

Méthode de Monte-Carlo et non-linéarités : de la physique du transfert radiatif à la cinétique des gaz

Guillaume Terrée

► To cite this version:

Guillaume Terrée. Méthode de Monte-Carlo et non-linéarités : de la physique du transfert radiatif à la cinétique des gaz. Génie des procédés. Ecole des Mines d'Albi-Carmaux, 2015. Français. <NNT : 2015EMAC0017>. <tel-01411882>

HAL Id: tel-01411882

<https://tel.archives-ouvertes.fr/tel-01411882>

Submitted on 7 Dec 2016

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 Fédérale



Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

*l'École Nationale Supérieure des Mines d'Albi-Carmaux
conjointement avec l'INP Toulouse*

Présentée et soutenue le 13 Octobre 2015 par :

GUILLAUME TERRÉE

**Méthode de Monte-Carlo et non-linéarités :
de la physique du transfert radiatif à la cinétique des gaz**

JURY

MOUNA EL HAFI	Maître assistante, École des Mines d'Albi	Directrice de thèse
PIERRE MARÉCHAL	Professeur, Université Paul Sabatier (Toulouse)	Co-directeur de thèse
RENÉE GATIGNOL	Professeure émérite, Université Pierre et Marie Curie (Paris 6)	Rapporteuse
DAVID LACROIX	Professeur des universités, Université de Lorraine (Nancy)	Rapporteur
OLIVIER SIMONIN	Professeur des universités, Institut National Polytechnique de Toulouse	Président du jury
VICTOR SOFONEA	Directeur de recherche, Academia Română (Timișoara)	Examineur
DAVID GUÉRY-ODELIN	Professeur, Université Paul Sabatier (Toulouse)	Examineur
LUC MIEUSSENS	Professeur des universités, Université de Bordeaux	Examineur
RICHARD FOURNIER	Professeur, Université Paul Sabatier (Toulouse)	Invité

École doctorale et spécialité :

MEGEP : Énergétique et transferts

Unité de Recherche :

Centre RAPSODEE, CNRS - UMR 5302 de l'École des Mines d'Albi-Carmaux

Directeurs de Thèse :

Mouna EL HAFI et Pierre MARÉCHAL

Rapporteurs :

Renée GATIGNOL et David LACROIX

Remerciements

Remerciements institutionnels

●●●● Le déroulement et la qualité d'un travail de thèse dépendent essentiellement du doctorant, et plus encore de son encadrement. À ce titre, je voudrait remercier mes encadrants en premier lieu.

J'adresse un immense merci à ma directrice de thèse, Mouna EL HAFI. Merci de m'avoir proposé ce doctorat, avant même que j'y aie pensé, et d'avoir remué l'administration régionale pour me décrocher un contrat. Merci pour ta présence toutes les semaines, venant sans cesse rappeler le droit chemin. Merci pour ton intelligence organisationnelle, qui a trop souvent rattrapé mes manquements à ce niveau. Merci pour ton inébranlable confiance ; et merci pour ton inépuisable patience, que je sais avoir largement usée durant ce qui fut une très, très longue rédaction.

Merci aussi à Stéphane BLANCO et Richard FOURNIER, mes encadrants toulousains (non-officiels). Votre connaissance quasi-extensive de la physique du transport m'a été essentielle. Merci pour votre direction et vos conseils avisés, amenant inmanquablement à d'intéressants résultats. Merci aussi pour votre expérience de la communication, si utile quand il faut rédiger des introductions.

Merci à mon co-directeur de thèse, Pierre MARÉCHAL. La bifurcation à mi-parcours de mon sujet de recherche t'a laissé de côté ; merci de l'intérêt que tu as néanmoins porté à mon travail, jusqu'à la fin. Au fait, malgré mon changement de sujet, tes cours de maths m'ont été bien utiles.

Merci à Olivier FUDYM, encadrant albigeois des premières heures, pour ton aide fournie en inversion paramétrique. Bon plaisir au Brésil !

Merci à David GUÉRY-ODELIN, en particulier pour l'exemple du chapitre 6 que tu nous as fourni. J'espère que mon travail pourra t'être utile.

Merci enfin à Victor SOFONEA, notre collaborateur en Roumanie ; en particulier pour ma visite à Timișoara, qui fut des plus agréables. Merci aussi d'être venu jusqu'à Albi pour assister à ma soutenance. J'espère qu'à toi aussi, ce travail pourra être utile.

●●●● Je remercie aussi tous les extérieurs qui ont bien voulu participer à mon jury de thèse.

Merci d’abord à Olivier SIMONIN, qui a accepté de présider mon jury, et ce avec une très bonne humeur. Merci pour vos remarques très constructives au sujet de la comparaison entre mon travail et la méthode DSMC.

Merci ensuite et surtout, à Renée GATIGNOL et David LACROIX, d’avoir accepté de rapporter mon volumineux manuscrit ; et ce, pendant vos vacances d’août qui ont été, en partie de ce fait, très studieuses.

Merci enfin à Luc MIEUSSENS d’être venu depuis Bordeaux assister à ma soutenance (même s’il n’est pas celui qui a fait le plus de kilomètres!).

Remerciements personnels

Je voudrais d’abord remercier mes collègues de toujours, Mathieu G. et Christophe S., avec qui j’ai fait mes études d’ingénieur, et avec qui j’ai poursuivi en thèse, dans la magnifique ville d’Albi. Sans vous, ça n’aurait pas eu la même saveur. Bon vent à vous, et à la prochaine !

Merci aux grands briscards de l’équipe, Vincent E., Benjamin P., Germain B., Jérémie D., Olivier F., Anaïs K.¹, Laetitia L., Sebastian W., Sébastien D., Sara A.C., Cyril C., et dans un genre différent Jean-Jacques B., Christophe C., et Jacques G. — j’espère que je n’oublie personne. Vous étiez et êtes encore, de fort bénéfiques éclaireurs sur mon chemin.

Bonne chance à tous les petits nouveaux ou moins nouveaux : Victor P., Julien C., Hugo U., Manon A., Alexandra D.C., Vincent F. . Je suis fier de vous servir d’exemple (ou pas. . .). Encore une fois bonne chance, et bon courage !

Merci enfin à tous les collègues de l’École des Mines : Nelly, Clément, Raphaël, Nicolas, Marion, Marta, Hélène, Baptiste, Abdou, Marwa, Rabab, Augustina, Lat’grand, Javier, Hugo, Briec, Lucia, et j’en oublie certainement. C’était un plaisir de vous voir chaque jour.

1. Au fait, j’attends toujours ton manuel de bonnes manières — avec un peu d’appréhension je l’avoue. J’ai pourtant essayé d’apprendre sans toi, mais les conseils de mes grands-mères sur comment-se-tenir-comme-il-faut-à-table me ressortent inévitablement par les trous de nez.

Table des matières

Remerciements	iii
Remerciements institutionnels	iii
Remerciements personnels	iv
Table des matières	vii
Nomenclature	xiii
Introduction	1
Contexte scientifique	1
Problématique	5
Quelques illustrations	11
Organisation du manuscrit	16
1 Introduction technique	19
1.1 Équations de Boltzmann	19
1.1.1 Fonctions de distribution	20
1.1.1.1 Fonctions de distribution, et approche mésoscopique	20
1.1.1.2 Fonction de distribution des molécules	21
1.1.1.3 Fonction de distribution des photons, et luminance	22
1.1.2 Fonction de distribution et observables macroscopiques . . .	24
1.1.3 Hypothèses, établissement, et conséquences de l'équation de	
Boltzmann	28
1.1.3.1 Hypothèses sur les molécules	28
1.1.3.2 Obtention de l'équation de Boltzmann	29
1.1.3.3 Distribution d'équilibre des molécules	32
1.1.4 Équation de transfert radiatif	34
1.2 Résolution des équations de Boltzmann le long d'un chemin balistique	37
1.2.1 Forme intégrale de l'équation de transfert radiatif	37
1.2.2 Forme intégrale de l'équation de Boltzmann	38
1.2.3 Conditions aux frontières	40
1.2.3.1 En transfert radiatif	40
1.2.3.2 En cinétique des gaz	42
1.3 La méthode de Monte-Carlo	42

1.3.1	Les bases de la méthode	43
1.3.2	Approche analogue et approche intégrale, et concept de variance nulle	48
1.3.2.1	Calcul de l'aire d'un étang	50
1.3.2.2	Calcul d'une transmittivité dans une colonne de gaz uniforme	51
1.3.3	Événements rares, et indicateurs statistiques avancés	55
1.4	(*) Suivi direct et suivi inverse des photons : un lien par l'adjonction	58
1.4.1	Principe	59
1.4.2	Remarques	62
1.5	(*) Diffusion multiple	63
1.5.1	Description du problème	63
1.5.2	Formulation intégrale, et notations	63
1.5.3	Choix des variables aléatoires	64
1.5.4	Modification du problème : ajout d'absorption	67
1.5.5	Calcul de sensibilités	70
2	Algorithmes à Collisions Nulles	73
2.1	Les ACNs comme technique d'échantillonnage des libres parcours .	74
2.1.1	Bref historique	74
2.1.2	La difficulté que l'on veut surmonter	75
2.1.3	Utilisation de collisionneurs fictifs	79
2.2	Fondements mathématiques	83
2.2.1	Validité de la formulation intégrale	84
2.2.2	Transformation en expression statistique	85
2.2.3	Partitionnement de l'énergie <i>vs.</i> roulette russe	86
2.2.4	ACNs et calcul de sensibilités	89
2.3	(*) Mise en œuvre pratique des alternatives algorithmiques	90
2.3.1	Dans le cas d'étude du chapitre (absorption/émission seule)	90
2.3.2	Avec diffusion	96
2.4	(*) Exemple académique	97
2.4.1	Introduction	97
2.4.2	Présentation du cas d'étude	98
2.4.3	Détails techniques sur le calcul effectué	98
2.4.4	Résultats et commentaires	102
2.5	ACNs et estimation de l'extinction	113
2.5.1	Intérêt (pour nous) de cette démarche	113
2.5.2	Situation du travail de Mathieu GALTIER	114
2.5.3	Problématique spectrale des milieux gazeux	114
2.5.3.1	Complexité du coefficient d'absorption	114
2.5.3.2	Conséquences de cette complexité en calcul numé- rique du transfert radiatif	117
2.5.4	Estimation de l'extinction dans un ACN	119
2.6	Les algorithmes à collisions nulles dans la suite de ce manuscrit .	121

3	Estimation d'une fonction non-linéaire d'une espérance	123
3.1	Introduction	123
3.1.1	Grandeurs dérivées de la fonction de distribution : le cas des photobioréacteurs	124
3.1.2	Problématique	127
3.2	Présentation de la technique utilisée	131
3.2.1	Cas d'un monôme ou d'un polynôme	131
3.2.2	Cas des fonctions analytiques	133
3.2.3	Parenthèse sur les sensibilités	137
3.3	(*) Calcul de la productivité annuelle d'une centrale solaire à concentration	139
3.3.1	Contexte	139
3.3.2	Démarche	142
3.3.3	Résultats	145
3.4	(*) Calcul de la productivité d'un photobioréacteur	147
3.4.1	Problématique	147
3.4.2	Modèle	148
3.4.3	Traitement mathématique	150
4	Méthode de Monte-Carlo et équadiffs non-linéaires	155
4.1	Introduction	155
4.2	Bases mathématiques de la méthode	157
4.3	Illustration sur un cas de cinétique chimique	163
4.3.1	Le cas d'étude	163
4.3.2	Premier algorithme de résolution proposé	164
4.3.3	Comportement de l'algorithme de résolution proposé : étude par modélisation de la récursivité et de la variance	166
4.3.4	Second algorithme de résolution proposé	171
4.3.5	Expériences numériques	174
4.4	Utilisation d'une solution approximative calculée <i>a priori</i>	180
4.5	Calcul de sensibilités paramétriques	181
4.6	Illustration sur un cas de cinétique chimique : suite et fin	183
4.6.1	Sensibilités paramétriques	183
4.6.1.1	Mise en place des calculs	183
4.6.1.2	Résultats des calculs	185
4.6.2	Utilisation d'une solution approximative	186
4.6.2.1	Étude théorique sur le cas choisi	186
4.6.2.2	Derniers choix algorithmiques	192
4.6.2.3	Résultats, et commentaires	193
5	Résultats obtenus concernant le mode BKW	199
5.1	Quelques explications sur le mode BKW (Bobylev-Krook-Wu)	200
5.1.1	Revue bibliographique partielle	200
5.1.2	Description du mode BKW	201

5.2	Développement d'un algorithme de Monte-Carlo simple pour le mode BKW	202
5.2.1	Formulation intégrale	204
5.2.2	Un choix pour $p_{T'}$, $p_{\vec{C}_*}$ et $p_{\vec{U}'}$	205
5.3	Premiers résultats obtenus	207
5.3.1	Campagne de tests, et résultats	207
5.3.2	Analyse des résultats	209
5.4	Une astuce algorithmique pour accroître les capacités de la méthode développée	214
5.5	Deuxième groupe de résultats obtenus	215
6	Résultats obtenus sur un nuage de gaz oscillant	221
6.1	Description de la situation physique, dite « de confinement harmo- nique »	222
6.1.1	Cas général	222
6.1.2	Nos choix	223
6.2	Solution intégrale partielle de l'équation de Boltzmann, et applica- tion à la situation de confinement harmonique	226
6.2.1	Écritures intégrale et statistique de l'équation de Boltzmann, basée sur les Algorithmes à Collisions Nulles	226
6.2.2	Transformations possibles de l'expression statistique	228
6.2.3	Choix algorithmiques	230
6.3	Résultats obtenus	238
	Perspectives, et conclusion	249
1	Calcul en régime permanent	251
2	Gestion des géométries complexes	252
2.1	Exemple	253
3	Perspectives formelles sur l'opérateur de collision de Boltzmann	255
3.1	Variation des sections efficaces	255
3.2	Combinaison du terme source et de la fréquence de collision	257
3.3	Utilisation d'information connue <i>a priori</i>	259
A	Rappels mathématiques	261
A.1	Développements en Série Entière	261
A.2	Adjonction Hermitienne	263
A.3	Fonction d'erreur complémentaire normalisée erfcx	265
B	Lien entre les ACNs et les DSEVIs	267
C	Polynôme d'une espérance	273
C.1	Calcul en ligne de la formulation (3.9)	273
C.2	Intérêt d'estimer un polynôme de l'espérance d'une V.A. grâce à un échantillon contenant plus de tirages de la V.A. que le degré du polynôme : exemple du calcul de la variance	275

D Fichiers sources	277
D.1 Chambre de combustion contenant une flamme axisymétrique (problème académique de la section 2.4) : calcul du terme source radiatif, avec gestion des hétérogénéités des propriétés optiques par un Algorithme à Collisions Nulles	278
D.2 Résolution d'un problème instationnaire simplifié de cinétique chimique d'ordre 2	291
D.2.1 Sans approximation de guidage y^\approx	291
D.2.2 Avec approximation de guidage y^\approx	297
D.3 Calcul de $f(\vec{c}; t)$ dans la situation du mode BKW	303
D.4 Calcul de $f(\vec{r}; \vec{c}; t)$ dans la situation de confinement harmonique . .	314
Bibliographie	335

Nomenclature

Les tables qui suivent listent les notations de ce manuscrit, d'abord en ce qui concerne les acronymes, ensuite les notions mathématiques usuelles (produit scalaire, ensembles, . . .), et enfin toutes les grandeurs que nous allons utiliser.

Avant de commencer, précisons qu'autant que possible :

- les vecteurs sont surmontés d'une flèche, comme \vec{a} ,
- les variables aléatoires sont désignées par des lettres capitales, comme A ,
- un estimateur d'une grandeur a est noté en majuscule (car c'est une variable aléatoire) et surmonté d'un tilde, ce qui donne \tilde{A} .

Acronymes :

Acronyme	Signification	1 ^{re} apparition
MMC	méthode de Monte-Carlo	page 5
DSMC	Direct Simulation Monte-Carlo	page 4
BsR	Boltzmann sur Réseau	page 4
VA	variable aléatoire	page 44
ACN	algorithme à collisions nulles	page 73
PNE	puissance nette échangée	page 98
DSE	développement en série entière	page 134
DSEVI	développement en série entière de variables indépendantes	page 135
mode BKW	mode « Bobylev-Krook-Wu »	page 199
méthode RK4	méthode Runge-Kutta d'ordre 4	page 192

Notations : pages suivantes

Nomenclature : pages suivantes

Notation	Signification
$\mathcal{F}(E; F)$	Ensemble des fonctions de l'ensemble E vers l'ensemble F
$\mathcal{L}(E; F)$	Ensemble des fonctions linéaires de l'ensemble E vers l'ensemble F
$\mathcal{B}(E; \vec{M}; R)$	Boule ouverte de l'ensemble E , de centre \vec{M} et de rayon $R : \mathcal{B}(E; \vec{M}; R) = \{\vec{x} \in E, \ \vec{x} - \vec{M}\ < R\}$
$\mathcal{B}_f(E; \vec{M}; R)$	Boule fermée de l'ensemble E , de centre \vec{M} et de rayon $R : \mathcal{B}_f(E; \vec{M}; R) = \{\vec{x} \in E, \ \vec{x} - \vec{M}\ \leq R\}$
$\mathcal{S}(E; \vec{M}; R)$	Sphère de l'ensemble E , de centre \vec{M} et de rayon $R : \mathcal{S}(E; \vec{M}; R) = \{\vec{x} \in E, \ \vec{x} - \vec{M}\ = R\}$
$E \setminus F$	Complémentaire de l'ensemble F dans l'ensemble E
$E \setminus_{\perp} F$	Supplémentaire orthogonal du sous-espace F dans l'espace E
$\mathcal{P}(E)$	Ensemble des parties de l'ensemble E
$\mathcal{E}(E)$	enveloppe convexe de l'ensemble E
\mathcal{D}_f	Ensemble de définition de la fonction f
$\llbracket n; m \rrbracket$	Ensemble des entiers de n à m . L'orientation des crochets de chaque côté a la même signification que pour les intervalles de \mathbb{R} .
$\Pr(A)$	Probabilité de l'événement A
$\Pr(A \mid B)$	Probabilité de l'événement A , sachant l'événement B
$p_{\vec{X}}$	Fonction de densité de probabilité de la variable aléatoire \vec{X}
F_X	Fonction de répartition de la variable aléatoire X à valeurs réelles : $\forall x \in \mathbb{R}, F_X(x) = \Pr(X \leq x)$
$\mathcal{U}_{\vec{X}}$	Univers de la variable aléatoire \vec{X} : c'est-à-dire l'ensemble de ses valeurs possibles
$E(\vec{X})$	Espérance de la variable aléatoire \vec{X}
$\text{Var}(X)$	Variance de la variable aléatoire X à valeurs réelles
$f(\bullet; y)$	Si $f \in \mathcal{F}(E_1 \times E_2; F)$, alors pour tout $y \in E_2$, $f(\bullet; y) \in \mathcal{F}(E_1; F)$ et $f(\bullet; y) : x \mapsto f(x; y)$.
$H(\text{cond})$	Fonction de Heaviside généralisée. $H(\text{cond}) = 1$ si la condition <i>cond</i> est vraie, et $H(\text{cond}) = 0$ si <i>cond</i> est fausse.
$\delta(\bullet)$	Distribution de Dirac

Notation	Signification
$\vec{a} \cdot \vec{b}$	Produit scalaire des vecteurs \vec{a} et \vec{b} . \vec{a} et \vec{b} sont des vecteurs de l'espace ordinaire.
$\langle a; b \rangle$	Produit scalaire des vecteurs a et b , qui appartiennent à un espace quelconque
$\vec{a} \otimes \vec{b}$	Produit dyadique des vecteurs \vec{a} et \vec{b} de l'espace ordinaire. Si dans une base orthonormale donnée, $\vec{a} = [a_i]$ et $\vec{b} = [b_j]$, alors $\vec{a} \otimes \vec{b} = [a_i b_j]$. On a en particulier : $\vec{a} \cdot \vec{b} = \text{Tr}(\vec{a} \otimes \vec{b})$.
\vec{a}^2	Produit scalaire du vecteur \vec{a} avec lui-même
$\partial_1 f$	Fonction dérivée selon la 1 ^{re} variable de la fonction f
$\partial_x A$	Dérivée de l'expression A selon le paramètre x
$\partial_{x \perp y} A$	Dérivée de l'expression A selon le paramètre x , au paramètre y constant
$\partial_{xy} A$	Dérivée seconde de l'expression A , selon les paramètres x et y
	Exemple : Soit $f : (x; y) \mapsto x^2 + y^2$, alors :
	$\partial_1 f(x; x + z) = 2x$ $\partial_x f(x; x + z) = \partial_{x \perp z} f(x; x + z) = 4x + 2z$ $\partial_{x \perp (x-z)} f(x; 2x - (x - z)) = 10x - 4(x - z) = 6x + 4z$ $\partial_{xx} f(x; x + z) = 4$
$\overrightarrow{\text{grad}}_{\vec{x}} A$	Gradient de l'expression scalaire A selon le paramètre vectoriel \vec{x}
$\overrightarrow{\text{grad}}_1 f$	Fonction gradient de la fonction à valeurs scalaires f selon sa 1 ^{re} variable (vectorielle)
$\text{div}_{\vec{x}} \vec{A}$	Divergence de l'expression vectorielle \vec{A} selon le paramètre vectoriel \vec{x}
$\text{div}_1 \vec{f}$	Fonction divergence de la fonction à valeurs vectorielles \vec{f} selon sa 1 ^{re} variable (vectorielle)

Symbole	Signification	Portée
\vec{r}	Vecteur position	Manuscrit entier
\vec{c}	Vecteur vitesse	Chapitres 1, 5, et 6
\vec{u}	Direction	Manuscrit entier
ν	Fréquence (d'un photon)	Chapitres 1, 2, et 3
t	Variable temps, instant	Manuscrit entier
E_r	Espace des positions (espace géométrique ordinaire)	Manuscrit entier
E_c	Espace des vitesses	Manuscrit entier
E_u	Espace des directions	Manuscrit entier
E_ν	Échelle spectrale (en transfert radiatif)	Chapitres 1 et 2
E_p	Espace des phases	Manuscrit entier
E_t	Échelle des temps	Manuscrit entier
f	Fonction de distribution des molécules, ou bien fonction de distribution des photons quand les propriétés spectrales n'interviennent pas	Chapitres 1, 2, 5, et 6
f_ν	Fonction de distribution des photons, avec différentielle spectrale en fréquence	Chapitres 1, 2, et 3
L_ν	Luminance, avec différentielle spectrale en fréquence	Chapitres 1 et 3
L	Luminance intégrée spectralement	Chapitres 1 et 3
g	Captance	Chapitre 1
M	Masse molaire	Chapitres 1 et 6
η	Concentration molaire	Manuscrit entier
\vec{j}_{mol}	Densité de débit molaire	Chapitre 1
\vec{v}	Vitesse moyenne locale du fluide	Chapitres 1 et 6
c_{qma}	Vitesse quadratique moyenne sur chaque axe d'agitation thermique	Chapitres 1, 5, et 6
S_r	terme source radiatif	Chapitres 1 et 2

Symbole	Signification	Portée
\vec{F}	Force d'origine extérieure	Chapitres 1 et 6
\vec{a}	Accélération due à une force d'origine extérieure	Chapitres 1 et 6
C_B	Opérateur de collision de Boltzmann	Chapitres 1, 5, et 6
σ_T	Section efficace totale de collision inter-moléculaire	Chapitre 1
σ_F	Section efficace différentielle de collision inter-moléculaire	Chapitres 1, 5, et 6
b_T	Facteur total de collision inter-moléculaire	Chapitre 1
b_F	Facteur différentiel de collision inter-moléculaire	Chapitres 1, 5, et 6
\vec{c}_*	Vitesse d'une partenaire de collision	Chapitres 1, 5, et 6
\vec{c}'	Vitesse avant collision d'une particule suivie	Chapitres 1, 5, et 6
\vec{c}'_*	Vitesse avant collision d'une partenaire de collision	Chapitres 1, 5, et 6
\vec{u}'	Direction d'arrivée d'une particule suivie dans une collision, dans le référentiel barycentrique de collision	Chapitres 1, 5, et 6
\vec{r}_b et \vec{c}_b	Position et vitesse, en tant que fonctions du temps, le long d'une trajectoire balistique	Chapitres 1 et 6
C_T	Opérateur de collision de l'équation de transfert radiatif	Chapitres 1 et 2
k_a	Coefficient d'absorption	Chapitres 1, 2, et 3
k_d	Coefficient de diffusion	Chapitres 1, et 2
$k_{d,F}$	Coefficient angulaire de diffusion	Chapitre 1
Φ	Fonction de phase de diffusion	Chapitres 1, et 2
k_t	Coefficient d'extinction	Chapitres 1 et 2
k_n	Coefficient de collision nulle	Chapitre 2
\hat{k}_a	Coefficient d'absorption majoré	Chapitre 2
\hat{k}_t	Coefficient d'extinction majoré	Chapitre 2
ε	Émissivité d'une surface	Chapitres 1 et 2
\vec{u}'	Direction d'un photon suivi, avant diffusion ou réflexion	Chapitres 1 et 2
s, s', s_{cur}	Abscisse curviligne le long d'un rayon	Chapitres 1 et 2

Symbole	Signification	Portée
π	total de pondération associé aux collisions restantes, lors de la construction d'un chemin optique pour estimer une grandeur radiative	Chapitres 1 et 2
Π	<i>idem</i> , considéré comme une variable aléatoire	Chapitres 1 et 2
cp_i	pondération associée à la i^e collision, lors de la construction d'un chemin optique pour estimer une grandeur radiative	Chapitres 1 et 2
Cp_i	<i>idem</i> , considérée comme une variable aléatoire	Chapitres 1 et 2
wp_i	valeur associée à la i^e collision, lors de la construction d'un chemin optique pour estimer une grandeur radiative	Chapitres 1 et 2
Wp_i	<i>idem</i> , considérée comme une variable aléatoire	Chapitres 1 et 2
\vec{x} et \vec{y}	Variables d'intégrations quelconques	Chapitres 1 et 3
\vec{X} et \vec{Y}	Variables aléatoires quelconques	Chapitres 1 et 3
z	Variable d'intégration quelconque, à valeurs réelles	Chapitres 3 et 4
Z	Variable aléatoire quelconque, à valeurs réelles	Chapitres 3 et 4
R	Variable aléatoire de loi uniforme standard	Manuscrit entier
\tilde{R}	Variable aléatoire position	Chapitres 1, 2, 3, et 6
\tilde{U}	Variable aléatoire direction	Manuscrit entier
$\vec{\gamma}$	Chemin optique	Chapitres 1 et 3
$\vec{\Gamma}$	Variable aléatoire chemin optique	Chapitres 1 et 3
Col	Variable aléatoire événement de collision	Chapitres 1, 2, et 6
\tilde{F}	Estimateur de la fonction de distribution f	Chapitres 1, 2, 5, et 6
\tilde{G}	Estimateur de la captance g	Chapitre 1

Symbole	Signification	Portée
f et g	fonctions quelconques à valeurs réelles	Chapitres 3 et 4
σ	permutation des entiers	Chapitres 1 et 3
\mathfrak{R}	ensemble pondéré de permutations des entiers	Chapitre 3
\mathfrak{S}_n	ensemble des permutations des entiers de 1 à n	Chapitres 1 et 3
λ	coefficient d'un développement en série entière	Chapitre 3
y	fonction solution d'un système différentiel ordinaire scalaire d'ordre 1	Chapitre 4
\tilde{Y}	estimateur de y	Chapitre 4

Introduction

Contexte scientifique

●●●● Le champ physico-mathématique dans lequel ce travail de thèse s'inscrit est celui des *modélisations mésoscopiques des phénomènes de transport*. Grossièrement, ces termes signifient qu'on a un phénomène physique conceptualisé comme issu des déplacements et des interactions désordonnés de très nombreuses particules, et qu'on le modélise mathématiquement via une fonction de distribution signifiant la distribution de ces particules dans l'espace de leurs possibles états (rassemblant leur position, leur vitesse, leur énergie interne, ...) (on appelle cet espace *l'espace des phases*).

Ces modélisations sont utiles quand celles basées sur les grandeurs macroscopiques locales usuelles (densité, vitesse moyenne, température, composition chimique, ...) ne suffisent plus, ou qu'il faut les justifier. Leurs applications sont nombreuses :

- la lecture atomistique de la thermodynamique dans les gaz — historiquement la 1^{re} modélisation de ce type, lors des travaux de Ludwig BOLTZMANN,
- l'interprétation atomistique de la mécanique des fluides, qui fournit des modèles de dépendance des propriétés des fluides (viscosité, diffusivité thermique, ...) aux conditions thermodynamiques. Cela a été fait dans les gaz parfaits, et aussi dans les gaz denses et les liquides [25].
- le calcul des taux et des vitesses de réaction nucléaire, dans les réacteurs à fission ou dans les bombes, qui sont directement liés au flux de neutrons. C'est ce domaine disciplinaire qui a développé et rendu populaire la méthode de Monte-Carlo, dont il est question dans ce manuscrit [38, 81, 112, 113].
- le calcul du transfert radiatif de toute nature, et dans toute sorte d'applications : la protection radiologique, le génie des procédés combustifs [33, 62, 90, 95], le génie des procédés solaires [7, 29, 31, 44], le calcul des flux thermiques atmosphériques (indispensable à la météorologie et à la climatologie) [42], l'imagerie médicale [15, 87, 104], ... De façon moins directe, la synthèse d'images photo-réalistes [92] en fait aussi partie.
- l'astrophysique ; d'une part par le transfert radiatif, nécessaire à la compréhension détaillée de l'émission lumineuse des étoiles ou des nuages interstellaires

de gaz et de poussières ; mais d'autre part par la dynamique des amas d'étoiles et des galaxies (les étoiles sont alors les particules),

- la cinétique des gaz raréfiés. Celle-ci va du dimensionnement des protections thermiques pour les rentrées atmosphériques, équipant les engins spatiaux, jusqu'à toute la micro-fluidique [3, 93, 110, 111].
- la compréhension des comportements collectifs, en biologie [23, 60, 69],
- la physique des plasmas [14, 19, 79, 123]. Ses applications sont par exemple la fusion thermonucléaire contrôlée, le four à micro-ondes, ou le traitement de surfaces.
- la physique des solides, et en particulier la physique des semi-conducteurs [10, 88, 97, 106]. Les particules considérées sont alors des phonons ou des charges électriques libres.
- ...

●●●● Nous nous restreignons dans cette thèse à ce qui obéit à l'équation de Boltzmann ; et ne serait-ce que parmi les applications précédemment listées cela fait déjà beaucoup. En fait l'équation de Boltzmann au sens large, c'est-à-dire par exemple linéaire ou non-linéaire, est constituante du sujet de cette étude.

Cette équation s'écrit de façon générale :

$$D(f) = C(f)$$

où :

- f est la fonction de distribution,
- D est la dérivation particulaire, appelée aussi dérivation en suivant le mouvement. Par exemple, si les particules sont des molécules identiques de masse m dont l'état est complètement décrit par une position \vec{r} et une vitesse \vec{c} , alors :

$$\forall(\vec{r}; \vec{c}; t), D(f)(\vec{r}; \vec{c}; t) = \partial_t f(\vec{r}; \vec{c}; t) + \vec{c} \cdot \overrightarrow{\text{grad}}_{\vec{r}} f(\vec{r}; \vec{c}; t) + \text{div}_{\vec{c}} \left(f(\vec{r}; \vec{c}; t) \frac{\vec{F}}{m}(\vec{r}; \vec{c}; t) \right)$$

où \vec{F} est la somme des forces issues de l'extérieur du système et s'appliquant aux particules.

- $C(f)$ est le *terme collisionnel*. Dans les équations de Boltzmann, ce terme est supposé :
 - local en position et en temps. Ceci signifie que la valeur de $C(f)$ en un point ne dépend que des valeurs de f en la même position \vec{r} et au même instant t . Les équations de Boltzmann s'opposent en cela aux équations de Vlasov, où on tient compte d'interactions à longue portée entre les particules.

- directement intégral sur $E_p \setminus_{\perp} E_r$ (les dimensions de l'espace des phases autres que la position). Cela veut dire que la variation de la densité de particules suivant une trajectoire balistique est due à la sortie et l'entrée de particules dans cette trajectoire, et que ces entrées/sorties se font par sauts instantanés dans l'espace des phases. Ces sauts sont appelés *collisions*. Les opérateurs de collision de type Boltzmann se distinguent en cela des opérateurs de collision de Landau-Fokker-Planck, modélisant des situations où les collisions sont rasantes, et qui font apparaître un Laplacien dans $E_p \setminus_{\perp} E_r$.

La linéarité d'une équation de Boltzmann dépend de la linéarité de son terme de collision. En transfert radiatif ou en neutronique par exemple, le transport est linéaire dans la plupart des cas ; alors $C(f)$ s'exprime comme une intégrale de f sur $E_p \setminus_{\perp} E_r$. En cinétique des gaz le transport est non-linéaire : en l'occurrence $C(f)$ est quadratique selon f , et s'exprime comme une intégrale sur $E_p \setminus_{\perp} E_r$ de $f \times f$.

●●●● Notre sujet d'étude, en lui-même, est d'essayer d'étendre aux transports non-linéaires des interprétations statistiques construites pour les transports linéaires. Les applications que nous avons en tête lors de ce travail étaient essentiellement fluidiques, si bien qu'il a été orienté, ainsi que ce manuscrit, vers des perspectives fluidiques ; cependant, nous sommes convaincus que cet écrit peut avoir un intérêt pour des lecteurs issus d'autres champs disciplinaires.

Par ailleurs, l'exemple de transport linéaire, que nous connaissons suite à de précédents travaux dans l'équipe où cette thèse a été encadrée, et qui nous sert de point de départ, est le transfert radiatif.

Ainsi, au cours de ce manuscrit nous parlerons systématiquement de transfert radiatif ou de cinétique des gaz : le premier nous servira de représentant des transports linéaires, et la deuxième sera notre exemple de transport non-linéaire.

●●●● Même si certains concepts construits dans ce manuscrit ont possiblement une valeur intuitive propre (nous parlerons notamment de propagation dans le monde du non-linéaire, où c'est réputé inopérant), notre objectif est entièrement contenu dans le montage de solutions numériques à l'équation de Boltzmann, en utilisant une approche statistique.

Cela nous amène à présenter maintenant les méthodes actuellement disponibles pour résoudre les équations de Boltzmann, en transfert radiatif et en cinétique des gaz. Souvent, les méthodes se correspondent strictement entre les deux domaines, même quand le vocabulaire diffère. Au fait, les équations d'évolution dans ces domaines sont appelées respectivement « équation de transfert radiatif » et « équation de Boltzmann » (sous-entendu la seule, celle d'origine).

Globalement, les méthodes de résolution se divisent en 2 catégories : les méthodes à discrétisation et/ou à quadratures fixes, et les méthodes statistiques. La façon dont ces méthodes se précisent dépend aussi du caractère linéaire ou non du transport.

●●● Les méthodes à discrétisation se distinguent essentiellement par les quadratures qu’elles utilisent dans l’espace des phases, et leur schéma de mise à jour (dans l’échelle des temps).

Dans l’espace ordinaire, un maillage usuel de type volumes finis ou éléments finis, régulier ou non, peut être utilisé. Les méthodes se distinguent plus souvent par la quadrature utilisée dans l’espace supplémentaire à l’espace usuel — espace des vitesses en cinétique des gaz, espace des directions en transfert radiatif. On peut utiliser par exemple des harmoniques sphériques couplés à des moments de la norme — en transfert radiatif la norme n’apparaît pas, car elle est constante. Dans le même esprit, les harmoniques sphériques peuvent être remplacées par des ordonnées discrètes. Dans un espace des vitesses réellement 3D, on peut utiliser des quadratures spécifiques comme une quadrature tridimensionnelle de Gauss-Hermite.

Cela amène par exemple aux méthodes :

- aux harmoniques sphériques [22],
- aux ordonnées discrètes (DOM) [24],
- Boltzmann sur réseau (BsR), en cinétique des gaz [94, 111].

●●● En cinétique des gaz, exemple type de transport non-linéaire, les méthodes statistiques se posent essentiellement comme une alternative à la discrétisation de l’espace des vitesses. Une difficulté importante qu’elles proposent de surmonter est le calcul du terme collisionnel de Boltzmann, qui est une intégrale au moins pentadimensionnelle de la fonction de distribution.

La méthode statistique pour la cinétique des gaz la plus utilisée est certainement la *simulation directe Monte-Carlo* (DSMC). Elle repose sur 2 fondements [9]. Un suivi d’un essaim de molécules fictives permet de prendre en compte naturellement le transport balistique. Les collisions sont prises en compte en faisant collisionner aléatoirement des molécules situées dans des mêmes petites cellules d’une grille. Plusieurs variantes se distinguent, par exemple selon la façon détaillée de simuler les collisions [72, 73], ou encore selon la gestion du temps considéré comme variant par pas ou de manière continue [66, 67].

En cinétique des plasmas, non-linéaire elle-aussi, on peut signaler la méthode de particules en cellules (PiC), qui sert à résoudre les équations de Vlasov. Comme en DSMC, on suit des particules fictives ; la différence essentielle avec la méthode DSMC est l’utilité de la grille. Plutôt que de faire apparaître des collisions sinon improbables, son usage est de permettre le calcul du champ électromagnétique, vecteur d’interactions à longue portée entre les particules, en effaçant les interactions à courte portée. La décroissance lente des forces électrostatiques avec la distance explique la prévalence des interactions à longue portée entre charges électriques, en cinétique des plasmas.

●●● En physique du transport linéaire, que ce soit en neutronique, en transfert radiatif, ou dans certains problèmes de physique des plasmas, la situation des méthodes statistiques est différente. Il n’y en a qu’une, et c’est la méthode de

Monte-Carlo (MMC), au sens où l'entendaient ses concepteurs neutroniciens, qui est aussi le sens où on l'entend dans ce manuscrit.

Ordinairement, la méthode de Monte-Carlo se conçoit comme la construction de trajectoires de particules fictives, à l'aide d'un générateur aléatoire, dont on peut tirer des observables moyennes. D'un point de vue mathématique, la méthode de Monte-Carlo est une méthode numérique de calcul d'intégrales. Ces 2 conceptions de la méthode sont complémentaires : nous en parlerons longuement au chapitre 1.

La différence essentielle entre la MMC pure et par exemple la DSMC, vient de la propagation indépendante des sources : en MMC les particules sont suivies une à une, indépendamment les unes des autres. Il n'y a pas non plus de maillage, comme on n'en a pas besoin pour introduire des collisions inter-particulaires, ou pour les empêcher.

La méthode de Monte-Carlo a de nombreux avantages reconnus :

- Elle gère facilement les espaces des phases aux nombreuses dimensions.
- Elle permet de prendre en compte simultanément de nombreux phénomènes physiques, sans augmentation drastique de la dépense computationnelle.
- Ses résultats sont certes entachés d'erreur statistique, mais elle estime cette erreur simultanément à l'estimation principale, ce qui permet de fournir des intervalles de confiance réglables pour les résultats.
- Ses résultats ne sont (normalement) pas entachés d'erreur systématique : si on recommence le calcul avec une initialisation différente des générateurs aléatoires, l'erreur générée est différente, et de moyenne nulle. En augmentant la taille d'échantillon à moyenner, on converge vers la valeur exacte recherchée.
- Elle est aisément parallélisable (sur CPUs du moins).
- Elle permet généralement le calcul simultané de plusieurs observables d'intérêt, et même simultanément avec leurs sensibilités à divers paramètres.

Problématique

Quand le transport est non-linéaire, il n'y a plus propagation indépendante des sources. Il est alors communément admis que la méthode de Monte-Carlo n'est plus utilisable :

“So far as the author is aware, the extension of Monte Carlo methods to nonlinear processes has not yet been accomplished and may be impossible.” [28]

“Monte Carlo methods are not generally effective for nonlinear problems mainly because expectations are linear in character.” [64]

Bien sûr, des aménagements de la méthode de Monte-Carlo existent, comme l'introduction d'une grille en DSMC — qui revient à une linéarisation par pas de temps. Mais ils effacent une partie des intérêts de la méthode de Monte-Carlo, comme l'absence d'erreur systématique :

“A nonlinear problem must usually be linearized in order to use Monte Carlo technique.” [64]

Cependant, plusieurs contre-exemples nous permettent aujourd’hui de modérer ce type d’affirmations, et nous allons les prendre comme point de départ pour réexaminer la question de l’utilisation de la méthode Monte-Carlo en transport non-linéaire.

On connaît depuis plus de 10 ans 2 situations où la méthode de Monte-Carlo a pu être employée, de façon exacte donc sans linéarisation aucune, pour des calculs non linéaires :

- la 1^{re} est le calcul d’intégrales de Fredholm contenant des non-linéarités polynomiales. Ivan T. DIMOV a exposé un moyen de le réaliser par méthode de Monte-Carlo, en utilisant des marches aléatoires avec branchement. Bien que peu reprise et peu citée, cette technique est totalement démontrée et fonctionnelle [36, 37], et son auteur affiche pleinement l’objectif d’étendre la méthode de Monte-Carlo en direction de problèmes non-linéaires.
- la 2^e est la prise en compte des propriétés d’absorption et de diffusion d’un milieu en transport linéaire, par exemple en neutronique ou en transfert radiatif, lorsque ces propriétés sont fortement variables — selon l’espace ou selon le temps. Face à ce type de difficultés, une technique très connue est la technique des *collisions nulles* : on rajoute des collisionneurs virtuels de façon à ce que la fréquence de collision devienne constante. La question de l’hétérogénéité est alors remplacée par la gestion (locale) du taux de collisions virtuelles (de façon à ce que la solution du problème reste identique à celle du problème initial).

Cette technique est très employée et très citée, mais personne ne la considère comme un exemple d’utilisation de la méthode de Monte-Carlo dans un problème non-linéaire. Elle est systématiquement présentée comme une technique d’échantillonnage des libres parcours en transport linéaire, et comme elle est très intuitive (nous la présenterons en détails au chapitre 2) aucune mention n’est faite à une quelconque non-linéarité. Pourtant nous verrons que de façon sous-jacente, il s’agit de gérer la non-linéarité de la décroissance exponentielle de la loi d’extinction de Beer, et que le succès de cette technique atteste de la possibilité concrète de faire appel à la méthode de Monte-Carlo dans des problèmes non-linéaires.

On peut exhiber la non-linéarité abordée dans cet exemple, en considérant l’espace des chemins optiques $\mathcal{U}_{\vec{r}}$, et en écrivant pour chaque chemin $\vec{\gamma}$ le coefficient d’extinction k_t comme une fonction de l’abscisse curviligne s le long du chemin. Le problème de transport se traduit alors comme un problème intégral du type :

$$\int_{\mathcal{U}_{\vec{r}}} d\vec{\gamma} \exp\left(-\int_0^{l_{\vec{\gamma}}} ds k_t(s)\right)$$

où on intègre sur l'espace des chemins une fonction non-linéaire d'une intégrale le long de chaque chemin. La méthode des collisions nulles se présente alors comme un moyen d'évaluer cette intégrale par la méthode de Monte-Carlo, malgré la non-linéarité de la combinaison de l'espace des chemins et de l'espace des abscisses curvilignes. Cette méthode est par contre fortement associée au cas particulier de l'exponentielle.

Plus récemment, lors de son travail de modélisation radiative des photobioréacteurs [29], Jérémie DAUCHET a été confronté à deux situations où :

- la complexité présente, notamment géométrique, était trop élevée pour envisager d'autres méthodes numériques que des méthodes statistiques,
- des non-linéarités étaient présentes et rendaient l'emploi de la méthode de Monte-Carlo apparemment impossible.

Ces non-linéarités n'étaient pas de forme exponentielle, donc l'astuce des collisions nulles n'était pas suffisante. Il ne s'agissait pas non plus de situations pouvant se ramener à la famille des problèmes abordées par I. T. DIMOV. J. DAUCHET est pourtant parvenu à résoudre ces deux problèmes, d'une façon qui ouvrait de très larges perspectives. Résumés rapidement, ces 2 exemples sont les suivants (nous reviendrons sur le 2^e au chapitre 3) :

1. Le calcul des propriétés radiatives de particules à partir de la théorie électromagnétique : calcul des coefficients d'absorption et de diffusion, et de la fonction de phase de diffusion, à partir d'une résolution numérique des équations de Maxwell. Les particules considérées étaient les micro-organismes cultivés, et leur formes pouvaient être complexes. En revanche, d'autres données du problème physique (le contraste d'indice de réfraction des particules vis-à-vis de leur milieu, très faible) permettaient l'emploi de l'approximation de la diffusion anormale ou de l'approximation de Schiff [30, 105, 120].

Selon ces approximations, le problème électromagnétique se ramène au balayage d'un espace de chemins $\mathcal{U}_{\mathbb{F}}$ de diffusion simple (un seul changement de direction par chemin) à travers la particule, et au calcul d'un poids $w(\vec{\gamma})$ associé à chaque chemin. La méthode de Monte-Carlo est alors particulièrement bien adaptée au traitement de la complexité géométrique.

Mais les propriétés radiatives sont des propriétés énergétiques, et on a donc besoin de prendre *le carré* du champ électromagnétique. Et ensuite, il s'agit d'intégrer sur l'ensemble des formes et des tailles de particules présentes dans le mélange étudié. En notant $\mathcal{U}_{\vec{X}}$ l'ensemble des configurations géométriques possibles, le problème posé est du type :

$$\int_{\mathcal{U}_{\vec{X}}} d\vec{x} f\left(\int_{\mathcal{U}_{\mathbb{F}}(\vec{x})} d\vec{\gamma} w(\vec{\gamma})\right)$$

avec $f : z \mapsto z^2$. On est donc face au même type de problème intégral que celui qu'on voulait résoudre par l'emploi des collisions nulles ; mais la

fonction non-linéaire par laquelle se combinent les intégrales n'est plus une exponentielle, mais un carré.

2. Le calcul de la productivité totale d'un photobioréacteur : on admet qu'en chaque point \vec{r} du réacteur on sait estimer le rayonnement présent comme une intégrale, sur l'espace des chemins optiques $\vec{\gamma}$ menant à \vec{r} , d'un poids $w(\vec{\gamma})$. On suppose aussi qu'on connaît la loi de conversion f de ce rayonnement en biomasse. Il s'agit d'intégrer cela sur tout le volume \mathcal{V} du réacteur ; le problème intégral résultant est semblable aux précédents :

$$\int_{\mathcal{V}} d\vec{r} f\left(\int_{\mathcal{U}_F(\vec{r})} d\vec{\gamma} w(\vec{\gamma})\right)$$

où la non-linéarité de f peut être relativement complexe.

Face à ces deux difficultés voisines, Jérémie DAUCHET a développé une approche passant par un développement en série entière de la non-linéarité, ce qui revient à aplanir la non-linéarité en l'injectant dans un espace de dimension infinie. Cette dimension infinie ne pose aucun problème dans le contexte de la méthode de Monte-Carlo.

On reconnaît en effet une astuce de ce type dans le cas particulier des collisions nulles : en ajoutant des collisionneurs fictifs, on introduit la récursivité typique des problèmes de diffusion multiple (espace des chemins sans diffusion, avec une diffusion, deux diffusions, *etc*). On remplace la non-linéarité de l'extinction exponentielle par le balayage d'un espace de dimension infinie.

Chez J. DAUCHET, l'espace de dimension infinie est un espace de polynômes. Une seconde idée est alors introduite : dans la méthode de Monte-Carlo, les intégrales sont vues comme des espérances de variables aléatoires, et dans le cas particulier où la fonction f est un monôme de degré n portant sur une variable aléatoire Z on peut écrire :

$$f(E(Z)) = a E(Z)^n = E(a Z_1 \cdots Z_n)$$

où les Z_i sont des variables aléatoires parentes à Z , c'est-à-dire indépendantes et distribuées identiquement à Z . La puissance n de la moyenne de Z est remplacée par la moyenne du produit de n variables aléatoires indépendantes. Aussi simple soit-elle, cette astuce joue un rôle fondamental dans la proposition de J. DAUCHET. Nous y consacrerons le chapitre 3.

Cette proposition a ensuite été reprise par Olivier FARGES pour calculer la productivité annuelle d'une centrale solaire à concentration, en tenant compte d'une conversion électrothermique non-linéaire [44]. On savait depuis longtemps que le calcul de l'énergie récoltée annuellement au récepteur d'une centrale ne poserait aucune difficulté supplémentaire majeure, comparativement au calcul de la puissance reçue instantanée. Cela revient à rajouter une intégration (sur le temps), ce qui est réputé facile en méthode de Monte-Carlo (voir [28], et aussi le chapitre 1.3.1). Dans un article récent [45], Olivier FARGES montre en effet que pour évaluer la puissance totale récoltée par une centrale solaire pendant l'ensemble de sa durée

de vie, le temps de calcul nécessaire est pratiquement identique à celui nécessaire pour évaluer la puissance reçue à un instant donné. Comme si l'intégration sur trente ans (en tenant compte d'une climatologie à l'heure!) ne coûtait rien...

C'est clairement une illustration des capacités de la méthode de Monte-Carlo, dans les problèmes de complexité élevée. La complexité de l'espace des chemins optiques dans un champ de plus d'un millier d'héliostats est déjà élevée. Combiner cet espace de chemins avec l'échelle des temps (variations de la position du soleil et des conditions météorologiques) ne change pas sensiblement le niveau de complexité, quand on l'échantillonne avec la quadrature statistique de la méthode de Monte-Carlo (voir chapitre 1.3.1).

La prise en compte du rendement de conversion de la centrale, dépendant de la puissance instantanée reçue, constitue une deuxième originalité du travail d'O. FARGES. Il est d'ailleurs remarquable que cela se fasse, là encore, sans une augmentation sensible de la dépense computationnelle. Dans ce dernier exemple, le problème intégral suit encore le même schéma :

$$\int_{\mathcal{U}_T} dt f\left(\int_{\mathcal{U}_F(t)} d\vec{\gamma} w(\vec{\gamma})\right)$$

où t est le temps et $\vec{\gamma}$ un chemin optique allant du Soleil au récepteur.

Dans tous ces exemples, on observe un détail algorithmique déjà montré par Ivan DIMOV [36, 37] : le processus d'estimation générant la grandeur d'entrée à la loi non-linéaire f doit généralement être répété — 2 fois dans l'exemple des propriétés optiques car f est un carré, un nombre variable de fois dans les autres exemples. Cela peut être l'occasion d'un alourdissement computationnel, cependant cet alourdissement reste très raisonnable dans tous les exemples testés jusqu'ici. Il est d'autant plus acceptable, qu'on se retrouve alors en situation de réaliser des calculs auparavant jugés impossibles !

Le propos de cette thèse est de mettre en œuvre une approche statistique du même type pour la résolution des équations de Boltzmann non-linéaires ; en particulier pour l'équation de Boltzmann de la cinétique des gaz. Initialement, le problème n'a pas strictement la même forme que dans les exemples précédents : il ne s'agit pas seulement de l'intégrale d'une fonction non-linéaire d'une intégrale. Mais les ingrédients sont les mêmes : nous avons affaire à un problème de transport et la technique des collisions nulles est immédiatement transposable ; la non-linéarité du terme collisionnel est quadratique, c'est à dire qu'elle passe par le produit de la fonction de distribution pour deux vitesses différentes, ce qui est très proche de l'exemple précédent d'électromagnétisme.

Nous verrons qu'une telle extension est possible et notre question sera ensuite de tester l'algorithmique qui en résulte. En particulier, est-ce que resteront valables en cinétique des gaz toutes les caractéristiques propres à la méthode de Monte-Carlo en transfert radiatif :

- La possibilité d'un calcul exact, c'est-à-dire sans biais et avec une estimation fiable de l'incertitude.

- La possibilité d'un calcul sonde, c'est à dire du calcul d'une grandeur en un point, sans avoir à évaluer l'ensemble du champ. C'est standard en transport linéaire, où l'on suit souvent les particules en sens inverse depuis un capteur considéré ponctuel. Mais ceci est réputé être associé aux images de propagation, qui ne sont plus valables en transport non-linéaire. On admet que l'on puisse prédire numériquement une photo prise par satellite sans calculer le rayonnement en tous les points de l'atmosphère ; mais en dynamique des fluides on imagine mal prédire la densité, la vitesse et la température en un point sans calculer l'ensemble du champ.
- La possibilité d'évaluer des événements rares. Il s'agit presque de la même idée que celle qui précède : si on peut faire un calcul sonde, alors on peut se concentrer sur des événements très rares et les évaluer sans difficulté particulière, là où les méthodes statistiques utilisant un ensemble de pseudo-particules seraient en échec, car elles auraient besoin d'un nombre immense de particules pour gérer la rareté des événements recherchés.
- La possibilité de calculer, en même temps que le calcul principal, les sensibilités de la grandeur étudiée aux paramètres du problème.
- La possibilité de faire appel à des échantillonnages par importance ou des reformulations intégrales, le tout en étant guidé par des approches à variance nulle.
- La possibilité de faire appels aux outils numériques de la synthèse d'image pour la gestion des géométries 3D complexes.
- ...

Comme, en cinétique des gaz tout comme en transfert radiatif, notre description physique est basée sur la fonction de distribution (notée f), nous aurons fait l'essentiel du chemin quand nous aurons exprimé un estimateur de $f(\vec{r}; \vec{c}; t)$ pour toutes position \vec{r} , vitesse \vec{c} et date t données. Quand cela sera fait, le raffinement pour estimer des grandeurs macroscopiques (des flux, des débits, ou même des grandeurs locales comme la densité, les densités de flux...), qui sont en fait des intégrales de f , sera immédiat.

En démarrant ce travail, l'inconnue principale est bien sûr l'expression que pourra avoir cet estimateur. *A priori*, on se basera sur des formulations totalement intégrales de l'équation de Boltzmann, comme il en existe pour l'équation de transfert radiatif. Cela donnera lieu à un suivi des particules en sens inverse, en partant d'un détecteur ou d'une caméra comme c'est ordinaire en transfert radiatif. Une différence notable sera cependant que les chemins reconstruits depuis le détecteur se diviseront pour devenir des arbres, comme c'est le cas dans les propositions techniques de I. T. DIMOV [37] ou de Jérémie DAUCHET [29].

Quelques illustrations

●●●● Il est inhabituel de présenter les résultats d'un travail de thèse dès l'introduction du manuscrit, mais nous choisissons tout de même de dévoiler dès à présent quelques caractéristiques des algorithmes de calcul que nous avons finalement obtenus.

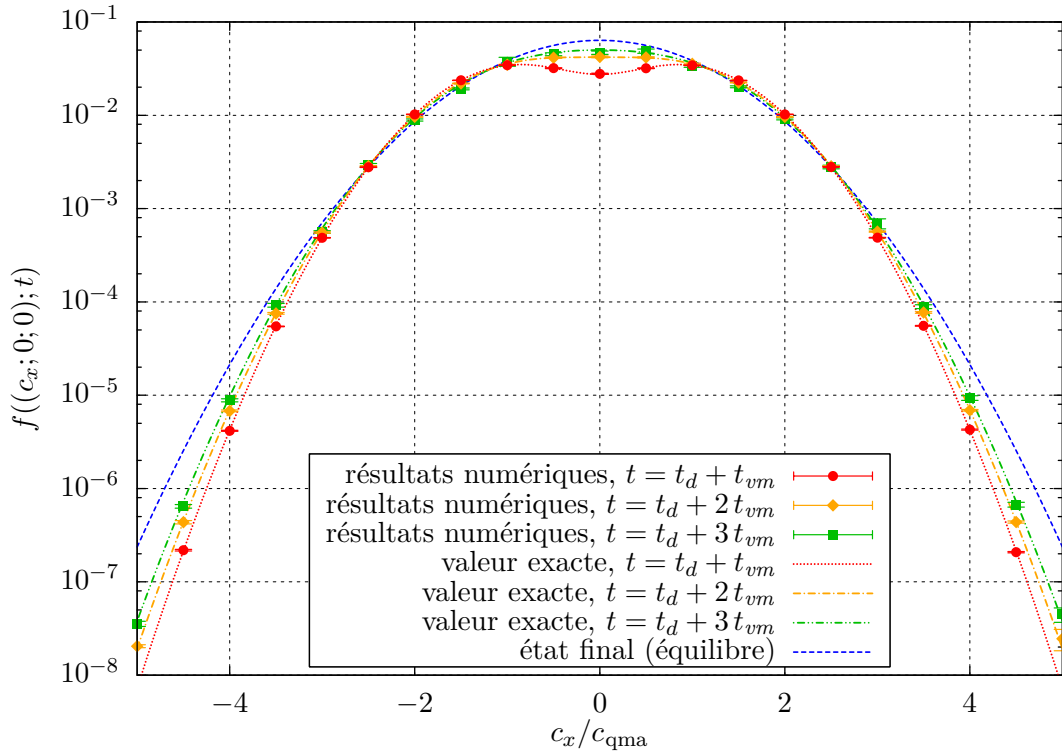
La raison première de ce choix est didactique. Dans ce document, une place majoritaire est consacrée à la présentation de précédents travaux, ou même à la relecture du savoir-faire usuel. Ce qui sera alors expliqué pourra sembler éloigné de notre objet d'étude ; et même si ces redites seront orientées selon les besoins de notre progression logique, elles resteront centrées sur les thématiques originales de chaque auteur. Le lecteur devra donc nous faire confiance, pendant les 3 premiers chapitres, sur l'affirmation que le temps qu'il dépense dans ces pages est utile à la compréhension de notre proposition. De plus, lorsque nous rentrerons dans ce que cette thèse a de pleinement original, nous commencerons par un exemple très académique qui ne relève pas à proprement parler de la cinétique des gaz : cela fera encore un chapitre en apparence hors du sujet de ce manuscrit.

Nous estimons donc que le lecteur a besoin, dès maintenant, d'illustrations concrètes de ce vers quoi le conduit le présent écrit, en termes pratiques. Ce choix implique aussi, d'une certaine façon, que nous accordons plus d'importance au cheminement formel mené durant cette recherche qu'à ses premières mises en œuvre : nous ne cherchons donc nullement à maintenir une sorte de *suspense* sur celles-ci. En fait, ces mises en œuvre ne sont que des tests de faisabilité, et nous manquons encore de recul pour vraiment évaluer la portée de la stricte application de la méthode de Monte-Carlo, sans linéarisation, à la cinétique des gaz. Les résultats que nous présentons de suite (et qui seront décrits entièrement dans les chapitres 5, 6, et lors des perspectives) illustrent donc avant tout le type de conséquences pratiques auxquelles amènent les développements formels.

Les figures 1 et 2 montrent des simulations effectuées dans 2 cas académiques, où la solution exacte est connue. Ces simulations ont été retenues comme des occasions de tester la validité de l'estimateur de f que nous allons construire.

●●● Le cas test de la figure 1 est habituellement dénommé « mode BKW » (pour Bobylev-Krook-Wu). C'est une relaxation vers l'équilibre d'un gaz uniforme et isotrope, mais dont la distribution des particules à l'instant initial n'est pas celle d'équilibre. Il a été découvert par des physiciens spécialistes de la chimie nucléaire, et proposé comme un moyen d'explorer la dynamique de relaxation vers l'équilibre des *queues de distributions*, c'est à dire de particules peu nombreuses mais de haute énergie cinétique, dont le rôle est essentiel en cinétique chimique. Pour nous, c'est l'occasion de contrôler :

- la fonctionnalité de notre estimateur de f , même quand le gaz est en très fort déséquilibre,



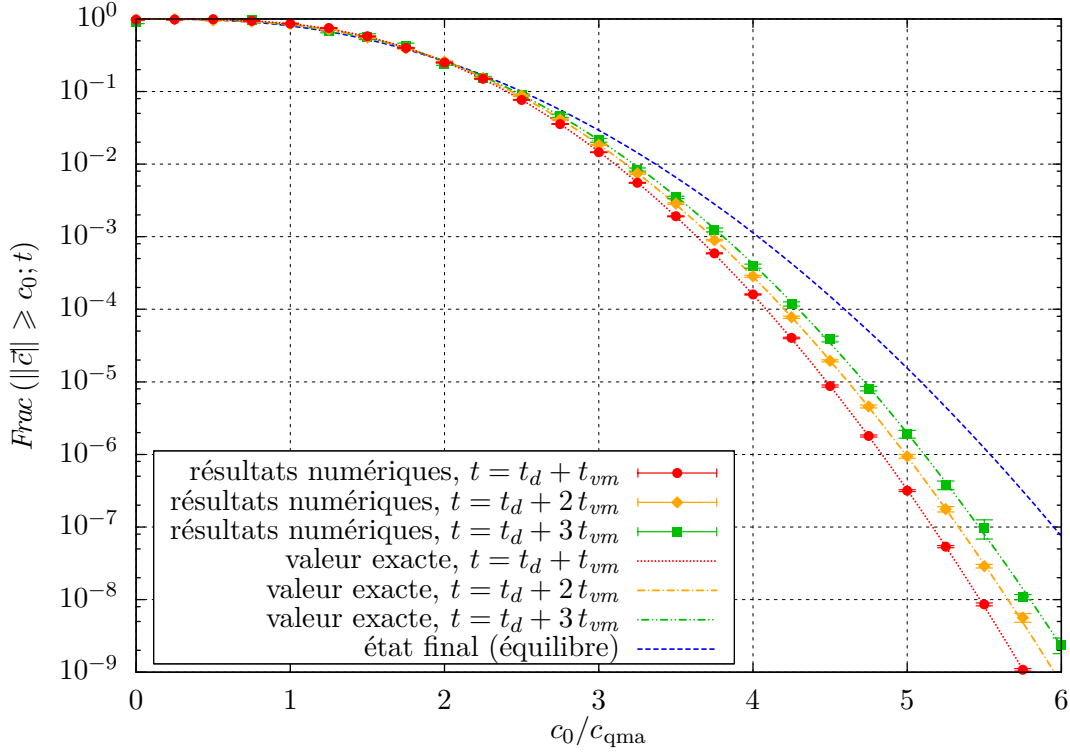
(a) Profils de la fonction de distribution f le long de l'axe (Ox) de l'espace des vitesses, en fonction de la coordonnée c_x de la vitesse

FIGURE 1 – Calculs concernant la distribution des particules, dans un cas académique appelé « mode BKW » (décrit au chapitre 5). Ce cas physique décrit un gaz uniforme, isotrope et sans paroi, c'est pourquoi la dépendance de la fonction de distribution f envers \vec{r} n'est pas mentionnée. f est fixée en un temps initial t_d , selon un profil largement hors-équilibre (suivant la formule (5.12)). t_{vm} est le temps de vol libre moyen d'une particule (totalement constant, dans ce cas physique précis), et c_{qma} est la vitesse quadratique moyenne d'agitation thermique sur chaque axe : le mode BKW ne varie que selon ces 2 paramètres, de façon homothétique.

La solution exacte, connue, est donnée en traits. Les points avec barres d'erreur sont calculés par l'algorithme 5.1, avec 10^4 réalisations en chaque point.

Suite de la figure page suivante.

- notre capacité à effectuer un calcul sonde dans l'espace des vitesses, c'est-à-dire à calculer la fonction de distribution f pour une vitesse donnée (sans aucune discrétisation de l'espace des vitesses),
- que notre estimateur reste précis même dans les zones peu peuplées de l'espace des vitesses (gestion des événements rares).



(b) Profils de la fraction des particules dont la vitesse dépasse un seuil donné, en fonction de ce seuil c_0

FIGURE 1 – Suite de la page précédente.

L'algorithme 5.1 sert à estimer f , en un point quelconque $(\vec{c}; t)$ de l'espace-temps des phases. Il est basé sur des formulations intégrales récursives de l'équation de Boltzmann, avec les quelques nécessaires adaptations aux particularités du mode BKW. On se sert de cet algorithme soit directement dans la figure (a), soit avec une intégration finale dans l'espace des vitesses pour calculer des fractions de particules de haute énergie cinétique (dans la figure (b)).

L'algorithme 5.1 est assez représentatif de ce qu'on obtient à la fin de ce travail : des algorithmes réalisant un calcul sonde de f , c'est-à-dire estimant f en un point sans la calculer dans tout l'espace-temps des phases, fonctionnels même dans des zones très peu peuplées de l'espace des phases, par contre qui connaissent des difficultés quand le temps physique simulé augmente (la variance d'estimation augmente avec t , cela est visible sur les graphes).

La figure 1a illustre ces trois points avec un accord parfait entre simulation et solution exacte. Les calculs sont menés jusqu'à atteindre une précision inférieure à 1 %, sans observer de différence significative, en terme de convergence numérique, entre les points de l'espace des vitesses très peuplés (le centre de la distribution)

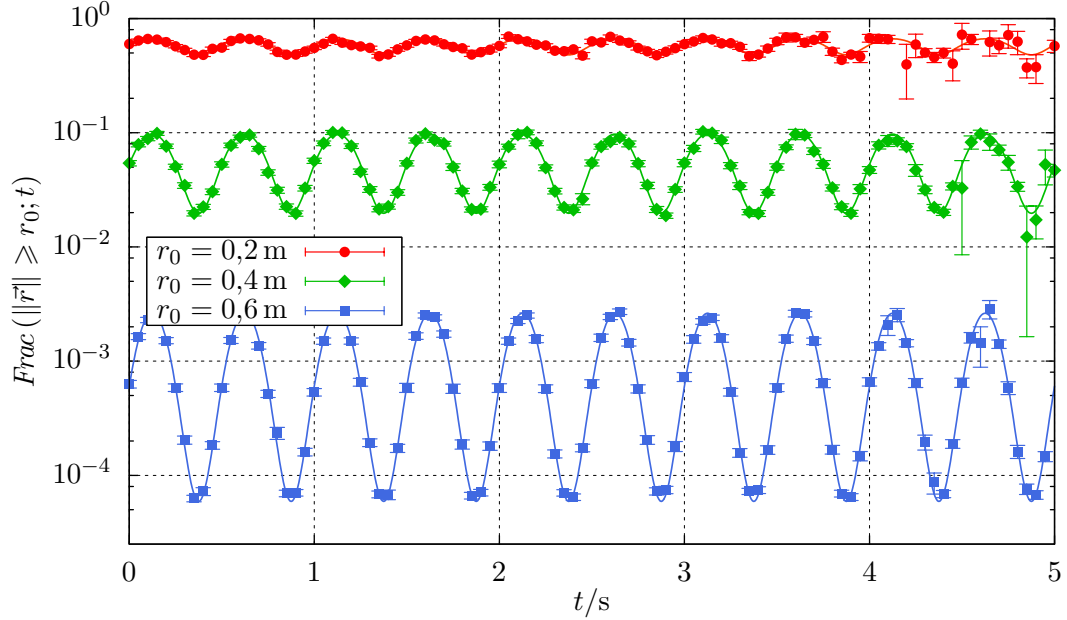


FIGURE 2 – Évolution de la fraction de la matière éloignée de l’origine des positions par une distance supérieure à certains seuils, dans le cas académique de confinement harmonique détaillé au chapitre 6. Dans ce cas physique, une force rappelle élastiquement toutes les molécules vers l’origine. Contrairement au mode BKW, ce cas physique fait donc apparaître une distribution non-uniforme du gaz dans l’espace. En l’occurrence, le gaz forme un nuage centré sur l’origine, gonflant et se dégonflant périodiquement sans jamais atteindre un équilibre.

La solution exacte, connue, est donnée en traits pleins. Les points avec barres d’erreur sont calculés par l’algorithme 6.4, avec 10^4 réalisations en chaque point. Comme l’algorithme 5.1 dont des résultats sont montrés dans la figure 1, l’algorithme 6.4 sert à estimer f en des points quelconques $(\vec{r}; \vec{c}; t)$ de l’espace-temps des phases. Le calcul des fractions de particules éloignées de l’origine est réalisé grâce à cet algorithme, couplé à une intégration statistique finale dans l’espace des positions. Ce couplage d’un échantillonnage de positions et de chemins de molécules (c’est ceci que réalise, en sous-jacent, l’algorithme 6.4) est réalisable par la méthode de Monte-Carlo, pour le coup de façon totalement standard et répandue.

Là encore, il n’y a pas de difficulté importante à calculer ce qui se passe à haute énergie (potentielle, cette fois-ci). Par contre, la variance d’estimation augmente avec le temps physique simulé — on représente environ 1,5 fois le temps de collision moyen, sur ce graphe.

et ceux très dépeuplés (la queue de la distribution). La figure 1b présente une évaluation de la fraction des particules ayant une vitesse supérieure à une valeur de référence c_0 donnée. Elle illustre notre gestion des événements rares dans l'espace des vitesses : nous atteignons les mêmes précisions relatives pour des événements ayant une probabilité de 10^{-8} que pour des événements de probabilité presque unité. On parvient là à évaluer des grandeurs qui seraient complètement inaccessibles par les approches statistiques représentant la distribution des vitesses à l'aide d'un grand nombre de pseudo-particules. Dans notre cas, les pseudo-particules sont remplacées par des chemins à branchements multiples qui partent du point étudié de l'espace des vitesses, aussi dépeuplé soit-il. Par conception même, ces chemins se concentrent sur l'objet d'étude. La faiblesse de l'approche est par contre la multiplication des branchements, qui limite le temps physique simulé atteignable.

●●● Dans ce premier exemple, le gaz était homogène et nous ne testions pas le transport dans l'espace usuel des positions. Le cas test de la figure 2 est au contraire une situation où les molécules d'un gaz sont rappelées élastiquement vers l'origine des positions. Il en résulte que ce gaz forme un nuage centré sur l'origine, et gonfle et se dégonfle de façon périodique sans jamais atteindre un équilibre. Pour nous, c'est l'occasion de montrer que nous sommes capables de prédire correctement la dynamique de très faibles portions de la masse, occupant des zones spécifiques de l'espace ordinaire : l'événement rare est ici la présence d'une molécule très loin du centre attracteur. On voit là encore que la précision des calculs est presque indépendante de la rareté de l'événement et que la limite est le temps physique simulé (du fait des branchements multiples).

La figure 3 montre les résultats obtenus, dans une configuration faisant intervenir une géométrie complexe, sur une grille de points de l'espace ordinaire. Dans ce cas-ci, nous ne disposons bien entendu pas d'une solution symbolique. C'était toutefois l'occasion de montrer que :

- Aucun problème n'émerge de la prise en compte d'une géométrie complexe. Comme dans les applications solaires précédemment évoquées, on utilise les outils usuels de la communauté de la synthèse d'image. L'algorithme est indépendant de la géométrie, et il n'y a donc aucun travail supplémentaire à fournir pour passer des exemples académiques précédents à une configuration de type industrielle.
- On est toujours capable de réaliser un calcul sonde, ici dans l'espace ordinaire, ce qui illustre le fait que nous parvenons bien à prédire les caractéristiques locales d'un écoulement (densité, vitesse, température) sans calculer l'ensemble de cet écoulement. La notion de calcul sonde est donc bien étendue à la mécanique des fluides, malgré la non-linéarité qui exclut les images propagatives. On pourrait à l'inverse dire que les branchements multiples qui caractérisent nos algorithmes permettent de raisonner avec des images de type propagatif, mais arborescentes.

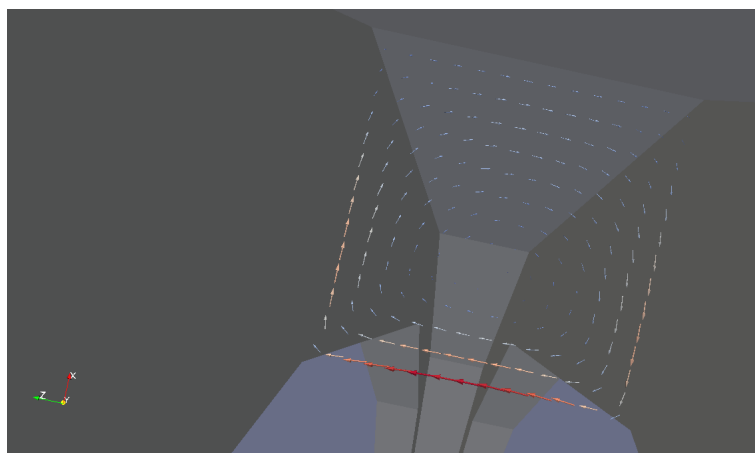


FIGURE 3 – Visualisation du champ de densité de débit, obtenue grâce à des calculs sondes effectués sur une grille de points. La situation physique est détaillée en partie dans le chapitre consacré aux perspectives ; elle fait intervenir une géométrie complexe.

La géométrie visible sur l'image est une gorge, dans une courte section de tube circulaire. Elle est modélisée à l'aide d'un maillage de triangles, c'est pourquoi elle est si anguleuse. À chaque extrémité du tube, la distribution des molécules entrantes est fixée — à une maxwellienne, de chaque côté à la même pression mais avec 2 températures différentes. Tous les autres paramètres nécessaires sont fixés aussi, comme le type de rebond aux parois, la section efficace de collision. . .

La prise en compte de la géométrie n'est pas difficile. Elle intervient uniquement par le biais de calculs d'intersections trajectoires/géométrie, et la forme de l'algorithme de calcul de f n'est pas modifiée par rapport aux exemples précédents des figures 1 et 2. De plus, pour ce calcul d'intersections on peut profiter du savoir-faire de la communauté de la synthèse d'image.

Organisation du manuscrit

Nous démarrerons à partir les travaux menés dans l'équipe où ce travail de thèse a été encadré, équipe répartie dans le centre RAPSODEE², le LAPLACE³, le laboratoire PROMES⁴, et l'IBP⁵. Les travaux visés ici sont spécifiquement :

1. ceux de J. DAUCHET au sujet de la prise en compte d'une loi de couplage non-linéaire [29],
2. ceux réalisés autour des algorithmes dits à *collisions nulles* [50].

2. centre de Recherche d'Albi en génie des Procédés des SOLides Divisés, de l'Énergie et de l'Environnement, UMR 5302

3. Laboratoire PLAsmas et Conversion d'Énergie, UMR 5213

4. laboratoire PROcédés, Matériaux, et Énergie Solaire, UPR 8521

5. Institut Blaise Pascal, UMR 6602

●●●● Bien que les points de départ de ce travail de thèse sont des concepts courants en physique du transport linéaire — excepté les propositions de J. DAUCHET en MMC, encore relativement confidentielles —, ce que nous allons obtenir en transport non-linéaire est très inhabituel, et pas forcément intuitif. Cela implique que nous devons progresser point par point, et avec beaucoup de rigueur. En particulier, nous redémontrons ce que nous utilisons habituellement pour le transport linéaire, afin de connaître ce que nous pourrions étendre sans abus vers la physique du transport non-linéaire.

Du coup, nous allons nous retrouver avec un déroulement logique très formel, contenant des résultats mathématiques assez simples mais nombreux. Nous en arriverons même à présenter certains résultats sous forme de théorèmes, avec leurs preuves (succinctes). Nous avons conscience qu'un tel format d'écriture est peu courant dans notre discipline ; nous avons néanmoins jugé que le lecteur issu de notre communauté, tout comme nous, considérerait nécessaire d'asseoir ou de ré-asseoir solidement chaque détail de sa théorie ou de sa pratique usuelle avant de l'utiliser dans un champ nouveau.

●●●● Compte tenu de la quantité de ce que nous voulons présenter, et afin que la lecture de cet écrit reste la plus agréable possible, nous avons fait le choix de décanter son contenu entre 2 niveaux de lecture. Ces niveaux sont distingués par des éditions différentes.

- Le niveau principal est écrit avec une seule colonne par page, et avec une police normale. Par exemple, tout cette introduction en fait partie. Ce niveau contient le fil conducteur de la thèse. Afin de comprendre tout son contenu, il est nécessaire de le lire dans l'ordre des pages.
- Le niveau secondaire est écrit avec 2 colonnes par page, et en police réduite. Cependant, pour des raisons techniques certaines équations et figures occupent toute la largeur de page. Dans la table des matières, il est signalé par une étoile (*) .

Ce niveau contient des éléments dispensables en première lecture. Il fait cependant partie intégrante de la thèse défendue. Il contient typiquement les phases de redites et d'approfondissement du savoir-faire existant, dont nous évoquons à l'instant l'utilité.

●●●● Par ailleurs, ce manuscrit s'organise principalement en 2 parties.

●●● Les chapitres 1, 2, et 3 sont consacrés à une appropriation suffisante des concepts et des techniques liés à la physique du transport ou à la méthode de Monte-Carlo, qui nous ont été nécessaires au montage d'une approche statistique en transport non-linéaire. Ces chapitres sont les seuls à contenir des sections du niveau de lecture secondaire évoqué à l'instant.

Le chapitre 1 est une relecture de quelques bases de la physique du transport et des méthodes de Monte-Carlo. L'accent est mis sur ce qui nous sera utile ensuite :

ceci comprend :

- les formulations intégrales des équations de transport,
- une compréhension mathématique de la méthode de Monte-Carlo,
- le lien qui existe entre cette dernière compréhension, et une pratique — usuelle — de la MMC comme analogie statistique du transport.

Le chapitre 2 est consacré aux « algorithmes à collisions nulles » : on désigne par ce terme une sorte d’astuce, quand on fait du calcul par MMC en physique du transport linéaire, pour gérer l’occurrence des collisions quand leur fréquence n’est pas *simple*. Ils constituent la clé essentielle nous ayant permis de proposer des estimateurs pour la fonction de distribution en cinétique des gaz.

Le chapitre 3 est une relecture de la technique développée par J. DAUCHET puis Olivier FARGES pour gérer les lois de couplage non-linéaire, quand on utilise la MMC. Nous l’utiliserons ensuite, combinée avec les algorithmes à collisions nulles, pour gérer le transport non-linéaire.

●●● Les chapitres 4, 5, et 6 détaillent la construction d’estimateurs de la fonction de distribution. La fonctionnalité de ces estimateurs est par la même occasion abondamment testée.

Dans le chapitre 4, on se concentre sur la résolution par MMC des équations différentielles ordinaires scalaires non-linéaires. Un système différentiel, pouvant être vu comme l’équation de Boltzmann dont on aurait retiré l’espace des phases, sert de test tout au long de ce chapitre.

Les chapitres 5 et 6 montrent la construction d’estimateurs de f dans 2 cas académiques de cinétique des gaz. Ces estimateurs sont testés, en particulier contre les solutions symboliques disponibles dans ces cas simples.

Chapitre 1

Équation de transfert radiatif, équation de Boltzmann, et méthode de Monte-Carlo

Le présent chapitre amène au lecteur les fondements physiques, mathématiques, et numériques sur lesquels s'appuie ce travail de thèse. Ce qui est présenté ici vise surtout à apporter au lecteur une compréhension particulière de la pratique de la méthode de Monte-Carlo en transport linéaire, dans laquelle la suite du manuscrit est aisément compréhensible.

Nous commençons en section 1.1 par rappeler les expressions précises des équations de Boltzmann dans les champs disciplinaires qui nous servent d'exemples (transfert radiatif et cinétique des gaz). Ces équations sont ré-exprimées en section 1.2 sous forme purement intégrale.

Puis nous parlons en section 1.3 de la méthode de Monte-Carlo, et de son utilisation en transport linéaire.

Ensuite, on s'attarde en section 1.4 sur la justification purement mathématique d'une pratique courante en transport linéaire : on peut résoudre les équations de Boltzmann linéaires par la méthode de Monte-Carlo, en suivant des particules au fil de leurs trajectoires. Pourtant, les formulations intégrales des équations de Boltzmann font uniquement apparaître un suivi des particules depuis leur point d'arrivée, donc en remontant leurs trajectoires.

Enfin, on donnera en section 1.5 un exemple concret d'utilisation de la méthode de Monte-Carlo pour résoudre un problème de transport linéaire.

1.1 Équations de Boltzmann

●●●● Comme nous l'avons annoncé dans le chapitre d'introduction, l'objet d'étude de ce manuscrit est la résolution des équations de Boltzmann.

La 1^{re} équation de ce type est celle posée par Ludwig BOLTZMANN vers la fin du 19^e siècle, lors de la construction de son modèle cinétique des gaz. C'est généralement celle-ci qu'on désigne sous le terme d'« équation de Boltzmann », comme nous le faisons dans ce manuscrit.

Par la suite, la modélisation d'autres phénomènes physiques a donné lieu à l'établissement d'équations similaires, qui ont été regroupées sous le nom d'« équations de Boltzmann », dans un sens plus générique. Comme expliqué dans le chapitre d'introduction, toutes ces équations s'écrivent sous la forme :

$$D(f) = C(f) \tag{1.1}$$

où f est la fonction de distribution des particules dont on considère le transport, D est la dérivée en suivant le mouvement, et C est un opérateur de collision.

1.1.1 Fonctions de distribution

1.1.1.1 Fonctions de distribution, et approche mésoscopique

●●● Toutes les descriptions et les modèles physiques pour la cinétique des gaz ou le transfert radiatif, dont il sera question dans ce manuscrit, seront donc basés sur des *fonctions de distribution*, respectivement des molécules ou des photons. La définition de telles fonctions de distribution implique que l'on se place dans une description mésoscopique, et ce à 2 titres :

1. On suppose l'existence d'une échelle dite mésoscopique, intermédiaire entre le système à l'étude et les corpuscules le constituant, permettant de considérer une multitude de petites parties du système telles que :
 - Chaque petite partie puisse être considérée comme ponctuelle, comparativement à l'extension totale du système.
L'échelle mésoscopique s'oppose ainsi à l'échelle globale, où on ne considère que quelques observables d'état simples du système ou de certains de ses sous-volumes.
 - Chaque petite partie soit assez étendue pour contenir de très nombreux corpuscules. De la sorte, le contenu de 2 parties centrées sur un même point du système est proportionnel au volume de ces parties.
Cela signifie que la fonction de distribution ignore la granularité de la matière. L'échelle mésoscopique s'oppose ainsi à l'échelle microscopique, où on essaie de caractériser l'état de chaque corpuscule constituant le système.
2. L'espace de définition de la fonction de distribution est l'espace des états possibles pour les corpuscules constituant le système. Cet espace est appelé l'espace des phases. Souvent, il ne se limite pas à l'espace des positions.
Une approche mésoscopique se rapproche en cela d'une approche microscopique, car on essaie de tenir compte de l'influence de toutes les variables d'état des corpuscules (bien qu'on ne considère pas les corpuscules individuellement).

Elle se distingue de même d'une approche macroscopique à l'échelle mésoscopique, où le système est décrit par les champs de certaines observables, définis uniquement sur l'espace des positions (l'espace ordinaire). Dans une telle approche, l'établissement des équations d'évolution nécessite plus que la simple application des lois de la mécanique aux corpuscules, comme c'est le cas avec les approches purement mésoscopiques.

●●●● Il reste maintenant à :

- préciser les espaces des phases dans lesquels les molécules ou les photons vont se répartir,
- donner les expressions exactes des équations de Boltzmann, applicables à nos cas d'étude.

1.1.1.2 Fonction de distribution des molécules

●●●● Pour rester simple, dans ce travail nous nous limitons, en cinétique des gaz, aux gaz monoatomiques et mono-espèce.

Un gaz est qualifié de monoatomique, à la base, quand ses molécules ne sont chacune constituée que d'un seul atome. Du point de vue de la thermodynamique, on qualifie un gaz de « monoatomique » quand ses molécules n'échangent pas d'énergie interne lors de leurs collisions — c'est-à-dire que leurs collisions préservent l'énergie cinétique. L'absence d'énergie interne observable des molécules est en pratique bel et bien liée à leur caractère monoatomique, du moins dans les conditions ambiantes de température et de pression. L'explication de cela, issue de la mécanique quantique, ne sera pas donnée ici. Mais à chaque fois que nous parlerons, dans la suite de ce manuscrit, du caractère monoatomique d'un gaz, ce sera pour signifier l'absence d'énergie interne aux molécules.

Un gaz est qualifié de mono-espèce quand ses molécules sont toutes identiques, du point de vue de la composition chimique et isotopique. Cela implique en particulier que toutes les molécules ont la même masse.

●●●● Finalement, dans un gaz mono-espèce et monoatomique, les molécules ne diffèrent que par 2 variables d'état : la position \vec{r} et la vitesse \vec{c} . Ainsi, leur espace des phases E_p égale $E_r \times E_c$, le produit de l'espace des positions E_r (l'espace ordinaire, en fait) et de l'espace des vitesses E_c .

La fonction de distribution f des molécules est donc définie sur l'espace-temps des phases $E_r \times E_c \times E_t$, où E_t est l'échelle des temps. Elle est définie comme la quantité de molécules par unité de volume de l'espace ordinaire et par unité de volume de l'espace des vitesses, et son unité est la $\text{mol.m}^{-6}.\text{s}^3$ ¹. Cela s'écrit :

$$f(\vec{r}; \vec{c}; t) = \frac{dn}{d\vec{r}d\vec{c}} \quad (1.2)$$

1. Il est très courant, dans la littérature consacrée à la cinétique des gaz ou des plasmas, de travailler avec le *nombre de molécules* par unité de volume de l'espace des phases. Dans ce cas, l'unité de f est le $\text{m}^{-6}.\text{s}^3$.

où dn est la quantité de molécules présentes à l'instant t , dans une partie infinitésimale de E_r de volume $d\vec{r}$ et centrée sur \vec{r} , et dans une partie infinitésimale de E_c de volume $d\vec{c}$ et centrée sur \vec{c} .

1.1.1.3 Fonction de distribution des photons, et luminance

●●●● La physique moderne explique que l'énergie lumineuse est transportée par des particules appelées photons. L'état de ceux-ci est décrit par 4 variables :

- la position \vec{r} ,
- la direction de propagation \vec{u} . La vitesse de propagation est imposée par le milieu, on ne peut pas faire accélérer ou ralentir un photon en lui « appliquant une force ».
- la fréquence ν de l'onde électromagnétique associée,
- le spin \vec{s} , lié à la direction du champ électrique transporté par l'onde électromagnétique associée.

Le comportement des photons est descriptible par la mécanique quantique ; celle-ci postule que les photons ont un comportement hybride, à la fois corpusculaire, ondulatoire, et aléatoire. Dans ce manuscrit, à chaque fois que nous parlerons de transfert radiatif nous envisagerons le comportement des photons comme purement corpusculaire, ce qui impose quelques restrictions dans les cas d'étude :

- Les effets du spin des photons, c'est-à-dire les effets de la polarisation de la lumière, sont totalement négligés. Dans les applications impliquant du transfert radiatif, étudiées jusqu'ici par l'équipe où ce travail de thèse a eu lieu — chambres de combustion, transferts thermiques atmosphériques, procédés solaires, imagerie médicale —, ils avaient peu d'influence. Qui plus est, leur prise en compte dans une modélisation corpusculaire du transfert radiatif est en fait assez malaisée².
- Aucun phénomène d'interférence n'est observable.
- Les ondes électromagnétiques dans le système sont incohérentes.

●●●● Ainsi, l'espace des phases dans lequel les photons se répartissent est $E_p = E_r \times E_u \times E_\nu$, où $E_u = \mathcal{S}(E_c; \vec{0}; 1)$ est l'espace des directions et $E_\nu \simeq \mathbb{R}^{+*}$ est l'échelle des fréquences (appelée aussi échelle spectrale).

La fonction de distribution des photons se définit donc comme la quantité de photons, par unité de volume de l'espace ordinaire, par unité d'angle solide, et par unité d'intervalle de l'échelle des fréquences. La définition de l'angle solide est une généralisation dans l'espace de la notion d'angle du plan ; l'angle solide correspond à ce que devrait être le volume dans l'espace des directions.

2. Le lecteur que cela intéresse peut se reporter entre autres à l'ouvrage [59] page 725.

●●●● Une subtilité existe en revanche quand on recherche la différentielle spectrale (sur l'échelle des fréquences). Selon les auteurs, la variable repérant les photons sur l'échelle spectrale peut être la fréquence ν , mais aussi la longueur d'onde $\lambda = c_0/\nu$ ou le nombre d'onde $\eta = \nu/c_0$ ³. Selon les cas l'élément différentiel sur l'échelle des fréquences change, et la définition de la fonction de distribution avec, même s'il y a une correspondance stricte entre les définitions possibles.

Ce dernier point aura peu d'importance dans ce manuscrit. Toutefois, en fonction de l'élément différentiel choisi sur l'échelle spectrale, nous décidons de noter la fonction de distribution f_ν , f_λ ou f_η . Par exemple, f_ν s'écrit :

$$f_\nu(\vec{r}; \vec{u}; \nu; t) = \frac{dn_\gamma}{d\vec{r}d\vec{u}d\nu} \quad (1.3)$$

où dn_γ est la quantité de photons présents à l'instant t , dans une partie infinitésimale de E_r de volume $d\vec{r}$ et centrée sur \vec{r} , dans un angle solide infinitésimal $d\vec{u}$ centré sur \vec{u} , et dans un intervalle infinitésimal de fréquence de longueur $d\nu$ et centré sur ν . L'unité de f_ν est donc le $\text{mol.m}^{-3}.\text{Hz}^{-1}$.

Du fait de la définition de la fréquence, de la longueur d'onde et du nombre d'onde, on a :

$$f_\lambda = \frac{c_0}{\lambda^2} f_\nu \quad \text{et} \quad f_\eta = c_0 f_\nu \quad (1.4)$$

car c'est la quantité totale de photons dans un intervalle spectral donné qui doit être conservé entre les différentes définitions de f .

Dans ce manuscrit, nous omettons la plupart du temps de préciser quelle est la différentielle spectrale de la fonction de distribution des photons, que nous noterons alors f . Plusieurs situations existent, où cette précision est en fait inutile :

- Soit tout le rayonnement présent dans le système est à une seule fréquence. On dit alors qu'il est monochromatique. Par exemple, les lasers sont des sources de rayonnement monochromatique.
- Soit, dans la plage spectrale où se produit le transfert radiatif qui nous intéresse, les propriétés du milieu de propagation sont indépendantes de la fréquence. On peut dans ce cas la retirer de l'espace des phases, comme elle n'a plus d'influence.
- On réalise une étude théorique, où le rayonnement est considéré monochromatique.

Les transferts radiatifs à fréquences différentes (même légèrement) sont indépendants dans beaucoup d'applications, dont toutes celles évoquées dans ce manuscrit. Il y a alors du sens à réfléchir sur des problèmes de transfert radiatif monochromatique. Le retour à des problèmes polychromatiques se fait en sommant simplement le transfert radiatif à chaque fréquence, ce qui est possible car ils sont indépendants.

3. c_0 est ici la vitesse de la lumière dans le vide, constante universelle. Il est plus prudent de travailler avec les longueurs ou les nombres d'onde dans le vide, car la longueur d'onde d'une onde électromagnétique change quand elle passe d'un milieu de propagation à un autre ; c'est la fréquence de l'onde qui est conservée.

●●●● Dans la communauté s'intéressant au transfert radiatif, même quand on utilise une approche mésoscopique, la grandeur descriptive n'est généralement pas la fonction de distribution des photons, mais la *luminance* (*irradiance* en anglais). Plutôt que compter le nombre de photons près d'un point de l'espace des phases, on compte l'énergie qu'ils transportent. Quand on se replace à l'échelle des systèmes étudiés, c'est cette énergie qui intéresse le physicien ou l'ingénieur, plus que les photons en tant que particules individuelles.

La luminance est ainsi définie comme la puissance lumineuse, à un instant t , se propageant dans une unité d'angle solide autour d'une direction \vec{u} , et divisée par la surface normale à la direction de propagation. Cette surface normale est centrée autour d'un point de l'espace ordinaire \vec{r} .

L'unité de L_ν , par exemple, est le $\text{W.m}^{-2}.\text{Hz}^{-1}$.

Étant donné qu'un photon :

- transporte une énergie $h\nu$, où $h = 6,626\,07 \times 10^{-34}$ J.s est la constante de Planck,
- se déplace à une vitesse notée c ,

la luminance L et la fonction de distribution des photons f sont directement liées par la formule :

$$L(\vec{r}; \vec{u}; \nu; t) = h\nu c(\vec{r}; \vec{u}; \nu; t) N_A f(\vec{r}; \vec{u}; \nu; t) \quad (1.5)$$

où :

- $c(\vec{r}; \vec{u}; \nu; t)$ est la vitesse de la lumière, au point de l'espace-temps des phases considéré.
- N_A est le nombre d'Avogadro, environ égal à $6,022\,141\,29 \times 10^{23} \text{ mol}^{-1}$. Si on exprime f comme un nombre de photons (sans unité) et non comme une quantité de photons (en moles), le nombre d'Avogadro N_A disparaît de la formule (1.5).

1.1.2 Fonction de distribution et observables macroscopiques

●●●● Nous expliquions précédemment qu'une description purement mésoscopique est plus précise qu'une description macroscopique à l'échelle mésoscopique ; car on décrit complètement la répartition des corpuscules dans l'espace de leurs états possibles (l'espace des phases), et pas seulement dans l'espace ordinaire.

Comme l'espace des phases contient l'espace ordinaire, on peut exprimer les variables macroscopiques locales selon la fonction de distribution, même si l'inverse n'est pas vrai.

En cinétique des gaz, si on a un gaz mono-espèce et monoatomique de masse molaire M , alors avec la définition de f donnée à la section 1.1.1.2 on peut exprimer :

- la densité molaire η et la densité massique ρ :

$$\eta(\vec{r}; t) = \int_{E_c} f(\vec{r}; \vec{c}; t) d\vec{c} \quad \rho(\vec{r}; t) = M\eta(\vec{r}; t) = M \int_{E_c} f(\vec{r}; \vec{c}; t) d\vec{c} \quad (1.6)$$

- les densités de débit molaire \vec{j}_{mol} et massique \vec{j}_{mas} , et la vitesse moyenne \vec{v} :

$$\vec{j}_{mol}(\vec{r}; t) = \eta(\vec{r}; t) \vec{v}(\vec{r}; t) = \int_{E_c} f(\vec{r}; \vec{c}; t) \vec{c} d\vec{c} \quad (1.7a)$$

$$\vec{j}_{mas}(\vec{r}; t) = \rho(\vec{r}; t) \vec{v}(\vec{r}; t) = M \int_{E_c} f(\vec{r}; \vec{c}; t) \vec{c} d\vec{c} \quad (1.7b)$$

- la densité d'énergie totale e :

$$e(\vec{r}; t) = \frac{M}{2} \int_{E_c} f(\vec{r}; \vec{c}; t) \vec{c}^2 d\vec{c} \quad (1.8)$$

Cette énergie totale est généralement décomposée entre énergie cinétique macroscopique et énergie thermique (due à l'agitation thermique microscopique)

u :

$$e(\vec{r}; t) = \underbrace{\frac{1}{2} \rho(\vec{r}; t) \vec{v}(\vec{r}; t)^2}_{\text{énergie cinétique macroscopique}} + u(\vec{r}; t) \quad (1.9a)$$

avec

$$u(\vec{r}; t) = \frac{M}{2} \int_{E_c} f(\vec{r}; \vec{c}; t) (\vec{c} - \vec{v}(\vec{r}; t))^2 d\vec{c} \quad (1.9b)$$

L'énergie thermique u est directement liée à la température par :

$$T(\vec{r}; t) = \frac{2u(\vec{r}; t)}{3R\eta(\vec{r}; t)} \quad (1.9c)$$

où $R \simeq 8,314\,462 \text{ J.mol}^{-1}.\text{K}^{-1}$ est la constante des gaz parfaits. La température est ainsi une mesure locale de l'énergie moyenne des particules par degré de liberté (ici il y en 3 : les coordonnées de la vitesse des particules).

- le tenseur de transfert de quantité de mouvement \vec{j}_p :

$$\vec{j}_p(\vec{r}; t) = M \int_{E_c} f(\vec{r}; \vec{c}; t) (\vec{c} \otimes \vec{c}) d\vec{c} \quad (1.10)$$

Ce transfert volumique de quantité de mouvement est ordinairement décomposé entre une part due au transport macroscopiquement observable et une part dénommée *tenseur des contraintes*. Nous notons ce dernier tenseur $\vec{\sigma}$ en accord avec les notations usuelles de la mécanique des milieux continus⁴.

$$\vec{j}_p(\vec{r}; t) = \underbrace{\rho(\vec{r}; t) \vec{v}(\vec{r}; t) \otimes \vec{v}(\vec{r}; t)}_{\text{transport macroscopique de quantité de mouvement}} + \vec{\sigma}(\vec{r}; t) \quad (1.11a)$$

4. Cependant, nous choisissons positives les contraintes en pression, contrairement au choix usuel de la mécanique des milieux continus qui est de choisir positives les contraintes en tension. Nous faisons ce choix pour rendre $\vec{\sigma}$ positif.

avec

$$\vec{\sigma}(\vec{r}; t) = M \int_{E_c} f(\vec{r}; \vec{c}; t) ((\vec{c} - \vec{v}(\vec{r}; t)) \otimes (\vec{c} - \vec{v}(\vec{r}; t))) d\vec{c} \quad (1.11b)$$

La pression est la partie sphérique du tenseur des contraintes : $p(\vec{r}; t) = \frac{1}{3} \text{Tr}(\vec{\sigma}(\vec{r}; t))$.

En remarquant que $u(\vec{r}; t) = \frac{1}{2} \text{Tr}(\vec{\sigma}(\vec{r}; t))$, on retrouve l'équation d'état des gaz parfaits monoatomiques : $p(\vec{r}; t) = \frac{2}{3} u(\vec{r}; t) = R \eta(\vec{r}; t) T(\vec{r}; t)$. Cette équation est donc une conséquence directe de la validité de l'approche corpusculaire pour la cinétique des gaz⁵.

— la densité de flux total \vec{j}_e :

$$\vec{j}_e(\vec{r}; t) = \frac{M}{2} \int_{E_c} f(\vec{r}; \vec{c}; t) \vec{c}^2 d\vec{c} \quad (1.12a)$$

On dégage ordinairement de cette densité de flux total la densité de flux total d'énergie thermique \vec{j}_u :

$$\vec{j}_u(\vec{r}; t) = \frac{M}{2} \int_{E_c} f(\vec{r}; \vec{c}; t) (\vec{c} - \vec{v}(\vec{r}; t))^2 d\vec{c} \quad (1.12b)$$

et de cette dernière la densité de flux diffusif d'énergie thermique \vec{j}_{Du} :

$$\vec{j}_{Du}(\vec{r}; t) = \frac{M}{2} \int_{E_c} f(\vec{r}; \vec{c}; t) (\vec{c} - \vec{v}(\vec{r}; t))^2 (\vec{c} - \vec{v}(\vec{r}; t)) d\vec{c} \quad (1.12c)$$

Par définition, ces densités de flux sont liées par les relations :

$$\vec{j}_e(\vec{r}; t) = \underbrace{\vec{j}_u(\vec{r}; t) + \frac{1}{2} \rho(\vec{r}; t) \vec{v}(\vec{r}; t)^2 \vec{v}(\vec{r}; t)}_{\text{flux macroscopique d'énergie cinétique macroscopique}} + \underbrace{\left(\vec{\sigma}(\vec{r}; t) \right) (\vec{v}(\vec{r}; t))}_{\text{puissance locale des contraintes}} \quad (1.12d)$$

et

$$\vec{j}_u(\vec{r}; t) = \underbrace{u(\vec{r}; t) \vec{v}(\vec{r}; t)}_{\text{flux macroscopique d'énergie thermique}} + \vec{j}_{Du}(\vec{r}; t) \quad (1.12e)$$

Une fois que l'on connaît ces grandeurs macroscopiques locales, on obtient les grandeurs globales par intégration sur l'espace ordinaire ou sur le temps. Ainsi peut-on calculer la masse totale, l'énergie totale, la force exercée sur une paroi, le flux de chaleur traversant une paroi...

En exprimant la conservation de la masse, de la quantité de mouvement et de l'énergie, il est possible de dégager des relations entre les précédentes grandeurs macroscopiques. Si une approche mésoscopique est indispensable, c'est que ce jeu de relations n'est pas fermé. En fait, il manque généralement l'expression de \vec{j}_{Du} et de la partie déviatrice de $\vec{\sigma}$.

5. et aussi de l'absence de force intermoléculaire à longue portée, présupposée lors de l'écriture du tenseur densité de flux de quantité de mouvement

En transfert radiatif, en intégrant sur l'espace des directions et l'échelle des fréquences on retrouve les observables habituelles du transfert de chaleur. On a ainsi les expressions de :

— la densité d'énergie radiative e :

$$e(\vec{r}; t) = \int_{E_u} d\vec{u} \int_{E_\nu} d\nu h\nu N_A f_\nu(\vec{r}; \vec{u}; \nu; t) \quad (1.13a)$$

$$= \int_{E_u} d\vec{u} \int_{E_\nu} d\nu \frac{L_\nu(\vec{r}; \vec{c}; \nu; t)}{c(\vec{r}; \vec{c}; \nu; t)} \quad (1.13b)$$

Dans la littérature consacrée au transfert radiatif, on rencontre plutôt l'irradiance G , strictement équivalente quand l'indice de réfraction est constant :

$$G(\vec{r}; t) = \int_{E_u} d\vec{u} \int_{E_\nu} d\nu h\nu c N_A f_\nu(\vec{r}; \vec{u}; \nu; t) \quad (1.13c)$$

$$= \int_{E_u} d\vec{u} \int_{E_\nu} d\nu L_\nu(\vec{r}; \vec{c}; \nu; t) \quad (1.13d)$$

— la densité de flux radiatif :

$$\vec{j}(\vec{r}; t) = \int_{E_u} d\vec{u} \int_{E_\nu} d\nu h\nu N_A c(\vec{r}; \vec{u}; \nu; t) f_\nu(\vec{r}; \vec{u}; \nu; t) \vec{u} \quad (1.14a)$$

$$= \int_{E_u} d\vec{u} \int_{E_\nu} d\nu L_\nu(\vec{r}; \vec{u}; \nu; t) \vec{u} \quad (1.14b)$$

On peut exprimer à partir de ce dernier le « terme source radiatif » :

$$S_r(\vec{r}; t) = -\text{div}_{\vec{r}} \vec{j}(\vec{r}; t) \quad (1.15)$$

Le terme source radiatif joue un rôle primordial dans l'étude des transferts thermiques par rayonnement. La création ou la destruction de photons se fait par des interactions avec le milieu de propagation, qui préservent l'énergie totale. Donc S_r est la puissance volumique prélevée par le milieu au « gaz de photons ». Dans les équations de la chaleur applicables aux milieux de propagation, le rayonnement intervient entièrement au travers de S_r .

On utilise fréquemment des versions *spectrales* des précédentes grandeurs : c'est-à-dire qu'on ne fait pas l'intégration sur l'échelle des fréquences. La raison de l'emploi de grandeurs spectrales est l'habitude indépendante des transferts radiatifs à fréquences différentes. On écrit alors $e_\nu(\vec{r}; \nu; t) = \int_{E_u} d\vec{u} h\nu N_A f_\nu(\vec{r}; \vec{u}; \nu; t)$, $\vec{j}_\nu(\vec{r}; \nu; t), \dots$. Si on utilise une différentielle autre qu'en fréquence sur l'échelle des fréquences, on notera plutôt $e_\lambda(\vec{r}; \lambda; t)$ et $\vec{j}_\lambda(\vec{r}; \lambda; t)$, ou encore $e_\eta(\vec{r}; \eta; t), \dots$

Par ailleurs, au lieu de noter « $\int_{E_u} d\vec{u} \dots$ » pour l'intégration sur l'espace des directions, on note généralement « $\int_{4\pi} d\vec{u} \dots$ ». La raison en est que l'espace des directions a une surface de 4π : $\int_{E_u} d\vec{u} = 4\pi$.

De même qu'en cinétique des gaz, il est possible de dégager des relations entre les précédentes grandeurs macroscopiques. Ce jeu de relations n'est pas fermé, il manque l'expression complète de \vec{j} .

1.1.3 Hypothèses, établissement, et conséquences de l'équation de Boltzmann

Nous donnons dans cette section une démonstration de l'équation de Boltzmann.

Pour cela, nous ne passons pas par la hiérarchie BBGKY (pour Bogoliubov-Born-Green-Kirkwood-Yvon) [16, 18, 70, 71, 124]. Celle-ci consiste, en résumé, à poser un système couplé d'équations décrivant l'évolution des fonctions de distribution à 1 corps⁶, à 2 corps, etc jusqu'à N corps — où N est le nombre de particules dans le système. Ensuite, en tronquant ce système on obtient une équation d'évolution sur la fonction de distribution à 1 corps.

Ici, on se limitera à contraindre l'évolution de la fonction de distribution (à 1 corps) par quelques hypothèses simples. Le but en est multiple. Premièrement, nous voulons rester transversal lors de la démonstration, dans le sens où celle-ci nécessiterait peu d'aménagement pour être transposée dans un autre domaine disciplinaire (par exemple la biologie). Deuxièmement, les hypothèses mises en œuvre sont directement liées à la dynamique des molécules : elles limitent les événements possibles pour ces dernières. Cela permet d'associer immédiatement des images physiques aux différents termes de l'équation de Boltzmann ; or dans la suite du présent manuscrit, nous essayons de permettre de telles images après de conséquents développements mathématiques sur cette équation.

1.1.3.1 Hypothèses sur les molécules

●●●● Nous faisons les hypothèses suivantes :

1. Les molécules suivent des trajectoires balistiques (sans interaction mutuelle) entre des *collisions*, ponctuelles dans l'espace et dans le temps.
2. Pendant leurs trajectoires balistiques, les molécules sont soumises à un champ de force extérieur connu (potentiellement nul), en toute généralité fonction de leur position, de leur vitesse, et du temps.
3. Les collisions impliquent chacune uniquement 2 particules, situées au même point de l'espace et du temps.
4. Le gaz est mono-espèce : toutes les molécules sont identiques.
5. Les molécules sont monoatomiques, les collisions conservent leur énergie cinétique.
6. Le taux de collision en un point de l'espace-temps est strictement proportionnel au carré de f , *via* un facteur ne dépendant que de la différence de vitesse entre les molécules.

Quelques remarques sur ces hypothèses :

- Un exemple de force dépendante de la vitesse des molécules est la composante magnétique de la force de Lorentz.

6. C'est bien de fonctions de distributions à 1 corps dont nous parlons depuis le début de ce manuscrit.

- La 6^e hypothèse sous-entend entre autres que les molécules n'ont pas d'orientation privilégiée. Ceci peut être mis en défaut par exemple en présence d'un intense champ électrique ou magnétique.

●●●● La 6^e hypothèse est liée à une notion nommée *chaos moléculaire*.

Cela signifie en premier lieu que les molécules ne s'ordonnent pas à petite échelle. Cette hypothèse est donc fautive dans les phases condensées, ou du moins ne s'y énonce pas de la même manière.

Le gaz est par ailleurs supposé ne pas avoir de mémoire fine. Cela signifie que son évolution est chaotique, et que s'il existe une fonction d'état très sensible aux conditions initiales du gaz, telle que sa valeur doivent être ré-évaluée régulièrement, cela n'a pas de conséquence notable sur l'évolution du système. De telles fonctions sensibles se rencontrent par exemple en arithmétique modulaire, ce qui justifie son usage en cryptographie ; nous supposons qu'il n'y a aucune observable telle dans le système.

Cette hypothèse signifie aussi qu'aucune fluctuation, due à la granularité de la matière, n'apparaît spontanément dans la fonction de distribution f .

Le chaos moléculaire, associé à la ponctualité des collisions, est la cause formelle de l'irréversibilité (que nous évoquerons brièvement en section 1.1.3.3) de l'équation de Boltzmann, qui s'oppose à la réversibilité du système à l'échelle microscopique. Cette opposition n'est pas gênante tant que le système contient un très grand nombre de molécules, et qu'on s'intéresse à la dynamique du système loin de l'équilibre (détaillé en section 1.1.3.3) : dans ce cas, l'équation de Boltzmann peut être une approximation très correcte de la dynamique moléculaire.

Le lecteur intéressé par la notion de chaos moléculaire, ses limites et ses défauts pourra se reporter par exemple aux publications [63, 83, 84].

1.1.3.2 Obtention de l'équation de Boltzmann

●●●● Soit une partie Ω_p de l'espace des phases E_p , qui soit le produit d'une partie Ω_r de E_r et d'une partie Ω_c de E_c : $\Omega_p = \Omega_r \times \Omega_c$. La frontière de Ω_r est notée $\partial\Omega_r$ et la frontière de Ω_c est notée $\partial\Omega_c$; chacune de ces frontières est supposée avoir une normale sortante en tous ses points. Ces normales sortantes sont notées respectivement $\vec{n}_r(\vec{r})$ et $\vec{n}_c(\vec{c})$. Ω_p est supposée fixe dans le temps. La quantité de molécules qu'elle contient est notée $n(t)$.

$n(t)$ varie à cause de 4 phénomènes :

1. Des particules proches de la frontière de Ω_r la traversent, parce qu'elles ont une vitesse.
2. Des particules proches de la frontière de Ω_c la traversent, parce qu'elles subissent une accélération progressive $\vec{a} = \frac{\vec{F}}{m}$ (due aux forces extérieures de longue portée).
3. Des particules dans Ω_c font un saut dans E_c , en subissant une collision.

4. Des particules font un saut dans E_c et se retrouvent dans Ω_c , en subissant une collision.

Cela s'écrit :

$$\begin{aligned} d_t n(t) = & \int_{\partial\Omega_r} d\vec{r} \int_{\Omega_c} d\vec{c} (-\vec{c} \cdot \vec{n}_r(\vec{r})) f(\vec{r}; \vec{c}; t) \\ & + \int_{\Omega_r} d\vec{r} \int_{\partial\Omega_c} d\vec{c} (-\vec{a}(\vec{r}; \vec{c}; t) \cdot \vec{n}_c(\vec{c})) f(\vec{r}; \vec{c}; t) \\ & - \int_{\Omega_r} d\vec{r} \int_{\Omega_c} d\vec{c} f(\vec{r}; \vec{c}; t) \int_{E_c} d\vec{c}_* f(\vec{r}; \vec{c}_*; t) \int_{E_c} d\vec{c}' \int_{E_c} d\vec{c}'_* b_F(\vec{c}'; \vec{c}'_* | \vec{c}; \vec{c}_*) \\ & + \int_{\Omega_r} d\vec{r} \int_{\Omega_c} d\vec{c} \int_{E_c} d\vec{c}' f(\vec{r}; \vec{c}'; t) \int_{E_c} d\vec{c}'_* f(\vec{r}; \vec{c}'_*; t) \int_{E_c} d\vec{c}_* b_F(\vec{c}; \vec{c}_* | \vec{c}'; \vec{c}'_*) \end{aligned} \quad (1.16)$$

où $b_F(\vec{c}_1; \vec{c}_2 | \vec{c}_3; \vec{c}_4)$ est la probabilité, par unité de temps et 2 fois par unité de volume de l'espace des vitesses, de 2 molécules situées au même point et de vitesses \vec{c}_3 et \vec{c}_4 , de collisionner et de se retrouver respectivement avec les vitesses \vec{c}_1 et \vec{c}_2 .

●●●● En appliquant le théorème de Green-Ostrogradski, et en considérant que la précédente expression est vraie quelle que soit Ω_p , on arrive à :

$$\begin{aligned} \partial_t f(\vec{r}; \vec{c}; t) + \text{div}_{\vec{r}} (f(\vec{r}; \vec{c}; t) \vec{c}) + \text{div}_{\vec{c}} (f(\vec{r}; \vec{c}; t) \vec{a}(\vec{r}; \vec{c}; t)) = \\ - f(\vec{r}; \vec{c}; t) \int_{E_c} d\vec{c}_* f(\vec{r}; \vec{c}_*; t) \int_{E_c} d\vec{c}' \int_{E_c} d\vec{c}'_* b_F(\vec{c}'; \vec{c}'_* | \vec{c}; \vec{c}_*) \\ + \int_{E_c} d\vec{c}' f(\vec{r}; \vec{c}'; t) \int_{E_c} d\vec{c}'_* f(\vec{r}; \vec{c}'_*; t) \int_{E_c} d\vec{c}_* b_F(\vec{c}; \vec{c}_* | \vec{c}'; \vec{c}'_*) \end{aligned} \quad (1.17)$$

Cette nouvelle expression peut être simplifiée. Premièrement, comme les variables \vec{r} et \vec{c} sont indépendantes, on a : $\text{div}_{\vec{r}} \vec{c} = 0$. Ensuite, comme les collisions sont réversibles (cf. la mécanique quantique) (on parle de *micro-réversibilité*) et n'ont pas de direction privilégiée : $b_F(\vec{c}'; \vec{c}'_* | \vec{c}; \vec{c}_*) = b_F(-\vec{c}; -\vec{c}_* | -\vec{c}'; -\vec{c}'_*) = b_F(\vec{c}; \vec{c}_* | \vec{c}'; \vec{c}'_*)$. Ainsi :

$$\begin{aligned} \partial_t f(\vec{r}; \vec{c}; t) + \vec{c} \cdot \overrightarrow{\text{grad}}_{\vec{r}} f(\vec{r}; \vec{c}; t) + \text{div}_{\vec{c}} (f(\vec{r}; \vec{c}; t) \vec{a}(\vec{r}; \vec{c}; t)) = \\ \int_{E_c} d\vec{c}_* \int_{E_c} d\vec{c}' \int_{E_c} d\vec{c}'_* b_F(\vec{c}; \vec{c}_* | \vec{c}'; \vec{c}'_*) (-f(\vec{r}; \vec{c}; t) f(\vec{r}; \vec{c}_*; t) + f(\vec{r}; \vec{c}'; t) f(\vec{r}; \vec{c}'_*; t)) \end{aligned} \quad (1.18)$$

Dans cette écriture, on peut interpréter les vitesses comme suit :

- \vec{c} est la vitesse d'une particule suivie
- \vec{c}_* est la vitesse d'une partenaire de collision
- \vec{c}' est la vitesse avant collision de la particule suivie
- \vec{c}'_* est la vitesse avant collision de la partenaire de collision

●●●● Le second membre de l'équation (1.18) peut être simplifié. Dans le cas d'un gaz monoatomique et monomoléculaire, les collisions conservent la quantité de mouvement et l'énergie cinétique, ce qui s'écrit :

$$\begin{cases} \vec{c} + \vec{c}_* = \vec{c}' + \vec{c}'_* \\ \vec{c}^2 + \vec{c}_*^2 = \vec{c}'^2 + \vec{c}'_*^2 \end{cases} \quad (1.19)$$

Il en résulte en particulier que :

$$\|\vec{c} - \vec{c}_*\| = \|\vec{c}' - \vec{c}'_*\| \quad (1.20)$$

Après calcul, on aboutit à ce que le système (1.19) est équivalent à la condition :

$$\exists \vec{u}' \in \mathcal{S}(E_c; \vec{0}; 1), \quad \left\{ \vec{c}' = \frac{\vec{c} + \vec{c}_*}{2} + \frac{\|\vec{c} - \vec{c}_*\|}{2} \vec{u}' \quad \text{et} \quad \vec{c}'_* = \frac{\vec{c} + \vec{c}_*}{2} - \frac{\|\vec{c} - \vec{c}_*\|}{2} \vec{u}' \right\} \quad (1.21)$$

En posant $\int_{E_c} \int_{E_c} \int_{E_c} \int_{E_c} b_F(\vec{c}; \vec{c}_* | \vec{c}'; \vec{c}'_*) \dots \equiv \int_{E_c} \int_{4\pi} b_F(\vec{c} - \vec{c}_*; \vec{u}') \dots$, on se ramène à :

$$\begin{aligned} \partial_t f(\vec{r}; \vec{c}; t) + \vec{c} \cdot \overrightarrow{\text{grad}}_{\vec{r}} f(\vec{r}; \vec{c}; t) + \text{div}_{\vec{c}} \left(f(\vec{r}; \vec{c}; t) \vec{a}(\vec{r}; \vec{c}; t) \right) = \\ \int_{E_c} d\vec{c}_* \int_{4\pi} d\vec{u}' b_F(\vec{c} - \vec{c}_*; \vec{u}') \left(-f(\vec{r}; \vec{c}; t) f(\vec{r}; \vec{c}_*; t) + f(\vec{r}; \vec{c}'; t) f(\vec{r}; \vec{c}'_*; t) \right) \end{aligned} \quad (1.22)$$

où \vec{c}' et \vec{c}'_* respectent les expressions (1.21). En considérant l'invariabilité de b_F selon le repère, on arrive finalement à l'équation de Boltzmann :

$$\begin{aligned} \partial_t f(\vec{r}; \vec{c}; t) + \vec{c} \cdot \overrightarrow{\text{grad}}_{\vec{r}} f(\vec{r}; \vec{c}; t) + \text{div}_{\vec{c}} \left(f(\vec{r}; \vec{c}; t) \vec{a}(\vec{r}; \vec{c}; t) \right) = \\ \int_{E_c} d\vec{c}_* \int_{4\pi} d\vec{u}' b_F \left(\|\vec{c} - \vec{c}_*\|; \vec{u}' \cdot \frac{\vec{c} - \vec{c}_*}{\|\vec{c} - \vec{c}_*\|} \right) \times \\ \left(-f(\vec{r}; \vec{c}; t) f(\vec{r}; \vec{c}_*; t) + f(\vec{r}; \vec{c}'; t) f(\vec{r}; \vec{c}'_*; t) \right) \end{aligned} \quad (1.23)$$

On ré-exprime souvent l'équation de Boltzmann sous forme globale, en posant l'opérateur de collision de Boltzmann C_B :

$$\partial_t f(\vec{r}; \vec{c}; t) + \vec{c} \cdot \overrightarrow{\text{grad}}_{\vec{r}} f(\vec{r}; \vec{c}; t) + \text{div}_{\vec{c}} \left(f(\vec{r}; \vec{c}; t) \vec{a}(\vec{r}; \vec{c}; t) \right) = C_B(f)(\vec{r}; \vec{c}; t) \quad (1.24a)$$

avec

$$\begin{aligned} C_B(f)(\vec{r}; \vec{c}; t) = \int_{E_c} d\vec{c}_* \int_{4\pi} d\vec{u}' b_F \left(\|\vec{c} - \vec{c}_*\|; \vec{u}' \cdot \frac{\vec{c} - \vec{c}_*}{\|\vec{c} - \vec{c}_*\|} \right) \times \\ \left(-f(\vec{r}; \vec{c}; t) f(\vec{r}; \vec{c}_*; t) + f(\vec{r}; \vec{c}'; t) f(\vec{r}; \vec{c}'_*; t) \right) \end{aligned} \quad (1.24b)$$

Notion de section efficace : Usuellement, la participation à la fréquence de collision, $b_F(g; u)$, est exprimée selon une grandeur appelée *section efficace*.

La *section efficace différentielle* de collision est définie comme :

$$\sigma_F(g; u) = b_F(g; u)/g \quad (1.25)$$

où g est la vitesse relative des particules et u le cosinus de l'angle de déviation lors de la collision. La *section efficace totale* de collision est définie comme :

$$\forall \vec{u} \in \mathcal{S}(E_c; \vec{0}; 1), \sigma_T(g) = \int_{4\pi} d\vec{u}' \sigma_F(g; \vec{u} \cdot \vec{u}') = 2\pi \int_{-1}^1 du \sigma_F(g; u) \quad (1.26)$$

quand cette intégrale est bien définie.

Le terme de « section efficace » est justifiée par l'image suivante : en circulant dans l'espace, chaque molécule balaie un cylindre où elle percute ses voisines.

Finalement, l'équation de Boltzmann s'écrit ordinairement, dans un gaz monoatomique et monomoléculaire :

$$\begin{aligned} \partial_t f(\vec{r}; \vec{c}; t) + \vec{c} \cdot \overrightarrow{\text{grad}}_{\vec{r}} f(\vec{r}; \vec{c}; t) + \text{div}_{\vec{c}} \left(f(\vec{r}; \vec{c}; t) \vec{a}(\vec{r}; \vec{c}; t) \right) = \\ \int_{E_c} d\vec{c}_* \int_{4\pi} d\vec{u}' \|\vec{c} - \vec{c}_*\| \sigma_F \left(\|\vec{c} - \vec{c}_*\|; \vec{u}' \cdot \frac{\vec{c} - \vec{c}_*}{\|\vec{c} - \vec{c}_*\|} \right) \times \\ \left(-f(\vec{r}; \vec{c}; t) f(\vec{r}; \vec{c}_*; t) + f(\vec{r}; \vec{c}'; t) f(\vec{r}; \vec{c}'_*; t) \right) \end{aligned} \quad (1.27)$$

où \vec{c}' et \vec{c}'_* suivent les expressions (1.21).

À toutes fins utiles, citons 2 modèles très classiques de section efficace :

sphères dures : Les molécules sont considérées comme des sphères parfaites, dures, élastiques, et lisses. L'application de la mécanique de Newton et de calculs géométriques aboutit à ce que dans ce modèle, la section efficace est constante : $\sigma_F(g; u) = \sigma_F$ constante.

particules de Maxwell : De façon générale, on désigne sous ce terme les modèles où $b_F(g; u)$ ne dépend pas de la différence de vitesse g . Bien que plus éloignés de la réalité que le modèle de sphères dures, ces modèles de collision ont l'avantage de rendre plus simple la résolution de l'équation de Boltzmann.

1.1.3.3 Distribution d'équilibre des molécules

●●●● Comme affirmé précédemment, l'équation de Boltzmann décrit une dynamique irréversible. Cela se montre par un résultat appelé le théorème H, dont nous donnons ici quelques éléments de démonstration.

Le départ est de prendre les termes de l'équation de Boltzmann, de les multiplier par $\ln f(\vec{r}; \vec{c}; t)$, puis de les intégrer selon l'espace des vitesses. En retirant le terme

dû aux forces à longue portée, on obtient :

$$\int_{E_c} d\vec{c} (\ln f(\dots)) \times (\partial_t f(\dots)) + \int_{E_c} d\vec{c} (\ln f(\dots)) \times (\operatorname{div}_{\vec{r}}(f(\dots)\vec{c})) = \int_{E_c} d\vec{c} (\ln f(\dots)) \times C_B(f)(\dots) \quad (1.28)$$

où on a omis les dépendances de f pour alléger l'écriture.

Posons maintenant une entropie volumique s et une densité de flux d'entropie \vec{j}_s :

$$s(\vec{r}; t) = \int_{E_c} d\vec{c} f(\vec{r}; \vec{c}; t) \ln f(\vec{r}; \vec{c}; t) \quad \vec{j}_s(\vec{r}; t) = \int_{E_c} d\vec{c} f(\vec{r}; \vec{c}; t) \ln f(\vec{r}; \vec{c}; t) \vec{c} \quad (1.29)$$

Après quelques traitements mathématiques, et l'utilisation de la conservation de la quantité de particules dans l'équation de Boltzmann, on arrive depuis l'équation (1.28) à :

$$\partial_t s(\vec{r}; t) + \operatorname{div}_{\vec{c}} \vec{j}_s(\vec{r}; t) = \int_{E_c} d\vec{c} (\ln f(\vec{r}; \vec{c}; t)) \times C_B(f)(\vec{r}; \vec{c}; t) \quad (1.30)$$

On peut montrer, en utilisant les invariants collisionnels⁷ (masse, quantité de mouvement, et énergie), que le terme de droite dans l'équation (1.30) ne peut être que négatif ou nul. On retrouve ici un axiome de la thermodynamique : il existe une grandeur d'état s qui ne peut varier que par échange ou par création locale (destruction, avec les définitions données ici). L'évolution des gaz, tant que l'équation de Boltzmann est valable, est irréversible.

Une distribution d'équilibre peut par ailleurs être dégagée depuis l'équation (1.30).

On sait que η , \vec{v} , et u sont des invariants collisionnels. Le gaz est donc à l'équilibre si s est minimale compte-tenu de η , \vec{v} , et u . Ce minimum est atteint quand la fonction de distribution égalise la fonction de distribution d'équilibre :

$$f_{eq}(\vec{r}; \vec{c}; t) = \frac{\eta(\vec{r}; t)}{(2\pi)^{3/2} c_{qma}(\vec{r}; t)^3} \times \exp\left(-\frac{(\vec{c} - \vec{v}(\vec{r}; t))^2}{2 c_{qma}(\vec{r}; t)^2}\right) \quad (1.31a)$$

où

$$c_{qma}(\vec{r}; t) = \sqrt{\frac{2u(\vec{r}; t)}{3\rho(\vec{r}; t)}} \quad (1.31b)$$

est la vitesse quadratique moyenne d'agitation thermique sur chaque axe.

L'équilibre thermodynamique est donc atteint quand $f = f_{eq}$, avec η , \vec{v} , et u uniformes (sinon le transport des molécules interromprait immédiatement l'équilibre).

7. les grandeurs que les collisions ne modifient pas

1.1.4 Équation de transfert radiatif

●●●● L'équation de transfert radiatif (ETR) s'obtient sous des hypothèses assez semblables à celles qui mènent à l'équation de Boltzmann. Cela explique que l'ETR soit une équation de Boltzmann linéaire, dans le sens expliqué dans l'introduction de ce manuscrit. En particulier :

- Les photons n'interagissent pas entre eux, mais avec un milieu matériel. On suppose ce dernier connu.
- Ces interactions sont ponctuelles dans l'espace et dans le temps.
- Ces interactions sont indépendantes.
- Pour un seul photon, la probabilité de collisionner dans un intervalle de temps infinitésimal ne dépend que de la position dans l'espace des phases. Cette hypothèse se rapproche du chaos moléculaire de la cinétique des gaz ; une différence est que cette fois-ci le chaos est situé dans le milieu. Par exemple, si on est dans un milieu poreux, on suppose qu'il n'est pas assez ordonné pour favoriser certaines longueurs particulières entre collisions.
- Il n'y a que 3 types d'interactions rayonnement/matière possibles, dans un milieu semi-transparent :

absorption : La matière récupère l'énergie transportée par un photon, celui-ci est détruit.

émission : La matière crée un photon, évacuant de la sorte une partie de son énergie.

diffusion : La matière fait changer à un photon sa direction de propagation. Formellement, on peut considérer un événement de diffusion comme une absorption suivi instantanément d'une ré-émission.

On fait quelques hypothèses simplificatrices, en plus des précédentes :

- La polarisation de la lumière est négligée.
- Le milieu est d'indice de réfraction constant.
- Les événements de diffusion sont élastiques, c'est-à-dire qu'ils ne font pas changer la fréquence des photons.
- Le milieu est en équilibre thermodynamique local. Ceci signifie que la distribution locale des constituants de la matière dans les états qu'ils peuvent occuper est proche d'une distribution d'équilibre imposée par la thermodynamique. En particulier, cet équilibre est très peu perturbé par les transferts radiatifs, si ce n'est par la puissance qu'ils transportent dans l'espace ordinaire.

Fonction de distribution d'équilibre (des photons) : on l'utilise dans l'équation de transfert radiatif. Dans un milieu fermé en équilibre thermodynamique global⁸ à la température T depuis un temps « infini », dans tout l'espace des phases

8. Ceci signifie que chaque point est en équilibre thermodynamique local, et que le système entier est de température égale.

la distribution des photons est à sa valeur d'équilibre f_{eq} , égale à⁹ :

$$f_{\nu eq}(T; \nu) = \frac{2 \nu^2}{N_A c^3} \times \frac{1}{\exp\left(\frac{h\nu}{k_B T}\right) - 1} \quad (1.32)$$

où $k_B = 1,380\,65 \times 10^{-23}$ J/K est la constante de Boltzmann — qu'on retrouve en cinétique des gaz, en effet $k_B = R/N_A$.

●●●● Suite à toutes ces considérations, et en utilisant un raisonnement semblable à celui permettant de dégager l'équation de Boltzmann pour les gaz dans la section précédente, l'équation de transfert radiatif (ETR) s'écrit :

$$\begin{aligned} \partial_t f_{\nu}(\vec{r}; \vec{u}; \nu; t) + c \vec{u} \cdot \overrightarrow{\text{grad}}_{\vec{r}} f_{\nu}(\vec{r}; \vec{u}; \nu; t) = \\ c k_a(\vec{r}; \vec{u}; \nu; t) \left(f_{\nu eq}(T(\vec{r}; t); \nu) - f_{\nu}(\vec{r}; \vec{u}; \nu; t) \right) + \\ \int_{4\pi} d\vec{u}' c k_{d,F}(\vec{r}; \vec{u}|\vec{u}'; \nu; t) \left(f_{\nu}(\vec{r}; \vec{u}'; \nu; t) - f_{\nu}(\vec{r}; \vec{u}; \nu; t) \right) \end{aligned} \quad (1.33)$$

où :

- $k_a(\vec{r}; \vec{u}; \nu; t)$ est le coefficient d'absorption, c'est-à-dire la probabilité pour un photon en \vec{r} , de direction \vec{u} , de fréquence ν , et autour de l'instant t , d'être absorbé en parcourant une longueur élémentaire ds de chemin,
- $k_{d,F}(\vec{r}; \vec{u}; \nu; t)$ est le coefficient angulaire de diffusion, c'est-à-dire la probabilité pour un photon en \vec{r} , de direction \vec{u}' , de fréquence ν , et autour de l'instant t , d'être diffusé vers la direction \vec{u} en parcourant une longueur élémentaire ds de chemin,
- f_{eq} est la distribution d'équilibre précédemment introduite.

Remarque. L'identité du terme d'émission avec $c k_a(\vec{r}; \vec{u}; \nu; t) f_{\nu eq}(T(\vec{r}; t); \nu)$ est une conséquence de l'équilibre thermodynamique local (ETL), associé au 2^e principe de la thermodynamique.

On ré-écrit habituellement l'ETR :

$$\begin{aligned} \frac{1}{c} \partial_t f_{\nu}(\vec{r}; \vec{u}; \nu; t) + \vec{u} \cdot \overrightarrow{\text{grad}}_{\vec{r}} f_{\nu}(\vec{r}; \vec{u}; \nu; t) = \\ k_a(\vec{r}; \vec{u}; \nu; t) \left(f_{\nu eq}(T(\vec{r}; t); \nu) - f_{\nu}(\vec{r}; \vec{u}; \nu; t) \right) + \\ \int_{4\pi} d\vec{u}' k_{d,F}(\vec{r}; \vec{u}|\vec{u}'; \nu; t) \left(f_{\nu}(\vec{r}; \vec{u}'; \nu; t) - f_{\nu}(\vec{r}; \vec{u}; \nu; t) \right) \end{aligned} \quad (1.34)$$

On dégage alors un terme collisionnel $C_T(f_{\nu})$:

$$\frac{1}{c} \partial_t f_{\nu}(\vec{r}; \vec{u}; \nu; t) + \vec{u} \cdot \overrightarrow{\text{grad}}_{\vec{r}} f_{\nu}(\vec{r}; \vec{u}; \nu; t) = C_T(f_{\nu})(\vec{r}; \vec{u}; \nu; t) \quad (1.35a)$$

9. Encore une fois, si f est exprimée comme un nombre de photons et non comme une quantité de photons (en moles), le nombre d'Avogadro disparaît de la formule qui suit.

avec :

$$C_T(f_\nu)(\vec{r}; \vec{u}; \nu; t) = k_a(\vec{r}; \vec{u}; \nu; t) \left(f_{\nu \text{ eq}}(T(\vec{r}; t); \nu) - f_\nu(\vec{r}; \vec{u}; \nu; t) \right) + \int_{4\pi} d\vec{u}' k_{d,F}(\vec{r}; \vec{u}|\vec{u}'; \nu; t) \left(f_\nu(\vec{r}; \vec{u}'; \nu; t) - f_\nu(\vec{r}; \vec{u}; \nu; t) \right) \quad (1.35b)$$

Plutôt que de parler de coefficient de diffusion angulaire, en transfert radiatif on parle plus de *coefficient de diffusion* k_d et de *fonction de phase* Φ :

$$k_d(\vec{r}; \vec{u}'; \nu; t) = \int_{4\pi} d\vec{u} k_{d,F}(\vec{r}; \vec{u}|\vec{u}'; \nu; t) \quad (1.36a)$$

$$k_{d,F}(\vec{r}; \vec{u}|\vec{u}'; \nu; t) = k_d(\vec{r}; \vec{u}'; \nu; t) \Phi(\vec{r}; \vec{u}|\vec{u}'; \nu; t) \quad (1.36b)$$

si l'intégrale définissant k_d est bien définie. On définit alors aussi le coefficient d'extinction $k_t = k_a + k_d$. On note alors :

$$C_T(f_\nu)(\vec{r}; \vec{u}; \nu; t) = -k_t(\vec{r}; \vec{u}; \nu; t) f_\nu(\vec{r}; \vec{u}; \nu; t) + k_a(\vec{r}; \vec{u}; \nu; t) f_{\nu \text{ eq}}(T(\vec{r}; t); \nu) + \int_{4\pi} d\vec{u}' k_d(\vec{r}; \vec{u}'; \nu; t) \Phi(\vec{r}; \vec{u}|\vec{u}'; \nu; t) f_\nu(\vec{r}; \vec{u}'; \nu; t) \quad (1.37)$$

●●●● Le 2^e principe de la thermodynamique pose des contraintes sur la fréquence des événements de collisions, qu'on appelle micro-réversibilité (ce n'est pas la même micro-réversibilité qu'en cinétique des gaz). On a, à tout instant t et pour tout $(\vec{r}; \vec{u}; \nu) \in E_p$:

$$k_t(\vec{r}; -\vec{u}; \nu; t) = k_t(\vec{r}; \vec{u}; \nu; t) \quad \text{et} \quad k_{d,F}(\vec{r}; -\vec{u}|\vec{u}'; \nu; t) = k_{d,F}(\vec{r}; \vec{u}'|\vec{u}; \nu; t) \quad (1.38)$$

Cette *micro-réversibilité* est une formulation équivalente de la *réversibilité des chemins optiques*. Celle-ci affirme que, dans une situation stationnaire, la probabilité d'un photon de se déplacer d'un point à un autre de l'espace des phases, est égale à celle d'un autre photon d'effectuer le trajet inverse.

Milieu isotrope : C'est une propriété fréquente des milieux semi-transparents. Leur influence sur la lumière ne dégage pas de direction particulière. L'équation de transfert radiatif se simplifie alors :

$$\begin{aligned} \frac{1}{c} \partial_t f_\nu(\vec{r}; \vec{u}; \nu; t) + \vec{u} \cdot \overrightarrow{\text{grad}}_{\vec{r}} f_\nu(\vec{r}; \vec{u}; \nu; t) = \\ -k_t(\vec{r}; \nu; t) f_\nu(\vec{r}; \vec{u}; \nu; t) + k_a(\vec{r}; \nu; t) f_{\nu \text{ eq}}(T(\vec{r}; t); \nu) + \\ k_d(\vec{r}; \nu; t) \int_{4\pi} d\vec{u}' \Phi(\vec{r}; \vec{u} \cdot \vec{u}'; \nu; t) f_\nu(\vec{r}; \vec{u}'; \nu; t) \end{aligned} \quad (1.39)$$

1.2 Résolution des équations de Boltzmann le long d'un chemin balistique

1.2.1 Forme intégrale de l'équation de transfert radiatif

●●●● Dans la section précédente 1.1.4, on donne l'équation de transfert radiatif (ETR) sous *forme différentielle*, c'est-à-dire sous la forme d'une équation aux dérivées partielles. Pour une utilisation dans certaines méthodes numériques, comme la méthode de Monte-Carlo ou la méthode de lancer de rayons [86], il est plus commode d'utiliser une écriture strictement équivalente, appelée *forme intégrale de l'ETR*.

Cette écriture intégrale s'obtient en résolvant l'ETR sous forme différentielle, en remontant un rayon lumineux.

●●●● En transfert radiatif, quand l'indice de réfraction est constant, les photons, entre 2 collisions successives, suivent à vitesse constante des trajectoires rectilignes.

Si on note \vec{r}_b et \vec{u}_b la position et la direction en fonction du temps d'un photon particulier, de position \vec{r} et de direction \vec{u} à l'instant t , alors tant qu'il n'interagit pas avec la matière :

$$\begin{cases} \partial_{t'} \vec{r}_b(t') = c \vec{u}_b(t') \\ \partial_{t'} \vec{u}_b(t') = \vec{0} \end{cases} \quad (1.40)$$

La solution de ce système est immédiate :

$$\begin{cases} \vec{r}_b(t') = \vec{r} + c(t' - t)\vec{u} \\ \vec{u}_b(t') = \vec{u} \end{cases} \quad (1.41)$$

Remarque. On a passé sous silence la fréquence du photon, car dans les hypothèses que nous avons faites celle-ci est constante.

Dérivons la fonction de distribution en suivant le photon :

$$\begin{aligned} & \frac{1}{c} \partial_{t'} f_\nu(\vec{r}_b(t'); \vec{u}_b(t'); \nu; t') \\ &= \frac{1}{c} \overrightarrow{\text{grad}}_1 f_\nu(\vec{r}_b(t'); \vec{u}_b(t'); \nu; t') \cdot \partial_{t'} \vec{r}_b(t') + \\ & \quad \frac{1}{c} \overrightarrow{\text{grad}}_2 f_\nu(\vec{r}_b(t'); \vec{u}_b(t'); \nu; t') \cdot \partial_{t'} \vec{u}_b(t') + \\ & \quad \frac{1}{c} \partial_4 f_\nu(\vec{r}_b(t'); \vec{u}_b(t'); \nu; t') \\ &= \vec{u}_b(t') \cdot \overrightarrow{\text{grad}}_1 f_\nu(\vec{r}_b(t'); \vec{u}_b(t'); \nu; t') + \\ & \quad \frac{1}{c} \partial_4 f_\nu(\vec{r}_b(t'); \vec{u}_b(t'); \nu; t') \\ &= C_T(f_\nu)(\vec{r}_b(t'); \vec{u}_b(t'); \nu; t) \end{aligned} \quad (1.42)$$

La dérivée en suivant le mouvement de la fonction de distribution égalise l'opérateur de collision. Ce résultat est connu sous le nom de théorème de Liouville.

●●●● La précédente identité (1.42) possède la solution générale :

$$\begin{aligned}
 f_\nu(\vec{r}; \vec{u}; \nu; t) = & \left(\int_0^{s_0} ds \exp\left(-\int_0^s ds' k_t(\vec{r} - s'\vec{u}; \vec{u}; \nu; t - \frac{s'}{c})\right) \times \right. \\
 & \left(k_a(\vec{r} - s\vec{u}; \vec{u}; \nu; t - \frac{s}{c}) f_{\nu \text{ eq}}(T(\vec{r} - s\vec{u}; t - \frac{s}{c}); \nu) + \right. \\
 & \left. \int_{4\pi} d\vec{u}' k_d(\vec{r} - s\vec{u}; \vec{u}'; \nu; t - \frac{s}{c}) \Phi(\vec{r} - s\vec{u}; \vec{u}|\vec{u}'; \nu; t - \frac{s}{c}) f_\nu(\vec{r} - s\vec{u}; \vec{u}'; \nu; t - \frac{s}{c}) \right) \Bigg) + \\
 & \exp\left(-\int_0^{s_0} ds k_t(\vec{r} - s\vec{u}; \vec{u}; \nu; t - \frac{s}{c})\right) f_\nu(\vec{r} - s_0\vec{u}; \vec{u}; \nu; t - \frac{s_0}{c}) \quad (1.43)
 \end{aligned}$$

où la variable muette s est une abscisse de remontée le long d'un rayon lumineux. On pose dans cette écriture qu'elle possède un maximum s_0 , mais il peut ne pas exister (on considère alors $s_0 = +\infty$). Dans la pratique, on considère généralement que $s_0 = s_m$, l'abscisse de la plus récente intersection avec une paroi : $s_m = \min\{s \in \mathbb{R}^+, \vec{r} - s\vec{u} \in \text{parois}\}$.

Une interprétation physique de l'écriture (1.43) peut être la suivante. La lumiance en $s = 0$ est la somme des contributions de tous les points du rayon arrivant en $(\vec{r}; \vec{u}; \nu)$, chacune de ces contributions étant atténuée par le milieu entre son point d'entrée et le point d'observation. Les contributions sont par hypothèse de 2 types : émission thermique et diffusion entrante. De même, l'atténuation est due à l'absorption et la diffusion sortante.

1.2.2 Forme intégrale de l'équation de Boltzmann

La démarche de la section précédente (1.2.1) peut être appliquée sur l'équation de Boltzmann, même si l'écriture résultante est usuellement moins utilisée que son pendant avec l'ETR.

Si on note \vec{r}_b et \vec{c}_b la position et la vitesse en fonction du temps d'une molécule particulière, de position \vec{r} et de vitesse \vec{c} à l'instant t , alors tant qu'elle ne subit aucune collision :

$$\begin{cases} \partial_{t'} \vec{r}_b(t') = \vec{c}_b(t') \\ \partial_{t'} \vec{c}_b(t') = \vec{a}(\vec{r}_b(t'); \vec{c}_b(t'); t') \end{cases} \quad (1.44)$$

où $\vec{a} = \frac{\vec{F}}{m}$ est l'accélération subie par la molécule, due aux forces à longue portée. Selon l'expression de \vec{a} , la résolution de ce système peut être difficile.

En dérivant la fonction de distribution en suivant la molécule :

$$\begin{aligned}
 & \partial_{t'} f(\vec{r}_b(t'); \vec{c}_b(t'); t') \\
 &= \overrightarrow{\text{grad}}_1 f(\vec{r}_b(t'); \vec{c}_b(t'); t') \cdot \partial_{t'} \vec{r}_b(t') + \\
 & \quad \overrightarrow{\text{grad}}_2 f(\vec{r}_b(t'); \vec{c}_b(t'); t') \cdot \partial_{t'} \vec{c}_b(t') + \\
 & \quad \partial_3 f(\vec{r}_b(t'); \vec{c}_b(t'); t') \\
 &= \vec{c}_b(t') \cdot \overrightarrow{\text{grad}}_1 f(\vec{r}_b(t'); \vec{c}_b(t'); t') + \\
 & \quad \vec{a}(\vec{r}_b(t'); \vec{c}_b(t'); t') \cdot \overrightarrow{\text{grad}}_2 f(\vec{r}_b(t'); \vec{c}_b(t'); t') + \\
 & \quad \partial_3 f(\vec{r}_b(t'); \vec{c}_b(t'); t') \\
 &= \vec{c}_b(t') \cdot \overrightarrow{\text{grad}}_1 f(\vec{r}_b(t'); \vec{c}_b(t'); t') + \\
 & \quad \text{div}_2 (f \times \vec{a})(\vec{r}_b(t'); \vec{c}_b(t'); t') - \\
 & \quad \left(\text{div}_2 \vec{a}(\vec{r}_b(t'); \vec{c}_b(t'); t') \right) f(\vec{r}_b(t'); \vec{c}_b(t'); t') + \\
 & \quad \partial_3 f(\vec{r}_b(t'); \vec{c}_b(t'); t') \\
 &= - \left(\text{div}_2 \vec{a}(\vec{r}_b(t'); \vec{c}_b(t'); t') \right) f(\vec{r}_b(t'); \vec{c}_b(t'); t') + C_B(f)(\vec{r}; \vec{c}; t) \tag{1.45}
 \end{aligned}$$

Cette précédente expression possède la solution générale :

$$\begin{aligned}
 f(\vec{r}; \vec{c}; t) &= \left(\int_{t_0}^t dt' \exp \left(- \int_{t'}^t dt'' \int_{E_c} d\vec{c}_* b_T(\|\vec{c}_b(t'') - \vec{c}_*\|) f(\vec{r}_b(t''); \vec{c}_*; t'') \right) \times \right. \\
 & \quad \exp \left(- \int_{t'}^t dt'' \text{div}_2 \vec{a}(\vec{r}_b(t''); \vec{c}_b(t''); t'') \right) \int_{E_c} d\vec{c}_* \int_{4\pi} d\vec{u}' \times \\
 & \quad \left. b_F \left(\|\vec{c}_b(t') - \vec{c}_*\|; \vec{u}' \cdot \frac{\vec{c}_b(t') - \vec{c}_*}{\|\vec{c}_b(t') - \vec{c}_*\|} \right) f(\vec{r}_b(t'); \vec{c}'; t') f(\vec{r}_b(t'); \vec{c}_*; t') \right) + \\
 & \quad \exp \left(- \int_{t_0}^t dt' \int_{E_c} d\vec{c}_* b_T(\|\vec{c}_b(t') - \vec{c}_*\|) f(\vec{r}_b(t'); \vec{c}_*; t') \right) \times \\
 & \quad \exp \left(- \int_{t_0}^t dt' \text{div}_2 \vec{a}(\vec{r}_b(t'); \vec{c}_b(t'); t') \right) f(\vec{r}_b(t_0); \vec{c}_b(t_0); t_0) \tag{1.46a}
 \end{aligned}$$

où

$$\vec{c}' = \frac{\vec{c}_b(t') + \vec{c}_*}{2} + \frac{\|\vec{c}_b(t') - \vec{c}_*\|}{2} \vec{u}' \quad \text{et} \quad \vec{c}_*' = \frac{\vec{c}_b(t') + \vec{c}_*}{2} - \frac{\|\vec{c}_b(t') - \vec{c}_*\|}{2} \vec{u}' \tag{1.46b}$$

Par commodité, nous avons choisi comme variable muette un temps t' , correspondant à une date d'entrée des particules dans la trajectoire balistique suivie. Pour sa borne inférieure, on choisirait naturellement $t_0 = t_m$ avec $t_m = \max\{t' \in] - \infty; t], \vec{r}_b(t') \in \text{parois}\}$.

Si le champ des forces à longue portée ne diverge pas dans l'espace des vitesses, l'expression (1.46a) se simplifie en :

$$\begin{aligned}
 f(\vec{r}; \vec{c}; t) = & \left(\int_{t_0}^t dt' \exp\left(-\int_{t'}^t dt'' \int_{E_c} d\vec{c}_* b_T(\|\vec{c}_b(t'') - \vec{c}_*\|) f(\vec{r}_b(t''); \vec{c}_*; t'') \right) \times \right. \\
 & \left. \int_{E_c} d\vec{c}_* \int_{4\pi} d\vec{u}' b_F\left(\|\vec{c}_b(t') - \vec{c}_*\|; \vec{u}' \cdot \frac{\vec{c}_b(t') - \vec{c}_*}{\|\vec{c}_b(t') - \vec{c}_*\|}\right) f(\vec{r}_b(t'); \vec{c}'; t') f(\vec{r}_b(t'); \vec{c}_*; t') \right) + \\
 & \exp\left(-\int_{t_0}^t dt' \int_{E_c} d\vec{c}_* b_T(\|\vec{c}_b(t') - \vec{c}_*\|) f(\vec{r}_b(t'); \vec{c}_*; t') \right) f(\vec{r}_b(t_0); \vec{c}_b(t_0); t_0) \quad (1.47)
 \end{aligned}$$

1.2.3 Conditions aux frontières

●●●● Jusqu'ici dans ce chapitre, nous avons considéré la dynamique des corpuscules (molécules ou photons) uniquement en volume, c'est-à-dire dans le volume du gaz ou dans un milieu semi-transparent. Nous y avons donné des équations régissant la fonction de distribution (des molécules ou des photons). Dans la plupart des problèmes concrets, il faut aussi tenir compte de la présence de parois, et plus généralement on doit disposer de conditions aux frontières.

Cette section détaille quelques conditions aux frontières couramment utilisées en cinétique des gaz et en transfert radiatif. Nous n'entrons pas dans le détail de leur justification, ou de la pertinence de leur utilisation pour tel ou tel problème physique. Par ailleurs, quand nous donnons des conditions aux frontières applicables sur des parois matérielles, nous considérons ces dernières systématiquement immobiles.

Du point de vue mathématique, il est intéressant de constater que ces conditions aux frontières s'expriment directement sous la forme d'équations intégrales, contrairement aux équations de Boltzmann en volume qui, pour nous, ont pris de prime abord des expressions intégrodifférentielles.

Nous considérons à présent que le système à l'étude occupe un volume \mathcal{M} , délimité par une frontière $\partial\mathcal{M}$. Cette frontière possède une normale en presque tous ses points, notée $\vec{n}_{\partial\mathcal{M}}(\vec{r})$. Cette normale est choisie sortante des parois, c'est-à-dire dirigée vers l'intérieur de \mathcal{M} .

1.2.3.1 En transfert radiatif

●●●● La condition frontière la plus simple est certainement celle qui consiste à imposer la luminance entrante. Cela s'écrit :

$$\forall \vec{u} \in E_u \text{ tq } \vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) \geq 0, f_\nu(\vec{r}; \vec{u}; \nu; t) \text{ est connue} \quad (1.48)$$

●●●● Sur une paroi opaque, en équilibre thermodynamique local à la température $T(\vec{r}; t)$, la condition frontière s'écrit en toute généralité [59] :

$$\begin{aligned} \forall \vec{u} \in E_u \text{ tq } \vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) \geq 0, \\ f_\nu(\vec{r}; \vec{u}; \nu; t) = \varepsilon(\vec{r}; \vec{u}; \nu; t) f_{\nu \text{ eq}}(T(\vec{r}; t); \nu) + \\ \int_{2\pi^+} d\vec{u}' (\vec{u}' \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r})) (1 - \varepsilon(\vec{r}; \vec{u}'; \nu; t)) \Phi_p(\vec{r}; \vec{u} | - \vec{u}'; \nu; t) f_\nu(\vec{r}; -\vec{u}'; \nu; t) \end{aligned} \quad (1.49)$$

où :

- $\int_{2\pi^+} d\vec{u}'$ signifie qu'on intègre sur l'hémisphère des directions \vec{u}' sortantes de la paroi, c'est-à-dire telles que $\vec{u}' \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) \geq 0$,
- ε est l'émissivité de la paroi,
- $\Phi_p(\vec{r}; \vec{u} | \vec{u}'; \nu; t)$ est la densité de probabilité d'un photon de fréquence ν et arrivant en \vec{r} à l'instant t selon la direction \vec{u}' d'être réfléchi dans la direction \vec{u} , sachant qu'il est réfléchi.

Cette condition frontière est le pendant sur une surface de l'ETR dans un milieu semi-transparent. Φ_p est ainsi l'équivalent surfacique de la fonction de phase de diffusion. Par ailleurs :

- ε est à valeurs comprises entre 0 et 1.
- La micro-réversibilité s'exprime :

$$\begin{aligned} (\vec{u}' \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r})) (1 - \varepsilon(\vec{r}; \vec{u}'; \nu; t)) \Phi_p(\vec{r}; \vec{u} | - \vec{u}'; \nu; t) = \\ (\vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r})) (1 - \varepsilon(\vec{r}; \vec{u}; \nu; t)) \Phi_p(\vec{r}; \vec{u}' | - \vec{u}; \nu; t) \end{aligned} \quad (1.50)$$

●●●● La relation (1.49) admet de nombreux cas particuliers. On peut citer :

le corps noir : $\varepsilon = 1$, et donc :

$$\forall \vec{u} \in E_u \text{ tq } \vec{u} \cdot \vec{n}_{\partial\mathcal{M}} \geq 0, f_\nu(\vec{r}; \vec{u}; \nu; t) = f_{\nu \text{ eq}}(T(\vec{r}; t); \nu)$$

Si la paroi est froide ($T(\vec{r}; t) = 0 \text{ K}$) :

$$\forall \vec{u} \in E_u \text{ tq } \vec{u} \cdot \vec{n}_{\partial\mathcal{M}} \geq 0, f_\nu(\vec{r}; \vec{u}; \nu; t) = 0$$

les corps gris : ε et Φ_p ne dépendent pas de ν .

la réflexion diffuse : elle est l'équivalent surfacique de la diffusion avec fonction de phase isotrope :

$$\begin{aligned} \forall \vec{u} \in E_u \text{ tq } \vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) \geq 0, f_\nu(\vec{r}; \vec{u}; \nu; t) = \\ \varepsilon(\vec{r}; \nu; t) f_{\nu \text{ eq}}(T(\vec{r}; t); \nu) + \int_{2\pi^-} d\vec{u}' \frac{-\vec{u}' \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r})}{\pi} (1 - \varepsilon(\vec{r}; \nu; t)) f_\nu(\vec{r}; \vec{u}'; \nu; t) \end{aligned}$$

1.2.3.2 En cinétique des gaz

●●●● Comme en transfert radiatif, la condition frontière la plus simple est d'imposer la distribution des particules entrantes.

Dans ce cas, on dit couramment que le milieu est connecté à un « réservoir », quand la distribution à l'entrée est une distribution d'équilibre, à densité, vitesse moyenne et température connues.

●●●● Pour tenir compte d'une paroi étanche, plusieurs conditions aux limites sont envisageables. En voici quelques unes, parmi les plus simples :

réflexion spéculaire :

$$\forall \vec{c} \in E_c \text{ tq } \vec{c} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) \geq 0, \\ f(\vec{r}; \vec{c}; t) = f(\vec{r}; \vec{c}_i; t) \quad \text{avec } \vec{c}_i = \vec{c} - 2(\vec{c} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}))\vec{n}_{\partial\mathcal{M}}(\vec{r})$$

réflexion arrière :

$$\forall \vec{c} \in E_c \text{ tq } \vec{c} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) \geq 0, f(\vec{r}; \vec{c}; t) = f(\vec{r}; -\vec{c}; t)$$

rebond Maxwellien (avec thermalisation) : Les particules partent de la paroi selon une distribution d'équilibre, à la vitesse moyenne et la température de la paroi. La densité associée à cette distribution sortante est telle que le flux total traversant la paroi soit nul :

$$\forall \vec{c} \in E_c \text{ tq } \vec{c} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) \geq 0, f(\vec{r}; \vec{c}; t) = \frac{j_{in}(\vec{r})}{c_{qma}^2} \times \frac{\exp(-\vec{c}^2 / 2 c_{qma}^2)}{2\pi c_{qma}^2}$$

avec j_{in} la densité de particules arrivant sur la paroi, et c_{qma} la vitesse quadratique moyenne des molécules associée à la température de la paroi :

$$j_{in} = \int_{E_c^-} d\vec{c}' f(\vec{r}; \vec{c}'; t) (-\vec{c}' \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r})) \quad \text{et} \quad c_{qma}^2 = \frac{R T_{\text{paroi}}}{2 M_{\text{gaz}}}$$

où R est la constante des gaz parfaits, M_{gaz} est la masse molaire du gaz, et $\int_{E_c^-} d\vec{c}'$ signifie qu'on intègre sur le demi-espace des vitesses dirigées vers la paroi (telles que $\vec{c}' \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) \leq 0$).

1.3 La méthode de Monte-Carlo

●●●● Dans les différentes disciplines physiques liées au transport linéaire, et en particulier en transfert radiatif, la méthode de Monte-Carlo (MMC) est la plupart du temps considérée comme une reproduction numérique du phénomène de transport. Elle consiste alors à générer l'histoire de particules représentatives dans le système d'étude, en leur faisant subir aléatoirement les événements qu'elles seraient susceptibles de connaître en réalité.

Bien sûr, on s'écarte assez vite de la stricte analogie avec le phénomène physique, dès que l'on veut rendre les calculs plus efficaces. Par exemple, on modifie l'échantillonnage des événements possibles, en compensant par une pondération des particules pour que les grandeurs physiques calculées restent les mêmes. Ou on travaille sur des problèmes physiques reformulés, plus simples à simuler, mais où les résultats sont strictement identiques. Tout ceci est cependant fait en préservant une caractéristique essentielle de la méthode de Monte-Carlo : son côté intuitif.

●●●● Nous n'avons évidemment pas oublié ce caractère intuitif, ni les pratiques existantes et répandues de la méthode de Monte-Carlo, au cours du travail de thèse ici exposé. Néanmoins, en voulant appliquer la méthode de Monte-Carlo en cinétique des gaz, *ce qui est une première à notre connaissance*, et en s'aidant pour cela seulement de quelques astuces mathématiques (exposées au chapitres 2 et 3), il nous sera difficile de construire rapidement un intuitif solide. À cause de ceci, ce sera uniquement dans un formalisme rigoureux et de la théorie statistique exacte que nous resterons en sécurité.

Afin de commencer dans une telle rigueur, la présente section repose complètement les bases de la méthode de Monte-Carlo. Les connaisseurs de la méthode y retrouveront presque exclusivement ce qu'ils savent déjà. Nous avons toutefois jugé nécessaire d'effectuer ce passage en revue théorique, dans le but de déterminer ce que nous pouvons utiliser sans crainte dans le monde du transport non-linéaire.

1.3.1 Les bases de la méthode

L'idée de base de la méthode de Monte-Carlo est de calculer une grandeur, définie comme l'espérance d'une variable aléatoire elle-même issue d'un processus aléatoire, en simulant n fois ce processus, à l'aide d'un générateur de nombres aléatoires. Un ordinateur opère, numériquement, n fois le processus — les n réalisations du processus doivent être indépendantes —, calcule à chaque fois une valeur pour la variable aléatoire, et à la fin dégage la moyenne de toutes les valeurs qu'il a calculé : cette moyenne est une estimation de la grandeur recherchée.

La méthode de Monte-Carlo est très appréciée en physique du transport corpusculaire (qui comprend la neutronique). Le transport corpusculaire est en général lui-même vu comme la résultante observable d'un processus aléatoire s'appliquant à de nombreuses particules (comme expliqué dans la section 1.1). Dans cette optique, la MMC propose de réaliser *in silico* le processus aléatoire sur un échantillon de particules, afin d'estimer directement les observables d'intérêts. Cette approche de la méthode de Monte-Carlo est qualifiée d'*analogue*.

De façon plus générale, la MMC permet de calculer toute expression intégrale, en fournissant une quadrature provenant d'un processus aléatoire arbitrairement introduit. Il n'est alors plus nécessaire d'avoir un processus physique en tête pour réfléchir, la MMC est conçue comme une pure méthode de calcul numérique. Cette approche est qualifiée d'*intégrale*.

Formellement, posons qu'on veut connaître l'espérance q d'une grandeur $w(\vec{X})$ issue d'un processus aléatoire, dont toute instance (c'est-à-dire toute réalisation) est caractérisée par un vecteur d'état \vec{X} . On écrit par convention « $q = E(w(\vec{X}))$ » pour signifier « q est l'espérance de $w(\vec{X})$ », ce qui signifie par définition de l'espérance que :

$$q = \int_{\mathcal{U}_{\vec{X}}} p_{\vec{X}}(\vec{x}) d\vec{x} w(\vec{x}) \quad (1.51)$$

où on note $\mathcal{U}_{\vec{X}}$ l'ensemble des valeurs possibles de \vec{X} (appelé *univers* de \vec{X}) et $p_{\vec{X}}$ la densité de probabilité de \vec{X} . Pour mémoire, $p_{\vec{X}}$ est définie¹⁰ par :

$$\forall \Omega \subset \mathcal{U}_{\vec{X}}, \Pr(\vec{X} \in \Omega) = \int_{\Omega} p_{\vec{X}}(\vec{x}) d\vec{x} \quad (1.52)$$

où \Pr dénote la probabilité.

Remarques.

- Dans les écritures précédentes, \vec{X} et $w(\vec{X})$ sont des variables aléatoires (VA). w est une fonction. On a implicitement posé que $w(\vec{X})$ est la variable aléatoire dont la valeur est l'image par w de la valeur de \vec{X} . Cette convention d'écriture tiendra pour toute variable aléatoire, et pour toute fonction à l'exception de : E , Var , σ , Asym , Kurt , Kurtex , max , et min .
- Autant que possible, dans ce manuscrit nous essaierons de noter les variables aléatoires en lettres capitales.

Définition 1.3.1 (Variance). Soit X une variable aléatoire à valeurs réelles. La variance de X , notée $\text{Var}(X)$, est définie par :

$$\text{Var}(X) = E((X - E(X))^2)$$

si cette espérance est bien définie ; sinon $\text{Var}(X) = +\infty$.

Théorème 1.3.2 (Formule de König-Huygens). Soit X une variable aléatoire à valeurs réelles, de variance finie. Alors les espérances de X et de X^2 sont bien définies, et $\text{Var}(X) = E(X^2) - E(X)^2$.

Théorème 1.3.3 (Loi faible des grands nombres). Soit $(X_n)_{n \in \mathbb{N}^*}$ une suite de variables aléatoires indépendantes de même univers inclus dans \mathbb{R} , ayant la même espérance $E(X)$ et la même variance finies. On définit, pour tout $n \in \mathbb{N}^*$, $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$. Alors :

$$\forall \varepsilon > 0, \lim_{n \rightarrow +\infty} \Pr(|\bar{X}_n - E(X)| \geq \varepsilon) = 0$$

10. $p_{\vec{X}}$ n'est pas, de façon générale, une fonction mais une *distribution*. Un exemple de distribution qui ne s'identifie pas à une fonction est la distribution de Dirac. Les distributions sont décrites dans une axiomatique appelée théorie des distributions.

Théorème 1.3.4 (Loi forte des grands nombres). Soit $(X_n)_{n \in \mathbb{N}^*}$ une suite de variables aléatoires indépendantes, à valeurs réelles, identiquement distribuées à X (c'est-à-dire de même univers et de même densité de probabilité), et intégrables (ce qui signifie $E(|X|) < +\infty$). On définit, pour tout $n \in \mathbb{N}^*$, $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$. Alors :

$$\Pr\left(\lim_{n \rightarrow +\infty} \bar{X}_n = E(X)\right) = 1$$

Les lois des grands nombres légitiment la méthode de Monte-Carlo. En échantillonnant de manière répétitive et indépendante une variable aléatoire, on peut estimer son espérance par la moyenne arithmétique. Un deuxième résultat mathématique fondamental envers la MMC est le théorème central limite.

Théorème 1.3.5 (Théorème central limite). Soit $(X_n)_{n \in \mathbb{N}^*}$ une suite de variables aléatoires indépendantes, à valeurs réelles, identiquement distribuées à X , avec X d'espérance et de variance finie. On définit, pour tout $n \in \mathbb{N}^*$, $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$ puis $Z_n = \sqrt{n/\text{Var}(X)} \times (\bar{X}_n - E(X))$.

Alors la suite des variables aléatoires $(Z_n)_{n \in \mathbb{N}^*}$ converge en loi vers la loi normale centrée réduite.

Remarque. Un résultat intermédiaire du théorème central limite est que, avec les mêmes notations : $\text{Var}(\bar{X}_n) = \frac{1}{n} \text{Var}(X)$. Ceci est en fait une conséquence presque immédiate des propriétés de la variance.

Théorème-Définition 1.3.6 (Variance empirique échantillon). Soit $(x_i)_{i \in \llbracket 1;n \rrbracket}$, où $n \in \mathbb{N}$, une famille de réalisations indépendantes de la variable aléatoire X à valeurs réelles. On définit la variance empirique s^2 de l'échantillon $(x_i)_{i \in \llbracket 1;n \rrbracket}$ par :

$$s^2((x_i)_{i \in \llbracket 1;n \rrbracket}) = \frac{1}{n-1} \sum_{i=1}^n \left(x_i - \frac{1}{n} \sum_{i=1}^n x_i\right)^2 = \frac{1}{n-1} \left(\left(\sum_{i=1}^n x_i^2\right) - \frac{1}{n} \left(\sum_{i=1}^n x_i\right)^2 \right)$$

$s^2((x_i)_{i \in \llbracket 1;n \rrbracket})$ est une estimation non-biaisée, et convergente quand $n \rightarrow +\infty$, de $\text{Var}(X)$.

Le théorème central limite et la formule de la variance empirique donnent à la MMC une de ses propriétés essentielles : elle permet d'estimer l'erreur de calcul qu'elle commet, et de donner son résultat sous forme d'intervalles de confiance, en même temps qu'elle effectue le calcul. De plus, les données à retenir entre les réalisations sont minimales : il faut juste mettre à jour une somme des réalisations, et une somme des carrés des réalisations.

●●●● Cet aspect pratique serait néanmoins de peu d'intérêt, si on ne savait pas échantillonner des variables aléatoires quelconques. Pour ce faire, on s'aide notamment de :

— la propriété suivante 1.3.7,

— la méthode de rejet.

Théorème-Définition 1.3.7 (Fonction de répartition). Soit X une variable aléatoire à valeurs réelles. On définit la fonction de répartition F_X de X par :

$$\begin{aligned} \mathbb{R} &\mapsto [0; 1] \\ F_X(x) : \quad x &\rightarrow \Pr(X \leq x) = \int_{]-\infty; x]} p_X(x') dx' \end{aligned} \quad (1.53)$$

On définit la pseudo-inverse F_X^{-1} de F_X par :

$$\begin{aligned} F_X^{-1}(r) : \quad [0; 1] &\mapsto \mathbb{R} \\ r &\rightarrow \inf\{x \in \mathbb{R} \text{ tq } F_X(x) \geq r\} \end{aligned} \quad (1.54)$$

Alors si R est de loi uniforme standard (c'est-à-dire de loi uniforme sur $]0; 1[$), $F_X^{-1}(R)$ est distribuée identiquement à X .

Les générateurs aléatoires disponibles sur ordinateur (en fait, ce sont presque toujours des générateurs pseudo-aléatoires, délivrant en une boucle très longue une séquence de nombres aux propriétés statistiques satisfaisantes) délivrent tous des valeurs entre 0 et 1. En appliquant à de telles valeurs la réciproque de la primitive de p_X centrée en $-\infty$, on échantillonne la variable aléatoire X .

Pour échantillonner une variable aléatoire \vec{X} à valeurs vectorielles, on tire ses coordonnées les unes après les autres.

●●● Si le théorème 1.3.7 ne donne pas immédiatement satisfaction pour échantillonner une variable aléatoire \vec{X} , une astuce disponible est la méthode de rejet.

L'idée est d'utiliser une variable aléatoire \vec{Y} dont l'univers $\mathcal{U}_{\vec{Y}}$ inclut celui de \vec{X} , et dont la densité de probabilité $p_{\vec{Y}}$ est proportionnelle à $p_{\vec{X}}$ sur $\mathcal{U}_{\vec{X}}$. Alors pour échantillonner \vec{X} , on échantillonne \vec{Y} — nous supposons qu'on sait le faire — jusqu'à trouver une valeur dans $\mathcal{U}_{\vec{X}}$. Pour que cela fonctionne, il faut que $\Pr(\vec{Y} \in \mathcal{U}_{\vec{X}}) > 0$.

Cette technique prend toute sa puissance en ajoutant une dimension à l'univers de \vec{X} .

Théorème 1.3.8 (Méthode de rejet). Considérons une variable aléatoire \vec{Y} recouvrant \vec{X} , ce qui signifie que :

$$\forall \Omega \subset \mathcal{U}_{\vec{X}}, \Pr(\vec{X} \in \Omega) > 0 \implies \Pr(\vec{Y} \in \Omega) > 0$$

On suppose qu'on sait échantillonner \vec{Y} , et aussi que le rapport $p_{\vec{X}}/p_{\vec{Y}}$ est majoré par $\alpha \in \mathbb{R}^{++}$ connu :

$$\forall \Omega \subset \mathcal{U}_{\vec{X}}, \Pr(\vec{X} \in \Omega) \leq \alpha \times \Pr(\vec{Y} \in \Omega)$$

Posons les variables aléatoires :

- $\vec{X}' = (\vec{X}; R_{\vec{X}})$, où $R_{\vec{X}}$ est indépendante de \vec{X} et de loi uniforme standard,
- $\vec{Y}' = (\vec{Y}; R_{\vec{Y}})$, où $R_{\vec{Y}}$ est de loi uniforme sur $]0; \frac{\alpha p_{\vec{Y}}(\vec{Y})}{p_{\vec{X}}(\vec{Y})}[$.

Alors $\alpha \geq 1$, $\mathcal{U}_{\vec{X}'} \subset \mathcal{U}_{\vec{Y}'}$, et $p_{\vec{X}'}$ est proportionnelle à $p_{\vec{Y}'}$. On peut donc échantillonner \vec{X}' , et donc \vec{X} , par une méthode de rejet, détaillée en algorithme 1.1.

Sorties : Un échantillon ponctuel \vec{x} de \vec{X}

1 répéter

2 | échantillonner \vec{Y} : on obtient \vec{y} ;

3 | échantillonner R de loi uniforme standard : on obtient r ;

4 jusqu'à $r \propto p_{\vec{Y}}(\vec{y}) < p_{\vec{X}}(\vec{y})$;

5 retourner \vec{y} ; // \vec{y} remplace \vec{x}

Algorithme 1.1 : Algorithme servant à échantillonner \vec{X} , par méthode de rejet, dans les hypothèses de la proposition 1.3.8.

●●●● Malgré ce qui est précédemment expliqué sur :

- la convergence de la moyenne arithmétique d'un échantillon de réalisations indépendantes,
- les moyens d'effectuer de telles réalisations, des points de vue mathématique et informatique,

on peut se demander quel est l'intérêt d'utiliser une « quadrature aléatoire » pour estimer $q = \int_{\mathcal{U}_{\vec{X}}} d\vec{x} p_{\vec{X}}(\vec{x}) w(\vec{x})$, plutôt qu'une quadrature régulière (comme dans la méthode des rectangles centrés, par exemple).

Cet intérêt apparaît quand $\mathcal{U}_{\vec{X}}$ est de dimension élevée. Une quadrature discrète doit comporter un nombre certain de coordonnées d'échantillonnage dans chaque dimension, sinon elle peut générer des erreurs systématiques par repli de spectre (ce phénomène s'appelle moiré en synthèse d'image). Si l'espace échantillonné contient beaucoup de dimensions, les points de quadrature peuvent devenir extrêmement nombreux, ce qui amène à un temps de calcul prohibitif.

Par ailleurs, les exigences sur une quadrature statistique de type Monte-Carlo et sur une quadrature régulière ne sont pas les mêmes quand des fortes difficultés de convergence sont attendues dans plusieurs dimensions d'intégration. Supposons que l'on n'ait que 2 dimensions d'intégration, devant être échantillonnées chacune en N_1 et N_2 points. Une quadrature régulière devra comporter $N_1 \times N_2$ points; à moins que l'on sache comment en enlever une partie, ce qui est un exercice mathématique laborieux. Une quadrature Monte-Carlo comportera $\max\{N_1; N_2\}$ points; car quand on ajoute une dimension, la coordonnée ajoutée à chacun des points possède une unique valeur aléatoire par point, et non pas une liste de valeurs (fixes). On dit qu'en MMC, les difficultés de convergence propres à chaque dimension d'intégration ne se multiplient pas. En contrepartie, N_1 et N_2 doivent de base être grands avec une quadrature Monte-Carlo, car sinon elle risque d'échantillonner insuffisamment certaines régions du domaine d'intégration, à cause de son irrégularité.

Tout cela fait que la MMC et sa quadrature deviennent performantes, comparées aux quadratures régulières, quand le domaine d'intégration est de dimension élevée (plus de 5 dimensions est la valeur habituellement retenue).

Un dernier avantage des quadratures statistiques est qu'elles sont sans biais. C'est-à-dire que si on recommence le même calcul avec des tirages aléatoires différents, les points de quadratures sont entièrement renouvelés, et l'erreur produite est différente. Les quadratures régulières ont généralement une technique fixe de positionnement associée : à moins de raffiner franchement la quadrature, l'erreur générée est donc systématique.

1.3.2 Approche analogue et approche intégrale, et concept de variance nulle

En physique du transport corpusculaire, il existe 2 façons complémentaires de concevoir la méthode de Monte-Carlo : l'approche analogue et l'approche intégrale, évoquées dans la section précédente 1.3.1.

●●●● La base d'une approche analogue de la MMC est le constat que la description physique du transport corpusculaire est une description statistique.

En transfert radiatif par exemple, un photon dans un milieu semi-transparent a une probabilité donnée d'être absorbé ou diffusé à chaque élément de longueur qu'il parcourt. S'il est diffusé ou s'il rencontre une paroi, sa probabilité d'acquiescer telle ou telle direction décrit la façon dont le flux lumineux est redistribué dans l'espace des directions.

L'approche analogue propose de mimer le phénomène physique, à l'aide d'un ordinateur et de son générateur aléatoire. On suit des particules fictives de lumière, indépendantes, dans le système physique, en générant leur parcours en tenant compte des probabilités d'occurrence des événements possibles (absorption, diffusion, réflexion, ...). Les grandeurs recherchées sont calculées comme des sommes de contributions des différents photons suivis¹¹.

En approche intégrale, on suppose que la grandeur d'intérêt q a été mise sous forme intégrale, lors d'une phase de physique et de mathématiques préalable au travail numérique. On a alors : $q = \int_{\mathcal{D}} d\vec{x} h(\vec{x})$. On introduit ensuite, arbitrairement, une variable aléatoire \vec{X} à valeurs dans \mathcal{D} . La densité de probabilité de \vec{X} doit recouvrir h sur \mathcal{D} au sens des distributions :

$$\forall \Omega \subset \mathcal{D}, \int_{\Omega} |h(\vec{x})| d\vec{x} > 0 \implies \int_{\Omega} p_{\vec{X}}(\vec{x}) d\vec{x} > 0 \quad (1.55)$$

11. Certains auteurs font remarquer le barbarisme qu'est l'appellation de « photon » pour ces particules que l'on simule (par exemple [96]). Les photons sont un concept de physique quantique, descriptibles à l'aide de fonctions d'onde à laquelle est liée une probabilité de présence ; or nous les envisageons sous un jour purement corpusculaire. Pire encore, ce qu'ils transportent n'est pas l'énergie $h\nu$, et n'est bien souvent même pas une énergie... Toutefois, par commodité et par habitude, nous emploierons le terme de « photon » pour cet usage.

Dès lors :

$$q = \int_{\mathcal{D}} d\vec{x} h(\vec{x}) \quad (1.56)$$

$$\begin{aligned} &= \int_{\mathcal{D}} p_{\vec{X}}(\vec{x}) d\vec{x} \frac{h(\vec{x})}{p_{\vec{X}}(\vec{x})} \\ &= \int_{\mathcal{D}} p_{\vec{X}}(\vec{x}) d\vec{x} w(\vec{x}) \quad \text{avec } w(\vec{x}) = \frac{h(\vec{x})}{p_{\vec{X}}(\vec{x})} \\ &= E(w(\vec{X})) \end{aligned} \quad (1.57)$$

La même démarche est applicable à une somme dénombrable. Soit $q = \sum_{n=0}^{+\infty} h_n$ la grandeur que l'on veut calculer. On pose N une variable aléatoire à valeurs dans \mathbb{N} , telle que $\forall n \in \mathbb{N}, h_n \neq 0 \implies \Pr(N = n) > 0$. Alors :

$$q = \sum_{n=0}^{+\infty} h_n \quad (1.58)$$

$$\begin{aligned} &= \sum_{n=0}^{+\infty} \Pr(N = n) \frac{h_n}{\Pr(N = n)} \\ &= \sum_{n=0}^{+\infty} \Pr(N = n) w_n \quad \text{avec } w_n = \frac{h_n}{\Pr(N = n)} \\ &= E(w_N) \end{aligned} \quad (1.59)$$

●●●● Nous allons maintenant présenter sur 2 exemples la mise en œuvre de la méthode de Monte-Carlo. Ces exemples — certes extrêmement simplifiés, et présents dans la plupart des manuels consacrés à la MMC — seront l'occasion d'illustrer les approches analogues et intégrales expliquées au-dessus, ainsi que leur complémentarité.

Le premier exemple est celui du calcul de l'aire d'un étang. L'approche sera très mathématique ; nous utiliserons peu d'intuitif, ou bien nous en créerons un de manière clairement artificielle.

Le deuxième exemple est celui du calcul de la transmittivité au travers d'une colonne de gaz uniforme. Nous l'attaquerons par une approche purement analogue : la MMC se réduira alors dans un premier temps à construire des trajectoire de photons, en simulant leur comportement statistique. Nous prendrons ensuite le temps de donner une formulation intégrale correspondant à l'algorithme monté. Ceci nous permettra d'introduire le concept de *variance nulle*, que nous ré-utiliserons aux chapitres 5 et 6 dans nos exemples de cinétique des gaz, en le combinant avec la distribution d'équilibre des molécules. Enfin, nous montrerons comment l'utilisation directe de la formulation intégrale de l'ETR aboutit à des algorithmes en apparence éloignés de toute analogie avec le phénomène physique de transfert radiatif.

1.3.2.1 Calcul de l'aire d'un étang

On a un étang à l'anglaise \mathcal{G} , dans lequel est installée une fontaine \mathcal{F} . On veut calculer la surface de l'étang $S_{\mathcal{G}}$, ainsi que la portion $F_{\mathcal{F}}$ de la surface de l'étang couverte par la fontaine. Les géométries de l'étang et de la fontaine sont connues, mais leur complexité rend le calcul de $S_{\mathcal{G}}$ et de $F_{\mathcal{F}}$ malaisé.

●●●● On pose alors un rectangle \mathcal{R} englobant l'étang. Grâce à l'énoncé 1.3.7, échantillonner uniformément un point dans \mathcal{R} est facile. Dès lors, on peut exhiber une expression de la surface $S_{\mathcal{G}}$ comme une intégrale sur \mathcal{R} :

$$S_{\mathcal{G}} = \int_{\mathcal{G}} d\vec{r} \quad (1.60a)$$

$$= \int_{\mathcal{R}} d\vec{r} H(\vec{r} \in \mathcal{G}) \quad (1.60b)$$

En utilisant une VA $\vec{R}_{\mathcal{R}}$ de loi uniforme sur \mathcal{R} , on arrive à une expression statistique de $S_{\mathcal{G}}$ utilisable immédiatement dans un calcul par méthode de Monte-Carlo :

$$S_{\mathcal{G}} = \int_{\mathcal{R}} p_{\vec{R}_{\mathcal{R}}}(\vec{r}) d\vec{r} \frac{H(\vec{r} \in \mathcal{G})}{p_{\vec{R}}(\vec{r})} \quad (1.60c)$$

$$= \int_{\mathcal{R}} p_{\vec{R}_{\mathcal{R}}}(\vec{r}) d\vec{r} H(\vec{r} \in \mathcal{G}) S_{\mathcal{R}} \quad (1.60d)$$

$$= E(H(\vec{R}_{\mathcal{R}} \in \mathcal{G}) S_{\mathcal{R}}) \quad (1.60e)$$

où $S_{\mathcal{R}}$ est la surface du rectangle englobant, facilement calculable.

Penchons-nous maintenant sur la fraction $F_{\mathcal{F}}$. La même démarche mathématique que pour $S_{\mathcal{G}}$ peut lui être appliquée, ce qui mène à :

$$F_{\mathcal{F}} = \int_{\mathcal{F}} d\vec{r} / S_{\mathcal{G}} \quad (1.61a)$$

$$= E(H(\vec{R}_{\mathcal{G}} \in \mathcal{F})) \quad (1.61b)$$

où $\vec{R}_{\mathcal{G}}$ est une variable aléatoire de loi uniforme sur l'étang \mathcal{G} . À cause de la géométrie de l'étang, ce tirage est difficile à réaliser directement ; mais en se basant sur un tirage uniforme sur \mathcal{R} , on peut utiliser une méthode de rejet.

Les précédentes techniques d'estimation sont synthétisées dans les algorithmes 1.2, 1.3, et 1.4.

●●●● Nous avons mis en place des algorithmes de Monte-Carlo, pour estimer $S_{\mathcal{G}}$ et $F_{\mathcal{F}}$, selon une approche purement intégrale. Néanmoins, une image physique analogue peut être rajoutée à peu de frais sur les algorithmes obtenus.

Supposons qu'il pleuve sur l'étang et ses alentours une hauteur d'eau uniforme. La probabilité, pour une goutte d'eau tombée sur le rectangle englobant \mathcal{R} , de tomber dans l'étang \mathcal{G} est égale à $S_{\mathcal{G}}/S_{\mathcal{R}}$. On sait simuler le choix aléatoire d'une

Sorties : Un tirage aléatoire $\vec{r}_{\mathcal{R}}$ uniforme sur \mathcal{R}

- 1 échantillonner R de loi uniforme standard, 2 fois indépendamment : on obtient r_1 et r_2 ;
- 2 **retourner** $A + r_1 \overrightarrow{AB} + r_2 \overrightarrow{AD}$;

Algorithme 1.2 : Algorithme servant à échantillonner un point uniformément sur un rectangle $ABCD$.

Sorties : Une estimation ponctuelle de $S_{\mathcal{G}}$

- 1 échantillonner $\vec{R}_{\mathcal{R}}$ de loi uniforme sur \mathcal{R} , grâce à l'algorithme 1.2 : on obtient $\vec{r}_{\mathcal{R}}$;
- 2 **retourner** $H(\vec{r}_{\mathcal{R}} \in \mathcal{G}) \times S_{\mathcal{R}}$;

Algorithme 1.3 : Algorithme estimant ponctuellement la surface $S_{\mathcal{G}}$, décrite dans la section 1.3.2.

Sorties : Une estimation ponctuelle de $F_{\mathcal{F}}$

- 1 **répéter**
- 2 échantillonner \vec{R} de loi uniforme sur \mathcal{R} , grâce à l'algorithme 1.2 : on obtient \vec{r} ;
- 3 **jusqu'à** $\vec{r} \in \mathcal{G}$;
- 4 **retourner** $H(\vec{r} \in \mathcal{F})$;

Algorithme 1.4 : Algorithme estimant ponctuellement la fraction $F_{\mathcal{F}}$, décrite dans la section 1.3.2.

position de chute sur \mathcal{R} , grâce à l'algorithme 1.2 issu du théorème 1.3.7. En opérant une telle simulation, on peut estimer $S_{\mathcal{G}}/S_{\mathcal{R}}$; comme $S_{\mathcal{R}}$ est aisément calculable, cela revient à estimer $S_{\mathcal{G}}$.

La même image tient pour calculer $F_{\mathcal{F}} = S_{\mathcal{F}}/S_{\mathcal{G}}$. Il faut cependant rajouter que l'on est capable d'échantillonner uniformément sur \mathcal{G} grâce à une méthode de rejet.

1.3.2.2 Calcul d'une transmittivité dans une colonne de gaz uniforme

On a un rayon allant de \vec{r}_d à \vec{r}_f , séparés par une longueur ℓ : $\vec{r}_f = \vec{r}_d + \ell \vec{u}$. Le système est en régime permanent, le milieu traversé est uniforme, et seulement absorbant (pas d'émission, pas de diffusion). On veut connaître la transmittivité τ de \vec{r}_d à \vec{r}_f , c'est-à-dire la portion de l'énergie lumineuse arrivant à atteindre \vec{r}_f depuis \vec{r}_d . En \vec{r}_d , la luminance dans la direction \vec{u} est connue : elle dépend de la fréquence ν .

Posons à présent :

$$L(\vec{r}; \vec{u}) = \int_{E_\nu} d\nu L_\nu(\vec{r}; \vec{u}; \nu) \quad (1.62a)$$

$$L'_\nu(\vec{r}; \nu; \nu) = L_\nu(\vec{r}; \vec{u}; \nu) / L(\vec{r}; \vec{u}) \quad (1.62b)$$

L'_ν caractérise la répartition spectrale de la puissance lumineuse en $(\vec{r}; \vec{u})$.

●●●● On peut proposer une expression statistique de la transmittivité par analogie avec le transport des photons le long du rayon. En l'occurrence :

$$\tau = E\left(H\left(EO/k_a(N) \geq \ell\right)\right) \quad (1.63)$$

où :

- la variable aléatoire N est une fréquence, de densité de probabilité : $p_N : \nu \mapsto L'_\nu(\vec{r}_d; \vec{u}; \nu)$.
- la variable aléatoire EO correspond à une épaisseur optique traversée.
L'épaisseur optique eo le long d'un segment de rayon lumineux $\{\vec{r}_0 + s\vec{u}, s \in [s_a; s_b]\}$ s'exprime de façon générale comme :

$$eo = \int_{s_a}^{s_b} ds k_t(\vec{r}_0 + s\vec{u}; \vec{u}; \nu) \quad (1.64)$$

Elle est l'intégrale de l'atténuation le long d'un chemin optique. Dans le cas à l'étude ici, $eo = k_a(s_b - s_a)$.

Comme la mémoire des photons se limite à :

- le fait qu'ils existent encore,
- leur fréquence, qui reste constante,

pour un photon quelconque, l'épaisseur optique restante à traverser avant la prochaine collision suit toujours une loi exponentielle d'espérance unité. Donc EO suit une loi exponentielle d'espérance unité, c'est-à-dire que $p_{EO} : eo \mapsto H(eo > 0) \times \exp(-eo)$.

Le déroulement d'une estimation ponctuelle de τ basée sur l'expression (1.63) est le suivant. On suit un photon depuis l'entrée dans le rayon. D'abord on lui choisit une fréquence, selon la distribution de l'énergie selon la fréquence à l'entrée du rayon. Puis on tire une longueur avant absorption de ce photon, sachant que l'épaisseur optique qu'il traverse suit une loi exponentielle unité. Si cette longueur l'amène après \vec{r}_f on compte 1 pour la transmittivité, sinon on compte 0.

●●●● La formulation statistique (1.63) de la transmittivité peut être convertie en une formulation intégrale, en utilisant la définition de l'espérance :

$$\tau = \int_{E_c} L'_\nu(\vec{r}_d; \vec{u}; \nu) d\nu \int_{\mathbb{R}^+} \exp(-eo) deo H(eo/k_a(\nu) \geq \ell) \quad (1.65)$$

Cette conversion est bien sûr réalisable à partir de toute formulation statistique. Donc même si on a monté un algorithme de Monte-Carlo en suivant une analogie avec la physique, une traduction intégrale de l'algorithme est toujours possible.

Par ailleurs, comme *a priori* on ne précise pas dans la formulation intégrale quelles sont les variables aléatoires servant à l'échantillonnage¹², à une seule formulation intégrale correspondent une multitude de traductions statistiques. Quand on vient de monter un algorithme de Monte-Carlo en suivant une approche analogue, exhiber sa formulation intégrale donne ainsi un accès direct à l'arsenal habituel des techniques d'optimisation disponible en MMC [38, 54] :

variables de contrôle : on ajoute dans l'expression intégrale une fonction arbitraire à l'intégrande. Cette fonction doit être d'intégrale connue, et choisie de sorte que la variance du nouveau poids échantillonné soit diminuée.

variables antithétiques : à chaque échantillonnage, on calcule non pas un poids mais une moyenne de plusieurs poids de Monte-Carlo, variant en sens opposés.

tirage stratifié : on divise le domaine d'intégration en *strates* où l'intégrande est de valeur peu variante. Le but est de faire diminuer la variance totale d'un facteur supérieur au nombre de strates choisies, pour gagner en efficacité.

suites à discrédance faible (non aléatoires) : elles peuvent être considérées comme un cas extrême de tirage stratifié.

échantillonnage préférentiel : on remplace les variables aléatoires de tirage.
reformulation intégrale complète.

Par exemple, notre expression intégrale précédente de la transmittivité τ peut être simplifiée en :

$$\tau = \int_{E_c} L'_\nu(\vec{r}_d; \vec{u}; \nu) d\nu \exp(-k_a(\nu)\ell) \quad (1.66)$$

ce qui est l'occasion de présenter une expression statistique de τ où le tirage de l'épaisseur optique a disparu :

$$\tau = E(\exp(-k_a(N)\ell)) \quad (1.67)$$

Si l'échantillonnage de N selon $\nu \mapsto L'_\nu(\vec{r}_d; \vec{u}; \nu)$ est trop compliqué, et qu'on ne veut pas utiliser une méthode de rejet, on peut remplacer N par une nouvelle variable aléatoire arbitraire N' . Alors l'expression intégrale (1.66) admet la traduction statistique :

$$\tau = E\left(\frac{L'_\nu(\vec{r}_d; \vec{u}; N')}{p_{N'}(N')} \exp(-k_a(N')\ell)\right) \quad (1.68)$$

12. Dans de nombreux écrits faisant part d'une utilisation de la MMC, et en particulier dans les thèses encadrées dans l'équipe où la présente l'a été, on se sert tout de même de la formulation intégrale pour détailler les algorithmes de Monte-Carlo. Simplement, les variables aléatoires échantillonnées sont précisées en clair dans les formulations intégrales *via* leurs densités de probabilité.

●●●● Quand on introduit dans une expression intégrale de nouvelles lois de tirage, comme c'est fait en passant de l'expression (1.66) à la (1.68), quelque chose qui est utile à garder en tête est l'approche à *variance nulle*. Celle-ci ne constitue pas à proprement parler une technique d'optimisation des algorithmes de Monte-Carlo, mais un guide à suivre quand on met en place une reformulation intégrale ou un échantillonnage préférentiel.

Supposons que l'on connaisse la valeur exacte de τ , et que dans l'expression statistique (1.68) N' ait la densité de probabilité :

$$p_{N'} : \nu \mapsto \frac{L'_\nu(\vec{r}_d; \vec{u}; \nu) \exp(-k_a(N')\ell)}{\tau} \quad (1.69)$$

alors effectuant un seul tirage de N' , l'estimateur de τ décrit en (1.68) vaut exactement τ . Sa variance est nulle.

Bien sûr, un tel estimateur n'est pas utilisable en pratique, car pour le construire il faut déjà disposer du résultat. Il n'y a alors plus aucun intérêt à mettre en place un calcul par MMC. Cependant, si on dispose d'une approximation relativement bonne $h^\approx(\vec{x})$ de $h(\vec{x})$ dans $q = \int_{\mathcal{D}} d\vec{x} h(\vec{x})$, alors il est utile de s'en servir pour choisir la densité de probabilité de la VA \vec{X} qu'on introduira pour mettre en place un calcul par MMC de q . De manière générale, plus le modèle de $h^\approx(\vec{x})$ dont on dispose est bon, plus l'estimateur qu'on construira avec son aide aura une variance faible.

Une condition de validité importante de l'approche à variance nulle décrite ci-dessus, est que l'intégrande approximée $h^\approx(\vec{x})$ doit être de signe constant ; sinon on ne peut pas poser une densité de probabilité qui lui soit proportionnelle. Toutefois, on peut en général contourner cette limitation grâce à une variable de contrôle.

●●●● Plutôt que de suivre une analogie du phénomène physique pour estimer la transmittivité, on pourrait se baser sur la forme intégrale de l'ETR (1.43). La transmittivité s'exprime comme :

$$\tau = \frac{L(\vec{r}_f; \vec{u})}{L(\vec{r}_d; \vec{u})} \quad (1.70)$$

à supposer qu'il n'y a pas d'apport de lumière sur la portion $[\vec{r}_d; \vec{r}_f]$ du rayon suivi, par émission ou par diffusion entrante. À supposer par ailleurs que la luminance présente en \vec{r}_d soit émise thermiquement, le long d'une portion $[\vec{r}_0; \vec{r}_d]$ du rayon suivi, on peut utiliser la formulation intégrale (1.43) de l'ETR :

$$\tau = \frac{1}{L(\vec{r}_d; \vec{u})} \int_{E_\nu} d\nu \int_0^{\vec{r}_f - s\vec{u} = \vec{r}_0} ds k_a(\nu) \exp(-k_a(\nu)s) L_{\nu \text{ eq}}(T(\vec{r}_d - s\vec{u}); \nu) \quad (1.71)$$

En introduisant les variables aléatoires S et N , avec N à valeurs dans E_ν et S à valeurs dans \mathbb{R}^+ de fonction de répartition $F_S : s \mapsto \mathbb{H}(s > 0) \times (1 - \exp(-k_a(\nu)s))$, on obtient :

$$\tau = \mathbb{E} \left(\frac{1}{L(\vec{r}_d; \vec{u}) p_N(N)} \mathbb{H}(S < \|\vec{r}_0 - \vec{r}_f\|) L_{\nu \text{ eq}}(T(\vec{r}_f - s\vec{u}); \nu) \right) \quad (1.72)$$

Contrairement à l'approche analogue, l'approche intégrale ne fournit pas les variables aléatoires paramétrant l'intégrande ; elles doivent être fournies dans un second temps.

Par ailleurs, en utilisant directement les formes intégrales des équations de transport dans une approche intégrale, on aboutit à un algorithme donnant l'impression de suivre des photons en sens inverse de leur parcours. Cela ne se voit pas forcément très bien dans cet exemple, mais se voit mieux quand il y a des diffusions ou réflexions multiples. Il y a là une différence forte avec des algorithmes analogues mimant le phénomène physique, et suivant les photons depuis leur émission jusqu'à leur absorption. Un lien entre les algorithmes analogues suivant les photons directement et les formulations intégrales des équations de transport est proposé en section 1.4.

1.3.3 Événements rares, et indicateurs statistiques avancés

●●●● Précédemment, nous expliquons grâce au théorème central limite 1.3.5 que la méthode de Monte-Carlo donne, en même temps qu'une estimation d'une grandeur q , une estimation de l'erreur de calcul commise dans cette estimation de q . Plus exactement, il s'agit d'une estimation de l'écart-type de l'estimation de q : on prend comme estimation de cet écart-type $s((\tilde{q}_i)_{[1;n]})/\sqrt{n}$, si on a tiré un échantillon $(\tilde{q}_i)_{[1;n]}$ de n réalisations indépendantes de l'estimateur \tilde{Q} de q (à valeurs réelles ici).

Au total, le résultat de la MMC se présente systématiquement sous la forme d'un couple « (estimation de la grandeur ; variance estimée de cette précédente estimation) ». Ceci nous permet de fournir des intervalles de confiance pour q , en supposant que l'on a effectué assez de tirages pour que notre estimation finale de q (la moyenne arithmétique \tilde{q}_n de n réalisations indépendantes de \tilde{Q}) suive une loi normale. Ces intervalles de confiance ne seront toutefois valables que si :

- on estime correctement $\text{Var}(\tilde{Q})$,
- la loi de \tilde{q}_n est effectivement proche d'une loi normale.

En pratique, on connaît souvent une contrainte sur l'estimateur \tilde{Q} que l'on va moyenner qui nous permet de savoir que ces conditions seront remplies. Cette contrainte peut par exemple être que \tilde{Q} a ses valeurs dans un intervalle restreint (assez pour que l'utilisateur puisse en tirer des conclusions).

Malheureusement, nous sommes confrontés, à partir du chapitre 4 à des estimateurs de variance infinie, sans savoir exactement quand ils ont cette caractéristique. On aimerait donc disposer d'un indicateur permettant de détecter, avec une confiance appréciable, cette situation.

Cet indicateur devrait, de façon générale, détecter toute sorte de situations où l'on sait que notre calcul de Monte-Carlo donne des résultats insatisfaisants. Ces

situations font intervenir, systématiquement, ce qu'on appelle des « événements rares ».

Un cas caricatural est certainement celui d'un estimateur dont la valeur est a sauf une fois sur (par exemple) 10^6 où il vaut b . Selon le nombre k de tirages valant b , le résultat d'une MMC tirant cet estimateur 10^6 fois serait :

$$\left(a + 10^{-6}k(b - a); \frac{1 - 10^{-6}k}{1 - 10^{-6}} \times 10^{-12}k(b - a)^2 \right)$$

Sur la moyenne des valeurs possibles pour k , la MMC donne une estimation non-biaisée de l'espérance $a + 10^{-6}(b - a)$ de l'estimateur. Néanmoins chacun des résultats ci-dessus, pour une valeur particulière de k , est clairement inadéquat pour une utilisation dans des intervalles de confiance basés sur la loi normale.

Comme indicateurs permettant de repérer ces événements rares, nous proposons les paramètres de forme, dont en particulier le kurtosis.

Le kurtosis est avec le paramètre d'asymétrie l'un des 2 *paramètres de forme* d'une distribution de probabilité sur \mathbb{R} . Ces 2 paramètres sont définis comme :

$$\text{Asym}(X) = \left(\left(\frac{X - E(X)}{\sigma(X)} \right)^3 \right) \quad \text{et} \quad \text{Kurt}(X) = \left(\left(\frac{X - E(X)}{\sigma(X)} \right)^4 \right) \quad (1.73a)$$

où $\sigma(X) = \sqrt{\text{Var}(X)}$ est l'écart-type de la distribution de X . On dit ainsi que le paramètre d'asymétrie et le kurtosis sont les moments centrés réduits d'ordre 3 et 4 de la distribution de X .

Une ré-écriture couramment employée est le kurtosis d'excès :

$$\text{Kurtex}(X) = \text{Kurt}(X) - 3 \quad (1.73b)$$

Les paramètres de forme décrivent l'aspect d'une distribution de probabilité sur \mathbb{R} , sans tenir compte de son centrage ou de sa largeur — décrits par l'espérance et l'écart-type. Les valeurs de ces paramètres changent en fonction de ce qu'on appelle couramment le type de la distribution : loi de Bernoulli, loi de Poisson, loi exponentielle, loi normale, loi uniforme, et bien d'autres encore. Une asymétrie positive signifie que, par rapport à la loi normale, la distribution considérée a un flanc gauche lourd et une longue queue à droite ; et vice-versa avec une asymétrie négative. Un kurtosis d'excès positif signifie que, par rapport à la loi normale, la distribution considérée a des queues lourdes, un centre pointu et des flancs appauvris ; et vice-versa avec un kurtosis d'excès négatif.

Proposition 1.3.9. Soit X une VA à valeurs réelles, et $(X_i)_{i \in \llbracket 1; n \rrbracket}$ une famille de VA indépendantes et distribuées identiquement à X . On note $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$. Les paramètres de forme respectent les assertions suivantes :

1. $\text{Kurt}(X) \geq \text{Asym}(X)^2 + 1$,

2. le minimum $\text{Kurt}(X) = 1$ est atteint par une seule loi : le pile ou face (Bernoulli équilibré),
3. si X est de loi normale, $\text{Kurtex}(X) = \text{Asym}(X) = 0$,
4. $\text{Asym}(\bar{X}) = \frac{1}{\sqrt{n}} \text{Asym}(X)$,
5. $\text{Kurtex}(\bar{X}) = \frac{1}{n} \text{Kurtex}(X)$,
6. $\text{Var}\left(s^2\left((X_i)_{i \in \llbracket 1; n \rrbracket}\right)\right) = \frac{\text{Var}(X)^2}{n} \left(\text{Kurt}(X) - \frac{n-3}{n-1} \right)$

Les points 3 et 5 de la proposition 1.3.9 montrent l'utilité du kurtosis d'excès : ses propriétés étendent le théorème central limite 1.3.5. En quelque sorte, le kurtosis d'excès quantifie la façon dont une moyenne d'échantillon tend vers la loi normale, quand la taille de l'échantillon augmente.

Pour estimer les paramètres de forme d'une distribution quelconque, on peut utiliser la méthodologie présentée dans ce manuscrit au chapitre 3.2.1 et en annexe C.2, et l'appliquer aux moments centrés d'ordre 3 et 4.

À partir d'un échantillon $(x_i)_{i \in \llbracket 1; n \rrbracket}$ de $n \geq 4$ réalisations indépendantes d'une VA X , on peut calculer des estimations sans biais $m'_{3 \text{ emp}}((x_i)_{i \in \llbracket 1; n \rrbracket})$ et $m'_{4 \text{ emp}}((x_i)_{i \in \llbracket 1; n \rrbracket})$ des moments centrés d'ordre 3 et 4 respectivement :

$$m'_{3 \text{ emp}}((x_i)_{i \in \llbracket 1; n \rrbracket}) = \frac{1}{A_n^3} (n^2 \Sigma_3 - 3n \Sigma_2 \Sigma_1 + 2 \Sigma_1^3) \quad (1.74)$$

$$m'_{4 \text{ emp}}((x_i)_{i \in \llbracket 1; n \rrbracket}) = \frac{1}{A_n^4} \times \left((n^2 - 2n + 3)(n \Sigma_4 - 4 \Sigma_3 \Sigma_1) + (-6n + 9) \Sigma_2^2 + 6n \Sigma_2 \Sigma_1^2 - 3 \Sigma_1^4 \right) \quad (1.75)$$

où $A_n^k = n!/(n-k)!$ est le nombre d'arrangements de k éléments parmi n , et $\Sigma_j = \sum_{i=1}^n X_i^j$.

Cela nous amène à utiliser les expressions suivantes, pour calculer les paramètres de forme empiriques d'un échantillon $(x_i)_{i \in \llbracket 1; n \rrbracket}$ de $n \geq 4$ réalisations indépendantes¹³ :

$$\text{Asym}_{\text{emp}}((x_i)_{i \in \llbracket 1; n \rrbracket}) = \frac{\sqrt{n(n-1)}}{n-2} \times \frac{n^2 \Sigma_3 - 3n \Sigma_2 \Sigma_1 + 2 \Sigma_1^3}{(n \Sigma_2 - \Sigma_1^2)^{3/2}} \quad (1.76)$$

$$\text{Kurtex}_{\text{emp}}((x_i)_{i \in \llbracket 1; n \rrbracket}) = \left(\frac{n(n-1)}{(n-2)(n-3)} \times \frac{(n^2 - 2n + 3)(n \Sigma_4 - 4 \Sigma_3 \Sigma_1) + (-6n + 9) \Sigma_2^2 + 6n \Sigma_2 \Sigma_1^2 - 3 \Sigma_1^4}{(n \Sigma_2 - \Sigma_1^2)^2} \right) - 3 \quad (1.77)$$

13. D'autres expressions des paramètres de forme, à calculer à partir d'un échantillon, existent, parfois appelées *statistiques de Fisher* [35].

Ces expressions sont obtenues par division des estimateurs non-biaisés des moments centrés d'ordre 3 et 4 par l'estimateur non-biaisé de la variance. Ce sont néanmoins des estimations biaisées du paramètre d'asymétrie et du kurtosis.

En particulier, quels que soient l'échantillon calculé $(x_i)_{i \in \llbracket 1;n \rrbracket}$ et la loi de probabilité dont il est issu, $|\text{Asym}_{emp}((x_i)_{i \in \llbracket 1;n \rrbracket})| \leq \sqrt{n}$ et $\text{Kurtex}_{emp}((x_i)_{i \in \llbracket 1;n \rrbracket}) + 3 \leq n$ (voir [61]). Il y a égalité dans le cas où toutes les valeurs de l'échantillon sont égales sauf une.

Au final, on obtient une estimation $(\text{Kurtex}_{emp}((x_i)_{i \in \llbracket 1;n \rrbracket})/n)$ du kurtosis d'excès de la distribution de la moyenne arithmétique $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$, qui constitue l'estimation de la grandeur ; et $(\text{Kurtex}_{emp}((x_i)_{i \in \llbracket 1;n \rrbracket})/n) \leq 1$. Ce kurtosis estimé est proche de 1 à la condition suivante : toutes les valeurs échantillonnées sont dans un intervalle restreint, sauf quelques-unes qui sont très éloignées de cet intervalle. Cette situation est exactement le signe que :

- soit on a pas assez échantillonné la VA pour donner des intervalles de confiance corrects. En utilisant plus de tirages, on pourra éventuellement donner des intervalles plus fiables, et le kurtosis le signalera.
- ou bien, la variance de la VA est en fait infinie. Dans ce cas, il est impossible de dégager des intervalles de confiance quelle que soit la taille de l'échantillon, en tout cas pas grâce au théorème central limite. En augmentant le nombre de tirages, le kurtosis restera près de l'unité.

Cela nous amène à dire que le kurtosis d'excès estimé $(\text{Kurtex}_{emp}((x_i)_{i \in \llbracket 1;n \rrbracket})/n)$ de la moyenne arithmétique est bien un indicateur comme nous le cherchions.

Remarque. Un calcul du kurtosis, depuis un échantillon, ne peut signaler une mauvaise convergence que si au moins une valeur « pathologique » a été tirée. Seule une étude préalable théorique (mais pas nécessairement complexe) du comportement de la variable aléatoire échantillonnée permet de décider d'une taille d'échantillon suffisante, pour ne pas être trompé par les résultats numériques de la méthode de Monte-Carlo.

1.4 (*) Suivi direct et suivi inverse des photons : un lien par l'adjonction

Dans la pratique courante de la méthode de Monte-Carlo en transport linéaire, on passe fréquemment d'algorithmes suivant des particules dans le sens de leur trajectoires à d'autres algorithmes suivant les particules en remontant leur historique, et on réalise autant le passage inverse. L'équivalence de ces fonctionnements algorithmiques est si connue et si éprouvée qu'on prend rarement la peine de la justifier spécifiquement.

Nous tentons de donner une compréhension de cette équivalence dans cette section, car elle ne sera plus vraie en transport non-linéaire.

1.4.1 Principe

•••• Dans la section 1.3.2, nous donnions un exemple de mise en œuvre de la méthode de Monte-Carlo pour le calcul d'une transmittivité au travers d'un milieu absorbant uniforme. Nous y montrions comment une formulation intégrale peut toujours être mise en place, qui décrit le fonctionnement d'un algorithme de Monte-Carlo, même si celui-ci a été construit comme une pure analogie du phénomène physique étudié.

Néanmoins, quand il s'agissait d'inventer l'algorithme de Monte-Carlo, 2 types de raisonnements distincts subsistaient :

- Soit on suit des photons dans le milieu, en cherchant à savoir s'ils vont le traverser ou non. Dans ce cas, on suit des photons fictifs, depuis leur entrée dans le milieu jusqu'à leur disparition (traversée ou absorption). On parle alors de *suivi direct* des photons.
- Soit on essaie de calculer la transmittivité définie comme une intégrale de la fonction de distribution f , détaillée grâce à la forme intégrale de l'Équation de Transfert Radiatif. On estime la quantité de photons présente en sortie du système, en remontant depuis cette sortie les chemins optiques qui y mènent. On parle alors de *suivi inverse* des photons.

- En toute rigueur, une 3^e possibilité doit être mentionnée : celle de compter les participations de chemins optiques reliant l'entrée à la sortie du système, et construits en partant d'une famille de points de passage obligés des photons allant de l'entrée à la sortie. Nous reviendrons là-dessus un peu plus tard dans cette section.

Cette distinction n'est pas seulement présente dans le précédent exemple de calcul de transmittivité. Elle est présente dans tout calcul d'une observable radiative par MMC — et partant, dans tout calcul Monte-Carlo en transfert radiatif.

Le lien qui existe entre ces différentes conceptions algorithmiques, desquelles émergent des formulations intégrales apparemment incompatibles, est issu de l'adjonction hermitienne¹⁴.

L'idée maîtresse est que toute grandeur radiative que l'on veut calculer s'exprime comme un produit scalaire entre la fonction de distribution f et un champ de sensation que nous noterons S_g .

Supposons que le système à l'étude soit un milieu semi-transparent \mathcal{M} , entourée de parois opaques $\partial\mathcal{M}$ possédant une normale en tout point. Cette normale est notée $\vec{n}_{\partial\mathcal{M}}(\vec{r})$, et est choisie sortante de la paroi, autrement dit dirigée vers l'intérieur de \mathcal{M} .

Si on étudie un régime permanent : pour commencer, notons $\mathcal{Q}_p = \mathcal{F}((\mathcal{M} \cup \partial\mathcal{M}) \times E_u \times E_v; \mathbb{R})$ l'ensemble des champs à valeurs réelles définis sur l'espace des phases. Posons ensuite les 2 produits scalaires suivants, définis sur \mathcal{Q}_p ² :

$$\langle a; b \rangle_{\text{obs I}} = \int_{E_v} d\nu \left\{ \int_{\mathcal{M}} d\vec{r} \int_{E_u} d\vec{u} a(\vec{r}; \vec{u}; \nu) b(\vec{r}; \vec{u}; \nu) + \int_{\partial\mathcal{M}} d\vec{r} \int_{\{\vec{u} \in E_u, \vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) \leq 0\}} d\vec{u} (-\vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r})) a(\vec{r}; \vec{u}; \nu) b(\vec{r}; \vec{u}; \nu) \right\} \quad (1.78a)$$

$$\langle a; b \rangle_{\text{obs D}} = \int_{E_v} d\nu \left\{ \int_{\mathcal{M}} d\vec{r} \int_{E_u} d\vec{u} a(\vec{r}; \vec{u}; \nu) b(\vec{r}; \vec{u}; \nu) + \int_{\partial\mathcal{M}} d\vec{r} \int_{\{\vec{u} \in E_u, \vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) \geq 0\}} d\vec{u} (\vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r})) a(\vec{r}; \vec{u}; \nu) b(\vec{r}; \vec{u}; \nu) \right\} \quad (1.78b)$$

Si on est en régime transitoire, entre les instants t_i et t_f : on considère $\mathcal{Q}_t = \mathcal{F}((\mathcal{M} \cup \partial\mathcal{M}) \times E_u \times E_v \times [t_i; t_f]; \mathbb{R})$ l'ensemble des champs à valeurs réelles définis sur l'espace-temps

14. L'annexe A.2 donne quelques rappels mathématiques au sujet de l'adjonction Hermitienne.

des phases, ainsi que les produits scalaires définis sur \mathcal{Q}_t^2 :

$$\langle a; b \rangle_{\text{obs I}} = \int_{E_\nu} d\nu \left\{ \begin{aligned} & \int_{t_i}^{t_f} dt \int_{\mathcal{M}} d\vec{r} \int_{E_u} d\vec{u} a(\vec{r}; \vec{u}; \nu; t) b(\vec{r}; \vec{u}; \nu; t) + \\ & \int_{t_i}^{t_f} dt \int_{\partial\mathcal{M}} d\vec{r} \int_{\{\vec{u} \in E_u, \vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) \leq 0\}} d\vec{u} (-\vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r})) a(\vec{r}; \vec{u}; \nu; t) b(\vec{r}; \vec{u}; \nu; t) + \\ & \int_{\mathcal{M}} d\vec{r} \int_{E_u} d\vec{u} a(\vec{r}; \vec{u}; \nu; t_f) b(\vec{r}; \vec{u}; \nu; t_f) \end{aligned} \right. \quad (1.79a)$$

$$\langle a; b \rangle_{\text{obs D}} = \int_{E_\nu} d\nu \left\{ \begin{aligned} & \int_{t_i}^{t_f} dt \int_{\mathcal{M}} d\vec{r} \int_{E_u} d\vec{u} a(\vec{r}; \vec{u}; \nu; t) b(\vec{r}; \vec{u}; \nu; t) + \\ & \int_{t_i}^{t_f} dt \int_{\partial\mathcal{M}} d\vec{r} \int_{\{\vec{u} \in E_u, \vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) \geq 0\}} d\vec{u} (\vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r})) a(\vec{r}; \vec{u}; \nu; t) b(\vec{r}; \vec{u}; \nu; t) + \\ & \int_{\mathcal{M}} d\vec{r} \int_{E_u} d\vec{u} a(\vec{r}; \vec{u}; \nu; t_i) b(\vec{r}; \vec{u}; \nu; t_i) \end{aligned} \right. \quad (1.79b)$$

Par exemple, en notant q l'observable radiative recherchée :

- Dans un problème instationnaire, on veut calculer la fonction de distribution en un point donné $(\vec{r}_0; \vec{c}_0; \nu_0; t_0)$ de l'espace temps des phases, avec $\vec{r}_0 \notin \partial\mathcal{M}$:

$$q = f(\vec{r}_0; \vec{c}_0; \nu_0; t_0) = \langle f_\nu; S_g \rangle_{\text{obs I}} \quad \text{avec } S_g(\vec{r}; \vec{c}; \nu; t) = \delta(\vec{r} - \vec{r}_0) \delta(\vec{c} - \vec{c}_0) \delta(\nu - \nu_0) \delta(t - t_0)$$

- Dans un problème stationnaire, on veut connaître le flux incident sur une paroi $\mathcal{S} \subset \partial\mathcal{M}$:

$$q = \langle f_\nu; S_g \rangle_{\text{obs I}} \quad \text{avec } S_g(\vec{r}; \vec{c}; \nu) = \mathbf{H}(\vec{r} \in \mathcal{S}) \mathbf{H}(\vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) \leq 0)$$

- Dans un problème stationnaire, on veut connaître le terme source radiatif en un point \vec{r}_0 :

$$\begin{aligned} q &= S_r(\vec{r}_0) = -\text{div}_{\vec{r}_0} \vec{J}(\vec{r}_0) \quad \text{selon la définition (1.15)} \\ &= \int_{E_\nu} d\nu \int_{4\pi} d\vec{u} k_a(\vec{r}_0; \vec{u}; \nu) (f_\nu(\vec{r}_0; \vec{u}; \nu) - f_{\nu \text{ eq}}(T(\vec{r}_0); \nu)) \quad \text{selon l'ETR} \\ &= \langle (f_\nu - f_{\nu \text{ eq}}); S_g \rangle_{\text{obs I}} \quad \text{avec } S_g(\vec{r}; \vec{c}; \nu) = \delta(\vec{r} - \vec{r}_0) k_a(\vec{r}_0; \vec{u}; \nu) \end{aligned}$$

●●● Dans l'équation de transfert radiatif sous forme différentielle, isolons le terme d'émission, que nous noterons S_f :

$$\begin{aligned} S_f(\vec{r}; \vec{u}; \nu; t) &= k_a(\vec{r}; \vec{u}; \nu; t) f_{\text{eq}}(T(\vec{r}; t); \nu) \\ &= \frac{1}{c} \partial_t f(\vec{r}; \vec{u}; \nu; t) + \vec{u} \cdot \overrightarrow{\text{grad}}_{\vec{r}} f(\vec{r}; \vec{u}; \nu; t) + k_a(\vec{r}; \vec{u}; \nu; t) f(\vec{r}; \vec{u}; \nu; t) + \\ & \quad \int_{4\pi} d\vec{u}' k_{d,F}(\vec{r}; \vec{u} | \vec{u}'; \nu; t) (f(\vec{r}; \vec{u}; \nu; t) - f(\vec{r}; \vec{u}'; \nu; t)) \end{aligned} \quad (1.80)$$

$$= CTR(f)(\vec{r}; \vec{u}; \nu; t) \quad (1.81)$$

où CTR est un opérateur déduisant le terme d'émission depuis la fonction de distribution. Son expression, sur l'intérieur de \mathcal{M} , se déduit de la ligne (1.80) précédant son invocation. Le même travail peut être fait sur les conditions aux limites. Au total, on trouve que CTR est linéaire.

La résolution d'un problème de transfert radiatif, répondant aux hypothèses détaillées en sections 1.1.1.3 et 1.1.4, revient formellement à inverser CTR . On connaît les champs de propriétés radiatives, le terme source d'émission S_f , et on en déduit $f = CTR^{-1}(S_f)$. Ainsi, une observable radiative q se calcule comme $q = \langle f; S_g \rangle_{\text{obs I}} = \langle CTR^{-1}(S_f); S_g \rangle_{\text{obs I}}$.

●●●● CTR est linéaire, et placée dans l'espace préhilbertien réel $(\mathcal{Q}; \langle \bullet; \bullet \rangle_{\text{obs D}})$ elle possède un adjoint dans l'espace préhilbertien réel $(\mathcal{Q}; \langle \bullet; \bullet \rangle_{\text{obs I}})$. Sans expliquer le détail de son obtention, voici son expression à l'intérieur d'un milieu semi-transparent :

$$CTR^*(g)(\vec{r}; \vec{u}; \nu; t) = -\frac{1}{c} \partial_t g(\vec{r}; \vec{u}; \nu; t) - \vec{u} \cdot \overrightarrow{\text{grad}}_{\vec{r}} g(\vec{r}; \vec{u}; \nu; t) + k_a(\vec{r}; \vec{u}; \nu; t) g(\vec{r}; \vec{u}; \nu; t) + \int_{4\pi} d\vec{u}' k_{d,F}(\vec{r}; \vec{u}' | \vec{u}; \nu; t) (g(\vec{r}; \vec{u}; \nu; t) - g(\vec{r}; \vec{u}'; \nu; t)) \quad (1.82)$$

Aux parois, on obtient des résultats similaires. Depuis la version générique de l'ETR applicable à une surface opaque en équilibre thermodynamique local (1.49), on sépare le terme d'émission $\varepsilon(\vec{r}; \vec{u}; \nu; t) f_{\nu,eq}(T(\vec{r}; t); \nu)$. Puis on dégage l'adjoint de CTR à la surface (en fait en même temps que l'adjoint à CTR dans le volume) :

$$\forall \vec{u} \in E_u \text{ tq } \vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) \leq 0, CTR^*(g)(\vec{r}; \vec{u}; \nu; t) = g(\vec{r}; \vec{u}; \nu; t) - \int_{2\pi^-} d\vec{u}' (-\vec{u}' \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r})) (1 - \varepsilon(\vec{r}; -\vec{u}; \nu; t)) \Phi_p(\vec{r}; -\vec{u}' | \vec{u}; \nu; t) g(\vec{r}; \vec{u}'; \nu; t) \quad (1.83)$$

où $\int_{2\pi^-} d\vec{u}'$ signifie qu'on intègre dans l'hémisphère des directions entrantes dans la paroi, telles que $\vec{u}' \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) \leq 0$.

Comme CTR a un adjoint dans $(\mathcal{Q}; \langle \bullet; \bullet \rangle_{\text{obs I}})$, CTR^{-1} a un adjoint dans $(\mathcal{Q}; \langle \bullet; \bullet \rangle_{\text{obs D}})$. Cela permet de ré-exprimer l'observable d'intérêt :

$$\begin{aligned} \langle f; S_g \rangle_{\text{obs I}} &= \langle CTR^{-1}(S_f); S_g \rangle_{\text{obs I}} \\ &= \langle S_f; (CTR^{-1})^*(S_g) \rangle_{\text{obs D}} \\ &= \langle S_f; (CTR^*)^{-1}(S_g) \rangle_{\text{obs D}} \\ &= \langle S_f; g \rangle_{\text{obs D}} \end{aligned} \quad (1.84)$$

où g est la solution de l'équation : « $CTR^*(g) = S_g$ ». Dans la suite de ce manuscrit, nous dénommerons g la « captance ».

●●●● On peut appliquer une démarche de résolution le long d'un rayon similaire à ce qui est présenté en section 1.2.1, pour obtenir une forme intégrale de l'équation régissant la captance. On obtient :

$$\begin{aligned} g(\vec{r}; \vec{u}; \nu; t) &= \left(\int_0^{s_0} ds \exp\left(-\int_0^s ds' k_t(\vec{r} + s'\vec{u}; \vec{u}; \nu; t + \frac{s'}{c})\right) \left(S_g(\vec{r} + s\vec{u}; \vec{u}; \nu; t + \frac{s}{c}) + \right. \right. \\ &\quad \left. \left. k_d(\vec{r} + s\vec{u}; \vec{u}; \nu; t + \frac{s}{c}) \int_{4\pi} d\vec{u}' \Phi(\vec{r} + s\vec{u}; \vec{u}' | \vec{u}; \nu; t + \frac{s}{c}) g(\vec{r} + s\vec{u}; \vec{u}'; \nu; t + \frac{s}{c}) \right) \right) + \\ &\quad \exp\left(-\int_0^{s_0} ds k_t(\vec{r} + s\vec{u}; \vec{u}; \nu; t + \frac{s}{c})\right) g(\vec{r} + s_0\vec{u}; \vec{u}; \nu; t + \frac{s_0}{c}) \end{aligned} \quad (1.85)$$

où la variable muette s est une abscisse de descente le long d'un rayon lumineux. Comme dans l'écriture (1.43) on pose qu'elle possède un maximum s_0 , avec éventuellement $s_0 = +\infty$. Dans la pratique, on considère généralement que $s_0 = s_m$, l'abscisse de l'intersection suivante avec une paroi : $s_m = \min\{s \in \mathbb{R}^+, \vec{r} + s\vec{u} \in \text{parois}\}$.

L'équation (1.85) se retrouve dans les formulations intégrales des algorithmes analogues de calcul de grandeurs radiatives, en suivi direct, comme celui calculant la transmitti-

vité dans l'exemple de la section 1.3.2. Elle en constitue, avec les adjointes des équations aux frontières, les parties récursives. La partie initiale est contenue dans le produit scalaire

$\langle \bullet; \bullet \rangle_{\text{obs D}}$. On peut finalement affirmer que du point de vue de la formulation intégrale du transport, l'image physique d'un échantillonnage de la captance justifie les algorithmes de Monte-Carlo en suivi direct.

1.4.2 Remarques

•••• Dans l'expression (1.85) on intègre le rayon sur $t + s/c \geq t$, au lieu de l'intégrer sur $t - s/c \leq t$ comme dans l'équation de transfert radiatif sous forme intégrale (1.43). L'image de somme de contributions avec atténuation, justifiant physiquement la forme intégrale de l'ETR (1.43), est néanmoins valable sur la forme intégrale (1.85) de l'adjointe à l'ETR; mais la causalité est renversée.

S'il y a une luminance non nulle en un point de l'espace-temps des phases, c'est parce que des sources de lumière sont présentes dans le passé. S'il y a de la captance en un point de l'espace-temps des phases, c'est parce que des zones sensibles sont présentes dans le futur.

À ce titre, si on s'intéresse à un problème instationnaire de transfert radiatif, la formulation différentielle en luminance du problème sera dotée d'une condition initiale, et la formulation différentielle en captance du problème sera dotée d'une condition finale.

•••• Plus tôt dans cette section, on évoquait la possibilité d'écrire une formulation intégrale d'une observable radiative, basée sur un espace de chemins optiques construit depuis une famille de points de passage obligés entre les sources de lumière et les zones sensibles. De telles formulations intégrales sont en fait exprimées à l'aide :

- d'un produit scalaire $\langle \bullet; \bullet \rangle_{\text{obs M}}$, à l'expression similaire de celles de $\langle \bullet; \bullet \rangle_{\text{obs I}}$ et de $\langle \bullet; \bullet \rangle_{\text{obs D}}$,
- de la formulation intégrale (1.43) de l'ETR, et de ses conditions aux limites,
- de la formulation intégrale (1.85) de l'adjointe à l'ETR, et de ses conditions aux limites,

et peuvent être expliquées grâce à la notion de *source de lumière secondaire*.

Si l'on dispose d'une famille \mathcal{P} de points de l'espace-temps des phases, telle que tout le flux allant des sources de lumière aux zones

sensibles passe une et une seule fois par \mathcal{P} , alors du point de vue des zones sensibles on peut remplacer les vraies sources de lumière par un ensemble de sources secondaires réparties sur \mathcal{P} . Ces sources secondaires émettent alors directement ce qu'elles reçoivent depuis les sources primaires.

En notant S_{f2} l'émission fictive des sources secondaires, qui égalise la luminance reçue par ces sources reçue depuis les sources primaires, on arrive à ce que :

$$\begin{aligned} q &= \langle f; S_g \rangle_{\text{obs I}} \\ &= \langle CTR_1^{-1}(S_{f2}); S_g \rangle_{\text{obs I}} \end{aligned}$$

où CTR_1 exprime l'émission fictive S_{f2} en fonction de la distribution de la lumière ayant traversé \mathcal{P} .

$$= \langle CTR_1^{-1}(CTR_2^{-1}(S_f)); S_g \rangle_{\text{obs I}} \quad (1.86a)$$

où CTR_2 exprime l'émission originelle S_f en fonction de la distribution de la lumière n'ayant pas traversé \mathcal{P} .

$$= \langle CTR_2^{-1}(S_f); CTR_1^{*-1}(S_g) \rangle_{\text{obs M}} \quad (1.86b)$$

$$= \langle S_f; CTR_2^{*-1}(CTR_1^{*-1}(S_g)) \rangle_{\text{obs D}} \quad (1.86c)$$

En matière de calcul de q , l'expression (1.86a) est associée à un suivi inverse, l'expression (1.86c) est associée à un suivi direct, et l'expression (1.86b) est associée à un suivi à deux sens et partant de \mathcal{P} .

Des formalismes intégraux basés sur une telle construction des chemins optiques depuis leur milieu se rencontrent par exemple :

- au chapitre 3.3 de ce manuscrit, où il est question de calculer la productivité d'une centrale solaire de concentration à tour,
- lorsqu'on utilise les décompositions en ordre de diffusion.

•••• Il est parfois mentionné qu'en transfert radiatif, la réversibilité des chemins optiques explique la validité des approches utilisant l'adjonction. C'est faux. Dans l'obtention de l'adjoint de l'opérateur CTR , la réversibilité des chemins optiques ne joue aucun rôle.

•••• Le discours mathématique tenu dans cette section, qui explique à l'aide de l'adjonction les algorithmes de Monte-Carlo suivant les particules de leur émission vers leur capture, ne tient pas en cinétique des gaz. En effet, l'équation de Boltzmann est non-linéaire, donc elle n'a pas d'adjointe.

Cela me fait dire qu'il est *a priori* impossible de créer des algorithmes de Monte-Carlo comme je les décris dans ce manuscrit, c'est-à-dire sans biais par rapport à une formulation intégrale justifiée à l'aide des équations de transport elles-mêmes, en cinétique des gaz et en suivant les particules en sens direct.

1.5 (*) Diffusion multiple

Lors de ce travail de thèse, les diverses pratiques algorithmiques utilisées en transfert radiatif dans les situations de diffusion multiple ont constituées pour nous des références essentielles.

Nous prenons le temps dans cette section d'étudier un problème physique de ce type. Nous explicitons les formulations intégrales et statistiques de plusieurs algorithmes utilisables, tout en gardant à l'esprit l'aide ce que fournit un intuitif analogue. C'est aussi l'occasion d'une part d'illustrer le propos de la section 1.4, et d'autre part de traiter de l'utilisation d'échantillonnages « non analogues ».

1.5.1 Description du problème

On veut calculer, pour un milieu \mathcal{M} borné, isotrope et uniformément diffusant (le milieu n'est ni absorbant ni émissif), les moments de la distribution des longueurs de premier retour d'une lumière monochromatique. L'éclairage du milieu est supposé uniforme et isotrope, à sa surface. Ces résultats sont utilisés dans la

publication [118], en considérant des géométries particulières.

Le problème se formule comme suit. On a un milieu \mathcal{M} , entouré d'une frontière $\partial\mathcal{M}$ dont la normale est définie presque partout. Le milieu est diffusant avec un coefficient de diffusion constant k_d et une fonction de phase Φ ne dépendant que de l'angle de déviation.

On veut calculer les grandeurs :

$$\langle L^n \rangle = \int_0^{+\infty} (ct)^n dt \int_{\partial\mathcal{M}} d\vec{r} \int_{2\pi^-} d\vec{u} (-\vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r})) f(\vec{r}; \vec{u}; t) \quad (1.87a)$$

avec $n \in \mathbb{N}$, où f est fixée par :

$$\begin{cases} \forall(\vec{r}; \vec{u}) \in (\mathcal{M} \setminus \partial\mathcal{M}) \times E_u, f(\vec{r}; \vec{u}; t) = 0 \\ \forall(\vec{r}; \vec{u}; t) \in \partial\mathcal{M} \times E_u \times E_t, \text{ tq } \vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) > 0, f(\vec{r}; \vec{u}; t) = \frac{1}{\pi S} \delta(t) \end{cases} \quad (1.87b)$$

où $\vec{n}_{\partial\mathcal{M}}(\vec{r})$ est la normale à $\partial\mathcal{M}$ en \vec{r} dirigée vers l'intérieur de \mathcal{M} , $2\pi^-$ signifie qu'on intègre sur l'hémisphère des directions sortantes de \mathcal{M} , et S est la surface de $\partial\mathcal{M}$. La dépendance de f en ν est omise, car la lumière présente est monochromatique.

1.5.2 Formulation intégrale, et notations

•••• En reprenant la section 1.4, on peut ré-exprimer les $\langle L^n \rangle$. Ils s'identifient à $\langle f; S_g \rangle_{obs}$, où :

$$\begin{cases} \forall(\vec{r}; \vec{u}; t) \in (\mathcal{M} \setminus \partial\mathcal{M}) \times E_u \times E_t, S_f(\vec{r}; \vec{u}; t) = 0 \\ \forall(\vec{r}; \vec{u}; t) \in \partial\mathcal{M} \times E_u \times E_t, S_f(\vec{r}; \vec{u}; t) = \frac{H(\vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) > 0)}{\pi S} \delta(t) \end{cases} \quad (1.88a)$$

$$\begin{cases} \forall(\vec{r}; \vec{u}; t) \in (\mathcal{M} \setminus \partial\mathcal{M}) \times E_u \times E_t, & S_g(\vec{r}; \vec{u}; t) = 0 \\ \forall(\vec{r}; \vec{u}; t) \in \partial\mathcal{M} \times E_u \times E_t, & S_g(\vec{r}; \vec{u}; t) = H(\vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}) < 0) (ct)^n \end{cases} \quad (1.88b)$$

•••• Pour pouvoir justifier facilement la mise en place de calculs de sensibilités paramétriques, à la section 1.5.5, nous utilisons une approche intégrale de la MMC, basée sur les formulations intégrales des équations de transport.

Étant données les formes des sources de luminance et de captance, nous choisissons, comme modèle de transport, l'adjointe à l'équation de transfert radiatif. En effet, si

on choisissait l'ETR elle-même, il faudrait intégrer la distribution de Dirac sur un espace de chemins optiques complexe. La MMC ne permet pas de faire cela, car sa quadrature ne permet pas de recouvrir la distribution de Dirac dans ce cas (les points d'échantillonnage tomberont à côté du pic de la distribution).

Nous allons ainsi monter des algorithmes de Monte-Carlo suivant des photons en sens direct.

L'estimateur $\widetilde{\langle L^n \rangle}$ de $\langle L^n \rangle$ que ces considérations font apparaître sera basé sur :

$$\langle L^n \rangle = \int_{\partial\mathcal{M}} \frac{d\vec{r}}{S} \int_{2\pi^+} \frac{d\vec{u}}{\pi} (\vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r})) g(\vec{r}; \vec{u}; t = 0) \quad (1.89a)$$

$$g(\vec{r}; \vec{u}; t) = \int_0^{+\infty} ds k_d \exp(-k_d s) \left(H(s \geq s_m) (ct + s_m)^n + H(s < s_m) \int_{4\pi} d\vec{u}' \Phi(\vec{u}' \cdot \vec{u}) g(\vec{r} + s\vec{u}; \vec{u}'; t + s/c) \right) \quad (1.89b)$$

où $2\pi^+$ signifie qu'on intègre sur l'hémisphère des directions entrantes dans \mathcal{M} , et $s_m = \min\{s \in \mathbb{R}^+, \vec{r} + s\vec{u} \in \partial\mathcal{M}\}$.

1.5.3 Choix des variables aléatoires

•••• Pour exprimer complètement $\widetilde{\langle L^n \rangle}$, il faut mettre ces précédentes expressions intégrales sous forme statistique, et donc introduire des VA pour tirer \vec{r} et \vec{u} dans (1.89a) et s et \vec{u}' dans (1.89b). Le choix le plus simple, présentant par ailleurs une analogie certaine avec la physique du problème, est sans doute :

$$\widetilde{\langle L^n \rangle} = \widetilde{G}(\vec{R}; \vec{U}; 0) \quad (1.90a)$$

$$\widetilde{G}(\vec{r}; \vec{u}; t) = H(S \geq s_m) (ct + s_m)^n + H(S < s_m) \widetilde{G}(\vec{r} + S\vec{u}; \vec{U}'; t + S/c) \quad (1.90b)$$

avec :

- \vec{R} de loi uniforme sur $\partial\mathcal{M}$: $p_{\vec{R}} : \vec{r} \mapsto 1/S$,
- \vec{U} de loi lambertienne : $p_{\vec{U}} : \vec{u} \mapsto (\vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}))/\pi$,
- S de loi exponentielle de constante k_d : $F_S : s \mapsto H(s > 0) \times (1 - \exp(-k_d s))$,
- \vec{U}' dont la loi est la fonction de phase : $p_{\vec{U}'} : \vec{u}' \mapsto \Phi(\vec{U}' \cdot \vec{u})$.

Cependant, le choix de \vec{R} , \vec{U} , S , et \vec{U}' est en fait arbitraire. De façon générale, un estimateur de $\langle L^n \rangle$ basé sur les expressions intégrales (1.89a) et (1.89b) aura l'expression :

$$\widetilde{\langle L^n \rangle} = \frac{\vec{U} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{R})}{\pi S p_{\vec{R}}(\vec{R}) p_{\vec{U}}(\vec{U})} \widetilde{G}(\vec{R}; \vec{U}; 0) \quad (1.91a)$$

$$\tilde{G}(\vec{r}; \vec{u}; t) = \int_0^{+\infty} ds \frac{k_d \exp(-k_d S)}{p_S(S)} \left(H(S \geq s_m) (ct + s_m)^n + H(S < s_m) \frac{\Phi(\vec{U}' \cdot \vec{u})}{p_{\vec{U}'}(\vec{U}')} \tilde{G}(\vec{r} + S\vec{u}; \vec{U}'; t + S/c) \right) \quad (1.91b)$$

et sera échantillonné par un algorithme suivant le prototype présenté en 1.5.

Sorties : Une estimation ponctuelle du moment $\langle L^n \rangle$

- 1 $i \leftarrow 0$;
- 2 $l_{tot} \leftarrow 0$;
- 3 échantillonner \vec{R}_0 sur $\partial\mathcal{M}$: on obtient \vec{r}_0 ;
- 4 échantillonner \vec{U}_0 sur $\{\vec{u} \in E_u, \vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}_0) > 0\}$: on obtient \vec{u}_0 ;
- 5 $\pi_0 \leftarrow \frac{\vec{u}_0 \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}_0)}{\pi S p_{\vec{R}_0}(\vec{r}_0) p_{\vec{U}_0}(\vec{u}_0)}$;
- 6 **répéter**
- 7 $i \leftarrow i + 1$;
- 8 échantillonner S_i sur $[0; +\infty[$: on obtient s_i ;
- 9 $\pi_i \leftarrow \frac{k_d \exp(-k_d s_i)}{p_{s_i}(s_i)} \pi_{i-1}$;
- 10 **si** $\{\vec{r}_{i-1} + s' \vec{u}_{i-1}, s' \in [0; s_i]\} \cap \partial\mathcal{M} = \emptyset$ **alors**
- 11 **sortie** $\leftarrow non$;
- 12 $l_{tot} \leftarrow l_{tot} + s_i$;
- 13 $\vec{r}_i \leftarrow \vec{r}_{i-1} + s_i \vec{u}_{i-1}$;
- 14 échantillonner \vec{U}_i sur la sphère unité : on obtient \vec{u}_i ;
- 15 $\pi_i \leftarrow \frac{\Phi(\vec{u}_i \cdot \vec{u}_{i-1})}{p_{\vec{U}_i}(\vec{u}_i)} \pi_i$;
- 16 **sinon**
- 17 **sortie** $\leftarrow oui$;
- 18 $l_{tot} \leftarrow l_{tot} + \min\{s' \in \mathbb{R}^+, \vec{r}_{i-1} + s' \vec{u}_{i-1} \in \text{parois}\}$;
- 19 **jusqu'à** **sortie** = oui;
- 20 **retourner** $\pi_i(l_{tot}^n)$;

Algorithme 1.5 : Patron d'algorithme estimant ponctuellement le moment $\langle L^n \rangle$ décrit en section 1.5.1, en échantillonnant l'expression statistique (1.91).

●●●● Considérant l'algorithme 1.5, posons les variables aléatoires suivantes :

- $\vec{\Gamma}$, qui est le chemin optique parcouru (le descriptif complet, pas seulement la longueur),
- I_m , la valeur de i quand le chemin optique revient aux parois,
- Π_i , avec $i \in \llbracket 0; I_m \rrbracket$, correspondant à π_i ,

ainsi que $l_{tot}(\vec{\Gamma})$, la longueur cumulée de $\vec{\Gamma}$. On a :

$$\widetilde{\langle L^n \rangle} = \Pi_{I_m}(l_{tot}(\vec{\Gamma})^n) \quad (1.92a)$$

avec :

$$\Pi_{I_m} = \frac{\vec{U}_0 \cdot \vec{n}_{\partial\mathcal{M}}(\vec{R}_0)}{\pi S p_{\vec{R}_0}(\vec{R}_0) p_{\vec{U}_0}(\vec{U}_0)} \times \frac{k_d \exp(-k_d S_{I_m})}{p_{S_{I_m}}(S_{I_m})} \prod_{i=1}^{I_m-1} \frac{k_d \exp(-k_d S_i)}{p_{S_i}(S_i)} \times \frac{\Phi(\vec{U}_i \cdot \vec{U}_{i-1})}{p_{\vec{U}_i}(\vec{U}_i)} \quad (1.92b)$$

$$l_{tot}(\vec{\Gamma}) = \sum_{i=1}^{I_m} S_i \quad (1.92c)$$

Si les \vec{R}_0 , S_i , et \vec{U}_i ont les lois de probabilités analogues à la physique décrites en (1.90), alors $\Pi_{I_m} = 1$.

Par ailleurs, $p_{l_{tot}(\vec{\Gamma})}$ décroît de façon exponentielle au voisinage de $+\infty$. Cela est démontrable grâce au théorème spectral, applicable à CTR^{-1} parce que \mathcal{M} est borné et CTR présente une certaine symétrie.

Grâce à cela, si on utilise les lois de tirages analogues décrites précédemment, pour tout $n \in \mathbb{N}$ la distribution de $\langle L^n \rangle$ a tous ses moments finis.

●●●● Il n'en va pas aussi commodément avec toutes les lois de tirages possibles pour les \vec{R}_0 , S_i , et \vec{U}_i . Pour comprendre cela, considérons le produit $\prod_{i=1}^{I_m-1} \dots$ dans (1.92b).

Supposons pour simplifier que \vec{R}_0 , I_m , les S_i , et les \vec{U}_i sont indépendants¹⁵, et posons :

$$\Pi'_i = \frac{k_d \exp(-k_d S_i) \Phi(\vec{U}_i \cdot \vec{U}_{i-1})}{p_{S_i}(S_i) p_{\vec{U}_i}(\vec{U}_i)} \quad (1.93)$$

On a : $E(\Pi'_i) = 1$.

Supposons maintenant aussi que les Π'_i sont identiquement distribués. Les propriétés de la variance amènent alors à ce que :

$$\forall n \in \mathbb{N}^*, \text{Var}\left(\prod_{i=1}^n \Pi'_i\right) = (\text{Var}(\Pi'_1) + 1)^n - 1 \quad (1.94)$$

Grossièrement, la variance du produit $\prod_{i=1}^{I_m-1} \dots$ suit une progression géométrique selon I_m .

Comme la loi de probabilité de I_m converge par ailleurs vers une suite géométrique décroissante (de la même façon que $l_{tot}(\vec{\Gamma})$), si $\text{Var}(\Pi'_1)$ est trop élevée la variance de $\langle L^n \rangle$ est infinie.

Bien sûr, quelle que soit l'expression intégrale qu'on convertit sous forme statistique,

il existe toujours des lois de tirage assez mal adaptées pour que la variance de l'estimateur obtenu soit infinie. Cependant, cela prend davantage d'importance quand l'expression intégrale est hautement récursive, comme dans l'expression (1.89) des moments $\langle L^n \rangle$. Cette importance vient de ce que grossièrement, la variance d'un produit est le produit des variances de ses composantes.

Il en résulte finalement, que l'expression analogue proposée pour les \vec{R}_0 , S_i , et \vec{U}_i n'est pas si peu optimale qu'on aurait pu l'imaginer au départ. Elle fonctionne, et c'est en fait déjà pas mal. Globalement, dans toute expression intégrale infiniment récursive comme la (1.89b), il vaut mieux poser les VA d'échantillonnage de sorte à compenser les coefficients multiplicateurs de la partie récursive de l'intégrale. En transport corpusculaire, une approche analogue mène souvent à des lois de tirage adéquates. Par exemple, dans un problème de transfert radiatif avec diffusion multiple (comme le cas d'étude de cette section), il vaut mieux utiliser exactement la fonction de phase du modèle pour tirer les directions de diffusion, quitte à utiliser une méthode de rejet, qu'utiliser une loi de tirage différente.

●●●● Cette difficulté à optimiser le tirage des chemins, quand on est en présence de diffusion multiple ou d'un équivalent, est connu des physiciens du transport corpusculaire depuis des décennies. Il était déjà présent dans le problème de protection neutronique pour lequel la MMC a été créée. Dans ce cas, on s'intéresse à la minuscule proportion des neutrons arrivant à traverser le bouclier. Un calcul analogue demande alors beaucoup d'estimations pour fournir une estimation moyenne précise (car les événements intéressants sont des événements rares), et chaque estimation requiert un

15. C'est impossible à obtenir tout-à-fait en pratique, car I_m est lié à la suite des \vec{R}_i échantillonnés.

grand effort computationnel (à cause des nombreuses collisions subies par chaque neutron suivi).

De nombreuses techniques d'optimisation ont alors été proposées, pour pallier à ce problème : échantillonnage préférentiel des longues distances de vol, échantillonnage préférentiel des directions amenant à la sortie, ... L'expérience a montré qu'elles amenaient souvent à des résultats bien pires que ceux qu'elles étaient censées corriger ; à cause de ce qui est expliqué précédemment dans cette section. L'approche de variance nulle a permis de comprendre pourquoi. Un neutron situé profondément dans le milieu, loin de toute sortie, subira de nombreuses collisions avant de sortir éventuellement du milieu. Quand on tire une distance de vol libre ou une direction de diffusion pour ce neutron, sa sortie du milieu dépend assez peu du résultat de ce tirage en particulier. La déformation à appliquer à ce tirage, depuis un tirage analogue, pour obtenir la plus faible variance possible, est en fait très faible ; et il est indispensable de la baser sur un modèle physique simplifié.

Ils s'écrivent :

$$\langle L^n \rangle = \int_0^{+\infty} (ct)^n dt \left(\int_{\partial\mathcal{M}} d\vec{r} \int_{2\pi^-} d\vec{u} (-\vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r})) f(\vec{r}; \vec{u}; t) + \int_{\mathcal{M}} d\vec{r} \int_{4\pi} d\vec{u} k_a f(\vec{r}; \vec{u}; t) \right) \quad (1.95)$$

On peut donner une expression intégrale récursive des $\langle L^n \rangle$, en utilisant à nouveau l'adjointe à l'ETR :

$$\langle L^n \rangle = \int_{\partial\mathcal{M}} \frac{d\vec{r}}{S} \int_{2\pi^+} \frac{d\vec{u}}{\pi} (\vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r})) g(\vec{r}; \vec{u}; t=0) \quad (1.96a)$$

$$g(\vec{r}; \vec{u}; t) = \int_0^{+\infty} ds k_t \exp(-k_t s) \left(H(s \geq s_m) (ct + s_m)^n + H(s < s_m) \times \left(\frac{k_a}{k_t} (ct + s)^n + \frac{k_d}{k_t} \int_{4\pi} d\vec{u}' \Phi(\vec{u}' \cdot \vec{u}) g(\vec{r} + s\vec{u}; \vec{u}'; t + s/c) \right) \right) \quad (1.96b)$$

•••• Des estimateurs $\widetilde{\langle L^n \rangle}$ de $\langle L^n \rangle$ peuvent à nouveau être donnés.

••• En utilisant encore une analogie avec la physique, on peut aboutir à un estimateur ayant une variance assurément finie :

$$\widetilde{\langle L^n \rangle} = \widetilde{G}(\vec{R}; \vec{U}; 0) \quad (1.97a)$$

$$\begin{aligned} \widetilde{G}(\vec{r}; \vec{u}; t) = & H(S \geq s_m) (ct + s_m)^n + H(S < s_m) \times \\ & (H(\text{Col} = \text{abs}) (ct + S)^n + H(\text{Col} = \text{diff}) \widetilde{G}(\vec{r} + S\vec{u}; \vec{U}'; t + S/c)) \end{aligned} \quad (1.97b)$$

avec :

Finalement, il apparaît dans l'exemple plus haut que la *diffusion épaisse* — situation physique où les corpuscules subissent un nombre moyen élevé de collisions dans le système — est un problème difficile pour la méthode de Monte-Carlo ; la difficulté dont il est question est due au coût computationnel élevé de chaque réalisation, plus qu'à une forte variance d'estimateur. Ce point d'achoppement de la MMC est très connu ; des techniques pour l'éviter, comme le couplage de la MMC avec une autre méthode de calcul, ont été développées (voir par exemple [103]).

1.5.4 Modification du problème : ajout d'absorption

•••• Supposons à présent qu'en plus d'être diffusant, le milieu \mathcal{M} est absorbant, avec un coefficient d'absorption constant k_a .

On définit à présent les $\langle L^n \rangle$ comme les moments de la distribution des longueurs avant absorption — dans le milieu, ou à la paroi supposée parfaitement absorbante.

- \vec{R} de loi uniforme sur $\partial\mathcal{M} : p_{\vec{R}} : \vec{r} \mapsto 1/S$,
- \vec{U} de loi lambertienne : $p_{\vec{U}} : \vec{u} \mapsto (\vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}))/\pi$,
- Col à valeurs dans $\{abs; diff\}$, de probabilités respectives k_a/k_t et k_d/k_t ,
- S de loi exponentielle de constante $k_t : F_S : s \mapsto \mathbb{H}(s > 0) \times \exp(-k_t s)$,
- \vec{U}' dont la loi est la fonction de phase : $p_{\vec{U}'} : \vec{u}' \mapsto \Phi(\vec{U}' \cdot \vec{u})$.

L'algorithme 1.6 montre la procédure d'échantillonnage de cet estimateur.

Sorties : Une estimation ponctuelle du moment $\langle L^n \rangle$

```

1   $i \leftarrow 0$ ;
2   $l_{tot} \leftarrow 0$ ;
3  échantillonner  $\vec{R}_0$  sur  $\partial\mathcal{M}$  de façon uniforme : on obtient  $\vec{r}_0$ ;
4  échantillonner  $\vec{U}_0$  sur  $\{\vec{u} \in E_u, \vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}_0) > 0\}$  de façon lambertienne : on obtient  $\vec{u}_0$ ;
5  répéter
6       $i \leftarrow i + 1$ ;
7      échantillonner  $S$  de loi exponentielle de constante  $k_t$  : on obtient  $s_i$ ;
8      si  $\{\vec{r}_{i-1} + s' \vec{u}_{i-1}, s' \in [0; s_i]\} \cap \partial\mathcal{M} = \emptyset$  alors
9           $l_{tot} \leftarrow l_{tot} + s_i$ ;
10          $\vec{r}_i \leftarrow \vec{r}_{i-1} + s_i \vec{u}_{i-1}$ ;
11         échantillonner  $R$  de loi uniforme standard : on obtient  $r_i$ ;
12         si  $r_i < k_d/k_t$  alors
13              $sortie \leftarrow non$ ;
14             échantillonner  $\vec{U}_i$  de densité de probabilité  $\vec{u}' \mapsto \Phi(\vec{u}' \cdot \vec{u}_{i-1})$  : on obtient  $\vec{u}_i$ ;
15         sinon  $sortie \leftarrow oui$ ;
16     sinon
17          $sortie \leftarrow oui$ ;
18          $l_{tot} \leftarrow l_{tot} + \min\{s' \in \mathbb{R}^+, \vec{r}_{i-1} + s' \vec{u}_{i-1} \in \text{parois}\}$ ;
19 jusqu'à  $sortie = oui$ ;
20 retourner  $l_{tot}^n$ ;
```

Algorithme 1.6 : Algorithme estimant ponctuellement le moment $\langle L^n \rangle$ décrit en section 1.5.4, en échantillonnant l'expression statistique (1.97).

••• Un autre estimateur très similaire peut être donné :

$$\widetilde{\langle L^n \rangle} = \widetilde{G}(\vec{R}; \vec{U}; 0) \quad (1.98a)$$

$$\begin{aligned} \widetilde{G}(\vec{r}; \vec{u}; t) = & \mathbb{H}(S \geq s_m) (ct + s_m)^n + \mathbb{H}(S < s_m) \times \\ & \left(\frac{k_a}{k_t} (ct + S)^n + \frac{k_d}{k_t} \widetilde{G}(\vec{r} + S\vec{u}; \vec{U}'; t + S/c) \right) \end{aligned} \quad (1.98b)$$

avec les mêmes variables aléatoires. L'algorithme 1.7 en montre la procédure d'échantillonnage.

Cet estimateur peut être décomposé en une somme de contributions le long du chemin optique construit :

$$\widetilde{\langle L^n \rangle} = \sum_{i=1}^{I_m} C p_i W p_i = \sum_{i=1}^{+\infty} C p_i W p_i \quad (1.99a)$$

Sorties : Une estimation ponctuelle du moment $\langle L^n \rangle$

```

1   $i \leftarrow 0$ ;
2   $l_{tot} \leftarrow 0$ ;
3   $\Sigma \leftarrow 0$ ;
4  échantillonner  $\vec{R}_0$  sur  $\partial\mathcal{M}$  de façon uniforme : on obtient  $\vec{r}_0$ ;
5  échantillonner  $\vec{U}_0$  sur  $\{\vec{u} \in E_u, \vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}_0) > 0\}$  de façon lambertienne : on obtient  $\vec{u}_0$ ;
6   $\pi_0 \leftarrow 1$ ;
7  répéter
8  |    $i \leftarrow i + 1$ ;
9  |   échantillonner  $S$  de loi exponentielle de constante  $k_t$  : on obtient  $s_i$ ;
10 |   si  $\{\vec{r}_{i-1} + s' \vec{u}_{i-1}, s' \in [0; s_i]\} \cap \partial\mathcal{M} = \emptyset$  alors
11 |   |    $sortie \leftarrow non$ ;
12 |   |    $l_{tot} \leftarrow l_{tot} + s_i$ ;
13 |   |    $\vec{r}_i \leftarrow \vec{r}_{i-1} + s_i \vec{u}_{i-1}$ ;
14 |   |    $\Sigma \leftarrow \Sigma + \pi_{i-1} k_a(l_{tot}^n)/k_t$ ;
15 |   |    $\pi_i \leftarrow k_d \pi_{i-1}/k_t$ ;
16 |   |   échantillonner  $\vec{U}_i$  de densité de probabilité  $\vec{u}' \mapsto \Phi(\vec{u}' \cdot \vec{u}_{i-1})$  : on obtient  $\vec{u}_i$ ;
17 |   sinon
18 |   |    $sortie \leftarrow oui$ ;
19 |   |    $l_{tot} \leftarrow l_{tot} + \min\{s' \in \mathbb{R}^+, \vec{r}_{i-1} + s' \vec{u}_{i-1} \in \text{parois}\}$ ;
20 |   |    $\Sigma \leftarrow \Sigma + \pi_{i-1}(l_{tot}^n)$ ;
21 jusqu'à  $sortie = oui$ ;
22 retourner  $\Sigma$ ;
```

Algorithme 1.7 : Algorithme estimant ponctuellement le moment $\langle L^n \rangle$ décrit en section 1.5.4, en échantillonnant l'expression statistique (1.98).

avec

$$\forall i \in \mathbb{N}^*, Cp_i = \begin{cases} \left(\prod_{j=1}^i \frac{k_d}{k_t} \right) \frac{k_a}{k_t} & \text{si } i < I_m \\ \prod_{j=1}^{I_m} \frac{k_d}{k_t} & \text{si } i = I_m \\ 0 & \text{si } i > I_m \end{cases} \quad (1.99b)$$

$$\forall i \in \mathbb{N}^*, Wp_i = \begin{cases} \left(\sum_{j=1}^i S_j \right)^n & \text{si } i < I_m \\ \left(s_m I_m + \sum_{j=1}^{I_m-1} S_j \right)^n = l_{tot}^n & \text{si } i \geq I_m \end{cases} \quad (1.99c)$$

●●● Il ressort des 2 formulations statistiques précédentes, 2 façons différentes de gérer l'absorption :

le « partitionnement de l'énergie » :
La contribution du chemin optique échantillonné est répartie entre plusieurs points qui l'émaillent. La 2^e formulation statistique proposée, associée

à l'algorithme 1.7, suit ce schéma.

la « roulette russe » : La contribution du chemin optique échantillonné se fait en une seule fois. Du coup, une fois qu'elle est faite on peut arrêter la construction du chemin, d'où le nom de roulette russe. La 1^{re} formulation statistique proposée, associée à l'algorithme

1.6, suit ce schéma.

Le terme de roulette russe s'applique aussi à une technique visant à arrêter arbitrairement l'évaluation d'une intégrale multidimensionnelle, sans créer de biais. À un certain moment, on effectue un test à 2 issues : survie ou interruption. Si on décide d'interrompre l'estimation, celle-ci est fixée à une valeur par défaut (0 par exemple). Si on décide de la continuer, sa contribution est augmentée par division par la probabilité préfixée de survie.

Avec une roulette russe l'estimation est en moyenne plus rapide, mais sa variance augmente.

1.5.5 Calcul de sensibilités

●●●● Le calcul de sensibilités, en méthode de Monte-Carlo, revêt une forme particulière.

D'abord, il faut savoir que la MMC n'est de façon générale pas compatible avec la méthode des différences finies. Cette dernière consiste, pour résumer grossièrement, à utiliser, quand on veut calculer la sensibilité d'une grandeur q à un paramètre π autour de $\pi = \pi_0$,

une approximation du type :

$$\partial_\pi q(\pi = \pi_0) \simeq \frac{q(\pi = \pi_0 + \delta\pi) - q(\pi = \pi_0)}{\delta\pi} \quad (1.100)$$

où $\delta\pi$ est une petite variation de π .

Cette approche est problématique combinée avec la MMC, car cette dernière génère inévitablement un bruit statistique. Si $\delta\pi$ est grand, l'approximation de la dérivée n'est pas bonne. Mais si $\delta\pi$ est petit, la différence entre $q(\pi = \pi_0 + \delta\pi)$ et $q(\pi = \pi_0)$ est recouverte par l'incertitude des estimations.

●●●● Considérons les algorithmes d'estimation des moments $\langle L^n \rangle$ détaillés dans cette section. Dans tous les cas, on construit un chemin optique, depuis l'entrée d'un photon dans le milieu jusqu'à son arrêt. Ensuite ou en même temps, selon qu'il y a de l'absorption et surtout selon l'estimateur de $\langle L^n \rangle$ qu'on échantillonne, on en déduit une estimation pour $\langle L^n \rangle$. On peut constater aisément, en relisant les algorithmes, que plusieurs moments $\langle L^n \rangle$ (avec des n différents) peuvent être estimés simultanément, suite à la construction d'un même chemin optique.

Il en va de même pour d'éventuelles sensibilités paramétriques de $\langle L^n \rangle$: elles peuvent être calculées simultanément à $\langle L^n \rangle$ grâce au même jeu de chemins optiques.

●●●● Considérons un paramètre π dont dépendent simultanément les coefficients d'absorption k_a et de diffusion k_d , et aussi la fonction de phase de diffusion Φ . En reprenant les expressions (1.96) et en dérivant sous l'intégrale on obtient :

$$\partial_\pi \langle L^n \rangle = \int_{\partial\mathcal{M}} \frac{d\vec{r}}{S} \int_{2\pi^+} \frac{d\vec{u}}{\pi} (\vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r})) \partial_\pi g(\vec{r}; \vec{u}; t=0) \quad (1.101a)$$

$$\begin{aligned} \partial_\pi g(\vec{r}; \vec{u}; t) = & \int_0^{+\infty} ds k_t \exp(-k_t s) \left(H(s \geq s_m) \left(\frac{\partial_\pi k_t}{k_t} - (\partial_\pi k_t) s \right) (ct + s_m)^n + \right. \\ & H(s < s_m) \times \left(\frac{k_a}{k_t} \left(\frac{\partial_\pi k_a}{k_a} - (\partial_\pi k_t) s \right) (ct + s)^n + \frac{k_d}{k_t} \int_{4\pi} d\vec{u}' \Phi(\vec{u}' \cdot \vec{u}) \times \right. \\ & \left. \left. \left(\left(\frac{\partial_\pi k_d}{k_d} - (\partial_\pi k_t) s + \frac{\partial_\pi \Phi(\vec{u}' \cdot \vec{u})}{\Phi(\vec{u}' \cdot \vec{u})} \right) g(\vec{r} + s\vec{u}; \vec{u}'; t + s/c) + \partial_\pi g(\vec{r} + s\vec{u}; \vec{u}'; t + s/c) \right) \right) \right) \end{aligned} \quad (1.101b)$$

Ceci peut se traduire statistiquement, par exemple, par l'estimateur $\widetilde{\langle L^n \rangle}_{\partial\pi}$ suivant de $\partial_\pi \langle L^n \rangle$:

$$\widetilde{\langle L^n \rangle}_{\partial\pi} = \widetilde{G}_{\partial\pi}(\vec{R}; \vec{U}; 0) \quad (1.102a)$$

$$\begin{aligned}
\tilde{G}_{\partial\pi}(\vec{r}; \vec{u}; t) = & \mathbb{H}(S \geq s_m) \left(\frac{\partial_\pi k_t}{k_t} - (\partial_\pi k_t) S \right) (ct + s_m)^n + \\
& \mathbb{H}(S < s_m) \left(\mathbb{H}(\text{Col} = \text{abs}) \left(\frac{\partial_\pi k_a}{k_a} - (\partial_\pi k_t) S \right) (ct + S)^n + \mathbb{H}(\text{Col} = \text{diff}) \times \right. \\
& \left. \left(\left(\frac{\partial_\pi k_d}{k_d} - (\partial_\pi k_t) S + \frac{\partial_\pi \Phi(\vec{U}' \cdot \vec{u})}{\Phi(\vec{U}' \cdot \vec{u})} \right) \tilde{G}(\vec{r} + S\vec{u}; \vec{U}'; t + S/c) + \tilde{G}_{\partial\pi}(\vec{r} + S\vec{u}; \vec{U}'; t + S/c) \right) \right)
\end{aligned} \tag{1.102b}$$

où les VA \vec{R} , \vec{U} , S , Col , et \vec{U}' suivent les mêmes lois analogues qu'en (1.97).

L'algorithme 1.8 propose une estimation simultanée de $\langle L^n \rangle$ et $\partial_\pi \langle L^n \rangle$, en se basant sur les formulations statistiques (1.97) et (1.102).

Sorties : Une estimation ponctuelle de $\langle L^n \rangle$ et $\partial_\pi \langle L^n \rangle$

```

1   $i \leftarrow 0$ ;
2   $l_{tot} \leftarrow 0$ ;
3   $\Pi_{\partial\pi} \leftarrow 0$ ;
4  échantillonner  $\vec{R}_0$  sur  $\partial\mathcal{M}$  de façon uniforme : on obtient  $\vec{r}_0$ ;
5  échantillonner  $\vec{U}_0$  sur  $\{\vec{u} \in E_u, \vec{u} \cdot \vec{n}_{\partial\mathcal{M}}(\vec{r}_0) > 0\}$  de façon lambertienne : on obtient  $\vec{u}_0$ ;
6  répéter
7       $i \leftarrow i + 1$ ;
8      échantillonner  $S$  de loi exponentielle de constante  $k_t$  : on obtient  $s_i$ ;
9      si  $\{\vec{r}_{i-1} + s' \vec{u}_{i-1}, s' \in [0; s_i]\} \cap \partial\mathcal{M} = \emptyset$  alors
10          $l_{tot} \leftarrow l_{tot} + s_i$ ;
11          $\vec{r}_i \leftarrow \vec{r}_{i-1} + s_i \vec{u}_{i-1}$ ;
12         échantillonner  $R$  de loi uniforme standard : on obtient  $r_i$ ;
13         si  $r_i < k_d/k_t$  alors
14              $\text{sortie} \leftarrow \text{non}$ ;
15             échantillonner  $\vec{U}_i$  de densité de probabilité  $\vec{u}' \mapsto \Phi(\vec{u}' \cdot \vec{u}_{i-1})$  : on obtient  $\vec{u}_i$ ;
16              $\Pi_{\partial\pi} \leftarrow \Pi_{\partial\pi} + \partial_\pi \ln k_d - s_i \partial_\pi k_t + \partial_\pi \ln \Phi(\vec{u}_i \cdot \vec{u}_{i-1})$ ;
17         sinon
18              $\text{sortie} \leftarrow \text{oui}$ ;
19              $\Pi_{\partial\pi} \leftarrow \Pi_{\partial\pi} + \partial_\pi \ln k_a - s_i \partial_\pi k_t$ ;
20     sinon
21          $\text{sortie} \leftarrow \text{oui}$ ;
22          $l_{tot} \leftarrow l_{tot} + \min\{s' \in \mathbb{R}^+, \vec{r}_{i-1} + s' \vec{u}_{i-1} \in \text{parois}\}$ ;
23          $\Pi_{\partial\pi} \leftarrow \Pi_{\partial\pi} + \partial_\pi \ln k_t - s_i \partial_\pi k_t$ ;
24 jusqu'à  $\text{sortie} = \text{oui}$ ;
25 retourner  $l_{tot}^n$  et  $\Pi_{\partial\pi}(l_{tot}^n)$ ;

```

Algorithme 1.8 : Algorithme estimant ponctuellement $\langle L^n \rangle$ et $\partial_\pi \langle L^n \rangle$ décrits en section 1.5.4 et 1.5.5, en échantillonnant les expressions statistiques (1.97) et (1.102).

•••• Cette démarche n'est pas seulement applicable à l'exemple de cette section, mais à tous les algorithmes de Monte-Carlo, tant que :

— le paramètre de dérivation n'intervient

pas dans l'expression du domaine d'intégration,

— l'intégrande ne dépend pas de façon discontinue du paramètre.

La récursivité de l'expression intégrale évaluée n'est pas un problème.

Quand on est dans l'un des cas listés à l'instant, le travail de formulation intégrale de la sensibilité est plus difficile. Il a néanmoins été abordé avec succès au cours de plusieurs travaux, dont [29, 44, 101].

Tout cela me fait dire qu'en matière de calcul de sensibilités, la MMC n'est pas en reste

par rapport à d'autres méthodes numériques, comme les méthodes avec maillage. Au cours de ce manuscrit, nous reviendrons de temps en temps sur la possibilité de calcul de sensibilités, afin de montrer qu'elle n'est pas obstruée au cours des développements méthodologiques orientés vers le transport non-linéaire.

Chapitre 2

Algorithmes à Collisions Nulles

Quand on met en place un calcul numérique par MMC pour un problème de transport linéaire, si le coefficient d'extinction est variable, alors on est presque inmanquablement gêné par le terme exponentiel d'intégrale manifestant la loi de Beer.

Les algorithmes à collisions nulles (ACNs) proposent une façon simple d'effectuer l'échantillonnage des libres parcours, dans cette situation. Leur principe est d'ajouter au problème physique des collisionneurs en quantité arbitraire, qui n'ont pas d'effet sur l'état des particules suivies, et en quantité adéquate pour rendre le coefficient d'extinction constant.

Les ACNs restent fonctionnels même quand le coefficient d'extinction n'est pas calculé à l'avance. Ceci a été utilisé en transfert radiatif dans les gaz, où le coefficient d'absorption est très variable et très difficile à calculer exactement. La méthode de Monte-Carlo prend alors le tout dans un seul calcul : elle intègre de façon combinée le modèle du transfert radiatif et le modèle de la dépendance du coefficient d'absorption aux conditions thermodynamiques [49]. Dans la suite de ce manuscrit, nous utiliserons cette propriété en cinétique des gaz, où l'extinction le long d'un chemin est de prime abord inconnue car dépendante de la solution du problème.

Les Algorithmes à Collisions Nulles (ACNs) sont, basiquement, une reformulation de la version intégrale des équations de transport. Ils sont connus depuis les années 1960, et depuis sont largement utilisés dans plusieurs disciplines de la physique du transport, comme la physique des plasmas, la neutronique, ou la synthèse d'image. Ils y servent presque exclusivement de technique algorithmique permettant de calculer des grandeurs liées au transport quand les coefficients d'extinction sont variables. Cependant, les possibilités qu'ils offrent sont plus larges que cela ; en particulier, ils sont au cœur de la technique algorithmique et du point de vue statistique sur la cinétique des gaz développés dans la suite de ce manuscrit.

Ce chapitre a pour but d'introduire petit-à-petit la majeure partie des propriétés et intérêts des Algorithmes à Collisions Nulles. Il balaye depuis le plus simple et le plus classique, jusqu'à certains développements récents utilisés dans la suite du manuscrit. Dans la section 2.1, on part d'une image physique simple, amenant aux formes les plus utilisées des Algorithmes à Collisions Nulles, pour ensuite développer mathématiquement et exhiber dans la section 2.2 des propriétés augmentant encore leurs caractères pratiques avantageux. Enfin, dans la section 2.5, on évoque le travail de thèse de Mathieu GALTIER [49] ; il montre que le transport corpusculaire est encore conceptualisable et calculable quand l'extinction balistique est inconnue en premier lieu — c'est-à-dire quand le coefficient d'extinction k_t résulte d'un modèle, et n'est pas pré-calculé — ; cette capacité sera centrale dans la façon dont nous aborderons la cinétique des gaz.

2.1 Les ACNs comme technique d'échantillonnage des libres parcours

2.1.1 Bref historique

Les Algorithmes à Collisions Nulles ont été expliqués pour la 1^{re} fois indépendamment par au moins 3 chercheurs à la fin des années 1960 : E. WOODCOCK en 1965 [122], neutronicien ; H.D. REES en 1968 [98], physicien des semi-conducteurs ; et H.R. SKULLERUD en 1968 [108], plasmicien. Chacun est considéré comme le découvreur de la méthode par sa propre communauté ; de même, les travaux qui en ont découlé se divisent en 3 ensembles indépendants qui semblent mutuellement s'ignorer jusqu'à aujourd'hui.

H.R. SKULLERUD aborde pour la 1^{re} fois dans [108] un Algorithme à Collisions Nulles, sans encore le dénommer ainsi, dans le but de pouvoir échantillonner des temps de vol libre pour des ions dans un gaz soumis à un champ électrique. De nombreux travaux vont suivre, comme [6, 14, 20, 57, 79, 80], consacrés essentiellement à la même problématique. Les ACNs s'y rencontrent essentiellement sous cette dénomination, ou plus rarement sous le terme d'« événements nuls » ou de « pseudocollisions ». Aujourd'hui, ils constituent une technique standard dans la physique des plasmas. Ils ont aussi été importés par la communauté s'intéressant aux gaz raréfiés — voir les publications [67, 72].

Par ailleurs, la technique développée par E. WOODCOCK sera étendue en 1968 par W. COLEMAN [26], et ensuite intensivement utilisée dans les sciences nucléaires. Parmi les travaux qui l'utilisent on peut citer [1, 81, 82, 100, 112]. On sait même qu'avant la publication de E. WOODCOCK, souvent citée comme référence, des techniques semblables étaient déjà utilisées dans les codes de calcul neutronique ; par exemple le rapport [21] en témoigne. L'utilisation des ACNs dans cette communauté est si répandue qu'ils sont nativement implémentés dans plusieurs codes de calcul pour le transport corpusculaire, comme SERPENT [75–77] ou MORET [47, 85]. Ces formes algorithmiques ont aussi diffusé vers d'autres domaines applicatifs, comme

la radiothérapie [5, 65], la tomographie [99, 121], et la synthèse d'image [89, 115, 116]. Dans la littérature consacrée à toutes ces sciences, les ACNs sont généralement désignés par le terme « Woodcock tracking », ou aussi « delta-scattering », « pseudo-scattering », « fictitious collisions », « fictitious scattering », « fictitious interaction tracking », « self-scattering », « hole-tracking »...

En physique des solides, H.D. REES crée et explique la technique dans [97, 98], sous le terme de « self scattering ». Cet apport sera très vite repris, par exemple dans des travaux sur les diodes Gunn [10, 46], puis largement employé dans la communauté. Nous n'avons hélas, par manque de temps, pas poussé plus loin la bibliographie sur les ACNs dans ce champ disciplinaire.

D'après ce que nous avons pu constater, ces différents groupes d'utilisateurs des ACNs ont rarement eu conscience de leur existence mutuelle. La seule publication que nous connaissons faisant référence à des travaux de ces trois ensembles bibliographiques simultanément est [14]. Avant, nous observons quelques croisements au sujet des ACNs entre les plasmiciens et les physiciens des semi-conducteurs, comme dans [6]. Nettement plus tard, nous savons que l'auteur Alfred I. KHISAMUTDINOV a cité les ensembles bibliographiques neutronique et plasma. Finalement, ces deux ensembles sont à nouveau cités dans [50], dont ce travail est une suite.

2.1.2 La difficulté que l'on veut surmonter

●●●● Supposons que l'on veuille calculer quelque grandeur, dans un problème de transfert radiatif. Comme on utilise une approche mésoscopique, cette grandeur est une valeur particulière de la fonction de distribution f , ou bien une intégrale de f (voir le chapitre 1.1).

Comme expliqué au chapitre 1.2.1, dans un milieu isotrope et en équilibre thermodynamique local, à indice de réfraction constant, la fonction de distribution des photons en un point se calcule comme :

$$\begin{aligned}
 f_\nu(\vec{r}; \vec{u}; \nu; t) = & \int_0^{s_m} ds k_t(\vec{r} - s\vec{u}; \vec{u}; \nu; t - \frac{s}{c}) \exp\left(-\int_0^s ds' k_t(\vec{r} - s'\vec{u}; \vec{u}; \nu; t - \frac{s'}{c})\right) \times \\
 & \left(\frac{k_a(\vec{r} - s\vec{u}; \vec{u}; \nu; t - \frac{s}{c})}{k_t(\vec{r} - s\vec{u}; \vec{u}; \nu; t - \frac{s}{c})} f_{\nu,eq}\left(T(\vec{r} - s\vec{u}; t - \frac{s}{c}); \nu\right) + \frac{k_d(\vec{r} - s\vec{u}; \vec{u}; \nu; t - \frac{s}{c})}{k_t(\vec{r} - s\vec{u}; \vec{u}; \nu; t - \frac{s}{c})} \right. \\
 & \left. \int_{4\pi} d\vec{u}' \Phi(\vec{r} - s\vec{u}; \vec{u} \cdot \vec{u}'; \nu; t - \frac{s}{c}) f_\nu(\vec{r} - s\vec{u}; \vec{u}'; \nu; t - \frac{s}{c}) \right) + \\
 & \exp\left(-\int_0^{s_m} ds k_t(\vec{r} - s\vec{u}; \vec{u}; \nu; t - \frac{s}{c})\right) f_\nu(\vec{r} - s_m\vec{u}; \vec{u}; \nu; t - \frac{s_m}{c}) \quad (2.1)
 \end{aligned}$$

où $s_m = \min\{s \in \mathbb{R}^+, \vec{r} - s\vec{u} \in \text{parois}\}$ est la distance de la plus proche paroi le long du rayon $\mathcal{R}(\vec{r}; -\vec{u})$.

En supposant que $\int_0^{+\infty} ds k_t(\dots) = +\infty$, l'expression précédente peut être

ré-écrite :

$$f_\nu(\vec{r}; \vec{u}; \nu; t) = \int_0^{+\infty} ds k_t(\vec{r} - s\vec{u}; \vec{u}; \nu; t - \frac{s}{c}) \exp\left(-\int_0^s ds' k_t(\vec{r} - s'\vec{u}; \vec{u}; \nu; t - \frac{s'}{c})\right) \times \\ \left(H(s < s_m) \left(\frac{k_a(\vec{r} - s\vec{u}; \vec{u}; \nu; t - \frac{s}{c})}{k_t(\vec{r} - s\vec{u}; \vec{u}; \nu; t - \frac{s}{c})} f_{\nu,eq}\left(T(\vec{r} - s\vec{u}; t - \frac{s}{c}); \nu\right) + \right. \right. \\ \left. \frac{k_d(\vec{r} - s\vec{u}; \vec{u}; \nu; t - \frac{s}{c})}{k_t(\vec{r} - s\vec{u}; \vec{u}; \nu; t - \frac{s}{c})} \int_{4\pi} d\vec{u}' \Phi(\vec{r} - s\vec{u}; \vec{u} \cdot \vec{u}'; \nu; t - \frac{s}{c}) f_\nu(\vec{r} - s\vec{u}; \vec{u}'; \nu; t - \frac{s}{c}) \right) + \\ \left. H(s \geq s_m) f_\nu\left(\vec{r} - s_m\vec{u}; \vec{u}; \nu; t - \frac{s_m}{c}\right) \right) \quad (2.2)$$

Pour des questions de simplicité, on se limitera dans la suite de ce chapitre à un cas monochromatique, en régime stationnaire, dans un milieu purement absorbant et émissif, le long d'une ligne de visée terminée par une paroi opaque. Le lecteur constatera au fur et à mesure de ce chapitre que ce cas physique est suffisamment représentatif des difficultés physico-mathématiques que nous pouvons dépasser grâce aux ACNs. Dans ce cas restreint, la formule précédente se simplifie en :

$$f(s) = \left(\int_s^{s_m} ds' k_a(s') \exp\left(-\int_s^{s'} ds'' k_a(s'')\right) f_{eq}(s') \right) + \\ \exp\left(-\int_s^{s_m} ds' k_a(s')\right) f(s_m) \quad (2.3)$$

où $f_{eq}(s) = f_{eq}(T(s))$, et s est l'abscisse le long du rayon, avec $s \leq s_m$. Comme précédemment, cela peut aussi se ré-écrire comme :

$$f(s) = \int_s^{+\infty} ds' k_a(s') \exp\left(-\int_s^{s'} ds'' k_a(s'')\right) \times \\ \left(H(s' < s_m) f_{eq}(s') + H(s' \geq s_m) f(s_m) \right) \quad (2.4)$$

Il s'agit maintenant d'être capable d'exploiter ces précédentes expressions, en particulier pour le calcul.

●●●● De la même façon qu'au chapitre 1.5, l'équation (2.4) peut être convertie sous forme statistique. Pour ce faire, on opte ici pour une approche intégrale ; mais une vision analogue peut être utilisée pour poser la variable aléatoire S' , qui correspondra à la longueur parcourue avant collision.

Avec une variable aléatoire analogue :

$$f(s) = E\left(H(S' < s_m) f_{eq}(S') + H(S' \geq s_m) f(s_m)\right) \quad (2.5a)$$

où S' est une variable aléatoire à valeurs dans $[s; +\infty[$ et de fonction de répartition :

$$F_{S'}(s') = 1 - \exp\left(-\int_s^{s'} ds'' k_a(s'')\right) \quad (2.5b)$$

S' est bien un libre parcours d'absorption.

Remarque. $F_{S'}$ est bien une fonction de répartition sur $[s; +\infty[$ ssi $\int_s^{s'} ds'' k_a(s'') \xrightarrow{s' \rightarrow +\infty} +\infty$. Par ailleurs, comme l'éventuel milieu derrière le mur n'a pas d'influence directe dans l'expression ci-dessus :

- On peut temporairement fixer la valeur de $k_a(s')$ pour $s' \geq s_m$ à n'importe quelle valeur arbitraire strictement positive, pour l'évaluation de $f(s)$. Cette considération permet de traiter les cas où la valeur de $k_a(s')$ pour $s' \geq s_m$ n'est pas définie *a priori*.
- Dans la formulation (2.5a), on peut très bien poser :

$$F_{S'} : \begin{cases} \mathbb{R} & \rightarrow [0; 1] \\ s' & \mapsto 0 & \text{si } s' \leq s \\ s' & \mapsto 1 - \exp\left(-\int_s^{s'} ds'' k_a(s'')\right) & \text{si } s \leq s' < s_m \\ s' & \mapsto 1 & \text{si } s_m \leq s' \end{cases}$$

cela ne change rien.

Approche intégrale arbitraire :

$$f(s) = E\left(\frac{\exp\left(-\int_0^{s'} ds'' k_a(s'')\right)}{p_{S'}(s')}\left(H(S' < s_m)f_{eq}(S') + H(S' \geq s_m)f(s_m)\right)\right) \quad (2.6)$$

où S' est une variable aléatoire à valeurs dans $[s; +\infty[$ et de densité de probabilité arbitraire (mais strictement positive). Choisir S' avec une répartition éloignée du libre parcours d'absorption présente des risques d'augmentation de la variance, comme expliqué au chapitre 1.3.2.

Remarque. Dans les expressions (2.5) et (2.6), on peut remplacer $f(s_m)$ par un estimateur sans biais $\tilde{F}(s_m)$, car $f(s)$ dépend linéairement de $f(s_m)$. Dans ce cas, les échantillonnages de $\tilde{F}(s_m)$ et de S' doivent être indépendants.

●●●● Disposant de l'expression statistique (2.5) ou (2.6) de $f(s)$, il est naturel de s'en servir pour monter un algorithme de Monte-Carlo visant à l'évaluer. En pratique, la difficulté se situera en 2 points :

- L'évaluation de s_m , qui peut être compliquée si la géométrie du problème l'est. Nous bénéficions ici toutefois des apports de la communauté de la synthèse d'image, pour qui le calcul de l'intersection entre un rayon et une géométrie est un problème fondamental et extensivement étudié.

- Le tirage de la VA S' si on utilise l'expression (2.5), ou l'évaluation de l'exponentielle si on utilise l'expression (2.6).

Si k_a est une constante, la 2^e difficulté mentionnée n'en est pas une. En approche analogue, il suffit de poser $S' = -(\ln R)/k_a$ avec R de loi uniforme standard, comme nous l'avons déjà montré au chapitre 1.3.1. En approche intégrale, le calcul du terme exponentiel est immédiat.

En approche analogue, si k_a n'est plus uniforme il faut d'abord tirer une épaisseur optique traversée $EO = -\ln R$, puis retrouver l'abscisse curviligne correspondante $S' \in [s; +\infty[$ telle que $\int_s^{S'} ds'' k_a(s'') = EO$. Dans la pratique, plusieurs démarches apparaissent :

- Si on dispose d'une formule donnant k_a en fonction de l'abscisse curviligne s' , on peut parfois réaliser l'inversion-intégration $EO \mapsto S'$ de manière symbolique puis implémenter le résultat. On peut être dans cette situation si la formule $s \mapsto k_a$ est simple sur l'ensemble du domaine d'intérêt (cas académiques), ou est définie par morceaux (cas des maillages).
- Dans le cas précédent, si on est capable d'exprimer l'épaisseur optique traversée eo en fonction de s' mais qu'on n'arrive pas à exprimer la réciproque $eo \mapsto s'$, on peut avoir recours à une procédure d'inversion numérique. Il n'y a pas là de grande difficulté théorique, mais les calculs peuvent être sensiblement rallongés.
- En dernier lieu, on pourra approximer le profil $s' \mapsto k_a$ par une fonction se prêtant aisément au traitement symbolique — intégration puis inversion —, comme une spline cubique. La méthode de Monte-Carlo perd alors son caractère non-biaisé, à moins de mettre en place de complexes contrôles de la précision.

Ces précédentes méthodes ont d'importants défauts. Parmi eux :

- L'inversion symbolique de la primitive du profil d'absorption n'est en général pas réutilisable. C'est-à-dire que dès que le type du profil (ou de la fonction d'interpolation, dans le cas d'un maillage) est modifié, le travail mathématique d'inversion doit être recommencé.
- Les procédures d'approximation (qui reposent d'habitude sur une reconstitution du profil d'absorption sur un maillage le long du rayon) sont naturellement laborieuses. Quand un contrôle de précision est utilisé, cela devient encore pire, car ce contrôle repose souvent sur une répétition de la procédure d'approximation.
- Quand un maillage est utilisé, la procédure de suivi d'un rayon dans le maillage peut être lourde (informatiquement parlant). Cela se fait sentir typiquement quand le transfert radiatif est couplé avec la mécanique des fluides, par exemple dans un problème de combustion. La CFD (Computational Fluid Dynamics) utilise ordinairement des maillages de plusieurs millions de mailles, soit approximativement $100 \times 100 \times 100$ mailles. Dans ces cas, une pratique

existante consiste à utiliser un maillage différent pour le calcul du transfert radiatif, plus lâche que celui servant aux calculs de mécanique des fluides. Cependant, lorsqu'on calcule le transfert radiatif par MMC sur un maillage et que l'on veut être précis :

- soit le maillage est fin, et donc les mailles à traverser sont nombreuses,
- soit la fonction d'interpolation permet de modéliser des profils complexes dans chaque maille, et donc les calculs à faire dans chaque maille sont plus coûteux.

En approche purement intégrale, c'est-à-dire en utilisant une VA S' arbitraire, la situation est relativement semblable bien que l'échantillonnage de S' ne soit *a priori* plus une difficulté ; car il faut connaître le terme $\exp\left(-\int_0^{s'} ds'' k_a(s'')\right)$. Estimer l'intégrale interne (par une méthode de Monte-Carlo) n'est pas une bonne technique, car la fonction exponentielle n'est pas affine : elle est convexe et donc amplifie le bruit statistique davantage vers le haut que vers le bas, d'où un biais de calcul. On doit donc connaître l'intégrale interne exactement, ce qui nous ramène à peu de choses près à la liste des cas de figure précédente.

Une solution serait d'utiliser la méthode présentée au chapitre 3, et d'utiliser un Développement en Série Entière de la fonction exponentielle. On serait alors capable d'estimer le terme $\exp\left(-\int_0^{s'} ds'' k_a(s'')\right)$ sans biais. En fait, les formes mathématiques qui en résultent sont proches des Algorithmes à Collisions Nulles ; l'annexe B donne plus de détails à ce sujet.

●●●● Ces difficultés sont connues depuis longtemps des physiciens du transport corpusculaire, même si elles peuvent prendre des formes légèrement différentes. Par exemple, en neutronique, en transfert radiatif, ou en synthèse d'image ce sont les inhomogénéités du milieu qui posent problème. En physique des plasmas, l'échantillonnage des longueurs de collision peut rencontrer la difficulté explicitée précédemment même dans un milieu uniforme. En effet, si un champ électrique est présent, la vitesse des particules chargées évolue au cours de leurs trajectoires balistiques, ainsi que leur section efficace de collision avec les autres particules présentes.

Les Algorithmes à Collisions Nulles proposent de court-circuiter totalement ces difficultés.

2.1.3 Utilisation de collisionneurs fictifs

L'idée de base qui sous-tend ces formes algorithmiques est d'introduire dans le problème physique des collisionneurs supplémentaires, en quantité arbitraire et ajustable, qui génèrent des *collisions sans effet*. On parle alors de *collisions nulles*.

Par exemple, dans un problème de transfert radiatif, on peut considérer que ces collisionneurs provoquent de la diffusion sans déviation des photons — d'où le

terme de « delta-scattering » : « delta » fait référence à la fonction de phase de diffusion, qui est ici un Dirac vers l'avant.

On passe ainsi de l'équation différentielle de transport (sans diffusion) :

$$\vec{u} \cdot \overrightarrow{\text{grad}}_{\vec{r}} f(\vec{r}; \vec{u}) = - \underbrace{k_a(\vec{r}) f(\vec{r}; \vec{u})}_{\text{atténuation}} + \underbrace{k_a(\vec{r}) f_{eq}(\vec{r})}_{\text{terme source}} \quad (2.7)$$

à :

$$\vec{u} \cdot \overrightarrow{\text{grad}}_{\vec{r}} f(\vec{r}; \vec{u}) = - \underbrace{(k_a(\vec{r}) + k_n(\vec{r})) f(\vec{r}; \vec{u})}_{\text{atténuation}} + \underbrace{k_a(\vec{r}) f_{eq}(\vec{r}) + k_n(\vec{r}) f(\vec{r}; \vec{u})}_{\text{terme source}} \quad (2.8)$$

qui revient, du point de vue du transport, strictement au même.

Par contre, du point de vue algorithmique, c'est tout-à-fait différent. On peut reprendre la même image de somme de contributions avec atténuation qu'au chapitre 1.2.1 pour obtenir une forme intégrale de notre version modifiée — mais exacte — de l'ETR (2.8) ; cette fois-ci on obtient :

$$f(s) = \int_s^{+\infty} ds' \hat{k}_a(s') \exp\left(-\int_s^{s'} ds'' \hat{k}_a(s'')\right) \times \left(H(s' < s_m) \left(\frac{k_a(s')}{\hat{k}_a(s')} f_{eq}(s') + \frac{k_n(s')}{\hat{k}_a(s')} f(s') \right) + H(s' \geq s_m) f(s_m) \right) \quad (2.9)$$

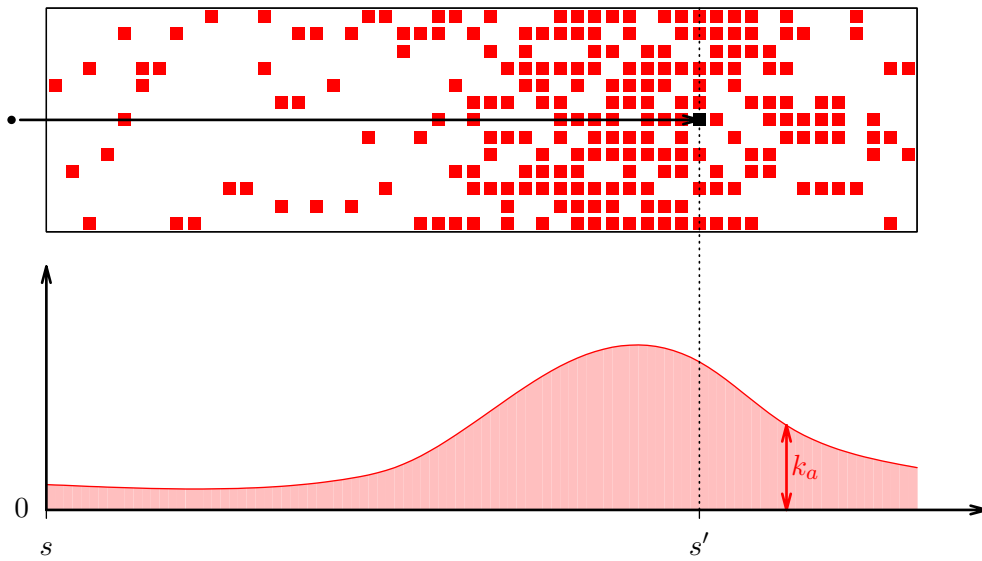
avec $\hat{k}_a = k_a + k_n$, appelé *coefficient d'absorption majoré*. Cette équation peut être convertie sous forme statistique, ici en utilisant des VAs analogues :

$$f(s) = E \left(H(S' < s_m) \left(H(Col = abs) f_{eq}(S') + H(Col = nul) \tilde{F}(S') \right) + H(S' \geq s_m) \tilde{F}(s_m) \right) \quad (2.10)$$

avec :

- S' une VA à valeurs dans $[s; +\infty[$, mais cette fois-ci de fonction de répartition $F_{S'} : s' \mapsto 1 - \exp\left(-\int_s^{s'} ds'' \hat{k}_a(s'')\right)$. Elle correspond au libre parcours d'extinction, compte tenu de l'absorption et des collisionneurs fictifs ajoutés.
- Col une VA à 2 valeurs possibles (*abs* et *nul*), prenant la valeur *abs* avec une probabilité $k_a(S')/\hat{k}_a(S')$. Elle correspond au type d'événement dont est issu le photon partant de S' : émission ou collision nulle.
- $\tilde{F}(s_m)$ un estimateur sans biais de $f(s_m)$.
- $\tilde{F}(S')$ un estimateur sans biais de $f(S')$. Cet estimateur est typiquement construit à l'aide de l'expression (2.10) elle-même, où la valeur de S' remplace s .

L'intérêt de cette nouvelle forme statistique/intégrale de f réside dans la loi de tirage de S' : S' suit désormais une loi exponentielle selon un coefficient d'extinction réglable, on peut choisir le profil et les valeurs de $\hat{k}_a(s')$ que l'on souhaite (un profil constant, par exemple). La complexité de l'extinction est désormais renvoyée dans la récursivité de l'expression intégrale ou statistique. La seule contrainte restante est que k_n doit être positif, d'ailleurs nous verrons dans la section suivante que cette contrainte peut être levée elle aussi.



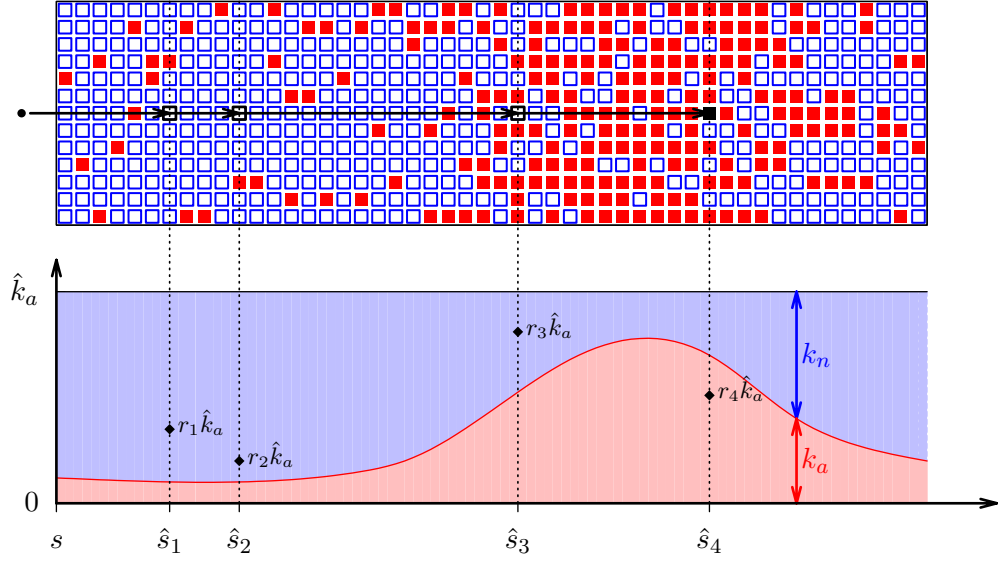
(a) Sans collisions nulles. Les centres absorbants sont figurés par des carrés rouges pleins.

On sait que l'épaisseur optique traversée suit, statistiquement parlant, une loi exponentielle unité. Une fois que l'on a choisi une épaisseur optique eo , il faut lui faire correspondre s' telle que l'aire sous la courbe rouge entre s et s' vaille bien eo . Selon la forme du profil de k_a , cela peut être très difficile.

FIGURE 2.1 – Échantillonnage de libres parcours d'absorption dans une colonne absorbante unidimensionnelle, effectué avec — (b), page suivante — ou sans — (a) — collisions nulles.

●●●● À titre d'illustration, la structure d'un algorithme à collisions nulles pour l'estimation ponctuelle de $f(s)$ est donnée en algorithme 2.2, à comparer à la structure d'un algorithme plus ordinaire donnée en algorithme 2.1¹. La figure 2.1 propose une image physique du fonctionnement de ces algorithmes.

1. Ces algorithmes peuvent être nettement améliorés pour éviter des calculs inutiles. En particulier, dans l'algorithme 2.2, un test $s_{cur} < s_m$ peut être placé dans la boucle pour éviter de continuer à échantillonner des libres parcours alors qu'on a déjà heurté la paroi. Cela n'a pas été fait ici afin de ne garder dans l'algorithme 2.2 que la substantifique moelle des algorithmes de type « à collisions nulles ».



(b) Avec collisions nulles. Des collisionneurs fictifs représentés par des carrés bleus creux sont ajoutés, ici de sorte que la densité totale de collisionneurs soit constante.

Il est désormais facile de déterminer \hat{s}_1 : l'épaisseur optique traversée correspond à la surface rouge et bleue entre s et \hat{s}_1 , rectangulaire. Par contre l'extinction peut être due à une absorption ou une collision nulle ; il faut donc faire un choix, ici guidé par les densités respectives d'absorbeurs et de collisionneurs fictifs (figurées par les aires rouge et bleue). Si une collision nulle est choisie, le photon suivi poursuit son chemin.

FIGURE 2.1 – L'ajout de collisionneurs fictifs rend récursive la procédure de choix d'une longueur d'absorption ; par contre, chaque itération de la procédure est facile. Au total, on peut gagner en facilité de calcul grâce à cet ajout.

Figure inspirée de la publication [89].

On voit que la différence se situe uniquement dans le calcul de s' ou s_{cur} . Comme les espérances de \tilde{f} sont égales à la fin des 2 algorithmes quel que soit le profil de f_{eq} , cela signifie que *l'échantillonnage de s_{cur} dans l'algorithme 2.2 est équivalent à l'échantillonnage de s' dans l'algorithme 2.1.* De manière plus formelle, en posant les variables aléatoires :

- S' de fonction de répartition $F_{S'}(s') = 1 - \exp\left(-\int_s^{s'} ds'' k_a(s'')\right)$,
- $\hat{S}_0 = s$,
- $\forall i \in \mathbb{N}, \hat{S}_{i+1}$ de fonction de répartition $F_{\hat{S}_{i+1}}(\hat{s}) = 1 - \exp\left(-\int_{\hat{S}_i}^{\hat{s}} ds'' \hat{k}_a(s'')\right)$,
- et I_{abs} à valeurs dans \mathbb{N}^* , telle que $\forall i \in \mathbb{N}^*, \Pr(I_{abs} = i \mid I_{abs} \geq i) = k_a(\hat{S}_i) / \hat{k}_a(\hat{S}_i)$,

$\hat{S}_{I_{abs}}$ est identiquement distribué à S' .

Cela permet d'affirmer que du point de vue numérique, dans leur forme la plus simple, les Algorithmes à Collisions Nulles ne constituent ni plus ni moins

Résultat : Une estimation \tilde{f} de $f(s)$

- 1 échantillonner EO selon une loi exponentielle unité : on obtient eo ;
// échantillonnage d'une épaisseur optique traversée
- 2 calculer s' tq $\int_s^{s'} ds'' k_a(s'') = eo$; // **POTENTIELLEMENT DIFFICILE**
- 3 si $s' < s_m$ alors $\tilde{f} \leftarrow f_{eq}(s')$ sinon $\tilde{f} \leftarrow f(s_m)$;

Algorithme 2.1 : Algorithme basique
et ordinaire pour l'estimation de $f(s)$

Résultat : Une estimation \tilde{f} de $f(s)$

- 1 $s_{cur} \leftarrow s$;
- 2 **répéter**
- 3 | échantillonner EO selon une loi exponentielle unité : on obtient eo ;
| // échantillonnage d'une épaisseur optique traversée
- 4 | calculer s' tq $\int_{s_{cur}}^{s'} ds'' \hat{k}_a(s'') = eo$; // **FACILE**
- 5 | $s_{cur} \leftarrow s'$;
- 6 | $P_{abs} \leftarrow k_a(s_{cur}) / \hat{k}_a(s_{cur})$;
- 7 | échantillonner R de loi uniforme standard : on obtient r ;
- 8 | si $r < P_{abs}$ alors $Col \leftarrow abs$ sinon $Col \leftarrow nul$;
- 9 jusqu'à $Col = abs$;
- 10 si $s_{cur} < s_m$ alors $\tilde{f} \leftarrow f_{eq}(s_{cur})$ sinon $\tilde{f} \leftarrow f(s_m)$;

Algorithme 2.2 : Algorithme à collisions
nulles basique pour l'estimation de $f(s)$

qu'une technique d'échantillonnage des libres parcours. Ils font penser en cela à une méthode de rejet. Nous verrons cependant dans la section suivante que sous des formes plus raffinées, leur applicabilité et leur utilité ne se réduit pas à une telle technique d'échantillonnage.

2.2 Fondements mathématiques

Dans la section précédente 2.1.3, nous avons justifié la formulation intégrale des ACNs (2.9) par l'image d'un problème de transport équivalent, où des collisionneurs sans effet sont ajoutés. En considérant ces collisionneurs fictifs comme des centres diffuseurs ne déviant en fait pas les photons, on peut se ramener au cas des milieux absorbants et diffusants ; la formulation intégrale de l'équation de transfert radiatif y est bien connue, ainsi que les formes algorithmiques de type Monte-Carlo associées. La question de la fonctionnalité de ces derniers algorithmes a déjà été traitée dans ce manuscrit au chapitre 1.5.4, et ne fait de toute façon plus de doute auprès de la communauté étudiant le transfert radiatif.

Néanmoins, une exploration purement mathématique de la formulation (2.9) est un passage obligé dans notre contexte :

- Elle permet d'en explorer pleinement la validité, indépendamment des images intuitives de la diffusion multiple, ce qui sera un élément de confiance important lorsque nous quitterons cet intuitif pour explorer la cinétique des gaz.
- Elle permet d'imaginer d'autres algorithmes, toujours basés sur la formulation (2.9), mais qui ne sont pas ancrés dans la pratique du transfert en milieu diffusant. On pourrait manquer ces astuces en raisonnant uniquement avec l'image d'une diffusion vers l'avant en tête.

2.2.1 Validité de la formulation intégrale

Rappelons l'équation simplifiée du cas que nous étudions :

$$\vec{u} \cdot \overrightarrow{\text{grad}_{\vec{r}}} f(\vec{r}; \vec{u}) = -k_a(\vec{r})f(\vec{r}; \vec{u}) + k_a(\vec{r})f_{eq}(\vec{r}) \quad (2.11)$$

Le long d'une ligne de visée selon $-\vec{u}$, cela se réduit à :

$$f'(s) = k_a(s)f(s) - k_a(s)f_{eq}(s) \quad (2.12)$$

avec la condition frontière : « $f(s_m)$ est connue ».

L'équation (2.9) est correcte si et seulement si l'expression qu'elle donne pour $f(s)$ satisfait à l'équation (2.12), ainsi qu'à la condition frontière. Faisons la vérification :

Preuve. L'équation (2.9) s'écrit :

$$f(s) = \int_s^{+\infty} ds' \hat{k}_a(s') \exp\left(-\int_s^{s'} ds'' \hat{k}_a(s'')\right) \times \\ \left(H(s' < s_m) \left(\frac{k_a(s')}{\hat{k}_a(s')} f_{eq}(s') + \frac{k_n(s')}{\hat{k}_a(s')} f(s') \right) + H(s' \geq s_m) f(s_m) \right)$$

ce qui se simplifie, si $\hat{k}_a(s')$ est non nul pour tout $s' \geq s$, en :

$$f(s) = \int_s^{+\infty} ds' \exp\left(-\int_s^{s'} ds'' \hat{k}_a(s'')\right) \times \\ \left(H(s' < s_m) \left(k_a(s') f_{eq}(s') + k_n(s') f(s') \right) + H(s' \geq s_m) \hat{k}_a(s') f(s_m) \right) \quad (2.13)$$

d'où (pour $s < s_m$) :

$$f'(s) = - \left(k_a(s) f_{eq}(s) + k_n(s) f(s) \right) + \\ \int_s^{+\infty} ds' \hat{k}_a(s) \exp\left(-\int_s^{s'} ds'' \hat{k}_a(s'')\right) \\ \left(H(s' < s_m) \left(k_a(s') f_{eq}(s') + k_n(s') f(s') \right) + H(s' \geq s_m) \hat{k}_a(s') f(s_m) \right) \quad (2.14)$$

ce qui aboutit bien à :

$$\begin{aligned} f'(s) &= - \left(k_a(s) f_{eq}(s) + k_n(s) f(s) \right) + \hat{k}_a(s) f(s) \\ &= k_a(s) f(s) - k_a(s) f_{eq}(s) \end{aligned}$$

L'expression (2.9) satisfait donc bien à l'équation différentielle (2.12) sur $[s; +\infty[$. En ce qui concerne la condition frontière, en reprenant l'équation (2.9) avec $s = s_m$:

$$\begin{aligned} f(s_m) &= \int_{s_m}^{+\infty} ds' \hat{k}_a(s') \exp\left(-\int_{s_m}^{s'} ds'' \hat{k}_a(s'')\right) f(s_m) \\ &= f(s_m) \quad \text{ssi} \quad \int_{s_m}^{s'} ds'' \hat{k}_a(s'') \xrightarrow{s' \rightarrow +\infty} +\infty \end{aligned} \quad \blacksquare$$

La démonstration que l'on vient d'effectuer ne laisse apparaître que 2 conditions nécessaires pour la validité de l'expression intégrale des ACNs (2.9). Ces 2 contraintes, en fait assez lâches, sont que :

- On doit avoir $\int_{s_m}^{s'} ds'' \hat{k}_a(s'') \xrightarrow{s' \rightarrow +\infty} +\infty$. Cette contrainte n'en est pas vraiment une, comme cela est expliqué précédemment à la section 2.1.2.
- $\hat{k}_a(s)$ doit être non-nulle en tout point. Si on se base directement sur l'expression (2.13), cette contrainte disparaît.

2.2.2 Transformation en expression statistique

●●● Pour obtenir une reformulation statistique générique de l'expression (2.9), utilisable en MMC, il nous faut introduire une variable aléatoire S' (pour l'instant arbitraire) sur $[s; +\infty[$. Alors :

$$\begin{aligned} f(s) &= \mathbb{E} \left(\frac{\hat{k}_a(S') \exp\left(-\int_s^{S'} ds'' \hat{k}_a(s'')\right)}{p_{S'}(S')} \left(\right. \right. \\ &\quad \left. \left. \begin{array}{c} \text{terme(s) de collision} \\ \underbrace{\left(\frac{k_a(S')}{\hat{k}_a(S')} f_{eq}(S') + \frac{k_n(S')}{\hat{k}_a(S')} \tilde{F}(S') \right)}_{\substack{\text{terme d'émission} \quad \text{terme de} \\ \text{collision nulle}}} \right) + \mathbb{H}(S' \geq s_m) \tilde{F}(s_m) \right) \right) \end{aligned} \quad (2.15)$$

où $\tilde{F}(s)$ est un estimateur sans biais de $f(s)$.

Deux remarques sont à faire à ce stade :

- Nous n'avons pas introduit de variable aléatoire pour le choix d'un type de collision, comme en (2.10). Comme il n'y a qu'un nombre fini de termes de collision, cela n'est *a priori* pas nécessaire pour que l'expression (2.15) soit utilisable directement dans un algorithme de type Monte-Carlo.

Dans la pratique, une telle procédure de choix peut redevenir nécessaire. En effet, parmi les 2 termes, celui de collision nulle est récursif — le terme

d'émission fait intervenir les propriétés locales du milieu, et n'est donc en général pas récursif. Pour que l'algorithme puisse s'effectuer, il faut donc un moyen d'interrompre cette récursivité. Cela peut souvent être réalisé par la condition frontière, mais introduire un choix du type de collision est aussi une solution très utilisée (et souvent pertinente).

- Aucune hypothèse n'est ici encore faite sur \hat{k}_a et $p_{S'}$. On peut ainsi aboutir à des formes algorithmiques tout-à-fait différentes de ce que nous allons présenter dans la suite de ce chapitre.

●●●● Pour la suite de ce chapitre, on reprendra S' comme une longueur d'extinction obéissant à une loi exponentielle de coefficient \hat{k}_a . Le but de ce choix algorithmique est de faire diminuer la variance d'une estimation utilisant l'expression (2.15), comme cela est expliqué au chapitre 1.5.3.

Ce choix élimine plusieurs optimisations concevables pour un algorithme MC calculant $f(s)$. Par exemple un tirage stratifié de S' , d'un côté sur $[s; s_m[$ et de l'autre sur $[s_m; +\infty[$; l'intégrale de la 2^e strate est d'ailleurs calculable symboliquement. Néanmoins, l'approche algorithmique consistant à utiliser \hat{k}_a pour tirer des longueurs d'extinction semble tout-à-fait suffisante pour les besoins qui seront explicités dans ce manuscrit.

Ainsi, la formulation statistique générale pour les Algorithmes à Collisions Nulles que nous utiliserons devient :

$$f(s) = E \left(H(S' < s_m) \left(\frac{k_a(S')}{\hat{k}_a(S')} f_{eq}(S') + \frac{k_n(S')}{\hat{k}_a(S')} \tilde{F}(S') \right) + H(S' \geq s_m) \tilde{F}(s_m) \right) \quad (2.16a)$$

avec

$$S' \text{ à valeurs dans } [s; +\infty[\text{ et } F_{S'}(s') = 1 - \exp\left(-\int_s^{s'} ds'' \hat{k}_a(s'')\right) \quad (2.16b)$$

2.2.3 Partitionnement de l'énergie *vs.* roulette russe

●●●● Lorsqu'on calcule un estimateur $\tilde{F}(s)$ de $f(s)$ construit selon l'expression (2.16), si le point de collision échantillonné est à l'intérieur du milieu (c'est-à-dire si $S' < s_m$), on doit calculer la somme des contributions du terme de collision nulle et du terme d'émission. Ce calcul peut être effectué de plusieurs façons :

1. Soit on calcule la somme, « tout simplement ». On doit alors calculer localement $k_a(S')$ et $f_{eq}(S')$, et récursivement $\tilde{F}(S')$.
2. Soit, comme cela est proposé dans la formulation (2.10), on choisit le terme d'émission avec une probabilité $k_a(S')/\hat{k}_a(S')$ ou alors le terme de collision nulle. Puis on considère que $\tilde{F}(s) = f_{eq}(S')$ si on a choisi le terme d'émission, ou $\tilde{F}(s) = \tilde{F}(S')$ si on a choisi le terme de collision nulle. On a donc besoin de calculer d'abord $k_a(S')$ puis, selon le choix qu'on a fait, $f_{eq}(S')$ ou $\tilde{F}(S')$.

On retrouve ici l'opposition entre *partitionnement de l'énergie* (la 1^{re} méthode) et *roulette russe* (la 2^e méthode), déjà mentionnée au chapitre 1.5.4. La 2^e technique est bien une roulette russe, car si on choisit le terme d'émission les calculs nécessaires sont uniquement locaux, et le suivi de photon peut se terminer.

3. Si on utilise une roulette russe — c'est-à-dire une épreuve de Bernoulli entre le terme d'émission et le terme de collision nulle — on peut poser arbitrairement les probabilités de choisir chacun des termes. En effet, l'équation (2.16) peut être ré-écrite :

$$\begin{aligned}
 f(s) &= \mathbb{E} \left(\mathbb{H}(S' < s_m) \left(\frac{P_{abs}}{P_{abs}} \frac{k_a(S')}{\hat{k}_a(S')} f_{eq}(S') + \frac{P_{nul}}{P_{nul}} \frac{k_n(S')}{\hat{k}_a(S')} \tilde{F}(S') \right) + \right. \\
 &\quad \left. \mathbb{H}(S' \geq s_m) \tilde{F}(s_m) \right) \\
 &= \mathbb{E} \left(\mathbb{H}(S' < s_m) \left(\mathbb{H}(Col = abs) \frac{1}{P_{abs}} \frac{k_a(S')}{\hat{k}_a(S')} f_{eq}(S') + \right. \right. \\
 &\quad \left. \left. \mathbb{H}(Col = nul) \frac{1}{P_{nul}} \frac{k_n(S')}{\hat{k}_a(S')} \tilde{F}(S') \right) + \mathbb{H}(S' \geq s_m) \tilde{F}(s_m) \right) \quad (2.17a)
 \end{aligned}$$

avec

$$\begin{cases} P_{abs} \text{ et } P_{nul} \text{ arbitraires telles que} \\ P_{abs} > 0, P_{nul} > 0, \text{ et } P_{abs} + P_{nul} = 1 \\ Col \text{ une VA à valeurs dans } \{abs; nul\} \\ \text{de probabilités respectives } P_{abs} \text{ et } P_{nul} \end{cases} \quad (2.17b)$$

4. Précédemment dans cette liste, on propose de mettre en place un choix de Bernoulli entre le terme d'émission et le terme de collision nulle, afin de créer un mécanisme de roulette russe.

On peut même ré-exprimer cette somme de termes, afin de faire apparaître de nouveaux termes entre lesquels choisir ! *A priori*, comme on doit alors exhiber à nouveau une somme, on va procéder par combinaison linéaire.

L'intérêt d'une telle astuce numérique est d'obtenir une estimation de $f(s)$ avec une variance plus faible que ce qu'autorise les démarches précédentes, en gardant l'utilité de la roulette russe : la possibilité d'arrêter le suivi de photon. On va donc essayer d'obtenir une somme de 2 termes, dont un seulement est récursif. Le terme non-récursif ne pouvant pas contenir une contribution de collision nulle, la formulation générale correspondante est la suivante :

$$\begin{aligned}
 f(s) &= E \left(H(S' < s_m) \left(\theta_{abs,arr} \frac{k_a(S')}{\hat{k}_a(S')} f_{eq}(S') + \theta_{abs,rec} \frac{k_a(S')}{\hat{k}_a(S')} f_{eq}(S') + \right. \right. \\
 &\quad \left. \left. \frac{k_n(S')}{\hat{k}_a(S')} \tilde{F}(S') \right) + H(S' \geq s_m) \tilde{F}(s_m) \right) \\
 &= E \left(H(S' < s_m) \left(\frac{H(Col = arr)}{P_{arr}} \theta_{abs,arr} \frac{k_a(S')}{\hat{k}_a(S')} f_{eq}(S') + \right. \right. \\
 &\quad \left. \left. \frac{H(Col = rec)}{P_{rec}} \left(\theta_{abs,rec} \frac{k_a(S')}{\hat{k}_a(S')} f_{eq}(S') + \frac{k_n(S')}{\hat{k}_a(S')} \tilde{F}(S') \right) \right) + \right. \\
 &\quad \left. H(S' \geq s_m) \tilde{F}(s_m) \right) \tag{2.18a}
 \end{aligned}$$

avec

$$\left\{ \begin{array}{l} \theta_{abs,arr} \text{ et } \theta_{abs,rec} \text{ arbitraires tels que} \\ \theta_{abs,arr} + \theta_{abs,rec} = 1 \\ P_{arr} \text{ et } P_{rec} \text{ arbitraires telles que} \\ P_{arr} > 0, P_{rec} > 0, \text{ et } P_{arr} + P_{rec} = 1 \\ Col \text{ une VA à valeurs dans } \{arr; rec\} \\ \text{de probabilités respectives } P_{arr} \text{ et } P_{rec} \end{array} \right. \tag{2.18b}$$

●●●● Ces méthodes algorithmiques peuvent être combinées entre elles. En effet, les Algorithmes à Collisions Nulles sont par nature récurrents, et quand on veut estimer $f(s)$, quelque soit la technique sus-explicitée utilisée, il existe une probabilité non-nulle (si $\hat{k}_a \neq k_a$) de devoir estimer f en une abscisse aléatoire $S' \geq s$ pour aboutir. Par contre, il n'est pas obligatoire de réutiliser exactement la même technique algorithmique pour estimer $f(S')$ et $f(s)$. Au cours du suivi d'un photon, on pourra donc utiliser des approches différentes pour estimer la somme des termes d'émission et de collision nulle.

Dans la littérature, on constate que ce sont les 1^{re} et 2^e approche qui sont le plus utilisées. Elles étaient toutes deux déjà connues en 1966 [112]. En neutronique c'est généralement la 2^e approche qui est utilisée, car les collisions sont généralement des diffusions, et donc l'emploi d'un partitionnement de l'énergie provoquerait une multiplication généralement inutile des chemins suivis. Dans les autres domaines c'est plus variable.

En fait, selon la situation, il vaut mieux privilégier une approche ou une autre. Les sections 2.3 et 2.4 fournissent des éléments de réponses au lecteur que ces choix algorithmiques intéressent. Le cas de milieux absorbants et diffusants, et non plus uniquement absorbants comme dans cette section, y est évoqué.

2.2.4 ACNs et calcul de sensibilités

●●●● Comme affirmé au chapitre 1.5.5, le calcul de sensibilités par la méthode de Monte-Carlo nécessite un travail spécifique sur la formulation intégrale, la méthode des différences finies n'étant pas utilisable.

Comparativement à l'exemple donné au chapitre 1.5.5, les Algorithmes à Collision Nulles n'ajoutent pas de difficulté. Si on se replace dans le cas d'étude simplifié de la section 2.1.2, et qu'on considère un paramètre π , la sensibilité $\partial_\pi f(s)$ peut venir de :

- la fonction de distribution d'équilibre f_{eq} ,
- le coefficient d'absorption k_a ,
- la luminance au bout de la ligne de visée $f(s_m)$,
- la distance du mur au bout de la ligne de visée s_m .

La sensibilité vis-à-vis d'un paramètre géométrique comme s_m est un cas particulier. Le lecteur intéressé pourra se reporter à la monographie [101] dont c'est le sujet, ou encore à [29] et [44]. Les 3 autres contributions possibles à $\partial_\pi f(s)$ constituent en revanche des problèmes bien maîtrisés.

●●●● Si on utilise un algorithme à collisions nulles, on passe de l'expression de f :

$$f(s) = \int_s^{+\infty} ds' k_a(s') \exp\left(-\int_s^{s'} ds'' k_a(s'')\right) \times \left(H(s' < s_m) f_{eq}(s') + H(s' \geq s_m) f(s_m)\right) \quad (2.4)^\dagger$$

à

$$f(s) = \int_s^{+\infty} ds' \exp\left(-\int_s^{s'} ds'' \hat{k}_a(s'')\right) \times \left(H(s' < s_m) \left(k_a(s') f_{eq}(s') + k_n(s') f(s')\right) + H(s' \geq s_m) \hat{k}_a(s') f(s_m)\right) \quad (2.9)^\dagger$$

La démarche pour obtenir la formulation intégrale d'une sensibilité $\partial_\pi f$ reste globalement la même : il faut dériver l'équation (2.9) (au lieu de (2.4)) selon π .

Une petite subtilité apparaît cependant : k_n est un paramètre libre, sans influence sur f , mais selon ce qu'il devient lors de la dérivation de (2.9) le résultat s'exprime différemment. On peut distinguer 2 attitudes simples : soit on laisse \hat{k}_a constant soit on laisse k_n constant.

Si on laisse \hat{k}_a constant :

$$\begin{aligned} \partial_{\pi \perp \hat{k}_a} f(s) = & \int_s^{+\infty} ds' \exp\left(-\int_s^{s'} ds'' \hat{k}_a(s'')\right) \times \\ & \left(H(s' < s_m) \left(\partial_\pi k_a(s') (f_{eq}(s') - f(s')) + k_a(s') \partial_\pi f_{eq}(s') + k_n(s') \partial_\pi f(s') \right) + \right. \\ & \left. H(s' \geq s_m) \hat{k}_a(s') \partial_\pi f(s_m) \right) \quad (2.19) \end{aligned}$$

Si on laisse k_n constant :

$$\begin{aligned} \partial_{\pi \perp k_n} f(s) = & \int_s^{+\infty} ds' \exp\left(-\int_s^{s'} ds'' \hat{k}_a(s'')\right) \times \\ & \left(\left(H(s' < s_m) \left(\partial_{\pi} k_a(s') f_{eq}(s') + k_a(s') \partial_{\pi} f_{eq}(s') + k_n(s') \partial_{\pi} f(s') \right) + \right. \right. \\ & H(s' \geq s_m) \left(\hat{k}_a(s') \partial_{\pi} f(s_m) + \partial_{\pi} k_a(s') f(s_m) \right) - \int_s^{s'} ds'' \partial_{\pi} k_a(s'') \times \\ & \left. \left. \left(H(s' < s_m) \left(k_a(s') f_{eq}(s') + k_n(s') f(s') \right) + H(s' \geq s_m) \hat{k}_a(s') f(s_m) \right) \right) \right) \quad (2.20) \end{aligned}$$

Bien entendu, $\partial_{\pi \perp \hat{k}_a} f(s) = \partial_{\pi \perp k_n} f(s)$; néanmoins, l'expression (2.19) semble plus simple pour le calcul, dans la majorité des situations.

2.3 (*) Mise en œuvre pratique des alternatives algorithmiques

Nous avons expliqué dans la section 2.2.3 que plusieurs choix algorithmiques sont possibles, avec partitionnement de l'énergie ou roulette russe. Mais que décider en pratique ? Comment faire un choix qui soit raisonnablement efficace ? Nous donnons ici quelques éléments de clarification.

2.3.1 Dans le cas d'étude du chapitre (absorption/émission seule)

Le choix d'une méthode de calcul de la somme des contributions de l'émission et des collisions nulles dans l'expression (2.16) devrait être guidé par les avantages et les défauts de ces méthodes. Ils sont de plusieurs ordres :

- applicabilité de la méthode
- récursivité générée
- variance du poids résultant

La roulette russe analogue (la 2^e méthode de la liste en section 2.2.3) n'est utilisable que si \hat{k}_a majore effectivement k_a . Si ce n'est pas le cas, $k_a(S')/\hat{k}_a(S')$ et $k_n(S')/\hat{k}_a(S')$ ne sont plus des probabilités. En pratique, et en particulier dans les problèmes de transfert radiatif atmosphérique, c'est une limitation forte. Quand on utilise un Algorithme à Collisions Nulles, et que $\hat{k}_a \geq k_a$, la récursivité moyenne

est proportionnelle à \hat{k}_a ; on a donc intérêt à prendre un \hat{k}_a le plus faible possible pour limiter le temps de calcul. Par ailleurs, dans les gaz le coefficient d'absorption est une fonction non monotone de la pression et de la température, avec une dépendance spectrale forte. Pour chaque fréquence donnée ν , il est donc difficile de connaître avec précision la borne supérieure du coefficient d'absorption sur le volume d'intérêt — surtout en science atmosphérique, où la pression varie très largement.

Ceci a amené l'équipe où la présente thèse a été encadrée, à proposer l'utilisation d'un choix émission/collision nulle modifié (la 3^e possibilité évoquée en section 2.2.3) [50] ; ce choix était accompagné d'une proposition pour P_{abs} et P_{nul} (valable ssi $k_a \geq 0$) :

$$\begin{aligned} P_{abs} &= \frac{k_a(S')}{k_a(S') + |k_n(S')|} \\ P_{nul} &= \frac{|k_n(S')|}{k_a(S') + |k_n(S')|} \end{aligned} \quad (2.21)$$

Tant que $\hat{k}_a \geq k_a$, ceci reste strictement équivalent à une roulette russe analogue. Par contre, si $\hat{k}_a < k_a$ par endroits, cette roulette russe avec probabilités modifiées reste encore utilisable, bien que la variance du poids augmente. En particulier, si la situation physique étudiée est en équilibre thermodynamique global², le poids estimant $f(s)$ n'est plus fixé à $f_{eq}(T_{globale})$, comme cela est le cas avec partitionnement de l'énergie ou avec roulette russe analogue.

La 4^e proposition de la section 2.2.3 permet de remédier partiellement à ce défaut. On peut en effet assurer que, avec cette roulette russe à nouveau modifiée :

- le mécanisme est identique à une roulette russe analogue si $\hat{k}_a \geq k_a$,
- le poids estimant $f(s)$ est de variance nulle en cas d'équilibre thermodynamique global,

si on utilise par exemple le jeu de paramètres suivants :

$$P_{arr} = \frac{k_a(S')}{k_a(S') + |k_n(S')|} \quad (2.22a)$$

$$P_{rec} = \frac{|k_n(S')|}{k_a(S') + |k_n(S')|}$$

$$\theta_{abs,arr} = \frac{k_a(S') + k_n(S')}{k_a(S') + |k_n(S')|} \quad (2.22b)$$

$$\theta_{abs,rec} = \frac{|k_n(S')| - k_n(S')}{|k_n(S')| + k_a(S')}$$

Étude des techniques sus-citées : On se place précisément dans le cas d'étude de ce chapitre, c'est-à-dire qu'on estime $f(s_{init})$. En réutilisant les notations introduites au chapitre 1.5.4, on peut donner l'expression totale du poids de Monte-Carlo $W = \tilde{F}(s_{init})$ qui va sortir d'un Algorithme à Collisions Nulles, selon la technique de sommation utilisée, afin d'anticiper le comportement de ce poids. Notons :

- $\hat{S}_0 = s_{init}$ la position de départ du suivi de photon,
- \hat{S}_i avec $i \in \mathbb{N}$ la i^e position de collision du photon qu'on échantillonne. $\forall i \in$

\mathbb{N} , \hat{S}_{i+1} est de fonction de répartition $F_{\hat{S}_{i+1}} : \hat{s} \mapsto 1 - \exp(-\int_{\hat{S}_i}^{\hat{s}} ds' \hat{k}_a(s'))$.

- I_{arr} l'indice de la position \hat{S} où le photon sera arrêté, si on utilise une approche en roulette russe. I_{arr} obéit à la propriété $\forall i \in \mathbb{N}^*, \Pr(I_{arr} = i \mid I_{arr} \geq i) = P_{arr,i}$ où $P_{arr,i} = P_{arr}(\hat{S}_i)$ est la probabilité d'arrêt du suivi en \hat{S}_i , selon la formule qu'on s'est donnée.
- I_m l'indice de la 1^{re} position \hat{S} échantillonnée au-delà du mur au bout de la ligne de visée.

Si on utilise exclusivement le partitionnement de l'énergie, en reprenant les notations du chapitre 1.5.4 on a, $\forall i \in \mathbb{N}^*$:

$$\Pi_i = \begin{cases} \prod_{j=1}^i \frac{k_n(\hat{S}_j)}{\hat{k}_a(\hat{S}_j)} & \text{si } i < I_m \\ 0 & \text{si } i \geq I_m \end{cases} \quad (2.23a)$$

$$Cp_i = \begin{cases} \left(\prod_{j=1}^{i-1} \frac{k_n(\hat{S}_j)}{\hat{k}_a(\hat{S}_j)} \right) \frac{k_a(\hat{S}_i)}{\hat{k}_a(\hat{S}_i)} & \text{si } i < I_m \\ \prod_{j=1}^{i-1} \frac{k_n(\hat{S}_j)}{\hat{k}_a(\hat{S}_j)} & \text{si } i = I_m \\ 0 & \text{si } i > I_m \end{cases} \quad (2.23b)$$

$$Wp_i = \begin{cases} f_{eq}(\hat{S}_i) & \text{si } i < I_m \\ \tilde{F}(s_m) & \text{si } i \geq I_m \end{cases} \quad (2.23c)$$

donc

$$W = \sum_{i=1}^{I_m-1} \left(\prod_{j=1}^{i-1} \frac{k_n(\hat{S}_j)}{\hat{k}_a(\hat{S}_j)} \right) \frac{k_a(\hat{S}_i)}{\hat{k}_a(\hat{S}_i)} f_{eq}(\hat{S}_i) + \prod_{i=1}^{I_m-1} \frac{k_n(\hat{S}_i)}{\hat{k}_a(\hat{S}_i)} \tilde{F}(s_m) \quad (2.23d)$$

2 remarques sont à faire à ce stade :

- W est une moyenne pondérée des $f_{eq}(\hat{S}_i)$ et de $\tilde{F}(s_m)$.
De plus, les coefficients de pondération Cp_i sont tous positifs si et seulement si $\hat{k}_a \geq k_a$ à toutes les positions de

2. Chaque point est en équilibre thermodynamique local, et le système entier est de température égale.

collisions choisies. Cela s'écrit :

$$\begin{aligned} (\forall i \in \llbracket 1; I_m \rrbracket, Cp_i \geq 0) &\iff \\ (\forall i \in \llbracket 1; I_m - 1 \rrbracket, \hat{k}_a(\hat{S}_i) \geq k_a(\hat{S}_i)) & \end{aligned} \quad (2.24)$$

Dans ce cas, on obtient nécessairement $W \in [\min \mathcal{F}; \max \mathcal{F}]$, où $\mathcal{F} = \{f_{eq}(\hat{S}_i), i \in \llbracket 1; I_m - 1 \rrbracket\} \cup \{\tilde{F}(s_m)\}$. De façon plus générale, si $\hat{k}_a \geq k_a$ tout le long de la ligne de visée, alors le poids W est nécessairement dans l'intervalle $[\inf \mathcal{F}_t; \sup \mathcal{F}_t]$ où $\mathcal{F}_t = \{f_{eq}(s), s \in [s_{init}; s_m]\} \cup \mathcal{U}_{\tilde{F}(s_m)}$; c'est une garantie de bonne convergence pour la moyenne de N poids w_n .

En revanche, si par endroit $\hat{k}_a < k_a$, alors on obtiendra des valeurs négatives pour certains coefficients de pondération. Cela mène à une augmentation de la variance de la distribution de W .

- Au fil des collisions, si $k_a < 2\hat{k}_a$, le total de pondération des collisions restantes Π_i tend vers 0. Plus précisément, arrivé à une certaine position s , le to-

tal de pondération restante Π a pour espérance la transmittivité de s vers s_{init} ; on peut s'en convaincre en prenant $Wp_i = H(\hat{S}_i \geq s)$ dans l'expression (2.23c).

Cela implique que dans un milieu optiquement épais, l'algorithme va passer du temps à sommer des contributions à la luminance en s_{init} , contributions dont la plupart seront très faibles (à moins que les zones contribuant réellement à éclairer le point initial soient éloignées).

Diminuer \hat{k}_a sous k_a n'est pas une solution. Le long du chemin, Π garde la même espérance quel que soit \hat{k}_a , par contre si \hat{k}_a est faible devant k_a cela fait augmenter la variance de W .

Vis-à-vis de l'utilisation exclusive d'une roulette russe, on peut effectuer la même étude que précédemment. On se placera dans le cas le plus général d'une roulette russe à termes modifiés (la 4^e proposition de la section 2.2.3).

On pose $I'_{arr} = \min\{I_{arr}; I_m\}$, alors $\forall i \in \mathbb{N}^*$:

$$\Pi_i = \begin{cases} \left(\prod_{j=1}^i \frac{1}{P_{rec,j}} \frac{k_n(\hat{S}_j)}{\hat{k}_a(\hat{S}_j)} \right) & \text{si } i < I'_{arr} \\ 0 & \text{si } i \geq I'_{arr} \end{cases} \quad (2.25a)$$

$$Cp_i = \begin{cases} \left(\prod_{j=1}^{i-1} \frac{1}{P_{rec,j}} \frac{k_n(\hat{S}_j)}{\hat{k}_a(\hat{S}_j)} \right) \frac{\theta_{abs,rec} \ k_a(\hat{S}_i)}{P_{rec,i} \ \hat{k}_a(\hat{S}_i)} & \text{si } i < I'_{arr} \\ \left(\prod_{j=1}^{i-1} \frac{1}{P_{rec,j}} \frac{k_n(\hat{S}_j)}{\hat{k}_a(\hat{S}_j)} \right) \frac{\theta_{abs,arr} \ k_a(\hat{S}_i)}{P_{arr,i} \ \hat{k}_a(\hat{S}_i)} & \text{si } i = I'_{arr} < I_m \\ \prod_{j=1}^{i-1} \frac{1}{P_{rec,j}} \frac{k_n(\hat{S}_j)}{\hat{k}_a(\hat{S}_j)} & \text{si } i = I'_{arr} = I_m \\ 0 & \text{si } i > I'_{arr} \end{cases} \quad (2.25b)$$

$$Wp_i = \begin{cases} f_{eq}(\hat{S}_i) & \text{si } i < I_m \\ \tilde{F}(s_m) & \text{si } i \geq I_m \end{cases} \quad (2.25c)$$

donc

$$\begin{aligned} W = \sum_{i=1}^{I'_{arr}-1} \left(\prod_{j=1}^{i-1} \frac{1}{P_{rec,j}} \frac{k_n(\hat{S}_j)}{\hat{k}_a(\hat{S}_j)} \right) \frac{\theta_{abs,rec} \ k_a(\hat{S}_i)}{P_{rec,i} \ \hat{k}_a(\hat{S}_i)} f_{eq}(\hat{S}_i) + \left(\prod_{i=1}^{I'_{arr}-1} \frac{1}{P_{rec,i} \ \hat{k}_a(\hat{S}_i)} \right) \times \\ \left(H(I_{arr} < I_m) \frac{\theta_{abs,arr} \ k_a(\hat{S}_{I_{arr}})}{P_{arr,I_{arr}} \ \hat{k}_a(\hat{S}_{I_{arr}})} f_{eq}(\hat{S}_{I_{arr}}) + H(I_{arr} \geq I_m) \tilde{F}(s_m) \right) \end{aligned} \quad (2.25d)$$

2.3. Mise en œuvre pratique des alternatives algorithmiques

Le comportement de la variable aléatoire W et le coût de l'algorithme qui la génère vont dépendre de l'expression détaillée des $P_{arr,i}$, $P_{rec,i}$, $\theta_{abs,rec\ i}$, et $\theta_{abs,arr\ i}$. Par exemple, avec la proposition faite en (2.22), l'expression de W se précise comme :

$$W = \sum_{i=1}^{I'_{arr}-1} \left(\prod_{j=1}^{i-1} \text{sgn}(k_n(\hat{S}_j)) \frac{k_a(\hat{S}_j) + |k_n(\hat{S}_j)|}{k_a(\hat{S}_j) + k_n(\hat{S}_j)} \right) \left(1 - \text{sgn}(k_n(\hat{S}_i)) \right) \frac{k_a(\hat{S}_i)}{\hat{k}_a(\hat{S}_i)} f_{eq}(\hat{S}_i) + \left(\prod_{i=1}^{I'_{arr}-1} \text{sgn}(k_n(\hat{S}_i)) \frac{k_a(\hat{S}_i) + |k_n(\hat{S}_i)|}{k_a(\hat{S}_i) + k_n(\hat{S}_i)} \right) \left(H(I_{arr} < I_m) f_{eq}(\hat{S}_{I_{arr}}) + H(I_{arr} \geq I_m) \tilde{F}(s_m) \right) \quad (2.26)$$

Si $\hat{k}_a \geq k_a$ (au moins aux positions de collisions \hat{S}_i) cette expression se réduit à :

$$W = H(I_{arr} < I_m) f_{eq}(\hat{S}_{I_{arr}}) + H(I_{arr} \geq I_m) \tilde{F}(s_m) \quad (2.27)$$

Plutôt que de faire la moyenne entre les valeurs de f_{eq} à différentes positions de collisions échantillonnées, on cherche à affecter à W la valeur de f_{eq} ou $\tilde{F}(s_m)$ en une seule position (la position d'arrêt du suivi de photon). On peut alors s'attendre à ce que la variance de W soit plus élevée que ce qu'on aurait obtenu avec un partitionnement de l'énergie.

Plus en détails, en suivant la proposition (2.22) :

- Si \hat{k}_a majore k_a en tout point, la probabilité que le suivi de photon arrive jusqu'à une abscisse curviligne s égale la transmittivité de s vers s_{init} . Si le milieu est optiquement épais, l'algorithme se terminera donc plus vite que si on avait utilisé un partitionnement de l'énergie.
- Si $\hat{k}_a \ll k_a$, la probabilité que le suivi de photon atteigne une certaine abscisse curviligne suit une loi exponentielle de coefficient d'extinction $\hat{k}_a/2$.
- Si au cours d'un suivi de photon on obtient $\hat{k}_a < k_a$ à certaines positions de collisions, W devient une moyenne pondérée entre la luminance d'équilibre à ces positions et la luminance d'équilibre à la position d'arrêt. Les coefficients de pondération alternent en signe, et le total de pondération des collisions restantes augmente en valeur absolue à chaque collision. Ce comportement est susceptible faire augmenter la variance de W .

Une autre proposition pour l'expression de P_{arr} , P_{rec} , $\theta_{abs,rec}$, et $\theta_{abs,arr}$ est donnée à

la section 2.4.

Roulette russe déclenchable : Une approche pour combiner les avantages du partitionnement de l'énergie et de la roulette russe est de commencer le suivi de photon en partitionnant l'énergie lors des collisions, puis de passer en roulette russe. Le critère de changement est généralement un seuil sur le total de pondération des collisions restantes.

●●●● La figure 2.2 donne un exemple de l'évolution de π et d'un échantillon de cp pour l'estimation de $f(s_{init})$. Deux façons de calculer la somme des termes d'émission et de collision nulle sont comparées : une roulette russe analogue (applicable dans l'exemple de la figure), et une roulette russe analogue déclenchable précédée d'un partitionnement de l'énergie.

Une remarque peut être faite au visionnage de cette figure : il est moins pénalisant d'avoir un \hat{k}_a trop élevé si on utilise un partitionnement de l'énergie plutôt qu'une roulette russe. Dans les deux cas, plus \hat{k}_a est grand plus chaque suivi de photon comportera de collisions et nécessitera donc de temps de calcul. Par contre, quand on utilise un partitionnement de l'énergie, $f(s_{init})$ est estimée comme une somme de contributions venant de plusieurs points du segment $[s_{init}; s_m]$; et plus \hat{k}_a est grand, plus l'ensemble des points contributeurs est raffiné, et donc plus l'estimation de $f(s_{init})$ est précise. En roulette russe on ne garde que la contribution du dernier point de

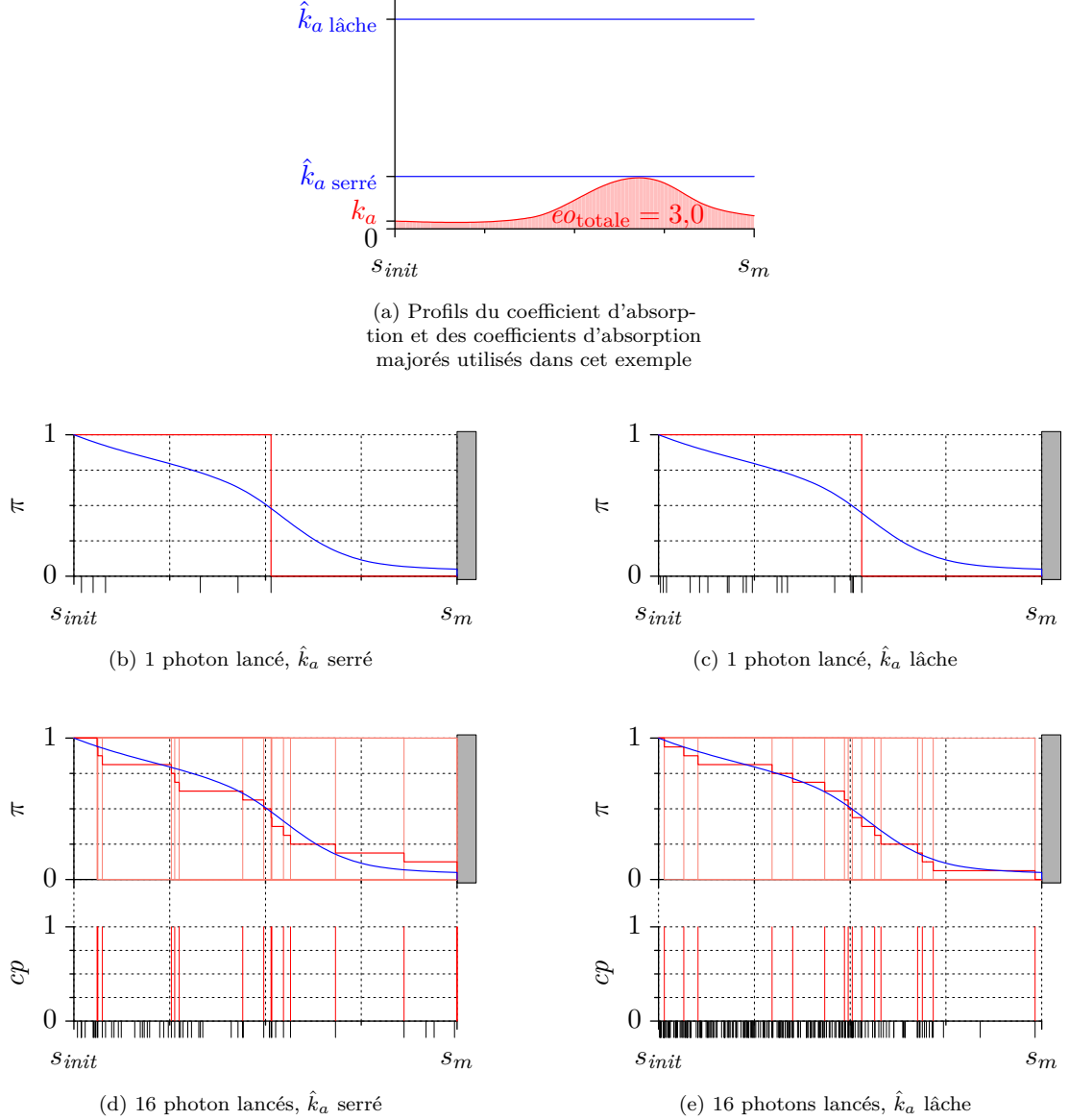


FIGURE 2.2 – Évolution, lors du calcul de f dans une situation décrite à la section 2.1.2 page 76, du total de pondération des collisions restantes π en fonction de l'abscisse curviligne s , π étant calculé avec 1 ou 16 photons lancés. Dans les graphes de (b) à (e), c'est une roulette russe analogue qui a été utilisée.

La courbe bleue donne la transmittivité exacte de s vers s_{init} , la courbe rouge donne π moyenné entre les photons lancés, et s'il y en a plusieurs les courbes roses fines donnent π pour chaque photon. Les marques noires fines indiquent les points de collision, réelle ou nulle, où k_a a dû être calculé. Les marques rouges fines indiquent ceux où $f_{eq}(s)$ entre dans le calcul de $f(s_{init})$.

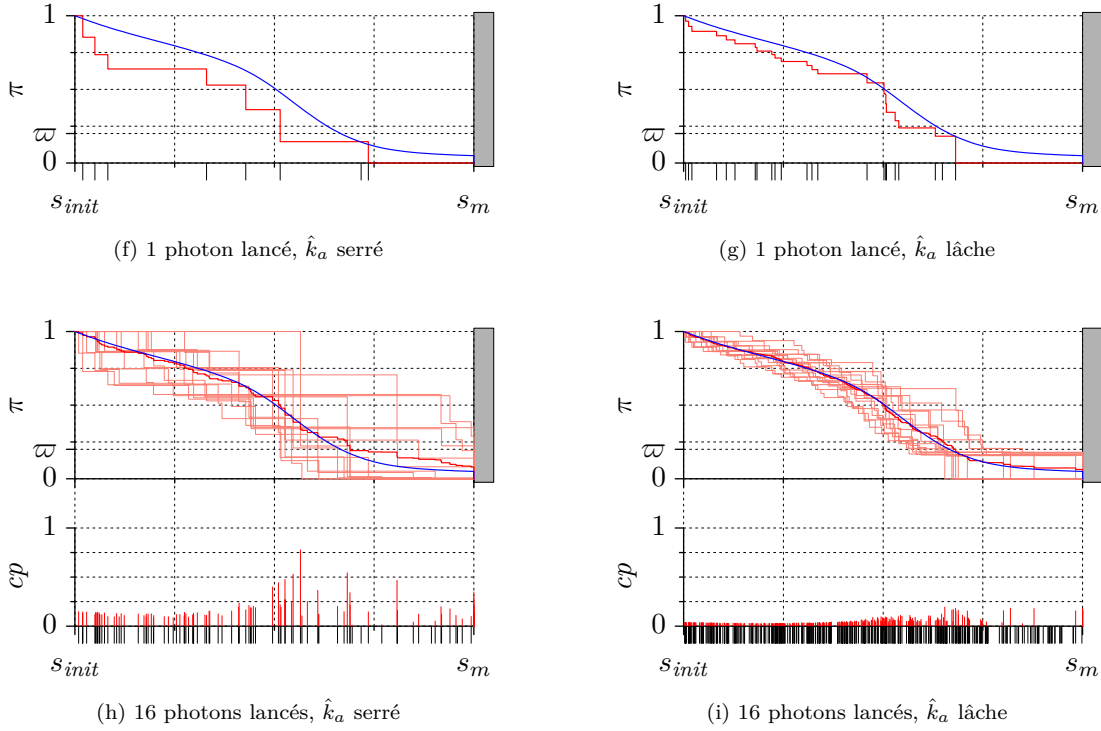


FIGURE 2.2 – Dans les graphes de (f) à (i), c'est une roulette russe déclenchable qui a été utilisée. Avant que cette roulette ne soit déclenchée, on utilise un partitionnement de l'énergie; la roulette se déclenche quand π passe en dessous d'un seuil ϖ (fixé à 0,2).

La hauteur des marques rouges est proportionnelle à la pondération associée à $f_{eq}(s)$ dans le calcul de $f(s_{init})$. Certaines marques se superposent.

Figure inspirée de la publication [89].

collision choisi, donc l'estimation de $f(s_{init})$ ne s'améliore pas quand \hat{k}_a augmente.

Il faut cependant garder à l'esprit que dans un calcul Monte-Carlo concret de transfert radiatif, l'estimation de $f(s_{init})$ entre généralement dans un processus d'estimation plus large, comme par exemple celui d'un terme source radiatif; alors $f(s_{init})$ devient dépendante de paramètres choisis antérieurement, comme par exemple une direction d'arrivée en un point. Dans ce cas il n'est pas très utile d'avoir une estimation ponctuelle très précise de $f(s_{init})$ pour un choix particulier de paramètres antérieurs, car l'estimation finale comprendra une part de variance issue de l'échantillonnage de ces paramètres. L'avantage sus-explicité du partitionnement de l'énergie par rapport à la roulette russe est alors plus faible.

2.3.2 Avec diffusion

•••• Les Algorithmes à Collisions Nulles restent parfaitement fonctionnels en présence de diffusion. Simplement, il faut tenir compte des termes de diffusion dans les équations intégrales de transport.

En présence de diffusion pure, la transformation à appliquer aux écritures précédentes de ce chapitre pour obtenir les expressions intégrales appropriées est évidente : il suffit de remplacer le coefficient d'absorption k_a par le coefficient de diffusion k_d , et le terme d'émission thermique $k_a f_{eq}$ par le terme source de

diffusion. Dans un milieu isotrope, ce terme s'écrit $k_d(\vec{r}) \int_{4\pi} d\vec{u}' \Phi(\vec{r}; \vec{u} \cdot \vec{u}') f(\vec{r}; \vec{u}')$.

Si on a en même temps diffusion, absorption, et émission, plusieurs approches sont *a priori* possibles. La 1^{re}, la plus efficace, est de tenir compte des 3 contributions (d'absorption/émission, de diffusion, et de collision nulle) dans le terme d'extinction et dans le terme source de l'équation de transport. Du point de vue algorithmique, l'émission et la diffusion entrante seront alors pris en compte simultanément. La 2^e approche consiste à d'abord générer des chemins optiques de diffusion en tenant compte uniquement du caractère diffusant du milieu, puis d'appliquer un algorithme adapté aux situations d'absorption/émission seule le long de ces chemins pour estimer f en leur origine; un ACN peut faciliter l'une de ces étapes, ou même les 2. Dans ce cas, la variable s utilisée dans les précédentes formulations de f en absorption/émission devient l'abscisse curviligne le long de chaque chemin optique de diffusion. On sait qu'une approche consistant à traiter ainsi séquentiellement la diffusion puis l'absorption est tout-à-fait valable; il existe d'ailleurs des démonstrations basées sur les ACNs de ce résultat [81]. Elle est cependant peu efficace, car on commence par générer des chemins optiques de diffusion destinés à être utilisés seulement partiellement ensuite par la prise en compte de l'absorption (à moins que le milieu soit optiquement peu épais).

•••• On s'intéresse ici uniquement à la 1^{re} approche. En présence de diffusion, d'absorption et d'émission thermique, et dans un milieu isotrope, l'équation de transport s'écrit sous forme statistique :

$$f(\vec{r}; \vec{u}) = E \left(\frac{\hat{k}_t(\vec{r} - S\vec{u}) \exp(-\int_s^S ds' \hat{k}_t(\vec{r} - s'\vec{u}))}{p_S(S)} \left(H(S < s_m) \right. \right. \\ \left. \left. \underbrace{\left(\frac{k_a(\vec{r} - S\vec{u})}{\hat{k}_t(\vec{r} - S\vec{u})} f_{eq}(\vec{r} - S\vec{u}) \right)}_{\text{terme d'émission}} + \underbrace{\left(\frac{k_d(\vec{r} - S\vec{u})}{\hat{k}_t(\vec{r} - S\vec{u})} \frac{\Phi(\vec{r} - S\vec{u}; \vec{U}' \cdot \vec{u})}{p_{\vec{U}'|S}(\vec{U}')} \tilde{F}(\vec{r} - S\vec{u}; \vec{U}') \right)}_{\text{terme de diffusion}} + \right. \\ \left. \underbrace{\left(\frac{k_n(\vec{r} - S\vec{u})}{\hat{k}_t(\vec{r} - S\vec{u})} \tilde{F}(\vec{r} - S\vec{u}; \vec{u}) \right)}_{\text{terme de collision nulle}} \left. + H(S \geq s_m) \tilde{F}(\vec{r} - s_m \vec{u}; \vec{u}) \right) \right) \quad (2.28)$$

où $\hat{k}_t = k_t + k_n = k_a + k_d + k_n$, et S et \vec{U}' sont deux variables aléatoires à valeurs respectivement dans \mathbb{R}^+ et $\mathcal{S}(\mathbb{R}^3; \vec{0}; 1)$.

Afin de faire diminuer la variance d'un estimateur de $f(\vec{r}; \vec{u})$ basé sur l'expression (2.28), selon ce que l'on a déduit au chapitre 1.5.3 il peut être judicieux de choisir :

- S comme une longueur d'extinction obéissant à une loi exponentielle de coefficient \hat{k}_t , c'est-à-dire que $F_S(s) = 1 - \exp(-\int_0^s ds' \hat{k}_t(s'))$,
- \vec{U}' comme une direction de diffusion obéissant à la fonction de phase de diffusion, c'est-à-dire que $p_{\vec{U}'|S}(\vec{u}') = \Phi(\vec{r} - S\vec{u}; \vec{u} \cdot \vec{u}')$,

auquel cas l'expression (2.28) se simplifie en :

$$f(\vec{r}; \vec{u}) = \mathbb{E} \left(\mathbb{H}(S < s_m) \left(\frac{k_a(\vec{r} - S\vec{u})}{\hat{k}_t(\vec{r} - S\vec{u})} f_{eq}(\vec{r} - S\vec{u}) + \frac{k_d(\vec{r} - S\vec{u})}{\hat{k}_t(\vec{r} - S\vec{u})} \tilde{F}(\vec{r} - S\vec{u}; \vec{U}') + \right. \right. \\ \left. \left. \frac{k_n(\vec{r} - S\vec{u})}{\hat{k}_t(\vec{r} - S\vec{u})} \tilde{F}(\vec{r} - S\vec{u}; \vec{u}) \right) + \mathbb{H}(S \geq s_m) \tilde{F}(\vec{r} - s_m \vec{u}; \vec{u}) \right) \quad (2.29)$$

●●●● Reste à faire un choix algorithmique pour calculer la somme des contributions de l'émission, de la diffusion entrante, et des collisions nulles. Nous ne détaillerons pas ici tout ce qui peut être mis au point et intéressant du point de vue calculatoire : les possibilités sont nombreuses, bien plus que lorsque le milieu n'est pas diffusant. Néanmoins, la section suivante présente un cas académique de transfert radiatif avec un milieu chaud absorbant et diffusant ; l'algorithme de Monte-Carlo à collisions nulles qui est utilisé pour les calculs est détaillé, offrant un exemple d'implémentation pratique.

Quoi qu'il en soit, il est important de noter qu'on utilise rarement un partitionnement de l'énergie pour calculer la somme du terme de diffusion entrante et du terme de collision nulle. En effet, chacun de ces termes est récursif et demande de recommencer l'estimation de

f en un nouveau point de l'espace des phases. Si on utilise un partitionnement de l'énergie pour calculer leur somme, le suivi de photon ne forme plus un chemin mais un arbre se divisant en deux branches à chaque nouvelle collision. Lorsqu'on traite un problème stationnaire (ce qu'on fait en transfert radiatif la plupart du temps), seules les conditions aux frontières ou l'émission sont susceptibles de terminer une branche du suivi de photon (car ce sont les sources de lumière). Ainsi, si l'épaisseur optique en diffusion du milieu est assez grande, il se produit un phénomène de criticité : l'espérance du nombre de branches d'un arbre de suivi de photon devient strictement infinie, et l'algorithme de calcul est alors susceptible de ne pas pouvoir se terminer. Une possibilité évoquée comme perspective, page 251, se situe dans le même cas.

2.4 (*) Exemple académique

Cet exemple est repris de la publication [43]. Il est académique, mais comprend une géométrie (simple) 3D avec des champs de propriétés anisothermes. Il permet d'illustrer le comportement d'un algorithme à collisions nulles, et en particulier sa sensibilité au choix du coefficient d'extinction majoré \hat{k}_t .

2.4.1 Introduction

Nous présentons dans cette section un cas académique de calcul du transfert radiatif, qui figure une chambre de combustion contenant une flamme axisymétrique. Ce cas académique

a déjà été présenté dans plusieurs publications : [43, 50].

Dans cette étude, on considère un rayonnement monochromatique. En effet, les spectres d'absorption des gaz sont très complexes, et la

prise en compte des propriétés spectrales des milieux gazeux constitue un sujet de réflexion à part entière. L'exemple que nous allons présenter ici a pour seule vocation d'illustrer la capacité à prendre en compte les hétérogénéités des milieux participants dans un calcul par méthode de Monte-Carlo, en utilisant les Algorithmes à Collisions Nulles ; on évitera donc temporairement de traiter la complexité spectrale.

Bien sûr on peut envisager que, gérées simultanément, la difficulté de la prise en compte des hétérogénéités géométriques et la difficulté liée à la prise en compte des propriétés spectrales fassent plus que s'ajouter. En réalité, quand on utilise des méthodes de Monte-Carlo basées sur des formulations intégrales avec collisions nulles c'est l'inverse. Les Algorithmes à Collisions Nulles permettent une gestion statistique des propriétés spectrales des milieux semi-transparentes, gestion qui dans le cas des gaz diminue considérablement la complexité mathématique et numérique liée au calcul de grandeurs énergétiques (luminance, flux, ...) . C'est la contribution originale principale de la thèse de Mathieu GALTIER [49], et le point de vue statistique et numérique pour la cinétique des gaz développé dans ce manuscrit en est directement inspiré. Cela sera développé dans la section suivante 2.5.

L'illustration faite ici a 2 objectifs :

- montrer une implémentation aboutie des formes algorithmiques à Collisions Nulles, permettant d'effectuer des bilans radiatifs dans des milieux participants aux hétérogénéités un tant soit peu complexes. Il s'agit de montrer qu'elle fonctionne, et comment elle fonctionne intérieurement.
- permettre de cerner les situations physiques et numériques qui vont mettre en difficulté l'approche algorithmique à collisions nulles. Les difficultés rencontrées seront analysées à l'aune de ce que nous avons expliqué dans la section précédente.

2.4.2 Présentation du cas d'étude

Le système considéré est l'intérieur d'un cube de côté $2L$, centré sur l'origine et aligné sur les axes. Ses faces, opaques, réfléchissent partiellement la lumière, de manière diffuse et uniforme — on note ε leur absorptivité — et elles sont à 0 K. Le milieu à l'intérieur est hétérogène à la fois en température et en propriétés optiques. Les champs de k_a , k_d , et f_{eq} sont donnés par les formules :

$$k_a(\vec{r}) = Fact(\vec{r}) \times k_{a \max} \quad (2.30a)$$

$$k_d(\vec{r}) = Fact(\vec{r}) \times k_{d \max} \quad (2.30b)$$

$$f_{eq}(\vec{r}) = Fact(\vec{r}) \times f_{eq \max} \quad (2.30c)$$

$$Fact(\vec{r}) = \frac{L - r_x}{2L} \left(1 - \sqrt{\frac{r_y^2 + r_z^2}{2L^2}} \right) \quad (2.30d)$$

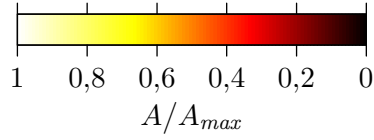
ils singent une flamme axisymétrique selon l'axe (Ox) — température et extinction maximale le long de l'axe, et décroissant linéairement selon la distance à l'axe jusqu'à 0 dans les coins. La figure 2.3 donne une représentation de ces champs. La fonction de phase de diffusion Φ est considérée isotrope.

Pour rester simple, le champ d'extinction majoré \hat{k}_t est pris uniforme. Étant donné que k_a et k_d ont leur valeur maximale au même endroit, $k_{t \max} = k_{a \max} + k_{d \max}$ est la valeur maximale du coefficient d'extinction et le ratio $\varrho = \hat{k}_t / k_{t \max}$ indique si des valeurs négatives de k_n sont obtensibles ($\varrho < 1$) ou non.

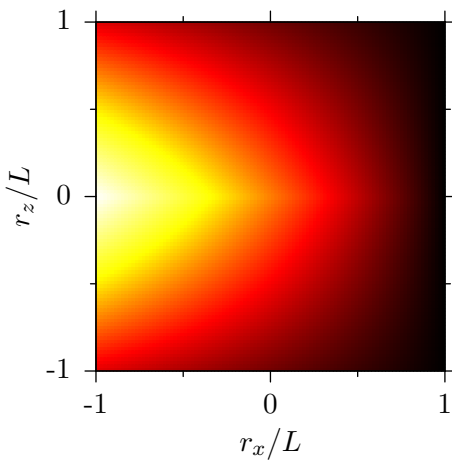
La grandeur que l'on veut calculer est le terme source radiatif $S_r(\vec{r}_0) = \int_{4\pi} d\vec{u} \, h\nu c k_a(\vec{r}_0) (f(\vec{r}_0; \vec{u}) - f_{eq}(\vec{r}_0))$ en 2 points sondes $\vec{r}_0 : (0; 0; 0)$ et $(-L; 0; 0)$. L'expression de S_r ne contient que 2 dimensions d'intégration en plus de l'expression de f , qui en comporte pour sa part une infinité (c'est une expression intégrale récursive) : on peut donc s'attendre à ce que cela ne rajoute que peu de difficultés.

2.4.3 Détails techniques sur le calcul effectué

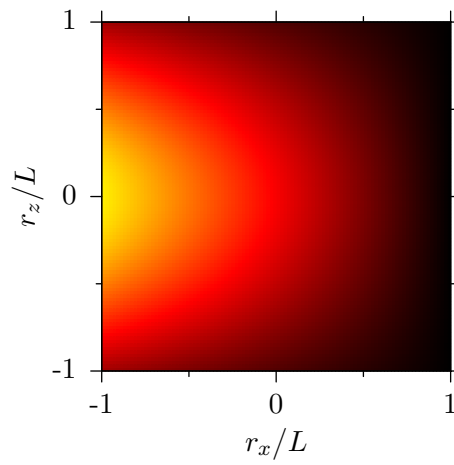
●●●● On utilisera le concept de Puissance Nette Échangée présenté dans les manuscrits [42, 48, 90]. C'est-à-dire que dans les formulations intégrales de $f(\vec{r}; \vec{u})$ précédemment dé-



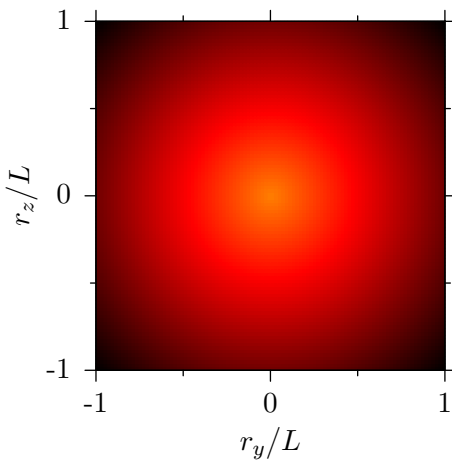
(a) Échelle de couleur utilisée dans les figures suivantes



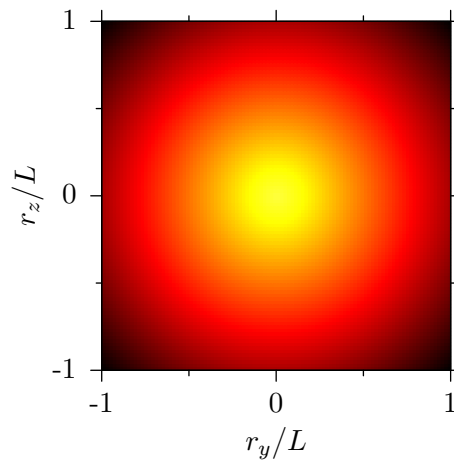
(b) Coupe selon le plan ($r_y = 0$)



(c) Coupe selon le plan ($r_y = -0.5$)



(d) Coupe selon le plan ($r_x = 0$)



(e) Coupe selon le plan ($r_x = -0.5$)

FIGURE 2.3 – Coupes de la géométrie du cas d'étude de la section 2.4, montrant les champs k_a , k_d et f_{eq} . A désigne au choix k_a , k_d ou f_{eq} (ils sont proportionnels).

taillées, on substituera « $f_{eq}(\vec{R}) - f_{eq}(\vec{r}_0)$ » à « $f_{eq}(\vec{R})$ », après quoi les dites formulations expriment $f(\vec{r}; \vec{u}) - f_{eq}(\vec{r}_0)$.

Une telle modification des écritures intégrales est validée par la possibilité d'un équilibre thermodynamique global à la température $T(\vec{r}_0)$. Dans ce cas, en tout point \vec{r} , $f(\vec{r}) = f_{eq}(\vec{r}) = f_{eq}(\vec{r}_0)$. Cette modification peut aussi s'assimiler à l'utilisation d'une variable de contrôle (voir [38]).

Dans la suite de cette section on notera :

$$\delta f(\vec{r}; \vec{u}) = f(\vec{r}; \vec{u}) - f_{eq}(\vec{r}_0) \quad (2.31a)$$

$$\delta f_{eq}(\vec{r}) = f_{eq}(\vec{r}) - f_{eq}(\vec{r}_0) \quad (2.31b)$$

●●● Par ailleurs, nous mettons en œuvre l'ensemble des astuces techniques détaillées dans la section précédente. L'algorithme de suivi de photon fonctionne globalement selon une roulette russe déclenchable. Jusqu'à ce que le total de pondération restante Π passe sous un seuil fixé ϖ , à chaque collision — dans le milieu ou sur les parois — l'énergie échangée est partitionnée entre une partie émise et une partie qui continue son chemin (avant diffusion, collision nulle, ou réflexion). Ensuite et jusqu'à la fin du suivi du photon, on choisit entre tous les événements possibles pour le photon à chaque collision — émission, diffusion, ou collision nulle si la collision a lieu dans le volume, ou émission ou réflexion si la collision se fait aux parois.

●●● Tant que l'algorithme partitionne l'énergie, la fonction de distribution est évaluée, à l'intérieur du milieu, selon l'expression suivante :

$$\begin{aligned} \delta f(\vec{r}; \vec{u}) = & \mathbb{E} \left(\mathbb{H}(S' \geq s_m) \delta \tilde{F}(\vec{r} - s_m \vec{u}; \vec{u}) + \right. \\ & \mathbb{H}(S' < s_m) \left(\frac{k_a(\vec{R}')}{\hat{k}_t(\vec{R}')} \delta f_{eq}(\vec{R}') + \right. \\ & \left. \left. \frac{\mathbb{H}(Col = diff)}{P_{diff}} \frac{k_d(\vec{R}')}{\hat{k}_t(\vec{R}')} \delta \tilde{F}(\vec{R}'; \vec{U}') + \right. \right. \end{aligned}$$

●●● À partir du 1^{er} passage de Π en deçà de ϖ une roulette russe est mise en place. Dans le milieu semi-transparent, la fonction de distribution est dès lors évaluée selon l'expression suivante :

$$\left. \left. \frac{\mathbb{H}(Col = nul)}{P_{nul}} \frac{k_n(\vec{R}')}{\hat{k}_t(\vec{R}')} \delta \tilde{F}(\vec{R}'; \vec{u}) \right) \right) \quad (2.32a)$$

avec

$$\left\{ \begin{array}{l} S' \text{ une VA à valeurs dans } \mathbb{R}^+ \\ \text{et de fonction de répartition } F_{S'}(s') = \\ 1 - \exp\left(-\int_0^{s'} ds'' \hat{k}_t(\vec{r} - s'' \vec{u})\right) \\ \vec{R}' = \vec{r} - S' \vec{u} \\ Col \text{ une VA à valeurs dans } \{diff; nul\}, \\ \text{de probabilités respectives } P_{diff} \text{ et } P_{nul} \\ \vec{U}' \text{ une VA à valeurs dans la sphère unité} \\ \text{et de densité de probabilité } 1/(4\pi) \\ \delta \tilde{F}(\vec{r}; \vec{u}) \text{ un estimateur de } \delta f(\vec{r}; \vec{u}) \end{array} \right. \quad (2.32b)$$

Lorsqu'on estime récursivement δf , le total de pondération restante est ainsi ajourné selon la formule :

$$\Pi_{i+1} = \left\{ \begin{array}{ll} \frac{k_d(\vec{R}')}{P_{diff} \hat{k}_t(\vec{R}')} \Pi_i & \text{si } Col = diff \\ \frac{k_n(\vec{R}')}{P_{nul} \hat{k}_t(\vec{R}')} \Pi_i & \text{si } Col = nul \end{array} \right. \quad (2.32c)$$

Pour choisir l'événement dont est originaire la partie non-émise de la puissance échangée (diffusion ou collision nulle), on fait le choix d'utiliser les probabilités :

$$\begin{aligned} P_{diff} &= \frac{k_d(\vec{R}')}{k_d(\vec{R}') + |k_n(\vec{R}')|} \\ P_{nul} &= \frac{|k_n(\vec{R}')|}{k_d(\vec{R}') + |k_n(\vec{R}')|} \end{aligned} \quad (2.32d)$$

Lorsque le photon suivi passe par une paroi, la fonction de distribution issue de la paroi est calculée selon :

$$\delta f(\vec{r}; \vec{u}_{out}) = \mathbb{E} \left((1 - \varepsilon) \delta \tilde{F}(\vec{r}; \vec{U}_{in}) - \varepsilon f_{eq}(\vec{r}_0) \right) \quad (2.33)$$

où \vec{U}_{in} est une direction aléatoire dans $\{\vec{u} \in \mathcal{S}(\mathbb{R}; \vec{0}; 1), \vec{u} \cdot \vec{n} \leq 0\}$, de densité de probabilité $p_{\vec{U}_{in}} : \vec{u} \mapsto -\frac{\vec{u} \cdot \vec{n}}{\pi}$, où \vec{n} est la normale à la surface intérieure du cube. La part absorbée est alors soustraite au total de pondération restante selon la formule : $\Pi_{i+1} = (1 - \varepsilon) \Pi_i$.

$$\delta f(\vec{r}; \vec{u}) = \mathbb{E} \left(\mathbb{H}(S' \geq s_m) \delta \tilde{F}(\vec{r} - s_m \vec{u}; \vec{u}) + \mathbb{H}(S' < s_m) \left(\frac{\mathbb{H}(Col = abs)}{P_{abs}} \theta_{abs,abs} \frac{k_a(\vec{R}')}{\hat{k}_t(\vec{R}')} \delta f_{eq}(\vec{R}') + \frac{\mathbb{H}(Col = diff)}{P_{diff}} \left(\theta_{abs,diff} \frac{k_a(\vec{R}')}{\hat{k}_t(\vec{R}')} \delta f_{eq}(\vec{R}') + \frac{k_d(\vec{R}')}{\hat{k}_t(\vec{R}')} \delta \tilde{F}(\vec{R}'; \vec{U}') \right) + \frac{\mathbb{H}(Col = nul)}{P_{nul}} \left(\theta_{abs,nul} \frac{k_a(\vec{R}')}{\hat{k}_t(\vec{R}')} \delta f_{eq}(\vec{R}') + \frac{k_n(\vec{R}')}{\hat{k}_t(\vec{R}')} \delta \tilde{F}(\vec{R}'; \vec{u}) \right) \right) \right) \quad (2.34a)$$

avec

$$\left\{ \begin{array}{l} S' \text{ une VA à valeurs dans } \mathbb{R}^+ \text{ et de fonction de répartition} \\ F_{S'}(s') = 1 - \exp\left(-\int_0^{s'} ds'' \hat{k}_t(\vec{r} - s'' \vec{u})\right) \\ \vec{R}' = \vec{r} - S' \vec{u} \\ Col \text{ une VA à valeurs dans } \{abs; diff; nul\}, \\ \text{de probabilités respectives } P_{abs}, P_{diff} \text{ et } P_{nul} \\ \vec{U}' \text{ une VA à valeurs dans la sphère unité} \\ \text{et de densité de probabilité } p_{\vec{U}'|S'} = \Phi(\vec{R}'; \vec{U}' \cdot \vec{u}) \\ \theta_{abs,abs} + \theta_{abs,diff} + \theta_{abs,nul} = 1 \\ \delta \tilde{F}(\vec{r}; \vec{u}) \text{ un estimateur de } \delta f(\vec{r}; \vec{u}) \end{array} \right. \quad (2.34b)$$

Lors de l'estimation récursive de δf , la mise à jour du total de pondération restante suit donc la formule :

$$\Pi_{i+1} = \begin{cases} \frac{k_d(\vec{R}')}{P_{diff} \hat{k}_t(\vec{R}')} \Pi_i & \text{si } Col = diff \\ \frac{k_n(\vec{R}')}{P_{nul} \hat{k}_t(\vec{R}')} \Pi_i & \text{si } Col = nul \\ 0 & \text{si } Col = abs \end{cases} \quad (2.34c)$$

Nous avons, dans le cas de l'utilisation d'une roulette russe, choisi des probabilités plus compliquées pour l'événement d'origine du photon, comparativement à ce que nous utilisons avec un partitionnement de l'énergie. Si, au point \vec{R}' de collision du photon, on a $k_n(\vec{R}') \geq -k_a(\vec{R}')/2$, on prend :

$$\begin{aligned} P_{abs} &= \frac{k_a(\vec{R}') + k_n(\vec{R}') - |k_n(\vec{R}')|}{\hat{k}_t(\vec{R}')} & \theta_{abs,abs} &= 1 + \frac{k_n(\vec{R}') - |k_n(\vec{R}')|}{k_a(\vec{R}')} \\ P_{diff} &= \frac{k_d(\vec{R}')}{\hat{k}_t(\vec{R}')} & \theta_{abs,diff} &= 0 \\ P_{nul} &= \frac{|k_n(\vec{R}')|}{\hat{k}_t(\vec{R}')} & \theta_{abs,nul} &= \frac{k_n(\vec{R}') - |k_n(\vec{R}')|}{k_a(\vec{R}')} \end{aligned} \quad (2.34d)$$

Sinon on choisit :

$$\begin{aligned}
 P_{abs} &= \frac{k_n(\vec{R}')}{2k_n(\vec{R}')} \times \frac{2k_n(\vec{R}') + k_a(\vec{R}')}{2k_d(\vec{R}') + k_a(\vec{R}')} \\
 \theta_{abs,abs} &= \frac{\hat{k}_t(\vec{R}')}{2k_n(\vec{R}')} \times \frac{2k_n(\vec{R}') + k_a(\vec{R}')}{2k_d(\vec{R}') + k_a(\vec{R}')} \\
 P_{diff} &= \frac{k_d(\vec{R}')(k_a(\vec{R}')^2 - 4k_d(\vec{R}')k_n(\vec{R}'))}{-2k_n(\vec{R}')(k_d(\vec{R}') - k_n(\vec{R}'))(2k_d(\vec{R}') + k_a(\vec{R}'))} \\
 \theta_{abs,diff} &= \frac{k_d(\vec{R}')}{k_a(\vec{R}')} \left(\frac{\hat{k}_t(\vec{R}')(k_a(\vec{R}')^2 - 4k_d(\vec{R}')k_n(\vec{R}'))}{-2k_n(\vec{R}')(k_d(\vec{R}') - k_n(\vec{R}'))(2k_d(\vec{R}') + k_a(\vec{R}'))} - 1 \right) \\
 P_{nul} &= \frac{k_a(\vec{R}')^2 - 4k_d(\vec{R}')k_n(\vec{R}')}{2(k_d(\vec{R}') - k_n(\vec{R}'))(2k_d(\vec{R}') + k_a(\vec{R}'))} \\
 \theta_{abs,nul} &= \frac{\hat{k}_t(\vec{R}')(k_a(\vec{R}')^2 - 4k_d(\vec{R}')k_n(\vec{R}'))}{2k_a(\vec{R}')(k_d(\vec{R}') - k_n(\vec{R}'))(2k_d(\vec{R}') + k_a(\vec{R}'))} - \frac{k_n(\vec{R}')}{k_a(\vec{R}')}
 \end{aligned} \tag{2.34e}$$

Ces choix paramétriques respectent les relations suivantes, quelle que soit la valeur de $k_n(\vec{R}')$ relativement à $k_a(\vec{R}')$:

$$\begin{aligned}
 \frac{\theta_{abs,abs}}{P_{abs}} \frac{k_a(\vec{R}')}{\hat{k}_t(\vec{R}')} &= 1 \\
 \frac{1}{P_{diff}} \left(\theta_{abs,diff} \frac{k_a(\vec{R}')}{\hat{k}_t(\vec{R}')} + \frac{k_d(\vec{R}')}{\hat{k}_t(\vec{R}')} \right) &= 1 \\
 \frac{1}{P_{nul}} \left(\theta_{abs,nul} \frac{k_a(\vec{R}')}{\hat{k}_t(\vec{R}')} + \frac{k_n(\vec{R}')}{\hat{k}_t(\vec{R}')} \right) &= 1
 \end{aligned} \tag{2.35a}$$

La relation suivante aussi est vérifiée, même en cas d'utilisation d'un partitionnement de l'énergie :

$$\frac{P_{diff}}{k_d(\vec{R}')} = \frac{P_{nul}}{|k_n(\vec{R}')|} \tag{2.35b}$$

Lorsque l'algorithme de suivi de photon passe en roulette russe, ce changement de fonctionnement a lieu aussi pour la prise en compte de la réflexion aux parois. Ainsi, si le photon suivi vient d'une paroi on estime désormais la fonction de distribution par :

$$\begin{aligned}
 \delta f(\vec{r}; \vec{u}_{out}) &= E \left(H(Col = refl) \delta \tilde{F}(\vec{r}; \vec{U}_{in}) + \right. \\
 &\quad \left. H(Col = abs) \times -f_{eq}(\vec{r}_0) \right)
 \end{aligned} \tag{2.36a}$$

où Col est une variable aléatoire à valeurs dans $\{abs; refl\}$ de probabilités respectives ε et $1 - \varepsilon$, et \vec{U}_{in} est la même variable aléatoire

qu'en (2.33), quand on partitionnait l'énergie. Le total de pondération restante est cette fois-ci actualisé selon un schéma tout-ou-rien :

$$\begin{cases} \Pi_{i+1} = \Pi_i & \text{si } Col = refl \\ \Pi_{i+1} = 0 & \text{si } Col = abs \end{cases} \tag{2.36b}$$

●●●● Le détail de l'implémentation qui a été réalisée (en langage C++) de l'algorithme complet détaillé au cours de cette section est donné en annexe D.1.

2.4.4 Résultats et commentaires

L'algorithme expliqué à la section précédente 2.4.3 a été testé avec une série de valeurs pour les paramètres précisant la situation physique expliquée précédemment. Les résultats sont présentés dans les figures de 2.4 à 2.13.

Dans ces figures, on s'intéresse à l'évolution de :

- la variance de l'estimateur de $S_r(\vec{r}_0)$ généré, notée $\text{Var}(W)$,
- le nombre moyen de points de collision dans le milieu durant chaque estimation ponctuelle de $S_r(\vec{r}_0)$, noté rec_p . Cette estimation se fait en effet en construisant un chemin optique passant par \vec{r}_0 . Les collisions sur les parois ne sont pas comptées dans rec_p . À chaque point de

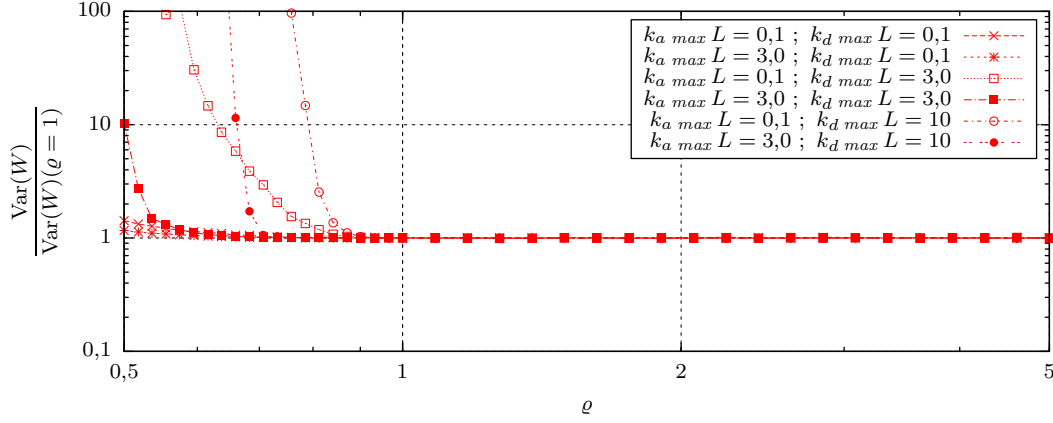
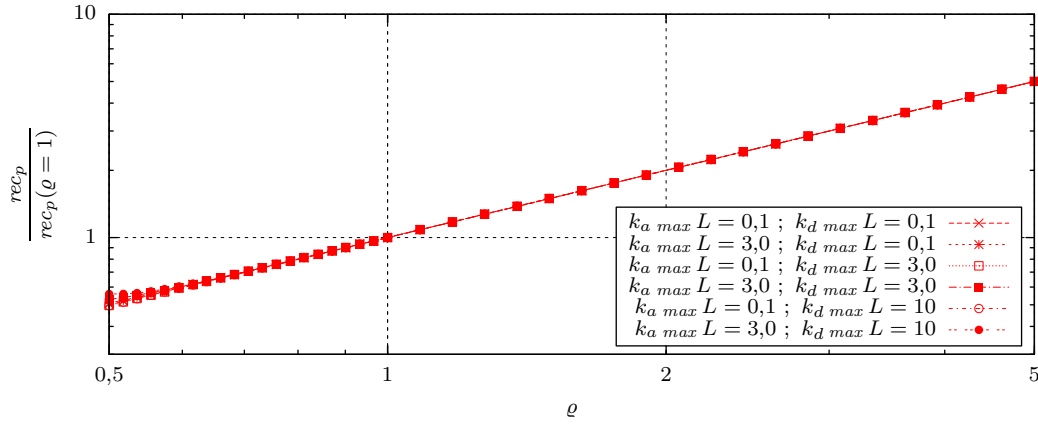
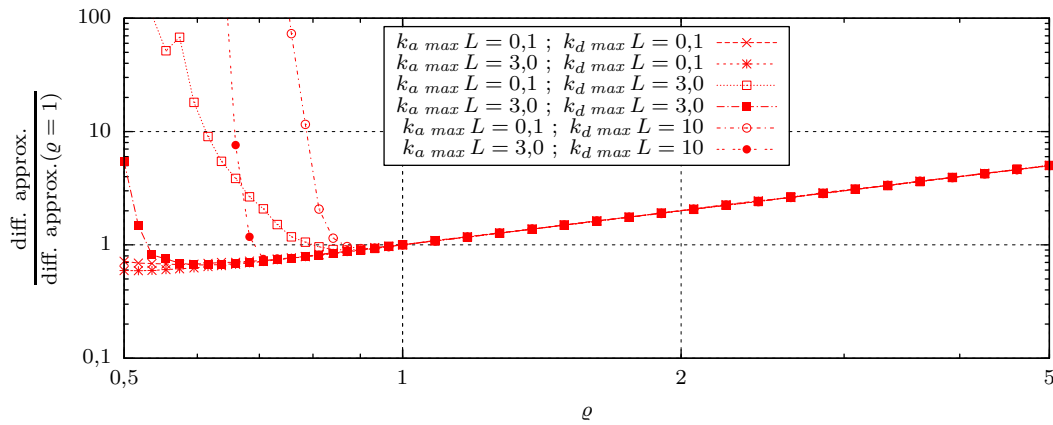

 (a) Profils de la variance $\text{Var}(W)$ de l'estimateur généré, en fonction de ρ

 (b) Profils de la moyenne rec_p du nombre de collisions dans le milieu par estimation, en fonction de ρ

 (c) Profils de la « difficulté du calcul », approximée comme $\text{Var}(W) \times \text{rec}_p$, en fonction de ρ

 FIGURE 2.4 – Tests de l'algorithme présenté à la section 2.4.3, en fonction du coefficient d'extinction majoré $\hat{k}_t = \rho k_{t \max}$. Les parois sont noires ($\varepsilon = 1$), le point sonde est le milieu du cube ($\vec{r}_0 = (0; 0; 0)$), et l'algorithme fonctionne en roulette russe pure ($\varpi = 1$).

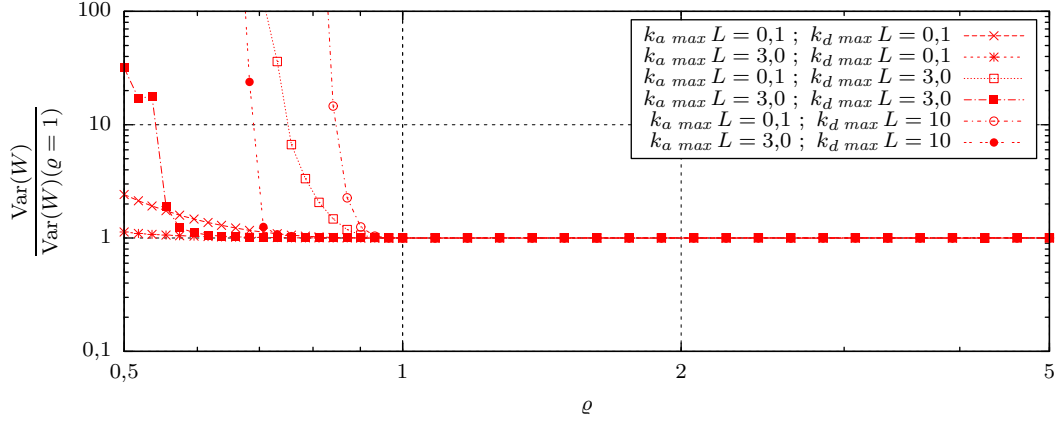
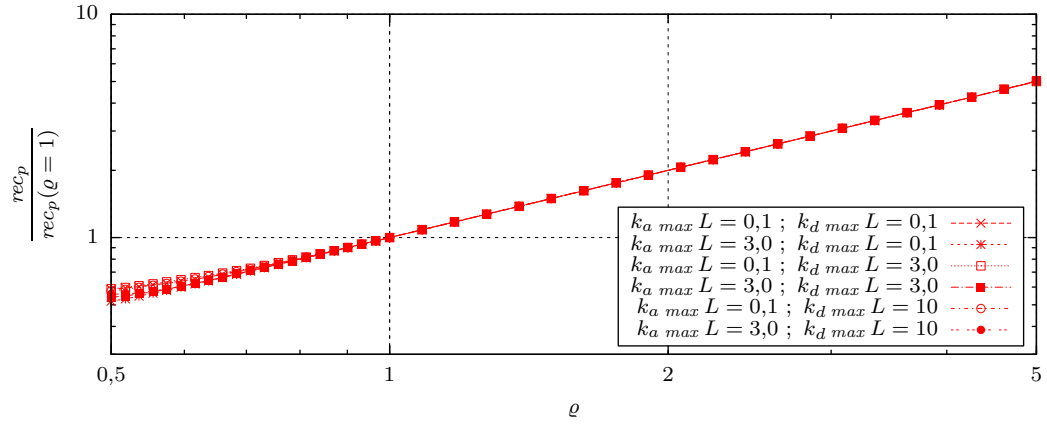
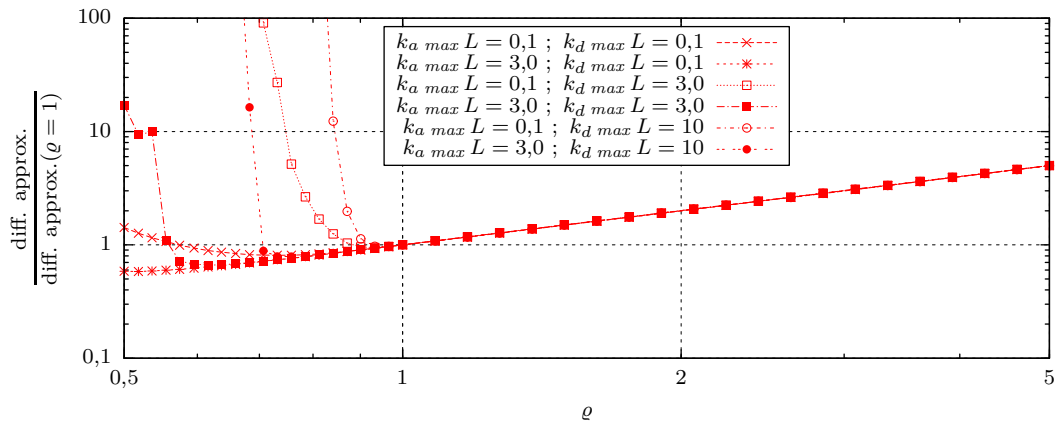

 (a) Profils de la variance $\text{Var}(W)$ de l'estimateur généré, en fonction de ρ

 (b) Profils de la moyenne rec_p du nombre de collisions dans le milieu par estimation, en fonction de ρ

 (c) Profils de la « difficulté du calcul », approximée comme $\text{Var}(W) \times \text{rec}_p$, en fonction de ρ

 FIGURE 2.5 – Tests de l'algorithme présenté à la section 2.4.3, en fonction du coefficient d'extinction majoré $\hat{k}_t = \rho k_{t \max}$. Les parois sont totalement réfléchissantes ($\varepsilon = 0$), le point sonde est le milieu du cube ($\vec{r}_0 = (0; 0; 0)$), et l'algorithme fonctionne en roulette russe pure ($\varpi = 1$).

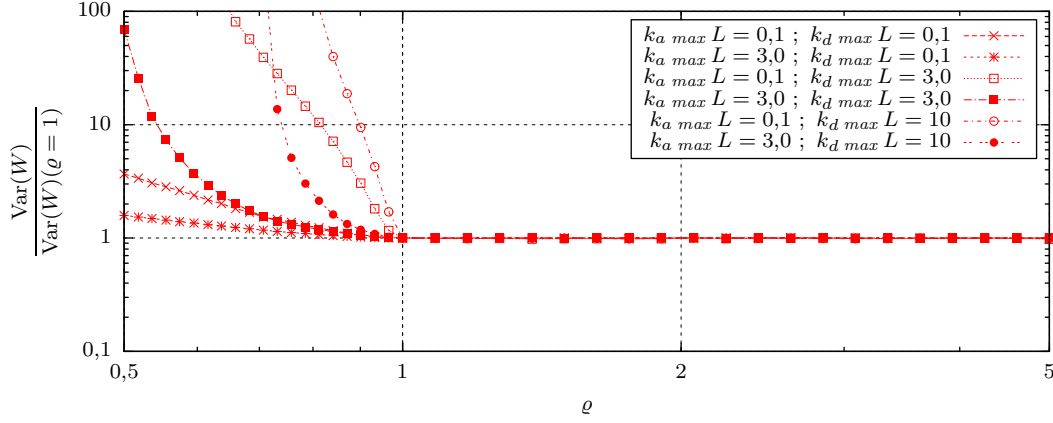
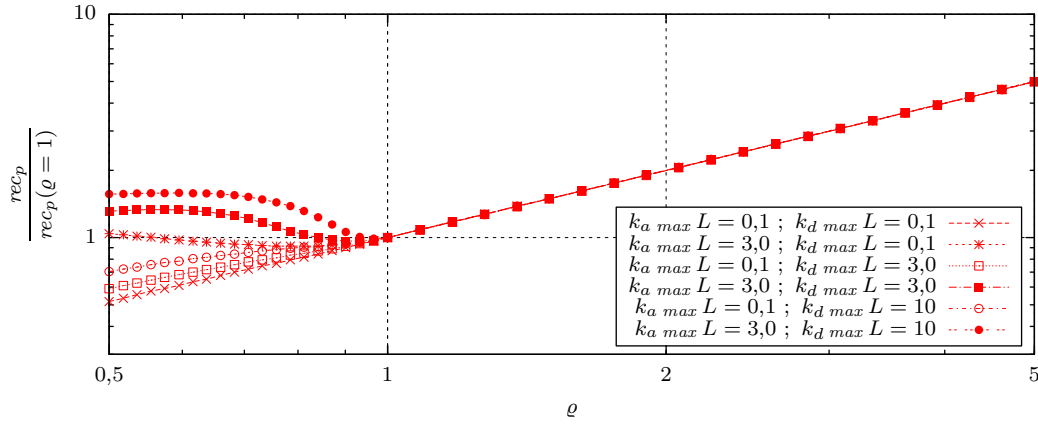
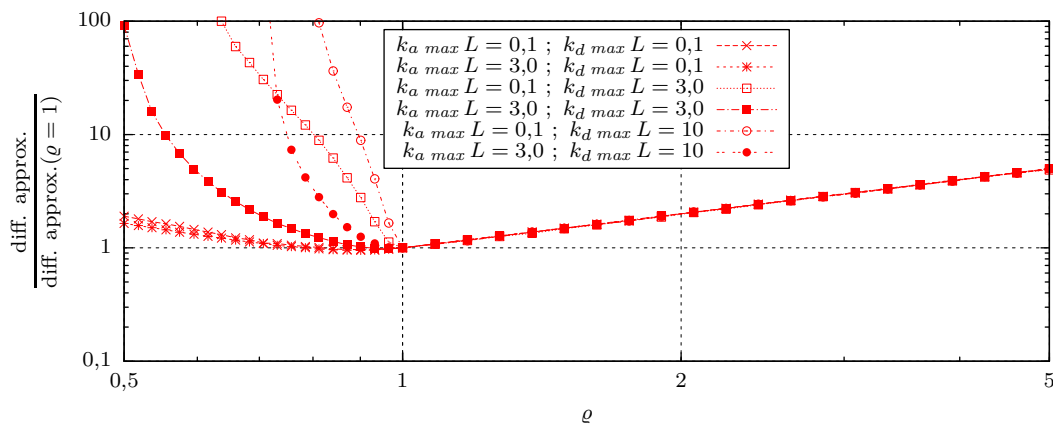

 (a) Profils de la variance $\text{Var}(W)$ de l'estimateur généré, en fonction de ρ

 (b) Profils de la moyenne rec_p du nombre de collisions dans le milieu par estimation, en fonction de ρ

 (c) Profils de la « difficulté du calcul », approximée comme $\text{Var}(W) \times \text{rec}_p$, en fonction de ρ

 FIGURE 2.6 – Tests de l'algorithme présenté à la section 2.4.3, en fonction du coefficient d'extinction majoré $\hat{k}_t = \rho k_t \max$. Les parois sont noires ($\varepsilon = 1$), le point sonde est sur une paroi ($\vec{r}_0 = (-L; 0; 0)$), et l'algorithme fonctionne en roulette russe pure ($\varpi = 1$).

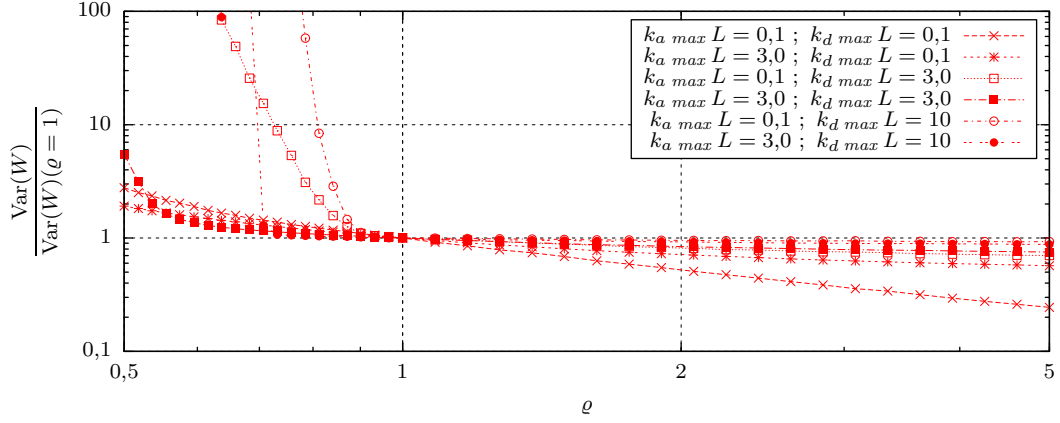
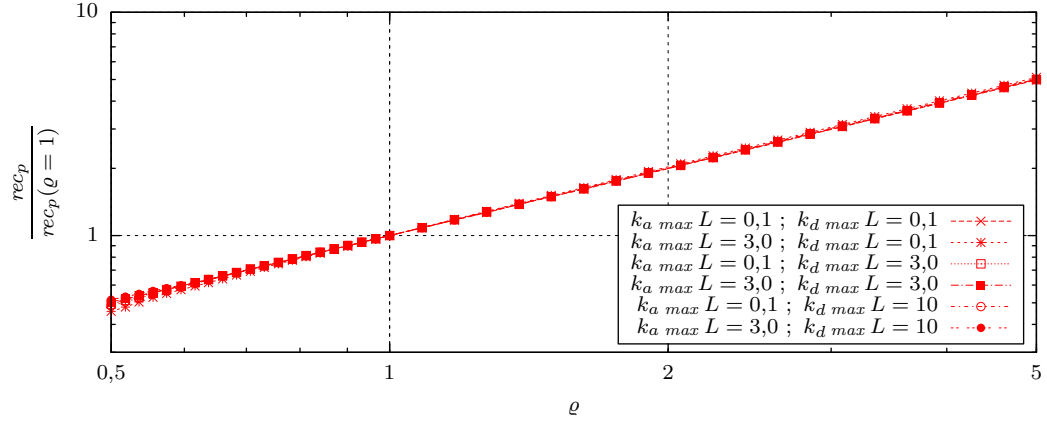
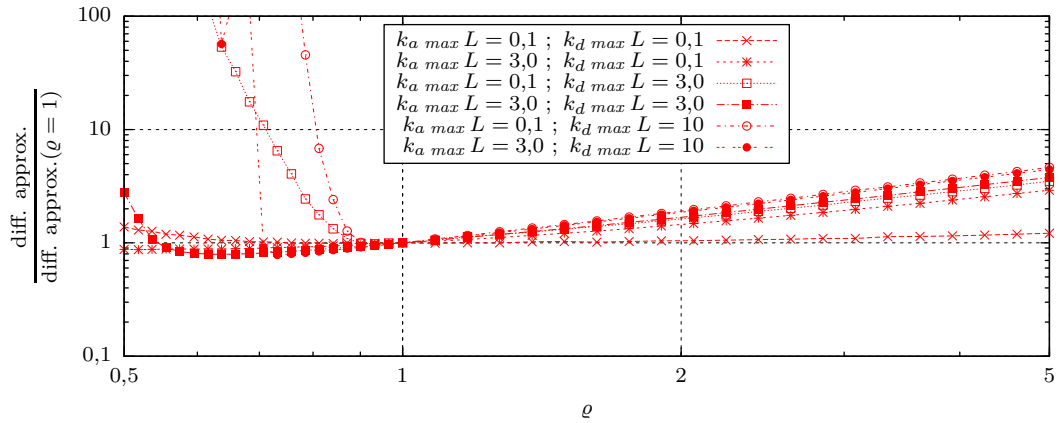

 (a) Profils de la variance $\text{Var}(W)$ de l'estimateur généré, en fonction de ρ

 (b) Profils de la moyenne rec_p du nombre de collisions dans le milieu par estimation, en fonction de ρ

 (c) Profils de la « difficulté du calcul », approximée comme $\text{Var}(W) \times \text{rec}_p$, en fonction de ρ

 FIGURE 2.7 – Tests de l'algorithme présenté à la section 2.4.3, en fonction du coefficient d'extinction majoré $\hat{k}_t = \rho k_{t \max}$. Les parois sont noires ($\varepsilon = 1$), le point sonde est le milieu du cube ($\vec{r}_0 = (0; 0; 0)$), et l'algorithme fonctionne en roulette russe déclenchable ($\varpi = 0,1$).

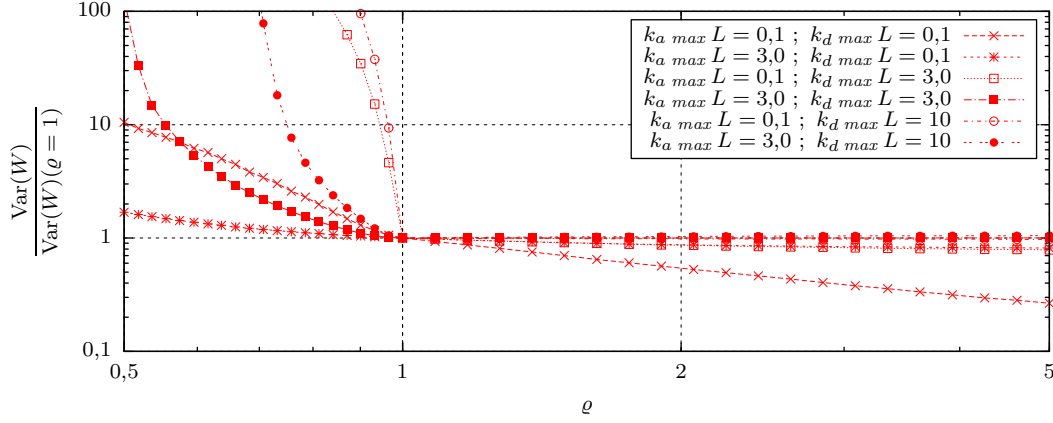
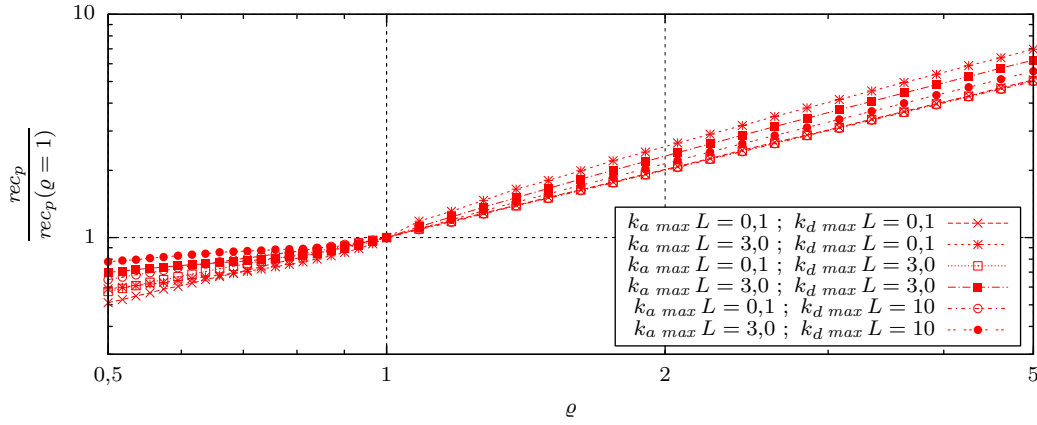
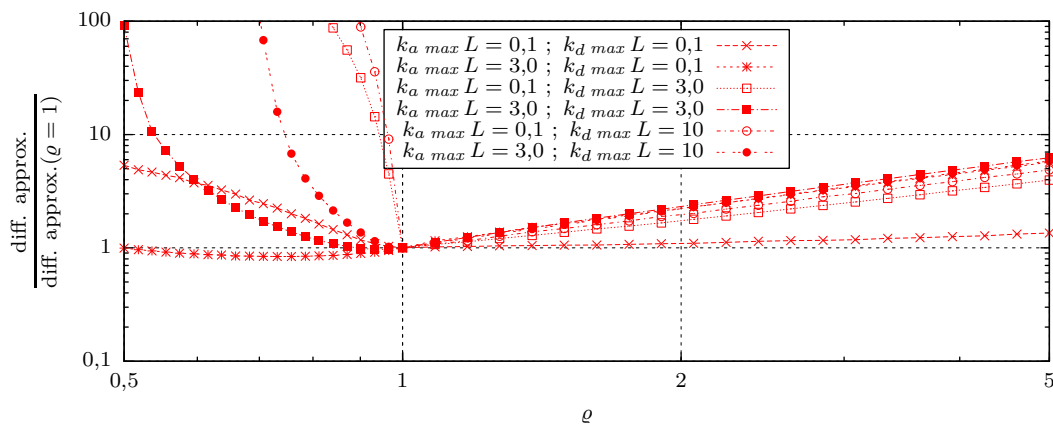

 (a) Profils de la variance $\text{Var}(W)$ de l'estimateur généré, en fonction de ρ

 (b) Profils de la moyenne rec_p du nombre de collisions dans le milieu par estimation, en fonction de ρ

 (c) Profils de la « difficulté du calcul », approximée comme $\text{Var}(W) \times \text{rec}_p$, en fonction de ρ

 FIGURE 2.8 – Tests de l'algorithme présenté à la section 2.4.3, en fonction du coefficient d'extinction majoré $\hat{k}_t = \rho k_{t \max}$. Les parois sont noires ($\varepsilon = 1$), le point sonde est sur une paroi ($\vec{r}_0 = (-L; 0; 0)$), et l'algorithme fonctionne en roulette russe déclenchable ($\varpi = 0,1$).

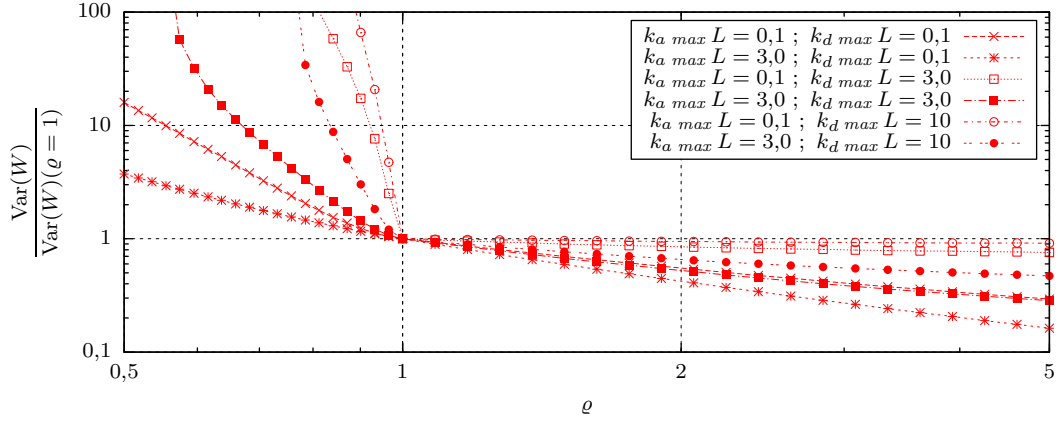
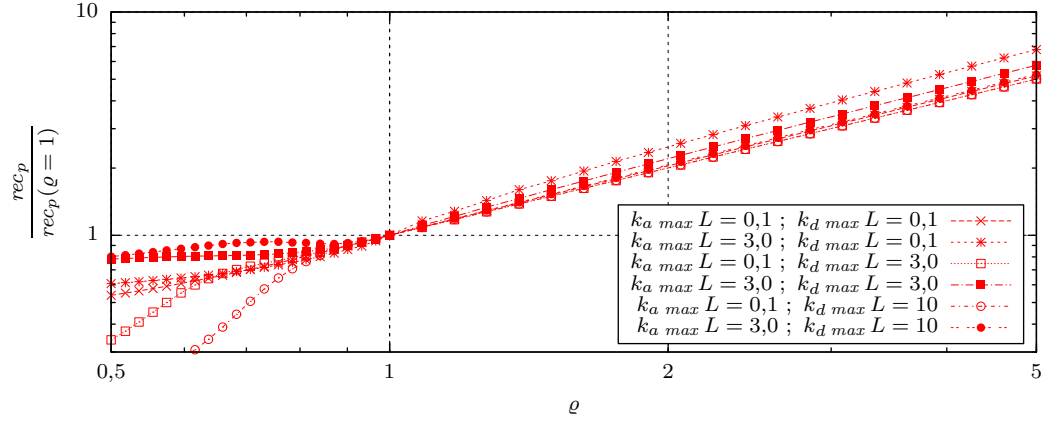
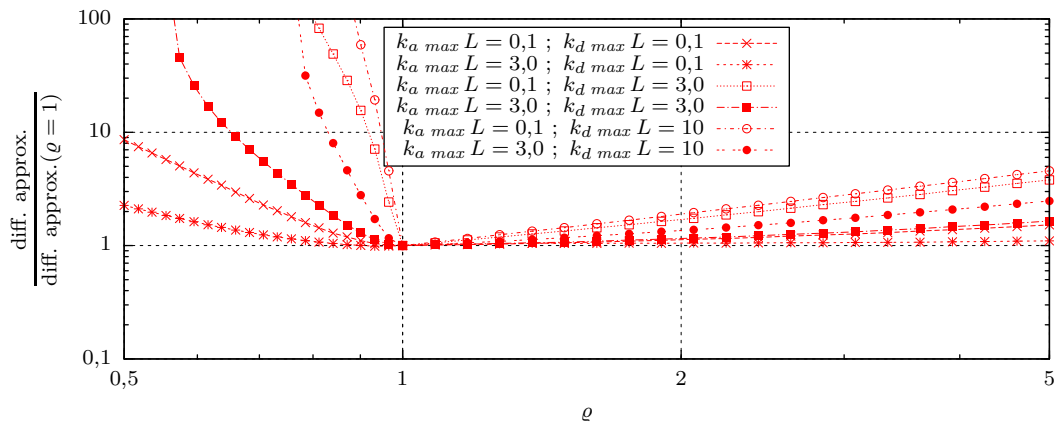
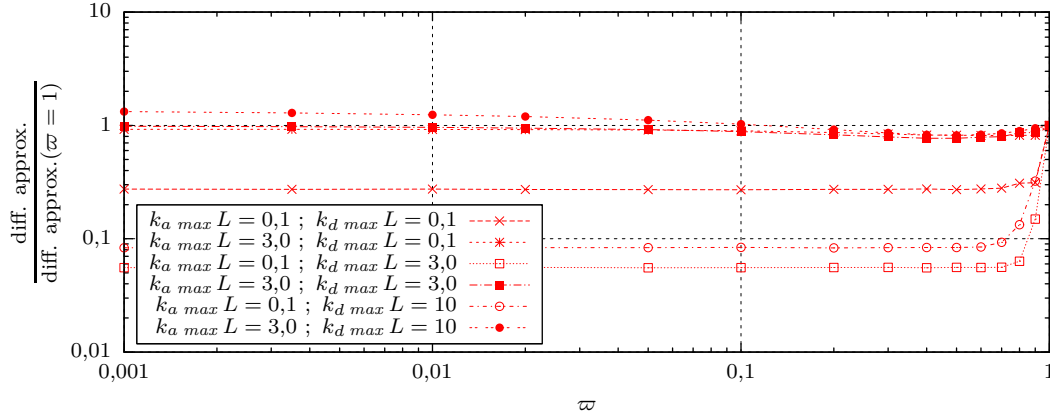
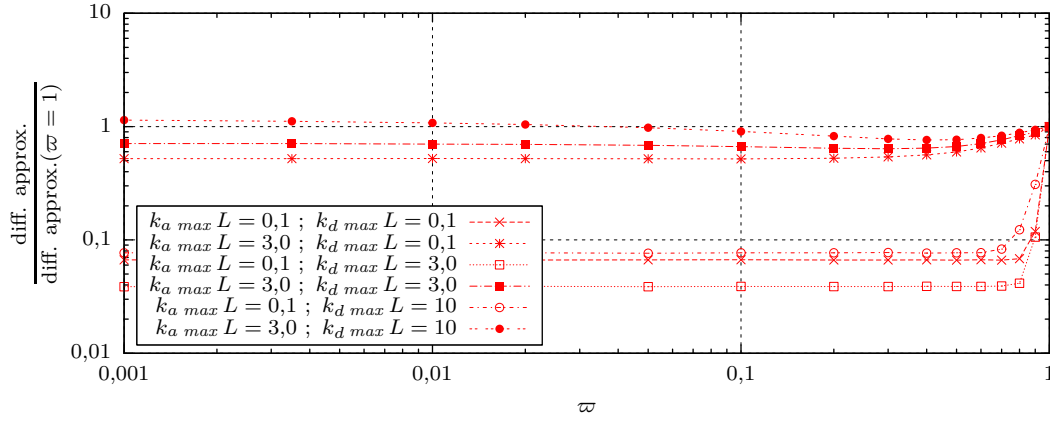

 (a) Profils de la variance $\text{Var}(W)$ de l'estimateur généré, en fonction de ρ

 (b) Profils de la moyenne rec_p du nombre de collisions dans le milieu par estimation, en fonction de ρ

 (c) Profils de la « difficulté du calcul », approximée comme $\text{Var}(W) \times \text{rec}_p$, en fonction de ρ

 FIGURE 2.9 – Tests de l'algorithme présenté à la section 2.4.3, en fonction du coefficient d'extinction majoré $\hat{k}_t = \rho k_{t \max}$. Les parois sont totalement réfléchissantes ($\varepsilon = 0$), le point sonde est sur une paroi ($\vec{r}_0 = (-L; 0; 0)$), et l'algorithme fonctionne en roulette russe déclenchable ($\varpi = 0,1$).



(a) Profils de la « difficulté du calcul », approximée comme $\text{Var}(W) \times \text{rec}_p$, en fonction de ϖ . ϱ est pris égal à 1.



(b) Idem (a), mais avec $\varrho = 5$

FIGURE 2.10 – Tests de l’algorithme présenté à la section 2.4.3, en fonction du seuil de déclenchement de la roulette russe. Les parois sont noires ($\varepsilon = 1$) et le point sonde est le milieu du cube ($\vec{r}_0 = (0; 0; 0)$).

collision dans le milieu, il faut déterminer les propriétés locales k_a , k_d et f_{eq} .

Ce qui est noté dans les figures « diff. approx. » est une approximation de la difficulté du calcul. Celle-ci est prise comme égale au produit de $\text{Var}(W)$ et de rec_p , car :

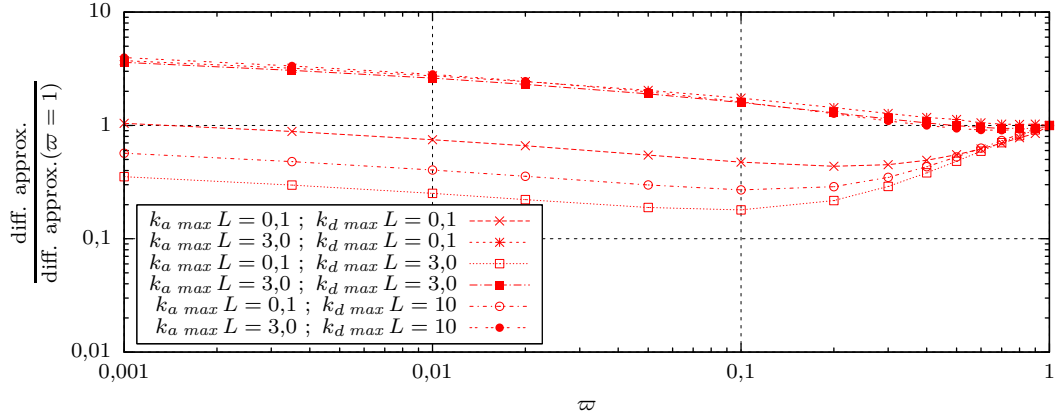
- Le nombre d’échantillonnages d’un estimateur nécessaires pour obtenir une précision donnée est proportionnel à la variance. Ceci est une propriété générale de la méthode de Monte-Carlo.
- La difficulté de calcul est considéré proportionnelle à rec_p . Cela veut dire que

la majorité de l’effort computationnel est dépensé à calculer les propriétés locales du milieu.

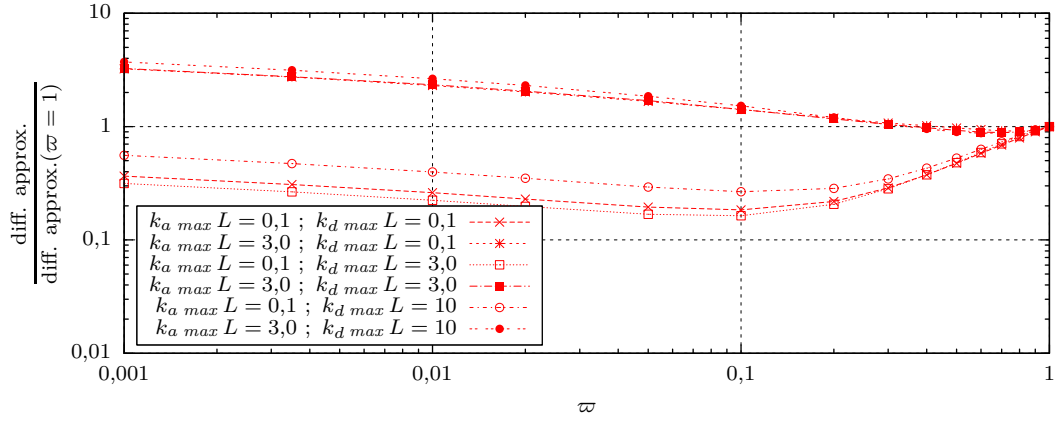
On aurait pu raisonner différemment. Par exemple si on considère que le calcul d’intersection rayon/géométrie est le plus coûteux, informatiquement parlant, il aurait fallu comptabiliser le nombre de recours à ce calcul plutôt que la grandeur rec_p choisie.

Ces grandeurs, $\text{Var}(W)$, rec_p et $\text{Var}(W) \times \text{rec}_p$ sont considérées dans leur évolution selon les paramètres algorithmiques :

- \hat{k}_t , le coefficient d’extinction majoré,



(a) Profils de la « difficulté du calcul », approximée comme $\text{Var}(W) \times \text{rec}_p$, en fonction de ϖ . ϱ est pris égal à 1.



(b) Idem (a), mais avec $\varrho = 5$

FIGURE 2.11 – Tests de l’algorithme présenté à la section 2.4.3, en fonction du seuil de déclenchement de la roulette russe. Les parois sont totalement réfléchissantes ($\varepsilon = 0$) et le point sonde est le milieu du cube ($\vec{r}_0 = (0; 0; 0)$).

défini à partir de ϱ ,

- ϖ , le seuil de déclenchement de la roulette russe,

pour quelques jeux des paramètres $(k_a \max; k_d \max; \varepsilon; \vec{r}_0)$.

Quelques remarques sur les résultats :

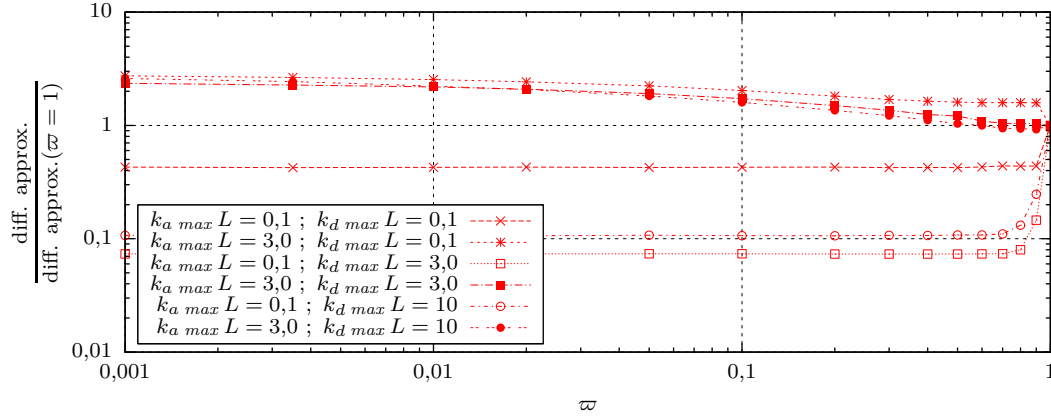
- Quand l’algorithme fonctionne en roulette russe pure ($\varpi = 1$), et tant que \hat{k}_t est bien un majorant de k_t :
 - \hat{k}_t n’a aucune influence sur la variance de l’estimateur obtenu,

- le nombre de collisions dans le milieu, réelles ou fictives, est proportionnel à \hat{k}_t .

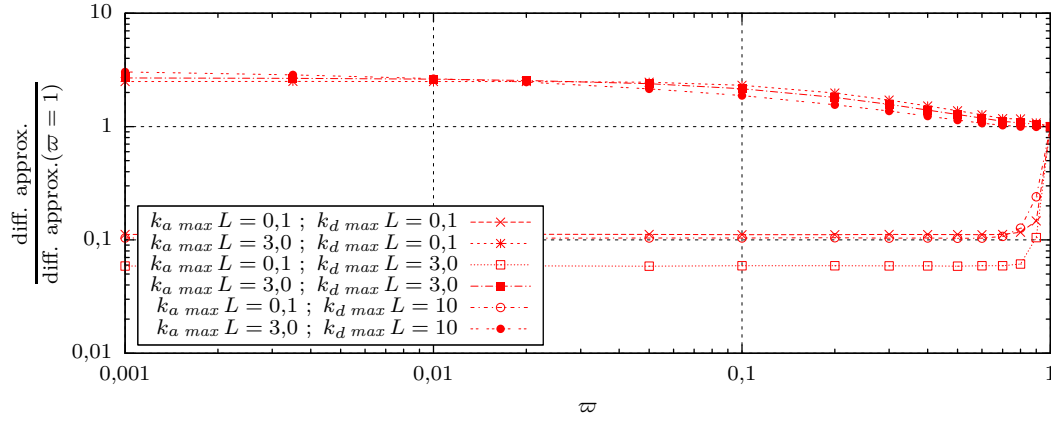
C’était attendu, compte-tenu de ce qu’on a pu expliquer au cours de la section 2.3.

- Il est plus grave de choisir un \hat{k}_t trop petit quand l’extinction est majoritairement due à de la diffusion.

Il faut probablement y voir une conséquence des choix algorithmiques détaillés que nous avons fait, dans l’expression des P_{event} et des $\theta_{abs,event}$. Nous avons en fait suivi la 4^e proposi-



(a) Profils de la « difficulté du calcul », approximée comme $\text{Var}(W) \times \text{rec}_p$, en fonction de ϖ . ϱ est pris égal à 1.



(b) Idem (a), mais avec $\varrho = 5$

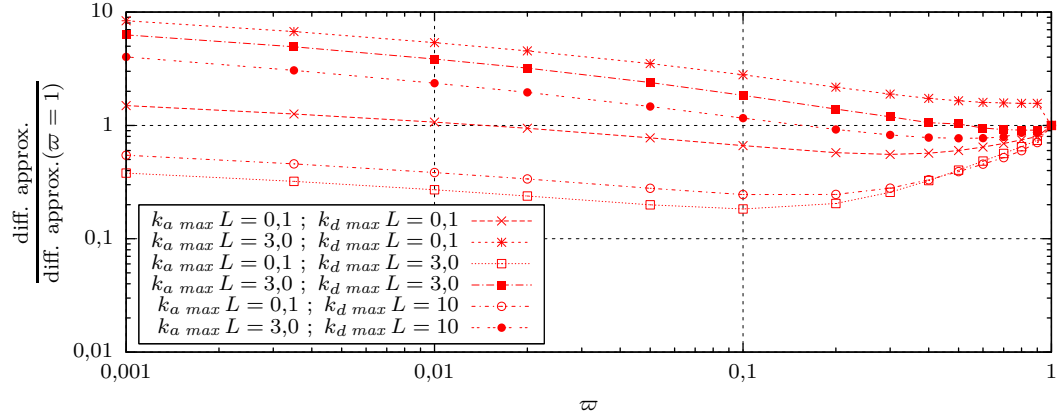
FIGURE 2.12 – Tests de l’algorithme présenté à la section 2.4.3, en fonction du seuil de déclenchement de la roulette russe. Les parois sont noires ($\varepsilon = 1$) et le point sonde est sur une paroi ($\vec{r}_0 = (-L; 0; 0)$).

tion de la section 2.2.3, en essayant d’obtenir un estimateur de f de variance nulle en cas d’équilibre thermodynamique global, comme cela est suggéré dans le début de la section 2.3.

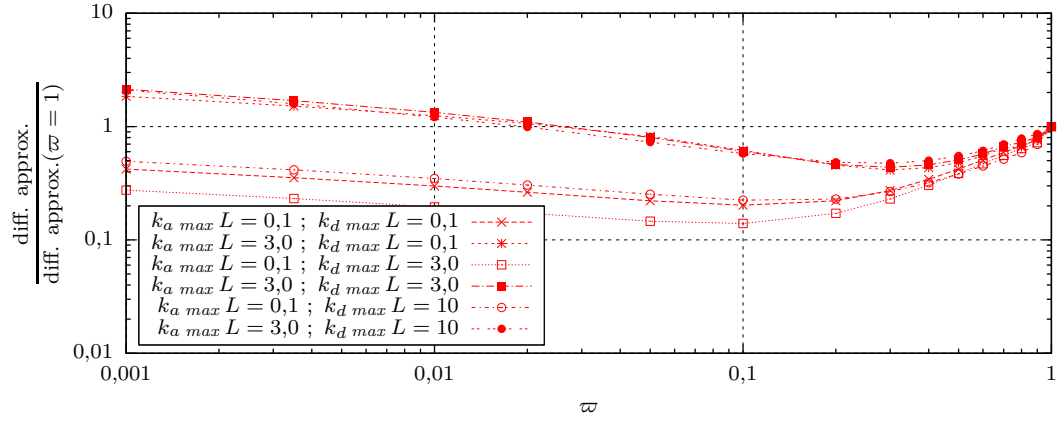
Cet objectif de variance nulle n’était qu’un guide pour la construction des P_{event} et des $\theta_{abs,event}$ ³. Mais il nous a fait placer des contributions de l’émission thermique dans les termes récurrents (de collision nulle et de diffusion),

de sorte que si \hat{k}_t se révèle trop petit, des termes d’émission thermique puissent se compenser entre les différents points de collision échantillonnés successivement. Si \hat{k}_t se révèle trop petit, on rogne prioritairement sur la probabilité de choisir un événement d’absorption (voir les détails de la section précédente), en espérant une compensation dans les collisions suivantes. Mais si $k_d \gg k_a$, si \hat{k}_t est trop petit cette astuce n’est rapidement plus utilisable.

3. D’ailleurs, le formalisme en PNE garantit à lui seul la nullité systématique de notre estimateur de $S_r(\vec{r}_0)$ en cas d’équilibre thermodynamique global.



(a) Profils de la « difficulté du calcul », approximée comme $\text{Var}(W) \times \text{rec}_p$, en fonction de ϖ . ρ est pris égal à 1.



(b) Idem (a), mais avec $\rho = 5$

FIGURE 2.13 – Tests de l’algorithme présenté à la section 2.4.3, en fonction du seuil de déclenchement de la roulette russe. Les parois sont totalement réfléchissantes ($\varepsilon = 0$) et le point sonde est sur une paroi ($\vec{r}_0 = (-L; 0; 0)$).

- Utiliser un partitionnement de l’énergie permet parfois, comme annoncé en section 2.3, de diminuer la pénalité (en efficacité algorithmique) provoquée par une trop large majoration de k_t . Mais c’est loin d’être toujours le cas. Par ailleurs, la pénalité provoquée par une majoration insuffisante de k_t semble plus grande quand on utilise un partitionnement de l’énergie.
- Si on utilise une roulette russe déclen-

chable, il n’est pas utile d’attendre d’avoir très fortement entamé le total des pondérations restantes Π pour déclencher la roulette russe. 0,1 est une valeur suffisamment basse pour ϖ dans toute notre campagne de tests.

●●● Le lecteur que les choix algorithmiques dans les ACNs intéressent pourra continuer par lire par exemple les publications [43, 89].

2.5 ACNs et estimation de l'extinction

2.5.1 Intérêt (pour nous) de cette démarche

Dans les sections précédentes, nous montrions comment les « algorithmes à collisions nulles » permettent de gérer la variabilité du coefficient d'absorption (ou de diffusion), quand on calcule le transfert radiatif par MMC. Dans la formulation intégrale sans collisions nulles de l'équation de transfert radiatif (1.43) (ou plus simplement (2.4)), le terme exponentiel d'intégrale provenant de la loi de Beer est susceptible de gêner. L'addition de collisionneurs fictifs permet de traiter ce terme directement dans tout algorithme de Monte-Carlo estimant une grandeur radiative. On évite ainsi toutes les approximations consistant à mailler le domaine pour évaluer l'épaisseur optique le long du chemin.

Mais dans ces raisonnements, on a toujours supposé que le coefficient d'extinction k_t était connu. La seule question était sa présence, sous forme intégrale, à l'intérieur de l'exponentielle. En cinétique des gaz k_t sera inconnu. En effet, comme les molécules d'un gaz collisionnent entre elles, la fréquence de collision pour une molécule donnée dépend de la fonction de distribution, et donc de la solution du problème que l'on voudra calculer.

Dans son travail de thèse [49], Mathieu GALTIER a détaillé une partie de la démarche que nous allons proposer pour la cinétique des gaz. L'objet d'étude de M. GALTIER était le transfert radiatif dans les milieux gazeux semi-transparents. Il considérait certes que le coefficient d'extinction était connaissable en entrée, parce que fonction seulement des conditions thermodynamiques du gaz supposées connues, mais sa problématique était que ce coefficient d'extinction était très difficilement pré-calculable. En effet, dans un gaz, le coefficient d'absorption est la somme des contributions de très nombreuses transitions entre états énergétiques internes des molécules (typiquement une centaine de millions). M. GALTIER a alors choisi d'écarter l'idée d'une évaluation préalable de k_a et de k_t , et il a vu dans les ACNs un moyen de faire traiter la sommation des contributions des transitions par l'algorithme de Monte-Carlo lui-même. L'idée est globalement la même que celles des ACNs usuels ; mais au lieu d'échantillonner seulement un espace de chemins optiques contenant des collisions nulles, on échantillonne un espace de dimension plus élevé, résultant d'un produit de l'espace des chemins et de l'espace des transitions.

Nous allons, dans cette section, reprendre ce propos de M. GALTIER. Nous allons d'abord introduire la physique et la problématique de calcul auquel il s'est confronté — afin que le lecteur saisisse le contexte, même si c'est un peu long. Ensuite nous introduirons sa proposition méthodologique. Partant de tout ce que nous avons précédemment expliqué sur les ACNs, cette étape finale sera en fait très courte.

2.5.2 Situation du travail de Mathieu Galtier

Comme expliqué et montré dans les parties précédentes de ce chapitre, les ACNs permettent d'outrepasser les difficultés liées au calcul numérique par méthode de Monte-Carlo du transfert radiatif (du transport corpusculaire, de façon plus générale) dans les milieux participants aux propriétés optiques hétérogènes — excepté l'indice de réfraction. Par ailleurs, le travail de thèse de Patrice PÉREZ [90], effectué en collaboration entre le centre RAPSODEE, l'IRIT⁴, et le LAPLACE, avait montré ce que pouvaient représenter les apports des outils conceptuels et informatiques de la communauté de la synthèse d'image pour gérer la complexité géométrique des problèmes rencontrés en transfert radiatif (voir aussi la publication [32]). Pour la majorité des gens utilisant les méthodes de Monte-Carlo pour calculer le transfert radiatif dans des milieux isotropes et non-uniformes en équilibre thermodynamique local et à indice de réfraction constant, sans tenir compte de la polarisation, la gestion de la complexité géométrique des surfaces et des hétérogénéités optiques des milieux semi-transparents ne représente plus une difficulté dès lors que l'on dispose de ressources informatiques bureautiques modernes. Les difficultés qui restent sont plutôt :

- la gestion des milieux optiquement épais en diffusion, notoirement très difficile pour les méthodes de Monte-Carlo,
- la connaissance et la prise en compte des propriétés diffusantes des milieux semi-transparents,
- toute la complexité spectrale des milieux gazeux.

Ce dernier point constitue le sujet de la thèse de Mathieu GALTIER, soutenue récemment en Décembre 2014.

2.5.3 Problématique spectrale des milieux gazeux

2.5.3.1 Complexité du coefficient d'absorption

●●● Dans les gaz de faible densité, l'absorption et l'émission de rayonnement thermique est un phénomène se déroulant à l'échelle moléculaire. Un photon est absorbé ou émis à l'occasion d'une transition d'une molécule entre deux états internes, associés à des énergies différentes.

On écrit ainsi le coefficient d'absorption comme la somme des contributions de toutes les espèces chimiques présentes. De même, on écrit la contribution à l'absorption de chaque espèce chimique comme la somme des contributions de toutes les transitions énergétiques que ses molécules sont susceptibles de subir. Premièrement :

$$k_a = \sum_{m=1}^{N_m} \eta_m \sigma_m \quad (2.37a)$$

4. Institut de Recherche en Informatique de Toulouse

où k_a est le coefficient d'absorption en m^{-1} , η_m est la concentration en l'espèce m (en mol/m^3 par exemple), et σ_m est la section efficace d'absorption/émission de l'espèce m (ici en m^2/mol). Ensuite :

$$\sigma_m = \sum_{j=1}^{N_t(m)} \sigma_{m,j} \quad (2.37b)$$

où $\sigma_{m,j}$ dénote la contribution de la transition j à la section efficace d'absorption de l'espèce m . Bien sûr, les sections efficaces dépendent de la fréquence du rayonnement considéré.

●●●● Selon la physique quantique, l'énergie interne d'une molécule isolée ne peut prendre qu'un ensemble discret de valeurs. Il s'ensuit qu'une molécule ne peut émettre ou absorber un photon que si l'énergie de celui-ci correspond *exactement* à une transition entre 2 de ses énergies internes discrètes possibles. Les contributions de chaque transition au coefficient d'absorption devraient, suivant ceci, être des raies d'épaisseur nulle sur l'échelle des fréquences.

Cela devrait *a priori* rendre l'absorption et l'émission du rayonnement dans les gaz impossible. En fait, dans un gaz les raies d'absorption associées aux transitions subissent chacune des élargissements. Ces élargissements peuvent être de 3 natures, que nous ne détaillerons pas plus avant : l'élargissement naturel (dû à l'incertitude quantique), l'effet Doppler, et l'élargissement collisionnel (dû aux collisions entre molécules).

Malgré ces élargissements, les raies restent flagrantes dans le spectre d'absorption d'un gaz. Et étant donné le nombre de ces raies (jusqu'à plusieurs centaines de millions!), le spectre d'absorption des gaz, même quand ils ne contiennent qu'une seule espèce chimique, exhibe généralement un profil très torturé. La figure 7 en donne un exemple.

●●●● Pour finir, les conditions thermodynamiques dans le gaz vont modifier le spectre d'absorption associé à chaque espèce. En effet :

- La température va influencer sur la distribution des molécules parmi les divers états qu'elles peuvent occuper. Cela influe en effet sur l'absorptivité, car la capacité d'une molécule à absorber ou émettre un photon dépend de son état initial.

Remarque. Dans les chambres de combustion ou dans les atmosphères planétaires, les collisions entre molécules sont beaucoup plus nombreuses que leurs interactions avec le rayonnement. Comme ces collisions sont aussi des occasions de modification de l'énergie interne des molécules, la répartition des états des molécules est en permanence proche de l'équilibre transitoire imposé par la cinétique des gaz : cela fait partie de l'équilibre thermodynamique local.

5. Dans la distinction des espèces chimiques, la nature isotopique compte aussi car la masse des noyaux des atomes influe sur l'énergie associée aux états.

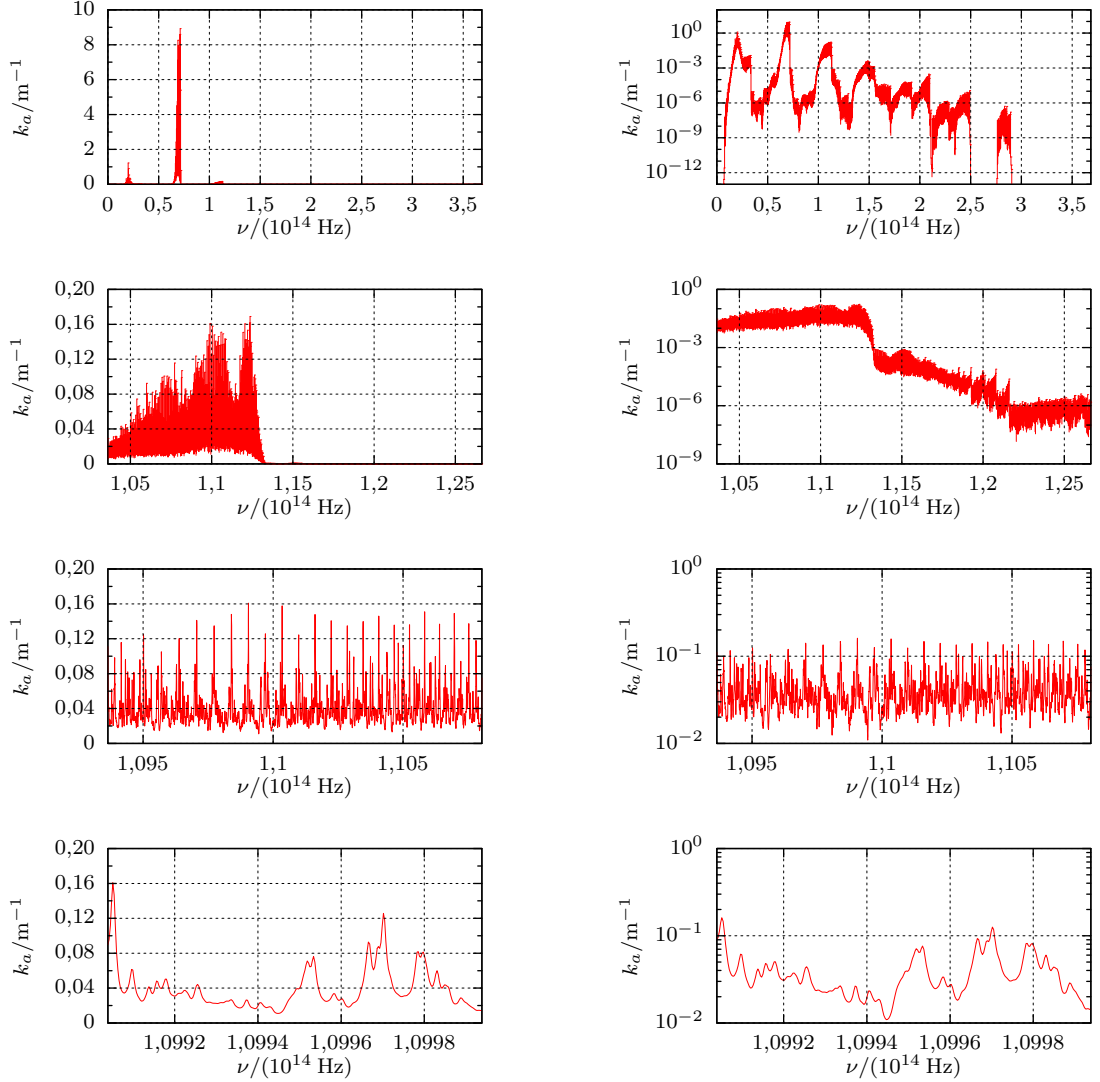


FIGURE 2.14 – Spectre d’absorption d’un mélange gazeux à une pression de 1 atm et une température de 1500 K, composé de 50 % de CO_2 et de 50 % de N_2 ⁶. Ces spectres sont calculés⁷ à partir de la base de données CDSD-4000. Les mêmes données sont représentées à gauche avec une échelle verticale linéaire et à droite avec une échelle verticale logarithmique, et entre chaque ligne il y a un grossissement sur l’échelle horizontale d’un facteur 16.

6. Dans ce cas, presque toute l’absorption/émission de rayonnement thermique est due au seul CO_2 .

7. Les profils de raie sont supposés Lorentzien, et sont tronqués à $7,5 \times 10^{11}$ Hz. Voir la table 2.1 page 118. Crédits : Mathieu GALTIER.

La micro-réversibilité entre absorption et émission (c'est-à-dire l'identité du terme d'émission avec $k_a f_{eq}(T(\vec{r}; t))$) est une des conséquence de cet équilibre.

- La température, la pression, et la masse des molécules en présence vont influencer sur leur vitesse moyenne et leur fréquence de collision, et ainsi sur les élargissements possibles des raies.

Ainsi, la section efficace d'absorption par une espèce chimique et isotopique va dépendre des conditions thermodynamiques locales du gaz.

Au total, le coefficient d'absorption se calcule comme ⁸ :

$$k_a(\vec{r}; \vec{u}; t) = \sum_{m=1}^{N_m} \eta_m(\vec{r}; t) \sum_{j=1}^{N_t(m)} \sigma_{m,j} \left(T(\vec{r}; t); p(\vec{r}; t); \left(\eta_{m'}(\vec{r}; t) \right)_{m' \in \llbracket 1; N_m \rrbracket} \right) \quad (2.38)$$

La dépendance des $\sigma_{m,j}$ envers la concentration en les différentes espèces est faible devant leur dépendance envers la pression et plus encore devant leur dépendance envers la température.

Pour aller plus loin dans la compréhension de l'absorption et de l'émission thermique dans les gaz, le lecteur intéressé pourra se reporter au manuscrit de thèse de M. GALTIER [49], ainsi qu'aux ouvrages [24, 52, 59, 86] considérés comme des références par les communautés scientifiques concernées.

2.5.3.2 Conséquences de cette complexité en calcul numérique du transfert radiatif

●●●● Le calcul prédictif de grandeurs radiatives dans une situation impliquant un milieu semi-transparent nécessite, *a priori*, de connaître le coefficient d'absorption en tout point, et pour toute fréquence.

Pour obtenir ce coefficient d'absorption, on aura généralement recours à une *base de données spectroscopiques*, appelée aussi *base de données de transitions*. Ces bases sont construites depuis plusieurs années par une large communauté de spectroscopistes. Leur principe est de rassembler un jeu de paramètres descriptifs pour chacune des transitions ayant un impact significatif sur le spectre d'absorption d'une espèce en particulier (les isotopes sont rassemblés par espèce chimique); et ce, pour toutes les espèces chimiques ayant de l'importance sur le transfert radiatif dans certaines applications. Chaque jeu de paramètres permet de calculer une section efficace de transition $\sigma_{m,j}$ dans une plage de pression et de température donnée. La table 2.1 liste quelques-unes de ces bases de données spectroscopiques.

Ces bases de données ont en général plusieurs versions, et sont encore aujourd'hui en cours d'édition, au fur et à mesure que de nouvelles transitions sont découvertes.

8. tant que l'on est en équilibre thermodynamique local

Base de données	Espèces	Gamme de températures
CDSD-1000	CO ₂	$\lesssim 1\,000\text{ K}$
CDSD-4000	CO ₂	$\lesssim 4\,000\text{ K}$
Geisa 2011	50 molécules	Basses températures
Hitemp Hitemp 2010	CO ₂ , H ₂ O, CO, NO, HO	$\lesssim 1\,000\text{ K}$ $\lesssim 3\,000\text{ K}$
Hitran 2008	42 molécules	Basses températures
Hitran 2012	47 molécules	

TABLE 2.1 – Principales bases de données spectroscopiques

Disposant de ces bases de données, on obtient le coefficient d'absorption par la formule (2.38). Imaginons maintenant qu'on effectue un calcul de transfert radiatif, par la MMC et en utilisant un ACN comme présenté précédemment dans ce chapitre. Comme le coefficient d'absorption k_a est évalué à chaque point de collision testé, son calcul répété va peser fortement sur la dépense computationnelle totale. Or, par exemple, la base de données CDSD-4000 recense 600 millions de transitions pour le seul CO₂ ! Il n'est donc pas raisonnable d'évaluer k_a , à chaque demande, par la formule (2.38).

Parmi les travaux traitant du calcul du transfert radiatif dans les milieux gazeux (par méthodes de Monte-Carlo ou autres), 2 approches ont principalement été retenues jusqu'à aujourd'hui :

- Pré-enregistrer le spectre d'absorption des différents constituants du mélange. Pendant le calcul radiatif proprement dit, on n'aura plus qu'à aller chercher les valeurs des σ_m depuis une mémoire informatique. C'est la base des calculs qualifiés de *raie par raie*. Le calcul reste globalement gourmand en ressources informatiques, car les spectres sont relativement lourds en mémoire et surtout longs à produire. Cette approche est qui plus est mise en difficulté dès que la température et/ou la pression varient continûment. Dans ce cas, on ne peut pas pré-enregistrer les spectres des σ_m pour toutes les conditions thermodynamiques présentes, mais seulement pour quelques-unes ; pour les autres conditions possibles, on devra procéder par interpolation. Dans ce cas, la maîtrise de la précision devient une tâche ardue.
- Opter pour une gestion statistique du coefficient d'absorption. On considère alors que sur différentes bandes spectrales, le coefficient d'absorption aura une certaine répartition statistique si on choisit la fréquence de manière uniforme. Dans l'expression des grandeurs radiatives, l'intégration spectrale est ainsi remplacée par la combinaison d'une intégration spectrale partielle et d'une intégration sur la distribution du coefficient d'absorption. C'est la base de la méthode des *k-distribution*, ou des modèles globaux comme WSGG (*Weighted Sum of Gray Gases*) [117].

La présence d'hétérogénéités dans les conditions thermodynamiques est aussi

un handicap pour ces méthodes. Il n'est en effet pas systématique que les distributions des σ_m en 2 points de pression et température différentes soient strictement corrélées, quand la fréquence varie.

2.5.4 Estimation de l'extinction dans un ACN

M. Galtier propose dans son manuscrit de thèse une méthode originale pour gérer la complexité des spectres d'absorption des gaz, dans un calcul de transfert radiatif. Son coût calculatoire est très faible comparativement aux calculs raie par raie, et elle n'introduit aucune approximation.

Le principe en est basé sur l'équation intégrale (2.9), mais au lieu d'en déduire l'équation statistique (2.10) ou (2.16) on déduit l'expression plus générale :

$$f(s) = E \left(H(S' < s_m) \left(\frac{\widetilde{K}_a(S')}{\hat{k}_a(S')} f_{eq}(S') + \frac{\hat{k}_a(S') - \widetilde{K}_a(S')}{\hat{k}_a(S')} \widetilde{F}(S') \right) + H(S' \geq s_m) \widetilde{F}(s_m) \right) \quad (2.39)$$

où :

- S' est une VA à valeurs dans $[s; +\infty[$ et de fonction de répartition $F_{S'}(s') = 1 - \exp\left(-\int_s^{s'} ds'' \hat{k}_a(s'')\right)$, comme en (2.16),
- $\widetilde{F}(S')$ est un estimateur de $f(S')$, comme en (2.16),
- $\widetilde{K}_a(S')$ est un estimateur de $k_a(S')$,
- sachant S' , les VA $\widetilde{F}(S')$ et $\widetilde{K}_a(S')$ sont indépendantes.

Cela signifie que pour pouvoir estimer $f(s)$, il n'est pas nécessaire d'être capable de calculer k_a (ou k_d) en tout point, mais seulement de l'estimer.

La figure 2.15 propose une illustration d'un algorithme estimant $f(s)$ à l'aide de l'expression (2.39), et utilisant une roulette russe seule pour le choix du type des événements de collisions.

Remarque. On franchit ici une étape supplémentaire par rapport à la démarche initiale des Algorithmes à Collisions Nulles. Initialement, il s'agit d'être capable d'estimer $f(s)$ sans connaître exactement l'intégrale de k_a le long d'un chemin — l'épaisseur optique d'absorption —, mais seulement des valeurs ponctuelles de k_a ; ce qui aboutit à utiliser des estimations de l'épaisseur optique d'absorption. Désormais, les valeurs ponctuelles de k_a ne sont elles-mêmes qu'estimées; cette logique d'estimation de l'épaisseur optique est ainsi poussée à son paroxysme.

●●●● Cette capacité à utiliser un estimateur du coefficient d'absorption est une aubaine : on peut en effet en construire un facilement à l'aide de l'équation (2.38).

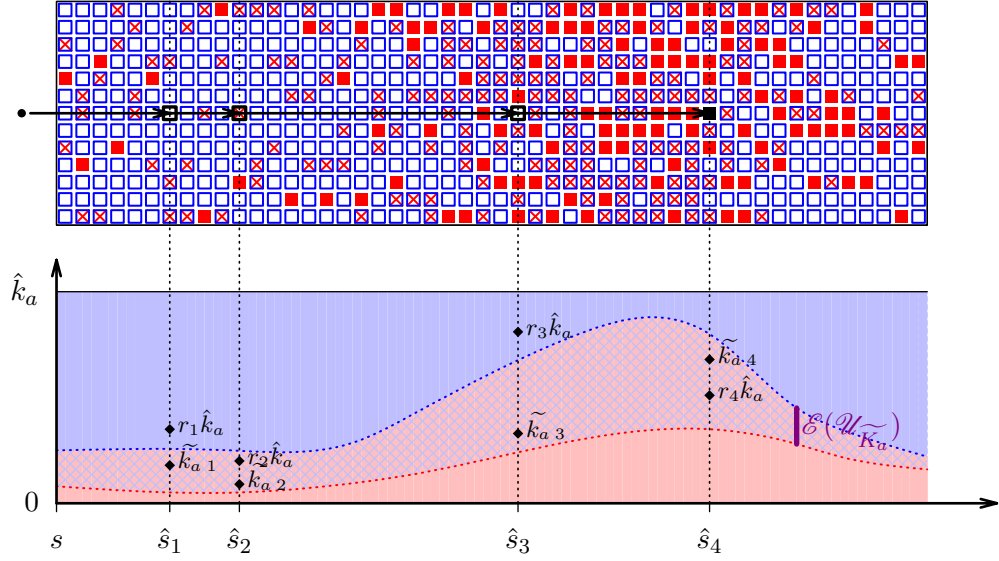


FIGURE 2.15 – Illustration, inspirée de la figure 2.1, du principe de fonctionnement d'un algorithme échantillonnant des libres parcours avant absorption, utilisant un estimateur \widehat{K}_a du coefficient d'absorption k_a .

Il y a une densité constante de collisionneurs, absorbants (carrés rouges) ou fictifs (carrés vides bleus). Cependant, certains collisionneurs sont incertains (carrés bleus avec croix rouge) : leur statut réel ou fictif n'est déterminé que quand une collision a lieu à leur emplacement. Ensuite, le type de collision est choisi selon les proportions de collisionneurs réels et fictifs qu'on a trouvées.

Il suffit de poser des probabilités de choix de molécules P_m et des probabilités conditionnelles de choix de transitions $P'_{j|m}$ et alors :

$$\begin{aligned} k_a &= \sum_{m=1}^{N_m} P_m \frac{\eta_m}{P_m} \sum_{j=1}^{N_t(m)} P'_{j|m} \frac{\sigma_{m,j}}{P'_{j|m}} \\ &= E\left(\frac{\eta_M \sigma_{M,J}}{P_M P'_{J|M}}\right) \end{aligned} \quad (2.40)$$

avec les variables aléatoires M et J vérifiant $\Pr(M = m) = P_m$ et $\Pr(J = j \mid M = m) = P'_{j|m}$. On peut donc choisir pour k_a l'estimateur

$$\widetilde{K}_a = \frac{\eta_M \sigma_{M,J}}{P_M P'_{J|M}} \quad (2.41)$$

Une particularité forte de la méthode proposée est, que la gestion de la complexité spectrale de l'absorption se fait simultanément à la gestion des hétérogénéités des conditions thermodynamiques. Ceci est une rupture par rapport aux précédentes démarches de calcul numérique existantes en transfert radiatif.

Dans la pratique, toute la difficulté se situe dans le choix des probabilités P_m et $P'_{j|m}$, devant assurer une variance suffisamment faible pour \widetilde{K}_a .

On sait que si \hat{k}_a majore \widetilde{K}_a , on est assuré d'avoir l'estimation de $f(s)$ dans l'intervalle $[\inf \mathcal{F}_t; \sup \mathcal{F}_t]$, où $\mathcal{F}_t = \{f_{eq}(s), s \in [s_{init}; s_m]\} \cup \mathcal{U}_{\widetilde{F}(s_m)}$; cette contrainte sur l'estimateur lui assure une faible variance. Donc si \widetilde{K}_a est susceptible de prendre des valeurs très dispersées, une solution pour que l'algorithme converge bien est de prendre une valeur élevée pour \hat{k}_a ; mais ce n'est pas une solution parfaite, car le suivi de photon devient très laborieux parce qu'encombré de nombreuses collisions nulles.

Il va donc falloir trouver un compromis entre 2 attitudes :

- Utiliser un choix rapide pour M et J , afin que l'échantillonnage de \widetilde{K}_a soit beaucoup plus rapide que le calcul complet de k_a par la formule (2.38).
- Choisir minutieusement les P_m et $P'_{j|m}$, de sorte à diminuer la variance de \widetilde{K}_a ou mieux encore à le majorer par une valeur proche de k_a , pour éviter une explosion de la variance de l'estimateur de $f(s)$ ou de la proportion de collisions nulles.

On peut aussi envisager d'utiliser un estimateur \widetilde{K}_a de k_a de définition différente à l'expression (2.41); par exemple on pourrait utiliser plusieurs transitions à la fois.

Cette problématique occupe une partie du manuscrit de thèse de M. GALTIER. Je renvoie à cet écrit pour les résultats obtenus pour l'instant.

2.6 Les algorithmes à collisions nulles dans la suite de ce manuscrit

●●●● Nous venons de voir que les algorithmes à collisions nulles permettent de lever les difficultés qui émergent, lors d'un calcul du transfert radiatif par MMC, de la complexité de l'expression de l'épaisseur optique d'extinction le long d'un rayon. Cette complexité peut émerger :

- de la variabilité du coefficient d'extinction le long du rayon,
- de l'expression du coefficient d'extinction lui-même (une somme des contributions d'un grand nombre de transitions dans le cas de M. GALTIER).

Les ACNs nous permettent, dans les 2 cas, de gérer statistiquement la complexité de l'épaisseur optique. Pour cela, leur principe est de faire virtuellement disparaître la non-linéarité de la loi de Beer, en la transformant en récursivité. Dès lors, l'épaisseur optique peut être échantillonnée comme les autres grandeurs, au cours d'un algorithme de Monte-Carlo ordinaire.

En cinétique des gaz, la complexité de l'épaisseur d'extinction n'aura qu'une seule source : l'extinction dépend de la solution elle-même. Cette complexité-là synthétise et dépasse à elle-seule celles que nous avons rencontrées dans ce chapitre.

Nous allons y remédier en augmentant encore la récursivité de nos algorithmes. En fait, on appliquera strictement la démarche exposée dans cette section : nous utiliserons un estimateur du coefficient d'extinction. Mais cette fois-ci, il sera construit à partir de la fonction de distribution elle-même ! Les chapitres 3 et 4 montreront la validité d'une telle approche. Ils montreront aussi comment gérer la source dans le terme collisionnel de Boltzmann ; car pour l'instant, nous n'avons parlé que de l'extinction.

Chapitre 3

Estimation d'une fonction non-linéaire d'une espérance statistique

Les utilisateurs de la méthode de Monte-Carlo admettent couramment que la force principale de la méthode est sa capacité à combiner des espaces de grande dimension. Mais ils admettent également que cela devient impossible dès que la combinaison passe par une fonction non linéaire. Il n'y a aucun problème à calculer la moyenne sur un espace 1 d'une fonction linéaire d'une moyenne sur un espace 2 ; mais seulement dans le cas linéaire.

Cette apparente incapacité peut être contournée, pour les fonctions analytiques. Le point central est d'estimer une puissance n entière d'une espérance par le produit de n réalisations indépendantes. Nous reprendrons cette idée en cinétique des gaz, en particulier pour tenir compte du terme source de l'équation de Boltzmann, quadratique en la fonction de distribution.

3.1 Introduction

●●●● Au chapitre précédent, nous avons présenté les algorithmes à collisions nulles (ACNs). Historiquement, ceux-ci ont été conçus comme une astuce algorithmique, pour pouvoir échantillonner des longueurs de vol libre dans un milieu où le coefficient d'extinction est variable. Mais d'un point de vue théorique, on peut les interpréter comme une façon de gérer la non-linéarité de l'extinction vis-à-vis de son coefficient. L'exponentielle d'intégrale, représentant l'extinction dans l'équation (1.43), est démontée en un empilement infini d'intégrales que l'on sait gérer dans un calcul de type Monte-Carlo.

Mais au stade actuel nous avons présenté cette approche d'une manière très intuitive, en faisant appel à l'image de collisionneurs fictifs. En conséquence, le

seul type de non-linéarité que ces ACNs peuvent gérer est le terme exponentiel d'extinction, caractéristique de la loi de Beer.

Ce chapitre propose une approche plus générale pour calculer par MMC les expressions se présentant sous la forme : « intégrale d'une fonction non-linéaire d'une intégrale ». Ses fondements théoriques ne sont rien d'autre que les développements en série entière et l'indépendance statistique. Au passage, cette approche permet aussi d'expliquer et de formaliser les ACNs sous un angle radicalement différent. Toutefois, comme dans cette thèse, à chaque fois que nous en aurons besoin, nous utiliserons les ACNs tels que présentés au chapitre précédent, nous ne détaillons pas ce travail de redite des ACNs dans le corps du manuscrit ; nous le reportons en annexe B. Par contre, nous aurons besoin de la généralisation pour traiter la non-linéarité du terme collisionnel.

En fait, dans les chapitres suivants de ce manuscrit, nous diviserons la non-linéarité du terme collisionnel de Boltzmann en deux :

- l'extinction, dépendante de la fonction de distribution, sera gérée par un ACN identique à ceux du chapitre précédent,
- le terme source sera géré grâce à l'approche généralisée présentée dans le présent chapitre.

Ce chapitre va plus loin que ce qui est strictement nécessaire pour traiter le terme source de l'équation de Boltzmann, qui est seulement quadratique en f . Mais encore une fois, l'équation de Boltzmann d'origine, celle de la cinétique des gaz, est pour nous une représentante de toutes les équations de Boltzmann non-linéaires. Nous avons donc donné à ce chapitre une portée plus générale que le seul objectif de traiter le terme source quadratique de Boltzmann.

●●●● La section 3.1 explique le type de difficultés numériques que l'on rencontre avec la MMC lorsque deux espaces sont combinés par l'intermédiaire d'une fonction non-linéaire. On commencera par un exemple concret, issu du génie des procédés photobiochimiques. Puis on définira la question posée en toute généralité avant d'entrer dans la section 3.2 qui détaillera la technique permettant de surmonter les difficultés annoncées.

Enfin, les 2 dernières sections de ce chapitre exposeront des exemples où la technique a été appliquée avec succès. Ils sont tirés des travaux de thèse de Jérémie DAUCHET [29] et d'Olivier FARGES [44].

3.1.1 Grandeurs dérivées de la fonction de distribution : le cas des photobioréacteurs

●●●● Dans les chapitres précédents, nous avons présenté plusieurs grandeurs d'intérêt macroscopiques pour le physicien ou l'ingénieur confronté à un problème de transfert radiatif ou de cinétique des gaz :

- la masse ou l'énergie totale contenue dans un volume donné,

- le flux lumineux absorbé par une surface,
- la pression sur une surface,
- le flux de chaleur échangé entre une surface et un gaz...

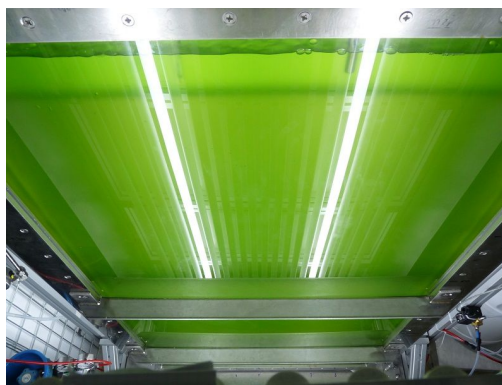
Comme ces grandeurs ne sont que des intégrales de la fonction de distribution f sur une partie de l'espace des phases, leur calcul, par une méthode de Monte-Carlo, ne pose aucune difficulté supplémentaire par rapport au calcul de f elle-même. Nous en avons discuté au chapitre 1.

Les choses se corsent sensiblement lorsque l'on veut calculer des grandeurs macroscopiques plus complexes, dépendant non-linéairement de f . Un problème concret de ce type est celui rencontré par Jérémie DAUCHET pendant sa thèse [29], quand il a voulu calculer la productivité totale d'un photobioréacteur. Ce cas est exposé partiellement ici, et ensuite dans la section 3.4.

●●●● Grossièrement résumé, le photobioréacteur étudié est une enceinte cylindrique fermée dans laquelle on fait se multiplier des micro-organismes photosynthétiques, dans un milieu aqueux mélangé en permanence. De ce milieu est constamment extrait de la biomasse, pendant qu'il est suffisamment alimenté en eau, en CO_2 , en autres nutriments, et en lumière pour que cette biomasse se renouvelle au plus vite. Aujourd'hui, l'homogénéisation et l'optimisation des conditions thermochimiques dans le réacteur (température, salinité, concentration en nutriments, concentration en biomasse) pour maximiser la croissance des micro-organismes n'est plus une difficulté : les technologies nécessaires sont totalement maîtrisées. Il n'en est hélas pas de même pour son éclairage ; c'est l'objet de la thèse de J. DAUCHET.



(a) Photobioréacteur « fait main »
Source : post du blog
www.inventgeek.com



(b) Photobioréacteur plan,
pour essais en laboratoire
Source : GEPEA¹

FIGURE 3.1 – Exemples de photobioréacteurs en milieu fermé

1. laboratoire GÉnie des Procédés Environnement et Agroalimentaire, UMR 6144

La lumière est la source primaire et unique d'énergie des micro-organismes dans le réacteur. La géométrie de ce dernier est prévue pour que, sachant les

propriétés optiques du milieu de croissance, l'absorption de la lumière fournie et surtout la production subséquente de biomasse soient maximales. Ceci n'est pas une tâche évidente, car le rendement de la photosynthèse est variable, et dépend en particulier de l'éclairage local ; il n'y a donc pas proportionnalité entre l'éclairage absorbé et la production finale. En résumé, la vraie grandeur d'intérêt est bien la production totale de biomasse par le réacteur, qui n'est que l'intégrale sur le volume du réacteur de la vitesse locale de production de biomasse, mais cette dernière est une fonction non-linéaire de l'irradiance locale.

Cela s'écrit, avec Q_x en kg/s la productivité totale en biomasse (matière sèche organique) du réacteur :

$$Q_x = \int_{\mathcal{V}} v_x(\vec{r}) d\vec{r} \quad (3.1a)$$

où v_x est la vitesse volumique locale de production de biomasse (en $\text{kg.m}^{-3}.\text{s}^{-1}$), et \mathcal{V} le volume réactionnel. Comme affirmé précédemment, $v_x(\vec{r})$ est fonction :

- du type de micro-organismes présents (fixé)
- des conditions thermochimiques (uniformes)
- de l'irradiance. Plus exactement, $v_x(\vec{r})$ est fonction de la vitesse volumique locale d'absorption des photons, parfois notée $A(\vec{r})$ ². Ces photons doivent appartenir à une certaine fenêtre spectrale, appelée PAR — pour *Photosynthetically Active Radiation*. A s'exprime ainsi comme :

$$A(\vec{r}) = \int_{\text{PAR}} d\nu k_a(\vec{r}; \nu) \int_{4\pi} d\vec{u} f_\nu(\vec{r}; \vec{u}; \nu) \quad (3.1b)$$

On note alors : $v_x(\vec{r}) = f_{v_x}(A(\vec{r}))$, où f_{v_x} est appelée la *loi de couplage cinétique*.

Finalement :

$$Q_x = \int_{\mathcal{V}} d\vec{r} f_{v_x}(A(\vec{r})) \quad (3.1c)$$

$$= \int_{\mathcal{V}} d\vec{r} f_{v_x} \left(\int_{\text{PAR}} d\nu k_a(\vec{r}; \nu) \int_{4\pi} d\vec{u} f_\nu(\vec{r}; \vec{u}; \nu) \right) \quad (3.1d)$$

où f_{v_x} est une fonction non-linéaire.

●●●● La détermination des conditions radiatives dans un photobioréacteur revêt une importance particulière, dès qu'on veut améliorer son rendement.

Un éclairage inadéquat peut en effet entailler largement le rendement d'un procédé photobioréactif. Si l'irradiance locale est trop faible, les micro-organismes ne génèrent pas plus de biomasse par photosynthèse qu'ils n'en consomment par respiration dans le même temps — ce 2^e mécanisme est toujours présent —, et donc ne produisent rien. Si l'irradiance est très élevée, les fonctions photosynthétiques

2. J. DAUCHET utilise dans son manuscrit une notation différente : il note \mathcal{A} pour la vitesse *spécifique* locale d'absorption des photons, ce qui correspond ici à A/C_x où C_x est la concentration locale massique en biomasse.

saturent, produisant certes de la biomasse à un rythme maximal, mais perdant une grande partie de l'énergie lumineuse reçue en chaleur inutile. Il y a donc bien un optimum à chercher dans l'irradiance locale.

Un des prototypes étudiés par J. DAUCHET, nommé le DiCoFluV, vise à une optimisation du rendement total³, en particulier par la recherche d'un bon éclairage. Technologiquement parlant, il est conçu comme une enceinte cylindrique où la lumière est amenée par des fibres optiques réparties dans tout le volume intérieur (voir figure 3.7 page 149). Cette disposition vise d'une part à utiliser efficacement le volume disponible : la surface d'entrée de la lumière est maximisée, pour que l'on puisse travailler avec une faible densité de flux lumineux entrant. Cette faible densité de flux est indispensable, car l'optimum de rendement photosynthétique vis-à-vis de l'éclairage est situé généralement à une irradiance faible, du moins comparativement à l'éclairage du jour. D'autre part, cette disposition des sources de lumière vise à rendre l'irradiance la plus uniforme possible, près de l'optimum.

L'uniformité de l'irradiance vise à ce que le rendement photosynthétique soit le meilleur en tout point. Mais elle est contradictoire avec la nécessité d'absorber la lumière fournie. L'absorption se fait dans le volume de la suspension de micro-organismes, et obligatoirement les points proches des entrées lumineuses seront plus éclairés que ceux qui en sont éloignés. Il est bien sûr faisable d'assurer une irradiance presque uniforme dans tout le volume : c'est le cas quand il y a peu d'absorption dans le milieu réactionnel ; mais cette situation serait incompatible avec la volonté de produire le plus possible avec la ressource lumineuse disponible.

Bref, la conception d'un photobioréacteur efficace, du point de vue de l'utilisation de sa ressource lumineuse, est une tâche difficile ; et la difficulté porte essentiellement sur la géométrie du réacteur et le placement des sources lumineuses, par leur influence sur le champ d'irradiance. On ne saurait se dispenser du calcul de la productivité totale du réacteur. Celle-ci s'exprime de façon générale avec la formule (3.1d).

Au vu de la complexité du problème de transfert radiatif posé (diffusion multiple, géométrie complexe, dépendance spectrale importante des propriétés optiques, diffusion anisotrope,...), il semble raisonnable d'évaluer cette productivité par la méthode de Monte-Carlo. Mais la présence de la loi de couplage cinétique, non-linéaire, entre 2 intégrations, pose problème.

3.1.2 Problématique

Dans la suite de ce chapitre, nous opterons pour des notations plus génériques, non-spécifiques aux photobioréacteurs.

●●●● Le problème global posé est celui de l'évaluation l'expression intégrale :

3. en opposition par exemple avec une optimisation du coût, ou une optimisation de l'encombrement

$$S_f = \int_{\mathcal{U}_{\vec{X}}} d\vec{x} f\left(\vec{x}; \int_{\mathcal{U}_{\vec{Y}}(\vec{x})} d\vec{y} g(\vec{x}; \vec{y})\right) \quad (3.2)$$

où $\mathcal{U}_{\vec{X}}$ et $\mathcal{U}_{\vec{Y}}(\vec{x})$ sont des parties d'espaces euclidiens, et f et g sont 2 fonctions quelconques à valeurs réelles (aucun rapport ici entre f et g et des fonctions de distribution).

En introduisant \vec{X} et \vec{Y} 2 variables aléatoires, à valeurs dans $\mathcal{U}_{\vec{X}}$ et $\mathcal{U}_{\vec{Y}}(\vec{X})$ et de densités de probabilité $p_{\vec{X}}$ et $p_{\vec{Y}|\vec{X}}$, l'expression recherchée peut-être ré-exprimée sous forme statistique :

$$\begin{aligned} S_f &= \int_{\mathcal{U}_{\vec{X}}} d\vec{x} p_{\vec{X}}(\vec{x}) w_f\left(\vec{x}; \left(\int_{\mathcal{U}_{\vec{Y}}(\vec{x})} d\vec{y} p_{\vec{Y}|\vec{X}}(\vec{y}|\vec{x}) w_g(\vec{x}; \vec{y})\right)\right) \\ &= E\left(w_f\left(\vec{X}; E_{|\vec{X}}\left(w_g\left(\vec{X}; \vec{Y}\right)\right)\right)\right) \end{aligned} \quad (3.3)$$

où $w_f(\vec{x}; z) = \frac{f(\vec{x}; z)}{p_{\vec{X}}(\vec{x})}$ et $w_g(\vec{x}; \vec{y}) = \frac{g(\vec{x}; \vec{y})}{p_{\vec{Y}|\vec{X}}(\vec{y}|\vec{x})}$.

Remarquons d'abord que si f est une fonction affine⁴ envers sa 2^e variable, le problème se simplifie. En effet, cela signifie qu'il existe 2 fonctions λ_0 et λ_1 telles que $f(\vec{x}; z) = \lambda_0(\vec{x}) + \lambda_1(\vec{x})z$, auquel cas on peut écrire :

$$\begin{aligned} S_f &= \int_{\mathcal{U}_{\vec{X}}} d\vec{x} \left(\lambda_0(\vec{x}) + \lambda_1(\vec{x}) \int_{\mathcal{U}_{\vec{Y}}(\vec{x})} d\vec{y} g(\vec{x}; \vec{y}) \right) \\ &= \int_{\mathcal{U}_{\vec{X}}} d\vec{x} \left(\lambda_0(\vec{x}) + \lambda_1(\vec{x}) \int_{\mathcal{U}_{\vec{Y}}(\vec{x})} p_{\vec{Y}|\vec{X}}(\vec{y}|\vec{x}) d\vec{y} w_g(\vec{x}; \vec{y}) \right) \\ &= \int_{\mathcal{U}_{\vec{X}}} d\vec{x} \int_{\mathcal{U}_{\vec{Y}}(\vec{x})} p_{\vec{Y}|\vec{X}}(\vec{y}|\vec{x}) d\vec{y} \left(\lambda_0(\vec{x}) + \lambda_1(\vec{x}) w_g(\vec{x}; \vec{y}) \right) \\ &= \int_{\mathcal{U}_{\vec{X}}} d\vec{x} \int_{\mathcal{U}_{\vec{Y}}(\vec{x})} p_{\vec{Y}|\vec{X}}(\vec{y}|\vec{x}) d\vec{y} f\left(\vec{x}; w_g(\vec{x}; \vec{y})\right) \\ &= \int_{\mathcal{U}_{\vec{X}}} p_{\vec{X}}(\vec{x}) d\vec{x} \int_{\mathcal{U}_{\vec{Y}}(\vec{x})} p_{\vec{Y}|\vec{X}}(\vec{y}|\vec{x}) d\vec{y} w_f\left(\vec{x}; w_g(\vec{x}; \vec{y})\right) \end{aligned} \quad (3.4a)$$

$$= E\left(w_f\left(\vec{X}; w_g\left(\vec{X}; \vec{Y}\right)\right)\right) \quad (3.4b)$$

On retrouve ici la formulation intégrale d'un algorithme de Monte-Carlo tout-à-fait standard.

Les choses se compliquent quand f n'est plus affine, car elle ne commute alors pas avec l'intégration sur \vec{y} comme montré plus haut. Il n'est alors *a priori* plus possible d'exprimer S_f comme l'espérance d'un processus aléatoire.

●●●● La question pratique reste alors : comment évaluer S_f ? Notons pour la suite $S_g(\vec{x}) = \int_{\mathcal{U}_{\vec{Y}}(\vec{x})} d\vec{y} g(\vec{x}; \vec{y}) = E_{|\vec{X}=\vec{x}}\left(w_g(\vec{x}; \vec{Y})\right)$, de sorte que $S_f = E\left(w_f\left(\vec{X}; S_g(\vec{X})\right)\right)$. En l'état, la formulation algorithmique qui se dégage est celle de l'algorithme 3.1.

Ce qui laisse plusieurs interrogations :

4. On dit très souvent « fonction linéaire » dans ce cas, même si cela ne correspond pas tout-à-fait à la définition de la linéarité.

<p>Résultat : Une estimation $(\overline{w}_f; s^2(\overline{w}_f))$ de S_f</p> <pre> 1 $\Sigma_f \leftarrow 0$; 2 $\Sigma_{f2} \leftarrow 0$; 3 pour $i \leftarrow 1$ à N_f faire 4 échantillonner \vec{X} : on obtient \vec{x}_i ; 5 $\Sigma_g \leftarrow 0$; 6 pour $j \leftarrow 1$ à N_g faire 7 échantillonner \vec{Y} : on obtient $y_{j,i}$; 8 $w_{g\ j,i} \leftarrow w_g(\vec{x}_i; \vec{y}_{j,i})$; 9 $\Sigma_g \leftarrow \Sigma_g + w_{g\ j,i}$; 10 $\overline{w}_{g\ i} \leftarrow \Sigma_g / N_g$; 11 $w_{f\ i} \leftarrow w_f(\vec{x}_i; \overline{w}_{g\ i})$; 12 $\Sigma_f \leftarrow \Sigma_f + w_{f\ i}$; 13 $\Sigma_{f2} \leftarrow \Sigma_{f2} + w_{f\ i}^2$; 14 $\overline{w}_f \leftarrow \Sigma_f / N_f$; 15 $s^2(\overline{w}_f) \leftarrow (\Sigma_{f2} / N_f - \overline{w}_f^2) / (N_f - 1)$; </pre>

Algorithme 3.1 : Algorithme naïf et biaisé pour le calcul de S_f

1. Comment évaluer l'erreur de calcul ?
2. Comment évoluera le temps de calcul ?

Les $\overline{w}_{g\ i}$ sont des estimateurs sans biais des $S_g(\vec{x}_i)$, dont on peut estimer la variance. Par contre, si f est non-linéaire en sa 2^e variable, les $w_f(\vec{x}_i; \overline{w}_{g\ i})$ sont des estimateurs en général biaisés de $w_f(\vec{x}_i; S_g(\vec{x}_i))$. Pour s'en convaincre, on pourra considérer le cas où $w_f(\vec{x}_i; \bullet)$ est strictement convexe (comme $z \mapsto z^2$, ou l'exponentielle) (équivalent à « $f(\vec{x}_i; \bullet)$ est strictement convexe ») ; la figure 3.2 en donne une illustration. Dans ce cas, en posant Z une variable aléatoire à valeurs réelles et de variance non-nulle, l'inégalité de Jensen affirme que $E(w_f(\vec{x}_i; Z)) > w_f(\vec{x}_i; E(Z))$. La différence augmente quand la variance de Z augmente et quand la convexité de $w_f(\vec{x}_i; \bullet)$ se creuse ; on ne connaît cependant pas de formule simple et systématique pour la calculer.

Tant que $w_f(\vec{x}_i; \bullet)$ est continue en $S_g(\vec{x}_i)$, $w_f(\vec{x}_i; \overline{w}_{g\ i})$ reste néanmoins un estimateur convergent de $w_f(\vec{x}_i; S_g(\vec{x}_i))$, et donc le résultat de l'algorithme 3.1 tend bien vers S_f quand N_f et N_g tendent vers l'infini. Au final :

- L'algorithme 3.1 donne une estimation $s^2(\overline{w}_f)$ non biaisée de la variance de son résultat \overline{w}_f , comme tout algorithme de Monte-Carlo respectant la constitution présentée au chapitre 1.3.1.
- Cet écart-type ne correspond cependant plus à une barre d'erreur sur la grandeur calculée, car au bruit statistique se rajoute un biais non-quantifiable.
- Ce biais tend vers 0 quand N_g tend vers l'infini ; on ne sait toutefois pas comment.

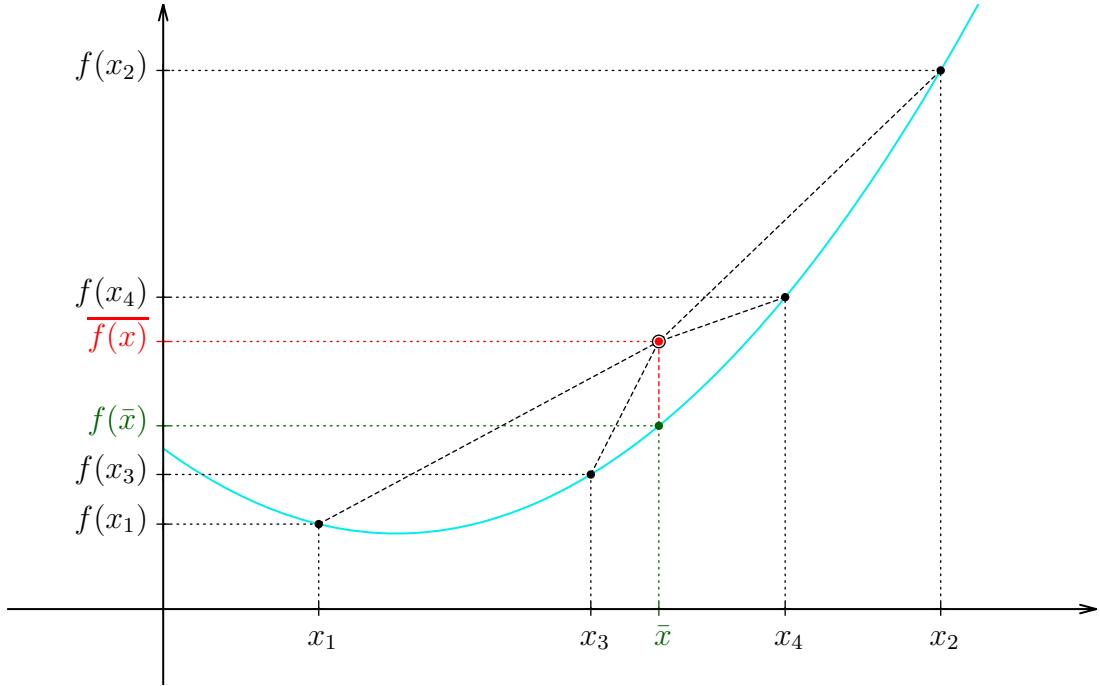


FIGURE 3.2 – Illustration de l'inégalité de Jensen.

Soit une fonction strictement convexe f . Le barycentre d'une famille de points du graphe de f distincts et strictement positivement pondérés est strictement au-dessus de ce graphe. La moyenne des images par f d'une famille de plusieurs réels est donc strictement supérieure à l'image par f de la moyenne de cette famille.

D'où la non-commutativité de f avec la prise de moyenne, ou l'espérance statistique.

— La complexité de l'algorithme n'est plus en N_f , mais en $N_f \times N_g$.

●●●● Pour le numéricien utilisateur de la MMC, la combinaison de ces 2 derniers points est un gros défaut. À fins d'illustrations, revenons au cas du photobioréacteur pré-cité.

Dans les travaux de J. DAUCHET sur le prototype DiCoFluV, la méthode de calcul choisie pour $A(\vec{r})$ était bien la méthode de Monte-Carlo. Elle permet de prendre en compte la pleine complexité du problème radiatif (diffusion multiple et anisotrope, géométrie complexe, propriétés spectrales) pour un coût computationnel réduit, et sans nécessiter de développement ou de validation spécifique au cas d'étude. Comme affirmé au chapitre 1.3.1, cette capacité à rester simple et moyennement coûteuse dans des situations très complexes est son principal intérêt.

Là, le calcul de A en un point \vec{r} donné nécessitait environ 10^5 réalisations (consistant chacune en l'échantillonnage d'un chemin optique, comme au chapitre 1.5) pour atteindre une précision relative de 1 %. En utilisant des outils informatiques issus de la communauté de la synthèse d'image, pour paralléliser les réalisations

et surtout accélérer le calcul d'intersections rayon/géométrie — seule véritable difficulté numérique présente dans le cas d'étude — ceci nécessitait environ 5 secondes sur un ordinateur de bureau usuel. Soit : un coût calculatoire parfaitement acceptable dans le contexte du génie des procédés.

Seulement, pouvoir calculer A , et ainsi la productivité volumique v_x , en tout point du réacteur est certes très intéressant quand on le conçoit, mais ce n'est pas la grandeur d'intérêt finale : c'est la productivité totale Q_x . Pour l'obtenir, il faut intégrer v_x sur tout le volume réactionnel.

Dans le cas considéré, en prenant une démarche calquée sur l'algorithme 3.1, l'expérience a montré qu'il fallait :

- 10^5 positions \vec{r} de calcul de $v_x(\vec{r})$ pour estimer Q_x avec un écart-type de 1 %. Si l'étape d'intégration volumique de $v_x(\vec{r})$ est réalisée avec une méthode déterministe, on reste dans le même besoin de calcul, pour obtenir une précision de 1 %.
- estimer $A(\vec{r})$, afin de calculer $v_x(\vec{r})$, avec un écart-type de 1 %, et donc avec 10^5 échantillonnage de chemins optiques.

Cela nécessite au total 10^{10} échantillonnages de chemins optiques soit plusieurs semaines de calcul !

●●●● Confronté à cet obstacle, J. DAUCHET a, en s'inspirant de la formulation intégrale des Algorithmes à Collisions Nulles, mis au point une astuce numérique, basée sur les développements en série entière et les propriétés de l'indépendance statistique. Pour son cas d'étude, elle écrase complètement la difficulté explicitée au-dessus.

La technique mathématique et numérique dont il est question est bien sûr présentée dans le manuscrit de thèse de Jérémie DAUCHET [29]. Elle est reprise dans le manuscrit de thèse d'Olivier FARGES [44], et présentée dans la section suivante.

3.2 Présentation de la technique utilisée

3.2.1 Cas d'un monôme ou d'un polynôme

La base de la technique dont il est question dans ce chapitre est la capacité à calculer un monôme de l'espérance d'une variable aléatoire (soit $E(Z)^i$). On rappellera que l'on ne peut pas la calculer comme l'espérance du même monôme de la variable aléatoire (soit $E(Z^i)$). Pour s'en convaincre, il suffit de considérer la formule de la variance : $\text{Var}(X) = E(X^2) - E(X)^2$.

Pour calculer la i^{e} puissance de l'espérance d'une variable aléatoire réelle Z , on utilise une famille $(Z_j)_{j \in \llbracket 1; i \rrbracket}$ de variables indépendantes identiquement distribuées

à Z (on parle alors de variables *parentes* à Z). En effet :

$$(E(Z))^i = \prod_{j=1}^i E(Z) \quad \text{par définition} \quad (3.5a)$$

$$= \prod_{j=1}^i E(Z_j) \quad \text{si les } Z_j \text{ sont identiquement distribuées à } Z \quad (3.5b)$$

$$= E\left(\prod_{j=1}^i Z_j\right) \quad \text{si les } Z_j \text{ sont indépendantes} \quad (3.5c)$$

Le nombre d'échantillonnages de Z nécessaires est alors multiplié par i . i peut certes être grand : mais il est à comparer avec le N_g de l'algorithme 3.1, qui devrait théoriquement être infini pour qu'aucun biais de calcul ne subsiste.

●●●● Une technique similaire peut être utilisée pour calculer un polynôme de degré k de $E(Z)$. Soit $h : z \mapsto \sum_{i=0}^k a_i z^i$ une fonction polynomiale, alors :

$$h(E(Z)) = \sum_{i=0}^k a_i E(Z)^i = E\left(\sum_{i=0}^k a_i \prod_{j=1}^i Z_j\right) \quad \text{avec les } Z_j \text{ parentes à } Z \quad (3.6)$$

Là-aussi, le nombre d'échantillonnages de Z nécessaires est multiplié par le degré du polynôme.

Astuce : Dans la formule 3.6, les Z_j sont parfaitement interchangeables. Si σ est une permutation donnée des entiers, on a donc :

$$\sum_{i=0}^k a_i E(Z)^i = E\left(\sum_{i=0}^k a_i \prod_{j=1}^i Z_{\sigma(j)}\right) \quad \text{avec les } Z_j \text{ parentes à } Z \quad (3.7)$$

Plus largement, si l'on dispose d'un ensemble fini \mathfrak{R} de couples de :

1. coefficients réels ς
2. permutations σ des entiers,

tel que $\sum_{(\varsigma;\sigma) \in \mathfrak{R}} \varsigma = 1$, alors on peut écrire :

$$\sum_{i=0}^k a_i E(Z)^i = E\left(\sum_{(\varsigma;\sigma) \in \mathfrak{R}} \varsigma \sum_{i=0}^k a_i \prod_{j=1}^i Z_{\sigma(j)}\right) \quad \text{avec les } Z_j \text{ parentes à } Z \quad (3.8)$$

Remarque. \mathfrak{R} doit être indépendant des Z_j .

La possibilité de permuter les Z_j n'est pas uniquement une curiosité formelle ; elle permet de diminuer la variance de l'estimation de $h(E(Z))$. En effet, comme cette somme comporte un terme en $E(Z)^k$ il faudra un échantillon d'au moins k tirages indépendants de Z pour l'estimer. Cependant, dans la formule (3.6), l'estimation de chaque terme $E(Z)^i$ n'utilise que les i premiers tirages de Z :

l'information utilisable dans les tirages suivants est perdue pour l'estimation du terme $E(Z)^i$.

Ainsi, une fois qu'on a échantillonné Z *au moins*⁵ $k \geq i$ fois, en permutant les tirages on peut calculer différentes estimations d'un terme $E(Z)^i$, toutes non biaisées. En moyennant ces différentes estimations, on récupère donc plus d'information de l'échantillon effectué que si on n'en avait utilisé que i éléments, une seule fois. Une fois qu'on a effectué $k' \geq k$ tirages de Z , le choix pour \mathfrak{R} donnant l'estimateur le plus efficace de $h(E(Z))$ est de prendre toutes les permutations de $\llbracket 1; k' \rrbracket$ (on note usuellement leur ensemble $\mathfrak{S}_{k'}$, et elles sont au nombre de $k'!$) et de les affecter du même coefficient ς :

$$\sum_{i=0}^k a_i E(Z)^i = E\left(\frac{1}{k'!} \sum_{\sigma \in \mathfrak{S}_{k'}} \sum_{i=0}^k a_i \prod_{j=1}^i Z_{\sigma(j)}\right) \quad \text{avec} \quad \begin{cases} \text{les } Z_j \text{ parentes à } Z \\ k' \in \llbracket k; +\infty \rrbracket \end{cases} \quad (3.9)$$

Pour le calcul de cette formulation, un algorithme est présenté dans l'annexe C.1.

Dans la suite de ce manuscrit, on reparlera peu de cette possibilité de permuter les estimations indépendantes de Z pour estimer $h(E(Z))$ avec une précision accrue. Les exemples d'application de la méthode exposée ici, détaillés à la fin de ce chapitre, sont directement repris des travaux de thèse de J. DAUCHET et O. FARGES, qui n'ont de fait pas utilisé de permutations.

De même, dans ce que nous avons mis en place pour la cinétique des gaz, l'utilité de telles techniques ne s'est pas encore fait sentir. Mais cela est lié à ce que le terme collisionnel de Boltzmann est purement quadratique en la fonction de distribution, sans partie linéaire.

Si nous évoquons une possible utilisation de permutations dans cette section, c'est parce que nous pensons que notre lecteur pourrait vouloir :

- utiliser la technique présentée dans ce chapitre, telle quelle, par exemple dans une application de génie des procédés (comme le sont les exemples illustratifs de ce chapitre),
- étendre notre démarche de montage d'un estimateur de la fonction de distribution, que nous avons appliqué à la cinétique des gaz, vers d'autres phénomènes de transport non-linéaire.

Dans ces 2 cas, des permutations des Z_j pourraient être plus utiles que ce qu'elles nous l'ont été durant cette thèse.

3.2.2 Cas des fonctions analytiques

On reprend les notations de la section 3.1.2, avec de plus $Z = w_g(\vec{X}; \vec{Y})$, ce qui permet d'exprimer avec concision $S_f = E(w_f(\vec{X}; E_{|\vec{X}}(Z))) = E(w_f(\vec{X}; S_g(\vec{X})))$.

5. Il est effectivement possible de diminuer le rapport $\text{Var}(\tilde{h}(E(Z)))/\text{Nb total tirages}_Z$, avec \tilde{h} un estimateur de la fonction polynomiale h , en prenant plus de tirages de Z que nécessaire pour calculer ponctuellement de $\tilde{h}(E(Z))$. Voir l'annexe C.2 pour plus de détails.

●●●● On suppose désormais que pour tout $\vec{x} \in \mathcal{U}_{\vec{X}}$, il existe $z_R \in \mathbb{R}$ tel que :

- f soit développable en série entière⁶ vis-à-vis de sa 2^e variable autour de z_R avec un rayon de convergence $R \in \overline{\mathbb{R}^{+*}}$, et
- $S_g(\vec{x})$ appartienne au disque ouvert de convergence $\mathcal{B}(z_R; R)$ du DSE sus-cité.

La développabilité en série entière de $f(\vec{x}; z)$ vis-à-vis de z se transmet immédiatement à $w_f(\vec{x}; z) = \frac{f(\vec{x}; z)}{p_{\vec{X}}(\vec{x})}$, ce qui permet de poser la suite des coefficients $\lambda(\vec{x})$ tels que :

$$\forall z \in \mathcal{B}(z_R; R), w_f(\vec{x}; z) = \sum_{i=0}^{+\infty} \lambda_i(\vec{x})(z - z_R)^i \quad (3.10)$$

On peut alors écrire :

$$w_f(\vec{x}; E_{|\vec{X}=\vec{x}}(Z)) = \sum_{i=0}^{+\infty} \lambda_i(\vec{x})(E_{|\vec{X}=\vec{x}}(Z) - z_R)^i \quad (3.11a)$$

$$= \sum_{i=0}^{+\infty} E_{|\vec{X}=\vec{x}} \left(\lambda_i(\vec{x}) \prod_{j=1}^i (Z_j - z_R) \right) \quad (3.11b)$$

$$= E_{|\vec{X}=\vec{x}} \left(\sum_{i=0}^{+\infty} \lambda_i(\vec{x}) \prod_{j=1}^i (Z_j - z_R) \right) \quad (3.11c)$$

Ainsi, les propriétés de l'indépendance statistique nous permettent de faire traverser une fonction analytique par l'espérance.

●●●● Cependant, l'équation (3.11c) donne certes l'expression d'une variable aléatoire dont $w_f(\vec{x}; S_g(\vec{x}))$ est l'espérance, mais le calcul de cette VA nécessite l'évaluation d'un échantillon strictement infini de Z ; en pratique, l'estimation de $w_f(\vec{x}; z)$ ne pourra donc pas être basée sur cette expression. On en restera donc à l'équation (3.11b), qui donne une trame pour des expressions statistiques calculables de $w_f(\vec{x}; S_g(\vec{x}))$. En effet, plutôt que de commuter directement l'espérance avec une somme infinie, on va transformer la somme infinie aussi en une espérance, comme la possibilité en est évoquée au chapitre 1.3.2.

Par exemple, en posant une variable aléatoire arbitraire I à valeurs dans \mathbb{N} , associée à la suite de probabilités $(P_i)_{i \in \mathbb{N}}$ toutes non nulles, on peut écrire :

$$\begin{aligned} w_f(\vec{x}; E_{|\vec{X}=\vec{x}}(Z)) &= \sum_{i=0}^{+\infty} E_{|\vec{X}=\vec{x}} \left(\lambda_i(\vec{x}) \prod_{j=1}^i (Z_j - z_R) \right) \\ &= \sum_{i=0}^{+\infty} P_i E_{|\vec{X}=\vec{x}} \left(\frac{\lambda_i(\vec{x})}{P_i} \prod_{j=1}^i (Z_j - z_R) \right) \\ &= E_{|\vec{X}=\vec{x}} \left(\frac{\lambda_I(\vec{x})}{P_I} \prod_{j=1}^I (Z_j - z_R) \right) \end{aligned} \quad (3.12)$$

6. L'annexe A.1 donne quelques rappels mathématiques au sujet des développements en série entière (DSE).

Finalement, en posant la variable aléatoire \widetilde{W}_f :

$$\widetilde{W}_f = \frac{\lambda_I(\vec{X})}{P_I} \prod_{j=1}^I (Z_j - z_R) \quad (3.13)$$

on a bien $S_f = E(E_{|\vec{X}}(\widetilde{W}_f)) = E(\widetilde{W}_f)$ avec \widetilde{W}_f calculable. La technique de calcul proposée reposant sur l'utilisation d'un développement en série entière calculée à partir de variables indépendantes, on la désignera dans la suite du manuscrit par « Développement en Série Entière de Variables Indépendantes » (ou DSEVI).

●●●● D'autres VA d'espérance S_f peuvent être construites à partir des Z_j . Nous en introduirons ici 2, en nous inspirant de ce qui a été introduit en section précédente 3.2.1.

L'expression de \widetilde{W}_f nous donne la procédure par laquelle cette V.A. sera échantillonnée (une fois \vec{X} connue) :

1. choisir $I \in \mathbb{N}$ selon les probabilités P_I
2. échantillonner Z I fois de manière indépendante
3. calculer $\widetilde{w}_{f,n}$ selon l'expression (3.13)

Cela signifie qu'on commence par choisir le terme de degré I du DSE de w_f en z , puis qu'on le calcule par I tirages de Z comme proposé à la section précédente. Étant donné qu'on doit échantillonner Z au moins I fois pour calculer le terme de degré I , il peut sembler dommage de ne calculer que le terme de degré I , sans les précédents. Ainsi, on peut construire une estimation de S_f basée sur un échantillon de taille I de Z , utilisant tous les termes du DSE de $w_f(\vec{X}; \bullet)$ jusqu'au degré I .

On pose maintenant (en accord avec le formalisme de la section précédente) une variable aléatoire K à valeurs dans \mathbb{N} , de probabilités $(P'_k)_{k \in \mathbb{N}}$ non-stationnaires à 0. On pose aussi la suite $(R_i^+)_{i \in \mathbb{N}}$ telle que :

$$\forall i \in \mathbb{N}, R_i^+ = \Pr(K \geq i + 1 \mid K \geq i) = \frac{\sum_{k=i+1}^{+\infty} P'_k}{\sum_{k=i}^{+\infty} P'_k} \quad (3.14a)$$

ce qui équivaut à :

$$\forall i \in \mathbb{N}, \prod_{j=0}^{i-1} R_j^+ = \sum_{k=i}^{+\infty} P'_k \quad (3.14b)$$

ou encore :

$$\forall k \in \mathbb{N}, P'_k = (1 - R_k^+) \prod_{i=0}^{k-1} R_i^+ \quad (3.14c)$$

Alors :

$$\begin{aligned}
 w_f(\vec{x}; E_{|\vec{X}=\vec{x}}(Z)) &= \sum_{i=0}^{+\infty} E_{|\vec{X}=\vec{x}} \left(\lambda_i(\vec{x}) \prod_{j=1}^i (Z_j - z_R) \right) \\
 &= \sum_{i=0}^{+\infty} \frac{\sum_{k=i}^{+\infty} P'_k}{\sum_{k=i}^{+\infty} P'_k} E_{|\vec{X}=\vec{x}} \left(\lambda_i(\vec{x}) \prod_{j=1}^i (Z_j - z_R) \right) \\
 &= \sum_{k=0}^{+\infty} P'_k \sum_{i=0}^k \frac{1}{\sum_{k=i}^{+\infty} P'_k} E_{|\vec{X}=\vec{x}} \left(\lambda_i(\vec{x}) \prod_{j=1}^i (Z_j - z_R) \right) \\
 &= E_{|\vec{X}=\vec{x}} \left(\sum_{i=0}^K \left(\frac{\lambda_i(\vec{x})}{\sum_{k=i}^{+\infty} P'_k} \prod_{j=1}^i (Z_j - z_R) \right) \right) \quad (3.15a)
 \end{aligned}$$

$$= E_{|\vec{X}=\vec{x}} \left(\sum_{i=0}^K \left(\frac{\lambda_i(\vec{x})}{\prod_{j=0}^{i-1} R_j^+} \prod_{j=1}^i (Z_j - z_R) \right) \right) \quad (3.15b)$$

En posant la variable aléatoire :

$$\widetilde{W}'_f = \sum_{i=0}^K \left(\frac{\lambda_i(\vec{X})}{\prod_{j=0}^{i-1} R_j^+} \prod_{j=1}^i (Z_j - z_R) \right) \quad (3.16)$$

on obtient un nouvel estimateur calculable de S_f .

Cette logique peut être poussée plus loin en utilisant des permutations. Pour tout $k \in \mathbb{N}$, soit \mathfrak{R}_k un ensemble fini de couples $(\varsigma; \sigma) \in \mathbb{R} \times \mathfrak{S}_k$ tel que $\sum_{(\varsigma; \sigma) \in \mathfrak{R}_k} \varsigma = 1$, on a alors :

$$\begin{aligned}
 w_f(\vec{x}; E_{|\vec{X}=\vec{x}}(Z)) &= E_{|\vec{X}=\vec{x}} \left(\sum_{i=0}^K \left(\frac{\lambda_i(\vec{x})}{\prod_{j=0}^{i-1} R_j^+} \prod_{j=1}^i (Z_j - z_R) \right) \right) \\
 &= E_{|\vec{X}=\vec{x}} \left(\sum_{i=0}^K \left(\frac{\lambda_i(\vec{x})}{\prod_{j=0}^{i-1} R_j^+} \sum_{(\varsigma; \sigma) \in \mathfrak{R}_K} \varsigma \prod_{j=1}^i (Z_{\sigma(j)} - z_R) \right) \right) \quad (3.17)
 \end{aligned}$$

En faisant le choix $\mathfrak{R}_k = \{1/k!\} \times \mathfrak{S}_k$, cela donne :

$$w_f(\vec{x}; E_{|\vec{X}=\vec{x}}(Z)) = E_{|\vec{X}=\vec{x}} \left(\sum_{i=0}^K \left(\frac{\lambda_i(\vec{x})}{\prod_{j=0}^{i-1} R_j^+} \times \frac{1}{K!} \sum_{\sigma \in \mathfrak{S}_K} \prod_{j=1}^i (Z_{\sigma(j)} - z_R) \right) \right) \quad (3.18)$$

On a ainsi de nouvelles expressions aléatoires calculables d'espérance S_f :

$$\widetilde{W}''_f = \sum_{i=0}^K \left(\frac{\lambda_i(\vec{X})}{\prod_{j=0}^{i-1} R_j^+} \sum_{(\varsigma; \sigma) \in \mathfrak{R}_K} \varsigma \prod_{j=1}^i (Z_{\sigma(j)} - z_R) \right) \quad (3.19)$$

$$\widetilde{W}'''_f = \sum_{i=0}^K \left(\frac{\lambda_i(\vec{X})}{\prod_{j=0}^{i-1} R_j^+} \times \frac{1}{K!} \sum_{\sigma \in \mathfrak{S}_K} \prod_{j=1}^i (Z_{\sigma(j)} - z_R) \right) \quad (3.20)$$

3.2.3 Parenthèse sur les sensibilités

Comme affirmé au chapitre 1.5.5, le calcul de sensibilités par la méthode de Monte-Carlo nécessite un travail spécifique sur la formulation intégrale, la méthode des différences finies n'étant pas utilisable. Ce travail de reformulation intégrale commence à être bien balisé, en particulier depuis les travaux de thèse d'Amaury de Guilhem DE LATAILLADE [33, 34], de Maxime ROGER [101, 102], Jérémie DE LA TORRE [31], Jérémie DAUCHET [29], et Olivier FARGES [44]. Le cas de la traversée d'une fonction non-linéaire par une sensibilité reste cependant peu étudié, sauf dans la thèse de J. DAUCHET (à la section 6.4.2, page 248). Ce qui suit reprend partiellement ses travaux, en les plaçant dans le formalisme utilisé dans ce manuscrit.

On s'intéresse au calcul de la sensibilité de S_f vis-à-vis d'un paramètre π n'intervenant dans l'expression de S_f que via S_g , ce qui s'écrit « $S_f = \int_{\mathcal{U}_{\vec{X}}} d\vec{x} f(\vec{x}; S_g(\vec{x}; \pi))$ ». Le cas plus général « $S_f = \int_{\mathcal{U}_{\vec{X}}(\pi)} d\vec{x} f(\vec{x}; S_g(\vec{x}; \pi); \pi)$ » se traite en sommant les contributions à la sensibilité vis-à-vis de π passant par $\mathcal{U}_{\vec{X}}(\pi)$, par $\partial_3 f$, et par $S_g(\vec{x}; \pi)$. Les 2 premières contributions font déjà l'objet de travaux extensifs dans [29, 31, 44, 101, 102], et sont de peu d'intérêt dans ce chapitre; nous nous concentrons donc sur la 3^e.

En supposant que $S_f = \int_{\mathcal{U}_{\vec{X}}} d\vec{x} f(\vec{x}; S_g(\vec{x}; \pi))$ on a :

$$\partial_\pi S_f = \int_{\mathcal{U}_{\vec{X}}} d\vec{x} \partial_\pi f(\vec{x}; S_g(\vec{x}; \pi)) \quad (3.21)$$

$$= \mathbb{E}(\partial_\pi w_f(\vec{X}; S_g(\vec{X}; \pi))) \quad (3.22)$$

La difficulté du calcul de $\partial_\pi S_f$ se transfère donc sur le calcul de $\partial_\pi w_f(\vec{X}; S_g(\vec{X}; \pi))$. En l'occurrence, par dérivation d'une composée :

$$\forall \vec{x} \in \mathcal{U}_{\vec{X}}, \partial_\pi w_f(\vec{x}; S_g(\vec{x}; \pi)) = \partial_2 w_f(\vec{x}; S_g(\vec{x}; \pi)) \times \partial_\pi S_g(\vec{x}; \pi) \quad (3.23)$$

Nous avons supposé précédemment que f était Développable en Série Entière sur sa 2^e variable, de sorte que pour tout $\vec{x} \in \mathcal{U}_{\vec{X}}$, $S_g(\vec{x}; \pi)$ soit dans le rayon de convergence du DSE de $f(\vec{x}; \bullet)$, ce qui s'écrivait :

$$w_f(\vec{x}; S_g(\vec{x}; \pi)) = \sum_{i=0}^{+\infty} \lambda_i(\vec{x}) (S_g(\vec{x}; \pi) - z_R)^i \quad (3.11a)^\dagger$$

De par les propriétés de dérivation des DSE, la formule (3.23), et l'identité $\mathbb{E}_{|\vec{X}=\vec{x}}(Z) = S_g(\vec{x}; \pi)$:

$$\partial_\pi w_f(\vec{x}; \mathbb{E}_{|\vec{X}=\vec{x}}(Z)) = \partial_2 w_f(\vec{x}; \mathbb{E}_{|\vec{X}=\vec{x}}(Z)) \times \partial_\pi \mathbb{E}_{|\vec{X}=\vec{x}}(Z) \quad (3.24)$$

$$= \sum_{i=1}^{+\infty} i \lambda_i(\vec{x}) (\mathbb{E}_{|\vec{X}=\vec{x}}(Z) - z_R)^{i-1} \partial_\pi \mathbb{E}_{|\vec{X}=\vec{x}}(Z) \quad (3.25)$$

On suppose maintenant qu'on dispose d'une variable aléatoire $Z_{\partial\pi}$ telle que $E_{|\vec{X}}(Z_{\partial\pi}) = \partial_\pi E_{|\vec{X}}(Z)$ ⁷.

$$\partial_\pi w_f(\vec{x}; E_{|\vec{X}=\vec{x}}(Z)) = \sum_{i=0}^{+\infty} E_{|\vec{X}=\vec{x}} \left(i \lambda_i(\vec{x}) Z_{\partial\pi,1} \prod_{j=2}^i (Z_j - z_R) \right) \quad (3.26)^8$$

On peut maintenant construire des variables aléatoires calculables d'espérance $\partial_\pi S_f$ sur la base des mêmes transformations de la somme infinie du DSE en espérance qu'on a utilisées pour l'expression de variables aléatoires calculables d'espérance S_f .

$$\begin{aligned} \partial_\pi w_f(\vec{x}; E_{|\vec{X}=\vec{x}}(Z)) &= \sum_{i=0}^{+\infty} E_{|\vec{X}=\vec{x}} \left(i \lambda_i(\vec{x}) Z_{\partial\pi,1} \prod_{j=2}^i (Z_j - z_R) \right) \\ &= E_{|\vec{X}=\vec{x}} \left(I \frac{\lambda_I(\vec{x})}{P_I} Z_{\partial\pi,1} \prod_{j=2}^I (Z_j - z_R) \right) \end{aligned} \quad (3.27)$$

D'où l'expression d'une V.A. $\widetilde{W_{f,\partial\pi}}$ d'espérance $\partial_\pi S_f$:

$$\widetilde{W_{f,\partial\pi}} = \frac{I \lambda_I(\vec{X})}{P_I} Z_{\partial\pi,1} \prod_{j=2}^I (Z_j - z_R) \quad (3.28)$$

Ou sinon :

$$\begin{aligned} \partial_\pi w_f(\vec{x}; E_{|\vec{X}=\vec{x}}(Z)) &= \sum_{i=0}^{+\infty} E_{|\vec{X}=\vec{x}} \left(i \lambda_i(\vec{x}) Z_{\partial\pi,1} \prod_{j=2}^i (Z_j - z_R) \right) \\ &= E_{|\vec{X}=\vec{x}} \left(Z_{\partial\pi,1} \sum_{i=0}^K \left(\frac{i \lambda_i(\vec{x})}{\sum_{k=i}^{+\infty} P'_k} \prod_{j=2}^i (Z_j - z_R) \right) \right) \end{aligned} \quad (3.29)$$

$$= E_{|\vec{X}=\vec{x}} \left(Z_{\partial\pi,1} \sum_{i=0}^K \left(\frac{i \lambda_i(\vec{x})}{\prod_{j=0}^{i-1} R_j^+} \prod_{j=2}^i (Z_j - z_R) \right) \right) \quad (3.30)$$

Ce qui donne de nouvelles façons d'estimer $\partial_\pi S_f$:

$$\widetilde{W'_{f,\partial\pi}} = \sum_{i=0}^K \frac{i \lambda_i(\vec{X})}{\prod_{j=0}^{i-1} R_j^+} Z_{\partial\pi,1} \prod_{j=2}^i (Z_j - z_R) \quad (3.31)$$

7. Au sens où Z est le produit d'un mécanisme pseudo-aléatoire impliquant π , on peut en général dériver Z par rapport à π . Si c'est faisable, un choix possible pour $Z_{\partial\pi}$ est de prendre $Z_{\partial\pi} = \partial_\pi Z$. Néanmoins, ce choix n'est nullement une obligation.

8. Strictement, la somme devrait commencer à $i = 1$. On peut formellement considérer qu'elle commence à $i = 0$, car à $i = 0$ son terme est nul ; cela permet de la définir sur le même ensemble qu'à la section précédente, et par là de mener les calculs de S_f et $\partial_\pi S_f$ simultanément.

Ou encore :

$$\begin{aligned} \partial_\pi w_f(\vec{x}; E_{|\vec{X}=\vec{x}}(Z)) &= E_{|\vec{X}=\vec{x}} \left(Z_{\partial\pi,1} \sum_{i=0}^K \left(\frac{i\lambda_i(\vec{x})}{\prod_{j=0}^{i-1} R_j^+} \prod_{j=2}^i (Z_j - z_R) \right) \right) \\ &= E_{|\vec{X}=\vec{x}} \left(\sum_{i=0}^K \left(\frac{i\lambda_i(\vec{x})}{\prod_{j=0}^{i-1} R_j^+} \sum_{(\varsigma;\sigma) \in \mathfrak{R}_K} \varsigma Z_{\partial\pi,\sigma(1)} \prod_{j=2}^i (Z_{\sigma(j)} - z_R) \right) \right) \end{aligned} \quad (3.32)$$

$$= E_{|\vec{X}=\vec{x}} \left(\sum_{i=0}^K \left(\frac{i\lambda_i(\vec{x})}{\prod_{j=0}^{i-1} R_j^+} \times \frac{1}{K!} \sum_{\sigma \in \mathfrak{S}_K} Z_{\partial\pi,\sigma(1)} \prod_{j=2}^i (Z_{\sigma(j)} - z_R) \right) \right) \quad (3.33)$$

Ce qui donne les V.A. d'espérance $\partial_\pi S_f$:

$$\widetilde{W''_{f,\partial\pi}} = \sum_{i=0}^K \frac{i\lambda_i(\vec{X})}{\prod_{j=0}^{i-1} R_j^+} \sum_{(\varsigma;\sigma) \in \mathfrak{R}_K} \varsigma Z_{\partial\pi,\sigma(1)} \prod_{j=2}^i (Z_{\sigma(j)} - z_R) \quad (3.34)$$

$$\widetilde{W'''_{f,\partial\pi}} = \sum_{i=0}^K \frac{i\lambda_i(\vec{X})}{\prod_{j=0}^{i-1} R_j^+} \times \frac{1}{K!} \sum_{\sigma \in \mathfrak{S}_K} Z_{\partial\pi,\sigma(1)} \prod_{j=2}^i (Z_{\sigma(j)} - z_R) \quad (3.35)$$

3.3 (*) Calcul de la productivité annuelle d'une centrale solaire à concentration

Nous disions en introduction du manuscrit que la MMC permet, sans aucune difficulté, de passer du calcul de la puissance instantanée d'une centrale solaire au calcul de sa puissance moyenne sur par exemple 1 an, et ce en utilisant des données climatologiques détaillées par pas de 15 minutes.

Mais nous pensions à la puissance reçue, sans tenir compte de sa conversion — non-linéaire. Ici, on tient compte de cette conversion ; il nous faut alors tirer plusieurs rayons à chaque date. Cependant, le temps de calcul n'est presque pas augmenté.

Ce calcul est repris de la thèse d'Olivier FARGES [44].

3.3.1 Contexte

●●●● La consommation croissante d'énergie par les sociétés modernes, associée à des difficultés croissantes d'approvisionnement (difficultés en cours ou prévues), amène à s'intéresser à de nouvelles sources primaires d'énergie convertible en électricité. Ces sources sont déjà plus ou moins connues ; l'une d'elles, qui attire

aujourd'hui beaucoup d'intérêt, est l'énergie solaire concentrée.

Rapidement, le principe est de concentrer la puissance lumineuse issue du Soleil et dirigée vers un terrain (1,36 kW/m² de surface face au Soleil, au sommet de l'atmosphère), sur un récepteur, à l'aide d'un jeu de miroir. Par temps clair, la luminance du disque solaire

étant très élevée ($20,0 \text{ MW.m}^{-2}.\text{sr}^{-1}$ au sommet de l'atmosphère), cela permet d'échauffer le récepteur jusqu'à plusieurs milliers de degrés Celsius pendant qu'on lui extrait de la chaleur : on peut ainsi alimenter un cycle thermodynamique produisant de l'énergie consommable, typiquement de l'électricité.

Les technologies de concentration de la lumière solaire sont ordinairement réparties en 2 catégories : concentration linéaire et concentration ponctuelle. La figure 3.3 en donne quelques exemples. La technologie étudiée par O. FARGES est appelée centrale Beam Down, et se rapproche des centrales à tour (où la concentration est ponctuelle).

●●●● Dans une centrale à tour, le terrain de la centrale est recouvert d'héliostats — des miroirs à orientation pilotée automatiquement à distance. Ces héliostats réfléchissent la lumière solaire vers un récepteur situé en hauteur.

La technologie *Beam Down* se distingue de la technologie à tour classique par le remplacement, en haut de la tour, du récepteur par un miroir secondaire renvoyant la lumière issue du champ d'héliostats vers le récepteur (situé cette fois-ci au sol). Ceci est illustré sur la figure 3.4.

La technologie Beam Down présente plusieurs avantages appréciables sur la technologie à tour ordinaire, et aussi quelques inconvénients :

- La tour n'a plus à supporter le poids du récepteur et d'une partie du cycle thermodynamique, mais celui du réflecteur secondaire, beaucoup plus léger. Cela représente des économies de construction substantielles.
- Le cycle thermodynamique est désormais complètement au sol, ce qui élimine des problèmes potentiels dus à l'éloignement de ses puits thermiques chaud et froid (ceux dus à une longue tuyauterie, par exemple).
- Comme le récepteur est au sol, sa disposition est plus facilement maîtrisable.
- Comme il y a deux réflexions au lieu d'une, les pertes par réflexion sur les miroirs augmentent.
- La réflexion sur le miroir secondaire courbe rend le système très sensible

aux erreurs de pointage : un léger défaut d'orientation d'un héliostat envoie sa contribution loin hors du récepteur. Ce défaut peut cependant être corrigé à l'aide d'optiques supplémentaires au sol.

●●●● Quoi qu'il en soit, en phase de conception, une fois qu'un type de design est choisi, il convient de l'optimiser. Il faut donc disposer d'une grandeur d'intérêt.

Jusqu'à la thèse d'O. FARGES, c'était la puissance thermique récupérable au récepteur qui était en général considérée. Cependant, ce choix, bien que raisonnable, n'est pas le mieux que l'on puisse faire, car si la puissance solaire est bien ce qui fait fonctionner la centrale, ce n'est pas son produit final, qui est vendu et utilisé ; ce sera plutôt de l'électricité, ou un produit chimique. Or le rendement puissance solaire \rightarrow puissance valorisée n'est pas forcément constant.

Tant que l'on reste dans une technique de design *à l'ancienne*, où l'on optimisait le design de la centrale vis-à-vis de l'éclairage d'une date donnée (dans la plupart des cas, le 21 Mars à midi solaire), ce précédent point n'a pas d'importance. Il en prend quand l'optimisation est faite en considérant la productivité annuelle de la centrale (bien plus représentative de ses capacités en utilisation). Si le rendement du cycle thermodynamique varie avec la puissance thermique disponible, il n'y a plus correspondance entre la chaleur totale reçue au récepteur et la production dans l'année de la centrale ; la répartition temporelle de la puissance joue aussi.

La grandeur d'intérêt devient donc la production annuelle de la centrale, calculée comme l'intégrale sur l'année de la puissance sortante de la centrale, cette puissance étant fonction de la puissance thermique récoltée au récepteur. Si cette puissance thermique est calculée par MMC, on retombe exactement dans la problématique de ce chapitre.

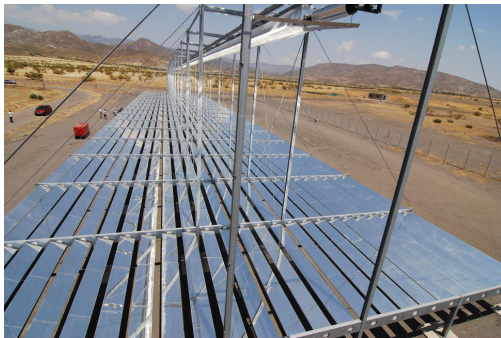
Remarques.

- Au lieu d'estimer la production de la centrale au cours d'une année, on peut carrément estimer sa production au cours de toute son existence prévue. Le calcul reste assez identique, mais il faut

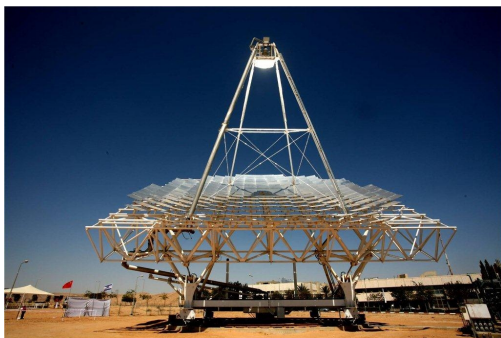
3.3. Productivité annuelle d'une centrale solaire



(a) Un concentrateur cylindro-parabolique.
Source : site web de NextEra Energy
Ressources (entreprise états-unienne)



(b) [Modèle pilote de concentrateur linéaire
de Fresnel, à la plate-forme d'Almeria.
Source : DLR



(c) Concentrateur Dish, avec miroir de Fresnel.
Installation d'Heliofocus (entreprise israélienne)
Source : CSP Korea



(d) Vue sur un des champs de concentrateurs cylindro-paraboliques du
complexe SEGS, en Californie.
Source : Wikimedia Commons



(e) Photo partielle de la centrale à tour
Solar Two en fonctionnement, au Ne-
vada. Centrale aujourd'hui démantelée.
Source : Wikimedia Commons



(f) Concentrateur Dish avec moteur Stirling au
foyer, en exposition au four solaire d'Odeillo.
Source : Wikimedia Commons

FIGURE 3.3 – Exemples de différentes technologies de concentration du rayonnement solaire. On parle de concentration *linéaire* ((a), (d), et (b)) ou *ponctuelle* ((c), (e), et (f)) selon la forme du récepteur. Si pour chaque récepteur, le miroir concentrateur est morcelé, on parle de *miroir de Fresnel*.



(a) Centrale à tour « classique » Solar Tres, en Espagne.
Source : CSP Korea. Photo commerciale.



(b) Installation pilote de type Beam Down en construction à Masdar, à Abu Dhabi.
Source : site web <http://nenmore.blogspot.fr>

FIGURE 3.4 – Comparatif de principe entre une centrale à tour Beam Down (b) et une centrale à tour plus ordinaire (a).

pouvoir prendre en compte les effets du vieillissement de l'installation.

- En plus du rendement de conversion de la puissance, on peut prendre en compte les variations du prix de l'énergie vendue selon la date et l'heure. Cela laisse la forme du calcul presque inchangée.

3.3.2 Démarche

Le calcul mené est fait sur une centrale à tour ordinaire. O. FARGES a fait ce choix en partie pour avoir une comparaison avec une situation expérimentale (la centrale PS10). Toutefois, l'algorithme de calcul entre ce cas d'étude et une centrale Beam Down change très peu.

Pour calculer la puissance arrivante sur le récepteur, on fait les hypothèses suivantes :

- Toute la puissance lumineuse arrivant sur le récepteur (noté \mathcal{R}) a auparavant subi une et une seule réflexion sur la face avant des héliostats (notée \mathcal{H}^+), et vient avant cela du disque solaire (noté \mathcal{S}). C'est-à-dire que les seuls trajets optiques amenant de la lumière au récepteur sont du type : $\mathcal{S} \rightarrow \mathcal{H}^+ \rightarrow \mathcal{R}$.
- Les héliostats sont des miroirs gris et spéculaires. Les défauts des miroirs sont pris en compte par une déviation statistique de la normale à leur surface, déviation qui suit le modèle de micro-

facettes de Blinn (détaillé un peu plus tard dans cette section, et dans [92]).

- L'air dans lequel baigne la centrale est transparent.

Alors la puissance lumineuse reçue par le récepteur s'exprime comme :

$$P_{th} = \int_{\mathcal{H}^+} d\vec{r}_1 \int_{\mathcal{S}} d\vec{u}_s \int_{\mathcal{W}_{\vec{n}_h}} p_{\vec{N}_h}(\vec{n}_h | \vec{u}_s; b) d\vec{n}_h (\vec{u}_s \cdot \vec{n}_h) \left\{ \begin{array}{l} H(\vec{r}_0 \notin \mathcal{S}) \times 0 + \\ H(\vec{r}_0 \in \mathcal{S}) \times \left\{ \begin{array}{l} H(\vec{r}_2 \notin \mathcal{R}) \times 0 + \\ H(\vec{r}_2 \in \mathcal{R}) \rho_{\mathcal{H}} L_{\mathcal{S}} \end{array} \right\} \end{array} \right\} \quad (3.36)$$

où :

- \vec{n}_h est la normale à la surface de l'héliostat, suivant une distribution statistique décrite par le modèle de Blinn.
- b est un paramètre du modèle de Blinn.
- \vec{r}_0 est le 1^{er} point d'intersection du rayon partant de \vec{r}_1 et de direction \vec{u}_s . On compte $\vec{r}_0 \in \mathcal{S}$ si le rayon « atteint l'infini » au niveau du disque solaire (ce sera toujours au niveau du disque solaire car on a choisi $\vec{u}_s \in \mathcal{S}$), et $\vec{r}_0 \notin \mathcal{S}$ si le rayon impacte un obstacle avant d'atteindre le ciel (un héliostat situé devant, la tour, ou le terrain).

- \vec{r}_2 est le 1^{er} point d'intersection du rayon réfléchi venant de \vec{r}_1 . On compte $\vec{r}_2 \in \mathcal{R}$ si le rayon atteint le récepteur, et $\vec{r}_2 \notin \mathcal{R}$ sinon (que ce soit parce que le rayon impacte l'arrière d'un autre héliostat, ou parce qu'il passe à côté du récepteur).
- $\rho_{\mathcal{H}}$ est la réflectivité des héliostats.
- $L_{\mathcal{S}}$ est la luminance du disque solaire.

Cette expression peut être justifiée par le formalisme détaillé au chapitre 1.4, en considérant qu'on fait sur la surface des héliostats le produit scalaire de la luminance issue du Soleil et de la captance issue du récepteur. Avec les hypothèses de travail faites, ce calcul doit donner le même résultat que l'intégration

de la luminance sur le récepteur, ou que le produit scalaire de la captance du récepteur sur la luminosité du ciel.

La luminance du disque solaire est souvent ré-exprimée selon une grandeur plus usuelle, la *DNI* (pour *Direct Normal Insulation*). Il mesure la puissance lumineuse reçue par une unité de surface, orientée face au Soleil. C'est une fonction de la date et de l'heure (à cause de l'atmosphère au-dessus de la centrale). Comme le disque solaire est très étroit dans le ciel on peut considérer que $L_{\mathcal{S}} \simeq DNI/\Omega_{\mathcal{S}}$, où $\Omega_{\mathcal{S}} = \int_{\mathcal{S}} d\vec{u}_s$ est l'angle solide du disque solaire. $\Omega_{\mathcal{S}}$ est calculable à l'aide du rayon angulaire du disque solaire $\theta_{\mathcal{S}} = 4,65$ mrad selon la formule $\Omega_{\mathcal{S}} = 2\pi(1 - \cos \theta_{\mathcal{S}}) = 68 \mu\text{sr}$.

En introduisant des densités de probabilité sur \mathcal{H}^+ et sur \mathcal{S} , on peut maintenant donner une expression utilisable par MMC de P_{th} :

$$P_{th} = \int_{\mathcal{H}^+} p_{\vec{R}_1}(\vec{r}_1) d\vec{r}_1 \int_{\mathcal{S}} p_{\vec{U}_s}(\vec{u}_s) d\vec{u}_s \int_{\mathcal{U}_{\vec{N}_h}} p_{\vec{N}_h}(\vec{n}_h|\vec{u}_s; b) d\vec{n}_h w_{P_{th}} \quad (3.37a)$$

avec

$$w_{P_{th}} = \left\{ \begin{array}{l} H(\vec{r}_0 \notin \mathcal{S}) \times 0 + \\ H(\vec{r}_0 \in \mathcal{S}) \times \left\{ \begin{array}{l} H(\vec{r}_2 \notin \mathcal{R}) \times 0 + \\ H(\vec{r}_2 \in \mathcal{R}) \frac{(\vec{u}_s \cdot \vec{n}_h) \rho_{\mathcal{H}} DNI}{p_{\vec{R}_1}(\vec{r}_1) \Omega_{\mathcal{S}} p_{\vec{U}_s}(\vec{u}_s)} \end{array} \right\} \end{array} \right\} \quad (3.37b)$$

Des choix usuels pour $p_{\vec{R}_1}$ et $p_{\vec{U}_s}$ sont :

$$\left\{ \begin{array}{l} p_{\vec{R}_1}(\vec{r}_1) = \frac{1}{S_{\mathcal{H}^+}} \\ p_{\vec{U}_s}(\vec{u}_s) = \frac{1}{\Omega_{\mathcal{S}}} = \frac{1}{2\pi(1 - \cos \theta_{\mathcal{S}})} \end{array} \right. \quad (3.38)$$

$p_{\vec{N}_h}(\vec{n}_h|\vec{u}_s; b)$ doit quant à elle suivre le modèle qu'on s'est fixé⁹, en l'occurrence le modèle de Blinn :

$$p_{\vec{N}_h}(\vec{n}_h|\vec{u}_s; b) = \frac{1 + \frac{1}{b}}{2\pi \left(1 - \cos\left(\frac{1}{2} \arcsin(\vec{u}_s \cdot \vec{n}_1)\right)^{2 + \frac{1}{b}}\right)} (\vec{n}_h \cdot \vec{n}_1)^{1 + \frac{1}{b}} \quad (3.39)$$

où \vec{n}_1 est la normale à la surface de l'héliostat, d'un point de vue purement géométrique.

On pose pour la suite $P_{ref} = DNI S_{\mathcal{H}^+}$ (pour rappel, DNI et donc P_{ref} dépendent de t), et ainsi $P_{th} = \eta_{opt} P_{ref}$, avec le rendement optique :

$$\eta_{opt} = \int_{\mathcal{H}^+} p_{\vec{R}_1}(\vec{r}_1) d\vec{r}_1 \int_{\mathcal{S}} p_{\vec{U}_s}(\vec{u}_s) d\vec{u}_s \int_{\mathcal{U}_{\vec{N}_h}} p_{\vec{N}_h}(\vec{n}_h|\vec{u}_s; b) d\vec{n}_h w_{\eta_{opt}} \quad (3.40a)$$

9. On aurait pu introduire une nouvelle densité de probabilité pour \vec{N}_h , de façon à modifier le tirage aléatoire comme la possibilité en est évoquée au chapitre 1.3.2. Ceci ne s'est pas avéré utile ici.

où

$$w_{\eta_{opt}} = \left\{ \begin{array}{l} H(\vec{r}_0 \notin \mathcal{S}) \times 0 + \\ H(\vec{r}_0 \in \mathcal{S}) \times \left\{ \begin{array}{l} H(\vec{r}_2 \notin \mathcal{R}) \times 0 + \\ H(\vec{r}_2 \in \mathcal{R}) \frac{(\vec{u}_s \cdot \vec{n}_h) \rho_{\mathcal{H}}}{S_{\mathcal{H}+} p_{\vec{R}_1}(\vec{r}_1) \Omega_{\mathcal{S}} p_{\vec{U}_s}(\vec{u}_s)} \end{array} \right\} \end{array} \right\} \quad (3.40b)$$

On pose par la suite $\vec{\gamma} = (\vec{r}_1; \vec{u}_s; \vec{n}_h)$ et la variable aléatoire $\vec{\Gamma} = (\vec{R}_1; \vec{U}_s; \vec{N}_h)$ (qui décrit bel et bien un chemin optique), ce qui permet d'écrire plus simplement :

$$\eta_{opt} = \int_{\mathcal{W}_{\vec{\Gamma}}} p_{\vec{\Gamma}}(\vec{\gamma}) d\vec{\gamma} w_{\eta_{opt}} \quad (3.41)$$

où $w_{\eta_{opt}}$ se calcule comme précédemment.

On reprend le cas de génération électrique étudié dans sa thèse par O. FARGES. Basiquement, comme on génère de l'électricité la puissance valorisée est $P_{elec} = \eta_{cycle} P_{th}$. Quand le DNI est supérieur à une valeur seuil DNI_l de démarrage de la centrale, le rendement du cycle thermodynamique est modélisé par une loi puissance :

$$\eta_{cycle} = k_{eff} \left(\frac{P_{th}}{P_{nom}} \right)^{\alpha_{eff}} \quad (3.42)$$

Ce qui amène à :

$$P_{elec} = k_{eff} \left(\frac{P_{ref}}{P_{nom}} \right)^{\alpha_{eff}} P_{ref} \eta_{opt}^{\alpha_{eff}+1} \quad (3.43)$$

où η_{opt} est la seule partie inconnue de l'expression. Si $DNI < DNI_l$, on considère que $P_{elec} = 0$.

Les valeurs des coefficients sont : $k_{eff} = 0,425\,267$, $\alpha_{eff} = 0,125\,788$, et $P_{nom} = 100$ MW.

L'expression du rendement optique η_{opt} montre que ce dernier est compris entre 0 et 1, on peut donc utiliser le développement en série entière de $x \mapsto (1+x)^\alpha$ autour de 0 :

$$\begin{aligned} \eta_{opt}^{\alpha_{eff}+1} &= (1 + \eta_{opt} - 1)^{\alpha_{eff}+1} \\ &= \sum_{i=0}^{+\infty} \left(\prod_{j=0}^{i-1} \frac{\alpha_{eff} + 1 - j}{j+1} \right) (\eta_{opt} - 1)^i \quad \text{ssi } \eta_{opt} - 1 \in]-1; 1[\end{aligned} \quad (3.44)$$

$$= \sum_{i=0}^{+\infty} \prod_{j=0}^{i-1} \left(\frac{\alpha_{eff} + 1 - j}{j+1} (\eta_{opt} - 1) \right) \quad (3.45)$$

On retrouve bien l'expression $\eta_{opt}^{\alpha_{eff}+1} = \sum_{i=0}^{+\infty} \lambda_i (E(Z) - z_R)^i$ avec :

$$\left\{ \begin{array}{l} \lambda_i = \prod_{j=0}^{i-1} \frac{\alpha_{eff} + 1 - j}{j+1} \\ Z = W_{\eta_{opt}} \\ z_R = 1 \end{array} \right. \quad (3.46)$$

Un autre choix est possible, avec l'avantage que les λ_i et les $Z_i - z_R$ sont presque tous positifs :

$$\left\{ \begin{array}{l} \lambda_i = \prod_{j=0}^{i-1} \frac{j - \alpha_{eff} - 1}{j+1} \\ Z = -W_{\eta_{opt}} \\ z_R = -1 \end{array} \right. \quad (3.47)$$

Ne reste maintenant plus qu'à convertir cette écriture en une expression calculable. On fera ici le même choix algorithmique qu'O. FARGES, directement repris des travaux de J. DAUCHET ;

c'est-à-dire qu'on utilise l'expression (3.15b) avec le choix $\forall j \in \mathbb{N}, R_j^+ = 1 - W_{\eta_{opt} \ j+1}$ (qui est bien compris entre 0 et 1).

La formulation statistique de l'algorithme qui en découle est la suivante :

$$\left\{ \begin{array}{l} P_{elec} = E \left(H(DNI \geq DNI_l) k_{eff} \left(\frac{P_{ref}}{P_{nom}} \right)^{\alpha_{eff}} P_{ref} w_{NL} \right) \\ w_{NL} = \sum_{i=0}^K \lambda_i \\ \lambda_i = \prod_{j=0}^{i-1} \frac{j - \alpha_{eff} - 1}{j + 1} \\ \mathcal{U}_K \subset \mathbb{N} \text{ et } \forall j \in \mathbb{N}, R_j^+ = \Pr(K \geq j + 1 \mid K \geq j) = 1 - W_{\eta_{opt} \ j+1} | T \\ T \text{ est la date (heure incluse), tirée uniformément sur l'année} \\ W_{\eta_{opt}} \text{ est précédemment décrit, dans l'équation (3.40b) et} \\ \text{dans le descriptif de l'équation (3.36)} \end{array} \right. \quad (3.48)$$

Remarque. Un point d'achoppement pourrait apparaître : si η_{opt} est rigoureusement nul (par exemple la nuit) on doit obtenir $w_{NL} = 0$. Cependant, si $\eta_{opt} = 0$, le DSE utilisé ne converge pas.

Cependant, la condition sur le DNI de mise en marche de la centrale rend ce cas impossible en pratique. On peut en effet espérer que le champ d'héliostats ne soit pas assez mal conçu pour que son rendement optique soit rigoureusement nul quand le Soleil brille !

L'algorithme finalement construit pour le calcul de la puissance électrique moyenne de la centrale $P_{M,elec}$ est décrit en 3.2.

3.3.3 Résultats

●●●● On se contentera dans cette section essentiellement d'épiloguer sur les résultats obtenus par Olivier FARGES. Je n'ai en effet pas recommencé ses simulations, d'une part à cause de la difficulté d'une telle entreprise, et d'autre part par manque de temps à y consacrer et manque d'intérêt dans cet acte. Cela aurait été d'autant moins intéressant que faisant les mêmes hypothèses de travail et utilisant le même algorithme de calcul, j'aurais dû obtenir les mêmes résultats. À mes yeux, reprendre le travail d'O. FARGES constitue essentiellement, dans ce manuscrit, l'occasion d'illustrer les principes méthodologiques et mathématiques énoncés dans ce chapitre.

Comme annoncé dans la section précédente, nous prendrons les paramètres $k_{eff} =$

0,425 267, $\alpha_{eff} = 0,125\,788$, et $P_{nom} = 100$ MW. À ceux-ci s'ajoutent tous les paramètres géométriques et optiques de la centrale simulée, ainsi que les données sur l'ensoleillement. Nous ne les listerons pas ici, étant donné leur quantité ; le lecteur intéressé pourra se reporter au manuscrit de thèse d'O. FARGES [44].

●●●● La figure 3.5 donne une représentation visuelle de la loi de conversion $P_{th} \rightarrow P_{elec}$. Comme on peut le constater, cette loi est finalement assez proche d'une loi affine : on pourrait par exemple l'approximer par la loi

$$P_{elec} = 0,430\,131 \times P_{th} - 1369,39 \text{ kW} \quad (3.49)$$

comme cela est montré sur la figure 3.5.

●●●● La figure 3.6 donne les résultats de simulation. Comme on peut le constater, la convergence de l'algorithme a l'air correcte.

De plus, il y a finalement peu de différence entre les résultats de l'approche non-linéaire et les résultats qu'aurait eu une approche linéaire basée sur l'expression (3.49), que ce soit en termes de valeur ou de convergence. Cela est dû à plusieurs facteurs qui se rassemblent dans ce cas :

- La faible non-linéarité de la loi de conversion $P_{th} \rightarrow P_{elec}$, qui explique à elle seule la proximité des valeurs obtenues.
- L'utilisation d'une valeur de référence P_{ref} pour mettre en œuvre le Dévelop-

Résultat : Une estimation $(\overline{w_{P_{M,elec}}}; s^2(\overline{w_{P_{M,elec}}}))$ de $P_{M,elec}$

```

1  $\Sigma_{P_{M,elec}} \leftarrow 0;$ 
2  $\Sigma_{P_{M,elec}^2} \leftarrow 0;$ 
3 pour  $i \leftarrow 1$  à  $N_{\text{tirages}}$  faire
4   échantillonner  $T$  : on obtient  $t_i$ ;
5   si  $DNI < DNI_l$  alors  $w_{NL\ i} \leftarrow 0;$ 
6   sinon
7      $\Sigma_\lambda \leftarrow 0;$ 
8      $\lambda \leftarrow 1;$ 
9      $j \leftarrow 0;$ 
10    répéter
11       $\Sigma_\lambda \leftarrow \Sigma_\lambda + \lambda;$ 
12       $\lambda \leftarrow \frac{j - \alpha_{eff} - 1}{j + 1} \lambda;$ 
13       $j \leftarrow j + 1;$ 
14      échantillonner  $\vec{\Gamma}$  : on obtient  $\vec{\gamma}_{j,i} = (\vec{r}_{1\ j,i}; \vec{u}_{s\ j,i}; \vec{n}_{h\ j,i});$ 
15       $w_{\eta_{opt\ j,i}} \leftarrow H(\vec{r}_{0\ j,i} \in \mathcal{S}) \times H(\vec{r}_{2\ j,i} \in \mathcal{R}) \times$ 
16         $\frac{(\vec{u}_{s\ j,i} \cdot \vec{n}_{h\ j,i}) \rho_{\mathcal{H}}}{S_{\mathcal{H}^+} p_{\vec{R}_1}(\vec{r}_{1\ j,i}) \Omega_{\mathcal{S}} p_{\vec{U}_s}(\vec{u}_{s\ j,i})};$ 
17       $R_{j-1,i}^+ \leftarrow 1 - w_{\eta_{opt\ j,i}};$ 
18      échantillonner  $R$  uniformément dans  $[0; 1]$  : on obtient  $r_{j-1,i};$ 
19      jusqu'à  $r_{j-1,i} > R_{j-1,i}^+;$ 
20       $w_{P_{M,elec\ i}} \leftarrow \frac{\Sigma_\lambda DNI(t_i) S_{\mathcal{H}^+}}{p_T(t_i) \times 1 \text{ an}};$ 
21       $\Sigma_{P_{M,elec}} \leftarrow \Sigma_{P_{M,elec}} + w_{P_{M,elec\ i}};$ 
22       $\Sigma_{P_{M,elec}^2} \leftarrow \Sigma_{P_{M,elec}^2} + w_{P_{M,elec\ i}}^2;$ 
23  $\overline{w_{P_{M,elec}}} \leftarrow \Sigma_{P_{M,elec}} / N_{\text{tirages}};$ 
24  $s^2(\overline{w_{P_{M,elec}}}) \leftarrow (\Sigma_{P_{M,elec}^2} / N_{\text{tirages}} - \overline{w_{P_{M,elec}}}^2) / (N_{\text{tirages}} - 1);$ 
    
```

Algorithme 3.2 : Algorithme pour le calcul de la puissance électrique moyenne d'une centrale solaire à tour

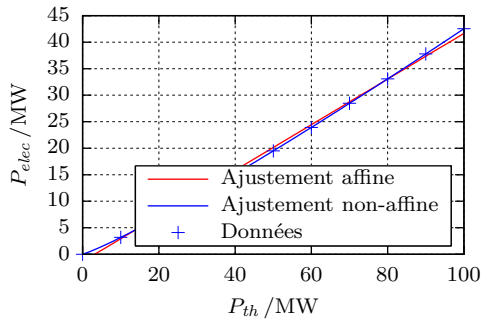


FIGURE 3.5 – Loi de conversion thermique \rightarrow électrique de la centrale étudiée
Crédits : Olivier FARGES

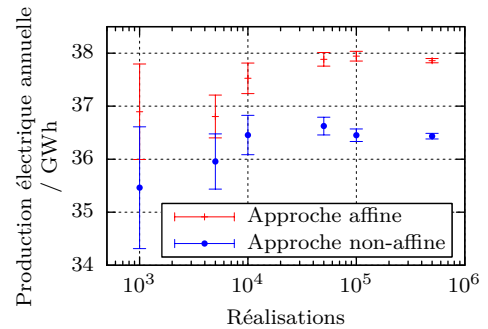


FIGURE 3.6 – Résultats de calcul obtenus, avec l'algorithme 3.2 ou en utilisant la loi de conversion affine (3.49)
Crédits : Olivier FARGES

pement en Série Entière proche de la valeur réelle de P_{th} . Le manuscrit [44] explique que la centrale étudiée est dimensionnée pour éviter les pertes optiques, que ce soit par ombrage ($\vec{r}_0 \notin \mathcal{S}$), blocage ($\vec{r}_2 \notin \mathcal{R}$ car l'arrière d'un héliostat gêne), ou débordement ($\vec{r}_2 \notin \mathcal{R}$ car le rayon réfléchi passe à côté du récepteur); il en résulte que le rendement optique η_{opt} , tel que nous l'avons défini, est proche de 1.

— Le choix que nous avons fait sur $R_j^+ =$

$1 - w_{\eta_{opt} j}$, combiné à la proximité statistique de $w_{\eta_{opt}}$ avec 1 (car $\eta_{opt} \simeq 1$ et $w_{\eta_{opt}} \leq 1$), fait que le DSEVI sur $w_{\eta_{opt}}$ n'est généralement pas poussé loin. L'addition d'un compteur permettant de compter les calculs de $w_{\eta_{opt}}$ pour chaque calcul de $w_{P_{M,elec}}$ montre en effet une moyenne de 1,08. Malgré cela, l'utilisation d'une valeur de référence adéquate ajoutée à la faible non-linéarité de la loi de conversion fait que le DSEVI converge très bien, à peu près aussi bien que l'approche affine.

3.4 (*) Calcul de la productivité d'un photobioréacteur

Cet exemple est similaire au précédent. On tient compte d'un couplage non-linéaire entre la production d'une installation photobiochimique et son champ interne de luminance. Cette fois, l'intégration de la loi de couplage ne se fait plus sur le temps mais dans l'espace.

Ici, le nombre de chemins optiques suivis est multiplié par environ 40 [29]. Le besoin en calcul n'est néanmoins plus bloquant pour les ingénieurs intéressés, alors qu'il l'était avant — ou alors, il fallait approximer fortement le modèle.

3.4.1 Problématique

●●● Le problème, d'ordre technique, est celui introduit en début de ce chapitre.

On veut cultiver des micro-organismes photo-synthétiques, afin d'en extraire des substances d'intérêt : biocarburants, nourriture, médicaments, intermédiaires chimiques... Ceci s'inscrit dans une problématique environnementale : en effet, tous ces produits sont aujourd'hui majoritairement issus de l'agriculture conventionnelle ou de la pétrochimie, ces 2 filières étant massivement développées, fonctionnelles, et compétitives. Typiquement, si on veut produire des biocarburants en photobioréacteurs, c'est pour atteindre des rendements ainsi qu'une compétitivité économique supérieurs à ce qu'on obtient avec des cultures en plein champ.

La possibilité théorique d'atteindre des rendements (vis-à-vis de la surface occupée au sol) supérieurs en photobioréacteur qu'en

culture plein champ a déjà été démontrée; ainsi, il serait possible d'augmenter ces rendements de 2 ordres de grandeur en utilisant des photobioréacteurs du type du prototype DiCo-FluV ([29]), qui est étudié dans cette section. L'intérêt est grand en notre période de difficultés liées à l'utilisation des ressources énergétiques fossiles. Les biocarburants ont déjà été testés comme approvisionnement de substitution à celles-ci, avec un succès très mitigé. Pour résumer, leur production requiert beaucoup trop de surface cultivée, et la production d'une quantité d'énergie consommable d'origine biologique nécessite un effort énergétique (pour le travail des sols, la moisson, la transformation...) comparable. Les photobioréacteurs laissent espérer d'éliminer ces blocages : encore faut-il éviter les pièges technologiques susceptibles d'anéantir leurs avantages, et optimiser un tant soit peu leur conception.

●●●● Le concept du réacteur DiCoFluV est dirigé vers une maximisation du rendement puissance lumineuse captée \rightarrow puissance chimique produite. En gros, la lumière solaire est injectée *via* un dispositif extérieur au réacteur en lui-même dans des fibres optiques, qui l'amènent et la redistribuent ensuite dans l'enceinte du réacteur, où les conditions sont contrôlées pour que la photosynthèse soit la plus efficace possible.

Le contrôle des conditions thermochimiques (température, pH, ...) dans le réacteur a fait l'objet de précédents travaux (d'après [29]) ; la difficulté est ici de ne pas consommer trop de puissance en pompage. La grande difficulté restante reste l'éclairage de l'enceinte. Comme affirmé au début de ce chapitre, il doit être le plus uniforme possible dans le volume réactionnel pour maximiser le rendement de la photosynthèse, ce qui est contradictoire avec la nécessité d'absorber tout le rayonnement fourni.

On est là en plein problème d'optimisation. Comme expliqué en début de chapitre, le calcul de la grandeur d'intérêt — la productivité totale du réacteur Q_x — est compréhensible dans le formalisme exposé en section 3.1.2.

3.4.2 Modèle

●●●● On reprend ici complètement le cas d'étude de J. DAUCHET dans sa thèse. Le photobioréacteur est une enceinte cylindrique où est disposé un maillage triangulaire de 979 fibres optiques à éclairage latéral (voir la figure 3.7). Le milieu réactionnel est absorbant, diffusant, uniforme et isotrope.

Le comportement d'un micro-organisme vis-à-vis de la lumière dépend de son espèce. On sait par exemple que les cyanobactéries soit respirent, soit réalisent la photosynthèse, mais ne font en général pas les 2 en même temps. Les micro-algues (eucaryotes) effectuent au contraire ces processus simultanément, avec un couplage entre les 2 ; on se placera pour cette succincte étude dans ce cas. De plus, on peut s'attendre à ce que le comportement des micro-organismes varie avec leur « état » : on supposera ici qu'il n'en est rien. C'est-à-dire que la réaction qui a lieu dans le bioréac-

teur est une réaction auto-catalytique, où les micro-organismes produisent la biomasse dont ils sont eux-même constitués. Cette biomasse leur sert à grandir, ce qui leur permet, une fois qu'ils ont dédoublé leur matériel cellulaire, de se diviser. On supposera que l'absorptivité lumineuse d'une quantité de matière vivante (biomasse, mesurée en masse sèche) ainsi que l'efficacité de la production de biomasse neuve qu'elle réalise grâce à la lumière absorbée ne dépend pas de l'état de la cellule dans laquelle elle est située (en cours de division ou non, par exemple).

Notations : On notera $A = C_x a$, c'est à dire que la vitesse volumique d'absorption des photons A (en $\text{mol.m}^{-3}.\text{s}^{-1}$) est égale à la vitesse spécifique d'absorption des photons a (en $\text{mol.kg}^{-1}.\text{s}^{-1}$) que multiplie la concentration en masse sèche C_x (en kg/m^3). Cela rappelle le concept de section efficace massique : $k_a = C_x \sigma_a$, où k_a est en m^{-1} et σ_a en m^2/kg .

●●●● La quantité de biomasse évolue essentiellement de 2 façons : elle est produite par photosynthèse, et consommée par respiration. On note : $v_x = v_{x,prod} - v_{x,conso}$.

La photosynthèse est considérée comme un processus à saturation. Quand la puissance lumineuse est basse, la lumière est convertie par le biais d'un complexe processus photochimique en biomasse (ou plutôt en énergie chimique) avec un rendement ρ_M . Quand la puissance lumineuse augmente, la production de biomasse approche un maximum. Cela est modélisé par une loi de conversion en fraction rationnelle. Le rendement de conversion ρ (en $\text{kg}_x/\text{mol}_{\text{photons}}$, l'indice x signifiant « biomasse sèche ») s'exprime comme :

$$\rho = \frac{\rho_M}{1 + \frac{a}{a_{0,p}}} = \frac{\rho_M}{1 + \frac{A}{C_x a_{0,p}}} \quad (3.50a)$$

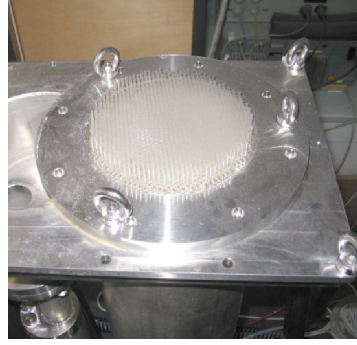
où $a_{0,p}$ est la vitesse spécifique maximale d'absorption utile des photons, en effet

$$v_{x,prod} = \rho A = \frac{\rho_M}{\frac{1}{A} + \frac{1}{C_x a_{0,p}}} = \frac{C_x \rho_M}{\frac{1}{a} + \frac{1}{a_{0,p}}} \quad (3.50b)$$

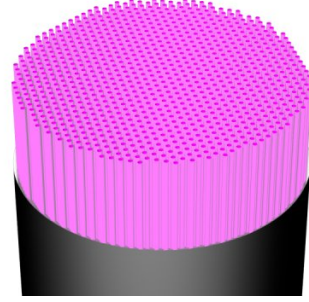
Le couplage entre la photosynthèse, alimentée par la lumière, et la respiration, active en permanence chez les eucaryotes, est encore



(a) Photo d'ensemble du prototype DiCoFluV



(b) Photo de l'entrée des fibres optiques dans le prototype DiCoFluV



(c) Vue en écorchée, par image de synthèse, de l'intérieur du prototype DiCoFluV. Les fibres optiques sont colorisées en mauve.

FIGURE 3.7 – Agencement du prototype de photobioréacteur DiCoFluV, du GEPEA.

Le réacteur se présente extérieurement sous la forme d'une enceinte cylindrique. À l'intérieur est placé un faisceau de 979 fibres optiques éclairant latéralement, disposées selon un maillage triangulaire. Les fibres sont chacune alimentées en lumière par une extrémité, grâce à un dispositif extérieur au réacteur en lui-même.

Crédits : J. DAUCHET

mal compris. Nous considérons un comportement d'inhibition de la respiration par la lumière, modélisé là-encore par une fraction rationnelle. En notant k_c (en s^{-1}) la constante de temps d'autodestruction de la biomasse en l'absence de lumière, cela donne :

$$v_{x,conso} = \frac{C_x k_c}{1 + \frac{a}{a_{0,c}}} = \frac{C_x k_c}{1 + \frac{A}{C_x a_{0,c}}} \quad (3.51)$$

où $a_{0,c}$ est le paramètre caractérisant l'inhibition de la respiration par l'absorption des photons.

Finalement :

$$v_x = \frac{\frac{\rho_M}{A} + \frac{1}{C_x a_{0,p}}}{1 + \frac{A}{C_x a_{0,c}}} \quad (3.52a)$$

$$= C_x \left(\frac{\frac{\rho_M}{A} + \frac{1}{C_x a_{0,p}}}{1 + \frac{A}{C_x a_{0,c}}} \right) \quad (3.52b)$$

On notera pour la suite $A_{0,p} = C_x a_{0,p}$ et $A_{0,c} = C_x a_{0,c}$.

On peut maintenant exprimer la loi de couplage cinétique fv_x , telle que définie à la section 3.1.1, c'est-à-dire telle que $v_x = fv_x(A)$:

$$fv_x : \begin{cases}]0; +\infty[\rightarrow \mathbb{R} \\ A \mapsto C_x \left(\frac{\frac{\rho_M}{A/C_x} + \frac{1}{a_{0,p}}}{1 + \frac{A}{C_x a_{0,c}}} \right) \end{cases} \quad (3.53a)$$

Ainsi :

$$fv_x(A) = \frac{\rho_M A}{1 + \frac{A}{A_{0,p}}} - \frac{C_x k_c}{1 + \frac{A}{A_{0,c}}} \quad (3.53b)$$

3.4.3 Traitement mathématique

●●●● Posons $A_R \in \mathbb{R}^+$, de sorte à exhiber une expression développable en série entière :

$$fv_x(A) = \frac{\rho_M A}{1 + \frac{A - A_R + A_R}{A_{0,p}}} - \frac{C_x k_c}{1 + \frac{A - A_R + A_R}{A_{0,c}}} \quad (3.54a)$$

$$= \frac{\rho_M A A_{0,p}}{A_{0,p} + A_R} \times \frac{1}{1 + \frac{A - A_R}{A_{0,p} + A_R}} - \frac{k_c C_x A_{0,c}}{A_{0,c} + A_R} \times \frac{1}{1 + \frac{A - A_R}{A_{0,c} + A_R}} \quad (3.54b)$$

On utilise le DSE autour de 0 de $x \mapsto \frac{1}{1+x}$:

$$fv_x(A) = \sum_{i=0}^{+\infty} \lambda_i (A - A_R)^i \quad (3.55a)$$

où :

$$\left\{ \begin{array}{l} \lambda_0 = \frac{\rho_M A_R A_{0,p}}{A_R + A_{0,p}} - \frac{C_x k_c A_{0,c}}{A_R + A_{0,c}} \\ \forall i \in \mathbb{N}^*, \lambda_i = (-1)^{i+1} \left(\frac{\rho_M A_{0,p}^2}{(A_R + A_{0,p})^{i+1}} + \frac{C_x k_c A_{0,c}}{(A_R + A_{0,c})^{i+1}} \right) \end{array} \right. \quad (3.55b)$$

La somme précédemment écrite converge ssi :

$$\left| \frac{A - A_R}{A_{0,p} + A_R} \right| < 1 \quad \text{et} \quad \left| \frac{A - A_R}{A_{0,c} + A_R} \right| < 1 \quad (3.56a)$$

ce qui correspond à :

$$A_R > \frac{A - \min\{A_{0,p}; A_{0,c}\}}{2} \quad (3.56b)$$

●●●● Il faut maintenant un estimateur de A pour continuer. Dans sa thèse, J. DAUCHET se base sur l'expression (3.1b) :

$$A(\vec{r}) = \int_{\text{PAR}} d\nu k_a(\vec{r}; \nu) \int_{4\pi} d\vec{u} f_\nu(\vec{r}; \vec{u}; \nu)$$

et sur un estimateur de $f_\nu(\vec{r}; \vec{u}; \nu)$ construit par un suivi de photon inverse. Cet estimateur est justifiable par une forme intégrale de l'équation de transfert radiatif le long d'un rayon, comme montré dans le chapitre précédent 1.2 du présent manuscrit, bien que les justifications données par J. DAUCHET dans sa thèse soient principalement basées sur une image analogue utilisant la réciprocité des chemins. Contrairement à ce qui est proposé dans le présent manuscrit, l'estimateur utilisé est construit en séparant la diffusion et l'absorption : on construit un chemin de diffusion-réflexion pure depuis le point sonde \vec{r} jusqu'à

l'émetteur (obligatoirement une fibre optique), puis la luminance est calculée comme la luminance émise par la fibre et exactement atténuée selon une loi exponentielle tenant compte de l'absorption totale le long du chemin de diffusion-réflexion. Cette technique est rendu possible par les spécificités du cas d'étude Di-CoFluV : le milieu participant est uniforme, et non émissif.

Construire un majorant de A , valable en tout point du réacteur, est une tâche relativement simple. Il suffit de considérer le maximum de k_a selon la fréquence — k_a ne varie en fait pas selon la position —, la luminance émise par les fibres, ainsi que leur réflectivité. Une fois que l'on dispose d'un tel majorant, on peut s'en servir de valeur de référence A_R pour centrer les DSE précédemment montrés de v_x selon A .

Ce majorant a une qualité intéressante : il est aussi majorant des valeurs de l'estimateur W_A de A . Cela permet d'utiliser le choix algorithmique

$$R_j^+ = 1 - \frac{W_A^{j+1}}{A_R} \quad (3.57)$$

où W_A est l'estimateur de A . J. DAUCHET détaille ce choix dans son manuscrit.

●●●● J. DAUCHET effectue aussi dans son manuscrit le calcul de la sensibilité de $Q_x = \int_{\mathcal{V}} d\vec{r} f_{v_x}(A(\vec{r}))$ vis-à-vis de plusieurs paramètres de fonctionnement ou de construction du photobioréacteur DiCoFluV : par exemple la concentration du milieu réactionnel en biomasse sèche, la concentration de la biomasse en pigments photosynthétiques, ou la réflectivité des fibres. Comme expliqué à la section 3.2.3, si on est capable d'estimer Q_x et la sensibilité de A vis-à-vis d'un paramètre, alors on est capable d'estimer la sensibilité de Q_x vis-à-vis de ce paramètre, du moins s'il n'intervient dans la définition de Q_x que via A .

C'est évidemment le cas de la réflectivité des fibres $\rho_{\mathcal{F}}$: elle n'intervient dans l'expression de Q_x qu'en modifiant le champ $f_{\nu}(\vec{r}; \vec{u}; \nu)$ continûment en chaque point. C'est *a priori* différent pour la concentration en biomasse sèche C_x et la teneur en pigments p , car C_x en particulier intervient dans l'expression de f_{v_x} . Cependant, si on exprime v_x selon la vitesse locale *massique* d'absorption des photons $a = A/C_x$, plutôt que selon la vitesse volumique A , cette difficulté est vite élimi-

née, comme en témoigne l'expression (3.52b). On peut en première approximation supposer qu'il en est de même pour la sensibilité $\partial_p f_{\nu}(\vec{r}; \vec{u}; \nu)$.

Les sensibilités de $f_{\nu}(\vec{r}; \vec{u}; \nu)$ selon $\rho_{\mathcal{F}}$, C_x , et p sont calculables simultanément avec $f_{\nu}(\vec{r}; \vec{u}; \nu)$ elle-même par des techniques présentées au chapitre 1.5.5. $\rho_{\mathcal{F}}$ intervient lors de l'interaction avec les fibres optiques, les coefficients d'absorption et de diffusion sont proportionnels à C_x , et p influe sur la fonction de phase de diffusion.

La situation aurait été plus complexe en cas de calcul d'une sensibilité de $Q_x = \int_{\mathcal{V}} d\vec{r} f_{v_x}(A(\vec{r}))$ vis-à-vis d'un paramètre géométrique. On peut alors penser que ce paramètre interviendrait dans l'expression du domaine d'intégration \mathcal{V} . De plus, l'évaluation de la sensibilité de $f_{\nu}(\vec{r}; \vec{u}; \nu)$ selon un tel paramètre nécessite généralement un échantillonnage dans un espace de chemins particulier, construit sur la base d'un *modèle de sensibilité*. Nous renvoyons le lecteur intéressé par plus de détails à ce sujet au manuscrit [29], dont un chapitre entier est consacré à ce sujet.

●●●● Avant de présenter les algorithmes utilisés par J. DAUCHET, quelques points restent à préciser :

- Pour réaliser le calcul des sensibilités, il assemble ses DSEVI d'une manière non-présentée à la section 3.2.3 (mais soluble dans le formalisme de la section 3.2.3, qui est conçu pour être généraliste). Sa technique peut être grossièrement résumée comme suit : d'abord il prend les tirages Z_i effectués pour le calcul de la grandeur Q_x et il les décale en indice, selon $Z_i \rightarrow Z_{i+1}$; de même, les R_i^+ sont décalés selon $R_i^+ \rightarrow R_{i+1}^+$ et $1 \rightarrow R_0^+$; puis il effectue un tirage du poids de sensibilité $Z_{\partial\pi,1}$; enfin il utilise la formulation (3.30).
- Nous avons besoin d'une expression de v_x en fonction de $a = A/C_x$. Cela s'écrit $v_x = f_{v_x}^m(a)$ avec :

$$f_{v_x}^m : \begin{cases}]0; +\infty[\rightarrow \mathbb{R} \\ a \mapsto C_x \left(\frac{\rho_M}{\frac{1}{a} + \frac{1}{a_{0,p}}} - \frac{k_c}{1 + \frac{a}{a_{0,c}}} \right) \end{cases} \quad (3.58)$$

qui admet notamment le DSE $f_{v_x}^m(a) = \sum_{i=0}^{+\infty} \lambda_i^m (a_R - a)^i$, avec :

$$\begin{cases} \lambda_0^m = C_x \left(\frac{\rho_M a_R a_{0,p}}{a_R + a_{0,p}} - \frac{k_c a_{0,c}}{a_R + a_{0,c}} \right) \\ \forall i \in \mathbb{N}^*, \lambda_i^m = -C_x \left(\frac{\rho_M a_{0,p}^2}{(a_R + a_{0,p})^{i+1}} + \frac{k_c a_{0,c}}{(a_R + a_{0,c})^{i+1}} \right) \end{cases} \quad (3.59)$$

Les algorithmes qui découlent finalement de ces considérations sont décrit en 3.3 et 3.4. L'algorithme 3.3 permet de calculer Q_x uniquement, tandis que l'algorithme 3.4 permet de calculer non seulement Q_x mais aussi ses sensibilités selon $\rho_{\mathcal{F}}$, C_x , et p . L'algorithme 3.4 est bien celui utilisé par J. DAUCHET dans le 6^e chapitre de sa thèse.

Résultat : Une estimation $(\overline{w_{Q_x}}; s^2(\overline{w_{Q_x}}))$ de Q_x

```

1  $\Sigma_{Q_x} \leftarrow 0;$ 
2  $\Sigma_{Q_x^2} \leftarrow 0;$ 
3 pour  $i \leftarrow 1$  à  $N_{\text{tirages}}$  faire
4   échantillonner une position uniformément dans le volume réactionnel  $\mathcal{V}$  : on obtient  $\vec{r}_{0\ i};$ 
5    $\Sigma_\lambda \leftarrow 0;$ 
6    $j \leftarrow 0;$ 
7   répéter
8      $\Sigma_\lambda \leftarrow \Sigma_\lambda + \lambda_j^m a_R^j;$  // cf. définition (3.59)
9      $j \leftarrow j + 1;$ 
10    tirer une fréquence sur  $[\nu_{\min}; +\infty[$  selon une densité de probabilité  $p_N$  : on obtient  $\nu_{j,i};$ 
11    tirer une direction de façon isotrope : on obtient  $\vec{u}_{0\ j,i};$ 
12    estimer  $f(\vec{r}_{0\ i}; \vec{u}_{0\ j,i}; \nu_{j,i})$ , grâce à un algorithme de suivi inverse : on obtient  $\tilde{f}(\vec{r}_{0\ i}; \vec{u}_{0\ j,i}; \nu_{j,i});$ 
13     $\tilde{a}_{j,i} \leftarrow \frac{4\pi k_a(\nu_{j,i}) \tilde{f}(\vec{r}_{0\ i}; \vec{u}_{0\ j,i}; \nu_{j,i})}{C_x p_N(\nu_{j,i})};$ 
14     $R_{j-1,i}^+ \leftarrow 1 - \tilde{a}_{j,i}/a_R;$ 
15    échantillonner  $R$  uniformément dans  $[0; 1]$  : on obtient  $r_{j-1,i};$ 
16  jusqu'à  $r_{j-1,i} > R_{j-1,i}^+;$ 
17   $w_{Q_x\ i} \leftarrow V_{\mathcal{V}} \Sigma_\lambda;$  // où  $V_{\mathcal{V}}$  est le volume total du milieu réactionnel
18   $\Sigma_{Q_x} \leftarrow \Sigma_{Q_x} + w_{Q_x\ i};$ 
19   $\Sigma_{Q_x^2} \leftarrow \Sigma_{Q_x^2} + w_{Q_x\ i}^2;$ 
20  $\overline{w_{Q_x}} \leftarrow \Sigma_{Q_x} / N_{\text{tirages}};$ 
21  $s^2(\overline{w_{Q_x}}) \leftarrow (\Sigma_{Q_x^2} / N_{\text{tirages}} - \overline{w_{Q_x}}^2) / (N_{\text{tirages}} - 1);$ 

```

Algorithme 3.3 : Algorithme pour le calcul de Q_x

Résultat : Une estimation $(\overline{w_{Q_x}}; s^2(\overline{w_{Q_x}}))$ de Q_x ,
 une estimation $(\overline{w_{\partial \rho_{\mathcal{F}}, Q_x}}; s^2(\overline{w_{\partial \rho_{\mathcal{F}}, Q_x}}))$ de $\partial_{\rho_{\mathcal{F}}} Q_x$,
 une estimation $(\overline{w_{\partial C_x, Q_x}}; s^2(\overline{w_{\partial C_x, Q_x}}))$ de $\partial_{C_x} Q_x$,
 et une estimation $(\overline{w_{\partial p, Q_x}}; s^2(\overline{w_{\partial p, Q_x}}))$ de $\partial_p Q_x$

- 1 mettre les sommes à 0;
- 2 **pour** $i \leftarrow 1$ à N_{tirages} **faire**
- 3 échantillonner une position uniformément dans le volume réactionnel \mathcal{V} : on obtient $\vec{r}_{0\ i}$;
- 4 tirer une fréquence sur $[\nu_{\min}; +\infty[$ selon une densité de probabilité p_N : on obtient $\nu_{0,i}$;
- 5 tirer une direction de façon isotrope : on obtient $\vec{u}_{0\ 0,i}$;
- 6 estimer les sensibilités de $f(\vec{r}_{0\ i}; \vec{u}_{0\ 0,i}; \nu_{0,i})$ vis-à-vis de $\rho_{\mathcal{F}}$, C_x , et p , grâce à un algorithme en suivi inverse : on obtient respectivement $w_{\partial \rho_{\mathcal{F}}, f\ i}$, $w_{\partial C_x, f\ i}$, et $w_{\partial p, f\ i}$;
- 7 $\Sigma_{\lambda} \leftarrow 0$;
- 8 $\Sigma'_{\lambda} \leftarrow 0$;
- 9 $j \leftarrow 0$;
- 10 **répéter**
- 11 $\Sigma_{\lambda} \leftarrow \Sigma_{\lambda} + \lambda_j^m a_R^j$; // cf. définition (3.59)
- 12 $\Sigma'_{\lambda} \leftarrow \Sigma'_{\lambda} + j \lambda_j^m a_R^j$;
- 13 $j \leftarrow j + 1$;
- 14 tirer une fréquence sur $[\nu_{\min}; +\infty[$ selon une densité de probabilité p_N : on obtient $\nu_{j,i}$;
- 15 tirer une direction de façon isotrope : on obtient $\vec{u}_{0\ j,i}$;
- 16 estimer $f(\vec{r}_{0\ i}; \vec{u}_{0\ j,i}; \nu_{j,i})$, grâce à un algorithme de suivi inverse : on obtient $\tilde{f}(\vec{r}_{0\ i}; \vec{u}_{0\ j,i}; \nu_{j,i})$;
- 17 $\tilde{a}_{j,i} \leftarrow \frac{4\pi k_a(\nu_{j,i}) \tilde{f}(\vec{r}_{0\ i}; \vec{u}_{0\ j,i}; \nu_{j,i})}{C_x p_N(\nu_{j,i})}$;
- 18 $R_{j-1,i}^+ \leftarrow 1 - \tilde{a}_{j,i}/a_R$;
- 19 échantillonner R uniformément dans $[0; 1]$: on obtient $r_{j-1,i}$;
- 20 **jusqu'à** $r_{j-1,i} > R_{j-1,i}^+$;
- 21 $w_{Q_x\ i} \leftarrow V_{\mathcal{V}} \Sigma_{\lambda}$; // où $V_{\mathcal{V}}$ est le volume total du milieu réactionnel
- 22 $w_{\partial \rho_{\mathcal{F}}, Q_x\ i} \leftarrow V_{\mathcal{V}} w_{\partial \rho_{\mathcal{F}}, f\ i} \Sigma'_{\lambda}$;
- 23 $w_{\partial C_x, Q_x\ i} \leftarrow V_{\mathcal{V}} \left(w_{\partial C_x, f\ i} \Sigma'_{\lambda} + \frac{\Sigma_{\lambda}}{C_x} \right)$;
- 24 $w_{\partial p, Q_x\ i} \leftarrow V_{\mathcal{V}} w_{\partial p, f\ i} \Sigma'_{\lambda}$;
- 25 mettre à jour les sommes;
- 26 extraire les résultats des sommes, de façon usuelle;

Algorithme 3.4 : Algorithme pour le calcul simultané de Q_x et de certaines de ses sensibilités

Chapitre 4

Résolution d'une équation différentielle ordinaire non-linéaire d'ordre 1 par une méthode de Monte-Carlo

Ce chapitre est consacré à la résolution, par MMC, d'une équation différentielle non-linéaire, où l'inconnue est à variable et à valeurs réelles. La solution proposée s'appuie sur les ACNs et les DSEVIs tels que présentés aux 2 précédents chapitres.

Il s'agit d'une première étape vers la résolution des équations de Boltzmann non-linéaires. Les non-linéarités des équations différentielles résolues ici peuvent être interprétées comme résultant de collisions inter-particulaires dans un modèle physique sous-jacent (comme dans une approche de cinétique macroscopique), mais il manque pour l'instant l'espace des phases.

4.1 Introduction

●●●● Nous proposons dans ce chapitre une technique de calcul de type Monte-Carlo pour la résolution de certaines équations différentielles ordinaires d'ordre 1. Plus précisément, on considère les systèmes différentiels de la forme :

$$\begin{cases} y'(t) = f(y(t); t) & \forall t \geq 0 \\ y(0) = y_0 & \text{avec } y_0 \in \mathbb{R} \end{cases} \quad (4.1)$$

où y et f sont des fonctions à valeurs réelles. f est supposé Développable en Série Entière vis-à-vis de sa 1^{re} variable. Ainsi, d'après le théorème de Cauchy-Lipschitz, le système différentiel précédent admet une unique solution maximale.

On suppose parfois qu'on dispose d'une solution approchée y^\sim au système (4.1). Cette solution est alors continue et de classe \mathcal{C}^1 par morceaux.

●●●● Cette proposition technique, présentée tout au long du chapitre, constitue uniquement une étape dans la construction d’une proposition similaire applicable aux équations de Boltzmann non-linéaires. Aujourd’hui et depuis longtemps, la résolution d’un système différentiel scalaire du type présenté en équation (4.1) ne pose plus aucune difficulté, en utilisant des méthodes à discrétisation de la variable indépendante (ici t). La nouvelle méthode de type Monte-Carlo proposée ici n’a donc pas vocation à être particulièrement performante sur un tel cas d’étude. D’ailleurs, le lecteur pourra constater, sur les exemples qui seront présentés, qu’elle est de fait nettement moins performante que les méthodes de discrétisation usuelles.

Peu importe, car l’objectif final est la résolution de l’équation de Boltzmann. La méthode de Monte-Carlo deviendra performante, comparée aux méthodes utilisant des discrétisations ou des quadratures, lorsque l’espace des variables indépendantes sera de grande dimension ou lorsque les conditions aux limites seront complexes. Dès que nous nous intéresserons à des problèmes concrets de cinétiques des gaz, l’espace des phases apportera 6 dimensions supplémentaires, et les contraintes aux limites géométriques du domaine pourront prendre tout type de forme (on peut penser notamment aux géométries 3D des systèmes industriels).

●●●● Par ailleurs, le système (4.1) revêt une expression très générique ; ceci est volontaire. Même si le montage d’une solution statistique utilisable en MMC pour l’équation de Boltzmann (en cinétique des gaz) est le fil conducteur de cette thèse, cette dernière équation est pour nous une représentante de toutes les équations de Boltzmann non-linéaires. C’est la raison pour laquelle nous n’avons pas cherché à préciser f et à rendre l’équation d’évolution aussi proche que possible de l’équation de Boltzmann (sans espace des phases toutefois, ce qui resterait une simplification lourde). Nous ne le ferons que lors des tests de mise en œuvre.

Nous fournissons donc d’abord des solutions générales au problème (4.1). Il nous faudra ensuite rajouter l’espace des phases. Cet ajout sera fait dans les chapitres suivants 5 et 6 comme une extension très directe de la proposition du présent chapitre. Mais pour le coup nous le ferons alors spécifiquement pour des cas de cinétique des gaz. Ce chapitre est ainsi le dernier à exhiber un caractère de généralité.

●●●● Les formulations intégrales solutions du système (4.1), que nous allons bientôt détailler, et qui seront utilisables dans le contexte de la méthode de Monte-Carlo, seront basées initialement sur les ACNs présentés au chapitre 2 et feront ensuite appel aux DSEVIs présentés au chapitre 3. Nous amènerons ces formulations dans la section 4.2, dans un premier temps de façon très formelle pour ne laisser aucune zone d’ombre, sachant que nous nous éloignons graduellement de l’intuitif usuel de la physique du transport et que nous ne pouvons plus nous appuyer sur lui pour contrôler la rigueur de nos développements.

Nous donnerons ensuite, en section 4.3, une démarche concrète de résolution. Ce sera l’occasion de tester le comportement des algorithmes, en particulier en termes

de variance et de récursivité. Nous montrerons aussi comment ce comportement peut être anticipé par le calcul symbolique.

Ensuite, nous montrerons en section 4.4 comment la résolution par MMC d'un système différentiel du type (4.1) peut être facilité lorsqu'on dispose, avant de commencer, d'une solution approximative. Cette solution pourrait être calculé par exemple à l'aide d'une méthode à discrétisation.

Ensuite encore, en section 4.5 nous montrerons comment le calcul de sensibilité est aisément compatible avec la démarche mathématique et numérique exposée dans ce chapitre. Notre intérêt pour ce détail, depuis le chapitre 2, vient de ce que nous voulons montrer que toutes les propriétés de la MMC sont utilisables dans la résolution des équations de Boltzmann non-linéaires ; y compris la façon de gérer les calculs de sensibilité tels que demandés de plus en plus fréquemment dans le contexte de l'ingénierie. Ce sera notre dernière digression sur le calcul de sensibilité dans le présent manuscrit.

Ce chapitre est donc à dominante formelle, mais comme la plupart des raisonnements s'étendront immédiatement à la cinétique des gaz, ils ne seront présentés qu'une seule fois ; et cela nous évitera de repasser par des considérations trop techniques dans les chapitres plus illustratifs de la fin de ce mémoire.

4.2 Bases mathématiques de la méthode

Proposition 4.2.1. Soit y^\sim une fonction de $[0; t_f[\subseteq \mathbb{R}^+$ vers \mathbb{R} , continue et de classe \mathcal{C}^1 par morceaux. Soit aussi \hat{k} une fonction de $] -\infty; t_f[$ vers \mathbb{R}^{+*} continue par morceaux et de morceaux continûment prolongeables vers \mathbb{R}^{+*} , telle que $\int_{-\infty}^0 \hat{k}(t) dt = +\infty$. Pour tout $t \in [0; t_f[$, on a :

$$y^\sim(t) = \int_{-\infty}^t dt' \hat{k}(t') \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right) \times \left(H(t' \leq 0) y^\sim(0) + H(t' > 0) \left(y^\sim(t') + \frac{y^{\sim'}(t')}{\hat{k}(t')} \right) \right) \quad (4.2)$$

Preuve. On note y_e^\sim le membre de droite dans l'identité (4.2). Comme \hat{k} ne s'annule pas, y_e^\sim peut être simplifié en :

$$y_e^\sim(t) = \int_{-\infty}^t dt' \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right) \times \left(H(t' \leq 0) \hat{k}(t') y^\sim(0) + H(t' > 0) \left(\hat{k}(t') y^\sim(t') + y^{\sim'}(t') \right) \right)$$

Dans cette expression intégrale de y_e^\sim , l'intégrande est une fonction continue par morceaux et dominée au voisinage de $-\infty$ par une fonction intégrable, donc l'intégrale est bien définie.

On va montrer que $y_e^{\sim'}(t) = y^{\sim'}(t)$ partout où $y^{\sim'}(t)$ existe, et que y_e^{\sim} et y^{\sim} satisfont à la même condition initiale. Comme y_e^{\sim} et y^{\sim} sont continues et de classe \mathcal{C}^1 par morceaux, cela suffit à montrer leur égalité.

En $t = 0$:

$$\begin{aligned} y_e^{\sim}(0) &= \int_{-\infty}^0 dt' \hat{k}(t') \exp\left(-\int_{t'}^0 dt'' \hat{k}(t'')\right) \left(H(t' \leq 0) y^{\sim}(0) + H(t' > 0) \dots\right) \\ &= y^{\sim}(0) \int_{-\infty}^0 dt' \hat{k}(t') \exp\left(-\int_{t'}^0 dt'' \hat{k}(t'')\right) \\ &= y^{\sim}(0) \left[\exp\left(-\int_{t'}^0 dt'' \hat{k}(t'')\right)\right]_{t'=-\infty}^{t'=0} \\ &= y^{\sim}(0) \end{aligned}$$

Par ailleurs, en $t > 0$ tel que $y^{\sim'}(t)$ existe :

$$\begin{aligned} \partial_t y_e^{\sim}(t) &= \partial_t \int_{-\infty}^0 dt' \hat{k}(t') \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right) y^{\sim}(0) + \\ &\quad \partial_t \int_0^t dt' \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right) \left(\hat{k}(t') y^{\sim}(t') + y^{\sim'}(t')\right) \\ &= y^{\sim}(0) \int_{-\infty}^0 dt' \hat{k}(t') \partial_t \left(-\int_{t'}^t dt'' \hat{k}(t'')\right) \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right) + \\ &\quad \int_0^t dt' \partial_t \left(-\int_{t'}^t dt'' \hat{k}(t'')\right) \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right) \left(\hat{k}(t') y^{\sim}(t') + y^{\sim'}(t')\right) + \\ &\quad \hat{k}(t) y^{\sim}(t) + y^{\sim'}(t) \\ &= -\hat{k}(t) y^{\sim}(0) \int_{-\infty}^0 dt' \hat{k}(t') \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right) - \\ &\quad \hat{k}(t) \int_0^t dt' \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right) \left(\hat{k}(t') y^{\sim}(t') + y^{\sim'}(t')\right) + \\ &\quad \hat{k}(t) y^{\sim}(t) + y^{\sim'}(t) \\ &= -\hat{k}(t) y^{\sim}(0) \left[\exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right)\right]_{t'=-\infty}^{t'=0} - \\ &\quad \hat{k}(t) \left[y^{\sim}(t') \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right)\right]_{t'=0}^{t'=t} + \hat{k}(t) y^{\sim}(t) + y^{\sim'}(t) \\ &= -\hat{k}(t) y^{\sim}(0) \exp\left(-\int_0^t dt' \hat{k}(t')\right) - \hat{k}(t) y^{\sim}(t) + \\ &\quad \hat{k}(t) y^{\sim}(0) \exp\left(-\int_0^t dt' \hat{k}(t')\right) + \hat{k}(t) y^{\sim}(t) + y^{\sim'}(t) \\ &= y^{\sim'}(t) \end{aligned}$$

Ceci clôt la démonstration. ■

Proposition 4.2.2. Soit y une fonction de $[0; t_f[\subseteq \mathbb{R}^+$ vers \mathbb{R} , solution du système :

$$\begin{cases} y'(t) = f(y(t); t) & \forall t \in [0; t_f[\\ y(0) = y_0 & y_0 \in \mathbb{R} \end{cases} \quad (4.3)$$

et supposée de classe \mathcal{C}^1 par morceaux. Soit aussi \hat{k} une fonction de $] -\infty; t_f[$ vers \mathbb{R}^{+*} , continue par morceaux et de morceaux continûment prolongeables vers \mathbb{R}^{+*} , telle que $\int_{-\infty}^0 \hat{k}(t) dt = +\infty$. Pour tout $t \in [0; t_f[$, on a :

$$y(t) = \int_{-\infty}^t dt' \hat{k}(t') \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right) \times \left(H(t' \leq 0) y_0 + H(t' > 0) \left(y(t') + \frac{f(y(t'); t')}{\hat{k}(t')} \right) \right) \quad (4.4)$$

Preuve. y est solution du système différentiel (4.3), donc y est continue. Ainsi, y satisfait aux conditions pour y^\approx dans la proposition 4.2.1, de même que \hat{k} . En substituant $f(y(t'); t')$ à $y'(t')$, on obtient le résultat. ■

Proposition 4.2.3. Soit le système différentiel :

$$\begin{cases} y'(t) = f(y(t); t) & \forall t \in [0; t_f[\\ y(0) = y_0 & y_0 \in \mathbb{R} \end{cases} \quad (4.3)^\dagger$$

avec $t_f \in \bar{\mathbb{R}}^+$, ainsi que $\hat{k} :] -\infty; t_f[\rightarrow \mathbb{R}^{+*}$ continue par morceaux et de morceaux continûment prolongeables vers \mathbb{R}^{+*} , et $B : [0; t_f[\rightarrow \mathcal{P}(\mathbb{R})$ répondant aux hypothèses suivantes :

1. $\forall t \in [0; t_f[, (\mathcal{E}(B(t)) \times \{t\}) \subset \mathcal{D}_f$
2. $\forall t \in [0; t_f[, y_0 \in B(t)$
3. $\forall t \in [0; t_f[$, si on dispose d'une variable aléatoire calculable Z à valeurs dans $B(t)$ et d'espérance finie, on peut construire un estimateur calculable non-biaisé convergeant de $\varphi(E(Z); t) = E(Z) + \frac{f(E(Z); t)}{\hat{k}(t)}$ à valeurs dans $\bigcap_{t' \in [t; t_f[} B(t')$. On note $C(t)$ un majorant, sur les VA Z précédemment définies, du nombre moyen de tirages de Z nécessaires pour estimer $\varphi(E(Z); t)$ grâce à cette construction.
4. C est intégrable sur $[0; t_f[$.

Soit y^* définie comme « une solution maximale du système différentiel (4.3), telle que $\forall t \in \mathcal{D}_{y^*}, y^*(t) \in \mathcal{E}(B(t))$ ».

Alors y^* est unique à répondre à cette définition, et satisfait à l'équation intégrale (4.4) sur tout son ensemble de définition. Pour tout $t \in \mathcal{D}_{y^*}$, il existe un estimateur calculable non-biaisé convergent de $y^*(t)$ à valeurs dans $B(t)$.

De plus :

- Si $B(t)$ est borné pour tout $t \in [0; t_f[$, alors y^* est définie sur tout $[0; t_f[$.
- Si $B(t)$ est ouvert pour tout $t \in [0; t_f[$, alors y^* est l'unique solution maximale du système (4.3), en toute généralité.

●●●● La proposition 4.2.2, couplée à la méthodologie expliquée au chapitre 3, fournit une méthode de résolution de type Monte-Carlo pour les équations différentielles étudiées dans ce chapitre. La proposition 4.2.3 donne des conditions suffisantes de fonctionnalité pour une telle méthode de résolution.

L'idée est d'utiliser le principe des Algorithmes à Collisions Nulles — tirer un temps ou une distance de recul selon une extinction arbitraire, puis tenir compte de la variation réelle de y par un mécanisme de compensation utilisant le ratio extinction réelle / extinction arbitraire — encore valable lorsque le coefficient d'extinction est *a priori* inconnu. Mais cette fois-ci, ce coefficient n'est plus simplement une grandeur complexe qu'on préfère estimer que calculer exactement (comme au chapitre 2.5), mais est une fonction de l'inconnue calculée y elle-même. On utilise alors un Développement en Série Entière de Variables Indépendantes pour estimer $f(y; t)$ à partir de l'estimateur récursif de y , qu'on aurait utilisé de toute façon.

Il y a tout de même une différence importante entre la forme algorithmique expliquée ici et un simple ACN : dès que f n'est pas affine envers y , le processus d'estimation de y en remontant le temps ne forme plus un chemin à proprement parler, mais un *arbre*. En effet, à chaque temps intermédiaire t_i de la remontée vers l'instant initial qui permet d'estimer $y(t)$, on doit estimer $y(t_i) + f(y(t_i); t_i) / \hat{k}(t_i)$: cette estimation requière plusieurs estimations indépendantes de $y(t_i)$ si f n'est pas affine en sa 1^{re} variable.

●●●● Afin de se repérer dans un tel arbre d'estimations, il est utile de mettre en place un indicage des estimations de y . Cet indicage doit permettre de situer de manière unique chaque estimation intermédiaire de y faite pour arriver au résultat. On indicera de même les temps t auxquels on remonte pour effectuer chaque estimation.

On s'inspire ici d'une notation utilisée par Ivan T. DIMOV [36, 37, 119]. On indice les estimations intermédiaires de y par des suites à coefficients naturels, selon les règles suivantes :

- Seules sont utilisées les suites $\mathbb{N}^* \rightarrow \mathbb{N}$, stationnaires à 0, dont tous les termes sont nuls à partir du 1^{er} terme nul. On note leur ensemble \mathcal{Z} .
- Le degré de la suite (l'indice de son plus lointain terme non nul) indique à quelle profondeur dans l'arbre de remontée se situe l'estimation repérée, c'est-à-dire combien de tirages de temps de recul fait-on avant d'avoir besoin de cette estimation.
- Parmi les termes non-nuls d'une suite i , le j^e terme désigne une estimation de $y(t_{(n \rightarrow H(n < j) \times i_n)})$ servant à estimer $y(t_{(n \rightarrow H(n < j) \times i_n)}) + f(y(t_{(n \rightarrow H(n < j) \times i_n)}); t_{(n \rightarrow H(n < j) \times i_n)})$. La donnée des numéros des estimations choisies à chaque temps de recul permet de déterminer la branche que l'on suit.

Bien sûr, pour un arbre de remontée en particulier seule une partie des suites indices de \mathcal{Z} est utilisée.

La figure 4.1 donne un exemple de fonctionnement de cet indicage.

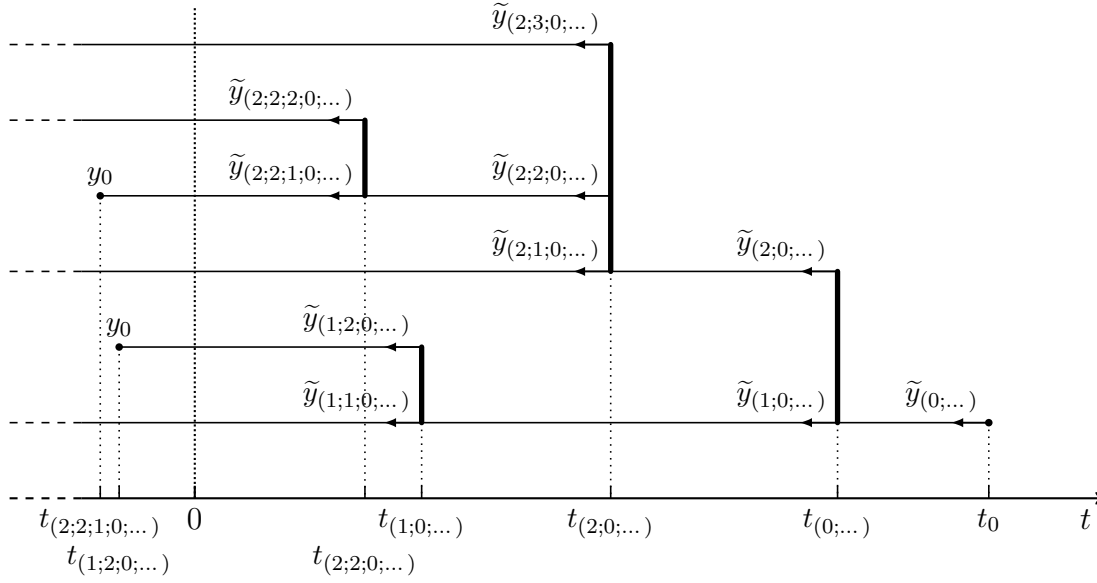


FIGURE 4.1 – Exemple de fonctionnement de l'indigage des branches d'un « arbre d'estimations » servant à l'estimation de $y(t_0)$

Preuve (proposition 4.2.3). Seules les grandes lignes de la preuve sont données ici.

On note φ l'application $(z; t) \mapsto z + \frac{f(z;t)}{\hat{k}(t)}$.

On note W_φ le mécanisme qui permet, à partir d'une VA Z d'espérance finie à valeurs dans $B(t)$, de construire un estimateur de $\varphi(E(Z); t) = E(W_\varphi \circ (Z; t))$. $E(Z) \in \mathcal{E}(B(t))$ donc $\varphi(E(Z); t)$ est bien défini, et le mécanisme W_φ existe par hypothèse.

Considérons la variable aléatoire $\tilde{Y}^\diamond(t)$ définie par :

$$\tilde{Y}^\diamond(t) = H(T' \leq 0) \times y_0 + H(T' > 0) \times W_\varphi \circ (\tilde{Y}^\diamond(T'); T')$$

avec T' une VA à valeurs dans $] -\infty; t]$ et de fonction de répartition $F_{T'} : t' \mapsto \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right)$.

Si $T' \leq 0$, $\tilde{Y}^\diamond(t)$ est calculable et à valeurs dans $B(t)$. Si $T' > 0$, et si $\tilde{Y}^\diamond(T')$ est calculable et à valeurs dans $B(T')$, alors $\tilde{Y}^\diamond(t)$ est calculable et à valeurs dans $B(t)$. Soit alors $rec(t)$ la moyenne du nombre total d'évaluations de $\tilde{Y}^\diamond(t_i)$ à différents t_i à effectuer pour obtenir une valeur de $\tilde{Y}^\diamond(t)$, récursivité comprise : $\tilde{Y}^\diamond(t)$ est calculable ssi $rec(t)$ est finie (voir [55]).

Pour tout $t \in [0; t_f]$, $rec(t)$ est positive et obéit à l'inéquation intégrale :

$$rec(t) \leq \int_{-\infty}^t dt' \hat{k}(t') \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right) \times \left(H(t' \leq 0) \times 1 + H(t' > 0) \left(1 + C(t') rec(t')\right)\right)$$

et donc au système différentiel :

$$\begin{cases} rec(0) = 1 \\ \partial_t rec(t) \leq \hat{k}(t) \left((C(t) - 1) rec(t) + 1 \right) \end{cases}$$

Soit rec^M obéissant au système différentiel :

$$\begin{cases} rec^M(0) = 1 \\ \partial_t rec^M(t) = \hat{k}(t) \left((C(t) - 1) rec^M(t) + 1 \right) \end{cases}$$

Il s'agit de la récursivité rec , dans l'hypothèse où le branchage généré par le mécanisme W_φ est égale au majorant C . Sans surprise, on a $rec^M \geq rec$.

Comme \hat{k} est continue par morceaux de morceaux continûment prolongeables vers \mathbb{R}^{+*} , et C est intégrable sur $[0; t]$, le précédent système admet la solution finie suivante :

$$rec^M(t) = \exp\left(-\int_0^t dt' \hat{k}(t') (C(t') - 1)\right) + \int_0^t dt' \hat{k}(t') \exp\left(-\int_0^{t'} dt'' \hat{k}(t'') (C(t'') - 1)\right)$$

donc $rec(t)$ est finie. Donc $\tilde{Y}^\diamond(t)$ est calculable.

Il s'en suit que chaque réalisation de la variable aléatoire $\tilde{Y}^\diamond(t)$ donnera lieu pour son calcul à un arbre de réalisations — de $\tilde{Y}^\diamond(t')$ avec $t' \in [0; t]$ — *fini*. Par récurrence depuis les feuilles de cet arbre vers sa racine, sur le plus grand intervalle de \mathbb{R}^+ contenant 0 où l'espérance de $\tilde{Y}^\diamond(t)$ existe, cette espérance obéit à l'équation intégrale (4.4) :

$$\begin{aligned} E(\tilde{Y}^\diamond(t)) = \int_{-\infty}^t dt' \hat{k}(t') \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right) \times \\ \left(H(t' \leq 0) y_0 + H(t' > 0) \varphi\left(E(\tilde{Y}^\diamond(t)); t\right) \right) \end{aligned}$$

De plus, quand l'espérance de $\tilde{Y}^\diamond(t)$ n'existe pas, c'est que l'intégrale précédente ne converge pas.

Par dérivation, l'espérance de $\tilde{Y}^\diamond(t)$ est solution du système différentiel (4.3).

L'unicité de l'espérance amène au résultat. $\tilde{Y}^\diamond(t)$ est un estimateur de $y^*(t)$.

Concernant les résultats annexes :

- Si $B(t)$ est borné, $\tilde{Y}^\diamond(t)$ est borné et possède donc une espérance, et donc $y^*(t)$ existe.
- Étant donné ce que l'on vient de montrer, si une solution au système (4.3) autre que y^* existe, lorsqu'elle diffère de y^* elle prend des valeurs hors de $\mathcal{E}(B(t))$. Comme une solution d'un problème différentiel est continue, si $B(t)$ est ouvert alors y^* est bien la seule solution maximale existante au système (4.3) : sinon il y aurait violation du théorème des valeurs intermédiaires. ■

Remarque. La proposition 4.2.3 fournit des conditions d'unicité pour la solution maximale du problème différentiel (4.3) qui sont plus étroites que la condition ordinaire du théorème de Cauchy-Lipschitz. Cette condition-là est que f soit localement lipschitzienne autour de sa 1^{re} variable. Dans la proposition 4.2.3 la condition est que la donnée d'un estimateur de $y(t)$ permet la construction d'un estimateur de $f(y(t); t)$: nous avons dans l'idée qu'une telle construction fasse appel à un Développement en Série Entière de f en sa 1^{re} variable, comme cela est expliqué au chapitre 3 ; or toute fonction développable en série entière est localement lipschitzienne sur le domaine de validité du développement.

Des contre-exemples permettent de montrer comment la méthode proposée peut être mise en défaut, et quels pièges sont à éviter. On peut citer le système suivant :

$$\begin{cases} y'(t) = a y(t)^{2/3} & \forall t \in \mathbb{R}^+ \\ y(0) = 0 \end{cases}$$

Ce système différentiel apparemment bien posé admet en fait un ensemble continu de solutions $\mathcal{S} = \{t \mapsto H(t > t_0)(a(t - t_0)/3)^3, t_0 \in \bar{\mathbb{R}}^+\}$. On peut vérifier qu'il ne satisfait pas à la condition du théorème de Cauchy-Lipschitz, en effet la fonction $y \mapsto y^{2/3}$ n'est pas localement lipschitzienne en 0, or toute solution du système passe en 0 du fait de la condition initiale.

On peut cependant être tenté d'appliquer la technique algorithmique de Monte-Carlo expliquée dans cette section pour estimer $y(t)$ avec $t > 0$. En effet, la fonction $y \mapsto y^{2/3}$ est développable en série entière, autour de tout $y_R > 0$ avec un rayon de convergence y_R . En plus, sur un tel intervalle de convergence la-dite fonction est bornée. Comme la fonction \hat{k} est arbitraire, il n'y a de ce côté pas de limitation. La technique devrait donc fonctionner, ce qui garantirait à la fois l'existence et l'unicité de la solution. Il n'en est en fait rien, car aucun développement en série entière de la fonction $y \mapsto y^{2/3}$ n'est convergent en $y = 0$.

4.3 Illustration sur un cas de cinétique chimique

4.3.1 Le cas d'étude

Nous illustrons maintenant la technique algorithmique développée dans la section précédente sur un cas simple. Ce cas est décrit par le système :

$$\begin{cases} y'(t) = -\alpha y(t)^2 \\ y(0) = y_0 \end{cases} \quad \text{avec} \quad \begin{cases} \alpha > 0 \\ y_0 \geq 0 \end{cases} \quad (4.5)$$

dont l'unique solution sur \mathbb{R}^+ est :

$$y(t) = \frac{y_0}{\alpha y_0 t + 1} \quad (4.6)$$

C'est, par exemple, l'évolution de la concentration en une espèce chimique d'un milieu uniforme, lorsque cette espèce subit une réaction de destruction d'ordre 2.

Ce cas d'étude, malgré sa simplicité, a pour nous suffisamment d'intérêt, à plusieurs titres :

- Bien qu'assez simple pour permettre des études par le calcul symbolique, il est représentatif de comportements fréquents de la méthode développée. Certains d'entre eux sont pathologiques et méritent d'être anticipés...
- L'équation d'évolution est quelque peu proche de l'équation de Boltzmann, dont la résolution numérique est l'objet d'étude de ce manuscrit. En effet, le terme de collision de Boltzmann comporte un terme source et un terme puits, tous deux quadratiques. Le terme puits est proportionnel à la fonction de distribution en le point considéré, alors que le terme source ne fait appel à la fonction de distribution que via une intégration sur l'espace des phases. Dans le cas d'étude de cette section, on a juste un terme puits, quadratique en y . En quelque sorte, il est une simplification grossière de l'équation de Boltzmann, dont on aurait retiré le terme source et l'espace des phases. À ce titre, l'image physique d'une réaction chimique de destruction d'ordre 2 est tout-à-fait justifiée ; l'ordre 2 est une caractéristique cinétique typique des réactions chimiques de destruction par interaction à 2 molécules (par exemple $2\text{O}_3 \rightarrow 3\text{O}_2$), de même selon la mécanique de Boltzmann les particules changent de vitesse en collisionnant 2 à 2.

Une conséquence de cette ressemblance est qu'on peut espérer qu'une partie des éventuels comportements de la méthode numérique que nous allons proposer pour l'équation de Boltzmann soit déjà observable sur ce cas d'étude simple.

4.3.2 Premier algorithme de résolution proposé

●●●● En reprenant ce qui est expliqué dans la section 4.2, et notamment l'équation (4.4), on peut donner une expression intégrale récursive de $y(t)$ dans le cas étudié :

$$y(t) = \int_{-\infty}^t dt' \hat{k}(t') \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right) \times \left(H(t' \leq 0) y_0 + H(t' > 0) \left(y(t') - \frac{\alpha y(t')^2}{\hat{k}(t')} \right) \right) \quad (4.7)$$

$$= \int_{-\infty}^t dt' \hat{k}(t') \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right) \times \left(H(t' \leq 0) y_0 + H(t' > 0) \left(1 - \frac{\alpha y(t')}{\hat{k}(t')} \right) y(t') \right) \quad (4.8)$$

À fins de simplicité, on choisira \hat{k} comme une fonction constante strictement

positive. Ainsi :

$$y(t) = \int_{-\infty}^t dt' \hat{k} \exp(-\hat{k}(t-t')) \left(H(t' \leq 0) y_0 + H(t' > 0) \left(1 - \frac{\alpha y(t')}{\hat{k}} \right) y(t') \right) \quad (4.9)$$

Cette expression intégrale peut être convertie sous forme statistique, en vue de monter un algorithme récursif d'estimation de $y(t)$. Plusieurs expressions statistiques peuvent être proposées, nous choisissons ici la suivante :

$$y(t) = E \left(H(T' \leq 0) y_0 + H(T' > 0) \left(1 - \frac{\alpha \tilde{Y}_1(T')}{\hat{k}} \right) \tilde{Y}_2(T') \right) \quad (4.10)$$

où T' est une VA à valeurs dans $] - \infty; t]$ et de fonction de répartition $F_{T'} : t' \mapsto \exp(-\hat{k}(t-t'))$, et $\tilde{Y}_1(T')$ et $\tilde{Y}_2(T')$ sont 2 estimateurs indépendants de y en T' .

L'algorithme 4.1 découle directement de cette formulation statistique.

Entrées : Le temps t où on veut estimer $y(t)$

Sorties : Une estimation ponctuelle de $y(t)$

- 1 échantillonner $T_{r,ad}$ selon une loi exponentielle d'espérance unité : on obtient $t_{r,ad}$;
- 2 $t' \leftarrow t - \frac{t_{r,ad}}{\hat{k}}$;
- 3 **si** $t' \leq 0$ **alors retourner** y_0 ;
- 4 **sinon**
- 5 estimer $y(t')$ grâce à cet algorithme : on obtient $\tilde{y}_1(t')$;
- 6 estimer à *nouveau* $y(t')$ grâce à cet algorithme : on obtient $\tilde{y}_2(t')$; // de manière indépendante à $\tilde{y}_1(t')$
- 7 **retourner** $\left(1 - \frac{\alpha \tilde{y}_1(t')}{\hat{k}} \right) \tilde{y}_2(t')$;

Algorithme 4.1 : Algorithme pour estimer ponctuellement $y(t)$

●●●● Comme le lecteur peut le constater en lisant l'algorithme 4.1, la façon dont nous l'avons écrit utilise ce qui est appelé la *programmation récursive* : en s'exécutant, l'algorithme 4.1 s'appelle lui-même, et même plusieurs fois. Cette structure de programmation est tout-à-fait valable, pourvu qu'on soit sûr que l'algorithme sera capable de s'exécuter — complètement, avec toutes ses ramifications — avec une espérance de ramification finie ; ceci est en partie l'objet de la proposition 4.2.3. La programmation récursive est naturellement autorisée par plusieurs langages de programmation modernes, comme le C++ que nous avons utilisé.

L'annexe D.2 fournit le code source que nous avons écrit pour tester le cas d'étude ; la partie du code correspondant vraiment à l'algorithme 4.1 est le *bloc*

fonction `Mon_algorithme::algorithme_ponctuel { }` du fichier `algo1.cpp`. Ce code possède la capacité supplémentaire de compter les ramifications des arbres de reconstruction qu'il génère. On peut constater qu'il n'y est nulle part fait référence à une façon d'enregistrer de tels arbres. Comme illustré dans la figure 4.2, le programme ne conserve en mémoire que :

- la branche qui est en cours d'exploration,
- les départs des sous-arbres partant de la branche d'exploration, qui sont encore à explorer,
- les résultats totaux d'estimations intermédiaires des sous-arbres déjà explorés, partant de la branche d'exploration,

le tout étant enregistré automatiquement par la *pile d'exécution*. Cette pile enregistre « où doit-on revenir dans le code quand on a terminé le bloc fonction où on se situe » pour tous les blocs fonctions dans lesquels on est entré successivement, ainsi que l'état (les valeurs des variables) du code appelant quand il appelle un bloc fonction (La pile d'exécution est bien sûr automatiquement déchargée quand un bloc fonction se termine et qu'on retourne au code appelant.). Cette structure permet à notre code d'être relativement économe en ressources informatiques, en plus d'être facile à écrire.

●●●● En C++, le compilateur n'effectue aucune vérification concernant la capacité de l'algorithme à se terminer. En revanche, pendant l'exécution, si la récursivité est potentiellement infinie la pile d'exécution se charge progressivement, jusqu'à saturer en provoquant le crash du programme.

On peut penser que la programmation récursive aurait aussi permis d'écrire les Algorithmes à Collisions Nulles utilisés pour traiter les équations différentielles linéaires, que nous avons présentés au chapitre 2. Il n'y en a en effet aucune impossibilité. Nous n'avons cependant pas retenu cette solution dans ce cas-là. Une raison est que les appels de fonction consomment du temps, pour charger et décharger la pile d'exécution. S'il n'y a pas de multiplication des branches suivies, une simple boucle permet d'effectuer le même travail pour des coûts computationnels moindres.

A priori on aurait pu concevoir un algorithme sans programmation récursive pour traiter le cas étudié ici, aussi performant, en utilisant des boucles, des tests, et une structure de mémorisation adaptée. Nous ne l'avons pas fait pour des questions de simplicité de codage.

4.3.3 Comportement de l'algorithme de résolution proposé : étude par modélisation de la récursivité et de la variance

Deux éléments dans le comportement de l'algorithme 4.1, que nous avons proposé comme générateur d'estimations ponctuelles de la grandeur étudiée $y(t)$, nous

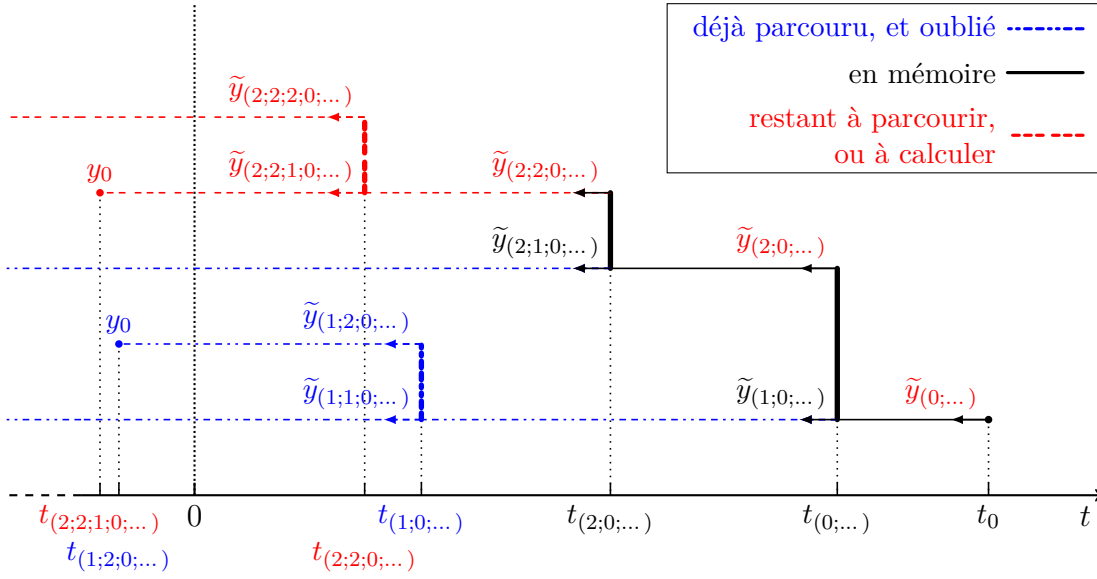


FIGURE 4.2 – Illustration du parcours par l'algorithme 4.1 d'un arbre d'estimations servant à l'estimation de $y(t_0)$. Cette figure reprend la figure 4.1 ; par souci de cohérence avec l'algorithme 4.1, on a retiré une branche.

On suppose ici qu'on vient juste de terminer le calcul de $\tilde{y}_{(2;1;0;\dots)}$; l'étage de calcul de $\tilde{y}_{(2;0;\dots)}$ n'en garde en mémoire que la valeur de $\tilde{y}_{(2;1;0;\dots)}$.

De même, il sait qu'il va devoir calculer $\tilde{y}_{(2;2;0;\dots)}$; mais on ne saura comment qu'en effectuant ce calcul, car la forme des sous-arbres d'estimations à parcourir dépend des valeurs aléatoires qui seront fournies.

À chaque étage de récursion, cette mémorisation est mise en place. De la sorte, l'algorithme sait ce qui lui reste globalement à faire pour finir l'estimation de $y(t_0)$.

intéressent particulièrement : sa récursivité moyenne, et la variance de l'estimateur qu'il génère.

●●●● L'étude de la récursivité générée en moyenne par l'algorithme peut reprendre le travail effectué dans la preuve de la proposition 4.2.3. Nous opterons toutefois ici pour une démarche détaillée, permettant d'expliciter ce précédent travail sur l'exemple étudié.

On note $rec(t)$ l'espérance du nombre de tirages de temps de recul T' nécessaires à une estimation ponctuelle de $y(t)$; un tirage de T' correspond à une traversée de l'intégrale sur t' dans l'expression (4.9).

Nous allons modéliser cette grandeur, c'est-à-dire que nous allons retrouver les équations auxquelles elle obéit. Pour cela, nous entrerons dans une démarche inverse à notre démarche mathématique habituelle visant à construire un algorithme de Monte-Carlo. Nous partirons de l'équation statistique ou intégrale vérifiée par rec , facile à obtenir quand on dispose de l'expression statistique vérifiée par y à partir

de laquelle on monte l'algorithme. Ensuite on dérive l'équation intégrale obtenue pour retrouver une équation différentielle, qui modélise l'évolution de rec . Dans le cas simple que nous étudions, le système différentiel obtenu est assez simple pour être résolu à la main.

Dans le cas où $T' \leq 0$, le suivi en remontée s'arrête, et on a donc besoin que du seul échantillonnage de T' que l'on a fait. Dans le cas inverse, en plus du tirage de T' effectué, il va falloir recommencer l'algorithme 2 fois, mais en partant à présent du temps T' . Cela amène à :

$$rec(t) = \int_{-\infty}^t dt' \hat{k} \exp(-\hat{k}(t-t')) \left(H(t' \leq 0) \times 1 + H(t' > 0) (1 + 2 rec(t')) \right) \quad (4.11)$$

En dérivant l'expression précédente on obtient un système différentiel sur t auquel rec obéit :

$$\begin{cases} rec(0) = 1 \\ \partial_t rec(t) = \hat{k} \times (1 + rec(t)) \end{cases} \quad (4.12)$$

Ce qui donne immédiatement :

$$rec(t) = 2 e^{\hat{k}t} - 1 \quad (4.13)$$

La récursivité de l'algorithme augmente de façon exponentielle avec le temps simulé. Plus le coefficient arbitraire d'extinction \hat{k} est grand, plus cette augmentation exponentielle est rapide.

●●●● La démarche de modélisation utilisée sur la récursivité de l'algorithme peut aussi être employée sur la variance. En fait, on l'appliquera au 2^e moment de la distribution de l'estimateur $\tilde{Y}(t)$, plutôt qu'à sa variance, car les équations intégrales vérifiées par les moments sont plus simples à écrire. Par ailleurs $\text{Var}(\tilde{Y}(t)) = E(\tilde{Y}(t)^2) - E(\tilde{Y}(t))^2 = E(\tilde{Y}(t)^2) - y(t)^2$, avec y connue ou alors grandeur d'intérêt principale, donc si on connaît $E(\tilde{Y}(t)^2)$ on connaît $\text{Var}(\tilde{Y}(t))$ et *vice versa*.

Parfois, cette démarche de modélisation n'est pas le plus simple moyen de connaître la variance de l'estimateur obtenu. On en donnera quelques exemples dans la suite de ce chapitre. Toutefois dans ce cas-ci c'est bien le seul moyen que nous avons trouvé. En fait, cette démarche semble pouvoir faire office de « technique par défaut » quand on ne sait pas obtenir la variance plus simplement (à part par l'expérience numérique, bien entendu).

En reprenant l'équation (4.10), on déduit :

$$\begin{aligned}
 E(\tilde{Y}(t)^2) &= \int_{-\infty}^t dt' \hat{k} \exp(-\hat{k}(t-t')) \left(H(t' \leq 0) y_0^2 + H(t' > 0) \right. \\
 &\quad \left. E\left(\left(1 - \frac{\alpha \tilde{Y}_1(t')}{\hat{k}}\right)^2\right) E(\tilde{Y}_2(t')^2) \right) \\
 &= \int_{-\infty}^t dt' \hat{k} \exp(-\hat{k}(t-t')) \left(H(t' \leq 0) y_0^2 + H(t' > 0) \right. \\
 &\quad \left. \left(1 - \frac{2\alpha E(\tilde{Y}_1(t'))}{\hat{k}} + \frac{\alpha^2 E(\tilde{Y}_1(t')^2)}{\hat{k}^2}\right) E(\tilde{Y}_2(t')^2) \right) \\
 &= \int_{-\infty}^t dt' \hat{k} \exp(-\hat{k}(t-t')) \left(H(t' \leq 0) y_0^2 + H(t' > 0) \right. \\
 &\quad \left. \left(1 - \frac{2\alpha y(t')}{\hat{k}} + \frac{\alpha^2 E(\tilde{Y}(t')^2)}{\hat{k}^2}\right) E(\tilde{Y}(t')^2) \right) \quad (4.14)
 \end{aligned}$$

En dérivant on obtient :

$$\begin{cases} E(\tilde{Y}(0)^2) = y_0^2 \\ \partial_t E(\tilde{Y}(t)^2) = \frac{\alpha^2}{\hat{k}} E(Y(t)^2)^2 - 2\alpha y(t) E(Y(t)^2) \\ \quad = \frac{\alpha^2}{\hat{k}} E(Y(t)^2)^2 - \frac{2\alpha y_0}{\alpha y_0 t + 1} E(Y(t)^2) \end{cases} \quad (4.15)$$

Ce système admet la solution :

$$E(\tilde{Y}(t)^2) = \begin{cases} \frac{y_0^2}{(1 + \alpha y_0 t) \left(1 + \left(1 - \frac{\alpha y_0}{\hat{k}}\right) \alpha y_0 t\right)} & \text{si } \begin{cases} \hat{k} \geq \alpha y_0 \text{ ou } \\ \alpha y_0 t < \frac{1}{\frac{\alpha y_0}{\hat{k}} - 1} \end{cases} \\ +\infty & \text{si } \begin{cases} \hat{k} < \alpha y_0 \text{ et } \\ \alpha y_0 t \geq \frac{1}{\frac{\alpha y_0}{\hat{k}} - 1} \end{cases} \end{cases} \quad (4.16)$$

Ce résultat peut être traduit immédiatement en termes de variance :

$$\text{Var}(\tilde{Y}(t)) = \begin{cases} y_0^2 \frac{\alpha y_0 t}{(1 + \alpha y_0 t)^2 \left(\frac{\hat{k}}{\alpha y_0} + \left(\frac{\hat{k}}{\alpha y_0} - 1\right) \alpha y_0 t\right)} & \text{si } \begin{cases} \hat{k} \geq \alpha y_0 \text{ ou } \\ \alpha y_0 t < \frac{1}{\frac{\alpha y_0}{\hat{k}} - 1} \end{cases} \\ +\infty & \text{si } \begin{cases} \hat{k} < \alpha y_0 \text{ et } \\ \alpha y_0 t \geq \frac{1}{\frac{\alpha y_0}{\hat{k}} - 1} \end{cases} \end{cases} \quad (4.17)$$

On peut en déduire comment la variance de notre algorithme se comporte, d'un point de vue qualitatif. 2 cas de figure se distinguent principalement, selon la valeur du rapport $\frac{\hat{k}}{\alpha y_0}$:

- Si $\frac{\hat{k}}{\alpha y_0} > 1$, la variance relative de l'estimateur $\tilde{Y}(t)$ — c'est-à-dire $\frac{\text{Var}(\tilde{Y}(t))}{\mathbb{E}(\tilde{Y}(t))^2}$ — tend vers la valeur $\frac{\alpha y_0}{\hat{k} - \alpha y_0}$ quand t tend vers l'infini.
- Si $\frac{\hat{k}}{\alpha y_0} < 1$, la variance, relative ou absolue, de l'estimateur $\tilde{Y}(t)$ devient infinie au bout d'un temps simulé t fini ! Le temps minimal au-delà duquel la variance est infinie vaut :

$$t_{echec} = \frac{1}{\alpha y_0} \times \frac{\hat{k}}{\alpha y_0 - \hat{k}} \quad (4.18)$$

●●●● La 1^{re} incidence de ces comportements combinés de la variance d'estimateur et de la récursivité moyenne de construction est sur le réglage de \hat{k} pour améliorer le temps de calcul. Un algorithme sera d'autant plus efficace que le temps de calcul nécessaire pour obtenir une estimation de $y(t)$ avec un écart-type donné (par exemple 1%) sera faible (voir chapitre 1.3.1). Ici, on peut grossièrement considérer que ce temps évolue comme : $t_{1\%} \propto rec \times \text{Var}(\tilde{Y})$.

Étant donné comment évoluent rec et $\text{Var}(\tilde{Y})$ en fonction du temps simulé, on peut en déduire l'attitude suivante : dès que le temps simulé dépasse $\frac{1}{\alpha y_0}$, il vaut mieux baisser \hat{k} autant que possible. Mais cependant, il est indispensable d'assurer que $\hat{k} \geq \alpha y_0$.

●●●● La 2^e chose à remarquer est que l'absence de comportement pathologique de la variance quand $\hat{k} \geq \alpha y_0$ était prévisible, grâce à la proposition 4.2.3.

En effet, le segment $[0; \hat{k}/\alpha]$ est stable par la fonction $(\tilde{Y}_1; \tilde{Y}_2) \mapsto (1 - \alpha \tilde{Y}_1/\hat{k})\tilde{Y}_2$. Dès lors, il semble naturel de choisir $B(t) = [0; \hat{k}/\alpha]$ pour tout t positif dans les hypothèses de la proposition 4.2.3, dès que $y_0 \in [0; \hat{k}/\alpha]$; on peut montrer assez vite que l'on peut compléter toutes les hypothèses de la proposition 4.2.3 dans ce cas et avec ce choix pour B . Ainsi, l'estimateur $\tilde{Y}(t)$ est à valeurs dans un ensemble borné, et sa variance est donc nécessairement finie.

Dans ce cas, il est tentant de parler d'estimateur *récurivement borné*. En effet, si les valeurs de l'estimateur de y sont coincées dans le segment $[0; \hat{k}/\alpha]$ en un point d'un arbre de reconstruction, c'est parce que les valeurs du même estimateur juste un peu plus haut dans l'arbre le sont aussi. C'est un mécanisme récursif.

Quelque chose à remarquer du même ordre, est que ce n'est pas aux temps simulés où un tel bornage récursif n'est pas constructible que l'estimateur exhibe un comportement pathologique. En effet, si \hat{k} est légèrement inférieur à αy_0 , le bornage récursif est *a priori* irréalisable dans les petits temps, pourtant c'est dans les grands temps que le calcul échoue. Cela laisse à penser que des trop grandes valeurs de \tilde{Y} vont être produites dans les petits temps calculés, qui vont ensuite être multipliées entre elles et finalement faire exploser la variance.

4.3.4 Second algorithme de résolution proposé

●●●● Comme affirmé précédemment, la formulation intégrale (4.9) :

$$y(t) = \int_{-\infty}^t dt' \hat{k} \exp(-\hat{k}(t-t')) \left(H(t' \leq 0) y_0 + H(t' > 0) \left(1 - \frac{\alpha y(t')}{\hat{k}} \right) y(t') \right)$$

n'admet pas uniquement l'expression (4.10) comme traduction statistique, même si cette dernière en est sûrement la plus évidente.

Considérons par exemple que comme la solution du système étudié (4.5) est positive et décroissante — ce qui n'est pas difficile à deviner même sans connaître exactement la solution — celle-ci est nécessairement à valeurs dans $[0; y_0]$. En choisissant $\hat{k} \geq \alpha y_0$, on a donc immédiatement

$$\forall t \in \mathbb{R}^+, 0 \leq 1 - \frac{\alpha y(t)}{\hat{k}} \leq 1 \quad (4.19)$$

ce qui permet d'utiliser $1 - \frac{\alpha y(t)}{\hat{k}}$ comme une probabilité. Dans ces conditions, on peut donner à la formulation intégrale (4.9) la traduction statistique :

$$y(t) = E \left(H(T' \leq 0) y_0 + H(T' > 0) H \left(1 - \frac{\alpha \tilde{Y}_1(T')}{\hat{k}} > R \right) \tilde{Y}_2(T') \right) \quad (4.20)$$

où T' est une VA à valeurs dans $] -\infty; t]$ et de fonction de répartition $F_{T'} : t' \mapsto \exp(-\hat{k}(t-t'))$, $\tilde{Y}_1(T')$ et $\tilde{Y}_2(T')$ sont 2 estimateurs de y en T' , avec $\tilde{Y}_1(T')$ à valeurs dans $[0; \hat{k}/\alpha]$, et R est une VA réelle de loi uniforme standard ; $\tilde{Y}_1(T')$, $\tilde{Y}_2(T')$, et R doivent être indépendantes.

Cette expression peut être immédiatement simplifiée en :

$$y(t) = E \left(H(T' \leq 0) y_0 + H(T' > 0) H \left(\frac{\alpha \tilde{Y}_1(T')}{\hat{k}} < R \right) \tilde{Y}_2(T') \right) \quad (4.21)$$

avec les mêmes éléments à l'intérieur. Il en découle l'algorithme 4.2.

Cet algorithme a une caractéristique originale par rapport au précédent algorithme 4.1 : la façon dont il branche dépend des valeurs obtenues pour \tilde{Y} . Cela est illustré par la figure 4.3.

●●●● Il est intéressant, aussi avec ce nouvel algorithme, de prévoir la récursivité moyenne générée ainsi que la variance de l'estimateur sortant.

L'étude de la variance est ici facile. En reprenant l'énoncé de la proposition 4.2.3 avec $B(t) = \{0; y_0\}$, on obtient rapidement que l'estimateur ne peut avoir pour valeurs que 0 ou y_0 . Comme par ailleurs l'espérance de l'estimateur est $y(t) = \frac{y_0}{\alpha y_0 t + 1}$, on obtient :

$$\text{Var}(\tilde{Y}(t)) = y_0^2 \frac{\alpha y_0 t}{(1 + \alpha y_0 t)^2} \quad (4.22)$$

Entrées : Le temps t où on veut estimer $y(t)$
Sorties : Une estimation ponctuelle de $y(t)$

- 1 échantillonner $T_{r,ad}$ selon une loi exponentielle d'espérance unité : on obtient $t_{r,ad}$;
- 2 $t' \leftarrow t - \frac{t_{r,ad}}{\hat{k}}$;
- 3 si $t' \leq 0$ alors retourner y_0 ;
- 4 sinon
 - 5 estimer $y(t')$ grâce à cet algorithme : on obtient $\tilde{y}_1(t')$;
 - 6 échantillonner R de loi uniforme standard : on obtient r ;
 - 7 si $\frac{\alpha \tilde{y}_1(t')}{\hat{k}} < r$ alors
 - 8 estimer à nouveau $y(t')$ grâce à cet algorithme : on obtient $\tilde{y}_2(t')$;
 - 9 retourner $\tilde{y}_2(t')$;
 - 10 sinon retourner 0;

Algorithme 4.2 : Un autre algorithme pour estimer ponctuellement $y(t)$

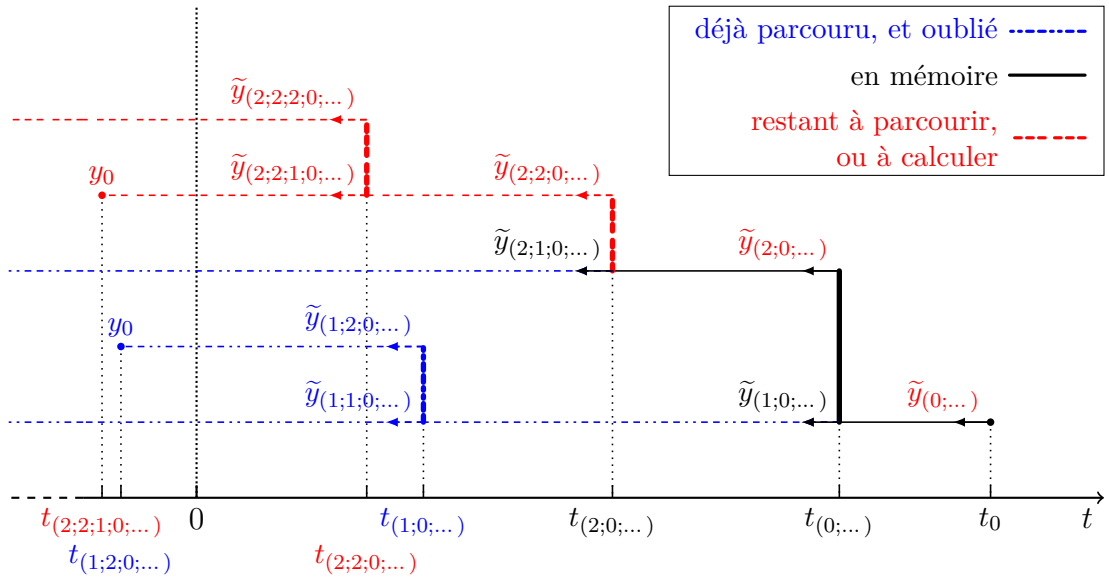


FIGURE 4.3 – Illustration du parcours par l'algorithme 4.2 d'un arbre d'estimations servant à l'estimation de $y(t_0)$, reprise de la figure 4.2.

On suppose ici qu'on vient de calculer $\tilde{y}_{(2;1;0;\dots)}$. La nécessité de calculer ou non un $\tilde{y}_{(2;2;0;\dots)}$ dépend de la valeur de $\tilde{y}_{(2;1;0;\dots)}$ et d'un tirage uniforme standard.

L'étude de la récursivité générée peut reprendre la méthodologie exposée à la section 4.3.3. Cette fois-ci, si $T' > 0$ on ne doit pas forcément recommencer 2 fois la procédure d'estimation. On aboutit à l'équation suivante sur la récursivité :

$$\begin{aligned}
 rec(t) &= \int_{-\infty}^t dt' \hat{k} \exp(-\hat{k}(t-t')) \left(H(t' \leq 0) \times 1 + \right. \\
 &\quad \left. H(t' > 0) \left(1 + rec(t') + \left(1 - \frac{\alpha y_0 / \hat{k}}{\alpha y_0 t + 1} \right) rec(t') \right) \right) \\
 &= \int_{-\infty}^t dt' \hat{k} \exp(-\hat{k}(t-t')) \left(H(t' \leq 0) \times 1 + \right. \\
 &\quad \left. H(t' > 0) \left(1 + \left(2 - \frac{\alpha y_0 / \hat{k}}{\alpha y_0 t + 1} \right) rec(t') \right) \right) \quad (4.23)
 \end{aligned}$$

En dérivant, ou plus simplement en procédant par identification dans l'équation (4.4), on arrive au système différentiel :

$$\begin{cases} \partial_t rec(t) = \hat{k} + \left(\hat{k} - \frac{\alpha y_0}{\alpha y_0 t + 1} \right) rec(t) \\ rec(0) = 1 \end{cases} \quad (4.24)$$

qui admet l'unique solution

$$rec(t) = \frac{1}{\alpha y_0 t + 1} \left(\left(2 + \frac{\alpha y_0}{\hat{k}} \right) e^{\hat{k}t} - \frac{\alpha y_0}{\hat{k}} \right) - 1 \quad (4.25)$$

Concernant l'efficacité globale de l'algorithme 4.2, on peut considérer à nouveau que $t_{1\%} \propto rec \times \text{Var}(\tilde{Y})$. En l'occurrence :

$$rec_{\text{algo 4.2}} \times \text{Var}(\tilde{Y})_{\text{algo 4.2}} = y_0^2 \frac{\alpha y_0 t}{(1 + \alpha y_0 t)^3} \left(\left(2 + \frac{\alpha y_0}{\hat{k}} \right) e^{\hat{k}t} - \alpha y_0 t - 1 - \frac{\alpha y_0}{\hat{k}} \right) \quad (4.26)$$

$$\sim_{t \rightarrow +\infty} \left(2 + \frac{\alpha y_0}{\hat{k}} \right) \times \frac{e^{\hat{k}t}}{(\alpha t)^2} \quad (4.27)$$

Ce résultat peut être comparé avec son équivalent pour l'algorithme 4.1 :

$$rec_{\text{algo 4.1}} \times \text{Var}(\tilde{Y})_{\text{algo 4.1}} = y_0^2 \frac{\alpha y_0 t}{(1 + \alpha y_0 t)^2} \times \frac{2e^{\hat{k}t} - 1}{\frac{\hat{k}}{\alpha y_0} + \left(\frac{\hat{k}}{\alpha y_0} - 1 \right) \alpha y_0 t} \quad (4.28)$$

$$\sim_{t \rightarrow +\infty} \frac{2}{\frac{\hat{k}}{\alpha y_0} - 1} \times \frac{e^{\hat{k}t}}{(\alpha t)^2} \quad (4.29)$$

Plusieurs remarques peuvent être tirées de ces expressions :

- Les 2 algorithmes montés sont aussi performants quand t tend vers l'infini ; c'est-à-dire que l'accroissement du temps simulé t ne rend pas l'un d'eux beaucoup plus efficace que l'autre, au niveau de $t_1\%$.
- Par contre leur comportement se différencie selon \hat{k} . Si $\hat{k} \gg \alpha y_0$, l'algorithme 4.1 est meilleur, car sa plus dense plage de valeurs possibles pour le résultat permet d'en faire diminuer la variance. Si $\hat{k} \simeq \alpha y_0$, l'algorithme 4.2 est plus efficace, car il ne perd pas de temps à parcourir des branches dont le résultat serait finalement peu pris en compte.
- Bien que l'algorithme 4.2 soit meilleur quand $\hat{k} \simeq \alpha y_0$, il faut assurer pour son bon fonctionnement que $\hat{k} \geq \alpha y_0$, comme avec l'algorithme 4.1. Cette fois-ci, ce n'est pas que l'estimateur résultant risque d'avoir une variance infinie, c'est que son espérance serait fausse.

On pourrait être déçu que, après avoir monté l'algorithme 4.2 de façon apparemment plus rusée que l'algorithme 4.1, les performances soient aussi similaires. Peut-être aurions-nous dû être plus inventifs encore : par exemple, on pourrait choisir entre une multiplication (algorithme 4.1) et un test de Bernoulli (algorithme 4.2) selon les valeurs de $1 - \frac{\alpha \tilde{Y}_1(T')}{\hat{k}}$. Globalement, la question de l'optimisation des algorithmes tels ceux présentés jusqu'ici dans ce chapitre reste ouverte.

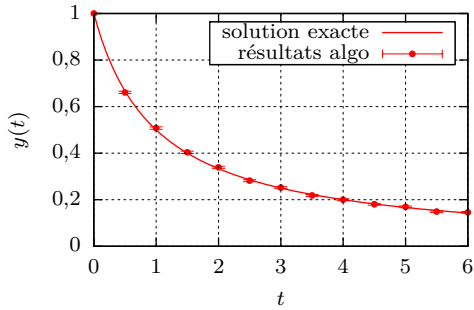
4.3.5 Expériences numériques

●●●● Pour cette campagne de tests, nous fixons les paramètres $\alpha = 1$ et $y_0 = 1$. Nous étudions le comportement des algorithmes 4.1 et 4.2 en fonction du temps final t et du majorant \hat{k} . Les résultats sont présentés dans les figures 4.4, 4.5, 4.6, et 4.7.

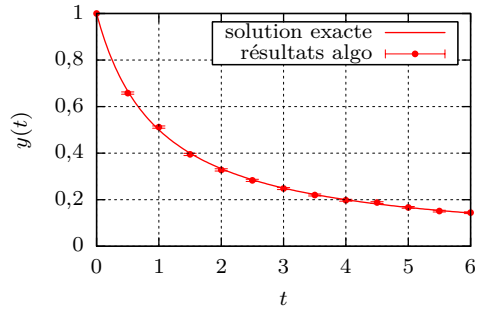
Les résultats ont été obtenus avec 10^4 réalisations pour chaque point représenté — certains points, dessinés sur des graphes différents, sont calculés simultanément. Ils comprennent les estimations obtenues de $y(t)$, avec leurs barres d'erreur, la variance relative des estimateurs utilisés, ainsi que la récursivité moyenne des algorithmes calculant ces estimateurs.

La *récursivité moyenne* est bien celle définie dans la section 4.3.3. Les formules prévisionnelles explicitées en sections 4.3.3 et 4.3.4 sont tracées en trait plein. Les estimations de la récursivité par les algorithmes eux-même sont données en points séparés avec barres d'erreur. Comme annoncé précédemment, les algorithmes possèdent un compteur de leur nombre d'appels par réalisation ; leur implémentation est détaillée en annexe D.2.

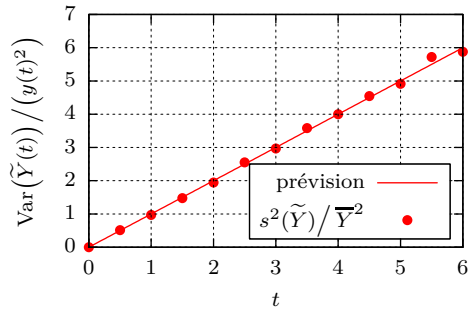
De même, la variance d'estimateur est donnée d'une part grâce aux formules détaillées en sections 4.3.3 et 4.3.4 en trait plein, et d'autre part en points séparés grâce aux résultats des algorithmes. C'est la variance relative qui est tracée, égale au quotient de la variance absolue par le carré de l'espérance. La variance relative « empirique » est calculée comme le quotient de la variance absolue empirique d'échantillon par le carré de la moyenne d'échantillon ; il est important de noter que cet estimateur de la variance relative est biaisé, bien que la variance empirique



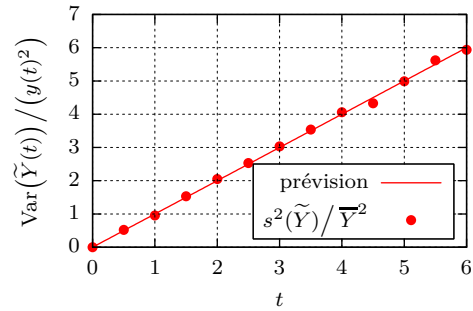
(a) Profil de y selon l'algorithme 4.1



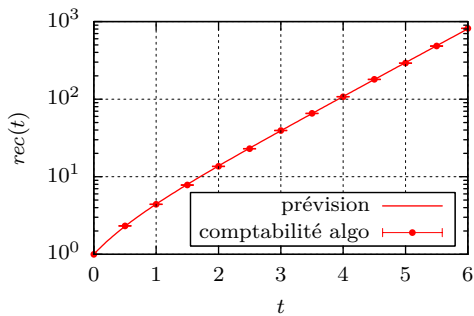
(b) Profil de y selon l'algorithme 4.2



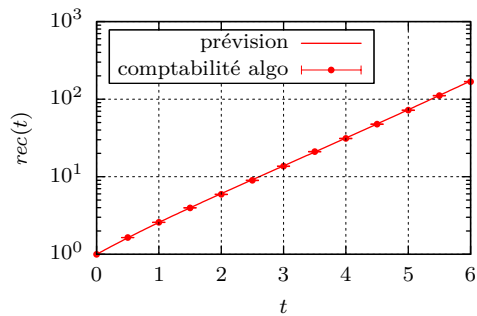
(c) Variance relative de l'estimateur $\tilde{Y}(t)$ calculé par l'algorithme 4.1



(d) Variance relative de l'estimateur $\tilde{Y}(t)$ calculé par l'algorithme 4.2

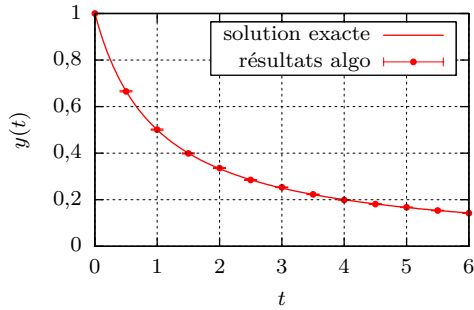


(e) Récursivité moyenne de l'algorithme 4.1 estimant $y(t)$

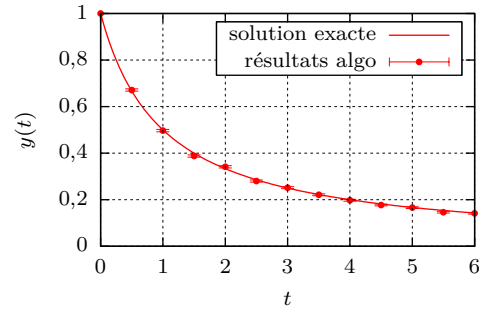


(f) Récursivité moyenne de l'algorithme 4.2 estimant $y(t)$

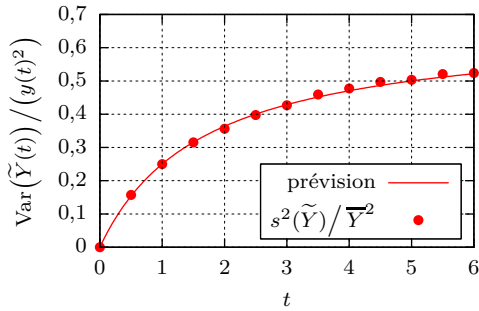
FIGURE 4.4 – Comportements des algorithmes 4.1 et 4.2 estimant $y(t)$, et utilisant $\hat{k} = 1$. En chaque point, 10^4 réalisations ont été effectuées.



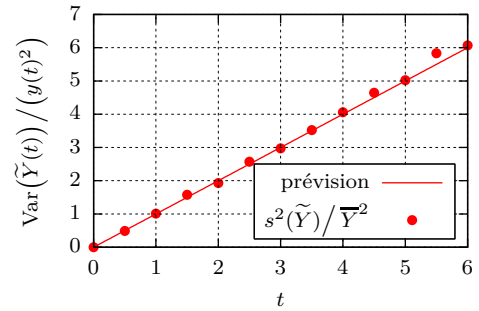
(a) Profil de y selon l'algorithme 4.1



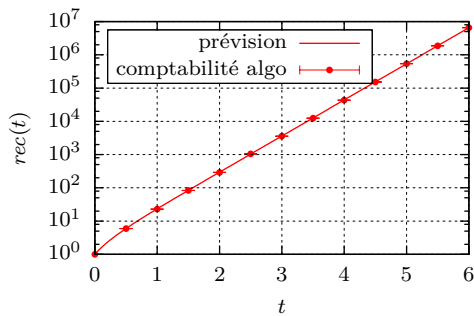
(b) Profil de y selon l'algorithme 4.2



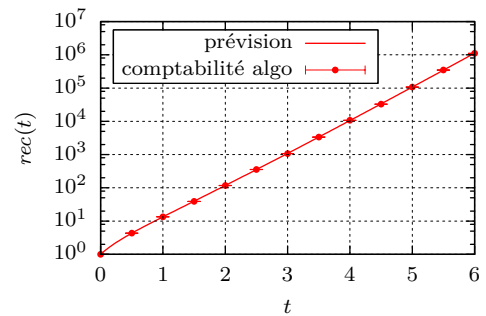
(c) Variance relative de l'estimateur $\tilde{Y}(t)$ calculé par l'algorithme 4.1



(d) Variance relative de l'estimateur $\tilde{Y}(t)$ calculé par l'algorithme 4.2



(e) Récursivité moyenne de l'algorithme 4.1 estimant $y(t)$



(f) Récursivité moyenne de l'algorithme 4.2 estimant $y(t)$

FIGURE 4.5 – Comportements des algorithmes 4.1 et 4.2 estimant $y(t)$, et utilisant $\hat{k} = 2,5$. En chaque point, 10^4 réalisations ont été effectuées.

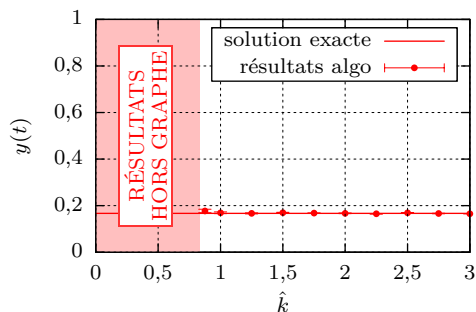
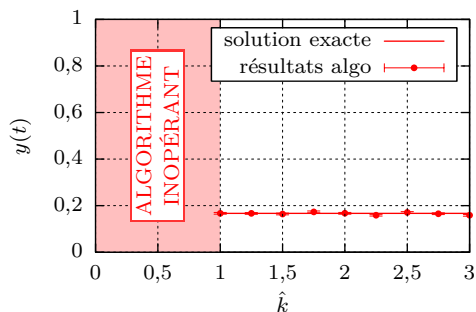
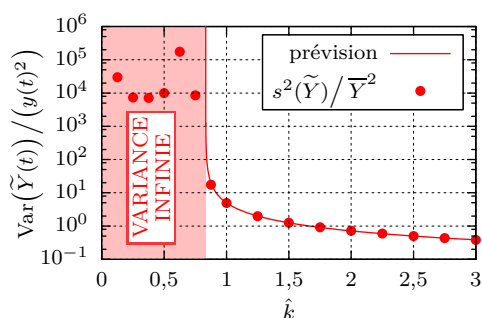
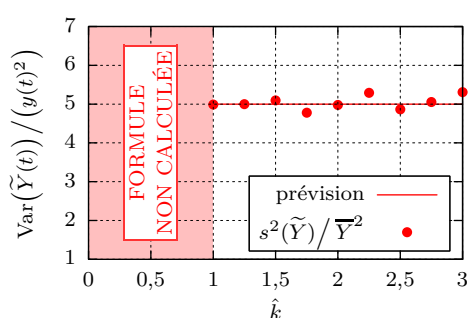
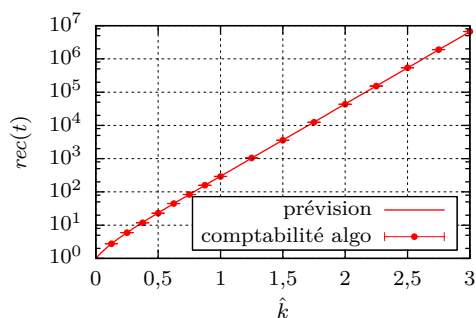
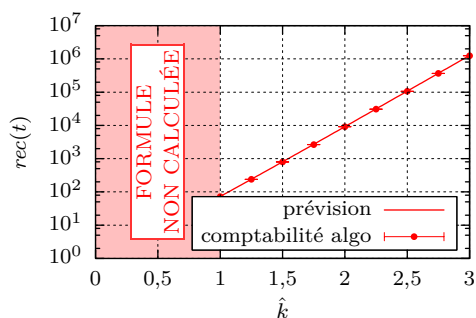
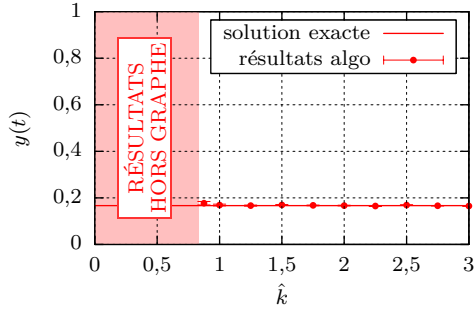
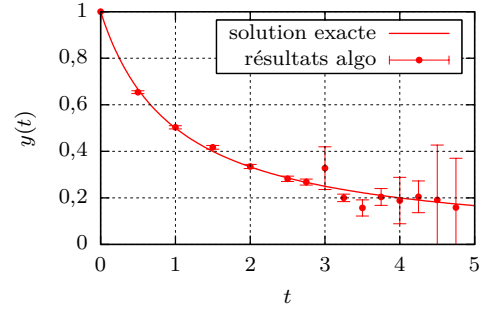

 (a) Estimation de $y(5)$ par l'algorithme 4.1

 (b) Estimation de $y(5)$ par l'algorithme 4.2

 (c) Variance relative de l'estimateur $\tilde{Y}(5)$ calculé par l'algorithme 4.1

 (d) Variance relative de l'estimateur $\tilde{Y}(5)$ calculé par l'algorithme 4.2

 (e) Récursivité moyenne de l'algorithme 4.1 estimant $y(5)$

 (f) Récursivité moyenne de l'algorithme 4.2 estimant $y(5)$

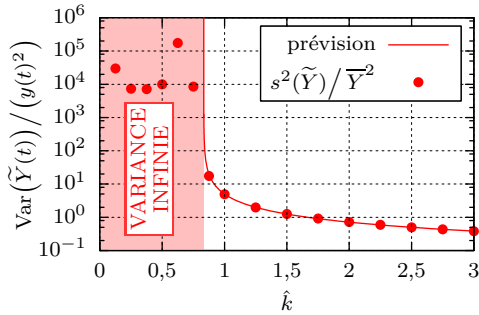
 FIGURE 4.6 – Comportements des algorithmes 4.1 et 4.2 estimant $y(t = 5)$, en fonction du \hat{k} utilisé. En chaque point, 10^4 réalisations ont été effectuées.



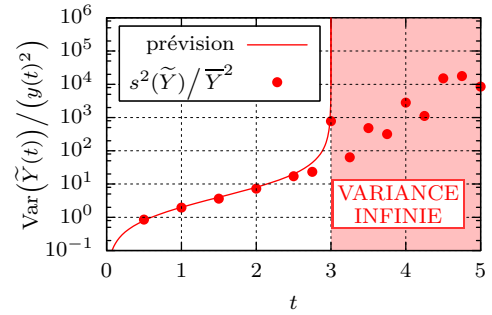
(a) Estimation de $y(5)$ par l'algorithme 4.1, en fonction de \hat{k}



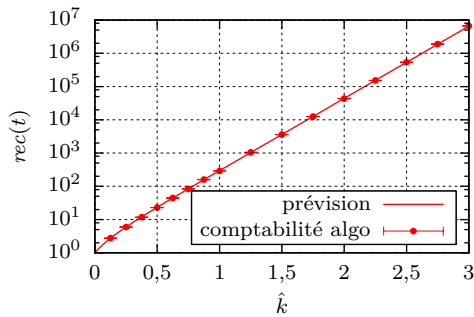
(b) Profil de y selon l'algorithme 4.1, travaillant avec $\hat{k} = 0,75$



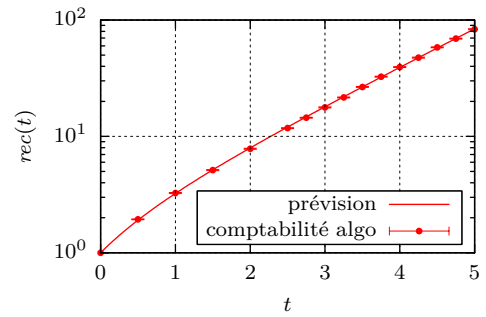
(c) Variance relative de l'estimateur $\tilde{Y}(5)$ calculé par l'algorithme 4.1, en fonction de \hat{k}



(d) Variance relative de l'estimateur $\tilde{Y}(t)$ calculé par l'algorithme 4.1, travaillant avec $\hat{k} = 0,75$



(e) Récursivité moyenne de l'algorithme 4.1 estimant $y(5)$, en fonction de \hat{k}



(f) Récursivité moyenne de l'algorithme 4.2 estimant $y(t)$, travaillant avec $\hat{k} = 0,75$

FIGURE 4.7 – Comportements de l'algorithme 4.1, dans des situations où il n'est pas fonctionnel. En chaque point, 10^4 réalisations ont été effectuées.

d'échantillon et la moyenne d'échantillon soient des estimateurs non-biaisés de la variance et de l'espérance respectivement. Pour estimer sans biais la variance relative, on pourrait remplacer la moyenne d'échantillon par l'espérance : mais il faut bien comprendre que quand on attaquera des problèmes dont les solutions seront inaccessibles symboliquement, on ne disposera pas de meilleur estimateur de la variance relative que celui utilisé ici.

●●●● La 1^{re} chose que l'on peut constater, c'est que les formules prévisionnelles que l'on a données pour la variance et la récursivité sont apparemment justes.

Aussi, il y a confirmation des performances comparées des algorithmes 4.1 et 4.2. Quand \hat{k} est à sa valeur minimale de « bon » comportement des algorithmes ($\hat{k} = \alpha y_0$), l'algorithme 4.2 est le plus efficace, grâce à sa récursivité plus faible. Cela change quand \hat{k} augmente, car la variance de l'estimateur obtenu par l'algorithme 4.1 diminue, ce qui n'est pas le cas avec l'algorithme 4.2.

Il ne faut cependant pas perdre de vue que le temps de calcul est proportionnel et non égal au produit variance \times récursivité ; d'autres facteurs entrent en jeu. Par exemple, si on considère que la grande majorité du temps de calcul est utilisé pour les générations aléatoires, l'algorithme 4.2 est à la base désavantagé d'un facteur presque 2 : en effet, à chaque appel de l'algorithme 4.2 2 tirages aléatoires sont faits (1 seul si on a traversé la condition initiale), contre 1 seulement dans l'algorithme 4.1.

●●●● Ensuite, le comportement des estimateurs de $y(t)$, quand leur variance devient infinie, est inquiétant.

Dans l'étude en fonction de \hat{k} du comportement de l'algorithme 4.1 calculant $y(5)$, on obtient que lorsque la variance doit devenir infinie, la variance relative estimée (avec biais, pour rappel) devient effectivement très élevée. Le résultat est qu'on obtient une barre d'erreur comparable à l'estimation, elle-même de norme très grande (ce n'est pas montré sur les graphes). Ce comportement est connu pour être le signe de la non-convergence d'une méthode de Monte-Carlo, encore que lorsque l'on dispose d'une unique estimation il ne soit pas forcément aisé à reconnaître.

Dans l'étude en fonction de t du comportement de l'algorithme 4.1 utilisant un \hat{k} trop faible, les résultats sont plus gênants. Quand la variance devient infinie, la variance relative estimée augmente progressivement : il est alors difficile de soupçonner le caractère infini de la variance. Le graphe des estimations montre cependant qu'elles ne sont pas très éloignées de la réalité ; mais, dans l'état de cette étude, il est impossible d'affirmer que ce sera toujours le cas.

Cela me fait dire qu'il serait utile :

- d'étudier la convergence d'estimateurs de variance infinie,
- d'essayer l'usage d'indicateurs statistiques supplémentaires au couple moyenne / variance empirique. Je pense en particulier au kurtosis, introduit au chapitre 1.3.3.

Malheureusement, par manque de temps je n'ai pas pu obtenir de résultat là-dessus.

4.4 Utilisation d'une solution approximative calculée *a priori*

On suppose ici qu'on est capable de donner une solution approximative et déterministe du problème différentiel étudié, du type donné en (4.3). Cette solution, notée y^\sim , est continue et de classe \mathcal{C}^1 par morceaux. Elle est précisément connue avant que l'algorithme de Monte-Carlo qui s'en sert comme guide ne démarre.

●●●● Posons maintenant $y^\Delta = y - y^\sim$. Comme y^\sim est connue, la résolution d'un problème en y^Δ est équivalente à la résolution d'un problème en y .

En soustrayant les équations (4.4) et (4.2), on obtient une équation intégrale à laquelle satisfait y^Δ :

$$y^\Delta(t) = \int_{-\infty}^t dt' \hat{k}(t') \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right) \left(H(t' \leq 0) y^\Delta(0) + H(t' > 0) \left(y^\Delta(t') + \frac{f(y^\Delta(t'); t') + y^\sim(t'); t') - f(y^\sim(t'); t')}{\hat{k}(t')} + \frac{f(y^\sim(t'); t') - y^{\sim'}(t')}{\hat{k}(t')} \right) \right) \quad (4.30)$$

Parmi les termes que multiplie l'exponentielle décroissante dans cette équation :

1. $y^\Delta(0)$ est connu, comme y_0 en (4.4).
2. $y^\Delta(t')$ est à estimer récursivement, comme $y(t')$ en (4.4).
3. $\frac{f(y^\Delta(t'); t') + y^\sim(t'); t') - f(y^\sim(t'); t')}{\hat{k}(t')}$ est à estimer récursivement, à l'instar de $\frac{f(y(t'); t')}{\hat{k}(t')}$ en (4.4).
4. $\frac{f(y^\sim(t'); t') - y^{\sim'}(t')}{\hat{k}(t')}$ est un terme nouveau par rapport à l'équation (4.4), mais il est calculable immédiatement.

Si y^\sim est proche de y , l'estimation de $y^\Delta(t)$ grâce à l'expression (4.30) sera certainement plus facile que l'estimation directe de $y(t)$ grâce à l'expression (4.4). En effet si y^\sim est proche de y :

- Le taux de non-linéarité du 3^e terme vis-à-vis de y^Δ sera moindre qu'en (4.4), donc pour obtenir des estimations intermédiaires satisfaisantes de ce terme il sera nécessaire de monter en moyenne moins loin dans les ordres des Développements en Série Entière de f sur sa 1^{re} variable. Comme la récursivité multiplicative de l'algorithme est due à ce terme, et qu'elle est d'autant plus multiplicative qu'il faut monter loin dans les ordres des DSE de f , la récursivité devrait chuter si y^\sim est bien choisie.
- Le 1^{er} et le 4^e terme seront proches de 0. Récursivement, le 2^e et le 3^e devraient l'être aussi, ainsi que l'estimateur final de $y^\Delta(t)$ dont la variance devrait

donc être amoindrie. C'est d'autant plus appréciable, par rapport au cas d'un système différentiel linéaire, que des estimations multiples de y sont multipliées pour obtenir le terme récursif non-linéaire, conduisant à une auto-amplification de la variance (parfois jusqu'à une divergence pathologique, comme cela est illustré en sections 4.3.3 et 4.3.5).

Une situation montrant ces dernières considérations poussées au maximum est le cas idéal $y^\approx = y$: dans ce cas, il est aisé de construire depuis l'expression (4.30) un estimateur de y^Δ valant systématiquement 0.

●●●● La disponibilité d'une solution approximative y^\approx a un autre intérêt : cette solution peut servir de référence autour de laquelle effectuer les DSEs de f . Avec l'exemple de cinétique chimique qui nous sert d'illustration dans ce chapitre ça n'est pas sensible, car nous avons choisi f polynomiale : il n'y a donc ici pas de question qui se pose concernant la convergence d'un Développement en Série Entière. Dans des cas non explorés dans ce manuscrit, l'utilisation d'une solution approximative semble donc pouvoir n'être pas seulement une question de temps de calcul, mais aussi carrément une question de faisabilité du calcul.

Cela ne remet pas en cause la proposition 4.2.3. Au contraire, il s'agit plutôt de trouver comment satisfaire à ses hypothèses, en particulier l'existence d'un estimateur de $f(y(t); t)$.

4.5 Calcul de sensibilités paramétriques

On explique brièvement dans cette section comment le calcul d'une sensibilité paramétrique de $y(t)$ peut-être mené dans un problème du type posé. 2 types de sensibilités se distinguent :

- La sensibilité à un paramètre π dont dépendent f ou y_0 ,
- La sensibilité au temps t . La démarche de calcul dans ce cas est immédiate, car par hypothèse $\partial_t y(t) = f(y(t); t)$.

●●●● Intéressons-nous donc au 1^{er} cas de figure. Remplaçons dans les écritures du problème $y(t)$ par $y(t; \pi)$, $f(y(t); t)$ par $f(y(t; \pi); t; \pi)$, et y_0 par $y_0(\pi)$. Par dérivation, on obtient depuis l'équation (4.4) :

$$\begin{aligned} \partial_\pi y(t; \pi) = \int_{-\infty}^t dt' \hat{k}(t') \exp\left(-\int_{t'}^t dt'' \hat{k}(t'')\right) & \left(H(t' \leq 0) \partial_\pi y_0(\pi) + H(t' > 0) \right. \\ & \left. \left(\partial_\pi y(t'; \pi) \left(1 + \frac{\partial_1 f(y(t'; \pi); t'; \pi)}{\hat{k}(t')} \right) + \frac{\partial_3 f(y(t'; \pi); t'; \pi)}{\hat{k}(t')} \right) \right) \end{aligned} \quad (4.31)$$

Plusieurs remarques sont à faire sur cette précédente écriture :

- Cette écriture intégrale est récursive en $\partial_\pi y$, mais cette récursivité est linéaire. Cela veut dire qu'il n'est pas nécessaire d'estimer $\partial_\pi y$ le long d'un arbre de reconstruction pour estimer $\partial_\pi y(t)$; cette reconstruction ne nécessite qu'un simple chemin, sans ramification.
- Néanmoins, à chaque étape de cette récursion en chemin on a besoin d'estimer $\partial_1 f(y(t_i; \pi); t_i; \pi)$. Si f n'est pas linéaire en sa 1^{re} variable cette sensibilité est bien à estimer récursivement, en estimant $y(t_i; \pi)$. Cela signifie que l'algorithme de calcul de sensibilité doit être épaulé par l'algorithme de calcul de la grandeur pour pouvoir fonctionner.
- Nous avons montré au chapitre 3.2.3 que les estimations de $f(y(t_i; \pi); t_i; \pi)$ et de $\partial_1 f(y(t_i; \pi); t_i; \pi)$ pouvait être réalisées simultanément, en utilisant les mêmes échantillons de $\tilde{Y}(t_i; \pi)$. Détail supplémentaire, l'estimation de la sensibilité peut consommer systématiquement 1 tirage de $\tilde{Y}(t_i; \pi)$ de moins que l'estimation de la grandeur, ce qui laisse une construction de sous-arbre indépendante pour estimer $\partial_\pi y(t_i; \pi)$. La conséquence est que, à moins d'une subtilité dans le terme $\partial_3 f(y(t'; \pi); t'; \pi)$, il n'est pas nécessaire d'étoffer les arbres de reconstruction pour estimer une sensibilité de la grandeur d'intérêt.
- Un développement en série entière conserve son rayon de convergence par dérivation. Donc si l'algorithme récursif d'estimation de la grandeur fonctionne, il peut aussi estimer la sensibilité (il peut y avoir exception si on est au bord du domaine de convergence) (encore une fois à moins d'une subtilité dans le terme $\partial_3 f(y(t'; \pi); t'; \pi)$). Ajoutons qu'il ne peut pas y avoir d'explosion de la variance dans la partie sensibilité de l'algorithme, car la récursivité en la sensibilité est linéaire.

●●●● Nous n'étudierons pas spécialement comment le calcul de sensibilités paramétriques peut être facilité par l'usage d'une solution approximative comme référence. Il y a plusieurs raisons à cela :

- Si l'on ne dispose pas d'un modèle de sensibilité sur y^\approx (c'est-à-dire de $\partial_\pi y^\approx$), la dérivation par rapport à π de l'équation (4.30) amène à l'équation (4.31), sans nouveauté.
- Même si l'on dispose d'un tel modèle, on n'obtiendra aucune diminution de la récursivité, et apparemment assez peu de diminution de la variance car celle-ci ne s'amplifie pas à la base du fait de la multiplication d'estimations différentes.

En fait, une solution approximative peut bel et bien être une aide au calcul de sensibilité... essentiellement en ce qu'elle est une aide au calcul de la grandeur, qui est utilisé par le calcul de sensibilité.

4.6 Illustration sur un cas de cinétique chimique : suite et fin

4.6.1 Sensibilités paramétriques

4.6.1.1 Mise en place des calculs

●●●● On reprend l'exemple de la section 4.3 : on considère le système :

$$\begin{cases} y'(t) = -\alpha y(t)^2 \\ y(0) = y_0 \end{cases} \quad \text{avec} \quad \begin{cases} \alpha > 0 \\ y_0 \geq 0 \end{cases} \quad (4.5)^\dagger$$

dont l'unique solution sur \mathbb{R}^+ est :

$$y(t) = \frac{y_0}{\alpha y_0 t + 1} \quad (4.6)^\dagger$$

On va calculer les sensibilités de $y(t)$ vis-à-vis des paramètres α et y_0 . Les résultats seront comparés aux solutions facilement obtensibles symboliquement :

$$\partial_\alpha y(t) = \frac{-y_0^2 t}{(\alpha y_0 t + 1)^2} \quad \partial_{y_0} y(t) = \frac{1}{(\alpha y_0 t + 1)^2} \quad (4.32)$$

●●●● Pour obtenir les équations intégrales vérifiées par $\partial_\alpha y(t)$ et $\partial_{y_0} y(t)$, il suffit de dériver l'équation intégrale (4.9) vérifiée par $y(t)$ (on garde \hat{k} constant) :

$$\partial_\alpha y(t) = \int_{-\infty}^t dt' \hat{k} \exp(-\hat{k}(t-t')) H(t' > 0) \frac{\partial_\alpha y(t') (\hat{k} - 2\alpha y(t')) - y(t')^2}{\hat{k}} \quad (4.33)$$

$$\begin{aligned} \partial_{y_0} y(t) = \int_{-\infty}^t dt' \hat{k} \exp(-\hat{k}(t-t')) & \left(H(t' \leq 0) \times 1 + \right. \\ & \left. H(t' > 0) \frac{\partial_{y_0} y(t') (\hat{k} - 2\alpha y(t'))}{\hat{k}} \right) \end{aligned} \quad (4.34)$$

Ces équations intégrales peuvent être immédiatement reformulés statistiquement :

$$\partial_\alpha y(t) = E \left(H(T' > 0) \frac{\widetilde{\partial_\alpha Y_1}(T') (\hat{k} - 2\alpha \widetilde{Y_2}(T')) - \widetilde{Y_1}(T') \widetilde{Y_2}(T')}{\hat{k}} \right) \quad (4.35)$$

$$\partial_{y_0} y(t) = E \left(H(T' \leq 0) \times 1 + H(T' > 0) \widetilde{\partial_{y_0} Y_1}(T') \left(1 - \frac{2\alpha \widetilde{Y_2}(T')}{\hat{k}} \right) \right) \quad (4.36)$$

où :

- T' est une VA à valeurs dans $] -\infty; t]$ et de fonction de répartition $F_{T'} : t' \mapsto \exp(-\hat{k}(t - t'))$,
- $\widetilde{Y}(T')$, $\widetilde{\partial_\alpha Y}(T')$, et $\widetilde{\partial_{y_0} Y}(T')$ sont des estimateurs de $y(T')$, $\partial_\alpha y(T')$ et $\partial_{y_0} y(T')$ respectivement,
- la notation « $\bullet_1 \dots \bullet_2$ » signifie comme précédemment l'indépendance de 2 estimateurs.

L'algorithme 4.3, qui estime simultanément $y(t)$, $\partial_\alpha y(t)$, et $\partial_{y_0} y(t)$, découle directement de ces formulations statistiques ainsi que de l'algorithme 4.1.

Entrées : Le temps t où on veut estimer $(y(t); \partial_\alpha y(t); \partial_{y_0} y(t))$

Sorties : Une estimation ponctuelle de $(y(t); \partial_\alpha y(t); \partial_{y_0} y(t))$

- 1 échantillonner $T_{r, ad}$ selon une loi exponentielle d'espérance unité : on obtient $t_{r, ad}$;
- 2 $t' \leftarrow t - \frac{t_{r, ad}}{\hat{k}}$;
- 3 **si** $t' \leq 0$ **alors retourner** $(y_0; 0; 1)$;
- 4 **sinon**
- 5 estimer $(y(t'); \partial_\alpha y(t'); \partial_{y_0} y(t'))$ grâce à cet algorithme : on obtient $(\widetilde{y}_1(t'); \widetilde{\partial_\alpha y_1}(t'); \widetilde{\partial_{y_0} y_1}(t'))$;
- 6 estimer à *nouveau* $y(t')$, grâce à l'algorithme 4.1 : on obtient $\widetilde{y}_2(t')$; // de manière indépendante à $(\widetilde{y}_1(t'); \widetilde{\partial_\alpha y_1}(t'); \widetilde{\partial_{y_0} y_1}(t'))$
- 7 **retourner** $\left(\left(1 - \frac{\alpha \widetilde{y}_1(t')}{\hat{k}} \right) \widetilde{y}_2(t'); \right.$
 $\left. \frac{\widetilde{\partial_\alpha y_1}(t') (\hat{k} - 2 \alpha \widetilde{y}_2(t')) - \widetilde{y}_1(t') \widetilde{y}_2(t')}{\hat{k}}; \widetilde{\partial_{y_0} y_1}(t') \left(1 - \frac{2 \alpha \widetilde{y}_2(t')}{\hat{k}} \right) \right)$;

Algorithme 4.3 : Algorithme pour estimer ponctuellement $y(t)$, $\partial_\alpha y(t)$, et $\partial_{y_0} y(t)$

●●●● Cette fois-ci, il ne semble *a priori* pas simple de mettre en place un test permettant de ne pas évaluer une partie de l'arbre d'estimations parcouru par l'algorithme 4.3, comme on l'a fait dans la section 4.3.4 ; du moins pas là où on veut estimer $y(t)$, $\partial_\alpha y(t)$, et $\partial_{y_0} y(t)$ simultanément. Le terme $1 - \frac{2 \alpha \widetilde{y}_2(t')}{\hat{k}}$ peut peut-être être borné récursivement, dans l'intervalle $[-1; 1]$; mais il semble difficile de rendre compatible un test basé sur la valeur de ce terme avec un « bon » comportement de l'estimation de $y(t)$. De plus, le terme en $\frac{\widetilde{y}_1(t') \widetilde{y}_2(t')}{\hat{k}}$ dans $\widetilde{\partial_\alpha y}(t')$ paraît lui-aussi peu compatible avec un tel test. Pour ces raisons, nous n'essayons pas de modifier les formulations statistiques (4.10), (4.35), et (4.36) de sorte à exhiber une épreuve de Bernoulli commune.

Cependant, cela n'empêche pas d'« élaguer » massivement l'arbre d'estimations parcouru par l'algorithme 4.3, car comme annoncé dans la section 4.5 l'estimation des sensibilités de $y(t)$ n'est nécessaire que le long *d'une seule branche* de cet arbre pour pouvoir aboutir — et une lecture attentive de l'algorithme 4.3 le confirme. Tout le reste de l'arbre d'estimation parcouru ne fait que prodiguer des estimations indépendantes de $y(t)$ en différents temps t : on a déjà montré, avec la formulation (4.21) et l'algorithme 4.2, que cette partie de l'arbre d'estimations peut être appauvrie facilement.

On fournit donc maintenant l'algorithme 4.4 pour le calcul simultané de $y(t)$ et de ses sensibilités vis-à-vis de α et de y_0 . Il ne diffère de l'algorithme 4.3 que par l'utilisation de l'algorithme 4.2 en lieu et place de l'algorithme 4.1, pour fournir une 2^e estimation de $y(T')$.

Entrées : Le temps t où on veut estimer $(y(t); \partial_\alpha y(t); \partial_{y_0} y(t))$

Sorties : Une estimation ponctuelle de $(y(t); \partial_\alpha y(t); \partial_{y_0} y(t))$

- 1 échantillonner $T_{r,ad}$ selon une loi exponentielle d'espérance unité : on obtient $t_{r,ad}$;
- 2 $t' \leftarrow t - \frac{t_{r,ad}}{\hat{k}}$;
- 3 **si** $t' \leq 0$ **alors retourner** $(y_0; 0; 1)$;
- 4 **sinon**
- 5 estimer $(y(t'); \partial_\alpha y(t'); \partial_{y_0} y(t'))$ grâce à cet algorithme : on obtient $(\tilde{y}_1(t'); \widetilde{\partial_\alpha y_1}(t'); \widetilde{\partial_{y_0} y_1}(t'))$;
- 6 estimer à *nouveau* $y(t')$, grâce à l'algorithme 4.2 : on obtient $\tilde{y}_2(t')$; // de manière indépendante à $(\tilde{y}_1(t'); \widetilde{\partial_\alpha y_1}(t'); \widetilde{\partial_{y_0} y_1}(t'))$
- 7 **retourner** $\left(\left(1 - \frac{\alpha \tilde{y}_1(t')}{\hat{k}} \right) \tilde{y}_2(t'); \frac{\widetilde{\partial_{y_0} y_1}(t') (\hat{k} - 2\alpha \tilde{y}_2(t')) - \tilde{y}_1(t') \tilde{y}_2(t')}{\hat{k}}; \widetilde{\partial_\alpha y_1}(t') \left(1 - \frac{2\alpha \tilde{y}_2(t')}{\hat{k}} \right) \right)$;

Algorithme 4.4 : Autre algorithme pour estimer ponctuellement $y(t)$, $\partial_\alpha y(t)$, et $\partial_{y_0} y(t)$

4.6.1.2 Résultats des calculs

Nous menons une 2^e campagne de tests très semblable à la 1^{re} : on prend les paramètres $\alpha = 1$ et $y_0 = 1$, et on effectue en chaque point de calcul 10^4 réalisations. Nous avons aussi choisi de faire évoluer t et \hat{k} parmi les mêmes valeurs qu'en section 4.3.5. L'implémentation détaillée des algorithmes est donnée en annexe D.2, et les résultats sont présentés dans les figures 4.8, 4.9, 4.10, et 4.11.

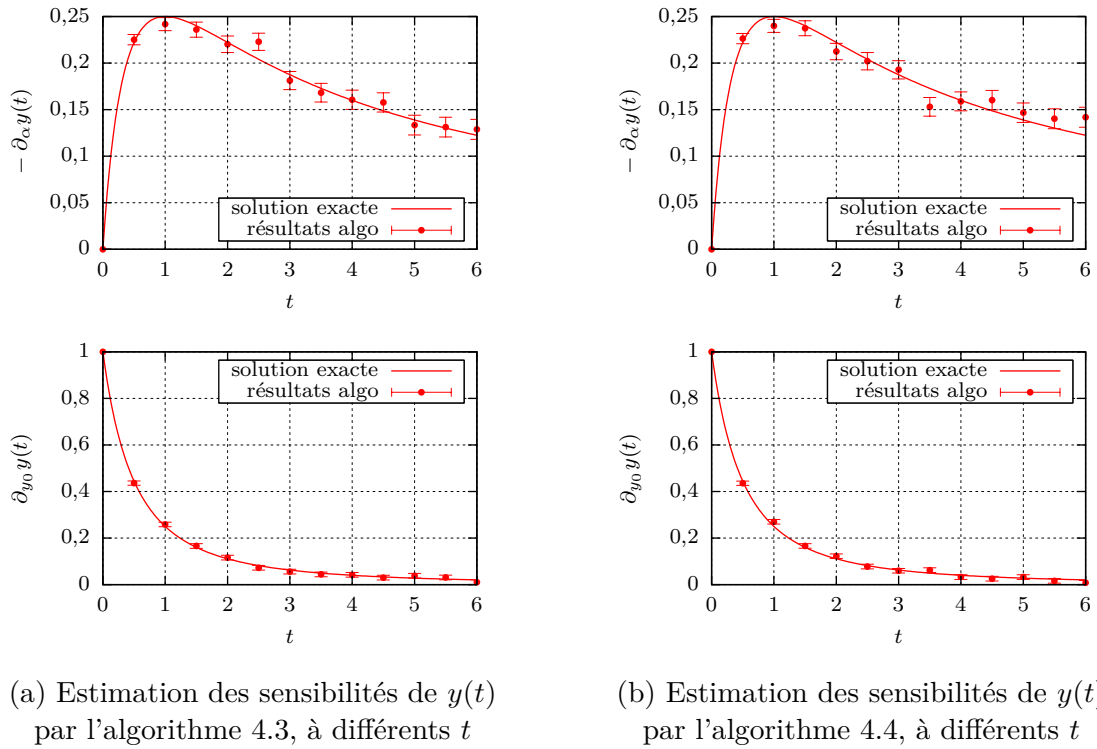


FIGURE 4.8 – Comportements des algorithmes 4.3 et 4.4 estimant $\partial_\alpha y(t)$ et $\partial_{y_0} y(t)$, et utilisant $\hat{k} = 1$. En chaque point, 10^4 réalisations ont été effectuées.

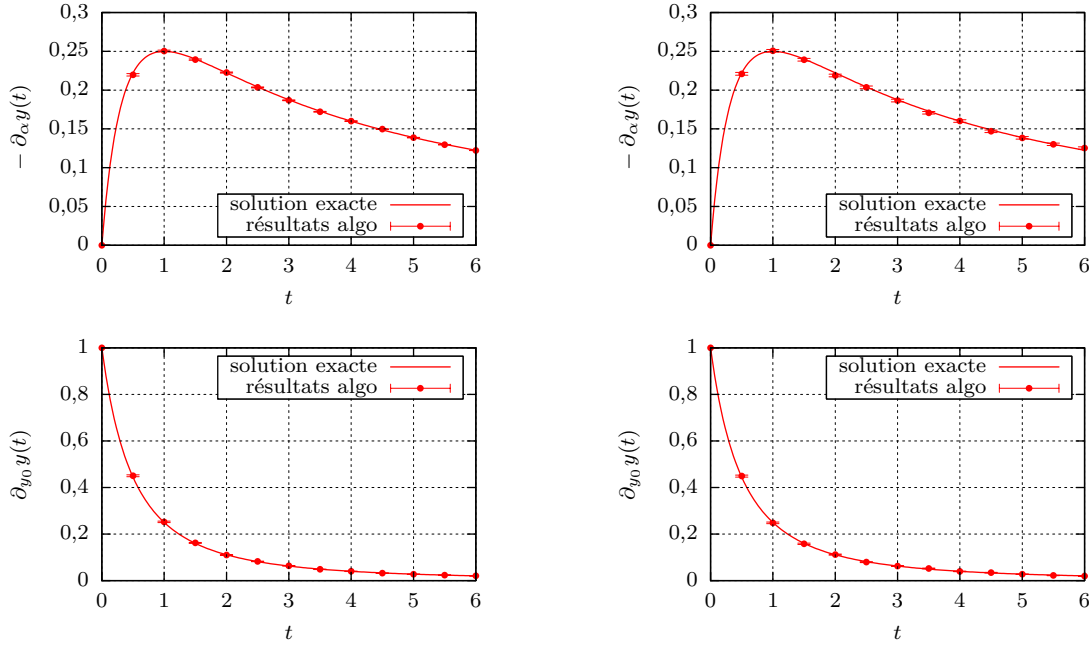
Nous ne faisons ici pas de commentaires très poussés sur les résultats obtenus. Nous remarquons tout de même que le comportement des estimateurs des sensibilités de $y(t)$ suit grossièrement celui de l'estimateur de $y(t)$. On observe les mêmes zones de non-fonctionnalité, c'est-à-dire que les estimateurs des sensibilités ont une variance finie si et seulement si l'estimateur de la grandeur en a une ; ceci était prévisible en analysant les formulations statistiques (4.35) et (4.36). Pareillement, les performances du calcul de sensibilités semblent liées aux performances du calcul de la grandeur ; là où l'algorithme 4.3 est le plus efficace (comparativement à l'algorithme 4.4) est là où l'algorithme 4.1, qu'il utilise en sous-jacent, est aussi le plus efficace, et vice-versa.

4.6.2 Utilisation d'une solution approximative

4.6.2.1 Étude théorique sur le cas choisi

●●●● On reprend encore une fois l'exemple de la section 4.3 :

$$\begin{cases} y'(t) = -\alpha y(t)^2 \\ y(0) = y_0 \end{cases} \quad \text{avec} \quad \begin{cases} \alpha > 0 \\ y_0 \geq 0 \end{cases} \quad (4.5)^\uparrow$$


 (a) Estimation des sensibilités de $y(t)$ par l'algorithme 4.3, à différents t

 (b) Estimation des sensibilités de $y(t)$ par l'algorithme 4.4, à différents t

 FIGURE 4.9 – Comportements des algorithmes 4.3 et 4.4 estimant $\partial_\alpha y(t)$ et $\partial_{y_0} y(t)$, et utilisant $\hat{k} = 2,5$. En chaque point, 10^4 réalisations ont été effectuées.

L'unique solution de ce système, sur \mathbb{R}^+ , est :

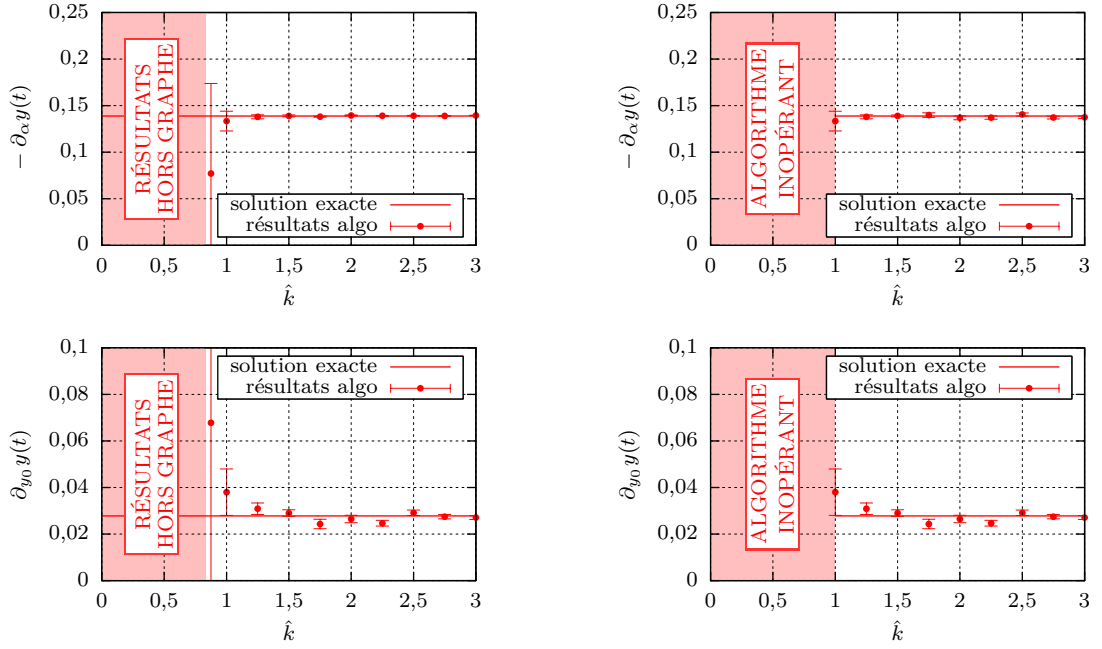
$$y(t) = \frac{y_0}{\alpha y_0 t + 1} \quad (4.6)^\dagger$$

On montre dans cette section une accélération de la convergence d'un algorithme de Monte-Carlo estimant $y(t)$, grâce à l'utilisation d'une solution approximative y^\approx connue à l'avance.

La possibilité de faire diminuer la récursivité de l'algorithme ne sera pas illustrée.

Nous avons montré à la section 4.3, en insérant un test de Bernoulli dans l'algorithme 4.1 qui devenait alors l'algorithme 4.2, qu'on pouvait faire diminuer la récursivité. Cette diminution était obtenue en créant un mécanisme de roulette russe, placé à la base de nombreuses branches des arbres d'estimations de y , élaguant de fait celles dont les résultats auraient été de toute façon peu pris en compte.

Excepté le cas particulier où $\hat{k} = \alpha y_0$, où les branches élaguées n'auraient *nullement* été prises en compte, cela se fait cependant au prix d'une augmentation de la variance de l'estimateur obtenu. Nous verrons par ailleurs au chapitre 5.4 que la démarche inverse — augmenter la récursivité pour faire diminuer la variance — peut être utilisée.


 (a) Estimation des sensibilités de $y(5)$ par l'algorithme 4.3, avec différents \hat{k}

 (b) Estimation des sensibilités de $y(5)$ par l'algorithme 4.4, avec différents \hat{k}

 FIGURE 4.10 – Comportements des algorithmes 4.3 et 4.4 estimant $\partial_\alpha y(t = 5)$ et $\partial_{y_0} y(t = 5)$, en fonction du \hat{k} utilisé. En chaque point, 10^4 réalisations ont été effectuées.

Il y a ainsi une équivalence partielle, entre :

- une diminution de la variance, sans augmentation de la récursivité,
- une diminution de la récursivité, sans augmentation de la variance.

C'est d'autant plus vrai que l'effort computationnel associé à une estimation de $y(t)$, avec un objectif de précision donné, est en fait juste proportionnel au produit de la récursivité et de la variance.

Si, dans cette section, on n'illustre que la diminution de la variance, c'est parce que c'est plus simple à mettre en place.

●●●● En injectant l'expression (4.5) de f dans l'équation (4.30), on arrive à la formulation intégrale :

$$y^\Delta(t) = \int_{-\infty}^t dt' \hat{k} \exp(-\hat{k}(t-t')) \left(H(t' \leq 0) y^\Delta(0) + H(t' > 0) \times \left(y^\Delta(t') + \frac{-\alpha (y^\Delta(t') + \tilde{y}'(t'))^2 + \alpha \tilde{y}'(t')^2}{\hat{k}} + \frac{-\alpha \tilde{y}'(t')^2 - \tilde{y}''(t')}{\hat{k}} \right) \right)$$

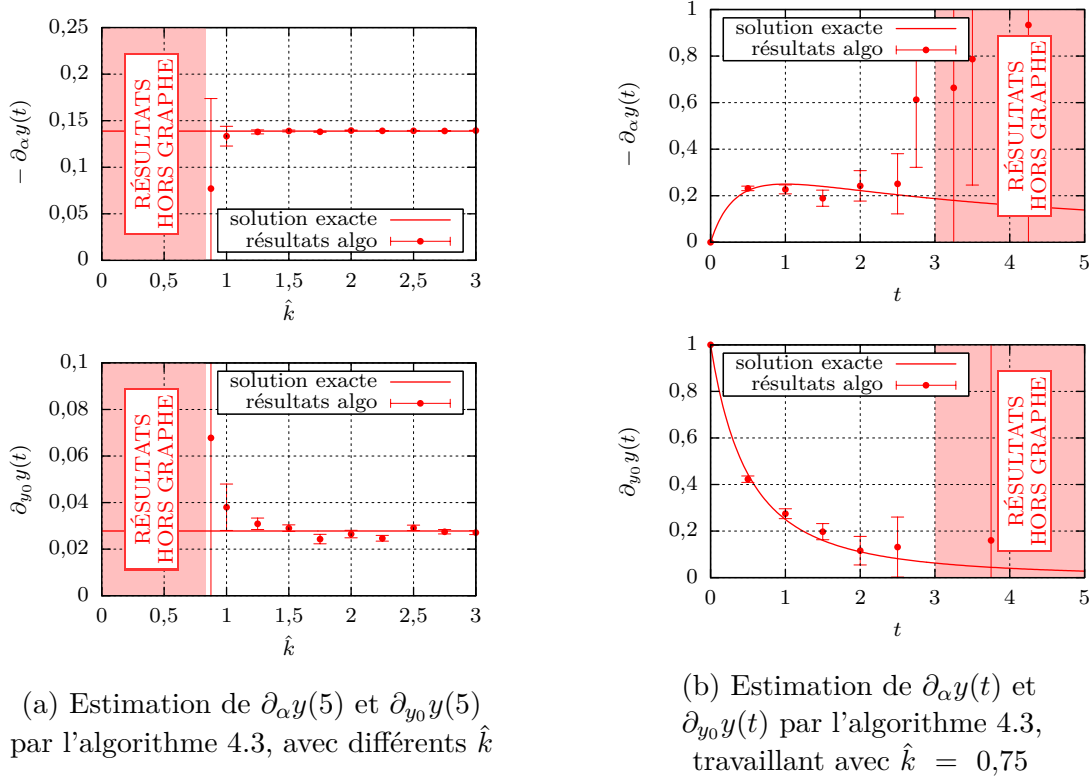


FIGURE 4.11 – Comportements de l'algorithme 4.3, dans des situations où il n'est pas fonctionnel. En chaque point, 10^4 réalisations ont été effectuées.

$$\begin{aligned}
 &= \int_{-\infty}^t dt' \hat{k} \exp(-\hat{k}(t-t')) \left(H(t' \leq 0) y^\Delta(0) + H(t' > 0) \times \right. \\
 &\quad \left. \left(y^\Delta(t') \left(1 - \frac{\alpha(2y^\sim(t') + y^\Delta(t'))}{\hat{k}} \right) - \frac{y^{\sim'}(t') + \alpha y^\sim(t')^2}{\hat{k}} \right) \right) \quad (4.37)
 \end{aligned}$$

qui peut être convertie en la formulation statistique :

$$\begin{aligned}
 &= E \left(H(T' \leq 0) y^\Delta(0) + H(T' > 0) \times \right. \\
 &\quad \left. \left(\tilde{Y}^{\Delta_1}(T') \left(1 - \frac{\alpha(2y^\sim(T') + \tilde{Y}^{\Delta_2}(T'))}{\hat{k}} \right) - \frac{y^{\sim'}(T') + \alpha y^\sim(T')^2}{\hat{k}} \right) \right) \quad (4.38)
 \end{aligned}$$

où :

- T' est une VA à valeurs dans $] -\infty; t]$ et de fonction de répartition $F_{T'} : t' \mapsto \exp(-\hat{k}(t-t'))$,
 - $\tilde{Y}^{\Delta_1}(T')$ et $\tilde{Y}^{\Delta_2}(T')$ sont 2 estimateurs indépendants de $y^\Delta(T')$.
- L'algorithme 4.5 illustre l'échantillonnage de la formulation (4.38).

<p>Entrées : Le temps t où on veut estimer $y^\Delta(t)$</p> <p>Sorties : Une estimation ponctuelle de $y^\Delta(t)$</p> <pre> 1 échantillonner $T_{r, ad}$ selon une loi exponentielle d'espérance unité : on obtient $t_{r, ad}$; 2 $t' \leftarrow t - \frac{t_{r, ad}}{\hat{k}}$; 3 si $t' \leq 0$ alors 4 calculer $y^\sim(0)$; 5 retourner $y_0 - y^\sim(0)$; 6 sinon 7 calculer $y^\sim(t')$ et $y^{\sim'}(t')$; 8 estimer $y^\Delta(t')$ grâce à cet algorithme : on obtient $\tilde{y}^{\Delta_1}(t')$; 9 estimer à nouveau $y^\Delta(t')$ grâce à cet algorithme : on obtient $\tilde{y}^{\Delta_2}(t')$; // de manière indépendante à $\tilde{y}^{\Delta_1}(t')$ 10 retourner $\tilde{y}^{\Delta_1}(t') \left(1 - \frac{\alpha (2 y^\sim(t') + \tilde{y}^{\Delta_2}(t'))}{\hat{k}} \right) - \frac{y^{\sim'}(t') + \alpha y^\sim(t')^2}{\hat{k}}$; </pre>

Algorithme 4.5 : Algorithme pour estimer ponctuellement $y^\Delta(t)$, basé sur la formulation statistique (4.38).

On suppose que la solution approximative $y^\sim(t)$ a été déterminée au préalable, et que ses valeurs sont calculables à la demande.

●●●● La récursivité moyenne *rec* de l'algorithme 4.5 est la même que celle de l'algorithme 4.1. Intéressons-nous maintenant à la variance de l'estimateur $\tilde{Y}^\Delta(t)$ généré.

Une telle étude de la variance peut reprendre la méthodologie présentée à la section 4.3.3. Elle consistait à dégager une formulation intégrale du moment d'ordre 2 de l'estimateur $\tilde{Y}(t)$, pour ensuite en tirer un système différentiel auquel ce moment obéit.

Ici, afin d'avoir une comparaison pertinente avec l'algorithme 4.1, on pousse la démarche un peu plus loin, jusqu'au dégagement d'un système différentiel sur la variance de $\tilde{Y}^\Delta(t)$. En effet, comme les estimateurs issus des algorithmes 4.1 et 4.5 n'ont pas la même espérance, comparer les 2^e moments de la distribution de leurs résultats n'est pas directement intéressant ; ce sont les variances (déduites facilement des 2^e moments) qu'il faut comparer.

La démarche pour dégager un système différentiel auquel $\text{Var}(\tilde{Y}^\Delta(t))$ obéit est finalement la suivante (très proche de ce qu'on a pu montrer en section 4.3.3) :

1. reprendre l'expression statistique (4.38), pour en tirer une expression statistique sur $E(\tilde{Y}^\Delta(t)^2)$,
2. convertir l'expression statistique obtenue en expression intégrale,
3. opérer une dérivation de l'expression intégrale obtenue, pour dégager un système différentiel sur t auquel $E(\tilde{Y}^\Delta(t)^2)$ obéit,

4. utiliser la formule de König-Huygens 1.3.2, pour transformer ce système différentiel en un système différentiel applicable à $\text{Var}(\tilde{Y}^\Delta(t))$,
5. enfin, simplifier ce dernier système à l'aide de l'équation différentielle (4.5) sur la grandeur $y(t)$.
6. Ce système ne peut semble-t-il pas être résolu symboliquement, car il fait intervenir y^\sim ; ou alors, il faut disposer d'une expression de y^\sim .

Sans entrer dans le détail des calculs, on arrive au système différentiel suivant pour la variance de $\tilde{Y}^\Delta(t)$, avec $\hat{y} = \hat{k}/\alpha$:

$$\left\{ \begin{array}{l} \frac{\partial_t \text{Var}(\tilde{Y}^\Delta(t))}{\hat{k}} = \frac{\text{Var}(\tilde{Y}^\Delta(t))^2}{\hat{y}^2} - \\ \quad 2 \left(\frac{y(t)}{\hat{y}} \left(1 - \frac{y(t)}{\hat{y}} \right) + \frac{y^\sim(t)}{\hat{y}} \left(1 - \frac{y^\sim(t)}{\hat{y}} \right) \right) \text{Var}(\tilde{Y}^\Delta(t)) + \\ \quad \left(\frac{y(t)^2 - y^\sim(t)^2}{\hat{y}} \right)^2 + \left(\frac{y^{\sim\prime}(t)}{\hat{k}} - \frac{y^\sim(t)^2}{\hat{y}} \right)^2 \\ \text{Var}(\tilde{Y}^\Delta(0)) = 0 \end{array} \right. \quad (4.39)$$

●●●● En posant $y^\sim(t) = 0$, ce qui équivaut à $y^\Delta(t) = y(t)$, on retrouve le système différentiel auquel obéit $\text{Var}(\tilde{Y}(t))$ si $\tilde{Y}(t)$ est obtenu par l'algorithme 4.1 :

$$\left\{ \begin{array}{l} \frac{\partial_t \text{Var}(\tilde{Y}(t))}{\hat{k}} = \frac{\text{Var}(\tilde{Y}(t))^2}{\hat{y}^2} - 2 \frac{y(t)}{\hat{y}} \left(1 - \frac{y(t)}{\hat{y}} \right) \text{Var}(\tilde{Y}(t)) + \frac{y(t)^4}{\hat{y}^2} \\ \text{Var}(\tilde{Y}(0)) = 0 \end{array} \right. \quad (4.40)$$

Nous n'avions pas exhibé ce système différentiel sur $\text{Var}(\tilde{Y}(t))$ durant la section 4.3.3, néanmoins il pouvait effectivement être obtenu à partir du système (4.15) portant sur $E(\tilde{Y}(t)^2)$.

On ne peut pas donner ici une solution symbolique du système (4.39), à cause de la présence des termes en y^\sim non précisés. Il est quand même possible de comparer l'évolution de la variance des estimateurs générés par les algorithmes 4.1 et 4.5, en comparant les termes contenus dans les systèmes différentiels (4.40) et (4.39).

Dans les 2 cas, on a la même condition initiale. Dans les équations d'évolution de la variance elles-mêmes :

- Le terme quadratique positif, seul susceptible de provoquer une divergence au bout d'un temps fini, est le même.
- On a un terme linéaire sur la variance. Dans le système (4.40), il est partout négatif ssi $\hat{k} \geq \alpha y_0$. Ce terme est plus grand — en valeur absolue — dans le système (4.39) tant que $\hat{k}/\alpha \geq y^\sim(t) \geq 0$.

- On a un terme d'augmentation, en $y(t)^4$. Si $2y(t) \geq y^\sim(t) \geq 0$, ce terme est plus faible dans le système (4.39).
- Un terme augmentant la variance s'ajoute dans le système (4.39). Ce terme dépend uniquement, et de façon quadratique, de l'erreur commise en construisant y^\sim voulue comme une solution du système (4.5).

Cela amène à penser que l'algorithme 4.5 construit un estimateur moins variant de $y(t)$ que l'algorithme 4.1, tant que la solution approximative y^\sim qu'il utilise comme guide est une « bonne » solution du système à résoudre (4.5). « Bonne » signifie ici que :

- y^\sim est proche de la solution exacte y , en particulier pour tout $t \geq 0, 0 \leq y^\sim(t) \leq y_0$,
- en tout point $t \geq 0$, y^\sim obéit approximativement à l'équation différentielle de (4.5), de sorte que $y^{\sim'}(t) \simeq -\alpha y^\sim(t)^2$.

A priori, pour que l'algorithme 4.5 fonctionne bien, il faudra aussi que $\hat{k} \geq \alpha y_0$, comme avec tous les algorithmes précédemment montrés dans ce chapitre.

4.6.2.2 Derniers choix algorithmiques

La méthode à discrétisation utilisée pour fournir une solution approximative au problème différentiel (4.5), avant de lancer le calcul par MMC, est la version usuelle de la méthode Runge-Kutta d'ordre 4 (abrégée RK4). On l'utilise avec un pas de temps constant Δt_\approx .

Dans la résolution des équations différentielles ordinaires, la méthode RK4 est presque systématiquement celle utilisée. Elle a l'avantage d'être :

- explicite (pas d'inversion à faire),
- très précise : elle est d'ordre 4, c'est-à-dire que quand le pas de temps tend vers 0, l'erreur de calcul est proportionnelle au pas de temps à la puissance 4,
- d'une stabilité comparable à celle de la méthode d'Euler explicite,
- peu coûteuse, en terme de temps de calcul et de difficulté de programmation. Son fonctionnement est similaire à celui de la méthode d'Euler, qui constitue elle-même la méthode Runge-Kutta d'ordre 1.

La méthode RK4 fournit une solution sous la forme d'une liste de points $(t_i; y^\sim(t_i); y^{\sim'}(t_i))$. Entre ces points, nous choisissons d'utiliser une interpolation polynomiale de degré 3, égalisant y^\sim et $y^{\sim'}$ aux points calculés les plus proches.

L'explication détaillée et l'implémentation des techniques d'obtention et d'interpolation de y^\sim est donnée en annexe D.2.

●●●● Simultanément à l'estimation de $y^\Delta(t)$, et donc de $y(t)$, par l'algorithme 4.5, nous estimons les sensibilités de $y(t)$ vis-à-vis des paramètres α et y_0 . Ce calcul de sensibilités est réalisé selon les formulations intégrales (4.35) et (4.36) ; quand on a besoin d'une estimation de $y(t)$, celle-ci est remplacée par $y^\sim(t)$ plus une estimation de $y^\Delta(t)$.

Le calcul de sensibilités est donc effectué selon une démarche semblable à celle prévalant dans les algorithmes 4.3 et 4.4, mais cette fois-ci c'est l'algorithme 4.5 qui fournit des estimations de $y(t)$. L'algorithme 4.5 n'a été en rien modifié pour permettre le calcul des sensibilités, ce calcul n'utilise que quelques instructions supplémentaires qui ne modifient pas la récursivité de l'algorithme 4.5.

La campagne de tests illustre 2 dépendances de $\text{Var}(\tilde{Y}^\Delta(t))$:

- la dépendance de $\text{Var}(\tilde{Y}^\Delta(t))$ selon t ,
- la dépendance de $\text{Var}(\tilde{Y}^\Delta(t))$ selon la précision de la solution y^\approx , donc selon Δt_\approx .

Cette campagne est menée avec un majorant $\hat{k} \geq \alpha y_0$, suffisant pour garantir que l'estimateur $\tilde{Y}^\Delta(t)$ obtenu est de variance finie — quand y^\approx n'est pas une approximation trop mauvaise.

Les paramètres α et y_0 sont tous les deux pris égaux à 1.

4.6.2.3 Résultats, et commentaires

Les résultats sont donnés en figures 4.12, 4.13, 4.14, et 4.15.

Le comportement de l'estimateur $\tilde{Y}^\Delta(t)$ peut être compris grâce à l'équation différentielle sur sa variance (4.39). Elle explique notamment pourquoi la variance relative de l'estimateur $\tilde{Y}(t)$ obtenu par l'algorithme 4.5 décroît avec le temps t , contrairement à celle de l'estimateur obtenu par l'algorithme 4.1.

Ceci est notamment dû à ce que :

- la solution approximative y^\approx est de bonne qualité, même avec $\Delta t_\approx = 1$,
- la solution $y(t)$ tend asymptotiquement vers l'axe des abscisses, quand $t \rightarrow +\infty$.

Dans l'équation différentielle (4.39), nous avons 4 termes :

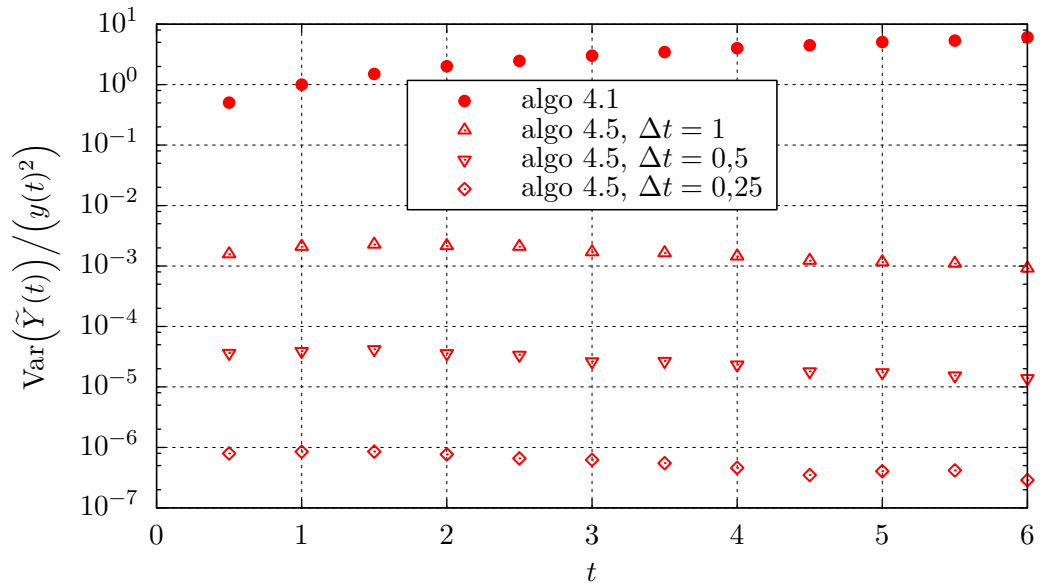
1. un terme quadratique, négligeable car \tilde{Y}^Δ est de faible variance,
2. un terme de décroissance linéaire,
3. un terme source en $(y(t)^2 - y^\approx(t)^2)^2$,
4. un terme source, dû à l'imprécision de y^\approx en tant que solution de (4.5).

Or, quand t augmente :

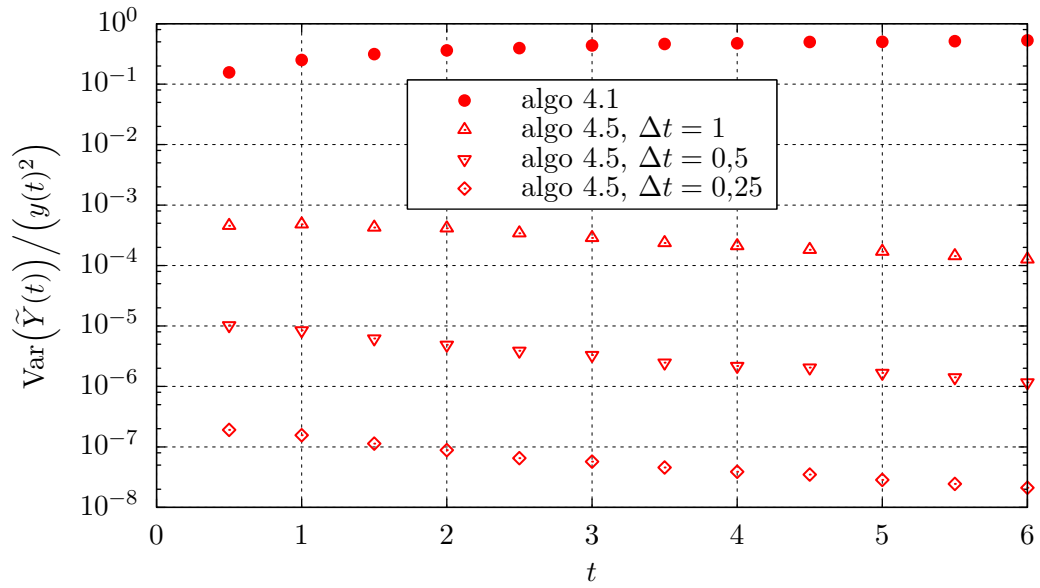
- Le 3^e terme tend rapidement vers 0, car $y(t)$ et $y^\approx(t)$ tendent vers 0.
- Comme y^\approx décroît de plus en plus lentement, et est de moins en moins convexe, la méthode RK4 résout le système (4.5) de plus en plus précisément. Donc le 4^e terme s'effondre.

Combiné avec le terme de décroissance linéaire, la chute des termes sources de variance amène une diminution notable de $\text{Var}(\tilde{Y}^\Delta(t))$ quand t augmente.

De même, quand Δt_\approx diminue, y^\approx est plus précise, et la variance de \tilde{Y}^Δ diminue.

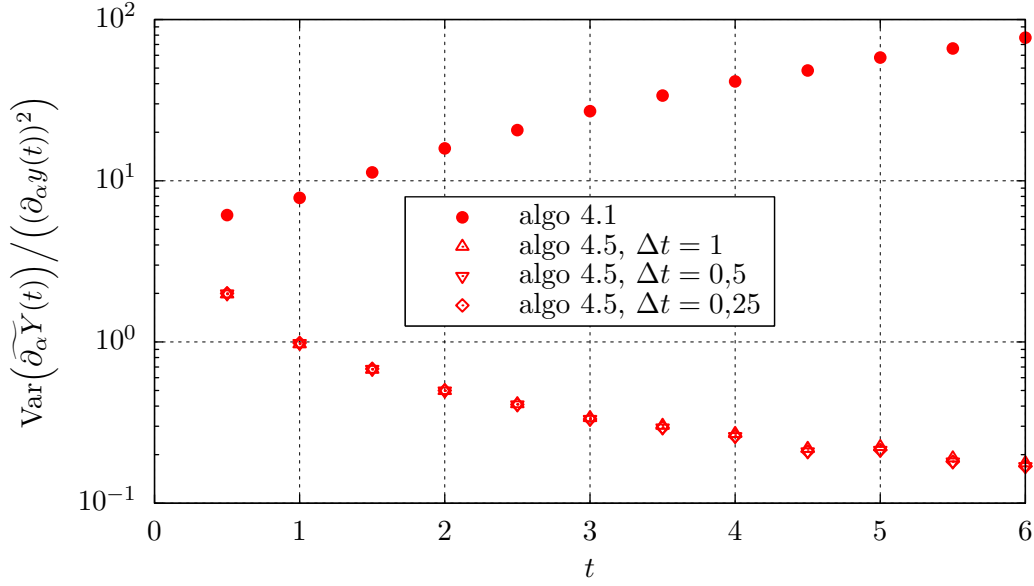
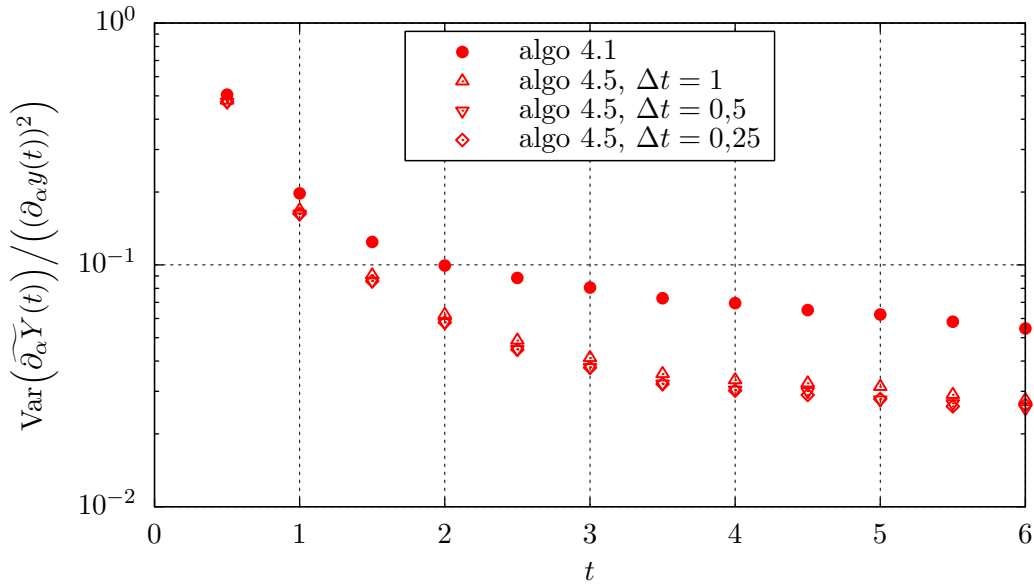


(a) Avec $\hat{k} = 1$



(b) Avec $\hat{k} = 2,5$

FIGURE 4.12 – Évolution selon t de la variance relative de plusieurs estimateurs $\tilde{Y}(t)$ de $y(t)$. Les estimateurs sont issus de l'algorithme 4.1 ou de l'algorithme 4.5 (avec $\tilde{Y}(t) = y^\sim(t) + \tilde{Y}^\Delta(t)$). Dans ce 2^e cas, plusieurs guides y^\sim plus ou moins précis sont considérés. 2 valeurs pour \hat{k} sont utilisées. À part ça, les conditions de l'expérience numérique sont celles décrites dans la section 4.6.2.2.

(a) Avec $\hat{k} = 1$ (b) Avec $\hat{k} = 2,5$ FIGURE 4.13 – Évolution selon t de la variance relative de plusieurs estimateurs $\widetilde{\partial_\alpha Y(t)}$ de $\partial_\alpha y(t)$.

Ces estimateurs sont tous construits à l'aide de la formulation statistique (4.35), mais en se basant sur des estimateurs $\widetilde{Y}(t)$ différents. L'algorithme prodiguant l'estimateur $\widetilde{Y}(t)$, utilisé de façon sous-jacente, est précisé sur la figure.

2 valeurs pour \hat{k} sont utilisées. À part ça, les conditions de l'expérience numérique sont celles décrites dans la section 4.6.2.2.

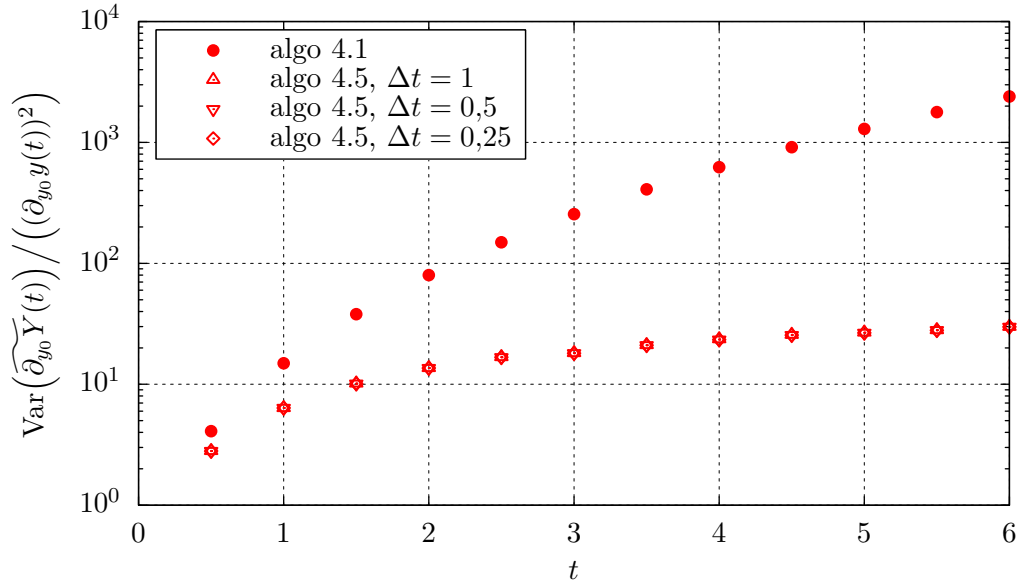
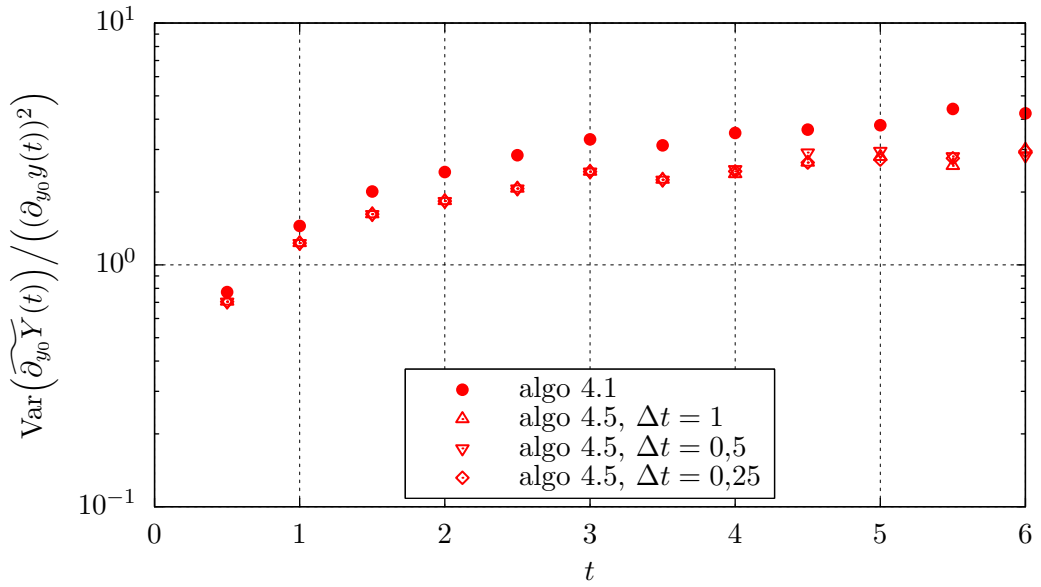

 (a) Avec $\hat{k} = 1$

 (b) Avec $\hat{k} = 2,5$

 FIGURE 4.14 – Évolution selon t de la variance relative de plusieurs estimateurs $\widetilde{\partial_{y_0} Y(t)}$ de $\partial_{y_0} y(t)$.

Ces estimateurs sont tous construits à l'aide de la formulation statistique (4.36), mais en se basant sur des estimateurs $\tilde{Y}(t)$ différents. L'algorithme prodiguant l'estimateur $\tilde{Y}(t)$, utilisé de façon sous-jacente, est précisé sur la figure.

2 valeurs pour \hat{k} sont utilisées. À part ça, les conditions de l'expérience numérique sont celles décrites dans la section 4.6.2.2.

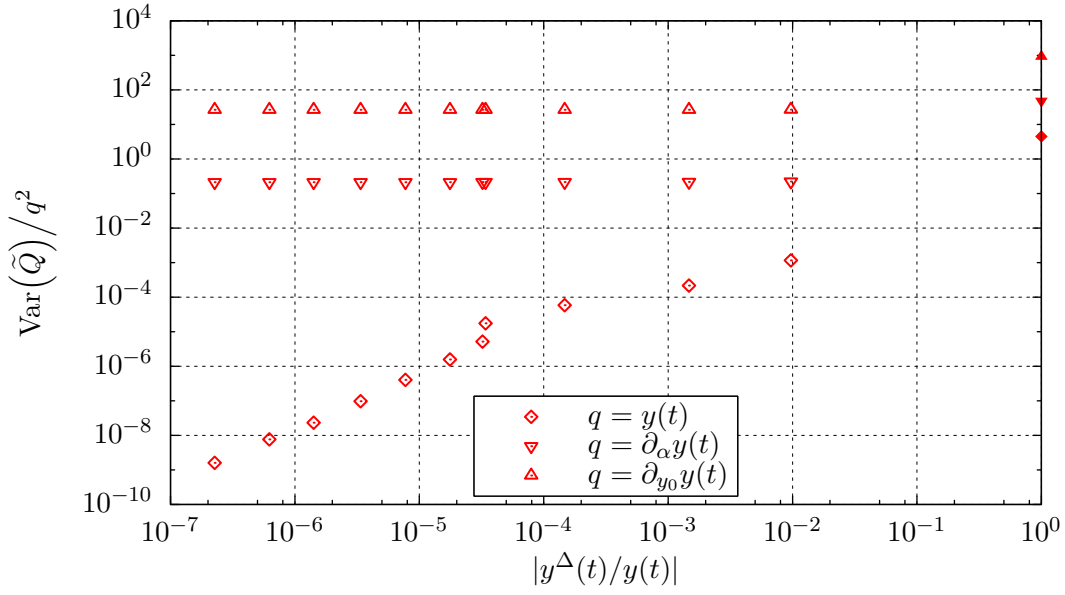
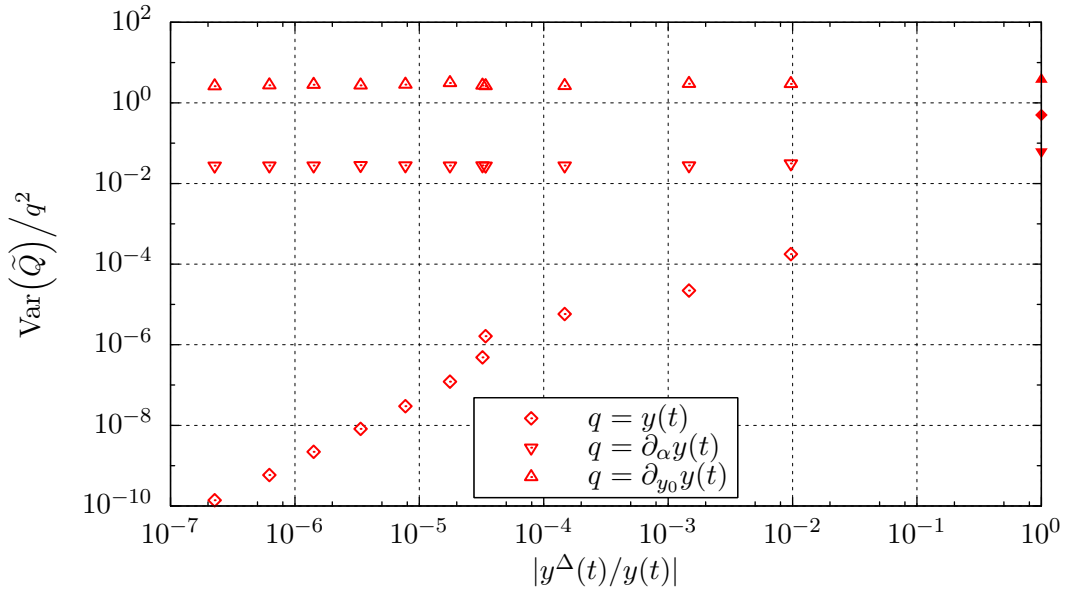
(a) Avec $\hat{k} = 1$ (b) Avec $\hat{k} = 2,5$

FIGURE 4.15 – Profils de la variance des estimateurs de $y(t)$, $\partial_\alpha y(t)$, et $\partial_{y_0} y(t)$, obtenus grâce à l'algorithme 4.5, selon la précision de y^\approx voulue comme une solution du système (4.5). On se place en $t = 5$, et la précision de y^\approx est évaluée par l'écart relatif $|y^\Delta(t)/y(t)|$ en $t = 5$. Les points pleins indiquent les résultats de l'algorithme 4.1, c'est-à-dire les résultats qu'aurait obtenu l'algorithme 4.5 avec $y^\approx = 0$. Les points creux sont obtenus en faisant varier Δt_\approx de 1 à 0,1. 2 valeurs pour \hat{k} sont utilisées. À part ça, les conditions de l'expérience numérique sont celles décrites dans la section 4.6.2.2.

Le calcul des sensibilités de $y(t)$ vis-à-vis de α et y_0 semble peu profiter de l'amélioration de la précision de l'estimation de $y(t)$. Ceci est particulièrement vrai quand \hat{k} est grand.

C'est dû à ce que le calcul de sensibilités ne se fait que sur une seule branche de l'arbre d'estimations, étant donnés les choix algorithmiques que l'on a fait. Dans ce cas, la valeur d'une estimation ponctuelle d'une sensibilité est surtout sensible au nombre de « collisions » qui ont eu lieu le long de cette branche. Ceci explique d'ailleurs que le calcul des sensibilités s'améliore nettement quand \hat{k} augmente. Dans ce cas les collisions sont plus nombreuses, mais chaque modification sur le poids en sensibilité est plus faible, et au final on échantillonne mieux l'évolution de la sensibilité selon le temps.

Chapitre 5

Résultats obtenus dans un gaz uniforme mais hors équilibre

On résout l'équation de Boltzmann par MMC dans un cas académique de cinétique des gaz, où une solution symbolique est disponible. Ce cas académique montre un gaz uniforme, où la distribution des vitesses revient vers l'équilibre en partant d'une situation initiale qui en est très éloignée. Avec cet exemple, par comparaison avec celui du chapitre précédent, on a maintenant un espace des phases (même si il se réduit à l'espace des vitesses) et un terme source correspondant aux collisions entrantes (une particule de vitesse \vec{c}' collisionne avec une particule de vitesse \vec{c}_ et se retrouve à la vitesse étudiée \vec{c}).*

Ce cas simple est aussi l'occasion de commencer à sonder les éventuelles difficultés de convergence associées à la résolution par MMC de l'équation de Boltzmann. Aucun problème de variance n'apparaît aux grandes vitesses, même lorsque les particules correspondantes sont extrêmement peu nombreuses. C'est notre premier exemple d'un calcul sonde se concentrant sur un événement rare. Par contre il apparaît un problème de variance aux grands temps (temps physiques) simulés. Une technique est proposée pour y remédier partiellement.

Ce chapitre présente une résolution par MMC de l'équation de Boltzmann, dans un cas académique usuellement nommé *mode BKW*. Entre autres particularités, il s'agit d'un gaz uniforme, dont la dynamique hors-équilibre est détaillé dans la 1^{re} section. C'est l'un des rares problèmes de cinétique des gaz pour lesquels on dispose d'une solution symbolique.

La formulation intégrale de l'équation de Boltzmann, que l'on va présenter en section 5.2 puis tester en section 5.3, est similaire à ce qu'on a pu montrer au chapitre précédent ; mais l'espace des vitesses est ajouté. Ce cas académique est l'occasion d'un premier test de la validité de notre approche : dégager un estimateur exact de la fonction de distribution dans des problèmes de cinétique des gaz, sans aucune discrétisation de l'espace des vitesses et sans chercher à représenter la

statistique de vitesse par le suivi d'un grand nombre de pseudo-particules.

Une particularité intéressante de l'estimateur que nous obtenons est qu'il a la même variance relative partout dans l'espace des vitesses. On a aucune difficulté à calculer ce qui se passe dans des zones peu peuplées de l'espace des vitesses, contrairement à ce qui se passe par exemple avec les méthodes BsR et DSMC. Et c'est bien normal : ces 2 méthodes suivent des particules au cours de leur évolution, soit par un suivi statistique direct (DSMC), soit grâce à une quadrature nécessairement centrée sur les zones très peuplées de l'espace des vitesses (BsR). L'estimateur que nous proposons ici est au contraire construit, du point de vue des chemins suivis, en partant de la zone d'intérêt : le suivi des particules se fait en sens inverse depuis le point d'étude. Comme expliqué au chapitre 1.2, partant de formulations intégrales de l'équation de Boltzmann, il est normal d'aboutir à un suivi des particules en remontant le temps. Cette capacité à rester précis dans des zones peu peuplées de l'espace des vitesses est très utile du point de vue applicatif. Par exemple en chimie, où des événements déterminants sur l'évolution du système se produisent uniquement à grande vitesse, donc à haute énergie, et n'impliquent nécessairement que peu de particules. Cette question était d'ailleurs l'objectif initial des concepteurs du mode BKW. Nous prenons la suite avec des algorithmes de Monte Carlo dont nous espérons qu'ils autoriseront de telles analyses au delà du mode BKW.

Une difficulté importante apparaît cependant : on ne peut atteindre que des temps physiques simulés assez faibles. Cela est dû à ce que la variance d'un produit de variables aléatoires égalise, grossièrement, le produit des variances des variables (voir chapitre 1.5.3). Or notre estimateur se détaille comme un grand produit, où chaque branche de l'arbre de suivi des particules amène un terme. Nous nous attendions à rencontrer des difficultés informatiques associée à la multiplication du nombre de branches, mais le problème s'avère numérique et non pas informatique. Nous présentons, dans les 2 dernières sections, une première tentative d'y remédier.

5.1 Quelques explications sur le mode BKW (Bobylev-Krook-Wu)

5.1.1 Revue bibliographique partielle

Le couramment dénommé *mode BKW* a été détaillé pour la 1^{re} fois parallèlement par Alexander V. BOBYLEV [11], et par Max KROOK et Tai Tsun WU [74], en 1976. C'est une solution symbolique de l'équation de Boltzmann sous forme complète.

De telles solutions de l'équation de Boltzmann sont relativement rares. Il y a bien sûr l'équilibre thermodynamique global, ainsi que des situations où les particules ne connaissent ni collision sauf aux parois, ni aucune force à longue portée ; ces configurations dont la compréhension est relativement simple sont cependant d'un intérêt assez limité. En dehors de cela sont connus :

- la situation de confinement harmonique, exposée au chapitre suivant,

- le mode BKW,
- des solutions semblables au mode BKW, mais en dimension autre que 3 (voir [39]),
- des solutions valables dans des conditions relativement semblables à celles du mode BKW (particules de Maxwell, absence de force à longue portée, uniformité, et isotropie), mais où la densité volumique d'énergie est infinie. Pour plus de détails à ce sujet, lire les publications [12, 13].
- de nombreux comportements asymptotiques. Le lecteur intéressé pourra se reporter par exemple aux publications [40, 41].

La découverte d'une solution exacte à l'équation de Boltzmann, à la fois instationnaire, dans laquelle les collisions jouent un rôle, et répondant aux exigences de réalisme physique les plus simples (positivité de f , densités de masse et d'énergie finies) a été une avancée importante pour le développement de la théorie cinétique des gaz. Le mode BKW a été l'objet d'une littérature conséquente, permettant le test de plusieurs hypothèses théoriques (voir par exemple [27, 51]). Pour une revue plus approfondie, le lecteur pourra utiliser la publication [56].

5.1.2 Description du mode BKW

●●●● On se place dans un gaz monomoléculaire et monoatomique, dans une situation uniforme et isotrope, ainsi : $f(\vec{r}; \vec{c}; t) = f(\|\vec{c}\|; t)$. Par commodité, on notera toutefois : $f(\vec{r}; \vec{c}; t) = f(\vec{c}; t)$. Il n'y a pas de force à longue portée s'exerçant sur les molécules, et les molécules respectent le modèle de Maxwell isotrope, c'est-à-dire que $\sigma_F(g; u) = \kappa/(4\pi g)$, où κ est une constante. Dans ce cas, l'équation de Boltzmann se simplifie en :

$$\partial_t f(\vec{c}; t) = -\kappa \eta f(\vec{c}; t) + \kappa \int_{E_c} d\vec{c}_* \int_{4\pi} \frac{d\vec{u}'}{4\pi} f(\vec{c}'; t) f(\vec{c}_*; t) \quad (5.1)$$

où

$$\begin{cases} \eta = \int_{E_c} d\vec{c} f(\vec{c}; t) \text{ est une constante} \\ \vec{c}' = \frac{\vec{c} + \vec{c}_*}{2} + \frac{\|\vec{c} - \vec{c}_*\|}{2} \vec{u}' \\ \vec{c}_*' = \frac{\vec{c} + \vec{c}_*}{2} - \frac{\|\vec{c} - \vec{c}_*\|}{2} \vec{u}' \end{cases}$$

Afin de simplifier encore les écritures on adimensionnalise partiellement cette précédente équation : f remplace f/η , et t remplace $\kappa \eta t$, alors :

$$\partial_t f(\vec{c}; t) = -f(\vec{c}; t) + \int_{E_c} d\vec{c}_* \int_{4\pi} \frac{d\vec{u}'}{4\pi} f(\vec{c}'; t) f(\vec{c}_*; t) \quad (5.2)$$

●●●● Cette équation admet la solution particulière suivante :

$$f_{\text{BKW}}(\vec{c}; t) = \frac{\exp(-\vec{c}^2/2Kc_{qma}^2)}{2(2\pi Kc_{qma}^2)^{3/2}} \left(\frac{5K-3}{K} + \frac{1-K}{K^2} \times \frac{\vec{c}^2}{c_{qma}^2} \right) \quad (5.3a)$$

où

$$K = 1 - \exp(-t/6) \quad (5.3b)$$

Pour que cette solution soit positive, il faut que $K \geq 3/5$, donc que $t \geq t_d = 6 \ln 5/2 \simeq 5,4977$. De plus, par la suite on prendra l'unique paramètre c_{qma} égal à 1 (Ce paramètre est bien la vitesse quadratique moyenne sur chaque axe : $3\eta c_{qma}^2 = \int_{E_c} d\vec{c} f(\vec{c}; t) \vec{c}^2$) : quand c_{qma} varie, f_{BKW} reste similaire par homothétie sur \vec{c} .

La figure 5.1 donne une représentation graphique de la solution f_{BKW} . En résumé, la situation initiale du mode BKW est une Maxwellienne à trop basse température, où les particules à très basses vitesses ont été déplacées vers les moyennes vitesses. Ensuite, quand le temps passe, les zones de moyenne vitesse se déplacent vers les zones de haute et basse vitesse.

●●●● Comme nous utilisons cette grandeur plus loin dans ce chapitre, nous donnons ici l'expression de la fraction des particules dont la vitesse dépasse un seuil c_0 donné :

$$\begin{aligned} & \text{Frac}(\|\vec{c}\| \geq c_0; t) \\ &= \int_{E_c} d\vec{c} H(\|\vec{c}\| \geq c_0) f_{\text{BKW}}(\vec{c}; t) \end{aligned} \quad (5.4)$$

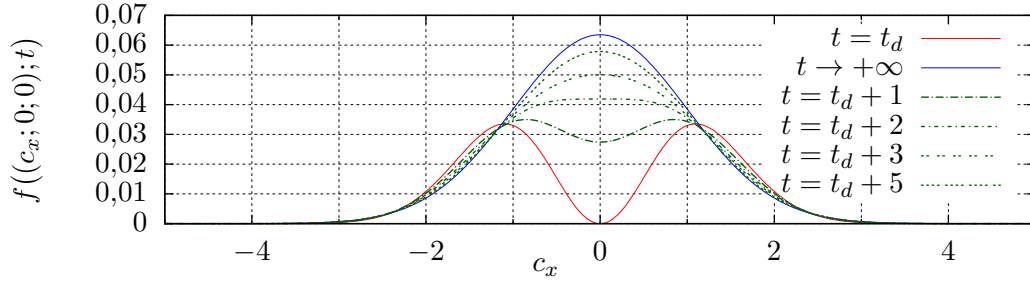
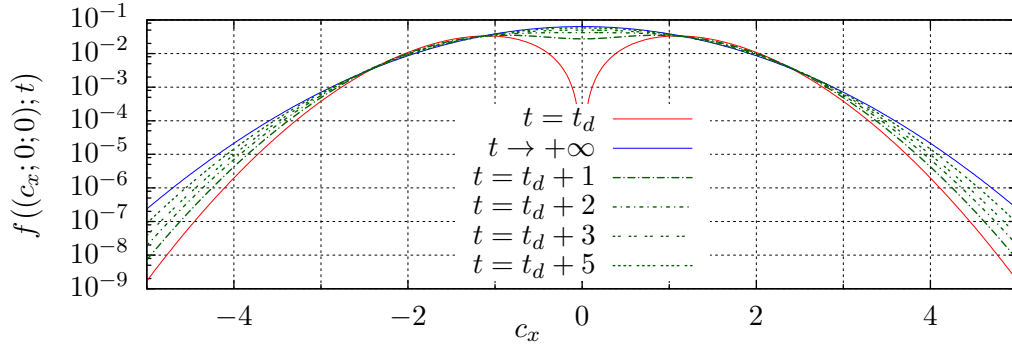
$$= \exp\left(\frac{-c_0^2}{2K c_{qma}^2}\right) \left(\frac{1}{\sqrt{\pi}} \frac{c_0}{\sqrt{2K} c_{qma}} \left(2 + \frac{1-K}{K^2} \frac{c_0^2}{c_{qma}^2} \right) + \text{erfcx}\left(\frac{c_0}{\sqrt{2K} c_{qma}}\right) \right) \quad (5.5)$$

où erfcx est la fonction d'erreur complémentaire normalisée (voir l'annexe A.3).

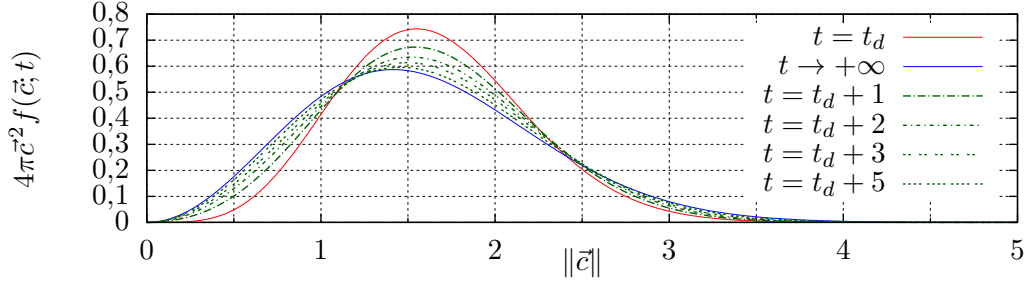
5.2 Développement d'un algorithme de Monte-Carlo simple pour le mode BKW

La disposition que nous avons d'une solution symbolique à l'équation de Boltzmann est pour nous l'occasion de tester la fonctionnalité de plusieurs algorithmes de type Monte-Carlo adaptés à la cinétique des gaz. Bien sûr, les aspects de tels algorithmes ne pourront pas être tous testés ici, à cause des particularités du mode BKW. Ses particularités les plus importantes sont certainement l'uniformité, qui fait qu'on ne testera pas la manière de gérer le transport, associée au modèle de particules de Maxwell, qui rend le taux de collision constant. Ces deux aspects seront travaillés sur la situation de piège harmonique, au chapitre suivant.

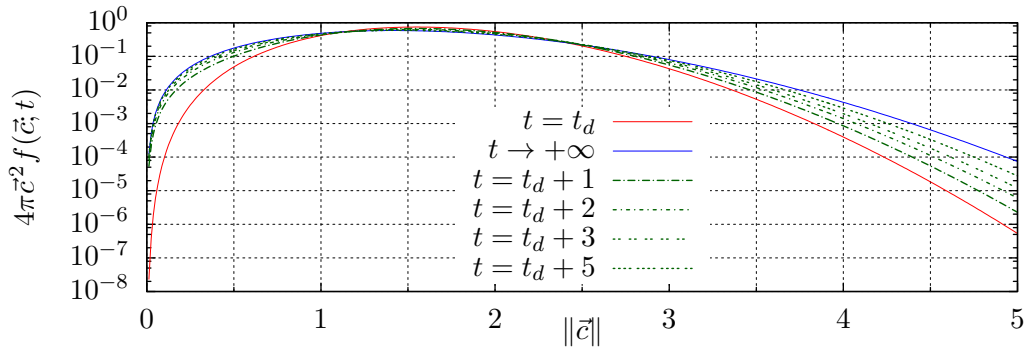
Étant donnée cette incapacité à être exhaustifs dans le test des formes algorithmiques que nous voulons monter, nous nous contenterons ici d'algorithmes spécifiquement adaptés au calcul de f dans le mode BKW. En particulier, nous profiterons de la constance du taux de collision pour simplifier l'échantillonnage des temps de vol libre des particules. Cette simplification n'est pas seulement portée


 (a) Densité de matière le long de l'axe (Ox) de l'espace des vitesses


(b) Idem (a), avec échelle logarithmique



(c) Répartition de la matière selon la norme de la vitesse



(d) Idem (c), avec échelle logarithmique

 FIGURE 5.1 – Profils de f à différents temps dans l'espace des vitesses, dans le cas du mode BKW décrit en section 5.1.2. f ne dépend que du temps et de $\|\vec{c}\|$.

par une volonté de se faciliter le travail ; c'est aussi l'occasion de distinguer les sources possibles de dysfonctionnement de nos algorithmes. Dans ce chapitre, nous devrons juste traiter la présence d'un terme source quadratique dans l'équation d'évolution (5.2), et il sera à lui seul source d'importantes difficultés. Quand nous rajouterons au chapitre suivant un échantillonnage plus fonctionnel des temps de vol libre, cela ajoutera des difficultés supplémentaires que nous pourrions distinguer des précédentes.

5.2.1 Formulation intégrale

On réécrit l'équation d'évolution (5.2) sous la forme :

$$\partial_t f(\vec{c}; t) + f(\vec{c}; t) = s(\vec{c}; t) \quad \text{avec} \quad s(\vec{c}; t) = \int_{E_c} d\vec{c}_* \int_{4\pi} \frac{d\vec{u}'}{4\pi} f(\vec{c}'; t) f(\vec{c}_*; t) \quad (5.6)$$

En supposant que f est connue à un temps $t_{init} \geq t_d$, on peut écrire une solution partielle à l'équation (5.6) :

$$f(\vec{c}; t) = \int_{-\infty}^t dt' e^{-(t-t')} \left(H(t' \leq t_{init}) f(\vec{c}; t_{init}) + H(t' > t_{init}) s(\vec{c}; t') \right) \quad (5.7)$$

$$= \int_{-\infty}^t dt' e^{-(t-t')} \left(H(t' \leq t_{init}) f(\vec{c}; t_{init}) + H(t' > t_{init}) \times \int_{E_c} d\vec{c}_* \int_{4\pi} \frac{d\vec{u}'}{4\pi} f(\vec{c}'; t') f(\vec{c}_*; t') \right) \quad (5.8)$$

Pour obtenir une expression statistique évaluable dans un algorithme de Monte-Carlo, il nous faut introduire une variable aléatoire pour chaque variable d'intégration. Posons donc les variables aléatoires T' , \vec{C}_* , et \vec{U}' à valeurs respectivement dans $] -\infty; t]$, E_c et $\mathcal{S}(E_c; \vec{0}; 1)$. Une interprétation algorithmique et physique de ces VA peut être donnée :

- T' est la date de la dernière collision d'une particule suivie.
- \vec{C}_* est la vitesse sortante de la partenaire, dans sa dernière collision, de la particule suivie.
- \vec{U}' est la direction d'entrée, de la particule suivie, dans le référentiel barycentrique de sa dernière collision.

On a :

$$f(\vec{c}; t) = \int_{-\infty}^t p_{T'}(t') dt' \frac{e^{-(t-t')}}{p_{T'}(t')} \left(H(t' \leq t_{init}) f(\vec{c}; t_{init}) + H(t' > t_{init}) \times \int_{E_c} p_{\vec{C}_*}(\vec{c}_*) d\vec{c}_* \int_{4\pi} p_{\vec{U}'}(\vec{u}') d\vec{u}' \frac{f(\vec{c}'; t') f(\vec{c}_*; t')}{4\pi p_{\vec{C}_*}(\vec{c}_*) p_{\vec{U}'}(\vec{u}')} \right) \quad (5.9)$$

L'expression statistique correspondante est :

$$f(\vec{c}; t) = \mathbb{E} \left(\frac{e^{-(t-T')}}{p_{T'}(T')} \left(H(T' \leq t_{init}) f(\vec{c}; t_{init}) + H(T' > t_{init}) \frac{\tilde{F}_1(\vec{C}'; T') \tilde{F}_2(\vec{C}_*; T')}{4\pi p_{\vec{C}_*}(\vec{C}_*) p_{\vec{U}'}(\vec{U}')} \right) \right) \quad (5.10a)$$

où

$$\vec{C}' = \frac{\vec{c} + \vec{C}_*}{2} + \frac{\|\vec{c} - \vec{C}_*\|}{2} \vec{U}', \quad \vec{C}_* = \frac{\vec{c} + \vec{C}_*}{2} - \frac{\|\vec{c} - \vec{C}_*\|}{2} \vec{U}' \quad (5.10b)$$

et les notations « $\tilde{F}_1(\vec{c}_1; t_1)$ » et « $\tilde{F}_2(\vec{c}_2; t_2)$ » signifient 2 estimateurs indépendants de $f(\vec{c}_1; t_1)$ et $f(\vec{c}_2; t_2)$.

5.2.2 Un choix pour $p_{T'}$, $p_{\vec{C}_*}$ et $p_{\vec{U}'}$

Le choix le plus évident pour les VA T' , \vec{C}_* et \vec{U}' est sans doute de prendre une loi exponentielle unité à gauche de t pour T' , une loi gaussienne centrée réduite (une Maxwellienne à vitesse nulle et vitesse quadratique unité sur chaque axe) pour \vec{C}_* , et une loi isotrope pour \vec{U}' . Cela s'écrit :

$$\begin{aligned} F_{T'}(t') &= H(t' < t) e^{-(t-t')} + H(t' \geq t) \times 1 \\ p_{\vec{C}_*}(\vec{c}_*) &= \frac{\exp(-\vec{c}_*^2/2)}{(2\pi)^{3/2}} \quad p_{\vec{U}'}(\vec{u}') = \frac{1}{4\pi} \end{aligned} \quad (5.11)$$

Au-delà de sa simplicité, ce choix algorithmique a une autre qualité : avec les hypothèses du mode BKW, si à t_{init} la fonction de distribution f est la distribution d'équilibre à densité unité, vitesse nulle et température unité — c'est-à-dire l'état final f_{fin} de la solution (5.3) —, alors l'estimateur généré est de variance nulle. En effet, en appliquant ce choix et cette distribution initiale dans l'expression (5.10) :

- le terme $\frac{e^{-(t-T')}}{p_{T'}(T')}$ se simplifie,
- si $T' \leq t_{init}$, alors $\tilde{F}(\vec{c}; t) = f_{fin}(\vec{c})$,
- si $T' > t_{init}$, et que $\tilde{F}_1(\vec{C}'; T') = f_{fin}(\vec{C}')$ et $\tilde{F}_2(\vec{C}_*; T') = f_{fin}(\vec{C}_*)$, alors $\tilde{F}(\vec{c}; t) = f_{fin}(\vec{c})$. Il suffit de réaliser le calcul suivant pour s'en convaincre :

$$\begin{aligned} & \frac{f_{fin}(\vec{C}'; t) f_{fin}(\vec{C}_*; t)}{4\pi p_{\vec{U}'}(\vec{U}') p_{\vec{C}_*}(\vec{C}_*)} \\ &= \frac{\exp(-\vec{C}'^2/2) \exp(-\vec{C}_*'^2/2)}{(2\pi)^{3/2} \exp(-\vec{C}_*^2/2)} \\ &= \frac{\exp(-(\vec{C}'^2 + \vec{C}_*'^2)/2)}{(2\pi)^{3/2} \exp(-\vec{C}_*^2/2)} \end{aligned}$$

comme les collisions conservent l'énergie cinétique (voir système (1.19)) :

$$\begin{aligned}
 &= \frac{\exp\left(-(\vec{c}^2 + \vec{C}_*^2)/2\right)}{(2\pi)^{3/2} \exp\left(-\vec{C}_*^2/2\right)} \\
 &= \frac{\exp(-\vec{c}^2/2)}{(2\pi)^{3/2}} \\
 &= f_{fin}(\vec{c})
 \end{aligned}$$

L'estimation de $f(\vec{c}; t)$ va générer un arbre d'estimations, comme au chapitre 4.2. Par récurrence depuis les feuilles jusqu'à la racine de cet arbre, on obtient l'égalité de $\tilde{F}(\vec{c}; t)$ avec $f_{fin}(\vec{c}; t)$ (qui est bien la valeur attendue en espérance dans ce cas).

●●●● Finalement, en se basant sur l'expression statistique (5.10) et en y injectant les densités de probabilités choisies en (5.11), on arrive à l'algorithme 5.1.

Entrées : Le point $(\vec{c}; t)$ où on veut estimer $f(\vec{c}; t)$	
Sorties : Une estimation ponctuelle de $f(\vec{c}; t)$	
1	échantillonner T_r selon une loi exponentielle unité : on obtient t_r ;
2	$t' \leftarrow t - t_r$;
3	si $t' \leq t_{init}$ alors retourner $f(\vec{c}; t_{init})$; // $f(\vec{c}; t_{init})$ est connue
4	sinon
5	échantillonner \vec{C}_* , selon la loi normale centrée réduite tridimensionnelle : on obtient \vec{c}_* ;
6	échantillonner \vec{U}' de façon isotrope : on obtient \vec{u}' ;
7	$\vec{c}' \leftarrow \frac{\vec{c} + \vec{c}_*}{2} + \frac{\ \vec{c} - \vec{c}_*\ }{2} \vec{u}'$;
8	$\vec{c}'_* \leftarrow \frac{\vec{c} + \vec{c}_*}{2} - \frac{\ \vec{c} - \vec{c}_*\ }{2} \vec{u}'$;
9	estimer $f(\vec{c}'; t')$ grâce à cet algorithme : on obtient $\tilde{F}_1(\vec{c}'; t')$;
10	estimer $f(\vec{c}'_*; t')$ grâce à cet algorithme : on obtient $\tilde{F}_2(\vec{c}'_*; t')$; // de façon indépendante à $\tilde{F}_1(\vec{c}'; t')$
11	retourner $(2\pi)^{3/2} \exp(\vec{c}_*^2/2) \times \tilde{F}_1(\vec{c}'; t') \times \tilde{F}_2(\vec{c}'_*; t')$;

Algorithme 5.1 : Algorithme servant à estimer $f(\vec{c}; t)$, valide dans les conditions du mode BKW décrites en section 5.1.2 (densité uniforme et unité, particules de Maxwell, ...).

En particulier, si $f(\vec{c}; t_{init}) = f_{BKW}(\vec{c}; t_{init})$, alors on a $f(\vec{c}; t) = f_{BKW}(\vec{c}; t)$. $f(\vec{c}; t)$ est supposée connue à un temps t_{init} .

5.3 Premiers résultats obtenus

5.3.1 Campagne de tests, et résultats

●●●● On a testé l'algorithme 5.1 en comparant ses résultats avec la formule (5.3). Plus précisément, on donne comme condition initiale dans l'algorithme 5.1 « $f(\vec{c}; t_d) = f_{BKW}(\vec{c}; t_d)$ », avec $t_d = 6 \ln^{5/2}$, puis on teste si l'algorithme estime correctement $f = f_{BKW}$ quand $t \geq t_d$. Pour mémoire, en se plaçant au temps initial t_d dans l'expression (5.3) on obtient :

$$f_{BKW}(\vec{c}; t_d) = \frac{25\sqrt{5}}{54\sqrt{6}\pi^{3/2}} \vec{c}^2 \exp\left(-\frac{5}{6} \vec{c}^2\right) \quad (5.12)$$

Cette expression est utilisée dans les codes ayant servi à cette campagne de tests, et présentés dans l'annexe D.3.

- Nous avons demandé à l'algorithme 5.1 de réaliser 2 types de calcul de f :
- des calculs sondes, c'est-à-dire des calculs de $f(\vec{c}; t)$ en divers points $(\vec{c}; t)$ donnés,
 - des calculs de la fraction des particules dont la vitesse dépasse un seuil, en certains temps t donnés.

Comme expliqué dans le chapitre 1, quand on utilise des méthodes de Monte-Carlo suivant les particules en sens inverse — comme c'est le cas ici — le calcul sonde est le test unique de fonctionnalité des algorithmes mis au point. Étant donné que les diverses grandeurs macroscopiques (densité de matière, densités de flux, fraction de la matière dans une partie du volume, ...) ne sont que des intégrales à 1 ou quelques étages de la fonction de distribution f , si on est capable de réaliser le calcul de $f(\vec{r}; \vec{c}; t)$ en tout point de l'espace-temps des phases — ce qu'on appelle le calcul sonde — alors on est capable de calculer toute grandeur macroscopique. Néanmoins, réaliser le calcul de fractions des particules dont l'énergie cinétique dépasse un seuil présente pour nous 2 intérêts :

- s'assurer qu'aucun problème n'apparaît durant le calcul de f quand la vitesse est élevée. Plusieurs calculs sondes en des points de l'espace des phases à $\|\vec{c}\|$ élevée auraient cependant suffi à cela.
- pouvoir affirmer, sans laisser subsister le moindre doute, que nous sommes capables de calculer ce qui se passe à haute vitesse.

●●●● Cette capacité à calculer correctement les événements de haute énergie (et plus largement, les événements rares) revêt pour nous une importance particulière. Ce type de calcul est connu pour mettre en difficulté toutes les méthodes numériques disponibles en cinétique des gaz — que ce soit Boltzmann sur Réseau, Direct Simulation Monte-Carlo, ... — ; pourtant, au niveau applicatif il sont très demandés,

car c'est à haute énergie que se produisent les phénomènes « intéressants » — il peut s'agir de réactions chimiques, d'ionisation, de dégradation de matériaux, . . .

Quand nous avons conceptualisé la méthode de calcul de f que nous présentons dans ce manuscrit, nous n'avons pas cherché spécifiquement à être capables de calculer les événements de haute énergie. Nous venions de lancer le travail présenté en [49], et il nous semblait extensible à la cinétique des gaz. C'est ainsi que ce travail a commencé. En progressant dans nos recherches et nos essais, nous avons constaté comme les problèmes de haute énergie peuvent être traités aisément par cette voie (les résultats de simulation en témoignent) ; même si en étant au stade purement théorique, nous aurions pu soupçonner cette capacité, car les méthodes de calcul de type Monte-Carlo utilisant un suivi des particules en sens inverse sont connues pour cela [1].

Finalement, c'est pour nous l'occasion de proposer dès maintenant une application à ce travail de recherche : le calcul des événements rares, et en particulier de haute énergie, en cinétique des gaz. Connaissant pour partie les problématiques auxquelles se confrontent les communautés scientifiques des gaz raréfiés et des plasmas, nous imaginons que cela pourrait les intéresser — plus rapidement que le calcul de référence, qui est une de nos motivations plus habituelles.

●●●● Pour connaître la fraction des particules dont la vitesse dépasse un certain seuil c_0 , nous devons évaluer l'intégrale :

$$Frac(\|\vec{c}\| \geq c_0; t) = \int_{E_c} d\vec{c} H(\|\vec{c}\| \geq c_0) f(\vec{c}; t) \quad (5.13)$$

$$= \int_{c_0}^{+\infty} dc_f \int_{4\pi} c^2 d\vec{u}_f f(c_f \vec{u}_f; t) \quad (5.14)$$

donc ajouter une (en fait 3) intégration au calcul de f .

On pose donc les variables aléatoires C_f et \vec{U}_f à valeurs dans $[c_0; +\infty[$ et $\mathcal{S}(E_c; \vec{0}; 1)$ respectivement, de sorte que :

$$Frac(\|\vec{c}\| \geq c_0; t) = \int_{c_0}^{+\infty} p_{C_f}(c_f) dc_f \int_{4\pi} p_{\vec{U}_f}(\vec{u}_f) d\vec{u}_f \frac{f(c_f \vec{u}_f; t) c_f^2}{p_{C_f}(c_f) p_{\vec{U}_f}(\vec{u}_f)} \quad (5.15)$$

$$= E \left(\frac{\tilde{F}(C_f \vec{U}_f; t) C_f^2}{p_{C_f}(C_f) p_{\vec{U}_f}(\vec{U}_f)} \right) \quad (5.16)$$

où $\tilde{F}(\vec{c}; t)$ est un estimateur de $f(\vec{c}; t)$. Cela nous amène à l'algorithme 5.2, qui comptabilise la fraction des particules de vitesse élevée.

Pour réaliser les calculs, on choisit arbitrairement :

$$\begin{cases} p_{C_f}(c_f) = \frac{c_0(c_0^2 - 2)}{\left((c_0^2 - 2)(c_f - c_0) + c_0\right)^2} & \text{si } c_0 > 2 \\ p_{C_f}(c_f) = \frac{2}{(2 + c_f - c_0)^2} & \text{si } c_0 \leq 2 \\ p_{\vec{U}_f}(\vec{u}_f) = \frac{1}{4\pi} \end{cases} \quad (5.17)$$

Entrées : Une vitesse c_0 et un temps t

Sorties : Une estimation ponctuelle de la fraction des particules dont la vitesse dépasse c_0 au temps t

- 1 échantillonner C_f : on obtient c_f ;
- 2 échantillonner \vec{U}_f : on obtient \vec{u}_f ;
- 3 estimer $f(c_f \vec{u}_f; t)$ grâce à l'algorithme 5.1 : on obtient $\tilde{f}(c_f \vec{u}_f; t)$;
- 4 retourner $\frac{c_f^2 \tilde{f}(c_f \vec{u}_f; t)}{p_{C_f}(c_f) p_{\vec{U}_f}(\vec{u}_f)}$;

Algorithme 5.2 : Algorithme servant à estimer la fraction des particules dont, à un instant fixé, la vitesse dépasse un certain seuil ; valide dans les conditions du mode BKW décrites en section 5.1.2.

●●●● Les résultats de simulation sont montrés en figures 5.2, 5.3, et 5.4.

5.3.2 Analyse des résultats

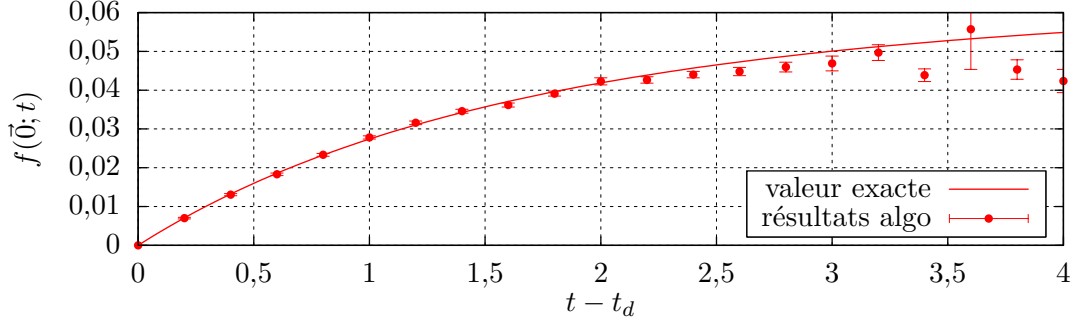
●●●● La 1^{re} constatation est que les résultats exhibent un comportement évoquant une variance d'estimateur infinie, comme dans la figure 4.7 au chapitre précédent. En comparant les profils des résultats obtenus en 4.7 et maintenant, on peut penser que la variance devient infinie vers $t = t_d + 2$. Nous n'avons cependant pas réalisé de modélisation de la variance afin de confirmer ou d'infirmer une telle intuition.

Le comportement des résultats fait penser au chapitre 4.3.5, quand on réalisait le calcul de $y(t)$ avec un \hat{k} trop faible. Ici, on est plutôt dans le cas « inquiétant » évoqué au chapitre 4.3.5, c'est-à-dire que la variance relative estimée augmente progressivement. En réalité, la variance vraie de notre estimateur de $f(\vec{c}; t)$ doit diverger à un certain t fini, mais on ne sait pas lequel.

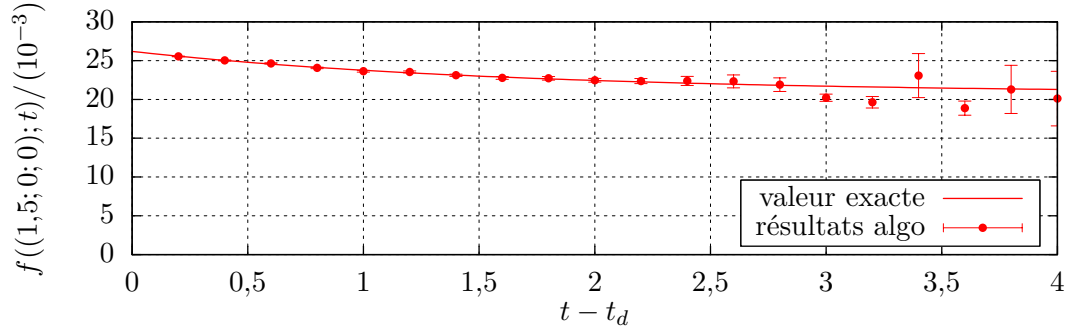
●●●● À certains égards, les résultats présentés ici sont encore plus inquiétants qu'au chapitre 4.3.5. Quand la variance d'estimateur est infinie, ils exhibent un comportement étonnant : certaines estimations tombent largement trop haut et avec une grande barre d'erreur, alors que les autres tombent trop bas avec une barre d'erreur faible, pouvant amener à fournir — sans se douter de rien — des intervalles de confiance faux. Quand on pousse les simulations à des temps non montrés dans les figures 5.2, 5.3, et 5.4, ce comportement s'accroît encore, jusqu'à ce que toutes les estimations soient bien trop faibles.

Ce comportement surprenant est probablement lié à la positivité de notre estimateur de $f(\vec{c}; t)$ (une lecture de l'algorithme 5.1 ou de la formulation 5.10 laisse deviner cette positivité). À cause de celle-ci, une grande dispersion de l'estimateur aboutit nécessairement à ce que dans sa répartition :

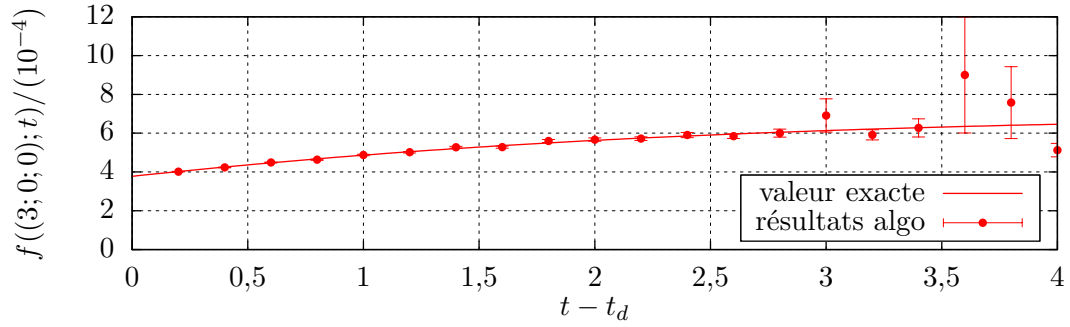
- on ait une « longue queue » dans les grands réels positifs, qui provoque l'infinité de la variance,



(a) Profil de $f_{BKW}(\vec{c}; t)$ en $\vec{c} = \vec{0}$ et en fonction de t



(b) Profil de $f_{BKW}(\vec{c}; t)$ en $\vec{c} = (1,5; 0; 0)$ et en fonction de t



(c) Profil de $f_{BKW}(\vec{c}; t)$ en $\vec{c} = (3; 0; 0)$ et en fonction de t

FIGURE 5.2 – Profils de $f_{BKW}(\vec{c}; t)$ au cours du temps, et en différents points de l'espace des vitesses. La solution symbolique est donnée en trait plein, et en points séparés avec barres d'erreur sont donnés les résultats de l'algorithme 5.1, travaillant avec la condition initiale « $f(\vec{c}; t_d)$ est connue » (voir section 5.3). En chaque point, 10^4 réalisations ont été effectuées.

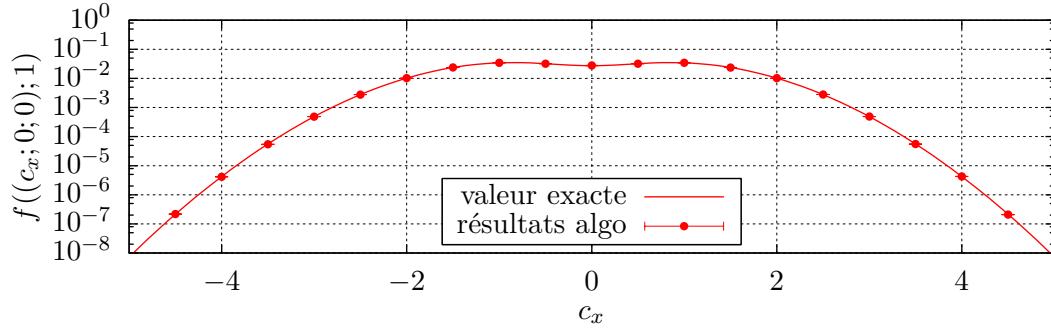
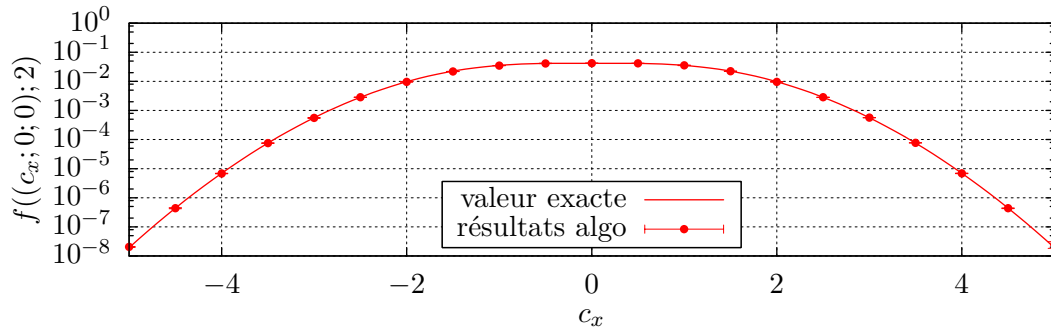
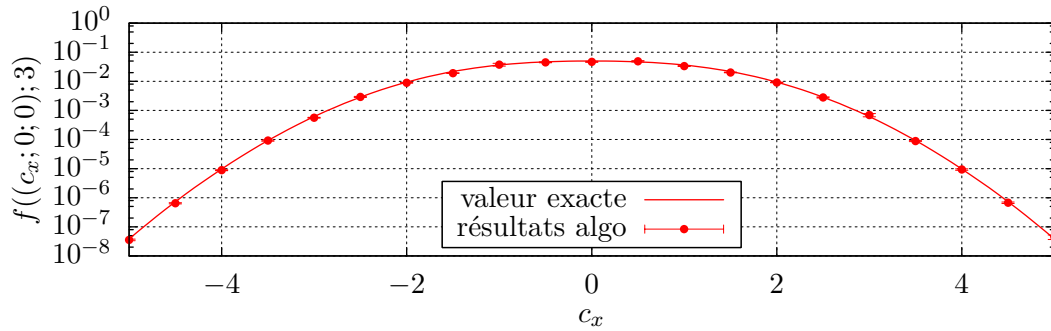
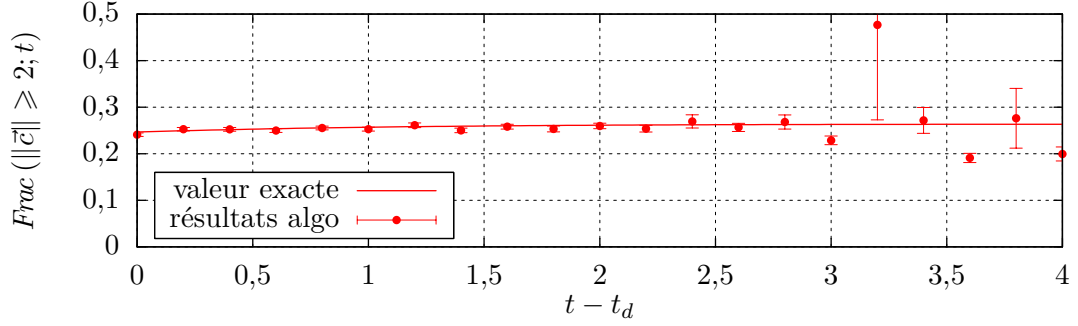
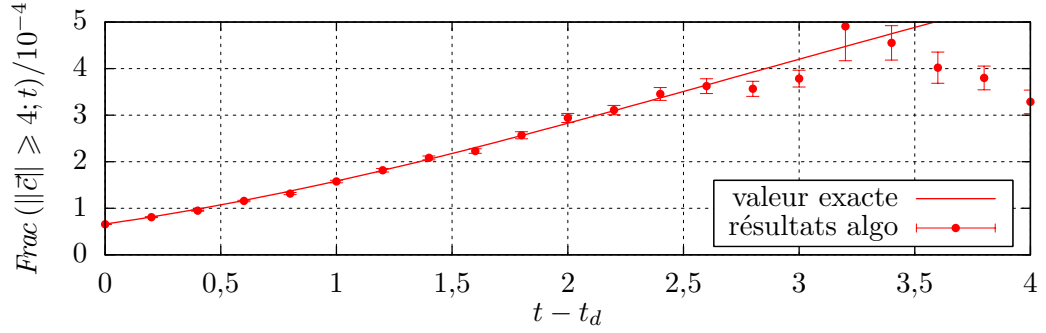
(a) Profil de $f_{BKW}((c_x; 0; 0); t)$ à $t = t_d + 1$ (b) Profil de $f_{BKW}((c_x; 0; 0); t)$ à $t = t_d + 2$ (c) Profil de $f_{BKW}((c_x; 0; 0); t)$ à $t = t_d + 3$

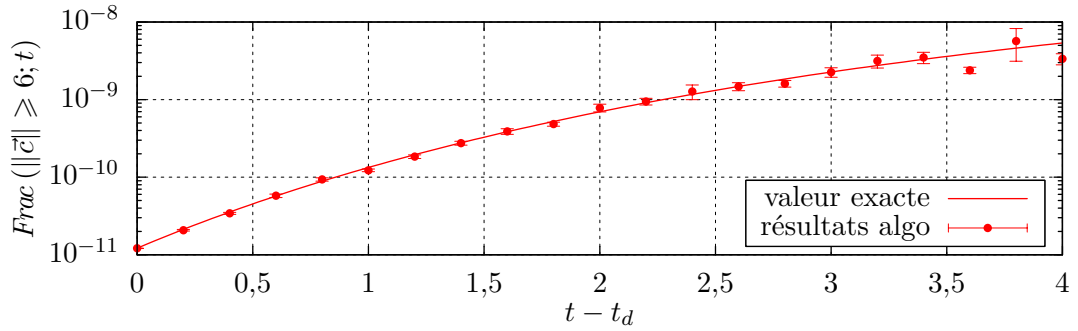
FIGURE 5.3 – Profils de f_{BKW} le long de l'axe (Ox) de l'espace des vitesses, à différents t fixés. La solution symbolique est donnée en trait plein, et en points séparés avec barres d'erreur sont donnés les résultats de l'algorithme 5.1, travaillant avec la condition initiale « $f(\vec{c}; t_d)$ est connue » (voir section 5.3). En chaque point, 10^4 réalisations ont été effectuées.



(a) Fraction des particules dont la vitesse dépasse $c_0 = 2$.



(b) Fraction des particules dont la vitesse dépasse $c_0 = 4$.



(c) Fraction des particules dont la vitesse dépasse $c_0 = 6$.

FIGURE 5.4 – Évolution selon le temps de fractions des particules dont la vitesse dépasse différents seuils c_0 donnés. La solution symbolique est donnée en trait plein, et en points séparés avec barres d'erreur sont donnés les résultats de l'algorithme 5.2, travaillant avec la condition initiale « $f(\vec{c}; t_d)$ est connue » (voir section 5.3). En chaque point, 10^4 réalisations ont été effectuées.

- la grande majorité de la « masse » soit proche de 0, compensant cette longue queue de sorte que l'espérance soit toujours correcte.

●●●● Peut-être aurions-nous pu prédire cette infinité de la variance de l'estimateur $f(\vec{c}; t)$, en considérant l'équation d'évolution (5.2). Sous la forme adimensionnalisée dans laquelle nous l'avons mise, elle est en effet *instable en densité*. C'est-à-dire qu'en reprenant la définition de la densité : $\eta = \int_{E_c} d\vec{c} f$, l'équation (5.2) amène à :

$$\partial_t \eta = \eta^2 - \eta \quad (5.18)$$

qui est clairement instable autour de $\eta = 1$. Selon la valeur initiale de η , soit on tend asymptotiquement vers 0, soit on arrive à $\eta \rightarrow +\infty$ au bout d'un temps fini !

Bien sûr cette instabilité en densité de l'équation (5.2) ne se ressent pas dans le comportement du phénomène physique étudié, car l'adimensionnalisation qui amène à l'équation (5.2) impose $\eta = 1$ exactement. Avec une condition initiale imposant $\eta = 1$, l'équation (5.2) amène bien à une densité η constante, et n'est donc pas en contradiction avec la physique qu'elle modélise.

Par contre, l'algorithme 5.1 qui la résout ignore la particularité de la condition initiale. Si $\eta \neq 1$ dans la condition initiale qui lui est donnée, il calculera une divergence sur η , suivant strictement l'équation (5.2). Mais comme l'algorithme 5.1 ne connaît pas exactement la densité initiale mais ne fait que l'échantillonner, il ne peut pas aboutir à ce que $\eta = 1$ exactement ; donc ses résultats divergent systématiquement (à part dans la situation de variance nulle décrite à la section 5.2.2).

●●●● La dernière chose que nous constatons est que conformément à ce que nous avons annoncé, la technique de calcul de $f(\vec{c}; t)$ que nous avons mise au point n'est pas mise en échec à grande vitesse. On peut le constater en premier lieu sur la figure 5.3 : l'écart-type d'estimation reste faible quand la norme de \vec{c} est grande (disons supérieure à 3).

Aussi, les calculs de fractions des particules à haute vitesse sont une réussite, comme en témoigne la figure 5.4. On observe certes que la variance relative des résultats de ces calculs est plus élevée, mais il n'y a rien de grave. Sûrement, cette augmentation de la variance est partiellement due à l'intégration finale sur $E_c \setminus \mathcal{B}(E_c; \vec{0}; c_0)$, car nous n'avons pas vraiment optimisé le tirage de C_f dans l'expression (5.16). On calcule sans difficulté des fractions de la masse totale inférieures à un milliardième (voir la figure 5.4c).

À ce niveau, nous repensons évidemment aux applications éventuelles précédemment listées — chimie gazeuse, physique des plasmas, fusion nucléaire... À notre connaissance, notre capacité à prévoir les fractions de particules de grande énergie cinétique, même quand elles sont aussi faibles qu'un milliardième, constitue une première. Il n'y a d'ailleurs *a priori* pas de limitation sur la petitesse des fractions que nous calculons.

●●●● Nous n'avons jusqu'ici pas parlé de la récursivité de l'algorithme 5.1. En fait, elle suit exactement la même évolution que la récursivité de l'algorithme 4.1 : comparer les formulations statistiques 5.10 (avec les choix de la section 5.2.2) et 4.10 permet de s'en convaincre. Plus exactement, avec la formulation 5.10 on a $\hat{k} = 1$; donc en reprenant l'expression (4.13) :

$$rec(t) = 2e^t - 1 \quad (5.19)$$

où $rec(t)$ est l'espérance du nombre total de tirages de temps de recul T' , quand on fait une estimation ponctuelle de $f(\vec{c}; t)$ par l'algorithme 5.1 (la valeur de \vec{c} ne joue pas).

Par exemple, si on considère que pour chaque tirage d'un temps de recul T' on effectue au plus 7 tirages uniformes standard (pour tirer T' , \vec{C}_* et \vec{U}' , voir l'annexe D.3), alors pour estimer $f(\vec{c}; t)$ avec un échantillon de 10^4 réalisations en suivant l'algorithme 5.1 il faudra moins de 10^6 tirages uniformes standards. Si on considère que ces tirages représentent la majorité de la dépense computationnelle, celle-ci est plutôt faible. Sur nos graphes de 5.2 à 5.4, nous avons effectivement utilisé 10^4 réalisations par points. Même si on considère les barres d'erreur trop larges à $t = 2$, rétrécir l'écart-type d'un facteur 10 est très abordable sur un ordinateur de bureau.

Bien sûr, à des temps physiques t simulés plus grands, la dépense computationnelle augmente de façon exponentielle. Nous en parlerons toutefois dans la section suivante, car à des temps simulés supérieurs à 2 nous sommes plutôt limités par un problème d'explosion de la variance.

5.4 Une astuce algorithmique pour accroître les capacités de la méthode développée

●●●● Au total, la méthode développée donne des résultats intéressants, mais il y a de quoi être frustré par les faibles temps qu'elle peut simuler, avant que la variance de ses résultats ne devienne infinie. Nous proposons ici une façon peu élégante d'améliorer cette situation, inspirée de la situation de variance nulle décrite en section 5.2.2.

On se base sur le constat que si la distribution initiale $f(\vec{c}; t_{init})$ était la distribution d'équilibre, alors notre algorithme donnerait un estimateur de variance nulle. Cela amène à penser que si l'état initial, qu'on a choisi comme $f(\vec{c}; t_d) = f_{BKW}(\vec{c}; t_d)$, était plus proche de l'état d'équilibre alors notre estimateur divergerait plus lentement. Or justement, l'équilibre est l'état final du mode BKW.

Au bout d'un temps simulé $t - t_{init}$ non nul, on s'est rapproché de l'équilibre. Cependant, notre estimateur de $f(\vec{c}; t)$ est maintenant entaché de variance, alors qu'à t_{init} il ne l'est pas (par hypothèse) ; et c'est par auto-multiplication de notre estimateur dans l'algorithme 5.1 que sa variance finit par exploser (comme au chapitre 1.5.3). Ce serait donc bien si au bout d'un temps simulé non nul on faisait baisser la variance de l'estimateur ; de la sorte, notre capacité d'estimer $f(\vec{c}; t)$ irait plus loin dans les temps simulés.

●●●● Une façon simple de faire baisser la variance d'un estimateur est de l'évaluer plusieurs fois, puis d'en prendre la moyenne. De la sorte, on passe de l'expression statistique (5.10) :

$$f(\vec{c}; t) = \mathbb{E} \left(\frac{e^{-(t-T')}}{p_{T'}(T')} \left(\mathbb{H}(T' \leq t_{init}) f(\vec{c}; t_{init}) + \mathbb{H}(T' > t_{init}) \frac{\tilde{F}_1(\vec{C}'; T') \tilde{F}_2(\vec{C}_*'; T')}{4\pi p_{\vec{C}_*}(\vec{C}_*) p_{\vec{U}'}(\vec{U}')} \right) \right)$$

à

$$f(\vec{c}; t) = \mathbb{E} \left(\frac{e^{-(t-T')}}{p_{T'}(T')} \left(\mathbb{H}(T' \leq t_{init}) f(\vec{c}; t_{init}) + \mathbb{H}(T' > t_{init}) \times \right. \right. \\ \left. \left. \frac{1}{N_m} \sum_{i=1}^{N_m} \frac{\tilde{F}_1(\vec{C}'_i; T') \tilde{F}_2(\vec{C}_*'; T')}{4\pi p_{\vec{C}_*}(\vec{C}_*) p_{\vec{U}'_i}(\vec{U}'_i)} \right) \right) \quad (5.20a)$$

où

$$\vec{C}'_i = \frac{\vec{c} + \vec{C}_{*i}}{2} + \frac{\|\vec{c} - \vec{C}_{*i}\|}{2} \vec{U}'_i \quad \vec{C}_*'_i = \frac{\vec{c} + \vec{C}_{*i}}{2} - \frac{\|\vec{c} - \vec{C}_{*i}\|}{2} \vec{U}'_i \quad (5.20b)$$

et N_m est un paramètre réglable, choisi coup sur coup en fonction d'au moins t et T' . Nous le choisirons égal à 1, sauf quand le suivi de particule traverse certains paliers en remontant le temps. Cela nous mène à l'algorithme 5.3.

5.5 Deuxième groupe de résultats obtenus

Pour cette campagne de tests, nous choisissons l'expression suivante pour N_m :

$$\begin{cases} \text{si } 1,2 \in [t'; t] & 9 \\ \text{sinon si } 2,5 \in [t'; t] & 4 \\ \text{sinon si } 4 \in [t'; t] & 4 \\ \text{sinon} & 1 \end{cases} \quad (5.21)$$

C'est-à-dire que N_m vaut 1, sauf si en faisant une remontée dans le temps (voir la formulation statistique 5.10) on traverse certains paliers. Si on franchit le palier $T' \leq 1,2$, on effectuera l'estimation de $s(\vec{c}; T')$ ¹ avec $N_m = 9$ échantillons. Sinon, si on franchit les paliers $T' \leq 2,5$ ou $T' \leq 4$, on procède de même avec $N_m = 4$.

L'implémentation de l'algorithme 5.3 qui en découle est détaillée en annexe D.3. Les résultats de la campagne de tests, menée avec des conditions similaires à celles de la précédente (section 5.3), sont donnés dans les figures 5.5, 5.6, 5.7, et 5.8.

Apparemment, l'estimateur sortant de l'algorithme 5.3, dans les conditions de cette campagne de tests, voit désormais sa variance exploser autour de $t - t_d = 5,5$, si on suit les figures. À moment-là, une estimation de $f(\vec{c}; t)$ basée sur un échantillon de 10^4 réalisations de l'algorithme 5.3 demandera entre 10^9 et 10^{10} tirages uniformes standards : ça commence à faire beaucoup, pour une machine de bureau actuelle.

1. voir la définition de s dans l'expression (5.6)

Entrées : Le point $(\vec{c}; t)$ où on veut estimer $f(\vec{c}; t)$
Sorties : Une estimation ponctuelle de $f(\vec{c}; t)$

- 1 échantillonner T_r selon une loi exponentielle unité : on obtient t_r ;
- 2 $t' \leftarrow t - t_r$;
- 3 **si** $t' \leq t_{init}$ **alors retourner** $f(\vec{c}; t_{init})$;
- 4 **sinon**
- 5 déterminer N_m en fonction de t et t' : on obtient n_m ;
- 6 $\Sigma_f \leftarrow 0$;
- 7 **pour** $i \leftarrow 1$ **à** n_m **faire**
- 8 échantillonner \vec{C}_* , selon la loi normale centrée réduite tridimensionnelle : on obtient \vec{c}_* ;
- 9 échantillonner \vec{U}' de façon isotrope : on obtient \vec{u}' ;
- 10 $\vec{c}' \leftarrow \frac{\vec{c} + \vec{c}_*}{2} + \frac{\|\vec{c} - \vec{c}_*\|}{2} \vec{u}'$;
- 11 $\vec{c}'_* \leftarrow \frac{\vec{c} + \vec{c}_*}{2} - \frac{\|\vec{c} - \vec{c}_*\|}{2} \vec{u}'$;
- 12 estimer $f(\vec{c}'; t')$ grâce à cet algorithme : on obtient $\tilde{F}_1(\vec{c}'; t')$;
- 13 estimer $f(\vec{c}'_*; t')$ grâce à cet algorithme : on obtient $\tilde{F}_2(\vec{c}'_*; t')$;
- 14 $\Sigma_f \leftarrow \Sigma_f + (2\pi)^{3/2} \exp(\vec{c}'_*^2/2) \times \tilde{F}_1(\vec{c}'; t') \times \tilde{F}_2(\vec{c}'_*; t')$;
- 15 **retourner** (Σ_f/n_m) ;

Algorithme 5.3 : Un nouvel algorithme servant à estimer $f(\vec{c}; t)$, valide dans les conditions du mode BKW décrites en section 5.1.2

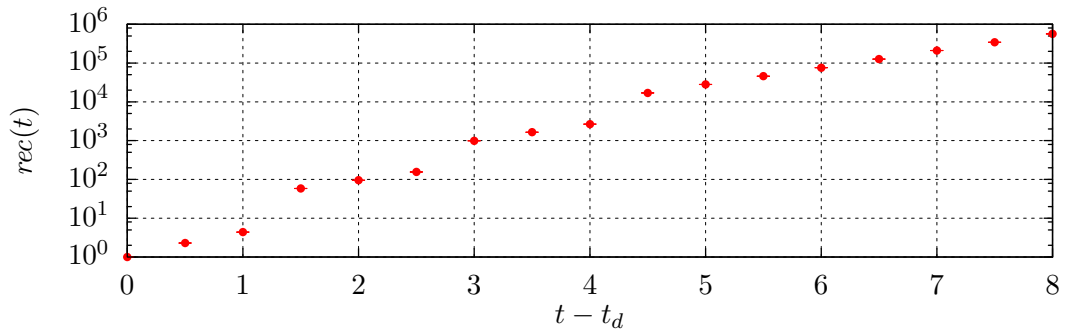


FIGURE 5.5 – Évolution de la récursivité de l'algorithme 5.3 selon le temps physique t simulé. La récursivité $rec(t)$ est celle définie à la fin de la section 5.3.2.

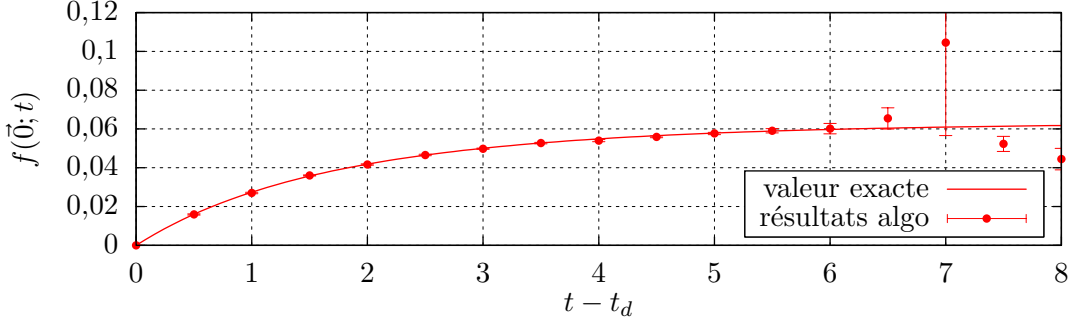
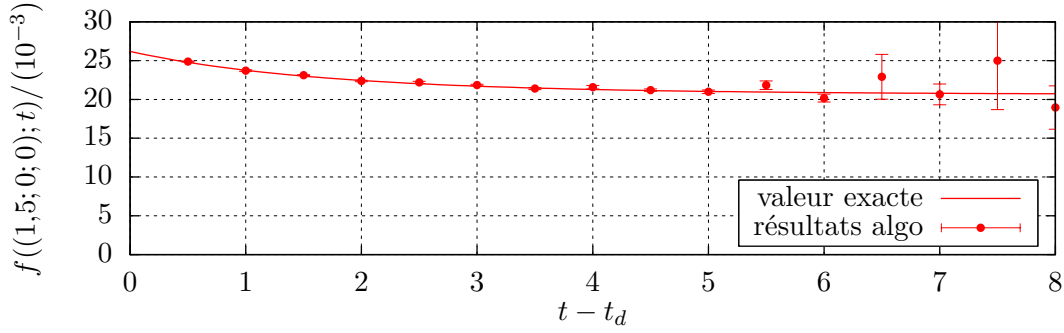
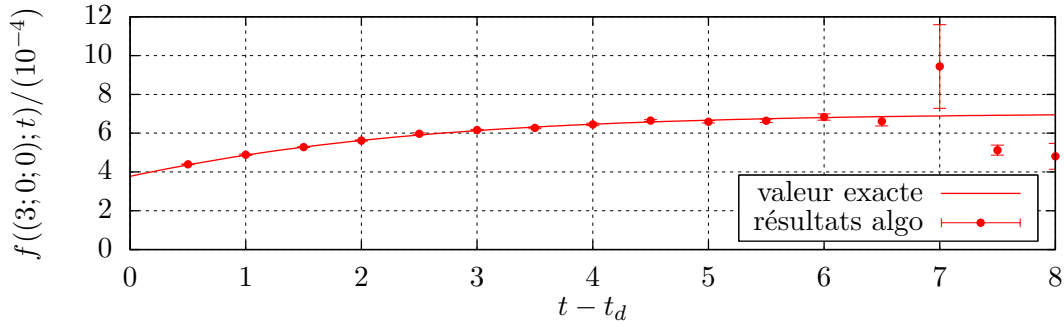
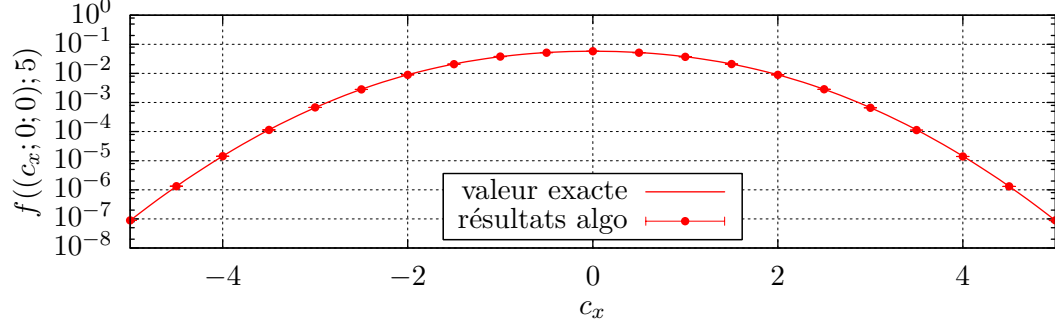
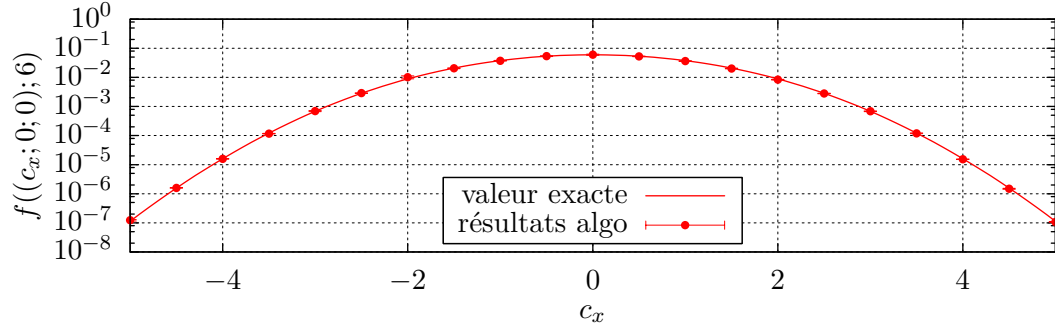
(a) Profil de $f_{BKW}(\vec{c}; t)$ en $\vec{c} = \vec{0}$ et en fonction de t (b) Profil de $f_{BKW}(\vec{c}; t)$ en $\vec{c} = (1,5; 0; 0)$ et en fonction de t (c) Profil de $f_{BKW}(\vec{c}; t)$ en $\vec{c} = (3; 0; 0)$ et en fonction de t

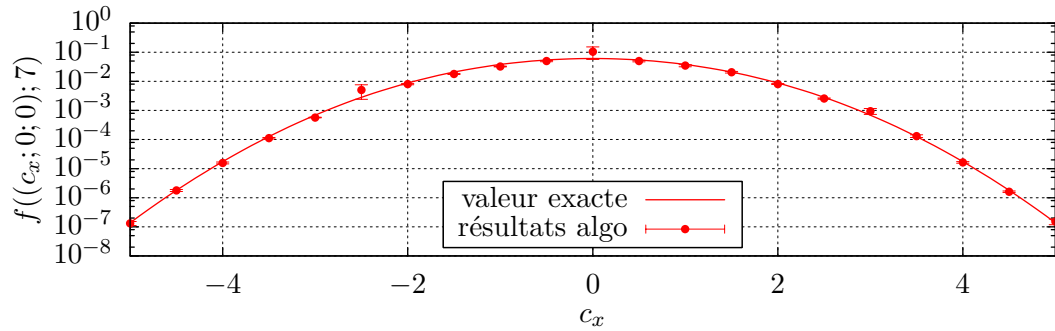
FIGURE 5.6 – Profils de $f_{BKW}(\vec{c}; t)$ au cours du temps, et en différents points de l'espace des vitesses. La solution symbolique est donnée en trait plein, et en points séparés avec barres d'erreur sont donnés les résultats de l'algorithme 5.3, travaillant avec la condition initiale « $f(\vec{c}; t_d)$ est connue ». En chaque point, 10^4 réalisations ont été effectuées.



(a) Profil de $f_{BKW}((c_x; 0; 0); t)$ à $t = t_d + 5$



(b) Profil de $f_{BKW}((c_x; 0; 0); t)$ à $t = t_d + 6$



(c) Profil de $f_{BKW}((c_x; 0; 0); t)$ à $t = t_d + 7$

FIGURE 5.7 – Profils de f_{BKW} le long de l'axe (Ox) de l'espace des vitesses, à différents t fixés. La solution symbolique est donnée en trait plein, et en points séparés avec barres d'erreur sont donnés les résultats de l'algorithme 5.3, travaillant avec la condition initiale « $f(\vec{c}; t_d)$ est connue ». En chaque point, 10^4 réalisations ont été effectuées.

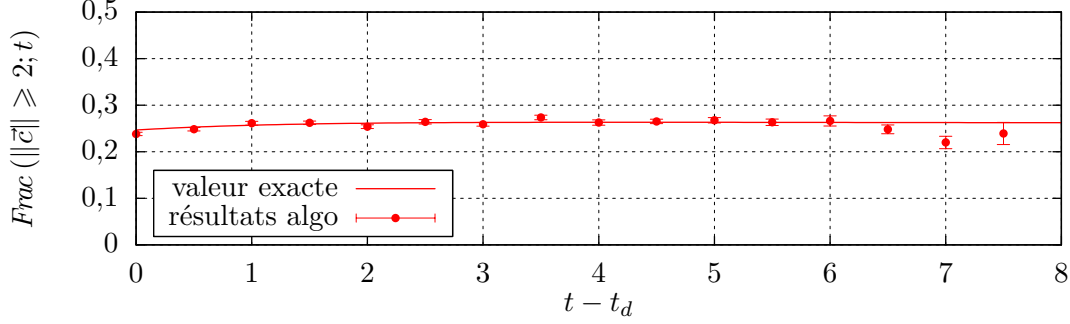
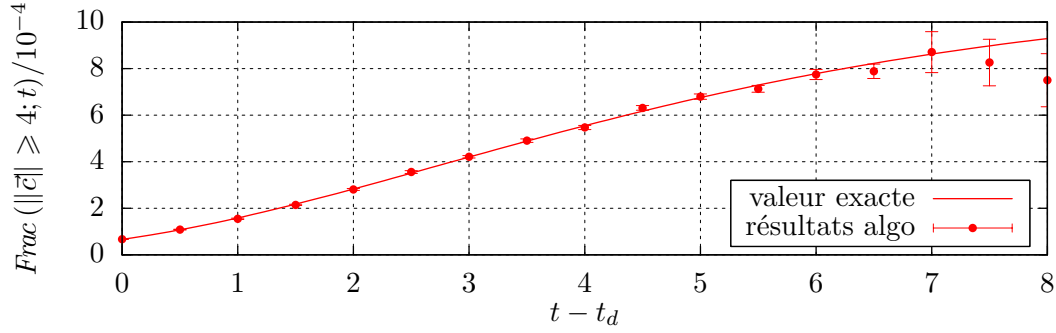
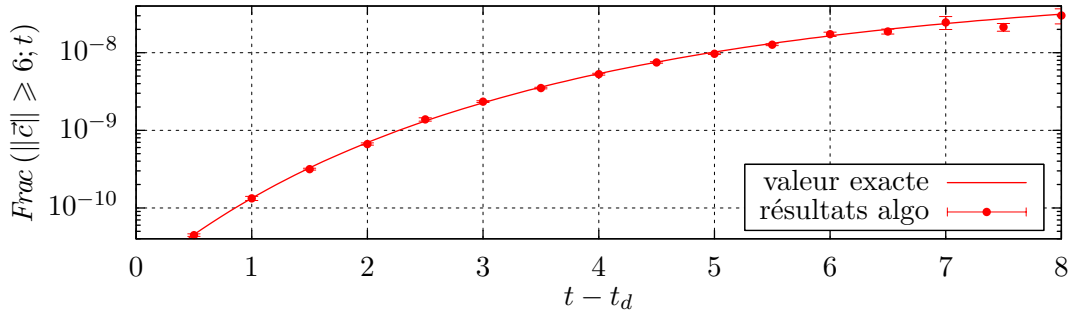
(a) Fraction des particules dont la vitesse dépasse $c_0 = 2$.(b) Fraction des particules dont la vitesse dépasse $c_0 = 4$.(c) Fraction des particules dont la vitesse dépasse $c_0 = 6$.

FIGURE 5.8 – Évolution selon le temps de fractions des particules dont la vitesse dépasse différents seuils c_0 donnés. La solution symbolique est donnée en trait plein, et en points séparés avec barres d'erreur sont donnés les résultats de l'algorithme 5.3 intégrés par l'algorithme 5.2, travaillant avec la condition initiale « $f(\vec{c}; t_d)$ est connue ». En chaque point, 10^4 réalisations ont été effectuées.

Chapitre 6

Résultats obtenus sur un nuage de gaz oscillant

On résout l'équation de Boltzmann par MMC dans un deuxième cas académique de cinétique des gaz, où une solution symbolique est disponible. Dans ce cas-ci, le gaz n'est plus uniforme, il forme un nuage autour de l'origine des positions, qui se contracte et se détend périodiquement. En plus de ce que nous avons réalisé au chapitre précédent, nous devons donc tenir compte du transport balistique dans l'espace des positions. Nous devons aussi ajouter la gestion d'une fréquence de collision variable, alors que le cas d'étude du précédent chapitre permettait de nous en dispenser.

Plusieurs alternatives algorithmiques apparaissent quand on veut gérer la somme de l'extinction et du terme source dans le terme collisionnel de Boltzmann. Nous proposerons plusieurs choix réalisables, dont nous comparerons la pertinence à l'aune d'une approche à variance nulle (voir le chapitre 1.3.2) dans laquelle nous supposerons la distribution des vitesses à l'équilibre. Puis nous testerons numériquement ces choix.

●●●● Nous allons montrer dans ce chapitre la résolution de l'équation de Boltzmann dans un 2^e cas académique. Ce dernier est détaillé dans la section 6.1, et possède comme le précédent une solution symbolique.

Dans ce nouveau cas, le gaz n'est pas uniforme, ce qui va nous obliger à prendre en compte les 2 aspects du transport que nous avons mis de côté au chapitre précédent :

- le transport balistique dans l'espace des positions,
- la variabilité de la fréquence de collision.

Pour tenir compte du transport entre les collisions, il nous suffira de mettre à jour la position des particules dans l'espace entre 2 collisions, ce que nous n'avions pas fait au chapitre précédent. Un champ de forces extérieures est présent dans

le cas d'étude ; heureusement, il est suffisamment simple pour que les trajectoires balistiques soient aisément paramétrisables.

La fréquence de collision variable, et dépendante de la fonction de distribution, est gérée en reprenant presque à l'identique la démarche du chapitre 4. L'espace des vitesses est cependant ajouté (comme au chapitre 5), et l'extinction due aux collisions agit en parallèle au terme source.

●●●● À ce stade, nous avons jugé utile de donner une formulation intégrale générique de l'équation de Boltzmann, pour en tirer ensuite la formulation particulière à notre problème. Ceci est fait dans la section 6.2.

Après cela, nous nous attardons sur la conversion de l'écriture intégrale en écriture statistique détaillée, et sur les alternatives que ce passage fait apparaître. Nous faisons des prédictions sur la variance de quelques estimateurs possibles, que finalement nous testons numériquement à la section 6.3.

●●●● La campagne de tests de la section 6.3 fait apparaître les mêmes comportements numériques qu'au chapitre précédent.

Aucune augmentation significative de la variance ne se produit quand nos estimateurs sondent des zones peu peuplées de l'espace des phases. Le cas d'étude de ce chapitre permet de tester ceci, dans les zones peu peuplées de l'espace des positions (Nous l'avons déjà testé aux grandes vitesses. Le présent cas d'étude n'a pas invalidé ce qu'on avait alors déduit.). En effet, parce que le champ de forces extérieures rappelle les particules à l'origine des positions, la masse est concentrée autour de celle-ci : les zones éloignées de l'origine sont dépeuplées, et nous testons nos estimateurs de f en particulier là.

Par ailleurs, la même fulgurante augmentation de la variance avec le temps physique simulé qu'au chapitre 5 est constatée. Il est à noter que les choix algorithmiques que l'on peut faire, à la section 6.2.3, influencent beaucoup la durée au bout duquel la variance explose.

6.1 Description de la situation physique, dite « de confinement harmonique »

6.1.1 Cas général

●●●● La situation de confinement harmonique est connue depuis les travaux de Ludwig BOLTZMANN lui-même.

Le théorème H explique que, lors des collisions, la quantité $\int_{E_r \times E_c} f \times \ln f$ ne peut que décroître — d'où son nom d'*entropie* en cinétique des gaz. La situation de confinement harmonique est, avec l'équilibre thermodynamique global, la seule situation connue où il n'y a pas de destruction de cette entropie. C'est-à-dire que c'est une situation sans dissipation, où les collisions ne jouent aucun rôle ; on dit qu'elle fait partie du *noyau de l'opérateur de collision*.

La combinaison de la minimisation de l'entropie et de la conservation des 3 premiers moments de f selon la vitesse \vec{c} amène à rechercher des solutions permanentes à l'équation de Boltzmann sous la forme (voir le chapitre 1.1.3.3) :

$$f(\vec{r}; \vec{c}; t) = \exp(-\alpha - \beta \vec{c}^2 - \vec{\gamma} \cdot \vec{c}) \quad (6.1)$$

où $\beta > 0$, α , et $\vec{\gamma}$ sont *a priori* des fonctions de l'espace et du temps. L'équilibre thermodynamique global est en effet décrit par une telle expression de f . Une autre solution moins connue existe, à condition que le gaz soit monoatomique et que le champ des forces extérieures suive un certain type de profil.

●●●● Nous reprendrons ici directement les résultats de la publication [53], de David GUÉRY-ODELIN, Juan Gonzalo MUGA, María José RUIZ MONTERO, et Emmanuel TRIZAC, sans les calculs détaillés.

On s'intéresse aux situations où la somme des forces à longue portée découle d'un potentiel scalaire. La distribution des particules peut appartenir au noyau de l'opérateur de collision de Boltzmann — quelle que soit la section efficace de collision — sans être la distribution d'équilibre hydrostatique si le potentiel V dont découle la force à longue portée s'écrit (aux translations près) :

$$\frac{V}{m} = \frac{V}{m}(\vec{r}; t) = \frac{\omega(t)^2 \vec{r}^2}{2} + \frac{b(\vec{r}/\|\vec{r}\|)}{\vec{r}^2} \quad (6.2)$$

où m est la masse des particules, et b une fonction positive. Si le gaz possède un moment cinétique total \vec{J} non nul autour de l'origine, il faut aussi que le terme b soit invariant par rotation d'axe \vec{J} , ce qui s'écrit : $b = b(\vec{r}/\|\vec{r}\|) = b\left(\frac{\vec{r}}{\|\vec{r}\|} \cdot \frac{\vec{J}}{\|\vec{J}\|}\right)$.

Par ailleurs, la température doit être uniforme, et son évolution régie par l'équation différentielle :

$$\partial_{ttt}\beta + 4\omega^2\partial_t\beta + 4\omega(\partial_t\omega)\beta = 0 \quad (6.3)$$

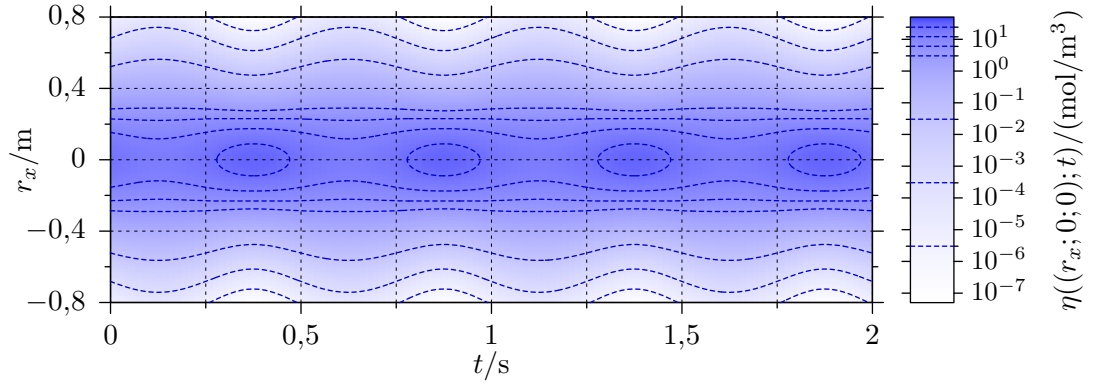
où β est le coefficient de l'équation (6.1).

Pour plus de détails, nous renvoyons à la publication [53].

6.1.2 Nos choix

Le cas physique précis décrit dans cette section est représenté sur la figure 6.1.

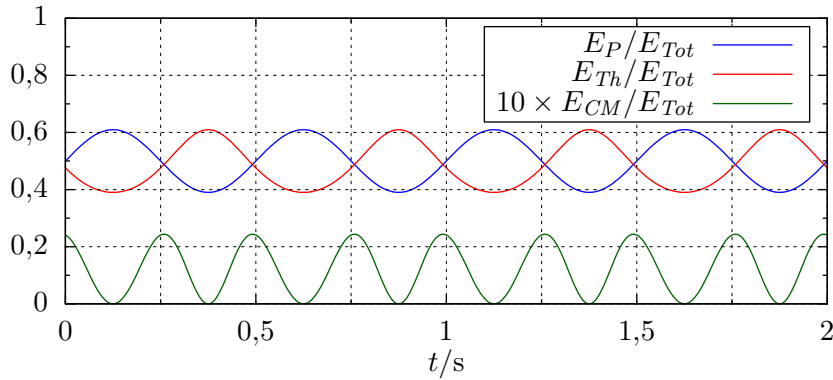
●●●● Pour simplifier notre cas d'étude, nous prenons un champ de force sans répulsion centrale ($b = 0$) et de raideur constante ($\omega = \text{cste}$), et nous choisissons un gaz mono-espèce, dont le moment cinétique total est nul. Alors la fonction de



(a) Densité de matière le long de l'axe (Ox) de l'espace ordinaire, en fonction du temps.

Les lignes bleues tiretées sont des isolignes de densité, écartées soit d'un facteur 2 soit d'un facteur 100.

En aucun point de l'espace la densité n'est constante. Même si cela ne se voit pas bien, les isolignes $\eta = 6 \text{ mol/m}^3$ ou $\eta = 3 \text{ mol/m}^3$ ondulent légèrement.



(b) Profils temporels de différentes énergies du gaz, exprimées en fractions de leur somme E_{Tot} . E_P est l'énergie potentielle cumulée des molécules, due au champ de force centrale; E_{Th} est l'énergie thermique totale du gaz, due à l'agitation moléculaire; et E_{CM} est l'énergie cinétique du gaz, due au mouvement observable à l'échelle macroscopique. E_{CM} est agrandie d'un facteur 10 pour des raisons de lisibilité.

FIGURE 6.1 – Présentation partielle de la situation physique qui sert de cas test au cours de ce chapitre.

Le problème est à symétrie sphérique. Un gaz monoatomique, dont chaque molécule est rappelée élastiquement vers l'origine des positions, forme un nuage gonflant et se dégonflant de façon strictement périodique.

Les paramètres de la situation dessinée ici sont bien aux valeurs utilisées dans les simulations.

distribution des particules s'écrit (voir [53]) :

$$f(\vec{r}; \vec{c}; t) = \frac{n \omega^3}{8\pi^3 c_{\text{qm min}}^3 c_{\text{qm max}}^3} \exp \left(\begin{aligned} & - \left(\frac{c_{\text{qm min}}^{-2} + c_{\text{qm max}}^{-2}}{4} + \frac{c_{\text{qm min}}^{-2} - c_{\text{qm max}}^{-2}}{4} \sin(2(\omega t + \phi_0)) \right) \vec{c}^2 \\ & + \frac{c_{\text{qm min}}^{-2} - c_{\text{qm max}}^{-2}}{2} \cos(2(\omega t + \phi_0)) \omega \vec{r} \cdot \vec{c} \\ & - \left(\frac{c_{\text{qm min}}^{-2} + c_{\text{qm max}}^{-2}}{4} - \frac{c_{\text{qm min}}^{-2} - c_{\text{qm max}}^{-2}}{4} \sin(2(\omega t + \phi_0)) \right) \omega^2 \vec{r}^2 \end{aligned} \right) \quad (6.4)$$

où n est la quantité totale de gaz, $c_{\text{qm min}}$ est le minimum au cours du temps de la vitesse quadratique moyenne sur chaque axe d'agitation thermique (uniforme), $c_{\text{qm max}}$ est le maximum de la même vitesse, et ϕ_0 est un décalage de phase à l'origine des temps.

À toutes fins utiles, on peut donner aussi l'expression des grandeurs macroscopiques associées à la distribution du gaz :

$$\eta(\vec{r}; t) = \frac{n \omega^3}{\pi^{3/2}} \left(c_{\text{qm max}}^2 + c_{\text{qm min}}^2 + (c_{\text{qm max}}^2 - c_{\text{qm min}}^2) \sin(2(\omega t + \phi_0)) \right)^{-3/2} \times \exp \left(\frac{-\omega^2 \vec{r}^2}{c_{\text{qm max}}^2 + c_{\text{qm min}}^2 + (c_{\text{qm max}}^2 - c_{\text{qm min}}^2) \sin(2(\omega t + \phi_0))} \right) \quad (6.5)$$

$$c_{\text{qma}}(t) = \sqrt{2} \left(c_{\text{qm min}}^{-2} + c_{\text{qm max}}^{-2} + (c_{\text{qm min}}^{-2} - c_{\text{qm max}}^{-2}) \sin(2(\omega t + \phi_0)) \right)^{-1/2} \quad (6.6)$$

$$\vec{v}(\vec{r}; t) = \frac{(c_{\text{qm max}}^2 - c_{\text{qm min}}^2) \cos(2(\omega t + \phi_0)) \omega \vec{r}}{(c_{\text{qm max}}^2 + c_{\text{qm min}}^2) + (c_{\text{qm max}}^2 - c_{\text{qm min}}^2) \sin(2(\omega t + \phi_0))} \quad (6.7)$$

Comme on calculera cette grandeur, on donne ici l'expression de la fraction des particules se situant à une distance de l'origine supérieure à un seuil r_0 donné :

$$\text{Frac}(\|\vec{r}\| \geq r_0; t) = \exp \left(\frac{-r_0^2}{2 r_{\text{qma}}^2} \right) \times \left(\sqrt{\frac{2}{\pi}} \times \frac{r_0}{r_{\text{qma}}} + \text{erfcx} \left(\frac{r_0}{\sqrt{2} r_{\text{qma}}} \right) \right) \quad (6.8a)$$

où

$$r_{\text{qma}} = \frac{1}{\omega \sqrt{2}} \sqrt{c_{\text{qm max}}^2 + c_{\text{qm min}}^2 + (c_{\text{qm max}}^2 - c_{\text{qm min}}^2) \sin(2(\omega t + \phi_0))} \quad (6.8b)^1$$

et erfcx est la fonction d'erreur complémentaire normalisée (voir l'annexe A.3).

1. Cette expression de r_{qma} est peu semblable à l'expression (6.6) de c_{qma} . C'est normal, car alors que r_{qma} est la distance quadratique moyenne sur chaque axe des particules *à l'origine des positions*, c_{qma} est la vitesse quadratique moyenne sur chaque axe *relative à la vitesse vectorielle moyenne locale* $\vec{v}(\vec{r}; t)$; d'où la différence.

●●●● Nous choisissons pour la suite de ce chapitre, afin de mettre en place les simulations, le jeu de paramètres suivant : $\omega = 2\pi$ Hz, $n = 1$ mol, $c_{\text{qm max}} = 1$ m/s, $\frac{c_{\text{qm min}}}{c_{\text{qm max}}} = 0,8$, et $\phi_0 = 0$. Ces valeurs sont purement arbitraires, et ne préjugent en rien de la physique sous-jacente.

Il nous faut aussi choisir un modèle de section efficace. Bien qu'il ne doive pas influencer la distribution des particules, il faut quand même en poser un, à moins de traiter uniquement du transport balistique (ce qui serait assez peu intéressant ici...). Pour nous simplifier les choses, nous prenons un modèle de section efficace de Maxwell isotrope, c'est-à-dire que $\sigma_F(g; u) = \kappa/(4\pi g)$, où κ est une constante. Nous fixons : $\frac{n\kappa\omega^2}{c_{\text{qm max}}^3} = 1,5$.

●●●● La figure 6.1 donne une représentation visuelle de la situation étudiée. Comme on peut s'y attendre en lisant les expressions de f et des grandeurs macroscopiques, on a un nuage de gaz centré sur l'origine, qui gonfle et se dégonfle de façon périodique.

6.2 Solution intégrale partielle de l'équation de Boltzmann, et application à la situation de confinement harmonique

6.2.1 Écritures intégrale et statistique de l'équation de Boltzmann, basée sur les Algorithmes à Collisions Nulles

●●●● On peut reprendre dans un premier temps directement l'équation de Boltzmann, dans sa forme générale 1.1.3 adaptée aux problèmes de cinétique des gaz monoatomiques et mono-espèce :

$$\begin{aligned} \partial_t f(\vec{r}; \vec{c}; t) + \vec{c} \cdot \overrightarrow{\text{grad}}_{\vec{r}} f(\vec{r}; \vec{c}; t) + \text{div}_{\vec{c}} \left(f(\vec{r}; \vec{c}; t) \vec{a}(\vec{r}; \vec{c}; t) \right) \\ = \int_{E_c} d\vec{c}_* \int_{4\pi} d\vec{u}' b_F \left(\|\vec{c} - \vec{c}_*\|; \vec{u}' \cdot \frac{\vec{c} - \vec{c}_*}{\|\vec{c} - \vec{c}_*\|} \right) \times \\ \left(-f(\vec{r}; \vec{c}; t) f(\vec{r}; \vec{c}_*; t) + f(\vec{r}; \vec{c}'; t) f(\vec{r}; \vec{c}'_*; t) \right) \end{aligned}$$

où $\vec{a} = \frac{\vec{F}}{m}$ est l'accélération subie par les particules due aux forces à longue portée, $b_F(g; u) = g \times \sigma_F(g; u)$ est une participation à la fréquence de collision, et :

$$\vec{c}' = \frac{\vec{c} + \vec{c}_*}{2} + \frac{\|\vec{c} - \vec{c}_*\|}{2} \vec{u}' \quad \vec{c}'_* = \frac{\vec{c} + \vec{c}_*}{2} - \frac{\|\vec{c} - \vec{c}_*\|}{2} \vec{u}'$$

Plutôt que d'en déduire l'expression intégrale (1.46a), on utilise le fonctionnement des Algorithmes à Collisions Nulles. Sans frontière et avec la condition

initiale « f est connue en un temps t_{init} », et en considérant que la force \vec{F} ne diverge pas dans l'espace des vitesses, on aboutit à l'expression suivante :

$$f(\vec{r}; \vec{c}; t) = \int_{-\infty}^t dt' \hat{\nu}(\vec{r}_b(t'); \vec{c}_b(t'); t') \exp\left(-\int_{t'}^t dt'' \hat{\nu}(\vec{r}_b(t''); \vec{c}_b(t''); t'')\right) \times \\ \left(H(t' \leq t_{init}) f(\vec{r}_b(t_{init}); \vec{c}_b(t_{init}); t_{init}) + H(t' > t_{init}) \times \right. \\ \left. \left(\left(1 - \frac{\nu_t(\vec{r}_b(t'); \vec{c}_b(t'); t')}{\hat{\nu}(\vec{r}_b(t'); \vec{c}_b(t'); t')} \right) f(\vec{r}_b(t'); \vec{c}_b(t'); t') + \frac{s_{is}(\vec{r}_b(t'); \vec{c}_b(t'); t')}{\hat{\nu}(\vec{r}_b(t'); \vec{c}_b(t'); t')} \right) \right) \quad (6.9a)$$

où $\hat{\nu}$ est une fréquence majorée de collision, \vec{r}_b et \vec{c}_b sont les solutions du problème différentiel :

$$\begin{cases} \partial_{t'} \vec{r}_b(t') = \vec{c}_b(t') \\ \partial_{t'} \vec{c}_b(t') = \vec{a}(\vec{r}_b(t'); \vec{c}_b(t'); t') \end{cases} \quad \text{avec les conditions finales} \quad \begin{cases} \vec{r}_b(t) = \vec{r} \\ \vec{c}_b(t) = \vec{c} \end{cases} \quad (6.9b)$$

et finalement $\nu_t(\vec{r}; \vec{c}; t)$ et $s_{is}(\vec{r}; \vec{c}; t)$ sont respectivement la fréquence réelle de collision et le terme source dû aux entrées par collision, en le point de l'espace-temps des phases $(\vec{r}; \vec{c}; t)$:

$$\nu_t(\vec{r}; \vec{c}; t) = \int_{E_c} d\vec{c}_* \int_{4\pi} d\vec{u}' b_F \left(\|\vec{c} - \vec{c}_*\|; \vec{u}' \cdot \frac{\vec{c} - \vec{c}_*}{\|\vec{c} - \vec{c}_*\|} \right) f(\vec{r}; \vec{c}_*; t) \quad (6.9c)$$

$$s_{is}(\vec{r}; \vec{c}; t) = \int_{E_c} d\vec{c}_* \int_{4\pi} d\vec{u}' b_F \left(\|\vec{c} - \vec{c}_*\|; \vec{u}' \cdot \frac{\vec{c} - \vec{c}_*}{\|\vec{c} - \vec{c}_*\|} \right) f(\vec{r}; \vec{c}'; t) f(\vec{r}; \vec{c}_*; t) \quad (6.9d)$$

●●●● En introduisant quelques variables aléatoires, on peut transformer l'expression intégrale précédente (6.9) en expression statistique utilisable dans un calcul de type Monte-Carlo :

$$f(\vec{r}; \vec{c}; t) = E \left(H(T' \leq t_{init}) f(\vec{r}_b(t_{init}); \vec{c}_b(t_{init}); t_{init}) + H(T' > t_{init}) \times \right. \\ \left. \left(\left(1 - \frac{\widetilde{N}_{t1}(\vec{r}_b(T'); \vec{c}_b(T'); T')}{\hat{\nu}(\vec{r}_b(T'); \vec{c}_b(T'); T')} \right) \widetilde{F}_2(\vec{r}_b(T'); \vec{c}_b(T'); T') + \frac{\widetilde{S}_{is}(\vec{r}_b(T'); \vec{c}_b(T'); T')}{\hat{\nu}(\vec{r}_b(T'); \vec{c}_b(T'); T')} \right) \right) \quad (6.10a)$$

où

$$\widetilde{N}_{t1}(\vec{r}; \vec{c}; t) = b_F \left(\|\vec{c} - \vec{C}_*\|; \vec{U}' \cdot \frac{\vec{c} - \vec{C}_*}{\|\vec{c} - \vec{C}_*\|} \right) \frac{\widetilde{F}_1(\vec{r}; \vec{C}_*; t)}{p_{\vec{C}_*}(\vec{C}_*) p_{\vec{U}'}(\vec{U}')} \quad (6.10b)$$

$$\widetilde{S}_{is}(\vec{r}; \vec{c}; t) = b_F \left(\|\vec{c} - \vec{C}_*\|; \vec{U}' \cdot \frac{\vec{c} - \vec{C}_*}{\|\vec{c} - \vec{C}_*\|} \right) \frac{\widetilde{F}_1(\vec{r}; \vec{C}'; t) \widetilde{F}_2(\vec{r}; \vec{C}_*; t)}{p_{\vec{C}_*}(\vec{C}_*) p_{\vec{U}'}(\vec{U}')} \quad (6.10c)$$

sont des estimateurs de la fréquence de collision ν_t et du terme source s_{is} . Dans l'expression (6.10c) :

$$\vec{C}' = \frac{\vec{c} + \vec{C}_*}{2} + \frac{\|\vec{c} - \vec{C}_*\|}{2} \vec{U}', \quad \vec{C}'_* = \frac{\vec{c} + \vec{C}_*}{2} - \frac{\|\vec{c} - \vec{C}_*\|}{2} \vec{U}' \quad (6.10d)$$

et $\tilde{F}(\vec{r}; \vec{c}; t)$ signifie un estimateur sans biais de $f(\vec{r}; \vec{c}; t)$. Dans les écritures (6.10a) et (6.10c), la notation « \tilde{F}_1, \tilde{F}_2 » signifie comme d'habitude l'indépendance de 2 estimateurs.

Les variables aléatoires à échantillonner sont les mêmes qu'au chapitre 5.2.1 :

- T' , à valeurs dans $] -\infty; t]$, est l'instant de la dernière collision d'une particule suivie.

T' est supposée suivre une loi de probabilité en analogie avec la physique ; sinon, dans l'expression (6.10a), une division par $p_{T'}$ serait présente. La loi que doit suivre T' est une loi exponentielle à gauche de t , dont le coefficient d'extinction est $\hat{\nu}$ (au fil de la trajectoire balistique de la particule suivie). Cela s'écrit :

$$F_{T'} : t' \mapsto H(t' < t) \exp\left(-\int_{t'}^t dt'' \hat{\nu}(\vec{r}_b(t''); \vec{c}_b(t''); t'')\right) + H(t' \geq t) \quad (6.10e)$$

- \vec{C}'_* à valeurs dans E_c , qui correspond à une vitesse sortante de la partenaire de la particule suivie dans sa dernière collision,
- \vec{U}' à valeurs dans $E_u = \mathcal{S}(E_c; \vec{0}; 1)$, qui correspond à la vitesse d'arrivée de la particule suivie dans le référentiel barycentrique de sa dernière collision.

Remarques.

- Il n'est pas nécessaire que les variables aléatoires \vec{C}'_* et \vec{U}' , dans l'expression de \tilde{N}_{t-1} d'une part, et dans l'expression de \tilde{S}_{is} d'autre part, soient bien les mêmes.
- Nous avons choisi la variable aléatoire T' comme une date de précédente collision, suivant la fréquence d'extinction $\hat{\nu}$. Au prix d'une légère modification de l'expression (6.10a) ce choix n'est pas obligatoire, même s'il est en général préférable comme cela est expliqué aux chapitres 1.5.3 et 2.2.2.

6.2.2 Transformations possibles de l'expression statistique

Considérant l'expression (6.10a), on se retrouve avec des possibilités similaires à ce qui était expliqué dans le chapitre 2.2.3. À savoir qu'une fois T' tiré et $\vec{r}_b(T')$ et $\vec{c}_b(T')$ calculés (dans le cas où $T' > t_{init}$), il faut calculer $(\hat{\nu} - \tilde{N}_{t-1})\tilde{F}_2 + \tilde{S}_{is}$ — la somme du terme de collision nulle et du terme source des collisions entrantes. C'est une somme dont les termes doivent être calculés récursivement, et on profiterait volontiers des possibilités offertes par les méthodes de Monte-Carlo pour ne pas la calculer entièrement à chaque demande.

Au contraire de ce qui est exposé au chapitre 2.2.3, les termes de la somme $(\hat{\nu} - \widetilde{N}_{t-1})\widetilde{F}_2 + \widetilde{S}_{is}$ sont tous 2 récursifs, et nécessitent chacun *a priori* 2 estimations de f pour être obtenus. Il n'y a pas d'intérêt à faire un choix de Bernoulli entre des combinaisons linéaires de ces termes (la 4^e proposition au chapitre 2.2.3), car l'un ne peut pas être partiellement ajouté à l'autre sans rendre le calcul de ce dernier nettement plus difficile². Par ailleurs, des probabilités analogues que la particule suivie ait subi ou non une collision ne paraissent pas aisées à dégager, car la fréquence vraie d'extinction ν_t est inconnue de prime abord. Restent donc fondamentalement 2 possibilités : calculer la somme $(\hat{\nu} - \widetilde{N}_{t-1})\widetilde{F}_2 + \widetilde{S}_{is}$ entièrement, ou utiliser un choix de Bernoulli plus ou moins arbitraire entre les 2 termes de cette somme.

En introduisant un choix de Bernoulli, l'équation (6.10a) se réécrit :

$$f(\vec{r}; \vec{c}; t) = \mathbb{E} \left(\mathbb{H}(T' \leq t_{init}) f(\vec{r}_b(t_{init}); \vec{c}_b(t_{init}); t_{init}) + \mathbb{H}(T' > t_{init}) \times \right. \\ \left. \left(P_{nul} \times \frac{1}{P_{nul}} \left(1 - \frac{\widetilde{N}_{t-1}(\vec{r}_b(T'); \vec{c}_b(T'); T')}{\hat{\nu}(\vec{r}_b(T'); \vec{c}_b(T'); T')} \right) \widetilde{F}_2(\vec{r}_b(T'); \vec{c}_b(T'); T') + \right. \right. \\ \left. \left. P_{diff} \times \frac{\widetilde{S}_{is}(\vec{r}_b(T'); \vec{c}_b(T'); T')}{P_{diff} \hat{\nu}(\vec{r}_b(T'); \vec{c}_b(T'); T')} \right) \right) \quad (6.11)$$

où P_{nul} et P_{diff} sont 2 probabilités telles que $P_{nul} + P_{diff} = 1$.

Il reste à faire un choix pour ces deux probabilités. Nous faisons ici 3 propositions que nous testerons :

pile ou face : $P_{nul} = P_{diff} = 1/2$

utilisation d'une approximation de f : On dispose d'une expression approximative de f . Grâce à elle, on dispose d'approximations ν_t^{\approx} et s_{is}^{\approx} respectivement de la fréquence de collision et du terme source. On en déduit des valeurs pour P_{nul} et P_{diff} grâce à une approche variance nulle, ou selon un calcul plus simple si l'approche variance nulle est trop difficile à mener à bien.

utilisation d'une estimation de f : Arrivé au point $(\vec{r}_b(T'); T')$, on effectue en premier lieu une estimation de $f(\vec{r}_b(T'); \vec{C}_*; T')$, où \vec{C}_* est une vitesse aléatoire pour une partenaire de collision. Cette estimation nous renseigne sur la fréquence de collision et le terme source, et on en déduit des valeurs pour P_{nul} et P_{diff} . L'estimation de $f(\vec{r}_b(T'); \vec{C}_*; T')$ peut éventuellement être réutilisée dans l'échantillonnage de \widetilde{N}_t ou de \widetilde{S}_{is} .

Dans le cas étudié d'un nuage de gaz pulsant dans un piège harmonique, le fait que l'on soit dans le noyau de l'opérateur de collision facilite les choses. En

2. À moins que le calcul de la somme de ces 2 termes ne soit *de base* plus simple que le calcul séparé de ces termes. Des propositions à ce sujet sont données page 258.

effet, il en résulte que $s_{is}(\vec{r}; \vec{c}; t) = \nu_t(\vec{r}; \vec{c}; t) f(\vec{r}; \vec{c}; t)$. Cela rend l'approximation ou l'estimation simultanée de ν_t et s_{is} plus facile — on pourrait s'attendre à ce que dans d'autres cas le calcul de s_{is} soit particulièrement difficile. Ainsi, dans la suite de ce chapitre nous justifierons nos choix essentiellement par l'approximation ou l'estimation de ν_t ; mais il n'est pas à exclure que dans d'autres situations, les choix à effectuer soient plus complexes.

6.2.3 Choix algorithmiques

●●●● Notre premier choix algorithmique sera de prendre la fréquence majorée de collision $\hat{\nu}$ constante. Ce choix est rendu raisonnable en particulier par le modèle des particules de Maxwell que nous avons choisi — pour nous faciliter le montage de la simulation justement —, qui fait que la fréquence de collision d'une particule est directement proportionnelle à la densité de particules en sa position (de l'espace ordinaire). Comme cette densité est majorée, la fréquence de collision l'est aussi; nous choisirons $\hat{\nu}$ comme égale au maximum de la fréquence de collision, calculable à la main, et valant $\hat{\nu} = \kappa \max(\eta) = \kappa \eta(\vec{r} = \vec{0}; \sin(2(\omega t + \phi_0)) = -1) = \frac{n\kappa\omega^3}{(2\pi)^{3/2} c_{qm \min}^3} \simeq 1,168\,776 \text{ Hz}$.

●●●● Si on calcule complètement la somme $(\hat{\nu} - \widetilde{N}_{t-1})\widetilde{F}_2 + \widetilde{S}_{is}$, ou qu'on choisit entre les 2 termes la composant par un test de pile ou face équilibré, la détermination des probabilités P_{nul} et P_{diff} n'est pas une question. Ces deux techniques seront d'ailleurs testées.

Si on utilise une approximation de f , il faudra *a priori* en tirer une approximation de ν_t et de s_{is} . Dans notre cas d'étude, le modèle de collisionneurs de Maxwell et l'annulation de l'opérateur de collision simplifient grandement les choses : $\nu_t(\vec{r}; \vec{c}; t) = \kappa \eta(\vec{r}; \vec{c}; t)$, et par ailleurs $s_{is}(\vec{r}; \vec{c}; t) = \nu_t(\vec{r}; \vec{c}; t) f(\vec{r}; \vec{c}; t)$.

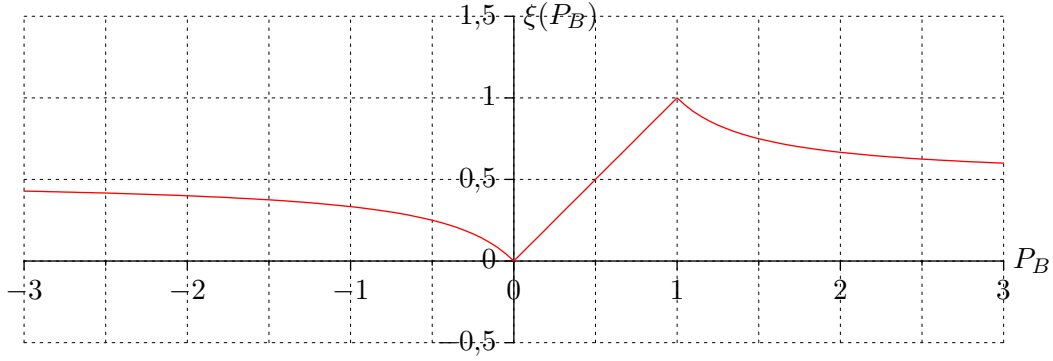
Sans nous intéresser au terme source s_{is} , nous choisissons $P_{nul} = \xi(P_{nul \ B})$, où :

- $P_{nul \ B} = 1 - \frac{\nu_t(\vec{r}; \vec{c}; t)}{\hat{\nu}} = 1 - \frac{\kappa \eta(\vec{r}; \vec{c}; t)}{\hat{\nu}}$, où η est une approximation de η que nous détaillerons bientôt.
- ξ est une fonction à valeurs dans $[0; 1]$, de sorte à assurer qu'on puisse utiliser P_{nul} comme une probabilité. On prendra :

$$\xi : \begin{cases} \mathbb{R} & \rightarrow & [0; 1] \\ P_B & \mapsto & P_B & \text{si } P_B \in [0; 1] \\ P_B & \mapsto & \frac{1}{2} \left(\frac{1}{2P_B - 1} + 1 \right) & \text{si } P_B \notin]0; 1[\end{cases} \quad (6.12)$$

, tracée en figure 6.2. Cette fonction respecte en particulier la propriété : $\forall P_B \in \mathbb{R}, \xi(1 - P_B) = 1 - \xi(P_B)$.

Ensuite P_{diff} est calculée comme $1 - P_{nul}$.

FIGURE 6.2 – Graphe de la fonction ξ

Si on effectue une première estimation \tilde{F} de f , alors on peut donner une estimation de la fréquence de collision :

$$\tilde{N}_t(\vec{r}; \vec{c}; t) = \frac{\kappa}{4\pi} \times \frac{\tilde{F}(\vec{r}; \vec{C}_*; t)}{p_{\vec{C}_*}(\vec{C}_*) p_{\vec{U}'}(\vec{U}')} \quad (6.13)$$

On peut alors choisir $P_{nul\ B} = 1 - \frac{\tilde{N}_t(\vec{r}; \vec{c}; t)}{\hat{\nu}}$, puis $P_{nul} = \xi(P_{nul\ B})$ comme en (6.12).

●●●● Restent les choix de \vec{C}_* et de \vec{U}' , pour l'estimation de ν_t et s_{is} .

S'il est possible que l'on doive calculer à la fois $\tilde{N}_{t\ 1}$ et \tilde{S}_{is} , il faut d'abord savoir si on choisit les mêmes \vec{C}_* et \vec{U}' pour le calcul de $\tilde{N}_{t\ 1}$ et de \tilde{S}_{is} . On est dans cette situation si on calcule extensivement la somme $(\hat{\nu} - \tilde{N}_{t\ 1})\tilde{F}_2 + \tilde{S}_{is}$, ou si on effectue un choix de Bernoulli entre ses 2 termes basé sur le calcul à l'avance de $\tilde{N}_{t\ 1}$.

En considérant que l'on se basera sur les formulations statistiques (6.10) ou (6.11), on peut lister dès maintenant les 6 techniques d'estimation de $(\hat{\nu} - \nu_t)f + s_{is}$ dont nous envisageons la construction — sans compter les divers choix possibles pour $p_{\vec{C}_*}$ et $p_{\vec{U}'}$:

1. calcul extensif de $(\hat{\nu} - \tilde{N}_{t\ 1})\tilde{F}_2 + \tilde{S}_{is}$, en utilisant les mêmes \vec{C}_* et \vec{U}' dans le calcul de $\tilde{N}_{t\ 1}$ et de \tilde{S}_{is}
2. calcul extensif de $(\hat{\nu} - \tilde{N}_{t\ 1})\tilde{F}_2 + \tilde{S}_{is}$, en utilisant 2 couples $(\vec{C}_*; \vec{U}')$ différents pour le calcul de $\tilde{N}_{t\ 1}$ et de \tilde{S}_{is}
3. choix de pile ou face entre les termes $(\hat{\nu} - \tilde{N}_{t\ 1})\tilde{F}_2$ et \tilde{S}_{is}
4. choix de Bernoulli entre les termes $(\hat{\nu} - \tilde{N}_{t\ 1})\tilde{F}_2$ et \tilde{S}_{is} . Les probabilités de choisir l'un ou l'autre terme sont déterminées suivant une approximation de ν_t , elle-même basée sur une approximation de f .
5. choix de Bernoulli entre les termes $(\hat{\nu} - \tilde{N}_{t\ 1})\tilde{F}_2$ et \tilde{S}_{is} . Les probabilités de choisir l'un ou l'autre terme sont déterminées en calculant d'office $\tilde{N}_{t\ 1}$. Si on choisit finalement le terme \tilde{S}_{is} , il est alors calculé en réutilisant \vec{C}_* et \vec{U}' tirées pour le calcul de $\tilde{N}_{t\ 1}$.

6. choix de Bernoulli entre les termes $(\hat{\nu} - \widetilde{N}_{t1})\widetilde{F}_2$ et \widetilde{S}_{is} . Les probabilités de choisir l'un ou l'autre terme sont déterminées en calculant d'office \widetilde{N}_{t1} . Si on choisit finalement le terme \widetilde{S}_{is} , il est calculé totalement indépendamment à \widetilde{N}_{t1} .

Les algorithmes qui découlent de ces différents choix sont détaillés respectivement en 6.1, 6.2, 6.3, 6.4, 6.5, et 6.6.

	Entrées : Le point $(\vec{r}; \vec{c}; t)$ où on veut estimer $f(\vec{r}; \vec{c}; t)$
	Sorties : Une estimation ponctuelle de $f(\vec{r}; \vec{c}; t)$
1	échantillonner T_r selon une loi exponentielle unité : on obtient t_r ;
2	$t' \leftarrow t - t_r/\hat{\nu}$;
3	si $t' \leq t_{init}$ alors retourner $f(\vec{r}_b(t_{init}); \vec{c}_b(t_{init}); t_{init})$;
4	sinon
5	échantillonner \vec{C}_* : on obtient \vec{c}_* ;
6	échantillonner \vec{U}' : on obtient \vec{u}' ;
7	$\vec{c}' \leftarrow \frac{\vec{c} + \vec{c}_*}{2} + \frac{\ \vec{c} - \vec{c}_*\ }{2} \vec{u}'$;
8	$\vec{c}'_* \leftarrow \frac{\vec{c} + \vec{c}_*}{2} - \frac{\ \vec{c} - \vec{c}_*\ }{2} \vec{u}'$;
9	estimer $f(\vec{r}_b(t'); \vec{c}_*; t')$ grâce à cet algorithme : on obtient $\widetilde{F}_1(\vec{r}_b(t'); \vec{c}_*; t')$;
10	estimer $f(\vec{r}_b(t'); \vec{c}_b(t'); t')$ grâce à cet algorithme : on obtient $\widetilde{F}_2(\vec{r}_b(t'); \vec{c}_b(t'); t')$;
11	estimer $f(\vec{r}_b(t'); \vec{c}'_*; t')$ grâce à cet algorithme : on obtient $\widetilde{F}_1(\vec{r}_b(t'); \vec{c}'_*; t')$;
12	estimer $f(\vec{r}_b(t'); \vec{c}'_*; t')$ grâce à cet algorithme : on obtient $\widetilde{F}_2(\vec{r}_b(t'); \vec{c}'_*; t')$;
13	retourner $\left(\left(1 - \frac{\kappa \widetilde{F}_1(\vec{r}_b(t'); \vec{c}_*; t')}{4\pi\hat{\nu} p_{\vec{C}_*}(\vec{c}_*) p_{\vec{U}'}(\vec{u}')} \right) \widetilde{F}_2(\vec{r}_b(t'); \vec{c}_b(t'); t') + \frac{\kappa \widetilde{F}_1(\vec{r}_b(t'); \vec{c}'_*; t') \widetilde{F}_2(\vec{r}_b(t'); \vec{c}'_*; t')}{4\pi\hat{\nu} p_{\vec{C}_*}(\vec{c}_*) p_{\vec{U}'}(\vec{u}')} \right)$;

Algorithme 6.1 : Algorithme servant à estimer $f(\vec{r}; \vec{c}; t)$, valide dans les conditions décrites à la section 6.1. $f(\vec{r}; \vec{c}; t)$ est supposée connue à un temps t_{init} .

●●● L'algorithme 6.1 est à variance nulle³ dans notre cas d'étude. L'algorithme 6.5 est de même à variance nulle, tant que $P_{nul\ B}$ reste dans l'intervalle $[0; 1]$.

Ceci est une conséquence de l'annulation de l'opérateur de collision. En tout point $(\vec{r}; t)$ de l'espace-temps, la distribution des vitesses des molécules est maxwellienne. Une propriété de cette distribution est la *balance détaillée* : pour tout

3. Le concept de variance nulle est, dans ce manuscrit, introduit au chapitre 1.3.2.

Entrées : Le point $(\vec{r}; \vec{c}; t)$ où on veut estimer $f(\vec{r}; \vec{c}; t)$

Sorties : Une estimation ponctuelle de $f(\vec{r}; \vec{c}; t)$

```

1 échantillonner  $T_r$  selon une loi exponentielle unité : on obtient  $t_r$ ;
2  $t' \leftarrow t - t_r/\hat{\nu}$ ;
3 si  $t' \leq t_{init}$  alors retourner  $f(\vec{r}_b(t_{init}); \vec{c}_b(t_{init}); t_{init})$ ;
4 sinon
5   échantillonner  $\vec{C}_{*1}$  : on obtient  $\vec{c}_{*1}$ ;
6   échantillonner  $\vec{U}'_1$  : on obtient  $\vec{u}'_1$ ;
7   estimer  $f(\vec{r}_b(t'); \vec{c}_{*1}; t')$  grâce à cet algorithme : on obtient
    $\tilde{F}_1(\vec{r}_b(t'); \vec{c}_{*1}; t')$ ;
8   estimer  $f(\vec{r}_b(t'); \vec{c}_b(t'); t')$  grâce à cet algorithme : on obtient
    $\tilde{F}_2(\vec{r}_b(t'); \vec{c}_b(t'); t')$ ;
9   échantillonner  $\vec{C}_{*2}$  : on obtient  $\vec{c}_{*2}$ ;
10  échantillonner  $\vec{U}'_2$  : on obtient  $\vec{u}'_2$ ;
11   $\vec{c}' \leftarrow \frac{\vec{c} + \vec{c}_{*2}}{2} + \frac{\|\vec{c} - \vec{c}_{*2}\|}{2} \vec{u}'_2$ ;
12   $\vec{c}'_* \leftarrow \frac{\vec{c} + \vec{c}_{*2}}{2} - \frac{\|\vec{c} - \vec{c}_{*2}\|}{2} \vec{u}'_2$ ;
13  estimer  $f(\vec{r}_b(t'); \vec{c}'; t')$  grâce à cet algorithme : on obtient  $\tilde{F}_1(\vec{r}_b(t'); \vec{c}'; t')$ ;
14  estimer  $f(\vec{r}_b(t'); \vec{c}'_*; t')$  grâce à cet algorithme : on obtient  $\tilde{F}_2(\vec{r}_b(t'); \vec{c}'_*; t')$ ;
15  retourner  $\left( \left( 1 - \frac{\kappa \tilde{F}_1(\vec{r}_b(t'); \vec{c}_{*1}; t')}{4\pi\hat{\nu} p_{\vec{C}_{*1}}(\vec{c}_{*1}) p_{\vec{U}'_1}(\vec{u}'_1)} \right) \tilde{F}_2(\vec{r}_b(t'); \vec{c}_b(t'); t') + \right.$ 
    $\left. \frac{\kappa \tilde{F}_1(\vec{r}_b(t'); \vec{c}'; t') \tilde{F}_2(\vec{r}_b(t'); \vec{c}'_*; t')}{4\pi\hat{\nu} p_{\vec{C}_{*2}}(\vec{c}_{*2}) p_{\vec{U}'_2}(\vec{u}'_2)} \right)$ ;

```

Algorithme 6.2 : Algorithme servant à estimer $f(\vec{r}; \vec{c}; t)$, valide dans les conditions décrites à la section 6.1. $f(\vec{r}; \vec{c}; t)$ est supposée connue à un temps t_{init} .

$(\vec{r}; \vec{c}; t; \vec{c}_*; \vec{u}') \in (E_r \times E_c \times E_t \times E_c \times E_u)$, on a :

$$f(\vec{r}; \vec{c}; t) f(\vec{r}; \vec{c}_*; t) = f(\vec{r}; \vec{c}'; t) f(\vec{r}; \vec{c}'_*; t) \quad (6.14)$$

Injectée dans la traduction statistique de l'algorithme 6.1, la balance détaillée amène à $(\hat{\nu} - \tilde{N}_{t-1})\tilde{F}_2 + \tilde{S}_{is} = \hat{\nu}\tilde{F}_2$, de façon récursive dans les arbres d'estimations. C'est aussi le cas avec l'algorithme 6.5, tant que $P_{nul} = P_{nul\ B}$ — tant que $P_{nul\ B} \in [0; 1]$, étant donné notre choix de ξ .

●●● Comme au chapitre précédent, on tire \vec{C}_* selon une maxwellienne et \vec{U}' de façon isotrope. Il nous faut donc décider d'une vitesse moyenne et d'une température pour mettre en place complètement le tirage de \vec{C}_* .

	Entrées : Le point $(\vec{r}; \vec{c}; t)$ où on veut estimer $f(\vec{r}; \vec{c}; t)$
	Sorties : Une estimation ponctuelle de $f(\vec{r}; \vec{c}; t)$
1	échantillonner T_r selon une loi exponentielle unité : on obtient t_r ;
2	$t' \leftarrow t - t_r/\hat{\nu}$;
3	si $t' \leq t_{init}$ alors retourner $f(\vec{r}_b(t_{init}); \vec{c}_b(t_{init}); t_{init})$;
4	sinon
5	échantillonner \vec{C}_* : on obtient \vec{c}_* ;
6	échantillonner \vec{U}' : on obtient \vec{u}' ;
7	échantillonner R de loi uniforme standard : on obtient r ;
8	si $r < 1/2$ alors
9	estimer $f(\vec{r}_b(t'); \vec{c}_*; t')$ grâce à cet algorithme : on obtient $\tilde{F}_1(\vec{r}_b(t'); \vec{c}_*; t')$;
10	estimer $f(\vec{r}_b(t'); \vec{c}_b(t'); t')$ grâce à cet algorithme : on obtient $\tilde{F}_2(\vec{r}_b(t'); \vec{c}_b(t'); t')$;
11	retourner $\left(2 \left(1 - \frac{\kappa \tilde{F}_1(\vec{r}_b(t'); \vec{c}_*; t')}{4\pi\hat{\nu} p_{\vec{C}_*}(\vec{c}_*) p_{\vec{U}'}(\vec{u}')} \right) \tilde{F}_2(\vec{r}_b(t'); \vec{c}_b(t'); t') \right)$;
12	sinon
13	$\vec{c}' \leftarrow \frac{\vec{c} + \vec{c}_*}{2} + \frac{\ \vec{c} - \vec{c}_*\ }{2} \vec{u}'$;
14	$\vec{c}'_* \leftarrow \frac{\vec{c} + \vec{c}_*}{2} - \frac{\ \vec{c} - \vec{c}_*\ }{2} \vec{u}'$;
15	estimer $f(\vec{r}_b(t'); \vec{c}'; t')$ grâce à cet algorithme : on obtient $\tilde{F}_1(\vec{r}_b(t'); \vec{c}'; t')$;
16	estimer $f(\vec{r}_b(t'); \vec{c}'_*; t')$ grâce à cet algorithme : on obtient $\tilde{F}_2(\vec{r}_b(t'); \vec{c}'_*; t')$;
17	retourner $\left(2 \times \frac{\kappa \tilde{F}_1(\vec{r}_b(t'); \vec{c}'; t') \tilde{F}_2(\vec{r}_b(t'); \vec{c}'_*; t')}{4\pi\hat{\nu} p_{\vec{C}_*}(\vec{c}_*) p_{\vec{U}'}(\vec{u}')} \right)$;

Algorithme 6.3 : Algorithme servant à estimer $f(\vec{r}; \vec{c}; t)$, valide dans les conditions décrites à la section 6.1. $f(\vec{r}; \vec{c}; t)$ est supposée connue à un temps t_{init} .

Si on choisit la vitesse moyenne et la température exactes du cas d'étude, les algorithmes 6.1, 6.2, 6.5 et 6.6 sont à variance nulle. Là-encore, c'est en partie lié à ce que la distribution f dans notre cas d'étude est maxwellienne en tout point. Si on choisit de plus la densité exacte pour η^\approx , l'algorithme 6.4 aussi est à variance nulle.

●●● Afin d'avoir des résultats intéressants, on choisit de paramétrer la maxwellienne de tirage de \vec{C}_* avec une vitesse et une température différentes de leur

Entrées : Le point $(\vec{r}; \vec{c}; t)$ où on veut estimer $f(\vec{r}; \vec{c}; t)$
Sorties : Une estimation ponctuelle de $f(\vec{r}; \vec{c}; t)$

- 1 échantillonner T_r selon une loi exponentielle unité : on obtient t_r ;
- 2 $t' \leftarrow t - t_r/\hat{\nu}$;
- 3 **si** $t' \leq t_{init}$ **alors retourner** $f(\vec{r}_b(t_{init}); \vec{c}_b(t_{init}); t_{init})$;
- 4 **sinon**
- 5 $\nu_t^\approx \leftarrow \kappa \eta^\approx(\vec{r}_b(t'); \vec{c}_b(t'); t')$;
- 6 $P_{nul\ B} \leftarrow 1 - \nu_t^\approx/\hat{\nu}$;
- 7 $P_{nul} \leftarrow \xi(P_{nul\ B})$;
- 8 échantillonner \vec{C}_* : on obtient \vec{c}_* ;
- 9 échantillonner \vec{U}' : on obtient \vec{u}' ;
- 10 échantillonner R de loi uniforme standard : on obtient r ;
- 11 **si** $r < P_{nul}$ **alors**
- 12 | estimer $f(\vec{r}_b(t'); \vec{c}_*; t')$ grâce à cet algorithme : on obtient $\tilde{F}_1(\vec{r}_b(t'); \vec{c}_*; t')$;
- 13 | estimer $f(\vec{r}_b(t'); \vec{c}_b(t'); t')$ grâce à cet algorithme : on obtient $\tilde{F}_2(\vec{r}_b(t'); \vec{c}_b(t'); t')$;
- 14 | **retourner** $\left(\frac{1}{P_{nul}} \left(1 - \frac{\kappa \tilde{F}_1(\vec{r}_b(t'); \vec{c}_*; t')}{4\pi\hat{\nu} p_{\vec{C}_*}(\vec{c}_*) p_{\vec{U}'}(\vec{u}')} \right) \tilde{F}_2(\vec{r}_b(t'); \vec{c}_b(t'); t') \right)$;
- 15 **sinon**
- 16 | $\vec{c}' \leftarrow \frac{\vec{c} + \vec{c}_*}{2} + \frac{\|\vec{c} - \vec{c}_*\|}{2} \vec{u}'$;
- 17 | $\vec{c}'_* \leftarrow \frac{\vec{c} + \vec{c}_*}{2} - \frac{\|\vec{c} - \vec{c}_*\|}{2} \vec{u}'$;
- 18 | estimer $f(\vec{r}_b(t'); \vec{c}'; t')$ grâce à cet algorithme : on obtient $\tilde{F}_1(\vec{r}_b(t'); \vec{c}'; t')$;
- 19 | estimer $f(\vec{r}_b(t'); \vec{c}'_*; t')$ grâce à cet algorithme : on obtient $\tilde{F}_2(\vec{r}_b(t'); \vec{c}'_*; t')$;
- 20 | **retourner** $\left(\frac{1}{1 - P_{nul}} \times \frac{\kappa \tilde{F}_1(\vec{r}_b(t'); \vec{c}'; t') \tilde{F}_2(\vec{r}_b(t'); \vec{c}'_*; t')}{4\pi\hat{\nu} p_{\vec{C}_*}(\vec{c}_*) p_{\vec{U}'}(\vec{u}')} \right)$;

Algorithme 6.4 : Algorithme servant à estimer $f(\vec{r}; \vec{c}; t)$, valide dans les conditions décrites à la section 6.1. $f(\vec{r}; \vec{c}; t)$ est supposée connue à un temps t_{init} . η^\approx est une approximation connue à l'avance de la densité η .

expressions exactes (connues). On fait le choix suivant : on prélève les paramètres macroscopiques (vitesse moyenne, température et densité) depuis une situation physique où la masse et l'énergie totale sont les mêmes, mais où il n'y a pas d'oscillations.

Entrées : Le point $(\vec{r}; \vec{c}; t)$ où on veut estimer $f(\vec{r}; \vec{c}; t)$
Sorties : Une estimation ponctuelle de $f(\vec{r}; \vec{c}; t)$

- 1 échantillonner T_r selon une loi exponentielle unité : on obtient t_r ;
- 2 $t' \leftarrow t - t_r/\hat{\nu}$;
- 3 **si** $t' \leq t_{init}$ **alors retourner** $f(\vec{r}_b(t_{init}); \vec{c}_b(t_{init}); t_{init})$;
- 4 **sinon**
- 5 échantillonner \vec{C}_* : on obtient \vec{c}_* ;
- 6 échantillonner \vec{U}' : on obtient \vec{u}' ;
- 7 estimer $f(\vec{r}_b(t'); \vec{c}_*; t')$ grâce à cet algorithme : on obtient $\tilde{F}_1(\vec{r}_b(t'); \vec{c}_*; t')$;
- 8 $\tilde{N}_{t\ 1} \leftarrow \frac{\kappa}{4\pi} \times \frac{\tilde{F}_1(\vec{r}_b(t'); \vec{c}_*; t')}{p_{\vec{C}_*}(\vec{c}_*)p_{\vec{U}'}(\vec{u}')};$
- 9 $P_{nul\ B} \leftarrow 1 - \tilde{N}_{t\ 1}/\hat{\nu}$;
- 10 $P_{nul} \leftarrow \xi(P_{nul\ B})$;
- 11 échantillonner R de loi uniforme standard : on obtient r ;
- 12 **si** $r < P_{nul}$ **alors**
- 13 estimer $f(\vec{r}_b(t'); \vec{c}_b(t'); t')$ grâce à cet algorithme : on obtient $\tilde{F}_2(\vec{r}_b(t'); \vec{c}_b(t'); t')$;
- 14 **retourner** $\left(\frac{P_{nul\ B}}{P_{nul}} \tilde{F}_2(\vec{r}_b(t'); \vec{c}_b(t'); t') \right)$;
- 15 **sinon**
- 16 $\vec{c}' \leftarrow \frac{\vec{c} + \vec{c}_*}{2} + \frac{\|\vec{c} - \vec{c}_*\|}{2} \vec{u}'$;
- 17 $\vec{c}'_* \leftarrow \frac{\vec{c} + \vec{c}_*}{2} - \frac{\|\vec{c} - \vec{c}_*\|}{2} \vec{u}'$;
- 18 estimer $f(\vec{r}_b(t'); \vec{c}'; t')$ grâce à cet algorithme : on obtient $\tilde{F}_1(\vec{r}_b(t'); \vec{c}'; t')$;
- 19 estimer $f(\vec{r}_b(t'); \vec{c}'_*; t')$ grâce à cet algorithme : on obtient $\tilde{F}_2(\vec{r}_b(t'); \vec{c}'_*; t')$;
- 20 **retourner** $\left(\frac{1}{1 - P_{nul}} \times \frac{\kappa \tilde{F}_1(\vec{r}_b(t'); \vec{c}'; t') \tilde{F}_2(\vec{r}_b(t'); \vec{c}'_*; t')}{4\pi\hat{\nu} p_{\vec{C}_*}(\vec{c}_*)p_{\vec{U}'}(\vec{u}')} \right)$;

Algorithme 6.5 : Algorithme servant à estimer $f(\vec{r}; \vec{c}; t)$, valide dans les conditions décrites à la section 6.1. $f(\vec{r}; \vec{c}; t)$ est supposée connue à un temps t_{init} .

Dans les choix que nous avons faits, la quantité de matière totale vaut $n = 1$ mol, et l'énergie totale vaut :

$$E_{Tot} = 3m \frac{c_{qm\ max}^2 + c_{qm\ min}^2}{2} \quad (6.15)$$

Entrées : Le point $(\vec{r}; \vec{c}; t)$ où on veut estimer $f(\vec{r}; \vec{c}; t)$
Sorties : Une estimation ponctuelle de $f(\vec{r}; \vec{c}; t)$

- 1 échantillonner T_r selon une loi exponentielle unité : on obtient t_r ;
- 2 $t' \leftarrow t - t_r/\hat{\nu}$;
- 3 **si** $t' \leq t_{init}$ **alors retourner** $f(\vec{r}_b(t_{init}); \vec{c}_b(t_{init}); t_{init})$;
- 4 **sinon**
- 5 échantillonner \vec{C}_{*1} : on obtient \vec{c}_{*1} ;
- 6 échantillonner \vec{U}'_1 : on obtient \vec{u}'_1 ;
- 7 estimer $f(\vec{r}_b(t'); \vec{c}_*(t'); t')$ grâce à cet algorithme : on obtient $\tilde{F}_1(\vec{r}_b(t'); \vec{c}_{*1}; t')$;
- 8 $\tilde{N}_{t1} \leftarrow \frac{\kappa}{4\pi} \times \frac{\tilde{F}_1(\vec{r}_b(t'); \vec{c}_{*1}; t')}{p_{\vec{C}_{*1}}(\vec{c}_{*1})p_{\vec{U}'_1}(\vec{u}'_1)}$;
- 9 $P_{nulB} \leftarrow 1 - \tilde{N}_{t1}/\hat{\nu}$;
- 10 $P_{nul} \leftarrow \xi(P_{nulB})$;
- 11 échantillonner R de loi uniforme standard : on obtient r ;
- 12 **si** $r < P_{nul}$ **alors**
- 13 estimer $f(\vec{r}_b(t'); \vec{c}_b(t'); t')$ grâce à cet algorithme : on obtient $\tilde{F}_2(\vec{r}_b(t'); \vec{c}_b(t'); t')$;
- 14 **retourner** $\left(\frac{P_{nulB}}{P_{nul}} \tilde{F}_2(\vec{r}_b(t'); \vec{c}_b(t'); t') \right)$;
- 15 **sinon**
- 16 échantillonner \vec{C}_{*2} : on obtient \vec{c}_{*2} ;
- 17 échantillonner \vec{U}'_2 : on obtient \vec{u}'_2 ;
- 18 $\vec{c}' \leftarrow \frac{\vec{c} + \vec{c}_{*2}}{2} + \frac{\|\vec{c} - \vec{c}_{*2}\|}{2} \vec{u}'_2$;
- 19 $\vec{c}'_* \leftarrow \frac{\vec{c} + \vec{c}_{*2}}{2} - \frac{\|\vec{c} - \vec{c}_{*2}\|}{2} \vec{u}'_2$;
- 20 estimer $f(\vec{r}_b(t'); \vec{c}'(t'); t')$ grâce à cet algorithme : on obtient $\tilde{F}_1(\vec{r}_b(t'); \vec{c}'; t')$;
- 21 estimer $f(\vec{r}_b(t'); \vec{c}'_*(t'); t')$ grâce à cet algorithme : on obtient $\tilde{F}_2(\vec{r}_b(t'); \vec{c}'_*; t')$;
- 22 **retourner** $\left(\frac{1}{1 - P_{nul}} \times \frac{\kappa \tilde{F}_1(\vec{r}_b(t'); \vec{c}'; t') \tilde{F}_2(\vec{r}_b(t'); \vec{c}'_*; t')}{4\pi\hat{\nu} p_{\vec{C}_{*2}}(\vec{c}_{*2})p_{\vec{U}'_2}(\vec{u}'_2)} \right)$;

Algorithme 6.6 : Algorithme servant à estimer $f(\vec{r}; \vec{c}; t)$, valide dans les conditions décrites à la section 6.1. $f(\vec{r}; \vec{c}; t)$ est supposée connue à un temps t_{init} .

où m est la masse totale contenue dans le système. Cela représente 6 fois l'énergie cinétique moyenne sur chaque axe. Le facteur 6 remplace ici l'habituel facteur

3, car aux degrés de liberté stockant de l'énergie que sont les coordonnées de la vitesse s'ajoutent les coordonnées de la position, qui représente de l'énergie à cause du champ de force extérieur.

Si la quantité de matière, la masse, le moment cinétique, et l'énergie sont les mêmes que dans la situation d'étude, ainsi que le champ de force de rappel vers le centre, mais qu'il n'y a pas d'oscillations, les paramètres macroscopiques s'expriment :

$$\eta(\vec{r}) = \frac{n \omega^3}{\pi^{3/2}} (c_{\text{qm max}}^2 + c_{\text{qm min}}^2)^{-3/2} \exp\left(\frac{-\omega^2 \vec{r}^2}{c_{\text{qm max}}^2 + c_{\text{qm min}}^2}\right) \quad (6.16a)$$

$$\vec{v}(\vec{r}) = \vec{0} \quad (6.16b)$$

$$c_{qma} = \sqrt{\frac{1}{2}(c_{\text{qm max}}^2 + c_{\text{qm min}}^2)} \quad (6.16c)$$

●●●● Les algorithmes qui résultent finalement de toutes ces considérations, et qui ont été utilisés dans la campagne de tests qui suit, sont présentés dans l'annexe D.4.

6.3 Résultats obtenus

●●●● La campagne de tests a été menée sur les 6 algorithmes proposés : 6.1, 6.2, 6.3, 6.4, 6.5, et 6.6. Ceux-ci sont testés dans le calcul de 3 grandeurs :

- $f(\vec{r}; \vec{c}; t)$ en des points fixés de l'espace des phases. 3 points $(\vec{r}; \vec{c})$ de l'espace des phases sont choisis, auxquels nous traçons des profils de $f(\vec{r}; \vec{c}; t)$ selon le temps t . Nous posons arbitrairement $r_{b1} = 0,2 \text{ m}$, puis les 3 points sont : $(r_{b1} \vec{e}_x; r_{b1} \omega \vec{e}_x)$, $(r_{b1} \vec{e}_x; r_{b1} \omega \vec{e}_y)$, et $(r_{b1} \vec{e}_x; -r_{b1} \omega \vec{e}_x)$, où \vec{e}_x et \vec{e}_y sont 2 vecteurs unitaires suivant les axes du repère.

En le 2^e point f est constante. En effet, dans la situation précise de piège harmonique qui nous sert de cas test, $f(\vec{r}; \vec{c}; t)$ est stationnaire ssi $\vec{r} \perp \vec{c}$ et $\|\vec{c}\| = \omega \|\vec{r}\|$.

- $\eta(\vec{r}; t)$ en des points fixés de l'espace usuel. 3 points \vec{r} de l'espace usuel sont choisis, auxquels nous traçons des profils de $\eta(\vec{r}; t)$ selon le temps t : $\vec{0}$, $r_{b2} \vec{e}_x$, et $2r_{b2} \vec{e}_x$, avec cette fois-ci $r_{b2} = 0,25 \text{ m}$.

L'intégration finale de f sur E_c est faite en choisissant une vitesse selon un profil maxwellien, avec la même vitesse moyenne et la même température que le tirage de \vec{C}_* dans les algorithmes de calcul récursif de f .

- la fraction du total des particules dont la distance au centre dépasse des seuils fixés r_0 . 3 seuils sont choisis, pour lesquels nous traçons le profil de la fraction selon le temps : r_{b1} , $2r_{b1}$, et $3r_{b1}$ — avec le même $r_{b1} = 0,2 \text{ m}$ qu'à l'instant.

L'intégration finale de f sur E_c est faite de la même manière que lors des calculs de η . L'intégration sur $E_r \setminus \mathcal{B}(E_r; \vec{0}; r_0)$ utilise une loi de probabilité

similaire à celle exhibée en (5.16) : $\vec{R}_f = R_f \vec{U}_f$, où $\vec{U}_f \in \mathcal{S}(E_r; \vec{0}; 1)$ et est de loi isotrope, et :

$$\begin{cases} p_{R_f}(r_f) = \frac{r_0(r_0^2 - 2\text{ m}^2)}{\left((r_0^2 - 2\text{ m}^2)(r_f - r_0) + r_0\right)^2} & \text{si } r_0 > 2\text{ m} \\ p_{R_f}(r_f) = \frac{2\text{ m}}{(2\text{ m} + r_f - r_0)^2} & \text{si } r_0 \leq 2\text{ m} \end{cases} \quad (6.17)$$

Les résultats en sont présentés en figures 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, et 6.9.

●●●● La campagne de tests révèle finalement peu de comportements que l'on ne suspectait pas déjà.

- L'algorithme 6.1 est à variance nulle. Les algorithmes 6.2 et 6.5 exhibent eux aussi une variance très faible.

Dans le cas de l'algorithme 6.2, cela est probablement dû à un mécanisme similaire à celui explicité dans le chapitre 5.4. En effet, à chaque nœud d'un arbre d'estimations généré par cet algorithme, non seulement on multiplie ensemble des estimateurs (la variance relative augmente) mais ensuite on les somme (la variance relative diminue).

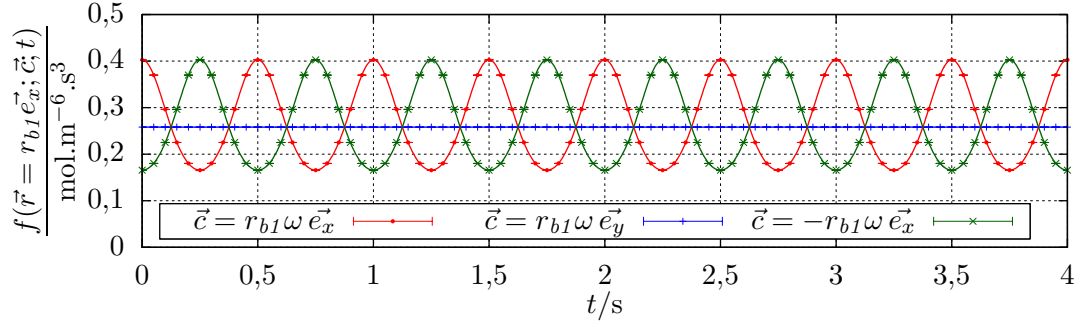
Ce mécanisme a par contre le même défaut, sur la récursivité, que celui évoqué au chapitre 5.5 : comme chaque nœud de l'arbre d'estimations multiplie sa branche par 4, la récursivité atteint vite de très hauts niveaux (cf. figure 6.3). C'est d'ailleurs pour cela que les algorithmes 6.1 et 6.2 n'ont été testés que jusqu'à $t = 4\text{ s}$ — contrairement aux autres, qui ont été testés jusqu'à $t = 5\text{ s}$. À $t = 4\text{ s}$, la récursivité (définie comme le nombre de tirages de T' dans un arbre d'estimations) moyenne atteint déjà 10^6 . En considérant :

- tous les tirages aléatoires effectués à chaque nœud d'un arbre — le temps de recul T' , mais aussi la vitesse \vec{C}_* et la direction \vec{U}' —, soit 11 tirages uniformes standards pour l'algorithme 6.2 si $T' > t_{\text{init}}$ (contre 7 tirages uniformes standards pour l'algorithme 6.1) (voir annexe D.4),
- qu'on effectue au total 10^4 réalisations,

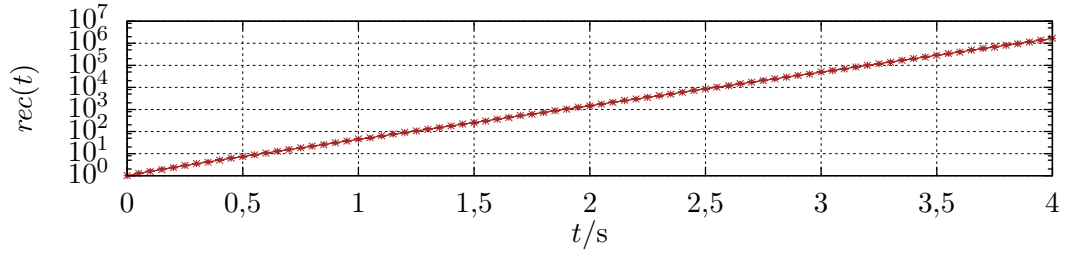
il faut effectuer 10^{11} tirages uniformes standards, pour donner une estimation à $t = 4\text{ s}$ en utilisant l'algorithme 6.2. Cela représente, encore à l'heure actuelle, un coût computationnel élevé (surtout si c'est pour calculer 1 seul point!).

- L'algorithme 6.3 est celui dont la variance explose le plus vite. Ce n'est pas surprenant, car parmi les algorithmes testés effectuant un choix de Bernoulli entre l'extinction et le terme source, c'est celui qui effectue ce choix de la façon la plus arbitraire. L'algorithme 6.4 fonctionne mieux, en utilisant de l'information préalable sur la distribution.

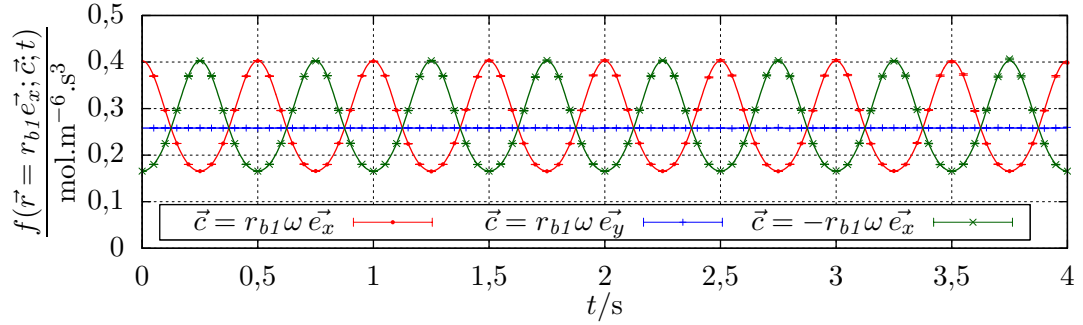
Il est intéressant de noter que l'algorithme 6.6 obtient des résultats similaires à l'algorithme 6.4, apparemment sans utiliser d'information *a priori* sur



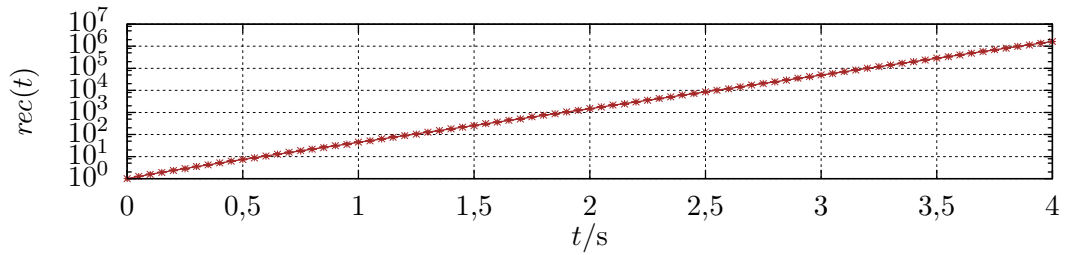
(a) Résultats de l'algorithme 6.1



(b) Récursivité moyenne de l'algorithme 6.1 calculant f , aux points de l'espace des phases décrit en (a)

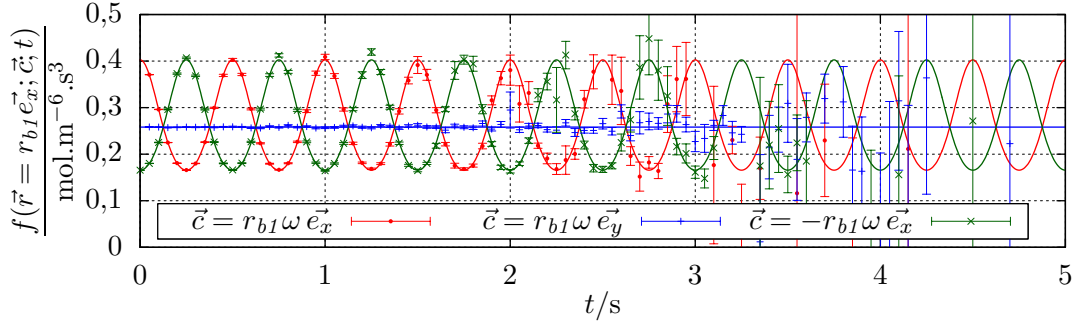


(c) Résultats de l'algorithme 6.2

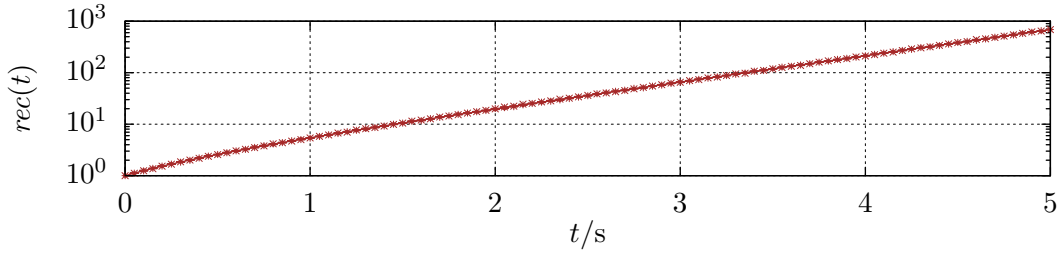
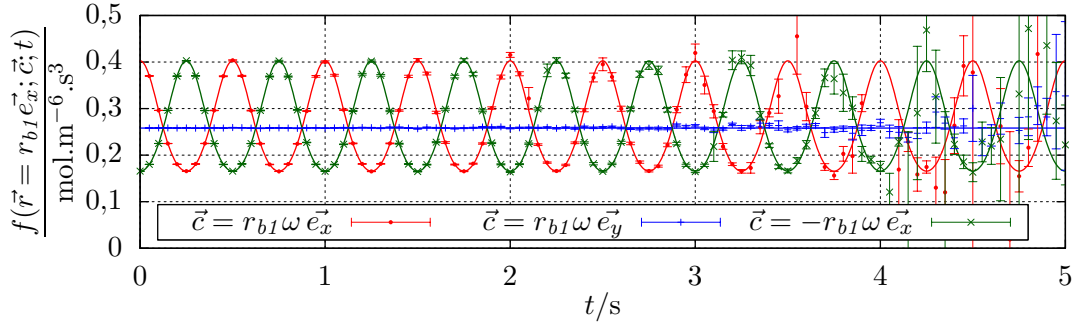


(d) Récursivité moyenne de l'algorithme 6.2 calculant f , aux points de l'espace des phases décrit en (c)

FIGURE 6.3 – Évolution de f selon le temps en 3 points fixes de l'espace des phases ; $r_{b1} = 0,2\text{m}$. Les solutions symboliques sont données en trait plein, et des résultats d'algorithmes Monte-Carlo sont donnés en points séparés avec barres d'erreur. En chaque point, 10^4 réalisations ont été utilisées.



(a) Résultats de l'algorithme 6.3

(b) Récursivité moyenne de l'algorithme 6.3 calculant f , aux points de l'espace des phases décrit en (a)

(c) Résultats de l'algorithme 6.4

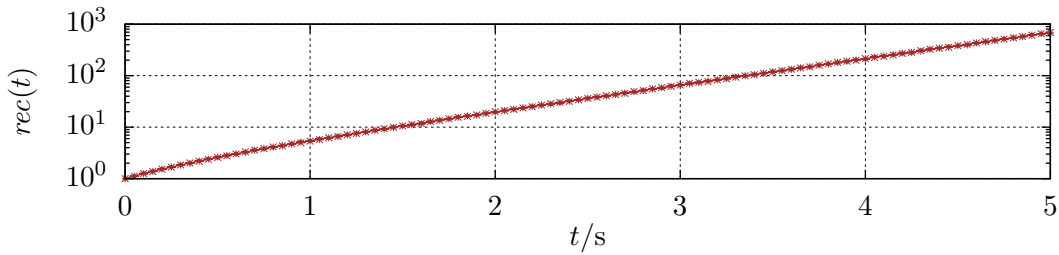
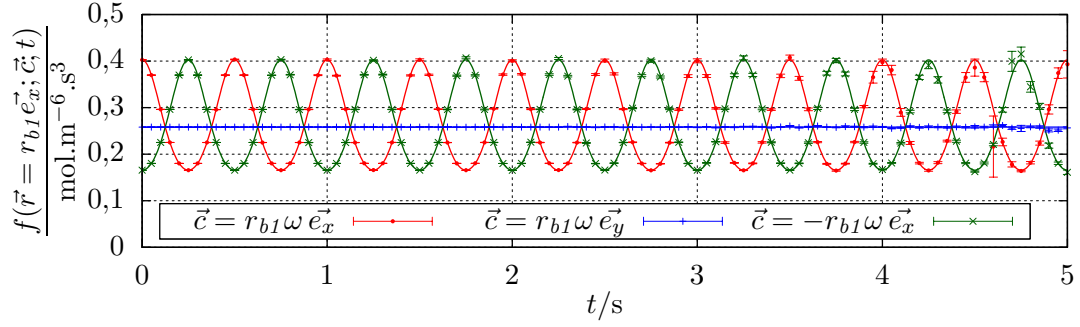
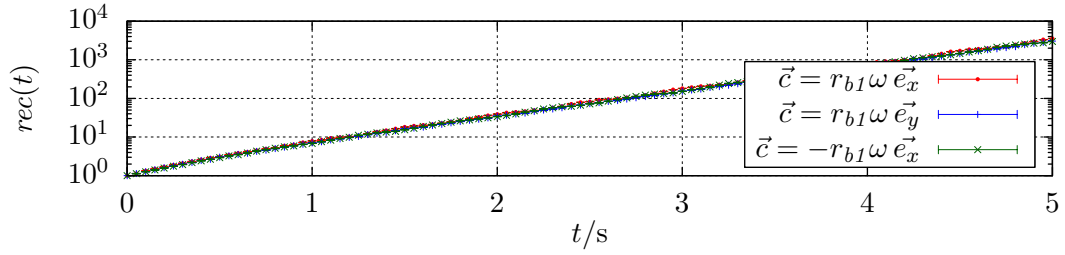
(d) Récursivité moyenne de l'algorithme 6.4 calculant f , aux points de l'espace des phases décrit en (c)

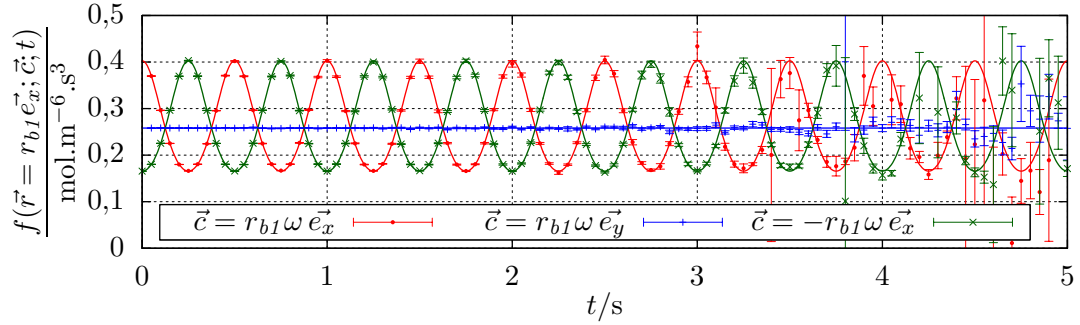
FIGURE 6.4 – Évolution de f selon le temps en 3 points fixes de l'espace des phases; $r_{b1} = 0,2\text{m}$. Les solutions symboliques sont données en trait plein, et des résultats d'algorithmes Monte-Carlo sont donnés en points séparés avec barres d'erreur. En chaque point, 10^4 réalisations ont été utilisées.



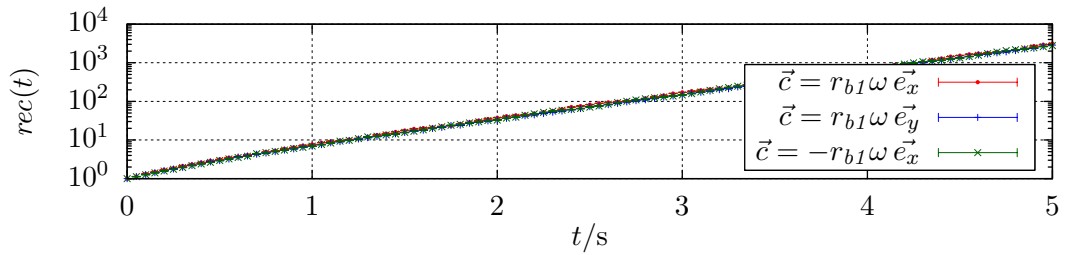
(a) Résultats de l'algorithme 6.5



(b) Récursivité moyenne de l'algorithme 6.5 calculant f , aux points de l'espace des phases décrit en (a)

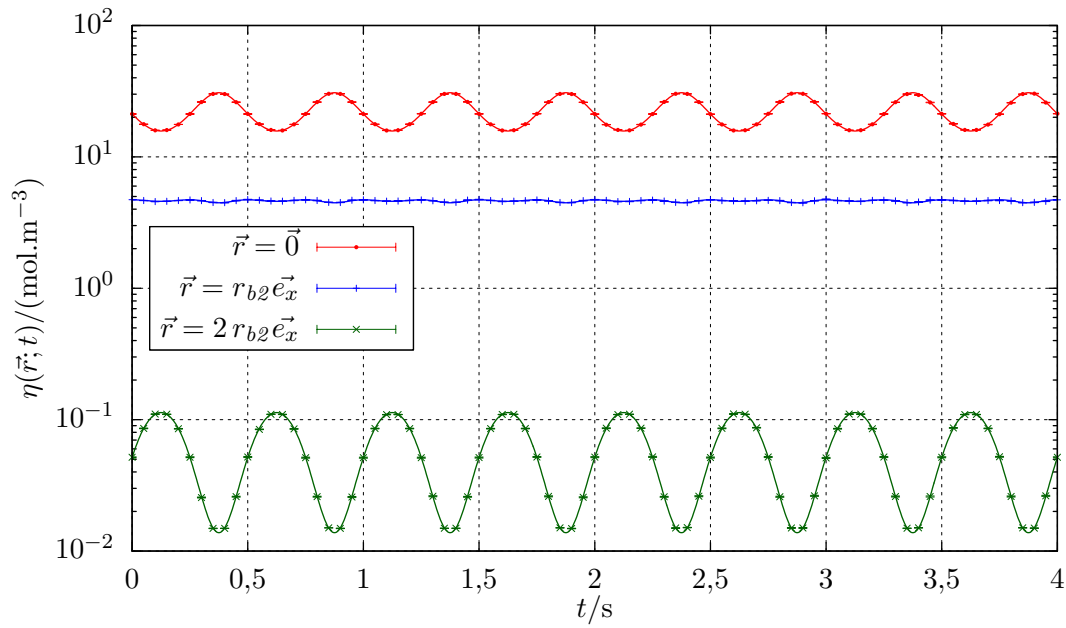


(c) Résultats de l'algorithme 6.6

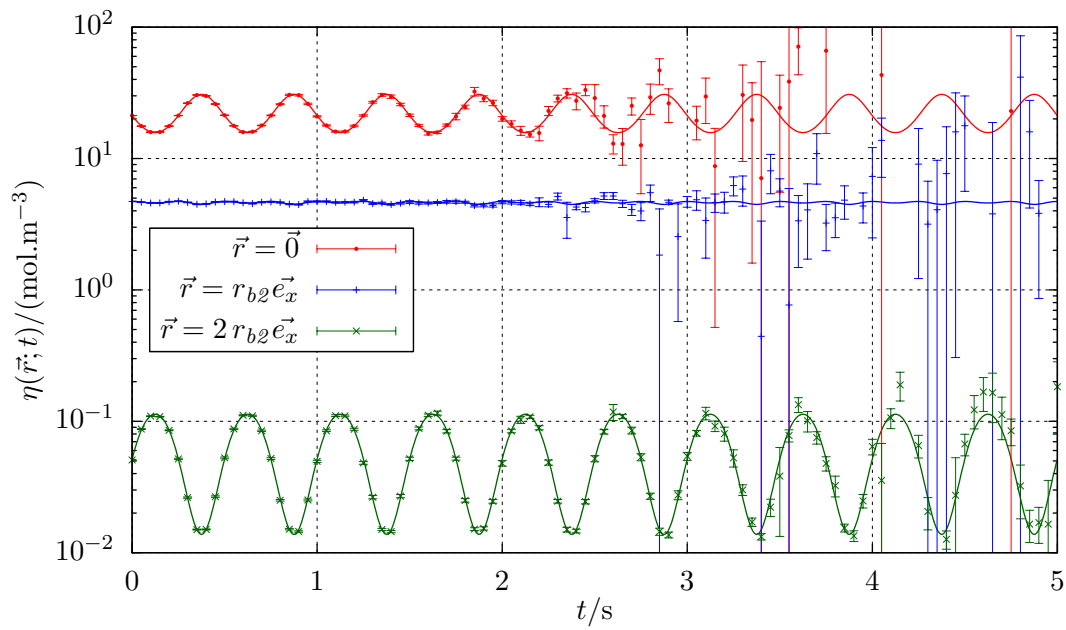


(d) Récursivité moyenne de l'algorithme 6.6 calculant f , aux points de l'espace des phases décrit en (c)

FIGURE 6.5 – Évolution de f selon le temps en 3 points fixes de l'espace des phases ; $r_{b1} = 0,2\text{m}$. Les solutions symboliques sont données en trait plein, et des résultats d'algorithmes Monte-Carlo sont donnés en points séparés avec barres d'erreur. En chaque point, 10^4 réalisations ont été utilisées.

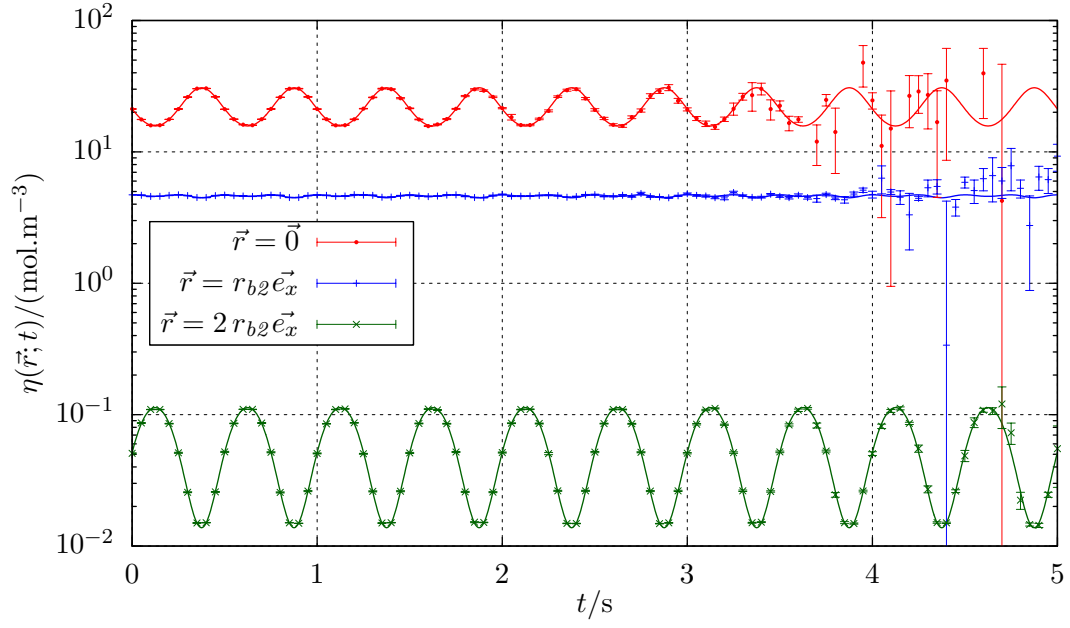


(a) Résultats de l'algorithme 6.2

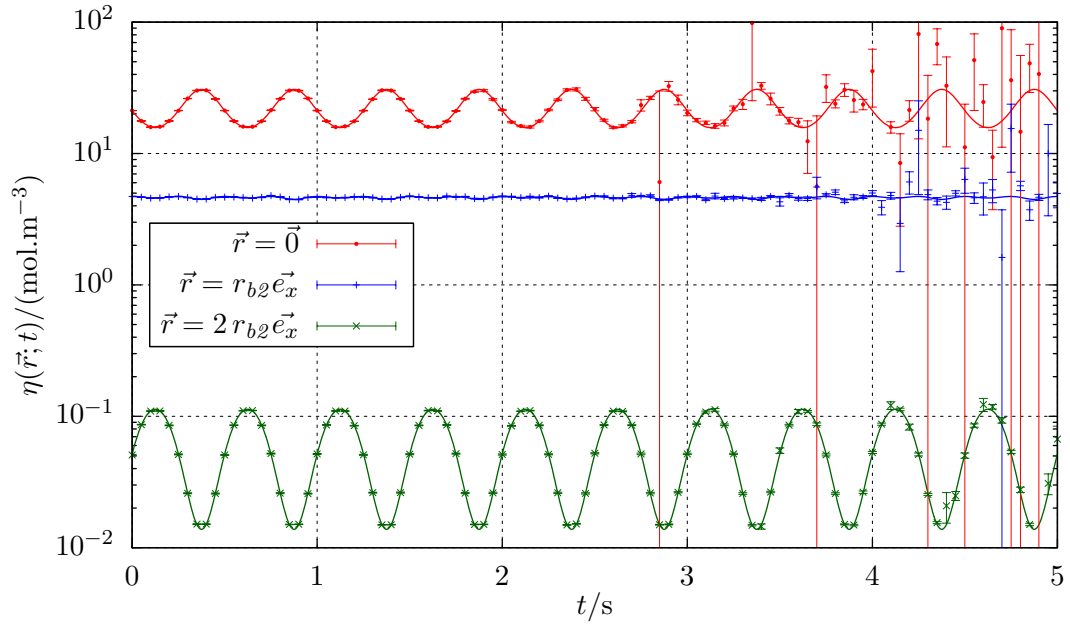


(b) Résultats de l'algorithme 6.3

FIGURE 6.6 – Évolution de η selon le temps en 3 points fixes de l'espace ; $r_{b2} = 0,25\text{ m}$. Les solutions symboliques sont données en trait plein, et des résultats d'algorithmes Monte-Carlo sont donnés en points séparés avec barres d'erreur. En chaque point, 10^4 réalisations ont été utilisées.

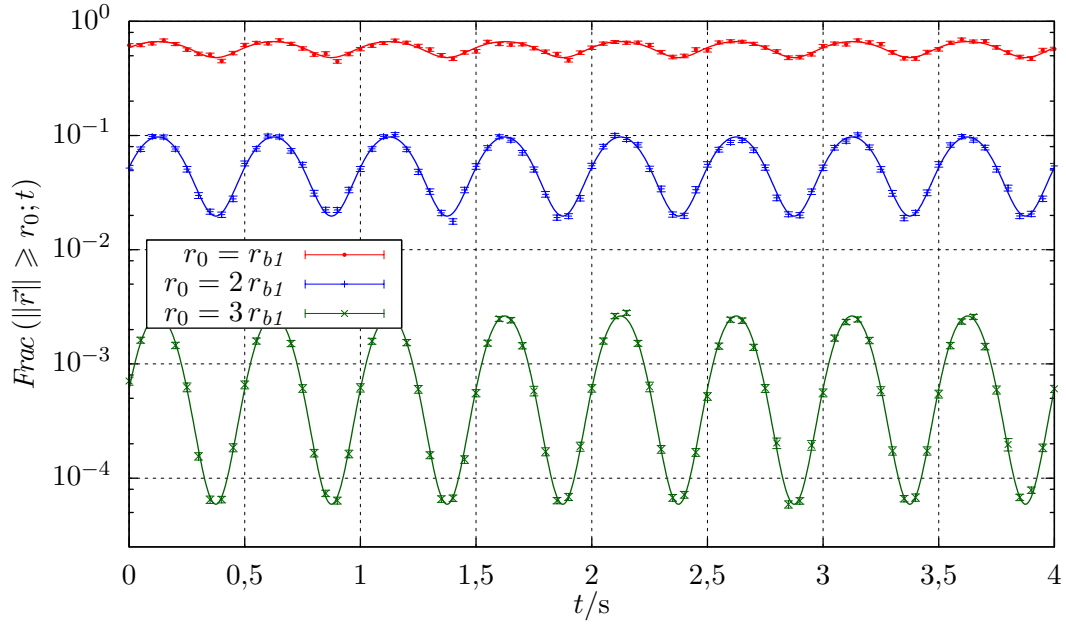


(a) Résultats de l'algorithme 6.4

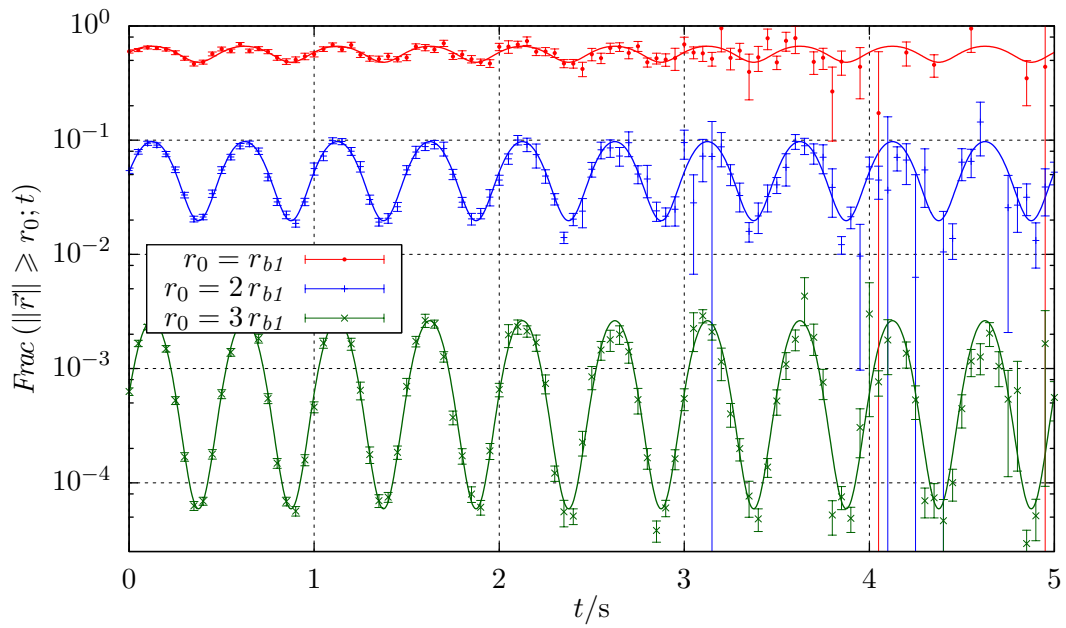


(b) Résultats de l'algorithme 6.6

FIGURE 6.7 – Évolution de η selon le temps en 3 points fixes de l'espace ; $r_{b2} = 0,25\text{ m}$. Les solutions symboliques sont données en trait plein, et des résultats d'algorithmes Monte-Carlo sont donnés en points séparés avec barres d'erreur. En chaque point, 10^4 réalisations ont été utilisées.

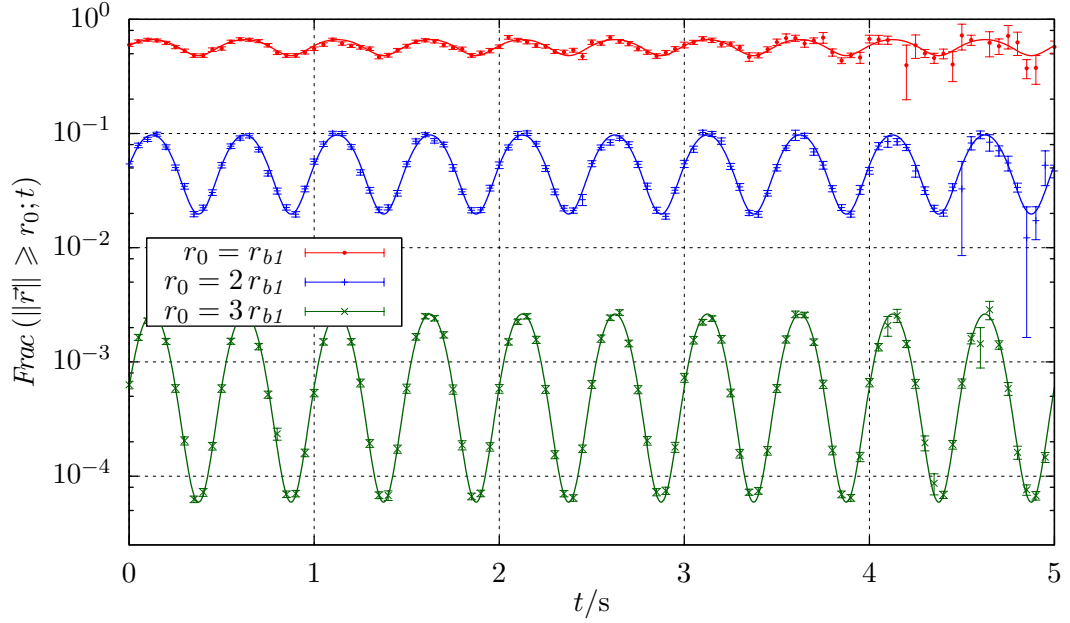


(a) Résultats de l'algorithme 6.2

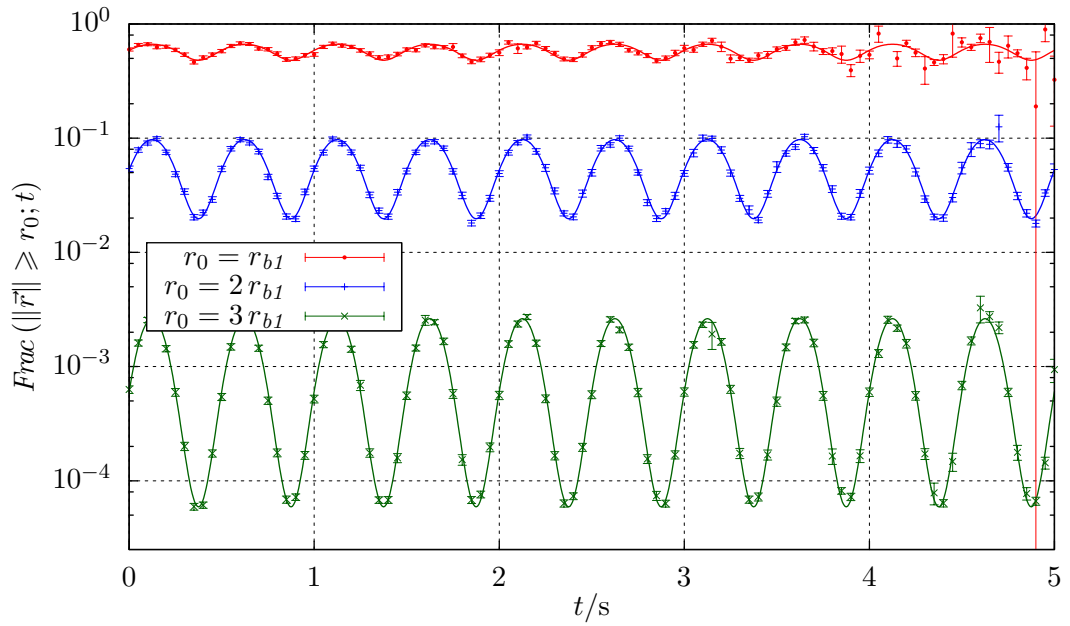


(b) Résultats de l'algorithme 6.3

FIGURE 6.8 – Évolution selon le temps de fractions des particules dont la distance au centre dépasse 3 seuils fixes ; $r_{b1} = 0,2$ m. Les solutions symboliques sont données en trait plein, et des résultats d'algorithmes Monte-Carlo sont donnés en points séparés avec barres d'erreur. En chaque point, 10^4 réalisations ont été utilisées.



(a) Résultats de l'algorithme 6.4



(b) Résultats de l'algorithme 6.6

FIGURE 6.9 – Évolution selon le temps de fractions des particules dont la distance au centre dépasse 3 seuils fixes ; $r_{b1} = 0,2$ m. Les solutions symboliques sont données en trait plein, et des résultats d'algorithmes Monte-Carlo sont donnés en points séparés avec barres d'erreur. En chaque point, 10^4 réalisations ont été utilisées.

la distribution. En fait, il dispose bien d'une information préalable sur la distribution : la vitesse moyenne et la température sont approximativement connues, car elles sont utilisées dans le tirage de \vec{C}_* . C'est la connaissance de la densité dont l'algorithme 6.6 arrive à se dispenser ; au prix d'une récursivité légèrement augmentée.

- On observe une difficulté à calculer les fractions de particules de grande énergie. Elle n'est pas rédhibitoire, mais plus marquée que ce qu'on avait observé avec le mode BKW.

Cela est peut-être lié à ce que cette fois-ci, nous nous sommes concentrés sur des événements rares *en position*, et non en vitesse comme au chapitre précédent. On sait que la distribution des particules, dans des zones peu peuplées de l'espace des phases, est très fortement influencée par la distribution dans les zones « majoritairement » peuplées de cet espace ; et que les particules de haute énergie (relativement aux autres) sont issues des zones fortement peuplées, avant de subir une courte et peu probable succession de collisions faisant augmenter leur énergie. Un propos de l'étude du mode BKW était d'ailleurs de comprendre ce mécanisme.

La configuration en confinement harmonique fait que les particules de haute énergie potentielle orbitent loin du centre attracteur, et de la majorité de la masse du gaz. Elles collisionnent donc entre elles de manière privilégiée, ce qui n'est nullement le cas dans le mode BKW. Si on ajoute que dans les 2 cas, l'information fournie aux algorithmes (sous forme de lois de tirage) est moins bonne dans les zones dépeuplées de l'espace des phases, cela donne un élément d'explication à la hausse de variance de notre estimateur de f , dans les zones de haute énergie potentielle de la configuration de confinement harmonique.

- Les algorithmes 6.1 et 6.5, où \vec{C}_* et \vec{U}' sont identiques dans le calcul de \widetilde{N}_{t-1} et \widetilde{S}_{is} , exhibent une variance largement plus faible que leurs équivalents 6.2 et 6.6, où \vec{C}_* et \vec{U}' sont tirées 2 fois.

Dans cet exemple on arrive à un résultat caricatural (variance nulle ou presque) à cause de la situation physique simulée. Comme expliqué précédemment, ceci est permis par la capacité de ces premiers algorithmes à profiter de la balance détaillée. Si la distribution des particules n'est plus maxwellienne en tout point, ils ne seront plus à variance nulle (ou presque), mais on peut penser qu'ils resteront les plus efficaces si la distribution des particules est proche d'une maxwellienne (c'est-à-dire proche de l'équilibre).

Du point de vue numérique, utiliser le même tirage du couple $(\vec{C}_*; \vec{U}')$ dans le calcul de \widetilde{N}_{t-1} et de \widetilde{S}_{is} , quand les 2 doivent être calculés, relève de l'utilisation de variables antithétiques (des variables aléatoires de covariance négative, comme annoncé au chapitre 1.3.2 et détaillé dans [54]). Même dans un calcul sur une situation de hors équilibre lointain, utiliser 2 tirages indépendants du couple $(\vec{C}_*; \vec{U}')$ dans les calculs \widetilde{N}_{t-1} et de \widetilde{S}_{is} paraît ainsi n'avoir, en première approche, aucun intérêt.

Perspectives, et conclusion

Ce **manuscrit** a commencé par la revisite en détails de 2 idées :

- les algorithmes à collisions nulles, au chapitre 2,
- les propositions de Jérémie DAUCHET sur le calcul par méthode de Monte-Carlo d'expressions non-linéaires d'intégrales, au chapitre 3.

Si nous avons fait leur révision avec autant de détails que dans ces 2 premiers chapitres, c'est parce que nous réutilisons ces 2 approches dans la suite du manuscrit. En fait, leur combinaison, dans le but d'une utilisation en cinétique des gaz, est constitutive de l'originalité de notre propos.

Pour construire, depuis ces 2 idées, une approche mathématique et numérique pour la cinétique des gaz, nous avons travaillé sur un cas simplifié. Ce cas est celui de la résolution des équations différentielles ordinaires scalaires d'ordre 1, présenté au chapitre 4. Nous l'avons imaginé comme une simplification de l'équation de Boltzmann, dont on aurait en particulier retiré l'espace des phases. Il nous permet de mettre en place toute la théorie utilisée ensuite ; les chapitres 5 et 6, qui montrent l'application de l'approche originale dans des problèmes de cinétique des gaz, ne font apparaître que de la mise en œuvre de ce qui est développé dans le chapitre 4.

Ce travail s'est fait sans hypothèse restrictive et sans accroc, même si nous nous sommes basés, tout au long du manuscrit, sur un formalisme mathématique détaillé pour ne pas commettre d'erreur. Nous avons fait le pari que des images physiques, permettant de ré-interpréter les développements mathématiques, pourraient être construites une fois que ces développements seraient en place ; et globalement, ce pari a réussi. Rien n'est venu invalider ou entraver notre construction d'un estimateur de la fonction de distribution des molécules, si ce n'est éventuellement l'apparition de variances d'estimateurs infinies, que nous avons étudiée aux chapitres 1.5.3, 4.3, 5.3 et 5.4.

En dépit de l'absence de spécificité — orientée vers la cinétique des gaz, parmi les différents phénomènes de transport descriptibles par les équations de Boltzmann — de l'approche que nous avons mis en place, celle-ci exhibe quelques particularités intéressantes du point de vue pratique, par exemple :

- Aucun maillage, de l'espace ou du temps, n'est utilisé pour résoudre numériquement l'équation de Boltzmann.

- On peut réaliser du calcul sonde, c'est-à-dire calculer la fonction de distribution en un point sans la calculer dans le reste de l'espace (ou du temps). À notre connaissance ceci constitue une nouveauté en cinétique des gaz, et plus généralement en mécanique des fluides.
- Calculer des événements rares, c'est-à-dire calculer la fonction de distribution en des lieux où elle est très faible, n'est pas spécialement difficile.

Ces inhabituelles propriétés sont apparues spontanément au fur et à mesure de la mise en place des algorithmes de calcul jusqu'aux 2 derniers chapitres, et sans lien avec des spécificités des exemples physiques alors à l'étude. Ainsi, bien que ces exemples soient académiques les conséquences pratiques observées sont intéressantes de façon générale.

●●●● Néanmoins, nous n'avons fait que défricher le sujet, et ce qui reste intéressant à explorer nous paraît vaste. Dans ce chapitre, nous listons quelques pistes pour continuer ce travail, qui nous paraissent prometteuses à court terme.

Premièrement, l'approche développée permet de réaliser des calculs en régime stationnaire.

Tous les cas physiques illustrant la méthode développée, et montrés dans ce manuscrit jusqu'ici, sont en régime stationnaire. Nous montrons dans ce chapitre un exemple de calcul dans un exemple de cinétique des gaz en régime stationnaire. Cet exemple sera toutefois détaillé dans la section consacrée à la 2^e piste d'exploration, pour faire d'une pierre deux coups.

Deuxièmement, dans nos cas physiques d'illustration il n'y avait jamais de parois aux systèmes.

Les expressions intégrales de la fonction de distribution, qui constituent l'élément clé de l'approche statistique pour la cinétique des gaz exposée dans ce manuscrit, ne font apparaître la géométrie que par des bornes d'intégration. La prise en compte de toutes sortes de géométries, éventuellement complexes, pourra être réalisée facilement, grâce aux outils issus de la synthèse d'image.

Troisièmement, nous n'avons pas essayé de travailler sur l'expression statistique de l'opérateur de collision de Boltzmann.

D'une part, nous nous sommes arrêtés, par simplicité, aux modèles de section efficaces de Maxwell. Même si nous savons que la technique numérique que nous avons développée ne se limite pas à ceux-ci, il serait bon d'en faire un exemple.

D'autre part, l'augmentation de la variance de nos estimateurs de la fonction de distribution avec le temps physique simulé, qui constitue le principal handicap à une utilisation directe de ce travail en calcul numérique appliqué, ne pourra être surmontée que grâce à un travail de fond sur la façon dont nous échantillonnons les termes de collision (extinction et terme source). Nous proposons dans ce chapitre plusieurs pistes de travail.

1 Calcul en régime permanent

●●●● Les cas académiques que nous avons traités, aux chapitres 5 et 6 sont en régime instationnaire. La fixation de la distribution du gaz s'y fait par une condition initiale.

D'un point de vue formel, le traitement d'une situation stationnaire, avec des entrées et sorties de matière et d'énergie, devrait être relativement semblable. Dans l'expression (6.9), la condition aux limites n'est plus prise en compte par :

$$f(\vec{r}; \vec{c}; t) = \int_{-\infty}^t dt' \hat{\nu}(\vec{r}_b(t'); \vec{c}_b(t'); t') \exp\left(-\int_{t'}^t dt'' \hat{\nu}(\vec{r}_b(t''); \vec{c}_b(t''); t'')\right) \times \\ \left(H(t' \leq t_{init}) f(\vec{r}_b(t_{init}); \vec{c}_b(t_{init}); t_{init}) + H(t' > t_{init}) \times \dots \right) \quad (1)$$

mais par l'expression plus générale :

$$f(\vec{r}; \vec{c}; t) = \int_{-\infty}^t dt' \hat{\nu}(\vec{r}_b(t'); \vec{c}_b(t'); t') \exp\left(-\int_{t'}^t dt'' \hat{\nu}(\vec{r}_b(t''); \vec{c}_b(t''); t'')\right) \times \\ \left(H(t' \leq t_m) f(\vec{r}_b(t_m); \vec{c}_b(t_m); t_m) + H(t' > t_m) \times \dots \right) \quad (2)$$

où $t_m = \max\{t' \in]-\infty; t], (\vec{r}_b(t'); \vec{c}_b(t'); t') \in \text{limites}\}$ est l'instant de la dernière intersection de la trajectoire balistique remontée avec une limite. Cette limite peut être un instant initial, une paroi, ou une frontière autre qu'une paroi.

●●●● S'il n'y a plus de condition initiale, plusieurs points d'achoppements peuvent néanmoins apparaître :

- En deçà d'un nombre de Knudsen critique, l'algorithme se met à suivre, en remontant le temps, un arbre de trajectoires gonflant de façon exponentielle. Sans condition initiale, on ne peut pas arrêter l'inflation d'un tel arbre à un temps physique fixe. Selon les tirages aléatoires que l'algorithme fait — surtout les premiers —, il peut devenir incapable de se terminer (voir [55] pour des détails).

Nous n'avons pour l'instant pas essayé de comprendre comment remédier à ce mécanisme.

- Le problème physico-mathématique peut être mal posé. Ceci est une difficulté courante en cinétique des gaz.

Elle apparaît par exemple quand on veut déterminer un écoulement dans un circuit fermé. Dans ce cas, si les parois sont étanches, les conditions aux frontières ne fixent pas la masse totale dans le système. Celle-ci reste donc un paramètre libre, à moins d'être déterminée par exemple par une condition initiale.

Ce n'est pas un problème pour les méthodes de calcul de type DSMC ou BsR, car elles utilisent toujours une condition initiale. Partant de celle-ci, elles calculent l'évolution du système jusqu'à ce qu'il ait convergé (au besoin, en truquant cette évolution de sorte qu'elle se finisse plus rapidement, sur le même état final). Une telle approche peut sûrement être tentée avec une technique de calcul inspirée de ce manuscrit, mais elle signifierait l'introduction d'une hypothèse nouvelle (« Le temps simulé est assez grand pour que le système ait convergé. »).

2 Gestion des géométries complexes

●●●● Dans les 2 cas académiques que nous avons traités, il n'y a pas de paroi.

L'existence de parois ne pose pas de problème dans les écritures intégrale (6.9) ou statistiques (6.10a) et (6.11) de l'équation de Boltzmann. On peut les prendre en compte comme expliqué précédemment en (2). Le cas échéant, il faut fixer une relation sur f exprimant le comportement du gaz à la paroi (réflexion spéculaire, rebond maxwellien, adsorption, émission, . . .), comme montré au chapitre 1.2.3. Il reste à savoir calculer les intersections entre les trajectoires balistiques et les parois.

●●●● Il est important de comprendre que les parois ne rentrent en compte, dans la méthode de calcul que nous proposons ainsi que dans les formulations mathématiques dont elle est issue, que *via* ce calcul d'intersections. En particulier, il n'y a aucun maillage ni aucune quadrature devant être réglé en fonction de la géométrie de sorte que les calculs fonctionnent. Du point de vue informatique, l'interaction entre l'algorithme de calcul et la géométrie ne se fait que par une routine de demande de calcul d'intersection.

Ce dernier point nous permet de profiter du travail de la communauté de la synthèse d'image. En effet, cette prise en compte de la géométrie uniquement par le calcul d'intersections se retrouve dans la technique dite de « lancer de rayon », utilisée en rendu photo-réaliste. Ainsi, une telle gestion de la géométrie est pour cette communauté une problématique fondamentale, ayant amené à la construction d'un savoir-faire étendu, concernant en particulier :

- la réalisation du calcul d'intersections dans toutes sortes de géométries, de façon générique, quelle que soit leur complexité,
- la transmission au code de calcul proprement dit de la géométrie, décrite extérieurement,
- la réalisation à faible coût computationnel du calcul d'intersections, dans les géométries très complexes.

Le lecteur cherchant un exposé extensif de ce savoir-faire pourra se reporter par exemple à l'ouvrage [92].

Ces outils sont surtout utilisables quand les trajectoires balistiques sont droites : car en synthèse d'image, les rayons lumineux sont majoritairement droits. En

cinétique des gaz, s'il y a une force à longue portée les trajectoires peuvent être courbes ; le calcul d'intersections trajectoires/géométrie devient alors plus compliqué.

2.1 Exemple

●●●● Nous avons demandé à Mésostar, une entreprise partenaire, de réaliser des implémentations de formes algorithmiques semblables à celles présentées au chapitre 6 dans des exemples géométriques, à l'aide d'EDstar, un environnement de développement que STARwest (l'équipe qui a encadré ce travail de thèse) a partiellement monté.

●●● La création d'EDstar a fait suite au constat que, lorsqu'on utilise la méthode de Monte-Carlo pour le calcul du transfert radiatif, la géométrie du problème n'intervient que *via* un calcul d'intersections rayon/géométrie — comme expliqué à l'instant au sujet de la cinétique des gaz. L'équipe STARwest avait alors déjà eu l'idée d'utiliser les avancées en synthèse d'image (voir [32, 104]). EDstar était aussi l'occasion de fixer et réutiliser des parties de codes systématiquement présentes dans tous ses codes de calcul par MMC, comme la parallélisation, dont l'écriture était un exercice peu utile effectué à chaque nouvelle simulation.

EDstar se présente comme un code source partiel, en C++, où l'utilisateur doit coder l'algorithme de Monte-Carlo qu'il veut implémenter. La génération de nombres aléatoires de loi uniforme standard, ainsi que le calcul d'intersections rayon/géométrie sont fournis sous forme de routines qu'il suffit d'appeler depuis l'algorithme. Par ailleurs, la parallélisation, la lecture du fichier de géométrie (écrit séparément), la garantie de l'indépendance des tirages aléatoires entre processus parallèles, ou le calcul final des moyennes et des écart-types sont déjà en place. L'utilisateur ne doit détailler qu'un bloc fonction fournissant un *vecteur résultat ponctuel* de l'algorithme (ses coordonnées sont des nombres réels) à chaque appel, appel qui se comprend comme une réalisation de l'algorithme de Monte-Carlo.

Pour mettre en place EDstar, STARwest s'est basé sur le code de synthèse d'image *Physically Based Rendering Techniques* (PBRT). Ce code, conçu comme une base d'entraînement pour des étudiants en synthèse d'image, a pour nous plusieurs avantages comme base pour EDstar :

- Il est sous licence libre.
 - Il est modulaire, et écrit sur une base physique en profitant du caractère *orienté objet* du langage C++.
- « Orienté objet » signifie qu'on peut définir des structures de données, appelées *classes*, avec des fonctions associées, appelées *méthodes*. Par exemple, pour représenter des vecteurs de l'espace ordinaire, on dispose dans PBRT de la classe **Vector**, qui possède 3 attributs : **float x, y, z** ;. De nombreuses méthodes lui sont associées, comme par exemple l'addition, la multiplication par un réel, ou le produit scalaire :

```

class Vector {
    float x, y, z;    // attributs
    Vector(float xx, float yy, float zz)
        : x(xx), y(yy), z(zz) { }    // constructeur
    ...
    Vector operator+(const Vector &v) const {
        return Vector(x + v.x, y + v.y, z + v.z);
    }
    Vector operator*(float f) const { return Vector(f*
        x, f*y, f*z); }
    ...
};
...
inline Vector operator*(float f, const Vector &v) {
    return v*f; }
inline float Dot(const Vector &v1, const Vector &v2)
    { return v1.x * v2.x + v1.y * v2.y + v1.z * v2.z
    ; }    // le produit scalaire
...

```

Grâce à une telle définition on peut écrire du code comme :

```

    Vector a, b, c;
...    // des calculs modifiant a et b
    c = a + 0.5 * Dot (a, b) * b;

```

qui signifie ce qu'il signifie intuitivement.

Des classes sont fournies pour toute sortes d'abstractions : des formes géométriques (disque, cône, maillage de triangles, ...), des matériaux de surface (mats de couleur fixe, avec réflectivité interpolée, ...) ... Le gain en vitesse d'écriture et en lisibilité du code est énorme.

- Il est conçu de sorte à ce que des fonctionnalités puissent lui être ajoutées facilement. Par exemple, il y a un modèle de classe pour les formes (la classe abstraite **Shape**) avec une interface standard ; si les formes disponibles par défaut dans PBRT ne conviennent pas à un utilisateur, il peut en rajouter une en ajoutant une sous-classe de **Shape** avec le détail de son implémentation.
- Et surtout, il en existe une documentation complète : [91].

●●● 2 séries de tests ont été réalisées, chacune dans une géométrie particulière : un tube tortueux, et une courte longueur de tube munie d'une gorge. Chacune de ces géométries est décrite par un maillage de triangles ; elles sont montrées en figure 1.

Dans les 2 géométries, on se place en régime permanent. Le tube relie 2 réservoirs. De chaque réservoir sort une distribution de particules à l'équilibre, avec une densité et une température fixes, et une vitesse moyenne nulle. Les 2 réservoirs sont à

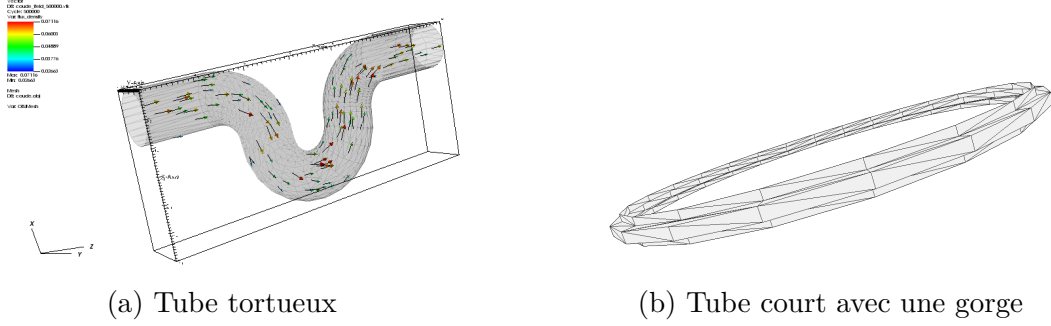


FIGURE 1 – Géométries maillées, utilisées dans des tests de prise en compte de géométries complexes.

pressions égales, mais à 2 températures différentes. Hors des connexions avec les réservoirs, les parois réfléchissent les particules en les thermalisant : la distribution des particules quittant les parois est maxwellienne en tout point, à vitesse nulle et température de la paroi, avec une densité réglée de sorte qu'aucun flux de matière ne traverse la paroi. La température des parois suit un profil affine selon la coordonnée z , continu avec les températures des réservoirs — les connexions aux réservoirs suivent bien des plans d'équation $z = \text{cste}$.

Les simulations illustrent le phénomène de *transpiration thermique* : dans de telles configurations, la mécanique des fluides à l'échelle macroscopique prévoit que le débit de matière dans le tube est nul ; lorsqu'on est en régime microfluidique, et qu'une approche macroscopique n'est plus valable, il en va différemment.

Dans le tube tortueux est simulé une situation sans collision inter-particules. La raison est qu'on n'arrivait pas à obtenir des effets significatifs des collisions, sans que l'algorithme de calcul ne cesse de fonctionner (voir la section précédente 1). On choisit plutôt d'illustrer la dépendance du débit de matière dans le tube envers la différence de température entre les extrémités (voir la figure 2).

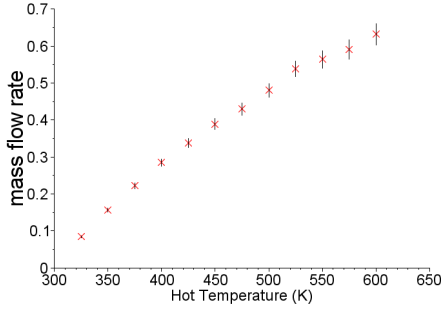
Dans le tube court, on illustre la dépendance du champ de densité de débit selon le taux de collision. On choisit un modèle de collision de Maxwell isotrope, et les températures des réservoirs sont fixées à 300 K et 350 K. Voir la figure 2.

3 Perspectives formelles sur l'opérateur de collision de Boltzmann

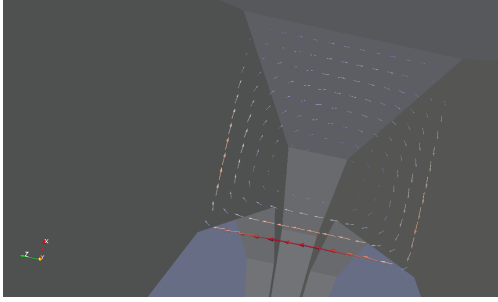
3.1 Variation des sections efficaces

Dans les cas tests de ce manuscrit, on a systématiquement utilisé le modèle de collisionneurs de Maxwell isotrope.

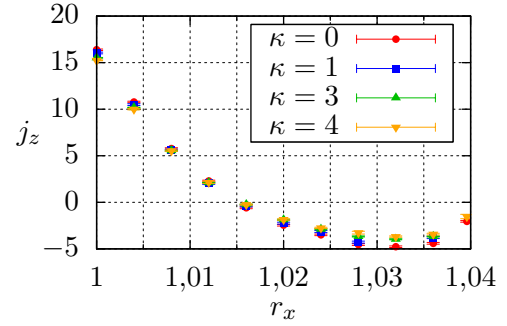
Il y avait là essentiellement une volonté de se simplifier le travail, en posant $b_F = \text{cste}$. Il serait bon de tester d'autres modèles de section efficace, à deux titres :



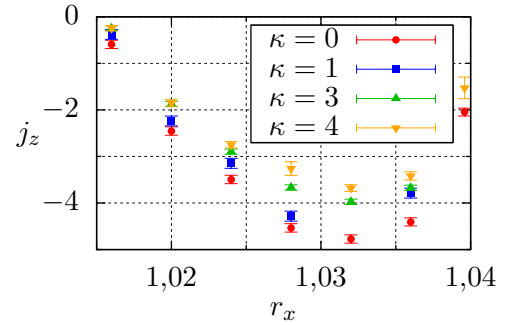
(a) Profil du débit de matière dans le tube sinueux, en fonction de la température de l'un des réservoirs, l'autre étant supposé à 300 K.



(b) Champ de densité de débit dans la gorge du tube court, obtenu par une série de calculs sondes sur les points d'une grille. Dans le calcul ayant donné ces résultats, il n'y a pas de collisions.



(c) Profil de la densité de débit traversant le plan de symétrie de la gorge du tube court, en fonction de la profondeur dans la gorge ($1 \leq r_x \leq 1,04$) et du coefficient réglant la section efficace de collision.



(d) Zoom sur une partie de (c)

FIGURE 2 – Résultats obtenus dans les configurations impliquant des géométries complexes, décrites dans ce chapitre.

- Prouver que la méthode que nous avons développée accepte toute sorte de modèle de section efficace (à condition d'adapter les lois de tirage, si on veut qu'elle reste un minimum efficiente), et pas seulement celui de Maxwell isotrope. Étant donnés les développements formels illustrés dans ce manuscrit nous en sommes convaincus, mais il serait mieux de le montrer concrètement.
- Le modèle de collisionneurs de Maxwell donne des résultats assez éloignés de ce qu'on constate expérimentalement. Il serait bien d'étendre la méthode développée vers des modèles de collisionneurs plus réalistes, ne serait-ce que le modèle de sphères dures.

Quelque chose que nous pouvons affirmer dès maintenant, est que les densités de probabilité pour \vec{C}_* et \vec{U}' assurant une variance nulle quand la situation physique

est à l'équilibre, ne seront plus les mêmes que dans les cas tests de ce manuscrit — loi normale multidimensionnelle pour \vec{C}_* et loi isotrope pour \vec{U}' .

3.2 Combinaison du terme source et de la fréquence de collision

●●●● Dans la formulation intégrale (6.9), générique pour la cinétique des gaz et utilisable en MMC, apparaît une récursivité de facteur 4. Nous montrons peu après dans le chapitre 6.2 comment cette récursivité peut être diminuée, en introduisant un choix de Bernoulli.

Nous faisons plusieurs propositions, pour orienter le choix de Bernoulli : appliquer un choix purement arbitraire, se baser sur une approximation de l'extinction connue au préalable, ou estimer récursivement l'extinction d'office. On ne se basait pas sur le terme source, car dans le cas de confinement harmonique étudié au chapitre 6 il compensait exactement l'extinction : $s_{is}(\vec{r}; \vec{c}; t) = \nu_t(\vec{r}; \vec{c}; t)f(\vec{r}; \vec{c}; t)$. Dans d'autres situations physiques il serait sûrement utile de le prendre en compte.

●●●● Par ailleurs il serait intéressant de prendre en compte le terme source et l'extinction simultanément, autrement qu'exposé dans les algorithmes 6.1 et 6.2. Le terme source est dû à la présence de particules à des vitesses autres que la vitesse d'intérêt, qui collisionnent ; de même l'extinction est due à la présence de particules qui collisionnent avec la particule suivie. Je pense qu'il est possible de lier leurs estimations, de façon à maîtriser la variance sans introduire une récursivité d'un facteur 4 à chaque collision.

Je pense aussi qu'un gain en variance est obtensible de la sorte même dans le chapitre 5. Dans ce chapitre, nous avons ignoré l'estimation de la fréquence d'extinction, car le cas d'étude la rendait constante. Cela rendait hélas le modèle mathématique associé à nos algorithmes instable en densité, ce qui explique en partie la façon dont la variance des estimateurs calculés diverge avec le temps simulé. Si on inclue une estimation de la fréquence de collision dans les algorithmes du chapitre 5, cela devrait corriger l'instabilité en densité.

Mais alors, les estimations de l'extinction et du terme source devraient être profondément liées, pas comme dans ce qu'on avait proposé au chapitre 6.2, en introduisant un choix de Bernoulli entre l'extinction et le terme source. Si on ne considère pas la conservation de la masse totale dans le terme collisionnel de Boltzmann C_B , la différence de l'extinction et du terme source, tous 2 quadratiques en f , provoque une instabilité en densité comme au chapitre 5.3.2. Cela me fait dire que pour construire un estimateur de la fonction de distribution f , dans un problème de cinétique des gaz, dont la variance ne diverge pas avec le temps simulé, et n'utilisant pas de récursivité accrue comme proposée au chapitre 5.4, il faudra tenir compte de la conservation de la masse dans C_B , d'une façon ou d'une autre.

●●●● On pourrait par exemple profiter des écritures alternatives suivantes de l'opérateur de collision de Boltzmann : en posant $\vec{g} = \vec{c} - \vec{c}_*$ (pour simplifier les

écritures), et aussi $\vec{k} = \vec{c}' - \vec{c} = \vec{c}_* - \vec{c}'_*$ de $\vec{l} = \vec{c}' - \vec{c} = \vec{c}_* - \vec{c}'$, on a :

$$\begin{aligned}
 & C_B(\vec{r}; \vec{c}; t) \\
 &= \int_{E_c} d\vec{c}_* \int_{4\pi} d\vec{u}' b_F\left(\|\vec{g}\|; \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|}\right) \times \\
 &\quad \left(-f(\vec{r}; \vec{c}; t) f(\vec{r}; \vec{c}_*; t) + f(\vec{r}; \vec{c}'; t) f(\vec{r}; \vec{c}'_*; t)\right) \\
 &= \int_{E_c} d\vec{c}_* \int_{E_c \setminus \perp \mathbb{R}\vec{l}} \frac{4 d\vec{k}}{\vec{g}^2} f(\vec{r}; \vec{c}'_*; t) \left(b_F\left(\|\vec{g}\|; \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|}\right) f(\vec{r}; \vec{c}'; t) - \right. \\
 &\quad \left. \frac{\vec{l}^2}{\vec{g}^2} b_F\left(\|\vec{l}\|; \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|}\right) f(\vec{r}; \vec{c}; t)\right) \tag{3a}
 \end{aligned}$$

$$\begin{aligned}
 &= \int_{E_c} d\vec{c}_* \int_{4\pi} d\vec{u}' f(\vec{r}; \vec{c}'_*; t) \left(b_F\left(\|\vec{g}\|; \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|}\right) f(\vec{r}; \vec{c}'; t) - \right. \\
 &\quad \left. \frac{1}{2} \left(1 + \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|}\right) b_F\left(\|\vec{l}\|; \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|}\right) f(\vec{r}; \vec{c}; t)\right) \tag{3b}
 \end{aligned}$$

$$\begin{aligned}
 &= \int_{E_c} d\vec{c}_* \int_{E_c \setminus \perp \mathbb{R}\vec{k}} \frac{4 d\vec{l}}{\vec{g}^2} f(\vec{r}; \vec{c}'; t) \left(b_F\left(\|\vec{g}\|; \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|}\right) f(\vec{r}; \vec{c}'_*; t) - \right. \\
 &\quad \left. \frac{\vec{k}^2}{\vec{g}^2} b_F\left(\|\vec{k}\|; \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|}\right) f(\vec{r}; \vec{c}; t)\right) \tag{3c}
 \end{aligned}$$

$$\begin{aligned}
 &= \int_{E_c} d\vec{c}_* \int_{4\pi} d\vec{u}' f(\vec{r}; \vec{c}'; t) \left(b_F\left(\|\vec{g}\|; \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|}\right) f(\vec{r}; \vec{c}'_*; t) - \right. \\
 &\quad \left. \frac{1}{2} \left(1 - \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|}\right) b_F\left(\|\vec{k}\|; \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|}\right) f(\vec{r}; \vec{c}; t)\right) \tag{3d}
 \end{aligned}$$

$$\begin{aligned}
 &= \int_{E_c} d\vec{c}_* \int_{4\pi} d\vec{u}' b_F\left(\|\vec{g}\|; \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|}\right) f(\vec{r}; \vec{c}'; t) f(\vec{r}; \vec{c}'_*; t) - \\
 &\quad \frac{1}{4} \left(1 - \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|}\right) b_F\left(\|\vec{k}\|; \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|}\right) f(\vec{r}; \vec{c}'; t) f(\vec{r}; \vec{c}; t) - \\
 &\quad \frac{1}{4} \left(1 + \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|}\right) b_F\left(\|\vec{l}\|; \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|}\right) f(\vec{r}; \vec{c}'_*; t) f(\vec{r}; \vec{c}; t) \tag{3e}
 \end{aligned}$$

où :

$$\left\{ \begin{array}{l} \vec{g} + \vec{k} + \vec{l} = \vec{0} \\ \vec{k} \perp \vec{l} \\ \vec{c}' = \frac{\vec{c} + \vec{c}_*}{2} + \frac{\|\vec{g}\|}{2} \vec{u}' = \vec{c} + \vec{k} = \vec{c}_* - \vec{l} \\ \vec{c}'_* = \frac{\vec{c} + \vec{c}_*}{2} - \frac{\|\vec{g}\|}{2} \vec{u}' = \vec{c}_* - \vec{k} = \vec{c} + \vec{l} \\ \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|} = \frac{\vec{l}^2 - \vec{k}^2}{\vec{l}^2 + \vec{k}^2} = \frac{\vec{l}^2 - \vec{k}^2}{\vec{g}^2} \\ \frac{\vec{l}^2}{\vec{g}^2} = \frac{1}{2} \left(1 + \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|} \right) \\ \frac{\vec{k}^2}{\vec{g}^2} = \frac{1}{2} \left(1 - \vec{u}' \cdot \frac{\vec{g}}{\|\vec{g}\|} \right) \end{array} \right. \quad (3f)$$

3.3 Utilisation d'information connue *a priori*

●●●● Au chapitre 4, on montrait comment on pouvait utiliser une solution approximative, connue de prime abord, pour faciliter la résolution d'un problème différentiel non-linéaire par méthode de Monte-Carlo. Ceci venait en complément :

- des approches à variance nulle [4, 17, 32, 50],
- de certaines techniques de calcul où on ne calcule par MMC que la différence entre la solution exacte d'un modèle physico-mathématique et une solution approximative, donnée par une méthode à discrétisation. La publication [103] en donne un exemple en transport linéaire, en transfert radiatif dans les milieux très diffusants. Nous expliquons, au chapitre 4.4, en quoi ces techniques deviennent encore plus avantageuses quand le problème différentiel résolu est non-linéaire.

Malheureusement, par manque de temps, nous n'avons pas appliqué cette méthodologie en cinétique des gaz.

●●●● Le test le plus rapide, venant à l'esprit de l'auteur, pourrait être effectué sur le mode BKW. On pourrait utiliser comme approximation de guidage le modèle collisionnel BGK [8]. Dans ce modèle, les collisions provoquent un rappel de la distribution vers l'équilibre, selon un temps de relaxation constant dans l'espace des vitesses :

$$C_{BGK}(\vec{r}; \vec{c}; t) = \frac{1}{\tau(\vec{r}; t)} \left(f_{eq}(\vec{c}; \eta(\vec{r}; t); \vec{v}(\vec{r}; t); c_{qma}(\vec{r}; t)) - f(\vec{r}; \vec{c}; t) \right) \quad (4)$$

où $\eta(\vec{r}; t)$, $\vec{v}(\vec{r}; t)$, et $c_{qma}(\vec{r}; t)$ sont les densité, vitesse moyenne et vitesse quadratique moyenne sur chaque axe locales de la distribution f , et $\tau(\vec{r}; t)$ est le temps de relaxation.

Ce modèle a l'avantage d'être solvable symboliquement dans le cas du mode BKW, et d'y donner une solution approximative très simple. De plus, il est avec ses modèles dérivés ES-BGK et Shakhov [2, 58, 107, 125] le modèle collisionnel simplifié le plus utilisé en calcul numérique de la cinétique des gaz, particulièrement avec les méthodes BsR [93, 94, 109–111].

●●●● Les avantages qu'il y a à utiliser une solution approximative comme guide pour la MMC, dans la résolution d'un problème différentiel non-linéaire, laisse penser qu'à moyen terme la MMC pourra être utilisée en cinétique des gaz en tandem avec une méthode à discrétisation, comme DSMC ou BsR. Un tel couple pourrait fonctionner selon un schéma prédicteur-correcteur : à chaque pas de temps, la méthode à discrétisation, travaillant à basse résolution, générerait une solution approximative, corrigée ensuite par la MMC utilisant tous les points déjà calculés comme guides et fonctionnant en méthode de référence sur le problème différentiel et ses conditions limites d'origine.

Annexe A

Rappels de mathématiques utilisées dans ce manuscrit

A.1 Développements en Série Entière

Définition A.1.1 (modes de convergence d'une suite de fonctions).

Soit $(f_n)_{n \in \mathbb{N}}$ une suite de fonctions à variable réelle et à valeurs dans un \mathbb{R} -espace vectoriel E de norme $\|\bullet\|_E$, soit f une fonction à variable réelle et à valeurs dans E , et soit I un intervalle non-dégénéré de \mathbb{R} .

On dit que :

- $(f_n)_{n \in \mathbb{N}}$ converge *simplement* sur I vers f , ssi pour tout $x \in I$ la suite $(f_n(x))_{n \in \mathbb{N}}$ est convergente, vers $f(x)$:

$$\forall x \in I, \forall \varepsilon > 0, \exists N \in \mathbb{N}, \forall n \geq N, \|f_n(x) - f(x)\|_E \leq \varepsilon$$

- $(f_n)_{n \in \mathbb{N}}$ converge *uniformément* sur I vers f , ssi la suite réelle $(\|f_n - f\|_{\infty, E})_{n \in \mathbb{N}}$ converge vers 0 :

$$\forall \varepsilon > 0, \exists N \in \mathbb{N}, \forall n \geq N, \forall x \in I, \|f_n(x) - f(x)\|_E \leq \varepsilon$$

Par extension, on dit que la suite $(f_n)_{n \in \mathbb{N}}$ converge simplement (resp. uniformément) sur I ssi il existe $f \in \mathcal{F}(I; E)$ telle que $(f_n)_{n \in \mathbb{N}}$ converge simplement (resp. uniformément) sur I vers f .

Cette définition s'étend immédiatement aux fonctions à variables complexes et à valeurs dans un \mathbb{C} -espace vectoriel.

Proposition A.1.2. La convergence uniforme implique la convergence simple.

Définition A.1.3 (modes de convergence d'une série de fonctions).

Soit $\Sigma(f_n)_{n \in \mathbb{N}}$ une série de fonctions à variable réelle et à valeurs dans un \mathbb{R} -espace vectoriel E de norme $\|\bullet\|_E$, soit f une fonction à variable réelle et à valeurs dans E , et soit I un intervalle non-dégénéré de \mathbb{R} .

On dit que :

- $\Sigma(f_n)_{n \in \mathbb{N}}$ converge *simplement* sur I vers f ssi la suite des sommes partielles de $\Sigma(f_n)_{n \in \mathbb{N}}$ converge simplement sur I vers f :

$$\forall x \in I, \forall \varepsilon > 0, \exists N \in \mathbb{N}, \forall n \geq N, \left\| \left(\sum_{i=0}^n f_i(x) \right) - f(x) \right\|_E \leq \varepsilon$$

- $\Sigma(f_n)_{n \in \mathbb{N}}$ converge *absolument* sur I ssi pour tout $x \in I$, la série numérique $\Sigma(\|f_n(x)\|_E)_{n \in \mathbb{N}}$ converge :

$$\forall x \in I, \exists l \in \mathbb{R}, \forall \varepsilon > 0, \exists N \in \mathbb{N}, \forall n \geq N, \left| \left(\sum_{i=0}^n \|f_i(x)\|_E \right) - l \right| \leq \varepsilon$$

- $\Sigma(f_n)_{n \in \mathbb{N}}$ converge *uniformément* sur I vers f ssi la suite des sommes partielles de $\Sigma(f_n)_{n \in \mathbb{N}}$ converge uniformément sur I vers f :

$$\forall \varepsilon > 0, \exists N \in \mathbb{N}, \forall n \geq N, \forall x \in I, \left\| \left(\sum_{i=0}^n f_i(x) \right) - f(x) \right\|_E \leq \varepsilon$$

- $\Sigma(f_n)_{n \in \mathbb{N}}$ converge *normalement* sur I ssi la série numérique $\Sigma(\|f_n\|_{\infty, E})_{n \in \mathbb{N}}$ converge :

$$\exists l \in \mathbb{R}, \forall \varepsilon > 0, \exists N \in \mathbb{N}, \forall n \geq N, \forall x \in I, \left| \left(\sum_{i=0}^n \|f_i(x)\|_E \right) - l \right| \leq \varepsilon$$

Par extension, on dit que la série $\Sigma(f_n)_{n \in \mathbb{N}}$ converge simplement (resp. uniformément) sur I ssi il existe $f \in \mathcal{F}(I; E)$ telle que $\Sigma(f_n)_{n \in \mathbb{N}}$ converge simplement (resp. uniformément) sur I vers f .

Proposition A.1.4.

- La convergence uniforme d'une série de fonctions implique sa convergence simple.
- La convergence absolue d'une série de fonctions implique sa convergence simple.
- La convergence normale d'une série de fonctions implique ses convergences uniforme, absolue, et simple.

Définition A.1.5 (Série entière). Une série entière est une série de fonctions à variable réelle ou complexe de la forme $\Sigma(z \mapsto a_n z^n)_{n \in \mathbb{N}}$, avec $(a_n)_{n \in \mathbb{N}}$ une série numérique réelle ou complexe. On la note usuellement $\Sigma_n a_n z^n$.

Théorème-Définition A.1.6 (Rayon de convergence d'une série entière). Soit $\Sigma_n a_n z^n$ une série entière. Il existe un unique $R \in \mathbb{R}^+$ tel que :

- $\Sigma_n a_n z^n$ converge normalement sur tout compact inclus dans le disque ouvert $\mathcal{B}(\mathbb{C}; 0; R) = \{z \in \mathbb{C}, |z| < R\}$, et

- la série numérique $\Sigma(a_n z^n)_{n \in \mathbb{N}}$ diverge grossièrement pour tout z hors du disque fermé $\mathcal{B}_f(0; R) = \{z \in \mathbb{C}, |z| \leq R\}$

R est appelé le *rayon de convergence* de la série entière $\Sigma_n a_n z^n$.

Définition A.1.7 (Développabilité en Série Entière). Soient f une fonction réelle ou complexe, et $z_R \in \mathbb{R}$ (ou \mathbb{C}).

On dit que f admet un Développement en Série Entière (DSE) autour de z_R sur un voisinage V ssi il existe $\Sigma_n a_n z^n$ une série entière telle que $\Sigma(z \mapsto a_n (z - z_R)^n)_{n \in \mathbb{N}}$ converge sur ce voisinage V de z_R , vers f .

Par extension, on dit que f admet un DSE autour de z_R , sans plus de précision, si il existe un voisinage V de z_R tel que f admette un DSE autour de z_R sur V .

Proposition A.1.8 (DSE et Développement en Série de Taylor). Soit f une fonction Développable en Série Entière sur un voisinage V ouvert autour de z_R .

Alors f est continûment dérivable sur V , et son DSE $\Sigma_n a_n z^n$ en z_R est unique et s'identifie au Développement en Série de Taylor de f en z_R . Cela s'écrit :

$$\begin{cases} f \in \mathcal{C}^\infty(V; \mathbb{K}) & \text{avec } \mathbb{K} = \mathbb{R} \text{ ou } \mathbb{C} \\ \forall n \in \mathbb{N}, a_n = \frac{f^{(n)}(z_R)}{n!} \end{cases}$$

Proposition A.1.9 (DSE et dérivation). Soit $\Sigma_n a_n z^n$ une série entière, convergente avec un rayon $R \in \bar{\mathbb{R}}^+$ vers une fonction f . Alors :

- f est dérivable sur le disque ouvert de convergence $\mathcal{B}(0; R)$, et
- la série entière $\Sigma(z \mapsto n a_n z^{n-1})_{n \in \mathbb{N}^*}$ (qui s'écrit aussi $\Sigma(z \mapsto (n+1) a_{n+1} z^n)_{n \in \mathbb{N}}$) a R pour rayon de convergence, et converge sur $\mathcal{B}(0; R)$ vers f' .

Définition A.1.10 (Fonction analytique). Une fonction analytique est une fonction réelle ou complexe, qui est développable en série entière autour de chaque point de son domaine de définition.

A.2 Adjonction Hermitienne

Dans cette section, on notera \mathbb{K} pour « \mathbb{R} ou \mathbb{C} ».

Définition A.2.1 (Produit scalaire réel). Soit E un \mathbb{R} -espace vectoriel, et φ une application de E^2 vers \mathbb{R} . On dit que φ est un produit scalaire sur E ssi elle est :

- bilinéaire : $\forall (x; y; z; \lambda) \in E^3 \times \mathbb{R}, \begin{cases} \varphi(\lambda x + z; y) = \lambda \varphi(x; y) + \varphi(z; y) \\ \varphi(x; \lambda y + z) = \lambda \varphi(x; y) + \varphi(x; z) \end{cases}$,
- symétrique : $\forall (x; y) \in E^2, \varphi(y; x) = \varphi(x; y)$,
- positive : $\forall x \in E, \varphi(x; x) \geq 0$,
- et « définie » : $\forall x \in E, (\varphi(x; x) = 0 \implies x = 0_E)$.

Dans ce manuscrit, on notera $\varphi(x; y) = \langle x; y \rangle$ ou $\varphi(x; y) = x \cdot y$.

Définition A.2.2 (Produit scalaire hermitien). Ici, « $\bar{\bullet}$ » dénote le complexe conjugué.

Soit E un \mathbb{C} -espace vectoriel, et φ une application de E^2 vers \mathbb{C} . On dit que φ est un produit scalaire sur E ssi elle est :

- sesquilinéaire à gauche : $\forall (x; y; z; \lambda) \in E^3 \times \mathbb{C}$,

$$\begin{cases} \varphi(\lambda x + z; y) = \bar{\lambda}\varphi(x; y) + \varphi(z; y) \\ \varphi(x; \lambda y + z) = \lambda\varphi(x; y) + \varphi(x; z) \end{cases} ,$$
- symétrique hermitienne : $\forall (x; y) \in E^2, \varphi(y; x) = \overline{\varphi(x; y)}$,
- positive : $\forall x \in E, \varphi(x; x) \in \mathbb{R}^+$,
- et « définie » : $\forall x \in E, (\varphi(x; x) = 0 \implies x = 0_E)$.

On utilisera la même notation $\varphi(x; y) = \langle x; y \rangle$ ou $\varphi(x; y) = x \cdot y$ que dans le cas réel.

Remarques.

- Dans le cas du produit scalaire hermitien, la sesquilinearité à gauche n'est pas une convention universelle, certains auteurs utilisent la sesquilinearité à droite.
- Un \mathbb{R} -espace vectoriel muni d'un produit scalaire est appelé *espace préhilbertien réel*.
- Un \mathbb{C} -espace vectoriel muni d'un produit scalaire est appelé *espace préhilbertien complexe*.
- Lorsqu'un \mathbb{K} -espace vectoriel possède un produit scalaire, celui-ci n'est pas unique. Néanmoins, la plupart des espaces vectoriels dont il est question dans ce manuscrit possèdent un produit scalaire *usuel*.

Théorème-Définition A.2.3. Soit E un \mathbb{K} -espace vectoriel muni d'un produit scalaire $\langle \bullet; \bullet \rangle_E$. L'application $x \mapsto \sqrt{\langle x; x \rangle_E}$ est une norme sur E , appelée *norme issue de $\langle \bullet; \bullet \rangle_E$* .

Définition A.2.4 (Adjoint). Soient E et F deux \mathbb{K} -espaces vectoriels munis des produits scalaires respectifs $\langle \bullet; \bullet \rangle_E$ et $\langle \bullet; \bullet \rangle_F$, et $u \in \mathcal{F}(E; F)$. Soit $v \in \mathcal{F}(F; E)$, on dit que v est adjointe à u ssi :

$$\forall (x; y) \in E \times F, \langle u(x); y \rangle_F = \langle x; v(y) \rangle_E$$

Proposition A.2.5. Soient E et F deux espaces préhilbertiens, ainsi que $u \in \mathcal{F}(E; F)$ et $v \in \mathcal{F}(F; E)$. Si v est adjointe à u alors :

- u est linéaire,
- v est linéaire,
- v est l'unique adjointe de u . On note : $v = u^*$.
- u est (l'unique) adjointe à v . C'est-à-dire que $(u^*)^* = u$.

- Si u est inversible, alors u^* est inversible, u^{-1} possède un adjoint, et $(u^*)^{-1} = (u^{-1})^*$.

Proposition A.2.6. Ici, « $\bar{\bullet}$ » dénote le complexe conjugué.

Soient E et F deux espaces préhilbertiens, $\lambda \in \mathbb{K}$, ainsi que u et $v \in \mathcal{L}(E; F)$ possédant chacune une adjointe.

Alors $\lambda u + v$ possède une adjointe, et : $(\lambda u + v)^* = \bar{\lambda}u^* + v^*$.

Proposition A.2.7. Soient E , F , et G trois espaces préhilbertiens, ainsi que $u \in \mathcal{L}(E; F)$ et $v \in \mathcal{L}(F; G)$ possédant chacune une adjointe.

Alors $v \circ u$ possède une adjointe, et : $(v \circ u)^* = u^* \circ v^*$.

A.3 Fonction d'erreur complémentaire normalisée erfcx

Définition A.3.1 (erfcx). La *fonction d'erreur complémentaire normalisée*, notée erfcx, est définie de \mathbb{R} vers \mathbb{R} par la formule :

$$\forall x \in \mathbb{R}, \text{erfcx}(x) = e^{x^2} \text{erfc}(x) = \frac{2}{\sqrt{\pi}} e^{x^2} \int_x^{+\infty} e^{-t^2} dt$$

où erfc est la fonction d'erreur complémentaire.

Proposition A.3.2.

- erfcx est l'unique solution sur \mathbb{R} du problème différentiel :

$$\begin{cases} \text{erfcx}'(x) = 2x \text{erfcx}(x) - \frac{2}{\sqrt{\pi}} \\ \text{erfcx}(0) = 1 \end{cases}$$
- $\text{erfcx}(0) = 1$, et $\text{erfcx}'(0) = -\frac{2}{\sqrt{\pi}}$.
- $\text{erfcx}(x) \sim_{x \rightarrow +\infty} \frac{1}{\sqrt{\pi}x}$.

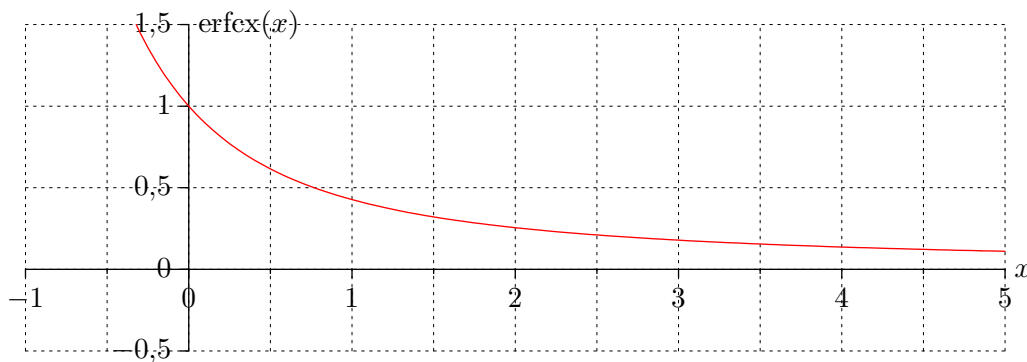


FIGURE A.1 – Une portion du graphe de la fonction erfcx

Annexe B

Lien entre les Algorithmes à Collisions Nulles et les Développements en Série Entière de Variables Indépendantes

Soit le problème différentiel sur \mathbb{R}^+ simplifié :

$$\begin{cases} f'(x) = -k(x)f(x) \\ f(0) = f_0 \end{cases} \quad (\text{B.1})$$

Les ACNs d'une part et les DSEVIs d'autre part offrent chacun une expression intégrale de $f(x)$ avec $x \in \mathbb{R}^+$, évaluable par méthode de Monte-Carlo.

Si on utilise un DSEVI : La solution symbolique au problème (B.1), est bien connue, c'est :

$$f(x) = \exp\left(-\int_0^x dx' k(x')\right) f_0 \quad (\text{B.2})$$

mais elle ne peut pas être évaluée directement par méthode de Monte-Carlo en toute généralité, comme c'est expliqué au chapitre 3.1.2.

On peut en dégager une expression évaluable par méthode de Monte-Carlo, en utilisant le développement en série entière de l'exponentielle (convergeant sur \mathbb{R} tout entier) :

$$f(x) = \sum_{n=0}^{+\infty} \frac{(-1)^n}{n!} \left(\int_0^x dx' k(x')\right)^n f_0 \quad (\text{B.3})$$

Si on utilise un ACN : On introduit une fonction \hat{k} arbitraire mais strictement positive, ensuite la formulation intégrale d'un ACN s'écrit :

$$f(x) = \exp\left(-\int_0^x dx' \hat{k}(x')\right) f_0 + \int_0^x dx' \exp\left(-\int_{x'}^x dx'' \hat{k}(x'')\right) (\hat{k}(x') - k(x')) f(x') \quad (\text{B.4})$$

Cette écriture peut être transformée en une expression proche de ce qu'on obtient avec un DSEVI :

$$f(x) = \exp\left(-\int_0^x dx'' \hat{k}(x'')\right) \left(f_0 + \int_0^x dx' \exp\left(\int_0^{x'} dx'' \hat{k}(x'')\right) (\hat{k}(x') - k(x')) f(x')\right)$$

d'où

$$\begin{aligned} & \exp\left(\int_0^x dx' \hat{k}(x')\right) f(x) \\ &= f_0 + \int_0^x dx' (\hat{k}(x') - k(x')) \exp\left(\int_0^{x'} dx'' \hat{k}(x'')\right) f(x') \\ &= f_0 + \int_0^x dx_1 (\hat{k}(x_1) - k(x_1)) \left(f_0 + \int_0^{x_1} dx_2 (\hat{k}(x_2) - k(x_2)) (f_0 + \dots)\right) \\ &= f_0 \times \left(1 + \sum_{n=1}^{+\infty} \int_0^x dx_1 \int_0^{x_1} dx_2 \cdots \int_0^{x_{n-1}} dx_n \prod_{i=1}^n (\hat{k}(x_i) - k(x_i))\right) \end{aligned}$$

En notant $x_0 = x$:

$$\exp\left(\int_0^x dx' \hat{k}(x')\right) f(x) = \left(1 + \sum_{n=1}^{+\infty} \int_0^{x_0} dx_1 \int_0^{x_1} dx_2 \cdots \int_0^{x_{n-1}} dx_n \prod_{i=1}^n (\hat{k}(x_i) - k(x_i))\right) f_0 \quad (\text{B.5})$$

●●●● Cette dernière expression peut être presque atteinte par un DSE. Il faut d'abord multiplier arbitrairement chaque terme de l'égalité (B.2) par $\exp\left(\int_0^x dx' \hat{k}(x')\right)$, et ensuite le développement en série entière de l'exponentielle nous mène à :

$$\exp\left(\int_0^x dx' \hat{k}(x')\right) f(x) = \sum_{n=0}^{+\infty} \frac{1}{n!} \left(\int_0^x dx' (\hat{k}(x') - k(x'))\right)^n f_0 \quad (\text{B.6})$$

La validité des algorithmes à collisions nulles, la validité du développement en série entière de l'exponentielle, combinées à l'unicité des développements en série entière impliquent l'égalité terme à terme des sommes dans (B.5) et (B.6).

Cela signifie, de façon plus générale, que pour toute fonction réelle k continue par morceaux :

$$\forall x_0 \in \mathbb{R}, \forall n \in \mathbb{N}^*,$$

$$\int_0^{x_0} dx_1 \int_0^{x_1} dx_2 \cdots \int_0^{x_{n-1}} dx_n \prod_{i=1}^n k(x_i) = \frac{1}{n!} \left(\int_0^{x_0} dx' k(x') \right)^n \quad (\text{B.7})$$

Ce résultat peut être montré indépendamment des ACNs et des DSEVIs de plusieurs façons, dont la suivante :

Preuve Identité (B.7). Pour tout $n \in \mathbb{N}^*$ et $x_0 \in \mathbb{R}^+$, on note :

$$\mathcal{T}(n; x_0) = \{ \vec{x} = (x_i)_{i \in \llbracket 1; n \rrbracket} \in [0; x_0]^n, \text{ tq } \forall (i; j) \in \llbracket 1; n \rrbracket^2, i < j \implies x_i > x_j \}$$

$\mathcal{T}(n; x_0)$ est en fait le domaine d'intégration du membre de gauche de l'identité (B.7). C'est un hypertétraèdre. Soit aussi, pour toute permutation $\sigma \in \mathfrak{S}_n$, l'hypertétraèdre :

$$\mathcal{T}(n; x_0; \sigma) = \{ (x_{\sigma(i)})_{i \in \llbracket 1; n \rrbracket}, \text{ avec } (x_i)_{i \in \llbracket 1; n \rrbracket} \in \mathcal{T}(n; x_0) \}$$

On a immédiatement $\mathcal{T}(n; x_0) = \mathcal{T}(n; x_0; id)$

Soit maintenant $\vec{x} = (x_i)_{i \in \llbracket 1; n \rrbracket} \in [0; x_0]^n$. Comme « $>$ » est une relation d'ordre strict totale sur $[0; x_0]$, s'il existe une permutation $\sigma \in \mathfrak{S}_n$ telle que $(x_{\sigma(i)})_{i \in \llbracket 1; n \rrbracket} \in \mathcal{T}(n; x_0)$ — ce qui équivaut à $(x_i)_{i \in \llbracket 1; n \rrbracket} \in \mathcal{T}(n; x_0; \sigma^{-1})$ —, cette permutation σ est unique. Si elle n'existe pas c'est que certains x_i sont égaux, les \vec{x} tels que ceci formant un ensemble de mesure nulle sur $[0; x_0]^n$.

Donc pour des n et x_0 fixés, quand σ parcourt \mathfrak{S}_n (équivalent à « σ^{-1} parcourt \mathfrak{S}_n ») les différents $\mathcal{T}(n; x_0; \sigma)$ forment presque (à une partie de mesure nulle près) une partition de l'hypercube $[0; x_0]^n$. Il en résulte que :

$$\begin{aligned} \left(\int_0^{x_0} dx' k(x') \right)^n &= \sum_{\sigma \in \mathfrak{S}_n} \int_{\vec{x} \in \mathcal{T}(n; x_0; \sigma)} \prod_{i=1}^n k(x_i) dx_i \\ &= \sum_{\sigma \in \mathfrak{S}_n} \int_{\vec{x} \in \mathcal{T}(n; x_0)} \prod_{i=1}^n k(x_{\sigma(i)}) dx_{\sigma(i)} \\ &= \sum_{\sigma \in \mathfrak{S}_n} \int_{\vec{x} \in \mathcal{T}(n; x_0)} \prod_{i=1}^n k(x_i) dx_i \end{aligned}$$

Comme par ailleurs \mathfrak{S}_n contient $n!$ permutations :

$$\left(\int_0^{x_0} dx' k(x') \right)^n = n! \int_{\vec{x} \in \mathcal{T}(n; x_0)} \prod_{i=1}^n k(x_i) dx_i \quad \blacksquare$$

●●●● Apparemment, les ACNs fournissent 2 éléments par rapport à un DSEVI direct de la solution symbolique de (B.1) :

- Si la fonction arbitraire \hat{k} majore k , chacun des termes sous la somme dans (B.5) (ou dans (B.6)) est positif, contrairement aux termes de la somme (B.3).

Comme il faut, lors d'une estimation d'une somme infinie, évaluer seulement un nombre fini de ses termes, leur égalité en signe devrait faciliter la convergence de l'estimateur obtenu.

- Les ACNs fournissent des probabilités analogues intuitives pour choisir n et $(x_i)_{i \in [1;n]}$, dans (B.5) ou (B.6). L'estimateur obtenu de $f(x)$ répond à un formalisme proche de celui décrit en (3.12). En effet, en se plaçant dans le cas le plus général possible, avec $p_{X'}(x')$ quelconque sur $] - \infty; x]$ et P_{nul} quelconque dans $]0; 1[$:

$$\tilde{F}(x) = \frac{\hat{k}(X') \exp\left(-\int_{X'}^x dx'' \hat{k}(x'')\right)}{p_{X'}(X')} \left(H(X' \leq 0) \tilde{F}_0 + H(X' > 0) \frac{H(Col = nul)}{P_{nul}} \times \frac{\hat{k}(X') - \tilde{k}(X')}{\hat{k}(X')} \tilde{F}(X') \right) \quad (B.8)$$

donc

$$\tilde{F}(x_0) = \prod_{i=1}^{I_m} \frac{\hat{k}(X_i)}{p_{X_i}(X_i)} \exp\left(-\int_{X_{I_m}}^{x_0} dx' \hat{k}(x')\right) \times \prod_{i=1}^{I_m-1} \frac{\hat{k}(X_i) - \tilde{k}(X_i)}{\hat{k}(X_i)} \times \frac{H(Col_i = nul)}{P_{nul\ i}} \tilde{F}_0 \quad (B.9)$$

où :

- $\forall i \in \mathbb{N}, X_{i+1}$ à valeurs dans $] - \infty; X_i]$,
- $I_m = \min\{i \in \mathbb{N}, X_i \leq 0\}$,
- les Col_i prennent chacune la valeur nul avec une probabilité $P_{nul\ i}$
- \tilde{F}_0 est un estimateur de f_0 ,
- les $\tilde{k}(X_i)$ sont estimateurs des $k(X_i)$.

Si on choisit les X_i comme des longueurs d'extinction selon \hat{k} , ce qui est conseillé au chapitre 2.2.2, l'expression précédente se réduit à :

$$\tilde{F}(x_0) = \prod_{i=1}^{I_m-1} \frac{\hat{k}(X_i) - \tilde{k}(X_i)}{\hat{k}(X_i)} \times \frac{H(Col_i = nul)}{P_{nul\ i}} \tilde{F}_0 \quad (B.10)$$

Cette dernière expression est totalement soluble dans le formalisme (3.12). Cela donne :

$$\tilde{F}(x_0) = \prod_{i=1}^{N_{nul}} \frac{\hat{k}(\check{X}_i) - \tilde{k}(\check{X}_i)}{\hat{k}(\check{X}_i)} \times \frac{H(Col_i = nul)}{P_{nul\ i}} \tilde{F}_0 \quad (B.11)$$

où

-
- N_{nul} suit une loi de Poisson de paramètre $\int_0^{x_0} dx \hat{k}(x)$,
 - les \check{X}_i sont indépendantes, à valeurs dans $]0; x_0[$ et de densité de probabilité $p_{\check{X}}(x) = \hat{k}(x) / \int_0^{x_0} dx' \hat{k}(x')$.

●●●● On peut pour finir remarquer qu'en utilisant un Algorithme à Collisions Nulles, on effectue *a priori* un DSEVI non centré sur 0. L'expression (B.11) montre en effet un DSEVI non correspondant à l'équation (B.3) mais plutôt à l'équation (B.6), où le développement en série entière est centré sur $\int_0^{x_0} dx' \hat{k}(x')$.

On peut en effet dégager les termes du DSE sous-jacent à (B.11) :

$$\begin{aligned}
f(x_0) &= E(\tilde{F}(x_0)) \\
&= E\left(\prod_{i=1}^{N_{nul}} \frac{\hat{k}(\check{X}_i) - \tilde{k}(\check{X}_i)}{\hat{k}(\check{X}_i)} \times \frac{H(Col_i = nul)}{P_{nul\ i}} \tilde{F}_0\right) \\
&= \sum_{n_{nul}=0}^{+\infty} p_{N_{nul}}(n_{nul}) \prod_{i=1}^{n_{nul}} \int_0^{x_0} p_{\check{X}}(\check{x}_i) d\check{x}_i \frac{\hat{k}(\check{x}_i) - \tilde{k}(\check{x}_i)}{\hat{k}(\check{x}_i)} f_0 \\
&= \sum_{n_{nul}=0}^{+\infty} \frac{\left(\int_0^{x_0} dx' \hat{k}(x')\right)^{n_{nul}}}{n_{nul}!} \exp\left(-\int_0^{x_0} dx' \hat{k}(x')\right) \prod_{i=1}^{n_{nul}} \int_0^{x_0} \frac{\hat{k}(\check{x}_i)}{\int_0^{x_0} dx' \hat{k}(x')} d\check{x}_i \times \\
&\quad \frac{\hat{k}(\check{x}_i) - \tilde{k}(\check{x}_i)}{\hat{k}(\check{x}_i)} f_0 \\
&= \exp\left(-\int_0^{x_0} dx' \hat{k}(x')\right) \sum_{n_{nul}=0}^{+\infty} \frac{1}{n_{nul}!} \prod_{i=1}^{n_{nul}} \int_0^{x_0} (\hat{k}(\check{x}_i) - \tilde{k}(\check{x}_i)) d\check{x}_i f_0 \\
&= \exp\left(-\int_0^{x_0} dx' \hat{k}(x')\right) \sum_{n_{nul}=0}^{+\infty} \frac{1}{n_{nul}!} \left(\int_0^{x_0} (\hat{k}(\check{x}) - \tilde{k}(\check{x})) d\check{x}\right)^{n_{nul}} f_0
\end{aligned}$$

ce qui est exactement l'équation (B.6).

●●●● Le propos de cette annexe peut être retrouvé au moins partiellement dans [68, 78–80, 114].

Annexe C

Éléments pour le calcul d'un polynôme de l'espérance d'une variable aléatoire

C.1 Calcul en ligne de la formulation (3.9)

Problème mathématique posé : On dispose d'une variable aléatoire Z , et on veut estimer $h(E(Z))$, où h est une fonction polynomiale. On note $h : z \mapsto \sum_{i=0}^k a_i z^i$ où les a_i sont des constantes réelles. Si $k' \in \mathbb{N}$ est tel que $k' \geq k$, on peut se servir de la formule (3.9) page 133 :

$$\sum_{i=0}^k a_i E(Z)^i = E \left(\frac{1}{k'!} \sum_{\sigma \in \mathfrak{S}_{k'}} \sum_{i=0}^k a_i \prod_{j=1}^i Z_{\sigma(j)} \right) \quad \text{avec les } (Z_i)_{i \in \mathbb{N}} \text{ parentes à } Z$$

pour obtenir une expression aléatoire $W_{h,Z}$ dérivée de Z dont $h(E(Z))$ soit l'espérance. On peut alors échantillonner $W_{h,Z}$ pour estimer $h(E(Z))$, dans une procédure Monte-Carlo tout-à-fait standard. En l'occurrence, l'expression $W_{h,Z}$ qui apparaît dans la formule (3.9) est

$$W_{h,Z} = \frac{1}{k'!} \sum_{\sigma \in \mathfrak{S}_{k'}} \sum_{i=0}^k a_i \prod_{j=1}^i Z_{\sigma(j)} \quad (\text{C.1})$$

Lorsqu'on dispose d'un échantillon $(z_{i,n})_{i \in [1;k'], n \in [1;N]}$ de Z , on pose ainsi pour n allant de 1 à N les

$$w_{h,Z} \ n = \frac{1}{k'!} \sum_{\sigma \in \mathfrak{S}_{k'}} \sum_{i=0}^k a_i \prod_{j=1}^i z_{\sigma(j),n} \quad (\text{C.2})$$

qui forment un échantillon de $W_{h,Z}$. Tous les $z_{i,n}$ doivent avoir été tirés indépendamment.

On propose ici une technique de calcul pour $w_{h,Z} \ n$, à partir des $(z_{i,n})_{i \in [1;k'], n}$, selon la formule précédente (C.2). L'ensemble $\mathfrak{S}_{k'}$ contenant $k'!$ permutations, ce

calcul peut être extrêmement fastidieux si mené dans un ordre peu efficace. La complexité de la méthode proposée ici est en $O(k'^2)$, ce qui est suffisamment faible pour les besoins exhibés dans ce manuscrit.

••••• k et k' étant indépendants car fixés, on peut ré-exprimer

$$w_{h,Z} = \sum_{i=0}^k \frac{a_i}{k'!} \sum_{\sigma \in \mathfrak{S}_{k'}} \prod_{j=1}^i z_{\sigma(j),n} \quad (\text{C.3})$$

La difficulté du calcul se reporte essentiellement sur le terme « $\sum_{\sigma \in \mathfrak{S}_{k'}} \prod_{j=1}^i z_{\sigma(j),n}$ ».

Posons la suite à 2 indices $(\tilde{\theta}_{k',i})_{k' \in \mathbb{N}, i \in \mathbb{N}}$ définie par récurrence :

$$\begin{cases} \tilde{\theta}_{k',i} = 1 & \text{si } i = 0 \\ \tilde{\theta}_{k',i} = 0 & \text{si } k' < i \\ \tilde{\theta}_{k'+1,i+1} = \tilde{\theta}_{k',i+1} + z_{k',n} \tilde{\theta}_{k',i} & \text{sinon} \end{cases} \quad (\text{C.4})$$

On peut obtenir assez aisément que :

— $\forall i \in \mathbb{N} :$

$$\tilde{\theta}_{i,i} = \prod_{j=1}^i z_{j,n}$$

d'où

$$\tilde{\theta}_{i,i} = \frac{1}{i!} \sum_{\sigma \in \mathfrak{S}_i} \prod_{j=1}^i z_{\sigma(j),n}$$

— $\forall (k', i) \in \mathbb{N}^2$ tq $k' \geq i$, $\tilde{\theta}_{k',i}$ est la somme de tous les produits $z_a z_b \cdots$ de i échantillons ponctuels de Z (échantillons d'indices 2 à 2 différents), avec les indices $\{a; b; \dots\} \subset \llbracket 1; k' \rrbracket$. Ces produits sont au nombre de $\binom{k'}{i}$, car l'ordre de leurs termes ne compte pas : en effet, dans $\tilde{\theta}_{k',i}$ chaque produit n'est calculé que dans un seul ordre.

Par ailleurs, dans le terme $\sum_{\sigma \in \mathfrak{S}_{k'}} \prod_{j=1}^i z_{\sigma(j),n}$, on a $k'!$ produits de ce genre (car $\mathfrak{S}_{k'}$ contient $k'!$ éléments), en comptant les répétitions qui auront lieu. Le nombre de répétitions est le même pour tous les produits.

Ce qui amène à :

$$\forall (k', i) \in \mathbb{N}^2 \text{ tq } k' \geq i, \quad \frac{1}{k'!} \sum_{\sigma \in \mathfrak{S}_{k'}} \prod_{j=1}^i z_{\sigma(j),n} = \frac{1}{\binom{k'}{i}} \tilde{\theta}_{k',i} \quad (\text{C.5})$$

•••• Les $\tilde{\theta}_{k',i}$ peuvent être calculés au fur et à mesure que l'on échantillonne Z , par l'algorithme suivant :

Résultat : Un échantillon ponctuel $w_{h,Z\ n}$ de $W_{h,Z}$

```

1  $\tilde{\theta}_{0,0} \leftarrow 1$ ;
2 pour  $\ell \leftarrow 1$  à  $k'$  faire                                     // Il faut que  $k' \geq \deg h$ .
3   échantillonner  $Z$  ; on obtient  $z_{\ell,n}$ ;
4    $\tilde{\theta}_{\ell,0} \leftarrow 1$ ;
5    $\tilde{\theta}_{\ell,\ell} \leftarrow z_{\ell,n} \tilde{\theta}_{\ell-1,\ell-1}$ ;
6   pour  $i \leftarrow 1$  à  $\min\{\ell - 1; \deg h\}$  faire  $\tilde{\theta}_{\ell,i} \leftarrow \tilde{\theta}_{\ell-1,i} + z_{\ell,n} \tilde{\theta}_{\ell-1,i-1}$ ;
7  $w_{h,Z\ n} \leftarrow 0$ ;
8 pour  $i \leftarrow 0$  à  $\deg h$  faire  $w_{h,Z\ n} \leftarrow w_{h,Z\ n} + a_i \tilde{\theta}_{k',i} / \binom{k'}{i}$ ;
```

Algorithme C.1 : Algorithme pour l'échantillonnage de $W_{h,Z}$, variable aléatoire d'espérance $h(E(Z))$ où h est polynômiale de coefficients a_i

C.2 Intérêt d'estimer un polynôme de l'espérance d'une V.A. grâce à un échantillon contenant plus de tirages de la V.A. que le degré du polynôme : exemple du calcul de la variance

•••• On considère ici une variable aléatoire Z à valeurs réelles, dont les moments $E(Z^i)$ sont finis au moins jusqu'à l'ordre $i = 4$.

Le problème de l'estimation d'un polynôme de l'espérance de Z trouve certainement sa plus simple expression dans l'estimation du monôme $E(Z)^2$. Ce problème est lié à l'estimation de la variance de Z , en effet $\text{Var}(Z) = E(Z^2) - E(Z)^2$ — l'estimation de $E(Z^2)$ ne posant aucune difficulté particulière.

•••• En reprenant la méthodologie exposée au chapitre 3.2.1, on obtient qu'un estimateur de $\text{Var}(Z)$ doit utiliser au moins 2 estimations de Z . Si on utilise en effet 2 estimations de Z , l'estimateur le plus efficace de $\text{Var}(Z)$ est ainsi :

$$\widetilde{\text{Var}(Z)}_P = \frac{1}{2!} \sum_{\sigma \in \mathfrak{S}_2} Z_{\sigma(1)}^2 - Z_{\sigma(1)} Z_{\sigma(2)} \quad (\text{C.6})$$

$$= \frac{1}{2} (Z_1^2 + Z_2^2) - Z_1 Z_2 \quad (\text{C.7})$$

Si on dispose de $n \geq 2$ tirages de Z , on peut les combiner pour former une seule estimation de $\text{Var}(Z)$. En utilisant les techniques les plus abouties présentées au

chapitre 3.2.1, on retrouve la formule de la variance empirique échantillon :

$$\widetilde{\text{Var}}(Z)_{MVU} = \frac{1}{n!} \sum_{\sigma \in \mathfrak{S}_n} Z_{\sigma(1)}^2 - Z_{\sigma(1)} Z_{\sigma(2)} \quad (\text{C.8})$$

$$= \frac{1}{n} \sum_{i=1}^n Z_i^2 - \frac{1}{n(n-1)} \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n Z_i Z_j \quad (\text{C.9})$$

$$= \frac{1}{n-1} \left(\sum_{i=1}^n Z_i^2 - \frac{1}{n} \left(\sum_{i=1}^n Z_i \right)^2 \right) \quad (\text{C.10})$$

Cet estimateur est bien l'estimateur *Minimal Variance and Unbiased* de $\text{Var}(Z)$ (voir par exemple [38, 54]).

●●●● Reprenons la formule (C.7). Pour estimer $\text{Var}(Z)$, il est tentant d'effectuer $2N$ tirages de Z puis de calculer l'estimateur :

$$\widetilde{\text{Var}}(Z)_{MP} = \frac{1}{N} \sum_{i=1}^N \frac{1}{2} (Z_{2i-1}^2 + Z_{2i}^2) - Z_{2i-1} Z_{2i} \quad (\text{C.11})$$

Mais quelle est sa variance ?

En considérant l'estimateur de $\text{Var}(Z)$ donné en C.10 avec n tirages de Z , on peut rappeler une des formule de la proposition 1.3.9 page 56 :

$$\text{Var}(\widetilde{\text{Var}}(Z)_{MVU}) = \frac{\text{Var}(Z)^2}{n} \left(\text{Kurt}(Z) - \frac{n-3}{n-1} \right)$$

En prenant $n = 2$:

$$\text{Var}(\widetilde{\text{Var}}(Z)_P) = \frac{\text{Var}(Z)^2}{2} (\text{Kurt}(Z) + 1) \quad (\text{C.12})$$

d'où, en effectuant $2N$ tirages :

$$\text{Var}(\widetilde{\text{Var}}(Z)_{MP}) = \frac{\text{Var}(Z)^2}{2N} (\text{Kurt}(Z) + 1) \quad (\text{C.13})$$

car les tirages sont censés être indépendants.

Par ailleurs, en effectuant aussi $2N$ tirages :

$$\text{Var}(\widetilde{\text{Var}}(Z)_{MVU}) = \frac{\text{Var}(Z)^2}{2N} \left(\text{Kurt}(Z) - \frac{2N-3}{2N-1} \right) \quad (\text{C.14})$$

Sans surprise, on constate que $\widetilde{\text{Var}}(Z)_{MP}$ est un estimateur de $\text{Var}(Z)$ moins efficace que $\widetilde{\text{Var}}(Z)_{MVU}$. Plus en détails, on a :

$$\frac{\text{Var}(\widetilde{\text{Var}}(Z)_{MP})}{\text{Var}(\widetilde{\text{Var}}(Z)_{MVU})} = \frac{2N(\text{Kurt}(Z) + 1) - \text{Kurt}(Z) - 1}{2N(\text{Kurt}(Z) - 1) - \text{Kurt}(Z) + 3} \quad (\text{C.15})$$

Par exemple :

- avec une loi de pile ou face équilibrée ($\text{Kurt}(Z) = 1$) : $\frac{\text{Var} \dots}{\text{Var} \dots} = 2N - 1$.
- avec une loi normale ($\text{Kurt}(Z) = 3$) : $\frac{\text{Var} \dots}{\text{Var} \dots} = 2 - \frac{1}{N}$.

Annexe D

Codes sources utilisés pour certains des calculs présentés

Cette annexe liste les codes sources utilisés pour plusieurs des calculs présentés au cours de ce manuscrit.

●●●● Tous ces codes sources sont prévus pour être compilés conjointement avec un environnement de développement nommé MCM (pour *Monte-Carlo Method*, en anglais). Cet environnement est un code source partiel, en C++, où l'utilisateur doit coder dans un emplacement prévu l'algorithme de Monte-Carlo qu'il veut implémenter.

Le principe en est le même que celui d'EDstar, présenté dans les perspectives page 253, dont MCM est d'ailleurs une des composantes¹. MCM gère plusieurs tâches utiles ou indispensables dans tous les calculs de type Monte-Carlo, de sorte que l'utilisateur n'ait pas à les ré-implémenter à chaque fois :

- le maintien de la somme et de la somme des carrés des réalisations, puis le calcul final de la moyenne, de la variance d'échantillon, ou encore des coefficients de corrélations entre différentes variables,
- la parallélisation,
- la fourniture des tirages uniformes standards, avec la garantie de l'indépendance des tirages, en particulier entre processus parallèles.

●●●● L'utilisateur doit détailler l'implémentation de la fonction `Mon_algorithme::calculUnique`, dont le prototype est : `void Mon_algorithme::calculUnique ()` ; (pas de valeurs d'entrée, pas de valeur de retour). Comme son nom l'indique, dans cette fonction doivent être explicités les calculs à réaliser pour obtenir une estimation ponctuelle. Cette estimation est l'échantillonnage ponctuel d'un estimateur à valeurs vectorielles, dont les composantes sont dans \mathbb{R} .

1. C'est un petit peu plus compliqué... La version de MCM utilisée dans ce travail de thèse est une version de travail customisée et personnelle, qui n'est pas celle sur laquelle a été construit EDstar. Cette dernière version est disponible sur le Web, contrairement à ma version personnelle.

L'interface disponible dans la fonction `Mon_algorithme::calculUnique`, avec le reste de MCM, est la suivante :

- **double** `Mon_algorithme::mcmRng ()`;
fournit des tirages uniformes standards.
- **void** `Mon_algorithme::setCoordonneeRes (unsigned long i, double w)`;
permet de placer la i^e coordonnée du vecteur résultat à la valeur w .
- **unsigned long** `Mon_algorithme::indiceCoordonneeRes (unsigned long indGrd, unsigned long indParam = 0) const`;
cette fonction fournit la valeur de i , à utiliser dans la fonction `setCoordonneeRes`. Son utilité est de simplifier la prise en charge du calcul de sensibilités paramétriques.
- **void** `Mon_algorithme::enregistrerRealisation ()`;
signale que l'on va pouvoir passer à la réalisation suivante, et donc que les sommes doivent être mises à jour.

D.1 Chambre de combustion contenant une flamme axisymétrique (problème académique de la section 2.4) : calcul du terme source radiatif, avec gestion des hétérogénéités des propriétés optiques par un Algorithme à Collisions Nulles

Le problème physique à l'étude est celui du chapitre 2.4. On cherche à calculer le terme source radiatif S_r en un point \vec{r}_0 , à l'intérieur d'une configuration stationnaire monochromatique partiellement décrite au chapitre 2.4.2. Ce terme source s'écrit :

$$S_r(\vec{r}_0) = \int_{4\pi} d\vec{u} k_a(\vec{r}_0) \delta f(\vec{r}_0; \vec{u})$$

où

$$\delta f(\vec{r}_0; \vec{u}) = f(\vec{r}_0; \vec{u}) - f_{eq}(\vec{r}_0)$$

On estime $\delta f(\vec{r}_0; \vec{u})$ grâce à un algorithme à collisions nulles, détaillé au chapitre 2.4.3.

Plusieurs paramètres, d'une part de la situation physique, d'autre part liés au fonctionnement de l'algorithme, sont variables selon les simulations, et doivent être passés lors de la compilation sous forme de macros. Ces paramètres sont :

- le point de départ \vec{r}_0 , devant être passé au travers de la macro `DEPART`,
- $k_{a\ max}$, devant être passé au travers de la macro `KAMAX`,
- $k_{d\ max}$, devant être passé au travers de la macro `KDMAX`,

- l'émissivité des parois ε , devant être passé au travers de la macro `EMISSIVITE`,
- ϱ , devant être passé au travers de la macro `SURESTIMATION`,
- ϖ , devant être passé au travers de la macro `SEUIL_ROULETTE`.

Les fichiers suivants sont prévus pour être compilés avec l'environnement MCM, expliqué à la page 277; d'où la ligne `#include "mcmGT2.h"` présente en tête du fichier contenant l'algorithme.

Fichier algorithme :

```
1 #include "mcmGT2.h"
2 #include "geom.h"
3 #include <math.h>
4 #define KTAPR ((double) (SURESTIMATION * (KAMAX + KDMAX)))
5
6 InfosMontageAlgo Mon_algorithme::giveDims () {
7
8     InfosMontageAlgo infosMontage;
9
10    infosMontage.nb_Grandeurs_RDR = 2;
11    infosMontage.nb_Parametres_RDR = 0;
12    infosMontage.pile_active = 0;
13    infosMontage.nb_Coordonnees_PDR = 2;
14    infosMontage.valeurDeBaseResume = 0.;
15    infosMontage.valeurDeBasePile = 0.;
16
17    return (infosMontage);
18 }
19
20 inline double kA (const Point & r)
21 {
22
23    return (((double) KAMAX) * .5 * (1. - r.x) * (1. - sqrt
24        (.5 * (r.y * r.y + r.z * r.z))));
25 }
26
27 inline double kD (const Point & r)
28 {
29
30    return (((double) KDMAX) * .5 * (1. - r.x) * (1. - sqrt
31        (.5 * (r.y * r.y + r.z * r.z))));
32 }
33
34 inline double fEq (const Point & r)
35 {
```



```
35     return (.5 * (1. - r.x) * (1. - sqrt (.5 * (r.y * r.y + r.
        z * r.z))));
36 }
37
38 void Mon_algorithme::calculUnique () // FONCTION ALGORITHME
39 {
40     double f0, pondDirInit, tpr, poidsCumuleChemin;
41     long nbAppels;
42     Point depart DEPART, arrivee;
43     Vector dir, normale;
44     BBox cube (Point (-1.,-1.,-1.), Point (1.,1.,1.));
45     double distExtinction, distMur, margeSlab;
46     bool murTouche, roulette;
47     int i, iChoc;
48     double ka, kd, feq, kn, Pdiff, Pnul, R;
49
50     nbAppels = 0;
51     poidsCumuleChemin = 0.;
52
53     dir = UniformSampleSphere (mcmRng (), mcmRng ()); //
        tirage de la direction de depart
54     pondDirInit = 4. * M_PI;
55     f0 = fEq (depart);
56
57     for (tpr = 1., roulette = false;; depart = arrivee) { //
        boucle de calcul
58
59         // calcul du point d'arret
60         // calcul d'un point d'arret selon l'extinction par le
            milieu semi-transparent
61         distExtinction = - log (mcmRng ()) / KTAPR;
62         arrivee = depart + distExtinction * dir;
63
64         // calcul d'une eventuelle intersection avec la bordure
            de la geometrie
65         murTouche = ! cube.Inside (arrivee);
66         if (murTouche) {
67             for (i = 0, distMur = INFINITY; i < 3; i++) {
68                 if (dir [i] == 0.) continue;
69                 margeSlab = ((dir [i] >= 0. ? cube.pMax [i] : cube.
                    pMin [i]) - depart [i]) / dir [i];
70                 if (margeSlab < distMur) {
71                     distMur = margeSlab;
72                     iChoc = i;
73                 }
74             }
75             for (i = 0; i < 3; i++) {
```

```
76     if (i == iChoc) {
77         if (dir [i] >= 0.) {
78             arrivee [i] = cube.pMax [i];
79             normale [i] = -1.;
80         }
81         else {
82             arrivee [i] = cube.pMin [i];
83             normale [i] = 1.;
84         }
85     }
86     else {
87         arrivee [i] = Clamp (depart [i] + distMur * dir [i
88             ], cube.pMin [i], cube.pMax[i]);
89         normale [i] = 0.;
90     }
91 }
92
93 if (tpr <= SEUIL_ROULETTE) roulette = true;
94
95 // calcul partiel du poids au point d'arret
96 if (murTouche) { // si le photon a atteint une paroi
97     if (roulette) {
98         if (mcmRng () <= EMISSIVITE) {
99             poidsCumuleChemin += tpr * - f0;
100             break;
101         }
102         else dir = CosineSampleHemisphere (mcmRng (), mcmRng
103             (), normale);
104     }
105     else {
106         poidsCumuleChemin += tpr * EMISSIVITE * - f0;
107         tpr *= (1. - EMISSIVITE);
108         dir = CosineSampleHemisphere (mcmRng (), mcmRng (),
109             normale);
110     }
111 }
112 else { // si le photon a collisionne dans le milieu
113     ka = kA (arrivee);
114     kd = kD (arrivee);
115     feq = fEq (arrivee);
116     kn = KTAPR - ka - kd;
117     R = mcmRng ();
118     nbAppels++;
119
120     if (roulette) { // utilisation d'une roulette russe
121         totale
```

```
119         if (kn >= 0.) {
120             if (R * KTAPR < ka) {
121                 poidsCumuleChemin += tpr * (feq - f0);
122                 break;
123             }
124             else if (R * KTAPR < kd + ka) dir = SampleHG (
                 mcmRng (), mcmRng (), ASYM, dir);
125         }
126         else if (-2. * kn <= ka) {
127             if (R * KTAPR < ka + 2. * kn) {
128                 poidsCumuleChemin += tpr * (feq - f0);
129                 break;
130             }
131             else if (R * KTAPR < ka + 2. * kn + kd) dir =
                 SampleHG (mcmRng (), mcmRng (), ASYM, dir);
132             else {
133                 poidsCumuleChemin += 2. * tpr * (feq - f0);
134                 tpr = - tpr;
135             }
136         }
137         else {
138             if (Pnul = (ka * ka - 4. * kd * kn) / (2. * (kd -
                 kn) * (2. * kd + ka)), 1. - R < Pnul) {
139                 poidsCumuleChemin += tpr * (1. - kn / (KTAPR *
                 Pnul)) * (feq - f0);
140                 tpr *= kn / (KTAPR * Pnul);
141             }
142             else if (Pdiff = - kd * Pnul / kn, 1. - R < Pnul +
                 Pdiff) {
143                 poidsCumuleChemin += tpr * (1. - kd / (KTAPR *
                 Pdiff)) * (feq - f0);
144                 tpr *= kd / (KTAPR * Pdiff);
145                 dir = SampleHG (mcmRng (), mcmRng (), ASYM, dir)
                 ;
146             }
147             else {
148                 poidsCumuleChemin += tpr * (feq - f0);
149                 break;
150             }
151         }
152     }
153     else { // utilisation d'un partitionnement de l'
             energie
154         if (kn >= 0.) {
155             if ((kd + kn) * R < kn) {
156                 poidsCumuleChemin += tpr * ka / KTAPR * (feq -
                 f0);
```

```
157         tpr *= (kd + kn) / KTAPR;
158     }
159     else {
160         poidsCumuleChemin += tpr * ka / KTAPR * (feq -
161             f0);
162         tpr *= (kd + kn) / KTAPR;
163         dir = SampleHG (mcmRng (), mcmRng (), ASYM, dir)
164             ;
165     }
166 }
167 else {
168     if ((kd - kn) * R < - kn) {
169         poidsCumuleChemin += tpr * ka / KTAPR * (feq -
170             f0);
171         tpr *= - (kd - kn) / KTAPR;
172     }
173     else {
174         poidsCumuleChemin += tpr * ka / KTAPR * (feq -
175             f0);
176         tpr *= (kd - kn) / KTAPR;
177         dir = SampleHG (mcmRng (), mcmRng (), ASYM, dir)
178             ;
179     }
180 }
181 }
182 }
183 if (tpr == 0.) break; // Normalement inutile, mais peut
184     servir si (EMISSIVITE == 1).
185 // Normalement inutile, car quand tpr tombe
186     a 0 on sort immédiatement de la boucle,
187     telle qu'elle est écrite ici.
188 }
```

Fichier geom.h :

```
1  /*****/
2  /* Repris depuis le code source de PBRT */
3  /*****/
```

```
4
5  /*
6      pbrt source code Copyright(c) 1998-2010 Matt Pharr and
          Greg Humphreys.
7      pbrt is free software; you can redistribute it and/or
          modify it under the terms of the GNU General Public
          License as published by the Free Software Foundation;
          either version 2 of the License, or (at your option)
          any later version. Note that the text contents of
          the book "Physically Based Rendering" are *not*
          licensed under the GNU GPL.
8      pbrt is distributed in the hope that it will be useful,
          but WITHOUT ANY WARRANTY; without even the implied
          warranty of MERCHANTABILITY or FITNESS FOR A
          PARTICULAR PURPOSE. See the GNU General Public
          License for more details.
9      You should have received a copy of the GNU General
          Public License along with this program. If not, see <
          http://www.gnu.org/licenses/>.
10  */
11
12
13  #ifndef HMGEOM_H
14  #define HMGEOM_H
15
16  #include <math.h>
17  #include <algorithm>
18  using std::min;
19  using std::max;
20
21  inline double Clamp(double val, double low, double high) {
22      if (val < low) return low;
23      else if (val > high) return high;
24      else return val;
25  }
26
27  class Vector {
28  public:
29      // Vector Public Methods
30      Vector() { x = y = z = 0.f; }
31      Vector(double xx, double yy, double zz)
32          : x(xx), y(yy), z(zz) { }
33      Vector operator+(const Vector &v) const {
34          return Vector(x + v.x, y + v.y, z + v.z);
35      }
36
37      Vector& operator+=(const Vector &v) {
```

```
38         x += v.x; y += v.y; z += v.z;
39         return *this;
40     }
41     Vector operator-(const Vector &v) const {
42         return Vector(x - v.x, y - v.y, z - v.z);
43     }
44
45     Vector& operator--(const Vector &v) {
46         x -= v.x; y -= v.y; z -= v.z;
47         return *this;
48     }
49     Vector operator*(double f) const { return Vector(f*x, f*
50         y, f*z); }
51
52     Vector &operator*=(double f) {
53         x *= f; y *= f; z *= f;
54         return *this;
55     }
56     Vector operator/(double f) const {
57         double inv = 1. / f;
58         return Vector(x * inv, y * inv, z * inv);
59     }
60
61     Vector &operator/=(double f) {
62         double inv = 1. / f;
63         x *= inv; y *= inv; z *= inv;
64         return *this;
65     }
66     Vector operator-() const { return Vector(-x, -y, -z); }
67     double operator[](int i) const {
68         return (&x)[i];
69     }
70
71     double &operator[](int i) {
72         return (&x)[i];
73     }
74     double LengthSquared() const { return x*x + y*y + z*z; }
75     double Length() const { return sqrt (LengthSquared()); }
76
77     bool operator==(const Vector &v) const {
78         return x == v.x && y == v.y && z == v.z;
79     }
80     bool operator!=(const Vector &v) const {
81         return x != v.x || y != v.y || z != v.z;
82     }
83     // Vector Public Data
```

```
84     double x, y, z;
85 };
86
87 inline Vector operator*(float f, const Vector &v) { return v
    *f; }
88 inline Vector Normalize(const Vector &v) { return v / v.
    Length(); }
89 inline double Dot(const Vector &v1, const Vector &v2) {
    return v1.x * v2.x + v1.y * v2.y + v1.z * v2.z; }
90 inline Vector Cross(const Vector &v1, const Vector &v2) {
91     double v1x = v1.x, v1y = v1.y, v1z = v1.z;
92     double v2x = v2.x, v2y = v2.y, v2z = v2.z;
93     return Vector((v1y * v2z) - (v1z * v2y), (v1z * v2x) - (
        v1x * v2z), (v1x * v2y) - (v1y * v2x));
94 }
95
96 class Point {
97 public:
98     // Point Public Methods
99     Point() { x = y = z = 0.f; }
100    Point(double xx, double yy, double zz)
101        : x(xx), y(yy), z(zz) {}
102    Point operator+(const Vector &v) const {
103        return Point(x + v.x, y + v.y, z + v.z);
104    }
105
106    Point &operator+=(const Vector &v) {
107        x += v.x; y += v.y; z += v.z;
108        return *this;
109    }
110    Vector operator-(const Point &p) const {
111        return Vector(x - p.x, y - p.y, z - p.z);
112    }
113
114    Point operator-(const Vector &v) const {
115        return Point(x - v.x, y - v.y, z - v.z);
116    }
117
118    Point &operator-=(const Vector &v) {
119        x -= v.x; y -= v.y; z -= v.z;
120        return *this;
121    }
122    Point &operator+=(const Point &p) {
123        x += p.x; y += p.y; z += p.z;
124        return *this;
125    }
126    Point operator+(const Point &p) const {
```

```
127         return Point(x + p.x, y + p.y, z + p.z);
128     }
129     Point operator* (double f) const {
130         return Point(f*x, f*y, f*z);
131     }
132     Point &operator*=(double f) {
133         x *= f; y *= f; z *= f;
134         return *this;
135     }
136     Point operator/ (double f) const {
137         double inv = 1./f;
138         return Point(inv*x, inv*y, inv*z);
139     }
140     Point &operator/=(double f) {
141         double inv = 1./f;
142         x *= inv; y *= inv; z *= inv;
143         return *this;
144     }
145     float operator[](int i) const {
146         return (&x)[i];
147     }
148
149     float &operator[](int i) {
150         return (&x)[i];
151     }
152
153     bool operator==(const Point &p) const {
154         return x == p.x && y == p.y && z == p.z;
155     }
156     bool operator!=(const Point &p) const {
157         return x != p.x || y != p.y || z != p.z;
158     }
159
160     // Point Public Data
161     float x, y, z;
162 };
163
164 class BBox {
165 public:
166     // BBox Public Methods
167     BBox() {
168         pMin = Point( INFINITY,  INFINITY,  INFINITY);
169         pMax = Point(-INFINITY, -INFINITY, -INFINITY);
170     }
171     BBox(const Point &p) : pMin(p), pMax(p) { }
172     BBox(const Point &p1, const Point &p2) {
```



```
173         pMin = Point(min(p1.x, p2.x), min(p1.y, p2.y), min(
174             p1.z, p2.z));
175         pMax = Point(max(p1.x, p2.x), max(p1.y, p2.y), max(
176             p1.z, p2.z));
177     }
178     bool Overlaps(const BBox &b) const {
179         bool x = (pMax.x >= b.pMin.x) && (pMin.x <= b.pMax.x
180             );
181         bool y = (pMax.y >= b.pMin.y) && (pMin.y <= b.pMax.y
182             );
183         bool z = (pMax.z >= b.pMin.z) && (pMin.z <= b.pMax.z
184             );
185         return (x && y && z);
186     }
187     bool Inside(const Point &pt) const {
188         return (pt.x >= pMin.x && pt.x <= pMax.x &&
189             pt.y >= pMin.y && pt.y <= pMax.y &&
190             pt.z >= pMin.z && pt.z <= pMax.z);
191     }
192     void Expand(double delta) {
193         pMin -= Vector(delta, delta, delta);
194         pMax += Vector(delta, delta, delta);
195     }
196     double SurfaceArea() const {
197         Vector d = pMax - pMin;
198         return 2. * (d.x * d.y + d.x * d.z + d.y * d.z);
199     }
200     double Volume() const {
201         Vector d = pMax - pMin;
202         return d.x * d.y * d.z;
203     }
204     int MaximumExtent() const {
205         Vector diag = pMax - pMin;
206         if (diag.x > diag.y && diag.x > diag.z)
207             return 0;
208         else if (diag.y > diag.z)
209             return 1;
210         else
211             return 2;
212     }
213     const Point &operator[](int i) const;
214     Point &operator[](int i);
215     Vector Offset(const Point &p) const {
216         return Vector((p.x - pMin.x) / (pMax.x - pMin.x),
217             (p.y - pMin.y) / (pMax.y - pMin.y),
218             (p.z - pMin.z) / (pMax.z - pMin.z));
219     }
```

```
215     bool operator==(const BBox &b) const {
216         return b.pMin == pMin && b.pMax == pMax;
217     }
218     bool operator!=(const BBox &b) const {
219         return b.pMin != pMin || b.pMax != pMax;
220     }
221
222     // BBox Public Data
223     Point pMin, pMax;
224 };
225
226 void CoordinateSystem (const Vector &v1, Vector *v2, Vector
    *v3);
227 Vector CosineSampleHemisphere (double u1, double u2, const
    Vector & normale);
228 Vector UniformSampleSphere (double u1, double u2);
229 Vector SampleHG (double r1, double r2, double g, const
    Vector &v1);
230
231 #endif // HMGEOM_H
```

Fichier geom.cpp :

```
1 #include "geom.h"
2
3 /*****
4  /* Partiellement repris depuis le code source de PBRT */
5  *****/
6
7 /*
8     pbrt source code Copyright(c) 1998-2010 Matt Pharr and
        Greg Humphreys.
9     pbrt is free software; you can redistribute it and/or
        modify it under the terms of the GNU General Public
        License as published by the Free Software Foundation;
        either version 2 of the License, or (at your option)
        any later version. Note that the text contents of
        the book "Physically Based Rendering" are *not*
        licensed under the GNU GPL.
10    pbrt is distributed in the hope that it will be useful,
        but WITHOUT ANY WARRANTY; without even the implied
        warranty of MERCHANTABILITY or FITNESS FOR A
        PARTICULAR PURPOSE. See the GNU General Public
        License for more details.
11    You should have received a copy of the GNU General
        Public License along with this program. If not, see <
        http://www.gnu.org/licenses/>.
```

```
12 */
13
14 void CoordinateSystem (const Vector &v1, Vector *v2, Vector
    *v3)
15 {
16     if (fabs (v1.x) > fabs (v1.y)) {
17         double invLen = 1. / sqrt (v1.x * v1.x + v1.z * v1.z);
18         *v2 = Vector (-v1.z * invLen, 0., v1.x * invLen);
19     }
20     else {
21         double invLen = 1. / sqrt (v1.y * v1.y + v1.z * v1.z);
22         *v2 = Vector (0., v1.z * invLen, -v1.y * invLen);
23     }
24     *v3 = Cross(v1, *v2);
25 }
26
27 Vector CosineSampleHemisphere (double u1, double u2, const
    Vector & normale)
28 {
29     double r, theta, sx, sy;
30
31     sx = 2. * u1 - 1.;
32     sy = 2. * u2 - 1.;
33
34     if (sx == 0. && sy == 0.) return (normale);
35
36     if (sx >= -sy) {
37         if (sx > sy) {
38             r = sx;
39             theta = sy / r;
40         }
41         else {
42             r = sy;
43             theta = 2. - sx / r;
44         }
45     }
46     else {
47         if (sx <= sy) {
48             r = - sx;
49             theta = 4. - sy / r;
50         }
51         else {
52             r = - sy;
53             theta = - 2. + sx / r;
54         }
55     }
56     theta *= M_PI / 4.;
```

```
57
58   Vector v2, v3;
59   CoordinateSystem (normale, & v2, & v3);
60   return (sqrt (1. - r * r) * normale + (r * cos (theta)) *
        v2 + (r * sin (theta)) * v3);
61 }
62
63 Vector UniformSampleSphere (double u1, double u2)
64 {
65     double z = 1. - 2. * u1;
66     double r = sqrt (1. - z * z);
67     double phi = 2. * M_PI * u2;
68     double x = r * cos (phi);
69     double y = r * sin (phi);
70     return Vector (x, y, z);
71 }
72
73 Vector SampleHG (double u1, double u2, double g, const
        Vector &v1)
74 {
75     double costheta, sintheta, tmp, phi;
76
77     tmp = (1. - g * (1. - 2. * u1));
78     tmp *= tmp;
79     costheta = (2. * (1. + g) * (1. + g) * u1 * (g * (u1 - 1.)
        + 1.) / tmp) - 1.;
80     costheta = Clamp (costheta, -1., 1.);
81     sintheta = sqrt (1. - costheta * costheta);
82     phi = (2 * M_PI) * u2;
83     Vector v2, v3;
84     CoordinateSystem (v1, & v2, & v3);
85     return (costheta * v1 + (sintheta * cos (phi)) * v2 + (
        sintheta * sin (phi)) * v3);
86 }
```

D.2 Résolution d'un problème instationnaire simplifié de cinétique chimique d'ordre 2

D.2.1 Sans approximation de guidage y^\sim

Le problème physico-mathématique à l'étude ici est celui qui sert d'exemple au fil du chapitre 4 : on cherche à calculer sur \mathbb{R}^+ :

$$\begin{cases} y'(t) = -\alpha y(t)^2 \\ y(0) = y_0 \end{cases} \quad \text{avec} \quad \begin{cases} \alpha > 0 \\ y_0 \geq 0 \end{cases} \quad (4.5)^\uparrow$$

La solution symbolique à ce système différentielle est aisément obtensible :

$$y(t) = \frac{y_0}{\alpha y_0 t + 1} \quad (4.6)^\uparrow$$

Les 4 fichiers algorithmes qui suivent servent à estimer, par une méthode de Monte-Carlo, $y(t)$ à t donné. Les algorithmes codés dans les fichiers `algo1-sensi.cpp` et `algo2-sensi.cpp` estiment aussi, simultanément, les sensibilités de $y(t)$ selon les paramètres α et y_0 .

Ces fichiers sont prévus pour être compilés avec l'environnement MCM, expliqué à la page 277 ; d'où la ligne `#include "mcmGT2.h"` présente en tête de tous ces fichiers.

Lors de la compilation de ces codes, les valeurs de α , y_0 , \hat{k} , et t doivent être fournies au travers des macros `ALPHA`, `Y0`, `MAJOR`, et `TF` respectivement.

Fichier `algo1.cpp` (code l'algorithme 4.1) :

```
1 #include "mcmGT2.h"
2
3 InfosMontageAlgo Mon_algorithme::giveDims () {
4
5     InfosMontageAlgo infosMontage;
6
7     infosMontage.nb_Grandeurs_RDR = 2;
8     infosMontage.nb_Parametres_RDR = 0;
9     infosMontage.pile_active = 0;
10    infosMontage.nb_Coordonnees_PDR = 2;
11    infosMontage.valeurDeBaseResume = 0.;
12    infosMontage.valeurDeBasePile = 0.;
13
14    return (infosMontage);
15 }
16
17 void Mon_algorithme::algorithme_ponctuel (double t, double *
18     y, long * nbAppels)
19 {
20     double tr, y1, y2;
21
22     (*nbAppels) ++;
23     tr = - log (mcmRng ()) / MAJOR;
24     if (t - tr <= 0.) {
25         *y = Y0;
26         return;
27     }
28     algorithme_ponctuel (t - tr, & y1, nbAppels);
29     algorithme_ponctuel (t - tr, & y2, nbAppels);
30     *y = (1. - ALPHA * y1 / MAJOR) * y2;
```

```
30     return;
31 }
32
33 void Mon_algorithme::calculUnique () // FONCTION ALGORITHME
34 {
35     double y;
36     long nbAppels;
37
38     nbAppels = 0;
39     algorithme_ponctuel (TF, & y, & nbAppels);
40
41     setCoordonneeRes (indiceCoordonneeRes (1), y);
42     setCoordonneeRes (indiceCoordonneeRes (2), nbAppels);
43     enregistrerRealisation ();
44 }
```

Fichier algo2.cpp (code l'algorithme 4.2) :

```
1 #include "mcmGT2.h"
2
3 InfosMontageAlgo Mon_algorithme::giveDims () {
4
5     InfosMontageAlgo infosMontage;
6
7     infosMontage.nb_Grandeurs_RDR = 2;
8     infosMontage.nb_Parametres_RDR = 0;
9     infosMontage.pile_active = 0;
10    infosMontage.nb_Coordonnees_PDR = 2;
11    infosMontage.valeurDeBaseResume = 0.;
12    infosMontage.valeurDeBasePile = 0.;
13
14    return (infosMontage);
15 }
16
17 void Mon_algorithme::algorithme_ponctuel (double t, double *
18     y, long * nbAppels)
19 {
20     double tr, y1;
21
22     (*nbAppels) ++;
23     tr = - log (mcmRng ()) / MAJOR;
24     if (t - tr <= 0.) {
25         *y = Y0;
26         return;
27     }
28     algorithme_ponctuel (t - tr, & y1, nbAppels);
```

```
28     if ((ALPHA * y1 / MAJOR) < mcmRng ()) algorithme_ponctuel
        (t - tr, y, nbAppels);
29     else *y = 0.;
30     return;
31 }
32
33 void Mon_algorithme::calculUnique () // FONCTION ALGORITHME
34 {
35     double y;
36     long nbAppels;
37
38     nbAppels = 0;
39     algorithme_ponctuel (TF, & y, & nbAppels);
40
41     setCoordonneeRes (indiceCoordonneeRes (1), y);
42     setCoordonneeRes (indiceCoordonneeRes (2), nbAppels);
43     enregistrerRealisation ();
44 }
```

Fichier algo1-sensi.cpp (code l'algorithme 4.3) :

```
1 #include "mcmGT2.h"
2
3 InfosMontageAlgo Mon_algorithme::giveDims () {
4
5     InfosMontageAlgo infosMontage;
6
7     infosMontage.nb_Grandeurs_RDR = 4;
8     infosMontage.nb_Parametres_RDR = 0;
9     infosMontage.pile_active = 0;
10    infosMontage.nb_Coordonnees_PDR = 2;
11    infosMontage.valeurDeBaseResume = 0.;
12    infosMontage.valeurDeBasePile = 0.;
13
14    return (infosMontage);
15 }
16
17 void Mon_algorithme::algorithme_ponctuel_avec_sensis (double
    t, double * y, double * day, double * d0y, long *
    nbAppels)
18 {
19     double tr, y1, y2, day1, d0y1;
20
21     (*nbAppels) ++;
22     tr = - log (mcmRng ()) / MAJOR;
23     if (t - tr <= 0.) {
24         *y = Y0;
```

```
25     *day = 0.;
26     *d0y = 1.;
27     return;
28 }
29     algorithme_ponctuel_avec_sensis (t - tr, & y1, & day1, &
        d0y1, nbAppels);
30     algorithme_ponctuel (t - tr, & y2, nbAppels);
31     *y = (1. - ALPHA * y1 / MAJOR) * y2;
32     *day = (day1 * (MAJOR - 2. * ALPHA * y2) - y1 * y2) /
        MAJOR;
33     *d0y = d0y1 * (1. - 2. * ALPHA * y2 / MAJOR);
34     return;
35 }
36
37 void Mon_algorithme::algorithme_ponctuel (double t, double *
        y, long * nbAppels)
38 {
39     double tr, y1, y2;
40
41     (*nbAppels) ++;
42     tr = - log (mcmRng ()) / MAJOR;
43     if (t - tr <= 0.) {
44         *y = Y0;
45         return;
46     }
47     algorithme_ponctuel (t - tr, & y1, nbAppels);
48     algorithme_ponctuel (t - tr, & y2, nbAppels);
49     *y = (1. - ALPHA * y1 / MAJOR) * y2;
50     return;
51 }
52
53 void Mon_algorithme::calculUnique () // FONCTION ALGORITHME
54 {
55     double y, day, d0y;
56     long nbAppels;
57
58     nbAppels = 0;
59     algorithme_ponctuel_avec_sensis (TF, & y, & day, & d0y, &
        nbAppels);
60
61     setCoordonneeRes (indiceCoordonneeRes (1), y);
62     setCoordonneeRes (indiceCoordonneeRes (2), day);
63     setCoordonneeRes (indiceCoordonneeRes (3), d0y);
64     setCoordonneeRes (indiceCoordonneeRes (4), nbAppels);
65     enregistrerRealisation ();
66 }
```

Fichier algo2-sensi.cpp (code l'algorithme 4.4) :

```
1  #include "mcmGT2.h"
2
3  InfosMontageAlgo Mon_algorithme::giveDims () {
4
5      InfosMontageAlgo infosMontage;
6
7      infosMontage.nb_Grandeurs_RDR = 4;
8      infosMontage.nb_Parametres_RDR = 0;
9      infosMontage.pile_active = 0;
10     infosMontage.nb_Coordonnees_PDR = 2;
11     infosMontage.valeurDeBaseResume = 0.;
12     infosMontage.valeurDeBasePile = 0.;
13
14     return (infosMontage);
15 }
16
17 void Mon_algorithme::algorithme_ponctuel_avec_sensis (double
    t, double * y, double * day, double * d0y, long *
    nbAppels)
18 {
19     double tr, y1, y2, day1, d0y1;
20
21     (*nbAppels) ++;
22     tr = - log (mcmRng ()) / MAJOR;
23     if (t - tr <= 0.) {
24         *y = Y0;
25         *day = 0.;
26         *d0y = 1.;
27         return;
28     }
29     algorithme_ponctuel_avec_sensis (t - tr, & y1, & day1, &
        d0y1, nbAppels);
30     algorithme_ponctuel (t - tr, & y2, nbAppels);
31     *y = (1. - ALPHA * y1 / MAJOR) * y2;
32     *day = (day1 * (MAJOR - 2. * ALPHA * y2) - y1 * y2) /
        MAJOR;
33     *d0y = d0y1 * (1. - 2. * ALPHA * y2 / MAJOR);
34     return;
35 }
36
37 void Mon_algorithme::algorithme_ponctuel (double t, double *
    y, long * nbAppels)
38 {
39     double tr, y1;
40
```

```

41  (*nbAppels) ++;
42  tr = - log (mcmRng ()) / MAJOR;
43  if (t - tr <= 0.) {
44      *y = Y0;
45      return;
46  }
47  algorithm_ponctuel (t - tr, & y1, nbAppels);
48  if ((ALPHA * y1 / MAJOR) < mcmRng ()) algorithm_ponctuel
    (t - tr, y, nbAppels);
49  else *y = 0.;
50  return;
51 }
52
53 void Mon_algorithme::calculUnique () // FONCTION ALGORITHME
54 {
55     double y, day, d0y;
56     long nbAppels;
57
58     nbAppels = 0;
59     algorithm_ponctuel_avec_sensis (TF, & y, & day, & d0y, &
        nbAppels);
60
61     setCoordonneeRes (indiceCoordonneeRes (1), y);
62     setCoordonneeRes (indiceCoordonneeRes (2), day);
63     setCoordonneeRes (indiceCoordonneeRes (3), d0y);
64     setCoordonneeRes (indiceCoordonneeRes (4), nbAppels);
65     enregistrerRealisation ();
66 }

```

D.2.2 Avec approximation de guidage y^\sim

●●● Dans le chapitre 4, quand nous testons les bénéfices de l'utilisation, dans un calcul par méthode de Monte-Carlo, d'une solution approximative y^\sim du système (4.5) connue à l'avance, cette solution est générée à l'aide de :

1. la méthode Runge-Kutta d'ordre 4 usuelle (méthode RK4), utilisée avec un pas de temps constant Δt_\approx ,
2. une interpolation cubique, entre les points calculés par la méthode RK4.

La méthode RK4 fournit un résultat sous la forme d'une liste de points $(t_i; y^\sim(t_i))$. C'est une méthode directement explicite, c'est-à-dire que y^\sim_{i+1} est calculée seulement à partir de y^\sim_i , t_i , et t_{i+1} , et ce sans technique d'inversion. La procédure est la suivante :

1. Disposant de $y^\sim(t_i)$, calculer un 1^{er} taux de variation $d_1 = f(y^\sim(t_i); t_i)$.
2. Calculer un 1^{er} point intermédiaire y_2 , en partant de $y^\sim(t_i)$ et en utilisant le taux de variation d_1 : $y_2 = y^\sim(t_i) + \frac{d_1 \Delta t_\approx}{2}$.

3. Calculer un 2^e taux de variation d_2 , au 1^{er} point intermédiaire : $d_2 = f(y_2; t_i + \frac{\Delta t_{\approx}}{2})$.
4. Calculer un 2^e point intermédiaire y_3 , en partant de $y^{\approx}(t_i)$ et en utilisant le taux de variation d_2 : $y_3 = y^{\approx}(t_i) + \frac{d_2 \Delta t_{\approx}}{2}$.
5. Calculer un 3^e taux de variation d_3 , au 2^e point intermédiaire : $d_3 = f(y_3; t_i + \frac{\Delta t_{\approx}}{2})$.
6. Calculer un point final y_4 , en partant de $y^{\approx}(t_i)$ et en utilisant le taux de variation d_3 : $y_4 = y^{\approx}(t_i) + d_3 \Delta t_{\approx}$.
7. Calculer un dernier taux de variation d_4 , au point final : $d_4 = f(y_4; t_i + \Delta t_{\approx})$.
8. Calculer le taux de variation total d_T comme la moyenne pondérée $d_T = (d_1 + 2 d_2 + 2 d_3 + d_4)/6$.
9. Le point $y^{\approx}(t_{i+1})$ est obtenu en partant de $y^{\approx}(t_i)$ et en utilisant le taux de variation d_T : $y^{\approx}(t_i) + d_T \Delta t_{\approx}$.

L'algorithme D.1 résume cette procédure.

Entrées : La condition initiale $(t_0; y^{\approx}(t_0))$, le temps final t_f , et le pas de temps Δt_{\approx}

Sorties : Une solution approximative d'un système du type (4.3), sous la forme d'une liste de points $(t_i; y^{\approx}(t_i))$

```

1   $i \leftarrow 0$ ;
2  tant que  $t_0 + (i + 1)\Delta t_{\approx} \leq t_f$  faire
3       $t_{i+1} \leftarrow t_0 + (i + 1)\Delta t_{\approx}$ ;
4       $d_1 \leftarrow f(y^{\approx}(t_i); t_i)$ ;
5       $y_2 \leftarrow y^{\approx}(t_i) + (d_1 \Delta t_{\approx})/2$ ;
6       $d_2 \leftarrow f(y_2; t_i + (\Delta t_{\approx}/2))$ ;
7       $y_3 \leftarrow y^{\approx}(t_i) + (d_2 \Delta t_{\approx})/2$ ;
8       $d_3 \leftarrow f(y_3; t_i + (\Delta t_{\approx}/2))$ ;
9       $y_4 \leftarrow y^{\approx}(t_i) + d_3 \Delta t_{\approx}$ ;
10      $d_4 \leftarrow f(y_4; t_i + \Delta t_{\approx})$ ;
11      $d_T \leftarrow (d_1 + 2 d_2 + 2 d_3 + d_4)/6$ ;
12      $y^{\approx}(t_{i+1}) \leftarrow y^{\approx}(t_i) + d_T \Delta t_{\approx}$ ;
13      $i \leftarrow i + 1$ ;

```

Algorithme D.1 : Méthode Runge-Kutta d'ordre 4 usuelle, avec pas de temps constant, pour résoudre une équation différentielle d'ordre 1 du type donné en (4.3)

Entre les points fournis par la méthode RK4, on utilise une interpolation cubique. Cette interpolation égalise y^{\approx} et sa dérivée en les points interpolés les plus proches.

Pour que cela fonctionne simplement, les dérivées sont enregistrées lors de la construction de la solution par la méthode RK4. C'est à dire que la méthode RK4 fournit une liste de points $(t_i; y^\sim(t_i); y^{\sim'}(t_i))$. Les $y^{\sim'}(t_i)$ sont calculées directement selon la formule : $y^{\sim'}(t_i) = f(y^\sim(t_i); t_i)$. Ce calcul ne rajoute d'ailleurs aucune charge computationnelle, car la méthode RK4 doit de toute façon l'effectuer.

Considérons ensuite un segment quelconque $[t_i; t_{i+1}]$: sur ce segment, y^\sim est définie par :

$$y^\sim(t) = y^\sim(t_i)P_0(t_{ad}) + y^\sim(t_{i+1})P_1(t_{ad}) + y^{\sim'}(t_i) \times (t_{i+1} - t_i)P_{0'}(t_{ad}) + y^{\sim'}(t_{i+1}) \times (t_{i+1} - t_i)P_{1'}(t_{ad})$$

où

$$t_{ad} = \frac{t - t_i}{t_{i+1} - t_i}$$

$$\begin{array}{ll} P_0 : x \mapsto (2x + 1)(x - 1)^2 & P_1 : x \mapsto x^2(3 - 2x) \\ P_{0'} : x \mapsto x(x - 1)^2 & P_{1'} : x \mapsto x^2(x - 1) \end{array}$$

Les fonctions polynomiales P_0 , P_1 , $P_{0'}$, et $P_{1'}$, respectent les contraintes :

$$\left\{ \begin{array}{llll} P_0(0) = 1 & P_0(1) = 0 & P_0'(0) = 0 & P_0'(1) = 0 \\ P_1(0) = 0 & P_1(1) = 1 & P_1'(0) = 0 & P_1'(1) = 0 \\ P_{0'}(0) = 0 & P_{0'}(1) = 0 & P_{0'}'(0) = 1 & P_{0'}'(1) = 0 \\ P_{1'}(0) = 0 & P_{1'}(1) = 0 & P_{1'}'(0) = 0 & P_{1'}'(1) = 1 \end{array} \right.$$

●●●● Le fichier `algo-guige-sensi.cpp` qui suit implémente à la fois l'algorithme 4.5, et la résolution par la méthode RK4. Celle-ci est effectuée avant que l'algorithme de Monte-Carlo ne démarre.

Ce fichier est prévu pour être compilé avec l'environnement MCM, expliqué à la page 277 ; d'où la ligne `#include "mcmGT2.h"` présente en tête.

Lors de la compilation de ce code, les valeurs de α , y_0 , \hat{k} , t , et Δt_\approx doivent être fournies au travers des macros `ALPHA`, `Y0`, `MAJOR`, `TF` et `TIMESTEP` respectivement.

Fichier `algo-guige-sensi.cpp` (code l'algorithme 4.5) :

```
1 #include "mcmGT2.h"
2
3 #ifdef TIMESTEP
4 Mon_algorithme::Mon_algorithme (int calcCov, int resumeCreux
    , int pileON, int pileCreuse, const gsl_rng_type * nomGen
    , McmTransmetteur * branchement) : McmAlgo (
    Mon_algorithme::giveDims (), calcCov, resumeCreux, pileON
    , pileCreuse, nomGen, branchement), approxTpts (ceil (((
    double) TF) / TIMESTEP) + 1), approxYpts (ceil (((double)
    TF) / TIMESTEP) + 1), approxDpts (ceil (((double) TF) /
    TIMESTEP) + 1)
```

```
5 {
6   RK4solver ();
7 }
8 #endif
9
10 InfosMontageAlgo Mon_algorithme::giveDims () {
11
12   InfosMontageAlgo infosMontage;
13
14   infosMontage.nb_Grandeurs_RDR = 5;
15   infosMontage.nb_Parametres_RDR = 0;
16   infosMontage.pile_active = 0;
17   infosMontage.nb_Coordonnees_PDR = 2;
18   infosMontage.valeurDeBaseResume = 0.;
19   infosMontage.valeurDeBasePile = 0.;
20
21   return (infosMontage);
22 }
23
24 void Mon_algorithme::RK4solver (void)
25 {
26   int step;
27   double y1, k1, y2, k2, y3, k3, y4, k4, yf;
28
29   approxpts [0] = 0.;
30   approxypts [0] = Y0;
31   approxdpts [0] = - ALPHA * Y0 * Y0;
32   for (step = 0; step < approxpts.size () - 1 ; step ++) {
33     approxpts [step + 1] = (step + 1) * TIMESTEP;
34     y1 = approxypts [step];
35     k1 = - ALPHA * y1 * y1;
36     y2 = y1 + .5 * TIMESTEP * k1;
37     k2 = - ALPHA * y2 * y2;
38     y3 = y1 + .5 * TIMESTEP * k2;
39     k3 = - ALPHA * y3 * y3;
40     y4 = y1 + TIMESTEP * k3;
41     k4 = - ALPHA * y4 * y4;
42     yf = y1 + (TIMESTEP / 6.) * (k1 + 2. * k2 + 2. * k3 + k4
43       );
44     approxypts [step + 1] = yf;
45     approxdpts [step + 1] = - ALPHA * yf * yf;
46   }
47
48 void Mon_algorithme::RK4interp (double t, double * y, double
49   * dty)
```

```

50  int dinter;
51  double tadim;
52
53  dinter = floor (t / TIMESTEP); // fonctionne parce que
    les intervalles sont reguliers
54  if (dinter >= approxtps.size () - 1) dinter = approxtps.
    size () - 2;
55  if (dinter < 0) dinter = 0;
56
57  tadim = (t - approxtps [dinter]) / TIMESTEP;
58  if (y) *y = (tadim - 1.) * (tadim - 1.) * ((2. * tadim +
    1.) * approxypts [dinter] + tadim * TIMESTEP *
    approxdpts [dinter]) + tadim * tadim * ((3. - 2. *
    tadim) * approxypts [dinter + 1] + (tadim - 1.) *
    TIMESTEP * approxdpts [dinter + 1]);
59  if (dty) *dty = (tadim - 1.) * (6. * tadim * approxypts [
    dinter] / TIMESTEP + (3. * tadim - 1.) * approxdpts [
    dinter]) + tadim * (6. * (1. - tadim) * approxypts [
    dinter + 1] / TIMESTEP + (3. * tadim - 2.) * approxdpts
    [dinter + 1]);
60 }
61
62 void Mon_algorithme::algorithme_ponctuel_avec_sensis (double
    t, double * yd, double * day, double * d0y, long *
    nbAppels)
63 {
64  double tr, yd1, yd2, y2, day1, d0y1, yg, dtyg;
65
66  (*nbAppels) ++;
67  tr = - log (mcmRng ()) / MAJOR;
68  if (t - tr <= 0.) {
69      RK4interp (0., & yg, NULL);
70      *yd = Y0 - yg;
71      *day = 0.;
72      *d0y = 1.;
73      return;
74  }
75  RK4interp (t - tr, & yg, & dtyg);
76  algorithme_ponctuel_avec_sensis (t - tr, & yd1, & day1, &
    d0y1, nbAppels);
77  algorithme_ponctuel (t - tr, & yd2, nbAppels);
78  y2 = yg + yd2;
79  *yd = yd1 * (1. - ALPHA / MAJOR * (2. * yg + yd2)) - (dtyg
    + ALPHA * yg * yg) / MAJOR;
80  *day = (day1 * (MAJOR - 2. * ALPHA * y2) - (yg + yd1) * y2
    ) / MAJOR;
81  *d0y = d0y1 * (1. - 2. * ALPHA * y2 / MAJOR);

```

```
82     return;
83 }
84
85 void Mon_algorithme::algorithme_ponctuel (double t, double *
      yd, long * nbAppels)
86 {
87     double tr, yd1, yd2, yg, dtyg;
88
89     (*nbAppels) ++;
90     tr = - log (mcmRng ()) / MAJOR;
91     if (t - tr <= 0.) {
92         RK4interp (0., & yg, NULL);
93         *yd = Y0 - yg;
94         return;
95     }
96     RK4interp (t - tr, & yg, & dtyg);
97     algorithme_ponctuel (t - tr, & yd1, nbAppels);
98     algorithme_ponctuel (t - tr, & yd2, nbAppels);
99     *yd = yd1 * (1. - ALPHA / MAJOR * (2. * yg + yd2)) - (dtyg
      + ALPHA * yg * yg) / MAJOR;
100    return;
101 }
102
103 void Mon_algorithme::calculUnique () // FONCTION ALGORITHME
104 {
105     double yg, yd, day, d0y;
106     long nbAppels;
107
108     nbAppels = 0;
109     RK4interp (TF, & yg, NULL);
110     algorithme_ponctuel_avec_sensis (TF, & yd, & day, & d0y, &
      nbAppels);
111
112     setCoordonneeRes (indiceCoordonneeRes (1), yg);
113     setCoordonneeRes (indiceCoordonneeRes (2), yd);
114     setCoordonneeRes (indiceCoordonneeRes (3), day);
115     setCoordonneeRes (indiceCoordonneeRes (4), d0y);
116     setCoordonneeRes (indiceCoordonneeRes (5), nbAppels);
117     enregistrerRealisation ();
118 }
```

D.3 Calcul de $f(\vec{c}; t)$ dans la situation du mode BKW

Le problème physique à l'étude est celui du chapitre 5. Les fichiers algorithmes détaillés ici calculent soit :

- $f(\vec{c}; t)$, pour \vec{c} et t spécifiés,
- la fraction des particules (c'est-à-dire de la masse) dont la vitesse dépasse un seuil c_0 à un instant t , pour c_0 et t spécifiés.

La distribution est connue en $t = t_d = 6 \ln^5/2$, et égalise celle du mode BKW décrite au chapitre 5.1.2.

Ces fichiers sont prévus pour être compilés avec l'environnement MCM, expliqué à la page 277; d'où la ligne `#include "mcmGT2.h"` présente en tête de tous ces fichiers.

Lors de la compilation de ces codes, les valeurs de $t - t_d$, et c_x ou c_0 doivent être fournies au travers des macros DELTAT, POS, et C0 respectivement.

Fichier `vector.h` (inclut par le biais de `mcmGT2.h`) :

```
1  /*****  
2  /* Repris depuis le code source de PBRT */  
3  /*****  
4  
5  /*  
6      pbrt source code Copyright(c) 1998-2010 Matt Pharr and  
          Greg Humphreys.  
7      pbrt is free software; you can redistribute it and/or  
          modify it under the terms of the GNU General Public  
          License as published by the Free Software Foundation;  
          either version 2 of the License, or (at your option)  
          any later version. Note that the text contents of  
          the book "Physically Based Rendering" are *not*  
          licensed under the GNU GPL.  
8      pbrt is distributed in the hope that it will be useful,  
          but WITHOUT ANY WARRANTY; without even the implied  
          warranty of MERCHANTABILITY or FITNESS FOR A  
          PARTICULAR PURPOSE. See the GNU General Public  
          License for more details.  
9      You should have received a copy of the GNU General  
          Public License along with this program. If not, see <  
          http://www.gnu.org/licenses/>.  
10 */  
11  
12 #ifndef KWVECTOR_H  
13 #define KWVECTOR_H  
14
```



```
15 #include <math.h>
16
17 class Vector {
18 public:
19     // Vector Public Methods
20     Vector() { x = y = z = 0.f; }
21     Vector(double xx, double yy, double zz)
22         : x(xx), y(yy), z(zz) { }
23     Vector operator+(const Vector &v) const {
24         return Vector(x + v.x, y + v.y, z + v.z);
25     }
26
27     Vector& operator+=(const Vector &v) {
28         x += v.x; y += v.y; z += v.z;
29         return *this;
30     }
31     Vector operator-(const Vector &v) const {
32         return Vector(x - v.x, y - v.y, z - v.z);
33     }
34
35     Vector& operator-=(const Vector &v) {
36         x -= v.x; y -= v.y; z -= v.z;
37         return *this;
38     }
39     Vector operator*(double f) const { return Vector(f*x, f*
40         y, f*z); }
41
42     Vector &operator*=(double f) {
43         x *= f; y *= f; z *= f;
44         return *this;
45     }
46     Vector operator/(double f) const {
47         double inv = 1. / f;
48         return Vector(x * inv, y * inv, z * inv);
49     }
50     Vector &operator/=(double f) {
51         double inv = 1. / f;
52         x *= inv; y *= inv; z *= inv;
53         return *this;
54     }
55     Vector operator-() const { return Vector(-x, -y, -z); }
56     double operator[](int i) const {
57         return (&x)[i];
58     }
59
60     double &operator[](int i) {
```

```
61         return (&x)[i];
62     }
63     double LengthSquared() const { return x*x + y*y + z*z; }
64     double Length() const { return sqrt (LengthSquared()); }
65
66     bool operator==(const Vector &v) const {
67         return x == v.x && y == v.y && z == v.z;
68     }
69     bool operator!=(const Vector &v) const {
70         return x != v.x || y != v.y || z != v.z;
71     }
72
73     // Vector Public Data
74     double x, y, z;
75 };
76
77 inline Vector operator*(float f, const Vector &v) { return v
    *f; }
78 inline Vector Normalize(const Vector &v) { return v / v.
    Length(); }
79 inline double Dot(const Vector &v1, const Vector &v2) {
    return v1.x * v2.x + v1.y * v2.y + v1.z * v2.z; }
80
81 void tirageGaussien2D (double r1, double r2, double * rn1,
    double * rn2);
82 Vector UniformSampleSphere (double u1, double u2);
83
84 #endif // KWVECTOR_H
```

Fichier `vector.cpp` (contient des routines utilisées par les algorithmes) :

```
1 #include "vector.h"
2
3 void tirageGaussien2D (double r1, double r2, double * rn1,
    double * rn2)
4 {
5     double r, theta, sx, sy;
6
7     // On commence par injecter continuellement et en preservant
        les surfaces le carre standard sur le disque unite...
8     sx = 2. * r1 - 1.;
9     sy = 2. * r2 - 1.;
10    if (sx == 0. && sy == 0.) {
11        if (rn1) *rn1 = 0.;
12        if (rn2) *rn2 = 0.;
13        return;
```

```
14     }
15     if (sx >= -sy) {
16         if (sx > sy) {
17             r = sx;
18             theta = sy / r;
19         }
20         else {
21             r = sy;
22             theta = 2. - sx / r;
23         }
24     }
25     else {
26         if (sx <= sy) {
27             r = - sx;
28             theta = 4. - sy / r;
29         }
30         else {
31             r = - sy;
32             theta = - 2. + sx / r;
33         }
34     }
35     theta *= M_PI / 4.;
36     // ...puis on utilise la methode de Box-Muller.
37     r = sqrt (- 2. * log (1. - r * r));
38     if (rn1) *rn1 = r * cos (theta);
39     if (rn2) *rn2 = r * sin (theta);
40 }
41
42 Vector UniformSampleSphere (double u1, double u2)
43 {
44     double z = 1. - 2. * u1;
45     double r = sqrt (1. - z * z);
46     double phi = 2. * M_PI * u2;
47     double x = r * cos (phi);
48     double y = r * sin (phi);
49     return Vector (x, y, z);
50 }
```

Fichier algo1.cpp (code l'algorithme 5.1) :

```
1 #include "mcmGT2.h"
2 #include <math.h>
3
4 InfosMontageAlgo Mon_algorithme::giveDims () {
5
6     InfosMontageAlgo infosMontage;
7 }
```

```
8   infosMontage.nb_Grandeurs_RDR = 2;
9   infosMontage.nb_Parametres_RDR = 0;
10  infosMontage.pile_active = 0;
11  infosMontage.nb_Coordonnees_PDR = 2;
12  infosMontage.valeurDeBaseResume = 0.;
13  infosMontage.valeurDeBasePile = 0.;
14
15  return (infosMontage);
16 }
17
18 #define T0 ((double) 5.497744391)
19
20 double distribution_initiale (const Vector & c)
21 {
22     double c2n;
23
24     c2n = c.LengthSquared ();
25     return (exp (- 5. / 6. * c2n) * c2n * (25. / (54. * M_PI *
26         sqrt (6. * M_PI / 5.))));
27 }
28
29 void Mon_algorithme::algorithme_ponctuel (double t, const
30     Vector & c, double * f, long * nbAppels)
31 {
32     Vector c2, omega, cm, c3, c4;
33     double dv, f3, f4;
34
35     (*nbAppels) ++;
36     t += log (mcmRng ());
37
38     // Si on remonte le temps avant l'instant initial, on
39     // prend en compte la distribution initiale.
40     if (t <= T0) {
41         *f = distribution_initiale (c);
42         return;
43     }
44
45     // Sinon, on choisit des vitesses entrantes.
46     // D'abord, on tire une vitesse sortante selon une
47     // distribution maxwellienne de vitesse projetee moyenne
48     // quadratique unite.
49     tirageGaussien2D (mcmRng (), mcmRng (), &c2.x, &c2.y);
50     tirageGaussien2D (mcmRng (), mcmRng (), &c2.z, NULL);
51     // Ensuite, on tire une direction entrante.
52     omega = UniformSampleSphere (mcmRng (), mcmRng ());
53     // Ensuite on calcule les directions entrantes
54     // correspondantes.
```

```
49   cm = .5 * (c + c2);
50   dv = .5 * (c - c2).Length ();
51   c3 = cm + dv * omega;
52   c4 = cm - dv * omega;
53
54   // Calcul du terme source
55   algorithme_ponctuel (t, c3, &f3, nbAppels);
56   algorithme_ponctuel (t, c4, &f4, nbAppels);
57   *f = f3 * f4 * 15.74960995 /* (2*pi)^1.5 */ * exp (.5 * c2
        .LengthSquared ());
58 }
59
60 void Mon_algorithme::calculUnique () // FONCTION ALGORITHME
61 {
62     double f;
63     long nbAppels;
64
65     nbAppels = 0;
66     algorithme_ponctuel (T0 + DELTAT, Vector (POS, 0., 0.), &
        f, & nbAppels);
67
68     setCoordonneeRes (indiceCoordonneeRes (1), f);
69     setCoordonneeRes (indiceCoordonneeRes (2), nbAppels);
70     enregistrerRealisation ();
71 }
```

Fichier algo1-queue.cpp (code l'algorithme 5.2) :

```
1 #include "mcmGT2.h"
2 #include <math.h>
3
4 InfosMontageAlgo Mon_algorithme::giveDims () {
5
6     InfosMontageAlgo infosMontage;
7
8     infosMontage.nb_Grandeurs_RDR = 2;
9     infosMontage.nb_Parametres_RDR = 0;
10    infosMontage.pile_active = 0;
11    infosMontage.nb_Coordonnees_PDR = 2;
12    infosMontage.valeurDeBaseResume = 0.;
13    infosMontage.valeurDeBasePile = 0.;
14
15    return (infosMontage);
16 }
17
18 #define T0 ((double) 5.497744391)
19
```

```
20 double distribution_initiale (const Vector & c)
21 {
22     double c2n;
23
24     c2n = c.LengthSquared ();
25     return (exp (- 5. / 6. * c2n) * c2n * (25. / (54. * M_PI *
26         sqrt (6. * M_PI / 5.))));
27 }
28 void Mon_algorithme::algorithme_ponctuel (double t, const
29     Vector & c, double * f, long * nbAppels)
30 {
31     Vector c2, omega, cm, c3, c4;
32     double dv, f3, f4;
33
34     (*nbAppels) ++;
35     t += log (mcmRng ());
36
37     // Si on remonte le temps avant l'instant initial, on
38     // prend en compte la distribution initiale.
39     if (t <= T0) {
40         *f = distribution_initiale (c);
41         return;
42     }
43
44     // Sinon, on choisit des vitesses entrantes.
45     // D'abord, on tire une vitesse sortante selon une
46     // distribution maxwellienne de vitesse projetee moyenne
47     // quadratique unite.
48     tirageGaussien2D (mcmRng (), mcmRng (), &c2.x, &c2.y);
49     tirageGaussien2D (mcmRng (), mcmRng (), &c2.z, NULL);
50     // Ensuite, on tire une direction entrante.
51     omega = UniformSampleSphere (mcmRng (), mcmRng ());
52     // Ensuite on calcule les directions entrantes
53     // correspondantes.
54     cm = .5 * (c + c2);
55     dv = .5 * (c - c2).Length ();
56     c3 = cm + dv * omega;
57     c4 = cm - dv * omega;
58
59     // Calcul du terme source
60     algorithme_ponctuel (t, c3, &f3, nbAppels);
61     algorithme_ponctuel (t, c4, &f4, nbAppels);
62     *f = f3 * f4 * 15.74960995 /* (2*pi)^1.5 */ * exp (.5 * c2
63         .LengthSquared ());
64 }
```

```
60 void Mon_algorithme::calculUnique () // FONCTION ALGORITHME
61 {
62     double f, norminit, tirage;
63     long nbAppels;
64
65     tirage = mcmRng ();
66     norminit = C0 > 2. ? C0 * (1. + ((1. / tirage) - 1.) * (1.
        / (C0 * C0 - 2.))) : 2. * (1. / tirage - 1.) + C0;
67     Vector cinit (norminit * UniformSampleSphere (mcmRng (),
        mcmRng ()));
68
69     nbAppels = 0;
70     algorithme_ponctuel (T0 + DELTAT, cinit, & f, & nbAppels);
71     f *= C0 > 2. ? (4 * M_PI * C0 / (C0 * C0 - 2.) * norminit
        * norminit) / (tirage * tirage) : 8. * M_PI * norminit
        * norminit / (tirage * tirage);
72
73     setCoordonneeRes (indiceCoordonneeRes (1), f);
74     setCoordonneeRes (indiceCoordonneeRes (2), nbAppels);
75     enregistrerRealisation ();
76 }
```

Fichier algo2.cpp (code l'algorithme 5.3) :

```
1 #include "mcmGT2.h"
2 #include <math.h>
3
4 InfosMontageAlgo Mon_algorithme::giveDims () {
5
6     InfosMontageAlgo infosMontage;
7
8     infosMontage.nb_Grandeurs_RDR = 2;
9     infosMontage.nb_Parametres_RDR = 0;
10    infosMontage.pile_active = 0;
11    infosMontage.nb_Coordonnees_PDR = 2;
12    infosMontage.valeurDeBaseResume = 0.;
13    infosMontage.valeurDeBasePile = 0.;
14
15    return (infosMontage);
16 }
17
18 #define T0 ((double) 5.497744391)
19 #define PALIER1 (T0 + 1.2)
20 #define PALIER2 (T0 + 2.5)
21 #define PALIER3 (T0 + 4.0)
22
23 double distribution_initiale (const Vector & c)
```

```
24 {
25     double c2n;
26
27     c2n = c.LengthSquared ();
28     return (exp (- 5. / 6. * c2n) * c2n * (25. / (54. * M_PI *
        sqrt (6. * M_PI / 5.))));
29 }
30
31 void Mon_algorithme::algorithme_ponctuel (double t, const
    Vector & c, double * f, long * nbAppels)
32 {
33     Vector c2, omega, cm, c3, c4;
34     double trecul, dv, f3, f4, f1;
35     int tirages, itirage;
36
37     (*nbAppels) ++;
38     trecul = log (mcmRng ());
39
40     // Si on remonte le temps avant l'instant initial, on
        prend en compte la distribution initiale.
41     if (t + trecul <= T0) {
42         *f = distribution_initiale (c);
43         return;
44     }
45
46     // Tests : si on franchit un des paliers, on multiplie les
        chemins suivis pour moyenner leur contribution
47     if ((t > PALIER1) && (t + trecul <= PALIER1)) tirages = 9;
48     else if ((t > PALIER2) && (t + trecul <= PALIER2)) tirages
        = 4;
49     else if ((t > PALIER3) && (t + trecul <= PALIER3)) tirages
        = 4;
50     else tirages = 1;
51
52     for (itirage = 0, f1 = 0.; itirage < tirages; itirage ++)
        {
53         // On choisit des vitesses entrantes.
54         // D'abord, on tire une vitesse sortante selon une
            distribution maxwellienne de vitesse projetee moyenne
            quadratique unite.
55         tirageGaussien2D (mcmRng (), mcmRng (), &c2.x, &c2.y);
56         tirageGaussien2D (mcmRng (), mcmRng (), &c2.z, NULL);
57         // Ensuite, on tire une direction entrante.
58         omega = UniformSampleSphere (mcmRng (), mcmRng ());
59         // Ensuite on calcule les directions entrantes
            correspondantes.
60         cm = .5 * (c + c2);
```



```
61     dv = .5 * (c - c2).Length ();
62     c3 = cm + dv * omega;
63     c4 = cm - dv * omega;
64     // Calcul du terme source
65     algorithme_ponctuel (t + trecul, c3, &f3, nbAppels);
66     algorithme_ponctuel (t + trecul, c4, &f4, nbAppels);
67     f1 += f3 * f4 * 15.74960995 /* (2*pi)^1.5 */ * exp (.5 *
        c2.LengthSquared ());
68 }
69 *f = f1 / tirages;
70 }
71
72 void Mon_algorithme::calculUnique () // FONCTION ALGORITHME
73 {
74     double f;
75     long nbAppels;
76
77     nbAppels = 0;
78     algorithme_ponctuel (T0 + DELTAT, Vector (POS, 0., 0.), &
        f, & nbAppels);
79
80     setCoordonneeRes (indiceCoordonneeRes (1), f);
81     setCoordonneeRes (indiceCoordonneeRes (2), nbAppels);
82     enregistrerRealisation ();
83 }
```

Fichier `algo2-queue.cpp` (algorithme calculant des fractions de la masse au-delà d'une certaine vitesse, sur le modèle de l'algorithme 5.2, mais en estimant $f(\vec{c}; t)$ grâce à l'algorithme 5.3) :

```
1 #include "mcmGT2.h"
2 #include <math.h>
3
4 InfosMontageAlgo Mon_algorithme::giveDims () {
5
6     InfosMontageAlgo infosMontage;
7
8     infosMontage.nb_Grandeurs_RDR = 2;
9     infosMontage.nb_Parametres_RDR = 0;
10    infosMontage.pile_active = 0;
11    infosMontage.nb_Coordonnees_PDR = 2;
12    infosMontage.valeurDeBaseResume = 0.;
13    infosMontage.valeurDeBasePile = 0.;
14
15    return (infosMontage);
16 }
17
```

```
18 #define T0 ((double) 5.497744391)
19 #define PALIER1 (T0 + 1.2)
20 #define PALIER2 (T0 + 2.5)
21 #define PALIER3 (T0 + 4.0)
22
23 double distribution_initiale (const Vector & c)
24 {
25     double c2n;
26
27     c2n = c.LengthSquared ();
28     return (exp (- 5. / 6. * c2n) * c2n * (25. / (54. * M_PI *
29         sqrt (6. * M_PI / 5.))));
30 }
31 void Mon_algorithme::algorithme_ponctuel (double t, const
32     Vector & c, double * f, long * nbAppels)
33 {
34     Vector c2, omega, cm, c3, c4;
35     double trecul, dv, f3, f4, f1;
36     int tirages, itirage;
37
38     (*nbAppels) ++;
39     trecul = log (mcmRng ());
40
41     // Si on remonte le temps avant l'instant initial, on
42     // prend en compte la distribution initiale.
43     if (t + trecul <= T0) {
44         *f = distribution_initiale (c);
45         return;
46     }
47
48     // Tests : si on franchit un des paliers, on multiplie les
49     // chemins suivis pour moyenner leur contribution
50     if ((t > PALIER1) && (t + trecul <= PALIER1)) tirages = 9;
51     else if ((t > PALIER2) && (t + trecul <= PALIER2)) tirages
52         = 4;
53     else if ((t > PALIER3) && (t + trecul <= PALIER3)) tirages
54         = 4;
55     else tirages = 1;
56
57     for (itirage = 0, f1 = 0.; itirage < tirages; itirage ++ )
58     {
59         // On choisit des vitesses entrantes.
60         // D'abord, on tire une vitesse sortante selon une
61         // distribution maxwellienne de vitesse projetee moyenne
62         // quadratique unite.
63         tirageGaussien2D (mcmRng (), mcmRng (), &c2.x, &c2.y);
```

```
56     tirageGaussien2D (mcmRng (), mcmRng (), &c2.z, NULL);
57     // Ensuite, on tire une direction entrante.
58     omega = UniformSampleSphere (mcmRng (), mcmRng ());
59     // Ensuite on calcule les directions entrantes
        correspondantes.
60     cm = .5 * (c + c2);
61     dv = .5 * (c - c2).Length ();
62     c3 = cm + dv * omega;
63     c4 = cm - dv * omega;
64     // Calcul du terme source
65     algorithme_ponctuel (t + trecul, c3, &f3, nbAppels);
66     algorithme_ponctuel (t + trecul, c4, &f4, nbAppels);
67     f1 += f3 * f4 * 15.74960995 /* (2*pi)^1.5 */ * exp (.5 *
        c2.LengthSquared ());
68 }
69 *f = f1 / tirages;
70 }
71
72 void Mon_algorithme::calculUnique () // FONCTION ALGORITHME
73 {
74     double f, norminit, tirage;
75     long nbAppels;
76
77     tirage = mcmRng ();
78     norminit = C0 > 2. ? C0 * (1. + ((1. / tirage) - 1.) * (1.
        / (C0 * C0 - 2.))) : 2. * (1. / tirage - 1.) + C0;
79     Vector cinit (norminit * UniformSampleSphere (mcmRng (),
        mcmRng ()));
80
81     nbAppels = 0;
82     algorithme_ponctuel (T0 + DELTAT, cinit, & f, & nbAppels);
83     f *= C0 > 2. ? (4 * M_PI * C0 / (C0 * C0 - 2.) * norminit
        * norminit) / (tirage * tirage) : 8. * M_PI * norminit
        * norminit / (tirage * tirage);
84
85     setCoordonneeRes (indiceCoordonneeRes (1), f);
86     setCoordonneeRes (indiceCoordonneeRes (2), nbAppels);
87     enregistrerRealisation ();
88 }
```

D.4 Calcul de $f(\vec{r}; \vec{c}; t)$ dans la situation de confinement harmonique

Le problème physique à l'étude est celui du chapitre 6. Les fichiers algorithmes détaillés ici calculent soit :

- $f(\vec{r}; \vec{c}; t)$, pour \vec{r} , \vec{c} et t spécifiés,
- $\eta(\vec{r}; t)$, pour \vec{r} et t spécifiés,
- la fraction des particules (c'est-à-dire de la masse) dont la distance à l'origine dépasse un seuil r_0 à un instant t , pour r_0 et t spécifiés.

La distribution est connue en $t = 0$, et égale celle décrite au chapitre 6.1.

Les calculs sont divisés entre plusieurs fichiers :

- Les fichiers `algoN.cpp` où $N \in \llbracket 1; 6 \rrbracket$ implémentent chacun une façon différente d'estimer $f(\vec{r}; \vec{c}; t)$, estimation à la base de tout les calculs effectués.
- Les fichiers `algo-X.cpp` où $X \in \{\text{f}; \text{dens}; \text{GR}\}$ effectuent si nécessaire l'intégration finale de f dans l'espace des phases.
- Les fichiers `approximateurs0.cpp` et `approximateurs1.cpp` contiennent chacun un jeu d'approximations des grandeurs macroscopiques, utilisées dans les procédures d'estimations de $f(\vec{r}; \vec{c}; t)$. Le fichier `approximateurs0.cpp` fournit les grandeurs exactes, ce qui amène presque tous les algorithmes d'estimation de f à la variance nulle. Le fichier `approximateurs1.cpp` fournit les approximations détaillées au chapitre 6.2.3.

Conformément à ce qui est détaillé au chapitre 6.1, les paramètres n , ω , et $c_{\text{qm max}}$ sont fixés d'emblée respectivement à 1 mol, 2π Hz, et 1 m/s. Les paramètres adimensionnels $\frac{c_{\text{qm min}}}{c_{\text{qm max}}}$, $\frac{n\kappa\omega^2}{c_{\text{qm max}}^3}$, et ϕ_0 doivent par contre être fournis par l'utilisateur lors de la compilation, au travers des macros `PI1`, `PI2` et `PHI0` respectivement. Il en va de même pour les paramètres du lieu de la mesure : t doit être fourni au travers de `TFIN` en secondes, r_0 (si nécessaire) au travers de `R0` en mètres, et \vec{r} et \vec{c} (si nécessaires) au travers de `POS` en unités SI.

Ces fichiers sont prévus pour être compilés avec l'environnement MCM, expliqué à la page 277 ; d'où la ligne `#include "mcmGT2.h"` présente en tête de tous ces fichiers.

Fichier `vector.h` (inclut par le biais de `mcmGT2.h`) : voir la section précédente D.3

Fichier `vector.cpp` (contient des routines utilisées par les algorithmes) : voir la section précédente D.3

Fichier `params.h` (contient les déclarations des paramètres physiques et des approximations) :

```

1 #ifndef PARAMS_H
2 #define PARAMS_H
3
4 #include <math.h>
5 #define OMEGA (2. * M_PI)
6 #define VMIN ((double) PI1)
7 #define KAPPA (((double) PI2) / (OMEGA * OMEGA))

```

```
8 #define NMAX (OMEGA * OMEGA * OMEGA / (15.74960995 /* (2*pi)
    ^ (3/2) */ * VMIN * VMIN * VMIN))
9 #define NEST (1. * NMAX)
10
11 #include "vector.h"
12 double distrib_exacte (const Vector & pos, const Vector &
    vit, double t);
13 double distrib_approx (const Vector & pos, const Vector &
    vit, double t);
14 double vitQCM_approx (double t);
15 Vector vitM_approx (const Vector & pos, double t);
16 double dens_approx (const Vector & pos, double t);
17
18 #endif
```

Fichier approximateurs0.cpp (fournit les grandeurs macroscopiques, exactes) :

```
1 #include <math.h>
2 #include "vector.h"
3 #include "params.h"
4
5 double distrib_exacte (const Vector & pos, const Vector &
    vit, double t)
6 {
7     double sint, cost;
8
9     sint = sin (2. * (OMEGA * t + PHIO));
10    cost = cos (2. * (OMEGA * t + PHIO));
11    return (((OMEGA * OMEGA * OMEGA) / (8. * M_PI * M_PI *
        M_PI * VMIN * VMIN * VMIN)) * exp (- (((1. / (VMIN *
        VMIN)) + 1.) * .25 - ((1. / (VMIN * VMIN)) - 1.) * .25
        * sint) * (OMEGA * OMEGA) * pos.LengthSquared () + ((1.
        / (VMIN * VMIN)) - 1.) * .5 * OMEGA * cost * Dot (pos,
        vit) - (((1. / (VMIN * VMIN)) + 1.) * .25 + ((1. / (
        VMIN * VMIN)) - 1.) * .25 * sint) * vit.LengthSquared
        ()))));
12 }
13
14 double distrib_approx (const Vector & pos, const Vector &
    vit, double t)
15 {
16     return (distrib_exacte (pos, vit, t));
17 }
18
19 double vitQCM_approx (double t)
20 {
```

```
21     return (1. / sqrt (((1. / (VMIN * VMIN)) + 1.) * .5 + ((1.
        / (VMIN * VMIN)) - 1.) * .5 * sin (2. * (OMEGA * t +
        PHIO))));
22 }
23
24 Vector vitM_approx (const Vector & pos, double t)
25 {
26     double fact;
27
28     fact = ((1. - VMIN * VMIN) * cos (2. * (OMEGA * t + PHIO))
        ) / ((1. + VMIN * VMIN) + (1. - VMIN * VMIN) * sin (2.
        * (OMEGA * t + PHIO)));
29     return (fact * OMEGA * pos);
30 }
31
32 double dens_approx (const Vector & pos, double t)
33 {
34     double vittmp;
35
36     vittmp = 1. + VMIN * VMIN + (1. - VMIN * VMIN) * sin (2. *
        (OMEGA * t + PHIO));
37     return ((OMEGA * OMEGA * OMEGA) / 5.568327997 / (vittmp *
        sqrt (vittmp)) * exp (- OMEGA * OMEGA * pos.
        LengthSquared () / vittmp));
38 }
```

Fichier `approximateurs1.cpp` (fournit les grandeurs macroscopiques, approximées) :

```
1 #include <math.h>
2 #include "vector.h"
3 #include "params.h"
4
5 double distrib_exacte (const Vector & pos, const Vector &
    vit, double t)
6 {
7     double sint, cost;
8
9     sint = sin (2. * (OMEGA * t + PHIO));
10    cost = cos (2. * (OMEGA * t + PHIO));
11    return (((OMEGA * OMEGA * OMEGA) / (8. * M_PI * M_PI *
        M_PI * VMIN * VMIN * VMIN)) * exp (- (((1. / (VMIN *
        VMIN)) + 1.) * .25 - ((1. / (VMIN * VMIN)) - 1.) * .25
        * sint) * (OMEGA * OMEGA) * pos.LengthSquared () + ((1.
        / (VMIN * VMIN)) - 1.) * .5 * OMEGA * cost * Dot (pos,
        vit) - (((1. / (VMIN * VMIN)) + 1.) * .25 + ((1. / (
        VMIN * VMIN)) - 1.) * .25 * sint) * vit.LengthSquared
```

```
        ());
12 }
13
14 double distrib_approx (const Vector & pos, const Vector &
    vit, double t)
15 {
16     return (((OMEGA * OMEGA * OMEGA) / (M_PI * M_PI * M_PI *
        (1. + VMIN * VMIN) * (1. + VMIN * VMIN) * (1. + VMIN *
        VMIN))) * exp (- 1. / (1. + VMIN * VMIN) * (OMEGA *
        OMEGA) * pos.LengthSquared () - 1. / (1. + VMIN * VMIN)
        * vit.LengthSquared ()));
17 }
18
19 double vitQCM_approx (double t)
20 {
21     return (sqrt (.5 * (1. + VMIN * VMIN)));
22 }
23
24 Vector vitM_approx (const Vector & pos, double t)
25 {
26     return (Vector (0., 0., 0.));
27 }
28
29 double dens_approx (const Vector & pos, double t)
30 {
31     return ((OMEGA * OMEGA * OMEGA) / 5.568327997 /*  $\pi^{(3/2)}$ 
        */ / ((1. + VMIN * VMIN) * sqrt (1. + VMIN * VMIN)) *
        exp (- OMEGA * OMEGA * pos.LengthSquared () / (1. +
        VMIN * VMIN)));
32 }
```

Fichier algo-f.cpp (permet le calcul de f en un point quelconque de l'espace des phases) :

```
1 #include "mcmGT2.h"
2 #include "vector.h"
3
4 InfosMontageAlgo Mon_algorithme::giveDims () {
5
6     InfosMontageAlgo infosMontage;
7
8     infosMontage.nb_Grandeurs_RDR = 2;
9     infosMontage.nb_Parametres_RDR = 0;
10    infosMontage.pile_active = 0;
11    infosMontage.nb_Coordonnees_PDR = 2;
12    infosMontage.valeurDeBaseResume = 0.;
13    infosMontage.valeurDeBasePile = 0.;
```

```
14
15     return (infosMontage);
16 }
17
18
19 void Mon_algorithme::calculUnique () // FONCTION ALGORITHME
20 {
21     double f;
22     long nbAppels;
23
24     nbAppels = 0;
25
26     algorithme_ponctuel (POS, TFIN, & f, & nbAppels);
27
28     setCoordonneeRes (indiceCoordonneeRes (1), f);
29     setCoordonneeRes (indiceCoordonneeRes (2), nbAppels);
30     enregistrerRealisation ();
31 }
```

Fichier algo-dens.cpp (permet le calcul de η en un point quelconque de l'espace usuel) :

```
1 #include "mcmGT2.h"
2 #include "vector.h"
3 #include <math.h>
4 #include "params.h"
5
6 InfosMontageAlgo Mon_algorithme::giveDims () {
7
8     InfosMontageAlgo infosMontage;
9
10    infosMontage.nb_Grandeurs_RDR = 2;
11    infosMontage.nb_Parametres_RDR = 0;
12    infosMontage.pile_active = 0;
13    infosMontage.nb_Coordonnees_PDR = 2;
14    infosMontage.valeurDeBaseResume = 0.;
15    infosMontage.valeurDeBasePile = 0.;
16
17    return (infosMontage);
18 }
19
20
21 void Mon_algorithme::calculUnique () // FONCTION ALGORITHME
22 {
23     double f, vitQCMdep;
24     Vector posDep, vitN, vitDep;
25     long nbAppels;
```



```
26
27     nbAppels = 0;
28
29     // choix d'une direction d'arrivee
30     tirageGaussien2D (mcmRng (), mcmRng (), &vitN.x, &vitN.y);
31     tirageGaussien2D (mcmRng (), mcmRng (), &vitN.z, NULL);
32     vitQCMdep = vitQCM_approx (TFIN);
33     vitDep = vitQCMdep * vitN;
34     vitDep += vitM_approx (POS, TFIN);
35
36     // lancement du calcul
37     algorithme_ponctuel (POS, vitDep, TFIN, & f, & nbAppels);
38     // correction par les probas de tirage
39     f *= 15.74960995 /* (2*pi)^1.5 */ * (vitQCMdep * vitQCMdep
        * vitQCMdep) * exp (.5 * vitN.LengthSquared ());
40
41     setCoordonneeRes (indiceCoordonneeRes (1), f);
42     setCoordonneeRes (indiceCoordonneeRes (2), nbAppels);
43     enregistrerRealisation ();
44 }
```

Fichier algo-GR.cpp (permet le calcul de la fraction des particules éloignées de l'origine des positions) :

```
1 #include "mcmGT2.h"
2 #include "vector.h"
3 #include <math.h>
4
5 InfosMontageAlgo Mon_algorithme::giveDims () {
6
7     InfosMontageAlgo infosMontage;
8
9     infosMontage.nb_Grandeurs_RDR = 2;
10    infosMontage.nb_Parametres_RDR = 0;
11    infosMontage.pile_active = 0;
12    infosMontage.nb_Coordonnees_PDR = 2;
13    infosMontage.valeurDeBaseResume = 0.;
14    infosMontage.valeurDeBasePile = 0.;
15
16    return (infosMontage);
17 }
18
19
20 void Mon_algorithme::calculUnique () // FONCTION ALGORITHME
21 {
22     double tirage, normPos, f, vitQCMdep;
23     Vector posDep, vitN, vitDep;
```

```
24  long nbAppels;
25
26  nbAppels = 0;
27
28  // choix d'un point d'arrivee
29  tirage = mcmRng ();
30  normPos = R0 > 2. ? R0 * (1. + ((1. / tirage) - 1.) * (1.
    / (R0 * R0 - 2.))) : 2. * (1. / tirage - 1.) + R0;
31  posDep = normPos * UniformSampleSphere (mcmRng (), mcmRng
    ());
32
33  // choix d'une direction d'arrivee
34  tirageGaussien2D (mcmRng (), mcmRng (), &vitN.x, &vitN.y);
35  tirageGaussien2D (mcmRng (), mcmRng (), &vitN.z, NULL);
36  vitQCMdep = vitQCM_approx (TFIN);
37  vitDep = vitQCMdep * vitN;
38  vitDep += vitM_approx (posDep, TFIN);
39
40  // lancement du calcul
41  algorithme_ponctuel (posDep, vitDep, TFIN, & f, & nbAppels
    );
42  // correction par les probas de tirage
43  f *= (R0 > 2. ? (4 * M_PI * R0 / (R0 * R0 - 2.) * normPos
    * normPos) / (tirage * tirage) : 8. * M_PI * normPos *
    normPos / (tirage * tirage)) * (15.74960995 /* (2*pi)
    ^1.5 */ * (vitQCMdep * vitQCMdep * vitQCMdep) * exp (.5
    * vitN.LengthSquared ()));
44
45  setCoordonneeRes (indiceCoordonneeRes (1), f);
46  setCoordonneeRes (indiceCoordonneeRes (2), nbAppels);
47  enregistrerRealisation ();
48 }
```

Fichier algo1.cpp (code l'algorithme 6.1) :

```
1  #include "mcmGT2.h"
2  #include "vector.h"
3  #include <math.h>
4  #include "params.h"
5
6  void Mon_algorithme::algorithme_ponctuel (const Vector & pos
    , const Vector & vit, double t, double * f, long *
    nbAppels)
7  {
8      Vector A, B, posC, vitC, vitN, vit1, omega, vit2, vit3,
        vitm;
```

```
9     double tvol, sint, cost, f1, f2, f3, f4, vitQCMtc, dvit,
      finv;
10
11     (*nbAppels) ++;
12
13     // On commence par choisir un temps de vol.
14     tvol = - log (mcmRng ()) / (KAPPA * NEST);
15
16     // Et on calcule les parametres de trajectoire du vol.
17     sint = sin (OMEGA * t);
18     cost = cos (OMEGA * t);
19     A = cost * pos - (sint / OMEGA) * vit;
20     B = sint * pos + (cost / OMEGA) * vit;
21
22     // Si on remonte le temps avant l'instant initial, on
      prend en compte la distribution initiale.
23     if (tvol >= t) {
24         *f = distrib_exacte (A, OMEGA * B, 0.);
25         return;
26     }
27
28     // Sinon, on retrouve la position de depart de la
      particule.
29     sint = sin (OMEGA * (t - tvol));
30     cost = cos (OMEGA * (t - tvol));
31     posC = cost * A + sint * B;
32     vitC = - OMEGA * sint * A + OMEGA * cost * B;
33
34     // On choisit une partenaire de collision.
35     tirageGaussien2D (mcmRng (), mcmRng (), &vitN.x, &vitN.y);
36     tirageGaussien2D (mcmRng (), mcmRng (), &vitN.z, NULL);
37     vitQCMtc = vitQCM_approx (t - tvol);
38     vit1 = vitQCMtc * vitN;
39     vit1 += vitM_approx (posC, t - tvol);
40
41     // On va chercher recursivement la fonction de
      distribution.
42     algorithme_ponctuel (posC, vit1, t - tvol, &f1, nbAppels);
43     algorithme_ponctuel (posC, vitC, t - tvol, &f2, nbAppels);
44
45     // On calcule des directions entrantes.
46     omega = UniformSampleSphere (mcmRng (), mcmRng ());
47     vitm = .5 * (vitC + vit1);
48     dvit = .5 * (vitC - vit1).Length ();
49     vit2 = vitm + dvit * omega;
50     vit3 = vitm - dvit * omega;
51
```

```
52 // On va chercher recursivement la fonction de
    distribution.
53 algorithme_ponctuel (posC, vit2, t - tvol, &f3, nbAppels);
54 algorithme_ponctuel (posC, vit3, t - tvol, &f4, nbAppels);
55 // On calcule un resultat.
56 finv = 15.74960995 /* (2*pi)^1.5 */ * (vitQCMtc * vitQCMtc
    * vitQCMtc) * exp (.5 * vitN.LengthSquared ());
57 *f = f2 * (1. - finv * f1 / NEST) + finv * f3 * f4 / NEST;
58 return;
59 }
```

Fichier algo2.cpp (code l'algorithme 6.2) :

```
1 #include "mcmGT2.h"
2 #include "vector.h"
3 #include <math.h>
4 #include "params.h"
5
6 void Mon_algorithme::algorithme_ponctuel (const Vector & pos
    , const Vector & vit, double t, double * f, long *
    nbAppels)
7 {
8     Vector A, B, posC, vitC, vitN, vit1, omega, vit2, vit3,
        vitm;
9     double tvol, sint, cost, f1, f2, f3, f4, vitQCMtc, dvit,
        finv;
10
11     (*nbAppels) ++;
12
13     // On commence par choisir un temps de vol.
14     tvol = - log (mcmRng ()) / (KAPPA * NEST);
15
16     // Et on calcule les parametres de trajectoire du vol.
17     sint = sin (OMEGA * t);
18     cost = cos (OMEGA * t);
19     A = cost * pos - (sint / OMEGA) * vit;
20     B = sint * pos + (cost / OMEGA) * vit;
21
22     // Si on remonte le temps avant l'instant initial, on
        prend en compte la distribution initiale.
23     if (tvol >= t) {
24         *f = distrib_exacte (A, OMEGA * B, 0.);
25         return;
26     }
27
28     // Sinon, on retrouve la position de depart de la
        particule.
```

```
29     sint = sin (OMEGA * (t - tvol));
30     cost = cos (OMEGA * (t - tvol));
31     posC = cost * A + sint * B;
32     vitC = - OMEGA * sint * A + OMEGA * cost * B;
33
34     // On choisit une partenaire de collision.
35     tirageGaussien2D (mcmRng (), mcmRng (), &vitN.x, &vitN.y);
36     tirageGaussien2D (mcmRng (), mcmRng (), &vitN.z, NULL);
37     vitQCMtc = vitQCM_approx (t - tvol);
38     vit1 = vitQCMtc * vitN;
39     vit1 += vitM_approx (posC, t - tvol);
40     finv = 15.74960995 /* (2*pi)^1.5 */ * (vitQCMtc * vitQCMtc
        * vitQCMtc) * exp (.5 * vitN.LengthSquared ());
41
42     // On va chercher recursivement la fonction de
        distribution.
43     algorithme_ponctuel (posC, vit1, t - tvol, &f1, nbAppels);
44     algorithme_ponctuel (posC, vitC, t - tvol, &f2, nbAppels);
45
46     // On calcule des directions entrantes, en considerant une
        nouvelle partenaire de collision.
47     tirageGaussien2D (mcmRng (), mcmRng (), &vitN.x, &vitN.y);
48     tirageGaussien2D (mcmRng (), mcmRng (), &vitN.z, NULL);
49     vitQCMtc = vitQCM_approx (t - tvol);
50     vit1 = vitQCMtc * vitN;
51     vit1 += vitM_approx (posC, t - tvol);
52     omega = UniformSampleSphere (mcmRng (), mcmRng ());
53     vitm = .5 * (vitC + vit1);
54     dvit = .5 * (vitC - vit1).Length ();
55     vit2 = vitm + dvit * omega;
56     vit3 = vitm - dvit * omega;
57
58     // On va chercher recursivement la fonction de
        distribution.
59     algorithme_ponctuel (posC, vit2, t - tvol, &f3, nbAppels);
60     algorithme_ponctuel (posC, vit3, t - tvol, &f4, nbAppels);
61     // On calcule un resultat.
62     *f = f2 * (1. - finv * f1 / NEST) + 15.74960995 /* (2*pi)
        ^1.5 */ * (vitQCMtc * vitQCMtc * vitQCMtc) * exp (.5 *
        vitN.LengthSquared ()) * f3 * f4 / NEST;
63     return;
64 }
```

Fichier algo3.cpp (code l'algorithme 6.3) :

```
1 #include "mcmGT2.h"
2 #include "vector.h"
```

```
3 #include <math.h>
4 #include "params.h"
5
6 void Mon_algorithme::algorithme_ponctuel (const Vector & pos
      , const Vector & vit, double t, double * f, long *
      nbAppels)
7 {
8     Vector A, B, posC, vitC, vitN, vit1, omega, vit2, vit3,
      vitm;
9     double tvol, sint, cost, f1, f2, vitQCMtc, dvit;
10
11     (*nbAppels) ++;
12
13     // On commence par choisir un temps de vol.
14     tvol = - log (mcmRng ()) / (KAPPA * NEST);
15
16     // Et on calcule les parametres de trajectoire du vol.
17     sint = sin (OMEGA * t);
18     cost = cos (OMEGA * t);
19     A = cost * pos - (sint / OMEGA) * vit;
20     B = sint * pos + (cost / OMEGA) * vit;
21
22     // Si on remonte le temps avant l'instant initial, on
      prend en compte la distribution initiale.
23     if (tvol >= t) {
24         *f = distrib_exacte (A, OMEGA * B, 0.);
25         return;
26     }
27
28     // Sinon, on retrouve la position de depart de la
      particule.
29     sint = sin (OMEGA * (t - tvol));
30     cost = cos (OMEGA * (t - tvol));
31     posC = cost * A + sint * B;
32     vitC = - OMEGA * sint * A + OMEGA * cost * B;
33
34     // On choisit si la particule a subi une collision ou non.
35     if (mcmRng () > .5) {
36         // si la collision est annulee
37         // On choisit une partenaire de collision fantome.
38         tirageGaussien2D (mcmRng (), mcmRng (), &vitN.x, &vitN.y
      );
39         tirageGaussien2D (mcmRng (), mcmRng (), &vitN.z, NULL);
40         vitQCMtc = vitQCM_approx (t - tvol);
41         vit1 = vitQCMtc * vitN;
42         vit1 += vitM_approx (posC, t - tvol);
43
```

```
44      // On va chercher recursivement la fonction de
         distribution.
45      algorithme_ponctuel (posC, vit1, t - tvol, &f1, nbAppels
         );
46      algorithme_ponctuel (posC, vitC, t - tvol, &f2, nbAppels
         );
47      // On calcule un resultat.
48      *f = 2. * f2 * (1. - 15.74960995 /* (2*pi)^1.5 */ * (
         vitQCMtc * vitQCMtc * vitQCMtc) * exp (.5 * vitN.
         LengthSquared ()) * f1 / NEST);
49      return;
50  }
51  else {
52      // si la collision est confirmee
53      // On calcule des directions entrantes.
54      tirageGaussien2D (mcmRng (), mcmRng (), &vitN.x, &vitN.y
         );
55      tirageGaussien2D (mcmRng (), mcmRng (), &vitN.z, NULL);
56      vitQCMtc = vitQCM_approx (t - tvol);
57      vit1 = vitQCMtc * vitN;
58      vit1 += vitM_approx (posC, t - tvol);
59      omega = UniformSampleSphere (mcmRng (), mcmRng ());
60      vitm = .5 * (vitC + vit1);
61      dvit = .5 * (vitC - vit1).Length ();
62      vit2 = vitm + dvit * omega;
63      vit3 = vitm - dvit * omega;
64
65      // On va chercher recursivement la fonction de
         distribution.
66      algorithme_ponctuel (posC, vit2, t - tvol, &f1, nbAppels
         );
67      algorithme_ponctuel (posC, vit3, t - tvol, &f2, nbAppels
         );
68      // On calcule un resultat.
69      *f = 2. * 15.74960995 /* (2*pi)^1.5 */ * (vitQCMtc *
         vitQCMtc * vitQCMtc) * exp (.5 * vitN.LengthSquared
         ()) * f1 * f2 / NEST;
70      return;
71  }
72 }
```

Fichier algo4.cpp (code l'algorithme 6.4) :

```
1 #include "mcmGT2.h"
2 #include "vector.h"
3 #include <math.h>
4 #include "params.h"
```

```
5
6 void Mon_algorithme::algorithme_ponctuel (const Vector & pos
    , const Vector & vit, double t, double * f, long *
    nbAppels)
7 {
8     Vector A, B, posC, vitC, vitN, vit1, vitMtc, omega, vit2,
        vit3, vitm;
9     double tvol, sint, cost, finv, f1, f2, vitQCMtc, dvit, PCA
        , RPN;
10
11     (*nbAppels) ++;
12
13     // On commence par choisir un temps de vol.
14     tvol = - log (mcmRng ()) / (KAPPA * NEST);
15
16     // Et on calcule les parametres de trajectoire du vol.
17     sint = sin (OMEGA * t);
18     cost = cos (OMEGA * t);
19     A = cost * pos - (sint / OMEGA) * vit;
20     B = sint * pos + (cost / OMEGA) * vit;
21
22     // Si on remonte le temps avant l'instant initial, on
        prend en compte la distribution initiale.
23     if (tvol >= t) {
24         *f = distrib_exacte (A, OMEGA * B, 0.);
25         return;
26     }
27
28     // Sinon, on retrouve la position de depart de la
        particule.
29     sint = sin (OMEGA * (t - tvol));
30     cost = cos (OMEGA * (t - tvol));
31     posC = cost * A + sint * B;
32     vitC = - OMEGA * sint * A + OMEGA * cost * B;
33
34     vitQCMtc = vitQCM_approx (t - tvol);
35     vitMtc = vitM_approx (posC, t - tvol);
36     // Puis on calcule une densite approximative en ce point.
37     f1 = dens_approx (posC, t - tvol);
38     // On en deduit une probabilite de collisionner.
39     finv = 1. / NEST;
40     PCA = f1 / NEST;
41     if (PCA > 1. || PCA < 0.) PCA /= 2. * PCA - 1.;
42
43     // On choisit si la particule a subi une collision ou non.
44     if (mcmRng () > PCA) {
45         // si la collision est annulee
```



```
46      // On choisit une direction entrante.
47      tirageGaussien2D (mcmRng (), mcmRng (), &vitN.x, &vitN.y
48      );
49      tirageGaussien2D (mcmRng (), mcmRng (), &vitN.z, NULL);
50      vit1 = vitQCMtc * vitN;
51      vit1 += vitMtc;
52      // On va chercher recursivement la fonction de
53      // distribution dans la direction entrante d'une
54      // partenaire de collision fantome, pour corriger l'
55      // approximation faite.
56      algorithme_ponctuel (posC, vit1, t - tvol, &f2, nbAppels
57      );
58      RPN = (1. - 15.74960995 /* (2*pi)^1.5 */ * (vitQCMtc *
59      vitQCMtc * vitQCMtc) * exp (.5 * vitN.LengthSquared
60      ()) * f2 / NEST) / (1. - PCA);
61      // On va chercher recursivement la fonction de
62      // distribution.
63      algorithme_ponctuel (posC, vitC, t - tvol, &f2, nbAppels
64      );
65      // On calcule un resultat.
66      *f = RPN * f2 ;
67      return;
68  }
69  else {
70      // si la collision est confirmee
71      // On calcule des directions entrantes.
72      tirageGaussien2D (mcmRng (), mcmRng (), &vitN.x, &vitN.y
73      );
74      tirageGaussien2D (mcmRng (), mcmRng (), &vitN.z, NULL);
75      vit1 = vitQCMtc * vitN;
76      vit1 += vitMtc;
77      omega = UniformSampleSphere (mcmRng (), mcmRng ());
78      vitm = .5 * (vitC + vit1);
79      dvit = .5 * (vitC - vit1).Length ();
80      vit2 = vitm + dvit * omega;
81      vit3 = vitm - dvit * omega;
82
83      // On va chercher recursivement la fonction de
84      // distribution.
85      algorithme_ponctuel (posC, vit2, t - tvol, &f1, nbAppels
86      );
87      algorithme_ponctuel (posC, vit3, t - tvol, &f2, nbAppels
88      );
89      // On calcule un resultat.
90      *f = 15.74960995 /* (2*pi)^1.5 */ * (vitQCMtc * vitQCMtc
91      * vitQCMtc) * exp (.5 * vitN.LengthSquared ()) * f1
92      * f2 / (NEST * PCA);
```

```
78     return;
79 }
80 }
```

Fichier algo5.cpp (code l'algorithme 6.5) :

```
1 #include "mcmGT2.h"
2 #include "vector.h"
3 #include <math.h>
4 #include "params.h"
5
6 void Mon_algorithme::algorithme_ponctuel (const Vector & pos
      , const Vector & vit, double t, double * f, long *
      nbAppels)
7 {
8     Vector A, B, posC, vitC, vitN, vit1, omega, vit2, vit3,
      vitm;
9     double tvol, sint, cost, f1, f2, f3, vitQCMtc, dvit, PCE,
      PCA, RPC, RPN;
10
11     (*nbAppels) ++;
12
13     // On commence par choisir un temps de vol.
14     tvol = - log (mcmRng ()) / (KAPPA * NEST);
15
16     // Et on calcule les parametres de trajectoire du vol.
17     sint = sin (OMEGA * t);
18     cost = cos (OMEGA * t);
19     A = cost * pos - (sint / OMEGA) * vit;
20     B = sint * pos + (cost / OMEGA) * vit;
21
22     // Si on remonte le temps avant l'instant initial, on
      prend en compte la distribution initiale.
23     if (tvol >= t) {
24         *f = distrib_exacte (A, OMEGA * B, 0.);
25         return;
26     }
27
28     // Sinon, on retrouve la position de depart de la
      particule.
29     sint = sin (OMEGA * (t - tvol));
30     cost = cos (OMEGA * (t - tvol));
31     posC = cost * A + sint * B;
32     vitC = - OMEGA * sint * A + OMEGA * cost * B;
33
34     // Puis on choisit une partenaire de collision eventuelle.
35     tirageGaussien2D (mcmRng (), mcmRng (), &vitN.x, &vitN.y);
```

```
36  tirageGaussien2D (mcmRng (), mcmRng (), &vitN.z, NULL);
37  vitQCMtc = vitQCM_approx (t - tvol);
38  vit1 = vitQCMtc * vitN;
39  vit1 += vitM_approx (posC, t - tvol);
40
41  // On va chercher recursivement la fonction de
    distribution.
42  algorithme_ponctuel (posC, vit1, t - tvol, &f1, nbAppels);
43  // On en deduit une probabilite de collisionner.
44  PCE = 15.74960995 /* (2*pi)^1.5 */ * (vitQCMtc * vitQCMtc
    * vitQCMtc) * exp (.5 * vitN.LengthSquared ()) * f1 /
    NEST;
45  if (PCE > 1. || PCE < 0.) {
46      PCA = PCE / (2. * PCE - 1.);
47      RPN = 1. - 2. * PCE;
48      RPC = - RPN;
49  }
50  else {
51      PCA = PCE;
52      RPN = 1.;
53      RPC = 1.;
54  }
55
56  // On choisit si la particule a subi une collision ou non.
57  if (mcmRng () > PCA) {
58      // si la collision est annulee
59      // On va chercher recursivement la fonction de
        distribution.
60      algorithme_ponctuel (posC, vitC, t - tvol, &f2, nbAppels
        );
61      // On calcule un resultat.
62      *f = RPN * f2 ;
63      return;
64  }
65  else {
66      // si la collision est confirmee
67      // On calcule des directions entrantes, ici en
        reutilisant la direction de la partenaire de
        collision.
68      omega = UniformSampleSphere (mcmRng (), mcmRng ());
69      vitm = .5 * (vitC + vit1);
70      dvit = .5 * (vitC - vit1).Length ();
71      vit2 = vitm + dvit * omega;
72      vit3 = vitm - dvit * omega;
73
74      // On va chercher recursivement la fonction de
        distribution.
```

```
75     algorithme_ponctuel (posC, vit2, t - tvol, &f2, nbAppels
76         );
77     algorithme_ponctuel (posC, vit3, t - tvol, &f3, nbAppels
78         );
79     // On calcule un resultat.
80     *f = RPC * f2 * f3 / f1;
81     return;
82 }
```

Fichier algo6.cpp (code l'algorithme 6.6) :

```
1  #include "mcmGT2.h"
2  #include "vector.h"
3  #include <math.h>
4  #include "params.h"
5
6  void Mon_algorithme::algorithme_ponctuel (const Vector & pos
7      , const Vector & vit, double t, double * f, long *
8      nbAppels)
9  {
10     Vector A, B, posC, vitC, vitN, vit1, vitMtc, omega, vit2,
11         vit3, vitm;
12     double tvol, sint, cost, f1, f2, vitQCMtc, dvit, PCE, PCA,
13         RPN;
14
15     (*nbAppels) ++;
16
17     // On commence par choisir un temps de vol.
18     tvol = - log (mcmRng ()) / (KAPPA * NEST);
19
20     // Et on calcule les parametres de trajectoire du vol.
21     sint = sin (OMEGA * t);
22     cost = cos (OMEGA * t);
23     A = cost * pos - (sint / OMEGA) * vit;
24     B = sint * pos + (cost / OMEGA) * vit;
25
26     // Si on remonte le temps avant l'instant initial, on
27     // prend en compte la distribution initiale.
28     if (tvol >= t) {
29         *f = distrib_exacte (A, OMEGA * B, 0.);
30         return;
31     }
32
33     // Sinon, on retrouve la position de depart de la
34     // particule.
35     sint = sin (OMEGA * (t - tvol));
```

```
30     cost = cos (OMEGA * (t - tvol));
31     posC = cost * A + sint * B;
32     vitC = - OMEGA * sint * A + OMEGA * cost * B;
33
34     // Puis on choisit une partenaire de collision eventuelle.
35     tirageGaussien2D (mcmRng (), mcmRng (), &vitN.x, &vitN.y);
36     tirageGaussien2D (mcmRng (), mcmRng (), &vitN.z, NULL);
37     vitQCMtc = vitQCM_approx (t - tvol);
38     vit1 = vitQCMtc * vitN;
39     vitMtc = vitM_approx (posC, t - tvol);
40     vit1 += vitMtc;
41
42     // On va chercher recursivement la fonction de
43     distribution.
44     algorithme_ponctuel (posC, vit1, t - tvol, &f1, nbAppels);
45     // On en deduit une probabilite de collisionner.
46     PCE = 15.74960995 /* (2*pi)^1.5 */ * (vitQCMtc * vitQCMtc
47         * vitQCMtc) * exp (.5 * vitN.LengthSquared ()) * f1 /
48         NEST;
49     if (PCE > 1. || PCE < 0.) {
50         PCA = PCE / (2. * PCE - 1.);
51         RPN = 1. - 2. * PCE;
52     }
53     else {
54         PCA = PCE;
55         RPN = 1.;
56     }
57
58     // On choisit si la particule a subi une collision ou non.
59     if (mcmRng () > PCA) {
60         // si la collision est annulee
61         // On va chercher recursivement la fonction de
62         distribution.
63         algorithme_ponctuel (posC, vitC, t - tvol, &f2, nbAppels
64             );
65         // On calcule un resultat.
66         *f = RPN * f2 ;
67         return;
68     }
69     else {
70         // si la collision est confirmee
71         // On calcule des directions entrantes.
72         tirageGaussien2D (mcmRng (), mcmRng (), &vitN.x, &vitN.y
73             );
74         tirageGaussien2D (mcmRng (), mcmRng (), &vitN.z, NULL);
75         vit1 = vitQCMtc * vitN;
76         vit1 += vitMtc;
```

```
71     omega = UniformSampleSphere (mcmRng (), mcmRng ());
72     vitm = .5 * (vitC + vit1);
73     dvit = .5 * (vitC - vit1).Length ();
74     vit2 = vitm + dvit * omega;
75     vit3 = vitm - dvit * omega;
76
77     // On va chercher recursivement la fonction de
78     // distribution.
79     algorithme_ponctuel (posC, vit2, t - tvol, &f1, nbAppels
80     );
81     algorithme_ponctuel (posC, vit3, t - tvol, &f2, nbAppels
82     );
83     // On calcule un resultat.
84     *f = 15.74960995 /* (2*pi)^1.5 */ * (vitQCMtc * vitQCMtc
85     * vitQCMtc) * exp (.5 * vitN.LengthSquared ()) * f1
86     * f2 / (NEST * PCA);
87     return;
88 }
89 }
```


Bibliographie

- [1] N. ANDREUCCI : Adjoint Monte-Carlo method with fictitious scattering in deep penetration and long-distance detector calculations. *In Lecture Notes in Physics*, page 291, 1985.
- [2] Pierre ANDRIES, Patrick LE TALLEC, Jean-Philippe PERLAT et Benoît PERTHAME : Gaussian-BGK model of Boltzmann equation with small Prandtl number. *European Journal of Mechanics, B/Fluids*, 19(6):813–830, 2000.
- [3] A. V. ANIKEENKO, N. N. MEDVEDEV, M. K. KOVALEV et M. S. MELGUNOV : Simulation of gaz diffusion in porous layers of varying structure. *Zhurnal Strukturnoi Khimii*, 50(3):425–432, 2009.
- [4] Roland ASSARAF et Michel CAFFAREL : Zero-variance principle for Monte Carlo algorithms. *Physical Review Letters*, 83(23):4682, 1999.
- [5] Andreu BADAL et Aldo BADANO : Monte Carlo Simulation of X-Ray Imaging Using a Graphics Processing Unit. *In Nuclear Science Symposium Conference*, pages 4081 – 4084, Orlando, 2009. IEEE.
- [6] J. N. BARDSLEY et S. L. LIN : The null-event method in computer simulation. *Computer Physics Communications*, 15:161–163, 1978.
- [7] Germain BAUD : *Conception de récepteurs solaires à lit fluidisé sous flux radiatif concentré*. Thèse de doctorat, Institut National Polytechnique de Toulouse, 2011.
- [8] P. L. BHATNAGAR, E. P. GROSS et Max KROOK : A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. *Physical Review*, 94(3):511–525, 1954.
- [9] Graeme Austin BIRD : *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Clarendon, 1994.
- [10] A. D. BOARDMAN, W. FAWCETT et H. D. REES : Monte Carlo calculation of the velocity-field relationship for gallium arsenide. *Solid State Communications*, 6(2):305–307, 1968.
- [11] Alexander V. BOBYLEV : One class of invariant solutions of the Boltzmann equation. *Doklady Akademii nauk SSSR*, 231:571–574, 1976.
- [12] Alexander V. BOBYLEV et Carlo CERCIGNANI : Exact Eternal Solutions of the Boltzmann Equation. *Journal of Statistical Physics*, 106(March):1019–1038, 2002.

- [13] Alexander V. BOBYLEV et Carlo CERCIGNANI : Self-Similar Solutions of the Boltzmann Equation and Their Applications. *Journal of Statistical Physics*, 106(March):1039–1071, 2002.
- [14] Jean-Pierre BOEUF et E. MARODE : A Monte Carlo analysis of an electron swarm in a nonuniform field : the cathode region of a glow discharge in helium. *Journal of Physics D : Applied Physics*, 15:2169 – 2187, 1982.
- [15] Matthieu BOFFETY : *Étude quantitative de la tomographie optique diffuse de luminescence. Application à la localisation de sources en imagerie moléculaire*. Thèse de doctorat, Ecole Centrale Paris, 2010.
- [16] BOGOLIUBOV : Problems of Dynamical Theory in Statistical Physics. *J. Phys.(USSR)*, 10:256, 1946.
- [17] Thomas E. BOOTH : Zero-variance solutions for linear Monte Carlo. *Nuclear Science and Engineering*, 102(4):332—340, 1989.
- [18] M. BORN et H. S. GREEN : *A General Kinetic Theory of Fluids*. Cambridge university press, Cambridge, 1949.
- [19] François BOUCHUT : Existence and Uniqueness of a Global Smooth Solution for the Vlasov-Poisson-Fokker-Planck System in Three Dimensions. *Journal of Functional Analysis*, 111:239–258, 1993.
- [20] M. J. BRENNAN : Optimization of Monte Carlo codes using null collision techniques for experimental simulation at low E/N. *IEEE Transactions on Plasma Science*, 19(2):256 – 261, 1991.
- [21] N. R. CANDELORE et R. C. GAST : A Monte Carlo program for estimating epithermal capture rates in rod arrays. Rapport technique, Westinghouse Electric Corp. Bettis Atomic Power Lab., Pittsburg, 1964.
- [22] Carlo CERCIGNANI : *The Boltzmann Equation and Its Applications*. Springer-Verlag, New York, 1988.
- [23] M. CHALLET, Vincent FOURCASSIÉ, Stéphane BLANCO, Richard FOURNIER, Guy THERAULAZ et Christian JOST : A new test of random walks in heterogeneous environments. *Die Naturwissenschaften*, 92(8):367–370, août 2005.
- [24] Subrahmanyan CHANDRASEKHAR : *Radiative transfer*. Dover Publications, 1960.
- [25] Sydney CHAPMAN et Thomas George COWLING : *The mathematical theory of non-uniform gases : an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases*. Cambridge university press, 3rd édition, 1970.
- [26] W. A. COLEMAN : Mathematical Verification of a Certain Monte Carlo Sampling Technique and Applications of the Technique to Radiation Transport Problems. *Nuclear Science and Engineering*, 32(1):76 – 81, 1968.

- [27] H. CORNILLE et A. GERVOIS : Powerlike Decreasing Solutions of the Boltzmann Equation for a Maxwell Gas. *Journal of Statistical Physics*, 26(1):181–217, 1981.
- [28] John H. CURTISS : Monte-Carlo methods for the iteration of linear operators. *Journal of Mathematics and Physics*, 32(4):209–232, 1954.
- [29] Jérémie DAUCHET : *Analyse radiative des photobioréacteurs*. Thèse de doctorat, Université Blaise Pascal - Clermont-Ferrand II, 2012.
- [30] Jérémie DAUCHET, Stéphane BLANCO, Jean-François CORNET et Richard FOURNIER : Calculation of the radiative properties of photosynthetic microorganisms. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 161:60–84, 2015.
- [31] Jérémie DE LA TORRE : *Calculs de sensibilités par méthode de Monte-Carlo, pour la conception de procédés à énergie solaire concentrée*. Thèse de doctorat, Institut National Polytechnique de Toulouse, 2011.
- [32] Jérémie DE LA TORRE, Germain BAUD, Jean-Jacques BÉZIAN, Stéphane BLANCO, Cyril CALIOT, Jean-François CORNET, Christophe COUSTET, Jérémie DAUCHET, Mouna EL HAFI, Vincent EYMET, Richard FOURNIER, Jacques GAUTRAIS, O. GOURMEL, David JOSEPH, Nicolas MEILHAC, A. PAJOT, Mathias PAULIN, Patrice PEREZ, Benjamin PIAUD, Maxime ROGER, Julien Yves ROLLAND, François VEYNANDT et Sebastian WEITZ : Monte Carlo advances and concentrated solar applications. *Solar Energy*, 103:653–681, 2014.
- [33] Amaury de Guilhem DE LATAILLADE : *Modélisation détaillée des transferts radiatifs et couplage avec la cinétique chimique dans des systèmes en combustion*. Thèse de doctorat, 2001.
- [34] Amaury de Guilhem DE LATAILLADE, Stéphane BLANCO, Y. CLERGENT, Jean-Louis DUFRESNE, Mouna EL HAFI et Richard FOURNIER : Monte Carlo method and sensitivity estimations. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 75(5):529–538, 2002.
- [35] Lawrence T. DECARLO : On the meaning and use of kurtosis. *Psychological Methods*, 2(3):292–307, 1997.
- [36] Ivan T. DIMOV : *Monte Carlo Methods for Applied Scientists*. 2008.
- [37] Ivan T. DIMOV et Todor V. GUROV : Monte Carlo algorithm for solving integral equations with polynomial non-linearity. Parallel implementation. *Pliska Studia Mathematica Bulgarica*, 13:117–132, 2000.
- [38] William L. DUNN et J. Kenneth SHULTIS : *Exploring Monte Carlo Methods*. Elsevier, 2012.
- [39] M. H. ERNST : Exact solution of the non-linear Boltzmann equation for Maxwell Models. *Physics Letters*, 69(6):390–392, 1979.
- [40] M. H. ERNST : Nonlinear model-Boltzmann equations and exact solutions. 78(1):171, 1981.

- [41] M. H. ERNST et R. BRITO : High-energy tails for inelastic Maxwell models. *Europhysics Letters*, 58(2):182–187, 2002.
- [42] Vincent EYMET : *Simulation Monte-Carlo et analyse en Puissances Nettes Échangées des transferts radiatifs infrarouges avec diffusion : vers une paramétrisation dans un Modèle de Circulation Générale atmosphérique*. Thèse de doctorat, Université Paul Sabatier, 2003.
- [43] Vincent EYMET, Damien POITOU, Mathieu GALTIER, Mouna EL HAFI, Guillaume TERRÉE et Richard FOURNIER : Null-collision meshless Monte-Carlo—Application to the validation of fast radiative transfer solvers embedded in combustion simulators. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 129:145–157, novembre 2013.
- [44] Olivier FARGES : *Conception optimale de centrales solaires à concentration : application aux centrales à tour et aux installations « beam down »*. Thèse de doctorat, Institut National Polytechnique de Toulouse - École Nationale Supérieure des Mines d’Albi-Carmaux, 2014.
- [45] Olivier FARGES, Jean-Jacques BÉZIAN, H. BRU, Mouna EL HAFI, Richard FOURNIER et Christophe SPIESSER : Life-time integration using Monte Carlo Methods when optimizing the design of concentrated solar power plants. *Solar Energy*, 113:57–62, 2015.
- [46] W. FAWCETT, A. D. BOARDMAN et S. SWAIN : Monte Carlo determination of electron transport properties in gallium arsenide. *Journal of Physics and Chemistry of Solids*, 31:1963–1990, 1970.
- [47] Benoit FORESTIER, Joachim MISS, Franck BERNARD, Aurélien DORVAL, Olivier JACQUET et Bernard VERBOOMEN : Criticality calculations on pebble-bed HTR-PROTEUS configuration as a validation for the pseudo-scattering tracking method implemented in the MORET 5 Monte Carlo code. *In International Conference on the Physics of Reactors*, page 8, 2008.
- [48] Richard FOURNIER : *Rayonnement thermique dans les gaz : analyse du couplage avec la convection naturelle*. Thèse de doctorat, 1994.
- [49] Mathieu GALTIER : *Approche statistique du rayonnement dans les milieux gazeux hétérogènes : de l’échantillonnage des transitions moléculaires au calcul de grandeurs radiatives*. Thèse de doctorat, Institut National Polytechnique de Toulouse - École Nationale Supérieure des Mines d’Albi-Carmaux, 2014.
- [50] Mathieu GALTIER, Stéphane BLANCO, Cyril CALIOT, Christophe COUSTET, Jérémie DAUCHET, Mouna EL HAFI, Vincent EYMET, Richard FOURNIER, Jacques GAUTRAIS, Anaïs KHUONG, Benjamin PIAUD et Guillaume TERRÉE : Integral formulation of null-collision Monte Carlo algorithms. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 125:57–68, août 2013.
- [51] A. J. M. GARRET : Successive derivatives of H for a solution of Boltzmann’s equation do not alternate. *Journal of Physics A : Mathematical and General*, 15:L351–L355, 1982.

- [52] R. GOODY et Y. YUNG : *Atmospheric radiation : theoretical basis*. Oxford University Press, 1996.
- [53] David GUÉRY-ODELIN, J. G. MUGA, M. J. RUIZ-MONTERO et E. TRIZAC : Nonequilibrium solutions of the Boltzmann equation under the action of an external force. *Physical Review Letters*, 112(May):1–5, 2014.
- [54] J.M. HAMMERSLEY et D.C. HANDSCOMB : *Monte Carlo Methods*. Fletcher & Son Ltd, 1964.
- [55] Theodore E. HARRIS : *The Theory of Branching Processes*. The RAND Corporation, 1964.
- [56] E. H. HAUGE et E. PRÆ STGAARD : The Bobylev Approach to the Nonlinear Boltzmann Equation. *Journal of Statistical Physics*, 24(1):21–38, 1981.
- [57] D. HEIFETZ, D. POST, M. PETRAVIC, J. WEISHEIT et G. BATEMAN : A Monte-Carlo Model of Neutral-Particle in Diverted Plasmas. *Journal of Computational Physics*, 46:309–327, 1982.
- [58] Jr. HOLWAY et H. LOWELL : Kinetic theory of shock structure using an ellipsoidal distribution function. *In Proceedings of the Fourth International Symposium held at the Institute for Aerospace Studies*, page 193, Toronto, 1965.
- [59] John R. HOWELL, Robert SIEGEL et M. PINAR MENGÜÇ : *Thermal Radiation, Heat Transfer*. CRC Press, 5 édition, 2011.
- [60] Raphaël JEANSON, Stéphane BLANCO, Richard FOURNIER, Jean Louis DENEUBOURG, Vincent FOURCASSIÉ et Guy THERAULAZ : A model of animal movements in a bounded space. *Journal of Theoretical Biology*, 225(4):443–451, décembre 2003.
- [61] Mark E. JOHNSON et Victor W. LOWE : Bounds on the Sample Skewness and Kurtosis. *Technometrics*, 21(3):377–378, 1979.
- [62] David JOSEPH : *Modélisation des transferts radiatifs en combustion par Méthode aux Ordonnées Discrètes sur des maillages non structurés tridimensionnels*. Thèse de doctorat, Institut National Polytechnique de Toulouse, 2004.
- [63] M. KAC : Foundations of Kinetic Theory. *In Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, volume 3, pages 171–197, Berkeley, Los Angeles, 1956. University of California Press.
- [64] Malvin H. KALOS et Paula A. WHITLOCK : *Monte carlo methods*. John Wiley & Sons, 2008.
- [65] Iwan KAWRAKOW et Matthias FIPPEL : Investigation of variance reduction techniques for Monte Carlo photon dose calculation using XVMC. *Physics in medicine and biology*, 45:2163–2183, 2000.
- [66] Alfred I. KHISAMUTDINOV : On connection between "Continuous time" and "Direct simulation" Monte Carlo methods for the Boltzmann equation and some new approximate methods.

- [67] Alfred I. KHISAMUTDINOV : On Development of Continuous Time Monte Carlo Methods for Problems of Boltzmann Equation with External Forces. *Transport Theory and Statistical Physics*, 33(1):69–89, 2004.
- [68] Alfred I. KHISAMUTDINOV et L. L. SIDORENKO : Monte Carlo fictitious collision algorithms for nonlinear Boltzmann equation. *Monte Carlo Methods and Applications*, 1(3):221–240, 1995.
- [69] Anaïs KHUONG : *Modèle comportemental de la dynamique de construction de la structure épigée du nid chez la fourmi Lasius niger. Approches expérimentales et théoriques*. Thèse de doctorat, Université Paul Sabatier - Toulouse III, 2013.
- [70] John G. KIRKWOOD : The Statistical Mechanical Theory of Transport Processes. I. General Theory. *The Journal of Chemical Physics*, 14(3):180–201, 1946.
- [71] John G. KIRKWOOD : The Statistical Mechanical Theory of Transport Processes II. Transport in Gases. *The Journal of Chemical Physics*, 15(1):72, 1947.
- [72] Katsuhisa KOURA : Improved Null-Collision Technique in the Direct Simulation Monte Carlo Method : Application to Vibrational Relaxation of Nitrogen. *Computers Math. Applic.*, 35(1):139–154, 1998.
- [73] Katsuhisa KOURA : Monte Carlo Direct (Test-Particle) Simulation of Rotational and Vibrational Relaxation and Dissociation of Diatomic Molecules Using Classical Trajectory Calculations. In T. J. BARTEL et M. A. GALLIS, éditeurs : *Rarefied Gas Dynamics : 22nd International Symposium*, pages 7–16. American Institute of Physics, 2001.
- [74] Max KROOK et Tai Tsun WU : Formation of Maxwellian Tails. *Physical Review Letters*, 36(19):1107–1109, 1976.
- [75] Jaakko LEPPÄNEN : *Development of a New Monte Carlo Reactor Physics Code*. 2007.
- [76] Jaakko LEPPÄNEN : Randomly dispersed particle fuel model in the PSG monte carlo neutron transport code. In *Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications*, page 12, 2007.
- [77] Jaakko LEPPÄNEN : Performance of Woodcock delta-tracking in lattice physics applications using the Serpent Monte Carlo reactor physics burnup calculation code. *Annals of Nuclear Energy*, 37(5):715–722, mai 2010.
- [78] Savino LONGO : Monte Carlo models of electron and ion transport in non-equilibrium plasmas. *Plasma Sources Science and Technology*, 9(4):468–476, 2000.
- [79] Savino LONGO : Direct derivation of Skullerud’s Monte Carlo method for charged particle transport from the linear Boltzmann equation. *Physica A*, 313:389–396, 2002.

- [80] Savino LONGO et P. DIOMEDE : Monte Carlo modeling of gas phase ion transport under thermal gradients and external fields. *The European Physical Journal : Applied Physics*, 26:177–185, 2004.
- [81] D. B. MACMILLAN : Note on a Paper by J. Spanier Concerning Monte Carlo Estimators. *SIAM Journal on Applied Mathematics*, 15(2):264–268, 1967.
- [82] William R. MARTIN et Forrest B. BROWN : Comparison of Monte Carlo methods for nonlinear radiation transport. Rapport technique September, Salt Lake City, 2001.
- [83] Stéphane MISCHLER et Clément MOUHOT : About Kac’s program in kinetic theory. *C. R. Acad. Sci. Paris, Ser. I*, 349:1245–1250, 2011.
- [84] Stéphane MISCHLER et Clément MOUHOT : Kac’s program in kinetic theory. *Inventiones Mathematicae*, 193(1):1–147, 2013.
- [85] Joachim MISS, Olivier JACQUET, Benoit FORESTIER, Wim HAECK et Bernard VERBOOMEN : Feasibility of using a universal point wise energy grid or a very fine multi-group energy grid in MORET5 criticality calculations. *In Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications*, page 11, 2007.
- [86] M. MODEST : *Radiative heat transfer*. Academic Press, 3 édition, 2013.
- [87] Bruno MONTCEL : *Tomographie optique diffuse résolue en temps : Applications fonctionnelles en neurosciences*. Thèse de doctorat, 2005.
- [88] M. NEDJALKOV, Todor V. GUROV et Ivan T. DIMOV : Statistical modeling of pulse excited electron quantum kinetics in a one-band semiconductor. *Mathematics and Computers in Simulation*, 47:391–402, 1998.
- [89] Jan NOVÁK, Andrew SELLE et Wojciech JAROSZ : Residual ratio tracking for estimating attenuation in participating media. *ACM Transactions on Graphics*, 33(6):11, novembre 2014.
- [90] Patrice PÉREZ : *Algorithmes de synthèse d’images et propriétés spectrales des gaz de combustion : méthode de Monte Carlo pour la simulation des transferts radiatifs dans les procédés à haute température*. Thèse de doctorat, Institut National Polytechnique de Toulouse, 2003.
- [91] Matt PHARR et Greg HUMPHREYS : *Physically based rendering : From theory to implementation*. Morgan Kaufmann Publishers Inc., 2004.
- [92] Matt PHARR et Greg HUMPHREYS : *Physically Based Rendering, Second Edition : From Theory To Implementation*. Morgan Kaufmann Publishers Inc., San Fransisco, 2 édition, 2010.
- [93] Benjamin PIAUD : *Modélisation mésoscopique des écoulements avec changement de phase à partir de l’équation de Boltzmann-Enskog. Introduction des effets thermiques*. Thèse de doctorat, Université Paul Sabatier - Toulouse III, 2007.

- [94] Benjamin PIAUD, Stéphane BLANCO, Richard FOURNIER, Victor Eugen AMBRUŞ et Victor SOFONEA : Gauss Quadratures – the Keystone of Lattice Boltzmann Models. *International Journal of Modern Physics C*, 25(01): 1340016, 2014.
- [95] Damien POITOU : *Modélisation du rayonnement dans la simulation aux grandes échelles de la combustion turbulente*. Thèse de doctorat, Institut National Polytechnique de Toulouse, 2009.
- [96] Olivier PUJOL : Comment on the (misused) concept of photon in radiative transfer, and proposition of a neologism. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 159:29–31, 2015.
- [97] H. D. REES : Calculation of distribution functions by exploiting the stability of the steady state. *Solid State Communications*, 6:643–655, 1968.
- [98] H. D. REES : Calculation of steady state distribution functions by exploiting stability. *Physics Letters A*, 26(9):416–417, 1968.
- [99] Niklas S. REHFELD, Simon STUTE, Marine SORET, John APOSTOLAKIS et Irène BUVAT : Optimization of photon tracking in GATE. In *Nuclear Science Symposium/Medical Imaging Conference*, pages 3288 – 3290, Dresden, 2008. IEEE.
- [100] S. RJASANOW et W. WAGNER : A Generalized Collision Mechanism for Stochastic Particle Schemes Approximating Boltzmann-Type Equations. *Computers & Mathematics with Applications*, 35:165–178, 1998.
- [101] Maxime ROGER : *Modèles de sensibilité dans le cadre de la méthode de Monte-Carlo : illustrations en transfert radiatif*. Thèse de doctorat, Institut National Polytechnique de Toulouse, 2006.
- [102] Maxime ROGER, Stéphane BLANCO, Mouna EL HAFI et Richard FOURNIER : Monte Carlo estimates of domain-deformation sensitivities. *Physical Review Letters*, 95(October):1–4, 2005.
- [103] Maxime ROGER, Cyril CALIOT, Nicolas CROUSEILLES et Pedro Jorge Martins COELHO : A hybrid transport-diffusion model for radiative transfer in absorbing and scattering media. *Journal of Computational Physics*, 275:346–362, 2014.
- [104] Julien Yves ROLLAND : *Étude statistique des chemins de premier retour aux nombres de Knudsen intermédiaires : De la simulation par méthode de Monte Carlo à l'utilisation de l'approximation de diffusion*. Thèse de doctorat, Institut National Polytechnique de Toulouse, 2009.
- [105] L. I. SCHIFF : Approximation Method for High-Energy Potential Scattering. *Physical Review*, 103(2):443–453, 1956.
- [106] J. M. SELIER et Ivan T. DIMOV : The Wigner-Boltzmann Monte Carlo method applied to electron transport in the presence of a single dopant. *Computer Physics Communications*, 185(10):2427–2435, 2014.

- [107] E. M. SHAKHOV : Generalization of the Krook Kinetic Relaxation Equation. *Mekhanika Zhidkosti i Gaza*, 3(5):95–96, 1968.
- [108] H. R. SKULLERUD : The stochastic computer simulation of ion motion in a gas subjected to a constant electric field. *Journal of Physics D : Applied Physics*, 1(2):1567–1568, 1968.
- [109] Victor SOFONEA et Robert F. SEKERKA : BGK models for diffusion in isothermal binary fluid systems. *Physica A : Statistical Mechanics and its Applications*, 299(3-4):494–520, 2001.
- [110] Victor SOFONEA et Robert F. SEKERKA : Viscosity of finite difference lattice Boltzmann models. *Journal of Computational Physics*, 184(2):422–434, 2003.
- [111] Victor SOFONEA et Robert F. SEKERKA : Boundary conditions for the upwind finite difference Lattice Boltzmann model : Evidence of slip velocity in micro-channel flow. *Journal of Computational Physics*, 207(2):639–659, 2005.
- [112] Jerome SPANIER : Two Pairs of Families of Estimators for Transport Problems. *SIAM Journal on Applied Mathematics*, 14(4):702–713, 1966.
- [113] Jerome SPANIER : An Analytic Approach to Variance Reduction. *SIAM Journal on Applied Mathematics*, 18(1):172–190, 1970.
- [114] Jerome SPANIER et Ely M. GELBARD : *Monte Carlo Principles and Neutron Transport Problems*. Addison-Wesley Publishing Company, 1969.
- [115] László SZIRMAY-KALOS, Balázs TÓTH et Milán MAGDICS : Efficient Free Path Sampling in Inhomogeneous Media. In *Eurographics*, page 2, 2010.
- [116] László SZIRMAY-KALOS, Balázs TÓTH et Milán MAGDICS : Free Path Sampling in High Resolution Inhomogeneous Participating Media. In *Computer Graphics Forum*, volume 0, page 12, 2010.
- [117] Jean TAINE et Anouar SOUFIANI : Gas IR Radiative Properties : From Spectroscopic Data to Approximate Models. *Advances in Heat Transfer*, 33(C):295–414, 1999.
- [118] Guillaume TERRÉE, Stéphane BLANCO, Mouna EL HAFI, Richard FOURNIER et Julien Yves ROLLAND : Diffusion approximation and short-path statistics at low to intermediate Knudsen numbers. *Europhysics Letters*, 110(2):20007, 2015.
- [119] Behrouz Fathi VAJARGAH et Mojtaba MORADI : Three classifications on branching processes and their behavior for finding the solution of nonlinear integral equations. *Mathematical Modelling and Analysis*, 15(3):371–381, 2010.
- [120] Hendrik Christoffel van de HULST : *Light Scattering by Small Particles*. Dover Publications, New York, 1981.
- [121] Lihong V. WANG, Robert E. NORDQUIST et Wei R. CHEN : Optimal beam size for light delivery to absorption-enhanced tumors buried in biological

- tissues and effect of multiple-beam delivery : a Monte Carlo study. *Applied Optics*, 36(31):8286–8291, novembre 1997.
- [122] E. WOODCOCK, T. MURPHY, P. HEMMINGS et S LONGWORTH : Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. *In Applications of Computing Methods to Reactor Problems*, 1965.
- [123] QingHua XIAO, LinJie XIONG et HuiJiang ZHAO : The Vlasov-Poisson-Boltzmann system for non-cutoff hard potentials. *Science China Mathematics*, 57(3):515–540, 2014.
- [124] Jacques YVON : La théorie statistique des fluides et l'équation d'état. *Actualités Scientifiques et Industrielles*, 203, 1935.
- [125] Y. ZHENG et H. STRUCHTRUP : Burnett equations for the ellipsoidal statistical BGK model. *Continuum Mechanics and Thermodynamics*, 16(1-2):97–108, 2004.

Monte-Carlo method and non-linearities : from radiative transfer physics to gas kinetics

Abstract : In transport physics, especially in radiative transfer physics, the Monte-Carlo method has been originally developed as the simulation of the history of numerous particles, from which are deduced mean observables. This numerical method owes its success to several qualities : a natural management of many-dimensional phase space, a null systematic error away from the mathematical and physical model, the confidence intervals given with the results, an ability to take into account simultaneously numerous physical phenomenons, the simultaneous sensitivities calculating possibility, and an easy parallelization.

In gas kinetics, particles collide each other, not with an external fixed medium ; it is said that their transport is non-linear. These mutual collisions put out of action the aforesaid approach of the Monte-Carlo method ; because in order to simulate the independent trajectories of multiple particles and thus estimate their distribution, this distribution must beforehand be exactly known. . .

This thesis follows on from those of Jérémy DAUCHET (2012) and of Mathieu GALTIER (2014), dedicated to radiative transfer physics. Between other works, these authors have shown how the Monte-Carlo method can bear non-linearities, while keeping its customary formalism and specificities. The then overcome non-linearities were respectively a chemistry/irradiance coupling law, and the dependence of the irradiance toward the absorption coefficient. We try in this manuscript to overcome the non-linearity of the transport. In this aim, our main tools are a reverse following of particles, based on integral formulations of the transport equations, formulations largely inspired from the so-called null collisions algorithms.

We show, on several academic examples, that we have indeed extended the Monte Carlo method to the resolution of the Boltzmann equation. These examples are also occasions to test the limits of what we have built. The most noteworthy results are certainly the absence of any mesh in the numerical method, and its capacity to calculate correctly the high-speed particles quantities (always rare compared to the total, in gas kinetics).

Beyond the given examples, this manuscript is wanted as a formalism attempt and an exploration of the developed method basics. The focus is made on the reasoning leading to the method, rather than on particular implementations which have been realized. In the eyes of the author, the method is still largely reworkable. In particular, the maximal times on which the evolution of particles is computable, which constitute the main weakness of the developed numerical method, can surely be increased.

Keywords : Monte-Carlo method, Gas kinetics, Radiative transfer, Boltzmann equation, Non-linear transport, Integral formulation

Abstract (intended to the general public) : The subject is the extension of a numerical computing method, named the Monte-Carlo method, from the radiative transfer physics (the study of heat transfer by the light, or the image synthesis) to the gas kinetics (a gas is thought as an ensemble of billions of billiard ball which bump into each other). Briefly, it consists in mimicking the physical phenomenon (the transport of photons or molecules), using a random generator for choosing the future of particles when they are subjected to an event (for example the encounter of a wall). In gas kinetics, molecules collide each other, so for evaluating the evolution of each one the evolution of the others has to be known previously : from where an apparent contradiction. The author proposes to solve this problem by following the molecules backward in the time, and by calculating on demand and recursively the past of the collision partners of the followed particles.

Méthode de Monte-Carlo et non-linéarités : de la physique du transfert radiatif à la cinétique des gaz

Résumé : En physique du transport, en particulier en physique du transfert radiatif, la méthode de Monte-Carlo a été développée à l'origine comme la simulation de l'histoire d'un grand nombre de particules, dont on déduit des observables moyennes. Cette méthode numérique doit son succès à plusieurs qualités : une gestion naturelle des espaces des phases aux nombreuses dimensions, une erreur systématique nulle par rapport au modèle physico-mathématique, les intervalles de confiance donnés avec les résultats, une capacité à prendre en compte simultanément de nombreux phénomènes physiques, la possibilité de calcul de sensibilités simultanées, et une parallélisation aisée.

En cinétique des gaz, les particules collisionnent entre elles et non pas avec un milieu extérieur ; on dit que leur transport est non-linéaire. Ces collisions mutuelles mettent en défaut l'approche évoquée ci-dessus de la méthode de Monte-Carlo ; car pour simuler des trajectoires indépendantes de multiples particules et ainsi estimer leur distribution, il faut connaître au préalable exactement cette même distribution. . .

Cette thèse fait suite à celles de Jérémie DAUCHET (2012) et de Mathieu GALTIER (2014), consacrées au transfert radiatif. Entre autres travaux, ces auteurs montraient comment la méthode de Monte-Carlo peut s'accommoder de non-linéarités, en gardant son formalisme et ses spécificités habituelles. Les non-linéarités alors franchies étaient respectivement une loi de couplage chimie/luminance, et la dépendance de la luminance envers le coefficient d'absorption. On essaie dans ce manuscrit d'outrepasser la non-linéarité du transport. Pour cela, nos principaux outils sont un suivi des particules en remontant le temps, basé sur des formulations intégrales des équations de transport, formulations largement inspirées des algorithmes dits à collisions nulles.

Nous montrons, sur plusieurs exemples académiques, que nous avons en effet étendu la méthode de Monte-Carlo à la résolution de l'équation de Boltzmann. Ces exemples sont aussi l'occasion de tester les limites de ce que nous avons mis en place. Les résultats les plus marquants sont certainement l'absence totale de maillage dans la méthode numérique, ainsi que sa capacité à calculer correctement les quantités de particules de haute énergie cinétique (toujours peu nombreuses par rapport au total, en cinétique des gaz).

Au-delà des exemples fournis, ce manuscrit est voulu comme un essai de formalisme et une exploration des bases de la méthode développée. L'accent est mis sur les raisonnements menant à la mise au point de la méthode, plutôt que sur les implémentations particulières qui ont été abouties. La méthode est encore, aux yeux de l'auteur, largement susceptible d'être retravaillée. En particulier, les temps maximaux sur lesquels l'évolution des particules est calculable, qui constituent la faiblesse principale de la méthode numérique développée, peuvent sûrement être augmentés.

Mots-clés : Méthode de Monte-Carlo, Cinétique des gaz, Transfert radiatif, Équation de Boltzmann, Transport non-linéaire, Formulation intégrale

Résumé (à l'adresse du grand public) : Le sujet est l'extension d'une méthode de calcul numérique, nommée méthode de Monte-Carlo, depuis la physique du transfert radiatif (l'étude du transfert de chaleur par la lumière, et aussi la synthèse d'image) vers la cinétique des gaz (on imagine le gaz comme un ensemble de milliards de boules de billard qui s'entrechoquent). Brièvement, elle consiste à mimer le phénomène physique (le transport des photons ou des molécules), à l'aide d'un générateur aléatoire pour choisir le devenir des particules quand elles subissent un événement (par exemple la rencontre d'une paroi). En cinétique des gaz, les molécules collisionnent entre elles, donc pour calculer l'évolution de chacune il faut connaître au préalable celle des autres : d'où une apparente contradiction. L'auteur propose de résoudre ce problème en suivant les molécules en remontant le temps, et en calculant à la demande et récursivement le passé des partenaires de collision des particules suivies.