Apport des méthodes cinétiques à la simulation d'écoulements dans les milieux poreux

Léonard De Izarra

▶ **To cite this version:**

Léonard De Izarra. Apport des méthodes cinétiques à la simulation d'écoulements dans les milieux poreux. Autre [cond-mat.other]. Université d'Orléans, 2012. Français. NNT : 2012ORLE2001 . tel-00675306v2

HAL Id: tel-00675306

https://theses.hal.science/tel-00675306v2

Submitted on 30 May 2012

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

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

UNIVERSITÉ D'ORLÉANS

ÉCOLE DOCTORALE SCIENCES ET TECHNOLOGIES

LABORATOIRES : Institut des Sciences de la Terre d'Orléans / Institut de Combustion Aérothermique Réactivité et Environnement

THÈSE présentée par :

Léonard DE IZARRA

soutenue le : **13 Janvier 2012**

pour obtenir le grade de : **Docteur de l'Université d'Orléans**

Discipline : **Physique**

Apport des méthodes cinétiques à la simulation d'écoulements dans les milieux poreux

THÈSE DIRIGÉE PAR :

Jean-Louis ROUET	Professeur, Université d'Orléans
Boujema IZRAR	Professeur, Université d'Orléans

RAPPORTEURS :

Giovanni MANFREDI	Directeur de Recherches, CNRS Strasbourg
Olivier SÉRO-GUILLAUME	Directeur de Recherches, CNRS Nancy

JURY :

Ary BRUAND	Professeur, Université d'Orléans, *Président*
Boujema IZRAR	Professeur, Université d'Orléans
Giovanni MANFREDI	Directeur de Recherches, CNRS Strasbourg
Vincent PAVAN	Maître de conférence, Université Aix-Marseille 1
Jean-Louis ROUET	Professeur, Université d'Orléans
Olivier SÉRO-GUILLAUME	Directeur de Recherches, CNRS Nancy

"Heureux qui a du goût pour l'étude ! Plus heureux qui réussit à délivrer par elle son esprit de ses illusions et son cœur de sa vanité ; but désirable où vous avez été conduit dans un âge encore tendre par les mains de la sagesse ; tandis que tant de pédants, après un demi-siècle de veilles et de travaux, plus courbés sous le faix des préjugés que sous celui du temps, semblent avoir tout appris, excepté à penser."

L'Homme-machine, Julien Offray de La Mettrie

À Stéphane, mon frère.

Remerciements

Je tiens tout d'abord à remercier la Collectivité, qui par l'intermédiaire de la Région Centre a financé ce travail de recherche.

Je remercie naturellement mes directeurs Jean-Louis Rouet et Boujema Izrar pour l'aide qu'ils m'ont apporté dans mon travail. Merci pour leur disponibilité et pour l'autonomie qu'ils m'ont donné.

Mes sincères remerciements à Monsieur Giovanni Manfredi, directeur de recherches au laboratoire IPCMS de Strasbourg, ainsi qu'à Monsieur Olivier Séro-Guillaume, directeur de recherches au laboratoire LEMTA de Nancy, pour l'intérêt qu'ils ont manifesté en acceptant d'être les rapporteurs de mon travail puis membres de mon jury.

Je remercie vivement Monsieur Ary Bruand, professeur à l'université d'Orléans et directeur de l'ISTO, d'avoir accepté de présider mon jury. Merci également pour sa confiance.

J'adresse également mes remerciements à Vincent Pavan, maître de conférence à l'université d'Aix-Marseille, pour l'intérêt qu'il a porté à mon travail, pour avoir accepter de participer à mon jury ainsi que pour son enthousiasme.

Merci à Emmanuel Le Trong pour l'aide précieuse qu'il m'a apporté dans l'écriture de mon code.

Merci également aux personnes de l'ISTO avec lesquelles j'ai pu partager quelques bons moments. Un merci tout particulier à Clairette.

Finalement, un grand merci à mon entourage, famille et ami(e)s. Merci à toi Delph !

Sommaire

Table des figures **5**

Liste des Tableaux **7**

Liste des symboles et notations **9**

1 Introduction **13**

1.1 Approches en mécanique des fluides et leurs limites de validité 14

1.2 Coexistence de différents régimes au sein d'un même milieu 16

1.3 Plan de la thèse 19

2 Éléments de théorie cinétique **21**

2.1 Espace des phases - Équations de Hamilton 21

2.2 Fonction de distribution dans l'espace des phases 23

2.2.1 Fonction de distribution à N particules 23

2.2.2 Fonction de distribution réduite 23

2.2.3 Fonction de distribution et grandeurs macroscopiques 24

2.2.4 Fonction de distribution d'équilibre 25

2.3 Évolution de la fonction de distribution dans l'espace des phases . . . 27

2.3.1 Équation de Liouville 27

2.3.2 Opérateur de collisions 28

2.4 Gaz dilués – Limite de validité de l'équation de Boltzmann 30

3 Limite hydrodynamique de l'équation de Boltzmann **33**

3.1 Hiérarchie des moments 33

3.2 Invariants collisionnels 34

3.3 Équations de bilan local sur les invariants collisionnels 35

3.3.1 Bilan de masse 35

3.3.2 Bilan de quantité de mouvement 36

3.3.3 Bilan d'énergie interne 36

3.4 Équations hydrodynamiques . . . 37
3.5 Temps de relaxation . . . 40
3.5.1 Temps de relaxation local . . . 40
3.5.2 Temps de relaxation et nombre de Knudsen . . . 41

4 Écoulements glissants et transitionnels 43
4.1 Conditions aux bords . . . 44
4.1.1 Conditions aux bords cinétiques . . . 44
4.1.2 Modèle de Maxwell . . . 46
4.2 Vitesse de glissement à la paroi . . . 47
4.3 Écoulement de Poiseuille avec glissement . . . 49
4.3.1 Profil de vitesse . . . 50
4.3.2 Vitesse débitante . . . 52
4.4 Saut de température et transpiration thermique . . . 55
4.4.1 Saut de température . . . 55
4.4.2 Transpiration thermique . . . 55

5 L'équation de Boltzmann sur réseau 57
5.1 Schémas de Boltzmann sur réseau . . . 57
5.2 Schéma général de résolution de l'équation de Boltzmann-BGK . . . 60
5.3 Projection de l'équation de Boltzmann-BGK 62
5.4 Quadrature et troncature de la fonction de distribution . . . 66
5.5 Fonction d'équilibre discrète . . . 68
5.6 Opérateur de collisions . . . 69
5.6.1 Splitting des opérateurs . . . 70
5.6.2 Moyenne des collisions . . . 72
5.7 Hiérarchie des coefficients d'Hermite . . . 73
5.8 Forces extérieures . . . 74
5.9 Conditions aux limites . . . 76
5.9.1 Discrétisation des conditions aux bords cinétiques . . . 76
5.9.2 Conditions aux bords de pression . . . 79
5.10 Quadratures et schémas hybrides . . . 82
5.10.1 Quadratures pures . . . 82
5.10.2 Quadratures composites . . . 83
5.11 Schéma de transport . . . 87

6 Validation et applications de la méthode 93

6.1 Gaz de Knudsen unidimensionnel 94

6.2 Validations de la thermique 102

6.2.1 Conductivité thermique 102

6.2.2 Saut de température à la paroi 103

6.3 Poiseuille plan – Paradoxe de Knudsen 108

6.3.1 Courbes de débits 108

6.3.2 Vitesses de glissement 110

6.3.3 Profils de vitesse et de température 112

6.4 Coût de calcul des différents schémas 116

6.5 Écoulement dans un long canal rectiligne 117

7 Conclusion 129

Appendices 131

A Polynômes d'Hermites et quadrature de Gauss-Hermite 133

A.1 Polynômes d'Hermite 133

A.1.1 Définition 133

A.1.2 Propriétés 133

A.1.3 Liste des premiers polynômes d'Hermite généralisés 134

A.1.4 Racines des polynômes d'Hermite généralisés 135

A.2 Tenseurs de Grad-Hermite généralisés 135

A.2.1 Définition 135

A.2.2 Propriétés 136

A.2.3 Liste des premiers tenseurs d'Hermite généralisés 137

A.3 Quadrature de Gauss-Hermite 137

A.3.1 Définition 137

A.3.2 Poids associés à la Quadrature de Gauss-Hermite 138

A.3.3 Quadrature de Gauss-Hermite en dimension D $>$ 1 138

B Nombres sans dimension et ordre de grandeur **141**

B.1 Nombre de Knudsen . . . 141

B.1.1 Libre parcours moyen . . . 141

B.1.2 Expression temporelle du nombre de Knudsen . . . 142

B.1.3 Ordres de grandeur . . . 142

B.2 Nombre de Mach . . . 143

B.2.1 Vitesse du son . . . 143

B.3 Nombre de Reynolds . . . 144

B.4 Relation entre K_n, M_a et R_e . . . 144

B.5 Nombre de Prandtl . . . 145

C Codes **147**

C.1 Mise en œuvre de la méthode en une dimension . . . 147

C.1.1 Quadrature de Gauss-Hermite . . . 147

C.1.2 Gaz de Knudsen unidimensionnel . . . 148

C.1.3 Boltzmann-BGK unidimensionnel . . . 150

C.2 Mise en œuvre de la méthode – Code maBLB . . . 154

C.2.1 En-tête de maBLB . . . 154

C.2.2 Exemples d'écoulements de Poiseuille . . . 156

C.2.3 Poiseuille plan – Force extérieure . . . 156

C.2.4 Poiseuille plan – Pression – Gradient de densité . . . 161

C.2.5 Géométrie . . . 167

C.2.6 Le réseau – Class Lattice . . . 175

C.2.7 Nœuds du réseau – Class Node . . . 184

C.2.8 Espace des vitesses – Class Vspace . . . 195

C.2.9 Variables globales . . . 198

Bibliographie **205**

Table des figures

1.1 Régimes d'écoulements, descriptions et méthodes associées (cf. Table 1 pour la signification des acronymes). Les régimes d'intérêts sont ceux susceptibles de coexister au sien des structures poreuses que l'on considère ici. On remarque que la description cinétique recouvre bien l'ensemble de ces régimes d'intérêts. 17

1.2 La structure confinante présente deux tailles caractéristiques L_1 et L_2. Un fluide dont le libre parcours moyen des particules est λ s'écoule dans cette structure. Ce problème peut donc présenter deux régimes d'écoulements distincts. 17

1.3 Écoulement sous pression à travers un long canal. La longueur L' du canal est très grande par rapport à sa largeur L. La longueur macroscopique pertinente pour définir le nombre de Knudsen est ici la largeur L du canal. 18

2.1 Transport de la fonction de distribution dans l'espace des phases sur un laps de temps δt. 28

2.2 Effet des collisions sur la fonction de distribution. 29

2.3 Limites des différentes approximations (cf. Bird [Bir94]). Les droites $\delta/d = 7$ (limite gaz dense - gaz dilué), $L/\delta = 100$ et $Kn = 0,1$ sont données à titre indicatif. La longueur L du système est prise telle que $L = 10\, l_{VER}$ pour un niveau de fluctuations admissible de 3%. Pour de l'air à TPS (*i.e.* $T = 273\, K$ et $P = 10^5\, Pa$), les différents régimes d'écoulement (fonction du nombre de Knudsen) on été représentés. MMC représente le régime hydrodynamique, Gl le régime de glissement, Tr le régime transitionnel et ML celui des molécules libres. 31

4.1 Illustration de la diffusion par une paroi d'un faisceau de particules incidentes à la vitesse $\boldsymbol{\xi}'$. La vitesse des particules réémises est notée $\boldsymbol{\xi}$. Le vecteur $\boldsymbol{n}$ est la normale à la paroi, dirigée vers le fluide. 44

4.2 Calcul de la vitesse du fluide sur la paroi. Les particules volent librement à l'intérieur de la couche d'épaisseur λ, leur vitesse n'est donc pas affectée à l'intérieur de celle-ci. La vitesse des particules réémises est $(1 - \beta)$ fois la vitesse des particules incidentes réfléchies, dont la vitesse tangentielle n'a pas changé plus β fois la vitesse des particules diffusées, dont la vitesse tangentielle est celle de la paroi. 48

4.3 Écoulement de Poiseuille plan. La vitesse U_0 est la vitesse maximale du champ de vitesse en régime hydrodynamique ($K_n \to 0$), représenté en ligne continue. Le champ de vitesse avec glissement est représenté par des flèches. 50

4.4 Représentation du débit normalisé en fonction du nombre de Knudsen. Le fait que cette courbe de débit présente un minimum est contre-intuitif en ce sens qu'à une augmentation de $K_n = \lambda/H$ (à λ fixé) correspond une diminution de la distance H entre les plaques. On nomme ce phénomène le paradoxe de Knudsen. 53

4.5 Débits expérimentaux par Dong [Don56] (1956), recueillis dans Cercignani [Cer94]. Les abscisses sont $\delta = (K_n)^{-1}$ et les ordonnés le débit normalisé $Q/(4\,U_0\,K_n)$. On remarque que les résultats ne dépendent que très peu du gaz en écoulement. 54

5.1 Débits de l'écoulement de Poiseuille en fonction du Knudsen selon différente approches : expérimentale, analytique et numérique. Pour plus de détails sur les courbes de débits représentées, le lecteur pourra se reporter section 4.3 page 49. 59

5.2 Exemple d'une géométrie générée à partir d'une image obtenue grâce aux programmes donnés dans l'annexe C.2.5. Les conditions aux bords cinétiques nécessitent la connaissance des normales aux bords représentées ici par des vecteurs. On observe que pour les parois selon les diagonales, la distance entre les nœuds de bords est dilatée d'un facteur $\sqrt{2}$ par rapport à celle des parois dans les directions principales. 76

5.3 Géométrie utilisée pour valider le bords "en coins". Le système est périodique selon x et y, et l'écoulement est mis en mouvement par une force extérieure parallèle à la paroi. Il s'agit donc ici d'un seul écoulement enroulé à la surface d'un tore. 78

5.4 Champ de vitesse obtenu pour l'écoulement de Poiseuille en diagonal. 79

5.5 Illustration du maillage près d'un bord. Le bord se trouve en x. Il est nécessaire de définir deux nœuds à l'intérieur de la paroi, en $x + \delta x$ et en $x + 2\delta x$, pour appliquer l'extrapolation. 80

5.6 Illustration du maillage près d'un bord. Le bord se trouve en maintenant en $x + 2\delta x$. 81

5.7 Exemples de quadratures. Les vitesses discrètes, racines des polynômes d'Hermite $\mathcal{H}_n$, ont étés redimensionnées par rapport à la vitesse maximale $v_{max}(\mathcal{H}_n)$, de sorte que le critère CFL soit respecté par tous les faisceaux pour $\delta x = \delta t = 1$. La distribution de Maxwell-Boltzmann représentée sur chaque graphe a pour température redimensionnée $\hat{\theta}_0 = \theta_0 / v^2_{max}(\mathcal{H}_n)$, θ_0 étant la température de référence du système. On remarque que les quadratures d'ordre impair, contrairement aux quadratures d'ordre pair, possèdent un faisceau de vitesse nulle dont le poids est le plus fort. Cette caractéristique a une influence, comme nous le verrons, sur les résultats produits par les schémas engendrés par ces différentes quadratures. 84

5.8 Exemples de maillages de l'espace des vitesses en dimension 2. Les vitesses discrètes, racines des polynômes de Grad-Hermite $\boldsymbol{\mathcal{H}}_n$, ont été redimensionnées par rapport à la vitesse maximale $v_{max}(\mathcal{H}_n)$, de sorte que le critère CFL soit respecté par tous les faisceaux pour $\delta x = \delta t = 1$. Elles sont engendrées par les permutations des vitesses en une dimension, prises deux à deux. Comme dans le cas unidimensionnel, les quadratures d'ordre impair comportent un faisceau de vitesse nulle, dont le poids est le plus fort. On remarquera que le système de vitesses construit sur $\mathcal{H}_3$ est identique au D_2Q_9 des LBM. 85

5.9 Exemples de quadratures composites. Les vitesses discrètes, racines des polynômes d'Hermite $\mathcal{H}_n$ et $\mathcal{H}_{n+1}$, ont été redimensionnées par rapport à la vitesse maximale $v_{max}(\mathcal{H}_{n+1})$, de sorte que le critère CFL soit respecté par tous les faisceaux pour $\delta x = \delta t = 1$. On remarquera que les quadratures composites, à nombre de faisceaux équivalent, discrétisent plus finement le "gras" de la fonction de distribution de Maxwell-Boltzmann (comparer par exemple $\mathcal{H}_{3|4}$ Fig. (a), avec $\mathcal{H}_6$, Fig. 5.7(d)). 86

5.10 Illustration du schéma de transport Upwind. La vitesse discrète v_i est une des racines du polynôme d'Hermite associé à la quadrature (cf. sous-section précédente) et v_{max} la vitesse discrète d'amplitude maximale, on a donc $|v_i|/v_{max} \leq 1$. Le rapport $\delta x/\delta t$ est la vitesse du réseau. Le produit $(|v_i|/v_{max}) \times (\delta x/\delta t)$, que l'on notera C_O le nombre de *Courant*, ce nombre permet de définir un critère de stabilité pour les schémas, le critère CFL : $C_O \leq 1$. 88

5.11 Comparaison des champs de vitesses obtenus par les limiteurs de flux superbee, koren et minmod sur Poiseuille plan. À chaque ligne de graphes correspond une valeur du nombre de Knudsen et des discrétisations de 15×4 et 40×4 pour les colonnes de gauche et de droite, respectivement. (**Suite à la figure 5.12.**) 91

5.12 (**Suite de la figure 5.11**) 92

6.1 Simulation de la relaxation d'une perturbation gaussienne de la densité pour le gaz de Knudsen. Les résultats montrent que l'augmentation du nombre de vitesses discrètes améliore la solution numérique. . 98

6.2 Valeur de la densité $\rho(L/2,t)$ en fonction du temps discret en $x = L/2$, position où la densité est maximale. 99

6.3 Comparaison du schéma composite $\mathcal{H}_{4|5}$ avec le schéma basé sur $\mathcal{H}_9$. Tous les schémas comportent 9 vitesses discrètes. 100

6.4 Valeur de la densité $\rho(L/2,t)$ en fonction du temps discret en $x = L/2$, position où la densité est maximale. Comparaison de la quadrature composite $\mathcal{H}_{4|5}$ avec deux quadratures "pures" d'ordres équivalents. . 101

6.5 Champs de températures obtenus avec les quadratures pures $\mathcal{H}_n$, pour $n = 3 \ldots 8$, à l'instant $t = 24,90$. Les paramètres de la simulation sont : $\delta t = 0,1$, $\delta x = 0,5$ et $\tau_0 = 0,1\,\delta t = 0,01$. 104

6.6 Champs de températures obtenus avec les quadratures composites $\mathcal{H}_{n|n+1}$, pour $n = 3 \ldots 5$, à l'instant $t = 24,90$. Les paramètres de la simulation sont : $\delta t = 0,1$, $\delta x = 0,5$ et $\tau_0 = 0,1\,\delta t = 0,01$. 105

6.7 Champs de température entre deux parois maintenues aux températures $\theta_0 \pm 0,05$ pour différentes valeurs du nombre de Knudsen. Ces résultats ont été obtenus grâce au schéma composite $\mathcal{H}_{4|5}$. On remarque que le saut de température augmente avec le Knudsen. . . . 106

6.8 Saut de température à la paroi $\hat{\theta}_{jump}$ obtenu par la mesure de $\hat{\theta}(y_W) - \hat{\theta}_W$. 106

6.9 Saut de température à la paroi $\hat{\theta}_{jump}$ obtenu par la mesure de la pente $\left.\frac{\partial\hat{\theta}}{\partial\hat{y}}\right|_{\hat{y}_W}$. La comparaison avec les courbes obtenues par la mesure de $\hat{\theta}(y_W) - \hat{\theta}_W$ pour le schéma basé sur $\mathcal{H}_{5|6}$, permet de conclure que le relation (6.24) est bien respectée par tous les schémas. 107

6.10 Courbes de débit, fonction du Knudsen, obtenues par les schémas basés sur les quadratures pures $\mathcal{H}_n$, pour $n = 3, \ldots, 8$. 109

6.11 Courbes de débits, fonction du Knudsen, obtenues par les schémas basés sur les quadratures composites $\mathcal{H}_{n|n+1}$, pour $n = 3, \ldots, 6$. 110

6.12 Vitesses de glissement en fonction du Knudsen. Résultats obtenus grâce aux schémas basés sur les quadratures pures $\mathcal{H}_n$, pour $n = 3, \ldots, 8$. 111

6.13 Vitesses de glissement en fonction du Knudsen. Résultats obtenus grâce aux schémas basés sur les quadratures composites $\mathcal{H}_{n|n+1}$, pour $n = 3, \ldots, 6$. 112

6.14 Profils de vitesse normalisée par la vitesse maximale de l'écoulement en régime hydrodynamique ($4\,U_0$). Résultats obtenus grâce au schéma basé sur $\mathcal{H}_{4|5}$, pour différentes valeurs du nombre de Knudsen et un nombre de Mach $M_a = 0,0001$. 113

6.15 Profils de vitesse normalisée par la vitesse maximale de l'écoulement en régime hydrodynamique ($4\,U_0$). Pour comparaison, les profils semi-analytiques (Eq. (6.26)) sont représentés en pointillés. Résultats obtenus grâce au schéma basé sur $\mathcal{H}_{4|5}$, pour différentes valeurs du nombres de Knudsenet un nombre de Mach $M_a = 0,0001$. 114

6.16 Champs de température pour différentes nombres de Knudsen et un nombre de Mach $M_a = 0,0001$. Résultats obtenus grâce au schéma basé sur $\mathcal{H}_{4|5}$. Sur la figure (d) les abscisses sont en échelle logarithme. 115

6.17 Temps de calcul en fonction du nombre de faisceaux (échelles Log-Log) pour les schémas basés sur les quadratures pures $\mathcal{H}_n$ et composites $\mathcal{H}_{n|n+1}$. Ces temps de calculs ont été normalisés par le temps de calcul du schéma basé sur $\mathcal{H}_3$. 116

6.18 Illustration du problème. 117

6.19 Champ de densité ρ/ρ_0 normalisé par la densité de référence ρ_0. . . . 120

6.20 Nombre de Knudsen local. 121

6.21 Champ de vitesse selon y (u_y/u_{max}) normalisé par la vitesse maximale selon y, u_{max}. 122

6.22 Déviation au gradient de densité linéaire. 123

6.23 Champ de vitesse selon x (u_x/u_{max}) normalisé par la vitesse maximale selon y, u_{max}. Il s'agit d'un tronçon du canal pris en son centre. . . . 124

6.24 Champ de vitesse selon x (u_x/u_{max}) normalisé par la vitesse maximale selon y, u_{max}. Il s'agit ici de la sortie du canal. 125

6.25 Champ d'impulsion selon y, normalisée ($\rho u_y/(\rho_0\, u_{max})$). 126

6.26 Champ de température θ/θ_0 normalisé par la température de référence θ_0. 127

C.1 Graphe d'inclusion des fichiers d'en-têtes (.hpp). 155

C.2 Graphe des appels aux fonctions pour l'exemple du Poiseuille plan soumis à une force extérieure (fichier poiseuille_fext.cpp). 157

C.3 Graphe des appels aux fonctions pour l'exemple du Poiseuille plan soumis à un gradient de pression (fichier poiseuille_pr.cpp). 162

C.4 Graph d'héritage des nœuds. 184

Liste des tableaux

1 Acronymes et abréviations 9
2 Nombres sans dimension (cf. annexe B) 9
3 Lettres grecques 10
4 Lettres latines 12
5 Objets et opérateurs 12

4.1 Débits expérimentaux $Q/(4\,U_0\,K_n)$. Valeurs prises dans le nuage de points de la figure 4.5, issue de la thèse de Dong [Don56]. 53

A.1 Racines des 5 premiers polynômes $\mathcal{H}^{[\sigma^2]}$. 135
A.2 Poids des quadratures de Gauss-Hermite de degrés $N = 0 \ldots 5$. Les poids sont donnés dans le même ordre que les racines du tableau A.1. 138

Liste des symboles et notations

Acronymes et abréviations

Acronymes/abréviations	Significations
BC	Boundary Conditions – Conditions aux bords
BGK	Bhatnagar-Gross-Krook (Opérateur de collision)
Boltzmann-BGK	Équation de Boltzmann munie de l'opérateur BGK
CFD	Computational Fluid Dynamics – Mécanique des fluides sur ordinateur
DSMC	Direct Simulation Monte Carlo (Méthode de résolution de l'équation de Boltzmann basée sur le tirage de nombres aléatoires)
LBM	Lattice Boltzmann Method – Méthode de Boltzmann sur réseau
MMC	Mécanique des Milieux Continus

Table 1: Acronymes et abréviations

Nombres sans dimension

Symboles	Significations
E_c	Nombre de Eckert
K_n	Nombre de Knudsen
M_a	Nombre de Mach
P_r	Nombre de Prandt
R_e	Nombre de Reynolds

Table 2: Nombres sans dimension (cf. annexe B)

Lettres grecques

Symboles	Significations
α	Diffusivité thermique
β	Coefficient d'accommodation en impulsion
β_T	Coefficient d'accommodation en énergie
γ	Indice adiabatique
δt	Pas de temps
δx	Pas d'espace
ϵ	Paramètre du développement de Chapman-Enskog
κ	Conductivité thermique
λ	Libre parcours moyen
$\mathbf{\Lambda}$	Tenseur des contraintes visqueuses
μ	Viscosité dynamique
ν	Viscosité cinématique
ω_i	Poids de la quadrature de Gauss-Hermite
$\psi(r)$	Limiteur de flux
ρ	Densité massique
$\rho\varepsilon$	Densité d'énergie interne
τ	Temps de relaxation vers l'équilibre
τ_M	Temps caractéristique d'évolution à l'échelle macroscopique
τ_m	Temps caractéristique d'évolution à l'échelle microscopique
θ	$k_B T/m$
θ_0	$k_B T_0/m$
$\boldsymbol{\xi}$	Vitesse particulaire exprimée dans le référentiel du fluide
$\boldsymbol{\xi}_i$	Vitesse particulaire discrète, dans le référentiel du fluide

Table 3: Lettres grecques

Lettres latines

Symboles	Significations
$\boldsymbol{a}_n, a_{n\alpha_1\ldots\alpha_n}$	Coefficient d'ordre n du développement d'Hermite de f
$\boldsymbol{a}_n^{eq}, a_{n\alpha_1\ldots\alpha_n}^{eq}$	Coefficient d'ordre n du développement d'Hermite de f^{eq}
c_0	Vitesse quadratique moyenne à la température de référence T_0
c_{max}	Vitesse discrète de plus grande valeur
c_s	Vitesse du son
C_p	Capacité calorifique à pression constante
C_v	Capacité calorifique à volume constant
f	Fonction de distribution dans l'espace des vitesses
f_i	Fonction de distribution discrétisée
D	Dimension de l'espace physique
f^{eq}	Fonction de distribution d'équilibre
f_i^{eq}	Fonction de distribution d'équilibre discrétisée
f^{neq}	Partie hors équilibre de la fonction de distribution
$\boldsymbol{F}$	Force extérieure
$\boldsymbol{g}$	$\boldsymbol{F}/m$
H	Fonction de Hamilton ou entropie de Boltzmann
$\mathcal{H}_n^{[\theta_0]}(v)$	Polynôme d'Hermite de degré n généralisé à la température θ_0
$\mathcal{H}_n(v)$	Polynôme d'Hermite de degré n
$\boldsymbol{\mathcal{H}}_n^{[\theta_0]}(\boldsymbol{v})$, $\mathcal{H}_{n\alpha_1\ldots\alpha_n}^{[\theta_0]}(\boldsymbol{v})$	Polynôme tensoriel d'Hermite de degré n généralisé à la température θ_0
$\boldsymbol{\mathcal{H}}_n(\boldsymbol{v})$, $\mathcal{H}_{n\alpha_1\ldots\alpha_n}(\boldsymbol{v})$	Polynôme tensoriel d'Hermite de degré n
k_B	Constante de Boltzmann
L	Longueur macroscopique caractéristique
$\mathcal{L}$	Opérateur de Liouville
m	Masse d'une particule
$\boldsymbol{\mathcal{M}}_k$	Moment d'ordre k de f
n	Densité numérique de particules ρ/m
$\boldsymbol{n}$	Vecteur normale à une paroi
$\mathcal{N}_A$	Nombre d'Avogadro
p	Pression cinétique
$\boldsymbol{P}$	Tenseur des pressions
$\boldsymbol{q}$	Flux d'énergie interne
$\boldsymbol{Q}$	Tenseur des flux d'énergie interne
$\boldsymbol{t}$	Vecteur tangente à une paroi
T	Température
T_0	Température de référence
T_W	Température de la paroi
t	Temps
$\boldsymbol{u}$	Vitesse macroscopique (ou moyenne)
u_{slip}	vitesse de glissement

Suite page suivante

Suite de la page précédente

Symboles	Significations
$\boldsymbol{U}_W$	Vitesse de la paroi
$\boldsymbol{v}_i$	Vitesse particulaire discrète, exprimée dans le référentiel du laboratoire
$\boldsymbol{v}$	Vitesse particulaire exprimée dans le référentiel du laboratoire
$w^{[\theta]}(\boldsymbol{\xi})$	Fonction de pondération pour les polynômes d'Hermite généralisés à la température θ
$w(\boldsymbol{\xi})$	Fonction de pondération pour les polynômes d'Hermite
$\boldsymbol{x}$	Position
$\boldsymbol{x}_W$	Position de la paroi

TABLE 4: Lettres latines

Objets et opérateurs

Notations	Significations
s	Scalaire
$\boldsymbol{v}$, v_α	Vecteur
$\boldsymbol{T}$, $T_{\alpha\beta\gamma\ldots}$	Tenseur
$\boldsymbol{u}\cdot\boldsymbol{v}$, $u_\alpha v_\alpha$	Produit scalaire
$\boldsymbol{S}:\boldsymbol{T}$, $S_{\alpha\beta}T_{\alpha\beta}$	Produit tensoriel totalement contracté
$\boldsymbol{uv}$,$\boldsymbol{u}\otimes\boldsymbol{v}$,$u_\alpha v_\beta$	Produit tensoriel
∂_t	Dérivé par rapport au temps
$\boldsymbol{\nabla}_{\boldsymbol{x}}$, $\partial_{x\alpha}, \partial_\alpha$	Nabla - Gradient ou divergence dans l'espace des positions
$\boldsymbol{\nabla}_{\boldsymbol{v}}$, $\partial_{v\alpha}$	Gradient ou divergence dans l'espace des vitesses
D_t	Dérivé particulaire
$\mathbb{I}$	Matrice identité

TABLE 5: Objets et opérateurs

1 Introduction

La présente thèse de doctorat porte sur l'apport de la théorie cinétique à la simulation d'écoulements de fluides, en particulier des gaz, dans des structures confinantes de petites tailles. Par petites tailles, il faut entendre, dans le cas des gaz et dans des conditions normales, des tailles comprises entre le millimètre et le nanomètre. Nous parlerons donc ici de "micro-écoulements" dont l'étude fait l'objet d'une discipline récente que l'on nomme la microfluidique [CB04, eH05] [1]. Bien que la microfluidique porte principalement sur l'étude de micro-systèmes manufacturés [2] (micro-turbines, micro-pompes, etc.), de tels écoulements abondent dans la nature, notamment dans les géomatériaux poreux, mais aussi dans le cadre industriel, où des matériaux poreux, naturels ou synthétiques, sont utilisés, par exemple, comme filtres, vannes, tamis moléculaires ou encore pompes sans pièce mobile [GG08] (motionless pumps). Les matériaux poreux ont la particularité de posséder des géométries relativement complexes, tortueuses et désordonnées, et aussi de pouvoir présenter plusieurs tailles caractéristiques. En effet, un matériau poreux peut posséder une distribution en tailles de pores très large, ou simplement comporter des fissures dont la taille est grande par rapport à la taille moyenne de ses pores. Ces deux caractéristiques constituent des difficultés du point de vue de la simulation numérique de ces écoulements. D'une part, les géométries complexes donnent naissance à des difficultés de conditions aux bords, d'autre part, comme nous le verrons dans la section suivante, la diversité des échelles de tailles des structures confinantes nécessite l'emploi d'une description du fluide plus fine que celle des milieux continus, qui cesse d'être valide aux petites échelles, pour lesquelles la structure granulaire/atomique de la matière commence à se manifester.

La description proposée ici sera celle de la physique statistique hors d'équilibre, grâce à laquelle le fluide sera décrit par une fonction de distribution de probabilité, dont l'évolution est régie par l'équation de Boltzmann. Plus précisément, il sera proposé d'étendre les méthodes de Boltzmann sur réseau (LBM), que l'on sait valide dans la limite hydrodynamique, aux écoulements non hydrodynamiques, en suivant la démarche systématique de construction des LBM proposée par Shan et He [SH98].

Commençons par définir de manière plus précise les différents régimes d'écoulement ainsi que les différentes descriptions et méthodes numériques qui leur sont associées.

1. La microfluidique est une discipline jeune. Cela vient du fait que les moyens techniques de la mise en œuvre de micro-systèmes, principalement issus de la mico-électronique, le sont aussi.
2. Ces micro-systèmes d'appels des MEMS, pour Micro-Electro-Mechanical Systems.

1.1 Les différentes approches en mécanique des fluides et leurs limites de validité

En mécanique des fluides, la description la plus communément utilisée est la mécanique des milieux continus (MMC). Celle-ci consiste à décrire l'évolution spatio-temporelle des champs macroscopiques de densité, d'impulsion et d'énergie par les équations de Navier-Stokes [Lan90, GHP01]. Ces champs sont, du point de vue de la physique statistique, à l'équilibre. Or, par définition, un système à l'équilibre ne peut évoluer dans le temps. L'évolution temporelle de ces champs macroscopiques nécessite donc l'introduction d'équations représentatives de l'éloignement du système à l'équilibre, source de l'évolution du système. Ces équations s'appellent *équations constitutives*[3] et nécessitent l'introduction de quantités empiriques appelées *coefficients de transport*[4]. Cependant, ces équations d'évolutions, dans le cadre de la MMC, ne représentent que de très faibles écarts à l'équilibre. Or, il est possible que dans certaines circonstances, un fluide puisse s'éloigner fortement de l'équilibre, et que, dans ce cas, les équations de la mécanique des milieux continus ne suffisent plus. Il est alors nécessaire d'utiliser une description plus fine de notre fluide. Nous pourrions par exemple considérer l'ensemble de toutes les molécules, ou atomes, qui le constitue. La résolution des équations du mouvement de cet ensemble de molécules, appelé problème à N corps, permet de retrouver le comportement des fluides, y compris lorsqu'ils s'éloignent fortement de l'équilibre. Cependant, une telle description est lourde et ne permettra la simulation de systèmes constitués que d'un petit nombre de molécules.

Entre ces deux descriptions (en terme de complexité), on trouve la théorie cinétique, qui consiste essentiellement en une description statistique du problème à N corps. La statistique permet d'une part de réduire considérablement le nombre de degrés de libertés du système, et permet d'autre part de passer d'un problème discontinu (les N corps), à un problème continu. Les N corps sont remplacés par une fonction de distribution statistique, qui bien que continue, conserve les caractéristiques discrètes sous-jacentes, à l'exception des fluctuations[5]. L'évolution de cette fonction de distribution est gouvernée par une équation aux dérivées partielles, l'équation de Boltzmann, qui est similaire aux équations rencontrées dans le cadre de la MMC et pour lesquelles les techniques numériques sont très développées.

Pour résoudre un problème de mécanique des fluides, il faudra donc choisir parmi l'une de ces descriptions. Ce choix dépendra évidement du problème. Mais comment

3. Ces équations sont par exemple la loi de Fick, pour la diffusion de masse, ou la loi de Fourier pour la diffusion de la chaleur.

4. Le coefficient associé à la loi de Fick sera le coefficient de diffusion, pour la loi de Fourier ce sera la conductivité thermique.

5. La réintroduction des fluctuations thermiques dans les LBM est un sujet de recherche d'actualité [ASCW05, DSL09, DSL07, Sch08]. Ces fluctuations sont importantes pour modéliser, notamment, le mouvement Brownien qui disperse des colloïdes en suspension dans des liquides ou des gaz.

déterminer rigoureusement quelle description est la plus adaptée ? La réponse sera donnée par la valeur prise par le nombre de Knudsen, qui caractérise la faculté qu'a le fluide à s'éloigner de l'équilibre (cf. annexe B.1 pour plus de détails). Le nombre de Knudsen, noté K_n, est défini par :

$$K_n = \frac{\text{Libre parcours moyen des particules du fluide}}{\text{Dimension caractéristique de l'écoulement}} = \frac{\lambda}{L}. \tag{1.1}$$

Selon la valeur prise par ce nombre, on qualifie alors le régime de l'écoulement comme suit :

- Si $K_n \ll 1$, *i.e.* si $\lambda \ll L$, alors les collisions dominent le fluide qui reste proche de l'équilibre. L'écoulement est qualifié d'hydrodynamique.
- Si $K_n \gg 1$, *i.e.* si $\lambda \gg L$, alors les collisions sont négligeables et le fluide peut s'éloigner énormément de l'équilibre. L'écoulement est qualifié d'écoulement de particules libres.
- Si $K_n \sim 1$, *i.e.* si $\lambda \sim L$, on parle de régime intermédiaire ou transitionnel. Le fluide peut s'éloigner de l'équilibre, mais des collisions tendent à l'en empêcher.

Un découpage plus fin en différents régimes peut être fait, comme illustré par la figure 1.1, sur laquelle on trouve les descriptions ainsi que quelques méthodes qui leur sont associées. Ce découpage place le régime hydrodynamique pour des Knudsen inférieurs à 0,01. Sur cet intervalle, les équations de la MMC, ainsi que ses hypothèses, peuvent être appliquées sans crainte. C'est le cadre actuel de la dynamique des fluides numérique (CFD), pour laquelle d'innombrables outils logiciels ont été développés, dont certains reposent sur la résolution des équations de Navier-Stokes, mais aussi sur des schémas particuliers de résolution de l'équation de Boltzmann-BGK, appelés méthodes de Boltzmann sur réseau (LBM), limitées à ce régime d'écoulement. Les LBM font l'objet de recherches intensives actuellement, car elles sont particulièrement bien adaptées à la structure parallèle des super-ordinateurs modernes [6] (les clusters ou grappes de calcul). Pour des nombres de Knudsen supérieurs à 10, se trouve le régime de particules libres. Il correspond typiquement aux conditions rencontrées dans l'aérospatiale, comme par exemple à la situation d'un satellite artificiel en mouvement dans le gaz ténu de la très haute atmosphère de la Terre. En effet, la taille L des satellites, y est plusieurs fois plus petite que le libre parcours moyen λ des molécules de l'atmosphère. Ce régime a lui aussi été l'objet d'intenses recherches et développements, bien que les outils logiciels soient moins abondants et plus confidentiels que ceux dédiés au régime hydrodynamique. Ces outils sont, en général, soit des codes déterministes de dynamique moléculaire, (on résout les équations du mouvement de toutes les particules du fluide, dont le nombre reste limité) soit des codes de résolution de l'équation de Boltzmann basés sur des techniques de tirages aléatoires (on parle alors de simulation directe par méthode de

6. Les LBM sont dites hautement parallélisables, du fait de la localité des opérations effectuées. En effet, Les collisions sont locales et le transport ne nécessite la connaissance que du voisinage proche du nœuds courant.

Monte Carlo[7] (DSMC) [ZGA02]). Entre ces deux régimes extrêmes, on distingue encore deux sous-régimes. Pour $0{,}01 \lesssim K_n \lesssim 0{,}1$ le régime est qualifié de *régime de glissement*. Il peut être vu comme un prolongement du régime hydrodynamique, en ce sens que la description du fluide peut encore y être faite par la MMC ; cependant, certaines hypothèses de celle-ci n'y sont plus valides. En particulier, il n'est plus possible de considérer que le fluide "adhère" aux parois solides qui le délimitent, le fluide peut y avoir une vitesse différente de celle de la paroi et semble glisser sur celle-ci, d'où l'appellation du "*régime de glissement*". Les techniques utilisées pour ce régime peuvent donc dériver de celles utilisées en MMC, en leur ajoutant des conditions aux bords (BC) spéciales de glissement (Slip BC). Ces conditions de glissement seront cependant obtenues grâce à la description cinétique (ou statistique) du fluide. Finalement, pour $0{,}1 \lesssim K_n \lesssim 3$, le régime sera qualifié de *régime transitionnel*, car il marque la transition entre les aspects continu et granulaire de la matière, où elle peut sembler présenter une forme de dualité. Les méthodes utilisables ici ne peuvent être dérivées que de la description statistique ; en effet, la MMC est insuffisante, car non particulaire, et si nous choisissions la description particulaire, alors nous devrions faire des statistiques *a posteriori* sur les solutions obtenues, pour retrouver le continu. Il est donc judicieux, comme cela est décrit dans le premier chapitre de cette thèse, de faire des statistiques *a priori* sur le système de particules constituant le fluide, et de déterminer les équations qui régissent l'évolution de ces propriétés statistiques. D'autant que cette démarche a l'avantage de diminuer considérablement le nombre de degrés de libertés nécessaires à la description du système. Partant de cette description statistique, deux grandes voies sont alors possibles. La première consiste à déterminer, par l'équation de Boltzmann, l'évolution des propriétés statistiques du système, puis de déduire des solutions obtenues, l'évolution des grandeurs macroscopiques. C'est la voie que nous choisirons. L'autre voie consiste à construire les équations d'évolution pour les champs macroscopiques à partir de l'équation de Boltzmann, on parle alors de méthode des moments [Gra49b, ST03].

Signalons enfin que pour $K_n = 0$, le fluide est dit parfait et il a la particularité d'avoir une viscosité et une conductivité thermique nulles. Les équations qui régissent son évolution sont les équations d'Euler. À l'opposé, lorsque $K_n \to \infty$, les molécules du gaz ne subissent aucune collision, on parle alors de gaz de Knudsen.

1.2 Coexistence de différents régimes au sein d'un même milieu

Illustrons maintenant la possibilité que plusieurs régimes d'écoulement puissent coexister. Une première circonstance envisageable est représentée figure 1.2, où l'on considère une structure confinante formée de deux canaux connectés. L'un est de

7. En référence aux jeux de dés, roulette, et autre jeux aléatoires, pratiqués des les casinos de ce quartier de Monaco.

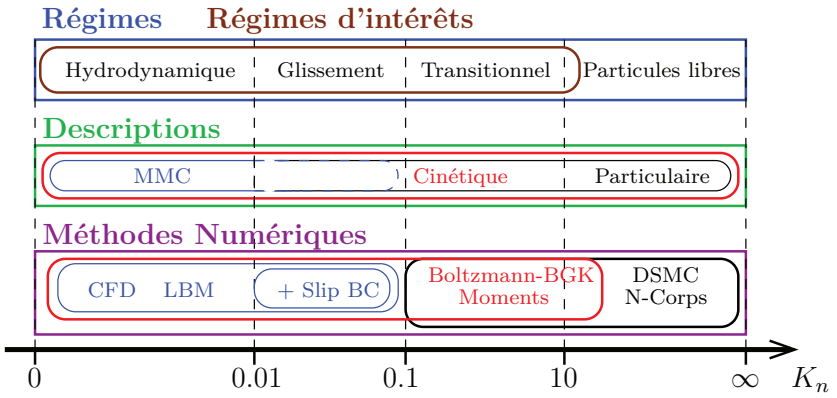


FIGURE 1.1 – Régimes d'écoulements, descriptions et méthodes associées (cf. Table 1 pour la signification des acronymes). Les régimes d'intérêts sont ceux susceptibles de coexister au sien des structures poreuses que l'on considère ici. On remarque que la description cinétique recouvre bien l'ensemble de ces régimes d'intérêts.

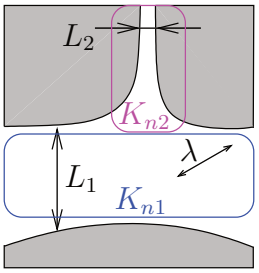


FIGURE 1.2 – La structure confinante présente deux tailles caractéristiques L_1 et L_2. Un fluide dont le libre parcours moyen des particules est λ s'écoule dans cette structure. Ce problème peut donc présenter deux régimes d'écoulements distincts.

taille L_1 et l'autre de taille $L_2 < L_1$. Supposons alors qu'un gaz, dans des conditions normales, s'y écoule. Le libre parcours moyen d'un gaz dans ces conditions est toujours de l'ordre de la centaine de nanomètres, soit $\lambda \sim 10^{-7}\,m$. Si nous considérons que l'écoulement dans le canal de taille L_2 est en régime de glissement, avec $K_{n\,2} \sim 0{,}1$, alors cela implique que $L_2 \sim 10\,\lambda \sim 1\,\mu m$. Pour que deux régimes coexistent, il suffit que l'écoulement dans le canal de taille L_1 soit en régime hydrodynamique, soit que $K_{n\,1} \gtrsim 0{,}01$, d'où $L_1 \gtrsim 10\,L_2 \sim 10\,\mu m$. Une telle situation n'est pas exceptionnelle dans nombre de matériaux poreux. Comme exemple, donnons le cas du charbon, dont les pores les plus petits ont une taille de l'ordre du nanomètre [RdS02] et pour lequel les effets de confinement modifient la perméabilité aux gaz [8] (voir par exemple [ZLSE07]).

8. Dans les liquides, le libre parcours moyen des molécules est de l'ordre de la taille des molécules, soit quelques angströms. Les écoulements de liquides sont donc toujours en régime hydrodynamique ; ils ne glissent donc pas, ou peu. Au contraire, un gaz est susceptible de glisser si la géométrie du milieu et son libre parcours moyen le permettent. Le glissement du gaz aux parois et le non-glissement du liquide, induit une différence de perméabilité au gaz et au liquide. Ce phénomène est connu sous le nom d'effet Klinkenberg.

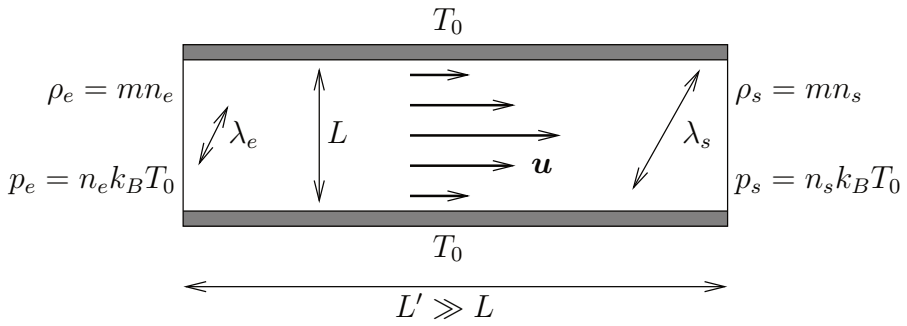


FIGURE 1.3 – Écoulement sous pression à travers un long canal. La longueur L' du canal est très grande par rapport à sa largeur L. La longueur macroscopique pertinente pour définir le nombre de Knudsen est ici la largeur L du canal.

La situation précédente montre que différents régimes d'écoulements peuvent coexister au sein d'un même domaine, pour des raisons d'ordre géométrique. Cependant, l'existence de plusieurs tailles caractéristiques dans une structure n'est pas l'unique cause possible à la coexistence de plusieurs régimes. En effet, une autre circonstance pouvant présenter le même type de propriété, est celle de l'écoulement d'un gaz à travers un long canal de largeur L constante, sous un gradient de pression important. Pour illustrer cela, considérons la situation représentée sur la figure 1.3. Pour simplifier le raisonnement, le problème est considéré isotherme à la température T_0. Le gaz est mis en mouvement par un gradient de pression $-\nabla p = (p_e - p_s)/L'$, imposée entre l'entrée et la sortie du canal. La température étant considérée uniforme et égale à T_0, la pression ne dépend que de la densité locale n dans le canal. Les pressions en entrée (p_e) et en sortie ($p_s < p_e$) ne dépendent donc que des valeurs des densités n_e et n_s, auxquelles correspondent des libres parcours moyens λ_e et λ_s différents et tels que $\lambda_s > \lambda_e$, car $\lambda \propto n^{-1}$. Il s'ensuit que la valeur du nombre de Knudsen varie le long du canal, avec une valeur K_{ne} à l'entrée et $K_{ns} > K_{ne}$ à la sortie. Ainsi, si la pression le long du canal varie dans de grandes proportions, il est encore possible que différents régimes d'écoulement coexistent, bien que la structure ne possède qu'une seule longueur caractéristique.

Dans le cas général, les deux effets énoncés ci-avant peuvent bien sûr se cumuler et le régime de l'écoulement dépendra à la fois de la géométrie et du libre parcours moyen local.

Ce qu'il faut retenir, c'est que dans certaines circonstances, plusieurs régimes d'écoulement peuvent coexister au sein du milieu dans lequel on cherche à simuler l'écoulement d'un fluide. Notamment, dans les milieux micro-poreux, où les écoulements de gaz sont susceptibles d'être dans des régimes allant de l'hydrodynamique jusqu'au régime transitionnel. Or, il n'existe pas encore à ce jour de méthode capable d'aborder de tels problèmes de manière efficace, et la recherche sur ce problème n'est qu'à ses débuts, bien que, comme il l'a été dit plus haut, des méthodes existent pour traiter ces différents régimes d'écoulement, mais dans des domaines indépendamment les uns des autres. Au vue du succès des méthodes de Boltzmann sur

réseaux (LBM) pour résoudre des écoulements en régime hydrodynamique et étant donné leur fondement cinétique, elles semblent être de bonnes candidates pour être étendues jusqu'au régime transitionnel. Tel est l'objet de cette thèse.

1.3 Plan de la thèse

Le plan de cette thèse est le suivant. Dans le chapitre 2, intitulé "Éléments de théorie cinétique", serons introduits les concepts et notions fondamentales de la physique statistique hors d'équilibre. Cette introduction à la théorie cinétique permettra de replacer les LBM dans ce cadre théorique [9]. Dans le chapitre 3, intitulé "Limite hydrodynamique de l'équation de Boltzmann", la relation entre la description cinétique et l'hydrodynamique sera présentée. Nous montrerons que dans la limite des petits nombres de Knudsen, la résolution de l'équation de Boltzmann est équivalente à la résolution des équations de Navier-Stokes. Dans le chapitre 4, intitulé "Écoulements glissants et transitionnels", nous introduirons les effets produits par l'éloignement à équilibre. Nous commencerons par présenter les conditions aux bords cinétiques de Maxwell, puis nous verrons que pour des nombres de Knudsen finis, un glissement du fluide sur les parois apparaît. L'effet de ce glissement sur un écoulement de Poiseuille sera alors discuté en détail. Ce chapitre se terminera par une description rapide d'effets thermiques induits par le confinement : le saut de température à la paroi et le phénomène de transpiration thermique. Dans le chapitre 5, intitulé "L'équation de Boltzmann sur réseau", sera proposée une nouvelle méthode de résolution de l'équation de Boltzmann-BGK, construite sur la base des LBM. Dans le chapitre 6, intitulé "Validation et applications de la méthode", la méthode proposée au chapitre 5 sera validée sur quelques cas tests, dont les mises en œuvre sont données et commentées en annexe C. Ces tests pourront constituer, moyennant des modification mineures, une base pour l'étude de cas concrets. Enfin, une conclusion à cette dissertation, ainsi que quelques perspectives, seront données au chapitre 7.

9. Celles-ci sont en effet souvent présentées comme des méthodes particulières de résolution des équations de Navier-Stokes, construites sur la base d'automates cellulaires (les "Lattice-gas cellular automata" introduit par Frisch et *al.* [FHP86] à la fin des années 80).

2 Éléments de théorie cinétique

L'objet de ce chapitre est de présenter le formalisme et les concepts fondamentaux utiles à la compréhension de ce mémoire. Il s'agit donc d'énoncer les bases de la *physique statistique*, qui est le formalisme utilisé pour décrire les systèmes mécaniques à grand nombre de degrés de liberté, aussi qualifiés de thermodynamiques. La physique statistique est une branche de la physique moderne, initiée par *Ludwig Boltzmann* à la fin du dix-huitième siècle, dans le but de comprendre comment les propriétés des corps à l'échelle que nous percevons, qualifiée de macroscopique, peuvent s'expliquer à partir de celles de leurs constituants microscopiques, que sont les atomes, les molécules, les électrons, etc.

Ce problème de passage de l'échelle microscopique à l'échelle macroscopique ne peut être résolu que par des méthodes statistiques. En effet, bien que la mécanique nous permette de décrire l'évolution de tout système, dès lors que nous connaissons ses équations du mouvement et son état à un instant donné, il est techniquement impossible de mesurer avec exactitude un tel état dans le cas d'un système thermodynamique [1]. Les méthodes statistiques présentées ici ont donc pour but de "lisser" les détails de la réalité et d'en extraire l'essentiel, suffisant à sa description microscopique.

La présentation de ce formalisme commencera par la définition de l'espace des phases, qui est l'espace le plus adapté à l'étude des systèmes thermodynamiques. Puis l'introduction de l'ensemble de Gibbs permettra de montrer que l'état macroscopique d'un système thermodynamique est déterminé par la simple connaissance de la densité d'états microscopiques dans l'espace des phases, celle-ci pouvant être reliée sans ambiguïté à l'état macroscopique du système, qui lui est mesurable. Enfin, l'équation de Boltzmann, régissant l'évolution temporelle de la densité d'états dans l'espace des phases, sera introduite.

2.1 Espace des phases - Équations de Hamilton

L'*espace des phases* est l'espace le plus adapté à l'étude des systèmes thermodynamiques. Il s'agit d'un espace abstrait qui, pour un gaz de N particules (indicées par k), est engendré par l'ensemble des coordonnées $\boldsymbol{x}_k$ et des impulsions $m_k\boldsymbol{v}_k$ des particules du gaz ; chaque point de cet espace représentant un état du système considéré. On appellera le sous-espace des positions de l'espace des phases, l'*espace des*

1. Même s'il était possible de réaliser une telle mesure, le nombre d'équations à résoudre simultanément (typiquement 10^{23}) serait alors hors de portée de n'importe quel ordinateur, aussi puissant soit-il.

configurations et le sous-espace des impulsions, l'*espace des impulsions*. Dans cet espace, l'évolution du système est alors régie par les *équations de Hamilton* :

$$m_k \frac{\partial \boldsymbol{x}_k}{\partial t} = \frac{\partial H}{\partial \boldsymbol{v}_k}, \tag{2.1}$$

$$m_k \frac{\partial \boldsymbol{v}_k}{\partial t} = -\frac{\partial H}{\partial \boldsymbol{x}_k}, \tag{2.2}$$

$$\{\boldsymbol{x}_{0k},\boldsymbol{v}_{0k}\} : \text{ État initial}, \tag{2.3}$$

où H est la *fonction de Hamilton* (ou *Hamiltonien*) du système. Il s'agit d'un système d'équations différentielles du premier ordre. Le nombre d'équations est égal à la dimension de l'espace des phases, soit $6N$ en 3 dimensions.

Pour illustrer cela, considérons le cas d'une particule de masse m soumise à un champ de force $\boldsymbol{F}(\boldsymbol{x},t)$ dérivant d'un potentiel $\Phi(\boldsymbol{x},t)$ (tq. $\boldsymbol{F}(\boldsymbol{x},t) = -\nabla\Phi(\boldsymbol{x},t)$). Ce système possède 3 degrés de liberté et son espace des phases est de dimension 6. Dans ce cas le Hamiltonien [2] du système est le suivant :

$$H(\boldsymbol{x},\boldsymbol{v},t) = \frac{1}{2} m\, v^2 + \Phi(\boldsymbol{x},t). \tag{2.4}$$

L'évolution du système est alors donnée par la résolution des équations de Hamilton :

$$m \frac{\partial \boldsymbol{x}}{\partial t} = \frac{\partial H}{\partial \boldsymbol{v}} = m\boldsymbol{v}, \tag{2.5}$$

$$m \frac{\partial \boldsymbol{v}}{\partial t} = -\frac{\partial H}{\partial \boldsymbol{x}} = -\boldsymbol{\nabla}_{\boldsymbol{x}}\Phi = \boldsymbol{F}, \tag{2.6}$$

$$(\boldsymbol{x}_0,\boldsymbol{v}_0) : \text{ État initial}. \tag{2.7}$$

On remarquera que dans le cas présent, la seconde équation de Hamilton correspond à la loi fondamentale de la *dynamique* : $\boldsymbol{F} = m\partial_t \boldsymbol{v}$. Quant à la première, elle compte pour définition de la *cinématique* : $\partial_t \boldsymbol{x} = \boldsymbol{v}$. Il s'agit donc d'un simple problème de dynamique, dont la généralisation à un système de N particules en interaction de potentiel de paires ne pose aucune difficulté. Cependant, pour les systèmes de taille macroscopique, N est de l'ordre du nombre d'Avogadro $\mathcal{N}_A \approx 6{,}02\,.\,10^{23}$ et il est impossible de construire un ordinateur capable de résoudre un système d'équations aussi grand. C'est pourquoi nous allons adopter le point de vue de la physique statistique et diminuer considérablement le nombre de degrés de liberté.

2. Pour les systèmes conservatifs, le Hamiltonien est confondu avec l'énergie totale (cinétique + potentielle). L'énergie potentielle est soit d'origine externe, soit due à l'interaction des particules.

2.2 Fonction de distribution dans l'espace des phases

2.2.1 Fonction de distribution à N particules

À chaque état macroscopique $(\rho, \boldsymbol{u}, T)$ d'un système thermodynamique correspond toute une collection d'états microscopiques $\{(\boldsymbol{x}_k,\boldsymbol{v}_k)_{k=1...N}\}$ différents [3]. On appellera *ensemble de Gibbs* une telle collection infinie de systèmes microscopiques différents, mais identiques du point de vue macroscopique.

Le nombre de systèmes de l'ensemble de Gibbs étant extrêmement grand, les points représentatifs correspondants sont denses dans l'espace des phases. Il est donc possible de définir une *densité d'états dans l'espace des phases* représentée par une *fonction de distribution* que l'on notera f^N. Cette fonction est dépendante des coordonnées $\Omega = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_N)$ et du temps t. Alors par définition, la quantité :

$$f^N(\Omega,t)\, d\Omega, \tag{2.8}$$

représente la probabilité de trouver, à l'instant t, le point représentatif de l'état microscopique du système, dans le volume élémentaire $d\Omega = d\boldsymbol{x}_1 \ldots d\boldsymbol{x}_N d\boldsymbol{v}_1 \ldots d\boldsymbol{v}_N$ de l'espace des phases centré sur Ω.

Cette fonction de distribution doit être normalisée :

$$\int_\Gamma f(\Omega,t)\, d\Omega = 1, \, \forall t, \tag{2.9}$$

où l'intégrale est faite sur tout l'espace des phases Γ.

2.2.2 Fonction de distribution réduite

La fonction de distribution définie dans la section précédente est qualifiée de *fonction de distribution à N particules* (notée ici f^N). Elle peut, dans le cas particulier où les particules du système ne sont pas corrélées [4], être réduite par intégration sur l'espace des phases de toutes les particules, sauf une, soit :

$$f^1(\boldsymbol{x}, \boldsymbol{v},t) = \int f^N(\Omega,t)\, d\Omega_1 \ldots d\Omega_{N-1}, \tag{2.10}$$

où $d\Omega_i = d\boldsymbol{x}_i d\boldsymbol{v}_i$. La fonction de distribution qui résulte de cette intégration est qualifiée de *fonction de distribution réduite* (ou à une particule). Cette fonction ne dépend plus que de 6 coordonnées (3 en configurations et 3 en vitesses). Dans ce

3. On parle d'états microscopiques compatibles.

4. Dans ce cas, la probabilité de trouver une des particules du système dans un état particulier $(\boldsymbol{x},\boldsymbol{v})$ est indépendant de l'état de toutes les autres particules. C'est notamment le cas des gaz parfaits, pour lesquels le temps d'interaction particule-particule est très court devant le temps de vol balistique des particules.

qui suit nous n'utiliserons plus que la fonction de distribution réduite, en gardant à l'esprit que nous nous restreignons aux gaz de particules non corrélées.

2.2.3 Fonction de distribution et grandeurs macroscopiques

Considérons une grandeur microscopique $\mathcal{A}(\boldsymbol{x},\boldsymbol{v},t)$, la grandeur macroscopique locale associée sera donnée par l'intégrale sur l'espace des vitesses de $\mathcal{A}$ pondérée par f :

$$\langle\mathcal{A}\rangle(\boldsymbol{x},t) = \int \mathcal{A}\, f\, d\boldsymbol{v}. \tag{2.11}$$

En particulier, si $\mathcal{A}(\boldsymbol{x},\boldsymbol{v},t) \equiv \Psi_k(\boldsymbol{v}) = \bigotimes^k \boldsymbol{v}$, la variable macroscopique associée $\mathcal{M}_k \equiv \langle\mathcal{A}\rangle$ est appelée *moment*[5] d'ordre k de f. Ces moments sont très importants, car il sont *essentiellement égaux* aux grandeurs macroscopiques que nous mesurons, comme la densité, la vitesse, la température, etc. Ils sont donnés par :

$$\mathcal{M}_k(\boldsymbol{x},t) = \int \Psi_k(\boldsymbol{v})\, f(\boldsymbol{x},\boldsymbol{v},t)\, d\boldsymbol{v}. \tag{2.12}$$

En particulier, pour $k = 0{,}1$ nous avons respectivement la densité et l'impulsion :

$$\mathcal{M}_0 = \rho(\boldsymbol{x},t) = \int f(\boldsymbol{x},\boldsymbol{v},t)\, d\boldsymbol{v}, \tag{2.13}$$

$$\mathcal{M}_1 = \rho(\boldsymbol{x},t)\, \boldsymbol{u}(\boldsymbol{x},t) = \int \boldsymbol{v}\, f(\boldsymbol{x},\boldsymbol{v},t)\, d\boldsymbol{v}. \tag{2.14}$$

Pour les moments d'ordres supérieurs, il est préférable de se placer dans le référentiel en mouvement avec le fluide. On prendra $\Psi_k(\boldsymbol{\xi}) = \bigotimes^k \boldsymbol{\xi}$, avec $\boldsymbol{\xi} = (\boldsymbol{v}-\boldsymbol{u}(\boldsymbol{x},t))$. Alors, pour $k = 2$ on obtient :

$$\mathcal{M}_2 = \boldsymbol{P}(\boldsymbol{x},t) = \int \boldsymbol{\xi}\boldsymbol{\xi}\, f(\boldsymbol{x},\boldsymbol{v},t)\, d\boldsymbol{\xi}, \tag{2.15}$$

où $\boldsymbol{P}(\boldsymbol{x},t)$ est le tenseur des pressions, que l'on peut décomposer en $\boldsymbol{P} = \boldsymbol{\Pi} + \mathbb{I}p$, avec p, la pression cinétique et $\boldsymbol{\Pi}$ le tenseur des contraintes visqueuses, qui est de trace nulle.

En l'absence de contraintes visqueuses, $\boldsymbol{P}$ se réduit à la pression cinétique qui est donnée par la trace de (2.15), soit en dimension D :

$$D\, p(\boldsymbol{x},t) = \int \boldsymbol{\xi}^2\, f(\boldsymbol{x},\boldsymbol{v},t)\, d\boldsymbol{\xi} = 2\,\rho\,\varepsilon(\boldsymbol{x},t), \tag{2.16}$$

où ε est l'énergie interne.

Or, le théorème d'équipartition de l'énergie nous donne $\varepsilon = \dfrac{D}{2}\dfrac{k_B T}{m}$. Dont on déduit

5. Tout comme la fonction Ψ_k, le moment $\mathcal{M}_k$ est un tenseur d'ordre k.

que la loi d'état du gaz sera celle d'un gaz parfait :

$$p = \rho \frac{k_B T}{m} = \rho\theta. \tag{2.17}$$

Finalement, pour $k = 3$, nous obtenons le tenseur des flux d'énergie interne par conduction :

$$\mathcal{M}_3 = 2\,\boldsymbol{Q}(\boldsymbol{x},t) = \int \boldsymbol{\xi}\boldsymbol{\xi}\boldsymbol{\xi}\, f(\boldsymbol{x},\boldsymbol{v},t)\, d\boldsymbol{\xi}. \tag{2.18}$$

Le flux de chaleur par conduction est alors donné par $\boldsymbol{q}_\alpha = \boldsymbol{Q}_{\alpha\beta\beta}$.

2.2.4 Fonction de distribution d'équilibre

Un état macroscopique particulier des systèmes thermodynamiques est l'*état d'équilibre*. Cet état d'équilibre peut être atteint *localement* ou *globalement*, on parlera alors respectivement de systèmes à l'équilibre thermodynamique local (ETL) ou global (ETG). L'équilibre thermodynamique global est caractérisé par l'homogénéité et la stationnarité du système dans son ensemble, la fonction d'équilibre est alors indépendante du temps et de l'espace. Signalons qu'à l'ETG, bien que la fonction de distribution soit stationnaire, le point représentatif de l'état du système évolue sous les équations de Hamilton dans l'espace des phases. En particulier, les particules du système collisionnent et la fonction de distribution d'équilibre devra être compatible avec les propriétés de ces collisions.

L'obtention de l'expression de la distribution d'équilibre, notée f^{eq}, nécessite l'introduction d'un théorème fondamental en physique statistique, dit *théorème H de Boltzmann*[6]. Ce théorème stipule qu'à l'équilibre, un système thermodynamique maximise une fonctionnelle de f, notée H, sous la *contrainte des trois premiers moments* de f. Ces trois moments sont les grandeurs invariantes (conservées) du système sous l'effet des collisions particule-particule ; il s'agit de la densité (2.13), de la quantité de mouvement (2.14) et de l'énergie (2.16). Cette fonctionnelle se nomme *fonctionnelle d'entropie* et son expression est la suivante :

$$H(f) = \int f \ln f d\boldsymbol{v}. \tag{2.19}$$

L'expression de la fonction d'équilibre f^{eq} s'obtient alors par la méthode d'optimisation sous contraintes des multiplicateurs de Lagrange. Cette méthode consiste à définir une nouvelle fonctionnelle :

$$\hat{H} := H + \lambda_\rho \rho + \boldsymbol{\lambda}_{\rho\boldsymbol{u}} \cdot \rho\boldsymbol{u} + \lambda_{\rho\varepsilon}\rho\varepsilon, \tag{2.20}$$

6. Le théorème H est l'équivalent cinétique du *second principe de la thermodynamique*.

augmentée de ses contraintes. Soit, en utilisant l'expression des moments :

$$\hat{H} := \int \left(f \ln f + \lambda_\rho f + \boldsymbol{\lambda}_{\rho \boldsymbol{u}} \cdot \boldsymbol{v} f + \lambda_{\rho\varepsilon} \frac{(\boldsymbol{v}-\boldsymbol{u})^2}{2} f \right) d\boldsymbol{v}, \tag{2.21}$$

où les $\lambda_{\rho,\rho\boldsymbol{u},\rho\varepsilon}$ sont les multiplicateurs de Lagrange associés aux contraintes des trois premiers moments. La fonction d'équilibre est alors solution de l'équation :

$$\left[\frac{\partial \hat{H}}{\partial f} \right]_{f=f^{eq}} = 0. \tag{2.22}$$

Soit :

$$\int_{-\infty}^{+\infty} \frac{\partial}{\partial f} \left(f \ln f + \lambda_\rho f + \boldsymbol{\lambda}_{\rho \boldsymbol{u}} \cdot \boldsymbol{v} f + \lambda_{\rho\varepsilon} \frac{(\boldsymbol{v}-\boldsymbol{u})^2}{2} f \right) d\boldsymbol{v} = 0, \ \forall v. \tag{2.23}$$

Et comme cette expression est vraie pour toute perturbation de f autour de l'équilibre, nous avons :

$$\ln f^{eq} + 1 + \lambda_\rho + \boldsymbol{\lambda}_{\rho \boldsymbol{u}} \cdot \boldsymbol{v} + \lambda_{\rho\varepsilon} \frac{(\boldsymbol{v}-\boldsymbol{u})^2}{2} = 0, \tag{2.24}$$

ou encore :

$$f^{eq} = A \, \exp \left(-\boldsymbol{\lambda}_{\rho \boldsymbol{u}} \cdot \boldsymbol{v} - \lambda_{\rho\varepsilon} \frac{(\boldsymbol{v}-\boldsymbol{u})^2}{2} \right), \tag{2.25}$$

avec $A = \exp(-1-\lambda_\rho)$.

Les inconnues $\{A,\boldsymbol{\lambda}_{\rho\boldsymbol{u}},\lambda_{\rho\varepsilon}\}$, s'obtiennent alors grâce aux contraintes d'invariance collisionnelle :

$$\rho = \int f d\boldsymbol{v} = \int f^{eq} d\boldsymbol{v}, \tag{2.26}$$

$$\rho \boldsymbol{u} = \int \boldsymbol{v} f d\boldsymbol{v} = \int \boldsymbol{v} f^{eq} d\boldsymbol{v}, \tag{2.27}$$

$$D\rho \frac{k_B T}{m} = \int |\boldsymbol{v}-\boldsymbol{u}|^2 f d\boldsymbol{v} = \int |\boldsymbol{v}-\boldsymbol{u}|^2 f^{eq} d\boldsymbol{v}. \tag{2.28}$$

Soit :

$$A = \rho \left(\frac{m}{2\pi k_B T} \right)^{D/2}, \tag{2.29}$$

$$\boldsymbol{\lambda}_{\rho\boldsymbol{u}} = \boldsymbol{0}, \tag{2.30}$$

$$\lambda_{\rho\varepsilon} = \frac{m}{k_B T}. \tag{2.31}$$

La fonction d'équilibre obtenue est finalement la *distribution de Maxwell-Boltzmann* :

$$f^{eq} = \rho \left(\frac{m}{2\pi k_B T} \right)^{D/2} \exp \left(-\frac{m(\boldsymbol{v}-\boldsymbol{u})^2}{2k_B T} \right). \tag{2.32}$$

Cette fonction est une Gaussienne dans l'espace des vitesses microscopiques $\boldsymbol{v}$, dont la surface sous la courbe est la densité ρ, le barycentre est la vitesse macroscopique $\boldsymbol{u}$ et dont la largeur à mi-hauteur dépend de la température T. On remarquera qu'elle ne dépend explicitement que des trois premiers moments de f.

2.3 Évolution de la fonction de distribution dans l'espace des phases

Nous sommes maintenant en possession d'une fonction de distribution f dans l'espace des phases, nous connaissons son expression à l'équilibre thermodynamique et nous savons que les grandeurs hydrodynamiques du système sont données par ses trois premiers moments dans l'espace des vitesses. Cependant, nous ne savons pas encore comment cette fonction de distribution doit évoluer. C'est ce que nous allons rechercher maintenant, dans le cas particulier où le système est conservatif. Cette classe de systèmes englobe tous les gaz de particules en interaction de potentiel de paire.

Nous allons d'abord rechercher l'équation d'évolution pour la fonction de distribution réduite (donc pour une particule seule – sans collision), dite *équation de Liouville*. Puis, par le biais d'un opérateur, nous introduirons les collisions entre particules, pour finalement obtenir l'*équation de Boltzmann*.

2.3.1 Équation de Liouville

En considérant l'évolution sous les équations de Hamilton d'un ensemble de points représentatifs, contenus dans un volume de l'espace des phases, on montre que pour les systèmes conservatifs, ce volume est conservé au cours du temps [7]. Une manière de déterminer l'équation d'évolution de la fonction f dans l'*espace des phases*, consiste donc à assimiler le contenu de cet espace à un *fluide incompressible*, de densité locale $f(\boldsymbol{x},\boldsymbol{v},t)$. L'équation d'évolution de ce fluide peut ainsi s'écrire en représentation lagrangienne sous la forme :

$$D_t f = 0, \tag{2.33}$$

où D_t est l'opérateur de dérivée particulaire. Dans l'espace des phases, la représentation Eulerienne de cet opérateur est la suivante :

$$D_t \equiv \partial_t + \partial_t \boldsymbol{x} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} + \partial_t \boldsymbol{v} \cdot \boldsymbol{\nabla}_{\boldsymbol{v}}. \tag{2.34}$$

En utilisant les équations de Hamilton (2.1)-(2.2), cet opérateur peut être réécrit sous la forme :

$$D_t \to \mathcal{L} = \partial_t + \frac{1}{m}\frac{\partial H}{\partial \boldsymbol{v}} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} - \frac{1}{m}\frac{\partial H}{\partial \boldsymbol{x}} \cdot \boldsymbol{\nabla}_{\boldsymbol{v}}, \tag{2.35}$$

7. Ce volume peut se déformer, s'effilocher, etc. Mais son volume restera toujours invariant.

que l'on appelle opérateur de Liouville. On obtient finalement l'équation d'évolution suivante pour f, appelée *équation de Liouville à une particule* :

$$\mathcal{L}f \equiv \partial_t f + \boldsymbol{v} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} f + \boldsymbol{g} \cdot \boldsymbol{\nabla}_{\boldsymbol{v}} f = 0, \tag{2.36}$$

où l'on a posé $\boldsymbol{g} = \boldsymbol{F}/m$.

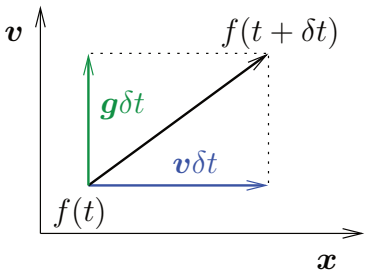


FIGURE 2.1 – Transport de la fonction de distribution dans l'espace des phases sur un laps de temps δt.

Il s'agit d'un *transport pur* de f dans l'espace des phases, à la vitesse $\boldsymbol{v}$ dans l'espace des configurations et à la "vitesse" $\boldsymbol{g}$ dans l'espace des vitesses. La figure 2.1 illustre l'action de l'opérateur $\mathcal{L}$ sur f. Signalons enfin que dans l'équation de Liouville à une particule, la force $\boldsymbol{F}$ est le *champ moyen* créé par l'ensemble des particules ou/et créé à l'extérieur du système.

2.3.2 Opérateur de collisions

L'équation de Liouville ne donnant l'évolution de la fonction de distribution que d'une particule seule, elle ne permet pas de rendre compte des collisions inter-particules au sein du fluide. Nous allons donc réintroduire ces collisions par le biais d'un opérateur agissant sur f dans l'équation de Liouville :

$$\mathcal{L}f = \left(\frac{\partial f}{\partial t}\right)_{coll}. \tag{2.37}$$

Cet opérateur traduira essentiellement la modification de la fonction de distribution due aux collisions binaires au sein du gaz [8]. Plusieurs opérateurs ont été proposés, le premier par Boltzmann lui-même, mais nous ne parlerons ici que du plus simple et le plus couramment utilisé : l'opérateur BGK [BGK54] (pour **B**hatnagar, **G**ross et **K**rook, ses auteurs).

Cet opérateur est construit à partir d'un raisonnement assez simple : l'effet des collisions est de faire relaxer la fonction de distribution f, vers la fonction d'équilibre

8. Les collisions binaires sont les plus fréquentes, car plus probables que les collisions à plus de deux particules.

locale f^{eq} (voir figure 2.2), en un temps caractéristique τ, inversement proportionnelle à la fréquence des collisions particule-particule.

Plus précisément, on supposera que f relaxe vers f^{eq} selon une loi exponentielle de temps τ [9] :

$$f(t) = f(t=0)\exp\left(-\frac{t}{\tau}\right) + \left[1 - \exp\left(-\frac{t}{\tau}\right)\right] f^{eq}. \tag{2.38}$$

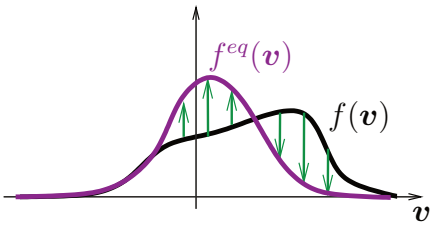


FIGURE 2.2 – Effet des collisions sur la fonction de distribution.

Or nous n'utiliserons cette expression que sur des temps courts par rapport au temps de relaxation, nous pouvons donc la linéariser. Le développement limité au premier ordre en $t = 0$ de $\exp(-t/\tau)$ étant $1 - t/\tau$, pour un petit intervalle de temps δt, nous avons :

$$f(\boldsymbol{x},\boldsymbol{v},t+\delta t) = f(\boldsymbol{x},\boldsymbol{v},t) - \frac{\delta t}{\tau}(f(\boldsymbol{x},\boldsymbol{v},t) - f^{eq}(\boldsymbol{x},\boldsymbol{v})), \tag{2.39}$$

d'où

$$\left(\frac{\partial f}{\partial t}\right)_{coll} = -\frac{1}{\tau}(f - f^{eq}). \tag{2.40}$$

Munie de cet opérateur de collisions, l'équation de Liouville devient :

$$\mathcal{L}f = -\frac{1}{\tau}(f - f^{eq}), \tag{2.41}$$

ou encore :

$$\partial_t f + \boldsymbol{v}\cdot\boldsymbol{\nabla}_{\boldsymbol{x}} f + \boldsymbol{g}\cdot\boldsymbol{\nabla}_{\boldsymbol{v}} f = -\frac{1}{\tau}(f - f^{eq}). \tag{2.42}$$

C'est l'*équation de Boltzmann-BGK*.

On remarquera que le terme $f^{neq} = f - f^{eq}$, correspond à la partie hors équilibre de f. On remarquera aussi que l'effet des collisions disparaît lorsque le système est à l'équilibre ($f^{neq} = f - f^{eq} = 0$).

9. Les collisions constituent un processus aléatoire, dont on cherche le nombre d'événements sur un laps de temps donné. Leur répartition suit donc une loi de Poisson, d'où la décroissance exponentielle.

2.4 Gaz dilués – Limite de validité de l'équation de Boltzmann

Les approximations faites lors de la dérivation de l'équation de Boltzmann, notamment l'absence de corrélations entre les particules et la définition des grandeurs macroscopiques comme moments de la fonction de distribution, nous amènent à définir les limites de validité de ces approximations.

Commençons par définir le critère de non-corrélations entre les particules du gaz. L'équation de Boltzmann dérivée dans ce chapitre représente l'évolution d'un *gaz dilué* de N molécules identiques de masse m confinées à l'intérieur d'une boîte de volume V. Le gaz est supposé parfait, *i.e.* que l'énergie d'interaction entre les molécules est négligeable devant leur énergie cinétique, ou de façon similaire, que le temps d'interaction entre les particules est très petit devant leur temps de vol libre. L'approximation "gaz parfait" sera d'autant meilleure que le gaz sera dilué. Pour définir la dilution, appelons d le diamètre effectif[10] des molécules et δ la distance intermoléculaire moyenne (voir Fig. 2.3), alors le gaz sera qualifié de *dilué* si :

$$\frac{\delta}{d} \gg 1, \tag{2.43}$$

dans le cas contraire, le gaz sera qualifié de *dense*.

La distance intermoléculaire s'exprime d'autre part en fonction de la densité locale $n = N/V$, comme $\delta \sim n^{-1/3}$ en dimension 3.

D'autre part, le nombre n de molécules contenues dans un volume élémentaire représentatif (VER) doit être suffisamment grand pour que les fluctuations statistiques y soient négligeables, sans quoi les définitions des grandeurs macroscopiques perdent leur sens. L'amplitude des fluctuations statistiques des grandeurs macroscopiques étant, pour des particules non corrélées, de l'ordre de $n/\sqrt{n}$ (cf. [BCDB89]), on a pour $n = 10^3$, des fluctuations de l'ordre de 3% et elles seront de 1% pour $n = 10^4$. Nous supposerons qu'un niveau de fluctuations de 3% est acceptable, d'où un nombre minimum de $n_{min} = 10^3$ molécules contenues dans le VER. Ainsi, la longueur minimale représentative l_{VER} devra répondre au critère $l_{VER}/\delta \geq 10$. La longueur macroscopique L caractéristique du système étudié devra donc être supérieure à la longueur élémentaire représentative l_{VER}, pour que les fluctuations soient négligeables.

Comme cela est fait en annexe (cf. annexe B), regardons le cas particulier de l'air dans des conditions normales de température et de pression (*i.e.* $T = 273\,K$ et $P = 10^5\,Pa$). Le diamètre classique de l'air, vu comme une molécule moyenne, est de l'ordre de $d \approx 0{,}35\,nm$, d'autre part, dans ces conditions standard, l'air est caractérisé par $\delta = 3{,}3\,nm$, un libre parcours moyen $\lambda = 65\,nm$ et une densité

10. d est le diamètre associé à la section efficace totale de la molécule, *i.e.* le diamètre de la molécule vue comme une sphère dure.

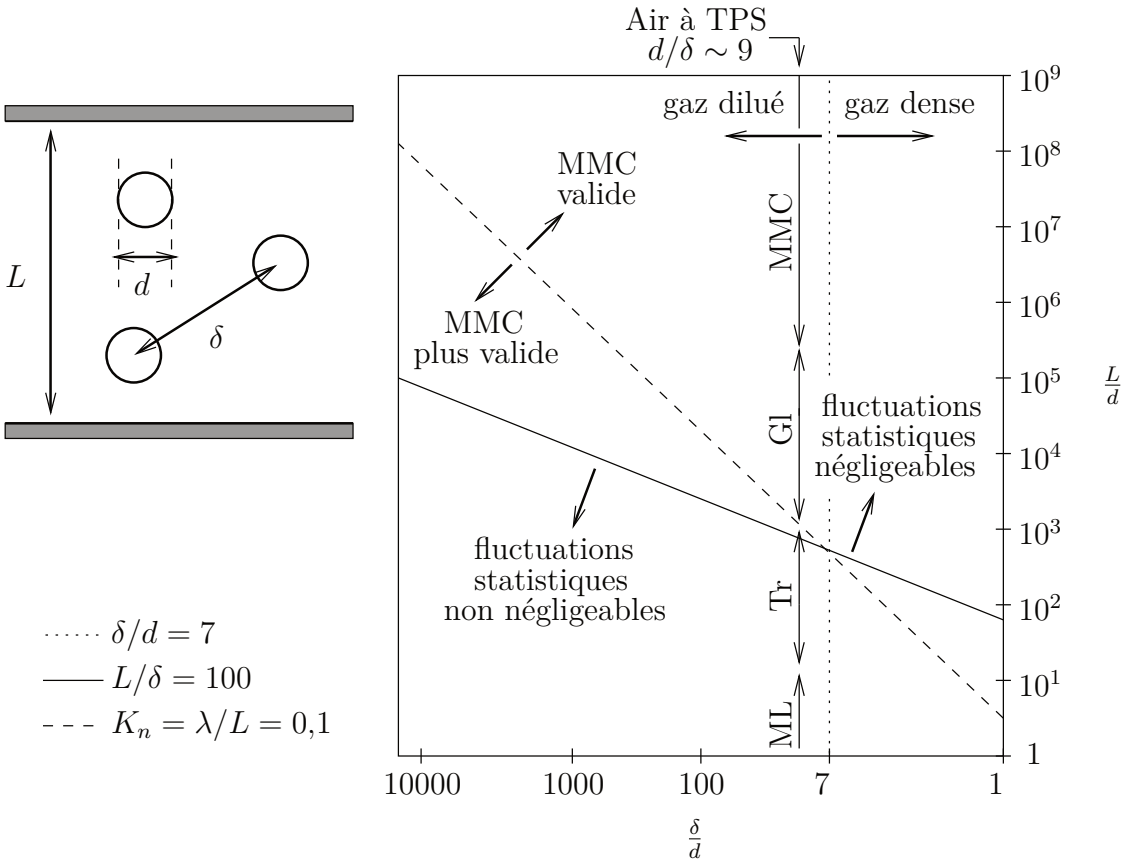


FIGURE 2.3 – Limites des différentes approximations (cf. Bird [Bir94]). Les droites $\delta/d = 7$ (limite gaz dense - gaz dilué), $L/\delta = 100$ et $Kn = 0{,}1$ sont données à titre indicatif. La longueur L du système est prise telle que $L = 10\, l_{VER}$ pour un niveau de fluctuations admissible de 3%. Pour de l'air à TPS (*i.e.* $T = 273\, K$ et $P = 10^5\, Pa$), les différents régimes d'écoulement (fonction du nombre de Knudsen) on été représentés. MMC représente le régime hydrodynamique, Gl le régime de glissement, Tr le régime transitionnel et ML celui des molécules libres.

numérique $n = 2{,}7 \cdot 10^{25}\, m^{-3}$. Dans ces conditions, la longueur représentative limite est $l_{VER} = 10\, \delta = 33\, nm$, soit $l_{VER} \approx \lambda/2$.

On remarque donc que lorsque l'on se rapproche du régime transitionnel (pour $K_n = \lambda/L \sim 1$), la longueur L devient du même ordre de grandeur que l_{VER} et alors il n'est pas évident que l'équation de Boltzmann, sous les hypothèses retenues ici, soit suffisante. Bien que l'objectif soit ici d'atteindre le régime transitionnel par résolution de l'équation de Boltzmann, ces considérations sur les fluctuations dépassent le cadre de cette thèse.

3 Limite hydrodynamique de l'équation de Boltzmann

Le comportement hydrodynamique d'un fluide ne se manifeste que sur des intervalles de temps beaucoup plus longs que le temps de relaxation τ et des distances beaucoup plus grandes que le libre parcours moyen λ des particules. Cette séparation des échelles micro/macro, caractérisée par de petits nombres de Knudsen ($K_n \ll 1$), implique que le fluide soit toujours très proche de l'équilibre.

De façon générale, l'équation de Boltzmann-BGK (2.42) est équivalente à une hiérarchie d'équations de bilan sur les moments de la fonction de distribution. Cette hiérarchie permet, tout comme l'équation de Boltzmann-BGK, de représenter l'évolution d'un gaz dont la fonction de distribution est arbitrairement éloignée de l'équilibre ; du moins en principe. Cependant, cette hiérarchie est ouverte et il est nécessaire, pour la rendre opérationnelle, de déterminer des relations de fermeture.

Dans ce chapitre, nous commencerons par écrire et commenter cette hiérarchie. Après avoir montré que les trois premières équations de la hiérarchie des moments sont des équations de bilan sur les invariants collisionnels, nous verrons que la résolution de l'équation de Boltzmann-BGK équivaut, lorsque la limite hydrodynamique est atteinte, à résoudre les équations classiques de la MMC, *i.e.* les équations de continuité sur les champs de densité, d'impulsion et d'énergie.

3.1 Hiérarchie des moments

L'obtention de l'évolution des moments associés à l'évolution de la fonction de distribution, se fait par intégration de l'équation de Boltzmann (2.42) pondérée par $\Psi_k(\boldsymbol{v}) = \bigotimes^k \boldsymbol{v}$ sur l'espace des vitesses. Dans le cas où le terme de force est nul, Ψ_k ne dépendant ni de l'espace, ni du temps, nous obtenons :

$$\frac{\partial}{\partial t}\int \Psi_k(\boldsymbol{v})\, f d\boldsymbol{v} + \boldsymbol{\nabla}_{\boldsymbol{x}} \int \underbrace{\Psi_k(\boldsymbol{v})\, \boldsymbol{v}}_{\Psi_{k+1}(\boldsymbol{v})}\, f d\boldsymbol{v} = \int \Psi_k(\boldsymbol{v}) \left(\frac{\partial f}{\partial t}\right)_{coll} d\boldsymbol{v}. \tag{3.1}$$

Ou encore, comme le moment d'ordre k de f est $\mathcal{M}_k = \int \Psi_k\, f d\boldsymbol{v}$:

$$\partial_t \mathcal{M}_k + \nabla \mathcal{M}_{k+1} = \left(\frac{\partial \mathcal{M}_k}{\partial t}\right)_{coll}. \tag{3.2}$$

Ce système d'équations aux dérivés partielles indicées en k s'appelle la *hiérarchie des moments de* f. Cette hiérarchie est ouverte, en ce sens que la résolution de

l'équation d'ordre k nécessite la solution de l'équation d'ordre $k+1$. Pour fermer cette hiérarchie, nous introduirons plus tard certaines hypothèses sur la fonction de distribution. Nous pouvons néanmoins lui donner un sens. En effet, à un rang k donné, cette équation signifie que dans un volume $\mathcal{V}$ de l'espace des configurations, délimité par la surface $\partial\mathcal{V}$, la variation temporelle du moment $\mathcal{M}_k$, est égale au flux entrant du moment $\mathcal{M}_{k+1}$ à travers $\partial\mathcal{V}$, plus la variation de $\mathcal{M}_k$ due aux collisions. Pour s'en convaincre, il suffit d'intégrer l'équation (3.2) sur le volume $\mathcal{V}$, puis grâce à la formule d'Ostrogradsky, à transformer l'intégrale de $\nabla\mathcal{M}_{k+1}$ en intégrale de surface sur $\partial\mathcal{V}$. Soit :

$$\int_{\mathcal{V}} \nabla\mathcal{M}_{k+1}\, dV = \int_{\partial\mathcal{V}} \mathcal{M}_{k+1} d\boldsymbol{s}, \tag{3.3}$$

où $d\boldsymbol{s}$ est le vecteur normal à la surface, dirigé vers l'extérieur, et de longueur égale à l'élément qu'il représente.

Soit, après remplacement :

$$\partial_t \int_{\mathcal{V}} \mathcal{M}_k dV = -\int_{\partial\mathcal{V}} \mathcal{M}_{k+1} d\boldsymbol{s} + \int_{\mathcal{V}} \left(\frac{\partial\mathcal{M}_k}{\partial t}\right)_{coll} dV. \tag{3.4}$$

Rappelons enfin que cette hiérarchie représente l'évolution d'un gaz dont la fonction de distribution est arbitraire. Nous pourrions donc envisager de l'utiliser en lieu et place de l'équation de Boltzmann.

3.2 Invariants collisionnels

Regardons dans un premier temps la variation que l'opérateur de collisions produit sur les moments de f. Pour ce faire, multiplions l'opérateur de collision par la fonction $\Psi_k(\boldsymbol{v}) = \bigotimes^k \boldsymbol{v}$, puis intégrons sur l'espace de vitesses :

$$\left(\frac{\partial\mathcal{M}_k}{\partial t}\right)_{coll} = -\frac{1}{\tau}\int \Psi_k(\boldsymbol{v})(f - f^{eq})d\boldsymbol{v} = -\frac{1}{\tau}(\mathcal{M}_k - \mathcal{M}_k^{eq}). \tag{3.5}$$

Nous avons vu, lors de la recherche de la fonction de distribution d'équilibre (cf. section 2.2.4), que la densité, l'impulsion et l'énergie interne étaient conservées lors des collisions :

$$\rho = \int f d\boldsymbol{v} = \int f^{eq} d\boldsymbol{v} \equiv \mathcal{M}_0 = \mathcal{M}_0^{eq}, \tag{3.6}$$

$$\rho\boldsymbol{u} = \int \boldsymbol{v} f d\boldsymbol{v} = \int \boldsymbol{v} f^{eq} d\boldsymbol{v} \equiv \mathcal{M}_1 = \mathcal{M}_1^{eq}, \tag{3.7}$$

$$2\,\rho\,\varepsilon = \int |\boldsymbol{v} - \boldsymbol{u}|^2 f d\boldsymbol{v} = \int |\boldsymbol{v} - \boldsymbol{u}|^2 f^{eq} d\boldsymbol{v} \equiv Tr(\mathcal{M}_2) = Tr(\mathcal{M}_2^{eq}). \tag{3.8}$$

Cette propriété impose donc :

$$\left(\frac{\partial \mathcal{M}_0}{\partial t}\right)_{coll} = \left(\frac{\partial \mathcal{M}_1}{\partial t}\right)_{coll} = \left(\frac{\partial Tr(\mathcal{M}_2)}{\partial t}\right)_{coll} = 0. \tag{3.9}$$

Nous pouvons donc écrire que si $\chi(\boldsymbol{x},\boldsymbol{v},t)$ est un invariant collisionnel, alors :

$$\int \chi(\boldsymbol{x},\boldsymbol{v},t) \left(\frac{\partial f}{\partial t}\right)_{coll} d\boldsymbol{v} = 0. \tag{3.10}$$

3.3 Équations de bilan local sur les invariants collisionnels

Ces équations s'obtiennent en multipliant l'équation de Boltzmann par un invariant collisionnel et en l'intégrant sur l'espace des vitesses :

$$\int \chi(\boldsymbol{x},\boldsymbol{v},t) \left(\partial_t f + \boldsymbol{v} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} f + \boldsymbol{g} \cdot \boldsymbol{\nabla}_{\boldsymbol{v}} f\right) d\boldsymbol{v} = 0. \tag{3.11}$$

Équation que l'on peut réécrire sous la forme suivante :

$$\partial_t \langle \rho\chi \rangle - \rho \langle \partial_t \chi \rangle + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \langle \rho\chi \boldsymbol{v} \rangle - \rho \langle \boldsymbol{v} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} \chi \rangle - \rho \langle \boldsymbol{g} \cdot \boldsymbol{\nabla}_{\boldsymbol{v}} \chi \rangle = 0, \tag{3.12}$$

où nous avons réintroduit l'opérateur de moyenne :

$$\langle \mathcal{A} \rangle(\boldsymbol{x},t) = \frac{1}{\rho} \int \mathcal{A}\, f\, d\boldsymbol{v}. \tag{3.13}$$

On déduit alors de cette équation générale de bilan local, les équations de bilans pour la masse, l'impulsion et l'énergie interne.

3.3.1 Bilan de masse

Pour $\chi = 1$, on obtient grâce à l'équation (3.12) :

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0. \tag{3.14}$$

Ou encore, en utilisant la dérivée particulaire $D_t = \partial_t + \boldsymbol{u}\nabla$:

$$D_t \rho + \rho \nabla \cdot \boldsymbol{u} = 0. \tag{3.15}$$

3.3.2 Bilan de quantité de mouvement

Pour $\chi = \boldsymbol{v}$, on obtient grâce à l'équation (3.12) :

$$\partial_t(\rho \boldsymbol{u}) + \nabla \cdot \langle \rho \boldsymbol{v}\boldsymbol{v} \rangle = \rho \boldsymbol{g}, \tag{3.16}$$

que l'on peut réécrire, connaissant l'expression du tenseur des pressions :

$$\boldsymbol{P} = \int \boldsymbol{\xi}\boldsymbol{\xi}\, f\, d\boldsymbol{\xi}, \quad (\boldsymbol{\xi} = \boldsymbol{v} - \boldsymbol{u}), \tag{3.17}$$

sous la forme :

$$\partial_t(\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u}\boldsymbol{u}) + \nabla \cdot \boldsymbol{P} = \rho \boldsymbol{g}, \tag{3.18}$$

ou encore :

$$\rho D_t \boldsymbol{u} = \rho \boldsymbol{g} - \nabla \cdot \boldsymbol{P}. \tag{3.19}$$

On a vu section 2.2.3 que $\boldsymbol{P} = \boldsymbol{\Pi} + \mathbb{I}p$, avec $p = \rho k_B T/m = \rho\theta$, la pression cinétique et $\boldsymbol{\Pi}$ le tenseur des contraintes visqueuses, qui est de trace nulle. On pourra donc encore écrire :

$$\rho D_t \boldsymbol{u} = \rho \boldsymbol{g} - \nabla p - \nabla \cdot \boldsymbol{\Pi}, \tag{3.20}$$

où $\rho \boldsymbol{g} - \nabla p$ est la pression motrice et $\nabla \cdot \boldsymbol{\Pi}$ la dissipation visqueuse.

3.3.3 Bilan d'énergie interne

Pour $\chi = |\boldsymbol{v} - \boldsymbol{u}|^2/2 = \boldsymbol{\xi}^2/2$, on obtient, grâce à l'équation (3.12) :

$$\partial_t(\rho\, \varepsilon + \rho\, \boldsymbol{u}^2) + \nabla \cdot \langle \frac{1}{2} \boldsymbol{\xi}\boldsymbol{\xi}^2 \rangle = 0, \tag{3.21}$$

que l'on réécrira :

$$\partial_t(\rho\, \varepsilon + \rho\, \boldsymbol{u}^2) + \nabla \cdot \left[\boldsymbol{u} \left(\rho\, \varepsilon + \rho\, \boldsymbol{u}^2 \right) + \boldsymbol{P} \cdot \boldsymbol{u} + \boldsymbol{q} \right] = 0, \tag{3.22}$$

sachant que le flux de chaleur (cf. Eq. (2.18)), ou d'énergie interne par conduction, est :

$$2\, \boldsymbol{q} = \int \boldsymbol{\xi}\boldsymbol{\xi}^2 f(\boldsymbol{x},\boldsymbol{v},t)\, d\boldsymbol{v}, \tag{3.23}$$

Finalement, en utilisant la dérivée particulaire D_t, on écrira sous forme compacte :

$$\rho D_t\, \varepsilon + \nabla \cdot \boldsymbol{q} + \boldsymbol{P} : (\nabla \boldsymbol{u}) = 0, \tag{3.24}$$

où l'opérateur ":" est le produit tensoriel totalement contracté.

3.4 Équations hydrodynamiques

En l'état, nous ne pouvons pas utiliser les équations de bilans (3.15), (3.20) et (3.24), car elles sont pour l'instant dépourvues de relations de fermeture ou autrement dit, d'équations constitutives. En effet, les équations de bilans de quantité de mouvement et d'énergie interne font intervenir le tenseur des contraintes visqueuses $\boldsymbol{P}$ et le flux de chaleur par conduction $\boldsymbol{q}$. Or, ces quantités sont des moments de la fonction de distribution f, dont le calcul nécessite d'en connaître l'expression. Il nous faut donc déterminer une forme pertinente de f pour en déduire $\boldsymbol{P}$ et $\boldsymbol{q}$ et ainsi donner aux équations de bilan précédentes un sens physique, celui des équations hydrodynamiques. Pour trouver une forme pertinente pour f, nous suivrons la démarche de Chapman et Enskog, présentée ici.

Rappelons d'abord que le comportement hydrodynamique d'un fluide nécessite la séparation des échelles micro/macro. Cette séparation, caractérisée par un nombre de Knudsen ($Kn = \lambda/L$) petit, implique que le fluide soit toujours très proche de l'équilibre thermodynamique local.

Pour de tels systèmes, proches de l'équilibre, il semble naturel d'utiliser la théorie des perturbations. C'est ce qui est fait dans le *développement de Chapman-Enskog*, qui consiste en un développement perturbatif de la fonction de distribution f, autour de l'équilibre local f^{eq}.

Le *petit paramètre* ϵ utilisé dans le développement devant refléter la *séparation des échelles*, il pourra être assimilé au nombre de Knudsen. De plus, nous supposerons que f ne s'écarte que linéairement de l'équilibre [1]. On écrira donc f sous la forme suivante :

$$f = f^{(0)} + f^{(1)}, \tag{3.25}$$

où $f^{(0)}$ est d'ordre $\epsilon^0 = 1$ et $f^{(1)}$ d'ordre ϵ, on à donc $f^{(1)} \ll f^{(0)}$ et $f^{(0)} \equiv f^{eq}$.

À l'ordre zéro en ϵ, pour $f = f^{eq}$, les contraintes visqueuses $\boldsymbol{\Pi}^{(0)}$ et le flux de chaleur $\boldsymbol{q}^{(0)}$, moments associés de la distribution de Maxwell-Boltzmann, sont nuls. On parle donc, à cet ordre, d'hydrodynamique non dissipative et les équations hydrodynamiques associées sont les équations d'Euler.

À l'ordre 1 en ϵ, la contribution aux contraintes visqueuses $\boldsymbol{P}^{(1)}$ et au flux de chaleur $\boldsymbol{q}^{(1)}$ sont les moments associés de la distribution $f^{(1)}$, dont nous ne connaissons pour l'instant pas l'expression. Tachons donc d'évaluer l'ordre de grandeur de cette partie

1. Il faut en effet se demander ici, s'il est justifié de poursuivre le développement de f au-delà de l'ordre en ϵ. En effet, étant donné la petitesse de $\epsilon \equiv K_n$, le fluide doit se trouver très proche de l'équilibre et son éloignement à celui-ci devrait donc se limiter à une réponse linéaire. Néanmoins, des développements à des ordres supérieurs sont pratiqués et débouchent sur les systèmes d'équations de Burnett [GCVU08,BL08,XJS01,XL04] (limitées à des nombres de Knudsen $K_n \leq 0{,}1$ pour les premières (cf. [XJS01])) et Super-Burnett [Xu03]. Ce problème dépasse le cadre de cette thèse et nous nous limiterons au cas de la réponse linéaire, qui est suffisant pour obtenir les équations de Navier-Stokes-Fourier.

hors équilibre de f. Sachant que la longueur caractéristique macroscopique de notre système est L et considérant la vitesse du son c_s comme vitesse caractéristique, nous pouvons approximer la partie transport de l'équation de Boltzmann (2.42) par :

$$\mathcal{L}f \sim \frac{c_s}{L} f. \tag{3.26}$$

Ainsi, nous aurons :

$$\frac{c_s}{L} f \sim -\frac{1}{\tau}(f - f^{eq}), \tag{3.27}$$

soit, en remplaçant f par son développement perturbatif (3.25) :

$$\frac{c_s}{L} f^{(0)} + \frac{c_s}{L} f^{(1)} \sim -\frac{1}{\tau}(f^{(0)} + f^{(1)} - f^{eq}), \tag{3.28}$$

qui selon les hypothèses $f^{(1)} \ll f^{(0)}$ et $f^{(0)} \equiv f^{eq}$, donne :

$$\frac{c_s}{L} f^{eq} \sim -\frac{1}{\tau} f^{(1)}. \tag{3.29}$$

Cette expression nous permet d'écrire :

$$f^{(1)} \sim -\frac{\tau c_s}{L} f^{eq} = -K_n f^{eq}, \tag{3.30}$$

car $\tau c_s = \lambda$ est le libre parcours moyen des particules. On montre ainsi que $f^{(1)}$ est bien d'ordre $\epsilon \equiv K_n$, et que ce développement ne sera valide que dans la limite hydrodynamique.

Formellement, nous écrirons finalement :

$$\mathcal{L}f^{eq} = -\frac{1}{\tau} f^{(1)}, \tag{3.31}$$

ou encore :

$$f^{(1)} = -\tau \left(\partial_t + \boldsymbol{v} \cdot \partial_{\boldsymbol{x}} + \boldsymbol{g} \cdot \partial_{\boldsymbol{v}}\right) f^{eq}, \tag{3.32}$$

à l'ordre d'approximation considéré.

L'équation (3.32) nous permet alors de calculer les contraintes visqueuses $\boldsymbol{P}^{(1)}$ et le flux de chaleur $\boldsymbol{q}^{(1)}$, moments associés de la distribution $f^{(1)}$, car l'expression de f^{eq} est connue. Après quelques calculs, relativement techniques (voir par exemple Pottier [Pot07] ou Wolf-Gladrow [WG00]), on obtient :

$$\boldsymbol{P}^{(1)} = -\rho\nu\boldsymbol{\Lambda}, \tag{3.33}$$

avec $\nu = \theta\tau$ la viscosité cinématique et $\boldsymbol{\Lambda}$ le tenseur déviatorique, tel que :

$$\Lambda_{\alpha\beta} = \partial_\alpha u_\beta + \partial_\beta u_\alpha - \frac{2}{D}\partial_\gamma u_\gamma \delta_{\alpha\beta}. \tag{3.34}$$

Le Tenseur des pressions est finalement $\boldsymbol{P} = \boldsymbol{P}^{(0)} + \boldsymbol{P}^{(1)}$:

$$\boldsymbol{P} = \rho\theta\mathbb{I} - \rho\nu\boldsymbol{\Lambda}, \tag{3.35}$$

la viscosité de volume est donc nulle.

Et pour $\boldsymbol{q}^{(1)} \equiv \boldsymbol{q}$:

$$\boldsymbol{q} = -\left(\frac{D+2}{2}\right)\rho\alpha\boldsymbol{\nabla}\theta. \tag{3.36}$$

Dans cette expression, $\alpha = \theta\tau$, est la diffusivité thermique et le facteur $(D+2)/2 = C_p$, est la capacité calorifique pour un gaz monoatomique.

Pour ce développement de f, au premier ordre en ϵ, les équations hydrodynamiques obtenues sont celles de Navier-Stokes-Fourier, valides pour les fluides dissipatifs, *i.e.* visqueux et conducteurs de la chaleur.

On remarquera que dans l'approximation du temps de relaxation, la viscosité cinématique (diffusion de quantité de mouvement) et la diffusivité thermique sont égales. Le nombre de Prandtl $P_r = \nu/\alpha$ sera donc égal à l'unité.

Une autre méthode pour déterminer une forme pertinente pour f a été proposée par H. Grad [Gra49b, Gra49a]. Elle consiste à rechercher des solutions de l'équation Boltzmann-BGK projetée sur la base des polynômes d'Hermite. La méthode de Grad permet de déterminer des équations pour le tenseur des pressions et le flux d'énergie, formant, avec les équations de bilan (3.15), (3.20) et (3.24) ce que l'on appelle la théorie des 13 moments de Grad. Nous n'entrerons pas ici dans les détails du développement de cette théorie, mais utiliserons le développement selon les polynômes d'Hermite. Car comme cela a été proposé récemment par Shan et He [SH98] (voir [ZSC06] pour une présentation plus simple de la méthode) et comme nous le verrons, partir d'un tel développement permet d'étendre les méthodes de Boltzmann sur réseau (LBM) au régime transitionnel. La démarche de développement des LBM proposée par Shan et He présente certains intérêts pour l'hydrodynamique. Notamment, pour corriger un défaut d'invariance galiléenne des schémas LBM [NSC08] ou encore supporter la thermique (voir en priorité les travaux de Karlin, Bardow et Ansumali). Signalons enfin que cette démarche n'a pour l'instant été que très peu exploitée pour étendre les LBM aux écoulements microscopiques. Les principaux travaux effectués en ce sens (principalement par Kim [KPB08a, KP08, KPB08b] et Ansumali [CAK06, AKFB06], bien que Ansumali ne développe pas la fonction d'équilibre de la même façon) sont tous basés sur l'utilisation d'une quadratique de Gauss-Hermite d'ordre 4 (que nous appellerons $\mathcal{H}_4$).

L'analyse systématique des différents ordres réalisée dans cette thèse est inédite. Cette analyse nous permettra par ailleurs de proposer un nouveau type de schémas, que nous qualifierons de composites. Ces schémas seront introduits pour accélérer la convergence de la procédure de Shan et He, que nous verrons être lente.

3.5 Temps de relaxation

Comme cela a été montré dans la section précédente, le temps de relaxation vers l'équilibre τ est un paramètre essentiel. Il permet en effet de définir la viscosité et la conductivité thermique du gaz (de façon non indépendante dans notre cas) donc de paramétrer nos simulations. En effet, pour des problèmes anisothermes –le nombre de Prandtl étant fixé à l'unité dans l'approximation du temps de relaxation unique– seuls deux nombres sans dimension seront nécessaires pour définir une similitude entre une situation physique et une simulation. Ces deux nombres seront un couple pris dans $\{K_n, R_e, M_a\}$, étant donnée la relation qui existe entre ces trois derniers (cf. annexe B.4).

Jusqu'à présent, aucune hypothèse sur la dépendance du temps de relaxation avec les grandeurs macroscopiques, telles que la densité ou la température, n'a été faite, or de telles dépendances existent. En effet, une augmentation de la densité ayant pour effet de diminuer la distance intermoléculaire moyenne, celle-ci aura pour effet d'augmenter la fréquence de collision et donc de diminuer le temps de relaxation. D'autre part, la section efficace de collision des molécules est dépendante de leur vitesse relative, et comme une augmentation de la température est assimilable (en moyenne quadratique), à une augmentation de cette vitesse relative. Alors, à une augmentation de la température correspondra une diminution de la section efficace de collision, donc à une diminution du temps de relaxation.

3.5.1 Temps de relaxation local

Le temps de relaxation vers l'équilibre τ détermine donc, ici, à la fois la viscosité ν et la diffusivité thermique α. Cependant, ce temps dépendra localement de la densité ρ et de la température θ. En effet, une augmentation de la densité ou de la température augmentera la fréquence de collision entre les particules du fluide et donc, diminuera le temps de relaxation.

La loi généralement admise pour définir $\tau(\rho,\theta)$ est la suivante :

$$\tau(\rho,\theta) = \tau_0 \frac{\rho_0}{\rho} \left(\frac{\theta_0}{\theta}\right)^{1-\varphi}, \tag{3.37}$$

où ρ_0 et θ_0 sont la densité et la température de référence, pour lesquelles le temps de relaxation est τ_0 et où φ est appelé l'*exposant thermique* du gaz. Il dépend du potentiel d'interaction entre les particules du fluide. Par la suite nous considérerons des gaz de sphères dures, pour lesquels $\varphi = 1/2$. Les ouvrages de Hirschfelder et *al.* [HCB54] et de Chapman et *al.* [CC91] fournissent des compléments sur ce sujet, entre autres.

3.5.2 Temps de relaxation et nombre de Knudsen

Partant de la relation entre les nombres de Knudsen, de Mach et de Reynolds (cf. Annexe B) :

$$K_n \sim \frac{M_a}{R_e}, \tag{3.38}$$

nous obtenons [2] :

$$K_n = \frac{A\nu}{L\sqrt{\theta}}. \tag{3.39}$$

Or, nous savons que la viscosité est reliée au temps de relaxation τ, par $\nu = \theta\tau$, où θ est la température. D'où, d'après l'expression (3.37) du temps de relaxation :

$$K_n = \frac{A}{L\sqrt{\theta}}\,\theta\,\tau_0\frac{\rho_0}{\rho}\left(\frac{\theta_0}{\theta}\right)^{1-\varphi}. \tag{3.40}$$

Considérons alors un Knudsen de référence K_{n0}, défini à la densité ρ_0 et à la température θ_0, il s'écrit, étant donné l'expression précédente :

$$K_{n0} = \frac{A}{L\sqrt{\theta_0}}\,\theta_0\,\tau_0, \tag{3.41}$$

d'où le temps de relaxation associé :

$$\tau_0 = K_{n0}\frac{L}{A\sqrt{\theta_0}}. \tag{3.42}$$

Finalement, en remplaçant τ_0 (Eq. (6.29)) dans l'expression de K_n (Eq. (6.28)), nous obtenons l'expression du nombre de Knudsen local :

$$K_n = K_{n0}\,\frac{\rho_0}{\rho}\left(\frac{\theta_0}{\theta}\right)^{1/2-\varphi}, \tag{3.43}$$

qui dans le cas d'un gaz de sphères dures ($\varphi = 1/2$) devient :

$$K_n = K_{n0}\,\frac{\rho_0}{\rho}, \tag{3.44}$$

que l'on remarquera être indépendant de la température pour les sphères dures.

Ce résultat est intéressant, car il montre la possibilité de faire varier localement le nombre de Knudsen, donc le régime de l'écoulement, par la variation de la densité locale ρ. Ce cas de figure sera étudié section 6.5, où un écoulement mis en mouvement par un gradient de densité dans un long canal sera simulé.

2. Le facteur A dépend du modèle utilisé pour obtenir la relation entre les nombres $K_{n,}$, M_a et R_e. Il est généralement pris égal à $A = \sqrt{\gamma\pi/2}$, cependant les schémas développés dans cette thèse ont pour coefficient $A = \sqrt{2.5} \neq \sqrt{\pi/2}$ en deux dimensions.

4 Écoulements glissants et transitionnels

Ce chapitre porte sur les régimes non hydrodynamiques, caractérisés par des nombres de Knudsen finis, compris dans la fourchette $0{,}01 \lesssim K_n \lesssim 10$.

Le régime de glissement correspond à des écoulements dont le nombre de Knudsen est typiquement compris entre 0,01 et 0,1. La particularité de ce régime est que le fluide peut encore être représenté par la MMC. Cependant, certaines hypothèses ne sont plus valides, notamment celle de non-glissement appliquée comme condition aux limites à la surface des corps solides délimitant l'écoulement. Cette hypothèse n'est pas la seule à ne plus être applicable, en effet, tous les champs présentent *a priori* dans ce régime, une discontinuité (du moins une variation sur une très courte distance) aux interfaces gaz-solide. L'épaisseur sur laquelle cette variation brutale des champs se produit s'appelle la *couche de Knudsen*. À l'intérieur de cette couche, dont la taille est de l'ordre du libre parcours moyen, les molécules ne collisionnent quasiment pas entre elles et y volent donc quasi librement. Au-delà de la couche de Knudsen, se trouve la *couche de Prandtl*, dans laquelle les molécules du fluide collisionnent suffisamment pour que le régime hydrodynamique soit atteint, donc que les équations de Navier-Stokes soient applicables.

Le régime transitionnel correspond, quant à lui, à des écoulements pour lesquels le nombre de Knudsen est compris entre 0,1 et 10. Il apparaît lorsque l'épaisseur de la couche de Knudsen devient non négligeable par rapport à la taille L caractéristique de l'écoulement. Typiquement, lorsque celle-ci est de l'ordre de $L/10$, étant donné que l'épaisseur de la couche de Knudsen est de l'ordre du libre parcours moyen. À partir de cette limite ($K_n \approx 0{,}1$), la couche de Knudsen commence à envahir l'ensemble de l'écoulement et les collisions n'y sont plus suffisantes pour que le régime hydrodynamique soit atteint.

Finalement, lorsque la taille caractéristique L de l'écoulement devient de l'ordre de l'épaisseur de la couche de Knudsen, alors les molécules de l'écoulement ne collisionnent quasiment plus entre elles et l'écoulement est un écoulement de particules libres, que l'on appelle aussi un gaz de Knudsen. Dans ce cas limite, les seules collisions que subissent les molécules du gaz sont celles avec les parois.

Ces deux derniers régimes, transitionnel et de Knudsen, ont donc pour point commun la couche de Knudsen, qui est un effet de bord et dont l'influence sur l'écoulement est d'autant plus marquée que le nombre de Knudsen est grand. L'étude de ces deux régimes commencera donc naturellement par l'étude des conditions aux limites sur les surfaces solides. Nous verrons ensuite l'effet que produit le glissement sur un écoulement de Poiseuille plan, cependant, nous nous limiterons au régime de

glissement, car aucune solution analytique n'existe au-delà de ce régime. Pour le régime transitionnel, nous aurons donc recours à des mesures expérimentales de débits pour vérifier les résultats numériques. Finalement, quelques effets thermiques serons introduits et commentés.

4.1 Conditions aux bords

Cette section concerne les conditions que doit satisfaire la fonction de distribution f sur une surface solide. Dans le cas d'un gaz s'écoulant autour ou à l'intérieur d'une région bornée par un ou plusieurs objets solides, ces conditions aux bords doivent rendre compte de l'interaction des particules du gaz avec les parois. La difficulté théorique de ce problème réside principalement dans le manque d'information sur la structure des surfaces et sur le potentiel d'interaction des molécules avec celles-ci. En effet, une molécule arrivant à la surface du solide peut, par exemple, être adsorbée, réagir chimiquement ou encore heurter une autre molécule de gaz déjà adsorbée. De plus, la surface du solide peut présenter une certaine porosité (à une échelle plus petite que celle que l'on considère), dans laquelle les molécules peuvent être piégées. Pour modéliser l'ensemble de ces phénomènes, nous allons introduire le modèle d'interaction simplifié, proposé par Maxwell [MN69, Cer94, Bir94] et considérerons que les parois sont non poreuses et non adsorbantes.

4.1.1 Conditions aux bords cinétiques

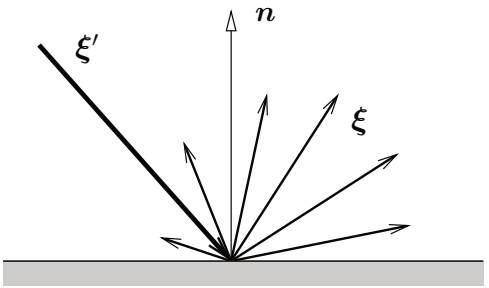


FIGURE 4.1 – Illustration de la diffusion par une paroi d'un faisceau de particules incidentes à la vitesse $\boldsymbol{\xi}'$. La vitesse des particules réémises est notée $\boldsymbol{\xi}$. Le vecteur $\boldsymbol{n}$ est la normale à la paroi, dirigée vers le fluide.

Considérons une paroi ∂R (cf. Fig. 4.1), définie en chacun de ses points $\boldsymbol{x}_W \in \partial R$, par la connaissance de sa vitesse $\boldsymbol{U}_W$, de sa température T_W et de sa normale sortante $\boldsymbol{n}$. Par la suite, nous exprimerons la fonction de distribution dans le référentiel de la paroi, soit $f(\boldsymbol{\xi})$, avec $\boldsymbol{\xi} = \boldsymbol{v} - \boldsymbol{U}_W$; de plus nous noterons $\boldsymbol{\xi}'$ (tq. $\boldsymbol{\xi}' \cdot \boldsymbol{n} < 0$) la vitesse des particules incidentes et $\boldsymbol{\xi}$ (tq. $\boldsymbol{\xi} \cdot \boldsymbol{n} > 0$) celle des particules réémises. Finalement, la fonction de distribution des particules à l'équilibre avec la paroi, dans le référentiel de celle-ci, sera notée $f_W^{eq} \equiv f^{eq}(\rho_W, 0, T_W)$.

Définissons maintenant un opérateur $\mathcal{B}(\boldsymbol{\xi}' \to \boldsymbol{\xi})$ représentant la densité de probabilité qu'une molécule de vitesse comprise entre $\boldsymbol{\xi}'$ et $\boldsymbol{\xi}' + \delta\boldsymbol{\xi}'$, heurte la paroi en $\boldsymbol{x}_W$ et en ressorte avec une vitesse comprise entre $\boldsymbol{\xi}$ et $\boldsymbol{\xi} + \delta\boldsymbol{\xi}$. Si la paroi est non poreuse et non adsorbante, nous pouvons écrire :

$$|\boldsymbol{\xi} \cdot \boldsymbol{n}|\, f(\boldsymbol{\xi}) = \int_{\boldsymbol{\xi}' \cdot \boldsymbol{n} < 0} |\boldsymbol{\xi}' \cdot \boldsymbol{n}|\, f(\boldsymbol{\xi}') \mathcal{B}(\boldsymbol{\xi}' \to \boldsymbol{\xi}) d\boldsymbol{\xi}', \qquad (\boldsymbol{\xi} \cdot \boldsymbol{n} > 0), \tag{4.1}$$

avec la propriété suivante sur $\mathcal{B}(\boldsymbol{\xi}' \to \boldsymbol{\xi})$:

$$\int_{\boldsymbol{\xi} \cdot \boldsymbol{n} > 0} \mathcal{B}(\boldsymbol{\xi}' \to \boldsymbol{\xi}) d\boldsymbol{\xi} = 1, \tag{4.2}$$

signifiant qu'une molécule incidente est sûre d'être réémise. En effet, ces deux équations (4.1) et (4.2) nous permettent de montrer que la composante normale de l'impulsion hydrodynamique s'annule à la paroi :

$$\begin{aligned} \boldsymbol{n} \cdot \rho \boldsymbol{u}_W &= \int \boldsymbol{n} \cdot \boldsymbol{\xi} f(\boldsymbol{\xi})\, d\boldsymbol{\xi} \\ &= \int_{\boldsymbol{\xi} \cdot \boldsymbol{n} > 0} |\boldsymbol{\xi} \cdot \boldsymbol{n}|\, f(\boldsymbol{\xi})\, d\boldsymbol{\xi} - \int_{\boldsymbol{\xi}' \cdot \boldsymbol{n} < 0} |\boldsymbol{\xi}' \cdot \boldsymbol{n}|\, f(\boldsymbol{\xi}')\, d\boldsymbol{\xi}' \\ &= \int_{\boldsymbol{\xi} \cdot \boldsymbol{n} > 0} d\boldsymbol{\xi} \left[\int_{\boldsymbol{\xi}' \cdot \boldsymbol{n} < 0} \mathcal{B}(\boldsymbol{\xi}' \to \boldsymbol{\xi}) |\boldsymbol{\xi}' \cdot \boldsymbol{n}|\, f(\boldsymbol{\xi}')\, d\boldsymbol{\xi}' \right] - \int_{\boldsymbol{\xi}' \cdot \boldsymbol{n} < 0} |\boldsymbol{\xi}' \cdot \boldsymbol{n}|\, f(\boldsymbol{\xi}')\, d\boldsymbol{\xi}' \\ &= \int_{\boldsymbol{\xi}' \cdot \boldsymbol{n} < 0} |\boldsymbol{\xi}' \cdot \boldsymbol{n}|\, f(\boldsymbol{\xi}')\, d\boldsymbol{\xi}' - \int_{\boldsymbol{\xi}' \cdot \boldsymbol{n} < 0} |\boldsymbol{\xi}' \cdot \boldsymbol{n}|\, f(\boldsymbol{\xi}')\, d\boldsymbol{\xi}' = 0, \end{aligned} \tag{4.3}$$

où nous avons tenu compte des équations (4.1) et (4.2).

Une autre propriété importante du noyau de collision $\mathcal{B}$ est la suivante :

$$|\boldsymbol{\xi}' \cdot \boldsymbol{n}|\, f_W^{eq}(\boldsymbol{\xi}')\, \mathcal{B}(\boldsymbol{\xi}' \to \boldsymbol{\xi}) = |\boldsymbol{\xi} \cdot \boldsymbol{n}|\, f_W^{eq}(\boldsymbol{\xi})\, \mathcal{B}(-\boldsymbol{\xi} \to -\boldsymbol{\xi}'), \tag{4.4}$$

que l'on appelle *condition de balance détaillée*, ou de *réciprocité* [Cer94]. Une interprétation physique de l'équation (4.4) est que si le fluide délimité par la paroi est à l'équilibre f_W^{eq} à la vitesse $\boldsymbol{U}_W$ et à la température T_W, alors le système est invariant dans le temps, par conséquent, il est invariant par un renversement du temps, dont l'effet est d'inverser les vitesses $\boldsymbol{\xi}$ et $\boldsymbol{\xi}'$. Le flux de particules de vitesse $\boldsymbol{\xi}'$ qui alimente la vitesse $\boldsymbol{\xi}$ de la fonction de distribution est donc égal, à l'équilibre, au flux de particules de vitesses $-\boldsymbol{\xi}$ pour peupler la vitesse $-\boldsymbol{\xi}'$ de la fonction de distribution.

Une conséquence de l'équation (4.4) est que si la fonction de distribution incidente est une maxwellienne de la paroi, alors la fonction de distribution émergente sera aussi une maxwellienne de la paroi. D'où la propriété suivante :

$$|\boldsymbol{\xi} \cdot \boldsymbol{n}|\, f_W^{eq}(\boldsymbol{\xi}) = \int_{\boldsymbol{\xi}' \cdot \boldsymbol{n} < 0} |\boldsymbol{\xi}' \cdot \boldsymbol{n}|\, f_W^{eq}(\boldsymbol{\xi}')\, \mathcal{B}(\boldsymbol{\xi}' \to \boldsymbol{\xi}) d\boldsymbol{\xi}', \tag{4.5}$$

équivalente à l'équation (4.1), mais pour la fonction d'équilibre.

4.1.2 Modèle de Maxwell

De façon rigoureuse, il est impossible de construire le noyau de collisions $\mathcal{B}(\boldsymbol{\xi}' \to \boldsymbol{\xi})$ du fait de la complexité des interactions sous-jacentes entre le fluide et la paroi. Le modèle de Maxwell consiste à considérer que l'interaction du fluide avec la paroi correspond au mélange de deux types d'interactions élémentaires. Les molécules peuvent soit être réflectées spéculairement sur la surface, nous avons alors :

$$f(\boldsymbol{x}_W,\boldsymbol{\xi},t) = f(\boldsymbol{x}_W,\underbrace{\boldsymbol{\xi} - 2\boldsymbol{n}(\boldsymbol{n}\cdot\boldsymbol{\xi})}_{\boldsymbol{\xi}'},t), \quad (\boldsymbol{\xi}\cdot\boldsymbol{n} > 0, \boldsymbol{\xi}'\cdot\boldsymbol{n} < 0), \tag{4.6}$$

soit être diffusées après avoir thermalisé avec la paroi. La fonction de distribution des particules réémises sera donc la distribution de Maxwell-Boltzmann f_W^{eq} dont la vitesse moyenne et la température sont celles de la paroi.

Le modèle de Maxwell tient compte de ces deux phénomènes, en les pondérant par un coefficient β (tq. $0 \leq \beta \leq 1$), appelé *coefficient d'accommodation*, d'où la fonction de distribution réémise suivante :

$$f(\boldsymbol{x}_W,\boldsymbol{\xi},t) = (1-\beta)\, f(\boldsymbol{x}_W,\boldsymbol{\xi} - 2\boldsymbol{n}(\boldsymbol{n}\cdot\boldsymbol{\xi}),t) + \beta\, f_W^{eq}(\boldsymbol{x}_W,\boldsymbol{\xi},t). \tag{4.7}$$

Considérons alors les cas particuliers $\beta = 0$ et $\beta = 1$. Pour $\beta = 0$, la paroi est totalement réfléchissante et le noyau de collision est simplement :

$$\mathcal{B}_{\text{sp}}(\boldsymbol{\xi}' \to \boldsymbol{\xi}) = \delta(\boldsymbol{\xi}' - \boldsymbol{\xi} - 2\boldsymbol{n}(\boldsymbol{n}\cdot\boldsymbol{\xi})). \tag{4.8}$$

Pour $\beta = 1$, le fluide qui émerge de la surface solide adopte la fonction de distribution d'équilibre de la paroi. Ainsi, la fonction de distribution émergente perd totalement la mémoire de ce qu'elle était, avant de heurter la surface. Le peuplement des faisceaux de vitesses $\boldsymbol{\xi}$ (fonction de distribution émergente) est donc indépendant du peuplement des faisceaux de vitesses $\boldsymbol{\xi}'$ (fonction de distribution incidente). À partir de l'équation (4.5), nous obtenons le noyau de collision pour une paroi totalement diffusante :

$$\mathcal{B}_{\text{diff}}(\boldsymbol{\xi}) \equiv \mathcal{B}(\boldsymbol{\xi}' \to \boldsymbol{\xi}) = \frac{|\boldsymbol{\xi}\cdot\boldsymbol{n}|\, f_W^{eq}(\boldsymbol{\xi})}{\int_{\boldsymbol{\xi}'\cdot\boldsymbol{n}<0} |\boldsymbol{\xi}'\cdot\boldsymbol{n}|\, f_W^{eq}(\boldsymbol{\xi}')d\boldsymbol{\xi}'}. \tag{4.9}$$

En réinjectant le noyau de collision précédent dans l'équation (4.5), nous obtenons l'expression pour la fonction de distribution émergente pour $\beta = 1$:

$$f(\boldsymbol{x}_W,\boldsymbol{\xi},t) = f_W^{eq}(\boldsymbol{x}_W,\boldsymbol{\xi},t)\frac{\int_{\boldsymbol{\xi}'\cdot\boldsymbol{n}<0} |\boldsymbol{\xi}\cdot\boldsymbol{n}|\, f(\boldsymbol{x}_W,\boldsymbol{\xi}',t)d\boldsymbol{\xi}}{\int_{\boldsymbol{\xi}'\cdot\boldsymbol{n}<0} |\boldsymbol{\xi}'\cdot\boldsymbol{n}|\, f_W^{eq}(\boldsymbol{x}_W,\boldsymbol{\xi}',t),\, d\boldsymbol{\xi}'}, \tag{4.10}$$

que l'on appelle les conditions aux bords diffus.

Finalement, pour β quelconque, l'expression pour la fonction de distribution émer-

gente est la suivante :

$$f(\boldsymbol{x}_W,\boldsymbol{\xi},t) = (1-\beta)\, f(\boldsymbol{x}_W,\boldsymbol{\xi} - 2\boldsymbol{n}(\boldsymbol{n}\cdot\boldsymbol{\xi}),t) + \beta\, f_W^{eq}(\boldsymbol{x}_W,\boldsymbol{\xi},t) \frac{\int_{\boldsymbol{\xi}'\cdot\boldsymbol{n}<0} |\boldsymbol{\xi}\cdot\boldsymbol{n}|\, f(\boldsymbol{x}_W,\boldsymbol{\xi}',t) d\boldsymbol{\xi}}{\int_{\boldsymbol{\xi}'\cdot\boldsymbol{n}<0} |\boldsymbol{\xi}'\cdot\boldsymbol{n}|\, f_W^{eq}(\boldsymbol{x}_W,\boldsymbol{\xi}',t) d\boldsymbol{\xi}'}. \tag{4.11}$$

4.2 Vitesse de glissement à la paroi

Maintenant que nous avons une expression explicite pour la fonction de distribution sur la paroi, nous pouvons, en calculant son second moment, évaluer la vitesse hydrodynamique sur celle-ci :

$$\rho \boldsymbol{u}_W = \int \boldsymbol{\xi} f(\boldsymbol{\xi})\, d\boldsymbol{\xi} \tag{4.12}$$

$$\begin{aligned} &= \int_{\boldsymbol{\xi}\cdot\boldsymbol{n}>0} \boldsymbol{\xi}\, f(\boldsymbol{\xi})\, d\boldsymbol{\xi} + \int_{\boldsymbol{\xi}'\cdot\boldsymbol{n}\leq 0} \boldsymbol{\xi}'\, f(\boldsymbol{\xi}')\, d\boldsymbol{\xi}' \\ &= \int_{\boldsymbol{\xi}'\cdot\boldsymbol{n}<0} \boldsymbol{\xi}'\, f(\boldsymbol{\xi}')\, d\boldsymbol{\xi}' \\ &\quad + (1-\beta) \int_{\boldsymbol{\xi}\cdot\boldsymbol{n}>0} \boldsymbol{\xi} f(\boldsymbol{x},\boldsymbol{\xi} - 2\boldsymbol{n}(\boldsymbol{n}\cdot\boldsymbol{\xi}),t) d\boldsymbol{\xi} \\ &\quad + \beta \int_{\boldsymbol{\xi}\cdot\boldsymbol{n}>0} \boldsymbol{\xi} f_W^{eq}(\boldsymbol{x},\boldsymbol{\xi},t) \frac{\int_{\boldsymbol{\xi}'\cdot\boldsymbol{n}<0} |\boldsymbol{\xi}\cdot\boldsymbol{n}|\, f(\boldsymbol{x},\boldsymbol{\xi}',t) d\boldsymbol{\xi}}{\int_{\boldsymbol{\xi}'\cdot\boldsymbol{n}<0} |\boldsymbol{\xi}'\cdot\boldsymbol{n}|\, f_W^{eq}(\boldsymbol{x},\boldsymbol{\xi}',t) d\boldsymbol{\xi}'} d\boldsymbol{\xi}. \end{aligned} \tag{4.13}$$

Et comme nous savons que la composante normale est nulle (cf. Eq. (4.3)), la vitesse tangentielle à la paroi sera simplement la norme de $u_w = |\boldsymbol{u}_W|$. Cependant, cette équation est difficile à résoudre, car elle nécessite la connaissance de la fonction de distribution incidente. Pour pallier ce problème nous allons chercher à exprimer la vitesse tangentielle sur la paroi en fonction de la vitesse tangentielle dans le fluide, à une distance de la surface de l'ordre du libre parcours moyen λ.

Dans cette couche d'une épaisseur d'un libre parcours moyen λ, aucune collision n'a lieu, les particules y volent donc librement, en gardant leur vitesse. Nous pouvons donc exprimer la vitesse tangentielle sur la paroi u_W comme la moyenne des vitesses tangentielles des particules incidentes depuis cette distance et celle des particules réémises à cette même distance (voir illustration Fig. 4.2), soit :

$$u_W = \frac{1}{2}\left[u(\boldsymbol{x}_W + \lambda\boldsymbol{n}) + (1-\beta)\, u(\boldsymbol{x}_W + \lambda\boldsymbol{n}) + \beta\, U_W\right], \tag{4.14}$$

où U_w est la composante tangentielle de la vitesse de la paroi $\boldsymbol{U}_W$.

Finalement, développer $u(\boldsymbol{x})$ selon un développement de Taylor dans la direction normale à la paroi :

$$u(\boldsymbol{x}_W + \lambda\boldsymbol{n}) = u_W + \lambda\boldsymbol{n}\, \left.\frac{\partial u}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_W} + \frac{1}{2}\lambda^2 \boldsymbol{n}\boldsymbol{n} : \left.\frac{\partial^2 u}{\partial \boldsymbol{x}\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_W} + \ldots, \tag{4.15}$$

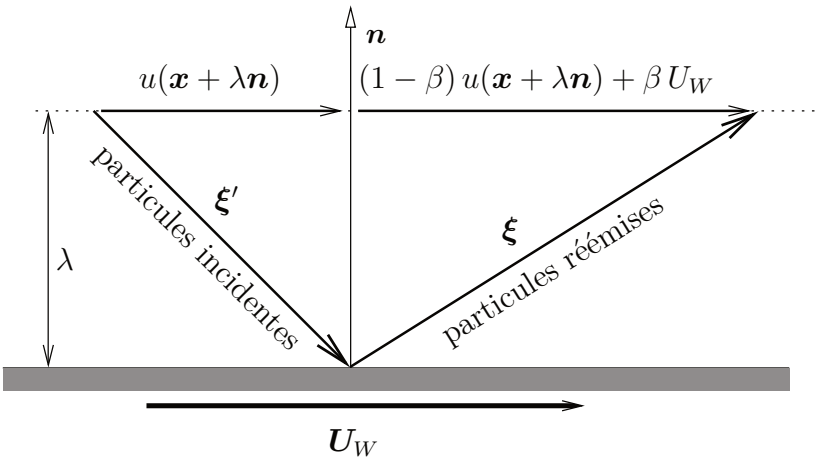


FIGURE 4.2 – Calcul de la vitesse du fluide sur la paroi. Les particules volent librement à l'intérieur de la couche d'épaisseur λ, leur vitesse n'est donc pas affectée à l'intérieur de celle-ci. La vitesse des particules réémises est $(1-\beta)$ fois la vitesse des particules incidentes réfléchies, dont la vitesse tangentielle n'a pas changé plus β fois la vitesse des particules diffusées, dont la vitesse tangentielle est celle de la paroi.

nous permet de résoudre l'équation pour u_W, qui nous donne :

$$u_W - U_W = \frac{2-\beta}{\beta}\left[\lambda\boldsymbol{n}\,\frac{\partial u}{\partial \boldsymbol{x}}\bigg|_{\boldsymbol{x}_W} + \frac{1}{2}\lambda^2\boldsymbol{n}\boldsymbol{n} : \frac{\partial^2 u}{\partial \boldsymbol{x}\partial\boldsymbol{x}}\bigg|_{\boldsymbol{x}_W} + \dots\right]. \tag{4.16}$$

Rappelons que la vitesse U_W est la vitesse de la paroi et que u_W est celle du fluide sur celle-ci. Le membre de droite de l'équation (4.16) étant *a priori* non nul, il apparaît que, contrairement à l'hypothèse classique de l'hydrodynamique, la vitesse du fluide u_W n'est pas égale à celle de la paroi U_W. Le fluide semble glisser et nous appellerons *vitesse de glissement*, notée u_{slip}, la différence entre ces deux vitesses.

Cette vitesse de glissement s'exprime sous une forme plus commune :

$$\hat{u}_{\text{slip}} = \frac{2-\beta}{\beta}\left[K_n\boldsymbol{n}\,\frac{\partial \hat{u}}{\partial \hat{\boldsymbol{x}}}\bigg|_{\hat{\boldsymbol{x}}_W} + \frac{1}{2}K_n^2\boldsymbol{n}\boldsymbol{n} : \frac{\partial^2 \hat{u}}{\partial \hat{\boldsymbol{x}}\partial\hat{\boldsymbol{x}}}\bigg|_{\hat{\boldsymbol{x}}_W} + \dots\right], \tag{4.17}$$

où la variable d'espace $\hat{\boldsymbol{x}} = \boldsymbol{x}/L$ et le temps $\hat{t} = t/\tau_M$ ont été redimensionnés respectivement pas la longueur L et le temps τ_M, caractéristiques de l'échelle macroscopique/hydrodynamique. Par conséquent les vitesses adimensionnées sont telles que $\hat{u} = u\,\tau_M/L$.

L'équation (4.17) montre que cette vitesse de glissement dépend des propriétés de la surface solide, via le coefficient d'accommodation β, et du nombre de Knudsen $K_n = \lambda/L$. En particulier, pour une paroi totalement accommodante ($\beta = 1$), donc des collisions aux bords totalement diffuses, on remarque que la vitesse de glissement s'annule lorsque $K_n \to 0$, correspondant bien à la limite hydrodynamique.

L'hypothèse de non-glissement utilisée pour la résolution des équations de Navier-Stokes est donc bien retrouvée ici. Néanmoins, pour des valeurs finies du nombre de Knudsen, typiquement pour $K_n \gtrsim 0{,}1$, ces effets de glissement ne pourront plus être négligés. Finalement, dans la limite d'une paroi totalement réfléchissante ($\beta \to 0$), la vitesse de glissement diverge, ce qui signifie que le fluide glisse librement sur la surface solide, sa vitesse y est donc arbitraire.

Par la suite, nous ne considérerons plus que des surfaces totalement diffuses, soit $\beta = 1$. Ce choix, pouvant paraître restrictif, est motivé par le fait que les mesures expérimentales faites de β sont toujours comprises entre 0,8 et 1. Les mesures de β, sont de plus très imprécises, car difficiles à réaliser.

En outre, l'expression de la vitesse de glissement (4.17) a été obtenue en faisant l'approximation de l'absence de collision au sein d'une couche d'épaisseur λ, ce qui n'est pas exact. Nous introduirons donc des coefficients, notés C_i, devant chaque terme du développement. Ces coefficients sont obtenus, soit par la mesure, soit par des calculs plus sophistiqués que ceux présentés ci-avant. L'expression de la vitesse de glissement que nous utiliserons par la suite sera donc, en se limitant à l'ordre deux :

$$u_{\text{slip}} = C_1 \, \lambda \, \boldsymbol{n} \left. \frac{\partial u}{\partial \boldsymbol{x}} \right|_{\boldsymbol{x}_W} + C_2 \, \lambda^2 \, \boldsymbol{nn} : \left. \frac{\partial^2 u}{\partial \boldsymbol{x} \partial \boldsymbol{x}} \right|_{\boldsymbol{x}_W}, \tag{4.18}$$

ou encore, sous forme adimensionnée :

$$\hat{u}_{\text{slip}} = C_1 \, K_n \boldsymbol{n} \left. \frac{\partial \hat{u}}{\partial \hat{\boldsymbol{x}}} \right|_{\hat{\boldsymbol{x}}_W} + C_2 \, K_n^2 \, \boldsymbol{nn} : \left. \frac{\partial^2 \hat{u}}{\partial \hat{\boldsymbol{x}} \partial \hat{\boldsymbol{x}}} \right|_{\hat{\boldsymbol{x}}_W}. \tag{4.19}$$

Signalons enfin que cette expression de la vitesse de glissement ne sera valide, étant donné l'ordre de son développement, que pour des nombres de Knudsen modérés, jamais supérieurs à 1. Nous choisirons d'imposer [1] $C_1 = 1$ et obtiendrons $C_2 = 0{,}13$ (cf. section 4.3.2), par ajustement avec des mesures expérimentales de débits d'un écoulement de Poiseuille plan, pour différents gaz, à différents Knudsen (cf. Fig. 4.5).

Nous vérifierons que la méthode de Boltzmann sur réseau développée dans cette thèse, supporte bien ce phénomène par l'étude d'un écoulement de Poiseuille plan (section 6.3).

4.3 Écoulement de Poiseuille avec glissement

Nous regardons ici l'effet du glissement sur les parois dans le cas d'un écoulement de Poiseuille plan, mis en mouvement par un champ de force extérieur $\boldsymbol{F} = \rho \boldsymbol{g}$ orienté parallèlement aux parois, selon les x positifs. Le problème est représenté

1. La valeur prise par le coefficient de glissement au premier ordre est généralement $C_1 = 1$, à l'exception de Cercignani qui donne $C_1 = 1{,}1466$. Le coefficient à l'ordre deux est quant à lui très différent selon les auteurs, en étant parfois même négatif. Un récapitulatif de ces différents coefficients est donné dans l'article de S. Colin et L. Baldas [CB04].

figure 4.3. Les calculs qui suivent s'appuient sur les équations de l'hydrodynamique, auxquelles nous allons apporter une première correction en K_n par l'incorporation des conditions de glissement aux parois considérées précédemment.

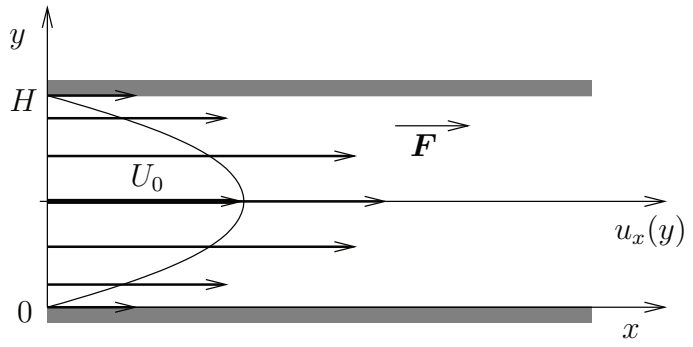


FIGURE 4.3 – Écoulement de Poiseuille plan. La vitesse U_0 est la vitesse maximale du champ de vitesse en régime hydrodynamique ($K_n \to 0$), représenté en ligne continue. Le champ de vitesse avec glissement est représenté par des flèches.

4.3.1 Profil de vitesse

En régime stationnaire, les équations de Navier-Stokes nous donnent pour le champ de vitesse :

$$\frac{d^2 u_x(y)}{dy^2} = \frac{g_x}{\nu}, \tag{4.20}$$

où ν est la viscosité cinématique du fluide et $g_x \equiv -g$ l'accélération extérieure.

Considérons alors les conditions aux limites de glissement données par l'équation (4.18) en $y = 0$ et $y = H$:

$$u_x(0) = C_1\,\lambda\,\partial_y u_x(0) - C_2\,\lambda^2 \partial_y^2 u_x(0), \tag{4.21}$$

$$u_x(H) = -C_1\,\lambda\,\partial_y u_x(H) - C_2\,\lambda^2 \partial_y^2 u_x(H). \tag{4.22}$$

Intégrons maintenant l'équation (4.20) :

$$\frac{du_x}{dy}(y) = \frac{g_x}{\nu}\,y + K_1, \tag{4.23}$$

$$u_x(y) = \frac{g_x}{2\nu}\,y^2 + K_1\,y + K_2. \tag{4.24}$$

Puis déterminons les constantes d'intégration K_1 et K_2 grâce aux conditions aux

limites. En $y = 0$:

$$u_x(0) = K_2 \tag{4.25}$$

$$= C_1\,\lambda\,\overbrace{\partial_y u_x(0)}^{(4.23)\Rightarrow K_1} - C_2\,\lambda^2\,\overbrace{\partial_y^2 u_x(0)}^{(4.20)\Rightarrow \frac{g_x}{\nu}} \tag{4.26}$$

$$= C_1\,\lambda\,K_1 - C_2\,\lambda^2\,\frac{g_x}{\nu}. \tag{4.27}$$

En $y = H$:

$$u_x(H) = \frac{g_x}{2\nu}\,H^2 + K_1\,H + K_2 \tag{4.28}$$

$$= -C_1\,\lambda\,\overbrace{\partial_y u_x(H)}^{(4.23)\Rightarrow \frac{g_x}{\nu}H+K_1} - C_2\,\lambda^2\,\overbrace{\partial_y^2 u_x(H)}^{(4.20)\Rightarrow \frac{g_x}{\nu}} \tag{4.29}$$

$$= -C_1\,\lambda\,\left(\frac{g_x}{\nu}H + K_1\right) - C_2\,\lambda^2\frac{g_x}{\nu}, \tag{4.30}$$

d'où :

$$\begin{cases} C_1\,\lambda\,K_1 - K_2 & = C_2\,\lambda^2\frac{g_x}{\nu}, \\ (H + C_1\,\lambda)\,K_1 + K_2 & = -C_2\,\lambda^2\frac{g_x}{\nu} - C_1\,\lambda\,\frac{g_x}{\nu}H - \frac{g_x}{2\nu}\,H^2. \end{cases}$$

Soit :

$$K_1 = -\frac{g_x}{2\nu}\,H, \tag{4.31}$$

$$K_2 = -\frac{g_x}{2\nu}\,H\,C_1\,\lambda - \frac{g_x}{\nu}\,H\,C_2\,\lambda^2. \tag{4.32}$$

La solution de l'équation (4.20) avec condition de glissement au second ordre s'écrit donc :

$$u_x(y) = \frac{g}{2\nu}\,H^2\left(\frac{y}{H} - \frac{y^2}{H^2} + C_1\frac{\lambda}{H} + 2\,C_2\frac{\lambda^2}{H^2}\right). \tag{4.33}$$

Il apparaît ici une dépendance au nombre de Knudsen $K_n = \lambda/H$ via les termes coefficientés en C_1 et C_2.

Isolons ces termes dans l'expression précédente, il vient :

$$u_x(y) = 4\,U_0\left(\frac{y}{H} - \frac{y^2}{H^2}\right) + 4\,U_0\left(C_1 K_n + 2\,C_2 K_n^2\right), \tag{4.34}$$

où l'on a posé $U_0 \equiv u_x(H/2) = \frac{g}{8\nu}\,H^2$, la vitesse maximale de l'écoulement en régime hydrodynamique. Dans le membre de droite de cette expression, le premier terme représente le champ de vitesse d'un écoulement de Poiseuille en régime hydrodynamique, c'est une parabole d'amplitude U_0 et dont la valeur s'annule sur les parois en 0 et en H. Le second terme correspond quant à lui à un profil de vitesse plat,

dont l'amplitude dépend de U_0 et du nombre de Knudsen ; il représente l'effet du glissement. On remarquera que dans la limite hydrodynamique ($K_n \to 0$), cet effet s'annule, comme attendu.

4.3.2 Vitesse débitante

Regardons maintenant la vitesse débitante (le débit), à travers une section de l'écoulement de Poiseuille, en fonction de la valeur du nombre de Knudsen. Par définition, le débit, noté Q, est la valeur moyenne de la vitesse sur une section du canal, soit :

$$Q = \frac{1}{H}\int_0^H u_x(y)dy. \tag{4.35}$$

En utilisant l'expression de $u_x(y)$ (Eq. 4.34), obtenue précédemment, nous obtenons :

$$Q = 4\,U_0\left(\frac{1}{6} + C_1 K_n + 2\,C_2 K_n^2\right). \tag{4.36}$$

Il est alors d'usage [Cer94, AKFB06] de normaliser ce débit Q par $4\,U_0\,K_n$. Le débit normalisé sera noté Q_0 et son expression sera :

$$Q_0 \equiv \frac{Q}{4\,U_0\,K_n} = \frac{1}{6\,K_n} + C_1 + 2\,C_2 K_n, \tag{4.37}$$

et ne sera valide que pour des nombres de Knudsen modérés $K_n \lesssim 1$. Dans l'expression de Q_0, le terme $1/(6K_n)$, correspondant au débit de l'écoulement de Poiseuille dans la limite hydrodynamique, *i.e.* sans glissement.

Nous proposons ici, pour déterminer les coefficients C_1 et C_2, d'imposer $C_1 = 1$, valeur communément admise, et d'évaluer C_2 par ajustement avec les valeurs expérimentales mesurées par Dong [Don56] (cf. Table 4.1). Étant donné l'ordre du développement, l'ajustement ne sera fait que pour $K_n < 1$ et obtenons ainsi $C_2 = 0{,}13$. La courbe de Q_0 pour ces valeurs de C_1 et de C_2, représenté figure 4.5, montre un bon accord de cette régression avec ces mesures.

Dans la limite de particules libres $K_n \to \infty$, les travaux de Cercignani [Cer94] montrent que ce débit normalisé croît comme le logarithme de K_n, nous noterons :

$$Q_\infty \sim \frac{1}{\sqrt{\pi}} \log K_n. \tag{4.38}$$

Contrairement à ce que l'on pourrait attendre, la vitesse débitante normalisée par $4\,U_0\,K_n$ n'est donc pas une fonction strictement décroissante de K_n. Elle possède en effet un minimum pour $K_n \sim 1$, comme le montre la représentation de la figure 4.4, sur laquelle les différents cas ont été tracés, ainsi que des valeurs expérimentales.

Les résultats représentés figure 4.4 seront utilisés dans le chapitre 6 (section 6.3) pour valider la méthode développée dans cette thèse.

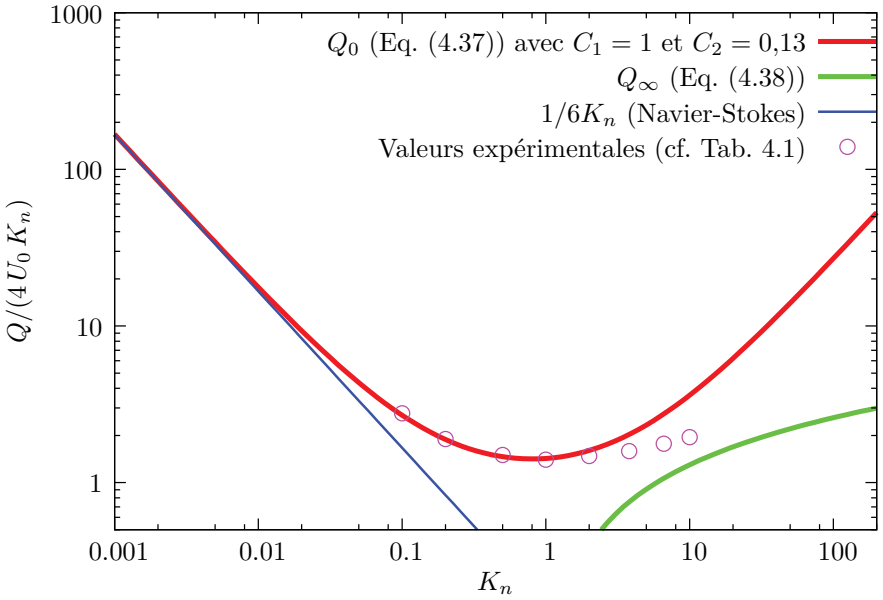


FIGURE 4.4 – Représentation du débit normalisé en fonction du nombre de Knudsen. Le fait que cette courbe de débit présente un minimum est contre-intuitif en ce sens qu'à une augmentation de $K_n = \lambda/H$ (à λ fixé) correspond une diminution de la distance H entre les plaques. On nomme ce phénomène le paradoxe de Knudsen.

K_n	0,1	0,2	0,5	1,0	2,0	3,8	6,6	10
$Q/(4\,U_0\,K_n)$	2,77	1,9	1,5	1,4	1,48	1,59	1,77	1,95

TABLE 4.1 – Débits expérimentaux $Q/(4\,U_0\,K_n)$. Valeurs prises dans le nuage de points de la figure 4.5, issue de la thèse de Dong [Don56].

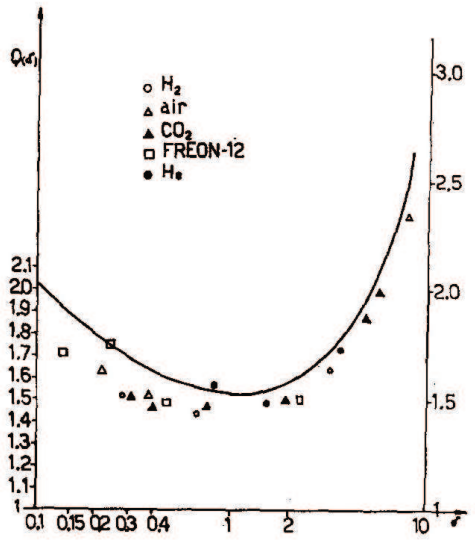


FIGURE 4.5 – Débits expérimentaux par Dong [Don56] (1956), recueillis dans Cercignani [Cer94]. Les abscisses sont $\delta = (K_n)^{-1}$ et les ordonnés le débit normalisé $Q/(4\,U_0\,K_n)$. On remarque que les résultats ne dépendent que très peu du gaz en écoulement.

4.4 Saut de température et transpiration thermique

Dans le cas anisotherme, d'autres phénomènes peuvent apparaître sur les parois. Il ne seront pas dérivés comme la vitesse de glissement vue précédemment, mais leurs expressions seront commentées.

4.4.1 Saut de température

De la même manière que précédemment pour la vitesse, définissons un coefficient d'accommodation, mais cette fois pour la température. Il sera noté β_T.

Alors la température du fluide sur la paroi $\hat{T}(\boldsymbol{x}_W)$ sera donnée, sous forme adimensionnée, par :

$$\hat{T}(\boldsymbol{x}_W) - \hat{T}_W = \frac{2-\beta_T}{\beta_T}\frac{2\gamma}{\gamma+1}\frac{K_n}{P_r}\boldsymbol{n}\left.\frac{\partial \hat{T}}{\partial \hat{\boldsymbol{x}}}\right|_{\hat{\boldsymbol{x}}_W}, \tag{4.39}$$

où γ est l'indice adiabatique et $P_r = \nu/\alpha$ est le nombre de Prandtl. L'équation (4.39) indique que le saut de température à la paroi [2], dépend, entre autres, du gradient de température dans la direction normale à celle-ci. Nous noterons $\hat{T}_{jump} = \hat{T}(\boldsymbol{x}_W) - \hat{T}_W$ ce saut de température.

Par simplicité, de même que précédemment pour la vitesse, nous considérons que le coefficient d'accommodation en température (ou en énergie) est égal à l'unité ($\beta_T = 1$). D'autre part, pour les gaz parfaits (cf. annexe B.2), nous savons qu'en dimension D, l'indice adiabatique est $\gamma = (D+2)/D$, nous pouvons donc écrire :

$$\hat{T}_{jump}(\boldsymbol{x}_W) = \frac{D+2}{D+1}\frac{K_n}{P_r}\boldsymbol{n}\left.\frac{\partial \hat{T}}{\partial \hat{\boldsymbol{x}}}\right|_{\hat{\boldsymbol{x}}_W}. \tag{4.40}$$

Nous vérifierons au chapitre 6 (section 6.2.2) que la méthode de Boltzmann sur réseau développée dans cette thèse supporte bien ce phénomène.

4.4.2 Transpiration thermique

Dans le cas où un gradient de température tangentiel à la paroi est présent, alors la vitesse de glissement $\hat{u}_W$ au premier ordre en $\hat{u}$ et en $\hat{T}$, s'exprime de la façon suivante :

$$\hat{u}_{\text{slip}} = \hat{u}_W - \hat{U}_W = \frac{2-\beta}{\beta}K_n\boldsymbol{n}\left.\frac{\partial \hat{u}}{\partial \hat{\boldsymbol{x}}}\right|_{\hat{\boldsymbol{x}}_W} + \frac{3}{2\pi}\frac{\gamma-1}{\gamma}\frac{M_a K_n}{E_c}\boldsymbol{t}\left.\frac{\partial \hat{T}}{\partial \hat{\boldsymbol{x}}}\right|_{\hat{\boldsymbol{x}}_W}, \tag{4.41}$$

2. Pour plus de détails voir par exemple le livre de Kogan [MN69].

où $\boldsymbol{t}$ est le vecteur tangent à la surface, M_a le nombre de Mach et E_c le nombre d'Eckert [3], défini par :

$$E_c = \frac{u_0^2}{C_p \, \Delta T}, \tag{4.42}$$

avec C_p la chaleur spécifique et ΔT le gradient de température.

Cette équation met en lumière un phénomène particulier. Supposons que le fluide soit immobile et délimité par une paroi, elle-même immobile et dont le champ de température présente un gradient tangentiel. Dans ce cas, l'équation précédente s'écrit :

$$\hat{u}_{\text{slip}} = \hat{u}_W = \frac{3}{2\pi} \frac{\gamma - 1}{\gamma} \frac{M_a \, K_n}{E_c} \boldsymbol{t} \left. \frac{\partial \hat{T}}{\partial \hat{\boldsymbol{x}}} \right|_{\hat{\boldsymbol{x}}_W}, \tag{4.43}$$

qui signifie que le fluide est mis en mouvement, dans la direction du gradient thermique, soit du froid vers le chaud. Ce phénomène de rampement (appelé "thermal transpiration" ou "thermal creep" en langue anglaise) dépend du nombre de Knudsen, et tend à disparaître dans la limite hydrodynamique. Il est exploité pour concevoir des pompes sans mouvement, aussi nommées compresseur de Knudsen [GG08, CB04, TZG+09]. Ce phénomène peut être simulé grâce à la méthode de Boltzmann sur réseau développée durant cette thèse. Cependant, nous ne connaissons pas la validité quantitative des résultats obtenus [4].

La simulation d'écoulements confinés mis en mouvement par ce processus fait l'objet de travaux, notamment par Mieussens et *al.* grâce à une méthode de résolution de l'équation de Boltzmann-BGK dite "à vitesses discrètes" [Mie00]. Cette méthode est différente de celle développée ici, le maillage de l'espace des vitesses y est régulier, la quadratique des moments est faite par une méthode de Newton-Cotes (et non Gauss-Hermite). La fonction de distribution d'équilibre y est de plus déterminée par application du théorème H de Boltzmann, par un schéma d'optimisation sous contrainte.

Signalons enfin que dans le cas où une paroi présente une courbure, tout ce qui a été présenté dans ce chapitre se complique [5] et nous ne vérifierons numériquement ces propriétés que dans le cas de parois planes.

3. Ce nombre permet de quantifier la dissipation d'énergie liée au frottement.

4. En particulier, nous ne savons pas évaluer le nombre d'Eckert. L'article de H. Xue et *al.* [XJS01] donne des indications utiles à ce sujet pour un micro-écoulement de Couette thermique, résolu par les équations de Burnett.

5. Voir par exemple Lockerby et *al.* [LREB04].

5 L'équation de Boltzmann sur réseau

Ce chapitre constitue le coeur de cette thèse. Il porte sur le développement d'une méthode de résolution de l'équation Boltzmann-BGK, que l'on souhaite adaptée à la résolution d'écoulements dans des régimes pouvant aller de l'hydrodynamique jusqu'au régime transitionnel. Ces différents régimes pouvant cohabiter aux sein du même domaine d'intégration.

Pour répondre à ce problème, nous allons procéder à une extension particulière des LBM, que l'on sait être valide dans le régime hydrodynamique, de sorte à les étendre jusqu'au régime transitionnel. La démarche que nous allons suivre est celle proposée par Shan et He dans l'article [SH98]. Il est montré dans cet article que les LBM peuvent se construire à partir de l'équation de Boltzmann-BGK, en développement d'une part la fonction de distribution sur la base des polynômes d'Hermite, comme l'a fait Grad [Gra49b] dans sa théorie des moments et d'autre part, en calculant les moments de cette fonction de distribution par une quadrature de Gauss-Hermite. Il est montré en particulier que les LBM classiques D_2Q_9 et D_3Q_{27} s'obtiennent en développant la fonction de distribution d'équilibre à l'ordre deux sur la base des polynômes d'Hermite et en calculant ses moments grâce à une quadrature de Gauss-Hermite à trois points par direction ; quadrature que nous noterons par la suite $\mathcal{H}_3$. Comme nous le verrons, l'ordre de la quadrature détermine, dans ce contexte, le moment d'ordre le plus élevé de la fonction de distribution qu'il est possible de calculer. Par exemple, une quadrature à trois points par direction ne permettra pas de calculer le flux de chaleur, ce qui explique que les LBM classiques ne supportent pas la thermique. Parti de ce constat, il est alors tentant de chercher à augmenter l'ordre de la quadrature, de façon à représenter (rendre calculable) les moments de la fonction de distribution d'ordres élevés, dont l'importance grandit avec le degré de raréfaction/confinement d'un écoulement.

Avant de présenter cette démarche particulière, commençons par faire un bref état de l'art concernant les différentes voies explorées pour étendre les LBM aux régimes non-hydrodynamiques.

5.1 Schémas de Boltzmann sur réseau pour les écoulements non-hydrodynamiques

Les *méthodes de Boltzmann sur réseau* (LBM) sont une classe de méthodes de dynamique des fluides sur ordinateur (CFD), bâties sur la base de l'*équation de*

Boltzmann-BGK sur réseau (LBE), à laquelle a été ajouté le support de phénomènes physiques [1] non régis initialement par l'équation de Boltzmann-BGK. L'équation de Boltzmann-BGK représente en effet, par sa définition, un gaz parfait dilué. La construction des LBM commence par supposer que l'équation de Boltzmann-BGK n'est pas valide uniquement pour les gaz parfaits dilués, mais aussi pour les liquides. L'objectif entendu des LBM n'est pas de simuler des liquides avec l'équation de Boltzmann-BGK, ce qui n'aurait pas de sens, mais de construire des schémas de résolution des équations de Navier-Stokes [2]. Cette vision "solver Navier-Stokes" de l'équation de Boltzmann-BGK, a amené les LBM à s'éloigner de ses fondements cinétiques. Beaucoup de choses couramment utilisées dans ces méthodes, à l'instar des conditions aux bords que l'on nomme "Bounce-Back" (retour en arrière), n'ont aucun sens dans le cadre de la théorie cinétique. Ces conditions aux bords sont néanmoins très efficace pour imposer la condition de *non-glissement* sur les parois qui délimitent le fluide, au-delà même du régime hydrodynamique, comme le montre la figure 5.1. On remarque notamment que ces conditions aux bords prolongent artificiellement le comportement hydrodynamique de l'équation de Boltzmann-BGK, jusqu'à des Knudsen de l'ordre de 0,1. En effet, pour $K_n \lesssim 0{,}1$, les débits obtenus avec le schéma D_2Q_9 correspondent à ceux donnés par les équations de Navier-Stokes sans glissement.

On remarquera que l'utilisation de conditions aux bords cinétiques, avec un schéma de Boltzmann sur réseau standard, permet de rendre compte du débit avec glissement pour $K_n \lesssim 0{,}1$.

On montre par ce simple exemple que les LBM, dans leur formulation classique, sont faites pour l'hydrodynamique et spécialement pour cela. On montre d'autre part que l'emploi des conditions aux bords cinétiques permet de rendre compte, dans une certaine mesure ($K_n \lesssim 0{,}1$), d'un effet de raréfaction : le glissement. Cependant, l'amélioration apportée par ce choix de conditions aux bords n'est pas suffisante pour étendre ces méthodes jusqu'au régime transitionnel.

Plusieurs démarches d'extension des LBM vers le régime transitionnel sont entreprises à ce jours. Toutes les méthodes développées en ce sens n'abandonnent pas les

1. Les LBM constituent maintenant une alternative viable et compétitive aux techniques classiques de CFD. Elles offrent la possibilité de simuler, par exemple, des écoulements turbulents, des écoulements multiphasiques, les phénomènes capillaires, entre autres. Il faut noter néanmoins qu'aucune des LBM ne supporte la thermique de manière intrinsèque, comme doit le permettre l'équation de Boltzmann-BGK. Les LBM anisothermes sont construites en couplant l'équation de Boltzmann-BGK sur réseau (isotherme) et l'équation de la chaleur.

2. L'analogie entre gaz et liquides peut sembler abusive au premier abord, d'autant qu'il est courant avec les LBM, de modifier la densité du liquide volontairement, notamment pour imposer une valeur de pression sur un bord, comme cela est fait avec les conditions aux bords de Zou et He [QX97]. En fait, cette démarche est acceptable dans le cadre "limité" d'application des LBM. Ces méthodes s'appliquent en effet à des écoulements isothermes, pour lesquels la pression du liquide, donnée par la loi des gaz $p = \rho\theta_0$, ne dépend que de la densité ρ. La résolution de l'équation de Boltzmann-BGK, pour la fonction de distribution, permet de résoudre simultanément les équations pour la pression (densité) et pour l'impulsion, contrairement aux méthodes classiques de résolution des équations de Navier-Stokes, pour lesquelles ces champs sont résolus séparément avant d'être couplés.

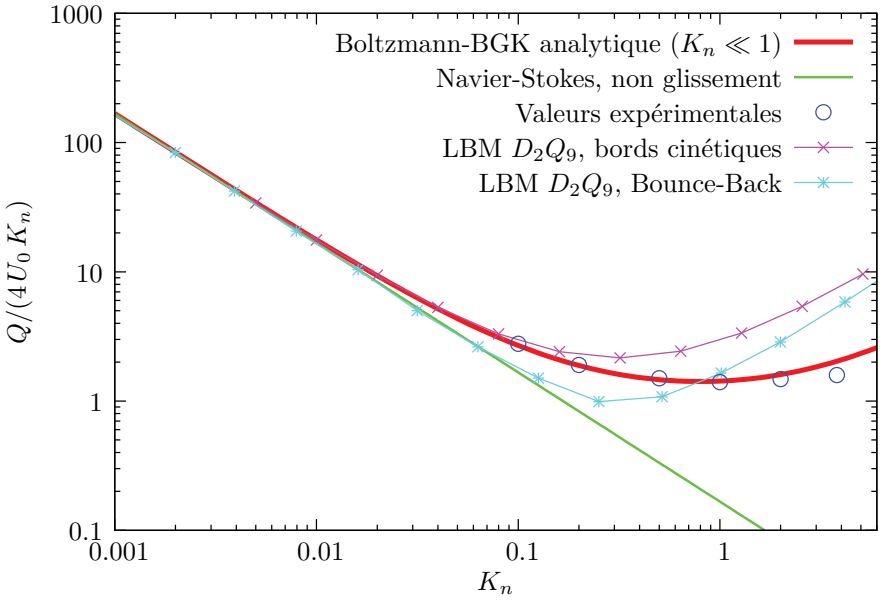


FIGURE 5.1 – Débits de l'écoulement de Poiseuille en fonction du Knudsen selon différente approches : expérimentale, analytique et numérique. Pour plus de détails sur les courbes de débits représentées, le lecteur pourra se reporter section 4.3 page 49.

conditions aux bords de type Bounce-Back, mais les composent avec des conditions aux bords réflectives, de sorte à construire des conditions similaires à celles de Maxwell, (Voir par exemple Tang et *al.* [TTH05] où l'écoulement d'un gaz à travers un milieu poreux est étudié, le schéma utilisé est en outre le D_2Q_9).

Une seconde approche consiste à utiliser des conditions aux bords cinétiques, tout en gardant une discrétisation de l'espace des vitesses simple (typiquement D_2Q_9). La démarche consiste alors à tenter de limiter le glissement aux parois, surestimé aux grands Knudsen par le schéma. Une telle démarche est présentée dans l'article de Toschi et Succi [TS05]. Ces auteurs assimilent le trop fort glissement à un manque d'isotropie de l'espace des vitesses [3]. En particulier, les faisceaux dont la vitesse est parallèle aux bords (dans le cas de l'écoulement de Poiseuille), ne rencontrent jamais la paroi. Dans le cas où le temps de relaxation est grand, ces faisceaux accélèrent quasi librement, et produisent une surestimation du débit (cf. figure 5.1). Pour corriger ce défaut, qu'ils nomment "*run-away pathology*", Toschi et Succi proposent d'ajouter aux collisions entre particules (opérateur BGK), des collisions virtuelles avec les parois du canal. Ces collisions virtuelles introduisent un temps de relaxation effectif, qui dépend de la distance à la paroi et de la direction de propagation du faisceau considéré. D'autres démarches du même type font appel à des temps de relaxation effectifs [KPB08b, ZGBE06], ou encore à des méthodes dites de régularisation [NHMS07].

Signalons l'existence d'une autre méthode de résolution de l'équation de Boltzmann-

3. La fonction de distribution ne peut se propager que dans un nombre de directions trop restreint.

BGK, dite *méthode à vitesses discrètes* [Mie00]. Celle-ci diffère des LBM par l'emploi d'une quadrature de Newton-Cotes pour calculer les moments et par la recherche de la fonction d'équilibre par le théorème H de Boltzmann.

Finalement, la démarche proposée par Shan et He [SH98] se base sur le développement de la fonction de distribution sur les polynômes d'Hermite, associé à l'emploi de la quadrature de Gauss-Hermite, pour en calculer les moments. Nous verrons que la quadrature des moments induit la discrétisation de l'espace des vitesses, car les vitesses discrètes seront les racines du polynôme d'Hermite associé à la quadrature. Les racines de ces polynômes étant en général dans des rapport irrationnels, nous serons amené à utiliser un schéma de transport plus sophistiqué que celui des LBM [4]. Signalons finalement que des quadratures de Gauss-Hermite spéciales, dont les racines des polynômes associés peuvent épouser un maillage régulier de l'espace des configurations, ont été proposées par Shan et *al.* [SYC06] et Nie et *al.* [NSC08], mais nous n'utiliserons pas cette méthode, lui préférant l'application des quadratures selon leur définition première.

5.2 Schéma général de résolution de l'équation de Boltzmann-BGK

Reprenons l'équation de Boltzmann-BGK (5.67), en supposant dans un premier temps le terme de force extérieur nul :

$$\partial_t f + \boldsymbol{v} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} f = -\frac{1}{\tau}(f - f^{eq}), \tag{5.1}$$

où f^{eq} est la distribution de Maxwell-Boltzmann :

$$f^{eq} = \rho \left(\frac{m}{2\pi k_B T}\right)^{D/2} \exp\left(-\frac{m(\boldsymbol{v}-\boldsymbol{u})^2}{2k_B T}\right). \tag{5.2}$$

Pour résoudre numériquement l'équation de Boltzmann-BGK, il nous faut la discrétiser dans l'espace des configurations, des vitesses et selon le temps ; pratiquons cette discrétisation par étapes en commencent par le temps : $dt \to \delta t$, avec δt fini et constant. L'équation (5.67) devient alors :

$$\frac{\delta f}{\delta t} + \boldsymbol{v} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} f = -\frac{1}{\tau}(f - f^{eq}). \tag{5.3}$$

Cette équation se résout par *splitting*, c'est-à-dire en résolvant dans un premier

4. Ces schémas, pour lesquels certains des faisceaux ne retombent pas sur les nœuds du réseau, ont été nommés "*Multispeed off-Lattice Boltzmann Models*" par Bardow et Karlin [BKG08].

temps l'équation sans second membre :

$$\frac{\delta f}{\delta t} + \boldsymbol{v} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} f = 0, \tag{5.4}$$

dont la solution est :

$$f(\boldsymbol{x}, \boldsymbol{v}, t + \delta t) = f(\boldsymbol{x} - \boldsymbol{v}\,\delta t, \boldsymbol{v}, t), \tag{5.5}$$

correspondant à un transport pur à vitesse $\boldsymbol{v}$ de la fonction de distribution f dans l'espace des configurations. Puis dans un second temps, à pratiquer les collisions :

$$\frac{\delta f}{\delta t} = -\frac{1}{\tau}(f - f^{eq}), \tag{5.6}$$

en quelque sorte instantanément, à l'instant $t + \delta t$. Il faudra néanmoins considérer l'ensemble des collisions qui ont lieu pendant l'intervalle de temps δt, ce que nous ferons section 5.6.

L'algorithme 1 résume la démarche.

Algorithme 1: Algorithme fondamental de résolution de l'équation de Boltzmann. Le pas de temps est ici $\delta t = 1$. On remarquera que la fonction de distribution f est initialisée à l'équilibre f^{eq}, pour les champs macroscopiques initiaux $\rho_{t=0}$,$\boldsymbol{u}_{t=0}$,$\theta_{t=0}$. Nous n'avons en effet pas d'autre choix, car la fonction de distribution d'équilibre est la seule pour laquelle il y a une bijection entre les grandeurs macroscopiques et microscopiques. Elle constitue en quelque sorte un pivot entre ces deux échelles.

Initialisation :
$f^{eq}_{t=0} \leftarrow f^{eq}(\rho_{t=0},\boldsymbol{u}_{t=0},\theta_{t=0})$
$f_{t=0} \leftarrow f^{eq}_{t=0}$
Boucle principale en temps :
for $t \leftarrow 0$ **to** T_{final} **do**
  $f_{t+1} \leftarrow transport(f_t)$
  $[\rho_{t+1},\boldsymbol{u}_{t+1},\theta_{t+1}] \leftarrow quadrature(\boldsymbol{v}^{0,1,2} f_{t+1})$
  $f^{eq}_{t+1} \leftarrow f^{eq}(\rho_{t+1},\boldsymbol{u}_{t+1},\theta_{t+1})$
  $f^{*}_{t+1} \leftarrow collisions(f_{t+1},f^{eq}_{t+1})$
  $f_t \leftarrow f^{*}_{t+1}$: initialisation du pas de temps suivant
end

Nous sommes donc amené à résoudre deux problèmes numériques, que nous verrons être couplés. Le premier est un problème d'intégration numérique et siège dans l'étape de collisions. En effet, l'étape de collisions nécessite de calculer la fonction de distribution à l'équilibre f^{eq} (5.2), donc des trois premiers moments de la fonction de distribution f elle-même. Le calcul de ces moments se faisant par intégration de la fonction $\boldsymbol{v}^n f$ (pour $n = 0, 1, 2$) sur l'espace des vitesses, nous allons être amené à induire la discrétisation de l'espace des vitesses par le choix d'une méthode de discrétisation de ces intégrales, d'une *quadrature*[5]. Nous verrons que la quadrature

5. Les méthodes de quadratures permettent de transformer des sommes continues (des inté-

la plus adaptée à ce problème est celle de Gauss-Hermite. Cependant, son utilisation nous obligera, dès lors que la quadrature sera faite sur plus de trois points, à interpoler la fonction de distribution dans l'espace des configurations pour pouvoir pratiquer l'étape de transport. En outre, nous verrons que, pour les quadratures à trois points et moins, nous retombons sur les méthodes LB, pour lesquelles il n'est pas nécessaire d'interpoler la fonction de distribution entre les nœuds du réseau, ce qui est leur principal atout.

5.3 Projection de l'équation de Boltzmann-BGK sur la base des polynômes d'Hermite

Le contexte théorique utilisé ici est similaire à celui utilisé par H. Grad [Gra49b, Gra49a] dans sa théorie des moments. Nous ne rentrerons pas dans les détails de cette théorie, mais utiliserons seulement l'idée selon laquelle l'équation de Boltzmann peut être exprimée sous forme de la projection de ses solutions (les moments), sur la base fonctionnelle des polynômes d'Hermite. Ce nouveau cadre théorique semble donner actuellement une nouvelle impulsion au développement des LBM, ainsi qu'une assise mieux définie dans la théorie cinétique [SYC06]. De méthodes de résolution pour les équations Navier-Stokes, elles deviennent des méthodes cinétiques, susceptibles d'aller au-delà de l'hydrodynamique, des milieux continus.

Repartons de la distribution d'équilibre thermodynamique local (2.32) :

$$f^{eq}(\boldsymbol{x},\boldsymbol{v},t,\theta) = \frac{\rho(\boldsymbol{x},t)}{(2\pi\theta)^{D/2}} \exp\left(-\frac{(\boldsymbol{v}-\boldsymbol{u}(\boldsymbol{x},t))^2}{2\,\theta}\right). \tag{5.7}$$

En développant l'argument de l'exponentielle, faisons apparaître

$$w^{[\theta]}(\boldsymbol{v}) = 1/(2\pi\theta)^{D/2} \exp\left(-\boldsymbol{v}^2/2\theta\right)$$

en facteur dans cette expression, il vient :

$$f^{eq} = \rho\, w^{[\theta]}(\boldsymbol{v}) \exp\left(\frac{\boldsymbol{v}\cdot\boldsymbol{u}}{\theta} - \frac{\boldsymbol{u}^2}{2\theta}\right). \tag{5.8}$$

Expression dans laquelle on peut reconnaître la fonction génératrice des polynômes

grales) en sommes discrètes. Deux grandes familles de méthodes de quadratures existent : les méthodes dites de Newton-Cotes (méthode des rectangles, des trapèzes, ...) et les méthodes dites de Gauss. Les premières sont les plus couramment utilisées, car elles permettent d'intégrer un grand nombre de fonctions avec une bonne précision, les secondes sont, elles, dédiées à des fonctions particulières. La méthode de quadrature la plus adaptée à notre problème est celle de Gauss-Hermite, dont l'emploi est préconisé pour l'intégration des fonctions Gaussiennes (une brève description de cette méthode ainsi qu'un formulaire sont donnés dans l'annexe A.3.). Outre Les qualités numériques de cette quadrature, nous allons voir que son utilisation, et l'utilisation des polynômes d'Hermite qui en découle, s'accorde bien avec la théorie des moments de H. Grad.

d'Hermite généralisés [6] $\boldsymbol{\mathcal{H}}_n^{[\theta]}$ (cf. Annexe A.2) :

$$\exp\left(\frac{\boldsymbol{v}\cdot\boldsymbol{u}}{\theta}-\frac{\boldsymbol{u}^2}{2\theta}\right)=\sum_{n=0}^{\infty}\frac{1}{n!}\bigotimes^{n}\boldsymbol{u}:\boldsymbol{\mathcal{H}}_n^{[\theta]}(\boldsymbol{v}). \tag{5.9}$$

Réécrivons maintenant l'équation (5.8) en tenant compte de l'équation (5.9) :

$$f^{eq}=\,w^{[\theta]}(\boldsymbol{v})\sum_{n=0}^{\infty}\frac{1}{n!}\bigotimes^{n}\rho\boldsymbol{u}:\boldsymbol{\mathcal{H}}_n^{[\theta]}(\boldsymbol{v}). \tag{5.10}$$

Cette forme de la fonction de distribution de Maxwell-Boltzmann est totalement équivalente à celle de départ (5.7), à la différence qu'elle est maintenant projetée sur une base de fonctions orthogonales (les $\boldsymbol{\mathcal{H}}_n^{[\theta]}$), dont nous pouvons bénéficier des propriétés mathématiques. Dans cette expression, les coefficients du développement sont les $\bigotimes^n \rho\boldsymbol{u}$, et ne font intervenir que les deux premiers moments de f^{eq}, ρ et $\boldsymbol{u}$. Le troisième moment θ, est maintenant contenu dans la base des polynômes d'Hermite, ce qui fait apparaître une difficulté. En effet, en un point $\boldsymbol{x}$ de l'espace des configurations, où la température est θ, va arriver de l'information (fonction de distribution) venant d'autres points $\boldsymbol{x}'$, où la température est θ'. Il faudrait donc savoir passer d'une projection à une autre, *i.e.* exprimer une fonction de distribution projetée sur $\boldsymbol{\mathcal{H}}_n^{[\theta']}$, sur la base des $\boldsymbol{\mathcal{H}}_n^{[\theta]}$ avec $\theta\neq\theta'$. Nous n'explorerons pas cette voie et considérerons plutôt pour f^{eq} un développement de la forme :

$$f^{eq}(\boldsymbol{x},\boldsymbol{v},t,\theta)=\,w^{[\theta_0]}(\boldsymbol{v})\sum_{n=0}^{\infty}\frac{1}{n!}\,\boldsymbol{a}_n^{eq}(\boldsymbol{x},t,\theta)\boldsymbol{\mathcal{H}}_n^{[\theta_0]}(\boldsymbol{v}), \tag{5.11}$$

où les $\boldsymbol{a}_n^{eq}(\boldsymbol{x},t,\theta)$ sont les coefficients du développement de $f^{eq}(\boldsymbol{x},t,\theta)$, projetée sur la base des polynômes $\boldsymbol{\mathcal{H}}_n^{[\theta_0]}$, avec θ_0 la température de référence, qui est uniforme. Notez que la température de référence θ_0 (fixe et uniforme), n'est pas celle (θ) de la fonction de distribution, qui elle est libre de varier dans l'espace et dans le temps.

Ces coefficients peuvent alors être calculés grâce à la relation d'orthogonalité entre

6. La démarche qui suit est celle de Grad, à la différence que Grad renormalise le problème par rapport à la température de référence θ_0. Il prend notamment comme température de référence normalisée $\hat{\theta}_0=\theta_0/\theta_0=1$, ce qui induit une projection sur la base des polynômes d'Hermite standards, mais qui a le défaut d'éliminer un degré de liberté au problème. Or, nous verrons que cette température θ_0, est importante pour la construction du schéma numérique.

les polynômes d'Hermite (A.9), en effet :

$$\left\langle f^{eq}(\boldsymbol{x},\boldsymbol{v},t,\theta)\middle|\boldsymbol{\mathcal{H}}_n^{[\theta_0]}(\boldsymbol{v})\right\rangle = \left\langle w^{[\theta_0]}(\boldsymbol{v})\sum_{n=0}^{\infty}\frac{1}{n!}\boldsymbol{a}_n^{eq}(\boldsymbol{x},t)\boldsymbol{\mathcal{H}}_n^{[\theta_0]}(\boldsymbol{v})\middle|\boldsymbol{\mathcal{H}}_m^{[\theta_0]}(\boldsymbol{v})\right\rangle \tag{5.12}$$

$$= \sum_{n=0}^{\infty}\frac{\boldsymbol{a}_n^{eq}(\boldsymbol{x},t)}{n!}\int \boldsymbol{\mathcal{H}}_n^{[\theta_0]}(\boldsymbol{v})\boldsymbol{\mathcal{H}}_m^{[\theta_0]}(\boldsymbol{v})w^{[\theta_0]}(\boldsymbol{v})\,d\boldsymbol{v} \tag{5.13}$$

$$= \frac{\boldsymbol{a}_n^{eq}}{n!}\,\frac{n!}{\theta_0^n}\,\delta_{nm} \tag{5.14}$$

$$= \frac{\boldsymbol{a}_m^{eq}}{\theta_0^m}. \tag{5.15}$$

On a donc :

$$\boldsymbol{a}_m^{eq}(\boldsymbol{x},t,\theta) = \theta_0^m\left\langle f^{eq}(\boldsymbol{x},\boldsymbol{v},t,\theta)\middle|\boldsymbol{\mathcal{H}}_m^{[\theta_0]}(\boldsymbol{v})\right\rangle, \tag{5.16}$$

ou encore :

$$\boldsymbol{a}_m^{eq}(\boldsymbol{x},t,\theta) = \theta_0^m\int f^{eq}(\boldsymbol{x},\boldsymbol{v},t,\theta)\boldsymbol{\mathcal{H}}_m^{[\theta_0]}(\boldsymbol{v})d\boldsymbol{v}. \tag{5.17}$$

Dans le cas de la distribution de Maxwell-Boltzmann, les coefficients $\boldsymbol{a}_n^{eq}$ peuvent être calculés explicitement en utilisant les expressions des polynômes d'Hermite données en annexe A.2, pour $n = 0\ldots4$, ils sont [7] :

$$a_0^{eq} = \rho, \tag{5.18}$$

$$a_{1\,\alpha}^{eq} = \rho\,u_\alpha, \tag{5.19}$$

$$a_{2\,\alpha\beta}^{eq} = \rho\,u_\alpha\,u_\beta + \rho\,(\theta-\theta_0)\,\delta_{\alpha\beta}, \tag{5.20}$$

$$a_{3\,\alpha\beta\gamma}^{eq} = \rho\,u_\alpha\,u_\beta\,u_\gamma + \rho\,(\theta-\theta_0)(u_\alpha\,\delta_{\beta\,\gamma} + u_\beta\,\delta_{\alpha\,\gamma} + u_\gamma\delta_{\alpha\,\beta}), \tag{5.21}$$

$$\begin{aligned} a_{4\,\alpha\beta\gamma\delta}^{eq} = {} & \rho\,u_\alpha u_\beta u_\gamma u_\delta + \rho(\theta-\theta_0)^2(\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}) \\ & + \rho(\theta-\theta_0)(\delta_{\alpha\beta}u_\gamma u_\delta + \delta_{\alpha\gamma}u_\beta u_\delta + \delta_{\alpha\delta}u_\beta u_\gamma + \delta_{\beta\gamma}u_\alpha u_\delta \\ & + \delta_{\beta\delta}u_\alpha u_\gamma + \delta_{\gamma\delta}u_\alpha u_\beta). \end{aligned} \tag{5.22}$$

On remarque que la température θ a bien été réintroduite ici. Les $\boldsymbol{a}_n^{eq}$ sont de la forme : $\boldsymbol{a}_n^{eq} = \bigotimes^n \rho\boldsymbol{u} + \boldsymbol{G}_n(\rho,\boldsymbol{u},(\theta-\theta_0))$ où $\boldsymbol{G}_n$ est une fonction tensorielle d'ordre n qui représente l'écart à la température de référence θ_0. Par la suite, par simplicité d'écriture, nous omettrons de mentionner la dépendance des coefficients d'Hermite

7. Un exemple en une dimension :

$$\begin{aligned} a_2^{eq} &= \theta_0^2\int_{-\infty}^{\infty} f^{eq}\,\mathcal{H}_2^{[\theta_0]}(v)\,dv = \theta_0^2\int_{-\infty}^{\infty} f^{eq}\,\left(\frac{v^2}{\theta_0^2} - \frac{1}{\theta_0}\right)dv \\ &= \rho\,u^2 + \rho\,\theta - \rho\,\theta_0 = \rho\,u^2 + (\theta - c_0^2)\,\rho. \end{aligned}$$

à la température.

Penchons-nous sur la fonction de distribution f elle-même. Étant donné que le fluide n'est pas à l'équilibre thermodynamique (mais en est proche, *i.e.* $f \sim f^{eq}$), la fonction f n'a *a priori* pas les mêmes coefficients que f^{eq}, nous la développerons comme suit :

$$f(\boldsymbol{x},\boldsymbol{v},t) = w^{[\theta_0]}(\boldsymbol{v}) \sum_{n=0}^{\infty} \frac{1}{n!} \boldsymbol{a}_n(\boldsymbol{x},t) \boldsymbol{\mathcal{H}}_n^{[\theta_0]}(\boldsymbol{v}). \tag{5.23}$$

Dont les coefficients $\boldsymbol{a}_n(\boldsymbol{x},t)$ se calculent de la même manière que ceux de f^{eq}, avec :

$$\boldsymbol{a}_m(\boldsymbol{x},t,\theta) = \theta_0^m \int f(\boldsymbol{x},\boldsymbol{v},t,\theta) \boldsymbol{\mathcal{H}}_m^{[\theta_0]}(\boldsymbol{v}) d\boldsymbol{v}, \tag{5.24}$$

mais cette fois en utilisant l'expression des moments de f (Eqs. (2.13), (2.14) et (2.15)). Leurs expressions sont, pour $n = 0 \dots 4$:

$$\begin{aligned}
a_0 &= \rho, && (5.25)\\
a_{1\,\alpha} &= \rho\, u_\alpha, && (5.26)\\
a_{2\,\alpha\beta} &= P_{\alpha\beta} + \rho(u_\alpha\, u_\beta - \theta_0 \delta_{\alpha\beta}),\\
&= \Pi_{\alpha\beta} + \rho\, u_\alpha\, u_\beta + \rho\,(\theta - \theta_0)\,\delta_{\alpha\beta}, && (5.27)\\
a_{3\,\alpha\beta\gamma} &= Q_{\alpha\beta\gamma} + u_\alpha a_{2\,\beta\gamma} + u_\beta a_{2\,\alpha\gamma} + u_\gamma a_{2\,\alpha\beta} + (1-D)\,\rho\, u_\alpha\, u_\beta\, u_\gamma, && (5.28)\\
a_{4\,\alpha\beta\gamma\delta} &= R_{\alpha\beta\gamma\delta} - (P_{\alpha\beta}\delta_{\gamma\delta} + P_{\alpha\gamma}\delta_{\beta\delta} + P_{\alpha\delta}\delta_{\beta\gamma} + P_{\beta\gamma}\delta_{\alpha\delta} + P_{\beta\delta}\delta_{\alpha\gamma} + P_{\gamma\delta}\delta_{\alpha\beta})\\
&\quad + (\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}). && (5.29)
\end{aligned}$$

Où l'on rappelle que

$$\boldsymbol{P} = \int \boldsymbol{\xi}\boldsymbol{\xi}\, f\, d\boldsymbol{\xi}, \quad \boldsymbol{Q} = \int \boldsymbol{\xi}\boldsymbol{\xi}\boldsymbol{\xi}\, f\, d\boldsymbol{\xi}, \tag{5.30}$$

et où

$$\boldsymbol{R} = \int \boldsymbol{\xi}\boldsymbol{\xi}\boldsymbol{\xi}\boldsymbol{\xi}\, f\, d\boldsymbol{\xi}. \tag{5.31}$$

Il faut remarquer ici que les $\boldsymbol{a}_n$ d'ordre successifs font apparaître successivement les moments de f, tout comme les $\boldsymbol{a}_n^{eq}$ font apparaître successivement les moments de f^{eq}. En outre, un coefficient d'ordre n donné, ne contiendra que des quantités calculables aux rangs inférieurs. En d'autres termes, les $\boldsymbol{a}_n$ forment une hiérarchie, équivalente à celle des moments de f. Pour nous en convaincre, nous reprendrons dans ce chapitre (section 5.7) la procédure de la section 3.1 du chapitre 3, pour l'équation de Boltzmann-BGK sans force extérieure.

En outre, les deux premiers coefficients de f et f^{eq} sont égaux, ainsi que les traces de $\boldsymbol{a}_2$ et de $\boldsymbol{a}_2^{eq}$, correspondant aux contraintes d'invariance collisionnelles.

5.4 Quadrature et troncature de la fonction de distribution

Comme cela a été énoncé dans la section 5.2, nous devons être capable, pour appliquer les collisions, de calculer le fonction de distribution d'équilibre f^{eq}, donc les trois premiers moments de la fonction de distribution f elle-même. Dans le formalisme de la section précédente, cela revient à être capable de calculer les coefficients a_0, $a_{1\,\alpha}$ et les éléments de la trace de $\boldsymbol{a}_2$, les $a_{2\,\alpha\alpha}$.

Ainsi, nous aurons :

$$\rho = a_0, \tag{5.32}$$

$$u_\alpha = \frac{a_{1\,\alpha}}{a_0} = \frac{a_{1\,\alpha}}{\rho}, \tag{5.33}$$

$$\begin{aligned}\theta &= \frac{a_{2\,\alpha\alpha}}{a_0} - \left(\frac{a_{1\,\alpha}}{a_0}\right)^2 + \theta_0 \\ &= \frac{a_{2\,\alpha\alpha}}{\rho} - u_\alpha u_\alpha + \theta_0,\end{aligned} \tag{5.34}$$

et nous pourrons alors calculer tous les coefficients $\boldsymbol{a}_n^{eq}(\rho,\boldsymbol{u},\theta)$, donc f^{eq}. Mais laissons pour l'instant de côté la fonction de distribution d'équilibre et penchons-nous sur la fonction de distribution f.

Les coefficients $\boldsymbol{a}_n$ de f étant calculés par une intégrale, la relation d'orthogonalité (5.24) et étant donné que f possède la gaussienne $w^{[\theta_0]}(\boldsymbol{v})$ en facteur, il semble naturel d'utiliser la quadrature de Gauss-Hermite pour les calculer.

Ainsi pour une quadrature à q points par direction, soit q^D points en dimension D, nous aurons :

$$\begin{aligned}\boldsymbol{a}_m(\boldsymbol{x},t) &= \theta_0^m \int f(\boldsymbol{x},\boldsymbol{v},t)\, \boldsymbol{\mathcal{H}}_m^{[\theta_0]}(\boldsymbol{v}) d\boldsymbol{v} \\ &= \theta_0^m \int w^{[\theta_0]}(\boldsymbol{v}) \frac{f(\boldsymbol{x},\boldsymbol{v},t)\, \boldsymbol{\mathcal{H}}_m^{[\theta_0]}(\boldsymbol{v})}{w^{[\theta_0]}(\boldsymbol{v})} d\boldsymbol{v} \\ &\approx \theta_0^m \sum_{i=1}^{q^D} \omega_i \frac{f(\boldsymbol{x},\boldsymbol{v}_i,t)\, \boldsymbol{\mathcal{H}}_m^{[\theta_0]}(\boldsymbol{v}_i)}{w^{[\theta_0]}(\boldsymbol{v}_i)},\end{aligned} \tag{5.35}$$

où les $\boldsymbol{v}_i$ sont les nœuds de la quadrature, *i.e.* les zéros de $\boldsymbol{\mathcal{H}}_q^{[\theta_0]}$ et les ω_i leurs poids (cf. annexe A.3).

On remarque alors immédiatement que l'évaluation de $\boldsymbol{a}_q$ avec une quadrature à q points, donne $\boldsymbol{a}_q = 0$, car les $\boldsymbol{v}_i$ sont les racines de $\boldsymbol{\mathcal{H}}_q^{[\theta_0]}$. D'autre part, nous savons qu'une quadrature de Gauss à q points par direction permet d'intégrer exactement, tout polynôme de degré $N \leq 2q$. Dans le cas où le développement de f est tronqué

à l'ordre q :

$$\tilde{f}(\boldsymbol{x},\boldsymbol{v},t) = w^{[\theta_0]}(\boldsymbol{v}) \sum_{n=0}^{q} \frac{1}{n!} \boldsymbol{a}_n(\boldsymbol{x},t) \boldsymbol{\mathcal{H}}_n^{[\theta_0]}(\boldsymbol{v}), \tag{5.36}$$

l'obtention des $q+1$ coefficients $\boldsymbol{a}_n$, de cette fonction $\tilde{f}$ tronquée, par la relation (5.35), correspond à des quadratures de tels polynômes, et les quadratures :

$$\boldsymbol{a}_m(\boldsymbol{x},t) = \theta_0^m \sum_{i=1}^{q^D} \omega_i \frac{\tilde{f}(\boldsymbol{x},\boldsymbol{v}_i,t)\, \boldsymbol{\mathcal{H}}_m^{[\theta_0]}(\boldsymbol{v}_i)}{w^{[\theta_0]}(\boldsymbol{v}_i)}, \tag{5.37}$$

sont exactes.

Néanmoins, le coefficient $\boldsymbol{a}_q$ étant nul par construction, $\tilde{f}$ ne sera *développable explicitement*[8] et de manière exacte, que jusqu'à l'ordre $q-1$.

Finalement, nous poserons :

$$\begin{aligned} f_i(\boldsymbol{x},t) &= \omega_i \sum_{n=0}^{q-1} \frac{1}{n!} \boldsymbol{a}_n(\boldsymbol{x},t) \boldsymbol{\mathcal{H}}_n^{[\theta_0]}(\boldsymbol{v}_i), \\ &= \omega_i \frac{\tilde{f}(\boldsymbol{x},\boldsymbol{v}_i,t)}{w^{[\theta_0]}(\boldsymbol{v}_i)}, \end{aligned} \tag{5.38}$$

avec

$$\boldsymbol{a}_m(\boldsymbol{x},t) = \theta_0^m \sum_{i=1}^{q^D} f_i(\boldsymbol{x},t)\, \boldsymbol{\mathcal{H}}_m^{[\theta_0]}(\boldsymbol{v}_i). \tag{5.39}$$

En particulier, les trois premiers moments seront :

$$\rho = \sum_{i=1}^{q^D} f_i, \tag{5.40}$$

$$\rho\, \boldsymbol{u} = \sum_{i=1}^{q^D} \boldsymbol{v_i}\, f_i, \tag{5.41}$$

$$D\, \rho\, \theta + \rho\, \boldsymbol{u}^2 = \sum_{i=1}^{q^D} \boldsymbol{v_i}^2\, f_i. \tag{5.42}$$

La fonction de distribution discrétisée f_i est maintenant caractérisée par sa vitesse discrète $\boldsymbol{v}_i$. Nous parlerons donc parfois de faisceau monocinétique de particules, ou simplement de faisceau. Chaque faisceau ayant son propre poids ω_i et son propre

8. Il est important de comprendre que, bien qu'en pratique, seuls les deux premiers ainsi que la trace du troisième coefficients de f sont calculés, de sorte à construire f^{eq}, les autres coefficients de f sont très importants. En effet, ces derniers comptent pour représentation du hors équilibre et devront donc être calculables exactement. Nous verrons, lors de la simulation d'un gaz sans collision (section 6.1), que plus l'ordre de la quadrature sera important et plus nous pourrons suivre le gaz s'éloigner de l'équilibre de manière importante, car plus nous représenterons, implicitement, un nombre important de coefficients de f.

peuplement, dépendant des $\boldsymbol{a}_n(\boldsymbol{x},t)$ projetés sur les $\boldsymbol{\mathcal{H}}_n^{[\theta_0]}(\boldsymbol{v}_i)$.

5.5 Fonction d'équilibre discrète

De même que pour la fonction de distribution f (cf. Eq. (5.36)), tronquons le développement de la fonction de distribution d'équilibre à un ordre[9] $N \leq q-1$:

$$\tilde{f^{eq}}(\boldsymbol{x},\boldsymbol{v},t) = w^{[\theta_0]}(\boldsymbol{v}) \sum_{n=0}^{N} \frac{1}{n!}\, \boldsymbol{a}_n^{eq}(\boldsymbol{x},t) \boldsymbol{\mathcal{H}}_m^{[\theta_0]}(\boldsymbol{v}), \tag{5.43}$$

où les cinq premiers coefficients $\boldsymbol{a}_n^{eq}$ sont donnés par les équations (5.18)-(5.22).

Et nous la discrétiserons de même :

$$f_i^{eq}(\boldsymbol{x},t) = \omega_i \sum_{n=0}^{N} \frac{1}{n!}\, \boldsymbol{a}_n^{eq}(\boldsymbol{x},t) \boldsymbol{\mathcal{H}}_n^{[\theta_0]}(\boldsymbol{v}_i), \tag{5.44}$$

avec

$$\boldsymbol{a}_m^{eq}(\boldsymbol{x},t) = \theta_0^m \sum_{i=1}^{q^D} f_i^{eq}(\boldsymbol{x},t)\, \boldsymbol{\mathcal{H}}_m^{[\theta_0]}(\boldsymbol{v}_i). \tag{5.45}$$

Les quatre premiers termes $\boldsymbol{a}_n^{eq}(\boldsymbol{x},t)\boldsymbol{\mathcal{H}}_n^{[\theta_0]}(\boldsymbol{v}_i)$ de f_i^{eq} peuvent être calculés explicitement :

$$a_0^{eq} \times \mathcal{H}_0^{[\theta_0]}(\boldsymbol{v}_i) = \rho, \tag{5.46}$$

$$\boldsymbol{a}_1^{eq} \cdot \boldsymbol{\mathcal{H}}_1^{[\theta_0]}(\boldsymbol{v}_i) = \rho\, \frac{\boldsymbol{u}\cdot\boldsymbol{v}_i}{\theta_0}, \tag{5.47}$$

$$\boldsymbol{a}_2^{eq} : \boldsymbol{\mathcal{H}}_2^{[\theta_0]}(\boldsymbol{v}_i) = \rho\, \frac{(\boldsymbol{u}\cdot\boldsymbol{v}_i)^2}{\theta_0^2} + \rho\,(\theta-\theta_0)\left(\frac{\boldsymbol{v}_i^2}{\theta_0^2} - \frac{D}{\theta_0}\right) - \rho\frac{\boldsymbol{u}^2}{\theta_0}, \tag{5.48}$$

$$\begin{aligned}\boldsymbol{a}_3^{eq} : \boldsymbol{\mathcal{H}}_3^{[\theta_0]}(\boldsymbol{v}_i) = {} & \rho\frac{(\boldsymbol{u}\cdot\boldsymbol{v}_i)^3}{\theta_0^3} + 3\,\rho\,(\theta-\theta_0)\left(\frac{\boldsymbol{v}_i^2}{\theta_0^2} - \frac{(D+2)}{\theta_0}\right)\frac{\boldsymbol{u}\cdot\boldsymbol{v}_i}{\theta_0} \\ & - 3\,\rho\frac{(\boldsymbol{u}\cdot\boldsymbol{v}_i)\boldsymbol{u}^2}{\theta_0^2}.\end{aligned} \tag{5.49}$$

Un développement important de f^{eq} est celui pour lequel $N=2$:

$$f_i^{eq} = \omega_i\,\rho\left(1 + \frac{\boldsymbol{u}\cdot\boldsymbol{v}_i}{\theta_0} + \frac{(\boldsymbol{u}\cdot\boldsymbol{v}_i)^2}{2\,\theta_0^2} + \frac{(\theta-\theta_0)}{2\,\theta_0}\left(\frac{\boldsymbol{v}_i^2}{\theta_0} - D\right) - \frac{\boldsymbol{u}^2}{2\,\theta_0}\right), \tag{5.50}$$

qui est le premier à faire apparaître la température θ, et donc celui avec lequel nous pourrons aborder les problèmes anisothermes.

9. L'ordre du développement de f^{eq} n'est pas forcément le même que celui de f, comme nous le verrons.

Une chose remarquable dans le développement qui vient d'être fait, par comparaison à celui utilisé initialement pour dériver les LBM (voir [WG00]), est la présence de la température θ dans la fonction de distribution d'équilibre, qui lorsqu'elle est égale à la température de référence θ_0, nous redonne, pour $N = 2$, la fonction d'équilibre des LBM. Nous verrons plus tard que ceci n'est pas un hasard, du fait que les LBM sont construites, pour les plus élémentaires, sur des quadratures à trois points par direction. Ces LBM, nommées $D2Q9$ en dimension 2 et $D3Q27$ en dimension 3, sont en effet isothermes, car elles ne possèdent pas assez de degrés de liberté (de faisceaux) pour que le flux de chaleur soit représenté. En effet, pour ces schémas, $\boldsymbol{a}_3 = 0$ par construction (quadrature) et seuls les éléments non diagonaux du tenseur des pressions (le coefficient $\boldsymbol{a}_2$) représentent le hors équilibre dans f, du fait de sa troncature (à l'ordre $q - 1 = 2$) par la quadrature (à $q = 3$ points par direction).

5.6 Opérateur de collisions

Dans les deux sections précédentes, nous avons discrétisé l'espace des vitesses par le choix d'une quadrature des moments et ainsi obtenu des formes discrètes pour f (Eq. (5.38)) et f^{eq} (Eq. (6.23)). Nous allons maintenant passer à la discrétisation du temps par la discrétisation de l'opérateur de collisions pour la fonction f_i, l'espace des configurations restant pour l'instant continu.

Nous cherchons donc ici à résoudre une équation différentielle ordinaire, pour f_i :

$$D_t f_i + \frac{f_i}{\tau} = \frac{f_i^{eq}}{\tau}, \tag{5.51}$$

où $D_t = \partial_t + \boldsymbol{v} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}$.

Deux méthodes distinctes permettent de réaliser cette tâche [XH03]. La première que l'on nomme *splitting des opérateurs* de transport et de collisions, est bien connue et fréquemment utilisée, notamment en DSMC ; c'est la méthode qui a été la plus utilisée dans le cadre des LBM [WG00, CD98]. Comme nous le verrons, elle impose un certain nombre d'hypothèses, notamment que le pas de temps d'intégration δt soit petit devant le temps de relaxation τ et nécessitera un développement multi-échelles, type Chapman-Enskog pour retrouver une cohérence avec le régime hydrodynamique. La seconde méthode consiste à faire une moyenne des collisions sur chaque pas de temps.

5.6.1 Splitting des opérateurs

Formellement [10] l'équation (5.51), s'intègre sur un laps de temps δt, et donne :

$$f_i(\boldsymbol{x}+\boldsymbol{v}_i\delta t, t+\delta t) = e^{-\frac{\delta t}{\tau}} f_i(\boldsymbol{x},t) + \frac{e^{-\frac{\delta t}{\tau}}}{\tau}\int_0^{\delta t} e^{\frac{t'}{\tau}} f_i^{eq}(\boldsymbol{x}+\boldsymbol{v}_i t', t+t')dt'. \tag{5.54}$$

Il nous faut donc maintenant évaluer le terme intégral contenu dans cette équation. Pour δt petit [11], nous pouvons approximer f_i^{eq} par son développement au premier ordre :

$$f_i^{eq}(\boldsymbol{x}+\boldsymbol{v}_i t', t+t') = f_i^{eq}(\boldsymbol{x},t) + t'\frac{f_i^{eq}(\boldsymbol{x}+\boldsymbol{v}_i\delta t, t+\delta t) - f_i^{eq}(\boldsymbol{x},t)}{\delta t} + \mathcal{O}(\delta t^2). \tag{5.55}$$

De même pour $\exp(-\delta t/\tau) = 1-\delta t/\tau+\mathcal{O}(\delta t^2)$ et pour $\exp(t'/\tau) = 1+t'/\tau+\mathcal{O}(\delta t^2)$.

Finalement, l'évaluation de l'équation (5.54) sous ces hypothèses, après élimination de tous les termes en $\mathcal{O}(\delta t^2)$, donne :

$$f_i(\boldsymbol{x}+\boldsymbol{v}_i\delta t, t+\delta t) = f_i(\boldsymbol{x},t) - \frac{\delta t}{\tau}(f_i(\boldsymbol{x},t) - f_i^{eq}(\boldsymbol{x},t)), \tag{5.56}$$

qui est maintenant discrétisée en vitesse et en temps.

Il nous faut cependant maintenant déterminer quels sont les coefficients de transport (viscosité et conductivité thermique) associés à cette équation discrète. Pour ce faire, nous devons pratiquer une expansion de Chapman-Enskog, similaire à celle présentée section 3.4, mais différente du fait du caractère discret de l'équation (5.56).

Commençons par exprimer le membre de gauche de l'équation (5.56) sous forme d'un développement de Taylor :

$$f_i(\boldsymbol{x}+\boldsymbol{v}_i\delta t, t+\delta t) = \sum_{n=0}^{\infty}\frac{\delta t^n}{n!} D_t^n f_i(\boldsymbol{x},t), \tag{5.57}$$

10. Il faut utiliser ici la méthode de la variation de la constante. En effet, pendant le temps d'intégration δt, la fonction de distribution d'équilibre est susceptible de varier, on résout donc une équation de la forme : $ay'(t) + y(t) = g(t)$, avec $a \equiv \tau$, $y \equiv f_i$ et $g \equiv f_i^{eq}$. La solution générale de cette équation différentielle est la suivante :

$$y(t) = e^{F(t)}\left(K + \int \frac{g(t)}{a} e^{-F(t)} dt\right), \tag{5.52}$$

avec $F(t) = -\int a^{-1}dt$. Pour l'équation (5.51), cette intégrale est faite le long de la trajectoire caractéristique $(\boldsymbol{x},t) \to (\boldsymbol{x}+\boldsymbol{v}_i\delta t, t+\delta t)$, d'où :

$$f_i(\boldsymbol{x}+\boldsymbol{v}_i\delta t, t+\delta t) = Ke^{-\frac{\delta t}{\tau}} + \frac{e^{-\frac{\delta t}{\tau}}}{\tau}\int_0^{\delta t} e^{\frac{t'}{\tau}} f_i^{eq}(\boldsymbol{x}+\boldsymbol{v}_i t', t+t')dt'. \tag{5.53}$$

Finalement, la condition initiale, pour $\delta t = 0$, nous donne $K = f_i(\boldsymbol{x},0)$. D'où l'expression (5.54).

11. Ou en considérant que f_i^{eq} varie linéairement.

puis, substituons ce développement, tronqué à l'ordre 2, dans l'expression (5.56), il vient :

$$D_t f_i(\boldsymbol{x},t) + \frac{\delta t}{2} D_t^2 f_i(\boldsymbol{x},t) = -\frac{1}{\tau}(f_i(\boldsymbol{x},t) - f_i^{eq}(\boldsymbol{x},t)). \tag{5.58}$$

Lorsque l'on étudie le comportement macroscopique d'un écoulement, aux échelles caractéristiques de temps et d'espace τ_M et L, avec $\tau \ll \tau_M$, l'équation (5.58) peut être réécrite sous la forme adimensionnée suivante :

$$D_{\hat{t}} f(\boldsymbol{x},t) + \frac{\epsilon_0}{2} D_{\hat{t}}^2 f_i(\boldsymbol{x},t) = -\frac{1}{\epsilon}(f_i(\boldsymbol{x},t) - f_i^{eq}(\boldsymbol{x},t)), \tag{5.59}$$

où $D_{\hat{t}} = \tau_M D_t$, $\epsilon = \tau/\tau_M$ et $\epsilon_0 = \delta t/\tau_M \ll \epsilon$.

Développons maintenant f sous forme de son développement perturbatif autour de l'équilibre :

$$f(\boldsymbol{x},t) = \sum_{m=0}^{\infty} \epsilon^m f_i^m = f_i^{eq} + \sum_{m=1}^{\infty} \epsilon^m f_i^m = f_i^{eq} + f_i^{neq}, \tag{5.60}$$

avec $f_i^{neq} = \epsilon f_i^{(1)} + \epsilon^2 f_i^{(2)} + \ldots$

Alors, par substitution de cette expression dans l'équation (5.60), nous obtenons :

$$\epsilon^0 : \quad D_{\hat{t}} f_i^{eq} = -f_i^{(1)}, \tag{5.61}$$

$$\epsilon^1 : \quad D_{\hat{t}} f_i^{(1)} + \frac{\epsilon_0}{2\epsilon} D_{\hat{t}}^2 f_i^{eq} = -f_i^{(2)}, \tag{5.62}$$

$$\epsilon^2 : \quad D_{\hat{t}} f_i^{(2)} + \frac{\epsilon_0}{2\epsilon} D_{\hat{t}}^2 f_i^{(1)} = -f_i^{(3)}, \tag{5.63}$$

$$\vdots$$

Équations auxquelles nous devons ajouter les contraintes suivantes :

$$\sum_{i=1}^{q^D} \begin{pmatrix} 1 \\ \boldsymbol{v}_i \\ \boldsymbol{v}_i \cdot \boldsymbol{v}_i \end{pmatrix} f_i^{(m)} = 0, \quad \text{pour} \quad m = 1,2,3,\ldots. \tag{5.64}$$

Car les composantes hors équilibres de f_i, ne contribuent pas aux trois premiers moments.

Là encore, le calcul des équations hydrodynamiques associées aux différents ordres en ϵ, permet d'obtenir les relations entre le temps de relaxation τ et les coefficients de transport ν et α. Cependant, la discrétisation du temps, introduite par le splitting des opérateurs, modifie les coefficients de transport macroscopiques. En particulier, la viscosité ν et la diffusivité thermique α ne seront plus ceux de l'équation continue de départ. Cette différence se répercutera cependant de la même manière sur ces deux coefficients, par l'ajout de la quantité $\delta t/2$ au temps de relaxation continu. Nous aurons :

$$\nu_{\delta t} = \theta(\tau - \delta t/2) \tag{5.65}$$

et

$$\alpha_{\delta t} = \theta(\tau - \delta t/2). \tag{5.66}$$

5.6.2 Moyenne des collisions

Ici, au lieu de faire l'hypothèse qu'aucune collision n'a lieu durant l'étape de transport, comme cela a été fait dans la section précédente, nous allons supposer que l'effet des collisions sur un intervalle de temps donné, correspond à la moyenne des collisions au début et à la fin de cet intervalle. Cette opération revient à intégrer le second membre de l'équation de Boltzmann-BGK, le membre de droite étant donné par la solution caractéristique :

$$f_i(\boldsymbol{x} + \boldsymbol{v}_i\delta t, t + \delta t) - f_i(\boldsymbol{x},t) = -\frac{1}{\tau}\int_t^{t+\delta t} (f_i(\boldsymbol{x},t') - f_i^{eq}(\boldsymbol{x},t'))dt', \tag{5.67}$$

par la méthode de trapèzes :

$$f_i(\boldsymbol{x}+\boldsymbol{v}_i\delta t, t+\delta t)-f_i(\boldsymbol{x},t) = -\frac{1}{\tau}\frac{\delta t}{2}(f_i^{neq}(\boldsymbol{x}+\boldsymbol{v}_i\delta t, t+\delta t)+f_i^{neq}(\boldsymbol{x},t))+\mathcal{O}(\delta t^2), \tag{5.68}$$

où $f_i^{neq} = f_i - f_i^{eq}$.

Ici, contrairement à la méthode de splitting, aucune hypothèse sur la petitesse de $\delta t/\tau$ n'a été faite. Et bien que le schéma résultant soit *implicite en temps*, il peut être rendu *explicite* par l'introduction du changement de variable suivant :

$$\bar{f}_i = f_i + \frac{\delta t}{2\tau} f_i^{neq}, \tag{5.69}$$

$$\bar{f}_i^{eq} = f_i^{eq}, \tag{5.70}$$

qui ne modifie pas les contraintes sur les invariants collisionnels, d'où :

$$\rho = \sum_{i=1}^{q^D} \bar{f}_i, \tag{5.71}$$

$$\rho\,\boldsymbol{u} = \sum_{i=1}^{q^D} \boldsymbol{v_i}\,\bar{f}_i, \tag{5.72}$$

$$D\,\rho\,\theta + \rho\,\boldsymbol{u}^2 = \sum_{i=1}^{q^D} \boldsymbol{v_i^2}\,\bar{f}_i, \tag{5.73}$$

et permettra d'écrire :

$$\bar{f}_i(\boldsymbol{x} + \boldsymbol{v}_i\delta t, t + \delta t) - \bar{f}_i(\boldsymbol{x},t) = -\frac{\delta t}{\tau + \delta t/2}(\bar{f}_i(\boldsymbol{x},t) - f_i^{eq}(\boldsymbol{x},t)). \tag{5.74}$$

Cette équation de Boltzmann-BGK diffère de l'équation (5.56) par son temps de

relaxation $\bar{\tau} = \tau + \delta t/2$, qui incorpore directement la modification des coefficients de transport liée à la discrétisation du temps. Dans le cas présent, la viscosité ν et la diffusivité thermique α seront les mêmes que pour l'équation continue, soit $\nu = \alpha = \theta\tau$, cependant comme $\tau = \bar{\tau} - \delta t/2$, l'équation (5.74) sera bien équivalente à l'équation (5.56) avec $\nu_{\delta t} = \alpha_{\delta t} = \theta(\bar{\tau} - \delta t/2)$.

Ce développement est discuté et généralisé par Bardow et *al.* [BKG06, BKG08].

5.7 Hiérarchie des coefficients d'Hermite

Dans le cas continu (cf. section 3.1), nous avons vu qu'à l'équation de Boltzmann-BGK correspondait une hiérarchie d'équations de bilan sur les moments $\boldsymbol{\mathcal{M}}_k$ de la fonction de distribution f. Nous allons voir ici, qu'il en est de même pour les coefficients d'Hermite du développement de f énoncé section 5.3. Après avoir écrit, dans le cas général, cette hiérarchie d'équations sur les coefficients $\boldsymbol{a}_n$, nous regarderons quel est l'effet d'une quadrature de Gauss-Hermite sur celle-ci. Le fait qu'une quadrature d'ordre q, appliquée à la fonction de distribution développée sur la base des polynômes d'Hermite, en annule le coefficient de rang q ($\boldsymbol{a}_q$), laisse en effet présager que l'effet de la quadrature sur la hiérarchie, sera de la tronquer à cet ordre. Nous allons ici vérifier cette propriété.

Reprenons l'équation de Boltzmann-BGK sans force extérieure :

$$\partial_t f + \boldsymbol{v} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} f = -\frac{1}{\tau}(f - f^{eq}), \tag{5.75}$$

où l'on considère maintenant les fonctions f et f^{eq} sous leur formes (5.23) et (5.11).

Prenons alors le produit Hermitien de (5.75) avec $\boldsymbol{\mathcal{H}}_n^{[\theta_0]}(\boldsymbol{v})$, on obtient :

$$\theta_0^{-n}\partial_t \boldsymbol{a}_n + \boldsymbol{\nabla}_{\boldsymbol{x}} \int \boldsymbol{v}\, \boldsymbol{\mathcal{H}}_n^{[\theta_0]} f d\boldsymbol{v} = -\frac{\theta_0^{-n}}{\tau}(\boldsymbol{a}_n - \boldsymbol{a}_n^{eq}), \tag{5.76}$$

avec

$$\boldsymbol{a}_n = \theta_0^n \int f\, \boldsymbol{\mathcal{H}}_n^{[\theta_0]} d\boldsymbol{v}, \quad \boldsymbol{a}_n^{eq} = \theta_0^n \int f^{eq}\, \boldsymbol{\mathcal{H}}_n^{[\theta_0]} d\boldsymbol{v}. \tag{5.77}$$

Or, $\boldsymbol{v}\, \boldsymbol{\mathcal{H}}_n^{[\theta_0]} = \theta_0^n\, \boldsymbol{\mathcal{H}}_{n+1}^{[\theta_0]} + \boldsymbol{\mathcal{H}}_{n-1}^{[\theta_0]}$ (cf. A.2), expression grâce à laquelle nous pouvons évaluer le terme intégral restant dans l'équation (5.76).

On obtient finalement :

$$\partial_t \boldsymbol{a}_n + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \boldsymbol{a}_{n+1} + \theta_0 \boldsymbol{\nabla}_{\boldsymbol{x}} \otimes \boldsymbol{a}_{n-1} = -\frac{1}{\tau}(\boldsymbol{a}_n - \boldsymbol{a}_n^{eq}), \tag{5.78}$$

avec $\boldsymbol{a}_{-1} \equiv 0$. Ici $\boldsymbol{\nabla}_{\boldsymbol{x}}\cdot$ est une divergence et $\boldsymbol{\nabla}_{\boldsymbol{x}}\otimes$ est un gradient.

Nous savons (cf. 5.4) que l'évaluation de $\boldsymbol{a}_q$ avec une quadrature à q points, donne $\boldsymbol{a}_q = 0$, car les $\boldsymbol{v}_i$ sont les racines de $\boldsymbol{\mathcal{H}}_q^{[\theta_0]}$. Regardons alors quelles conséquences ceci

a sur la hiérarchie des coefficients (5.78), en se plaçant au rang q et en supposant que les $\boldsymbol{a}_n^{eq}$ sont calculés de la même manière que les $\boldsymbol{a}_n$ (Eq. (5.35)). Il vient :

$$\begin{aligned} q-1: &\quad \partial_t \boldsymbol{a}_{q-1} + \theta_0 \boldsymbol{\nabla}_{\boldsymbol{x}} \otimes \boldsymbol{a}_{q-2} = -\frac{1}{\tau}(\boldsymbol{a}_{q-1} - \boldsymbol{a}_{q-1}^{eq}) \\ q: &\quad \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \boldsymbol{a}_{q+1} + \theta_0 \boldsymbol{\nabla}_{\boldsymbol{x}} \otimes \boldsymbol{a}_{q-1} = 0 \\ q+1: &\quad \partial_t \boldsymbol{a}_{q+1} + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \boldsymbol{a}_{q+2} = -\frac{1}{\tau}(\boldsymbol{a}_{q+1} - \boldsymbol{a}_{q+1}^{eq}) \end{aligned} \tag{5.79}$$

On peut donc voir ici un couplage à l'ordre q, entre les ordres $q-1$ et $q+1$, par la relation $\boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \boldsymbol{a}_{q+1} + \theta_0 \boldsymbol{\nabla}_{\boldsymbol{x}} \otimes \boldsymbol{a}_{q-1} = 0$.

Nous savons qu'une quadrature à q points ne permet pas de calculer exactement les coefficients $\boldsymbol{a}_n$ d'ordre $n \geq q$. Un transit d'information à travers la hiérarchie, de l'ordre $q+1$ à l'ordre $q-1$, introduirait donc des erreurs dans la restriction de la hiérarchie que nous voulons représenter. Ce couplage à l'ordre q, est cependant sans effet sur les équations d'ordres inférieurs et supérieurs à q et ne représente qu'une propriété de la méthode.

5.8 Forces extérieures

Jusqu'à présent, nous avons laissé de côté les forces extérieures $\boldsymbol{F} = m\boldsymbol{g}$, correspondant au terme $\boldsymbol{g} \cdot \boldsymbol{\nabla}_{\boldsymbol{v}} f$ dans l'équation de Boltzmann-BGK. Comme cela a été dit dans la section 2.3.1, l'application d'une force extérieure correspond à un transport à la "vitesse" $\boldsymbol{g}$ de la fonction de distribution dans l'espace des vitesses. Bien que l'application d'une procédure de transport soit possible dans l'espace des vitesses, nous ne procéderons pas de cette façon et appliquerons les forces extérieures par une modification appropriée de la fonction de distribution d'équilibre.

Rappelons que l'opérateur BGK consiste à faire relaxer la fonction de distribution $f(\boldsymbol{x},\boldsymbol{v},t)$ vers la fonction de distribution d'équilibre $f^{eq}(\boldsymbol{x},\boldsymbol{v},t)$, dont l'expression est :

$$f^{eq} = \rho\,(2\pi\theta)^{-D/2} \exp\left(-\frac{(\boldsymbol{v}-\boldsymbol{u})^2}{2\theta}\right). \tag{5.80}$$

Considérons alors la loi fondamentale de la dynamique, $\boldsymbol{g} = d\boldsymbol{u}/dt$, qui nous permet d'écrire l'accroissement de la vitesse $\delta\boldsymbol{u}$ sur un laps de temps δt en présence de la force $m\boldsymbol{g}$:

$$\delta\boldsymbol{u} = \boldsymbol{g}\delta t, \tag{5.81}$$

d'où, en première approximation :

$$\boldsymbol{u}(\boldsymbol{x},t+\delta t) = \boldsymbol{u}(\boldsymbol{x},t) + \boldsymbol{g}\delta t. \tag{5.82}$$

Considérons alors une nouvelle fonction de distribution d'équilibre :

$$f^{eq^*} = \rho\,(2\pi\theta)^{-D/2} \exp\left(-\frac{(\boldsymbol{v} - [\boldsymbol{u} + A])^2}{2\theta}\right), \tag{5.83}$$

dont la vitesse moyenne a été augmentée d'une certaine quantité A, puis calculons l'accroissement de la vitesse durant l'étape de collisions.

La variation de f due aux collisions étant :

$$\left.\frac{\delta f}{\delta t}\right|_{coll} = -\frac{1}{\tau}(f(\boldsymbol{x},t) - f^{eq}(\boldsymbol{x},t)), \tag{5.84}$$

nous avons pour la vitesse :

$$\left.\frac{\delta u}{\delta t}\right|_{coll} = -\frac{1}{\tau}\int \boldsymbol{v}(f(\boldsymbol{x},t) - f^{eq}(\boldsymbol{x},t))d\boldsymbol{v} = \frac{\rho A}{\tau}, \tag{5.85}$$

soit :

$$\boldsymbol{u}(\boldsymbol{x},t+\delta t) = \boldsymbol{u}(\boldsymbol{x},t) + \frac{\rho A}{\tau}\delta t. \tag{5.86}$$

Finalement, par identification avec l'équation (5.82), nous obtenons $A = \boldsymbol{g}\rho/\tau$.

Ainsi, l'application d'une force extérieure se fera en définissant une fonction de distribution d'équilibre translatée de $\boldsymbol{g}\rho/\tau$ dans l'espace des vitesses, soit :

$$f^{eq^*} = \rho\,(2\pi\theta)^{-D/2} \exp\left(-\frac{(\boldsymbol{v} - [\boldsymbol{u} + \boldsymbol{g}\rho/\tau])^2}{2\theta}\right). \tag{5.87}$$

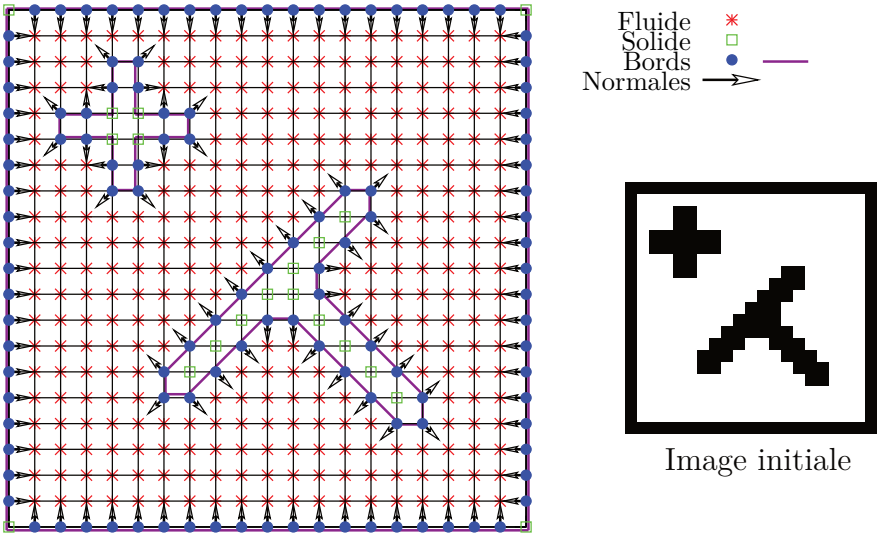


FIGURE 5.2 – Exemple d'une géométrie générée à partir d'une image obtenue grâce aux programmes donnés dans l'annexe C.2.5. Les conditions aux bords cinétiques nécessitent la connaissance des normales aux bords représentées ici par des vecteurs. On observe que pour les parois selon les diagonales, la distance entre les nœuds de bords est dilatée d'un facteur $\sqrt{2}$ par rapport à celle des parois dans les directions principales.

5.9 Conditions aux limites

Deux types de conditions aux limites vont être présentés dans cette section. Le premier type correspond à la discrétisation par la quadrature de Gauss-Hermite des conditions aux bords de Maxwell avec accommodation totale, présentées sections 4.1.2 dans le cas continu. Ces conditions aux bords nous permettront d'imposer, dans les simulations, la vitesse $\boldsymbol{U}_W$ et la température θ_W sur les parois solides délimitant le gaz en écoulement. Le second type de conditions aux limites sera des conditions de pression.

5.9.1 Discrétisation des conditions aux bords cinétiques

L'objectif est ici de discrétiser les conditions aux bords de Maxwell avec accommodation totale énoncées 4.1.2. Nous avions obtenu comme expression pour la fonction de distribution émergeant de la surface d'une paroi solide :

$$f(\boldsymbol{x}_W,\boldsymbol{\xi},t) = f_W^{eq}(\boldsymbol{x}_W,\boldsymbol{\xi},t)\frac{\int_{\boldsymbol{\xi}'\cdot\boldsymbol{n}<0}|\boldsymbol{\xi}\cdot\boldsymbol{n}|\,f(\boldsymbol{x}_W,\boldsymbol{\xi}',t)d\boldsymbol{\xi}}{\int_{\boldsymbol{\xi}'\cdot\boldsymbol{n}<0}|\boldsymbol{\xi}'\cdot\boldsymbol{n}|\,f_W^{eq}(\boldsymbol{x}_W,\boldsymbol{\xi}',t),d\boldsymbol{\xi}'}, \quad (\boldsymbol{\xi}\cdot\boldsymbol{n}>0), \tag{5.88}$$

où $\boldsymbol{n}$ est la normale dirigée vers le fluide, $\boldsymbol{\xi} = \boldsymbol{v} - \boldsymbol{U}_W$ et $\boldsymbol{\xi}' = \boldsymbol{v}' - \boldsymbol{U}_W$ sont respectivement les vitesses des particules sortants et entrants de la paroi, enfin f_W^{eq}

est la distribution de Maxwell-Boltzmann à la vitesse $\boldsymbol{U}_W$ et à la température T_W de la paroi.

Tout comme les moments $\boldsymbol{a}_n$, les intégrales seront calculées numériquement grâce à une quadrature de Gauss-Hermite. Posons $\boldsymbol{\xi}_i = \boldsymbol{v}_i - \boldsymbol{U}_W$ et $\boldsymbol{\xi}'_i = \boldsymbol{v}'_i - \boldsymbol{U}_W$, les vitesses discrètes, nœuds de la quadrature, exprimées dans le référentiel de la paroi, il vient :

$$f_i = f^{eq}_{iW} \frac{\sum_{\boldsymbol{\xi}'_i \cdot \boldsymbol{n} < 0} |\boldsymbol{\xi}_i \cdot \boldsymbol{n}| \, f_i}{\sum_{\boldsymbol{\xi}'_i \cdot \boldsymbol{n} < 0} |\boldsymbol{\xi}'_i \cdot \boldsymbol{n}| \, f^{eq}_{iW}}, \quad (\boldsymbol{v}_i - \boldsymbol{U}_W) \cdot \boldsymbol{n} > 0, \tag{5.89}$$

avec :

$$f^{eq}_{iW} = \omega_i \, \rho_W \left(1 + \frac{\boldsymbol{U}_W \cdot \boldsymbol{v}_i}{\theta_0} + \frac{(\boldsymbol{U}_W \cdot \boldsymbol{v}_i)^2}{2\,\theta_0^2} + \frac{(\theta_W - \theta_0)}{2\,\theta_0} \left(\frac{\boldsymbol{v}_i^2}{\theta_0} - D \right) - \frac{\boldsymbol{U}_W^2}{2\,\theta_0} \right), \tag{5.90}$$

si nous choisissons de développer la fonction d'équilibre au second ordre.

Pour être appliquées, ces conditions aux bords nécessitent la connaissance de la normale à la paroi. Dans le cadre de cette thèse, tout comme dans la majorité des méthodes de Boltzmann sur réseau, l'espace des configurations est maillé uniformément. Les nœuds du réseau auront alors la possibilité de faire partie, soit d'une partie du domaine contenant du fluide, soit d'une partie contenant du solide. À l'interface entre ces deux types de nœuds se trouveront donc des bords, sur lesquels les normales $\boldsymbol{n}$ devront être définies. La recherche des bords et la définition des normales à ceux-ci, constituent un problème en soi. Ce problème dépendra en partie des choix fait dans la mise en œuvre de la méthode. En effet, considérons l'exemple de deux canaux parallèles séparés par une membrane solide, alors si nous choisissons que les nœuds de bords ne possèdent qu'une normale, cette membrane devra être constituée d'au moins deux couches de nœuds de bords, une couche supportant les normales à la paroi pour l'un des canaux et l'autre couche pour le second. *A contrario*, si nous choisissons de mettre en œuvre des nœuds de bords pouvant supporter, par exemple, deux normales de sens opposés, alors une seule couche de nœuds permettrait de définir une telle membrane. Le choix qui a été fait dans la mise en œuvre donnée en annexe (annexe C.2) de ce mémoire est le premier ; une seule normale est définie pour les nœuds de bords, une vérification de la géométrie est donc nécessaire, de façon à garantir qu'aucun nœuds ne soit isolé. D'autres problèmes doivent par ailleurs être pris en considération, en particulier lorsqu'une paroi n'est pas définie selon une direction principale de l'espace. En effet, dans ce cas la discrétisation de l'espace produit une discrétisation du bord en forme de marches d'escalier, comportant deux types de coins que l'on peut qualifier, soit d'entrant, soit de sortant. Là encore, un choix de mise en œuvre est à faire. Le choix retenu ici est de ne considérer que les coins sortants comme faisant partie du bords, les coins entrants étant quant à eux considérés comme faisant partie du solide. La conséquence de ce choix est simplement que les nœuds d'un bord en diagonal sont séparés par un distance de $\sqrt{2}\delta x$, avec δx le pas du réseau. L'illustration d'une géométrie, générée à partir d'une image, est représentée figure 5.2. Cette géométrie a été obtenue grâce aux programmes donnés dans l'annexe C.2.5. Signalons que la démarche employée ici

(cf. code de l'annexe C.2.5) est indépendante de la dimension du problème car toute les opérations peuvent être "splittées".

Le point le plus délicat concernant les bords est celui des bords en coins, d'autant que ces derniers sont indispensables pour simuler des écoulements dans des géométries complexes. Pour vérifier que les choix énoncés plus haut n'affectent pas la validité des résultats, considérons le cas d'un écoulement de Poiseuille dans un canal en diagonal. Ce problème n'est peut-être pas le pire des cas, mais il présente l'intérêt de posséder une solution analytique, contrairement à un écoulement autour d'un disque, par exemple. Considérons donc la géométrie représentée figure 5.3.

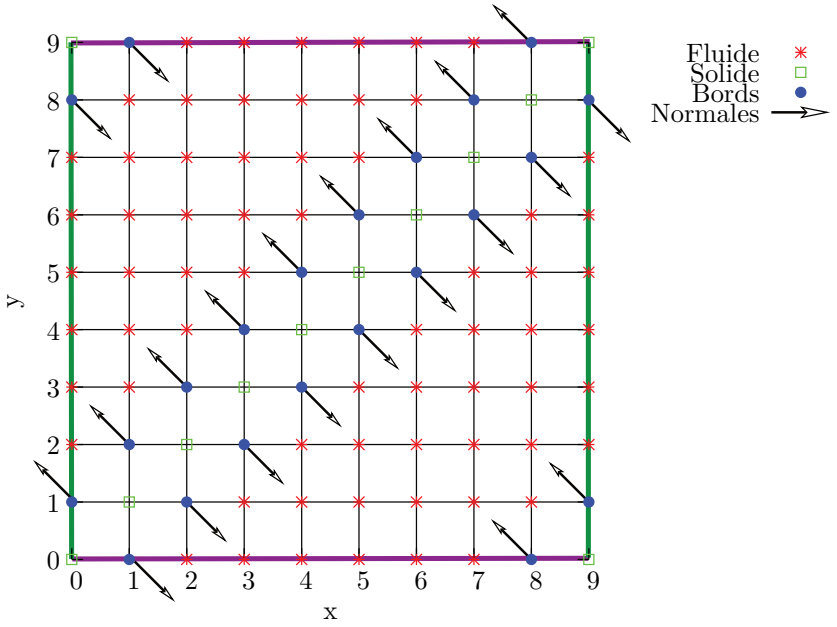


FIGURE 5.3 – Géométrie utilisée pour valider le bords "en coins". Le système est périodique selon x et y, et l'écoulement est mis en mouvement par une force extérieure parallèle à la paroi. Il s'agit donc ici d'un seul écoulement enroulé à la surface d'un tore.

La simulation de cet écoulement grâce au code maBLB (donné en annexe C.2), pour un nombre de Knudsen $K_n = 0{,}1$, est représentée figure 5.4. La solution analytique montre que la solution numérique obtenue est en très bon accord avec la théorie.

Ces conditions aux bords discrètes ont été introduites par Ansumali et *al.* dans le cadre des LBM entropiques (ELBM). Il est montré (cf. [AV02]) que l'erreur commise sur les bords par la quadrature est la même que celle commise dans le fluide. Cette analyse est importante, car contrairement à la fonction de distribution du fluide, la fonction de distribution sur les bords n'est pas gaussienne, mais formée de deux gaussiennes, une pour la paroi, à la température θ_W et à la vitesse $\boldsymbol{U}_W$ et une pour le fluide adjacent à la paroi et dont la température est $\theta(\boldsymbol{x}_W) \neq \theta_W$ et la vitesse est $\boldsymbol{u}(\boldsymbol{x}_W) \neq \boldsymbol{U}_W$. La fonction de distribution présente donc *a priori* une discontinuité sur les parois et il n'est pas évident qu'une quadrature de Gauss-Hermite y soit aussi

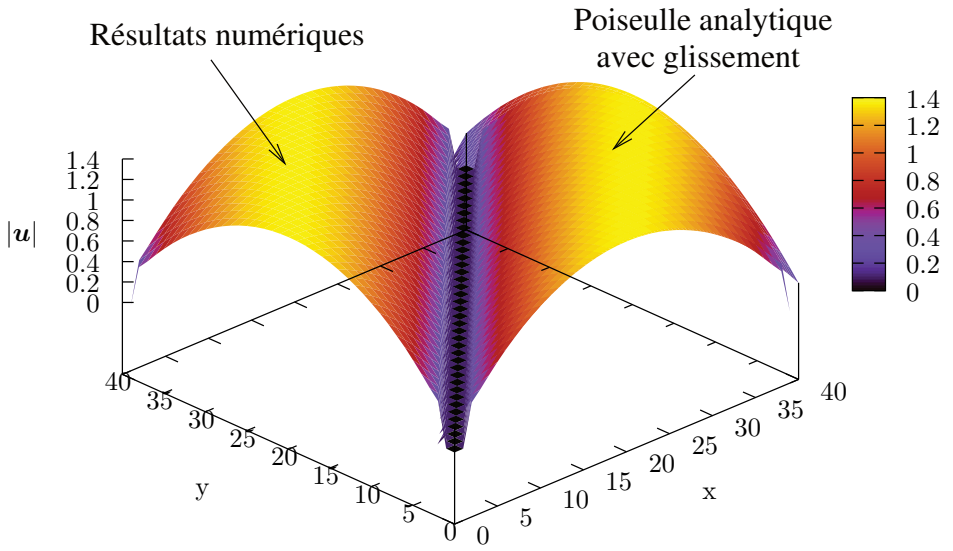


FIGURE 5.4 – Champ de vitesse obtenu pour l'écoulement de Poiseuille en diagonal.

bien adaptée qu'au sein du fluide.

5.9.2 Conditions aux bords de pression

Les conditions aux bords présentées ici seront utilisées pour réaliser les simulations présentées section 6.5, où l'écoulement d'un gaz à travers un long canal est imposé par des consignes sur la densité à l'entrée et à la sortie du canal.

Rappelons qu'en théorie cinétique, la pression d'un gaz parfait dilué est donnée par la relation $p = \rho\theta$, où ρ est la densité et $\theta = k_B T/m$. La pression dépend donc à la fois de la densité et de la température locale. Imposer une pression sur le bord du domaine d'intégration consistera donc essentiellement à imposer ρ et θ sur ce bord. Cependant, ne faire que cela ne suffit pas. D'une part, la connaissance de la pression que l'on impose, n'informe pas sur la valeur à donner à la vitesse macroscopique sur le bord, il nous faudra donc définir une stratégie pour la déterminer. D'autre part, nous devons, à partir de la valeur des champs macroscopiques, construire la fonction de distribution qui leur est associée. Or cette tâche n'est pas si simple qu'il y paraît. En effet, la seule fonction de distribution que l'on sache construire sans ambiguïté à partir d'un état macroscopique $(\rho,\boldsymbol{u},\theta)$, est la fonction de distribution d'équilibre de Maxwell-Boltzmann, et rien ne permet de dire *a priori*, que la fonction de distribution soit à l'équilibre sur les bords où une valeur de la pression est imposée. Ces bords ne sont en effet pas du même type que ceux définis dans la section précédente, ce ne sont pas des parois, mais des limites mathématiques imposées par la finitude du domaine d'intégration. En d'autres termes, du fluide est implicitement présent au-delà de ces bords, à l'extérieur du domaine. Nous devons donc faire en sorte de rendre compte de sa présence en définissant correctement la fonction de distribution sur ces bords.

Commençons par nous intéresser aux champs macroscopiques (ρ,$\boldsymbol{u}$,θ), en laissant de côté l'aspect cinétique. Comme cela vient d'être énoncé, un bord du domaine sur lequel nous cherchons à imposer une pression doit être virtuellement ouvert. La variation du champ de vitesse $\boldsymbol{u}$ doit donc être nulle, en première approximation au passage de ce bord. Si nous appelons $\partial\Omega$ la frontière du domaine Ω sur laquelle nous voulons imposer la pression, alors nous devons avoir :

$$\boldsymbol{n} \cdot \boldsymbol{\nabla} u_\alpha|_{\partial\Omega} = 0, \tag{5.91}$$

où $\boldsymbol{n}$ est le vecteur normal au bord $\partial\Omega$.

Considérons alors le cas discret représenté figure 5.5, où une limite est placée sur un des nœuds d'un maillage régulier de pas δx.

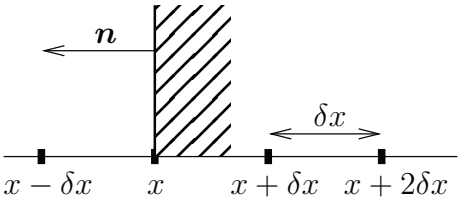


FIGURE 5.5 – Illustration du maillage près d'un bord. Le bord se trouve en x. Il est nécessaire de définir deux nœuds à l'intérieur de la paroi, en $x + \delta x$ et en $x + 2\delta x$, pour appliquer l'extrapolation.

La solution au problème (5.91) sur ce maillage est donnée par la procédure d'extrapolation dite de Richardson, dont le principe est donné ci-après. Par simplicité d'écriture, nous ne considérerons que le cas 1D, la généralisation aux dimensions supérieures est évidente et consiste à appliquer la procédure à chaque composante des champs vectoriels ou tensoriels. Commençons par écrire $u(x+\delta x)$ et $u(x+2\delta x)$ sous forme de développements de Taylor au troisième ordre :

$$u(x+\delta x) = u(x) + \delta x\, u'(x) + \frac{\delta x^2}{2} u''(x) + \mathcal{O}(\delta x^3), \tag{5.92}$$

$$u(x+2\delta x) = u(x) + 2\delta x\, u'(x) + \frac{4\delta x^2}{2} u''(x) + \mathcal{O}(\delta x^3). \tag{5.93}$$

Or la condition de non-variation du champs sur le bord (Eq. 5.91), impose $u'(x) = 0$. D'où, en prenant 4×(5.92) - (5.93) :

$$u(x) = \frac{4}{3} u(x+\delta x) - \frac{1}{3} u(x+2\delta x). \tag{5.94}$$

Cette expression permet, connaissant $u(x+\delta x)$ et $u(x+2\delta x)$, d'imposer $u(x)$ de telle sorte que l'équation (5.91) soit vérifiée.

Cette expression n'est cependant pas exactement celle dont nous avons besoin, car nous ne connaissons pas la valeur de u à l'extérieur du domaine, en $x+\delta x$ et $x+2\delta x$.

C'est pourquoi nous allons prendre le problème à l'envers, en partant du principe que le champ u est connu dans le domaine et chercher la valeur à imposer sur le bord. Nous placerons donc le bords en $x + 2\delta x$, comme illustré sur la figure 5.6, où u connu partout, sauf sur le bord.

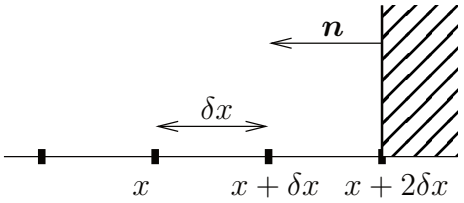


FIGURE 5.6 – Illustration du maillage près d'un bord. Le bord se trouve en maintenant en $x + 2\delta x$.

Alors, en inversant l'équation (5.94), nous avons :

$$u(x + 2\delta x) = 4\,u(x + \delta x) - 3\,u(x). \tag{5.95}$$

Expression que l'on utilisera pour extrapoler le champ de vitesse depuis le fluide vers le bord sur lequel la pression est imposée.

Signalons que ce traitement n'est pas réservé qu'au champ de vitesse $\boldsymbol{u}$, il est en effet possible de n'imposer sur le bord que la valeur de la densité et d'extrapoler le champ de température grâce à la relation (5.95), auquel cas la pression sur le bord dépendra de la température imposée par l'écoulement.

Passons maintenant à la partie cinétique du problème. Le but est de définir la fonction de distribution sur un bord du domaine où les champs macroscopiques ont été, soit imposés, soit extrapolés par la méthode de Richardson énoncée précédemment. Comme cela a été brièvement énoncé au début de cette section, nous ne savons définir sans ambiguïté, à partir de la connaissance des champs macroscopiques de densité, de vitesse et de température, qu'une seule fonction de distribution, qui est celle d'équilibre f^{eq}. Or, imposer cette fonction de distribution sur les bords que nous considérons ici, aurait comme effet de détruire la partie hors d'équilibre (f^{neq}) de f, que l'on rappelle être de la forme $f = f^{eq} + f^{neq}$. La solution proposée ici sera d'imposer l'équilibre f^{eq} sur le bord, puis de réintroduire la composante hors d'équilibre de la fonction de distribution en l'extrapolant, par la méthode de Richardson, depuis le fluide vers le bord. En reprenant le maillage représenté figure 5.6, cette procédure s'écrira :

$$f^{neq}(x + 2\delta x) = 4\,f^{neq}(x + \delta x) - 3\,f^{neq}(x), \tag{5.96}$$

avec $f^{neq} = f - f^{eq}$.

La mise en œuvre de ces conditions aux bords est donnée en annexe C.2.4.

5.10 Quadratures et schémas hybrides

Pour que la méthode soit complète, il faut maintenant déterminer un schéma de transport de la fonction de distribution.

Rappelons que la méthode présentée peut se voir comme une généralisation des LBM, en ce sens que nous cherchons à calculer les moments de la fonction de distribution, par des quadratures de Gauss-Hermite d'ordres arbitraires, alors que les LBM sont construites sur la base de quadratures d'ordre 3. Pour ces LBM, il y a donc à 3 faisceaux par direction, dont les vitesses discrètes sont les racines du polynôme $\mathcal{H}_3$, soit $\{-\sqrt{3},0,\sqrt{3}\}$, en dimension 1. Cette particularité rend les LBM particulièrement simples, car, dans ce cas, il est toujours possible de définir des pas d'espace δx et de temps δt, tels que la distance parcourue par les faisceaux sur un pas de temps, soit égale au pas du réseau. Ou, de façon équivalente, à choisir $\delta x = \delta t = 1$ et à redimensionner les vitesses discrètes par la vitesse de plus grande amplitude v_{max} [12]. Le schéma de transport en est alors simplifié et consiste en une procédure de copie de la fonction de distribution du nœud courant, vers le nœud pointé par la vitesse discrète à laquelle se propage le faisceau (on dit que le transport est de type *automate cellulaire*). Malheureusement, un tel schéma de transport n'est plus applicable dès lors que le degré de la quadrature de Gauss-Hermite est supérieur à 3. La raison en est l'irrationalité des racines des polynômes d'Hermite $\mathcal{H}_n$, pour $n > 3$. Avant d'aborder le schéma de transport, nous commencerons par décrire les quadratures de Gauss-Hermite.

5.10.1 Quadratures pures

Les figures 5.7 et 5.8 donnent des représentations de systèmes de vitesses respectivement en une et en deux dimensions. Sur la figure 5.7, sont aussi donnés, en ordonnés, les poids associés à chacune des vitesses. On remarque que l'accroissement de l'ordre de la quadrature permet, d'une part de mieux représenter le "gras" de la distribution de Maxwell-Boltzmann (voir figure 5.7), d'autre part, dans le cas où le dimension est supérieure à 1, d'augmenter le nombre de directions pointées par les faisceaux.

Il apparaît cependant que l'augmentation de l'ordre de la quadrature ajoute rapidement des faisceaux dont les poids sont très faibles, ils ne transporteront donc que très peu de particules et on peut se demander si leur existence améliore réellement les choses. Par exemple, on voit que la quadrature associée à $\mathcal{H}_6$ (Fig. 5.7(d)), ne comporte que 4 faisceaux de poids réellement significatifs, (tout comme $\mathcal{H}_4$ (Fig. 5.7(b))), les deux faisceaux les plus rapides de $\mathcal{H}_6$ ne transportant que très peu de particules. Étant donné qu'en dimension supérieure à un, les poids associés aux vitesses sont égaux aux produits des poids des vitesses 1D combinées pour générer la vitesse 2D, le poids du faisceau de cordonnées (1,1) de $\mathcal{H}_6$ (Fig. 5.8(d)), par exemple sera encore extrêmement petit.

12. Procéder ainsi aura pour effet de redimensionner la température par v_{max}^2. En particulier, la température de référence sera $\hat{\theta}_0 = \theta_0/v_{max}^2$.

5.10.2 Quadratures composites

Pour pallier l'introduction de faisceaux de poids trop faibles, nous proposons d'utiliser une composition de quadratures d'ordre peu élevé, par exemple $\mathcal{H}_3$ et $\mathcal{H}_4$, plutôt qu'une quadrature d'ordre élevé avec un nombre de faisceaux équivalents. De telles *quadratures composites* sont représentées figure 5.9. Il existe d'autres possibilités de compositions, cependant nous nous limiterons ici à celles dont la construction nous a semblé la plus naturelle. Ces quadratures sont composées de deux quadratures "pures" d'ordres successifs n et $n+1$ et seront notées $\mathcal{H}_{n|n+1}$.

La construction de la quadrature composite $\mathcal{H}_{n|n+1}$ est relativement simple. Elle consiste à construire les quadratures $\mathcal{H}_n$ et $\mathcal{H}_{n+1}$ en dimension souhaitée, à redimensionner les vitesses de chacune d'elles par la vitesse de plus grande amplitude $v_{max}(\mathcal{H}_{n+1})$ et finalement à diviser les poids ω_i de chacune, par 2. La somme des poids de la quadrature composite résultante sera donc toujours égale à l'unité, tout comme pour les quadratures initiales. Les vitesses discrètes sont donc celles des deux quadratures prises indépendamment. C'est au moment de l'application des collisions que l'ensemble des faisceaux interagissent.

Sur la figure 5.9, sont représentées les quadratures composites $\mathcal{H}_{3|4}$ et $\mathcal{H}_{4|5}$ en une et en deux dimensions. La comparaison de ces deux dernières avec les quadratures pures représentées figures 5.7 et 5.8, permet de constater que cette procédure de composition permet, à nombre de faisceaux équivalent, un bien meilleur échantillonnage de la distribution de Maxwell-Boltzmann. À noter que par le mode de construction défini ici, toutes les quadratures composites possèdent un faisceau de vitesse nulle, contrairement aux quadratures pures, pour lesquelles l'existence de ce faisceau dépendait de la parité de celle-ci.

Dans la partie application (cf. chapitre 6), nous analyserons systématiquement les résultats obtenus avec des quadratures pures et des quadratures composites. Nous verrons alors que la quadrature $\mathcal{H}_{4|5}$ semble être quasi optimale dans le cadre que nous nous sommes fixé, *i.e.* pour simuler des écoulements dont le nombre de Knudsen est compris dans l'intervalle $0{,}0001 < K_n < 10$. Les quadratures composites présentent en effet de bien meilleurs résultats que les quadratures pures, cependant les améliorations apportées par les quadratures composites $\mathcal{H}_{n|n+1}$ pour $n > 4$ ne sont que minimes.

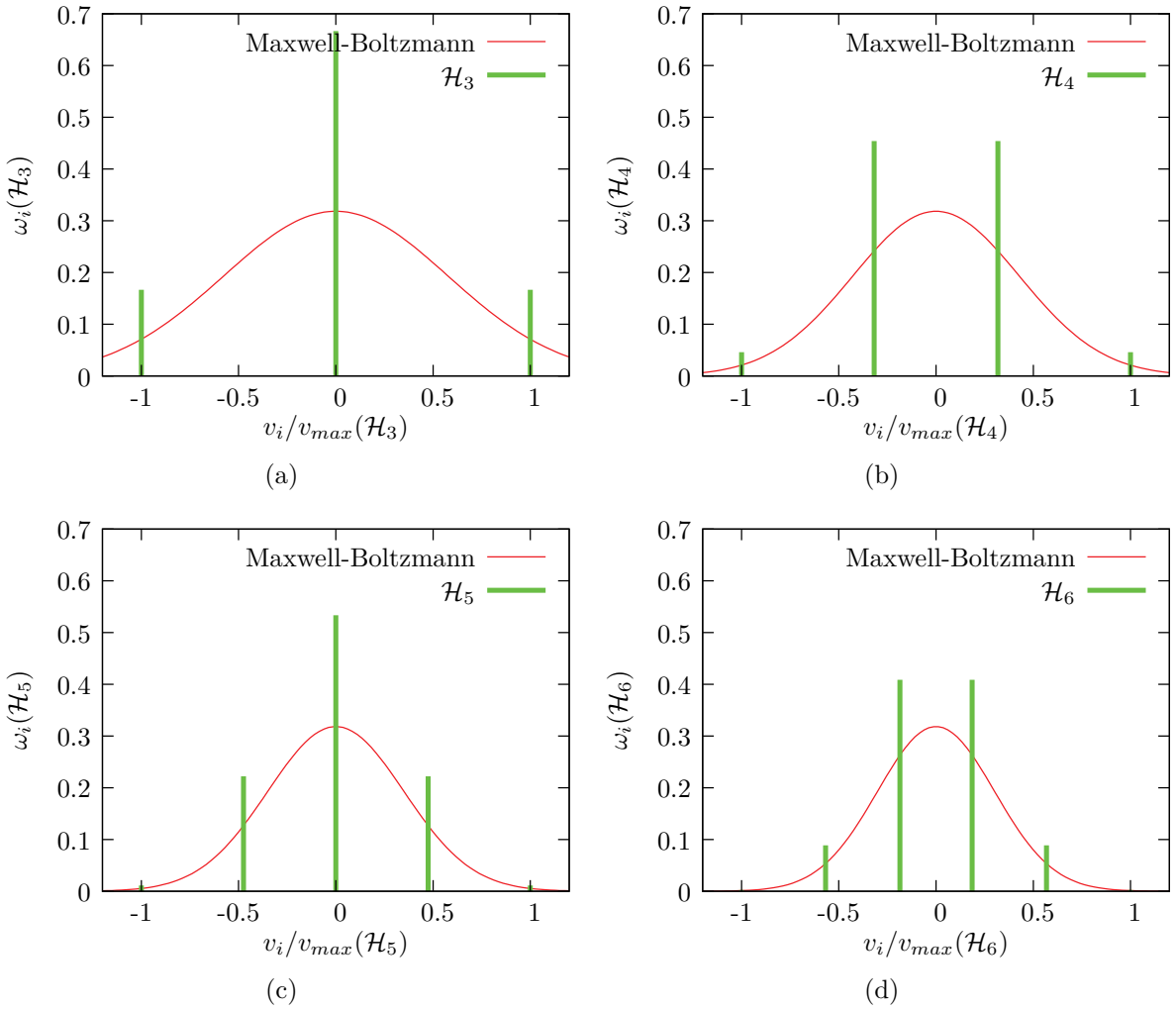


FIGURE 5.7 – Exemples de quadratures. Les vitesses discrètes, racines des polynômes d'Hermite $\mathcal{H}_n$, ont étés redimensionnées par rapport à la vitesse maximale $v_{max}(\mathcal{H}_n)$, de sorte que le critère CFL soit respecté par tous les faisceaux pour $\delta x = \delta t = 1$. La distribution de Maxwell-Boltzmann représentée sur chaque graphe a pour température redimensionnée $\hat{\theta}_0 = \theta_0/v^2_{max}(\mathcal{H}_n)$, θ_0 étant la température de référence du système. On remarque que les quadratures d'ordre impair, contrairement aux quadratures d'ordre pair, possèdent un faisceau de vitesse nulle dont le poids est le plus fort. Cette caractéristique a une influence, comme nous le verrons, sur les résultats produits par les schémas engendrés par ces différentes quadratures.

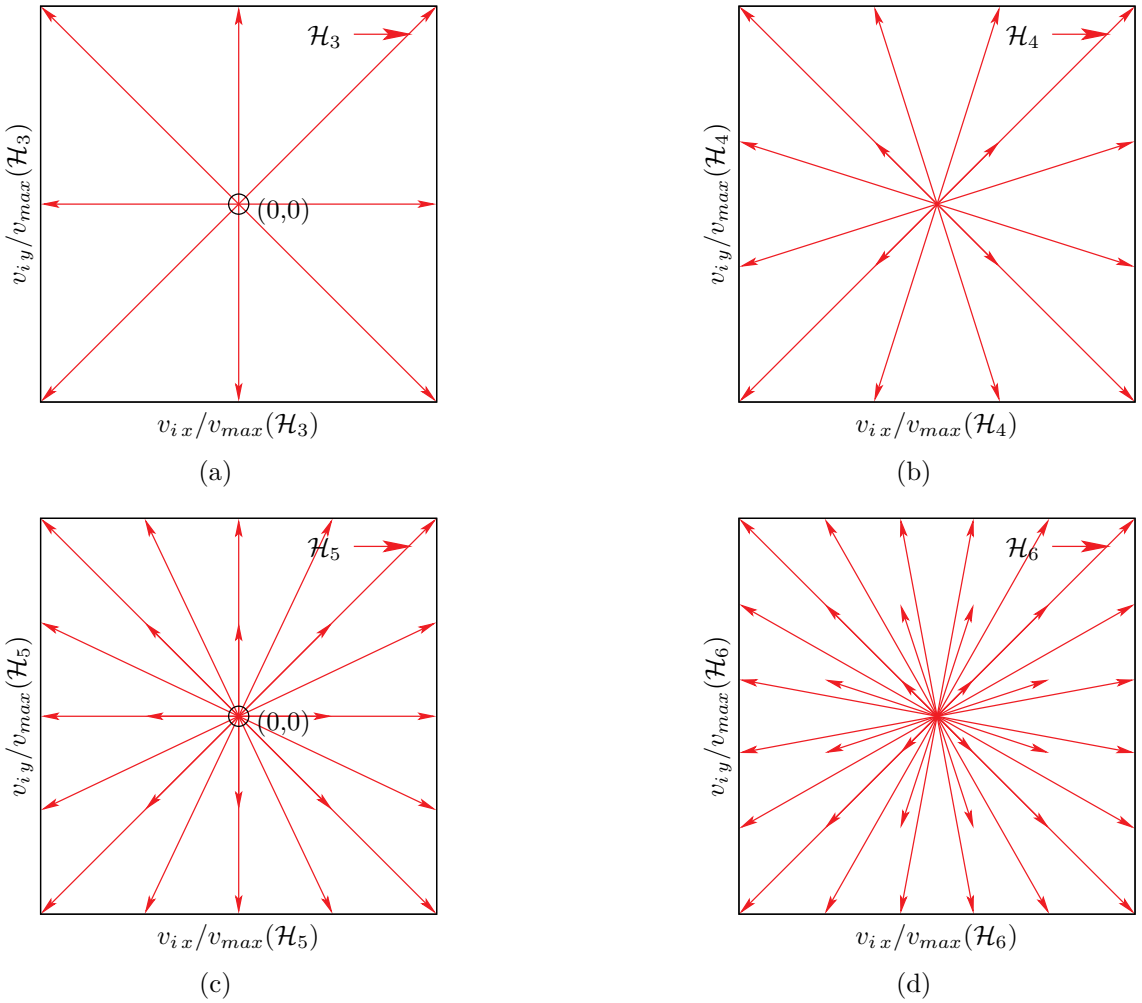


FIGURE 5.8 – Exemples de maillages de l'espace des vitesses en dimension 2. Les vitesses discrètes, racines des polynômes de Grad-Hermite $\boldsymbol{\mathcal{H}}_n$, ont été redimensionnées par rapport à la vitesse maximale $v_{max}(\mathcal{H}_n)$, de sorte que le critère CFL soit respecté par tous les faisceaux pour $\delta x = \delta t = 1$. Elles sont engendrées par les permutations des vitesses en une dimension, prises deux à deux. Comme dans le cas unidimensionnel, les quadratures d'ordre impair comportent un faisceau de vitesse nulle, dont le poids est le plus fort. On remarquera que le système de vitesses construit sur $\mathcal{H}_3$ est identique au D_2Q_9 des LBM.

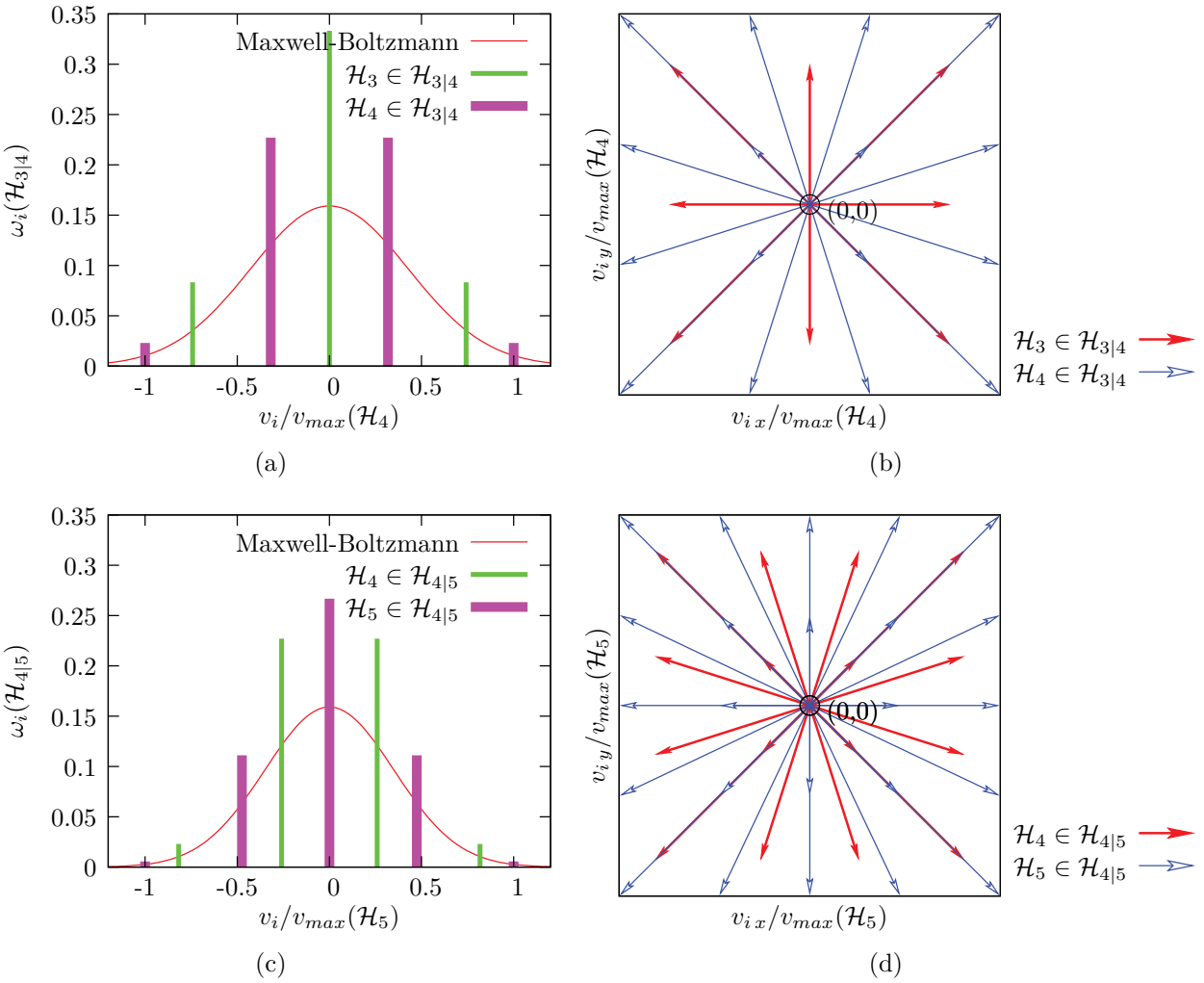


FIGURE 5.9 – Exemples de quadratures composites. Les vitesses discrètes, racines des polynômes d'Hermite $\mathcal{H}_n$ et $\mathcal{H}_{n+1}$, ont été redimensionnées par rapport à la vitesse maximale $v_{max}(\mathcal{H}_{n+1})$, de sorte que le critère CFL soit respecté par tous les faisceaux pour $\delta x = \delta t = 1$. On remarquera que les quadratures composites, à nombre de faisceaux équivalent, discrétisent plus finement le "gras" de la fonction de distribution de Maxwell-Boltzmann (comparer par exemple $\mathcal{H}_{3|4}$ Fig. (a), avec $\mathcal{H}_6$, Fig. 5.7(d)).

5.11 Schéma de transport

Les vitesses discrètes introduites par les quadratures de Gauss-Hermite n'étant pas rationnelles, nous allons être amené à considérer un schéma de transport plus complexe que celui adopté pour les schémas de type LBM, pour lesquels la fonction de distribution passe, en un pas de temps, d'un nœud du réseau à un autre.

Le problème abordé ici est celui de la résolution de la partie transport de l'équation de Boltzmann-BGK :

$$\partial_t f_i + \hat{v}_i \, \partial_x f_i = 0. \tag{5.97}$$

Les collisions ne sont pas considérées ici car le transport et les collisions sont découplés, ils peuvent donc être traités séparément.

Le temps et l'espace des configurations seront découpés régulièrement, avec des pas respectifs δt et δx. L'espace des vitesses est quant à lui discrétisé selon les vitesses $\boldsymbol{v}_i$, imposées par les quadratures présentées dans la section précédente. Nous avons posé $\hat{v}_i = v_i / v_{max}$ où v_{max} est la vitesse de plus grande amplitude. Les quadratures ayant pour nœuds les racines des polynômes d'Hermite, dont les valeurs sont irrationnelles, nous ne pourrons pas pratiquer le transport par un schéma de type automate cellulaire, et nous allons devoir interpoler la fonction de distribution entre les nœuds du réseau. De nombreux schémas permettent de réaliser cette tâche [13].

Commençons par présenter le plus simple de ces schémas, appelé *Upwind* et dont l'illustration est donnée figure 5.10. Il consiste simplement à interpoler le champ à transporter (ici la fonction de distribution), par une fonction affine. Il s'agit donc d'un schéma de transport du premier ordre. Sur la figure 5.10, la fonction de distribution discrète f_i est transportée à la vitesse $C_O = (|v_i|/v_{max}) \times (\delta x / \delta t)$ sur un pas de temps δt. L'objectif est de déterminer la valeur à affecter à la fonction de distribution au point x, après transport. Dans le cas où $C_O \delta t \neq \delta x$, cette valeur n'est pas connue, le schéma Upwind consiste donc à évaluer la fonction en $x - C_O \delta t$ par son interpolation au premier ordre, représentée par la droite sur la figure 5.10. Pour simplifier les écritures, nous poserons $\delta x = \delta t = 1$ à partir de maintenant ; ce choix, étant donné que $|\hat{v}_i| = |v_i|/v_{max} \leq 1$, permettra de respecter le critère CFL pour tous le faisceaux. Le schéma Upwind s'écrit alors de la façon suivante :

$$\begin{cases} f_i(x, t+1) = f_i(x,t) - |\hat{v}_i| \, [f_i(x,t) - f_i(x-1,t)] & \text{si } \hat{v}_i \geq 0 \\ f_i(x, t+1) = f_i(x,t) - |\hat{v}_i| \, [f_i(x,t) - f_i(x+1,t)] & \text{si } \hat{v}_i \leq 0 \end{cases} \tag{5.98}$$

Ce schéma n'est pas symétrique et la règle à appliquer dépendra du signe de la vitesse.

La même démarche avec une interpolation par un polynôme du second degré, né-

13. Les codes donnés en annexe C implémentent deux schémas de transport. La bibliothèque maBLB (annexe C.2) implémente les différents limiteurs de flux présentés ici. Dans les codes 1D (annexe C.1), le transport est réalisé par méthode spectrale (par transformée de Fourier). Le transport par transformée de Fourier n'est pas développé dans ce mémoire.

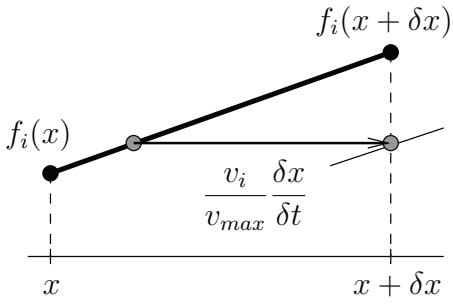


FIGURE 5.10 – Illustration du schéma de transport Upwind. La vitesse discrète v_i est une des racines du polynôme d'Hermite associé à la quadrature (cf. sous-section précédente) et v_{max} la vitesse discrète d'amplitude maximale, on a donc $|v_i|/v_{max} \leq 1$. Le rapport $\delta x/\delta t$ est la vitesse du réseau. Le produit $(|v_i|/v_{max}) \times (\delta x/\delta t)$, que l'on notera C_O le nombre de *Courant*, ce nombre permet de définir un critère de stabilité pour les schémas, le critère CFL : $C_O \leq 1$.

cessitant donc trois points, produit un schéma de transport au second ordre appelé *schéma de Lax-Wendroff*. Contrairement au Upwind, il est symétrique et s'écrit :

$$f_i(x,t+1) = f_i(x,t) + \frac{|\hat{v}_i|}{2}\left[f_i(x-1,t) - f_i(x+1,t)\right] + \frac{\hat{v}_i^2}{2}\left[f_i(x-1,t) - 2\,f_i(x,t) + f_i(x+1,t)\right]. \tag{5.99}$$

Cela ne sera pas montré dans ce mémoire, mais ni le schéma Upwind (Eq. (5.98)), ni celui de Lax-Wendroff (Eq. (5.99)) ne sont de qualité suffisante pour les problèmes que nous cherchons à résoudre. Les caractéristiques respectives de ces deux schémas sont une diffusion numérique importante pour le schéma Upwind, quant au schéma de Lax-Wendroff, il est très dispersif (il produit des ondes). Ces caractéristiques ont un effet différent selon la valeur du nombre de Knudsen de l'écoulement que l'on simule. Aux petits Knudsen, typiquement pour $K_n < 0{,}01$, le schéma de Lax-Wendroff montre une très bonne stabilité et permet d'obtenir de très bons résultats, par comparaison à la solution analytique pour l'écoulement de Poiseuille par exemple. Le schéma Upwind est quant à lui très instable aux petits Knudsen, la raison de cette instabilité est malheureusement mal comprise. Aux grands Knudsen, typiquement pour $K_n > 0{,}3$, la tendance s'inverse, en ce sens que le schéma de Lax-Wendroff devient très instable alors que le schéma Upwind se stabilise et donne de bons résultats. Cette observation s'explique simplement. Aux grands Knudsen, les collisions sont peu fréquentes et l'évolution de la fonction de distribution devient essentiellement un transport pur. Les défauts des schémas de transport sont donc libres de se manifester, en particulier, les oscillations produites par le schéma de Lax-Wendroff ont la possibilité de se développer, rendant la simulation instable. *A contrario*, la diffusion numérique introduite par le schéma Upwind tend à stabiliser la simulation, en introduisant en quelque sorte des collisions. Partant des observations aux grands

Knudsen, dont l'interprétation est simple, nous pouvons proposer des explications sur les comportements observés aux petits Knudsen. On peut en effet supposer que la stabilité du schéma de Lax-Wendroff y est améliorée par les collisions qui tendent à détruire les oscillations produites par la dispersion. On peut enfin supposer que la diffusion introduite par le schéma Upwind, assimilable à des collisions, rend la simulation instable en faisant que la viscosité effective du fluide simulé soit nulle, voire négative, car la viscosité dépend du temps de relaxation vers l'équilibre.

Au vu des qualités et défauts des deux schémas de transport discutés précédemment, nous allons nous orienter vers une classe de schémas de transports qui les combine de façon judicieuse : *les limiteurs de flux*. Nous n'entrerons pas dans les détails du développement de ces schémas et n'en donnerons que l'expression, qui est la suivante :

$$\begin{aligned} f_i(x,t+1) &= f_i(x,t) - |\hat{v}_i| \left[f_i(x-1,t) - f_i(x+1,t) \right] \\ &- \frac{|\hat{v}_i|}{2}(1-|\hat{v}_i|) \left\{ \psi(r_a) \left[f_i(x+1,t) - f_i(x,t) \right] - \psi(r_b) \left[f_i(x,t) - f_i(x-1,t) \right] \right\}, \end{aligned} \tag{5.100}$$

où

$$r_a = \frac{f_i(x,t) - f_i(x-1,t)}{f_i(x+1,t) - f_i(x,t)}, \tag{5.101}$$

$$r_b = \frac{f_i(x-1,t) - f_i(x+2,t)}{f_i(x,t) - f_i(x-1,t)} \tag{5.102}$$

et où la fonction ψ s'appelle le limiteur de flux.

Dans la sous-section suivante, nous allons, grâce à quelques tests, déterminer quelle fonction ψ, donc quel schéma limiteur de flux est le plus performant pour résoudre notre problème. Pour plus de détails sur les schémas de transport, voir par exemple [Tor99].

Choix du limiteur de flux

Trois limiteurs de flux sont testés ici. Le cas test choisi est un écoulement de Poiseuille plan, mis en mouvement par une force extérieure, problème que l'on étudiera par ailleurs au chapitre suivant (section 6.3). Le principe est ici de comparer les résultats obtenus avec la méthode de Boltzmann sur réseau, basée sur la quadrature $\mathcal{H}_{4|5}$, pour différentes discrétisations de l'espace (maillage plus ou moins fin) et pour différents nombres de Knudsen.

Les limiteurs de flux testés sont les suivants :
- Superbee : $\psi(r) = max(0, max(min(2\,r,1), min(r,2))$,
- Koren : $\psi(r) = max(0, min(min(2\,r,(2+r)/3), 2)$,
- Minmod : $\psi(r) = max(0, min(r,1))$.

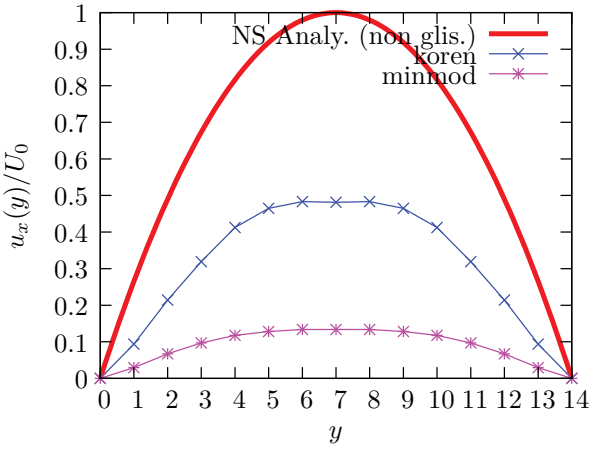
Les résultats obtenus, pour des nombres de Knudsen compris entre 0,001 et 8,1924, sont représentés figures 5.11 et 5.12. Le nombre de points de discrétisation L, étant de 15 pour la colonne de gauche et de 40 pour les figures de la colonne de droite.

Ces résultats montrent que pour les grands Knudsen (Fig. 5.12), ces trois limiteurs de flux donnent des résultats quasi identiques pour les deux discrétisations $L = 15$ et $L = 40$.

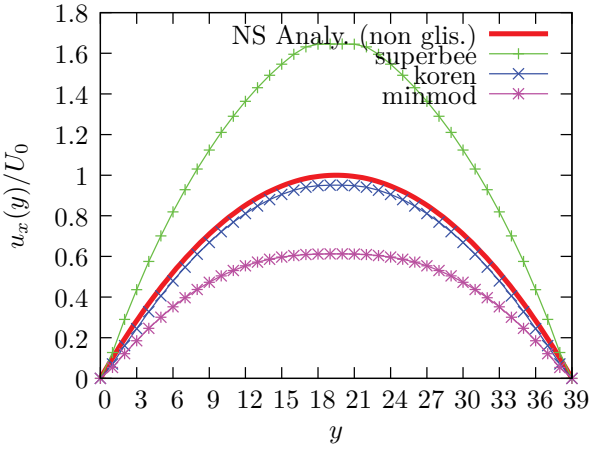
Pour un nombre de Knudsen plus petit et correspondant au régime de glissement (Fig. 5.11(e)-(f)), une légère différence apparaît entre les limiteurs de flux, mais seulement pour la discrétisation la plus grossière.

Des différences marquées n'apparaissent que dans le régime hydrodynamique, (Fig. 5.11(a)-(d)). Notamment, pour la discrétisation la plus fine, pour $L = 40$, le limiteur Koren se montre le plus performant, en particulier pour $K_n = 0{,}001$. Concernant la discrétisation grossière, avec $L = 15$ et pour $K_n = 0{,}001$, aucun des limiteurs n'est bon et Superbee se montre instable (c'est pourquoi la solution n'est pas représentée figure 5.11(a) pour ce limiteur). Ces observations nous permettent donc de conclure que le limiteur Koren est le plus adapté à notre problème.

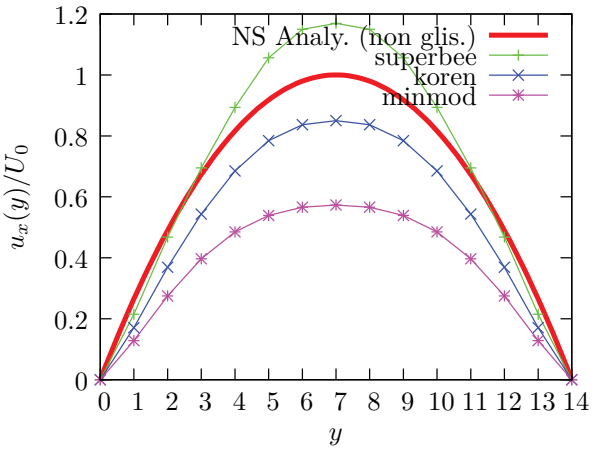
Il faudra donc simplement veiller, lors de la conception d'une simulation, que la discrétisation de la plus petite structure confinente soit suffisamment fine, selon la valeur locale du Knudsen. Le maillage de l'espace des configuration étant régulier, les grandes structures, dans lesquelles le Knudsen sera plus petit, seront naturellement discrétisées par un plus grand nombre de points.



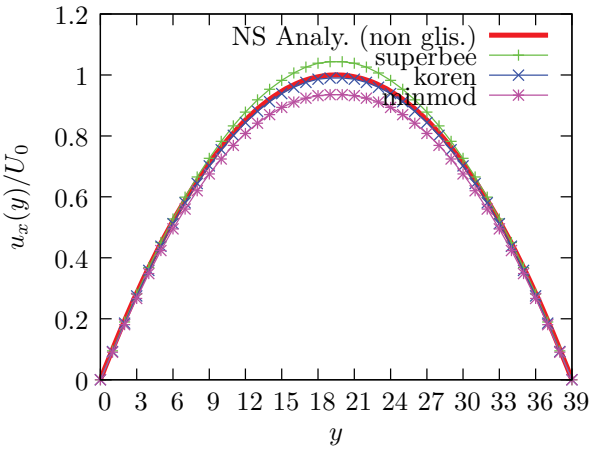
(a) $K_n = 0{,}001$, $L = 15$



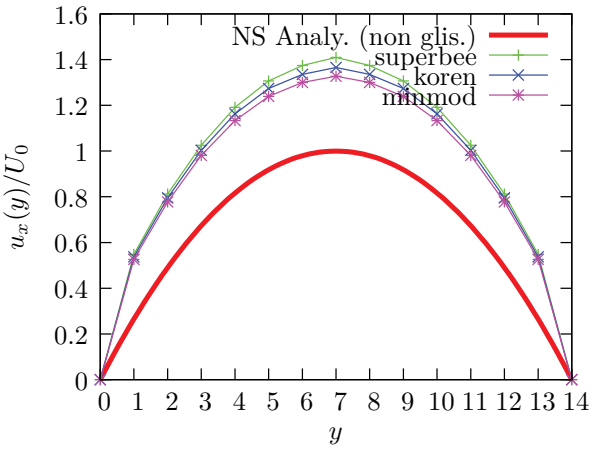
(b) $K_n = 0{,}001$, $L = 40$



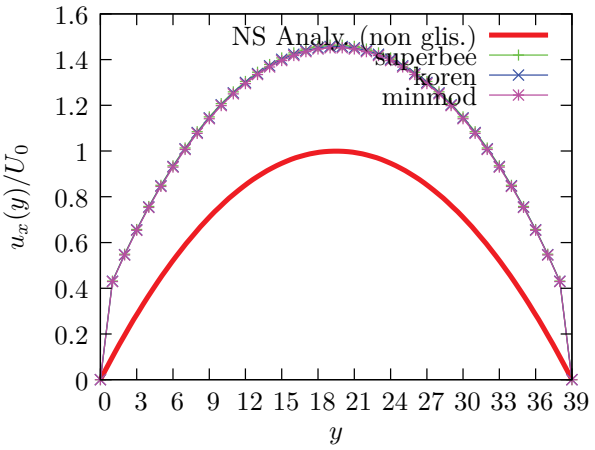
(c) $K_n = 0{,}008$, $L = 15$



(d) $K_n = 0{,}008$, $L = 40$

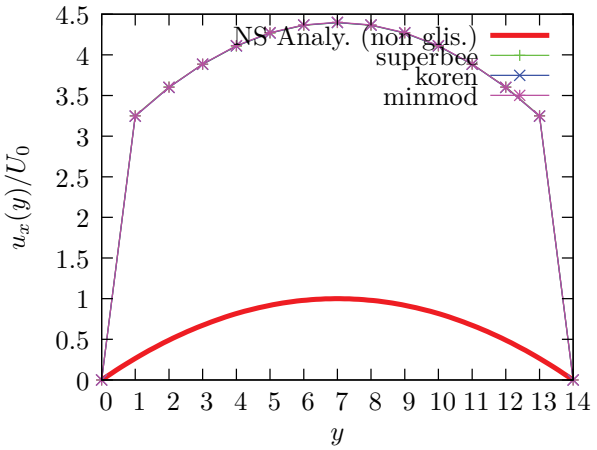


(e) $K_n = 0{,}128$, $L = 15$

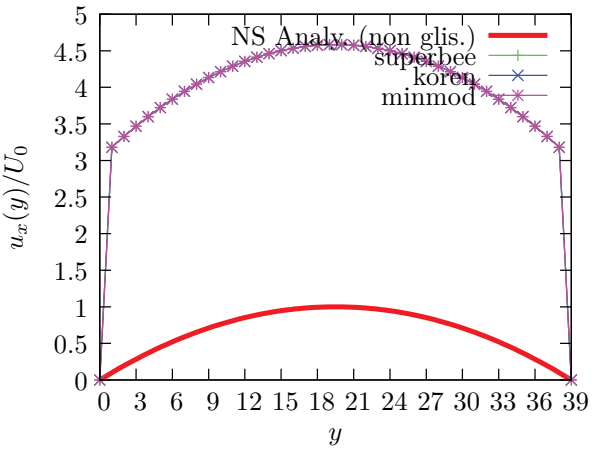


(f) $K_n = 0{,}128$, $L = 40$

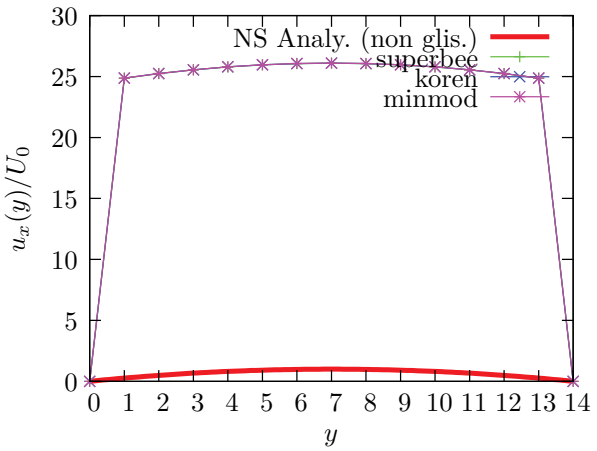
FIGURE 5.11 – Comparaison des champs de vitesses obtenus par les limiteurs de flux superbee, koren et minmod sur Poiseuille plan. À chaque ligne de graphes correspond une valeur du nombre de Knudsen et des discrétisations de 15×4 et 40×4 pour les colonnes de gauche et de droite, respectivement. (**Suite à la figure 5.12.**)



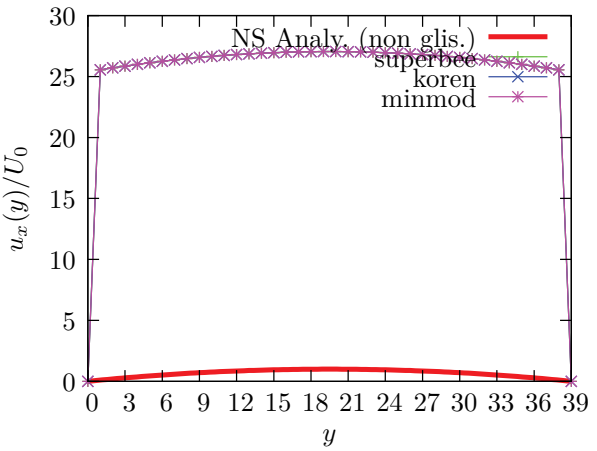
(a) $K_n = 1{,}024$, $L = 15$



(b) $K_n = 1{,}024$, $L = 40$



(c) $K_n = 8{,}1924$, $L = 15$



(d) $K_n = 8{,}192$, $L = 40$

FIGURE 5.12 – (**Suite de la figure 5.11**)

6 Validation et applications de la méthode

Dans ce chapitre, nous allons appliquer la méthode développée dans le chapitre précédent à différents problèmes [1]. Parmi les problèmes abordés, certains nous serviront simplement à valider l'aptitude des schémas d'une part, à capturer les effets de raréfaction et d'autre part, à représenter la conduction de la chaleur. Ces validations seront effectuées de façon systématique, pour des schémas basés sur des quadratures de différents ordres, pures et composites. En fait, nous verrons que l'idée de construire les schémas composites découle des comportements observés lors de l'application des schémas (*a posteriori*) purs.

Nous commencerons par l'étude d'un gaz unidimensionnel et périodique [2]. L'étude de ce système très simple nous permettra de considérer, en premier lieu, le cas limite pour lequel les molécules du gaz ne subissent aucune collision. Ce problème permettra d'observer le comportement de la méthode dans le cas limite des particules libres. En second lieu, toujours en dimension 1, nous passerons à la validation de la thermique, en particulier, nous vérifierons que la diffusivité thermique du gaz simulé est bien celle que nous souhaitons.

Nous passerons ensuite à des problèmes à deux dimensions. En commençant par l'étude d'un gaz placé entre deux plans parallèles et infinis. Le domaine d'intégration sera périodique dans une direction de l'espace et délimité par des bords cinétiques dans l'autre direction. Ce problème nous permettra dans un premier temps de vérifier que le saut de température à la paroi (dépendant du nombre de Knudsen) est bien capturé par la méthode. Dans ce cas les deux parois seront maintenues à des températures différentes. Cette même géométrie nous permettra ensuite d'étudier l'écoulement de Poiseuille, mû par une force extérieure. Nous procéderons alors de même, par une étude systématique des résultats obtenus par les différents schémas. Nous regarderons en priorité les valeurs du débit et la vitesse de glissement à la paroi, en fonction du nombre de Knudsen. Une fois le schéma le plus performant déterminé, nous l'utiliserons pour calculer des profils des champs de vitesse et de température de l'écoulement de Poiseuille, pour différentes valeurs du nombre de Knudsen.

Finalement, ce chapitre ce terminera par l'étude d'un écoulement à travers un long canal, mû par un gradient de densité. Nous verrons par cet exemple que le nombre de Knudsen peut varier au point que l'écoulement change de régime, non seulement

1. Les codes de calcul utilisés pour résoudre ces problèmes sont donnés en annexe C de ce mémoire.

2. Une mise en œuvre de la méthode en dimension 1 est donnée en annexe C.1. Le transport n'est pas réalisé par un schéma limiteur de flux, mais par méthode spectrale.

pour des raisons géométriques, mais aussi pour des raisons thermodynamiques (ici les variations de densité le long du canal).

6.1 Gaz de Knudsen unidimensionnel

On propose en premier lieu d'étudier, en dimension 1, la relaxation d'une perturbation gaussienne de la densité dans un gaz sans collision, aussi appelé *gaz de Knudsen*[3]. Cette situation limite est celle rencontrée à l'intérieur de la couche de Knudsen, au sein de laquelle les molécules du gaz ne collisionnent quasiment pas. Rappelons que la couche de Knudsen est présente à proximité des surfaces solides qui délimitent le fluide. Le gaz présent à l'intérieur de cette couche provient donc en partie de la masse du fluide et en partie de la surface solide ; surface sur laquelle la fonction de distribution est thermalisée. L'exercice proposé ici aura pour but de voir dans quelle mesure différents schémas, bâtis sur des quadratures d'ordre différents, sont susceptibles de décrire la partie de la fonction de distribution qui se propage à l'intérieur de la couche de Knudsen en provenance de la paroi. Nous supposerons donc que la fonction de distribution est initialement à l'équilibre et que celle-ci se propage librement, *i.e.* sans collision. De plus, nous supposerons qu'à la fonction de distribution initiale correspond une distribution de densité gaussienne. Cette condition initiale aura l'avantage de donner au problème une solution analytique, à laquelle nous pourrons comparer les résultats numériques.

Partons de l'équation de Boltzmann sans collision :

$$\partial_t f + v\,\partial_x f = 0, \tag{6.1}$$

dont la solution à l'instant t est la solution caractéristique :

$$f(x, v, t) = f(x - v\,t, v, 0). \tag{6.2}$$

Considérons alors la distribution d'ETG de Maxwell-Boltzmann :

$$w_0(v) = \frac{\rho_0}{\sqrt{2\pi\, c_0^2}}\, \exp\left(-\frac{v^2}{2\, c_0^2}\right), \tag{6.3}$$

où c_0 est la vitesse thermique et ρ_0 la densité de référence, que nous prendrons telles que $\rho_0 = 1$ et $c_0 = 1$, d'où :

$$w_0(v) = \frac{1}{\sqrt{2\pi}}\, \exp\left(-\frac{v^2}{2}\right). \tag{6.4}$$

Considérons alors dans l'espace des configurations, une perturbation gaussienne,

3. Les résultats présentés ici pourront être reproduits grâce au code scilab "*Gaz_Kn.sci*" donné en annexe C.1.2.

d'amplitude $\alpha \ll 1$ et de largeur unité :

$$f(x,v,t=0) = w_0(v)\left[1+\alpha\,\exp\left(-\frac{x^2}{2}\right)\right]. \tag{6.5}$$

La fonction de distribution (6.4) sera utilisée comme condition initiale. En utilisant la solution (6.2), on obtient alors pour f, au temps t :

$$f(x,v,t) = w_0(v)\left[1+\alpha\,\exp\left(-\frac{(x-v\,t)^2}{2}\right)\right]. \tag{6.6}$$

On ne s'intéressera ici qu'aux trois premiers moments de f à savoir, la densité $\rho(x,t)$, l'impulsion $\rho(x,t)\,u(x,t)$ et l'énergie totale $\rho(x,t)\,\varepsilon(x,t)$, définis par les intégrales suivantes :

$$\begin{aligned}
\rho(x,t) &= \int f(x,v,t)\,dv, && (6.7)\\
\rho(x,t)\,u(x,t) &= \int v\,f(x,v,t)\,dv, && (6.8)\\
\rho(x,t)\,\varepsilon(x,t) &= \int \frac{1}{2}|v-u(x,t)|^2\,f(x,v,t)\,dv && (6.9)\\
&= \rho(x,t)\,u^2(x,t)+\rho(x,t)\,\theta(x,t). && (6.10)
\end{aligned}$$

Le calcul de ces intégrales permet alors d'exprimer les champs macroscopiques de densité $\rho(x,t)$, de vitesse $u(x,t)$ et de température $\theta(x,t)$, donnés à l'instant initial par :

$$\begin{aligned}
\rho(x,0) &= 1+\alpha\,\exp\left(-\frac{x^2}{2}\right), && (6.11)\\
u(x,0) &= 0, && (6.12)\\
\theta(x,0) &= 1, && (6.13)
\end{aligned}$$

dont on se servira pour initialiser la fonction de distribution d'équilibre $w_0(v)$ (Eq. 6.4).

Finalement, pour $t>0$, ces champs suivent les équations d'évolution suivantes :

$$\begin{aligned}
\rho(x,t) &= 1+\alpha\,\frac{1}{\sqrt{1+t^2}}\,\exp\left(-\frac{x^2}{2\,(1+t^2)}\right), && (6.14)\\
u(x,0) &= \frac{1}{\rho(x,t)}\alpha\,\frac{x\,t}{\sqrt{(1+t^2)^3}}\,\exp\left(-\frac{x^2}{2\,(1+t^2)}\right), && (6.15)\\
\theta(x,t) &= \frac{1}{\rho(x,t)}+\frac{\alpha}{\rho(x,t)}\frac{1+t^2+(x\,t)^2}{\sqrt{(1+t^2)^5}}\,\exp\left(-\frac{x^2}{2\,(1+t^2)}\right)\\
&\quad -u^2(x,t). && (6.16)
\end{aligned}$$

Les figures 6.1 et 6.3 représentent des résultats de simulations de ce problème obtenus grâce au programme donné en annexe C.1.2. La longueur de l'espace est $L = 50$ et le nombre de nœuds est $N = 2^8$. La figure 6.1 compare les résultats obtenus pour trois quadratures "pures", $\mathcal{H}_3$, $\mathcal{H}_6$ et $\mathcal{H}_9$ aux solutions analytiques (6.14)-(6.16). Les champs de densité, de vitesse moyenne et de température ont été représentés à deux instants différents. On remarque aisément que l'évolution est simulée correctement sur des temps d'autant plus long que l'ordre de la quadrature est élevé, *i.e.* que le nombre de vitesses discrètes est grand. En outre, on remarque que les erreurs apparaissent sous forme d'oscillations autour des solutions théoriques. Ces oscillations s'expliquent par la "séparation" des différents faisceaux transportant chacun une partie de la densité, à sa propre vitesse. Chaque faisceaux transporte en effet simplement, comme il n'y a pas de collision, la fonction d'équilibre discrète initiale (cf. Eq. (5.50) section 5.5) :

$$f_i^{eq}(x,t=0) = \omega_i\,\rho(x,t=0), \tag{6.17}$$

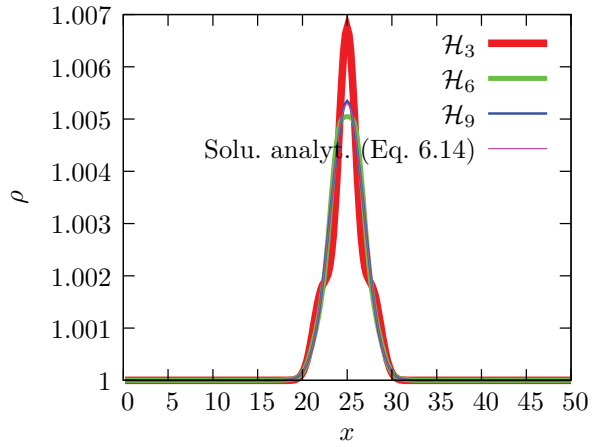
à la vitesse discrète v_i.

Ceci explique l'apparition, pour des temps suffisamment longs, d'autant de gaussiennes dans le champ de densité, que de faisceaux. L'amplitude de la gaussienne transportée à vitesse v_i étant $(1+\alpha)\,\omega_i$. En particulier, pour les quadratures d'ordre impair, comportant un faisceau de vitesse nulle (dont le poids ω_i est le plus fort), une gaussienne reste fixe, comme cela s'observe pour les solutions de $\mathcal{H}_3$ et $\mathcal{H}_9$. Au contraire, la solution obtenue avec $\mathcal{H}_6$ présente un creux à cette position, ce qui est nettement visible sur le graphe (b) (Fig. 6.1). Cette observation nous amène à représenter l'évolution dans le temps de la densité en ce point $x = L/2$, où se trouve le maximum de la gaussienne initiale. La figure 6.2 donne l'évolution temporelle de la densité $\rho(L/2,t)$ obtenue pour les quadratures $\mathcal{H}_{n=3\ldots8}$, ainsi que la solution analytique. Deux choses sont remarquables. La première est que l'accroissement de l'ordre de la quadrature permet de représenter l'évolution de la densité sur des temps de plus en plus longs, avec cependant une augmentation très lente du temps de "décrochage" à la courbe théorique avec l'accroissement de l'ordre de la quadrature. La seconde chose remarquable est la manière dont les schémas d'ordres successifs convergent vers la solution analytique. Il apparaît en effet une *convergence alternée* des schémas vers la solution théorique, avec une surestimation systématique par les schémas d'ordres impairs et une sous-estimation par les schémas d'ordres pairs. Nous verrons plus bas, lors de l'étude de l'écoulement de Poiseuille, que les courbes de débits en fonction du nombre de Knudsen présentent une convergence du même type. Signalons enfin pour clore ce point, que l'évolution du champ de température présente le même comportement que celui de la densité.

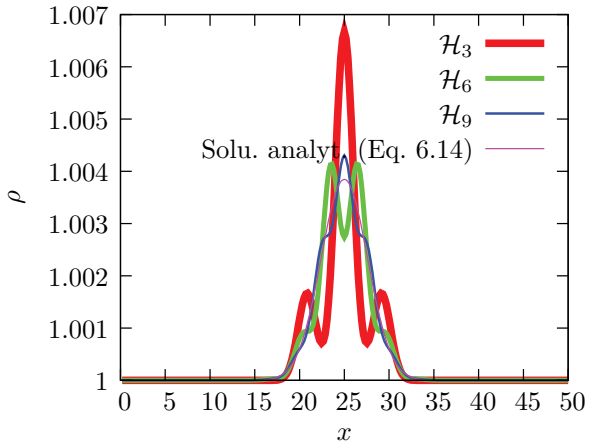
La convergence alternée, observée ici, nous a donné l'idée de tenter de compenser la sous-estimation de la densité et de la température au point $x = L/2$ des schémas pairs, par la surestimation des schémas impairs, en combinant deux schémas d'ordres successifs, donc de parités différentes. Ces schémas sont à la base des quadratures composites présentées section 5.10.1. Regardons ce qu'ils donnent sur le

problème précédent. La figure 6.4 représente l'évolution de la densité au maximum de la perturbation, obtenu avec le schéma basé sur la quadrature composite $\mathcal{H}_{4|5}$. Cette quadrature composite, représentée figure 5.9(c), est composée de 9 vitesses discrètes et nous comparerons les résultats obtenus en l'utilisant avec ceux obtenus avec les quadratures pures $\mathcal{H}_7$ et $\mathcal{H}_8$. Le nombre de vitesses discrètes de ces trois schémas étant comparable, il en sera de même des efforts numériques à réaliser pour obtenir les solutions. On remarque alors que le résultat obtenu avec $\mathcal{H}_{4|5}$ suit beaucoup plus longtemps la solution théorique que ceux obtenus avec $\mathcal{H}_7$ et $\mathcal{H}_8$. Le résultat de $\mathcal{H}_{4|5}$ n'est cependant pas parfait car des erreurs apparaissent plus tôt, aux temps où les schémas purs $\mathcal{H}_4$ et $\mathcal{H}_5$ qui composent $\mathcal{H}_{4|5}$ "décrochent" de la courbe théorique (cf. Fig. 6.2).

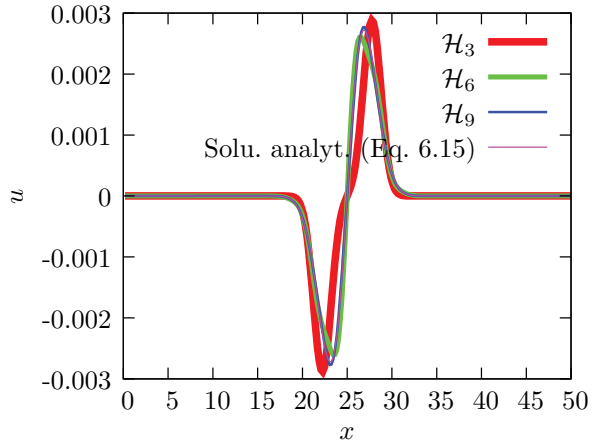
Regardons ce qu'il en est des champs de densité, de vitesse et de température obtenus avec les schémas basés sur $\mathcal{H}_{4|5}$ et $\mathcal{H}_9$, composés chacun de 9 faisceaux. Les résultats sont représentés figure 6.3, aux instants $t = 2{,}3$ (colonne de gauche, Pour une comparaison plus complète, la figure 6.1 donne les solutions obtenues au même instant avec les schémas purs) et $t = 4$ (colonne de droite). L'analyse de ces courbes permet de voir que pour des temps moyennement longs ($t = 2{,}3$, Fig. 6.3 colonne de gauche), les schémas basés sur $\mathcal{H}_{4|5}$ et $\mathcal{H}_9$ sont de qualités similaires. L'observation des solutions aux temps longs $t = 4$ (Fig. 6.3 colonne de droite), permet par contre de les différencier. On observe en effet que les oscillations des solutions autour des solutions théoriques (6.14)-(6.16), sont d'amplitudes plus faibles pour $\mathcal{H}_{4|5}$ que pour $\mathcal{H}_9$.



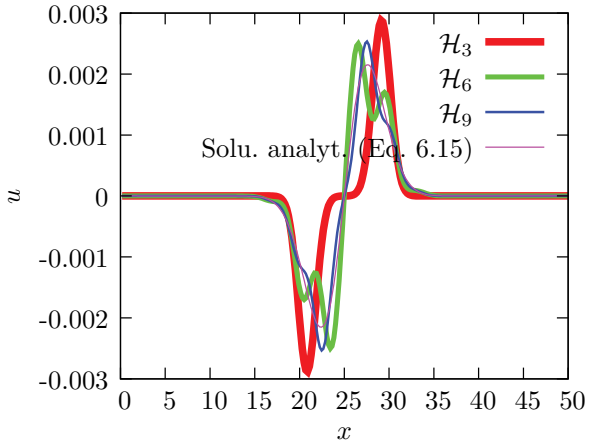
(a) Densité à l'instant $t = 1{,}5$.



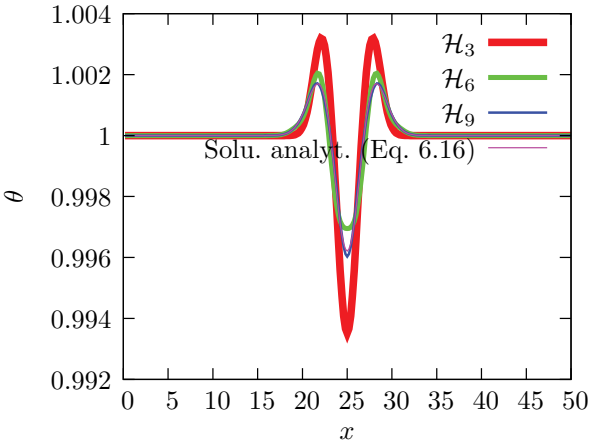
(b) Densité à l'instant $t = 2{,}3$.



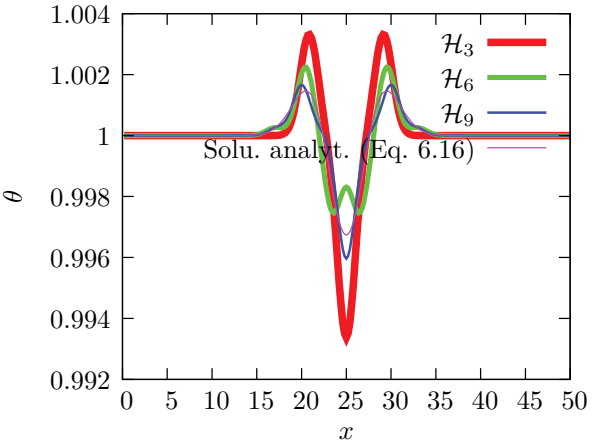
(c) Vitesse moyenne à l'instant $t = 1{,}5$.



(d) Vitesse moyenne à l'instant $t = 2{,}3$.



(e) Température à l'instant $t = 1{,}5$.



(f) Température à l'instant $t = 2{,}3$.

FIGURE 6.1 – Simulation de la relaxation d'une perturbation gaussienne de la densité pour le gaz de Knudsen. Les résultats montrent que l'augmentation du nombre de vitesses discrètes améliore la solution numérique.

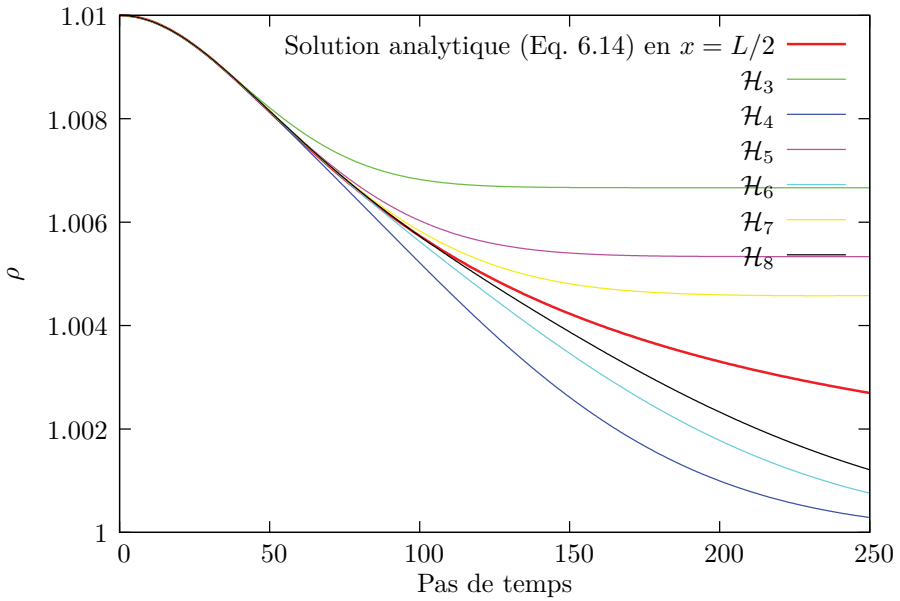
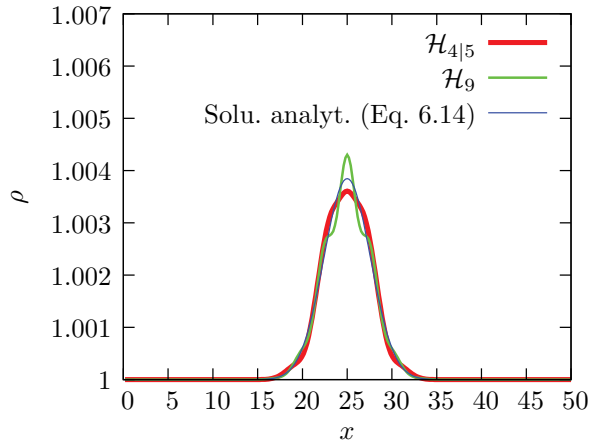
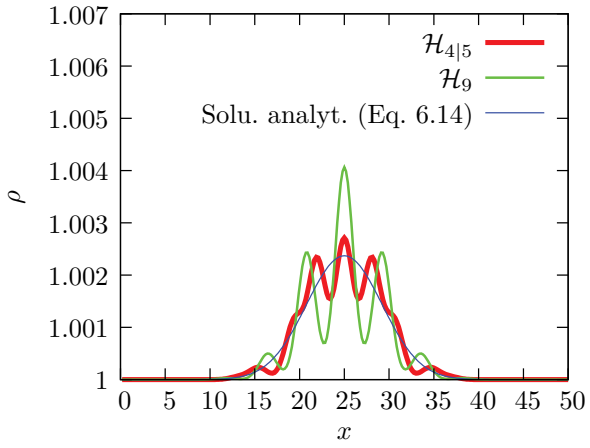


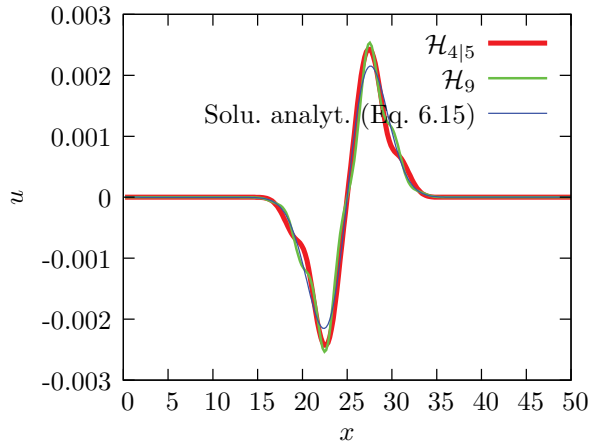
FIGURE 6.2 – Valeur de la densité $\rho(L/2,t)$ en fonction du temps discret en $x = L/2$, position où la densité est maximale.



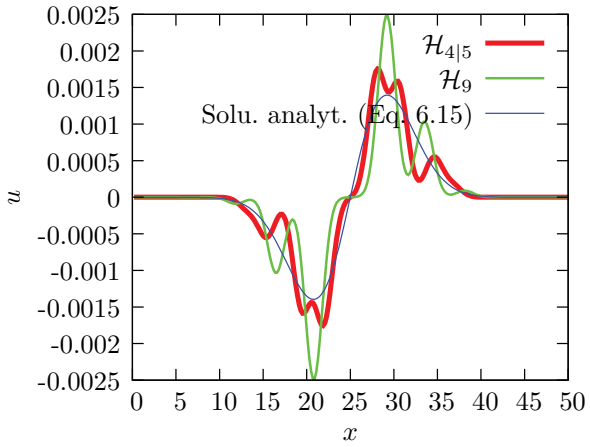
(a) Densité à l'instant $t = 2{,}3$.



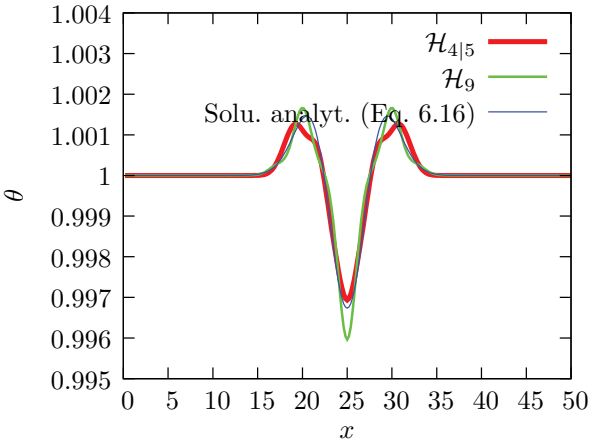
(b) Densité à l'instant $t = 4$.



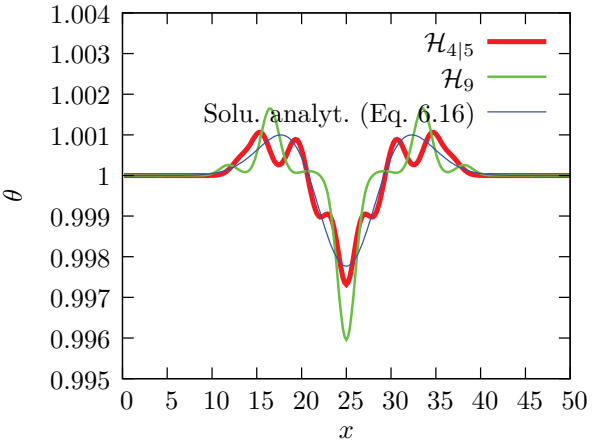
(c) Vitesse moyenne à l'instant $t = 2{,}3$.



(d) Vitesse moyenne à l'instant $t = 4$.



(e) Température à l'instant $t = 2{,}3$.



(f) Température à l'instant $t = 4$.

FIGURE 6.3 – Comparaison du schéma composite $\mathcal{H}_{4|5}$ avec le schéma basé sur $\mathcal{H}_9$. Tous les schémas comportent 9 vitesses discrètes.

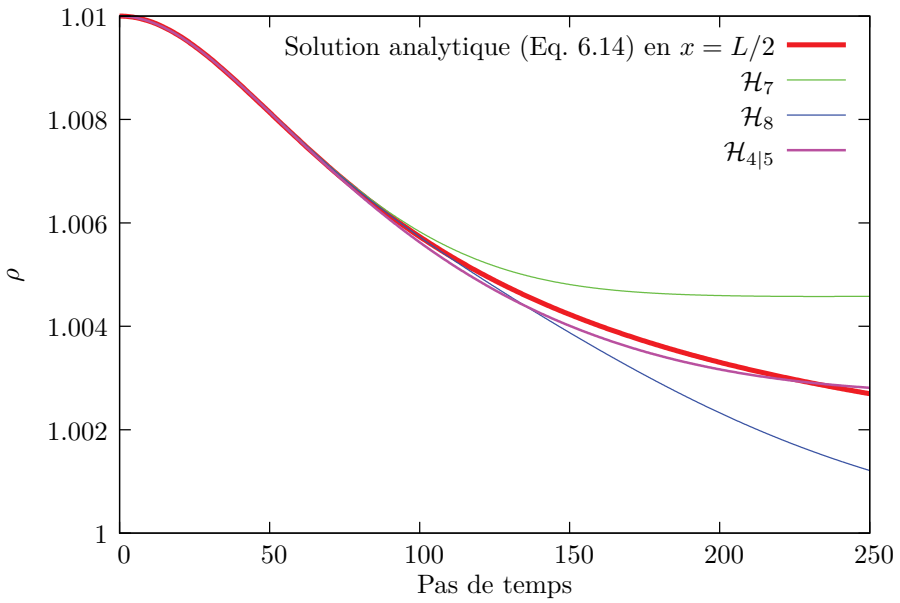


FIGURE 6.4 – Valeur de la densité $\rho(L/2,t)$ en fonction du temps discret en $x = L/2$, position où la densité est maximale. Comparaison de la quadrature composite $\mathcal{H}_{4|5}$ avec deux quadratures “pures” d’ordres équivalents.

6.2 Validations de la thermique

Nous allons nous pencher ici sur les processus thermiques, en commençant par vérifier que la diffusivité thermique α du gaz simulé, dans la limite hydrodynamique (*i.e.* pour $\tau \ll 1$), est bien reliée au temps de relaxation vers l'équilibre τ par la relation $\alpha = \theta\tau$. Finalement, nous vérifierons que le saut de température à la paroi en fonction du nombre de Knudsen, suit bien la relation discutée section 4.4.1.

6.2.1 Conductivité thermique

Rappelons que, dans la limite non collisionelle (*i.e.* pour $\tau \to \infty$), nous avons pu constater (cf. section 6.1) que l'évolution du champ de température pouvait être représentée correctement par les schémas de Boltzmann sur réseau proposés dans cette thèse. Nous avons en particulier observé que l'évolution des champs macroscopiques pouvait être représentée sur des temps d'autant plus longs que l'ordre de la quadrature était élevé.

Dans cette section, nous allons vérifier que, dans la limite hydrodynamique, *i.e.* pour $\tau \ll 1$, ces schémas représentent bien un gaz dont la diffusivité thermique est telle que $\alpha = \theta\tau$. Pour vérifier cette propriété, nous proposons de considérer un problème unidimensionnel [4] similaire à celui traité section 6.1 pour le gaz sans collision.

La méthode de validation proposée est de résoudre simultanément la diffusion d'un champ de température non uniforme, par la méthode développée ici et par la résolution de l'équation de la chaleur :

$$\partial_t \theta = \Delta\left(\alpha\,\theta\right), \tag{6.18}$$

en faisant en sorte que la pression soit uniforme, de façon à éviter tout transport d'énergie interne par advection. Alors, si la diffusivité thermique est bien telle que $\alpha = \theta\tau$, les solutions obtenues par ces deux méthodes devront être les mêmes.

La résolution de l'équation (6.18) sera réalisée par le schéma aux différences finies :

$$\theta(x,t+\delta t) = \theta(x,t) + \alpha(x,t)\frac{\delta t}{\delta x^2}\left[\theta(x+\delta x,t) - 2\,\theta(x,t) + \theta(x-\delta x,t)\right], \tag{6.19}$$

où le coefficient de diffusion thermique $\alpha(x,t)$ est local. Il dépend de $\rho(x,t)$ et $\theta(x,t)$ par la relation :

$$\alpha(x,t) = \theta(x,t)\frac{\rho_0}{\rho(x,t)}\left(\frac{\theta_0}{\theta(x,t)}\right)^{1/2}\tau_0, \tag{6.20}$$

où τ_0 est le temps de relaxation défini à la densité ρ_0 et à la température θ_0 de référence.

4. Les résultats présentés ici pourront être reproduits grâce au code scilab "*boltzmannBGK.sci*" donné en annexe C.1.3.

Nous choisirons d'imposer comme champ de température initial, une perturbation gaussienne d'amplitude η et de largeur à mi-hauteur égale à 2 autour de la température de référence θ_0 :

$$\theta(x,t=0) = \theta_0 \left(1 + \eta \exp\left(-\frac{(x - L/2)^2}{4}\right)\right), \tag{6.21}$$

où $L = 30$, est la longueur de l'espace d'intégration.

Pour que la pression initiale $p_0(x) \equiv p_0$ soit uniforme et telle que $p_0 = \rho_0\,\theta_0 = 1$, le champ de densité initial sera pris tel que :

$$\rho(x,0) = (\theta(x,0))^{-1}. \tag{6.22}$$

La simulation de ce problème en utilisant diverses quadratures pures (Fig. 6.5) et composites (Fig. 6.6) et une fonction de distribution d'équilibre développée à l'ordre $N = 2$, permet de montrer que la diffusivité thermique est bien donnée par $\alpha = \theta\tau$. On vérifie bien que ceci n'est vrai que pour les quadratures pures $\mathcal{H}_n$ et les quadratures composites $\mathcal{H}_{n|n+1}$ pour $n > 3$, et pour une fonction de distribution d'équilibre f_i^{eq} :

$$f_i^{eq}(x,t) = \omega_i \sum_{n=0}^{N} \frac{1}{n!}\, a_n^{eq}(x,t)\, \mathcal{H}_n^{[\theta_0]}(v_i), \tag{6.23}$$

développée à un ordre $N \geq 2$ (cf. section 5.5).

En effet, il a par ailleurs été montré de manière analytique, que le premier de nos schémas à supporter la thermique est $\mathcal{H}_4$ [KP08]. Il est d'autre part connu que les LBM, D_2Q_9 et D_3Q_{27}, équivalant aux schémas construits sur $\mathcal{H}_3$ en 2 et 3 dimensions ne supportent pas la thermique [AKA+07]. Par ailleurs, on remarque ici que ceci reste vrai, même si le développement de la fonction de distribution d'équilibre est fait jusqu'à l'ordre 2 (et donc réintroduit la température θ). Ce résultat montre que ces schémas à trois faisceaux par direction ne peuvent pas rendre compte du flux de chaleur. Nous aurions pu espérer que le schéma résultant de la combinaison de $\mathcal{H}_3$ et $\mathcal{H}_4$, comportant 7 faisceaux par direction, n'hérite pas du défaut de $\mathcal{H}_3$, cependant ce n'est pas le cas. Comme le montre la figure 6.6, le schéma basé sur $\mathcal{H}_{3|4}$ ne présente pas la diffusivité thermique espérée. En conclusion, le premier schéma composite supportant la thermique est basé sur $\mathcal{H}_{4|5}$.

6.2.2 Saut de température à la paroi

L'objectif est ici de vérifier l'aptitude des différents schémas à rendre compte du saut de température à la paroi, abordé section 4.4.1. Pour faire cela nous proposons d'étudier le champ de température d'un gaz contenu entre deux plaques parallèles et infinies, placées en $y = 0$ et $y = L$ et dont les températures normalisées $\hat{\theta} = \theta/\theta_0$ respectives sont $\theta(0)/\theta_0 = (1 + 0.05)$ et $\theta(L)/\theta_0 = (1 - 0.05)$.

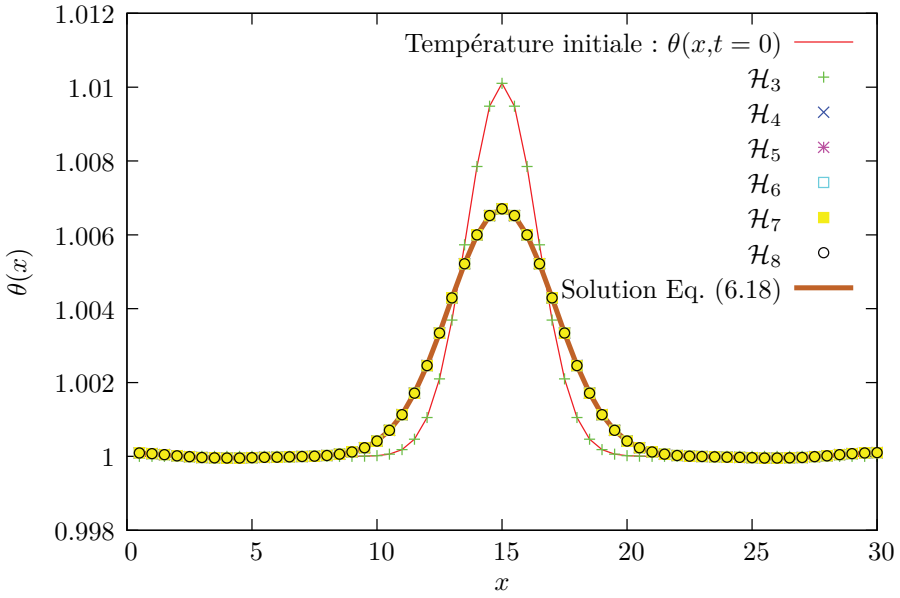


FIGURE 6.5 – Champs de températures obtenus avec les quadratures pures $\mathcal{H}_n$, pour $n = 3 \dots 8$, à l'instant $t = 24{,}90$. Les paramètres de la simulation sont : $\delta t = 0{,}1$, $\delta x = 0{,}5$ et $\tau_0 = 0{,}1\,\delta t = 0{,}01$.

Ce problème est de dimension 2, et l'expression du saut de température s'écrit :

$$\hat{\theta}_{jump} = \hat{\theta}(\hat{y}_W) - \hat{\theta}_W = \frac{4}{3} K_n \left.\frac{\partial \hat{\theta}}{\partial \hat{y}}\right|_{\hat{y}_W}, \tag{6.24}$$

où l'on rappelle que $\hat{\theta}_W$ est la température de la paroi solide et $\hat{\theta}(\hat{y}_W)$, la température du gaz au voisinage direct de celle-ci.

Pour vérifier que les différents schémas suivent bien cette relation, nous allons comparer les valeurs du saut de température $\hat{\theta}_{jump}$ obtenues pour différents nombres de Knudsen, en la mesurant par $(\hat{\theta}(\hat{y}_W) - \hat{\theta}_W)$ d'une part et par $\frac{4}{3} K_n \left.\frac{\partial \hat{\theta}}{\partial \hat{y}}\right|_{\hat{y}_W}$ d'autre part, et ceci avec différents schémas. La figure 6.7 montre les profils de température obtenus pour des nombres de Knudsen $0{,}005 \leq K_n \leq 5{,}12$, grâce au schéma construit sur $\mathcal{H}_{4|5}$. On observe bien un saut de température à la paroi, dont l'amplitude augmente avec le nombre de Knudsen.

Pour les schémas basés sur $\mathcal{H}_{3,4,5}$, $\mathcal{H}_{4|5}$ et $\mathcal{H}_{5|6}$, le saut de température, *i.e.* la différence entre la température de la paroi placée en $y = 0$ et celle du fluide en $y = 1$, est représenté en fonction de K_n sur la figure 6.8. Ces résultats montrent que les différents schémas donnent des résultats très similaires. Une chose étonnante est que le schéma basé sur $\mathcal{H}_3$, bien qu'il ne supporte pas la diffusion de la chaleur, capture aussi bien le saut de température que les schémas d'ordre plus élevé. Ce comportement est probablement produit par une différence de propriété entre les nœuds fluides et les nœuds de bords, remarque corroborée par le fait que les profils

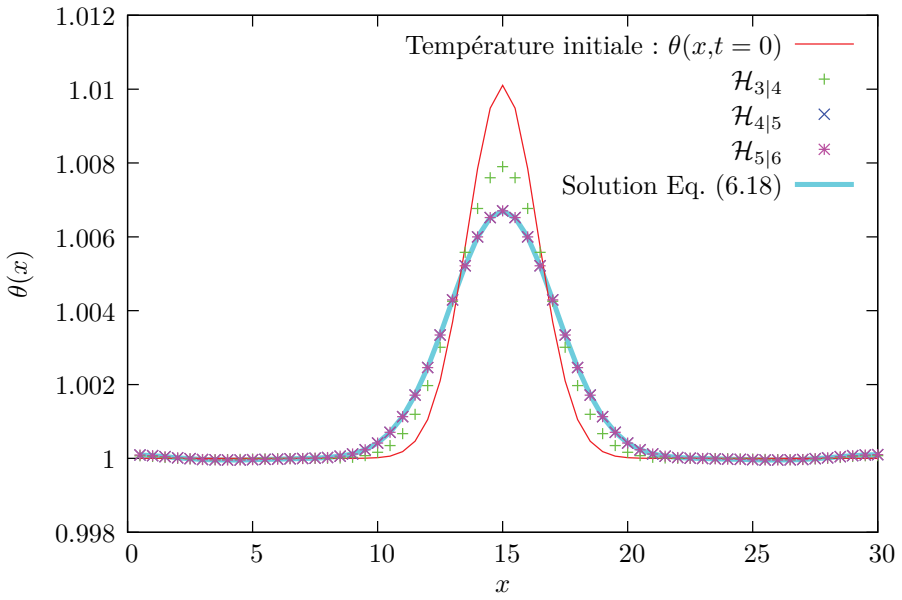


FIGURE 6.6 – Champs de températures obtenus avec les quadratures composites $\mathcal{H}_{n|n+1}$, pour $n = 3 \dots 5$, à l'instant $t = 24{,}90$. Les paramètres de la simulation sont : $\delta t = 0{,}1$, $\delta x = 0{,}5$ et $\tau_0 = 0{,}1\,\delta t = 0{,}01$.

de température obtenus par $\mathcal{H}_3$ ne sont pas corrects.

La figure 6.9 donne les sauts de températures obtenus par la mesure de la pente du profil de température $\left.\frac{\partial\hat{\theta}}{\partial\hat{y}}\right|_{\hat{y}_W}$. Les courbes obtenues y sont comparées à l'une des courbes obtenue par la première méthode de mesure. Bien que les mesures de $\hat{\theta}_{jump}$ par calcul de la pente soient plus éparpillées que celles représentées figure 6.8, elles montrent une bonne correspondance avec les mesures précédentes, permettant de conclure que les schémas présentés ici respectent une propriété des gaz hors d'équilibre, liée à leur température par la relation (6.24).

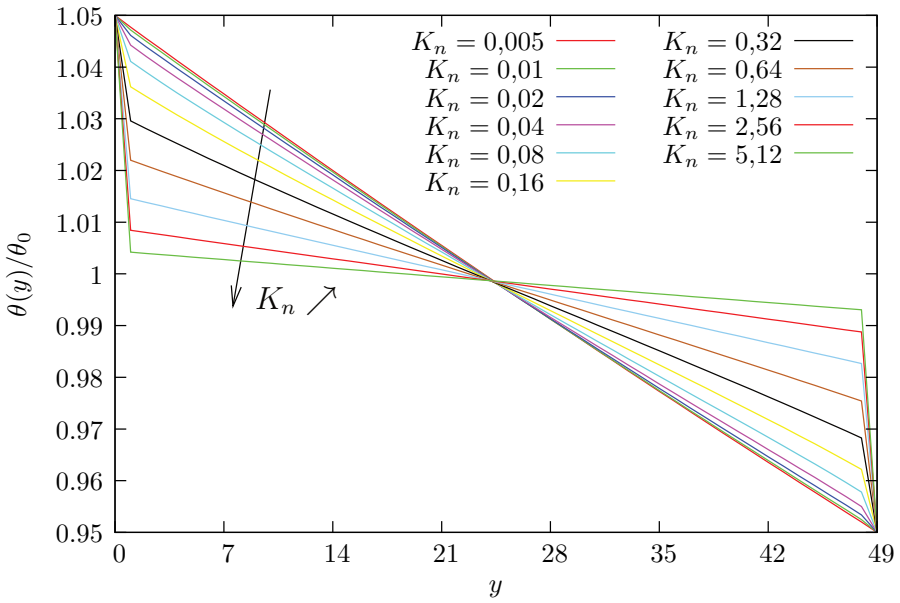


FIGURE 6.7 – Champs de température entre deux parois maintenues aux températures $\theta_0 \pm 0{,}05$ pour différentes valeurs du nombre de Knudsen. Ces résultats ont été obtenus grâce au schéma composite $\mathcal{H}_{4|5}$. On remarque que le saut de température augmente avec le Knudsen.

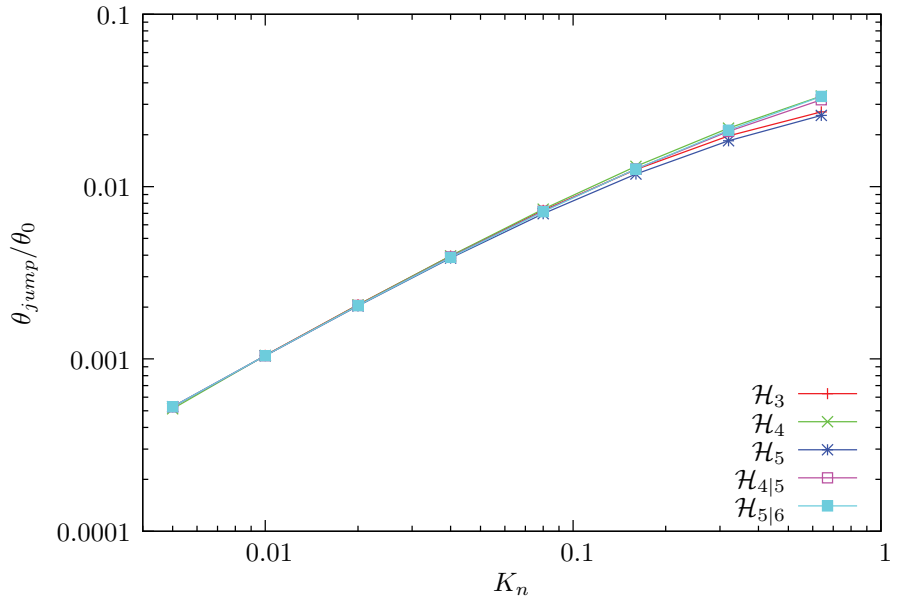


FIGURE 6.8 – Saut de température à la paroi $\hat{\theta}_{jump}$ obtenu par la mesure de $\hat{\theta}(y_W) - \hat{\theta}_W$.

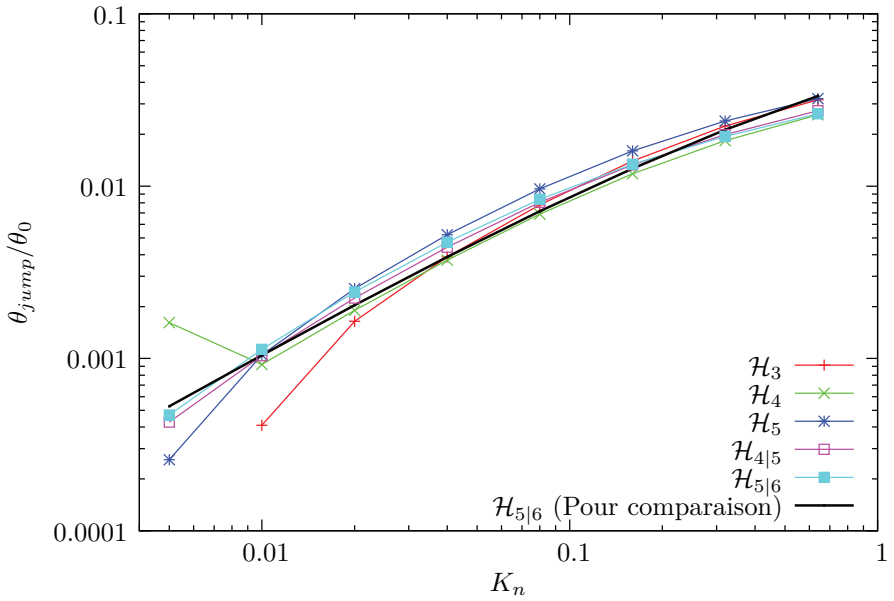


FIGURE 6.9 – Saut de température à la paroi $\hat{\theta}_{jump}$ obtenu par la mesure de la pente $\frac{\partial\hat{\theta}}{\partial\hat{y}}\Big|_{\hat{y}_W}$. La comparaison avec les courbes obtenues par la mesure de $\hat{\theta}(y_W) - \hat{\theta}_W$ pour le schéma basé sur $\mathcal{H}_{5|6}$, permet de conclure que le relation (6.24) est bien respectée par tous les schémas.

6.3 Poiseuille plan – Paradoxe de Knudsen

Cette section porte sur l'étude de l'écoulement de Poiseuille plan mis en mouvement par une force extérieure. Nous allons utiliser les résultats énoncés dans la section 4.3 du chapitre 4 pour déterminer, dans un premier temps, quelle quadrature présente les meilleures qualités pour résoudre cet écoulement sur une large gamme de nombres de Knudsen (typiquement : $0{,}001 < K_n < 10$) correspondant aux régimes hydrodynamique à transitionnel.

Nous commencerons par étudier les débits et les vitesses de glissements en fonction du Knudsen, obtenus avec différentes quadratures pures et composites. La comparaison avec des solutions semi-analytiques et des valeurs expérimentales, permettra de classer les schémas par ordre de qualité. La mesure des temps de calculs pour les différents schémas permettra finalement de déterminer quelle quadrature est la plus performante.

6.3.1 Courbes de débits

Nous avons vu section 4.3.2 que le débit d'un écoulement de Poiseuille, normalisé par $4\,U_0\,K_n$, était donné en fonction du nombre de Knudsen K_n par la relation :

$$Q_0 = \frac{Q}{4\,U_0\,K_n} = \frac{1}{6\,K_n} + 1 + 2 \times 0{,}13\,K_n, \tag{6.25}$$

où U_0 est la vitesse maximale dans l'écoulement en *régime hydrodynamique*. Rappelons que cette expression est obtenue avec un glissement au second ordre en K_n, sa validité est donc limitée aux valeurs de K_n relativement faibles.

Commençons par comparer les résultats obtenus grâce aux schémas basés sur différentes quadratures pures. La figure 6.10 représente les courbes de débits obtenues avec de tels schémas. Pour comparaison, sont représentées les courbes théoriques Q_0 (Eq. (6.25)) et $1/(6\,K_n)$ (Navier-Stokes sans glissement) ainsi que des valeurs expérimentales (Dong [Don56]).

L'observation de ces résultats permet les constats suivants :

1. Dans la limite hydrodynamique, pour $K_n < 0{,}01$, tous les schémas convergent vers les valeurs de débits données par les équations de Navier-Stokes sans glissement.
2. Les débits obtenus avec les schémas $\mathcal{H}_n$, pour $n > 3$, sont en très bon accord avec les débits théoriques avec glissement (Q_0), pour des Knudsen allant jusqu'à 0,1. Le schéma basé sur $\mathcal{H}_3$ trouve cependant sa limite de validité pour $K_n \lessapprox 0{,}1$.
3. De façon générale, l'augmentation de l'ordre de la quadrature augmente la valeur du nombre de Knudsen au-delà duquel les résultats cessent d'être en

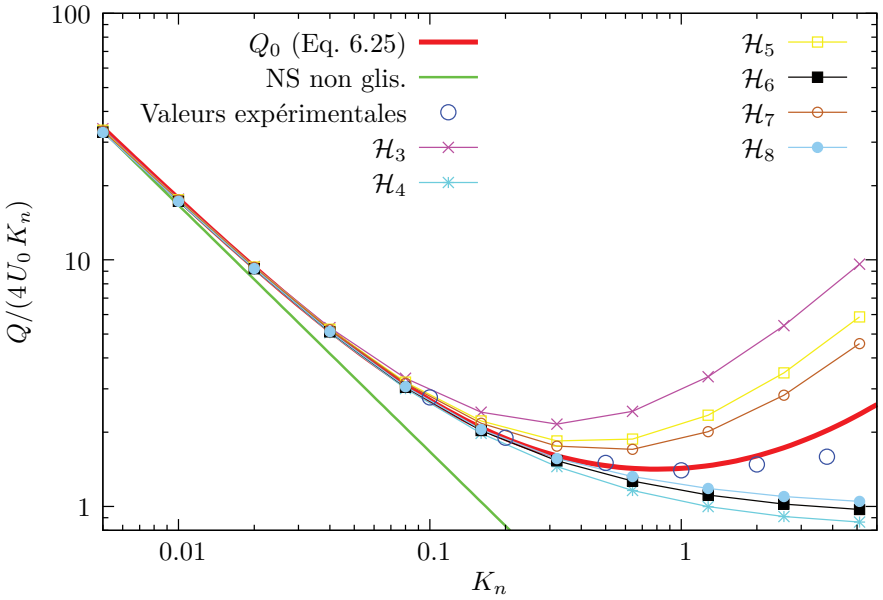


FIGURE 6.10 – Courbes de débit, fonction du Knudsen, obtenues par les schémas basés sur les quadratures pures $\mathcal{H}_n$, pour $n = 3, \ldots, 8$.

accord avec les valeurs de référence. Il y a donc convergence des schémas d'ordre successifs vers la solution.

4. La convergence des schémas d'ordres successifs vers la solution est lente. Elle est de plus alternée, avec une surestimation par les schémas d'ordre impair et une sous-estimation par les schémas d'ordre pair. En outre, on remarque que la sous-famille des schémas d'ordre pair converge plus lentement vers la solution que celle des schémas d'ordre impair.
5. Les schémas d'ordre impair présentent un minimum, contrairement aux schémas d'ordre pair. Ce minimum de débit donne aux schémas impairs une supériorité sur ceux d'ordre pair, en ce sens qu'ils représentent, au moins qualitativement, le paradoxe de Knudsen.

Ces observations nous amènent à conclure que tels quels, ces schémas ne nous permettent que difficilement de résoudre le régime transitionnel. En effet, les coûts numériques (abordés plus bas, cf. Fig. 6.17 page 116) des schémas $\mathcal{H}_7$ et $\mathcal{H}_8$ sont déjà très élevés, et ils n'ont toujours pas convergés vers la solution pour $K_n = 1$. On remarque néanmoins que le schéma basé sur $\mathcal{H}_4$, permettra de résoudre des écoulements pour des Knudsen $K_n \lesssim 0{,}3$, correspondant au début du régime transitionnel, donc à la fin du régime de glissement.

Dans la section concernant le gaz de Knudsen (section 6.1), nous avions déjà observé la convergence alternée des résultats obtenus par des schémas d'ordres successifs. Il avait alors été proposé de les combiner, pour accélérer la convergence des schémas.

La figure 6.11 donne les courbes de débits obtenues grâce aux schémas composites. On remarque immédiatement que la composition des quadratures permet d'améliorer

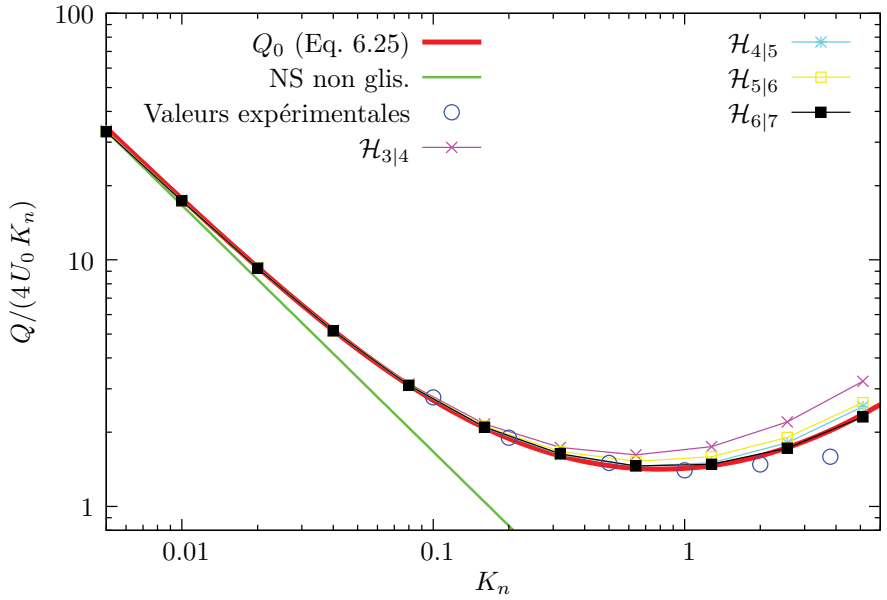


FIGURE 6.11 – Courbes de débits, fonction du Knudsen, obtenues par les schémas basés sur les quadratures composites $\mathcal{H}_{n|n+1}$, pour $n = 3, \ldots, 6$.

considérablement les solutions. La qualité de ces résultats est bien sûr à mettre en balance avec leur coût numérique respectif. La lecture de la figure 6.17, sur laquelle sont donnés les temps de calculs pour les différents schémas, permet de voir par exemple que $\mathcal{H}_{4|5}$ est comparable à $\mathcal{H}_6$ et que $\mathcal{H}_{5|6}$ est comparable à $\mathcal{H}_7$.

On montre ainsi que les schémas composites proposés dans cette thèse permettent d'accélérer la convergence de la procédure de construction des LBM proposée par Shan et He, dont le défaut avéré est une convergence lente.

On remarque, d'autre part, que l'augmentation de l'ordre des schémas composites $\mathcal{H}_{n|n+1}$, n'améliore que très peu la solution pour $n > 3$. La quadrature $\mathcal{H}_{4|5}$ constitue un optimal, en ce sens que les quadratures d'ordres supérieurs ne font qu'augmenter les temps de calcul.

6.3.2 Vitesses de glissement

Section 4.3.1, nous avons obtenu pour l'écoulement de Poiseuille le profil de vitesses suivant :

$$u_x(y) = 4\,U_0 \left(\frac{y}{H} - \frac{y^2}{H^2} \right) + 4\,U_0 \left(K_n + 2 \times 0{,}13\,K_n^2 \right), \tag{6.26}$$

avec glissement au second ordre en K_n.

Cette expression nous permet d'écrire la vitesse de glissement $U_{slip} \equiv u_x(0)$ à la

paroi, normalisée par U_0 :

$$\frac{U_{slip}}{U_0} = 4\,K_n + 4 \times 2 \times 0{,}13\,K_n^2. \tag{6.27}$$

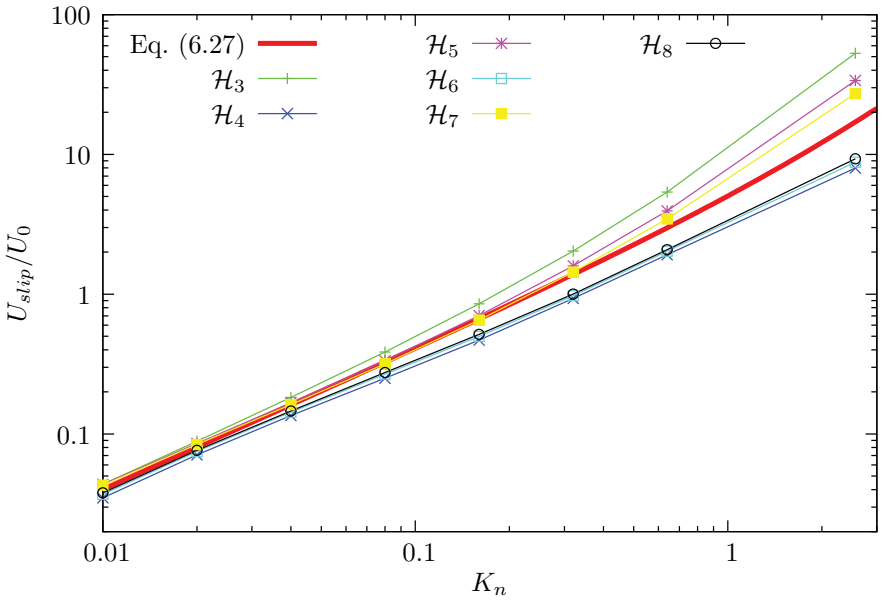


FIGURE 6.12 – Vitesses de glissement en fonction du Knudsen. Résultats obtenus grâce aux schémas basés sur les quadratures pures $\mathcal{H}_n$, pour $n = 3, \ldots, 8$.

Les vitesses de glissement normalisées, obtenues avec les schémas purs et composites ont été représentées, respectivement, sur les figures 6.12 et 6.13. Ces vitesses de glissement on été mesurées sur le premier nœud fluide, à proximité d'une des paroi de l'écoulement [5].

L'observation des courbes obtenues par les schémas purs (Fig. 6.12) nous amène aux mêmes conclusions que précédemment pour les débits. En effet, les vitesses de glissement, données par l'ensemble des schémas, tendent toutes vers les mêmes valeurs, données par l'équation (6.27), pour $K_n \lesssim 0{,}1$. Pour les valeurs supérieures du nombre de Knudsen, on remarque encore une convergence alternée des schémas d'ordres successifs, avec une surestimation systématique de la vitesse de glissement par les schémas impairs et une sous-estimation par les schémas pairs.

Là encore, l'introduction de schémas composés permet d'accélérer la convergence vers la solution (Eq.6.27), comme le montre la figure 6.13. On remarquera d'autre part que la convergence des schémas composites $\mathcal{H}_{n|n+1}$ est atteinte, sur la plage de nombre de Knudsen d'intérêt, pour $n = 4$.

5. Une représentation des champs de vitesse, pour différentes valeurs du nombre de Knudsen, est donnée figure 6.14 page 113.

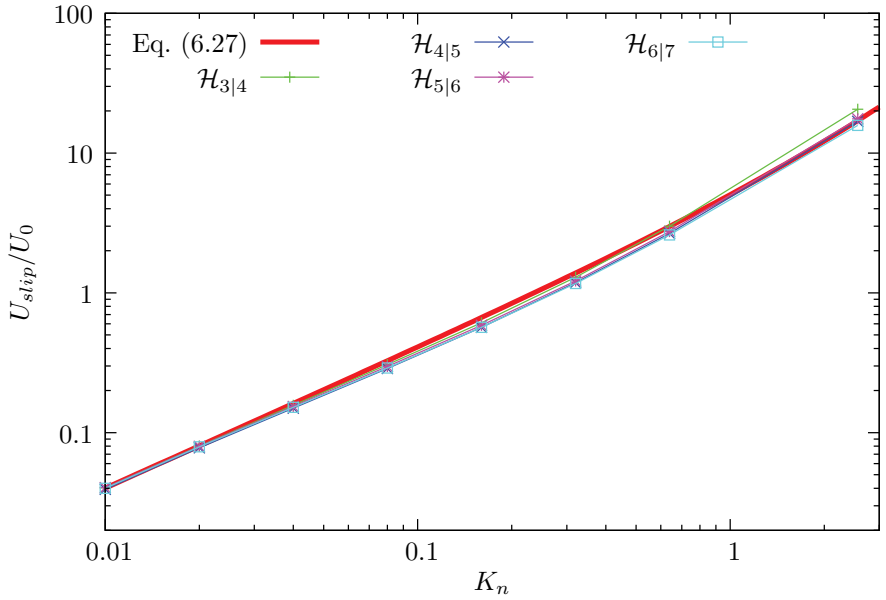


FIGURE 6.13 – Vitesses de glissement en fonction du Knudsen. Résultats obtenus grâce aux schémas basés sur les quadratures composites $\mathcal{H}_{n|n+1}$, pour $n = 3, \ldots, 6$.

6.3.3 Profils de vitesse et de température

Par les études systématiques réalisées dans les sections précédentes de ce chapitre, nous avons pu conclure que le schéma composite $\mathcal{H}_{4|5}$ était le meilleur compromis entre temps de calcul et qualité des résultats produits. Nous allons donc maintenant nous restreindre à son emploi et regarder les profils de vitesse et de température de l'écoulement de Poiseuille, obtenus pour différentes valeurs du nombre de Knudsen. Le nombre de Knudsen ne suffisant pas pour paramétrer une simulation, le nombre de Mach sera fixé[6] à $M_a = 0{,}0001$.

Sur la figure 6.14 sont représentés les profils de vitesse normalisée par la vitesse maximale de l'écoulement en régime hydrodynamique ($4\,U_0$), pour des nombres de Knudsen compris dans l'intervalle $0{,}005 < K_n < 0{,}64$. Cette représentation permet de constater que le profil de vitesse dans l'écoulement est la superposition d'un profil parabolique et d'un profil plan. On pourra remarquer que pour des courbes représentées, celles pour lesquelles $K_n \leq 0{,}16$, sont simplement des profils paraboliques d'amplitude $4\,U_0$ (solutions hydrodynamiques), translatées de la vitesse de glissements. Par contre, la partie parabolique des profils pour $K_n = 0{,}32$ et $K_n = 0{,}64$, sont d'amplitude plus importante qu'en régime hydrodynamique (de l'ordre de $1{,}2 \times 4\,U_0$ et $1{,}5 \times 4\,U_0$ respectivement). Cette limite pour $K_n \sim 0{,}1$, pour laquelle le champ de vitesse n'est plus que la superposition linéaire du profil hydrodynamique sans glissement et d'un profil plat, correspond à l'entrée dans le régime transitionnel. Sur la figure 6.15 sont tracés, pour des nombres de Knudsen compris dans l'intervalle

6. Cette valeur $M_a = 0{,}0001$ du nombre de Mach correspond à une vitesse U_0 de l'ordre de $3\,cm/s$, pour de l'air à TPS.

$0{,}005 < K_n < 2{,}56$, les profils de vitesse normalisés $4\,U_0$ numériques (par $\mathcal{H}4|5$) et semi-analytiques (Eq. (6.26)). Cette représentation montre bien qu'il ne s'agit pas simplement des profils hydrodynamiques augmentés de la vitesse de glissement.

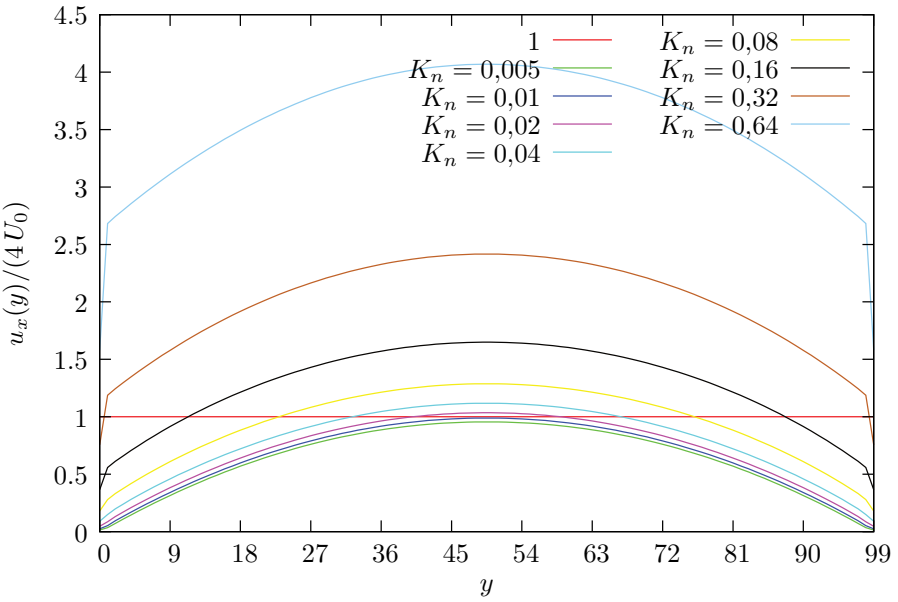


FIGURE 6.14 – Profils de vitesse normalisée par la vitesse maximale de l'écoulement en régime hydrodynamique ($4\,U_0$). Résultats obtenus grâce au schéma basé sur $\mathcal{H}_{4|5}$, pour différentes valeurs du nombre de Knudsen et un nombre de Mach $M_a = 0{,}0001$.

Sur la figure 6.16 sont représentés les profils de température normalisée par la température de référence θ_0, à laquelle a été retranché 1. La figure 6.16(d) donne ces profils pour des nombres de Knudsen compris dans l'intervalle $0{,}005 < K_n < 5{,}12$, avec en ordonnées une échelle logarithmique. Les figures 6.16(a)-(c) donnent quant à elles les profils en échelles linéaires pour des nombres de Knudsen $K_n = 0{,}005$, $K_n = 0{,}16$ et $K_n = 1{,}28$ correspondant respectivement aux régimes hydrodynamique, de glissement et transitionnel. Le profil de température représenté 6.16(b) pourra être comparé à celui donné par Zheng et *al.* [ZGA02]. Il est montré dans l'article de Zheng, que pour un écoulement de Knudsen de 0,1, le profil de température doit avoir une forme bimodale ; forme que l'on retrouve bien ici. L'observation des courbes 6.16(a)-(c) permet donc de dire que cette forme bimodale du profil de température disparaît aux grands Knudsen, pour faire place à un profil parabolique.

La dernière figure 6.16(d) montre quant à elle une augmentation de la valeur du saut de température avec le nombre de Knudsen.

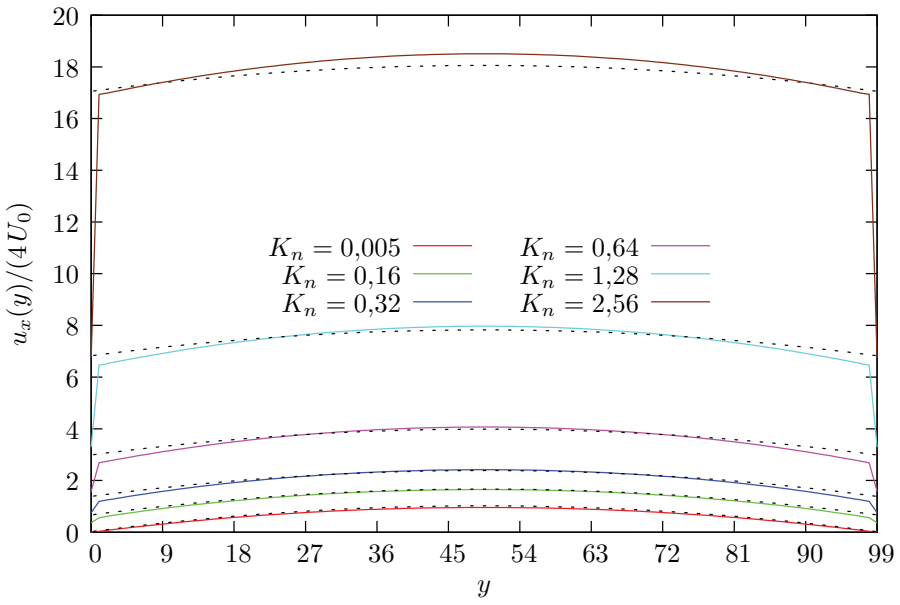


FIGURE 6.15 – Profils de vitesse normalisée par la vitesse maximale de l'écoulement en régime hydrodynamique ($4\,U_0$). Pour comparaison, les profils semi-analytiques (Eq. (6.26)) sont représentés en pointillés. Résultats obtenus grâce au schéma basé sur $\mathcal{H}_{4|5}$, pour différentes valeurs du nombres de Knudsenet un nombre de Mach $M_a = 0{,}0001$.

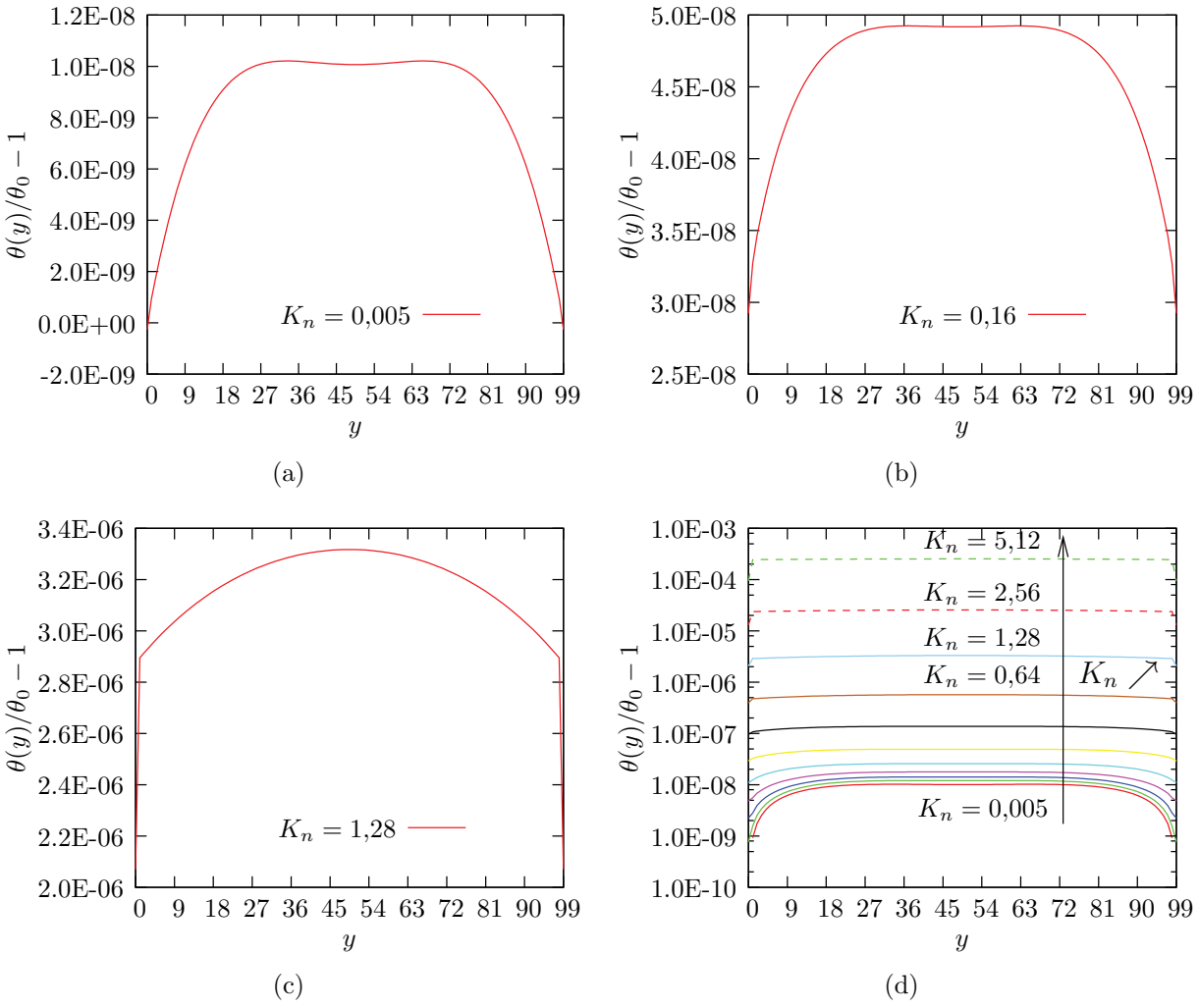


FIGURE 6.16 – Champs de température pour différentes nombres de Knudsen et un nombre de Mach $M_a = 0{,}0001$. Résultats obtenus grâce au schéma basé sur $\mathcal{H}_{4|5}$. Sur la figure (d) les abscisses sont en échelle logarithme.

6.4 Coût de calcul des différents schémas

Cette section traite des coûts de calcul associés aux différents schémas de Boltzmann sur réseau abordés précédemment. Cette évaluation est importante, car la qualité des solutions obtenues par une méthode de résolution n'est pas le seul critère à considérer. Le choix d'une méthode dépendra aussi du temps nécessaire pour obtenir la solution [7]. Les LBM étant considérées comme des méthodes performantes, nous proposons de comparer les coûts de calcul des différents schémas avec celui du modèle D_2Q_9, (équivalent au schéma basé sur $\mathcal{H}_3$). La figure 6.17 représente les temps de calcul, normalisés par celui du schéma basé sur $\mathcal{H}_3$ en fonction du nombre de faisceaux.

Le problème traité, pour réaliser ces mesures de temps, est un écoulement de Poiseuille plan, mis en mouvement par une force extérieure et dont le nombre de Knudsen est 0,1. Le problème est initialisé en imposant des champs de densité, de vitesse et de température uniformes, avec des valeurs respectives $\rho = 1$, $\boldsymbol{u} = \mathbf{0}$ et $\theta = \theta_0$. Le temps de calcul pour chacun des schémas est alors pris comme le temps nécessaire pour atteindre la stationnarité du champ de vitesses.

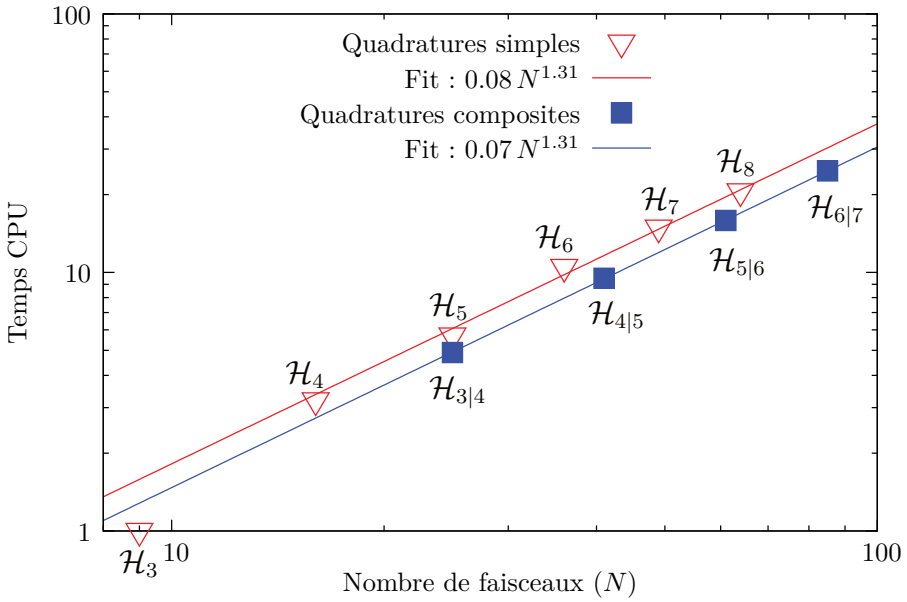


FIGURE 6.17 – Temps de calcul en fonction du nombre de faisceaux (échelles Log-Log) pour les schémas basés sur les quadratures pures $\mathcal{H}_n$ et composites $\mathcal{H}_{n|n+1}$. Ces temps de calculs ont été normalisés par le temps de calcul du schéma basé sur $\mathcal{H}_3$.

La figure 6.17 montre que pour tous les schémas, les temps de calculs en fonction du nombre de faisceaux N suivent une loi de puissance d'exposant 1.31, à l'exception du schéma basé sur $\mathcal{H}_3$, qui tire bénéfice du transport de type automate cellulaire.

7. La quantité de données mise en œuvre par une méthode devrait aussi être considérée, cependant les capacités mémoire des ordinateurs modernes rend le critère du temps de calcul prioritaire.

D'autre part, on remarque que les schémas composites sont, à nombre de faisceaux identique, plus performants que les schémas purs.

Finalement, nous pouvons voir que le schéma composite $\mathcal{H}_{4|5}$, dont la qualité a été montrée dans les sections précédentes, engendre des coûts de calcul environ 10 fois plus importants que le schéma basé sur $\mathcal{H}_3$, donc aux LBM. C'est cependant le prix à payer pour pouvoir traiter des écoulements fluides hors du régime hydrodynamique, mettant en jeu les phénomènes de glissement et de saut de température énoncés précédemment.

6.5 Écoulement dans un long canal rectiligne

Le dernier exemple d'application présenté ici est celui d'un écoulement à travers un long canal, mis en mouvement par un gradient de densité. Ce problème est traité par ailleurs de façon analytique par Arkilic et *al.* [AMB97] et par Dongari et *al.* [DAA07]. Ces auteurs utilisent les équations de Navier-Stokes dans le cas compressible avec des conditions de glissement au premier ordre selon le nombre de Knudsen.

Ce long canal commence à être utilisé pour tester des schémas numériques (voir par exemple Verhaeghe et *al.* [VLB09], Guo et *al.* [GZS06] ou encore Lee et *al.* [LL05]). Nous ferons référence aux résultats de ces auteurs pour analyser nos propres résultats. Des résultats expérimentaux sont aussi disponibles (Arkilic et *al.* [AMB97] par exemple). Ils montrent en particulier que les effets de compressibilité sont importants dans les écoulements en micro-canaux.

Le problème est illustré figure 6.18. Il s'agit de simuler en dimension 2, l'écoulement d'un gaz entre deux plaques parallèles, de longueur L_y et espacées d'une distance $L_x \ll L_y$. Ces parois, de type cinétique avec accommodation totale, sont maintenue à la température de référence θ_0. L'écoulement est mis en mouvement par un gradient de densité $(\rho_s - \rho_e)/L_y$, avec ρ_e la densité en entrée et $\rho_s < \rho_e$ la densité en sortie du canal. Les valeurs de ρ_e et de ρ_s sont maintenues constantes.

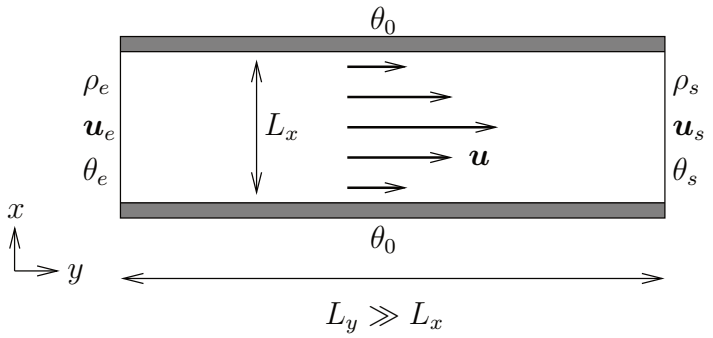


FIGURE 6.18 – Illustration du problème.

D'un point de vue pratique (cf. code annexe C.2.4), les conditions aux bords utilisées en entrée et en sortie consistent à imposer les densités ρ_e et ρ_s, puis à interpoler

les vitesses $\boldsymbol{u}_e$ et $\boldsymbol{u}_s$ et les températures θ_e et θ_s, (ces conditions aux bords ont été abordées section 5.9.2). D'autre part, le nombre de Knudsen local est, pour un gaz de sphères dures :

$$K_n = \frac{\sqrt{2.5\,\theta_0}}{L_x}\,\tau_0 \frac{\rho_0}{\rho}, \tag{6.28}$$

avec

$$\tau_0 = K_{n0} \frac{L_x}{\sqrt{2.5\,\theta_0}}. \tag{6.29}$$

Nous choisirons de prendre $\rho_0 = \rho_s = 1$, comme densité de référence. Ainsi, le Knudsen local K_n sera défini par rapport au Knudsen de référence K_{n0}, en sortie du canal, par :

$$K_n = \frac{K_{n0}}{\rho}. \tag{6.30}$$

La densité en entrée étant fixée à ρ_e, il en découle que si $K_{n0} = K_{n\,s}$, le nombre de Knudsen en sortie, alors le nombre de Knudsen en entrée est aussi fixé : $K_{n\,e} = K_{n\,s}/\rho_e$.

Les résultats en régime stationnaire de quatre simulations sont présentés figures 6.19 à 6.26. Les dimensions du domaine d'intégration sont $L_x = 25$ et $L_y = 1000$, nous avons donc bien $L_x \ll L_y$. Les quatre simulations effectuées correspondent aux combinaisons de $\rho_e = \{3; 10\}$ et $K_{n0} = \{0.1; 1\}$.

L'évolution du fluide, lors de son écoulement le long du canal, met en jeu deux principaux phénomènes physiques. Lors de son trajet à l'intérieur du canal, un volume de gaz voit sa densité diminuer, il subit donc une détente. Cette détente a comme conséquences, d'une part de refroidir le gaz, d'autre part d'augmenter le libre parcours moyen des particules qui le constituent, donc d'augmenter la valeur du nombre de Knudsen [8].

Sont représentés figures 6.19(a)-(d), les champs de densité pour les différents cas traités. À ces champs de densité sont associés des champs de nombres de Knudsen, calculés par la relation (6.30) et représentés sur les figures 6.20(a)-(d). Comme énoncé précédemment, la raréfaction du gaz produit une variation du nombre de Knudsen, qui comme le montre la figure 6.20, augmente entre l'entrée et la sortie du canal. À cette augmentation du nombre de Knudsen est associée une augmentation du glissement sur les parois du canal. Cette augmentation du glissement peut être observée sur la figure 6.21 qui représente le champ de vitesse dans la direction de l'écoulement.

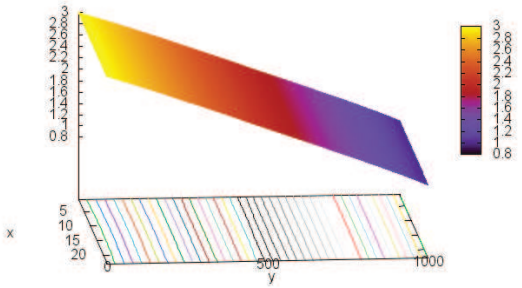
L'augmentation, le long du canal, de la vitesse de glissement, tend à produire une augmentation du débit massique $\int \rho\, u_y\, dx$, or ce débit doit rester constant tout le long du canal. Pour maintenir ce débit massique, une migration des particules doit se produire depuis les parois vers le centre du canal (l'impulsion locale selon x est peuplée au détriment de l'impulsion selon y), comme le montrent les figures 6.23 et

8. Dans ce cas, le nombre de Knudsen pourra être nommé *nombre de raréfaction local*.

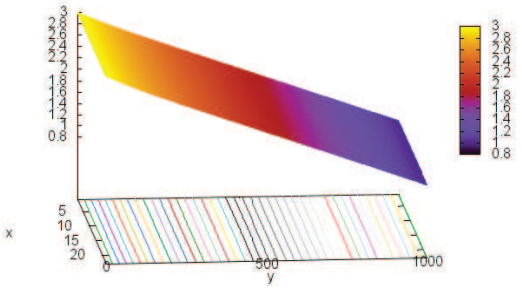
6.24, où sont représentées les vitesses transversales à l'écoulement (selon x), respectivement sur un tronçon au milieu du canal et en sortie du canal. On remarquera que l'amplitude de la vitesse selon x, relative à u_{max} la vitesse maximale l'écoulement selon y, est plus importante (d'un ordre de grandeur) aux petits Knudsen qu'aux grands. Cet effet semble avoir des répercussions sur la distribution de la densité dans la direction transversale à l'écoulement. On peut en effet constater (cf. figure 6.22) une déplétion de la densité près des parois et une réplétion au centre du canal, en particulier pour les cas où $K_{n0} = 0{,}1$. Un des cas traité est particulier, il s'agit de celui pour lequel $\rho_e = 10$ et $K_{n0} = 0{,}1$. Ce cas présente en effet une non-constance très marquée du nombre de Knudsen dans la direction transversale à l'écoulement (x). Le nombre de Knudsen y est plus grand près des parois du canal qu'en son centre. Cette variation selon x du Knudsen local est due à une déviation importante de la densité le long du canal par rapport au gradient de densité linéaire, comme le montrent les figures 6.19(c) et 6.22(c). Cette migration des parois vers le centre du canal peut aussi être observée sur la figure 6.25, où sont représentés les champs d'impulsions totales $\rho |\boldsymbol{u}|$ normalisées par l'impulsion $\rho_0 u_{max}$, avec u_{max} la vitesse maximale de l'écoulement (en sortie du canal) et ρ_0 la densité imposée en sortie.

Finalement, la figure 6.26 représente les champs de températures normalisées par θ_0.

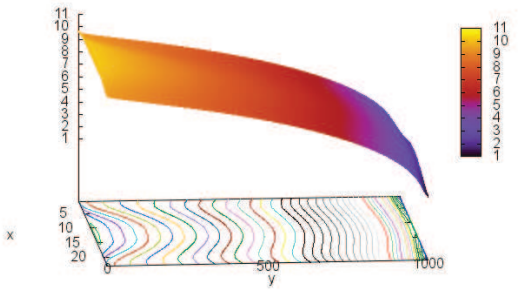
Les résultats obtenus pour un Knudsen de 0,1 et une pression d'entrée $\rho_e = 3$ (sous figures (a)) sont très semblables aux solutions analytiques obtenues par Arkilic et *al.* [AMB97]. En particulier pour la forme particulière de la vitesse transversale en sortie du canal.



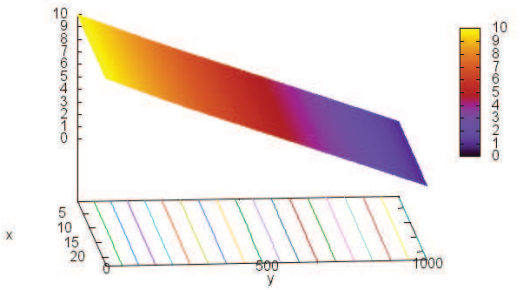
(a) $\rho_e = 3$, $K_{n0} = 0.1$



(b) $\rho_e = 3$, $K_{n0} = 1$

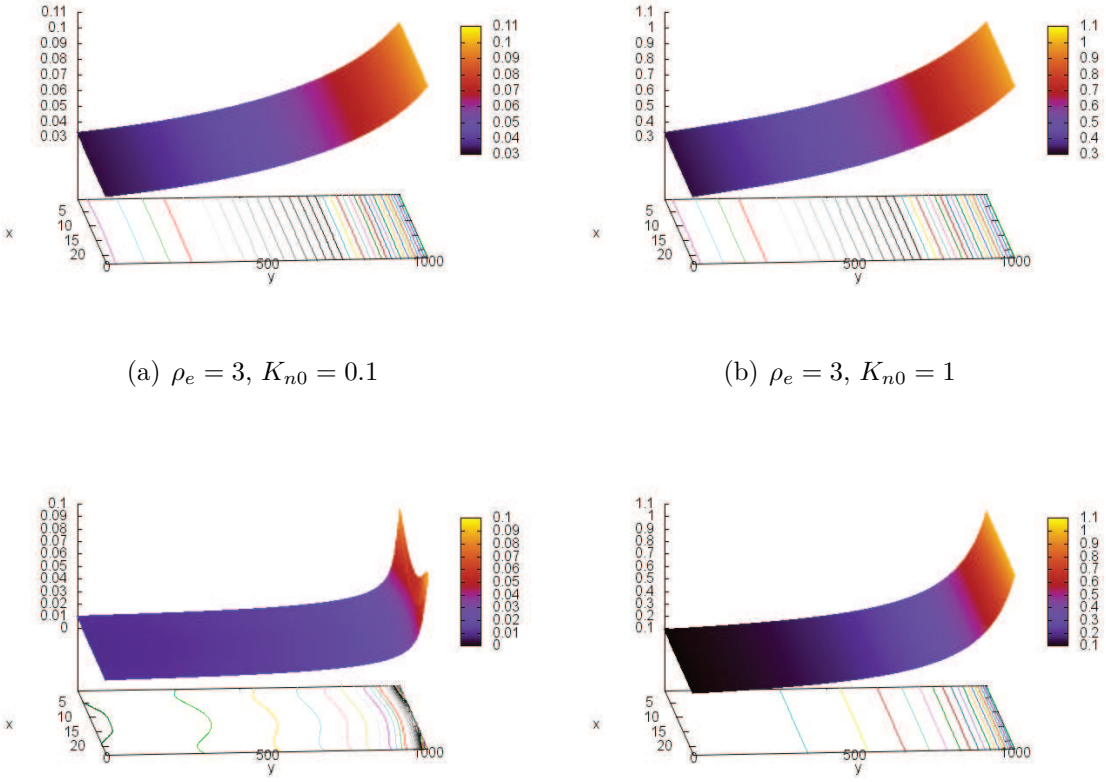


(c) $\rho_e = 10$, $K_{n0} = 0.1$



(d) $\rho_e = 10$, $K_{n0} = 1$

FIGURE 6.19 – Champ de densité ρ/ρ_0 normalisé par la densité de référence ρ_0.



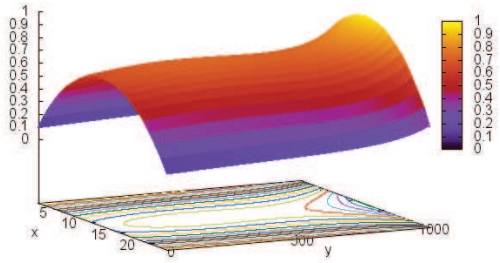
(a) $\rho_e = 3$, $K_{n0} = 0.1$

(b) $\rho_e = 3$, $K_{n0} = 1$

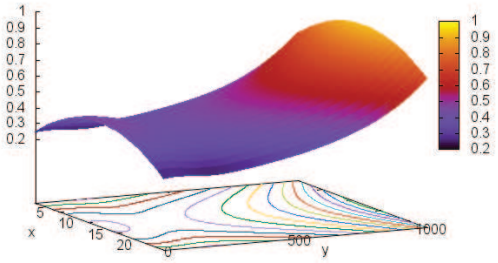
(c) $\rho_e = 10$, $K_{n0} = 0.1$

(d) $\rho_e = 10$, $K_{n0} = 1$

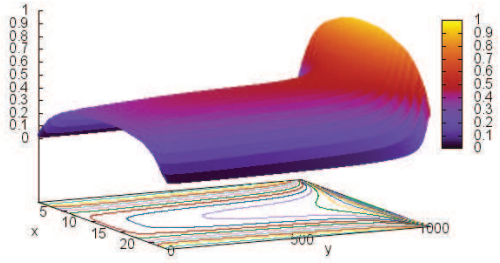
FIGURE 6.20 – Nombre de Knudsen local.



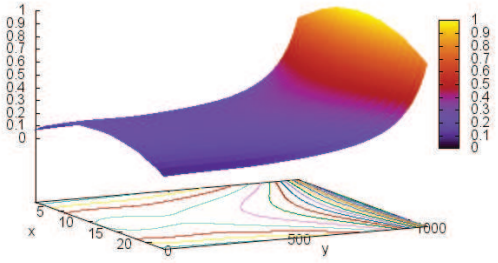
(a) $\rho_e = 3$, $K_{n0} = 0.1$



(b) $\rho_e = 3$, $K_{n0} = 1$

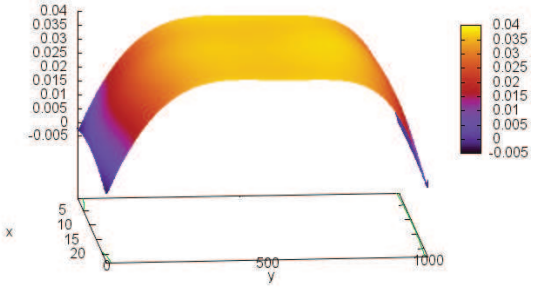


(c) $\rho_e = 10$, $K_{n0} = 0.1$

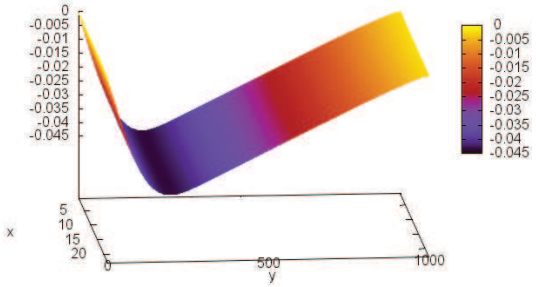


(d) $\rho_e = 10$, $K_{n0} = 1$

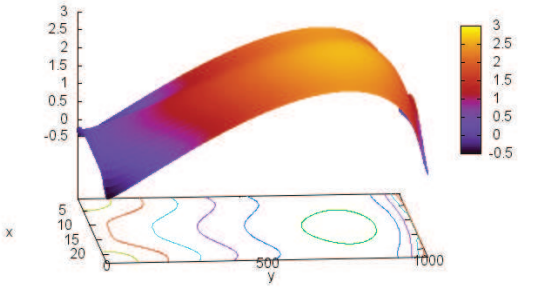
FIGURE 6.21 – Champ de vitesse selon y (u_y/u_{max}) normalisé par la vitesse maximale selon y, u_{max}.



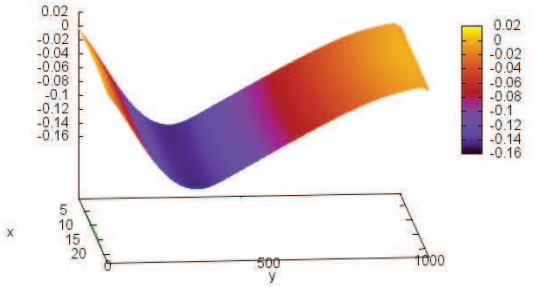
(a) $\rho_e = 3$, $K_{n0} = 0.1$



(b) $\rho_e = 3$, $K_{n0} = 1$

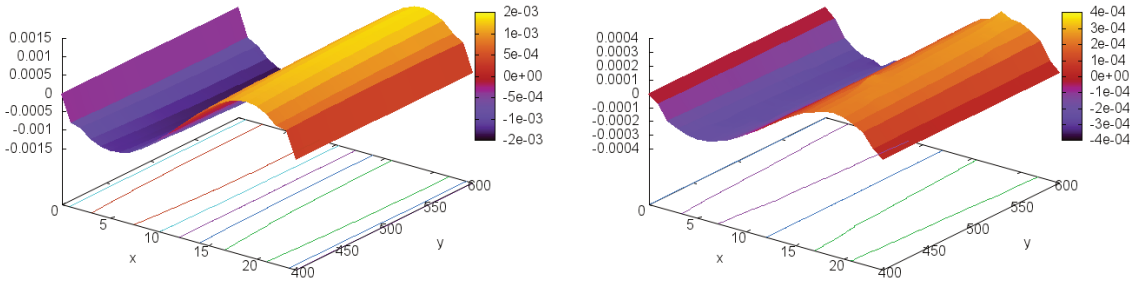


(c) $\rho_e = 10$, $K_{n0} = 0.1$

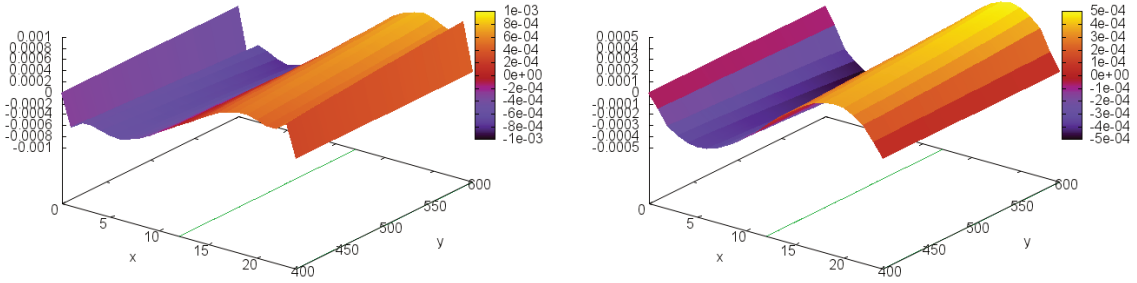


(d) $\rho_e = 10$, $K_{n0} = 1$

FIGURE 6.22 – Déviation au gradient de densité linéaire.

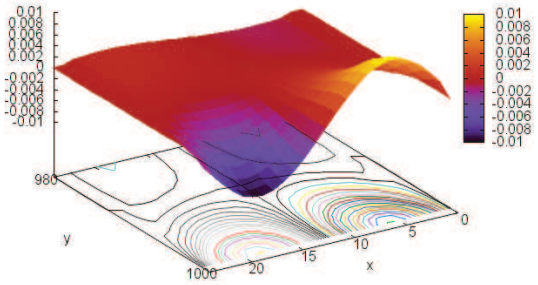


(a) $\rho_e = 3$, $K_{n0} = 0.1$ (b) $\rho_e = 3$, $K_{n0} = 1$

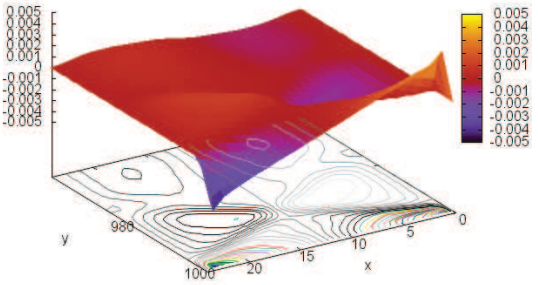


(c) $\rho_e = 10$, $K_{n0} = 0.1$ (d) $\rho_e = 10$, $K_{n0} = 1$

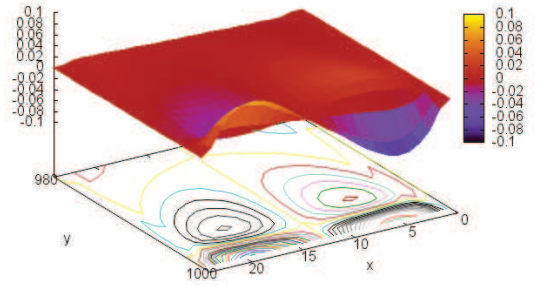
FIGURE 6.23 – Champ de vitesse selon x (u_x/u_{max}) normalisé par la vitesse maximale selon y, u_{max}. Il s'agit d'un tronçon du canal pris en son centre.



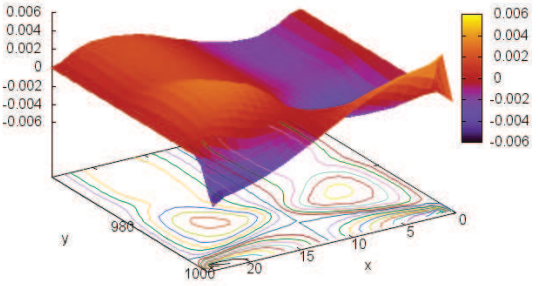
(a) $\rho_e = 3$, $K_{n0} = 0.1$



(b) $\rho_e = 3$, $K_{n0} = 1$

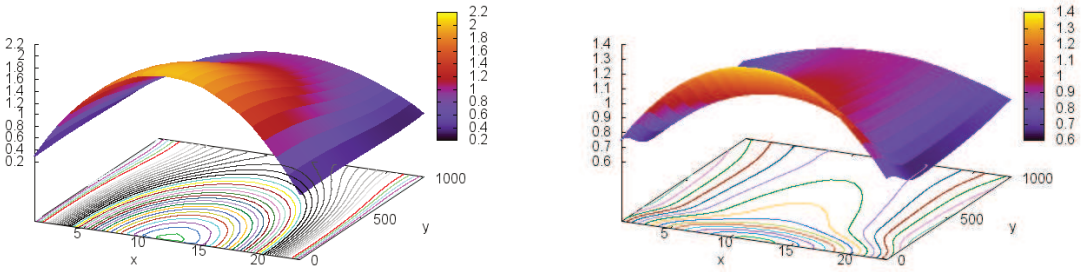


(c) $\rho_e = 10$, $K_{n0} = 0.1$



(d) $\rho_e = 10$, $K_{n0} = 1$

FIGURE 6.24 – Champ de vitesse selon x (u_x/u_{max}) normalisé par la vitesse maximale selon y, u_{max}. Il s'agit ici de la sortie du canal.



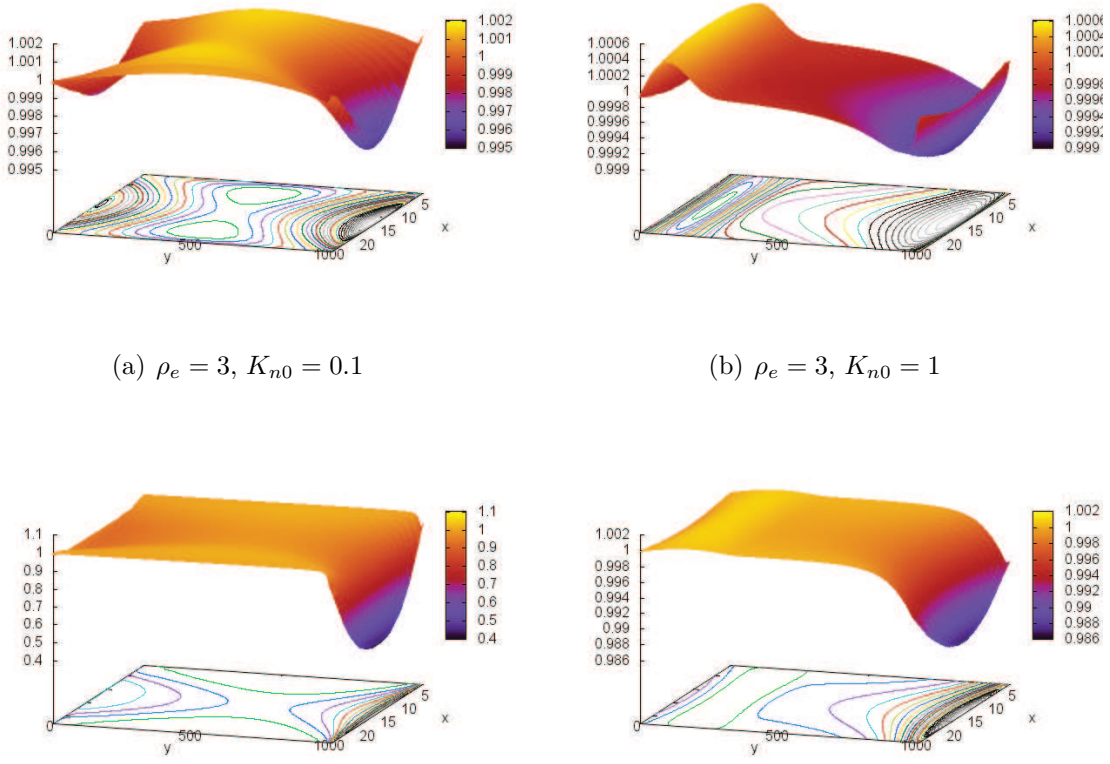
(a) $\rho_e = 3$, $K_{n0} = 0.1$

(b) $\rho_e = 3$, $K_{n0} = 1$

(c) $\rho_e = 10$, $K_{n0} = 0.1$

(d) $\rho_e = 10$, $K_{n0} = 1$

FIGURE 6.25 – Champ d'impulsion selon y, normalisée $(\rho u_y/(\rho_0\, u_{max}))$.



(a) $\rho_e = 3$, $K_{n0} = 0.1$

(b) $\rho_e = 3$, $K_{n0} = 1$

(c) $\rho_e = 10$, $K_{n0} = 0.1$

(d) $\rho_e = 10$, $K_{n0} = 1$

FIGURE 6.26 – Champ de température θ/θ_0 normalisé par la température de référence θ_0.

7 Conclusion

Dans cette thèse nous avons proposé une méthode de simulation d'écoulements anisothermes, valide sur une large gamme de nombres de Knudsen, typiquement $10^{-4} \lesssim K_n \lesssim 3$. Cette méthode a été développée en replaçant les méthodes de Boltzmann sur réseau (LBM) dans le contexte de la théorie cinétique, en nous basant d'une part sur la démarche de construction systématique des méthodes de Boltzmann sur réseau (LBM), proposée récemment par Shan et He et sur l'emploi de conditions aux bords de Maxwell.

Une confrontation systématique des solutions obtenues par une famille de schémas construits selon la procédure de Shan et He, avec des solutions de référence, nous a permis de constater que la convergence de cette procédure était lente.

La convergence alternée des solutions obtenues pour le problème de relaxation du gaz de Knudsen et du débit de l'écoulement de Poiseuille, nous a amené à combiner des schémas d'ordres successifs que nous avons appelé schémas composites. L'un des modèles composites ainsi produit ($\mathcal{H}_{4|5}$) s'est montré être un bon compromis en permettant de résoudre des écoulements anisothermes dans des régimes allant de l'hydrodynamique au régime transitionnel, avec un coût de calcul raisonnable. Dans ce travail, les schémas composites considérés ont été construits par combinaison de deux quadratures de Gauss-Hermite successives dont les poids ont simplement été divisés par deux. Les mêmes poids ont donc été accordés aux deux quadratures composées. Ce choix nous a semblé être le plus naturel pour introduire ces schémas. Il est cependant envisageable de considérer d'autres pondérations (sous la contrainte que la somme des poids soit unitaire) ou encore de composer plus de deux quadratures qui ne se succèdent pas forcément.

De nouvelles conditions aux bords ont été introduites. Elle permettent d'imposer la densité, la température ou la pression sur les limites d'un domaine. Ces conditions aux bords nous ont permis de simuler un écoulement mis en mouvement par un gradient de densité à travers un long canal. Ce problème est intéressant car il montre que le nombre de Knudsen peut varier, non seulement pour des raisons géométriques mais aussi thermodynamiques ; dans ce cas, la densité locale le long du canal. En outre, cet exemple a permis de montrer l'importance de la thermique, indispensable pour suivre les variations de températures mise en jeu lors de la détente du gaz.

Une bibliothèque mettant en œuvre la méthode en 2D et 3D, écrite en C++ est fournie en annexe de ce mémoire. Les deux problèmes d'écoulements de Poiseuille, mis en mouvement par une force extérieure et par un gradient de densité sont donnés comme exemples. Ces deux problèmes constitueront une bonne base pour construire des simulations d'écoulements dans des géométries plus complexes que de simples canaux, notamment des milieux poreux. Une étude de ce type a été réalisée par

Rouet et *al.* [RIDI11]. Elle met en évidence l'effet Klinkenberg dans un milieu poreux modèle. Il s'agit d'un effet lié à la géométrie du milieu (à la taille des structures confinentes) ; on observe une modification (augmentation), produite par le glissement sur les parois, de la perméabilité d'un milieu poreux avec le nombre de Knudsen.

Signalons aussi que cette méthode présente un intérêt pour l'étude de MEMS mettant en œuvre des écoulements gazeux. Ces systèmes ayant pour principe de tirer profit des effets liés à la raréfaction (au confinement) capturés par notre méthode. Un aspect non abordé numériquement dans ce mémoire, présente un intérêt particulier pour la communauté des MEMS. Il s'agit de la transpiration thermique qui est une mise en mouvement fluide à la surface d'une paroi présentant un gradient de tangentiel de température (cf. section 4.4.2). Disons simplement que cet effet est capturé par notre méthode, mais nous n'en connaissons pas la validité d'un point de vu quantitatif et il serait intéressant de mener une telle étude.

Pour finir, rappelons que la méthode ne permet de simuler que des fluides dont le nombre de Prandtl est unitaire. Cette limitation est imposée par l'opérateur de collision BGK (à temps de relaxation unique), que nous avons utilisé. Plusieurs pistes sont envisageables pour permettre de choisir la valeur du nombre de Prandtl. Citons les modèles à plusieurs temps de relaxations, introduits par D'Humière et *al.* [DGK$^+$02] ou encore les modèles quasi à l'équilibre (quasi-equilibrium) (voir par exemple Ansumali et *al.* [AAC$^+$07] ou encore Xu et *al.* [XHC08]).

Appendices

A Polynômes d'Hermites et quadrature de Gauss-Hermite

Communément, les polynômes d'Hermite et la quadrature de Gauss-Hermite sont définis pour la fonction de pondération $w = e^{-x^2}$ à laquelle sont associés les polynômes d'Hermite dans leur forme dite physique. Or, en physique statistique, nous sommes amené à considérer des fonctions de distributions gaussiennes (de Maxwell-Boltzmann) dont la largeur à mi-hauteur (la température) n'est pas fixée à l'unité. Nous considérerons donc, ici, les polynômes d'Hermite $\mathcal{H}_n^{[\sigma^2]}$ dans leur forme dite généralisée. Ces polynômes sont orthogonaux par rapport à la distribution de Maxwell-Boltzmann centrée $w^{[\sigma^2]}(\xi) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\xi^2/2\sigma^2}$, pour laquelle $\sigma = \sqrt{\frac{k_B T}{m}}$ est la vitesse du son dans le milieu.

A.1 Polynômes d'Hermite

A.1.1 Définition

Les polynômes d'Hermite sous leur forme dite généralisée peuvent être définis de la façon suivante :

$$\mathcal{H}_n^{[\sigma^2]}(x) = \frac{(-1)^n}{w^{[\sigma^2]}(x)} \frac{d^n}{dx^n} w^{[\sigma^2]}(x) \tag{A.1}$$

Où $w^{[\sigma^2]}(x)$ est la fonction de pondération [1] pour ces polynômes, telle que :

$$w^{[\sigma^2]}(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\, exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{A.2}$$

Remarque A.1.1. Il existe deux formes usuelles des polynômes d'Hermite, une forme dite *Probabiliste*, pour $\sigma^2 = 1$ et une forme dite *Physique*, pour $\sigma^2 = 1/2$.

A.1.2 Propriétés

Orthogonalité

Par construction, les polynômes d'Hermite sont orthogonaux entre eux, par rapport à la mesure $w^{[\sigma^2]}(x)$, en ce sens que :

1. On dit aussi une mesure.

$$\left\langle \mathcal{H}_n^{[\sigma^2]}(x) \middle| \mathcal{H}_m^{[\sigma^2]}(x) \right\rangle = \int_{-\infty}^{\infty} \mathcal{H}_n^{[\sigma^2]}(x)\, \mathcal{H}_m^{[\sigma^2]}(x)\, w^{[\sigma^2]}(x)\, dx = \frac{n!}{\sigma^{2n}}\, \delta_{nm}, \tag{A.3}$$

où δ_{nm} est le symbole de Kronecker et où l'opérateur $\langle|\rangle$ représente le produit scalaire (ou Hermitien) par rapport à $w^{[\sigma^2]}(x)$.

Fonction génératrice

On appelle fonction génératrice des polynômes $\mathcal{H}_n^{[\sigma^2]}$:

$$\exp\left(\frac{xt}{\sigma^2} - \frac{t^2}{2\sigma^2}\right) = \sum_{n=0}^{\infty} \mathcal{H}_n^{[\sigma^2]}(x) \frac{t^n}{n!}. \tag{A.4}$$

Le second membre de cette expression est connu sous le nom de développement de Gram-Charlier. Cette série est convergente si et seulement si $|t| < 1$.

Formules de récurrence

Les polynômes d'Hermite peuvent être construits récursivement grâce à la formule suivante :

$$\mathcal{H}_{n+1}^{[\sigma^2]}(x) = \frac{1}{\sigma^2} x \mathcal{H}_n^{[\sigma^2]}(x) - \frac{1}{\sigma^2} n \mathcal{H}_{n-1}^{[\sigma^2]}(x) \tag{A.5}$$

Avec $\mathcal{H}_{-1}^{[\sigma^2]} = 0$ et $\mathcal{H}_0^{[\sigma^2]} = 1$.

A.1.3 Liste des premiers polynômes d'Hermite généralisés

En voici une liste pour $n = 0, \ldots, 5$:

$$\begin{aligned}
\mathcal{H}_0^{[\sigma^2]}(x) &= 1 \\
\mathcal{H}_1^{[\sigma^2]}(x) &= \frac{x}{\sigma^2} \\
\mathcal{H}_2^{[\sigma^2]}(x) &= \frac{x^2}{\sigma^4} - \frac{1}{\sigma^2} \\
\mathcal{H}_3^{[\sigma^2]}(x) &= \frac{x^3}{\sigma^6} - \frac{3\,x}{\sigma^4} \\
\mathcal{H}_4^{[\sigma^2]}(x) &= \frac{x^4}{\sigma^8} - \frac{6\,x^2}{\sigma^6} + \frac{3}{\sigma^4} \\
\mathcal{H}_5^{[\sigma^2]}(x) &= \frac{x^5}{\sigma^{10}} - \frac{10\,x^3}{\sigma^8} + \frac{15\,x}{\sigma^6}
\end{aligned}$$

A.1.4 Racines des polynômes d'Hermite généralisés

Le tableau A.1 donne les racines des cinq premiers polynômes d'Hermite généralisés $\mathcal{H}_n^{[\sigma^2]}$.

n	Racines de $\mathcal{H}_n^{[\sigma^2]}$
0	$\varnothing$
1	0
2	$-\sigma\,,\sigma$
3	$-\sigma\sqrt{3}\,,0\,,\sigma\sqrt{3}$
4	$-\sigma\sqrt{\sqrt{6}+3}\,,\ -\sigma\sqrt{3-\sqrt{6}}\,,\sigma\sqrt{3-\sqrt{6}}\,,\sigma\sqrt{\sqrt{6}+3}$
5	$-\sigma\sqrt{\sqrt{10}+5}\,,\ -\sigma\sqrt{5-\sqrt{10}}\,,0\,,\sigma\sqrt{5-\sqrt{10}}\,,\sigma\sqrt{\sqrt{10}+5}$

TABLE A.1 – Racines des 5 premiers polynômes $\mathcal{H}^{[\sigma^2]}$.

Remarque A.1.2. Si $x_i^{[\sigma^2]}$ et $x_i^{[1]}$ sont respectivement les i-ièmes racines des polynômes d'Hermite généralisés $\mathcal{H}_n^{[\sigma^2]}$ et probabilistes $\mathcal{H}_n^{[1]}$, alors la relation entre ces racines est simplement :

$$x_i^{[\sigma^2]} = \sigma x_i^{[1]} \tag{A.6}$$

On a donc une affinité de σ entre les racines des $\mathcal{H}_n^{[\sigma^2]}$ et des $\mathcal{H}_n^{[1]}$.

A.2 Tenseurs de Grad-Hermite généralisés

A.2.1 Définition

En dimension $D > 1$, la variable $\boldsymbol{x}$ est un vecteur, les polynômes d'Hermite prennent alors une forme tensorielle, que l'on appellera tenseur de Grad-Hermite [Gra49a]. Ils s'obtiennent de la même manière qu'en une dimension :

$$\boldsymbol{\mathcal{H}}^{[\sigma^2]}_{n\alpha_1\ldots\alpha_n}(\boldsymbol{x}) = \frac{(-1)^n}{w^{[\sigma^2]}(\boldsymbol{x})}\frac{d}{dx_{\alpha_1}}\ldots\frac{d}{dx_{\alpha_n}}w^{[\sigma^2]}(\boldsymbol{x}) \tag{A.7}$$

Avec maintenant :

$$w^{[\sigma^2]}(\boldsymbol{x}) = \frac{1}{(2\pi\sigma^2)^{D/2}}\,exp\left(-\frac{\boldsymbol{x}^2}{2\sigma^2}\right) \tag{A.8}$$

A.2.2 Propriétés

Orthogonalité

La relation d'orthogonalité est alors la suivante :

$$\begin{aligned}\left\langle \mathcal{H}^{[\sigma^2]}_{n\alpha_1\dots\alpha_n}(\boldsymbol{x}) \middle| \mathcal{H}^{[\sigma^2]}_{m\beta_1\dots\beta_m}(\boldsymbol{x}) \right\rangle &= \int_{-\infty}^{\infty} \mathcal{H}^{[\sigma^2]}_{n\alpha_1\dots\alpha_n}(\boldsymbol{x})\, \mathcal{H}^{[\sigma^2]}_{m\beta_1\dots\beta_m}(\boldsymbol{x})\, w^{[\sigma^2]}(\boldsymbol{x})\, dx \\ &= \frac{1}{\sigma^{2n}}\, \delta_{nm} \boldsymbol{\delta}_{n\boldsymbol{\alpha}\boldsymbol{\beta}}, \end{aligned} \tag{A.9}$$

où $\boldsymbol{\delta}_{n\boldsymbol{\alpha}\boldsymbol{\beta}} = 1$ si $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$ est une permutation de $\boldsymbol{\beta} = (\beta_1, \dots, \beta_{m=n})$ et zéro sinon. Le nombre de permutations pour lesquelles $\boldsymbol{\delta}_{n\boldsymbol{\alpha}\boldsymbol{\beta}} = 1$ est $n!$.

Fonction génératrice

On appelle fonction génératrice des Tenseurs de Grad-Hermite $\boldsymbol{\mathcal{H}}^{[\sigma^2]}_n$:

$$\exp\left(\frac{\boldsymbol{x}\boldsymbol{t}}{\sigma^2} - \frac{\boldsymbol{t}^2}{2\sigma^2}\right) = \sum_{n=0}^{\infty} \frac{1}{n!} \bigotimes^n \boldsymbol{t} : \boldsymbol{\mathcal{H}}^{[\sigma^2]}_n(\boldsymbol{x}), \tag{A.10}$$

où $\bigotimes^n \boldsymbol{t} = \underbrace{\boldsymbol{t} \otimes \boldsymbol{t} \otimes \dots \boldsymbol{t} \otimes \boldsymbol{t}}_{n \text{ fois}}$ et où ":" est le produit tensoriel totalement contracté.

Le second membre de cette expression est connu sous le nom de développement de Gram-Charlier. Cette série est convergente si et seulement si $|\boldsymbol{t}| < 1$.

Formules de récurrence

Les polynômes d'Hermite peuvent êtres construit récursivement grâce à la formule suivante :

$$\boldsymbol{\mathcal{H}}^{[\sigma^2]}_{n+1\,\alpha_0\dots\alpha_n}(\boldsymbol{x}) = \frac{1}{\sigma^2} x_{\alpha_0} \boldsymbol{\mathcal{H}}^{[\sigma^2]}_{n\,\alpha_1\dots\alpha_n}(\boldsymbol{x}) - \frac{1}{\sigma^2} \sum_{i=1}^{n} \delta_{\alpha_0\alpha_i} \boldsymbol{\mathcal{H}}^{[\sigma^2]}_{n-1\,\alpha_0\dots\alpha_{i-1}\alpha_{i+1}\dots\alpha_n}(\boldsymbol{x}) \tag{A.11}$$

Avec $\boldsymbol{\mathcal{H}}^{[\sigma^2]}_{-1} = 0$ et $\boldsymbol{\mathcal{H}}^{[\sigma^2]}_{0} = 1$.

A.2.3 Liste des premiers tenseurs d'Hermite généralisés

En voici une liste pour $n = 0 \dots 4$:

$$
\begin{aligned}
\boldsymbol{\mathcal{H}}_{0}^{[\sigma^2]}(\boldsymbol{x}) &= 1 \\
\boldsymbol{\mathcal{H}}_{1\alpha}^{[\sigma^2]}(\boldsymbol{x}) &= \frac{x_\alpha}{\sigma^2} \\
\boldsymbol{\mathcal{H}}_{2\alpha\beta}^{[\sigma^2]}(x) &= \frac{x_\alpha\, x_\beta}{\sigma^4} - \frac{\delta_{\alpha\,\beta}}{\sigma^2} \\
\boldsymbol{\mathcal{H}}_{3\alpha\beta\gamma}^{[\sigma^2]}(\boldsymbol{x}) &= \frac{x_\alpha\, x_\beta\, x_\gamma}{\sigma^6} - \frac{x_\alpha\, \delta_{\beta\gamma} + x_\beta \delta_{\alpha\gamma} + x_\gamma\, \delta_{\alpha\beta}}{\sigma^4} \\
\boldsymbol{\mathcal{H}}_{4\alpha\beta\gamma\varepsilon}^{[\sigma^2]}(\boldsymbol{x}) &= \frac{x_\alpha\, x_\beta\, x_\gamma\, x_\epsilon}{\sigma^8} \\
&\quad - \frac{x_\alpha\, x_\beta\, \delta_{\gamma\epsilon} + x_\alpha\, x_\gamma \delta_{\beta\epsilon} + x_\alpha\, x_\epsilon\, \delta_{\beta\gamma} + x_\beta\, x_\gamma\, \delta_{\alpha\epsilon} + x_\beta\, x_\epsilon\, \delta_{\alpha\gamma} + x_\gamma\, x_\epsilon\, \delta_{\alpha\beta}}{\sigma^6} \\
&\quad + \frac{\delta_{\alpha\beta}\, \delta_{\gamma\epsilon} + \delta_{\alpha\gamma}\, \delta_{\beta\epsilon} + \delta_{\alpha\epsilon}\, \delta_{\beta\gamma}}{\sigma^4}
\end{aligned}
$$

A.3 Quadrature de Gauss-Hermite

La quadrature de Gauss-Hermite est utilisée, pour l'intégration numérique, soit de fonctions gaussiennes, soit pour intégrer des fonctions f sur $\mathbb{R}$ (pour pouvoir faire cela, il faut [2] que $f\left(w^{[\sigma^2]}\right)^{-1/2} \in \mathcal{L}^2(\mathbb{R})$, *i.e.* que f décroisse plus rapidement que $e^{-x^2/4\sigma^2}$ lorsque $x \to \infty$).

A.3.1 Définition

Soit f une fonction de la variable x de décroissance plus rapide que $e^{-x^2/4\sigma^2}$ lorsque $x \to \infty$. La quadrature de Gauss-Hermite permet d'approximer l'intégrale de cette fonction sur $\mathbb{R}$ de la façon suivante :

2. Les polynômes $\mathcal{H}_n^{[\sigma^2]}$ forment une base de l'espace de Hilbert pour les fonctions f telles que $|f|^2 w^{[\sigma^2]}$ soit de carré sommable, *i.e.* telles que

$$f\left(w^{[\sigma^2]}\right)^{-\frac{1}{2}} \in \mathcal{L}^2(\mathbb{R}) \Longleftrightarrow \int_{-\infty}^{\infty} |f(x)|^2\, e^{-\frac{x^2}{2\sigma^2}}\, dx < \infty. \tag{A.12}$$

Dans cet espace, le produit scalaire (Hermitique) de deux fonctions f et g est défini par :

$$< f|g >= \int_{-\infty}^{\infty} f(x)\overline{g(x)}w(x)dx. \tag{A.13}$$

$$\int_{\mathbb{R}} f(x)dx = \int_{\mathbb{R}} w^{[\sigma^2]}(x) \left(\frac{f(x)}{w^{[\sigma^2]}(x)} \right) dx = \sum_{i=1}^{N} \omega_i \left(\frac{f(x_i)}{w^{[\sigma^2]}(x_i)} \right) + \mathcal{O}\left(x^{2N}\right), \quad \text{(A.14)}$$

avec $w^{[\sigma^2]}(x) = \dfrac{1}{\sqrt{2\pi\sigma^2}}\, e^{-x^2/2\sigma^2}$.

Dans cette expression, les $\{x_i\}_{i=1,\ldots,N}$ sont les racines du polynôme d'Hermite $\mathcal{H}_N^{[\sigma^2]}$ de degré N et les $\{\omega_i\}_{i=1,\ldots,N}$ les poids associés.

Une propriété importante est que si la fonction $f/w^{[\sigma^2]}$ est un polynôme de degré $2N$, alors cette quadrature est exacte avec seulement N points d'intégration.

A.3.2 Poids associés à la Quadrature de Gauss-Hermite

Pour une quadrature d'Hermite à N points le poids $\omega_i^{[\sigma^2]}$ associé à la i-ième racine du polynôme généralisé $\mathcal{H}_N^{[\sigma^2]}(x)$, est donné par :

$$\omega_i^{[\sigma^2]} = N!\, \left[N \mathcal{H}_{N-1}^{[\sigma^2]}\left(x_i^{[\sigma^2]}\right) \right]^{-2}\ ,\ i = 1, \ldots, n \quad \text{(A.15)}$$

Où $x_i^{[\sigma^2]}$ est la i-ième racine du polynôme d'Hermite généralisé $\mathcal{H}_N^{[\sigma^2]}(x)$. Ces poids sont indépendants de σ.

Les poids des quadratures jusqu'à l'ordre 5 sont donnés dans le tableau A.2.

N	Poids ω_i
0	$\varnothing$
1	1
2	$1/2\,,1/2$
3	$1/6\,,2/3\,,1/6$
4	$\frac{1}{4\left(\sqrt{6}+3\right)}\,,\frac{1}{4\left(3-\sqrt{6}\right)}\,,\frac{1}{4\left(3-\sqrt{6}\right)}\,,\frac{1}{4\left(\sqrt{6}+3\right)}$
5	$\frac{3}{10\left(\sqrt{10}+2\right)^2}\,,\frac{3}{10\left(\sqrt{10}-2\right)^2}\,,8/15\,,\frac{3}{10\left(\sqrt{10}-2\right)^2}\,,\frac{3}{10\left(\sqrt{10}+2\right)^2}$

TABLE A.2 – Poids des quadratures de Gauss-Hermite de degrés $N = 0 \ldots 5$. Les poids sont donnés dans le même ordre que les racines du tableau A.1.

A.3.3 Quadrature de Gauss-Hermite en dimension D > 1

En dimension D > 1, la quadrature de Gauss-Hermite n'est pas unique, en ce sens que pour construire une quadrature d'un ordre donné, plusieurs ensembles de couples $\{\boldsymbol{x}_i, \omega_i\}_{i=1,\ldots,N}$ sont admissibles (cf. Nie et *al.* [NSC08]). Néanmoins, la voie la plus

naturelle est de construire les quadratures en dimensions supérieures à partir de la quadrature en 1 dimension. Considérons un polynôme p de degré N tel que $p(\boldsymbol{x}) = \prod_{\alpha=1}^{D} x_\alpha^{n_\alpha}$, avec $\sum_{\alpha=1}^{D} n_\alpha = N$. Nous avons alors :

$$\int \omega(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} = \int \omega(\boldsymbol{x}) \prod_{\alpha=1}^{D} x_\alpha^{n_\alpha} d\boldsymbol{x} \tag{A.16}$$

$$= \prod_{\alpha=1}^{D} \int \omega(\boldsymbol{x}) x_\alpha^{n_\alpha} d\boldsymbol{x} \tag{A.17}$$

Calculons alors $\int \omega(\boldsymbol{x}) x_\alpha^{n_\alpha} d\boldsymbol{x}$ par une quadrature à q_α points tel que $n_\alpha \leq q_\alpha/2$, de telle sorte à ce que celle-ci soit exacte, il vient :

$$\int \omega(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} = \prod_{\alpha=1}^{D} \sum_{i_\alpha}^{q_\alpha} \omega_{i_\alpha} x_{i_\alpha}^{n_\alpha} \tag{A.18}$$

$$= \sum_{i_1}^{q_1} \dots \sum_{i_D}^{q_D} \omega_{i_1} \dots \omega_{i_D} \, x_{i_1}^{n_1} \dots x_{i_D}^{n_D} \tag{A.19}$$

On remarque alors qu'il s'agit d'une nouvelle quadrature, dont les poids sont les produits des poids associés aux x_{i_α} des différentes quadratures 1D. Les points d'intégration en dimension D sont :

$$\{\boldsymbol{x}_{i_1 \dots i_D} = (x_{i_1}, \dots, x_{i_D}) \mid i_1 = 1, \dots, q_1 \, ; \dots \, ; i_D = 1, \dots, q_D\}, \tag{A.20}$$

et les poids qui leur sont associés :

$$\{\omega_{i_1 \dots i_D} = \omega_{i_1} \dots \omega_{i_D} \mid i_1 = 1, \dots, q_1 \, ; \dots \, ; i_D = 1, \dots, q_D\}. \tag{A.21}$$

Leur nombre est $\prod_{\alpha=1}^{D} q_\alpha$ et si le nombre de points par direction est le même, il y en aura simplement q^D.

B Nombres sans dimension et ordre de grandeur

B.1 Nombre de Knudsen

On définit le nombre de Knudsen, noté K_n, par :

$$K_n \equiv \frac{\text{Libre parcours moyen des particules du fluide}}{\text{Dimension caractéristique de l'écoulement}} = \frac{\lambda}{L}. \tag{B.1}$$

La valeur du nombre K_n caractérise la capacité du fluide à s'éloigner de l'équilibre thermodynamique local (ETL) :

- Si $K_n \ll 1$, *i.e.* si $\lambda \ll L$, alors les collisions dominent le fluide qui ne peut s'éloigner de l'ETL. L'écoulement est qualifié d'hydrodynamique.
- Si $K_n \gg 1$, *i.e.* si $\lambda \gg L$, alors les collisions sont négligeables et le fluide peut s'éloigner énormément de l'ETL. L'écoulement est qualifié d'écoulement de particules libres.
- Si $K_n \sim 1$, *i.e.* si $\lambda \sim L$ on parle de régime intermédiaire. Le fluide peut s'éloigner de l'ETL, mais pas trop car les collisions tendent à l'en empêcher.

B.1.1 Libre parcours moyen

Pour les gaz parfaits dilués, le libre parcours moyen λ peut être calculé par la relation :

$$\lambda = \frac{1}{n\,\sigma}, \tag{B.2}$$

où n est la densité numérique de particules et $\sigma = \pi\, r^2$ leur section efficace de collision, avec r le rayon moléculaire classique [1].

La loi des gaz parfaits nous donne de plus :

$$n = \frac{p}{k_B T}, \tag{B.3}$$

où p est la pression (Pa), T est la température du fluide (K) et k_B la constante de

1. Le rayon classique des atomes est de l'ordre de l'Angström ($1\,\text{Å} = 1.10^{-10}\,m$).

Boltzmann ($k_B = 1{,}38.10^{-23}$ J/K). Finalement, le libre parcours moyen pourra aussi être évalué par :

$$\lambda = \frac{k_B T}{\sigma p}. \tag{B.4}$$

B.1.2 Expression temporelle du nombre de Knudsen

De manière générale, le nombre de Knudsen représente le degré de séparation entre les échelles micro/macro. Dans la première définition (B.1) du nombre de Knudsen, nous avons choisi de comparer deux échelles de longueurs, λ et L. Mais nous pourrions tout aussi bien utiliser deux échelles de temps, sachant qu'une longueur caractéristique l_c est reliée à un temps caractéristique t_c par l'intermédiaire d'une vitesse caractéristique v_c : $l_c = v_c\, t_c$. Donc, si nous trouvons une vitesse v_c indépendante de l'échelle, nous pourrons alors exprimer le nombre de Knudsen sous la forme :

$$K_n = \frac{\lambda}{L} = \frac{v_c\, \tau_m}{v_c\, \tau_M} = \frac{\tau_m}{\tau_M}, \tag{B.5}$$

où τ_m et τ_M sont respectivement les temps caractéristiques d'évolution microscopique et macroscopique. La vitesse du son c_s (cf. section suivante) constitue une telle vitesse caractéristique. Dans ce cas :

$$\tau_m = \frac{\lambda}{c_s}, \tag{B.6}$$

représente le temps moyen entre deux collisions particule-particule et

$$\tau_M = \frac{L}{c_s}, \tag{B.7}$$

le temps nécessaire à une onde acoustique pour parcourir la longueur caractéristique L de l'échelle macroscopique.

En outre, l'inverse de τ_m représente la fréquence de collisions entre les particules du fluide, notée $\omega_{coll} = c_s/\lambda$. C'est aussi le temps de relaxation vers l'équilibre de l'opérateur de collision BGK. D'autre part, l'inverse de $\tau_M^{-1} = c_s/L$ correspondra à la fréquence du mode de résonance fondamental de la "cavité" de taille L.

B.1.3 Ordres de grandeur

Pour de l'air dans des conditions standards, $\lambda \approx 70\,\text{nm}$. Pour un tel gaz, un Knudsen de 10 donne $L \approx 7\,nm$ et un Knudsen de 0,01, correspondant à la limite du régime hydrodynamique, donne $L \approx 7\,\mu m$.

B.2 Nombre de Mach

On définit le nombre de Mach, noté M_a, par :

$$M_a \equiv \frac{\text{Vitesse caractéristique de l'écoulement}}{\text{Vitesse du son}} = \frac{u}{c_s}, \tag{B.8}$$

ou encore :

$$M_a \equiv \frac{\text{Énergie cinétique}}{\text{Énergie interne}} = \frac{1/2\ \rho u^2}{D/2\ \rho k_B T} = \frac{u^2}{D\, k_B T}, \tag{B.9}$$

avec ρ la masse volumique du fluide, T sa température et D la dimension de l'espace.

B.2.1 Vitesse du son

La vitesse du son tient une place importante en théorie cinétique. Elle est la vitesse à laquelle se propage une perturbation de la pression.

Elle est définie de la façon suivante :

$$c_s = \sqrt{\frac{\gamma p}{\rho}}, \tag{B.10}$$

avec $p = \rho\, \theta$ est la pression thermodynamique, ρ la densité massique, $\theta = k_B T/m$ la "température", et γ l'*indice adiabatique* du fluide.

L'indice γ est tel que si la transformation d'un gaz parfait est adiabatique, alors $pV^\gamma = C^{te}$. Il peut être calculé par $\gamma = C_p/C_v$, où C_v et C_p sont respectivement les capacités calorifiques à volume constant et à pression constante.

Dans le cadre de la physique statistique, on montre (cf. [Lan90, BCDB89] par exemple) que pour les gaz parfaits, $C_v = g/2$ et $C_p = (g+2)/2$, où g est le nombre de degrés de liberté d'une particule du fluide. Nous avons donc $\gamma = (g+2)/g$. Si la gaz est monoatomique, alors $g = D$, la dimension de l'espace. D'où, pour un gaz parfait monoatomique :

$$c_s = \sqrt{\frac{D+2}{D}\theta}. \tag{B.11}$$

La vitesse du son ne devra pas être confondue avec la vitesse quadratique moyenne c_0 des particules qui composent le fluide, qui est donnée par :

$$c_0 = \sqrt{\langle (v - \langle v \rangle)^2 \rangle} = \sqrt{D\, \theta}, \tag{B.12}$$

où v est la vitesse microscopique et $\langle \bullet \rangle$ est l'opérateur (3.13) défini aux section 2.2.3. Néanmoins, ces deux vitesses s'égalent pour $D = 2$.

Ordres de grandeur

Pour de l'air dans des conditions standard, la vitesse du son $c_s \approx 330\, ms^{-1}$. Pour les écoulements qui nous intéressent, à savoir des écoulements de gaz dans des canaux de taille micrométrique ($10^{-6}\, m$) à nanométrique ($10^{-9}\, m$), les vitesses débitantes sont au maximum de l'ordre du centimètre par seconde ($10^{-2}\, m\, s^{-1}$). Pour ces écoulements, le nombre de Mach est donc au maximum de l'ordre de $M_a \sim 10^{-4}$.

B.3 Nombre de Reynolds

On définit le nombre de Reynolds, noté R_e, par :

$$R_e \equiv \frac{\text{Vitesse caractéristique} \times \text{Dimension caractéristique de l'écoulement}}{\text{Viscosité (cinématique) du fluide}} = \frac{uL}{\nu}, \tag{B.13}$$

ou encore :

$$R_e \equiv \frac{\text{Forces d'inerties}}{\text{Forces visqueuses}} = \frac{\rho u^2/L}{\mu u/L^2}, \tag{B.14}$$

avec ρ est la densité massique, μ la viscosité dynamique et $\nu = \mu/\rho$ la viscosité cinématique du fluide.

Ordres de grandeur

Pour de l'air dans des conditions standard, la viscosité cinématique est $\nu_{air} = 15.6\, 10^{-6}\, m^2\, s^{-1} \sim 10^{-5}\, m^2\, s^{-1}$. Et comme dit dans la section précédente, nous nous intéressons à des écoulements de gaz, dans des canaux de taille micrométrique ($10^{-6} - 10^{-4}\, m$) à manométrique ($10^{-9}\, m$), dont la vitesse débitante est au maximum de l'ordre du centimètre par seconde ($10^{-2}\, m\, s^{-1}$). Dans ces conditions, le nombre de Reynolds est de l'ordre de $R_e \sim 10^{-1}$ à $R_e \sim 10^{-6}$, selon les cas. Dans tous les cas, nous aurons $R_e \lesssim 1$ caractéristique des écoulements rampants.

B.4 Relation entre K_n, M_a et R_e

En utilisant l'expression de la viscosité cinématique obtenue par la méthode du libre parcours moyen (cf. [MN69]), à savoir :

$$\nu = \frac{1}{3}\lambda \bar{v}, \tag{B.15}$$

où

$$\bar{v} = \sqrt{\frac{8k_BT}{\pi m}} = \sqrt{\frac{8}{\pi}}\, c_s \tag{B.16}$$

est la vitesse thermique. On obtient l'expression suivante pour le nombre de Reynolds :

$$R_e = \frac{3uL}{\lambda \bar{v}} = 3\left(\frac{u}{\bar{v}}\right)\frac{L}{\lambda} = 3\left(\frac{u}{\bar{v}}\right)\frac{1}{K_n}. \tag{B.17}$$

En remplaçant $\bar{v}$ par son expression (B.16), on obtient :

$$R_e = 3\,\frac{1}{K_n}\sqrt{\frac{\pi}{8}}\frac{u}{c_s}. \tag{B.18}$$

Finalement, on obtient la relation suivante entre K_n, M_a et R_e :

$$K_n = \sqrt{\frac{9\pi}{8}}\,\frac{M_a}{R_e}. \tag{B.19}$$

Ce résultat montre que les écoulement hors d'équilibre se rencontrent généralement pour de grands nombres de Mach et pour de petits nombres de Reynolds.

B.5 Nombre de Prandtl

On définit le nombre de Prandtl, noté P_r, par :

$$P_r \equiv \frac{\text{Diffusion de quantité de mouvement}}{\text{Diffusion de chaleur}} = \frac{\nu}{\alpha}. \tag{B.20}$$

Pour les gaz, notamment pour l'air, le nombre de Prandtl est de l'ordre de 0,6 - 0,7 et ne varie que très peu avec la température. En outre, le développement de Chapman-Enskog pour des gaz monoatomiques avec l'opérateur de collision de Boltzmann, donne $2/3 \sim 0{,}66$. Le modèle BGK, à simple temps de relaxation, donne quant à lui, $P_r = 1$.

C Codes

C.1 Mise en œuvre de la méthode en une dimension

C.1.1 Quadrature de Gauss-Hermite

Le code suivant, écrit en langage scilab, permet de construire les quadratures de Gauss-Hermite. Le lecteur intéressé pourra réaliser les simulations discutées dans le texte en se procurent de logiciel de calcul *scilab*, qui est libre et multiplateforme, sur le site : http://www.scilab.org/.

```
1  //===========================================================
2  // Fichier  : GaussHermite.sci
3  // Language : Scilab.
4  //
5  // Quadrature de Gauss-Hermite.
6  //===========================================================
7
8  //*********************************************************
9  // Fonction hermite()
10 //*********************************************************
11 //***  Renvoie le polynome d'Hermite de degre N
12 //***
13 function [Hn] = hermite(N)
14 //     Ajout de 1 a N pour compenser
15 //     le fait que les tableaux sont indicés a
16 //     partir de 1 en scilab
17   N=N+1;
18
19   // H est la matrice des coefficients polynomiaux
20   H = eye(N,N);
21
22   // Remplissage de la premiere colonne de H
23   for n = 3:N,
24     H(n,1) = -((n-1)-1)*H(n-2,1);
25   end;
26
27   // Remplissage du reste de H
28   for n = 3:N,
29     for i = 2:n,
30       H(n,i) = H(n-1,i-1) - ((n-1)-1)*H(n-2,i);
31     end;
32   end;
33
34   // Sortie de la fonction
```

```
35   Hn = poly(H(N,:),'x','coeff');
36 endfunction
37
38 //*********************************************************
39 // Fonction racines()
40 //*********************************************************
41 //*** Renvoie les N racines du polynome
42 //*** d'Hermite de degre N
43 //***
44 function [xi] = racines(Hn)
45   xitmp = gsort(roots(Hn)); // Racines dans l'ordre décroissant
46   N = degree(Hn);
47   for i = 1:length(xitmp),
48     xi(i) = xitmp((N+1)-i); // Racines dans l'ordre croissant
49   end;
50 endfunction
51
52 //*********************************************************
53 // Fonction poids()
54 //*********************************************************
55 //*** Renvoie le i-eme poids de la quadrature de
56 //*** Gauss-Hermite a N points
57 //***
58
59 //    Cette fonction prend comme argument le polynome
60 //    d'Hermite Hn (dans sa forme probaliliste) et renvoie
61 //    les poids de la quadrature de G-H associée
62 function [wi] = poids(Hn)
63   // N est le degré de Hn
64   N = degree(Hn);
65   wi = zeros(N);
66   // xi contient les racines de Hn
67   xi = racines(Hn)
68   // Hn1 est le polynome d'hermite de degre N-1
69   Hn1 = hermite(N-1);
70   // Boucle de calcul des poids
71   for i=1:N,
72     wi(i) = factorial(N)*(1/(N*horner(Hn1,xi(i)))^2);
73   end; //end_for
74 endfunction
```

C.1.2 Gaz de Knudsen unidimensionnel

Le code *Gaz_Kn.sci*, écrit en scilab, simule la relaxation d'un gaz de Knudsen (sans collision) en une dimension. Le transport est ici réalisé par méthode spectrale, le système est donc périodique. La condition initiale est une perturbation gaussienne de la densité, une vitesse moyenne nulle et une température uniforme. Ce code permet de réaliser les simulations présentées section 6.1 où l'importance du nombre de vitesses discrètes, ainsi que l'amélioration apportée par les quadratures composites ont été montrées.

```
//=========================================================
// Fichier  : Gaz_Kn.sci
// Language : Scilab.
//
// Résolution de l'equation Boltzmann 1D sans collision.
//=========================================================
clear;

//*** Function feq()
//*** Fonction de distribution d'équilibre.
// Liste des arguments :
// Mi : Moments de f.
//       - M0 : Densité.
//       - M1 : Impulsion.
//       - M2 : Energie totale.
// ci : Noeuds des l'espace des vitesses.
// wi : Poids de la quadrature.
function [fi] = feq(M0,M1,M2,cw)
  rho = M0;
  u = M1./rho;
  theta = (M2 - (u.^2)./rho)./rho;

  // feq pour N=2
  deff( '[y] = maxwellienne(c)', ...
        '[y] = rho.*(1.+ c*u + (c*u)^2/2. + (theta-1)*(c^2-1)/2. -u^2/2.)');

  for i= 1:length(cw(1,:)),
    fi(:,i) = cw(2,i).*maxwellienne(cw(1,i));
  end;
endfunction

//*******************************************************
// Initialisation
//*******************************************************

// Définition des paramètres de la simulation
// Mi0  : Moments initiaux
//       - M00 : Densité.
//       - M10 : Impulsion.
//       - M20 : Energie totale.
// tau : Temps de relaxation vers l'équilibre.
// Nx   : Nombre de noeuds en configurations.
// L    : Longueur de l'espace des configuration.
// x : Variable de configurations (position).
// Nv   : Nombre de noeuds en vitesses.
// k    : Variable de Fourier.
// k0   : Vecteur d'onde du mode fondamantal.
// dt   : Pas de temps.
// Tf   : Temps discret final.

Nv = 9;
//*** Quadrature de G-H.
exec("GaussHermite.sci");
// Quadrature "pure"
cw = [racines(hermite(Nv))' ; poids(hermite(Nv))'];
```

```
56
57 // Quadrature composite
58 // cw = [[racines(hermite(Nv-1)); racines(hermite(Nv))]'; ...
59 // [poids(hermite(Nv-1))./2;poids(hermite(Nv))./2]'];
60 // Nv=2*Nv-1;
61
62 L  = 50;
63 Nx = 2^8;
64 x  = L*[1:Nx]'/Nx;
65 dx = L/Nx;
66 k0 = 2*%pi/L;
67 k  = k0*[-[0:Nx/2],[Nx/2-1:-1:1]]';
68 Tf = Nx/2;
69 dt = (L/Nx)/(cw(1,length(cw(1,:)))); // Condition CFL (dx/dt = c_{max})
70
71 // ** Definition des moments initiaux ** //
72
73 //=== Densité initiale (M0 = rho).
74 // Densité de référence :
75 rho0  = 1.;
76 // Perturbation gaussienne d'amplitude alpha de la densité initiale :
77 alpha = .01; M0 = rho0 * (1. + alpha.*exp(-((x-L/2).^2)/(2)));
78
79 //=== Impulsion initiale (M1 = rho*u0).
80 u0 = .0; // Vitesse initiale
81 M1 = M0.*u0;
82
83 //=== Energie totale initiale (M2 = D*rho*theta + rho*u^2)
84 theta0 = 1.; // "Température" de référence (imposée à 1 du fait de la quadrature)
85 M2 = M0*theta0 + (M1.^2)./M0;
86
87 //***************************************
88 // Simulation
89 //***************************************
90
91 // Initialisation de la fonction de distribution.
92 f = feq(M0,M1,M2,cw);
93
94 // Boucle principale.
95 for t = 1:Tf,
96   // Transport //
97   for i=1:Nv,
98     fk(:,i) = fft(f(:,i));
99     f(:,i)  = real(ifft(fk(:,i).*exp(%i*k*cw(1,i)*dt)));
100  end;
101 end; // Fin de la boucle principale.
```

C.1.3 Boltzmann-BGK unidimensionnel

Le code Le code *boltzmannBGK.sci*, écrit en scilab, simule la diffusion de la chaleur et de la matière en une dimension. Le transport est réalisé par méthode spectrale, le système est donc périodique. La condition initiale est une perturbation gaussienne

de la densité, une vitesse moyenne nulle et une température telle que la pression soit constante. Ce code permet de réaliser les simulations présentées section 6.2.1 où la validation du coefficient de conductivité thermique a été montrée.

```
//=========================================================
// Fichier : boltzmannBGK.sci
// Language : Scilab
//
// Résolution de l'equation Boltzmann-BGK 1D
//=========================================================
clear;

//*** Function feq()
//*** Fonction de distribution d'équilibre.
// Liste des arguments :
// Mi : Moments de f.
//       - M0 : Densité.
//       - M1 : Impulsion.
//       - M2 : Energie totale.
// ci : Noeuds des l'espace des vitesses.
// wi : Poids de la quadrature.
function [fi] = feq(M0,M1,M2,cw)
rho = M0;
u = M1./rho;
theta = (M2 - (u.^2)./rho)./rho;

// feq pour N=0
//deff( '[y] = maxwellienne(c)', '[y] = rho');

// feq pour N=1
//deff( '[y] = maxwellienne(c)', '[y] = rho.*(1.+ c*u)');

// feq pour N=2
deff( '[y] = maxwellienne(c)', ...
'[y] = rho.*(1.+ c*u + (c*u)^2/2. + (theta-1)*(c^2-1)/2. -u^2/2.)');

// feq pour N=3
// deff( '[y] = maxwellienne(c)', ...
//       '[y] = rho.*(1.+ c*u + (c*u)^2/2. +   (theta-1.)*(c^2-1)/2. -u^2/2. + ...
//       (c*u)^3/6.- (c*u).*u.^2/2. + (theta-1.).*(c.^2-2).*(c.*u)./2.)');

for i= 1:length(cw(1,:)),
  fi(:,i) = cw(2,i).*maxwellienne(cw(1,i));
end;
endfunction

//*******************************************************
//*** Initialisation
//*******************************************************

// Définition des paramètres de la simulation
// Mi   : Moments initiaux
//       - M0 : Densité.
//       - M1 : Impulsion.
```

```
51 //       - M2 : Energie totale.
52 // tau : Temps de relaxation vers l'équilibre.
53 // Nx   : Nombre de noeuds en configurations.
54 // L    : Longueur de l'espace des configurations.
55 // x    : Variable de configurations (position).
56 // dx   : Pas d'espace.
57 // Nv   : Nombre de noeuds en vitesses.
58 // k    : Variable de Fourier.
59 // k0   : Vecteur d'onde du mode fondamental.
60 // dt   : Pas de temps.
61 // Tf   : Temps discret final.
62 // Nv   : Nombre de vitesses discretes.
63
64 //*** Quadrature de G-H.
65 exec("GaussHermite.sci");
66 Nv = 5;
67 //* Schemas 'Pures' :
68 //cw = [racines(hermite(Nv))'; poids(hermite(Nv))'];
69
70 //* Schemas composites :
71 cw = [[racines(hermite(Nv-1)); racines(hermite(Nv))]' ; [poids(hermite(Nv-1))./2;poids(hermite(Nv))./2]'];
72 Nv=2*Nv-1; // Nombre de vitesses discretes du schema composites.
73
74 L  = 30;
75 Nx = 2*L;
76 x  = L*[1:Nx]'/Nx;
77 dx = L/Nx;
78 k0 = 2*%pi/L;
79 k  = k0*[-[0:Nx/2],[Nx/2-1:-1:1]]';
80 Tf = 5*Nx;
81 dt = (L/Nx)/(cw(1,length(cw(1,:)))); // Condition CFL : dx/dt <= c_{max} (= ici).
82 tau = .5*dt*ones(Nx,Nv);
83
84 // ** Definition des moments initiaux ** //
85
86 //=== Densité initiale (M00 = rho).
87 rho0  = 1.; // Densité de référence.
88 // Densité initiale constante :
89 //M0 = rho0;
90
91 // Perturbation gaussienne d'amplitude alpha de la densité initiale :
92 alpha = .01; M0 = rho0 * (1. - alpha.*exp(-(((x)-L/2).^2)/(4)));
93
94 // Perturbation sinusoidale d'amplitude alpha de la densité initiale :
95 //alpha = .1; M00 = rho0 * (1 + alpha*sin(2*%pi*(x-1)/(L)));
96
97 //=== Impulsion initiale.
98 // M1 = rho*u0
99 u0 = .0; // Vitesse initiale
100 M1 = M0.*u0;
101
102 //=== Energie totale initiale.
103 theta0 = 1.; // 'Température' de référence (imposée à 1 du fait de la quadrature).
104 // M2 = D*rho*theta + rho*u^2
105 // Rmq. : theta = k_B * T / m   et D = 1 (Dimension de l'espace).
```

```
106
107 // Condition d'équilibre thermodynamique :
108 //M2 = M0*theta0 + (M1.^2)./M0; theta = (theta0./M0).*M0;
109
110 // Condition de Pression constante :
111 theta = theta0./M0; M2 = M0.*theta + (M1.^2)./M0;
112
113 // Diffusivité thermique.
114 D = zeros(Nx,1);
115
116 // Variables pour la résolution de l'équation de la chaleur :
117 temp = theta; tempTmp = temp;
118
119 //*********************************************************
120 //*** Simulation
121 //*********************************************************
122 set figure_style new;
123 fig=gcf();
124 fig.pixmap='on';
125 clf();
126
127 // Initialisation de la fonction de distribution.
128 f = feq(M0,M1,M2,cw);
129
130 unix_g('rm therm.res')
131 fd = mopen("./therm.res",'w');
132 mfprintf(fd,'#Nv=%d, Tau=%f, dt=%f\n',Nv,tau,dt);
133
134 // Boucle principale
135 for t = 1:Tf,
136
137   //** Résolution de l'équation de Boltzmann-BGK
138   // Transport
139   for i=1:Nv,
140     fk(:,i) = fft(f(:,i));
141     f(:,i)  = real(ifft(fk(:,i).*exp(%i*k*cw(1,i)*dt)));
142   end;
143
144   // Calcul des Moments
145   M0 = sum(f,'c');
146
147   for i=1:Nx,
148     M1(i) = sum((cw(1,:)).*f(i,:),'c');
149   end;
150
151   for i=1:Nx,
152     M2(i) = sum(((cw(1,:)).^2).*f(i,:),'c');
153   end;
154
155   // Calcul du temps de relaxation local
156   theta = (M2 - M1.*M1)./M0;
157   for i=1:Nv,
158     tau2(:,i) = (M0.*theta./(rho0*theta0)).*(tau(i) + dt/2.);
159   end;
160
```

```
161   // Collisions
162   f = f - (dt./tau2).*(f - feq(M0,M1,M2,cw));
163
164   //** Résolution de l'équation de la chaleur
165   // Calcul de la diffusivite thermique locale
166   D = tau2(:,1)-dt./2.;
167
168   tempTmp(1) = temp(1) + temp(1)*D(1)*(dt/dx^2)*(temp(2) + temp(Nx) -2*temp(1));
169   tempTmp(Nx) = temp(Nx) + temp(Nx)*D(Nx)*(dt/dx^2)*(temp(1) + temp(Nx-1) -2*temp(Nx));
170   for i=2:(Nx-1),
171     tempTmp(i)  = temp(i) + temp(i)*D(i)*(dt/dx^2)*(temp(i+1) + temp(i-1) -2*temp(i));
172   end
173   temp = tempTmp;
174
175   //** Ecriture du résultat et affichage
176   mfprintf(fd,'#Index %d, t = %f\n',t,(t-1)*dt);
177   for i=1:Nx,
178     mfprintf(fd,'%f %f %f\n',i*dx,(M2(i) - M1(i)*M1(i))./M0(i),theta(i));
179   end
180   fprintf(fd,'\n\n');
181
182   delete(gce()); plot(x,theta,'.',x,temp); show_pixmap;
183
184 end; // Fin de la boucle principale.
185
186 mclose (fd);
```

C.2 Mise en œuvre de la méthode – Code maBLB

La méthode numérique présentée au chapitre 5 a été mise en œuvre sous forme d'une bibliothèque lors de cette thèse. Cette bibliothèque, nommée *maBLB* (**ma**ny **B**eams **L**attice **B**oltzmann), est écrite en langage C++. Elle permet de réaliser des simulations d'écoulements de gaz, dans des géométries complexes, en 2 ou 3 dimensions, et sur une large gamme de nombres de Knudsen.

Cette librairie utilise la librairie C++ Boost [1], les objets de cette librairie commencent par `boost::`, définissant leur espace de nom.

C.2.1 En-tête de maBLB

Le fichier d'en-tête de la librairie maBLB (*maBLB.hpp*) devra être inclus dans chaque programme de simulation. Son contenu est le suivant :

```
1 /// \file    maBLB.h
2 /// \author Léonard Izarra <leonardizarra@gmail.com>
3 /// \date    Tue Jun 22 18:46:36 2010
```

1. http://www.boost.org/

```
4 /// \brief   maBLB headers file.
5
6 #ifndef _MABLB_H_
7 #define _MABLB_H_
8
9 #include "globals.hpp"
10 #include "geometry.hpp"
11 #include "vspace.hpp"
12 #include "lattice.hpp"
13
14 #endif /* _MABLB_H_ */
```

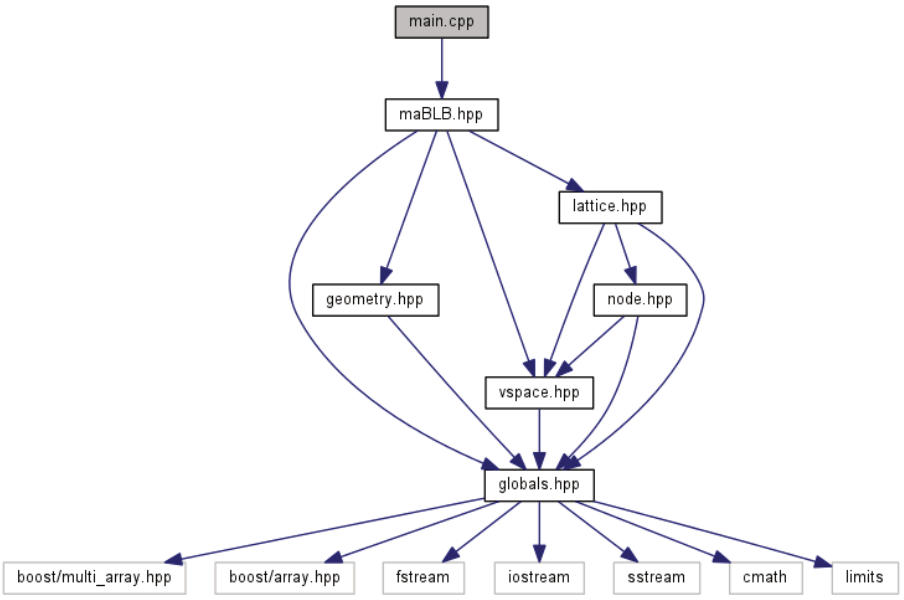


FIGURE C.1 – Graphe d'inclusion des fichiers d'en-têtes (.hpp).

La figure C.1 montre le graphe d'inclusion des fichiers d'en-têtes pour un programme *main.cpp*, incluant la librairie maBLB, on y trouve :

- ***maBLB.hpp*** le fichier d'en-tête de la librairie.
- ***geometry.hpp*** contient la définition de la *classe Geom*. Cette classe contient un ensemble de méthodes qui permettent de préparer la géométrie d'un problème, *i.e.* détecter les bords, trouver les normales à ces bords, etc. Le code en est donné section C.2.5, avec un exemple d'utilisation particulier, où une géométrie est générée à partir d'une image au format png.
- ***lattice.hpp*** contient la définition de la *classe Lattice*. Le code en est donné section C.2.6.
- ***node.hpp*** contient la définition de la *classe abstraite Node* ainsi que celle de ses classes héritières *Fluid_Node*, *Bound_Node* et *Solid_Node*. Le code en est donné section C.2.7.
- ***vspace.hpp*** contient la définition de la *classe Vspace*. Cette classe contient l'espace des vitesses. Le code en est donné section C.2.8.

– ***globals.hpp*** contient des définitions globales, notamment les types de nœuds et les inclusions d'en-têtes de librairies installées sur le système. Le code en est donné section C.2.9.

C.2.2 Exemples d'écoulements de Poiseuille

Deux exemples d'utilisation de maBLB sont donnés ici. Il s'agit des deux écoulements de Poiseuille plans présentés sections 6.3 et 6.5, dont l'un est mis en mouvement par une force extérieure et l'autre par un gradient de densité. Les programmes principaux de ces exemples sont donnés ci-après.

C.2.3 Poiseuille plan – Force extérieure

Ce premier exemple est donc celui d'un écoulement de Poiseuille plan mis en mouvement par une force extérieure. La technique d'application de la force extérieure est énoncée section 5.8. Le graphe des appels de fonctions pour cet exemple est donné figure C.2.

La boucle principale de calcul consiste à itérer la méthode `propagate_and_collide()` de la classe `L` déclarée ligne 81. L'action de cette méthode est de transporter puis faire collisionner la fonction de distribution.

L'essentiel du programme consiste à construire et initialiser la simulation, comme on le voit sur la figure C.2. La construction de la simulation est réalisée par la fonction `Create_geometry()` qui utilise la classe `Geom` décrite section C.2.5. La géométrie est alors chargée dans la "lattice" `L` par la méthode `load_geometry()` (ligne 87). Il est important de noter que le domaine d'intégration est par défaut périodique, comme on peut le voir à la lecture de la méthode `propagate()` (ligne 293) de la classe `Lattice` (cf. section C.2.6). Cette périodicité peut être rompue en plaçant des nœuds `Solid` aux extrémités du domaine. Dans l'exemple de la section suivante, la méthode `box()` de la classe `Geom` place des nœuds `Solid` tout autour du domaine et rompt ainsi la périodicité dans les deux directions de l'espace.

De même que pour l'espace des configurations, l'espace des vitesses est chargé par la méthode `load_velocity_space()` (ligne 90), alias dans la classe `Lattice<TD,TN>` de la méthode `load()`, membre de la classe `Vspace<TD,TN>`.

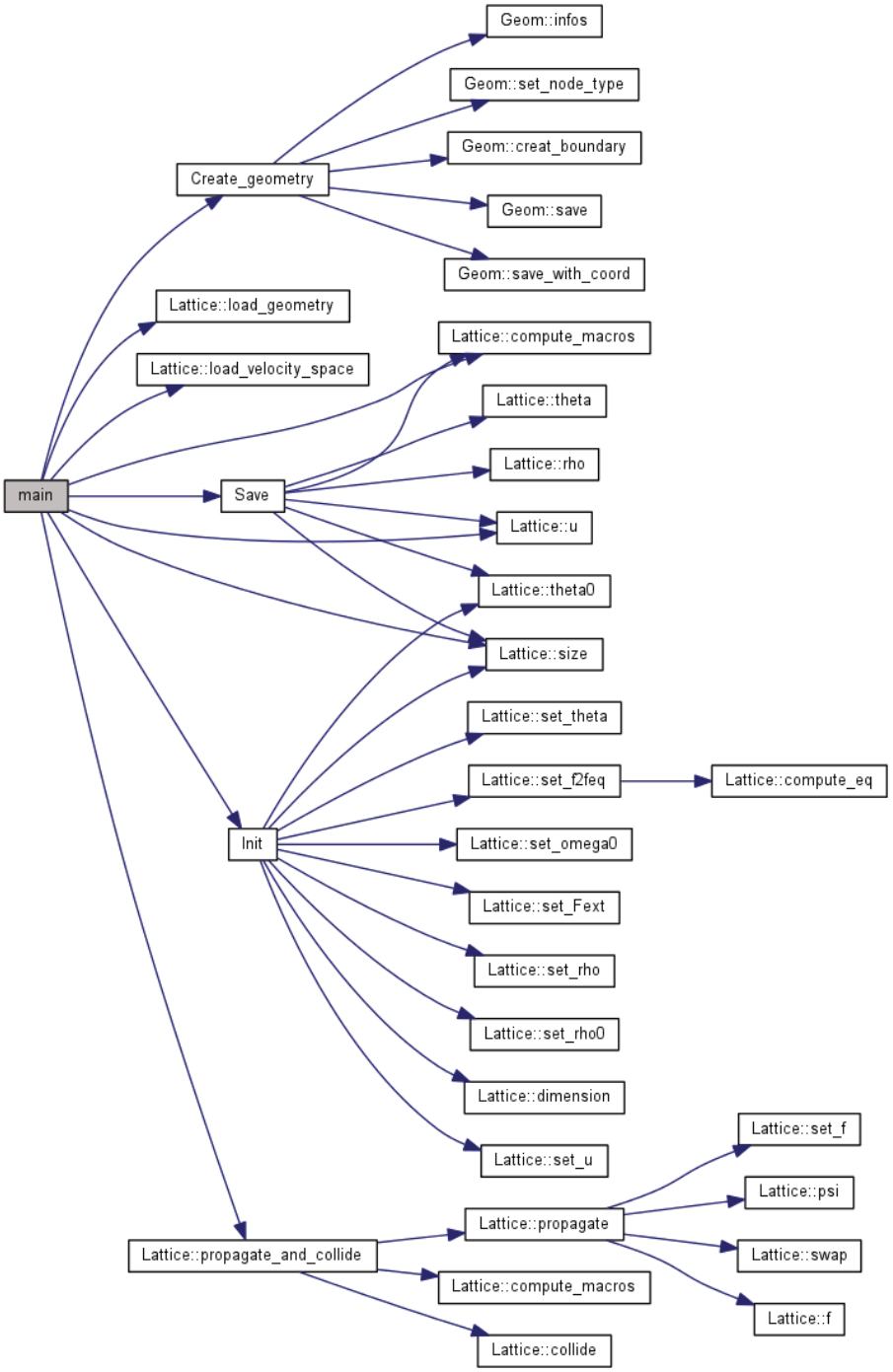


FIGURE C.2 – Graphe des appels aux fonctions pour l'exemple du Poiseuille plan soumis à une force extérieure (fichier poiseuille_fext.cpp).

S'ensuit l'initialisation de `L`, réalisée dans la fonction
`Init (Lattice<TD,TN> & Lattice, double Kn, double Ma)` (ligne 30).
Cette fonction prend en argument une référence sur un objet de type `Lattice<TD,TN>` (nous lui passerons `L`) et les nombres de Knudsen et de Mach. Ces deux nombres permettent alors de calculer le temps de relaxation et la force extérieure à appliquer au fluide. En effet, pour un écoulement de Poiseuille, de vitesse maximale `uMax` et à la température `theta0`, le nombre de Mach est tel que : `Ma` = `uMax` / $\sqrt{\texttt{theta0}}$, d'où la définition de la ligne 33. D'autre part, la relation (6.28) entre viscosité et nombre de Knudsen discutée 3.5.2 nous permet d'écrire `nu0` ligne 34, utilisé ligne 39 pour définir la fréquence de collision grâce à la méthode `set_omega()`. La force extérieure (discutée section 4.3.1) est ensuite initialisée grâce à la méthode `set_Fext()` (ligne 40).

Finalement, avant d'imposer l'équilibre comme condition initiale partout dans le domaine, grâce à la méthode `set_f2feq()` (ligne 51), les champs de densité, densité de référence, de vitesse et de température sont initialisés respectivement à 1., 1., 0. et `theta0`.

La fonction `Save()` (ligne 57) écrit du résultat dans des fichiers. Cette fonction fait appel aux méthodes d'accès aux champs `rho()`, etc. Signalons qu'avant d'appeler une de ces méthodes d'accès, les champs doivent être calculés par appel à la méthode `compute_macros()`, ce qui est fait ligne 66.

```
1 /**
2  * @file    poiseuille_fext.cpp
3  * @author Léonard de Izarra <leonardizarra@gmail.com>
4  * @date    Wed Jan 12 21:08:31 2011
5  */
6
7 #include "maBLB.hpp"
8
9 using namespace std;
10
11 const size_t TD=2, TN=16;
12 const size_t Lx=4, Ly=15;
13
14 void Create_geometry () {
15   Geom G(Lx,Ly,1);
16
17   for (size_t iX = 0; iX != Lx; ++iX) {
18     for (size_t iY = 0; iY != Ly; ++iY) {
19       if ( iY == 0 || iY == Ly-1)  G.set_node_type(iX, iY, 0, Solid);
20       else G.set_node_type(iX, iY, 0, Fluid);
21     } //~ for iY
22   } //~ for iX
23
24   G.creat_boundary();
25   G.save("in_geom.txt");
26   G.save_with_coord("coord_geom.txt");
27   G.infos();
28 } //~ Geometry ()
```

```
29
30 void Init (Lattice<TD,TN> & Lattice, double Kn, double Ma) {
31
32   double theta0 = Lattice.theta0(0,0,0);
33   double uMax = Ma * std::sqrt(theta0);
34   double nu0 = Kn * ((double)Ly-2.) * std::sqrt(theta0) / std::sqrt(2.5);
35
36   for (size_t iX = 0; iX != Lattice.size(0); ++iX) {
37     for (size_t iY = 0; iY != Lattice.size(1); ++iY) {
38
39       Lattice.set_omega0(iX, iY, 0, 1./(nu0/theta0 + .5));
40       Lattice.set_Fext(iX, iY, 0, 0, 8. * uMax * nu0 /(((double)Ly-2.)*((double)Ly)-2.));
41       Lattice.set_Fext(iX, iY, 0, 1, 0.);
42       Lattice.set_rho(iX, iY, 0, 1.);
43       Lattice.set_rho0(iX, iY, 0, 1.);
44
45       for (size_t axis = 0; axis != Lattice.dimension(); ++axis) {
46         Lattice.set_u(iX, iY, 0, axis, 0.);
47       } //~ for axis
48
49       Lattice.set_theta(iX, iY, 0, Lattice.theta0(0,0,0));
50
51       Lattice.set_f2feq(iX, iY, 0);
52     } //~ for iY
53   } //~ for iX
54
55 } //~ Init ()
56
57 void Save (Lattice<TD,TN> & Lattice, double Kn, double Ma) {
58   double theta0 = Lattice.theta0(0,0,0);
59   double uMax = Ma * std::sqrt(theta0);
60   std::ofstream o;
61   o.precision(PRECISION);
62   o.open("profils.res", std::ofstream::app);
63
64   o << "#Kn = " << Kn << "\n";
65   for (size_t iY = 0; iY != Lattice.size(1); ++iY) {
66     Lattice.compute_macros(Lx/2, iY, 0);
67     o << Lattice.rho(Lx/2, iY, 0) << " ";
68     o << Lattice.u(Lx/2, iY, 0, 0)/uMax << " ";
69     o << Lattice.u(Lx/2, iY, 0, 1)/uMax << " ";
70     o << Lattice.theta(Lx/2, iY, 0)/theta0 << " ";
71     o << Lattice.rho(Lx/2, iY, 0)*Lattice.theta(Lx/2, iY, 0)/theta0 << " ";
72     o << std::endl;
73   } //~ for iY
74   o << "\n\n";
75
76   o.close();
77 } //~ Save ()
78
79 int main(int argc, char *argv[]) {
80   // L : Lattice<Dimension, Total_number_of_beams> ...
81   Lattice<TD,TN> L;
82
83   // Creat geometry ...
```

```
84   Create_geometry();
85
86   // Load geometry ...
87   L.load_geometry("in_geom.txt");
88
89   // Load velocity space ...
90   L.load_velocity_space("vspace.txt");
91
92   // The Mach number (Ma = uMax/c0).
93   double Ma = .01;
94
95   // The Knudsen number.
96   double Kn = .01;
97
98   // Then Init ...
99   Init(L,Kn,Ma);
100
101  // Main loop over time ...
102  bool conv = false; double last = 1., present, epsilon = 1e-8;
103  size_t time = 0, save_step = 100;
104
105  while ( !conv ) {
106    L.propagate_and_collide();
107
108    // Convergence test.
109    present = 0.;
110    for (size_t iY = 0; iY != Ly; ++iY) {
111      L.compute_macros(L.size(0)/2.,iY,L.size(2));
112      present += std::abs(L.u(L.size(0)/2.,iY,L.size(2),0));
113    } //~ for iY
114
115    if ( time % save_step == 0 ) {
116      Save(L,Kn,Ma);
117      std::cout << "espilon = " << std::abs(present - last) << "\n";
118    } time++;
119
120    if ( std::abs(present - last) < epsilon || std::abs(present - last) > 10) {
121      Save(L,Kn,Ma);
122      conv = true;
123    } //~ if (conv)
124    last = present;
125    //~ Convergence test.
126
127  } //~ while ( !conv )
128
129  // The End.
130  std::cout << "\nThe Simulation is done.\n";
131  return 0;
132 }
```

C.2.4 Poiseuille plan – Pression – Gradient de densité

Ce second et dernier exemple, dont le graphe des appels aux fonctions est donné figure C.3, est très similaire au précédent, cependant, le fluide n'est plus mis en mouvement par une force extérieure, mais par un *gradient de densité*.

La différence majeure avec l'exemple précédent réside dans l'emploi des conditions aux bords présentées section 5.9.2 en entrée et en sortie du canal. Ces conditions aux bords sont mises en œuvre dans la fonction `PrBounds()`, ligne 66 du programme. La procédure consiste à imposer la valeur de la densité à l'entrée et à la sortie du canal grâce à la méthode `set_rho()` (lignes 75 et 89), puis à extrapoler les champs de vitesse et de température, par une extrapolation de Richardson au second ordre, depuis le fluide vers la paroi. Cette extrapolation impose que la variation de ces champs soit nulle au voisinage de la paroi. Finalement, la partie hors équilibre de la fonction de distribution est elle-même extrapolée du fluide vers la paroi, ceci grâce aux deux méthodes `set_fInPrX()` et `set_fOutPrX()` de la classe `Lattice` (cf. lignes 133 et 143 du ficher *lattice.hpp*, section C.2.6). La fonction `PrBounds()` est alors appliquée avant chaque itération de la méthode `propagate_and_collide()`, ligne 206.

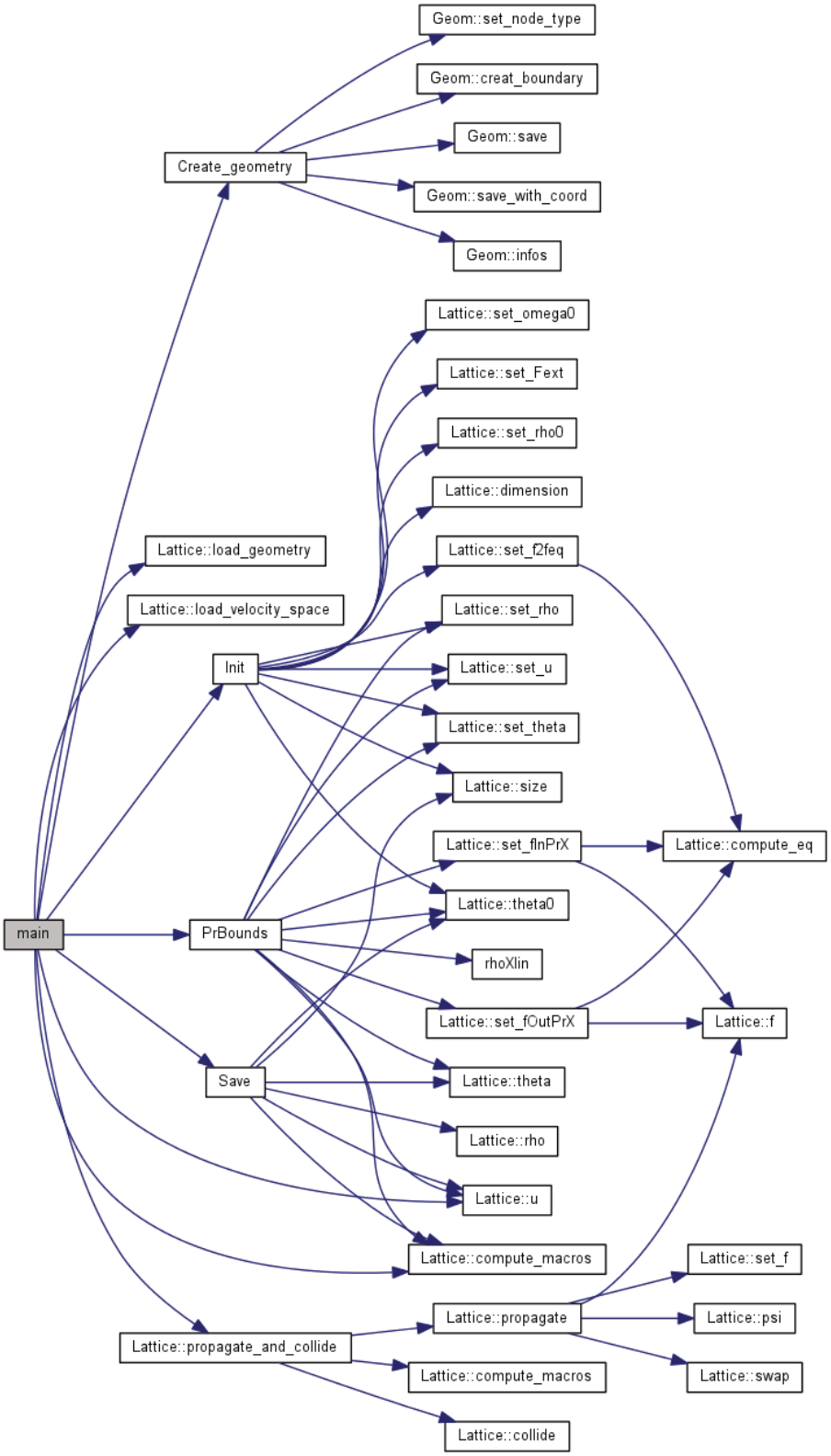


FIGURE C.3 – Graphe des appels aux fonctions pour l'exemple du Poiseuille plan soumis à un gradient de pression (fichier poiseuille_pr.cpp).

```
1  /**
2   * @file    poiseuille_pr.cpp
3   * @author Léonard de Izarra <leonardizarra@gmail.com>
4   * @date    Wed Jan 12 21:08:31 2011
5   */
6
7  #include "maBLB.hpp"
8
9  using namespace std;
10
11 const size_t TD=2, TN=16;
12 const size_t Lx=250, Ly=25;
13 double Ly0, Lx0;
14 double rhoIn = 3, rhoOut = 1.;
15
16 void Create_geometry () {
17   Geom G(Lx,Ly,1);
18
19   G.box();
20   Ly0 = (double)Ly;
21   Lx0 = (double)Lx;
22   G.creat_boundary();
23   G.save("in_geom.txt");
24   G.save_with_coord("coord_geom.txt");
25   G.infos();
26 } //~ Geometry ()
27
28 double rhoXlin (size_t iX) {
29   return (rhoOut - rhoIn) * iX / Lx + rhoIn;
30 }
31
32 void Init (Lattice<TD,TN> & Lattice, double Kn, double Ma) {
33
34   double c0 = std::sqrt(Lattice.theta0(0,0,0));
35   double uMax = Ma * c0;
36   double nu0 = Kn * Ly0 * c0 / sqrt(2.5);
37
38   for (size_t iX = 0; iX != Lattice.size(0); ++iX) {
39     for (size_t iY = 0; iY != Lattice.size(1); ++iY) {
40       Lattice.set_omega0(iX, iY, 0, 1./(nu0 / (c0*c0) + 1./2.));
41       Lattice.set_rho(iX, iY, 0, 1.);
42       Lattice.set_rho0(iX, iY, 0, 1.);
43
44       for (size_t axis = 0; axis != Lattice.dimension(); ++axis) {
45         Lattice.set_u(iX, iY, 0, axis, 0.);
46         Lattice.set_Fext(iX, iY, 0, axis, 0.);
47       } //~ for axis
48       Lattice.set_theta(iX, iY, 0, Lattice.theta0(0,0,0));
49     } //~ for iY
50   } //~ for iX
51
52   for (size_t iX = 0; iX != Lattice.size(0); ++iX) {
53     for (size_t iY = 1; iY != Lattice.size(1)-1; ++iY) {
54       Lattice.set_rho(iX, iY, 0, rhoXlin(iX));
55     } //~ for iY
```

```
56   } //~ for iX
57
58   for (size_t iX = 0; iX != Lattice.size(0); ++iX) {
59     for (size_t iY = 0; iY != Lattice.size(1); ++iY) {
60       Lattice.set_f2feq(iX, iY, 0);
61     } //~ for iY
62   } //~ for iX
63
64 } //~ Init ()
65
66 void PrBounds (Lattice<TD,TN> & Lattice, double Kn, double Ma) {
67
68   double c0 = std::sqrt(Lattice.theta0(0,0,0));
69   double uMax = Ma * c0;
70   double nu0 = Kn * Ly0 * c0 / sqrt(2.5);
71
72   for (size_t iY = 1; iY != Ly-1; ++iY) {
73
74     /// Inlet.
75     Lattice.set_rho(0, iY, 0, rhoXlin(0));
76     /// Macroscopic fields extrapolation.
77     Lattice.compute_macros(1,iY,0);
78     Lattice.compute_macros(2,iY,0);
79     Lattice.set_u(0, iY, 0, 0, 4.*Lattice.u(1, iY, 0, 0)
80                   - 3.*Lattice.u(2, iY, 0, 0));
81     Lattice.set_u(0, iY, 0, 1, 4.*Lattice.u(1, iY, 0, 1)
82                   - 3.*Lattice.u(2, iY, 0, 1));
83     Lattice.set_theta(0, iY, 0, 4.*Lattice.theta(1, iY, 0)
84                       - 3.*Lattice.theta(2, iY, 0));
85     /// Non-equilibrium distribution function extrapolation.
86     Lattice.set_fInPrX(0, iY, 0);
87
88     /// Outlet.
89     Lattice.set_rho(Lx-1, iY, 0, rhoXlin(Lx));
90     /// Macroscopic fields extrapolation.
91     Lattice.compute_macros(Lx-2,iY,0);
92     Lattice.compute_macros(Lx-3,iY,0);
93     Lattice.set_u(Lx-1, iY, 0, 0,4.*Lattice.u(Lx-2, iY, 0, 0)
94                   - 3.*Lattice.u(Lx-3, iY, 0, 0));
95     Lattice.set_u(Lx-1, iY, 0, 1, 4.*Lattice.u(Lx-2, iY, 0, 1)
96                   - 3.*Lattice.u(Lx-3, iY, 0, 1));
97     Lattice.set_theta(Lx-1, iY, 0, 4.*Lattice.theta(Lx-2, iY, 0)
98                       - 3.*Lattice.theta(Lx-3, iY, 0));
99     /// Non-equilibrium distribution function extrapolation.
100    Lattice.set_fOutPrX(Lx-1, iY, 0);
101
102  } //~ for iY
103
104 } //~ PrBounds ()
105
106 void Save (Lattice<TD,TN> & Lattice, double Kn, double Ma){
107   double theta0 = Lattice.theta0(0,0,0);
108   double uMax = Ma * std::sqrt(theta0);
109   double c0 = std::sqrt(Lattice.theta0(0,0,0));
110   double nu0 = Kn * Ly0 * c0 / sqrt(2.5);
```

```
111  std::ofstream o;
112  o.precision(PRECISION);
113
114  o.open("sim.res", std::ofstream::app);
115  for (size_t iX = 0; iX != Lattice.size(0); ++iX) {
116    for (size_t iY = 0; iY != Lattice.size(1); ++iY) {
117      Lattice.compute_macros(iX, iY, 0);
118      o << std::setprecision(12) << Lattice.rho(iX, iY, 0) << " "
119        << Lattice.rho(iX, iY, 0) - rhoXlin(iX) << " ";
120      double u_sq = 0;
121      for (size_t axis = 0; axis < Lattice.dimension(); ++axis) {
122        o << std::setprecision(12) << Lattice.u(iX, iY, 0, axis)/uMax << " ";
123        u_sq += std::pow(Lattice.u(iX, iY, 0, axis),2);
124      } //~ for axis
125      o << std::setprecision(12) << std::sqrt(u_sq)/uMax << " ";
126      o << std::setprecision(12) << Lattice.theta(iX, iY, 0)
127        / Lattice.theta0(0,0,0) << " ";
128      o << std::setprecision(12) <<
129        Lattice.rho(iX, iY, 0) * Lattice.theta(iX, iY, 0) / Lattice.theta0(0,0,0) << "\n";
130    } //~ for iY
131    o <<  "\n";
132  }//~ for iX
133  o <<  "\n\n";
134  o.close();
135
136  o.open("u_vec.res", std::ofstream::app);
137  for (size_t iX = 0; iX != Lattice.size(0); ++iX) {
138    for (size_t iY = 0; iY != Lattice.size(1); ++iY) {
139      if ( (iX % 1 == 0) && (iY % 1 == 0)) {
140        o << iX << " " << iY << " ";
141        for (size_t axis = 0; axis < Lattice.dimension(); ++axis)
142          o << std::setprecision(12) << Lattice.u(iX, iY, 0, axis)/uMax << " ";
143        o << std::endl;
144      } //~ if
145    } //~ for iY
146  }//~ for iX
147  o << "\n\n";
148  o.close();
149
150  o.open("profils.res", std::ofstream::app);
151  o << "#Kn = " << Kn << "\n";
152  for (size_t iY = 0; iY != Lattice.size(1); ++iY) {
153    o << Lattice.rho(Lx-1, iY, 0) << " ";
154    o << Lattice.u(Lx-1, iY, 0, 0)/uMax << " ";
155    o << Lattice.u(Lx-1, iY, 0, 1)/uMax << " ";
156    o << Lattice.theta(Lx-1, iY, 0)/theta0 << " ";
157    o << Lattice.rho(Lx-1, iY, 0)*Lattice.theta(Lx/2, iY, 0)/theta0 << " ";
158    o << std::endl;
159  } //~ for iY
160  o << "\n\n";
161  o.close();
162
163  o.open("profilsX.res", std::ofstream::app);
164  o << "#Kn = " << Kn << "\n";
165  for (size_t iX = 0; iX != Lx; ++iX) {
```

```
166     o << Lattice.rho(iX, Ly/2, 0) << " ";
167     o << Lattice.rho(iX, Ly/2, 0) - rhoXlin(iX) << " ";
168     o << Lattice.u(iX, Ly/2, 0, 0)/uMax << " ";
169     o << Lattice.u(iX, Ly/2 , 0, 1)/uMax << " ";
170     o << Lattice.theta(iX, Ly/2, 0)/theta0 << " ";
171     o << Lattice.rho(iX, Ly/2, 0)*Lattice.theta(iX, Ly/2, 0)/theta0 << " ";
172     o << (Lattice.rho(iX, Ly/2, 0) - rhoXlin(iX)) / theta0 << " ";
173     o << std::endl;
174   } //~ for iY
175   o << "\n\n";
176   o.close();
177 } //~ Save ()
178
179 int main(int argc, char *argv[]) {
180   // L : Lattice<Dimension, Total_number_of_beams> ...
181   Lattice<TD,TN> L;
182
183   // Creat geometry ...
184   Create_geometry();
185
186   // Load geometry ...
187   L.load_geometry("in_geom.txt");
188
189   // Load velocity space ...
190   L.load_velocity_space("vspace.txt");
191
192   // The Mach number (Ma = uMax/c0).
193   double Ma = .01;
194
195   // The Knudsen number.
196   double Kn = .01;
197
198   // Then Init ...
199   Init(L,Kn,Ma);
200
201   // Main loop over time ...
202   bool conv = false; double last = 1., present, epsilon = 5e-6;
203   size_t time = 0, save_step = 10;
204
205   while ( !conv ) {
206     PrBounds(L,Kn,Ma);
207     L.propagate_and_collide();
208
209     // Convergence test.
210     present = 0.;
211     for (size_t iY = 0; iY != Ly; ++iY) {
212       L.compute_macros(Lx/2.,iY,0);
213       present += std::abs(L.u(Lx/2,iY,0,0));
214     } //~ for iY
215
216     if ( time % save_step == 0 ) {
217       //Save(L,Kn,Ma);
218       std::cout << "espilon = " << std::abs(present - last) << "\n";
219     } time++;
220
```

```
221     if ( std::abs(present - last) < epsilon ) {
222       Save(L,Kn,Ma);
223       conv = true;
224     } //~ if (conv)
225     last = present;
226     //~ Convergence test.
227
228   } //~ while (!conv )
229
230   // The End.
231   std::cout << "\nThe Simulation is done.\n";
232   return 0;
233 }
```

C.2.5 Géométrie

Cette section porte sur les outils de création des géométries données en entrée au noyau de calcul de maBLB. Ces outils sont une classe nommée `Geom` et un script shell [2]. Ce script est optionnel et constitue une solution rapide à la génération de géométries à partir d'images. Une autre façon de procéder est utilisée dans les deux exemples donnés précédemment (cf. fonctions `Create_geometry()` des programmes d'exemples). Elle consiste à instancier la classe `Geom`, donnée ci-après, grâce à son constructeur `Geom(Lx,Ly,Lz)` (cf. ligne 45), où `(Lx,Ly,Lz)` sont les tailles du domaine d'intégration selon les différentes directions de l'espace. Dans le cas particulier où `Lz=1`, le problème sera 2D et si `Ly=Lz=1`, il sera 1D. Le paramètre *Template* `T_N` de toutes les classes devra être en cohérence avec la définition de ces tailles. Cette remarque est importante et s'explique par le fait qu'il a été choisi, pour rendre le code quasi indépendant de la dimension du problème, d'utiliser partout des tableaux à trois dimensions et d'imposer, pour les problèmes en dimensions inférieures, la taille des tableaux à 1 selon les directions "inexistantes". En outre, beaucoup de méthodes et opérateurs sont dépendants de la dimension, comme le calcul de la fonction d'équilibre ; cette information est passée au compilateur via le paramètre *Template* `T_N`. Les patrons de classe (`Template`) ne sont donc pas utilisés ici dans le but de rendre le code générique (au sens indépendant des types de données), mais plutôt comme un *préprocesseur*, un *métalangage*.

À l'instanciation, la classe `Geom(Lx,Ly,Lz)` ne contient que des nœuds de type fluide (`Fluid`). Construire une géométrie consiste à poser des nœuds de type solide (`Solid`) à l'intérieur du domaine d'intégration, opération réalisée grâce à la méthode `set_node_type(x,y,z,Solid)` de la classe `Geom` (ligne 55). Une fois la géométrie décrite par placement d'un ensemble de nœuds `Solid`, les bords doivent être détectés et leur normale construite, ce que permet la méthode `creat_boundary()` de la classe `Geom` (ligne 59).

2. Ce script n'a été testé que sur un système GNU/Linux. Son emploi nécessite qu'un shell Bash – ou compatible –, Imagemagick et Awk soient installés sur le système.

Class Geom

```
/**
 * @file    geometry.hpp
 * @author Léonard de Izarra <leonardizarra@gmail.com>
 * @date    Wed Jun  2 20:57:13 2010
 *
 */

#ifndef _GEOMETRY_H_
#define _GEOMETRY_H_

#include "globals.hpp"

/**
 * @class Geom
 *
 */
class Geom {
public:
  /// Default Ctor. -- Creat an empty geometry.
  Geom () { }

  /// Ctor. -- Creat geometry from picture created with the png2txt script.
  Geom (const char* inputFile) {
    std::stringstream fileName; fileName << inputFile;
    std::fstream in; in.open(fileName.str().c_str(), std::fstream::in);

    in >> m_lx  >> m_ly  >> m_lz;

    m_geom.resize(boost::extents[m_lx][m_ly][m_lz]);
    m_normal.resize(boost::extents[m_lx][m_ly][m_lz][3]);

    for (int k=0; k != m_lz; ++k)
      for (int j=0; j != m_ly; ++j)
        for (int i=0; i != m_lx; ++i)
          {
            int value;
            in >> value;
            m_geom[i][j][k] = static_cast<node_type>(value);
          }

    in.close();
  }

  /// Ctor. -- Creat an empty geometry of size (lx,ly,lz).
  Geom (size_t lx, size_t ly, size_t lz)
    : m_lx(lx), m_ly(ly), m_lz(lz)
  {
    m_geom.resize(boost::extents[m_lx][m_ly][m_lz]);
    std::fill(m_geom.data(), m_geom.data() + m_geom.num_elements(), Fluid);

    m_normal.resize(boost::extents[m_lx][m_ly][m_lz][3]);
    std::fill(m_normal.data(), m_normal.data() + m_normal.num_elements(), 0.);
  }
```

```
54
55  void set_node_type (size_t iX, size_t iY, size_t iZ, node_type NodeType) {
56    m_geom[iX][iY][iZ] = NodeType;
57  }
58
59  void creat_boundary () {
60    m_geom_tmp.resize(boost::extents[m_lx][m_ly][m_lz]);
61    m_geom_tmp = m_geom;
62
63    for(;;) {
64      /// Make the geometry clean ...
65      for (int i=0; i != m_lx; ++i)
66        for (int j=0; j != m_ly; ++j)
67          for (int k=0; k != m_lz; ++k)
68            {
69              int inext  = (i + 1 + 2*m_lx)%m_lx;
70              int ipre   = (i - 1 + 2*m_lx)%m_lx;
71              int jnext  = (j + 1 + 2*m_ly)%m_ly;
72              int jpre   = (j - 1 + 2*m_ly)%m_ly;
73              int knext  = (k + 1 + 2*m_lz)%m_lz;
74              int kpre   = (k - 1 + 2*m_lz)%m_lz;
75
76              if (m_geom[i][j][k] == Solid &&
77                  ((m_geom[ipre][j][k] == Fluid && m_geom[inext][j][k] == Fluid) ||
78                   (m_geom[i][jpre][k] == Fluid && m_geom[i][jnext][k] == Fluid) ||
79                   (m_geom[i][j][kpre] == Fluid && m_geom[i][j][knext] == Fluid) ))
80                {
81                  m_geom_tmp[i][j][k] = Fluid;
82                }
83            }
84      if (m_geom == m_geom_tmp)
85        break;
86
87      m_geom = m_geom_tmp;
88    }
89
90    /// Now, creat boundaries and find normals ...
91    for (int i = 0; i != m_lx; ++i)
92      for (int j = 0; j != m_ly; ++j)
93        for (int k = 0; k != m_lz; ++k)
94          {
95            int inext  = (i + 1 + 2*m_lx)%m_lx;
96            int ipre   = (i - 1 + 2*m_lx)%m_lx;
97
98            if (m_geom[i][j][k] == Solid && m_geom[ipre][j][k] == Fluid)
99              {
100               m_geom[i][j][k] = Bound;
101               m_normal[i][j][k][0] = -1;
102             }
103           else if (m_geom[i][j][k] == Solid && m_geom[inext][j][k] == Fluid)
104             {
105               m_geom[i][j][k] = Bound;
106               m_normal[i][j][k][0] = 1;
107             }
108
```

```
109             int jnext  = (j + 1 + 2*m_ly)%m_ly;
110             int jpre   = (j - 1 + 2*m_ly)%m_ly;
111
112             if ((m_geom[i][j][k] == Solid || m_geom[i][j][k] == Bound)
113                 && m_geom[i][jpre][k] == Fluid)
114               {
115                 m_geom[i][j][k] = Bound;
116                 m_normal[i][j][k][1] = -1;
117               }
118             else if ((m_geom[i][j][k] == Solid || m_geom[i][j][k] == Bound)
119                      && m_geom[i][jnext][k] == Fluid)
120               {
121                 m_geom[i][j][k] = Bound;
122                 m_normal[i][j][k][1] = 1;
123               }
124
125             if (m_lz != 1)
126               {
127                 int knext  = (k + 1 + 2*m_lz)%m_lz;
128                 int kpre   = (k - 1 + 2*m_lz)%m_lz;
129
130                 if ((m_geom[i][j][k] == Solid || m_geom[i][j][k] == Bound)
131                     && m_geom[i][j][kpre] == Fluid)
132                   {
133                     m_geom[i][j][k] = Bound;
134                     m_normal[i][j][k][2] = -1;
135                   }
136                 else if ((m_geom[i][j][k] == Solid || m_geom[i][j][k] == Bound)
137                          && m_geom[i][j][knext] == Fluid)
138                   {
139                     m_geom[i][j][k] = Bound;
140                     m_normal[i][j][k][2] = 1;
141                   }
142               }
143           }
144
145       /// Finally, we normalise normals ...
146       for (int i = 0; i != m_lx; ++i)
147         for (int j = 0; j != m_ly; ++j)
148           for (int k = 0; k != m_lz; ++k)
149             {
150               double norm = 0.;
151               if (m_lz == 1)
152                 {
153                   for (int l = 0; l != 2; ++l)
154                     norm += std::pow(m_normal[i][j][0][l],2);
155                   for (int l = 0; l != 2; ++l)
156                     m_normal[i][j][0][l] /= std::sqrt(norm);
157                 }
158               else
159                 {
160                   for (int l = 0; l != 3; ++l)
161                     norm += std::pow(m_normal[i][j][k][l],2);
162                   for (int l = 0; l != 3; ++l)
163                     m_normal[i][j][k][l] /= std::sqrt(norm);
```

```
164             }
165         }
166 }
167
168 /// Load picture from file created with the png2txt script.
169 void load (const char* inputFile) {
170   std::stringstream fileName; fileName << inputFile;
171   std::fstream in; in.open(fileName.str().c_str(), std::fstream::in);
172
173   in >> m_lx  >> m_ly  >> m_lz;
174
175   m_geom.resize(boost::extents[m_lx][m_ly][m_lz]);
176   m_normal.resize(boost::extents[m_lx][m_ly][m_lz][3]);
177
178   for (int k=0; k != m_lz; ++k)
179     for (int j=0; j != m_ly; ++j)
180       for (int i=0; i != m_lx; ++i)
181         {
182           int value; in >> value;
183           m_geom[i][j][k] = static_cast<node_type>(value);
184         }
185
186   in.close();
187 }
188
189 /// Save created geometry in file.
190 /// This file is the input for simulation.
191 void save (const char* outputFile) {
192   std::stringstream fileName; fileName << outputFile;
193   std::fstream out; out.open(fileName.str().c_str(), std::fstream::out);
194   out.precision(std::numeric_limits<double>::digits);
195
196   out << m_lx << " " << m_ly << " " << m_lz << "\n";
197
198   for (int i=0; i != m_lx; ++i)
199     for (int j=0; j != m_ly; ++j)
200       for (int k=0; k != m_lz; ++k)
201         if (m_geom[i][j][k] != Bound)
202           out << m_geom[i][j][k]      << "\n";
203         else
204           if (m_lz == 1)
205             out << m_geom[i][j][k]      << " "
206                 << m_normal[i][j][k][0] << " "
207                 << m_normal[i][j][k][1] << "\n";
208           else
209             out << m_geom[i][j][k]      << " "
210                 << m_normal[i][j][k][0] << " "
211                 << m_normal[i][j][k][1] << " "
212                 << m_normal[i][j][k][2] << "\n";
213
214   out.close();
215 }
216
217 /// Save created geometry in file.
218 /// This file is formated for the gnuplot software and is only do for geometry checking.
```

```
219  void save_with_coord (const char* outputFile) {
220    std::stringstream fileName; fileName << outputFile;
221    std::fstream out; out.open(fileName.str().c_str(), std::fstream::out);
222    out.precision(std::numeric_limits<double>::digits);
223
224    // index 0 : Fluid
225    for (int i=0; i != m_lx; ++i)
226      for (int j=0; j != m_ly; ++j)
227        for (int k=0; k != m_lz; ++k)
228          if (m_geom[i][j][k] == Fluid)
229            if (m_lz == 1)
230              out << i  << " "
231                  << j  << "\n";
232            else
233              out << i << " "
234                  << j << " "
235                  << k << "\n";
236
237    out << "\n\n";
238
239    // index 1
240    for (int i=0; i != m_lx; ++i)
241      for (int j=0; j != m_ly; ++j)
242        for (int k=0; k != m_lz; ++k)
243          if ((m_geom[i][j][k] == Solid))
244            if (m_lz == 1)
245              out << i  << " "
246                  << j  << "\n";
247            else
248              out << i << " "
249                  << j << " "
250                  << k << "\n";
251
252    out << "\n\n";
253
254    // index 2 : Bound
255    for (int i=0; i != m_lx; ++i)
256      for (int j=0; j != m_ly; ++j)
257        for (int k=0; k != m_lz; ++k)
258          if (m_geom[i][j][k] == Bound)
259            if (m_lz == 1)
260              out << i  << " "
261                  << j  << "\n";
262            else
263              out << i << " "
264                  << j << " "
265                  << k << "\n";
266    out << "\n\n";
267
268    // index 3 : Normals
269    for (int i=0; i != m_lx; ++i)
270      for (int j=0; j != m_ly; ++j)
271        for (int k=0; k != m_lz; ++k)
272          if (m_geom[i][j][k] == Bound)
273            if (m_lz == 1)
```

```
274             out << i                    << " "
275                 << j                    << " "
276                 << m_normal[i][j][k][0] << " "
277                 << m_normal[i][j][k][1] << "\n";
278           else
279             out << i << " "
280                 << j << " "
281                 << k << " "
282                 << m_normal[i][j][k][0] << " "
283                 << m_normal[i][j][k][1] << " "
284                 << m_normal[i][j][k][2] << "\n";
285
286   out << "\n\n";
287
288   out.close();
289 }
290
291 /// Infos is:
292 /// - Size of the geometry \f$ ( lx, ly, lz ) \f$.
293 /// - The porosity \f$ \epsilon \f$ of the geometry,
294 ///    i.e. the ratio of the non-solid volume \f$ V_{fluid} \f$ (excluding boundary)
295 ///    and the total volume (namely \f$ V_{total} = lx\,ly\,lz \f$). \n
296 ///    Mathematicaly: \f$ \displaystyle \epsilon = \frac{V_{fluid}}{V_{total}}\f$
297 /// \brief Provide informations about the geometry on the standard output.
298 void infos () {
299   std::cout << "Sizes = "
300             << m_lx << ", " << m_ly << ", " << m_lz
301             << "\n";
302
303   int vol_fluid = 0;
304   for (int i=0; i != m_lx; ++i)
305     for (int j=0; j != m_ly; ++j)
306       for (int k=0; k != m_lz; ++k)
307         if (m_geom[i][j][k] == Fluid) ++vol_fluid;
308
309   std::cout << "Porosity = "
310             << (double)vol_fluid / (double)(m_lx * m_ly * m_lz)
311             << "\n";
312 }
313
314 void box () { // 2D box
315
316   for (size_t i = 0 ; i != m_lx; ++i)
317     {
318       m_geom[i][0][0] = Solid;
319       m_geom[i][m_ly-1][0] = Solid;
320     }
321
322   for (size_t i = 0 ; i != m_ly; ++i)
323     {
324       m_geom[0][i][0] = Solid;
325       m_geom[m_lx-1][i][0] = Solid;
326     }
327 }
328
```

```
329 private:
330   boost::multi_array<node_type, 3> m_geom, m_geom_tmp;
331   /// Boundary normals'.
332   boost::multi_array<double, 4> m_normal;
333   ///
334   size_t m_lx, m_ly, m_lz;
335 };
336
337 #endif /* __GEOMETRY_H__ */
```

Convertisseur png vers ascii

Le script suivant permet de transformer des images au format *png* codées en *RVB 24 bits*, le blanc [3] correspondant à du fluide et le noir [4] à du solide. Il permet de traiter des images 2D et 3D.

```
1  #!/bin/bash
2  #
3  # File : png2txt.sh
4  # Deps : awk, imagemagick.
5
6  if [ $# -eq 0 ] then
7      echo "Usage: $0  (input.png || directory/input*.png)"
8      exit 1
9  fi
10
11 echo  $(identify $1 | awk '{print $3}' | uniq | tr "x" " ") "$#"
12
13 for i in $*
14 do
15     convert -flip $i txt:- | awk -F "[,: )(]" '$1 != "#" {print and($7,255)/255}'
16 done
17
18 exit 0
```

Exemple d'utilisation en 2D, sous UNIX :

```
$ png2txt.sh test.png >| test.txt
$ geoforge test.txt in_test.txt coord_test.txt
```

Où *geoforge* est le programme dont le code est donné ci-après :

```
1  /**
2   * @file    geoforge.cpp
3   * @author Leonard de Izarra <leonardizarra@gmail.com>
4   * @date    Sun Aug 21 20:10:51 2011
5   *
6   */
7  #include "geometry.hpp"
8  #include <iostream>
9
10 int main(int argc, char* argv[])
```

3. (255,255,255) en RVB.
4. (0, 0, 0) en RVB.

```
11 {
12   switch ( argc )
13     {
14     case 3:
15       {
16         Geom G;
17         G.load(argv[1]);
18         G.creat_boundary();
19         G.save(argv[2]);
20         G.infos();
21         std::cout << "Done.\n";
22         return 0;
23       }
24     case 4:
25       {
26         Geom G;
27         G.load(argv[1]);
28         G.creat_boundary();
29         G.save(argv[2]);
30         G.save_with_coord(argv[3]);
31         G.infos();
32         std::cout << "Done.\n";
33         return 0;
34       }
35     default:
36       {
37         std::cout << "Usage: geoforge inputFile outputFile [CoordOutputFile]\n";
38         exit(1);
39       }
40     }
41 }
```

C.2.6 Le réseau – Class Lattice

La classe Lattice est l'objet principal de la librairie. Elle est constituée de deux *buffers* contenant chacun un exemplaire de la simulation. Ces deux buffers sont nécessaires pour réaliser l'étape de transport de la fonction de distribution, pratiquée par la technique du double buffering (cf. lignes 274-405).

Ces buffers sont définis ligne 412 par :

```
boost::array<boost::multi_array<Node<T_D, T_N>*,3>,2> m_buffer;
```

Ce sont donc deux tableaux 3D `boost::multi_array` de pointeurs sur des `Node<T_D, T_N>` (cf. section C.2.7) contenus dans un tableau de deux éléments `boost::array<T,2>`.

Une particularité de la technique utilisée ici vient de l'abstraction qui est faite initialement sur les éléments `Node<T_D, T_N>*` que contiennent les deux buffers. En effet, il n'est pas possible d'instancier d'objet de type `Node<T_D, T_N>`, mais uniquement des pointeurs sur ces objets, on parle alors de classe abstraite. Cependant,

il est possible de les faire pointer vers les “concrétisations” de la classe abstraite `Node<T_D, T_N>` que sont les classes `Fluid_Node<T_D, T_N>`, `Bound_Node<T_D, T_N>` et `Solid_Node<T_D, T_N>` grâce à l'instruction `new`, par exemple :

```
m_buffer[0][x][y][z] = new Fluid_Node<T_D,T_N>;
```

Cette technique est utilisée pour construire la simulation à partir du fichier qui en décrit la géométrie, comme cela est fait dans la méthode `load_geometry()` de la classe Lattice (cf. ligne 73). Ainsi, une fois la géométrie construite, chaque fonction membre de `Node<T_D, T_N>` appelée sera celle de la classe fille de `Node<T_D, T_N>` qui aura été instanciée dans `m_buffer[nbuf][x][y][z]`. En particulier, la fonction `collide()` sera celle du fluide, du bord ou du solide (cf. section C.2.7).

```
1  /// \file    lattice.hpp
2  /// \author Léonard de Izarra <leonardizarra@gmail.com>
3  /// \date    Tue Jun 22 18:47:01 2010
4
5  #ifndef _LATTICE_H_
6  #define _LATTICE_H_
7
8  #include "globals.hpp"
9  #include "node.hpp"
10 #include "vspace.hpp"
11
12 /// \class Lattice
13 /// Structurally, it's an array of nodes.
14 /// \param T_D the dimension of the simulation.
15 /// \param T_N Total number of beams.
16 template<size_t T_D, size_t T_N>
17 class Lattice {
18 public:
19   /// Ctor.
20   Lattice () : m_cbuf (0), m_obuf (1) { }
21
22   /// Dtor.
23   ~Lattice () {
24     for (size_t iX = 0; iX != m_size[0]; ++iX) {
25       for (size_t iY = 0; iY != m_size[1]; ++iY) {
26         for (size_t iZ = 0; iZ != m_size[2]; ++iZ) {
27           delete m_buffer[0][iX][iY][iZ];
28           delete m_buffer[1][iX][iY][iZ];
29         } //~ for iZ
30       } //~ for iY
31     } //~ for iX
32   } //~ ~Lattice()
33
34   /// Dimensionality of the simulation.
35   size_t dimension () {
36     return T_D;
37   }
38
39   /// Total number of beams.
```

```
40  size_t nbeams () {
41    return T_N;
42  }
43
44  /// Size accessor.
45  /// \pre 0 < \a axis < T_D
46  size_t size (size_t axis) {
47    return m_size[axis];
48  }
49
50  /// Total number of nodes in the simulation.
51  size_t population () {
52    return m_buffer[m_cbuf].num_elements();
53  }
54
55  /// Node type accessor.
56  node_type type  (size_t iX, size_t iY, size_t iZ) {
57    return  m_buffer[m_cbuf][iX][iY][iZ]->type();
58  } //~ type ()
59
60  /// Load the velocity space discretisation created with the "vsforge" tool.
61  void load_velocity_space (const char* inputFile) {
62    for (size_t iX = 0; iX != m_size[0]; ++iX) {
63      for (size_t iY = 0; iY != m_size[1]; ++iY) {
64        for (size_t iZ = 0; iZ != m_size[2]; ++iZ) {
65          m_buffer[m_cbuf][iX][iY][iZ]->vspace.load(inputFile);
66          m_buffer[m_obuf][iX][iY][iZ]->vspace.load(inputFile);
67        } //~ for iZ
68      } //~ for iY
69    } //~ for iX
70  } //~ load_velocity_space()
71
72  /// Load the geometry created with the the Geom class, cf. geometry.hpp.
73  void load_geometry (const char* inputFile) {
74    std::stringstream fileName; fileName << inputFile;
75    std::fstream in; in.open(fileName.str().c_str(), std::fstream::in);
76
77    // Read sizes of the simulation.
78    in >> m_size[0]  >> m_size[1]  >> m_size[2];
79
80    m_buffer[0].resize(boost::extents[m_size[0]][m_size[1]][m_size[2]]);
81    m_buffer[1].resize(boost::extents[m_size[0]][m_size[1]][m_size[2]]);
82
83    /// Nodes differentiation -- cf. node_type in globals.hpp.
84    for (size_t iX = 0; iX != m_size[0]; ++iX) {
85      for (size_t iY = 0; iY != m_size[1]; ++iY) {
86        for (size_t iZ = 0; iZ != m_size[2]; ++iZ) {
87          size_t value; in >> value;
88          switch (value) {
89          case Solid : {
90            m_buffer[m_cbuf][iX][iY][iZ] = new Solid_Node<T_D,T_N>;
91            m_buffer[m_obuf][iX][iY][iZ] = new Solid_Node<T_D,T_N>;
92            break;
93          } //~ case Solid
94          case Fluid : {
```

```
95            m_buffer[m_cbuf][iX][iY][iZ] = new Fluid_Node<T_D,T_N>;
96            m_buffer[m_obuf][iX][iY][iZ] = new Fluid_Node<T_D,T_N>;
97            break;
98          } //~ case Fluid
99          case Bound : {
100           m_buffer[m_cbuf][iX][iY][iZ] = new Bound_Node<T_D,T_N>;
101           m_buffer[m_obuf][iX][iY][iZ] = new Bound_Node<T_D,T_N>;
102           /// If the node is a boundary node, then we read
103           /// his normal vector from the file.
104           for (size_t axis = 0; axis < T_D; ++axis) {
105             double normal; in >> normal;
106             m_buffer[0][iX][iY][iZ]->set_normal(axis, normal);
107             m_buffer[1][iX][iY][iZ]->set_normal(axis, normal);
108           }
109           break;
110         } //~ case Bound
111         default : {
112           std::cout << "Erreur: Node type " << value << " don't exist\n";
113           exit(8);
114         } //~ default
115         } //~ switch
116       } //~ for iZ
117     } //~ for iY
118   } //~ for iX
119
120   in.close();
121
122 } //~ load_geometry ()
123
124 /// Sets f to equilibrium on each nodes of the two buffers.
125 void set_f2feq (size_t iX, size_t iY, size_t iZ) {
126   for (size_t index = 0; index < T_N; ++index) {
127     double a_feq = compute_eq(iX, iY, iZ, index);
128     m_buffer[m_cbuf][iX][iY][iZ]->set_f(index, a_feq);
129     m_buffer[m_obuf][iX][iY][iZ]->set_f(index, a_feq);
130   } //~ for index
131 } //~ set_f2feq (x,y,z)
132
133 void set_fInPrX (size_t iX, size_t iY, size_t iZ) {
134   for (size_t index = 0; index < T_N; ++index) {
135     double a_feq = compute_eq(iX, iY, iZ, index);
136     double a_fneq1 = 4.*(f(iX+1, iY, iZ, index) - compute_eq(iX+1, iY, iZ, index)) -
137       3.*(f(iX+2, iY, iZ, index) - compute_eq(iX+2, iY, iZ, index));
138     m_buffer[m_cbuf][iX][iY][iZ]->set_f(index, a_feq + a_fneq1);
139     m_buffer[m_obuf][iX][iY][iZ]->set_f(index, a_feq + a_fneq1);
140   } //~ for index
141 } //~ set_fInPrX (x,y,z)
142
143 void set_fOutPrX (size_t iX, size_t iY, size_t iZ) {
144   for (size_t index = 0; index < T_N; ++index) {
145     double a_feq = compute_eq(iX, iY, iZ, index);
146     double a_fneq1 = 4.*(f(iX-1, iY, iZ, index) - compute_eq(iX-1, iY, iZ, index)) -
147       3.*(f(iX-2, iY, iZ, index) - compute_eq(iX-2, iY, iZ, index));
148     m_buffer[m_cbuf][iX][iY][iZ]->set_f(index, a_feq + a_fneq1);
149     m_buffer[m_obuf][iX][iY][iZ]->set_f(index, a_feq + a_fneq1);
```

```
150     } //~ for index
151   } //~ set_fOutPrX (x,y,z)
152
153   /// Distribution function accessor.
154   double f (size_t iX, size_t iY, size_t iZ, size_t index) {
155     return m_buffer[m_cbuf][iX][iY][iZ]->f(index);
156   } //~ f ()
157
158   /// Distribution function setter.
159   void set_f (size_t iX, size_t iY, size_t iZ, size_t index, double value) {
160     m_buffer[m_cbuf][iX][iY][iZ]->set_f(index, value);
161   } //~ set_f ()
162
163   /// Reference density accessor.
164   double rho0 (size_t iX, size_t iY, size_t iZ) {
165     return m_buffer[m_cbuf][iX][iY][iZ]->rho0();
166   } //~ rho0 ()
167
168   /// Reference density setter.
169   void set_rho0 (size_t iX, size_t iY, size_t iZ, double value) {
170     m_buffer[m_cbuf][iX][iY][iZ]->set_rho0(value);
171     m_buffer[m_obuf][iX][iY][iZ]->set_rho0(value);
172   } //~  set_rho0 ()
173
174   /// Density accessor.
175   double rho (size_t iX, size_t iY, size_t iZ) {
176     return m_buffer[m_cbuf][iX][iY][iZ]->rho();
177   } //~ rho ()
178
179   /// Density setter.
180   void set_rho (size_t iX, size_t iY, size_t iZ, double value) {
181     m_buffer[m_cbuf][iX][iY][iZ]->set_rho(value);
182     m_buffer[m_obuf][iX][iY][iZ]->set_rho(value);
183   } //~ set_rho ()
184
185   /// Macroscopic velocity accessor.
186   double u (size_t iX, size_t iY, size_t iZ, size_t axis) {
187     return m_buffer[m_cbuf][iX][iY][iZ]->u(axis);
188   } //~ u ()
189
190   /// Macroscopic velocity setter.
191   void set_u (size_t iX, size_t iY, size_t iZ, size_t axis, double value) {
192     m_buffer[m_cbuf][iX][iY][iZ]->set_u(axis, value);
193     m_buffer[m_obuf][iX][iY][iZ]->set_u(axis, value);
194   } //~ set_u ()
195
196   /// Temperature accessor.
197   double theta (size_t iX, size_t iY, size_t iZ) {
198     return m_buffer[m_cbuf][iX][iY][iZ]->theta();
199   } //~ theta ()
200
201   /// Temperature setter.
202   void set_theta (size_t iX, size_t iY, size_t iZ, double value) {
203     m_buffer[m_cbuf][iX][iY][iZ]->set_theta(value);
204     m_buffer[m_obuf][iX][iY][iZ]->set_theta(value);
```

```
205   } //~ theta ()
206
207   /// Reference temperature accessor.
208   double theta0 (size_t iX, size_t iY, size_t iZ) {
209     return m_buffer[m_cbuf][iX][iY][iZ]->theta0();
210   } //~ theta0 ()
211
212   /// External force accessor.
213   double Fext (size_t iX, size_t iY, size_t iZ, size_t axis) {
214     return m_buffer[m_cbuf][iX][iY][iZ]->Fext(axis);
215   } //~ Fext ()
216
217   /// External force setter.
218   void set_Fext (size_t iX, size_t iY, size_t iZ, size_t axis, double value) {
219     m_buffer[m_cbuf][iX][iY][iZ]->set_Fext(axis, value);
220     m_buffer[m_obuf][iX][iY][iZ]->set_Fext(axis, value);
221   } //~ set_Fext ()
222
223   /// Collision frequency accessor.
224   double omega (size_t iX, size_t iY, size_t iZ) {
225     return m_buffer[m_cbuf][iX][iY][iZ]->omega();
226   } //~ omega ()
227
228   /// Collision frequency setter.
229   void set_omega0 (size_t iX, size_t iY, size_t iZ, double value) {
230     m_buffer[m_cbuf][iX][iY][iZ]->set_omega0(value);
231     m_buffer[m_obuf][iX][iY][iZ]->set_omega0(value);
232   } //~ set_omega ()
233
234   /// Compute macroscopic fields from distribution function
235   /// on every nodes of the current buffer.
236   void compute_macros (size_t iX, size_t iY, size_t iZ) {
237     m_buffer[m_cbuf][iX][iY][iZ]->compute_macros();
238   } //~ compute_macros ()
239
240   /// Compute equilibrium distribution function from macroscopic fields (Momenta).
241   double compute_eq (size_t iX, size_t iY, size_t iZ, size_t index) {
242     return m_buffer[m_cbuf][iX][iY][iZ]->compute_eq(index);
243   } //~ compute_eq ()
244
245   /// Compute macroscopic fields on every nodes of the current buffer.
246   void compute_macros () {
247     for (size_t iX=0; iX < m_size[0]; ++iX) {
248       for (size_t iY=0; iY < m_size[1]; ++iY) {
249         for (size_t iZ=0; iZ < m_size[2]; ++iZ) {
250           m_buffer[m_cbuf][iX][iY][iZ]->compute_macros();
251         } //~ for iZ
252       } //~ for iY
253     } //~ for iX
254   } //~  compute_macros ()
255
256   /// Simulation step procedure.
257   void propagate_and_collide () {
258     propagate();
259     compute_macros();
```

```
260     collide();
261   } //~ propagate_and_collide ()
262
263   /// Collisions.
264   void collide () {
265     for (size_t iX=0; iX < m_size[0]; ++iX) {
266       for (size_t iY=0; iY < m_size[1]; ++iY) {
267         for (size_t iZ=0; iZ < m_size[2]; ++iZ) {
268           m_buffer[m_cbuf][iX][iY][iZ]->collide();
269         } //~ for iZ
270       } //~ for iY
271     } //~ for iX
272   } //~ collide ()
273
274   /// Buffer swapping -- double buffering.
275   void swap () {
276     m_cbuf += 1; m_cbuf %= 2;
277     m_obuf += 1; m_obuf %= 2;
278   } //~ swap()
279
280   /// Flux limiter's Psi function.
281   double psi(double r) {
282     // Minmod
283     //return std::max(0.,std::min(1.,r));
284     // Superbee
285     //return std::max(0.,std::max(std::min(2.*r,1.),std::min(r,2.)));
286     // Koren
287     return std::max(0.,std::min(std::min(2.*r,(2.+r)/3.),2.));
288     // MC
289     //return std::max(0.,std::min(std::min(2.*r,.5*(1.+r)),2.));
290   } //~ psi()
291
292   /// Propagation.
293   void propagate () {
294
295     /// Transport along X.
296     for (size_t iX = 0; iX < m_size[0]; ++iX) {
297       for (size_t iY = 0; iY < m_size[1]; ++iY) {
298         for (size_t iZ = 0; iZ < m_size[2]; ++iZ) {
299
300           node_type currentNode = m_buffer[m_cbuf][iX][iY][iZ]->type();
301
302           if (currentNode != Solid) {
303
304             for (size_t index = 0; index < T_N; ++index) {
305
306               int vu = m_buffer[m_cbuf][iX][iY][iZ]->vspace.vunit(index,0);
307
308               size_t next  = (iX + vu + 2*m_size[0])%m_size[0];
309               node_type nextNode = m_buffer[m_cbuf][next][iY][iZ]->type();
310
311               size_t pre   = (iX - vu + 2*m_size[0])%m_size[0];
312               node_type preNode = m_buffer[m_cbuf][pre][iY][iZ]->type();
313
314               size_t pre2  = (iX - 2*vu + 2*m_size[0])%m_size[0];
```

```
315             node_type pre2Node = m_buffer[m_cbuf][pre2][iY][iZ]->type();
316
317             double fnext = m_buffer[m_cbuf][next][iY][iZ]->f(index);
318             double f = m_buffer[m_cbuf][iX][iY][iZ]->f(index);
319             double fpre = m_buffer[m_cbuf][pre][iY][iZ]->f(index);
320             double fpre2 = m_buffer[m_cbuf][pre2][iY][iZ]->f(index);
321
322             if (preNode == Solid || (preNode == Bound && (iX == 0 || iX == (m_size[0]-1)))) {
323               fpre = (4./3.)*f - (1./3.)*fnext;
324               fpre2 = (4./3.)*fpre - (1./3.)*f;
325             }
326             else if (pre2Node == Solid || pre2Node == Bound)
327               fpre2 = (4./3.)*fpre - (1./3.)*f;
328
329             double a  = f - fpre;
330             double b  = fnext - f;
331             double rb = (fpre - fpre2)/(f - fpre);
332
333             double absv  = m_buffer[m_cbuf][iX][iY][iZ]->vspace.vabs(index,0);
334             double absv2 = absv*(1.-absv)/2.;
335
336             m_buffer[m_obuf][iX][iY][iZ]->
337               set_f(index,f - absv*a - absv2*(psi(a/b)*b - psi(rb)*a));
338
339           } //~ for index
340
341         } //~ if != Solid
342       } //~ for iZ
343     } //~ for iY
344   } //~ for iX
345   swap();
346
347   /// Transport along Y.
348   if (T_D >= 2) { // i.e. if we run a D>1 simulation.
349     for (size_t iX=0; iX < m_size[0]; ++iX) {
350       for (size_t iY=0; iY < m_size[1]; ++iY) {
351         for (size_t iZ=0; iZ < m_size[2]; ++iZ) {
352
353           node_type currentNode = m_buffer[m_cbuf][iX][iY][iZ]->type();
354
355           if (currentNode != Solid) {
356
357             for (size_t index=0; index < T_N; ++index) {
358
359               int vu = m_buffer[m_cbuf][iX][iY][iZ]->vspace.vunit(index,1);
360
361               size_t next  = (iY +   vu + 2*m_size[1])%m_size[1];
362               node_type nextNode = m_buffer[m_cbuf][iX][next][iZ]->type();
363
364               size_t pre   = (iY -   vu + 2*m_size[1])%m_size[1];
365               node_type preNode = m_buffer[m_cbuf][iX][pre][iZ]->type();
366
367               size_t pre2  = (iY - 2*vu + 2*m_size[1])%m_size[1];
368               node_type pre2Node = m_buffer[m_cbuf][iX][pre2][iZ]->type();
369
```

```
370                 double fnext = m_buffer[m_cbuf][iX][next][iZ]->f(index);
371                 double f     = m_buffer[m_cbuf][iX][iY][iZ]->f(index);
372                 double fpre  = m_buffer[m_cbuf][iX][pre][iZ]->f(index);
373                 double fpre2 = m_buffer[m_cbuf][iX][pre2][iZ]->f(index);
374
375                 if (preNode == Solid || (preNode == Bound && (iY == 0 || iY == (m_size[1]-1))))
{
376                   fpre = (4./3.)*f - (1./3.)*fnext;
377                   fpre2 = (4./3.)*fpre - (1./3.)*f;
378                 }
379                 else if (pre2Node == Solid || pre2Node == Bound)
380                   fpre2 = (4./3.)*fpre - (1./3.)*f;
381
382                 double a  = f - fpre;
383                 double b  = fnext - f;
384                 double rb = (fpre - fpre2)/(f - fpre);
385
386                 double absv  = m_buffer[m_cbuf][iX][iY][iZ]->vspace.vabs(index,1);
387                 double absv2 = absv*(1.-absv)/2.;
388
389                 m_buffer[m_obuf][iX][iY][iZ]->
390                   set_f(index,f - absv*a - absv2*(psi(a/b)*b - psi(rb)*a));
391
392               } //~ for index
393             } //~ if != Solid
394           } //~ for iZ
395         } //~ for iY
396       } //~ for iX
397       swap();
398     } //~ if (T_D >= 2)
399
400     /// Transport in direction Z.
401     // if (T_D == 3) { // i.e. if we run a 3D simulation.
402     //
403     // } //~ if (T_D == 3)
404
405   } //~ propagate ()
406
407 private:
408   /// The main simulation container.
409   // m_buffer contain 2 buffers (boost::array<-,2>).
410   // Each one is a 3-dimensional array boost::multi_array<-,3>
411   // of pointers on nodes  Node<T_D, T_N>*.
412   boost::array<boost::multi_array<Node<T_D, T_N>*,3>,2> m_buffer;
413
414   /// Size of the simulation.
415   boost::array<size_t,3> m_size;
416   /// The current buffer.
417   unsigned short int m_cbuf;
418   /// The other buffer.
419   unsigned short int m_obuf;
420
421 }; //~ class Lattice
422
423 #endif /* _LATTICE_H_ */
```

C.2.7 Nœuds du réseau – Class Node

Comme énoncé dans la section précédente, les éléments (ou nœuds) du réseau sont des classes filles d'une classe abstraite `Node<T_D,T_N>` paramétrée par la dimension du problème `T_D` et le nombre de vitesses discrètes `T_N`. Cette classe abstraite est la classe mère des différents types de nœuds utilisés pour construire des simulations. Trois types de nœuds sont implémentés ici. Les deux premiers types sont très similaires ; ce sont les nœuds de type `Fluid_Node<T_D,T_N>` (ligne 87) et `Bound_Node<T_D,T_N>` (ligne 286), décrivant respectivement le fluide et les bords (interfaces fluide-solide). La différence principale entre ces deux types de nœuds est leur méthode *collide()*, qui pour le nœud `Fluid_Node<T_D,T_N>` est l'opérateur de collisions BGK et pour le nœud `Bound_Node<T_D,T_N>` est l'opérateur de collisions aux bords de Maxwell avec accommodation totale. Le nœud `Bound_Node<T_D,T_N>` comporte en outre l'information sur la normale aux bords. Le dernier type de nœud `Solid_Node<T_D,T_N>` (ligne 476), décrit quant à lui le comportement du solide. Ses méthodes sont vides ; seuls les assesseurs aux champs macroscopiques retournent des valeurs constantes et codées en dur.

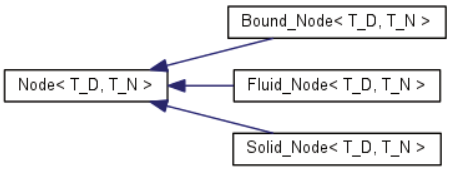


FIGURE C.4 – Graph d'héritage des nœuds.

```
1  /// \file    node.hpp
2  /// \author Léonard de Izarra <leonardizarra@gmail.com>
3  /// \date    Tue Jun 22 18:47:26 2010
4
5  #ifndef _NODE_H_
6  #define _NODE_H_
7
8  #include "globals.hpp"
9  #include "vspace.hpp"
10
11 /// \class Node
12 /// \bref Abstract base class for lattice nodes.
13 /// This classe should be differentiaded into Solide_Node,
14 /// Fluid_Node, etc ... for built simulations.
15 /// \param T_D the dimension of the simulation.
16 /// \param T_N the total number of discretes velocity.
17 template<size_t T_D, size_t T_N>
18 class Node {
19 public:
20   /// Ctor.
21   Node () { }
```

```
22  /// Dtor.
23  virtual ~Node () { }
24
25  /// Node type accessor.
26  virtual node_type type () { }
27
28  /// Velocity space.
29  Vspace<T_D, T_N> vspace;
30
31  /// Distribution function accessor.
32  virtual double f (size_t index) { }
33  /// Distribution function setter.
34  virtual void set_f (size_t index, double value) { }
35  /// Density accessor.
36  virtual double rho () { }
37  /// Density setter.
38  virtual void set_rho (double value) { }
39  /// Referece density accessor.
40  virtual double rho0 () { }
41  /// Referance density setter.
42  virtual void set_rho0 (double value) { }
43  /// Macroscopic velocity accessor.
44  virtual double u (size_t axis) { }
45  /// Macroscopic velocity setter.
46  virtual void set_u (size_t axis, double value) { }
47  /// Temperature accessor.
48  virtual double theta () { }
49  /// Temperature setter.
50  virtual void set_theta (double value) { }
51  /// Reference temperature accessor (this temperature can't be modified).
52  virtual double theta0 () { }
53  /// External force density accessor.
54  virtual double Fext (size_t axis) { }
55  /// External force density setter.
56  virtual void set_Fext (size_t axis, double value) { }
57  /// Collision freqency accessor.
58  virtual double omega () { }
59  /// Collision freqency setter.
60  virtual void set_omega0 (double value) { }
61
62  /// Compute \f$\rho\f$ from distribution function.
63  virtual void compute_rho () { }
64  /// Compute \f$u_{axis}\f$ component from distribution function.
65  virtual void compute_u (size_t axis) { }
66  /// Compute \f$\theta\f$ from distribution function.
67  virtual void compute_theta () { }
68  /// Compute \f$\rho\f$, \f$\vec{u}\f$ and \f$\theta\f$ from distribution function.
69  virtual void compute_macros () { }
70  /// Compute \f$f_{i}^{eq}\f$ from macroscopic fields (Momenta).
71  virtual double compute_eq (size_t index) { }
72
73  /// Normal accessor (only for boundary nodes)
74  virtual double nomal (size_t axis) { }
75  /// Normal setter(only for boundary nodes).
76  virtual void set_normal (size_t axis, double value) { }
```

```
77
78   /// The collide procedure.
79   virtual void collide () = 0;
80
81 }; //~ class Node
82
83 /// \class Fluid_Node
84 /// Fluid node class.
85 /// \param T_D the dimension of the simulation.
86 /// \param T_N the total number of discretes velocity.
87 template<size_t T_D, size_t T_N>
88 class Fluid_Node : public Node<T_D, T_N> {
89 public:
90   /// Ctor
91   Fluid_Node () : m_node_type (Fluid) { }
92   /// Dtor
93   virtual ~Fluid_Node () { }
94
95   /// Node type accessor.
96   node_type type () {
97     return m_node_type;
98   } //~ type ()
99
100  /// Distribution function accessor.
101  double f (size_t index) {
102    return m_f[index];
103  } //~ f ()
104
105  /// Distribution function setter.
106  void set_f (size_t index, double value) {
107    m_f[index] = value;
108  } //~ set_f ()
109
110  /// Reference density accessor.
111  double rho0 () {
112    return m_rho0;
113  } //~ rho0 ()
114
115  /// Reference density setter.
116  void set_rho0 (double value) {
117    m_rho0 = value;
118  } //~ set_rho0 ()
119
120  /// Density accessor.
121  double rho () {
122    return m_rho;
123  } //~ rho ()
124
125  /// Density setter.
126  void set_rho (double value) {
127    m_rho = value;
128  } //~ set_rho ()
129
130  /// Macroscopic velocity accessor.
131  double u (size_t axis) {
```

```
132     return m_u[axis];
133   } //~ u ()
134
135   /// Macroscopic velocity setter.
136   void set_u (size_t axis, double value) {
137     m_u[axis] = value;
138   } //~ set_u ()
139
140   /// Temperature accessor.
141   double theta () {
142     return m_theta;
143   } //~ theta ()
144
145   /// Temperature setter.
146   void set_theta (double value) {
147     m_theta = value;
148   } //~ set_theta ()
149
150   /// Temperature accessor.
151   double theta0 () {
152     return Node<T_D,T_N>::vspace.c02();
153   } //~ theta0 ()
154
155   /// External force density accessor.
156   double Fext (size_t axis) {
157     return m_Fext[axis];
158   } //~  Fext  ()
159
160   /// External force density setter.
161   void set_Fext (size_t axis, double value) {
162     m_Fext[axis] = value;
163   } //~ set_Fext ()
164
165   /// Collision freqency accessor.
166   /// Compute the local collision frequency.
167   double omega () {
168     return m_omega0*(m_rho*std::sqrt(m_theta))/(m_rho0*Node<T_D,T_N>::vspace.c0());
169   } //~ omega ()
170
171   /// Reference collision freqency setter.
172   void set_omega0 (double value) {
173     m_omega0 = value;
174   } //~ set_omega0 ()
175
176   /// Compute \f$\rho\f$ from distribution function.
177   /// \todo Documenter cette fonction.
178   void compute_rho () {
179     double a_rho = 0.;
180     for(size_t index = 0; index < T_N; ++index) {
181       a_rho += m_f[index];
182     } //~ for index
183     m_rho = a_rho;
184   } //~ compute_rho ()
185
186   /// Compute \f$u_{axis}\f$ component from distribution function.
```

```
187   /// \todo Documenter cette fonction.
188   void compute_u (size_t axis) {
189     double a_u = 0.;
190     for(size_t index = 0; index < T_N; ++index) {
191       a_u += Node<T_D,T_N>::vspace.v(index,axis) * m_f[index];
192     } //~ for index
193     m_u[axis] = a_u / m_rho;
194   } //~ compute_u ()
195
196   /// Compute \f$\theta\f$ from distribution function.
197   /// \todo Documenter cette fonction.
198   void compute_theta () {
199     double a_theta = 0.;
200     for(size_t index = 0; index < T_N; ++index) {
201       a_theta += Node<T_D,T_N>::vspace.v2(index) * m_f[index];
202     } //~ for index
203     double u2 = 0.;
204     for (size_t axis = 0; axis < T_D; ++axis) {
205       u2 += m_u[axis] * m_u[axis];
206     } //~ for axis
207     m_theta = (a_theta / m_rho - u2) / (double)T_D;
208   } //~ compute_theta ()
209
210   /// Compute \f$\rho\f$, \f$\vec{u}\f$ and \f$\theta\f$ from distribution function.
211   /// \todo Documenter cette fonction.
212   void compute_macros () {
213     compute_rho();
214     for (size_t axis = 0; axis < T_D; ++axis) {
215       compute_u(axis);
216     } //~ for axis
217     compute_theta();
218   } //~ compute_macros()
219
220   /// Compute \f$f_{i}^{eq}\f$ from macroscopic fields (Momenta).
221   /// \todo Documenter cette fonction.
222   double compute_eq (size_t index) {
223     double u2 = 0., vu = 0.;
224     for (size_t axis = 0; axis < T_D; ++axis) {
225       u2 += (m_u[axis] + m_Fext[axis] / (omega() * m_rho))*
226         (m_u[axis] + m_Fext[axis] / (omega() * m_rho));
227       vu += Node<T_D,T_N>::vspace.v(index,axis) * (m_u[axis] + m_Fext[axis] / (omega() * m_rho));
228     } //~ for axis
229
230     double vu2 = vu*vu;
231     double vu3 = vu2*vu;
232
233     double v2    = Node<T_D,T_N>::vspace.v2(index);
234     double w     = Node<T_D,T_N>::vspace.w(index);
235     double k     = Node<T_D,T_N>::vspace.k();
236     double kon2  = Node<T_D,T_N>::vspace.kon2();
237     double k2    = Node<T_D,T_N>::vspace.k2();
238     double k2on2 = Node<T_D,T_N>::vspace.k2on2();
239     double k3    = Node<T_D,T_N>::vspace.k3();
240     double k3on6 = Node<T_D,T_N>::vspace.k3on6();
241
```

```
242     /// f^eq with N = 2
243     return w * m_rho * (1. + k*vu + k2on2*vu2  - kon2*u2 + kon2*(m_theta - theta0())*(k*v2 - T_D));
244
245     /// f^eq with N = 3
246     // return w * m_rho * (1. + k*vu + k2on2*vu2  - kon2*u2 - k2on2*vu*u2 + k3on6*vu3
247     //                         + kon2*(m_theta - theta0())*(k*v2 - T_D)
248     //                         + k2on2*(m_theta - theta0())*(k*v2 - 2*T_D)*vu);
249   } //~ compute_eq
250
251   /// The collide procedure.
252   /// \todo Documenter cette fonction. Operateur BGK
253   void collide () {
254     for(size_t index = 0; index < T_N; ++index) {
255       m_f[index] *= 1. - omega();
256       m_f[index] += omega() * compute_eq(index);
257     } //~ for index
258   } //~ collide()
259
260 private:
261   /// Distribution function.
262   boost::array<double, T_N> m_f;
263   /// Reference collision freqency \f$\omega_0 = \frac{1}{\tau_0}\f$
264   /// where \f$\tau_0\f$ is the reference relaxation time.
265   double m_omega0;
266   /// \f$\rho_0\f$ : Reference density.
267   double m_rho0;
268   /// \f$\rho\f$ : Density.
269   double m_rho;
270   /// \f$\vec{u}\f$ : Macroscopic velocity.
271   boost::array<double, T_D> m_u;
272   /// \f$\equiv \theta = \frac{k_B T}{m}\f$ where \f$T\f$ is the Temperature.
273   double m_theta;
274   /// External force density.
275   boost::array<double, T_D> m_Fext;
276   /// Node type.
277   node_type m_node_type;
278
279 }; //~ class Fluid_Node
280
281
282 /// \class Bound_Node
283 /// Boundary node class.
284 /// \param T_D the dimension of the simulation.
285 /// \param T_N the total number of discretes velocity.
286 template<size_t T_D, size_t T_N>
287 class Bound_Node : public Node<T_D, T_N> {
288 public:
289   /// Ctor.
290   Bound_Node () : m_node_type (Bound) { }
291   /// Dtor.
292   virtual ~Bound_Node () { }
293
294   /// Node type accessor.
295   node_type type () {
296     return m_node_type;
```

```
297   } //~ type ()
298
299   /// Normal accessor.
300   double normal (size_t axis) {
301     return m_normal[axis];
302   } //~ normal ()
303
304   /// Normal setter.
305   void set_normal (size_t axis, double value) {
306     m_normal[axis] = value;
307   } //~ set_normal ()
308
309   /// Distribution function accessor.
310   double f (size_t index) {
311     return m_f[index];
312   } //~ f ()
313
314   /// Distribution function setter.
315   void set_f (size_t index, double value) {
316     m_f[index] = value;
317   } //~ set_f ()
318
319   /// Reference density accessor.
320   double rho0 () {
321     return m_rho0;
322   } //~ rho0 ()
323
324   /// Reference density setter.
325   void set_rho0 (double value) {
326     m_rho0 = value;
327   } //~ set_rho0 ()
328
329   /// Density accessor.
330   double rho () {
331     return m_rho;
332   } //~ rho ()
333
334   /// Density setter.
335   void set_rho (double value) {
336     m_rho = value;
337   } //~ set_rho ()
338
339   /// Macroscopic velocity accessor.
340   double u (size_t axis) {
341     return m_u[axis];
342   } //~ u ()
343
344   /// Macroscopic velocity setter.
345   void set_u (size_t axis, double value) {
346     m_u[axis] = value;
347   } //~ set_u ()
348
349   /// Temperature accessor.
350   double theta () {
351     return m_theta;
```

```
352   } //~ theta ()
353
354   /// Temperature setter.
355   void set_theta (double value) {
356     m_theta = value;
357   } //~ set_theta ()
358
359   /// Reference temperature accessor.
360   double theta0 () {
361     return Node<T_D,T_N>::vspace.c02();
362   } //~ theta0 ()
363
364   /// Compute rho from distribution function.
365   void compute_rho () {
366     double a_rho = 0.;
367     for(size_t index = 0; index < T_N; ++index) {
368       a_rho += m_f[index];
369     } //~ for index
370     m_rho = a_rho;
371   } //~ compute_rho ()
372
373   /// Compute \f$u_{axis}\f$ component from distribution function.
374   void compute_u (size_t axis) {
375   } //~ compute_u ()
376
377   /// Compute \f$\theta\f$ from distribution function.
378   void compute_theta () {
379   } //~ compute_theta ()
380
381   /// Compute \f$\rho\f$, \f$\vec{u}\f$ and \f$\theta\f$ from distribution function.
382   void compute_macros () {
383     compute_rho();
384   } //~ compute_macros ()
385
386   /// Compute \f$f_{i}^{eq}\f$ from macroscopic fields (Momenta).
387   double compute_eq (size_t index) {
388
389     double u2 = 0., vu = 0.;
390     for (size_t axis = 0; axis < T_D; ++axis) {
391       u2 += std::pow(m_u[axis],2);
392       vu += Node<T_D,T_N>::vspace.v(index,axis) * (m_u[axis]);
393     } //~ for axis
394
395     double vu2 = vu*vu;
396     double vu3 = vu2*vu;
397
398     double v2    = Node<T_D,T_N>::vspace.v2(index);
399     double w     = Node<T_D,T_N>::vspace.w(index);
400     double k     = Node<T_D,T_N>::vspace.k();
401     double kon2  = Node<T_D,T_N>::vspace.kon2();
402     double k2    = Node<T_D,T_N>::vspace.k2();
403     double k2on2 = Node<T_D,T_N>::vspace.k2on2();
404     double k3    = Node<T_D,T_N>::vspace.k3();
405     double k3on6 = Node<T_D,T_N>::vspace.k3on6();
406
```

```
407     /// f^eq with N = 2
408     return w * m_rho * (1. + k*vu + k2on2*vu2  - kon2*u2 + kon2*(m_theta - theta0())*(k*v2 - T_D));
409
410     /// f^eq with N = 3
411     // return w * m_rho * (1. + k*vu + k2on2*vu2  - kon2*u2 - k2on2*vu*u2 + k3on6*vu3
412     //                         + kon2*(m_theta - theta0())*(k*v2 - T_D)
413     //                         + k2on2*(m_theta - theta0())*(k*v2 - 2*T_D)*vu);
414   } //~ compute_eq ()
415
416   /// The collide procedure.
417   void collide () {
418
419     double num = 0., denom = 0.;
420     for (size_t index = 0; index < T_N; ++index) {
421
422       double vn = 0.; double vnn = 0.;
423       for (size_t axis = 0; axis < T_D; ++axis) {
424         vn += (Node<T_D,T_N>::vspace.v(index,axis)-u(axis)) * m_normal[axis];
425         vnn += vn*vn;//std::pow((Node<T_D,T_N>::vspace.v(index,axis)) * m_normal[axis],2);
426       } //~ for axis
427
428       if (vn < 0) {
429         num += std::sqrt(vnn) * m_f[index];
430         denom += std::sqrt(vnn) * compute_eq(index);
431       }
432
433     } //~ for index
434
435     for (size_t index = 0; index < T_N; ++index) {
436
437       double vn = 0.;
438       for (size_t axis = 0; axis < T_D; ++axis) {
439         vn += Node<T_D,T_N>::vspace.v(index,axis) * m_normal[axis];
440       } //~ for axis
441
442       if (vn > 0) {
443         m_f[index] = (num / denom) * compute_eq(index);
444       }
445
446     } //~ for index
447
448   }//~ collide ()
449
450 private:
451   /// Distribution function.
452   boost::array<double, T_N> m_f;
453   /// \f$\rho_0\f$ : Reference density.
454   double m_rho0;
455   /// \f$\rho\f$ : Density.
456   double m_rho;
457   /// \f$\vec{u}_{(\vec{n},\vec{t})}\f$ : Imposed macroscopic velocity.
458   /// \f$\vec{u}_{(\vec{n},\vec{t})}\f$ is expressed in the  \f${(\vec{n},\vec{t})}\f$
459   /// basis, where the tangant vector \f$\vec{t}\f$ is build by a \f$\frac{\pi}{2}\f$
460   /// rotation of the normal vector \f$\vec{n}\f$ in the xOy plan.
461   boost::array<double, T_D> m_u;
```

```
462   /// \f$\equiv \theta = \frac{k_B T}{m}\f$ where \f$T\f$ is the Temperature.
463   double m_theta;
464   /// Normal to boundary.
465   boost::array<double, T_D> m_normal;
466   /// Node type.
467   node_type m_node_type;
468
469 }; //~ class Bound_Node
470
471 /// \class Solid_Node
472 /// Solid node class.
473 /// \param T_D the dimension of the simulation.
474 /// \param T_N the total number of discretes velocity.
475 template<size_t T_D, size_t T_N>
476 class Solid_Node : public Node<T_D, T_N> {
477 public:
478   /// Ctor
479   Solid_Node () : m_node_type (Solid) { }
480   /// Dtor
481   virtual ~Solid_Node () { }
482
483   /// Node type accessor.
484   node_type type () {
485     return m_node_type;
486   } //~ type ()
487
488   /// Distribution function accessor.
489   double f (size_t index) {
490     return Node<T_D,T_N>::vspace.w(index);;
491   } //~ f ()
492
493   /// Distribution function setter.
494   void set_f (size_t index, double value) {
495   } //~ set_f ()
496
497   /// Reference density accessor.
498   double rho0 () {
499     return 1.;
500   } //~ rho0 ()
501
502   /// Reference density setter.
503   void set_rho0 (double value) {
504   } //~ set_rho0 ()
505
506   /// Density accessor.
507   double rho () {
508     return 1.;
509   } //~ rho ()
510
511   /// Density setter.
512   void set_rho (double value) {
513   } //~ set_rho ()
514
515   /// Macroscopic velocity accessor.
516   double u (size_t axis) {
```

```
517     return 0.;
518   } //~ u ()
519
520   /// Macroscopic velocity setter.
521   void set_u (size_t axis, double value) {
522   } //~ set_u ()
523
524   /// Temperature accessor.
525   double theta () {
526     return Node<T_D,T_N>::vspace.c02();
527   } //~ theta ()
528
529   /// Temperature setter.
530   void set_theta (double value) {
531   } //~ set_theta ()
532
533   /// Temperature accessor.
534   double theta0 () {
535     return Node<T_D,T_N>::vspace.c02();
536   } //~ theta0 ()
537
538   /// Collision freqency accessor.
539   double omega () {
540   } //~ omega ()
541
542   /// Collision freqency setter.
543   void set_omega0 (double value) {
544   } //~ set_omega0 ()
545
546   /// Compute \f$\rho\f$ from distribution function.
547   void compute_rho () {
548   } //~ compute_rho ()
549
550   /// Compute \f$u_{axis}\f$ component from distribution function.
551   void compute_u (size_t axis) {
552   } //~ compute_u ()
553
554   /// Compute \f$\theta\f$ from distribution function.
555   void compute_theta () {
556   } //~ compute_theta ()
557
558   /// Compute \f$\rho\f$, \f$\vec{u}\f$ and \f$\theta\f$ from distribution function.
559   void compute_macros () {
560   } //~ compute_macros()
561
562   /// Compute \f$f_{i}^{eq}\f$ from macroscopic fields (Momenta).
563   double compute_eq (size_t index) {
564   } //~ compute_eq ()
565
566   /// The collide procedure.
567   void collide () {
568   } //~ collide()
569
570 private:
571   /// Node type.
```

```
572   node_type m_node_type;
573
574 }; //~ class Solid_Node
575
576 #endif /* _NODE_H_ */
```

C.2.8 Espace des vitesses – Class Vspace

Dans les programmes donnés en exemple, la dimension de l'espace est 2 (TD=2) et le nombre de vitesses discrètes est 16 (TN=16), il s'agit donc d'un schéma basé sur $\mathcal{H}_4$. Dans ces exemples, le système de vitesses (vitesses discrètes et poids associés) sont chargés par la méthode *load_velocity_space()* qui n'est qu'un lien vers la méthode *load()* de la classe Vspace dans la classe Lattice ; la commande est la suivante :

```
L.load_velocity_space("vspace.txt");
```

La lecture de cette méthode de Vspace (ligne 18) permet de déduire l'organisation (le format) que doit avoir le fichier *vspace.txt*. Ce fichier doit contenir comme premier élément la norme de la vitesse maximale, puis les coordonnées de chacune des vitesses discrètes **normalisées** suivies de leur poids.

Soit, en 2D :

v_{max}
$v_{0\,x}/v_{max} \quad v_{0\,y}/v_{max} \quad \omega_0$
...
$v_{q\,x}/v_{max} \quad v_{q\,y}/v_{max} \quad \omega_q$

Exemple pour un schéma basé sur $\mathcal{H}_4$ en dimension 2 :

```
2.33441421833898
-1 -1 0.00210459405070091
-1 -0.317837245195782 0.0208333333333333
-1 0.317837245195782 0.0208333333333333
-1 1 0.00210459405070091
-0.317837245195782 -1 0.0208333333333333
-0.317837245195782 -0.317837245195782 0.206228739282632
-0.317837245195782 0.317837245195782 0.206228739282632
-0.317837245195782 1 0.0208333333333333
0.317837245195782 -1 0.0208333333333333
0.317837245195782 -0.317837245195782 0.206228739282632
0.317837245195782 0.317837245195782 0.206228739282632
0.317837245195782 1 0.0208333333333333
1 -1 0.00210459405070091
1 -0.317837245195782 0.0208333333333333
1 0.317837245195782 0.0208333333333333
1 1 0.00210459405070091
```

```
1 /// \file    vspace.hpp
2 /// \author Léonard de Izarra <leonardizarra@gmail.com>
3 /// \date    Mon Jun 14 13:42:58 2010
4
```

```
5 #ifndef _VSPACE_H_
6 #define _VSPACE_H_
7
8 #include "globals.hpp"
9
10 /// \class Vspace
11 /// \bref Velocity Space template class.
12 /// \param T_D the dimension of the simulation.
13 /// \param T_N the total number of discretes velocity.
14 template<size_t T_D, size_t T_N>
15 class Vspace {
16 public:
17
18   /// Inititialisation: load velocity space from file.
19   void load (const char* inputFile) {
20     std::stringstream fileT_Name; fileT_Name << inputFile;
21     std::fstream in;
22     in.open(fileT_Name.str().c_str(), std::fstream::in);
23
24     in >> m_c0;
25     m_c0 = 1./m_c0;
26
27     for (size_t index = 0; index < T_N; ++index) {
28       for (size_t axis = 0; axis < T_D; ++axis) {
29         in >> m_v[index][axis];
30       } //~ for axis
31       in >> m_w[index];
32     }  //~ for index
33
34     for (size_t index = 0; index < T_N; ++index) {
35       double v2 = 0.;
36       for (size_t axis = 0; axis < T_D; ++axis) {
37
38         if (m_v[index][axis] == 0) {
39           m_vunit[index][axis] = 0;
40         } //~ if
41         else {
42           m_vunit[index][axis] = m_v[index][axis] / std::abs(m_v[index][axis]);
43         } //~ else
44
45         v2 += m_v[index][axis] * m_v[index][axis];
46         m_vabs[index][axis] = std::abs(m_v[index][axis]);
47       }//~ for axis
48       m_v2[index] = v2;
49     } //~ for index
50
51     m_c02   = m_c0*m_c0;
52     m_k     = 1./m_c02;
53     m_kon2  = m_k/2.;
54     m_k2    = m_k*m_k;
55     m_k2on2 = m_k2/2.;
56     m_k3    = m_k*m_k*m_k;
57     m_k3on6 = m_k3/6.;
58
59   } //~ load()
```

```
60
61  /// Size accessor.
62  size_t size() const {
63    return T_N;
64  }
65
66  /// Velocity accessor.
67  double v (size_t index, size_t axis) const {
68    return m_v[index][axis];
69  }
70
71  /// Squar velocity accessor.
72  double v2 (size_t index) const {
73    return m_v2[index];
74  }
75
76  /// Normalised velocity accessor -- The beam's direction.
77  double vunit (size_t index, size_t axis) const {
78    return m_vunit[index][axis];
79  }
80
81  /// Normalised velocity accessor -- The beam's direction.
82  double vabs (size_t index, size_t axis) const {
83    return m_vabs[index][axis];
84  }
85
86  /// Weight accesor.
87  double w (size_t index) const {
88    return m_w[index];
89  }
90
91  /// Maximal velocity accessor -- Physiclaly, the speed of the sound.
92  double c0 () const {
93    return m_c0;
94  }
95
96  /// Constants accessors.
97  double c02 () const {
98    return m_c02;
99  }
100 double k () const {
101   return m_k;
102 }
103 double kon2 () const {
104   return m_kon2;
105 }
106 double k2 () const {
107   return m_k2;
108 }
109 double k2on2 () const {
110   return m_k2on2;
111 }
112
113 double k3 () const {
114   return m_k3;
```

```
115   }
116
117   double k3on6 () const {
118     return m_k3on6;
119   }
120 private:
121   /// Data
122   boost::array<boost::array<double,T_D>,T_N> m_v;
123   boost::array<double,T_N> m_w;
124   double m_c0;
125
126   boost::array<double,T_N> m_v2;
127   boost::array<boost::array<double,T_D>,T_N> m_vunit;
128   boost::array<boost::array<double,T_D>,T_N> m_vabs;
129   boost::array<size_t,T_N> m_Iop;
130
131   /// Some constants.
132   double m_c02, m_k, m_kon2, m_k2, m_k2on2, m_k3, m_k3on6;
133 };
134
135 #endif /* _VSPACE_H_ */
```

C.2.9 Variables globales

Le fichier suivant, *globals.hpp*, contient les inclusions de fichiers d'en-têtes utiles ainsi qu'une énumération des types de nœuds (ligne 21) implémentés dans le code maBLB.

```
1  /// \file    globals.hpp
2  /// \author Léonard Izarra <leonardizarra@gmail.com>
3  /// \date    Tue Jun 22 18:46:49 2010
4  /// \brief   Global definitions
5
6  #ifndef _GLOBALS_H_
7  #define _GLOBALS_H_
8
9  #include <boost/multi_array.hpp>
10 #include <boost/array.hpp>
11
12 #include <fstream>
13 #include <iostream>
14 #include <sstream>
15
16 #include <cmath>
17 #include <limits>
18
19 enum node_type {Solid = 0, Fluid = 1, Bound = 2};
20
21 #define PRECISION std::numeric_limits<double>::digits10
22
23 #endif /* _GLOBALS_H_ */
24
```

Bibliographie

[AAC+07] S. Ansumali, S. Arcidiacono, S. S. Chikatamarla, N. I. Prasianakis, A. N. Gorban et I. V. Karlin : Quasi-equilibrium lattice boltzmann method. *The European Physical Journal B - Condensed Matter and Complex Systems*, 56:135–139, 2007. 10.1140/epjb/e2007-00100-1.

[AK02] Santosh Ansumali et Iliya V. Karlin : Single relaxation time model for entropic lattice boltzmann methods. *Phys. Rev. E*, 65(5):056312, May 2002.

[AKA+07] S. Ansumali, I. V. Karlin, S. Arcidiacono, A. Abbas et N. I. Prasianakis : Hydrodynamics beyond navier-stokes: Exact solution to the lattice boltzmann hierarchy. *Phys. Rev. Lett.*, 98(12):124502, Mar 2007.

[AKFB06] S. Ansumali, I.V. Karlin, C.E. Frouzakis et K.B. Boulouchos : Entropic lattice boltzmann method for microflows. *Physica A: Statistical Mechanics and its Applications*, 359:289–305, 2006.

[AMB97] E.B. Arkilic, M.A.Schmidt et K.S. Breuer : Gaseous slip flow in long microchannels. *Microelectromechanical Systems*, 6:167 – 178, 1997.

[ASCW05] R. Adhikari, K. Stratford, M. E. Cates et A. J. Wagner : Fluctuating lattice boltzmann. *EPL (Europhysics Letters)*, 71(3):473, 2005.

[AV02] Santosh Ansumali et Iliya V. Karlin : Kinetic boundary conditions in the lattice boltzmann method. *Phys. Rev. E*, 66(2):026311, Aug 2002.

[BCDB89] Diu Bernard, Guthmann Claudine, Lederer Danielle et ROULET Bernard : *Eléments de physique statistique*. Hermann, Paris, 1989.

[BGK54] P. L. Bhatnagar, E. P. Gross et M. Krook : A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. *Phys. Rev.*, 94(3):511–525, May 1954.

[Bir94] G. A. Bird : *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Clarendon, Oxford, 1994.

[BKG06] A. Bardow, I. V. Karlin et A. A. Gusev : General characteristic-based algorithm for off-lattice boltzmann simulations. *EPL (Europhysics Letters)*, 75(3):434, 2006.

[BKG08] André Bardow, Iliya V. Karlin et Andrei A. Gusev : Multispeed models in off-lattice boltzmann simulations. *Phys. Rev. E*, 77(2):025701, Feb 2008.

[BL08] Fubing Bao et Jianzhong Lin : Burnett simulation of gas flow and heat transfer in micro poiseuille flow. *International Journal of Heat and Mass Transfer*, 51(15-16):4139 – 4144, 2008.

[BYCW01] B. M. Boghosian, J. Yepez, P. V. Coveney et A. Wagner : Entropic lattice Boltzmann methods. *Royal Society of London Proceedings Series A*, 457:717–+, mars 2001.

[CAK06] S. S. Chikatamarla, S. Ansumali et I. V. Karlin : Grad's approximation for missing data in lattice boltzmann simulations. *EPL (Europhysics Letters)*, 74(2):215, 2006.

[CB04] Stéphane Colin et Lucien Baldas : Effets de raréfaction dans les micro-écoulements gazeux. *Comptes Rendus Physique*, 5(5):521 – 530, 2004. Microfluidics.

[CC91] S. Chapman et T.G. Cowling : *The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction, and diffusion in gases*. Cambridge Mathematical Library. Cambridge University Press, 1991.

[CD98] Shiyi Chen et Gary D Doolen : Lattice boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, 30(1):329–364, 1998.

[Cer94] C. Cercignani : *The Boltzmann Equation and Its Applications*, volume 67 de *Applied Mathematical Sciences*. Springer, Berlin, Germany; New York, U.S.A., 1994.

[CJG09] W. K.S. Chiu, A. S. Joshi et K. N. Grew : Lattice boltzmann model for multi-component mass transfer in a solid oxide fuel cell anode with heterogeneous internal reformation and electrochemistry. *The European Physical Journal - Special Topics*, 171:159–165, 2009. 10.1140/epjst/e2009-01024-8.

[CLSG11] Zhenhua Chai, Jianhua Lu, Baochang Shi et Zhaoli Guo : Gas slippage effect on the permeability of circular cylinders in a square array. *International Journal of Heat and Mass Transfer*, 54(13-14):3009 – 3014, 2011.

[CT09] Sheng Chen et Zhiwei Tian : Simulation of microchannel flow using the lattice boltzmann method. *Physica A: Statistical Mechanics and its Applications*, 388(23):4803 – 4810, 2009.

[CWB03] Chang-Hwan Choi, K. Johan A. Westin et Kenneth S. Breuer : Apparent slip flows in hydrophilic and hydrophobic microchannels. *Phys. Of Fluids*, 15(10):2897–2902, 2003.

[DAA07] Nishanth Dongari, Abhishek Agrawal et Amit Agrawal : Analytical solution of gaseous slip flow in long microchannels. *International Journal of Heat and Mass Transfer*, 50(17-18):3411 – 3421, 2007.

[dG02] P. G. de Gennes : On fluid/wall slippage. *Langmuir*, 18(9):3413–3414, 2002.

[DGK+02] Dominique D'Humières, Irina Ginzburg, Manfred Krafczyk, Pierre Lallemand et Li-Shi Luo : Multiple-relaxation-time lattice boltzmann models in three dimensions. *Philosophical Transactions of the Royal Society - Series A: Mathematical, Physical and Engineering Sciences*, 360(1792):437–451, 2002.

[DIRI09] Léonard De Izarra, Jean-Louis Rouet et Boujema Izrar : Construction d'une méthode multifaisceaux pour les écoulements en milieux poreux. *In 19ème Congrès Français de Mécanique, Marseille, France*, 2009.

[Don56] W. Dong : *Vacuum Flow of Gases Through Channels with Circular, Annular and Rectangular Cross-Sections.* Thèse de doctorat, Univ. California, 1956.

[DSL07] Burkhard Dünweg, Ulf D. Schiller et Anthony J. C. Ladd : Statistical mechanics of the fluctuating lattice boltzmann equation. *Phys. Rev. E*, 76(3):036704, Sep 2007.

[DSL09] Burkhard Dünweg, Ulf D. Schiller et Anthony J.C. Ladd : Progress in the understanding of the fluctuating lattice boltzmann equation. *Computer Physics Communications*, 180(4):605 – 608, 2009. Special issue based on the Conference on Computational Physics 2008 - CCP 2008.

[eH05] Mohamed Gad el Hak : *The MEMS Handbook*. Mohamed Gad-el-Hak, 2005.

[FHP86] U. Frisch, B. Hasslacher et Y. Pomeau : Lattice-gas automata for the navier-stokes equation. *Phys. Rev. Lett.*, 56(14):1505–1508, Apr 1986.

[GCVU08] L.S. García-Colín, R.M. Velasco et F.J. Uribe : Beyond the navier-stokes equations: Burnett hydrodynamics. *Physics Reports*, 465(4):149 – 189, 2008.

[GG08] Naveen K. Gupta et Yogesh B. Gianchandani : Thermal transpiration in zeolites: A mechanism for motionless gas pumps. *Phys. Of Fluids*, 93(19):193511, 2008.

[GHP01] É. Guyon, J.P. Hulin et L. Petit : *Hydrodynamique physique*. Savoirs actuels. EDP sciences, 2001.

[GIB+88] A. Ghizzo, B. Izrar, P. Bertrand, E. Fijalkow, M. R. Feix et M. Shoucri : Stability of bernstein–greene–kruskal plasma equilibria. numerical experiments over a long time. *Physics of Fluids*, 31(1):72–82, 1988.

[GNGB97] Martha A. Gallivan, David R. Noble, John G. Georgiadis et Richard O. Buckius : An evaluation of the bounce-back boundary condition for lattice boltzmann simulations. *International Journal for Numerical Methods in Fluids*, 25(3):249–263, 1997.

[Gra49a] H. Grad : Note on N-dimensional Hermite polynomials. *Comm. Pure Appl. Math.*, 2:325–330, 1949.

[Gra49b] H. Grad : On the kinetic theory of rarefied gases. *Comm. Pure Appl. Math.*, 2:331, 1949.

[GZS02] Zhaoli Guo, Chuguang Zheng et Baochang Shi : An extrapolation method for boundary conditions in lattice boltzmann method. *Phys. Of Fluids*, 14(6):2007–2010, 2002.

[GZS06] Zhaoli Guo, T. S. Zhao et Yong Shi : Physical symmetry, spatial accuracy, and relaxation time of the lattice boltzmann equation for microgas flows. *Phys. Of Fluids*, 99(7):074903, 2006.

[HCB54] J. O. Hirschfelder, C. F. Curtiss et R. B. Bird : *Molecular Theory of Gases and Liquids*. Wiley, New York, 1954.

[HISA11] A. Homayoon, A.H. Meghdadi Isfahani, E. Shirani et M. Ashrafizadeh : A novel modified lattice boltzmann method for simulation of gas flows in wide range of knudsen number. *International Communications in Heat and Mass Transfer*, 38(6):827 – 832, 2011.

[HL97] Xiaoyi He et Li-Shi Luo : Theory of the lattice boltzmann method: From the boltzmann equation to the lattice boltzmann equation. *Phys. Rev. E*, 56(6):6811–6817, Dec 1997.

[HS06] Jürgen Horbach et Sauro Succi : Lattice boltzmann versus molecular dynamics simulation of nanoscale hydrodynamic flows. *Phys. Rev. Lett.*, 96(22):224503, Jun 2006.

[HZLD97] Xiaoyi He, Qisu Zou, Li-Shi Luo et Micah Dembo : Analytic solutions of simple flows and analysis of nonslip boundary conditions for the lattice boltzmann bgk model. *Journal of Statistical Physics*, 87:115–136, 1997. 10.1007/BF02181482.

[IGB+89] B. Izrar, A. Ghizzo, P. Bertrand, E. Fijalkow et M. R. Feix : Integration of vlasov equation by a fast fourier eulerian code. *Computer Physics Communications*, 52(3):375–382, 1989.

[JGPC07] Abhijit S. Joshi, Kyle N. Grew, Aldo A. Peracchio et Wilson K.S. Chiu : Lattice boltzmann modeling of 2d gas transport in a solid oxide fuel cell anode. *Journal of Power Sources*, 164(2):631 – 638, 2007.

[JPGC07] Abhijit S Joshi, Aldo A Peracchio, Kyle N Grew et Wilson K S Chiu : Lattice boltzmann method for multi-component, non-continuum mass diffusion. *Journal of Physics D: Applied Physics*, 40(23):7593, 2007.

[KAM06] Muhammed E. Kutay, Ahmet H. Aydilek et Eyad Masad : Laboratory validation of lattice boltzmann method for modeling pore-scale flow in granular materials. *Computers and Geotechnics*, 33(8):381–395, 2006.

[KP08] Seung Hyun Kim et Heinz Pitsch : Analytic solution for a higher-order lattice boltzmann method: Slip velocity and knudsen layer. *Phys. Rev. E*, 78(1):016702, Jul 2008.

[KPB08a] Seung Hyun Kim, Heinz Pitsch et Iain D. Boyd : Accuracy of higher-order lattice boltzmann methods for microscale flows with finite knudsen numbers. *Journal of Computational Physics*, 227(19):8655 – 8671, 2008.

[KPB08b] Seung Hyun Kim, Heinz Pitsch et Iain D. Boyd : Slip velocity and knudsen layer in the lattice boltzmann method for microscale flows. *Phys. Rev. E*, 77(2):026704, Feb 2008.

[KZCH02] Qinjun Kang, Dongxiao Zhang, Shiyi Chen et Xiaoyi He : Lattice boltzmann simulation of chemical dissolution in porous media. *Phys. Rev. E*, 65(3):036318, Mar 2002.

[Lan90] L.D. Landau : *Mécanique des fluides*. Mécanique des fluides. Editions Mir, 1990.

[LCM+08] Jonas Latt, Bastien Chopard, Orestis Malaspinas, Michel Deville et Andreas Michler : Straight velocity boundaries in the lattice boltzmann method. *Phys. Rev. E*, 77(5):056703, May 2008.

[LG11] Xiuliang Liu et Zhaoli Guo : A lattice boltzmann study of gas flows in a long micro-channel. *Computers & Mathematics with Applications*, In Press, Corrected Proof:–, 2011.

[LL05] Taehun Lee et Ching-Long Lin : Rarefaction and compressibility effects of the lattice-boltzmann-equation method in a gas microchannel. *Phys. Rev. E*, 71(4):046706, Apr 2005.

[LREB04] Duncan A. Lockerby, Jason M. Reese, David R. Emerson et Robert W. Barber : Velocity boundary condition at solid walls in rarefied gas calculations. *Phys. Rev. E*, 70(1):017303, Jul 2004.

[LY09] Y. Liu et J. Yang : Higher-order lattice boltzmann simulation of energy conversion of electrokinetic nanobatteries. *The European Physical Journal - Special Topics*, 171:99–104, 2009. 10.1140/epjst/e2009-01016-8.

[Mal09] Orestis Pileas Malaspinas : *Lattice Boltzmann method for the simulation of viscoelastic fluid flows*. Thèse de doctorat, EPFL, 2009.

[Mie00] Luc Mieussens : Discrete-velocity models and numerical schemes for the boltzmann-bgk equation in plane and axisymmetric geometries. *Journal of Computational Physics*, 162(2):429–466, 2000.

[MLR06] R.S. Myong, D.A. Lockerby et J.M. Reese : The effect of gaseous slip on microscale heat transfer: An extended graetz problem. *International Journal of Heat and Mass Transfer*, 49(15-16):2502 – 2513, 2006.

[MN69] Kogan Mikhail N. : *Rarefied Gas Dynamics*. Plenum Presse. New York, 1969.

[NDC02] Xiaobo Nie, Gary D. Doolen et Shiyi Chen : Lattice-boltzmann simulations of fluid flows in mems. *Journal of Statistical Physics*, 107: 279–289, 2002. 10.1023/A:1014523007427.

[NHMS07] Xiao-Dong Niu, Shi-Aki Hyodo, Toshihisa Munekata et Kazuhiko Suga : Kinetic lattice boltzmann method for microscale gas flows: Issues on boundary condition, relaxation time, and regularization. *Phys. Rev. E*, 76(3):036711, Sep 2007.

[NSC07] X.D. Niu, C. Shu et Y.T. Chew : A thermal lattice boltzmann model with diffuse scattering boundary condition for micro thermal flows. *Computers & Fluids*, 36(2):273 – 281, 2007.

[NSC08] X. B. Nie, X. Shan et H. Chen : Galilean invariance of lattice boltzmann models. *EPL (Europhysics Letters)*, 81(3):34005, 2008.

[PK08] Nikolaos I. Prasianakis et Iliya V. Karlin : Lattice boltzmann method for simulation of compressible flows on standard lattices. *Phys. Rev. E*, 78(1):016704, Jul 2008.

[Pot07] N. Pottier : *Physique statistique hors d'équilibre: processus irréversibles linéaires*. Savoirs actuels. Série Physique. EDP Sciences, 2007.

[QX97] Zou Qisu et He Xiaoyi : On pressure and velocity flow boundary conditions and bounceback for the lattice boltzmann bgk model. *Phys. Fluids*, 9, 1997.

[RdS02] C. F. Rodrigues et M. J. Lemos de Sousa : The measurement of coal porosity with different gases. *International Journal of Coal Geology*, 48(3-4):245–251, 2002.

[RIDI11] Jean-Louis Rouet, Boujema Izrar et Léonard De Izarra : Mise en évidence numérique de l'effet klinkenberg par une méthode boltzmann-bgk sur réseau. *In 20ème Congrès Français de Mécanique, Besançon, France*, 2011.

[RRC+03] Subrata Roy, Reni Raju, Helen F. Chuang, Brett A. Cruden et M. Meyyappan : Modeling gas flow through microchannels and nanopores. *Phys. Of Fluids*, 93(8):4870–4879, 2003.

[Sch08] Ulf D. Schiller : *Thermal fluctuations and boundary conditions in the lattice Boltzmann method*. Thèse de doctorat, Johannes Gutenberg University Mainz, 2008.

[SH98] Xiaowen Shan et Xiaoyi He : Discretization of the velocity space in the solution of the boltzmann equation. *Phys. Rev. Lett.*, 80(1):65–68, Jan 1998.

[SJ07] E. Shirani et S. Jafari : Application of lbm in simulation of flow in simple micro-geometries and micro porous media. *The African Review of Physics*, 1(1), 2007.

[SS05] Victor Sofonea et Robert F. Sekerka : Diffuse-reflection boundary conditions for a thermal lattice boltzmann model in two dimensions: Evidence of temperature jump and slip velocity in microchannels. *Phys. Rev. E*, 71(6):066709, Jun 2005.

[ST03] Henning Struchtrup et Manuel Torrilhon : Regularization of grad's 13 moment equations: Derivation and linear analysis. *Phys. Of Fluids*, 15(9):2668–2680, 2003.

[ST07] Michael C. Sukop et Daniel T. Jr Thorne : *Lattice Boltzmann Modeling: An Introduction for Geoscientists and Engineers*. Springer Publishing Company, Incorporated, 1 édition, 2007.

[Suc01] S. Succi : *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond (Numerical Mathematics and Scientific Computation)*. Oxford University Press, août 2001.

[SYC06] Xiaowen Shan, Xue-Feng Yuan et Hudong Chen : Kinetic theory representation of hydrodynamics: a way beyond the navier–stokes equation. *Journal of Fluid Mechanics*, 550:413–441, 2006.

[Sza07] L. Szalmás : Knudsen layer theory for high-order lattice boltzmann models. *EPL (Europhysics Letters)*, 80(2):24003, 2007.

[SZG07] Yong Shi, T. S. Zhao et Zhaoli Guo : Lattice boltzmann simulation of dense gas flows in microchannels. *Phys. Rev. E*, 76(1):016707, Jul 2007.

[TI94] Gombosi Tamas I. : *Gaskinetic Theory*. Cambridge atmospheric and space science series. Cambridge University Press, 1994.

[Tor99] Eleuterio F. Toro : *Riemann solvers and numerical methods for fluid dynamics - A practical introduction - 2nd edition*. Springer, Berlin, 1999.

[TS05] Federico Toschi et Sauro Succi : Lattice boltzmann method at finite-knudsen numbers. *Europhys. Lett.*, 69:549, 2005.

[TTH05] G. H. Tang, W. Q. Tao et Y. L. He : Gas slippage effect on microscale porous flow using the lattice boltzmann method. *Phys. Rev. E*, 72(5): 056301, Nov 2005.

[TTS09] Peyman Taheri, Manuel Torrilhon et Henning Struchtrup : Couette and poiseuille microflows: Analytical solutions for regularized 13-moment equations. 21(1):017102, 2009.

[TZE08] Gui-Hua Tang, Yong-Hao Zhang et David R. Emerson : Lattice boltzmann models for nonequilibrium gas flows. *Phys. Rev. E*, 77(4): 046701, Apr 2008.

[TZG+09] G. H. Tang, Y. H. Zhang, X. J. Gu, R. W. Barber et D. R. Emerson : Lattice boltzmann model for thermal transpiration. *Phys. Rev. E*, 79(2):027701, Feb 2009.

[TZL+07] Zhi-Wei Tian, Chun Zou, Hong-Juan Liu, Zhao-Li Guo, Zhao-Hui Liu et Chu-Guang Zheng : Lattice boltzmann scheme for simulating thermal micro-flow. *Physica A: Statistical Mechanics and its Applications*, 385(1):59 – 68, 2007.

[VLB09] Frederik Verhaeghe, Li-Shi Luo et Bart Blanpain : Lattice boltzmann modeling of microchannel flow in slip flow regime. *Journal of Computational Physics*, 228(1):147 – 157, 2009.

[WG00] Dieter A. Wolf-Gladrow : *Lattice-Gas Cellular Automata and lattice Boltzmann Models*. Springer, 2000.

[WPP98] Yu-Shu Wu, Karsten Pruess et peter Persoff : Gas flow in porous media with klinkenberg effects. *Transport in Porous Media*, 32:117–137, 1998. 10.1023/A:1006535211684.

[WT07] G. R. Willmott et J. L. Tallon : Measurement of newtonian fluid slip using a torsional ultrasonic oscillator. *Phys. Rev. E*, 76(6):066306, Dec 2007.

[XH03] Kun Xu et Xiaoyi He : Lattice boltzmann method and gas-kinetic bgk scheme in the low-mach number viscous flow simulations. *Journal of Computational Physics*, 190(1):100 – 117, 2003.

[XHC08] Kun Xu, Xin He et Chunpei Cai : Multiple temperature kinetic model and gas-kinetic method for hypersonic non-equilibrium flow computations. *J. Comput. Phys.*, 227:6779–6794, July 2008.

[XJS01] H. Xue, H. M. Ji et C. Shu : Analysis of micro-couette flow using the burnett equations. *International Journal of Heat and Mass Transfer*, 44(21):4139 – 4146, 2001.

[XL04] Kun Xu et Zhihui Li : Microchannel flow in the slip regime: gas-kinetic bgk–burnett solutions. *Journal of Fluid Mechanics*, 513:87–110, 2004.

[Xu03] Kun Xu : Super-burnett solutions for poiseuille flow. *Phys. Of Fluids*, 15(7):2077–2080, 2003.

[YAK08] Wahyu P. Yudistiawan, Santosh Ansumali et Iliya V. Karlin : Hydrodynamics beyond navier-stokes: The slip flow model. *Phys. Rev. E*, 78(1):016705, Jul 2008.

[ZGA02] Yihao Zheng, Alejandro L. Garcia et Berni J. Alder : Comparison of kinetic theory and hydrodynamics for poiseuille flow. *Journal of Statistical Physics*, 109:495–505, 2002. 10.1023/A:1020498111819.

[ZGBE06] Yong-Hao Zhang, Xiao-Jun Gu, Robert W. Barber et David R. Emerson : Capturing knudsen layer phenomena using a lattice boltzmann model. *Phys. Rev. E*, 74:046704, Oct 2006.

[ZLSE07] W.C. Zhu, J. Liu, J.C. Sheng et D. Elsworth : Analysis of coupled gas flow and deformation process with desorption and klinkenberg effects in coal seams. *International Journal of Rock Mechanics and Mining Sciences*, 44(7):971 – 980, 2007.

[ZQE05] Yonghao Zhang, Rongshan Qin et David R. Emerson : Lattice boltzmann simulation of rarefied gas flows in microchannels. *Phys. Rev. E*, 71(4):047702, Apr 2005.

[ZSC06] Raoyang Zhang, Xiaowen Shan et Hudong Chen : Efficient kinetic method for fluid simulation beyond the navier-stokes equation. *Phys. Rev. E*, 74(4):046703, Oct 2006.

[ZW07] Dongyuan Zhao et Ying Wan : Chapter 8 the synthesis of mesoporous molecular sieves. *In* Avelino Corma Jirí Cejka, Herman van Bekkum et Ferdi Schüth, éditeurs : *Introduction to Zeolite Science and Practice*, volume 168 de *Studies in Surface Science and Catalysis*, pages 241 – 300, I–III. Elsevier, 2007.

Léonard DE IZARRA

Apport des méthodes cinétiques à la simulation d'écoulements dans les milieux poreux

Les méthodes de Boltzmann sur réseaux (LBM) ont été appliquées avec beaucoup de succès aux écoulements hydrodynamiques en milieux poreux. Cependant, la limitation de ces méthodes aux écoulements hydrodynamiques et isothermes, les rendent insuffisantes pour simuler des écoulements de gaz dans des milieux micro-poreux. Dans ce cas, il est en effet fréquent que le libre parcours moyen des molécules du gaz, soit du même ordre de grandeur que la taille des pores dans lesquels il s'écoule. De tels écoulements ne seront alors plus en régime hydrodynamique, mais dans des régimes qualifiés de glissement et de transitionnel ; régimes pour lesquels les LBM standards ne sont plus valides. D'autre part, le caractère isotherme des LBM les rendent inutilisables, par exemple dans le cas où le gaz subit une détente à travers le milieu. Il est nécessaire, pour décrire de tels écoulements et phénomènes, de se placer au niveau cinétique. La démarche proposée repose sur la décomposition de la fonction de distribution sur la base des polynômes d'Hermite et l'emploi de la quadrature de Gauss-Hermite associée à cette projection. L'aspect systématique de ce développement amène naturellement à considérer divers ordres d'approximation de l'équation de Boltzmann-BGK sous diverses quadratures. Il résulte alors de ces différentes approximations toute une famille de discrétisations de l'équation de Boltzmann-BGK, dont les LBM classiques ne sont qu'un membre. La détermination de l'approximation la plus adaptée est réalisée par analyse systématique des résultats obtenus aux différents ordres d'approximation. Ces méthodes sont testées avec succès dans des cas modèles.

Mots clés : Milieux poreux, Boltzmann, BGK, LBM, régime transitionnel, théorie cinétique

Contribution of kinetic methods for the simulation of flows in porous media

The lattice Boltzmann method (LBM) have been applied very successfully to hydrodynamic flows in porous media. However, the limitation of these methods to isothermal and hydrodynamic flows, make them inadequate to simulate gas flows in micro-porous media. Indeed, in these conditions, the mean free path of the molecules could be of the same magnitude order as the pore size in which gas flows. Such flows will not be in hydrodynamic regime, but in regimes qualified of, slip or transitional ; for which the LBM are no longer valid. On the other hand, the isothermal character of LBM make them unusable, for example, in the case where the gas undergoes expansion through the media. It is then necessary, to take the kinetic point of view to describe such flows and phenomena. The proposed approach is based on the decomposition of the distribution function on the Hermite polynomials basis and the use of Gauss-Hermite quadrature associated with this projection. The systematic nature of this development naturally leads to consider different order of approximation of the Boltzmann-BGK equation in various quadratures. It then follows from these various approximations, a family of discretizations of the Boltzmann-BGK equation, whose classical LBM are a member. Determining the most suitable approximation is achieved by systematic analysis of the results obtained with different approximation orders. These methods are successfully tested in model cases.

Keywords : Porous media, Boltzmann, BGK, LBM, transitional regime, kinetic theory

Institut des Sciences de la Terre d'Orléans / Institut de Combustion Aérothermique Réactivité et Environnement