds suivant  $x$  (respectivement  $z$ ) et  $N = N_x N_z$  le nombre de points. On travaille à pas d'espace constant. Posons  $\Delta x$  (respectivement  $\Delta z$ ) le pas d'espace suivant  $x$  (respectivement  $z$ ), défini par  $\Delta x = \frac{L_x}{N_x}$  (respectivement  $\Delta z = \frac{L_z}{N_z}$ ). On définit  $h$  par  $h = \sqrt{\Delta x \Delta z}$  et  $\epsilon_h$  l'erreur associée.

Cette étude consiste à calculer successivement les erreurs  $\epsilon_h$  entre la solution de référence  $S_{p_{ref}}$  et la solution,  $S_{p_{num}}$ , obtenue sur chaque maillage  $G = (N_x - 1)(N_z - 1)$  de pas d'espace de plus en plus petit.

Lorsque la solution analytique n'existe pas, la solution de référence est calculée sur un maillage très fin de façon à ce que la solution soit suffisamment résolue. Comme son nom

l'indique, cette solution sert de référence. Sachant que l'erreur  $\epsilon$  évolue selon la relation  $\epsilon = k(\frac{L}{\sqrt{N}})^q$ , on peut déduire l'ordre de convergence  $q$  en traçant la courbe  $\ln(\epsilon_h) = \ln(k) - q\ln(\frac{L}{\sqrt{N}})$ , où  $k$  est une constante,  $L = \sqrt{L_x L_z}$  et  $q$  l'ordre de convergence. L'erreur  $\epsilon_h$  est définie par :

$$\epsilon_h = \frac{\|S_{pnum} - S_{pref}\|_2}{\|S_{pref}\|_2} \quad (3.2)$$

L'objectif est d'estimer l'ordre de convergence limite  $q$ .

Dans un deuxième temps, on réalise la même étude avec la méthode *AMR*. Les calculs d'erreur  $\epsilon_h$  successifs entre la solution de référence et les solutions *AMR*, pour des pas d'espace  $h$  de plus en plus petit, s'obtiennent en augmentant le nombre de niveaux *AMR*. On prend comme solution de référence la tache cisailée sur un maillage  $G = 1296 * 1296$  au temps  $t = 8s$ . Le pas de temps choisi est  $\Delta t = 4.10^{-4}s$ . On a rassemblé les erreurs à chaque maillage  $G$  dans le tableau 3.4 pour différents coefficients de diffusion sans *AMR*. On voit que pour  $D = 10^{-8}m^2.s^{-1}$ , les ordres de convergence successifs  $q_h$  (cf figure 3.11 haut) tendent vers l'ordre limite  $q = 2$ , ce qui concorde avec l'ordre théorique du schéma utilisé. Par contre, on remarque que pour  $D = 10^{-9}m^2.s^{-1}$  la convergence se fait plus difficilement puisque  $q_4 = 1.99$  pour  $D = 10^{-8}m^2.s^{-1}$  contre seulement  $q_4 = 1.29$  pour  $D = 10^{-9}m^2.s^{-1}$  sur la figure ?? bas.

		$D = 10^{-8}m^2.s^{-1}$	$D = 10^{-9}m^2.s^{-1}$
maillage	$N$	$\epsilon_h$	$\epsilon_h$
144 * 144	21025	0.3151	
162 * 162	26569	0.2854	0.3264
324 * 324	105625	0.1370	0.2017
432 * 432	187489	$8.5657 \cdot 10^{-2}$	0.1533
648 * 648	421201	$3.8298 \cdot 10^{-2}$	$9.0825 \cdot 10^{-2}$

Tableau 3.4 : *Etude de convergence en maillage sans AMR pour le cas de la tache cisailée*

Ceci s'explique par le fait que pour un même temps  $t_f$ , la longueur caractéristique de diffusion  $\lambda$  est plus grande pour un coefficient de diffusion  $D = 10^{-8}m^2.s^{-1}$  que

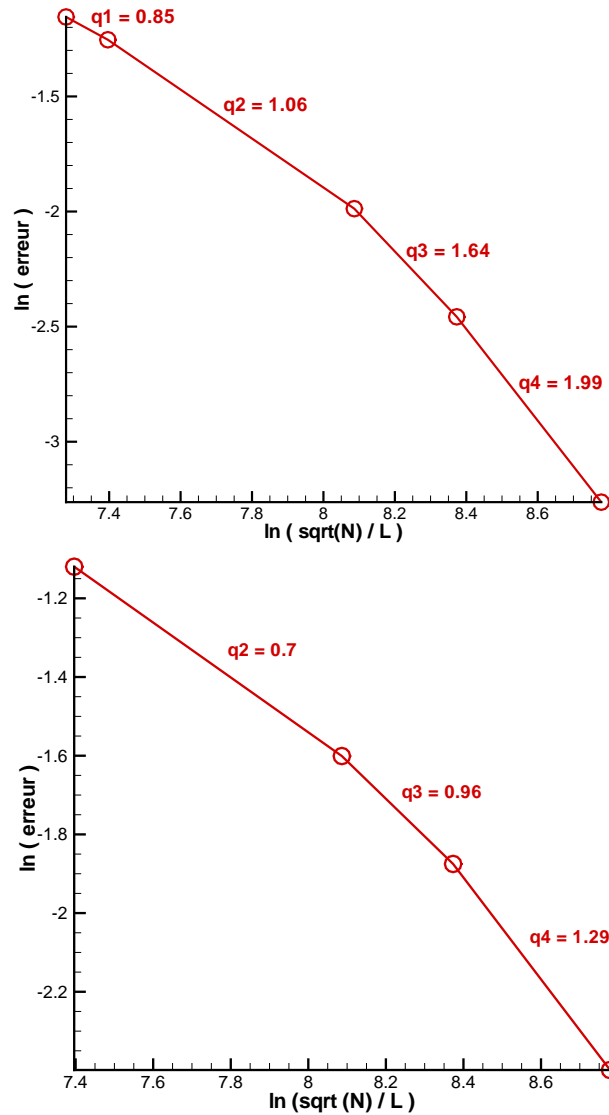


Figure 3.11 : Courbe de convergence pour un coefficient  $D = 10^{-8}m^2.s^{-1}$  sans AMR (haut). Courbe de convergence pour un coefficient  $D = 10^{-9}m^2.s^{-1}$  sans AMR (bas).

$D = 10^{-9}m^2.s^{-1}$ . L'ordre de grandeur de cette longueur caractéristique associée au coefficient de diffusion  $D = 10^{-d}$ , est  $\lambda_d = \sqrt{2Dt_f}$  en faisant une analogie avec le cas d'un mur semi-infini initialement isotherme, auquel on impose initialement une température de paroi (cf [TAINÉ 03]). Avec un temps  $t_f = 8s$ , on trouve  $\lambda_8 = 5.656 \cdot 10^{-4}m$  et  $\lambda_9 = 1.788 \cdot 10^{-4}m$ . Il est aussi intéressant de noter que  $\lambda_9 = \frac{\lambda_8}{\sqrt{10}}$ .

Il est clair que l'ordre du schéma ne pourra pas avoir atteint sa valeur limite tant que la longueur caractéristique  $\lambda$  de la diffusion physique sera inférieure à la longueur caractéristique  $\lambda_d$ .

téristique de la diffusion numérique. Sachant que la diffusion numérique s'étale sur deux à trois mailles, on détermine un pas d'espace critique  $\Delta x_{ci}$  (on se limite au pas d'espace suivant  $x$  car dans ce cas  $\Delta x = \Delta z$ ) pour lequel  $\lambda_i$ ,  $i \in [8, 9]$  est supérieure à trois mailles. On pourra alors considérer que la solution a convergé en maillage lorsque  $\Delta x \leq \Delta x_{c8}$  (respectivement  $\Delta x \leq \Delta x_{c9}$ ) pour  $D = 10^{-8}m^2.s^{-1}$  (respectivement  $D = 10^{-9}m^2.s^{-1}$ ). Les pas d'espaces critiques sont rassemblés dans le tableau 3.5, ainsi que les maillages équivalents, pour les deux coefficients de diffusion.

On note que les résultats du tableau 3.5 concordent avec ceux de la figure 3.11. En

	$\Delta x_c$	$G_c$
$D = 10^{-8}m^2.s^{-1}$	$1.885 \cdot 10^{-3}$	$529 * 529$
$D = 10^{-9}m^2.s^{-1}$	$5.96 \cdot 10^{-4}$	$1676 * 1676$

Tableau 3.5 : Pas d'espace et maillages critiques pour différents coefficients de diffusion.

effet, pour un maillage  $648 * 648$  (supérieur au maillage critique  $G_c = 529 * 529$ ), on obtient un ordre de convergence  $q_4 = 1.99$ . On peut estimer que  $\lambda_8$  est de l'ordre de  $\frac{\lambda_8}{\Delta x} = \frac{\lambda_8}{\Delta z} = 3.68 \sim 4$  mailles. Par contre, pour  $D = 10^{-9}m^2.s^{-1}$ , le maillage  $648 * 648$  est inférieur au maillage critique  $G_c = 1676 * 1676$ . il est donc illusoire de penser que l'ordre limite est atteint. En effet, la figure 3.11 bas montre que l'ordre de convergence n'est que  $q_4 = 1.29$ . On calcule d'ailleurs que  $\frac{\lambda_9}{\Delta x} = \frac{\lambda_9}{\Delta z} = 1.2$  mailles.

On s'intéresse maintenant à l'ordre de convergence avec la méthode *AMR*. Pour cela, on considère un maillage grossier  $G_0 = 16 * 16$ , un pas de temps  $\Delta t = 0.308 \cdot 10^{-4}s$  et la solution obtenue sans *AMR* sur le maillage  $G = 1620 * 1620$ , comme solution de référence. La diminution du pas d'espace le plus petit se fait en ajoutant des niveaux de raffinement. Le tableau 3.6 montre les différentes erreurs  $\epsilon_h$  lorsque le nombre de niveaux fins augmente. La figure 3.12, utilisant les données du tableau, montre que l'ordre de convergence tend vers une valeur limite  $q = 2$ . La méthode *AMR* permet donc de conserver l'ordre de convergence des schémas.

On peut refaire la même étude de convergence pour  $D = 10^{-9}m^2.s^{-1}$  avec la méthode *AMR*. Cependant  $G_4 = 1296 * 1296$  est inférieur au maillage critique. Il faudrait donc comparer les solutions obtenues sur différents maillage avec une solution de référence sans *AMR* sur un maillage  $G = 11664 * 11664$ . L'obtention de cette solution demandant beaucoup de ressources informatiques et de temps, on se contentera des résultats obtenus pour

		$D = 10^{-8}m^2.s^{-1}$
maillage	$NBP$	$\epsilon_h$
60 * 60	2633	2.1726 10 <sup>-2</sup>
180 * 180	14068	1.2804 10 <sup>-2</sup>
540 * 540	61395	3.54065 10 <sup>-3</sup>

Tableau 3.6 : Etude de convergence en maillage avec AMR pour le cas de la tache cisailée avec  $D = 10^{-8}m^2.s^{-1}$

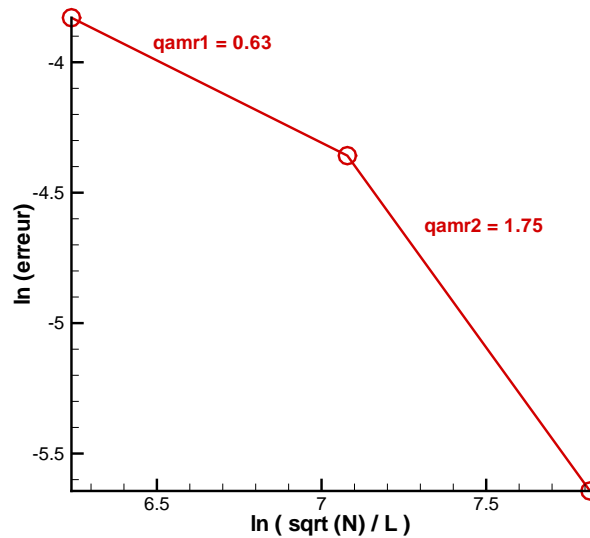


Figure 3.12 : Courbe de convergence pour un coefficient  $D = 10^{-8}m^2.s^{-1}$  avec AMR .

un coefficient de diffusion  $D = 10^{-8}m^2.s^{-1}$  pour conclure sur l'efficacité de la méthode. On peut visualiser les solutions obtenues avec AMR sur les figures (3.13) et (3.14). On peut remarquer que les cellules AMR (disposées de part et d'autre du front de concentration) recouvrent parfaitement la zone de diffusion. Ceci permet de conserver la totalité de l'information à l'intérieur de chaque niveau fin, et donc de réduire efficacement la diffusion numérique sans dégrader la répartition de la concentration massique. On peut observer que plus on ajoute de niveaux de raffinement, plus on se rapproche de la solution de référence (cf figure 3.15).

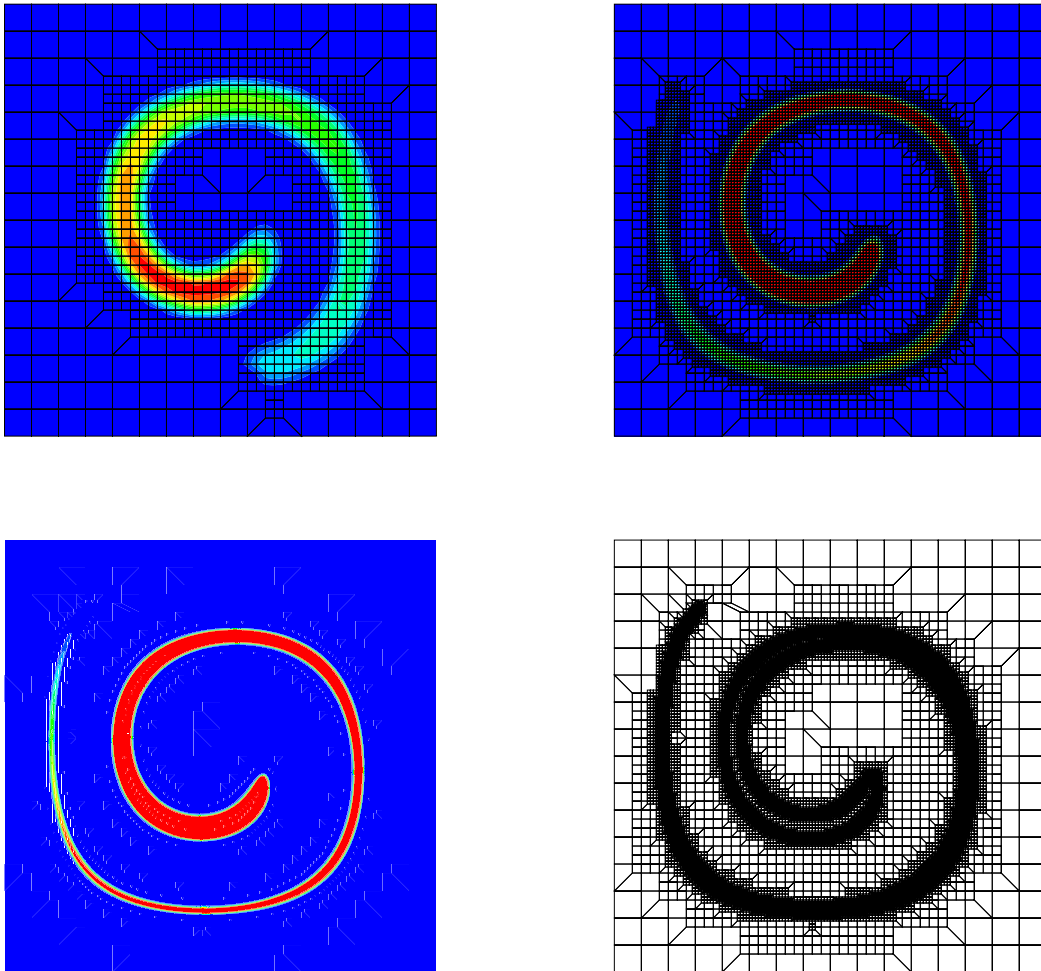


Figure 3.13 : Concentrations massiques et maillages associés à  $t = 8s$ , de un à trois niveaux de raffinement, de gauche à droite et de haut en bas. Pour une meilleure visualisation, on sépare  $S_p$  du maillage lorsqu'il y a plus de deux niveaux. Le champ de vitesse est analytique sur tous les niveaux. Seuls sont représentés les nœuds résolus de niveau le plus fin. Maillage de départ,  $G_0 = 16 * 16$ .

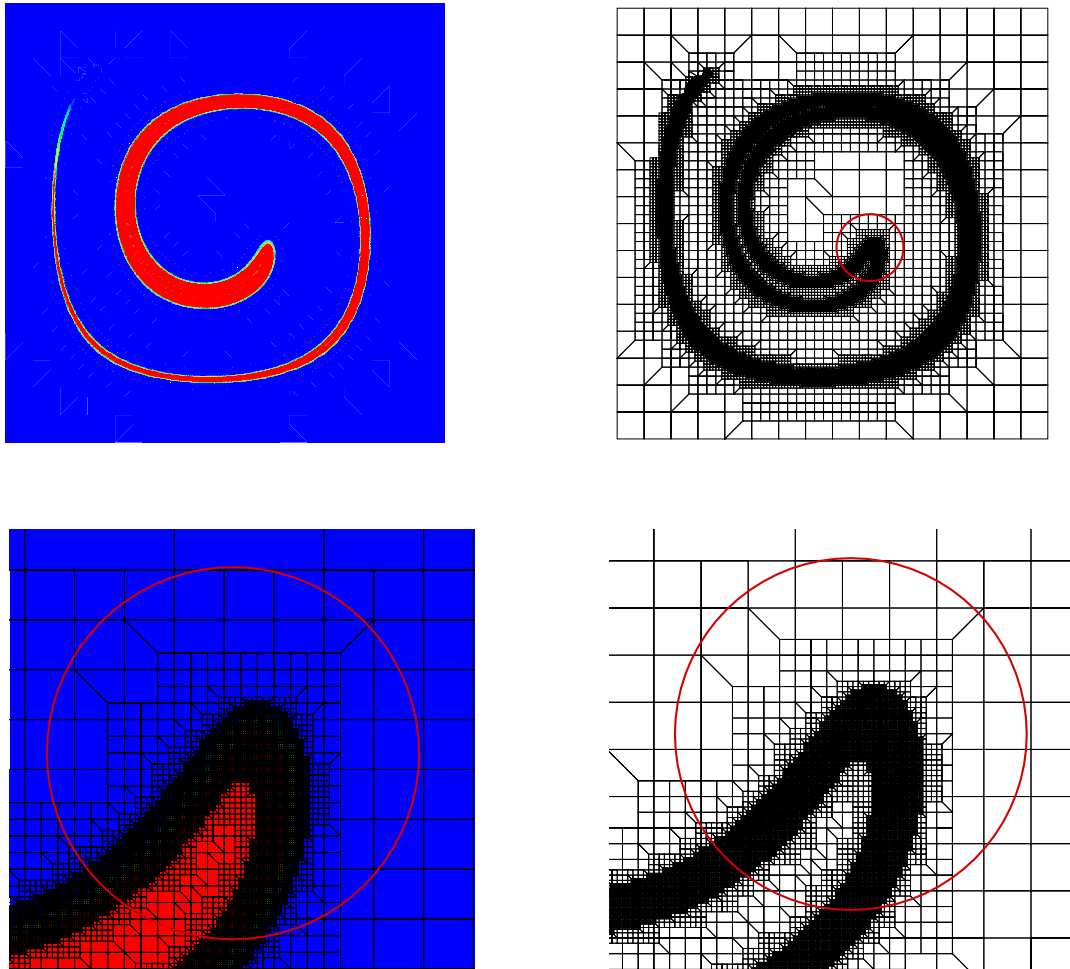


Figure 3.14 : Concentrations massiques et maillages associés à  $t = 8s$ , sur quatre niveaux de raffinement avec zoom, de gauche à droite et de haut en bas. Le champ de vitesse est analytique sur tous les niveaux. Seuls sont représentés les nœuds résolus de niveau le plus fin. Maillage de départ,  $G_0 = 16 * 16$ .

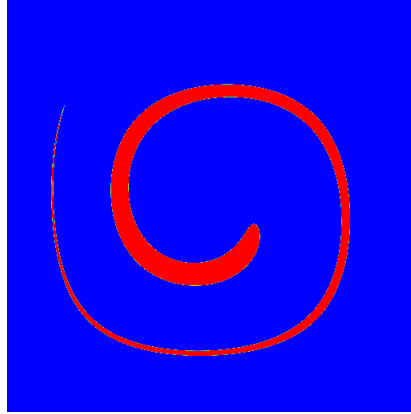


Figure 3.15 : *Concentration massique de référence à  $t = 8s$ , sur  $G = 3888 * 3888$ . Le champ de vitesse est analytique.*

### 3.1.6 Performances en temps CPU et mémoire de la méthode de raffinement

Ayant montré les performances de convergence en maillage de la méthode *AMR*, on s'intéresse à sa capacité à optimiser le temps *CPU* et la taille mémoire. Pour cela, on analyse plusieurs paramètres liés au temps de simulation et à la taille mémoire.

On choisit un maillage grossier  $G_0 = 20 * 20$  et un pas de temps  $\Delta t = 3.08 * 10^{-4}s$ . De plus, la tache est cisailée pendant une durée de  $8s$ .

#### Analyse de la mémoire

Pour estimer les performances de cette méthode en terme de mémoire, on compare la mémoire vive utilisée pour simuler le cas avec *AMR* et celle utilisée sur maillage fin équivalent sans *AMR*. On rassemble les résultats obtenus dans le tableau 3.7. Dans la première colonne est indiqué le pas d'espace  $\Delta x = \Delta z$  dans le cas sans *AMR* et celui sur le niveau le plus fin dans le cas avec raffinement. Dans la deuxième colonne (respectivement la troisième colonne), on donne la mémoire utilisée, en Mega octets (*Mo*), avec (respectivement sans) *AMR*,  $mem_{AMR}$  (respectivement  $mem$ ). Dans la dernière colonne, on a calculé le rapport  $rap = \frac{mem_{AMR}}{mem}$ .

Afin de mieux interpréter les résultats du tableau 3.7, on s'appuie en plus sur les résultats de la figure 3.16, représentant  $rgl = \frac{NBGL}{KKT_L}$  en fonction du temps, et ceux du tableau 3.8.  $NBGL$  représente le nombre de nœuds résolus avec la méthode *AMR* sur les  $L+1$  niveaux



et  $KKT_L$  le nombre de nœuds résolus sans la méthode de raffinement sur un maillage fin équivalent. On observe sur le tableau 3.7 que pour de petits maillages tels  $G = 60 * 60$ , la méthode *AMR* est plus coûteuse en mémoire que celle sans. Ceci est normal car on résout avec l'*AMR* un nombre de nœuds qui correspond à 71% des nœuds qu'il aurait fallu résoudre sur un maillage fin équivalent sans *AMR* (cf tableau 3.8). Quand le nombre de niveaux fins augmente le rapport des mémoires *rap* diminue (cf tableau 3.7), ainsi que le rapport *rgl* (cf tableau 3.8). On obtient par exemple un gain de 70% de mémoire avec trois niveaux de raffinement et la résolution de seulement 21% des nœuds qu'il aurait fallu résoudre sur un maillage fin équivalent sans *AMR*.

Ce résultat n'est pas surprenant dans la mesure où cette méthode de raffinement nécessite la création de tableaux supplémentaires (tableau de connection entre cellules fines, tableau indiquant la structure en arbre des nœuds père-fils, ...) pour gérer le raffinement et le déraffinement. Donc cette méthode sera performante en terme de mémoire à partir du moment où les ressources dont elle a besoin sont faibles devant celles demandée par le reste du code.

On peut quantifier ce rapport en estimant une valeur critique de *rgl*,  $rgl_{crit}$ , en deçà de laquelle il est intéressant d'utiliser la méthode *AMR*. Soit  $rap_{crit} = 0.75$ , la valeur critique de *rap* en deçà de laquelle on estime que le gain en taille mémoire est suffisant pour préférer la méthode *AMR*, vu l'effort de mise en oeuvre. On considère la fonction  $rap = f(rgl)$ . A partir des données des tableaux 3.7 et 3.8, une régression linéaire de la fonction  $rap = f(rgl)$  donne un coefficient de corrélation de 0.995 et la droite d'équation  $rap = 4.605 * rgl - 0.7376$ . On en déduit  $rgl_{crit} = 32\%$ .

Ainsi, tant que le nombre de nœuds résolus avec *AMR* est inférieur à 32% du nombre de nœuds qu'il aurait fallu résoudre sans *AMR* sur un maillage fin équivalent, cette méthode de raffinement permet d'obtenir des gains en taille mémoire très intéressants.

Dans les conditions qui ont été choisies, le critère sur *rgl* n'est vérifié qu'à partir du troisième niveau de raffinement, donc cette méthode devient très efficace en termes de taille mémoire lorsqu'il y a au moins trois niveaux fins.

On observe dans le tableau 3.8 que le rapport *rgl* diminue quand le nombre de niveaux augmente. Ceci est simplement du au fait que le nombre de nœuds ayant une concentration massique  $S_p = 1g.m^{-3}$  devient petit devant celui ayant une concentration massique  $S_p = 0g.m^{-3}$  quand le pas d'espace du maillage diminue. Donc, si on choisit un maillage  $G_0$  assez fin pour que le critère sur *rgl* soit vérifié pour  $G_1$ , alors il le sera pour tous les autres niveaux  $G_L$ ,  $L > 1$ . Ainsi la méthode *AMR* peut être intéressante en terme de

taille mémoire dès le premier niveau de raffinement.

En effet, si on simule le même cas avec un niveau de raffinement en partant du maillage  $G_0 = 180 * 180$ , le coût en taille mémoire s'élève à  $111Mo$ , soit un gain de 60%. Ceci est normal, puisqu'on résout 29% des nœuds qu'il aurait fallu résoudre si on avait utilisé un maillage  $G = 540 * 540$  sans *AMR*.

maillage fin	Mémoire avec <i>AMR</i> (Mo)	Mémoire sans <i>AMR</i> (Mo)	rap
60 * 60	18	7	2.6
180 * 180	37	34	1.1
540 * 540	82	281	0.3
1620 * 1620	830	2550	0.33

Tableau 3.7 : Mémoire utilisée (en Mo) pour simuler la tache cisailée avec et sans *AMR*, sur différents maillages. Pour les cas *AMR*, le maillage grossier utilisé est  $G_0 = 20 * 20$

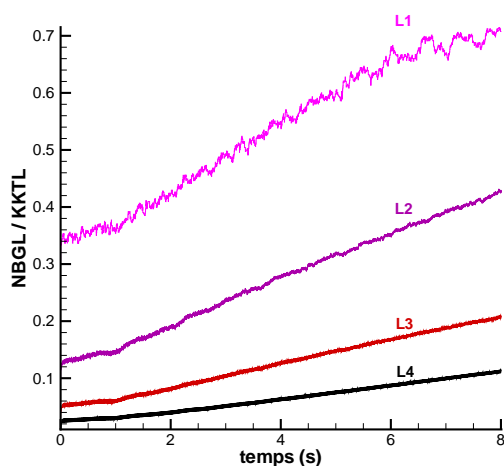


Figure 3.16 : Evolution temporelle du rapport  $r_{gl}$  pour plusieurs niveaux *AMR*.

maillage fin	$NBG_L$	$KKT_L$	$rgl$
60 * 60	2650	3721	0.7122
180 * 180	14018	32761	0.4279
540 * 540	61170	292681	0.2090
1620 * 1620	289981	2627641	0.1104

Tableau 3.8 : Rapport  $rgl$  pour différents nombres de niveaux fins au temps  $t = 8s$ .  
Maillage grossier  $G_0 = 20 * 20$

### Analyse du temps CPU

On cherche à estimer les performances de la méthode de raffinement de maillage en termes de temps de simulation. Le temps *CPU* total du cas *AMR* est la contribution des temps de gestion du raffinement, gestion du déraffinement et de résolution. On décide de quantifier ces différentes contributions en estimant leur temps moyen  $t_{moyen} = \frac{1}{NT} \sum_{i=1,NT} t_i$ , où  $NT$  est le nombre d'itérations en temps et  $t_i$  le temps *CPU* par itération. Soient  $t_{raf}$  le temps moyen pour gérer le raffinement,  $t_{deraf}$  le temps moyen pour gérer le déraffinement,  $t_{resol}$  le temps moyen pour résoudre l'équation de transport et  $t_{total}$  le temps moyen total tel que  $t_{total} = t_{raf} + t_{deraf} + t_{resol}$ . Les figures 3.17, 3.18 et 3.19 représentent l'évolution de ces différents temps moyens lorsqu'on augmente le nombre de niveaux de raffinement. On remarque que le déraffinement a un coût très faible jusqu'à trois niveaux de raffinement, puis il devient un peu plus coûteux pour quatre niveaux. Précisons que la gestion du déraffinement se fait de la même façon que celle du raffinement, sauf que celui-ci se fait à une fréquence choisie, contrairement au raffinement qui se fait à toutes les itérations. Par contre, on observe que l'écart entre  $t_{raf}$  et  $t_{resol}$  s'amenuise quand on augmente le nombre de niveaux puis la gestion du raffinement devient plus coûteuse que la résolution elle-même. On peut donc se demander si la méthode *AMR* reste intéressante d'utilisation sur plusieurs niveaux. On compare donc les temps moyen de résolution et totaux avec et sans *AMR* sur des maillages équivalents (cf figure 3.20). De plus, on indique dans les tableaux 3.9 et 3.10, les temps *CPU* de simulation pour les cas avec et sans *AMR*, ainsi que les différents temps moyens. La comparaison des temps de simulation *CPU* avec et sans *AMR* nous amène à conclure que l'*AMR* est intéressante

en terme de gain en temps *CPU* à partir de trois niveaux de raffinement où ce gain s'élève à 25%.

La question que l'on se pose maintenant est la suivante: Peut-on relier directement les performances en temps *CPU* avec le nombre de nœuds traités?

Pour répondre à cette question, on compare les temps moyens  $t_{tot}$  avec et sans *AMR* pour un nombre de niveaux fins différent (cf figure 3.20). La figure 3.20 bas montre que le temps  $t_{tot}$  avec *AMR* est égal à 75% de ce même temps sans *AMR*, ce qui rejoint le résultat obtenu sur le temps *CPU* de simulation avec trois niveaux.

Par contre, pour un niveau de raffinement (cf figure 3.20 haut), les courbes représentant  $t_{tot}$  en fonction du nombre d'itérations avec et sans *AMR* se coupent en un point d'intersection  $(NT_{int}, T_{totint})$ . Ceci montre qu'à partir du temps  $t_{int} = NT_{int} * \Delta t$  la méthode *AMR* devient plus coûteuse en temps *CPU* qu'une méthode classique. Notons  $t_{totcrit} = 0.75t_{totint}$  le temps critique à partir duquel on estime que la méthode *AMR* est trop coûteuse en temps *CPU* et  $NT_{crit}$  l'itération correspondant à  $t_{totcrit}$ . On observe dans le tableau que pour un niveau de raffinement, la méthode serait intéressante en terme de temps *CPU* si  $rgl < 0.53$ , pour deux niveaux, si  $rgl < 0.27$ . On ne peut donc pas relier directement les performances de la méthode *AMR* en terme de gain en temps *CPU* et en taille mémoire. Ceci nous amène à confirmer qu'un autre facteur rend l'*AMR* plus coûteuse en temps *CPU* et pas en taille mémoire. On pense donc immédiatement aux nombreuses allocations-désallocations de tableaux, indispensables à la gestion dynamique du raffinement-déraffinement et aux tests de connection des cellules *AMR*.

nombre de niveaux	$t_{raf}$	$t_{resol}$	$t_{total}$	$\frac{t_{raf}}{t_{total}}$	$\frac{t_{resol}}{t_{total}}$	temps <i>CPU</i> total
1	0.002499	0.01193	0.02001	0.1249	0.5962	522.1
2	0.04609	0.07379	0.1343	0.3432	0.5494	3495
3	0.4509	0.2476	0.7481	0.6027	0.3310	19460
4	3.4735	1.6411	6.4551	0.5381	0.2542	167940

Tableau 3.9 : Comparaison des temps *CPU* moyens à l'itération  $NT = 26000$  pour différents nombres de niveaux de raffinement.

D'après les résultats obtenus dans les tableaux 3.7, 3.8, 3.9, 3.10 et 3.11 on peut

maillage	$t_{resol}$	$t_{total}$	$\frac{t_{resol}}{t_{total}}$	temps CPU total
60 * 60	0.01156	0.0182	0.6352	475.3
180 * 180	0.1078	0.1207	0.8865	3144
540 * 540	0.9192	0.9932	0.9255	25860

Tableau 3.10 : Comparaison des temps CPU moyens à l'itération  $NT = 26000$  pour différents maillages sans AMR.

conclure que si le critère sur  $r_{gl}$  est vérifié (c'est-à-dire : si l'intérêt de la méthode AMR en terme de taille mémoire est confirmé), alors son efficacité en terme de gain en temps CPU est vérifiée. Attention, la réciproque est fausse!

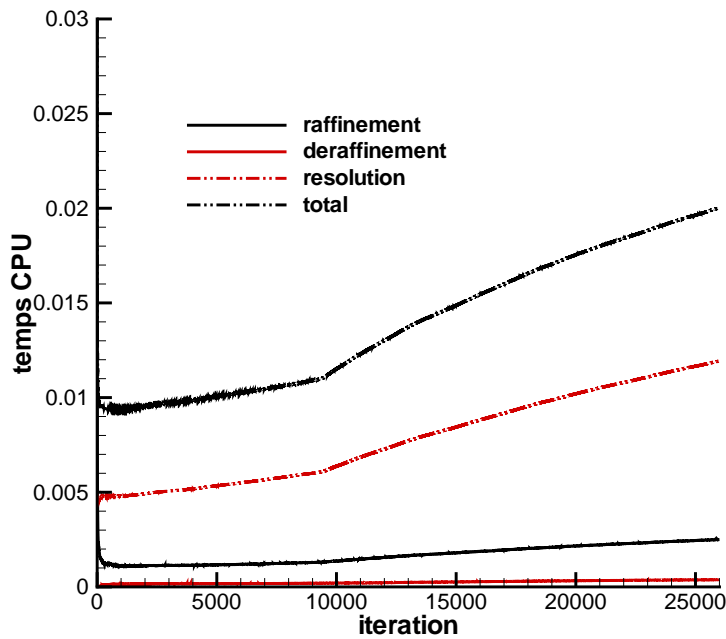


Figure 3.17 : Temps CPU moyens avec un niveau de raffinement.

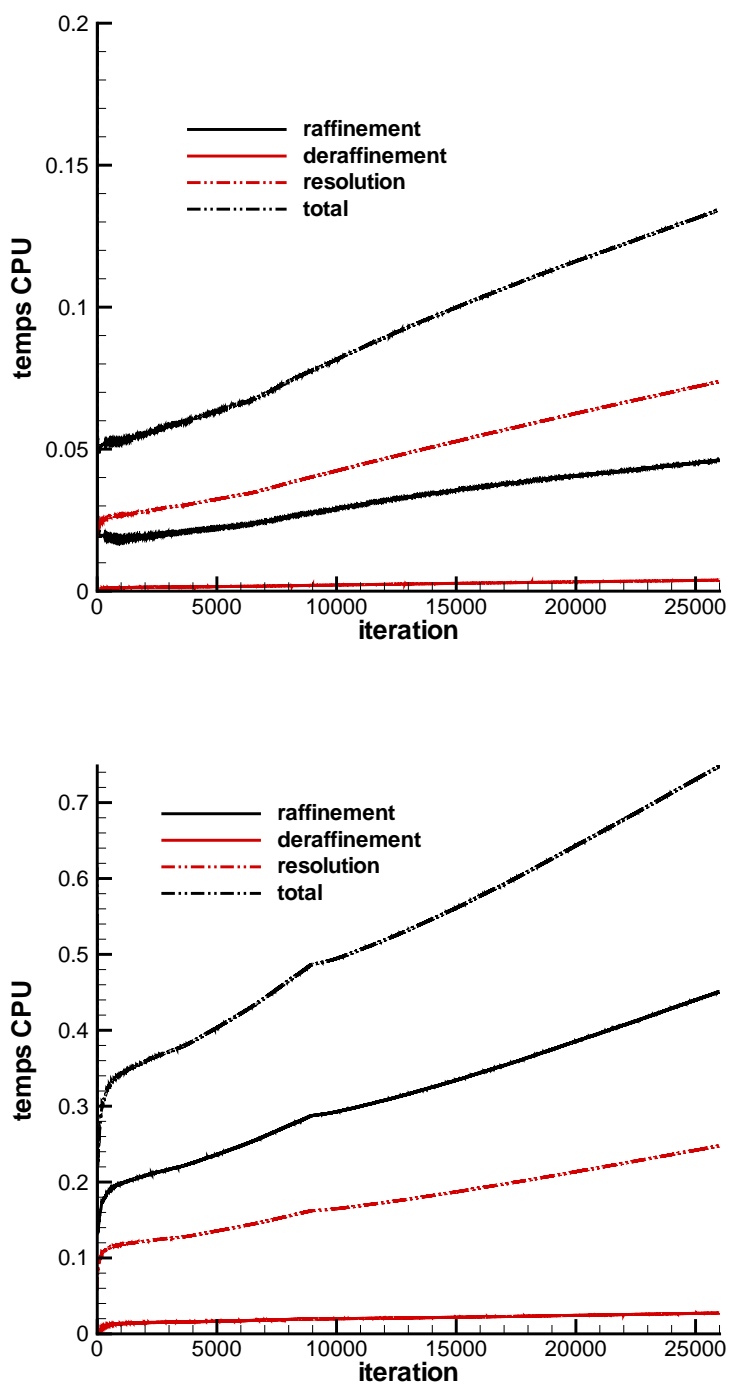


Figure 3.18 : Temps CPU moyens avec deux niveau fin en haut et trois en bas.

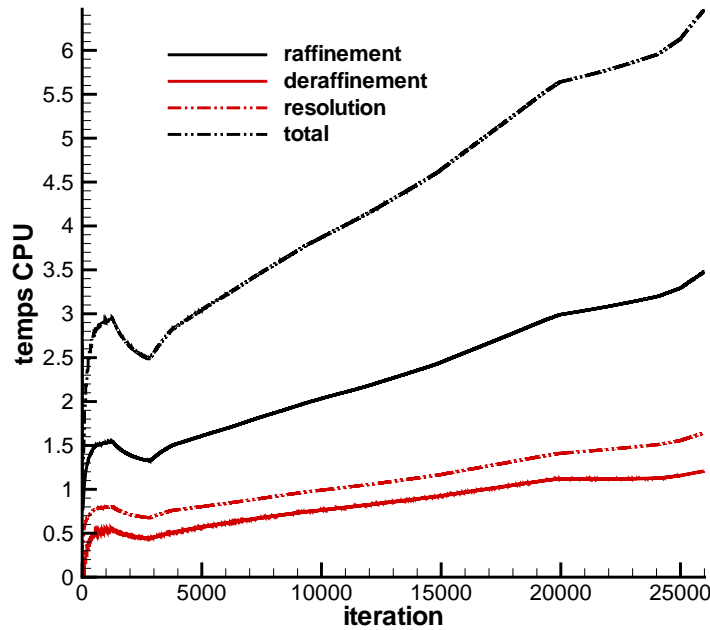


Figure 3.19 : Temps CPU moyens avec quatre niveaux de raffinement.

niveau fin	$NT_{int}$	$t_{totint}$	$NT_{crit}$	$t_{totcrit}$	$rgl$
1	21170	0.018	12769	0.0135	0.53
2	21766	0.12	12709	0.09	0.27

Tableau 3.11 : Comparaison des temps CPU moyens pour un et deux niveaux fins.

### Bilan

Après avoir analysé les performances de la méthode *AMR*, en termes de gain en taille mémoire et en temps *CPU*, il apparaît clairement que cette méthode offre un intérêt important quand le nombre de nœuds *AMR* à résoudre est inférieur à 32% du nombre de nœuds qu'il aurait fallu résoudre sans *AMR*, sur un maillage fin équivalent.

Sous cette condition, la méthode de raffinement assure aussi des gains en temps *CPU* satisfaisants lorsque la condition précédente est vérifiée, même si la gestion dynamique du raffinement est coûteuse.

On rappelle que seule l'équation de transport d'espèce a été résolue pour ce cas. Étant

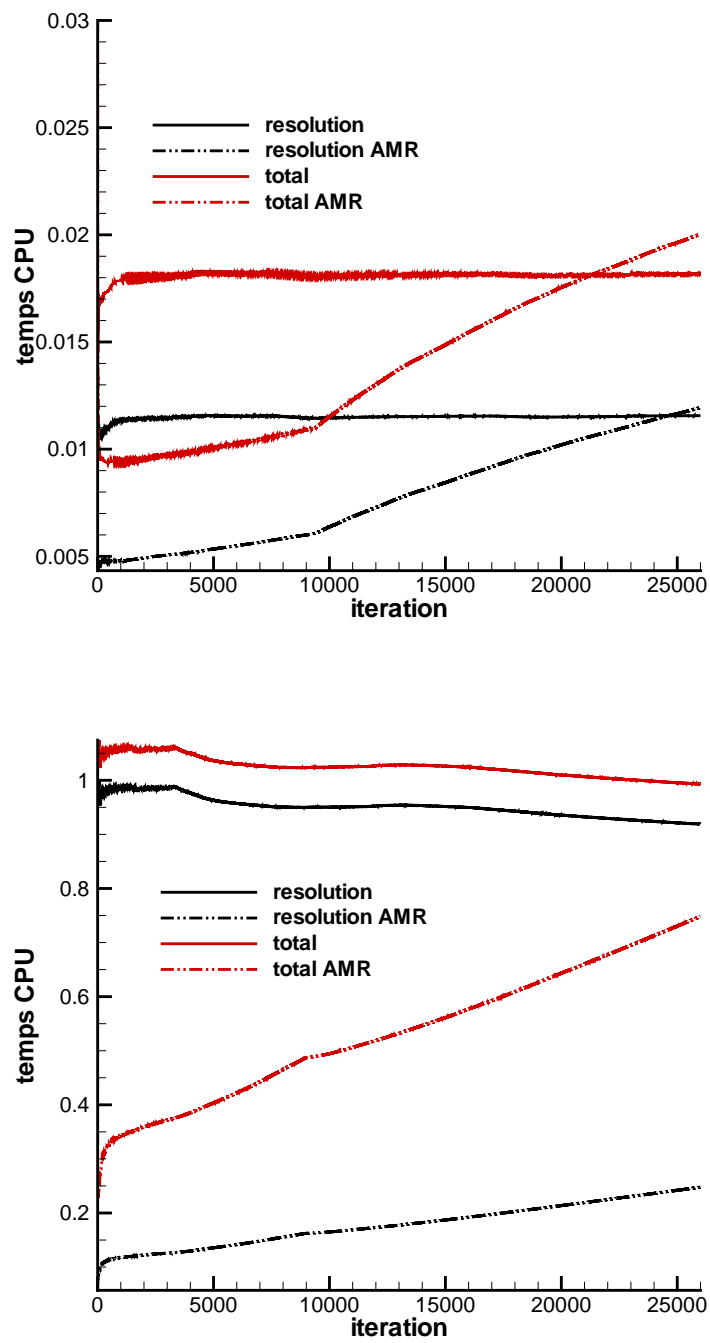


Figure 3.20 : Comparaison des temps CPU moyens avec un niveau de raffinement et sans AMR (haut) et avec trois niveaux fins et sans AMR (bas).



donné que le coût de cette méthode est directement liée au nombre de nœuds *AMR* résolu, il est évident que si on résout à la fois les équations de transport et de Navier-Stokes, ses performances en termes de gain en temps *CPU* seront très nettement accrues. Par contre, le nombre de nœuds *AMR* étant indépendant du nombre d'équations à résoudre, ses performances en terme de gain en taille mémoire resteront inchangées.

## 3.2 Etude d'une tache cisailée par un tourbillon dans une cavité 2D en diphasique

Ayant déjà évalué les performances de l'*AMR* en temps *CPU* et en taille mémoire sur un cas similaire en monophasique, on se contente ici d'estimer ses performances en terme de suivi d'interface. On se place, dans le cas de deux fluides non miscibles dans un domaine carré d'arête  $0.1m$ : le *fluide*<sub>1</sub> occupe initialement un disque de centre  $(x = 0.075m, z = 0.05m)$  et de rayon  $r = 0.015m$ , tandis que le *fluide*<sub>2</sub> occupe le reste du domaine (cf figure 3.1). On impose un champ de vitesse tournant d'intensité moyenne  $V_{moyen} = 0.1m.s^{-1}$  et de forme analytique (voir figure 3.1):

$$\begin{aligned} u(x, z) &= -V_{moyen} \cos\left(\frac{\pi(x - x_0)}{L}\right) \sin\left(\frac{\pi(z - z_0)}{L}\right) \\ w(x, z) &= V_{moyen} \sin\left(\frac{\pi(x - x_0)}{L}\right) \cos\left(\frac{\pi(z - z_0)}{L}\right) \end{aligned}$$

$(x_0 = 0.05m, z_0 = 0.05m)$  sont les coordonnées du centre du domaine.

### 3.2.1 Couplage *AMR* et *VOF* – *CIAM*

On cherche à montrer dans cette sous-section l'apport de la méthode *AMR* à l'approche *VOF* – *CIAM* (se référer à [ZALESKI] pour plus de détails). On va donc commencer par montrer les avantages et inconvénients de la méthode *VOF* – *CIAM*. On montre ensuite comment la méthode *AMR* peut accroître le nombre de cas pour lesquels l'approche *VOF* – *CIAM* peut être utilisée.

La méthode *VOF* – *CIAM* consiste à reconstruire l'interface géométriquement, ce qui permet de connaître la position de l'interface, au cours du temps, de manière très précise. L'inconvénient de cette méthode est que si l'interface est trop cisailée par rapport à un maillage donné (i.e. l'épaisseur du fluide est trop fine par rapport à la taille des mailles),

elle se fragmente (voir figure 3.21 sur le maillage  $G = 20 * 20$ ). On peut réduire cette fragmentation, jusqu'à la faire disparaître, quand on diminue la taille des mailles. C'est ce qui est observé sur la figure 3.21. Cependant, si ce cisaillement est trop important, au point de nécessiter un maillage dont le nombre de nœuds à résoudre dépasse la taille mémoire des machines mises à disposition, la méthode  $VOF - CIAM$  devient inutilisable. La tache cisailée de la figure 3.21 n'étant pas fragmentée pour un maillage  $G = 540 * 540$ ,

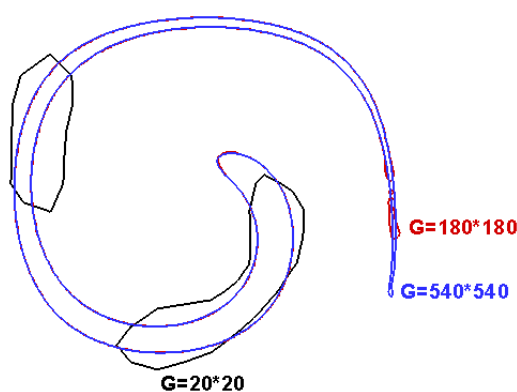


Figure 3.21 : Position de l'interface à  $t = 6.3s$  pour des maillages de pas d'espace différents.  $\Delta t = 0.0009s$ .

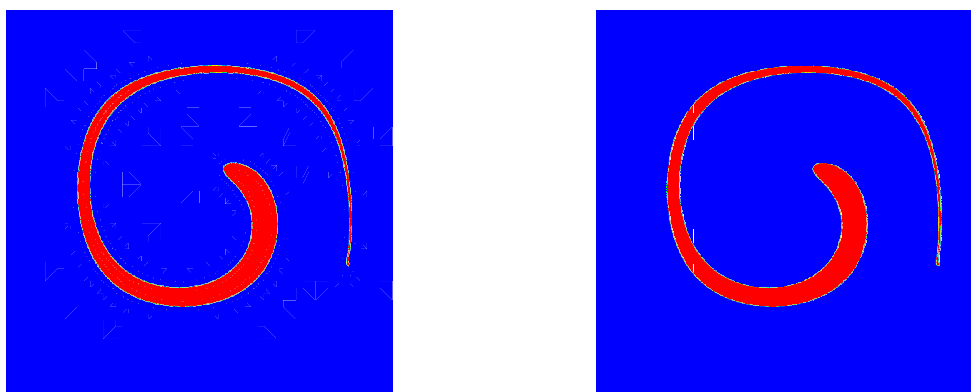


Figure 3.22 : Comparaison de la position de l'interface avec trois niveaux de raffinement (gauche) et sans AMR sur un maillage fin équivalent (droite) à  $t = 6.3s$ .

on décide de comparer cette solution à celle obtenue avec trois niveaux de raffinement en

partant d'un maillage grossier  $G_0 = 20 * 20$ . La vitesse analytique est imposée sur tous les niveaux de raffinement et les nœuds scalaires de limite de niveau sont traités avec une interpolation binaire. Au bout de  $t = 6.3s$ , on calcule une erreur  $\epsilon = 6 \cdot 10^{-14}$ , ce qui montre l'efficacité de l'*AMR* à obtenir une solution aussi précise que si on avait utilisé un maillage fin équivalent sans *AMR*, ce qui est visible sur les figures 3.22 et 3.23.

L'approche *AMR* ne prétend pas améliorer la méthode *VOF - CIAM*, mais seulement rendre accessible la finesse du maillage requise afin que la méthode *VOF - CIAM* reste efficace (pas de fragmentation d'interface).

Pour s'assurer que l'approche *AMR* se comporte de la même manière qu'une méthode sans raffinement, on effectue un deuxième test, qui consiste à imposer le champ de vitesses (3.1) pendant la première moitié du temps de simulation, puis à imposer à la tache cisailée un champ de vitesses opposé pendant la deuxième moitié du temps de la simulation (test de réversibilité). On compare les solutions obtenues avec et sans *AMR* sur une durée de  $t = 6.3s$  (cf figures 3.24, 3.25) puis de  $t = 12.6s$  (cf figures 3.26, 3.27).

On observe que les solutions se comportent de la même façon avec et sans *AMR*, à savoir une bonne réversibilité du phénomène quand la tache est cisailée pendant une courte durée (voir figure 3.25), car la tache retrouve sa forme initiale à  $t = 6.3s$ . Par contre, pour une simulation plus longue, la solution finale (à  $t = 12.6s$ ) ressemble à un disque déformé (cf figure 3.27). Pour le cas non raffiné, on retrouve des résultats similaires à ceux obtenus par [BREIL 01].

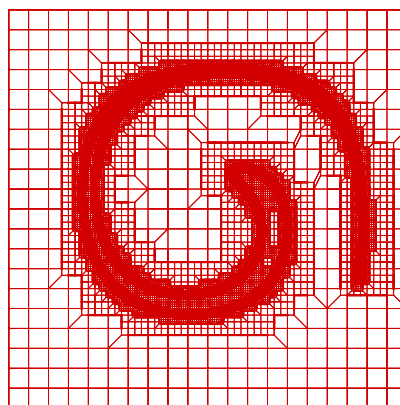


Figure 3.23 : Maillage utilisé pour suivre l'interface. Trois niveaux de raffinement,  
 $t = 6.3s$



Figure 3.24 : Comparaison de la position de l'interface avec trois niveaux de raffinement (gauche) et sans AMR sur un maillage fin équivalent (droite), sur un aller-retour de  $t = 6.3s$ .  $\Delta t = 0.0009s$ .



Figure 3.25 : Comparaison de la position de l'interface avec trois niveaux fins (gauche) et sans AMR sur un maillage fin équivalent (droite), sur un aller-retour de  $t = 6.3s$ .  $\Delta t = 0.0009s$ .

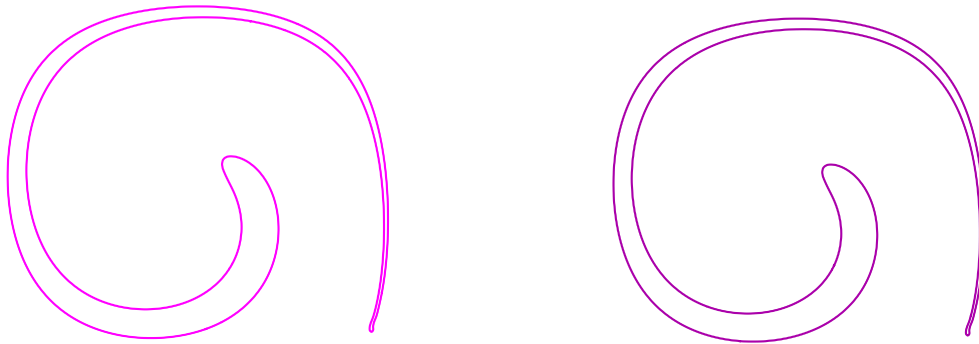


Figure 3.26 : Comparaison de la position de l'interface avec trois niveaux fins (gauche) et sans AMR sur un maillage fin équivalent (droite), sur un aller-retour de  $t = 12.6s$ .  
 $\Delta t = 0.0009s$ .



Figure 3.27 : Comparaison de la position de l'interface avec trois niveaux fins (gauche) et sans AMR sur un maillage fin équivalent (droite), sur un aller-retour de  $t = 12.6s$ .  
 $\Delta t = 0.0009s$ .

### 3.2.2 Couplage *AMR* et *VOF* – *TVD*

On cherche à montrer dans cette sous-section l'efficacité de la méthode *AMR* à réduire la diffusion numérique introduite par l'utilisation d'un schéma *TVD* pour transporter l'interface. Ayant déjà démontré les performances de cette approche dans le cas monophasique avec le transport d'un front de concentration, on se contente ici de montrer que les solutions obtenues avec cette méthode ont le même comportement qu'avec une approche classique. De la même manière que dans la sous-section 3.2.1, on impose au disque initial le champ de vitesses (3.1) pendant la première moitié du temps de simulation, puis le champ de vitesses inverse pendant la deuxième moitié du temps de simulation. On compare les solutions obtenues avec et sans *AMR* sur une durée de  $t = 6.3s$  (cf figures 3.28, 3.29) puis de  $t = 12.6s$  (cf figures 3.30, 3.31).

Comme dans le cas avec la méthode *VOF* – *CIAM*, on observe que les solutions se comportent de manière similaire avec et sans *AMR*, à savoir une bonne réversibilité du phénomène quand la tache est cisailée pendant une courte durée (voir figure 3.25), car la tache retrouve sa forme initiale à  $t = 6.3s$ . Par contre, pour une simulation plus longue, la forme du disque au temps final est déformé sur sa partie droite (cf figure 3.31).



Figure 3.28 : Comparaison de la position de l'interface avec trois niveaux fins (gauche) et sans *AMR* sur un maillage fin équivalent (droite), sur un aller-retour de  $t = 6.3s$ .

$$\Delta t = 0.0009s.$$



Figure 3.29 : Comparaison de la position de l'interface avec trois niveaux fins (gauche) et sans AMR sur un maillage fin équivalent (droite), sur un aller-retour de  $t = 6.3s$ .  
 $\Delta t = 0.0009s$ .

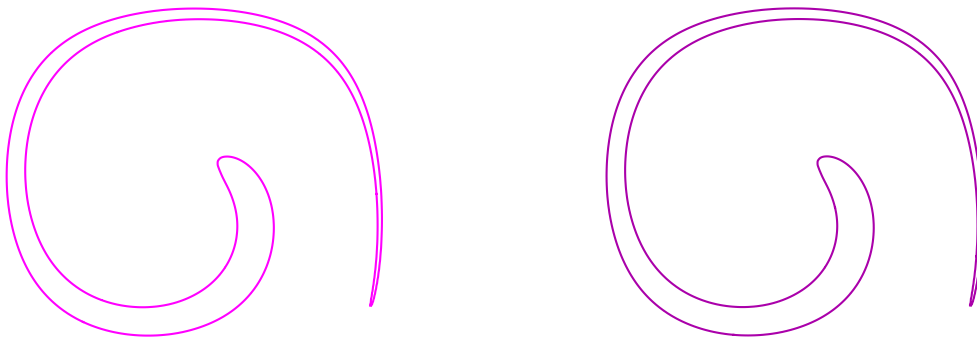


Figure 3.30 : Comparaison de la position de l'interface avec trois niveaux fins (gauche) et sans AMR sur un maillage fin équivalent (droite), sur un aller-retour de  $t = 12.6s$ .  
 $\Delta t = 0.0009s$ .



Figure 3.31 : *Comparaison de la position de l'interface avec trois niveaux fins (gauche) et sans AMR sur un maillage fin équivalent (droite), sur un aller-retour de  $t = 12.6s$ .  $\Delta t = 0.0009s$ .*

### 3.3 Etude d'un cas de thermique en 2D : cas de Whitaker

On considère un fluide incompressible dans un domaine rectangulaire de hauteur  $H$  et de longueur  $b$ . Les bords gauche, droit et inférieur sont maintenus à une température constante  $T_0$ , tandis que le bord supérieur est maintenu à une température  $T_f(x)$  de classe  $C^1$ . La vitesse est nulle dans tout le domaine. L'équation de conservation de l'énergie (équation (1.16)) se réduit donc aux termes de variation temporelle et de diffusion (cf [WHITAKER 83]). La figure 3.32 illustre le cas traité. On s'intéresse dans cet exemple au régime stationnaire. On impose sur le bord supérieur une température  $T_f$  de classe  $C^1$ , comme cela est décrit dans l'annexe A.2. On a représenté sur la figure (3.33) la répartition analytique de la température dans le domaine sur un maillage  $G = 48 * 48$ . On a fixé  $T_0 = 10^\circ C$  et  $T_1 = 100^\circ C$ . La forme analytique de la température est détaillée dans l'annexe A.2.

Dans un premier temps, on cherche à estimer l'influence de la méthode *AMR* sur la précision d'une solution. On désire ensuite déterminer comment le traitement des limites sur les niveaux de raffinement agit sur la qualité des solutions.



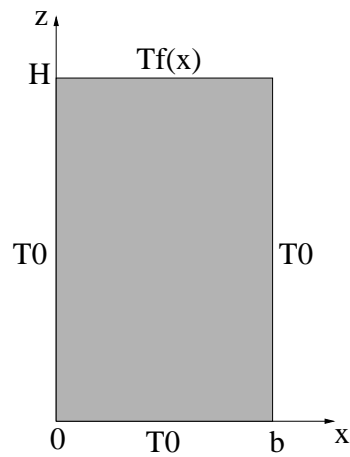


Figure 3.32 : Conduction 2D dans un rectangle dont les bords sont chauffés à des températures différentes.

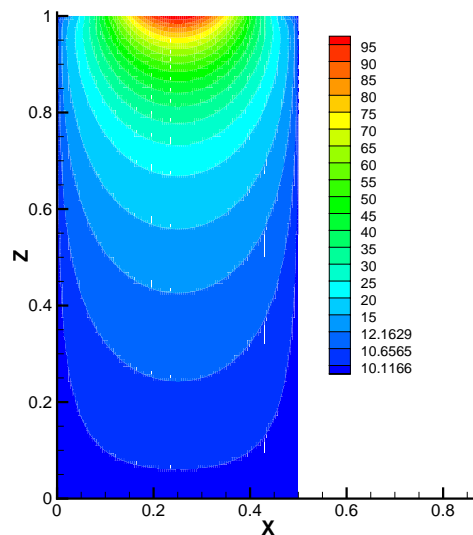


Figure 3.33 : Répartition analytique de la température sur un maillage  $G = 48 * 48$ .

### 3.3.1 Traitement des limites scalaires

On a exposé dans la sous-section 2.5 différentes techniques de traitement des conditions aux limites pour les niveaux fins. On désire montrer ici leur influence sur la précision de la solution via le cas de Whitaker.

Trois cas sont alors comparés. Dans le premier, la répartition de la température dans le domaine est obtenue sur un maillage  $G = 48 * 48$  sans *AMR*. Dans le deuxième cas,

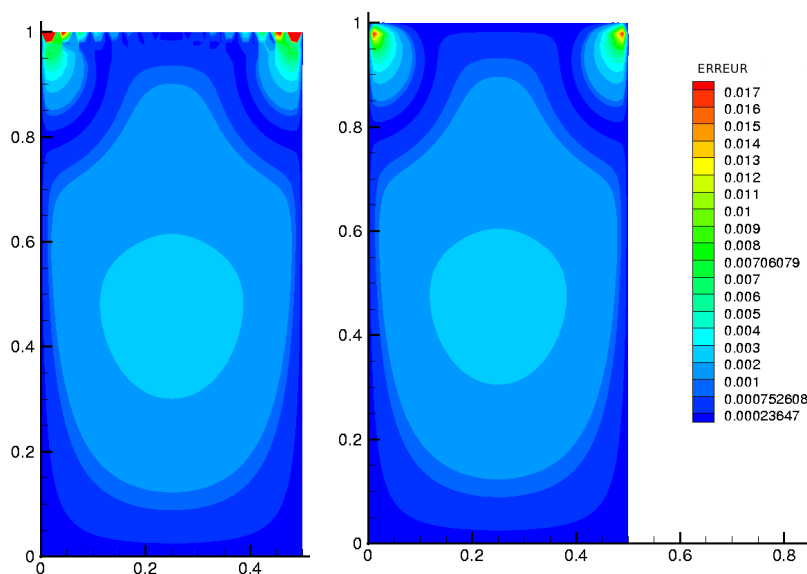


Figure 3.34 : Répartition de l'erreur  $e_r$  lorsque la température est imposée sur les bords du domaine à droite et interpolée à gauche. Maillage  $G_0 = 16 * 16$  avec un niveau fin.

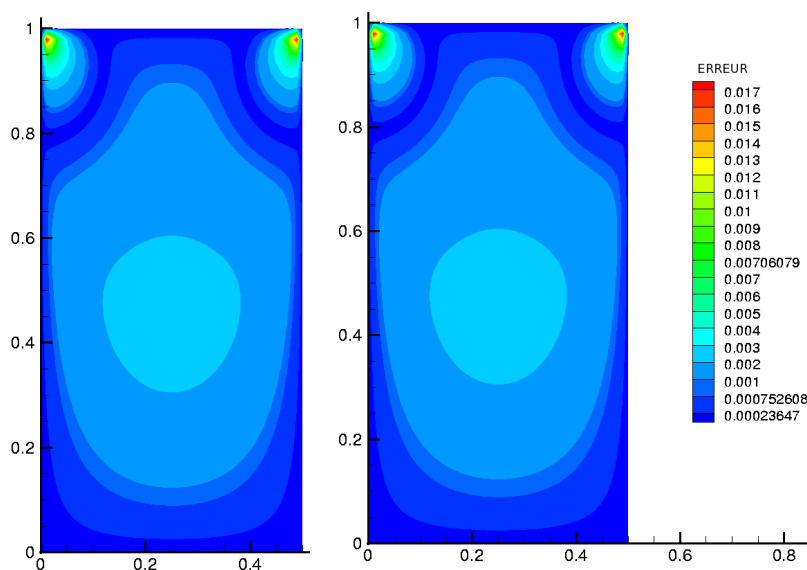


Figure 3.35 : Répartition de l'erreur  $e_r$  lorsque la température est imposée sur les bords du domaine avec 1 niveau d'AMR ( $G_0 = 16 * 16$ ) à droite et sans AMR ( $G = 48 * 48$ ) à gauche.

on calcule la répartition de la température sur un niveau de raffinement, à partir d'un maillage  $G_0 = 16 * 16$ , qui s'étend sur tout le domaine et les conditions aux limites

sont traitées par interpolation  $Q_2$ . Le troisième cas est identique au deuxième excepté le traitement des conditions aux limites; celles-ci sont imposées sur le niveau  $G_1$  par une méthode de pénalisation comme cela est fait sur le niveau grossier  $G_0$ . Pour chaque cas, on représente la répartition de l'erreur  $e_r(x, z), \forall (x, z) \in \Omega$ . Celle-ci est déterminée pour tout couple  $(x, z) \in \Omega$ , par :

$$e_r(x, z) = \frac{|T_{num}(x, z) - T_{an}(x, z)|}{|T_{an}(x, z)|}$$

avec  $T_{num}$  la température numérique et  $T_{an}$  celle analytique.

La figure 3.34 compare les deux cas avec un niveau d'AMR. A gauche, la température sur les bords du domaine est interpolée alors qu'à droite, on impose les mêmes conditions aux limites sur les deux niveaux. On note sur la figure 3.34 gauche, une erreur plus importante sur le bord supérieur (oscillations) que celle sur la figure de droite. La figure 3.35 gauche représente la répartition de l'erreur dans le cas AMR avec conditions aux limites imposées sur le niveau  $G_1$ . Celle de droite montre le cas sans AMR. Comme on pouvait s'y attendre, la répartition de l'erreur est la même dans les deux cas.

Les trois cas que l'on vient d'exposer montrent que lorsque les températures imposées sur les bords du domaine sont de classe  $C^1$ , l'imposition des conditions aux limites sur le niveau fin est plus efficace que l'interpolation. Plus précisément, on a remarqué que l'interpolation introduit des erreurs supplémentaires sur le bord supérieur. Par contre, si on impose les conditions aux limites de la même façon sur les niveaux fin et grossier (i.e. par la méthode de pénalisation décrite dans la sous-section 1.2.5), on observe une conservation de la précision de la solution; cette approche est utilisée dans la suite du document.

### 3.3.2 Méthode AMR et précision des solutions

Notons  $T_{an}$ , la température analytique,  $T_{num1}$ , la température calculée sans AMR sur un maillage  $G$ , et  $T_{num2}$ , celle obtenue par la méthode AMR. On souhaite déterminer sa précision. Pour cela, on compare les erreurs obtenues entre les solutions  $T_{num1}$  et  $T_{an}$  à celles obtenues entre les solutions  $T_{num2}$  et  $T_{an}$ .

Soient  $L_x$  (respectivement  $L_z$ ) la longueur du domaine suivant  $x$  (respectivement  $z$ ),  $N_x$  (respectivement  $N_z$ ) le nombre de nœuds suivant  $x$  (respectivement  $z$ ) et  $N = N_x N_z$  le nombre de points. On travaille à pas d'espace constant. Posons  $\Delta x$  (respectivement  $\Delta z$ ) le pas d'espace suivant  $x$  (respectivement  $z$ ), défini par  $\Delta x = \frac{L_x}{N_x}$  (respectivement

$\Delta z = \frac{L_z}{N_z}$ ). On définit  $h$  par  $h = \sqrt{\Delta x \Delta z}$ . On note  $T_{num}$  la solution numérique représentant  $T_{num1}$  ou  $T_{num2}$ .

L'erreur  $\epsilon$  est définie par :

$$\begin{aligned} \epsilon &= \frac{\|T_{num} - T_{an}\|_2}{\|T_{an}\|_2} \\ \epsilon &= \frac{[\int_V |T_{num}(x, z) - T_{an}(x, z)|^2 dx dz]^{\frac{1}{2}}}{[\int_V |T_{an}(x, z)|^2 dx dz]^{\frac{1}{2}}} \\ \epsilon &\sim \frac{[\sum_{i,j} |T_{num}(x_i, z_j) - T_{an}(x_i, z_j)|^2 \Delta x_i \Delta z_j]^{\frac{1}{2}}}{[\sum_{i,j} |T_{an}(x_i, z_j)|^2 \Delta x_i \Delta z_j]^{\frac{1}{2}}}, i \in [1, N_x], j \in [1, N_z] \end{aligned} \quad (3.3)$$

La démarche consiste à faire un test de convergence en maillage pour le cas sans *AMR* (tableau 3.12), pour celui avec un niveau d'*AMR* (tableau 3.13) et pour celui avec deux niveaux d'*AMR* (tableau 3.14), c'est à dire, calculer des erreurs  $\epsilon$  propres à chaque maillage  $G = (N_x - 1)(N_z - 1)$ , pour chaque cas. Sachant que l'erreur  $\epsilon$  évolue suivant la relation  $\epsilon = k(\frac{L}{\sqrt{N}})^q$ , il suffit de tracer la courbe  $\ln(\epsilon) = \ln(k) - q \ln(\frac{L}{\sqrt{N}})$ , où  $k$  est une constante,  $L = \sqrt{L_x L_z}$  et  $q$  l'ordre de convergence, pour déterminer les ordres locaux de convergence. Ceux-ci tendent vers un ordre limite.

On précise que ce cas n'est pas propice à la valorisation de la méthode de raffinement de maillage concernant l'ordre de convergence, l'optimisation du temps *CPU* et de la taille mémoire car on raffine sur tout le domaine. Étant donné qu'on calcule la solution sur tous les niveaux fins, on utilise plus de nœuds avec que sans *AMR* pour une solution de précision similaire. On ne peut donc ni espérer une amélioration de l'ordre de convergence, ni un gain en temps *CPU* ou taille mémoire via la méthode *AMR* dans ce cas. On cherche seulement à montrer que cette méthode conserve la précision des solutions et l'ordre de convergence.

On montre sur la figure 3.36 les ordres de convergence successifs obtenus sans *AMR* et calculés à partir des résultats du tableau 3.12. On observe une convergence en maillage quasi linéaire tendant vers l'ordre limite  $q = 2$ . Ce résultat est en concordance avec l'ordre théorique des schémas utilisés.

On calcule maintenant les ordres de convergence obtenus avec l'approche *AMR*, pour savoir si la méthode ne dégrade pas l'ordre de convergence des schémas. On commence donc par évaluer les ordres de convergences successifs avec un niveau fin et un maillage grossier de départ de pas d'espace variable. La figure 3.37 (s'appuyant des résultats du tableau 3.13) permet de noter une convergence quasi linéaire et un ordre de convergence

tendant vers  $q = 2$ . On réitère cette démarche avec deux niveaux d'AMR. On obtient de nouveau un ordre limite  $q = 2$  (voir figure 3.37 et tableau 3.14), ce qui est très satisfaisant. Ainsi, au vu des résultats obtenus sur un et deux niveaux fins, on peut conclure que cette approche conserve l'ordre de convergence des schémas.

maillage	N	$\epsilon$
8 * 8	81	1.2998 $10^{-2}$
16 * 16	289	4.2915 $10^{-3}$
32 * 32	1089	1.2316 $10^{-3}$
64 * 64	4225	3.2712 $10^{-4}$
128 * 128	16641	8.3961 $10^{-5}$
256 * 256	65536	2.1340 $10^{-5}$

Tableau 3.12 : Etude de convergence en maillage sans AMR pour le cas de Whitaker

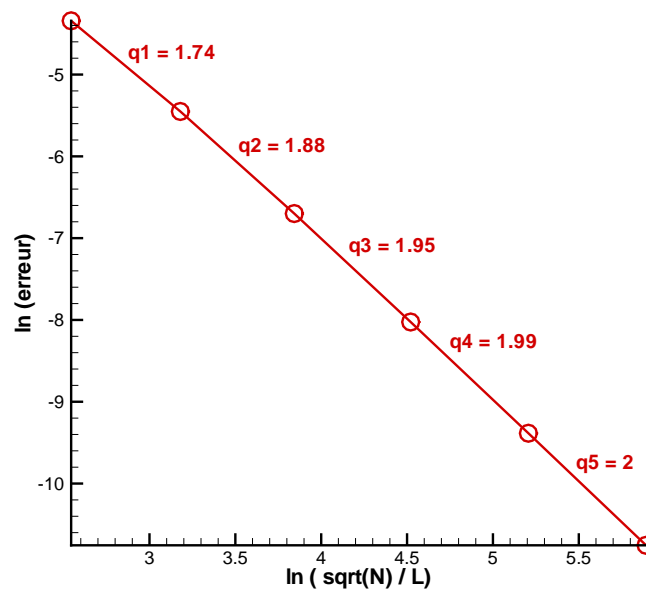


Figure 3.36 : Ordres de convergence successifs sans AMR. Les données proviennent du tableau 3.12.

maillage $G_0$	maillage $G_1$	$N$	$\epsilon$
$4 * 4$	$12 * 12$	194	$6.933 \cdot 10^{-3}$
$8 * 8$	$24 * 24$	706	$2.094 \cdot 10^{-3}$
$12 * 12$	$36 * 36$	1538	$9.871 \cdot 10^{-4}$
$24 * 24$	$72 * 72$	5954	$2.601 \cdot 10^{-4}$
$40 * 40$	$120 * 120$	16322	$9.537 \cdot 10^{-5}$
$70 * 70$	$210 * 210$	49562	$3.147 \cdot 10^{-5}$

Tableau 3.13 : *Etude de convergence en maillage avec la méthode AMR et un niveau fin pour le cas de Whitaker*

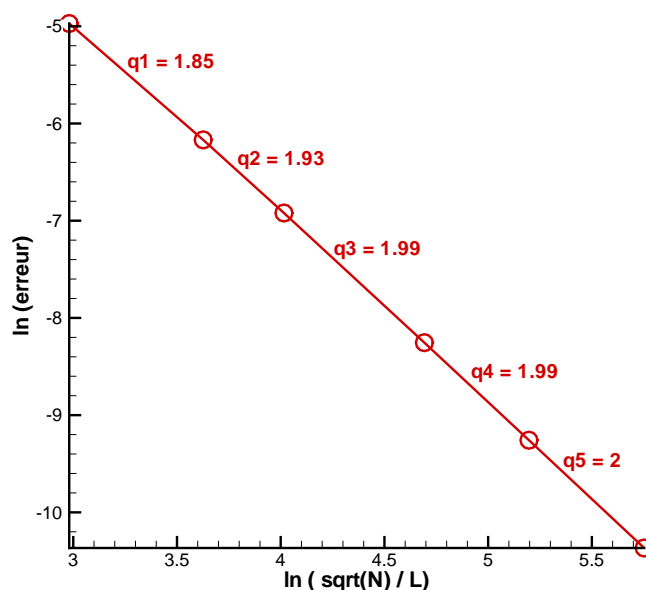


Figure 3.37 : *Convergence en maillage avec un niveau d'AMR. Les données proviennent du tableau 3.13.*

On décide de faire une dernière étude de convergence qui consiste à prendre un maillage grossier fixe ( $G_0 = 4 * 4$ ) et d'augmenter le nombre de niveaux de raffinement pour réduire le pas d'espace sur le niveau *AMR* le plus fin. Le tableau 3.15 montre l'erreur  $\epsilon$  sur le niveau *AMR* le plus fin et le nombre de nœuds résolus correspondants. La figure 3.39, montre une convergence en maillage quasi linéaire. On estime l'ordre limite à  $q = 2$ . La

maillage $G_0$	maillage $G_2$	$N$	$\epsilon$
4 * 4	36 * 36	1563	$9.87 \cdot 10^{-4}$
8 * 8	72 * 72	6035	$2.601 \cdot 10^{-4}$
12 * 12	108 * 108	13419	$1.1740 \cdot 10^{-4}$
16 * 16	144 * 144	23715	$6.651 \cdot 10^{-5}$
24 * 24	216 * 216	53043	$2.980 \cdot 10^{-5}$

Tableau 3.14 : *Etude de convergence en maillage avec la méthode AMR et deux niveaux fins pour le cas de Whitaker*

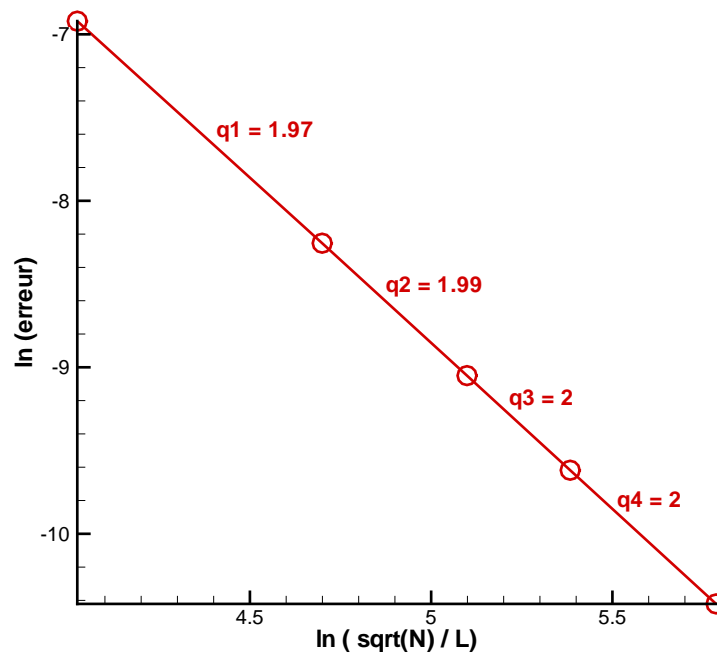


Figure 3.38 : *Convergence en maillage avec deux niveaux fins. Les données proviennent du tableau 3.14.*

méthode *AMR* permet donc de conserver l'ordre de convergence des schémas.

Niveau le plus fin	$N$	$\epsilon$
$G_1$	194	$6.933110^{-3}$
$G_2$	1563	$9.87 \cdot 10^{-4}$
$G_3$	13444	$1.174 \cdot 10^{-4}$
$G_4$	119069	$1.328 \cdot 10^{-5}$

Tableau 3.15 : *Etude de convergence en maillage avec la méthode AMR pour un maillage grossier  $G_0 = 4 * 4$  et un nombre de niveaux fins variable, pour le cas de Whitaker*

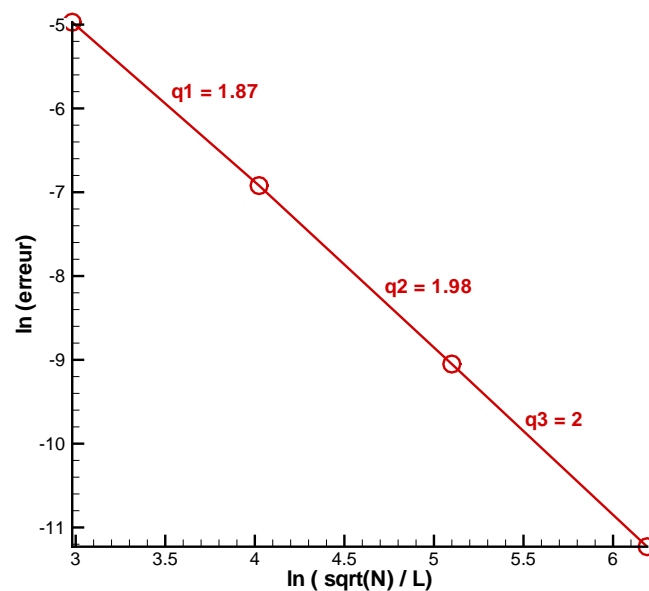


Figure 3.39 : *Convergence de l'AMR en augmentant le nombre de niveaux. Les données proviennent du tableau 3.15.*

### 3.3.3 Choix des opérateurs de restriction et de prolongement

On cherche les opérateurs de restriction et de prolongement les mieux adaptés au cas de Whitaker (i.e. cas pour lequel les gradients sont situés dans l'ensemble du domaine  $\Omega$ ), en ce qui concerne la conservation de l'énergie et l'obtention de résultats satisfaisants en termes d'erreurs.



### Opérateurs de restriction

On souhaite tester deux opérateurs de restriction, à savoir l'injection conservative (*FWCV*) et l'injection directe pour savoir lequel est le mieux adapté à ce cas test. Notons  $LP$  un nœud de niveau  $G_L$  ayant généré une cellule *AMR* (qui est constituée de  $(RAF + 2)(RAF + 2)$  fils).

Dans un premier temps, on raffine l'ensemble du domaine avec un niveau d'*AMR*, en partant d'un maillage grossier  $G_0 = 16 * 16$ . On remarque sur le tableau 3.16 que l'injection conservative dégrade la solution sur le niveau grossier. L'injection directe améliore la répartition de la température sur le maillage grossier car l'erreur  $\epsilon$  est divisée par 9 par rapport au cas sans injection. Donc l'injection directe semble plus adaptée pour améliorer la précision des solutions. La figure 3.40 vient confirmer le fait que l'injection directe paraît appropriée à ce cas particulier en terme d'erreur. Mais qu'en est-il de la conservation de l'énergie?

maillage	$\epsilon$ avec injection <i>FWCV</i>	$\epsilon$ avec injection directe	$\epsilon$ sans injection
$G_0$	$3.8 \cdot 10^{-2}$	$4.97 \cdot 10^{-4}$	$4.29 \cdot 10^{-3}$
$G_1$	$5.7 \cdot 10^{-4}$	$5.7 \cdot 10^{-4}$	$5.7 \cdot 10^{-4}$

Tableau 3.16 : Comparaison des erreurs entre la solution numérique et analytique, avec et sans injection. Maillage  $G_0 = 16 * 16$ , un niveau fin sur la totalité du domaine  $\Omega$ .

On calcule maintenant l'erreur faite sur la conservation de l'énergie lorsque le régime stationnaire est atteint. Connaissant la température analytique  $T_{an}$  et numérique  $T_{num}$ , on peut calculer l'énergie analytique  $E_{an}$ ,

$$E_{an} = \int_{\Omega} \rho c T_{an}(x, z) dx dz$$

et la comparer à l'énergie  $E_{num}$ , obtenue numériquement. Pour obtenir une valeur de  $E_{an}$  la plus précise possible, on la calcule sur un maillage fin,  $G = 400 * 400$ . D'après les résultats du tableau 3.17, on calcule une erreur entre  $E_{an}$  et  $E_{num}$  de 0.156% pour l'injection conservative et de 0.772% pour l'injection directe. Ceci montre que dans le cas particulier de Whitaker, les deux injections donnent des résultats corrects car l'énergie est assez bien conservée dans les deux cas : l'erreur ne dépasse pas 1%. Cependant, on commet

une erreur sur la conservation de l'énergie, cinq fois plus élevée avec l'injection directe qu'avec l'injection conservative. Il est donc dangereux d'utiliser l'injection directe pour un cas plus compliqué à résoudre que celui de Whitaker. De plus, on utilise une méthode de volumes finis qui est basée sur la conservation de l'énergie. Il est donc recommandé d'utiliser une injection conservative .

	$E_{num}$ , injection <i>FWCV</i>	$E_{num}$ , injection directe	$E_{an}$ sur $G = 400 * 400$
$maillageG_0$	8.9848	9.0033	8.9734
$\epsilon$	0.156 %	0.772 %	0 %

Tableau 3.17 : Comparaison des énergies avec injection à l'énergie analytique. Maillage  $G_0 = 16 * 16$ , un niveau de raffinement sur la totalité du domaine  $\Omega$ .

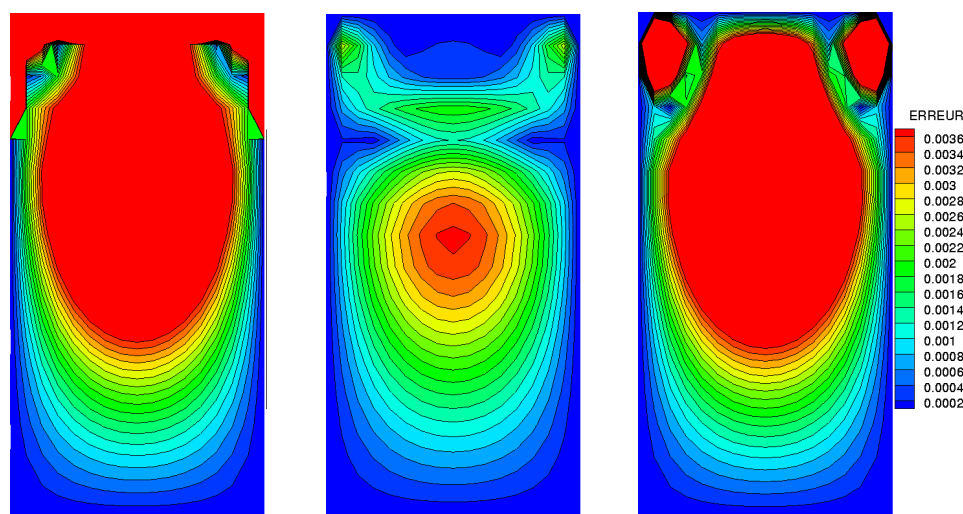


Figure 3.40 : Erreurs entre la solution analytique et numérique sur le niveau grossier  $G_0 = 16 * 16$  avec une injection *FWCV* (gauche), directe (centre) sans injection (droite) lorsque l'ensemble du domaine est raffiné.

### Opérateurs de prolongement

On s'intéresse maintenant à l'impact que peut avoir l'injection *FWCV* sur l'interpolation non conservative ( $Q_1$ ,  $Q_2$  ou  $Q_3$ ). Pour cela, on raffine une partie du domaine  $\Omega$ ,

soit la zone  $[0, 0.5] \times [0.8, 1]$  (voir figure 3.41). On étudie le comportement de l'erreur sur le niveau grossier lorsque différents types d'opérateurs de prolongement et restriction sont testés.

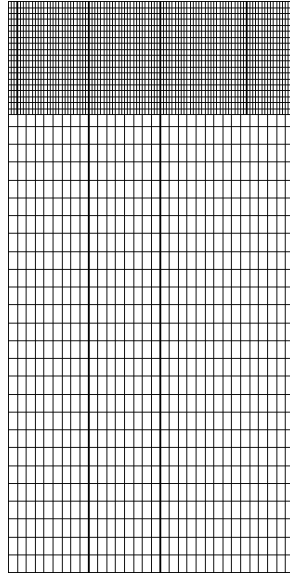


Figure 3.41 : *Maillage raffiné sur la zone  $[0, 0.5] \times [0.8, 1]$ .*

Pour comprendre l'influence que peut avoir la combinaison d'une interpolation non conservative et d'une injection conservative sur la répartition de la température, on considère une cellule *AMR* appartenant au niveau  $G_L$ , générée par le nœud père *LP* et dont la partie inférieure est interpolée (cf figure 3.42).

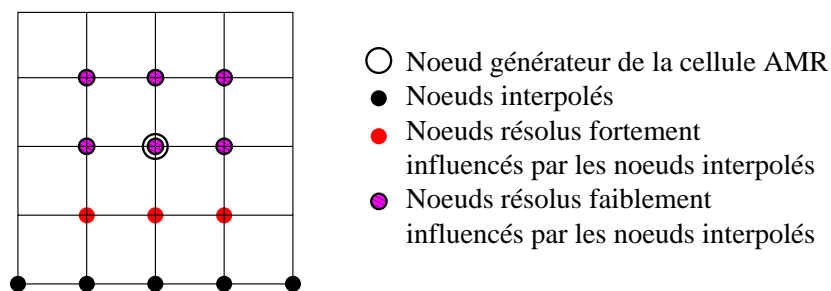


Figure 3.42 : *Influence de l'interpolation  $Q_2$  sur les nœuds résolus par discrétisation.*

Si on utilise une injection conservative (*FWCV*), les nœuds intérieurs à la cellule *AMR*

interviennent pour imposer au nœud  $LP$  une nouvelle valeur. Parmi ces nœuds fins, certains ont un voisin interpolé, ils sont donc fortement influencés par l'interpolation. Cette dernière étant non conservative, elle donne des solutions moins bonnes que si l'équation de conservation de l'énergie discrétisée avait été résolue. On impose alors au nœud  $LP$  une valeur influencée par l'interpolation. Or le nœud  $LP$  intervenant dans l'interpolation des nœuds de la limite inférieure de la cellule  $AMR$ , les valeurs en ces nœuds (interpolés) sont à leur tour dégradées. Celui-ci dégrade à son tour la valeur injectée au nœud  $LP$ . De proche en proche, la dégradation se propage vers l'intérieur du niveau  $AMR$  (voir la propagation de l'erreur avec une interpolation  $Q_2$  sur la figure 3.43).

Cette propagation est moins prononcée en utilisant une injection directe car la valeur

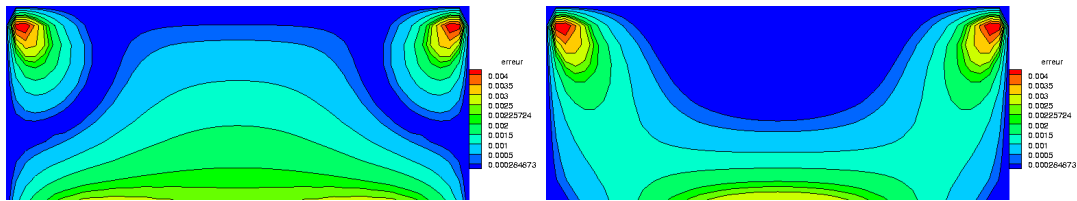


Figure 3.43 : Répartition de l'erreur sur le niveau fin avec interpolation  $Q_2$  et injection conservative sur la gauche puis injection directe sur la droite.

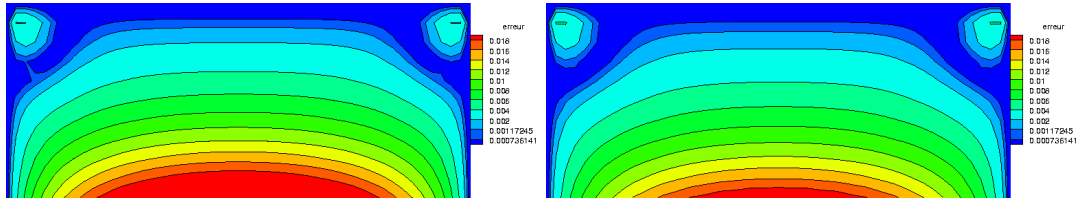


Figure 3.44 : Répartition de l'erreur sur le niveau fin avec interpolation  $Q_1$  et injection conservative sur la gauche puis injection directe sur la droite.

injectée au nœud  $LP$  dépend faiblement de l'interpolation (ceci peut être observé sur la figure (3.43) de droite). Le phénomène de propagation lié à la combinaison de l'injection conservative et de l'interpolation  $Q_2$  est beaucoup moins marqué avec une interpolation  $Q_1$ , car très peu de nœuds intervenant dans l'interpolation subissent une injection  $FWCV$  (voir figure 3.44).

Notons  $\epsilon_{G_1}$  (respectivement  $\epsilon_{G_0}$ ) l'erreur obtenue sur le niveau  $G_1$  ( $G_0$ ) et  $\epsilon_{composite}$  celle obtenue en ne comptabilisant que les nœuds résolus par discrétisation ou pénalisation

Injection-Interpolation	$\epsilon_{G_1}$	$\epsilon_{G_0}$	$\epsilon_{composite}$	$E_{num}$ sur $G_0$
$FWCV - Q_1$	$6.92 \cdot 10^{-3}$	$3.88 \cdot 10^{-2}$	$7.145 \cdot 10^{-3}$	9.0369
$FWCV - Q_2$	$9.61 \cdot 10^{-4}$	$3.8 \cdot 10^{-2}$	$1.29 \cdot 10^{-3}$	8.9937
$directe - Q_1$	$5.57 \cdot 10^{-3}$	$5.92 \cdot 10^{-3}$	$5.714 \cdot 10^{-3}$	9.049
$directe - Q_2$	$7.41 \cdot 10^{-4}$	$1.01 \cdot 10^{-3}$	$8.06 \cdot 10^{-4}$	9.01

Tableau 3.18 : Comparaison des erreurs et de l'énergie pour différents opérateurs de restriction et de prolongement. Maillage  $G_0 = 16 * 16$ , un niveau de raffinement sur la zone bornée  $[0, 0.5] \times [0.8, 1]$ .

(nœuds des limites physiques de  $\Omega$ ) et n'ayant pas généré de cellules *AMR*. On a rassemblé dans le tableau 3.18 ces erreurs ainsi que l'énergie numérique obtenue sur le niveau grossier.

On observe que pour une même injection, l'interpolation  $Q_2$  donne de meilleurs résultats en termes d'erreur et de conservation d'énergie que l'interpolation  $Q_1$ . Si on considère les cas avec une interpolation  $Q_2$ , on note que l'injection conservative donne des erreurs un peu plus élevées que celles obtenues avec une injection directe, tout en restant satisfaisantes. Par contre, l'injection *FWCV* permet de mieux conserver l'énergie. En effet, on calcule une erreur sur l'énergie de 0.23% pour l'injection *FWCV* contre 0.36% pour l'injection directe. Compte tenu des résultats du tableau 3.18, on conclut que l'utilisation d'une injection *FWCV* combinée avec une interpolation  $Q_2$  semble être le choix le mieux adapté au cas de Whitaker. Cependant, le phénomène de propagation des erreurs (dû aux interpolations) inhérent à ce choix reste préoccupant.

Quelques propositions peuvent être faites pour éviter ce problème :

- Soit l'effet de l'interpolation devient négligeable quand le nombre de nœuds résolus par discrétisation de l'équation de conservation d'énergie, devient important devant celui des nœuds résolus par interpolation.
- Soit on utilise des interpolations conservatives comme cela est déjà fait pour les méthodes multigrilles adaptatives (cf [TROTTEMBERG 01], [WASHIO 00]), la décomposition de domaine (cf [CADAFALCH 99]) ou le multibloc (cf [CADAFALCH 98]). Cette méthode consiste à faire un premier bilan de flux composite sur chaque volume de contrôle, de niveau  $G_{L+1}$ , dont au

moins une des faces est à l'interface  $G_{L+1} - G_L$ . Ces faces n'étant pas conjointes à celles du niveau  $G_L$ , on détermine le flux composite via un deuxième bilan fait sur un volume de contrôle du niveau  $G_{L+1}$  qui correspond au premier volume de contrôle dont les dimensions ont été prolongées jusqu'à ce que ses faces (seulement celles sur l'interface des deux niveaux) soient contiguës à celles des volumes de contrôle du niveau  $G_L$  (pour plus de détails voir [TROTTEBERG 01]).

- Soit on utilise une méthode *FIC* (Flux Interface Conservation), proposée par [ANGOT 92], qui consiste à faire des bilans de flux composites (flux provenant de deux niveaux  $G_L$  ou  $G_{L+1}$  suivant que la face du volume de contrôle est située à l'interface des deux niveaux ou non) sur les volumes de contrôles des nœuds situés à l'interface de deux niveaux.

Dans notre cas, la somme (au sens de l'union) des volumes de contrôle de chaque cellule *AMR* de niveau  $G_{L+1}$  est confondue avec celui du père de niveau  $G_L$ . Ainsi les faces des volumes de contrôles de niveau  $G_{L+1}$  appartiennent à l'interface  $G_{L+1} - G_L$  sont contiguës à celles des volumes de contrôle de niveau  $G_L$  appartenant à l'interface  $G_{L+1} - G_L$ . Donc l'interpolation conservative proposée par [TROTTEBERG 01] revient ici à une méthode *FIC* proposée par [ANGOT 92].

On décide donc de comparer l'opérateur de prolongement *FIC* à l'interpolation  $Q_2$ . Pour cela, on calcule les erreurs  $\epsilon_{G_1}$ ,  $\epsilon_{G_0}$  et  $\epsilon_{composite}$ , ainsi que l'énergie avec un niveau d'*AMR* dans la zone bornée  $[0, 0.5] \times [0.8, 1]$ , en partant d'un niveau  $G_0$  de finesse différente. On note  $N$  le nombre de nœuds utilisés pour la résolution.

Le tableau 3.19 montre que l'utilisation de l'opérateur de prolongement *FIC* avec celui de restriction *FWCV* est très performante en terme de conservation d'énergie et d'erreur (à condition d'avoir pris un maillage de départ assez fin). On note de plus que pour un maillage  $G_0 = 32 * 32$ , par exemple, l'erreur  $\epsilon_{G_0}$  est la même avec l'interpolation  $Q_2$  qu'avec la méthode *FIC* (tableau 3.19). Par contre, la répartition de cette erreur diffère suivant l'opérateur de prolongement choisi (voir figure 3.45 sur un maillage  $G_0 = 32 * 32$ ). Sachant que l'énergie est conservée à 0.001% près pour la méthode *FIC* contre 0.086% près pour l'interpolation  $Q_2$ , on en déduit qu'on peut obtenir deux solutions identiques à  $10^{-2}$  près, qui ont des propriétés de conservation différentes. On note de plus, que les erreurs  $\epsilon_{G_1}$  et  $\epsilon_{composite}$  sont meilleures avec une méthode *FIC* qu'avec une interpolation  $Q_2$ , excepté pour le maillage  $G_0 = 16 * 16$ .

On décide de calculer l'ordre de convergence des deux méthodes : *FIC* – *FWCV* et

$G_0$	Interpolation	$N$	$\epsilon_{G_1}$	$\epsilon_{G_0}$	$\epsilon_{composite}$	$E_{num}$ sur $G_0$
16 * 16	$Q_2$	907	$9.61 \cdot 10^{-4}$	$3.8 \cdot 10^{-2}$	$1.29 \cdot 10^{-3}$	8.9937
	<i>FIC</i>	828	$2.148 \cdot 10^{-3}$	$3.8 \cdot 10^{-2}$	$2.133 \cdot 10^{-3}$	8.9738
32 * 32	$Q_2$	3188	$5.762 \cdot 10^{-4}$	$1.48 \cdot 10^{-2}$	$6.561 \cdot 10^{-4}$	8.9811
	<i>FIC</i>	3029	$5 \cdot 10^{-4}$	$1.48 \cdot 10^{-2}$	$5 \cdot 10^{-4}$	8.9735
48 * 48	$Q_2$	6845	$3.3 \cdot 10^{-4}$	$8.33 \cdot 10^{-3}$	$3.67 \cdot 10^{-4}$	8.977
	<i>FIC</i>	6606	$2.174 \cdot 10^{-4}$	$8.31 \cdot 10^{-3}$	$2.168 \cdot 10^{-4}$	8.97346
64 * 64	$Q_2$	11878	$2.09 \cdot 10^{-4}$	$5.498 \cdot 10^{-3}$	$2.306 \cdot 10^{-4}$	8.9758
	<i>FIC</i>	11559	$1.211 \cdot 10^{-4}$	$5.48 \cdot 10^{-3}$	$1.208 \cdot 10^{-4}$	8.9734
96 * 96	$Q_2$	26939	$1 \cdot 10^{-4}$	$3.042 \cdot 10^{-3}$	$1.092 \cdot 10^{-4}$	8.9745
	<i>FIC</i>	26460	$5.05 \cdot 10^{-5}$	$3.037 \cdot 10^{-3}$	$5.049 \cdot 10^{-5}$	8.9734

Tableau 3.19 : Comparaison des erreurs et de l'énergie pour deux opérateurs de prolongement et une injection FWCV. Maillage  $G_0$  variable, un niveau de raffinement sur la zone bornée  $[0, 0.5] \times [0.8, 1]$ .

$Q_2 - FWCV$ . Pour cela, on représente sur la figure (3.46) le logarithme népérien de l'erreur  $\epsilon_{composite}$  en fonction du logarithme népérien de la quantité  $\sqrt{(\frac{N}{L_x L_z})}$ , où  $N$  représente le nombre de nœuds nécessaires à la résolution. On note que les deux méthodes permettent d'atteindre l'ordre limite  $q = 2$ . Cependant, l'utilisation d'une interpolation  $Q_2$  freine l'obtention de cet ordre limite car l'ordre local n'est que de 1.8 contre 2.1 avec un prolongement *FIC*. De plus, si on se réfère à l'ordre local équivalent obtenu sans *AMR* (voir figure 3.36), on note qu'il est égal à 1.99.

L'ensemble de ces résultats nous amène à conclure qu'il est impératif d'utiliser un opérateur de restriction et de prolongement conservatifs pour des phénomènes globaux, afin d'obtenir des résultats satisfaisants en terme d'erreur et de convergence tout en conservant l'énergie. Néanmoins, on peut envisager d'utiliser un opérateur de prolongement par interpolation à condition d'utiliser un maillage grossier de départ très fin.

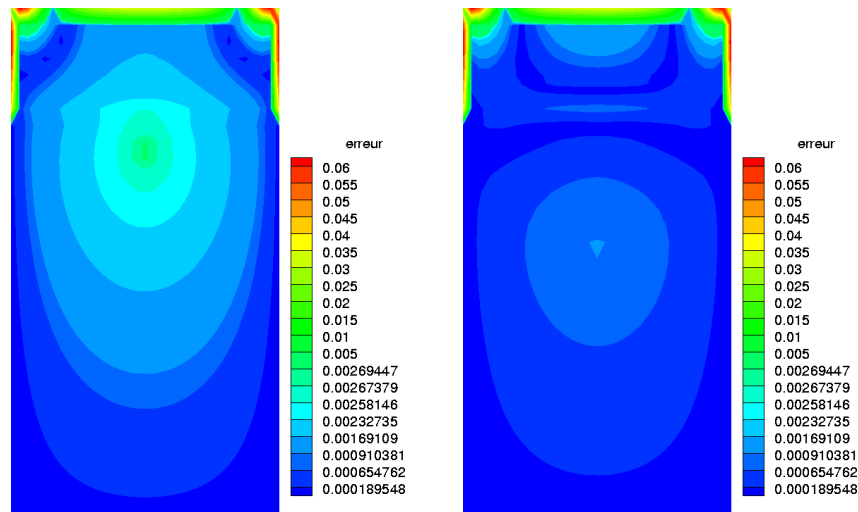


Figure 3.45 : Répartition de l'erreur sur le niveau  $G_0 = 32 * 32$  avec interpolation  $Q_2$  sur la gauche puis une interpolation conservatrice (FIC) sur la droite. Injection FWCV.

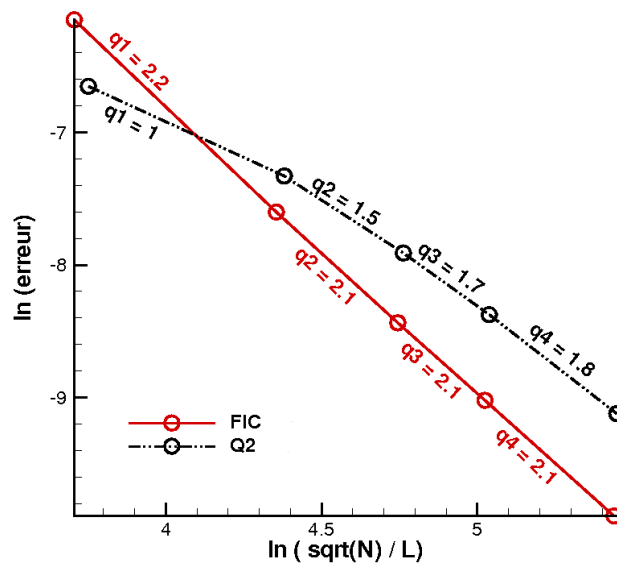


Figure 3.46 : Comparaison des ordres de convergence obtenus avec une interpolation  $Q_2$  à ceux obtenus avec une interpolation conservatrice (FIC). Injection FWCV.

### 3.4 Tache cisillée dans un écoulement monophasique par un tourbillon dans une cavité 3D

On considère un domaine cubique d'arête  $L = 0.1m$  centré en  $(x = 0.05m, z = 0.05m, y = 0.05m)$ . On y a disposé une sphère centrée en  $(x_0, z_0, y_0)$  et de rayon  $R =$



0.015m. Celle-ci est submergée dans un fluide de concentration massique nulle. Dans un premier temps, le fluide immergé a un coefficient de diffusion faible :  $D = 10^{-8}m.s^{-2}$ . On rappelle que l'équation résolue est celle de transport d'espèces.

On impose un champ de vitesses tournant d'intensité moyenne  $V_{moyen} = 0.1m^2.s^{-1}$  autour de l'axe  $y$ , de forme analytique :

$$\begin{aligned} u(x, z) &= -V_{moyen} \cos\left(\frac{\pi(x - x_0)}{L}\right) \sin\left(\frac{\pi(z - z_0)}{L}\right) \\ w(x, z) &= V_{moyen} \sin\left(\frac{\pi(x - x_0)}{L}\right) \cos\left(\frac{\pi(z - z_0)}{L}\right) \\ v(x, z) &= 0 \end{aligned}$$

Avec  $x_0 = 0.075m$ ,  $z_0 = 0.05m$  et  $y_0 = 0.05m$ .

Pour valider la méthode *AMR*, il convient de réaliser deux autres cas où l'axe de rotation du champ est  $z$  puis  $x$ . On aura ainsi testé toutes les directions d'espace. Le champ tournant d'axe  $z$  s'écrit :

$$\begin{aligned} u(x, y) &= -V_{moyen} \cos\left(\frac{\pi(x - x_0)}{L}\right) \sin\left(\frac{\pi(y - y_0)}{L}\right) \\ w(x, y) &= 0 \\ v(x, y) &= V_{moyen} \sin\left(\frac{\pi(x - x_0)}{L}\right) \cos\left(\frac{\pi(y - y_0)}{L}\right) \end{aligned}$$

Avec  $x_0 = 0.075m$ ,  $z_0 = 0.05m$  et  $y_0 = 0.05m$ .

Le champ tournant d'axe  $x$  s'écrit :

$$\begin{aligned} u(z, y) &= 0 \\ w(z, y) &= V_{moyen} \sin\left(\frac{\pi(y - y_0)}{L}\right) \cos\left(\frac{\pi(z - z_0)}{L}\right) \\ v(z, y) &= -V_{moyen} \cos\left(\frac{\pi(y - y_0)}{L}\right) \sin\left(\frac{\pi(z - z_0)}{L}\right) \end{aligned}$$

Avec  $x_0 = 0.05m$ ,  $z_0 = 0.05m$  et  $y_0 = 0.075m$ .

On teste donc trois cas : rotation autour de l'axe  $y$ ,  $z$  puis  $x$ . Pour chacun d'eux, on compare la solution obtenue au bout de  $t = 4s$  avec et sans *AMR*. On utilise un schéma *TVD* pour résoudre la partie advection de l'équation de transport. La figure 3.47 montre le cisaillement de la sphère autour de l'axe  $y$  pendant une durée de  $4s$ , sur le niveau de raffinement  $G_1 = 48 * 48 * 48$ .

On compare les masses  $M_0$ ,  $M$ ,  $M_1$ ,  $M_2$  et  $M_3$  introduites pour la tache cisailée en 2D. On adapte les définitions des masses au 3D.  $M_0$  est la masse théorique de la sphère calculée

à partir de son volume initial  $V = \frac{4}{3}\pi R^3$ .

$M$  est l'équivalent de la masse  $M_0$  calculée à partir de la solution numérique.  $M_1$  (resp  $M_2, M_3$ ) représente la masse pour laquelle la concentration massique est supérieure à  $0.9g.m^{-3}$  (resp  $0.95g.m^{-3}, 0.99g.m^{-3}$ ).

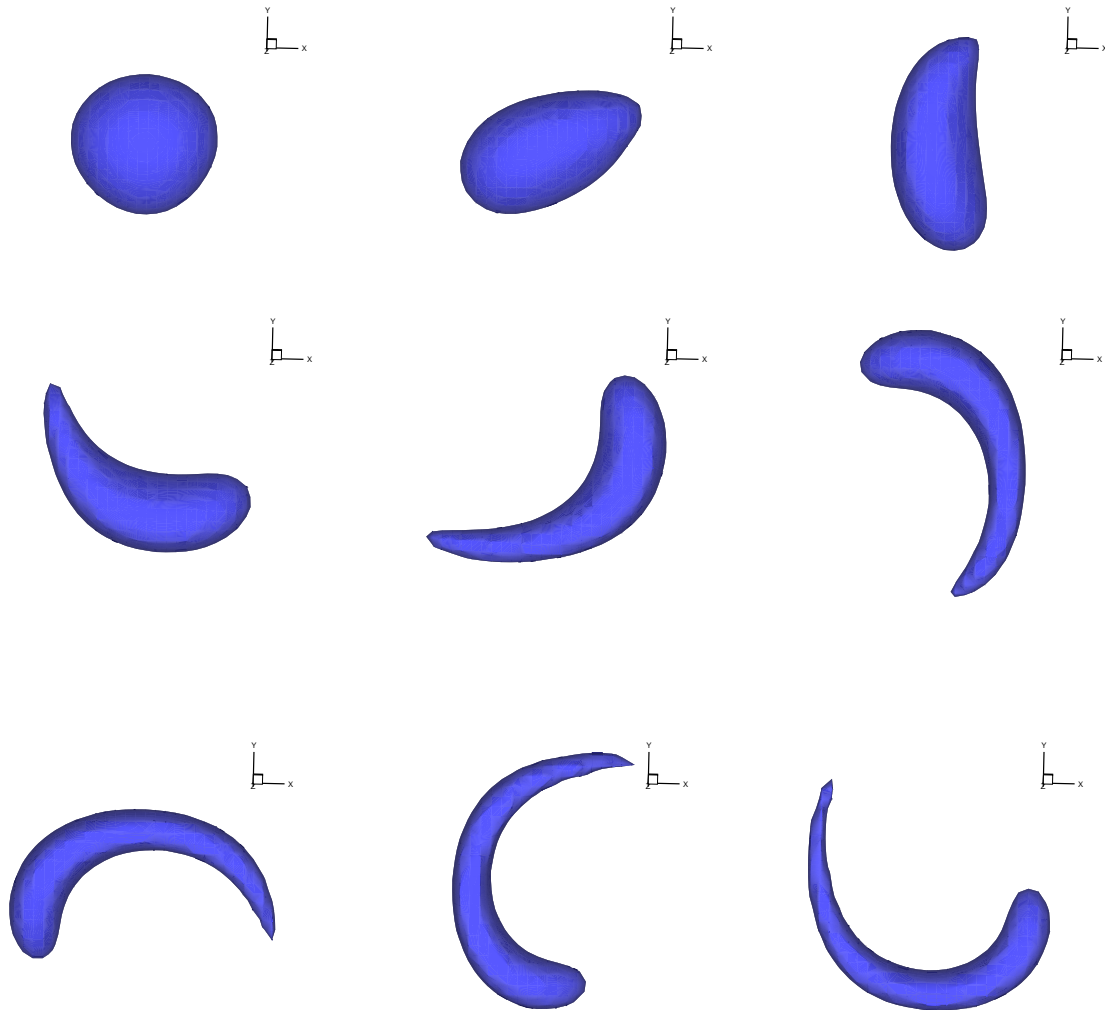


Figure 3.47 : Sketch de l'iso-concentration  $S_p = 0.5g.m^{-3}$  sur le maillage fin  $G_1 = 48^3$  jusqu'à  $t = 4s$ .  $D = 10^{-8}m^2.s^{-1}$ .

### 3.4.1 Tache cisailée autour de l'axe $y$

#### Précision des solutions

Les figures 3.48 gauche et droite représentent l'évolution des masses  $M$ ,  $M_1$ ,  $M_2$  et  $M_3$  de la sphère cisailée jusqu'à  $t = 4s$ , sur un maillage  $G_0 = 16 * 16 * 16$  avec un niveau d'AMR (gauche) et sur un maillage fin équivalent  $G = 48 * 48 * 48$  sans AMR (droite). Dans les deux cas, on note la superposition des courbes de masse  $M$ ,  $M_1$ ,  $M_2$  et  $M_3$  avec et sans AMR. En considérant que la solution obtenue sans AMR est une solution de référence, on calcule l'erreur  $e_r$  entre chaque masse  $M_{amr}$  (représentant  $M$ ,  $M_1$ ,  $M_2$  ou  $M_3$ ) sur un niveau de raffinement et  $M_{ref}$  (représentant  $M$ ,  $M_1$ ,  $M_2$  ou  $M_3$ ) sur une monogrille.  $e_r$  est calculée comme suit :  $e_r = \frac{|M_{amr} - M_{ref}|}{|M_{ref}|}$ . On place dans le tableau 3.20 une valeur moyenne de  $e_r$ . Celui-ci vient confirmer cette superposition puisque les erreurs ne dépassent pas 2%. Cette différence est due au fait qu'on considère qu'en dehors de la zone de raffinement,  $S_p$  est soit nul soit égal à  $1g.m^{-3}$ . Ceci montre que cette méthode de raffinement est efficace pour réduire la diffusion numérique en 3D. Pour vérifier que la concentration massique est répartie de la même manière avec et sans AMR, on compare les iso-concentrations  $S_p = 0.2g.m^{-3}$ ,  $0.4g.m^{-3}$ ,  $0.6g.m^{-3}$  et  $0.8g.m^{-3}$  obtenues avec et sans AMR à  $t = 4s$ , avec  $D = 10^{-8}m^2.s^{-1}$  (cf figure 3.49). Comme on peut le remarquer, elles sont confondues, ce qui montre une bonne répartition de la concentration massique. On calcule aussi l'erreur relative définie précédemment  $\epsilon$  entre la solution sans et avec AMR au même instant : on obtient  $\epsilon = 1,637.10^{-2}$ , ce qui est très satisfaisant pour un maillage fin  $G_1$  équivalent à un maillage  $G = 48 * 48 * 48$  sans AMR. Ce résultat vient confirmer l'efficacité de l'approche AMR à donner une bonne précision des solutions. La figure 3.47 permet de visualiser le cisaillement de la tache sur un niveau d'AMR ( $G_1 = 48 * 48 * 48$ ) au cours du temps jusqu'à  $t = 4s$ . On y a représenté l'iso-concentration  $S_p = 0.5g.m^{-3}$  à intervalle de temps régulier.

	$M$	$M_1$	$M_2$	$M_3$
$e_r$	1.28%	1%	1.24%	1.78%

Tableau 3.20 : Erreur moyenne entre les masses  $M_{amr}$  et  $M_{ref}$ .

On s'intéresse maintenant aux performances en taille mémoire et en temps CPU de cette méthode. Les calculs nécessitant des ressources informatiques importantes en 3D, on

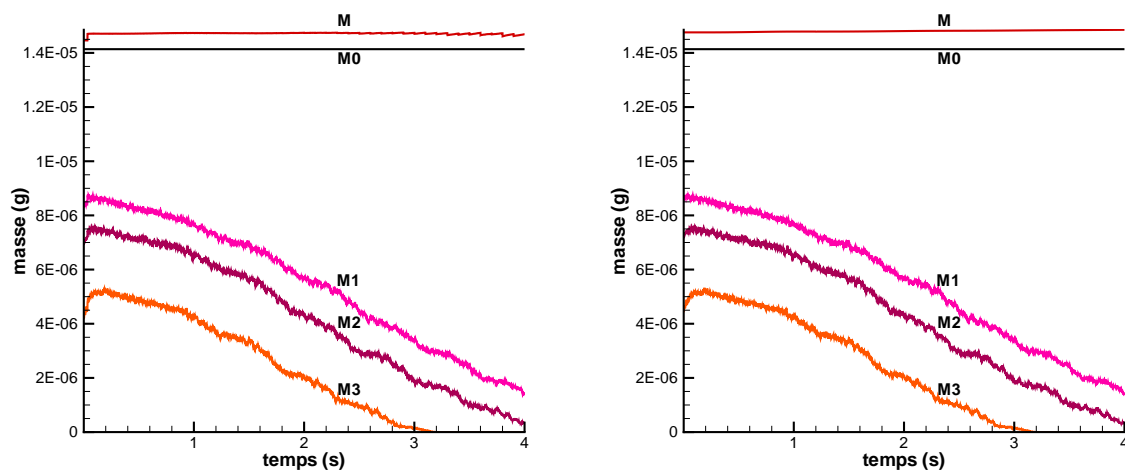


Figure 3.48 : Evolution des masses pour la tache cisailée autour de l'axe  $y$  avec un niveau de raffinement (gauche) et sans AMR (droite) pendant 4s.  $D = 10^{-8}m^2.s^{-1}$ .

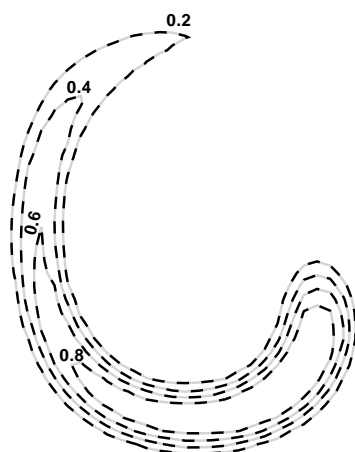


Figure 3.49 : Iso-concentrations avec (gris) et sans AMR (noir) à  $t = 4s$  et  $D = 10^{-8}m^2.s^{-1}$ . Coupe dans le plan  $xz$ , pour  $y = 0.05m$ .

se contente de comparer deux cas *AMR* à leurs homologues en maillage, sans *AMR*. Dans le premier, on part d'un maillage  $G_0 = 16 * 16 * 16$  avec un niveau de raffinement pendant un temps  $t = 4s$ . Dans le deuxième cas, on choisit un maillage grossier  $G_0 = 12 * 12 * 12$

avec deux niveaux de raffinement pendant une durée de 4s. On utilise les mêmes notations que dans le cas de la tache cisailée en 2D (cf sous-section 3.1.6).

**Analyse des performances en mémoire**

En adoptant la même démarche que dans la sous-section 3.1.6 et en utilisant les tableaux 3.21 et 3.22, on considère que l’approche *AMR* est intéressante en termes de coût en mémoire tant que  $rap < 0.75$ , c’est-à-dire  $rgl < 57\%$ . On rappelle que pour le même cas en dimension d’espace deux, on a évalué le pourcentage de points en deçà duquel l’approche *AMR* était intéressante à  $rgl < 32\%$ . Ceci montre que la méthode *AMR* est bien plus efficace, en termes de gain en taille mémoire, en 3D qu’en 2D.

maillage fin	Mémoire avec <i>AMR</i> (Mo)	Mémoire sans <i>AMR</i> (Mo)	rap
48 * 48 * 48	185	200	0.93
108 * 108 * 108	1400	2100	0.67

Tableau 3.21 : *Mémoire utilisée (en Mo) pour simuler la tache cisailée avec et sans AMR, sur différents maillages.*

maillage fin	$NBG_L$	$KKT_L$	$rgl$
48 * 48 * 48	85413	117649	0.7
108 * 108 * 108	616072	1259712	0.5

Tableau 3.22 : *Rapport rgl pour différents nombres de niveaux fins au temps  $t = 4s$ .*

**Analyse des performances en temps CPU**

On souhaite montrer les performances de la méthode *AMR* en termes de gain en temps *CPU*, sachant que son optimisation en 3D est incomplète (résolution de tous les nœuds *AMR* à l’intérieur du domaine  $\Omega$ ).

Comme dans le cas de la tache cisailée en 2D, on estime dans un premier temps la répartition des différents temps moyens définis dans la sous-section 3.1.6. Ceux-ci sont représentés sur la figure 3.50.

On observe sur cette figure que la gestion dynamique du raffinement reste toujours aussi

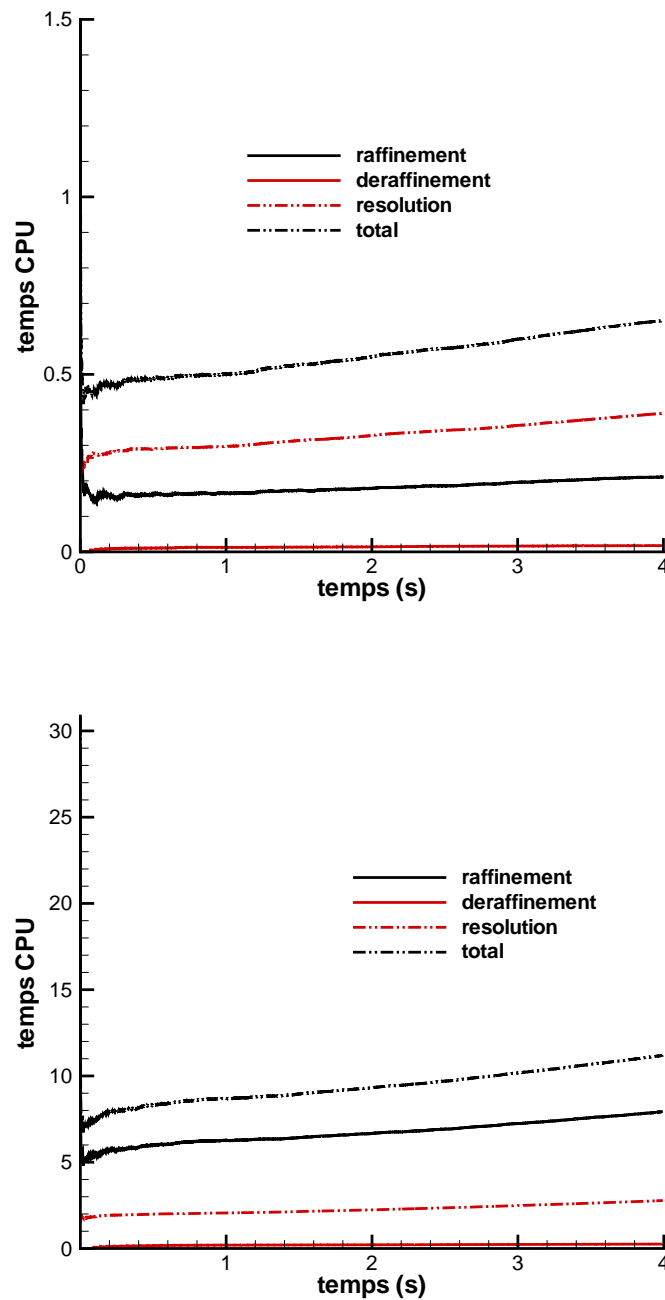


Figure 3.50 : Comparaison des différents temps moyens avec AMR pendant  $t = 4s$ .  $G_0 = 16 * 16 * 16$  avec un niveau à gauche.  $G_0 = 12 * 12 * 12$  avec deux niveaux à droite.

coûteuse en  $2D$  qu'en  $3D$ , notamment à cause des nombreuses allocations/désallocations de tableaux.

Étant donné que les cas sans *AMR* sont coûteux en temps *CPU* et que le temps moyen  $t_{tot}$  est pratiquement constant au cours du temps, on a calculé le temps *CPU* total pour faire quelques itérations, puis on a extrapolé le résultats sur une durée de 4s.

Les résultats du tableau 3.23 montrent qu'il faut continuer à optimiser la méthode.

maillage fin	temps <i>CPU</i> total sans <i>AMR</i>	temps <i>CPU</i> total avec <i>AMR</i>
108 * 108 * 108	71601	97490

Tableau 3.23 : Comparaison des temps *CPU* avec et sans *AMR*.

### 3.4.2 Tache cisailée autour de l'axe $z$

De même que pour la tache cisailée autour de l'axe  $y$ , les figures 3.52 gauche et droite représentent l'évolution des masses  $M$ ,  $M_1$ ,  $M_2$  et  $M_3$  de la sphère cisailée jusqu'à  $t = 4s$  autour de l'axe  $z$ . Dans les deux cas, on note la superposition des courbes de masse  $M$ ,  $M_1$ ,  $M_2$  et  $M_3$  avec et sans *AMR*. De même que pour le cas de la sphère cisailée autour de l'axe  $y$ , le tableau 3.24 confirme une évolution similaire dans les deux cas à quelques % près. Donc la méthode *AMR* permet d'obtenir une solution fine (sur  $G_1 = 48^3$ ) aussi précise que celle qu'on aurait obtenu sur un maillage équivalent sans *AMR* ( $G = 48^3$ ) à  $10^{-2}$  près (valeur d' $\epsilon$  calculé de la même façon que dans le cas de la tache cisailée autour de l'axe  $y$ ). On compare aussi les iso-concentrations  $S_p = 0.2g.m^{-3}$ ,  $0.4g.m^{-3}$ ,  $0.6g.m^{-3}$  et  $0.8g.m^{-3}$  obtenues avec et sans *AMR* à  $t = 4s$ , avec  $D = 10^{-8}m^2.s^{-1}$  (cf figure 3.51). Celles-ci sont confondues, ce qui vient conforter le résultat précédent.

	$M$	$M_1$	$M_2$	$M_3$
$e_r$	1.28%	1%	1.24%	1.78%

Tableau 3.24 : Erreur moyenne entre les masses  $M_{amr}$  et  $M_{ref}$ .

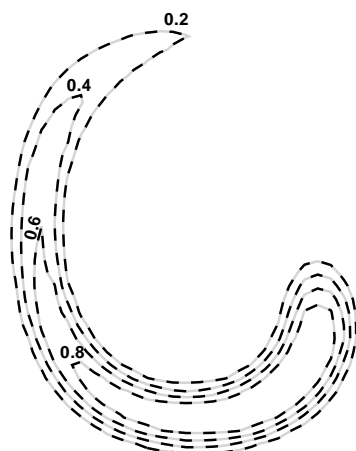


Figure 3.51 : Iso-concentrations avec (gris) et sans AMR (noir) autour de l'axe  $z$  à  $t = 4s$  et  $D = 10^{-8}m^2.s^{-1}$ . Coupe dans le plan  $xy$ , pour  $z = 0.05m$ .

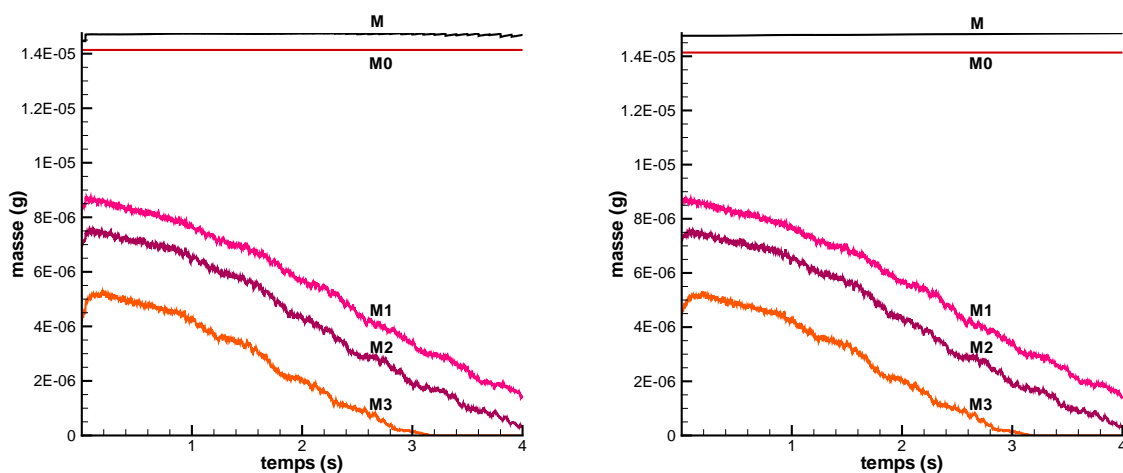


Figure 3.52 : Evolution des masses pour la tache cisailée autour de l'axe  $z$  avec un niveau de raffinement (gauche) et sans AMR (droite) pendant  $4s$ .  $D = 10^{-8}m^2.s^{-1}$ .

### 3.4.3 Tache cisailée autour de l'axe $x$

On teste enfin la troisième direction d'espace. Les figures 3.54 gauche et droite représentent l'évolution des masses  $M$ ,  $M_1$ ,  $M_2$  et  $M_3$  de la sphère cisailée jusqu'à  $t = 4s$  autour



### 3.4 Tache cisailée dans un écoulement monophasique par un tourbillon dans une cavité 3D 139

de l'axe  $x$ . Dans les deux cas, on note la superposition des courbes de masse  $M$ ,  $M_1$ ,  $M_2$  et  $M_3$  avec et sans  $AMR$ . De même que pour le cas de la sphère cisailée autour de l'axe  $y$ , le tableau 3.25 confirme une évolution similaire dans les deux cas à quelques % près. On en conclut que cette méthode permet d'obtenir une solution fine (sur  $G_1 = 48^3$ ) aussi précise que celle qu'on aurait obtenu sur un maillage équivalent sans  $AMR$  ( $G = 48^3$ ) à  $10^{-2}$  près.

Les iso-concentrations  $S_p = 0.2g.m^{-3}$ ,  $0.4g.m^{-3}$ ,  $0.6g.m^{-3}$  et  $0.8g.m^{-3}$  obtenues avec et sans  $AMR$  à  $t = 4s$ , avec  $D = 10^{-8}m^2.s^{-1}$  sur la figure 3.53 sont confondues, ce qui vient conforter le résultat précédent.

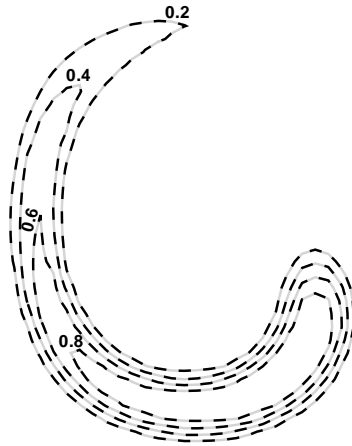


Figure 3.53 : Iso-concentrations de la tache cisailée avec (gris) et sans  $AMR$  (noir) autour de l'axe  $z$  à  $t = 4s$  et un coefficient de diffusion  $D = 10^{-8}m^2.s^{-1}$ . Coupe dans le plan  $zy$ , pour  $x = 0.05m$ .

	$M$	$M_1$	$M_2$	$M_3$
$e_r$	1.28%	1%	1.24%	1.78%

Tableau 3.25 : Erreur moyenne entre les masses  $M_{amr}$  et  $M_{ref}$ .

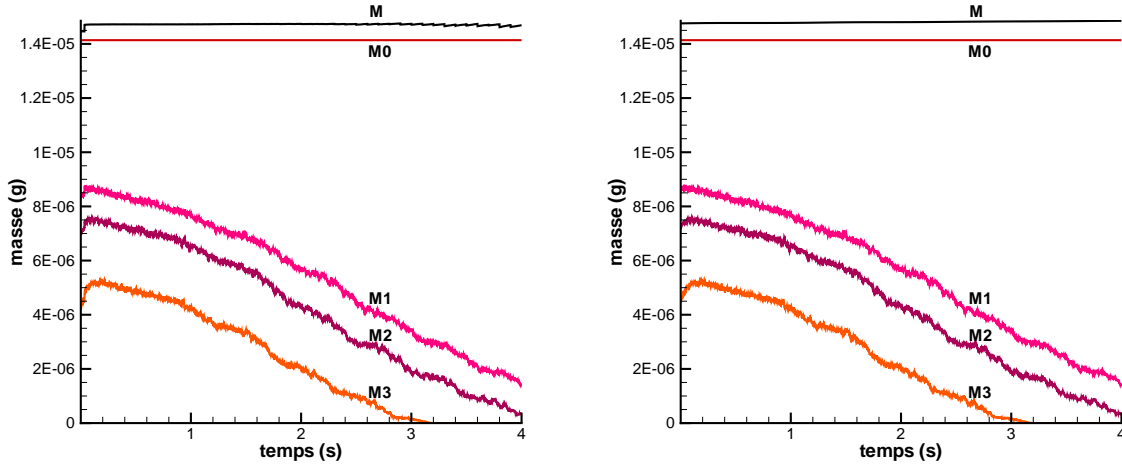


Figure 3.54 : Evolution des masses pour la tache cisailée autour de l'axe  $x$  avec un niveau de raffinement (gauche) et sans AMR (droite) pendant 4s.  $D = 10^{-8}m^2.s^{-1}$ .

### 3.5 Tache cisailée par un champ tournant périodique dans un écoulement diphasique $3D$

On a repris ici le cas proposé par [LEVEQUE 96] et [LIOVIC 06] dans le but de comparer les résultats obtenus par [LIOVIC 06] et par le code *Aquilon*. Dans ce cas, on considère un domaine cubique d'arête  $L = 1m$  centré en  $(x = 0.5m, z = 0.5m, y = 0.5m)$ . Un premier fluide de forme sphérique est centré en  $(x_0, z_0, y_0)$  et a pour rayon  $R = 0.15m$ . Le reste du domaine est occupé par un deuxième fluide. L'équation de transport d'interface est résolue. On impose un champ de vitesses tournant  $2T$ -périodique de forme analytique :

$$\begin{aligned} u(x, z, y) &= 2 \sin^2(\pi x) \sin(2\pi z) \sin(2\pi y) \cos\left(\frac{\pi t}{T}\right) \\ w(x, z, y) &= -\sin(2\pi x) \sin^2(\pi z) \sin(2\pi y) \cos\left(\frac{\pi t}{T}\right) \\ v(x, z, y) &= -\sin(2\pi x) \sin(2\pi z) \sin^2(\pi y) \cos\left(\frac{\pi t}{T}\right) \end{aligned}$$

Ce cas est particulièrement intéressant car la sphère est cisailée à tel point que son épaisseur devient très fine lorsque le cisaillement devient maximum (au bout du temps  $t = \frac{(2k-1)T}{2}$ ,  $k$  entier strictement positif). Si l'épaisseur de la tache est plus fine que la taille d'une maille à ce moment là, l'interface est dégradée (cf figure 3.55). Il en résulte une perte de matière qu'on peut estimer à l'instant suivant  $t = kT$ . On impose  $T = 3s$ . On

a représenté sur la figure 3.55 les résultats obtenus par [LIOVIC 06] sur deux maillages  $G = 128 * 128 * 128$  et  $G = 256 * 256 * 256$  au bout de  $t = 1.5s$  et  $t = 3s$ , en séquentiel. Ces cas étant très coûteux en mémoire, on se limite à une étude purement qualitative. On

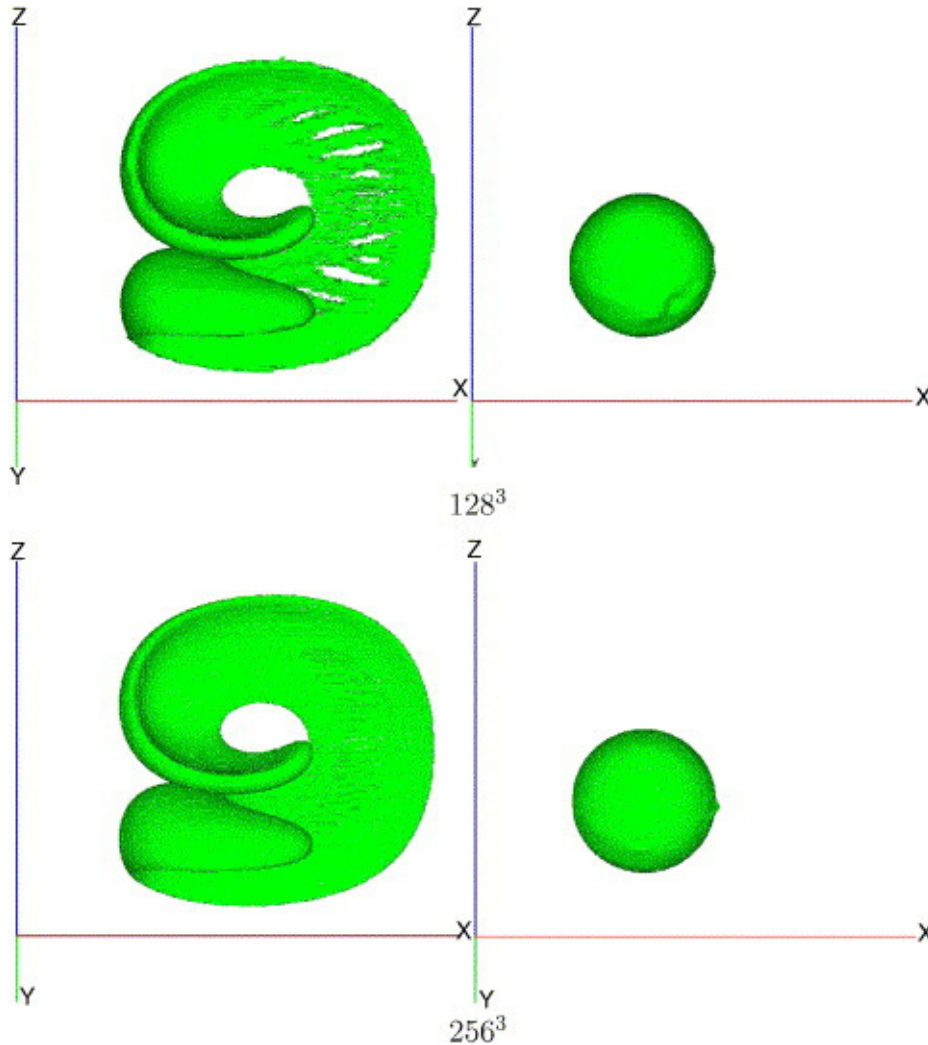


Figure 3.55 : Représentation de l'interface à son maximum de déformation ( $t = 1.5s$ ) et à son minimum ( $t = 3s$ ) pour un maillage  $G = 128 * 128 * 128$  (haut) et  $G = 256 * 256 * 256$  (bas) ([LIOVIC 06]).

donne aussi des ordres de grandeurs concernant la taille mémoire nécessaire pour simuler ce cas avec *AMR*.

On compare sur les figures 3.56 et 3.57 deux cas *AMR* avec un schéma *TVD*, au bout de  $0.75s$  et  $1.5s$ . L'interface sur les figures 3.56 et 3.57 de gauche (respectivement droite) est obtenue avec un maillage grossier  $G_0 = 12 * 12 * 12$  (respectivement  $G_0 = 22 * 22 * 22$ ) et

deux niveaux *AMR*, soit un maillage fin équivalent  $G = 108 * 108 * 108$  (respectivement  $G = 198 * 198 * 198$ ).

On observe que la partie où l'épaisseur de la tache est la plus fine a été diffusée. Comme on pouvait s'y attendre, on capte une partie supplémentaire de la sphère cisailée lorsque le pas d'espace du maillage le plus fin diminue. Cependant, il aurait fallu atteindre un maillage fin équivalent minimum de  $G = 256 * 256 * 256$  pour observer une solution proche de celle présentée par [LIOVIC 06]. On précise que le premier cas a nécessité une taille mémoire d'environ  $1.6Go$ , tandis que le second en a requis  $10Go$ .

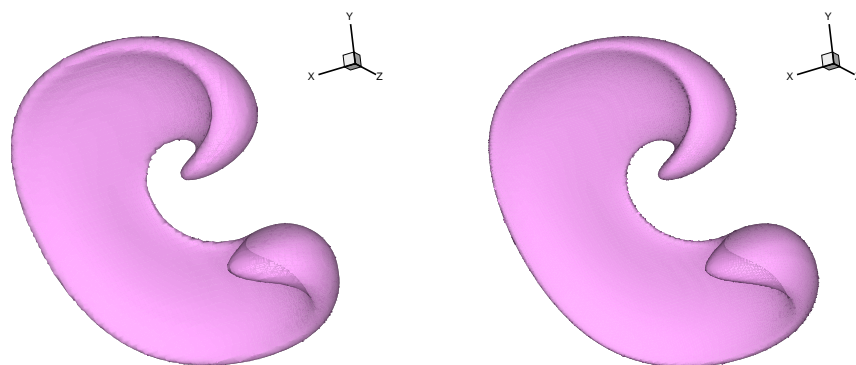


Figure 3.56 : Représentation de l'interface à  $t = 0.75s$ , pour un maillage  $G_0 = 12 * 12 * 12$  (gauche) et  $G = 22 * 22 * 22$  (droite), avec deux niveaux fins et un schéma TVD.

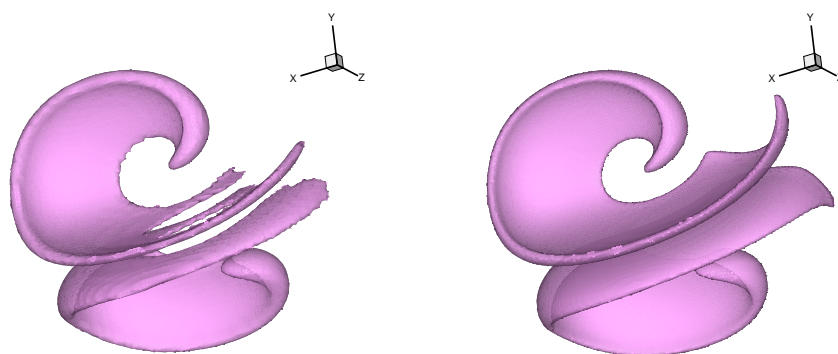


Figure 3.57 : Représentation de l'interface à son maximum de déformation ( $t = 1.5s$ ) pour un maillage  $G_0 = 12 * 12 * 12$  (gauche) et  $G = 22 * 22 * 22$  (droite), avec deux niveaux fins et un schéma TVD.

## Chapitre 4

# Résultats sur les équations de Navier-Stokes

Dans cette partie, on cherche à valider la méthode *AMR* pour l'équation de conservation de quantité de mouvement sous la contrainte d'incompressibilité. Le second objectif est d'évaluer ses performances en terme de précision des solutions et en terme de gain en taille mémoire et temps *CPU*.

### 4.1 Résultats préliminaires sur un écoulement de Poiseuille 2D

L'objectif de cette section est de tester la méthode *AMR* sur les équations de Navier-Stokes. On va donc étudier son influence sur le comportement des solutions, via le changement de plusieurs paramètres.

On considère un écoulement de Poiseuille 2D, sur une longueur  $L = 1m$ , entre deux plans parallèles séparés d'une distance  $h = 0.02m$ . Celui-ci est représenté sur la figure 4.1 (pour plus de clarté, cette figure n'est pas à l'échelle). La vitesse moyenne est  $U_{moyen} = 0.1m.s^{-1}$  et le nombre de Reynolds est  $Re = 100$ . La solution analytique est indiquée dans l'annexe A.

On choisit de simuler cet écoulement pendant une durée de 60s (temps au bout duquel le régime stationnaire a été atteint), avec un pas de temps  $\Delta t = 1s$ . On étudie l'évolution de l'erreur entre la solution numérique et analytique au cours du temps.

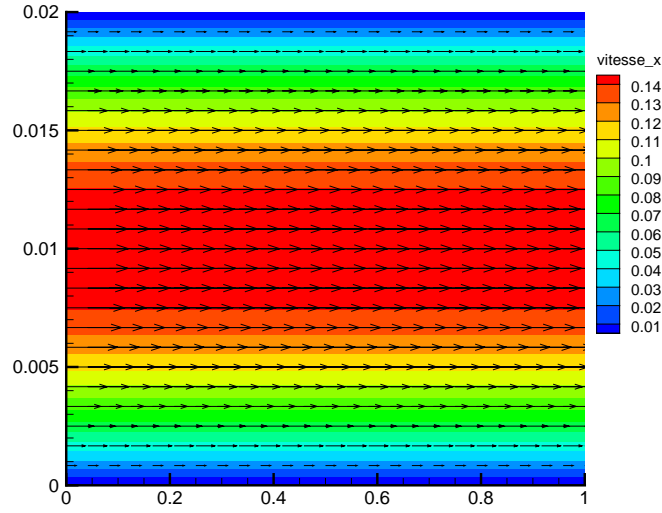


Figure 4.1 : Écoulement de Poiseuille sur un maillage  $G = 24 * 24$  : représentation du champ de vitesse et de son amplitude.

La solution numérique sur les niveaux de raffinement est calculée comme cela a été indiqué dans la sous-section 2.1.1, à savoir que les nœuds de limite de niveau sont interpolés tandis que ceux à l'intérieur sont discrétisés. On utilise la méthode du Lagrangien augmenté pour résoudre les équations de Navier-Stokes (cf sous-section 1.2.3). On note  $(u_{ex}, w_{ex})$  le vecteur vitesse analytique ou exact, calculé comme dans l'annexe A, et  $(u_{res}, w_{res})$  le vecteur vitesse obtenu numériquement. On définit l'erreur en norme  $L^2$ , de la manière suivante :

$$erreur^2 = \frac{\int_{\Omega} |u_{ex}(x, z) - u_{res}(x, z)|^2 + |w_{ex}(x, z) - w_{res}(x, z)|^2}{\int_{i, j \in \Omega} |u_{ex}(x, z)|^2 + |w_{ex}(x, z)|^2}$$

soit sous forme discrétisée :

$$erreur^2 = \frac{\sum_{i, j \in \Omega} |u_{ex}(x_i, z_j) - u_{res}(x_i, z_j)|^2 + |w_{ex}(x_i, z_j) - w_{res}(x_i, z_j)|^2}{\sum_{i, j \in \Omega} |u_{ex}(x_i, z_j)|^2 + |w_{ex}(x_i, z_j)|^2}$$

### 4.1.1 Influence du degré du polynôme d'interpolation

On cherche à déterminer l'évolution de l'erreur en fonction de l'ordre d'interpolation ( $Q_1$ ,  $Q_2$  et  $Q_3$ ) des nœuds de limite de niveau fin. Pour cela, un niveau d'AMR est construit dans la zone bornée  $[0.3, 0.7] \times [0.007, 0.011]$  avec un coefficient de raffinement  $RAF = 3$  (voir figure 4.2). On montre sur la figure 4.3 l'évolution temporelle de l'erreur sur le niveau  $G_1$  (figure 4.3) jusqu'au temps  $t = 60s$ . Les erreurs au temps final sont indiquées dans le tableau 4.1.

On note une évolution identique pour l'interpolation  $Q_2$  et  $Q_3$  jusqu'au temps  $t = 45s$ , où l'erreur est de l'ordre de  $10^{-13}$ . Ceci s'explique par le fait que la solution analytique est un polynôme de degré 2 (cf annexe A.2) et donc on attribue aux nœuds interpolés la solution analytique via l'interpolation  $Q_2$  ou  $Q_3$ . Ensuite, on observe un phénomène de saturation de l'erreur à  $10^{-13}$  pour l'interpolation  $Q_3$ , qui est du aux termes de degré 3 du polynôme d'interpolation  $Q_3$  qui sont théoriquement nuls, mais qui sont égaux à  $10^{-16}$  numériquement. Quant à l'interpolation  $Q_1$ , on a une erreur de l'ordre de  $10^{-1}$ , ce qui est normal puisque qu'on a approché la solution par un polynôme de degré un.

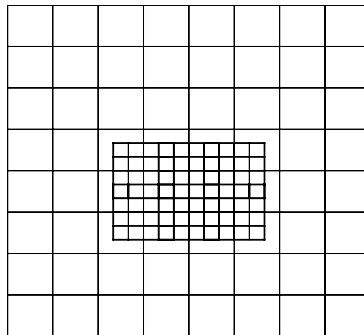


Figure 4.2 : Maillage global avec un niveau fin.

	$Q_1$	$Q_2$	$Q_3$
Erreur sur $G_1$	0.17	$8.2 \cdot 10^{-14}$	$1.6 \cdot 10^{-13}$

Tableau 4.1 : Erreur sur  $G_1$ , à  $t = 60s$  pour trois types d'interpolation

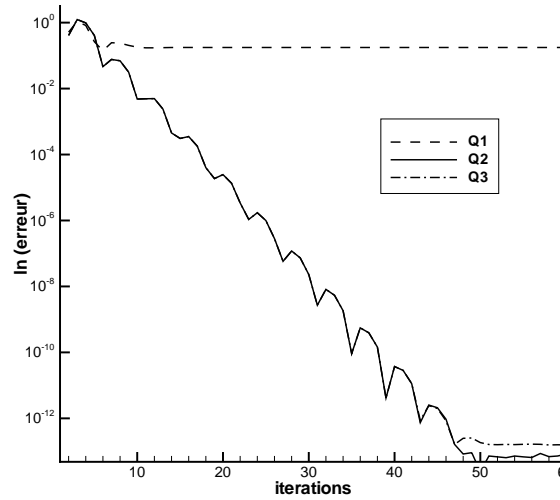


Figure 4.3 : Evolution de l'erreur sur le niveau  $G_1$  pour différents types d'interpolation

#### 4.1.2 Influence du nombre de niveaux de raffinement

On cherche à savoir comment se comporte l'évolution de l'erreur quand le nombre de niveaux fins varie. On décide donc de construire successivement un, deux puis trois niveaux fins dans la zone bornée  $[0.3, 0.7] \times [0.007, 0.011]$ , avec un coefficient de raffinement  $RAF = 3$  et une interpolation de type  $Q_2$  (voir figure 4.5 pour un maillage à deux niveaux d'*AMR*).

On observe sur la figure 4.4 une évolution temporelle de l'erreur, très similaire quelque soit le nombre de niveaux utilisé. On compare ce comportement à celui obtenu sur un maillage  $G = 8 * 8$  sans *AMR*. On note encore une évolution similaire aux autres courbes sur la figure 4.4. La méthode *AMR* donne donc des résultats satisfaisants indépendamment du nombre de niveaux fins choisi. Ceci est confirmé par les résultats du tableau 4.2, qui montre les erreurs obtenues au temps  $t = 60s$  sur les différents niveaux *AMR*.

Erreur sur $G = 8 * 8$	Erreur sur $G_1$	Erreur sur $G_2$	Erreur sur $G_3$
$5.5 \cdot 10^{-14}$	$7.3 \cdot 10^{-14}$	$2 \cdot 10^{-13}$	$6.5 \cdot 10^{-13}$

Tableau 4.2 : Erreur sur chaque niveau à  $t = 60s$  avec une interpolation  $Q_2$



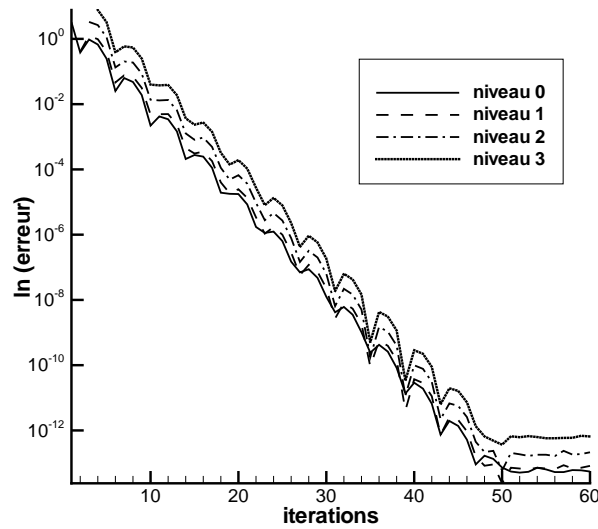


Figure 4.4 : Evolution de l'erreur sur différents niveaux pour une interpolation  $Q_2$

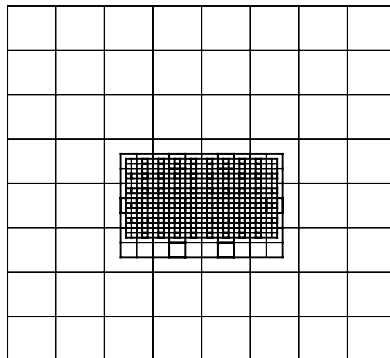


Figure 4.5 : Maillage avec deux niveaux d'AMR

### 4.1.3 Influence du coefficient de raffinement

On souhaite déterminer l'influence du coefficient de raffinement,  $RAF$ , sur la précision de la solution. Dans cette optique, on fait varier le paramètre  $RAF$ , sur un puis deux niveaux d'AMR dans la zone bornée  $[0.3, 0.7] \times [0.007, 0.011]$ . Pour plus de clarté, on a représenté sur les figures 4.2 et 4.6, les maillages utilisés avec un niveau AMR et un

coefficient de raffinement égal à 3, 5, puis 7. On montre l'évolution des erreurs au bout d'un temps  $t = 60s$  avec une interpolation de type  $Q_2$ , sur le niveau le plus fin sur la figure 4.7.

On constate que les erreurs sont faiblement dépendantes du coefficient de raffinement (malgré une légère détérioration avec  $RAF$  sur la figure 4.7 droite), et ce, aussi bien pour un que pour deux niveaux de raffinement (voir figure 4.7). Ainsi, la méthode  $AMR$  reste satisfaisante quand le coefficient de raffinement  $RAF$  varie, ce que vient confirmer quantitativement le tableau 4.3.

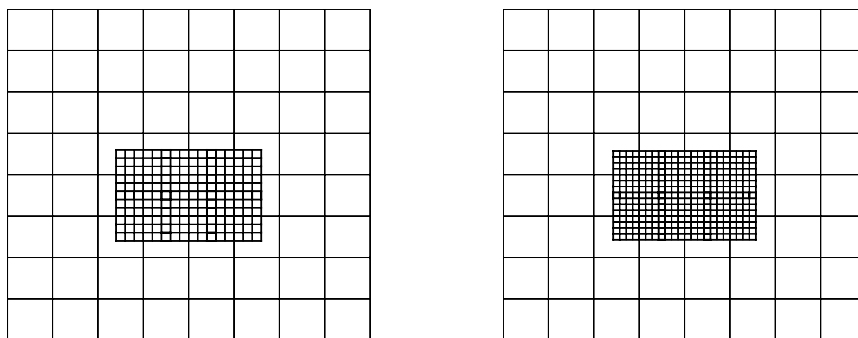


Figure 4.6 : Maillage avec un niveau fin pour  $RAF = 5$  à gauche et  $RAF = 7$  à droite.

	Erreur sur $G_1$	Erreur sur $G_2$
$raf = 3$	$7.3 \cdot 10^{-14}$	$2 \cdot 10^{-13}$
$raf = 5$	$8.9 \cdot 10^{-14}$	$4.5 \cdot 10^{-13}$
$raf = 7$	$2 \cdot 10^{-13}$	$9 \cdot 10^{-13}$

Tableau 4.3 : Erreur sur chaque niveau à  $t = 60s$  avec une interpolation  $Q_2$

Dans la suite du document, on utilisera un coefficient de raffinement  $RAF = 3$ .

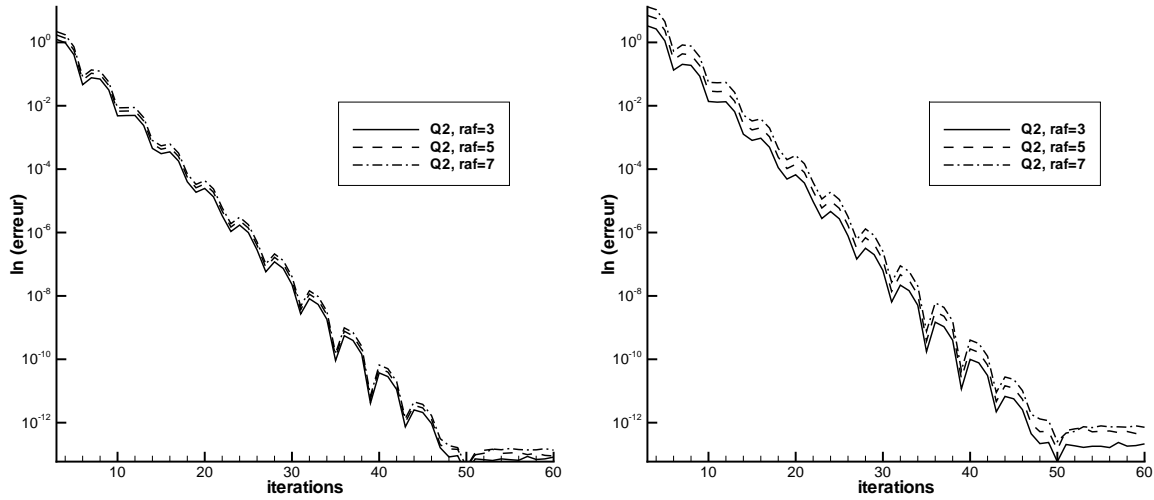


Figure 4.7 : Evolution de l'erreur sur le niveau  $G_1$  (gauche) et  $G_2$  (droite) pour une interpolation de type  $Q_2$  et un coefficient de raffinement variable.

## 4.2 Tourbillon de Green-Taylor 2D

L'objectif de cette section est de montrer le comportement de la méthode *AMR* dans la résolution des équations de Navier-Stokes, concernant la précision des solutions et les gains en terme de taille mémoire et de temps *CPU*.

On se place dans une cavité  $\Omega = [-0.1m, 0.1m] \times [-0.1m, 0.1m]$  de côté  $H = 0.2m$  dans laquelle existe un écoulement instationnaire, appelé tourbillon de Green-Taylor. Celui-ci a été modifié en intégrant un terme source  $\vec{S}_0$  dans les équations de Navier-Stokes, dans le but de faire apparaître une solution stationnaire non nulle (cf [CALTAGIRONE 99]). Le second membre  $\vec{S}_0$  a pour expression :

$$\begin{aligned} u_0(x, z) &= -\frac{\pi^2 \mu}{2H^2} \cos\left(\frac{\pi x}{2H}\right) \sin\left(\frac{\pi z}{2H}\right) \\ w_0(x, z) &= \frac{\pi^2 \mu}{2H^2} \sin\left(\frac{\pi x}{2H}\right) \cos\left(\frac{\pi z}{2H}\right) \end{aligned} \quad (4.1)$$

Ce qui donne le champ de vitesse analytique (voir figure 4.8) :

$$\begin{aligned} u(x, z, t) &= -\cos\left(\frac{\pi x}{2H}\right) \sin\left(\frac{\pi z}{2H}\right) \left(1 - \exp\left(\frac{-\pi^2 \mu t}{2\rho H^2}\right)\right) \\ w(x, z, t) &= \sin\left(\frac{\pi x}{2H}\right) \cos\left(\frac{\pi z}{2H}\right) \left(1 - \exp\left(\frac{-\pi^2 \mu t}{2\rho H^2}\right)\right) \end{aligned} \quad (4.2)$$

et la pression analytique :

$$p(x, z, t) = -\frac{\rho}{2} \left( \cos\left(\frac{\pi x}{2H}\right)^2 + \cos\left(\frac{\pi z}{2H}\right)^2 \right) \left( 1 - 2 \exp\left(\frac{-\pi^2 \mu t}{2\rho H^2}\right) + \exp\left(\frac{-\pi^2 \mu t}{\rho H^2}\right) \right) \quad (4.3)$$

La condition initiale correspond à un champ de vitesses nul. Les conditions aux limites sont de type Dirichlet. Elles sont obtenues en imposant le champ de vitesses analytique au niveau des parois du domaine  $\Omega$ , ce qui donne, au temps  $t = 0.001s$ , le champ de vitesse de la figure 4.8 gauche. On a introduit initialement un fluide coloré, dont la surface est un cercle de centre  $(0.05m, 0m)$  et de rayon  $0.02m$  (cf figure 4.8 droite). Pour générer les cellules *AMR*, on utilise un critère de raffinement basé sur le gradient de concentration. On utilise un schéma *TVD* pour résoudre l'équation de transport et la méthode du Lagrangien augmenté pour les équations de Navier-Stokes.

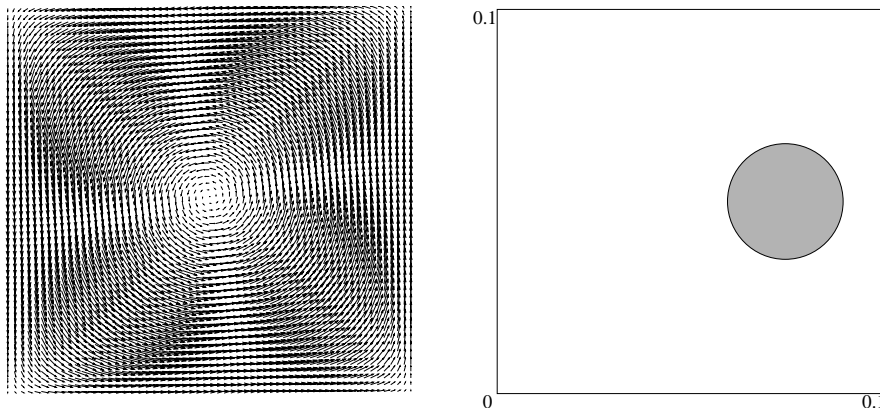


Figure 4.8 : *Champ de vitesses analytique à gauche à  $t = 0.001s$  et fluide coloré introduit initialement à droite, maillage  $G = 90 * 90$ .*

#### 4.2.1 Interpolations non conservatives et condition d'incompressibilité

On s'intéresse dans cette sous-section au comportement de la méthode *AMR* lorsqu'une méthode de lagrangien augmenté est utilisée pour les équations de Navier-Stokes. On compare donc, dans un premier temps, la valeur maximale en valeur absolue de la divergence sur les niveaux  $G_0 = 90 * 90$  et  $G_1$  lorsque le nombre d'itérations de lagrangien augmenté  $NT_{la}$  varie. Le tableau 4.4 montre que la divergence baisse sur le niveau fin quand  $NT_{la}$  augmente. Par contre, la divergence sur le niveau grossier reste de l'ordre de  $10^{-4}$  et diminue peu quand le nombre d'itérations en lagrangien augmente.

On compare les valeurs des divergences obtenues pour  $NT_{la} = 3$  sur les niveaux  $G_0$  et  $G_1$  aux valeurs obtenues sur des maillages  $G = 90 * 90$  et  $G = 270 * 270$ . On observe dans le tableau 4.5 que la méthode de raffinement de maillage ne dégrade pas la divergence du

maillage grossier et que celle obtenue sur le niveau de raffinement est équivalente à celle obtenue sur un maillage  $G = 270 * 270$ .

$NT_{la}$	résidu	divergence $G_0$	divergence $G_1$
1	$10^{-15}$	$6 \cdot 10^{-4}$	$10^{-4}$
3	$10^{-15}$	$2 \cdot 10^{-4}$	$10^{-6}$
6	$10^{-15}$	$10^{-4}$	$10^{-8}$

Tableau 4.4 : Valeur de la divergence par niveau en fonction du nombre d'itérations en lagrangien augmenté ( $NT_{la}$ ) au temps  $t = 2s$ .  $\Delta t = 0.005s$ , interpolation  $Q_2$  sur les nœuds de limites du niveau  $G_1$ .

divergence $G_0$	divergence $G_1$	divergence $G = 90 * 90$	divergence $G = 270 * 270$
$2 \cdot 10^{-4}$	$10^{-6}$	$2 \cdot 10^{-4}$	$10^{-6}$

Tableau 4.5 : Comparaison des divergences avec et sans AMR au temps  $t = 2s$ .  $\Delta t = 0.005s$ , interpolation  $Q_2$  sur les nœuds de limites du niveau  $G_1$ ,  $NT_{la} = 3$ ,  $G_0 = 90 * 90$ .

### 4.2.2 Précision du champ de vitesse

Ce cas a été choisi car il existe une solution analytique du champ de vitesse. Dans ce cas, celle-ci est obtenue en imposant cette solution sur les limites et en introduisant un terme source qui permet de converger vers le champ analytique souhaité. Ainsi, l'imposition des limites joue un rôle essentiel dans la convergence du champ de vitesses vers le champ analytique. On rappelle qu'on utilise une interpolation non conservative pour les nœuds appartenant aux limites des niveaux AMR. On impose donc aux limites de chacun d'eux une valeur du champ des vitesses non analytique. Ainsi, le champ de vitesses des niveaux fin est probablement altéré par l'interpolation. Pour le vérifier, on étudie l'évolution des erreurs sur le niveau grossier  $G_0 = 90 * 90$  et sur le niveau fin  $G_1$  en fonction de l'interpolation non conservative utilisée ( $Q_1$ ,  $Q_2$  et  $Q_3$ ). On impose trois itérations de lagrangien augmenté avec  $dp = dr = 0.001$  constant et un maximum de vingt itérations

en *BicgStabII*.

On observe sur la figure 4.9 que l'erreur (par rapport à la solution analytique) sur le niveau grossier oscille jusqu'à se stabiliser autour de  $\epsilon_{G_0} = 0.0114$ . L'erreur sur le niveau  $G_1$ , a le même comportement que celle sur  $G_0$  et se stabilise autour de  $\epsilon_{G_1} = 0.0038$ , quelque soit l'interpolation utilisée. On note d'ailleurs pour ce cas que les trois interpolations donnent des comportements de l'erreur sur le niveau fin très similaires pendant la durée des oscillations pour converger vers la même valeur à  $10^{-2}$  près. Le tableau 4.6 montre les erreurs obtenues au temps  $t = 1.1s$ . On peut noter que pour le tourbillon de Green-Taylor, l'utilisation d'une interpolation  $Q_1$ ,  $Q_2$  ou  $Q_3$  divise l'erreur obtenue sur le niveau  $G_1$  par 3 par rapport à  $G_0$ .

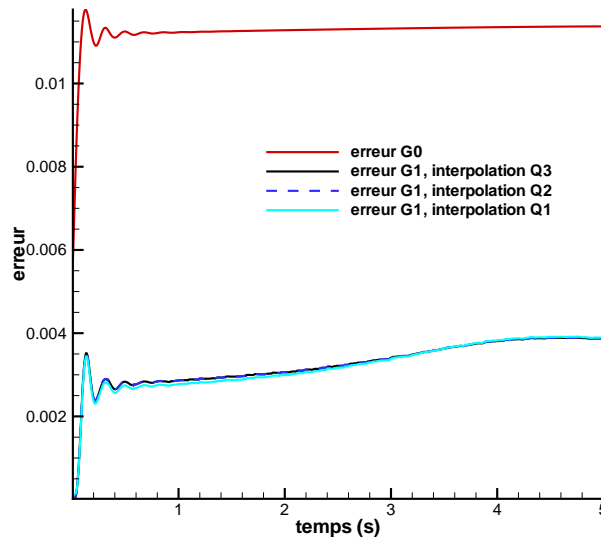


Figure 4.9 : Evolution des erreurs sur le niveau  $G_0$  et  $G_1$  jusqu'au temps  $t = 1.1s$ .

	$\epsilon_{G_0}$	$\epsilon_{G_1, Q_1}$	$\epsilon_{G_1, Q_2}$	$\epsilon_{G_1, Q_3}$
	0.0113726	0.003885	0.003864	0.003864
$\frac{\epsilon_{G_0}}{\epsilon_{G_1}}$	1	2.93	2.94	2.94

Tableau 4.6 : Erreurs faites sur le champ de vitesse sur  $G_0$  et  $G_1$  au temps  $t = 1.1s$

### 4.2.3 Performances de l'AMR en taille mémoire et temps CPU

Dans cette sous-section, on cherche à montrer les performances de l'approche *AMR* en termes de taille mémoire et de temps *CPU* lorsque l'on résout les équations de Navier-Stokes et celle du transport d'interface. Pour cela, on analyse plusieurs paramètres liés au temps de simulation et à la mémoire, puis on compare les résultats avec et sans *AMR*. Pour le cas *AMR* (avec un, deux puis trois niveaux de raffinement), on prend un maillage  $G_0 = 20 * 20$  et un pas de temps  $\Delta t = 0.001s$ . La tache est cisailée pendant 0.5s.

#### Analyse en taille mémoire

On adopte ici la même démarche que dans la sous-section 3.1.6 et on reprend les mêmes notations. En effet, dans un premier temps, on compare la mémoire nécessaire pour simuler le cas *AMR* avec celle utilisée sur un maillage fin équivalent sans *AMR* (voir tableau 4.7).

On note dans le tableau 4.7, que dès le premier niveau fin, le rapport *rap* est strictement inférieur à 1, et continue à diminuer quand le nombre de niveaux *AMR* augmente, ce qui montre que la méthode *AMR* reste moins coûteuse que celle sans. Pour interpréter ces résultats, on établit un lien avec le nombre de nœuds résolus. Étant donné qu'on résout les équations de Navier-Stokes et celle de transport d'interface sur un maillage décalé (*MAC*), la relation est établie sur le nombre total de nœuds résolus (vitesses + scalaire). Soit *KKT* (respectivement *KKV*) le nombre de nœuds scalaires (respectivement vitesses) résolus sur un maillage sans *AMR*. Soit *NBGT<sub>L</sub>* (respectivement *NBGV<sub>L</sub>*) le nombre de nœuds scalaires (respectivement vitesses) nécessaire à la résolution des équations avec la méthode *AMR*. *r<sub>gl</sub>* représente ici le rapport entre le nombre de nœuds résolus avec *AMR* et sans, soit :  $r_{gl} = \frac{NBGT_L + NBGV_L}{KKT + KKV}$ . Ces résultats sont rassemblés dans le tableau 4.8. Soit  $r_{ap_{crit}} = 0.75$ , la valeur critique de *rap* en deçà de laquelle on estime que le gain en taille mémoire est suffisant pour préférer la méthode *AMR*. On considère la fonction  $r_{ap} = f(r_{gl})$ . A partir des données des tableaux 4.7 et 4.8, une régression linéaire donne la droite d'équation  $r_{ap} = 2.55 * r_{gl} - 0.015$  avec un coefficient de corrélation de 0.9996. On en déduit  $r_{gl_{crit}} = 30\%$ .

Ainsi, tant que le nombre de nœuds résolus avec *AMR* est inférieur à 30% du nombre de nœuds qu'il aurait fallu résoudre sans cette méthode sur un maillage fin équivalent, celle-ci permet d'obtenir des gains en taille mémoire très satisfaisants. Ce résultat est d'ailleurs similaire à celui obtenu dans la sous-section 3.1.6, ce qui tend à montrer que

maillage fin	Mémoire avec <i>AMR</i> (Mo)	Mémoire sans <i>AMR</i> (Mo)	rap
60 * 60	7	8	0.9
180 * 180	15	50	0.3
540 * 540	38	420	0.1

Tableau 4.7 : Mémoire utilisée (en Mo) pour simuler un disque cisailé par le tourbillon de Green-Taylor, avec et sans *AMR*, sur différents maillages. Pour les cas *AMR*, le maillage grossier utilisé est  $G_0 = 20 * 20$

maillage fin	$NBGV_L$	$NBGT_L$	$KKV$	$KKT$	$rgl$
60 * 60	2332	1566	7564	3721	0.35
180 * 180	6927	5216	65884	32761	0.12
540 * 540	21716	17066	586444	292681	0.02

Tableau 4.8 : Rapport  $rgl$  pour différents nombres de niveaux *AMR* au temps  $t = s$ .  
Maillage grossier  $G_0 = 20 * 20$

le gain en mémoire apporté par cette méthode dépend essentiellement de  $rgl$  et non du nombre d'équations résolues.

### Analyse en temps CPU

On s'intéresse aux performances de la méthode *AMR* en termes de gain en temps de simulation.

Dans un premier temps on évalue les temps *CPU* moyens utilisés pour la gestion dynamique du raffinement-déraffinement, la résolution de chaque équation et le temps total, en gardant les notations de la sous-section 3.1.6. Étant donné qu'on résout les équations de Navier-Stokes et celle de transport d'interface, on note ici  $t_{resol}$  la somme du temps moyen de résolution des équations de Navier-Stokes (noté  $t_{resol-NS}$ ) et du temps moyen de l'équation d'advection (noté  $t_{resol-ad}$ ).

On observe d'après les figures 4.10, 4.11 et le tableau 4.9 que la résolution des équations de Navier-Stokes est très coûteuse ( $\sim 99\%$  du temps moyen total) devant celle de l'équation d'advection et devant la gestion dynamique de l'*AMR*. Ceci montre que la



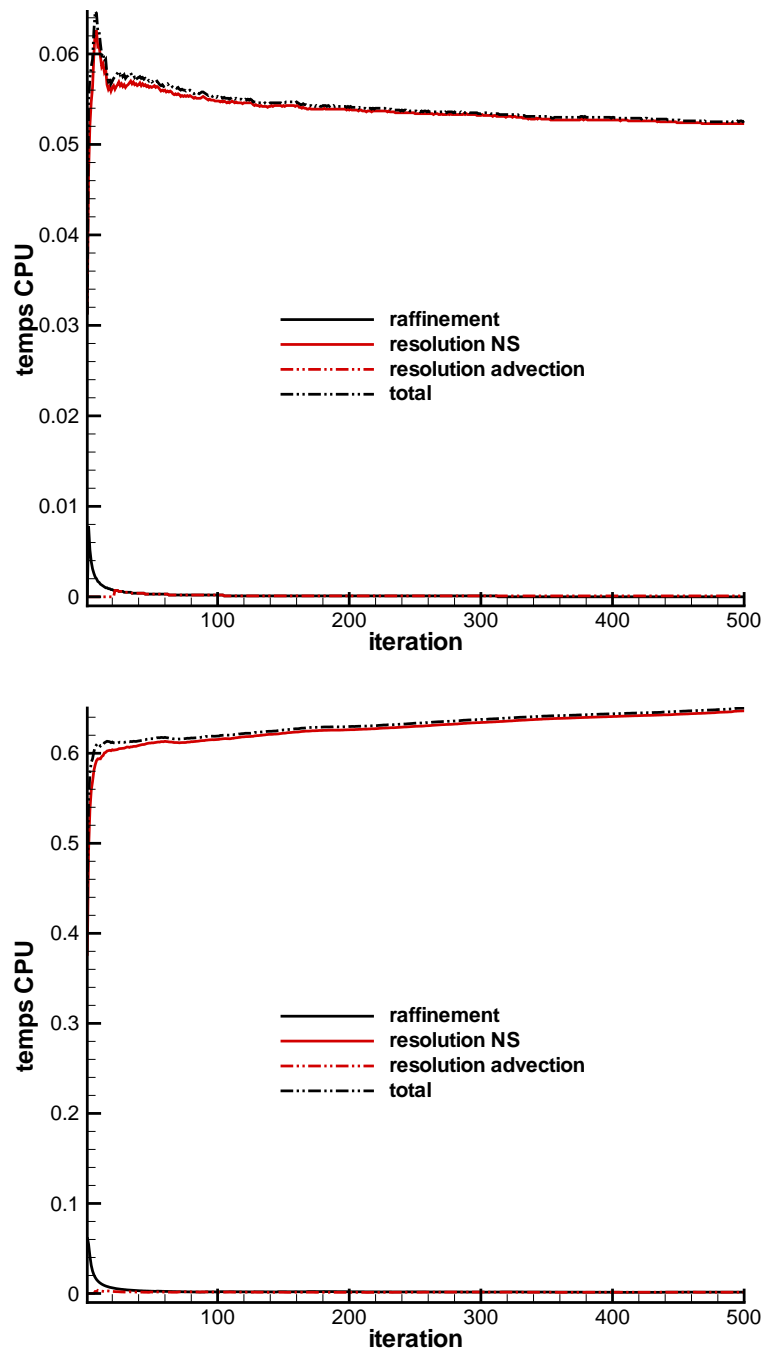


Figure 4.10 : Comparaison des temps CPU moyens avec un niveau d'AMR (resp deux niveaux) et sans AMR en haut (resp en bas).

contribution de cette méthode de raffinement devient négligeable devant la résolution des équations (particulièrement lorsque celles-ci sont complexes à résoudre, cf 3.1.6). Ainsi,

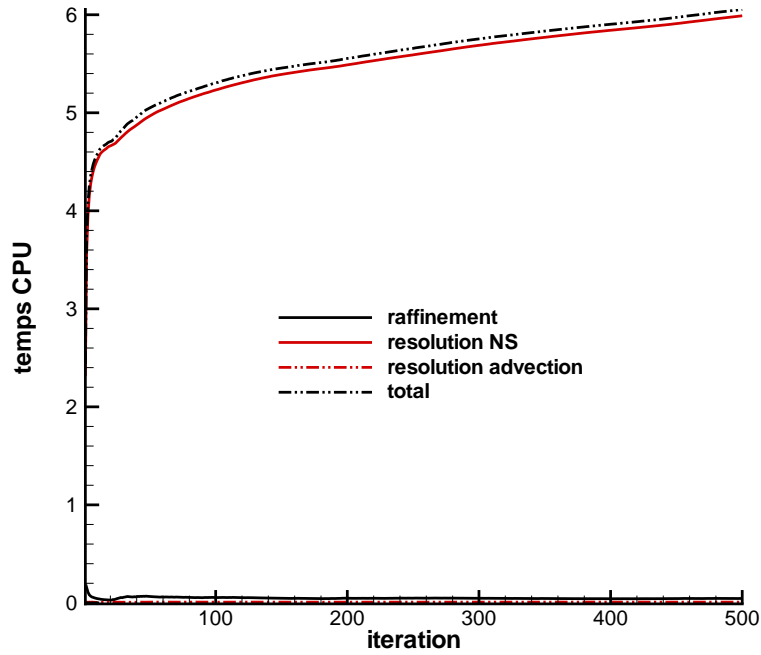


Figure 4.11 : Comparaison des temps *CPU* moyens avec trois niveaux fins et sans *AMR*.

l'approche *AMR* devient intéressante en termes de gain en temps *CPU*. Ceci est confirmé par les résultats du tableau 4.10 qui montrent qu'elle permet de gagner entre 30% et 60% de temps *CPU* pour un niveau (respectivement trois niveaux) de raffinement. Il est cependant difficile de généraliser cette tendance car aucun lien n'a pu être établi entre le rapport du temps *CPU* avec et sans *AMR*, et *r<sub>gl</sub>*.

nombre de niveaux	$t_{raf}$	$t_{resol-NS}$	$t_{resol-ad}$	$t_{total}$	$\frac{t_{raf}}{t_{total}}$	$\frac{t_{resol}}{t_{total}}$
1	0	0.0523	0.0001	0.0526	0	1
2	0.001	0.6471	0.0013	0.6502	0.0023	0.997
3	0.0452	5.9895	0.008	6.0542	0.0075	0.991

Tableau 4.9 : Comparaison des temps *CPU* moyens à l'itération  $NT = 500$  pour différents nombres de niveaux *AMR*.

maillage fin	$t_{AMR}$	$t_{SAMR}$	$rt$
60 * 60	26.44	36.98	0.7
180 * 180	325.2	565.2	0.6
540 * 540	3027	7274	0.4

Tableau 4.10 : Comparaison du temps CPU de simulation avec et sans AMR. Maillage grossier  $G_0 = 20 * 20$ .

### 4.3 Ruptures de barrage en 2D

On aborde dans cette section, des cas de rupture de barrage en diphasique. Initialement, une retenue d'eau est présente dans un domaine rectangulaire. L'effondrement de la colonne d'eau se fait soit sur fond sec (c'est-à-dire qu'il n'y a pas d'eau dans la zone en aval de la colonne d'eau), soit sur fond mouillé (c'est-à-dire qu'il y a de l'eau dans la zone en aval de la colonne d'eau). Le reste du domaine est occupé par de l'air. On résout les équations de Navier-Stokes et celle du transport d'interface. On donne les paramètres suivant :  $\frac{\rho_{eau}}{\rho_{air}} = 850$ ,  $\frac{\mu_{eau}}{\mu_{air}} = 54$ . Pour analyser les résultats obtenus avec la méthode AMR, on les compare aux résultats expérimentaux présentés par [MARTIN 52] sur fond sec et par [STANSBY 98] sur fond mouillé.

#### 4.3.1 Rupture de barrage sur fond sec

On se place dans les mêmes conditions que celles décrites dans [MARTIN 52]. On considère un domaine rectangulaire 2D de dimensions  $\Omega = [0, 1.2] \times [0, 0.14]$ . Une colonne d'eau, d'épaisseur  $a = 0.057m$  et de hauteur  $H_0 = n^2 a$ , où  $n$  est un réel, est initialement présente sur la partie gauche du domaine  $\Omega$ . La rupture du barrage se fait sur fond sec. On s'intéresse ici à la capacité de la méthode AMR à appréhender ce type d'écoulement. On compare donc l'avancée du front d'eau  $z$  ainsi que la hauteur résiduelle de la colonne d'eau  $h$  au cours du temps obtenus avec la méthode de raffinement aux résultats expérimentaux de [MARTIN 52]. Les deux cas étudiés diffèrent de part la hauteur d'eau initiale  $H_0$ . Dans le premier cas, on choisit  $n^2 = 1$  et dans le deuxième  $n^2 = 2$ . On adopte les notations suivantes :  $T_a = nt\sqrt{\frac{g}{a}}$ ,  $Z = \frac{z}{a}$ ,  $H = \frac{h}{an^2}$ . On choisit de simuler le cas  $n^2 = 1$  en partant d'un maillage  $G_0 = 100 * 30$  avec un niveau

de raffinement pendant un temps  $T_a = 3.3$ .

La figure 4.12 montre les différentes étapes de la rupture de barrage sur fond sec pour

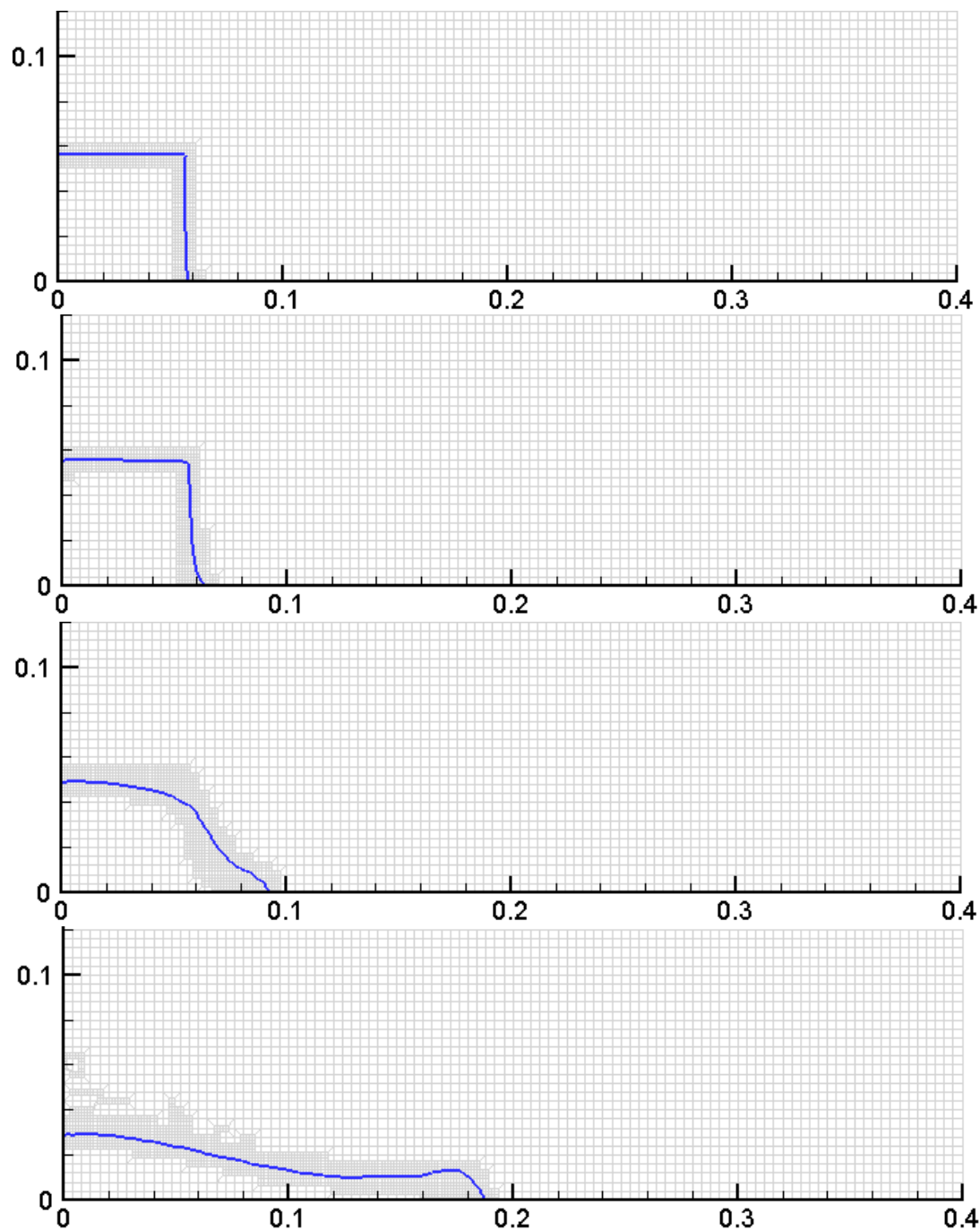


Figure 4.12 : Avancée de l'interface à  $t = 0.01s$ ,  $t = 0.02s$ ,  $t = 0.06s$  puis  $t = 0.16s$  (de haut en bas) pour la rupture de barrage sur fond sec avec  $n^2 = 1$ .  $G_0 = 100 * 30$  avec un niveau de raffinement,  $\Delta t = 2.10^{-4}s$ .

$n^2 = 1$ . On note un effondrement de la colonne d'eau initiale qui se propage sur le fond initialement sec (propagation d'une onde de détente).

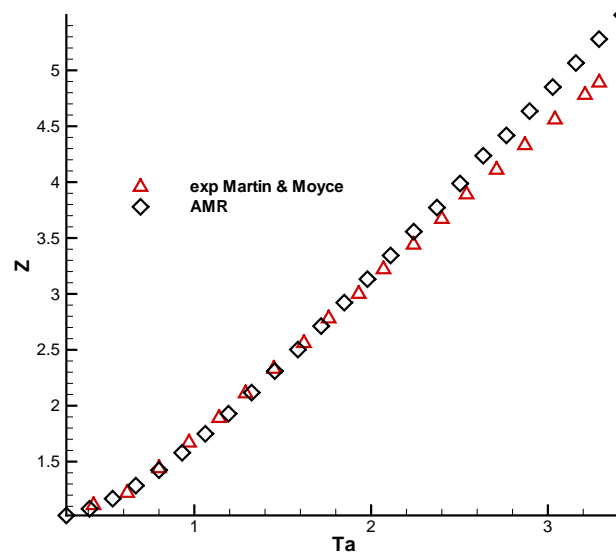


Figure 4.13 : Position adimensionnée de l'interface  $Z$  pour la rupture de barrage sur fond sec avec  $n^2 = 1$ .  $G_0 = 100 * 30$  avec un niveau de raffinement,  $\Delta t = 2.10^{-4}s$ .

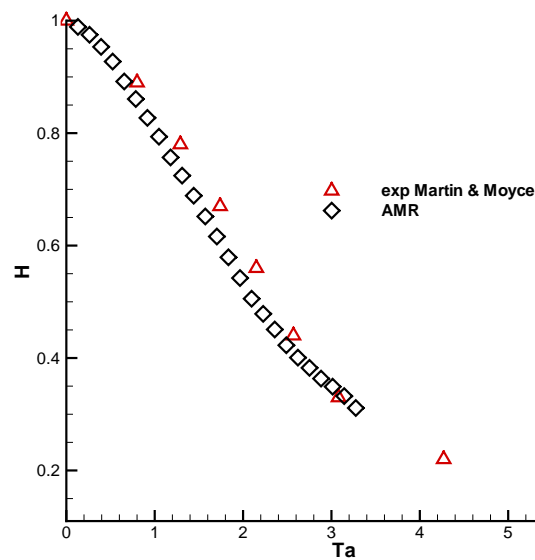


Figure 4.14 : Hauteur d'eau adimensionnée de l'interface  $H$  pour la rupture de barrage sur fond sec avec  $n^2 = 1$ .  $G_0 = 100 * 30$  avec un niveau de raffinement,  $\Delta t = 2.10^{-4}s$ .

On observe sur la figure 4.13 que la vitesse du front augmente jusqu'à  $T_a = 0.5$ , puis atteint une vitesse constante. Les résultats expérimentaux et ceux obtenus avec *AMR* sont très proches jusqu'à  $T_a = 2.4$  puis les courbes s'écartent pour atteindre une erreur de 8% au temps final de simulation.

Quant à l'évolution de la hauteur d'eau de la colonne résiduelle, la figure 4.14 montre un comportement similaire avec un maximum d'erreur de 6%. Ainsi la méthode choisie permet d'appréhender ce cas de rupture de façon satisfaisante.

De même que pour  $n^2 = 1$  on simule le cas  $n^2 = 2$  en partant d'un maillage  $G_0 = 100 * 30$  avec un niveau de raffinement pendant un temps  $T_a = 3.3$ .

La figure 4.16 montre l'affaissement de la colonne d'eau qui génère une onde de détente. Cette fois-ci, on compare les résultats provenant de trois sources : les expériences de [MARTIN 52], les simulations numériques de [ISHII 06] et la méthode *AMR*. Concernant

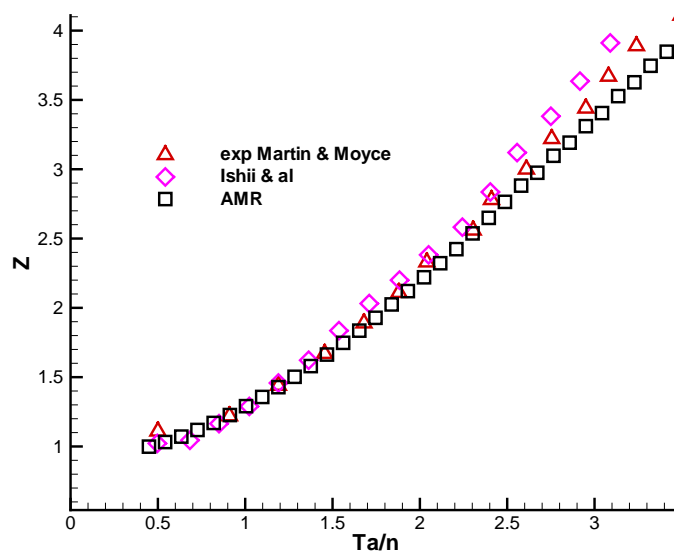


Figure 4.15 : Position adimensionnée de l'interface  $Z$  pour la rupture de barrage sur fond sec avec  $n^2 = 2$ .  $G_0 = 100 * 30$  avec un niveau de raffinement,  $\Delta t = 10^{-4}s$ .

la variation de la hauteur d'eau, on constate sur la figure 4.17 une diminution linéaire au cours du temps jusqu'à un ralentissement de cette baisse quand la hauteur résiduelle de la colonne d'eau passe en dessous de  $H = 0.4$ . Les comportements des deux méthodes numériques sont très proches de l'expérience. On note aussi sur la figure 4.15 une bonne appréhension de l'avancée du front puisque l'erreur entre la simulation et l'expérience

s'élève à 6% quelsoit la méthode utilisée. On note cependant que l'*AMR* a tendance à sous-estimer l'avancée du front tandis que l'approche utilisée par [ISHII 06] la sur-estime.

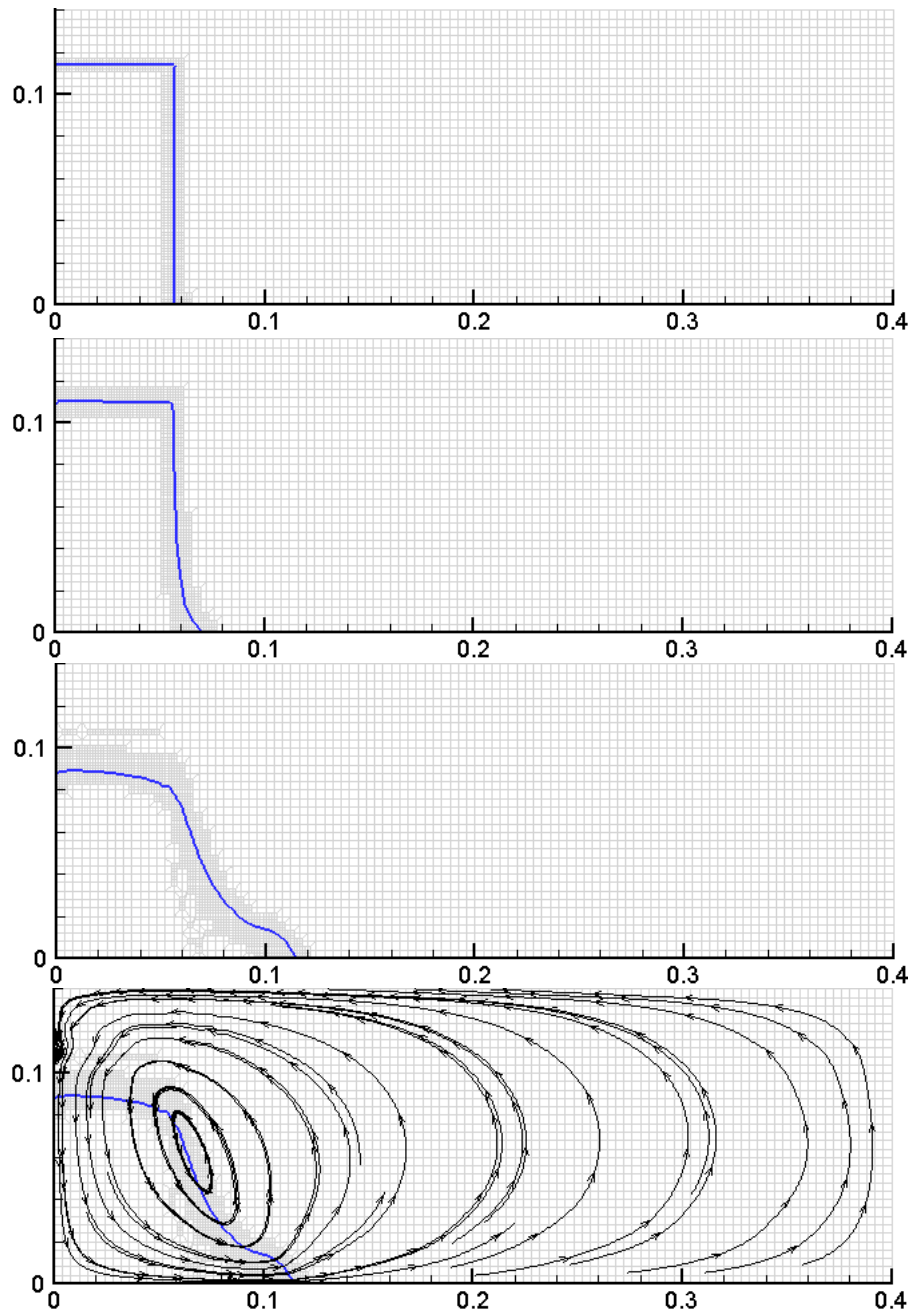


Figure 4.16 : Avancée de l'interface à  $t = 0.005s$ ,  $t = 0.03s$  puis  $t = 0.08s$  (de haut en bas) pour la rupture de barrage sur fond sec avec  $n^2 = 2$ . Lignes de courant pour  $t = 0.08s$ .  $G_0 = 100 * 30$  avec un niveau de raffinement,  $\Delta t = 10^{-4}s$ .

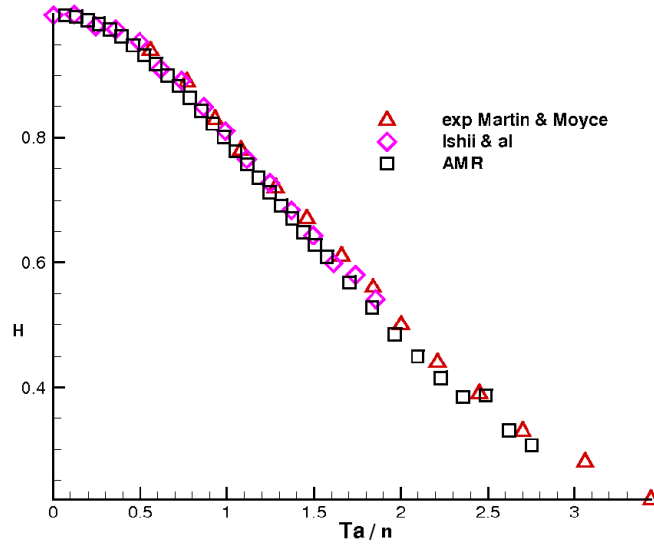


Figure 4.17 : Hauteur d'eau adimensionnée de l'interface  $H$  pour la rupture de barrage sur fond sec avec  $n^2 = 2$ .  $G_0 = 100 * 30$  avec un niveau de raffinement,  $\Delta t = 10^{-4}s$ .

### 4.3.2 Rupture de barrage sur fond mouillé

On se place dans les mêmes conditions que celles décrites dans [STANSBY 98]. On considère un domaine rectangulaire  $2D$  de dimensions  $\Omega = [0, 1.2] \times [0, 0.14]$ . Une colonne d'eau d'épaisseur  $a = 0.6m$  et de hauteur  $H_0 = 0.1m$ , est initialement présente sur la partie gauche d' $\Omega$ .

On s'intéresse ici à la capacité de la méthode *AMR* à appréhender l'écoulement induit par la rupture d'une onde de barrage sur un fond mouillé de hauteur  $h_f$ . On compare donc l'avancée du front d'eau  $z$  ainsi que la hauteur résiduelle de la colonne d'eau  $h$  au cours du temps obtenus avec la méthode de raffinement aux résultats expérimentaux de [STANSBY 98] et aux résultats numériques de [VINCENT 99]. Les deux cas étudiés diffèrent de part la hauteur d'eau initiale  $h_f$ . Dans le premier cas, on choisit  $h_f = 0.01m$  et dans le deuxième  $h_f = 0.045m$ . On considère un maillage  $G_0 = 400 * 47$  avec un niveau de raffinement.

- Dans le cas d'un fond initialement mouillé de hauteur  $h_f = 0.01m$ , on observe sur la figure 4.18 un affaissement de la colonne d'eau sous l'effet de l'apesanteur, génération d'un jet, puis le déferlement (propagation d'une onde de choc). La figure 4.20 montre



que les différentes méthodes donnent une bonne estimation de la zone de déferlement autour de  $x = 0.8m$  avec une erreur entre 1% et 3%. Cependant l'estimation de la hauteur de la surface libre dans la zone de déferlement est moins bien estimée, peut être du fait de la difficulté à évaluer ce paramètre expérimentalement (voir photos dans [STANSBY 98]).

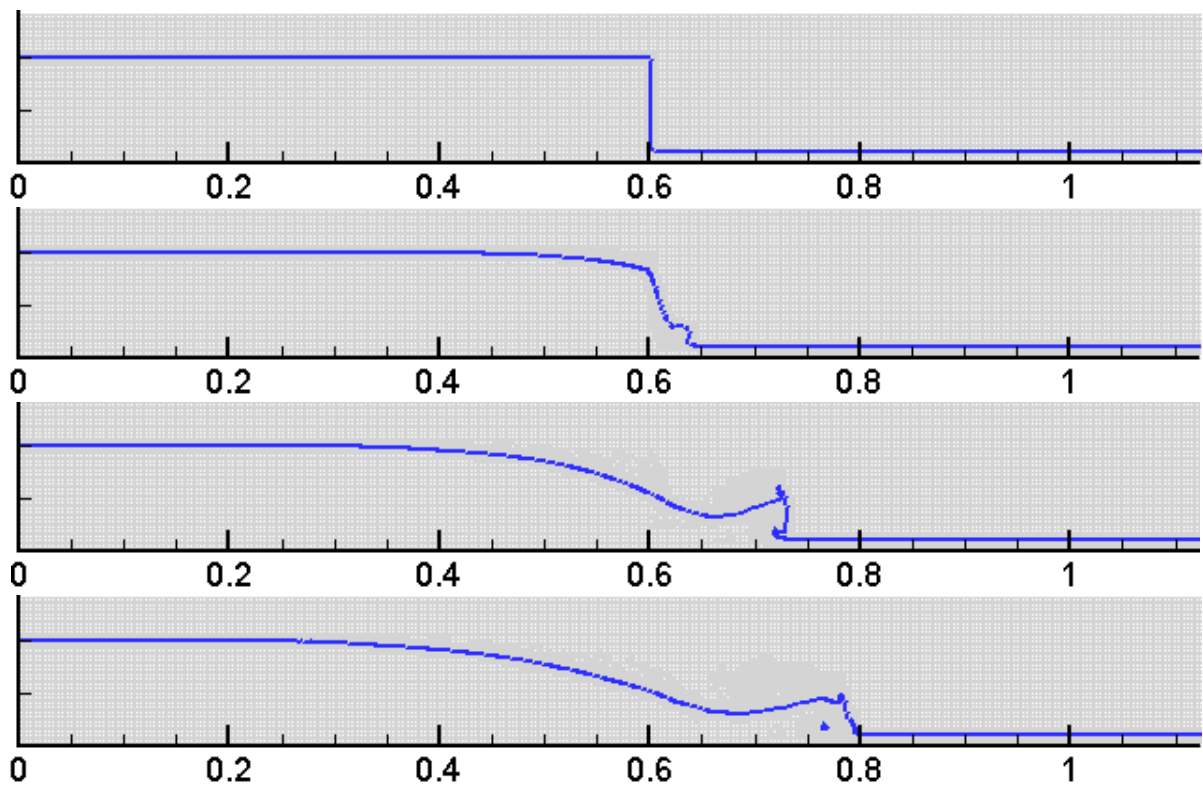


Figure 4.18 : Avancée de l'interface à  $t = 0.01s$ ,  $t = 0.06s$ ,  $t = 0.16s$  puis  $t = 0.22s$  (de haut en bas).  $G_0 = 400 * 47$  avec un niveau de raffinement,  $\Delta t = 2 \cdot 10^{-4}s$ . Rupture sur fond mouillé,  $h_f = 0.01m$ .

- Dans le cas d'un fond initialement mouillé de hauteur  $h_f = 0.045m$ , on observe

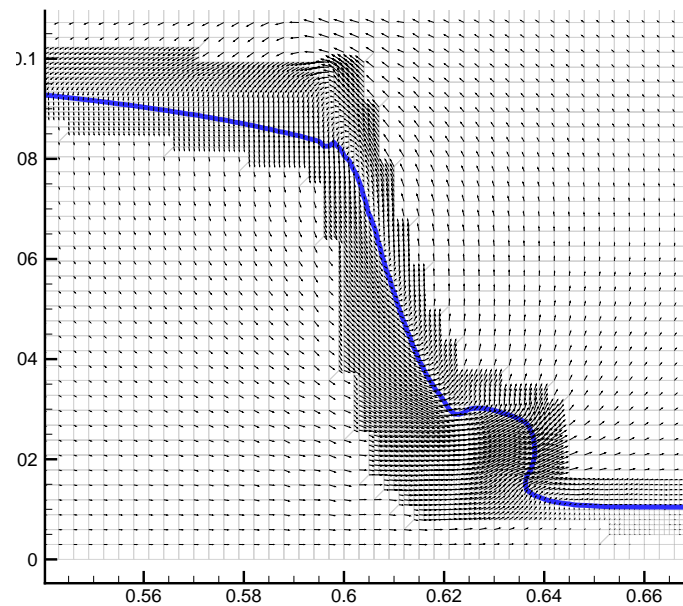


Figure 4.19 : *Champ de vitesses à  $t = 0.06s$ .  $G_0 = 400 * 47$  avec un niveau de raffinement,  $\Delta t = 2 \cdot 10^{-4}s$ . Rupture sur fond mouillé,  $h_f = 0.01m$ .*

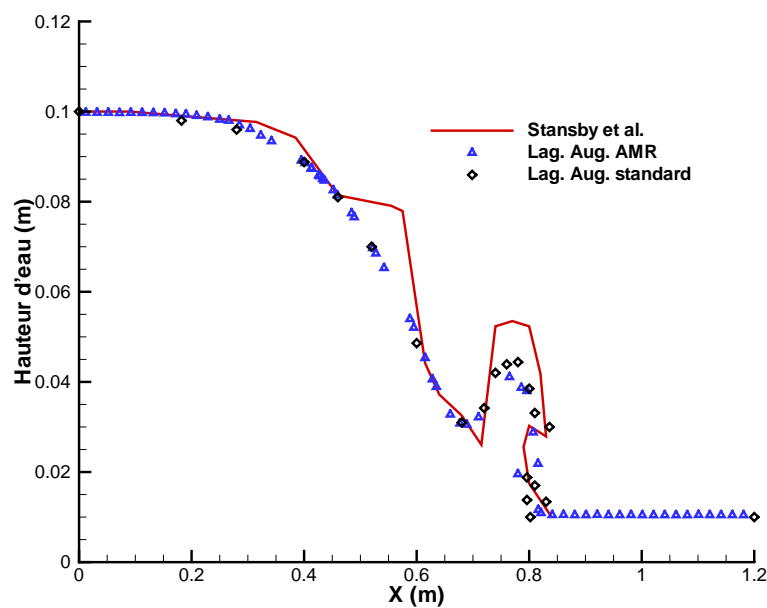


Figure 4.20 : *Hauteur d'eau dans le domaine à l'instant  $t = 0.24s$ .  $G_0 = 400 * 47$  avec un niveau de raffinement,  $\Delta t = 2 \cdot 10^{-4}s$ . Rupture sur fond mouillé,  $h_f = 0.01m$ . Rupture sur fond mouillé,  $h_f = 0.01m$ .*

sur la figure 4.21 différentes étapes de l'écoulement. On peut noter l'affaissement de la colonne d'eau sous l'effet de la gravité, puis génération d'un jet à cause de la présence d'un fond d'eau  $h_f = 0.045m$ . Celle-ci étant plus élevée que dans le premier cas de rupture de barrage sur fond mouillé, la lèvre qui s'est formée suit l'écoulement mais ne déferle pas. Cette onde qui se développe à la surface libre correspond à la propagation d'un choc de gauche à droite tandis qu'une onde de détente se déplace dans la direction inverse. Les figures 4.23 et 4.24 montrent que les résultats sur l'épaisseur de cette lèvre sont corrects, cependant sa hauteur est très sur-estimée.

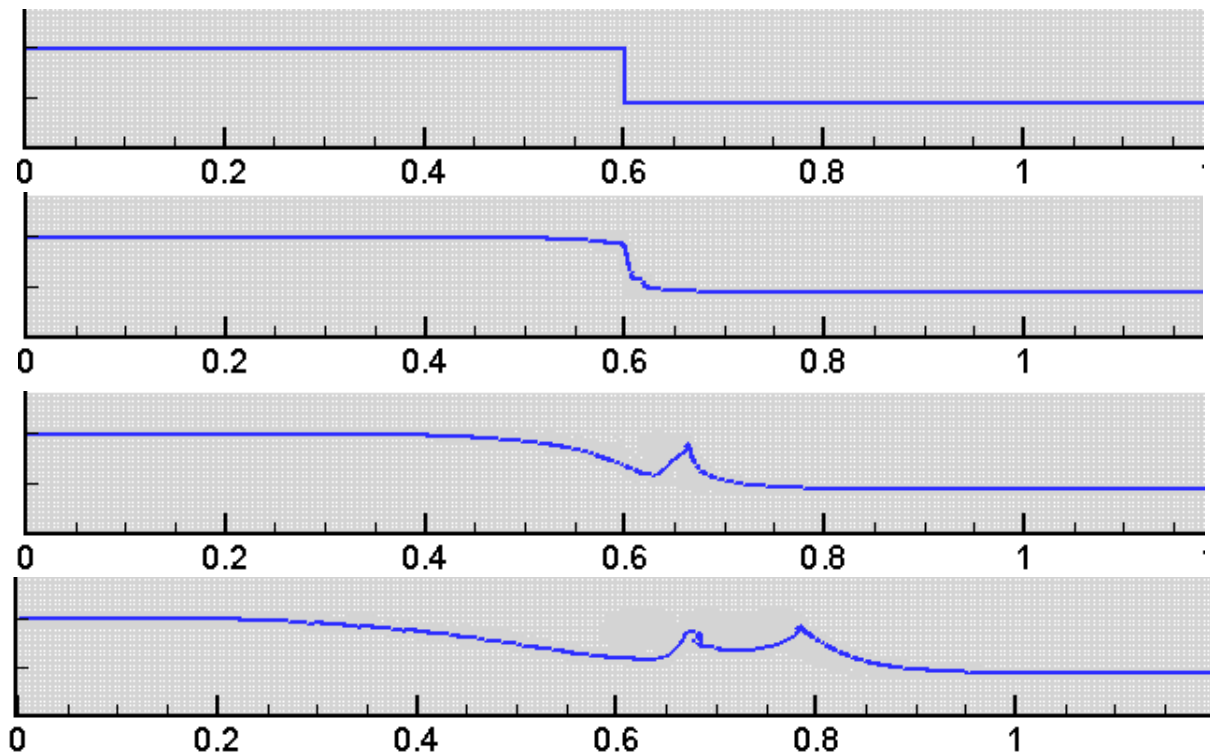


Figure 4.21 : Avancée de l'interface à  $t = 0.005s$ ,  $t = 0.04s$ ,  $t = 0.12s$  puis  $t = 0.3s$  (de haut en bas).  $G_0 = 400 * 47$  avec un niveau de raffinement,  $\Delta t = 10^{-4}s$ . Rupture sur fond mouillé,  $h_f = 0.045m$ .

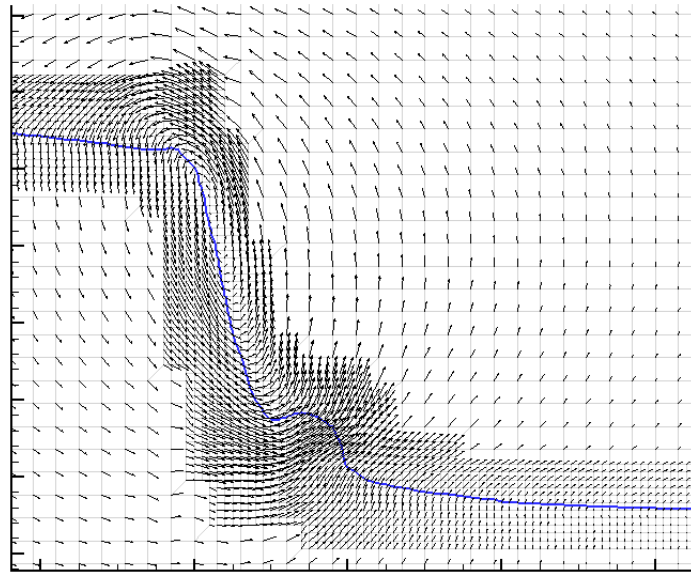


Figure 4.22 : *Champ de vitesses à  $t = 0.04s$ .  $G_0 = 400 * 47$  avec un niveau de raffinement,  $\Delta t = 10^{-4}s$ . Rupture sur fond mouillé,  $h_f = 0.045m$ .*

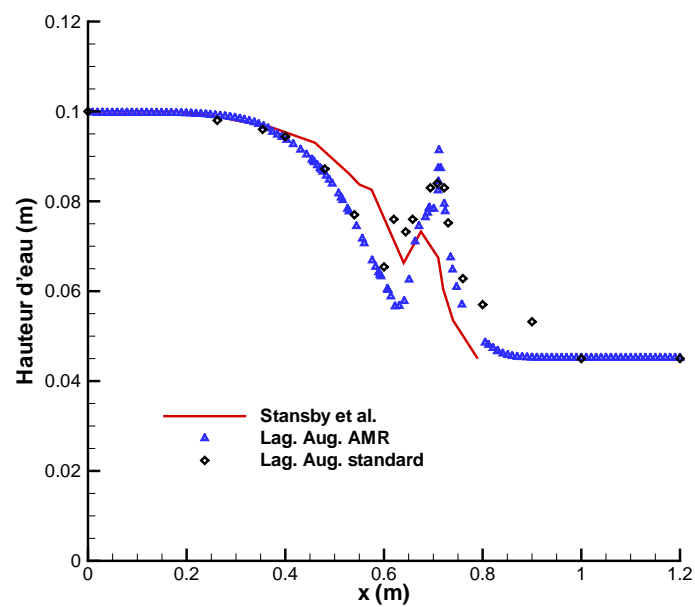


Figure 4.23 : *Hauteur d'eau dans le domaine à l'instant  $t = 0.2s$ .  $G_0 = 400 * 47$  avec un niveau de raffinement,  $\Delta t = 10^{-4}s$ . Rupture sur fond mouillé,  $h_f = 0.045m$ .*

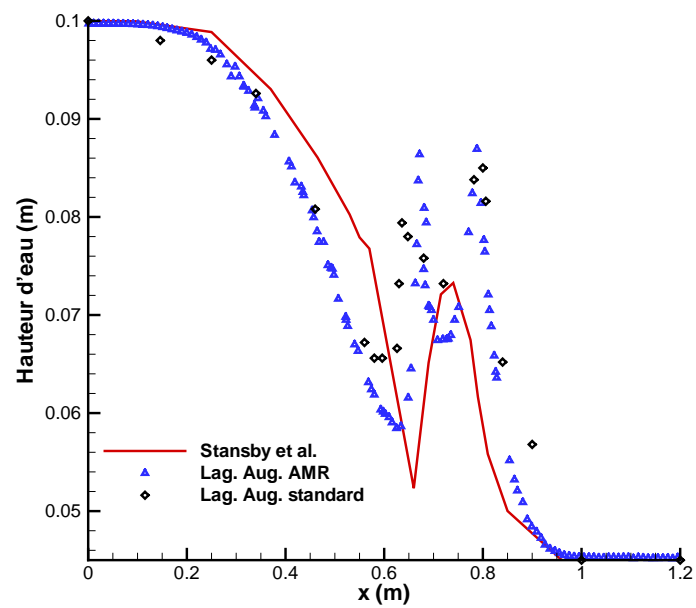


Figure 4.24 : Hauteur d'eau dans le domaine à l'instant  $t = 0.3s$ .  $G_0 = 400 * 47$  avec un niveau de raffinement,  $\Delta t = 10^{-4}s$ . Rupture sur fond mouillé,  $h_f = 0.045m$ .



## Conclusions et perspectives

Cette thèse avait pour objectif d'améliorer et d'optimiser la méthode *OCLM* pour mieux décrire les interfaces dans les écoulements incompressibles diphasiques et le transport d'espèces inertes faiblement diffusives dans les écoulements monophasiques.

L'originalité de la méthode de raffinement de maillage adaptatif (*AMR*) implémentée dans le code *Aquilon* au cours de la thèse, réside en trois points :

- Le premier est la structure même des niveaux de raffinement, que l'on peut considérer comme une entité dont la cohésion est assurée par les connexions entre cellules *AMR*.
- Le deuxième est l'implication de la méthode *AMR*, qui permet le passage d'informations entre niveaux de raffinement et entre cellules *AMR* d'un même niveau au même instant.
- Le troisième est le couplage de la méthode du lagrangien augmenté avec la méthode *AMR* en implicite.

Sur un cas de tache cisailée en monophasique, on a montré l'efficacité de l'approche développée pour réduire la diffusion des schémas et conserver la précision des solutions et l'ordre de convergence des schémas, lorsque le champ de vitesse est analytique. Lorsque le champ de vitesse est interpolé sur les niveaux fins, il s'est avéré nécessaire de partir d'un maillage  $G_0$  assez fin pour obtenir une précision des solutions satisfaisante. Cette observation était prévisible dans la mesure où les gradients de concentration d'espèce sont contenus à l'intérieur des niveaux de raffinement, contrairement au champ de vitesse. Des conclusions similaires ont été obtenues sur le même cas de tache cisailée en diphasique. Les premiers résultats en trois dimensions d'espace pour les cas de tache cisailée en monophasique et diphasique ont montré que la méthode *AMR* permet de réduire la diffusion numérique de façon efficace et ne dégrade pas la précision des solutions.

Pour estimer l'influence de l'interpolation non conservative sur la précision des solutions dans le cas où les gradients de cette quantité ne sont pas tous contenus dans les niveaux fins, une étude sur un cas de thermique a été menée (cas de Whitaker). Les résultats ont permis de montrer que le caractère non conservatif des interpolations de type  $Q_2$  altère la précision des solutions, l'ordre de convergence des schémas et ne conserve pas l'énergie. Ainsi, il est nécessaire d'utiliser des maillages  $G_0$  très fins pour espérer obtenir des solutions avec une précision satisfaisante quand ce type d'interpolation (non conservative) est choisi. Pour essayer de palier à ce problème, on a utilisé une méthode de correction des flux aux interfaces (*FIC*) des niveaux  $G_{L-1}/G_L$ , initialement proposée par [ANGOT 92]. Celle-ci s'est avérée plus efficace car elle permet de conserver l'ordre de convergence des schémas, ne dégrade pas la précision des solutions et conserve l'énergie.

L'approche *AMR* a ensuite été appliquée à la résolution des équations de Navier-Stokes en deux dimensions d'espace au travers de l'étude de trois cas : Poiseuille, tourbillon de Green-Taylor et rupture de barrage. La principale difficulté a été le couplage de cette approche avec la méthode du Lagrangien augmenté en implicite. Des résultats satisfaisants ont pu être obtenus à condition que le maillage de départ soit assez fin.

Pour l'ensemble des cas bidimensionnels, la méthode *AMR* a montré des gains en taille mémoire très intéressants tant que le nombre de points résolus reste inférieur à 30% du nombre de nœuds qu'il aurait fallu résoudre sur un maillage fin équivalent sans *AMR*, et ce quelquesoit l'équation résolue. En termes de gain en temps *CPU*, ses performances augmentent quand la complexité de l'équation à résoudre augmente. Ainsi, les gains en temps sont nettement plus importants lorsque l'on résout les équations de Navier-Stokes par rapport à une équation d'advection-diffusion pour des quantités scalaires.

Lorsque l'on passe à un cas tridimensionnel (sphère cisailée), les performances en termes de gain en taille mémoire sont fortement accrues puisque la méthode reste efficace ( $rgl < 0.75$ ) tant que le nombre de nœuds résolus avec l'*AMR* reste inférieur à 60% des nœuds qu'il aurait fallu résoudre sur un maillage fin équivalent contre 30% en deux dimensions d'espace.

Ainsi, les résultats obtenus au cours de ces travaux ont montré que cette méthode peut s'avérer très performante en termes de gains en taille mémoire et en temps *CPU*, tout en conservant une bonne qualité des solutions.

En perspective, on peut citer la mise en place d'une interpolation conservative (voir annexe C) pour les nœuds vitesses appartenant à l'interface de deux niveaux *AMR*. Ceci permettra d'avoir des écoulements mieux résolus localement, et donc un déplacement d'in-



---

terface plus précis. On envisage aussi l'extension de la méthode *AMR* aux équations de Navier-Stokes en trois dimensions d'espace. Il serait également intéressant de paralléliser la méthode.

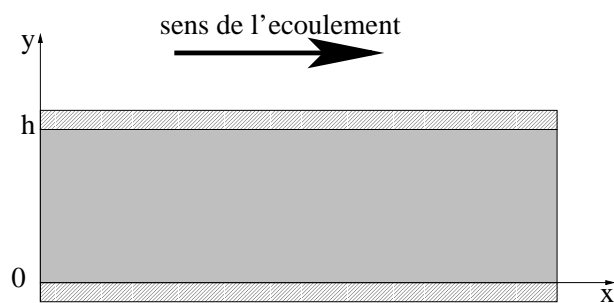
Des simulations sur des cas académiques de changement de phase ainsi que sur des injections de polluant dans une cavité de dimension d'espace deux sont en cours.



# Annexe A

## Solutions analytiques

### A.1 Écoulement de Poiseuille 2D



Soit un fluide incompressible se déplaçant entre deux plans horizontaux, distant d'une hauteur  $h$ . On considère un écoulement plan et on néglige la pesanteur. On s'intéresse de plus au cas stationnaire.

Les équations de Navier-Stokes se réduisent alors à :

$$\left\{ \begin{array}{l} \frac{\partial u}{\partial x} = 0, \\ -\frac{\partial p}{\partial x} + \nu \frac{\partial^2 u}{\partial z^2} = 0, \\ -\frac{\partial p}{\partial z} = 0, \end{array} \right. \quad (\text{A.1})$$

On impose sur les bords du domaine  $u(x, z = 0) = 0$  et  $u(x, z = h) = 0$ .

En résolvant les équations de Navier-Stokes en 2D, on obtient la solution analytique :

$$\begin{cases} u(x, z) = \frac{1}{2\nu} \frac{dp}{dx} z(h - z), \\ v(x, z) = 0, \end{cases} \quad (\text{A.2})$$

Comme il est plus facile d'imposer la vitesse moyenne  $U_{moyen}$  que le gradient de pression  $\frac{dp}{dx}$ , on va trouver une relation entre les deux grandeurs.

$$U_{moyen} = \frac{1}{h} \int_0^h \frac{1}{2\nu} \frac{dp}{dx} z(h - z) dz$$

donc

$$U_{moyen} = \frac{1}{2\nu} \frac{dp}{dx} \frac{h^2}{6}$$

d'où

$$\begin{cases} u(x, z) = U_{moyen} \frac{6}{h^2} z(h - z), \\ v(x, z) = 0. \end{cases} \quad (\text{A.3})$$

## A.2 Conduction 2D dans un rectangle dont les bords sont chauffés à des températures différentes

[WHITAKER 83] a proposé des solutions à ce cas en 1983. On considère un rectangle de hauteur  $H$  et de longueur  $b$  dont trois bords (gauche, droit, inférieur) sont maintenus à une température  $T_0$  et le dernier bord (supérieur) à une température  $T_f(x)$ . On place un fluide incompressible à l'intérieur de ce domaine (figure A.1).

### A.2.1 Cas où $T_f$ est constante par morceaux

Pour tout  $x$  appartenant au bord supérieur, on impose une température constante  $T_f(x) = T_1$ , avec  $T_1 > T_0$ . On crée donc une discontinuité au niveau des coins supérieurs

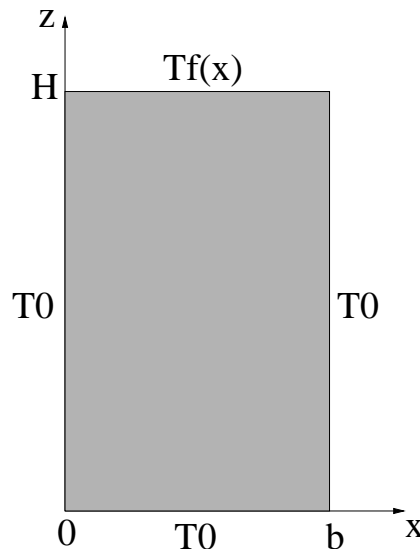


Figure A.1 : Conduction 2D dans un rectangle dont les bords sont chauffés à des températures différentes.

(figure A.2). On se place en régime stationnaire et on considère que le terme d'advection de l'équation de conservation de l'énergie est nul. Celle-ci se réduit donc au seul terme de conduction, ce qui donne :

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial z^2} = 0. \quad (\text{A.4})$$

Les conditions aux limites sont de type Dirichlet :

- $T(x = 0, z) = T_0$
- $T(x, z = 0) = T_0$
- $T(x = H, z) = T_0$
- $T(x, z = H) = T_1$

On suppose que la température  $T$  peut s'écrire  $T(x, z) = F(x)G(z)$ . Donc l'équation de l'énergie peut se mettre sous la forme

$$\frac{1}{F} \frac{d^2 F}{dx^2} = -\frac{1}{G} \frac{d^2 G}{dz^2} = C$$

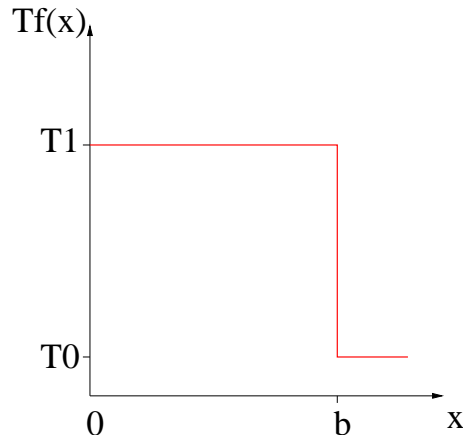


Figure A.2 : Température  $T_f$  imposée sur le bord supérieur du domaine.

Trois valeurs de  $C$  sont alors possibles :

- Si  $C = 0$  alors  $T_a(x, z) = (C_1 + C_2x)(C_3 + C_4z)$ ,
- si  $C > 0$  alors  $T_b(x, z) = (C_5e^{\lambda x} + C_6e^{-\lambda x})(C_7e^{i\lambda z} + C_8e^{-i\lambda z})$ ,
- si  $C < 0$  alors  $T_c(x, z) = (C_9\cos(\lambda x) + C_{10}\sin(\lambda x))(C_{11}\cosh(\lambda z) + C_{12}\sinh(\lambda z))$ .

$T_a$ ,  $T_b$  et  $T_c$  vérifient séparément l'équation différentielle A.4. Cependant, aucune des trois vérifient toutes les conditions aux limites. Seule une combinaison linéaire de  $T_a(x, z)$  et  $T_c(x, z)$ , par exemple  $T(x, z) = T_a(x, z) + T_c(x, z)$  est une solution acceptable, car elle vérifie à la fois l'équation différentielle et l'ensemble des conditions aux limites. Après quelques développements de calculs, cf [WHITAKER 83], on trouve :

$$T(x, z) = T_0 + 2(T_1 - T_0) \sum_{n=1}^{\infty} \frac{(1 - (-1)^n)}{n\pi \sinh(\frac{n\pi H}{b})} \sin(\frac{n\pi x}{b}) \sinh(\frac{n\pi z}{b}), \quad (\text{A.5})$$

### A.2.2 Cas où $T_f$ est de classe $C^1$

On choisit d'imposer une température  $T_f$  de classe  $C^1$  sur le bord supérieur. Au niveau des coins supérieurs, on a  $T(0, H) = T(b, H) = T_0$  et  $\frac{\partial T}{\partial x}(0, H) = \frac{\partial T}{\partial x}(b, H) = 0$ .

Donc la fonction  $T_f$  doit vérifier les mêmes conditions, ie :  $T_f(0) = T_f(b) = T_0$  et  $T_f'(0) =$

$T'_f(b) = 0$ . On fixe de plus que  $T_f$  atteigne son maximum  $T_1$  en  $x = \frac{b}{2}$ , donc  $T_f(\frac{b}{2}) = T_1$ .  
D'où

$$T_f(x) = T_0 - \frac{16}{b^4}(T_0 - T_1)x^2(x - b)^2$$

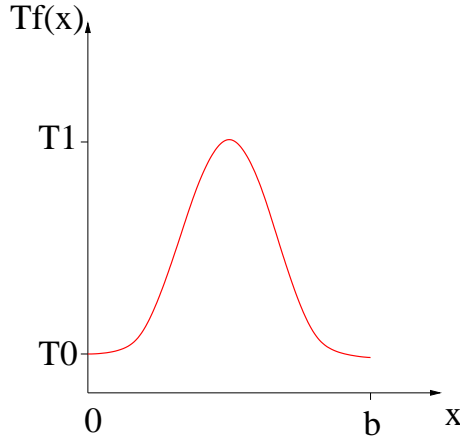


Figure A.3 : Température  $T_f$ , imposée sur le bord supérieur du domaine.

On se place dans les mêmes conditions que dans le cas où la température imposée sur le bord supérieur est constant. On retrouve ainsi l'équation (A.4) à résoudre.

Les conditions aux limites se traduisent par ,

- $T(x = 0, z) = T_0$
- $T(x, z = 0) = T_0$
- $T(x = H, z) = T_0$
- $T(x, z = H) = f(x)$

En suivant le même raisonnement que dans le cas exposé dans la sous section précédente et en se référant à [WHITAKER 83], on en déduit la température qui vérifie l'équation (A.4) et les conditions aux limites. Soit :

$$T(x, z) = T_0 + \sum_{n=1}^{\infty} C_n \sin\left(\frac{n\pi x}{b}\right) \sinh\left(\frac{n\pi z}{b}\right),$$

où

$$C_n = \frac{2}{b \sinh\left(\frac{n\pi H}{b}\right)} \int_0^b (T_f(x) - T_0) \sin\left(\frac{n\pi x}{b}\right) dx.$$

Dans notre cas,

$$C_n = \frac{2}{b \sinh\left(\frac{n\pi H}{b}\right)} \int_0^b \frac{16}{b^4} (T_0 - T_1) x^2 (x - b)^2 \sin\left(\frac{n\pi x}{b}\right) dx.$$

Après quelques intégrations par partie, on obtient :

$$C_n = 4(T_1 - T_0) 16 \frac{(1 - (-1)^n)}{(n\pi)^3 \sinh\left(\frac{n\pi H}{b}\right)} \left(\frac{12}{(n\pi)^2} - 1\right).$$

D'où,

$$T(x, z) = T_0 + 16(T_1 - T_0) \sum_{n=1}^{\infty} \frac{1 - (-1)^n}{(n\pi)^3 \sinh\left(\frac{n\pi H}{b}\right)} \sin\left(\frac{n\pi x}{b}\right) \sinh\left(\frac{n\pi z}{b}\right) \quad (\text{A.6})$$



## Annexe B

# Interpolation non conservative et éléments finis

L'idée de base consiste à approcher une fonction  $u$  (vitesse, pression,...) qui est, soit connue analytiquement mais inexploitable, soit connue en un nombre fini de points. Appelons  $X \in \{x_1, x_2, \dots, x_n\}$  cet ensemble de points.

La fonction d'approximation  $f$  de la fonction  $u$  peut se construire de deux façons différentes :

- Soit on force  $f$  à être égale à  $u$  aux points où cette dernière est connue i.e. pour tout point  $x$  de  $X$ ,  $f(x) = u(x)$ ,
- soit on force  $f$  à être proche de  $u$  aux points où cette dernière est connue i.e. pour tout point  $x$  de  $X$ ,  $\|f(x) - u(x)\| < \epsilon$ ,  $\epsilon$  très petit.

On s'intéresse par la suite à la première construction qui porte le nom d'interpolation. On précise que la fonction d'interpolation sera construite à partir des polynômes de Lagrange. Notons que les fonctions qui seront utilisées ultérieurement seront par exemple la pression, la température, la concentration massique ou les vitesses.

Soit  $dim$  la dimension d'espace. On s'est inspiré des éléments finis de Lagrange dans  $\mathfrak{R}^{dim}$  pour construire les fonctions d'interpolation. On définit un élément fini de la façon suivante :

- soit  $K$  une partie de  $\mathfrak{R}^{dim}$  compacte d'intérieur non vide,
- Soit  $\Sigma = \{a_j\}_{j=1,N}$  un ensemble de  $N$  points distincts de  $K$ ,

- Soit  $P$  un espace vectoriel de fonctions  $f : K \rightarrow \mathfrak{R}$ , de dimension fini. Ici, ce sont des fonctions polynômiales.
- $\Sigma$  est unisolvant, i. e. : si  $\{\alpha_j\}_{j=1,N}$  sont donnés,  $\exists! f \in P$  tel que  $f(a_j) = \alpha_j$ ,  $j = 1, N$ . La base adoptée est alors  $\{f_j\}_{j=1,N}$ ,  $f \in P$  tel que  $f_i(a_j) = \delta_{ij}$ .  $\delta$  est le symbole de Kronecker caractérisé par  $\delta_{ij} = 1$  si  $i = j$  et  $\delta_{ij} = 0$  sinon.

Soit  $u$  une fonction définie sur  $K, \exists! f \in P$  tel que  $u(a_j) = f(a_j)$ ,  $j = 1, N$ .  $f$  interpole  $u$  en un point  $a \in \Omega$  de coordonnées  $a = (x, z)$  en  $2D$  ou  $a = (x, z, y)$  en  $3D$  :

$$f(a) = \sum_{j=1}^N u(a_j) f_j(a).$$

On commence par construire un élément de référence  $(K, \Sigma, P)$ . On lui applique ensuite une transformation affine pour parcourir l'ensemble du domaine étudié. Ayant besoin d'utiliser des maillages à pas d'espace variables (ex : *Tchebicheff*, exponentiel, ...), il est indispensable de choisir des éléments de référence à pas d'espace variables. On précise que les mailles sont quadrangulaires. On teste donc des interpolations de type  $Q_i$ , où  $i$  est le degré du polynôme d'interpolation  $f$ . Les fonctions à interpoler sont soit des scalaires (température, concentration, pression, ... sur la grille de pression), soit des vecteurs (vitesses, ... sur les grilles de vitesses respectives).

### Interpolation pour des pas d'espace variables en 2D

Considérons  $dx_1, dx_2, dx_3$  trois pas d'espace suivant  $x$  et  $dz_1, dz_2, dz_3$  trois pas d'espace suivant  $z$ . On donne ci-après les éléments de références  $(K, \Sigma, P)$  propres à chaque interpolation, ainsi qu'une représentation graphique sur une grille quelconque (pression ou vitesses).

#### Interpolation Q1

L'élément de référence représenté sur la figure B.1 s'écrit :

$$K = [0, dx_1] \times [0, dz_1],$$

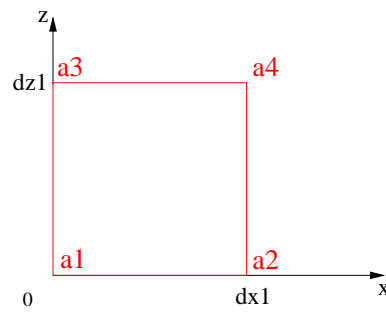
$$P = Q1 = \text{vect}\{1, x, z, xz\},$$

$$\Sigma = \{a_j\}_{j=1,4}.$$

#### Interpolation Q2

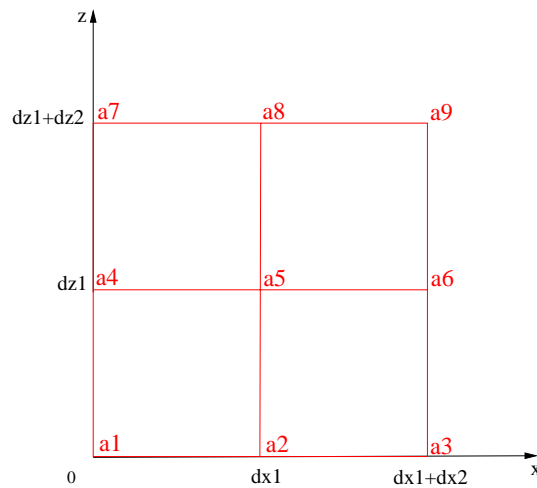
L'élément de référence choisi est (voir aussi la figure B.2) :

$$K = [0, dx_1 + dx_2] \times [0, dz_1 + dz_2],$$

Figure B.1 : Élément de référence pour l'interpolation  $Q1$ 

$$P = Q2 = \text{vect}\{x^j z^k\} \quad j, k \in 0, \dots, 2,$$

$$\Sigma = \{a_j\}_{j=1,9}.$$

Figure B.2 : Élément de référence pour l'interpolation  $Q2$ 

### Interpolation $Q3$

L'élément de référence, représenté sur la figure B.3 est composé de :

$$K = [0, dx_1 + dx_2 + dx_3] \times [0, dz_1 + dz_2 + dz_3],$$

$$P = Q3 = \text{vect}\{x^j z^k\} \quad j, k \in 0, \dots, 3,$$

$$\Sigma = \{a_j\}_{j=1,16}.$$

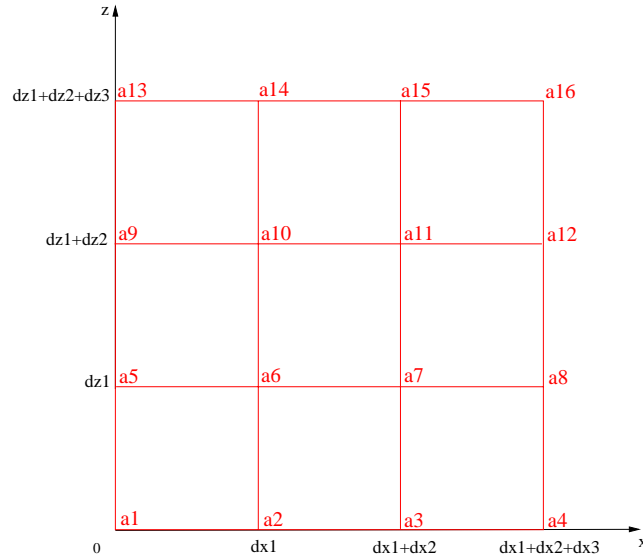


Figure B.3 : Éléments de référence pour l'interpolation  $Q3$

### Interpolation pour des pas d'espace variables en 3D

Considérons  $dx_1, dx_2, dx_3$  trois pas d'espace suivant  $x$ ,  $dz_1, dz_2, dz_3$  trois pas d'espace suivant  $z$  et  $dy_1, dy_2, dy_3$  trois pas d'espace suivant  $y$ . Comme dans le cas 2D, on donne ci-après les éléments de références  $(K, \Sigma, P)$  propre à chaque type d'interpolation, ainsi qu'une représentation graphique sur une grille représentant aussi bien celle de pression que celles des vitesses.

#### Interpolation Q1

L'élément de référence, représenté sur la figure B.4, est construit comme suit :

$$K = [0, dx_1] \times [0, dz_1] \times [0, dy_1],$$

$$P = Q1 = \text{vect}\{x^j z^k y^m\} \quad j, k, m \in 0, \dots, 1,$$

$$\Sigma = \{a_j\}_{j=1,8}.$$

#### Interpolation Q2

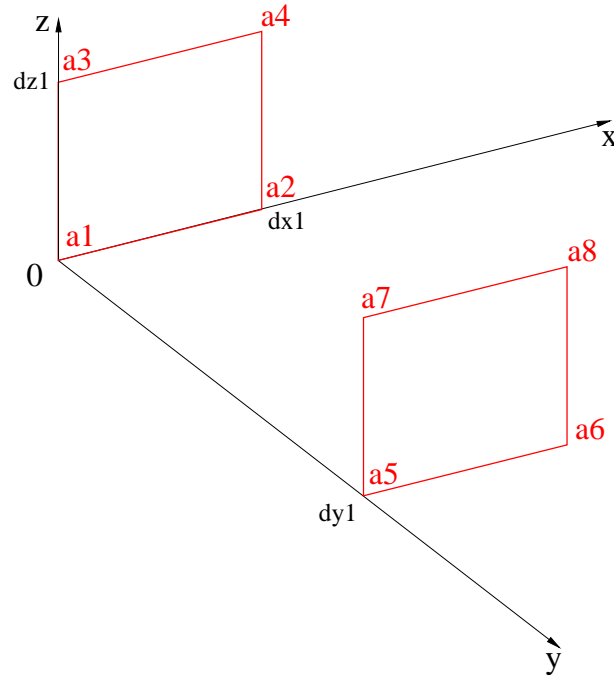
On écrit l'élément de référence de la façon suivante (voir aussi la figure B.5) :

$$K = [0, dx_1 + dx_2] \times [0, dz_1 + dz_2] \times [0, dy_1 + dy_2],$$

$$P = Q2 = \text{vect}\{x^j z^k y^m\} \quad j, k, m \in 0, \dots, 2,$$

$$\Sigma = \{a_j\}_{j=1,27}.$$

#### Interpolation Q3

Figure B.4 : Éléments de référence pour l'interpolation  $Q1$ 

L'élément de référence, représenté sur la figure B.6 s'écrit :

$$K = [0, dx_1 + dx_2 + dx_3] \times [0, dz_1 + dz_2 + dz_3] \times [0, dy_1 + dy_2 + dy_3],$$

$$P = Q3 = \text{vect}\{x^j z^k y^m\} \quad j, k, m \in 0, \dots, 3,$$

$$\Sigma = \{a_j\}_{j=1,64}.$$

## Détermination des polynômes de base

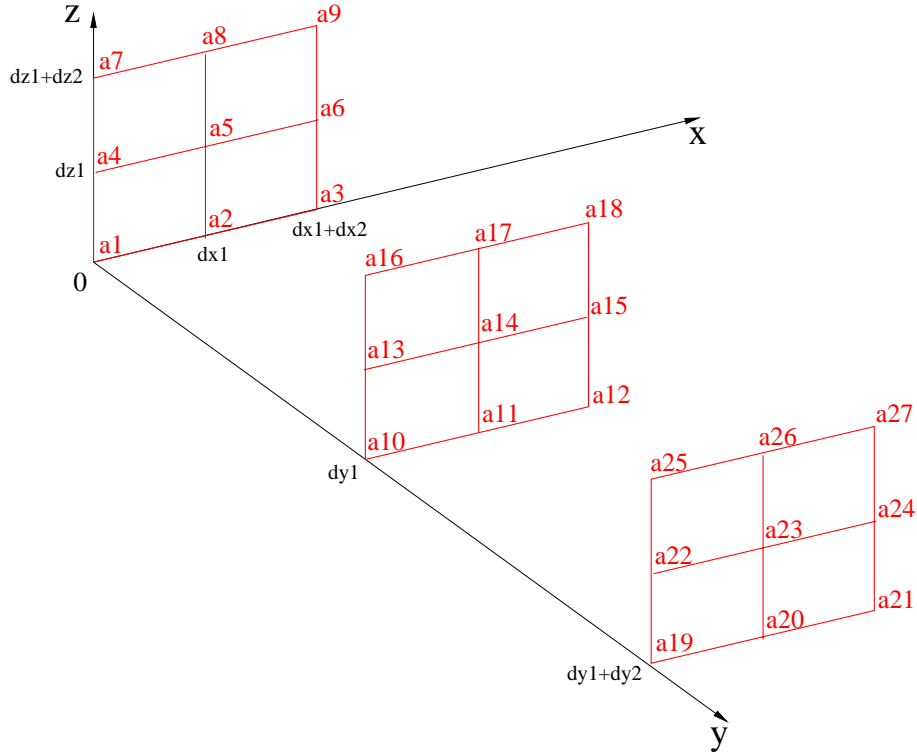
### en 2D

Soit  $n$  l'ordre du polynôme d'interpolation  $f$ . On rappelle que  $Q = \text{vect}(x^l y^m)$  pour tout  $l \in [0, n]$  et  $m \in [0, n]$ , c'est-à-dire qu'il existe un ensemble de réels  $\lambda_{jk}^i$ ,  $j \in [0, 3]$ ,  $k \in [0, 3]$  pour chaque polynôme de base  $f_i$  tel que,  $\forall a \in \Omega$  de coordonnées  $a = (x, z)$  :

$$f_i(a) = f_i(x, z) = \sum_{j=0}^n \sum_{k=0}^n \lambda_{jk}^i x^j z^k$$

Chaque polynôme  $f_i$  vérifie  $f_i(x_m, z_m) = \delta_{im}$  où  $(x_m, z_m)$  représentent les coordonnées du nœud  $a_m$  de la cellule de référence et  $\delta$  le symbole de Kronecker.

Étant donné que la cellule de référence est constituée de  $(n + 1)^2$  nœuds, il suffit de

Figure B.5 : Élément de référence pour l'interpolation  $Q_2$ 

résoudre pour chaque polynôme  $f_i, i \in [1, (n+1)^2]$  les  $(n+1)^2$  équations

$$f_i(x_m, z_m) = \delta_{im}, m \in [1, (n+1)^2]$$

pour trouver les inconnues  $\lambda_{jk}^i, j \in [0, n], k \in [0, n]$ .

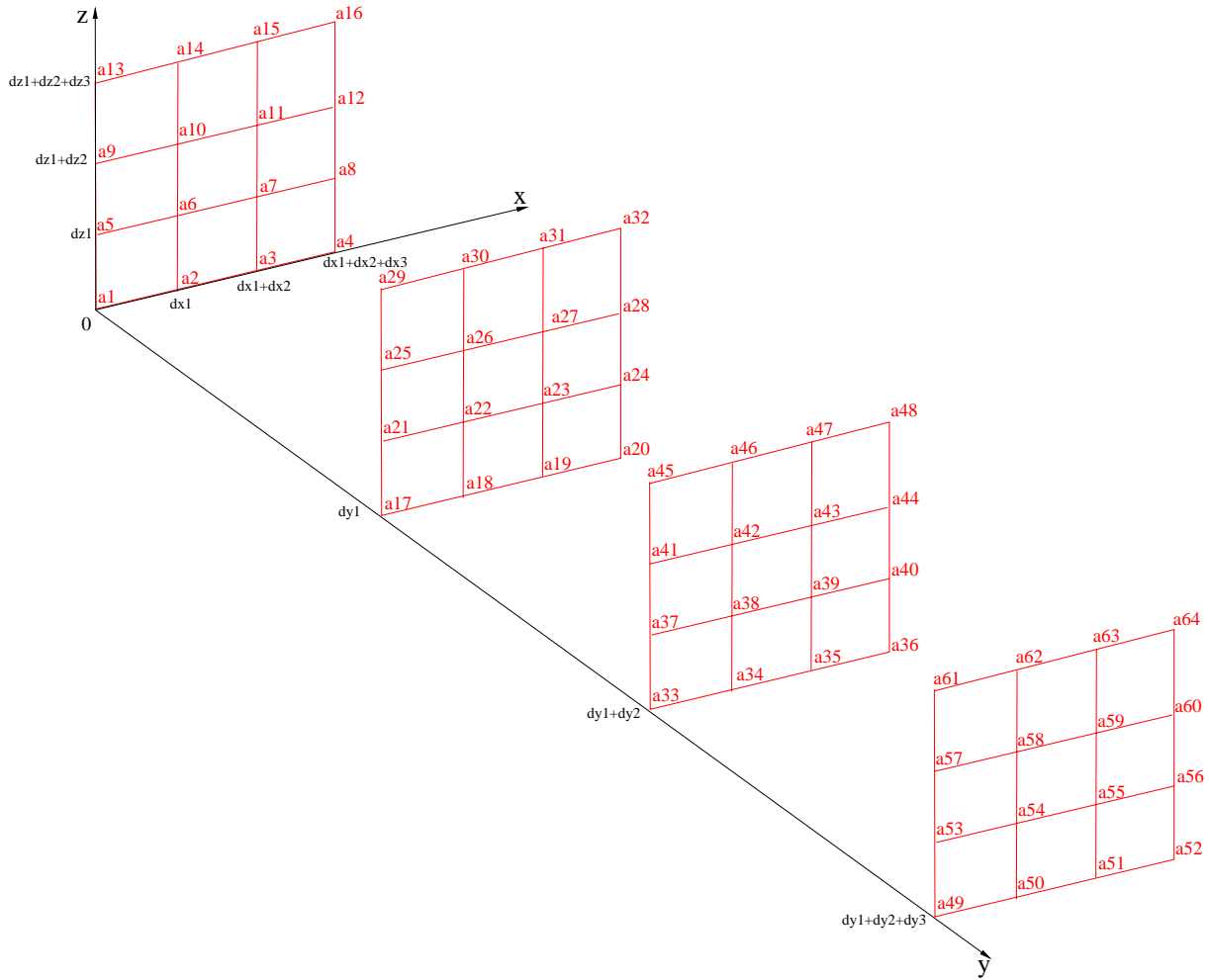
En 2D, la résolution de ces systèmes peut se faire avec le logiciel MAPLE, les résultats étant obtenus instantanément et étant de taille raisonnable pour être ensuite inscrits dans un code en Fortran 90. Soulignons qu'on obtient une formulation analytique des  $\lambda_{jk}^i, j \in [0, n], k \in [0, n]$ .

### en 3D

On désigne par  $n$  l'ordre du polynôme d'interpolation. On rappelle que  $Q = \text{vect}(x^l y^m z^q)$  pour tout  $l \in [0, n], m \in [0, n]$  et  $q \in [0, n]$ . Pour chaque polynôme de base  $f_l, l \in [1, (n+1)^3]$  tel que  $\forall a \in \Omega$  de coordonnées  $a = (x, z, y)$  :

$$f_l(a) = f_l(x, z, y) = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^n \lambda_{ijk}^l x^i y^j z^k$$

Notons que chaque polynôme de base  $p_l$  vérifie  $p_l(x_m, z_m, y_m) = \delta_{lm}, l \in [1, (n+1)^3], m \in [1, (n+1)^3]$  où  $(x_m, z_m, y_m)$  représentent les coordonnées du nœud  $a_m$  de la cellule de

Figure B.6 : Éléments de référence pour l'interpolation  $Q_3$ 

référence.

Sachant qu'il y a  $(n+1)^3$  nœuds dans la cellule de référence, on doit résoudre un système de  $(n+1)^3$  équations à  $(n+1)^3$  inconnues pour chaque polynôme de base. Donc, il faut résoudre en tout  $(n+1)^3 * (n+1)^3$  équations à  $(n+1)^3 * (n+1)^3$  inconnues :

$$f_l(x_m, z_m, y_m) = \delta_{lm}, \forall l \in [1, (n+1)^3], \forall m \in [1, (n+1)^3]$$

.

Cela revient à résoudre le système :

$$\sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^n \lambda_{ijk}^l x^i y^j z^k = \delta_{lm}, \forall l \in [1, (n+1)^3], \forall m \in [1, (n+1)^3]. \quad (\text{B.1})$$

Pour le cas  $3D$ , le logiciel *MAPLE* réussit à résoudre les systèmes de façon analytique

mais les résultats obtenus sont inutilisables du fait de leur taille trop conséquente. En effet, cela représente environ 100000 lignes de code. Cela reste donc très fastidieux à programmer et inutilisable. Il faut alors trouver une méthode qui résolve le système de manière exacte.

Seules les méthodes directes peuvent y répondre. Nous avons donc choisi la méthode d'inversion  $LU$  fourni par *Numerical Recipes*.

Cette méthode consiste à décomposer une matrice  $A$  en un produit de deux matrices  $L$  triangulaire inférieure et  $U$  triangulaire supérieure. Donc, au lieu de résoudre le système  $AX = B$  où  $X$  est l'inconnue,  $B$  le second membre, on résout dans un premier temps le système  $LY = B$  (descente) puis  $UX = Y$  (remontée).

Rappelons que cette méthode s'applique à des matrices à diagonale fortement dominante. La matrice que nous utilisons ne possède malheureusement pas cette propriété à cause de la présence de zéros sur la diagonale principale. Pour contrer cette difficulté, il faut rechercher des pivots, ce qui revient à échanger des lignes ou des colonnes afin de se débarrasser des zéros gênants. Cela revient donc à faire des permutations, ce qui se traduit sous forme matricielle par la multiplication à gauche de la matrice  $A$  par une matrice de permutation  $P$ . Il s'agit ainsi de résoudre le système  $PAX = PB$ . C'est alors la matrice  $PA$  qu'on décompose sous la forme  $LU$ .

Afin de fixer les idées, nous allons représenter les différents systèmes à résoudre sous forme matricielle. Déterminer les  $\lambda_{ijk}^l \in [0, n], j \in [0, n], k \in [0, n]$  de chaque polynôme de base  $f_l$  de la relation B.1 correspond à résoudre le système matriciel de la figure B.7.  $L(x_L, z_L, y_L)$  représente les coefficients  $x^i y^j z^k, (i, j, k) \in ([0, n])^3$  associés aux nœuds de pression de coordonnées  $a_L = (x_L, z_L, y_L)$ .

Pour trouver l'ensemble des coefficients associés aux polynômes de base, on résout le système  $LA_{0INT} = B$  de la figure B.8 :



$$\begin{bmatrix} L(x_1, y_1, z_1) \\ L(x_2, y_2, z_2) \\ \vdots \\ L(x_L, y_L, z_L) \\ \vdots \\ L(x_{(n+1)^3}, y_{(n+1)^3}, z_{(n+1)^3}) \end{bmatrix} \begin{bmatrix} \lambda_{000}^1 \\ \lambda_{100}^1 \\ \vdots \\ \lambda_{nnn}^1 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow \text{ligne L}$$

Figure B.7 : Système à résoudre pour chaque polynôme de base  $f_l$

$$\underbrace{\begin{bmatrix} L(x_1, y_1, z_1) \\ L(x_2, y_2, z_2) \\ \vdots \\ L(x_i, y_i, z_i) \\ \vdots \\ L(x_{(n+1)^3}, y_{(n+1)^3}, z_{(n+1)^3}) \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} \lambda_{000}^1 & \lambda_{000}^i & \lambda_{000}^{(n+1)^3} \\ \lambda_{100}^1 & \lambda_{100}^i & \lambda_{100}^{(n+1)^3} \\ \vdots & \vdots & \vdots \\ \lambda_{nnn}^1 & \lambda_{nnn}^i & \lambda_{nnn}^{(n+1)^3} \end{bmatrix}}_{\mathbf{A0int}} = \underbrace{\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 1 \end{bmatrix}}_{\mathbf{B}}$$

Figure B.8 : Système global à résoudre



## Annexe C

# Interpolation conservative pour les nœuds vitesses

### C.1 Vitesses horizontales

Dans cette annexe, on désire appliquer aux nœuds vitesses la méthode *FIC* détaillée dans la sous-section 2.3.2, pour les scalaires. La difficulté réside ici dans le fait que les volumes de contrôle des nœuds vitesses de niveau  $G_L$  appartenant à l'interface  $G_L - G_{L-1}$  ne sont pas toujours contigus à ceux des nœuds appartenant au niveau  $G_{L-1}$ .

Les figures C.1 et C.2 montrent une continuité des volumes de contrôle entre les deux niveaux pour les faces haut et bas de la cellule *AMR*. Ce n'est pas le cas pour les faces gauches et droites de cette cellule. Ainsi, pour les nœuds de niveau  $G_L$  appartenant à la face gauche et droite de la cellule *AMR*, on applique une interpolation conservative, comme cela est expliqué dans [TROTTEMBERG 01]. Pour les autres nœuds des faces haut et bas, on applique la méthode *FIC* détaillée dans la sous-section 2.3.2, pour les nœuds vitesses cette fois-ci. On rappelle que la méthode *FIC* permet de résoudre les nœuds discrétisés (voir figure 1.19) sans faire intervenir les nœuds de limite de cellules (traités par interpolation non conservative sur la figure 1.19). Les nœuds vitesses ne pouvant bénéficier de cette méthode, nécessitent de connaître la solution sur certains nœuds vitesses de limites de cellule comme cela est montré sur la figure 1.19. Ceux-ci sont traités maintenant avec une interpolation conservative et apparaissent sur les figures C.1 et C.2.

On considère l'équation (1.32), à laquelle on applique la méthode des volumes finis. Le

terme de pénalisation utilisé pour le traitement des limites est omis :

$$\begin{aligned}
\int_{\Omega_L^p} \rho^n \frac{\vec{V}^{n+1}}{\Delta t} d\Omega_L^p + \int_{\Omega_L^p} \vec{\nabla} \cdot (\rho^n \vec{V}^n \vec{V}^{n+1}) d\Omega_L^p &= \int_{\Omega_L^p} \rho^n \vec{g} d\Omega_L^p - \int_{\Omega_L^p} \vec{\nabla} p^{n+1} d\Omega_L^p \\
&+ \int_{\Omega_L^p} \vec{\nabla} \cdot (\mu^n (\vec{\nabla} \vec{V}^{n+1} + \vec{\nabla}^T \vec{V}^{n+1})) d\Omega_L^p \\
+ \int_{\Omega_L^p} \rho^n \left( \frac{\vec{V}^n}{\Delta t} \right) d\Omega_L^p + dr \int_{\Omega_L^p} \vec{\nabla} \cdot (\vec{\nabla} \cdot \vec{V}^{n+1}) d\Omega_L^p &
\end{aligned} \tag{C.1}$$

On s'appuie sur la figure C.1 pour exposer rapidement l'interpolation conservative (pour plus de détails, on consultera [TROTTEBERG 01]). L'idée consiste à trouver la vitesse  $u_w \in G_L$  à partir d'une approximation sur les flux. Dans un premier temps, on fait un bilan de flux sur le volume de contrôle hachuré ( $\Omega_L$ ) : soit

$$Flux(\Omega_L) = F_e - G_w + F_n - F_s$$

Tous ces flux font intervenir des quantités appartenant au niveau  $G_L$ . Ne connaissant pas  $G_w$  (car la face gauche de la cellule  $AMR$  de niveau  $G_L$  n'est pas *contigüe* à une face droite de niveau  $G_{L-1}$ ), on prolonge le volume  $\Omega_L$  vers la gauche jusqu'à ce qu'il y ait contiguïté. On note  $\widehat{\Omega}_L$  ce volume. On remarque d'ailleurs la relation  $\widehat{\Omega}_L = 3\Omega_L$ . Un deuxième bilan de flux est effectué sur le volume  $\widehat{\Omega}_L$  : soit

$$Flux(\widehat{\Omega}_L) = \widehat{F}_e - F_W + \widehat{F}_n - \widehat{F}_s$$

En supposant,

$$\begin{aligned}
- Flux(\Omega_L) &= \frac{1}{3} Flux(\widehat{\Omega}_L), \\
- \widehat{F}_e &= F_e, \\
- \widehat{F}_n &= 3F_n, \\
- \widehat{F}_s &= 3F_s,
\end{aligned}$$

on trouve  $G_w = \frac{2}{3}F_e + \frac{1}{3}F_W$ .  $G_w$  dépendant de  $u_w$ , on en déduit la vitesse  $u_w$ . On précise que le flux  $F_W$  est calculé à partir d'éléments appartenant au niveau  $G_{L-1}$  comme cela a été fait dans la méthode *FIC*.

L'avantage de cette méthode est qu'elle permet la conservation de la quantité de mouvement et que tous les nœuds résolus font intervenir les termes du lagrangien augmenté. Ainsi, la méthode d'optimisation s'applique à tous les nœuds résolus, ce qui pourrait rendre le couplage *AMR*-lagrangien augmenté plus efficace qu'elle ne l'est actuellement.

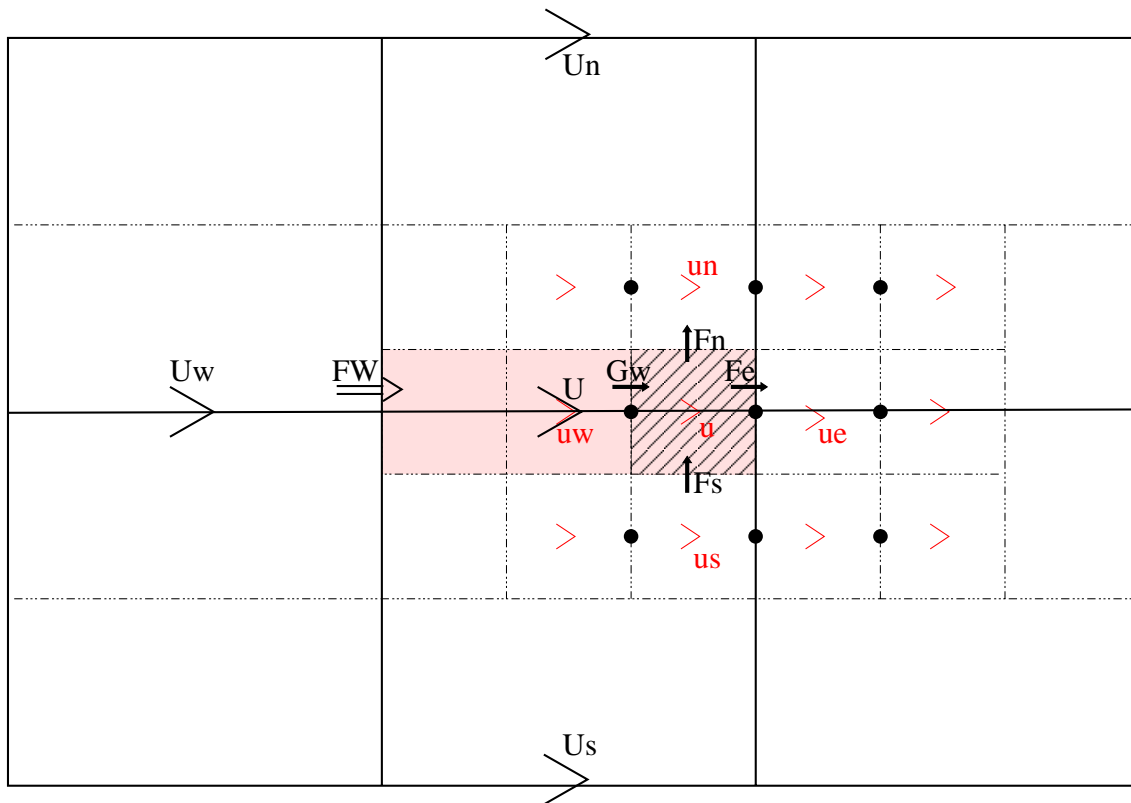
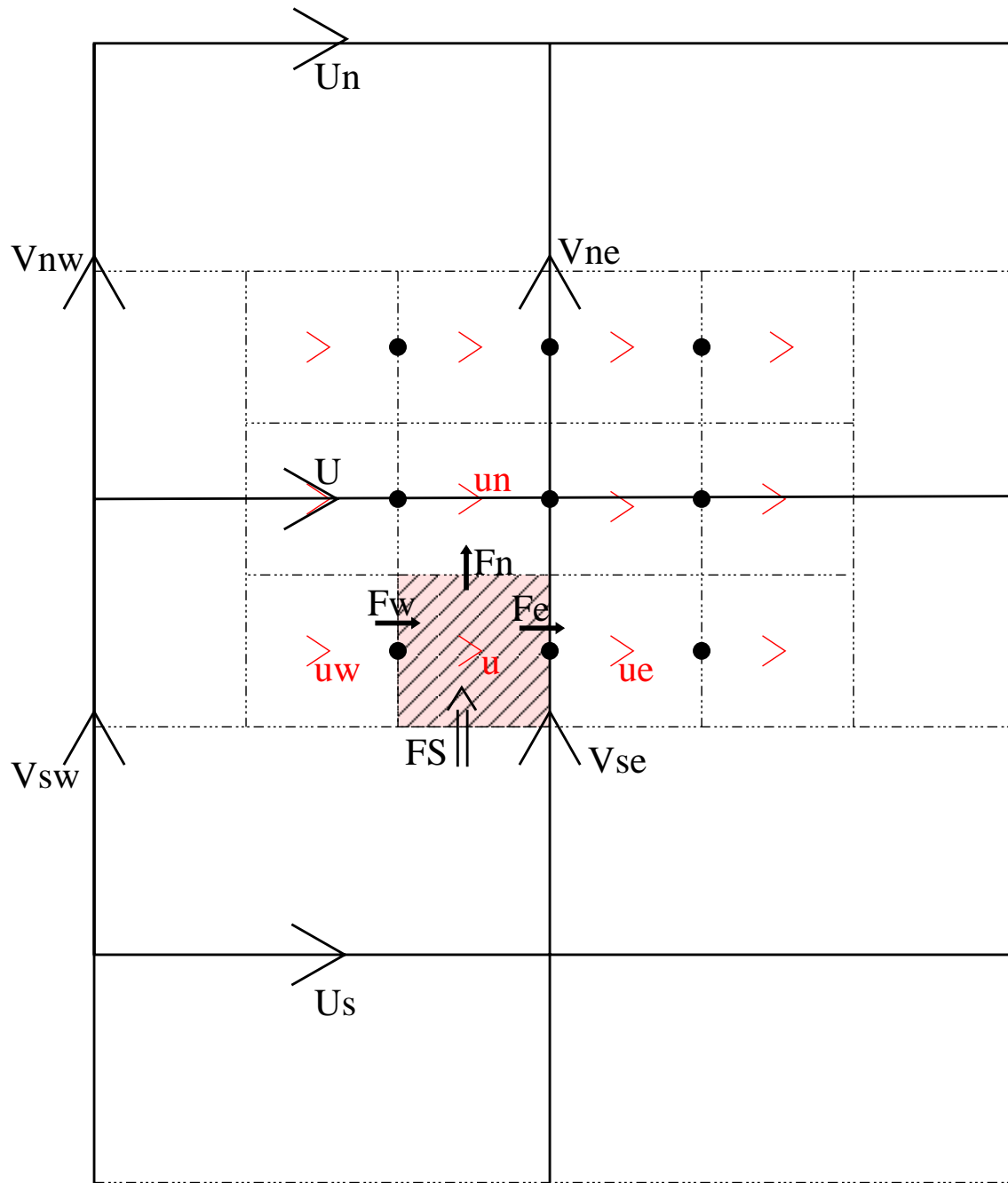


Figure C.1 : Interpolation conservative pour déterminer  $u_w$ .

Figure C.2 : Méthode FIC pour déterminer  $u$ .

# Bibliographie

- [AIT-ALI-YAHA 00] D. AIT-ALI-YAHA, W. HABASHI, A. TAM, M. VALLET & M. FORTIN. *A directionally adaptive methodology using an edge-based error estimate on quadrilateral grids*. International Journal for Numerical Methods in Fluids, vol. 32, p. 725–744, 2000.
- [ANGOT 89] P. ANGOT. *Contribution à l'étude des transferts thermiques dans des systèmes complexes; application aux composants électroniques*. Thèse de doctorat, Université BordeauxI, 1989.
- [ANGOT 92] P. ANGOT, J. P. CALTAGIRONE & K. KHADRA. *Une méthode adaptative de raffinement local: la Correction du Flux à l'Interface*. C. R. Acad. Sci. Paris, Série I, vol. 315, p. 739–745, 1992.
- [BAKER 97] T. BAKER. *Mesh adaptation strategies for problems in fluid dynamics*. Finite Elements in Analysis and Design, vol. 25, p. 243–273, 1997.
- [BASU 83] P. BASU & A. PEANO. *Adaptivity in p-version finite element analysis*. J. Struct. Eng., vol. 109, p. 2310–2324, 1983.
- [BELL 91] J. B. BELL, M. BERGER, J. S. SALTZMAN & M. WELCOME. *Three Dimensional Adaptive Mesh Refinement for Hyperbolic Conservation Laws*. LLNL report UCRL-JC-108794, december, 1991.
- [BELLENGER 05] E. BELLENGER & P. COOREVITS. *Adaptive mesh refinement for the control of cost and quality in finite element analysis*.

- Finite Elements in Analysis and Design, vol. 41, p. 1413–1440, 2005.
- [BERGER 84] M. BERGER & J. OLIGER. *Adaptive mesh refinement for hyperbolic partial differential equations*. Journal of Computational Physics, vol. 53, p. 484–512, 1984.
- [BERGER 89] M. BERGER & P. COLELLA. *Local adaptive mesh refinement for shock hydrodynamics*. Journal of Computational Physics, vol. 82, p. 64–84, 1989.
- [BRACKBILL 92] J. U. BRACKBILL, D.B. KOTHE & C. ZEMACH. *A continuum method for modeling surface tension*. J. Comput. Phys., vol. 100, p. 335–354, 1992.
- [BREIL 01] J. BREIL. *Modélisation du remplissage en propergol de moteur à propulsion solide*. Thèse de doctorat, Université BordeauxI, 2001.
- [CADAVALCH 98] J. CADAVALCH, C. D. PEREZ-SEGARRA, M. SORIA & A. OLIVA. *Fully conservative multiblock method for the resolution of turbulent incompressible flow*. In Proceedings of the Fourth ECCOMAS Computational Fluid Dynamics Conference, vol. 1.2, p. 1234–1239, 1998.
- [CADAVALCH 99] J. CADAVALCH, A. OLIVA, C. D. PEREZ-SEGARRA, M. COSTA & J. SALOM. *Comparative study of conservative and nonconservative interpolation schemes for the domain decomposition domain on laminar incompressible flow*. Numerical Heat Transfer, Part B, vol. 35, p. 65–84, 1999.
- [CALTAGIRONE 99] J. P. CALTAGIRONE & J. BREIL. *Sur une méthode de projection vectorielle pour la résolution des équations de Navier-Stokes*. C. R. Acad. Sci. Paris, Serie II b, Mécanique des fluides numérique, vol. 327, p. 1179–1184, 1999.



- [CHASSAING 00] P. CHASSAING. Turbulence en mécanique des fluides. Analyse du phénomène en vue de sa modélisation à l'usage de l'ingénieur. Collection Polytech, cépaduès-editions edition, 2000.
- [DELAGE 06] S. DELAGE, S.VINCENT, J.P. CALTAGIRONE & J.P. HELIOT. *A hybrid linking approach for solving the conservation equations with an Adaptive Mesh Refinement method*. Journal of Computational and Applied Mathematics, vol. 191, p. 280–296, 2006.
- [DELHAYE 74] J. M. DELHAYE. *Jump conditions and entropy sources in two phase systems. Local instant formulation*. Int. J. Multiphase Flow, vol. 1, p. 395–409, 1974.
- [DEMKOWICZ 89] L. DEMKOWICZ, J. ODEN, W. RACHOWICZ & O. HARDY. *Toward a universal h-p adaptive finite element strategy, Part 1: Constrained approximation and data structure*. Computer Methods in Applied Mechanics and Engineering, vol. 77, p. 79–112, 1989.
- [DONGARRA 03] J. DONGARRA, I. FOSTER, G. FOX, W. GROPP, K. KENNEDY, L. TORCZON & A. WHITE. Sourcebook of parallel computing. Morgan kaufmann publishers, an imprint of elsevier science edition, 2003.
- [DORR 02] M. R. DORR, F. X. GARAIZAR & J. A. F. HITTINGER. *Simulation of Laser Plasma filamentation using adaptive mesh refinement*. Journal of Computational Physics, vol. 177, p. 233–263, 2002.
- [FORTIN 82] C. FORTIN & R. GLOWINSKI. Méthodes de Lagrangien augmenté. Application à la résolution numérique de problèmes aux limites. Dunod, Paris, 1982.
- [GUO 86] B. GUO & I. BABUŠKA. *The h-p version of the finite element method. Part 1*. J. Comput. Mech., vol. 1, p. 21–42, 1986.

- [GUSTAFSSON 78] P. GUSTAFSSON. On first and second order symmetric factorization methods for the solution of elliptic difference equations. Chalmers University of Technology, 1978.
- [HABASHI 98] W. HABASHI, J. DOMPIERRE, Y. BOURGAULT, M. FORTIN & M. VALLET. *Certifiable computational fluid dynamics through mesh optimization*. AIAA journal, vol. 36, p. 703–711, 1998.
- [HACKBUSH 84] W. HACKBUSH. *Local defect correction method and domain decomposition techniques, in defect correction methods: Theory and applications*. K. Böhmer and H. J. Stetter, eds computation supplementation, Springer-Verlag, Wien, vol. 5, p. 89–113, 1984.
- [HARLOW 65] H. HARLOW & J.E. WELCH. *Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface*. Phys. Fluids, vol. 8, p. 2182–2189, 1965.
- [HAY 04] A. HAY. *Etude des stratégies d'estimation d'erreur numérique et d'adaptation locale de maillages non-structurés pour les équations de Navier-Stokes en moyenne de Reynolds*. Thèse de doctorat, Université de Nantes, 2004.
- [HAY 05] A. HAY & M. VISONNEAU. *Computation of free-surface flows with local mesh adaptation*. Int. J. Numer. Meth. Fluids, vol. 49, p. 785–816, 2005.
- [HETU 92] J. HETU & D.PELLETIER. *Adaptive remeshing for viscous incompressible flows*. AIAA Journal, vol. 30, p. 1986–1992, August 1992.
- [HIRSCH 90] C. HIRSCH. Numerical computation of internal and external flows, volume 2: computational methods for inviscid and viscous flows. John Wiley and Sons, Chichester, 1990.
- [HIRT 81] C. W. HIRT & B. D. NICHOLS. *Volume of fluids (VOF) method for the dynamics of free boundaries*. Journal of Computational Physics, vol. 39(1), p. 201–225, 1981.

- [ILINCA 00] C. ILINCA, X. ZHANG, J. Y. TREPANIER & R. CAMARERO. *A comparison of three error estimation techniques for Finite Volume solutions of compressible flows*. Computer methods in Applied Mechanics and Engineering, vol. 189, p. 1277–1294, 2000.
- [ISHII 06] E. ISHII, T. ISHIKAWA & Y. TANABE. *Hybrid Particle/Grid method for predicting motion of micro and macro-free surfaces*. Journal of Fluids Engineering, in press, 2006.
- [KATAOKA 86] I. KATAOKA. *Local instant formulation of two phase flow*. Int. J. Multiphase Flow, vol. 12, p. 745–758, 1986.
- [KHADRA 94] K. KHADRA. *Méthodes adaptatives de raffinement local multi-grille, applications aux équations de Navier-Stokes et de l'énergie*. Thèse de doctorat, Université BordeauxI, 1994.
- [KHADRA 95] K. KHADRA, J.P. CALTAGIRONE & P. ANGOT. *On a local multigrid mesh refinement for solving Navier-Stokes equations*. Comptes rendus de l'académie des sciences, Série II-Mécanique-Physique-Chimie-Astronomie, vol. 320(6), p. 295–302, Mars 1995.
- [KHADRA 96] K. KHADRA, P. ANGOT, J.P. CALTAGIRONE & P. MOREL. *Concept de zoom adaptatif en architecture multigrille locale; étude comparative des méthodes L.D.C., F.A.C. et F.I.C.* RAIRO, Modélisation mathématique et analyse numérique, vol. 30(1), p. 39–82, 1996.
- [KLEIN 94] R. I. KLEIN, C. F. MCKEE & P. COLELLA. *On the hydrodynamic interaction of shock waves with interstellar clouds. I. Non radiativenshocks in small clouds*. Astrophysical journal, vol. 420(1), p. 213–236, 1994.
- [LASCAUX 00] P. LASCAUX & R. THEODOR. *Analyse numérique matricielle appliquée à l'art de l'ingénieur : tome 2 Méthodes itératives*. Dunod, Paris, 2000.

- [LEVEQUE 90] R. J. LEVEQUE. *Numerical methods for conservation laws*. Lectures in Mathematics, ETH Zurich, Birkhauser, 1990.
- [LEVEQUE 92] R.J. LEVEQUE. *Numerical methods for conservation laws*. Lectures in mathematics, Landford, Birkhauser, Zurich, page 173, 1992.
- [LEVEQUE 96] R. LEVEQUE. *High-resolution conservative algorithms, for advection in incompressible flow*. SIAM J. Numer. Anal., vol. 33, p. 627–665, 1996.
- [LIOVIC 06] P. LIOVIC, M. RUDMAN, J-L. LIOW, D. LAKEHAL & D. KOTHE. *A 3D unsplit-advection volume tracking algorithm with planarity-preserving interface reconstruction*. Comput. and Fluids, 2006.
- [LIU 94] X. D. LIU, S. OSCHER & T. CHAN. *Weighted Essentially Non Oscillatory Schemes*. J. Comput. Phys., vol. 115, p. 200–212, 1994.
- [LOSASSO 06] F. LOSASSO, R. FEDKIW & S. OSHER. *Spatially adaptive techniques for level set methods and incompressible flows*. Computers and Fluids, in press, corrected proof, 2006.
- [MARTIN 52] J. C. MARTIN & W. J. MOYCE. *An experimental study of the collapse of liquids columns on a rigid horizontal plane*. Phys. Trans. Serie A, Math. Phys. Sci., vol. 244, p. 312–325, 1952.
- [MINION 97] M. L. MINION & D. L. BROWN. *Performances of under-resolved two-dimensional incompressible flow simulation, II*. Journal of Computational Physics, vol. 138(2), p. 734–765, 1997.
- [ODEN 89] J. ODEN, L. DEMKOWICZ, W. RACHOWICZ & T. WESTERMANN. *Toward a universal h-p adaptive finite element strategy, Part 2: A-posteriori error estimation*. Computer Methods in Applied Mechanics and Engineering, vol. 77, p. 113–180, 1989.

- [PALMERIO 88] B. PALMERIO. *A two dimensional FEM adaptive moving-node method for steady Euler flow simulation*. Computer methods in Applied Mechanics and Engineering, vol. 71, p. 315–340, 1988.
- [PALMERIO 94] B. PALMERIO. *An attraction-repulsion mesh adaptation model for flow solution on unstructured grids*. Computer and Fluids, vol. 23, p. 487–506, 1994.
- [PATANKAR 80] S. V. PATANKAR. Numerical heat transfer and fluid flow. Hemisphere Publishing Corporation, New York (USA), 1980.
- [PEMBER 95] R. PEMBER, J. BELL, P. COLELLA, W. CRUTCHFIELD & M. WELCOM. *An adaptive grid method for unsteady compressible flow in irregular regions*. Journal of Computational Physics, vol. 120, p. 278–304, 1995.
- [PEYRET 83] R. PEYRET & T. D. TAYLOR. Computational methods for fluid flows. Springer series in computational physics, springer, verlag, new york edition, 1983.
- [PIERCE 03] N. PIERCE & M. GILES. *Adjoint and defect error bounding and correction for functional estimates*. In the 16th AIAA Computational Fluid Dynamics Conference, p. paper 2003–3846, 2003.
- [PIRZADEH 99] S. PIRZADEH. *An adaptive unstructured grid method by grid subdivision, local remeshing and grid movement*. In the 14th AIAA Computational Fluid Dynamics Conference, p. paper 99–3255, 1999.
- [PROVATAS 99] N. PROVATAS, N. GOLDENFELD & J. DANTZIG. *Adaptive mesh refinement computation of solidification microstructures*. Journal of Computational Physics, vol. 148, p. 265–290, 1999.
- [RACHOWICZ 89] W. RACHOWICZ, J. ODEN & L. DEMKOWICZ. *Toward a universal h-p adaptive finite element strategy, Part 3: Design of h-p meshes*. Computer Methods in Applied Mechanics and Engineering, vol. 77, p. 181–212, 1989.

- [SAGAUT 06] P. SAGAUT. Large Eddy Simulation for incompressible Flows, an introduction. Springer, third edition edition, 2006.
- [SMAGORINSKY 63] J. SMAGORINSKY. *General circulation experiments with the primitive equations*. Month. Weath. Rev., vol. 93, p. 99–165, 1963.
- [STANSBY 98] P. K. STANSBY, A. CHEGINI & T. C. D. BARNES. *The initial stages of dam-break flow*. Journal of Fluid Mechanics, vol. 374, p. 407–424, 1998.
- [SUSSMAN 98] M. SUSSMAN, A. S. ALMGREN, J. B. BELL, P. COLLELLA, L. H. HOWELL & M. L. WELCOME. *A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations*. Journal of Computational Physics, vol. 142, p. 1–46, 1998.
- [SUSSMAN 99] M. SUSSMAN, A. S. ALMGREN, J. B. BELL, P. COLLELLA, L. H. HOWELL & M. L. WELCOME. *An adaptive Level Set Approach for Incompressible Two-Phase Flows*. Journal of Computational Physics, vol. 148, p. 81–124, 1999.
- [TAINE 03] J. TAINE & J-P PETIT. Transferts thermiques, Introduction aux sciences des transferts. Dunod, paris, 3ème édition edition, 2003.
- [TENCHEV 05] R. T. TENCHEV, J.A. MACHENZIE, T.J. SCANLON & M.T. STICKLAND. *Finite element moving mesh analysis of phase change problems with natural convection*. International Journal of Heat and Fluid Flow, vol. 26, p. 597–612, 2005.
- [TRANGENSTEIN 02] J. A. TRANGENSTEIN. *Multi-scale iterative techniques and adaptive mesh refinement for flow in porous media*. Advances in Water Resources, vol. 25, p. 1175–1213, 2002.
- [TROTTEBERG 01] U. TROTTEBERG, C. OOSTERLEE & A. SCHULLER. Multigrid. Academic press edition, 2001.

- [VENDITTI 02] D. VENDITTI & D. DARMOFAL. *Grid adaptation for functional outputs: Application to two-dimensional inviscid flows*. Journal of Computational Physics, vol. 176, p. 40–46, 2002.
- [VINCENT 99] S. VINCENT. *Modélisation d'écoulements incompressibles de fluides non-miscibles*. Thèse de doctorat, Université BordeauxI, 1999.
- [VINCENT 00] S. VINCENT & J.P. CALTAGIRONE. *One Cell Local Multigrid method for solving unsteady incompressible multi-phase flows*. J. Comput. Phys., vol. 163, p. 172–215, 2000.
- [VINCENT 04] S. VINCENT, J-P. CALTAGIRONE, P. LUBIN & T. N. RANDRIANARIVELO. *An adaptive augmented Lagrangien method for three dimensional multimaterial flows*. Computers and Fluids, vol. 33, p. 1273–1289, 2004.
- [VORST 92] H. A. VAN DER VORST. *A fast and smoothly converging variant of BI-CG for the solution of non-symmetric linear systems*. SIAM J. Sci. Stat. Comput., vol. 13(2), p. 631–644, 1992.
- [WASHIO 00] T. WASHIO & C. W. OOSTERLEE. *Error analysis for a potential problem on locally refined grids*. Numerische Mathematik, vol. 86, p. 539–563, 2000.
- [WHITAKER 83] S. WHITAKER. *Fundamental Principles of Heat Transfer*. Robert E. Krieger Publishing company, malabar, florida edition, 1983.
- [YOUNGS 82] D. L. YOUNGS. *Time-dependent multimaterial flow with large fluid distorsion*. K. W. Morton and M. J. Baines, New York (USA), 1982.
- [ZALESKI ] S. ZALESKI. *Computation of multiphase flow by volume of fluid and high-order front tracking methods*.