

UNIVERSITÉ DE POITIERS

Propagation de la lumière dans un milieu participant de type fumée

Auteur

Elsa TAMISIER

Encadrants

Lilian AVENEAU
Pierre COMBEAU



06 JUILLET 2018

SOMMAIRE

Introduction	3
1 Monte Carlo	4
1.1 La méthode	4
1.2 L'échantillonnage	5
1.2.1 Méthode d'inversion	5
1.2.2 Méthode de rejet	5
1.2.3 Direction aléatoire en coordonnées sphériques	6
2 LTE	8
2.1 Deux équations	8
2.2 Equation en potentiel avec réflexions	9
2.3 Equation en potentiel optimisée	9
2.4 Equation de rendu avec réflexions	12
2.5 Equation de rendu optimisée	14
3 Fonctions de phase	15
3.1 Présentation	15
3.2 Henyey-Greenstein	16
3.3 Schlick	16
3.4 Rayleigh	17
3.5 Mie	17
4 Interactions	18
4.1 Absorption	18
4.2 Out-Scattering	19
4.3 Extinction	19
4.4 Émission	20
4.5 In-Scattering	21
4.6 Source	21
5 Équation de transfert	23
6 Algorithme 1	25
6.1 Présentation	25
6.2 Intégration continue	25
6.3 Implémentation discrète	26
6.4 Algorithme	26

6.5	Résultats	27
7	Algorithme 2	32
7.1	Présentation	32
7.2	Intégration continue	32
7.3	Découpe du rayon	33
7.4	Problème de temporalité	35
7.5	Algorithme	37
7.6	Résultats	38
8	Algorithme 3	39
8.1	Présentation	39
8.2	Découpe du rayon	39
8.3	Algorithme	39
8.4	Résultats	40
8.4.1	Coefficient de dispersion	40
8.4.2	Paramètre g	41
9	Algorithme 4	44
9.1	Présentation	44
9.2	Découpe du rayon	44
10	Algorithme 5	45
10.1	Présentation	45
10.2	Idée	45

Introduction

En informatique graphique, et plus précisément dans le domaine du rendu, il est courant de trouver des recherches, expérimentations, démonstrations, ... sur la façon dont la lumière va être rendue. La lumière a toujours intéressé les savants et il existe de nombreux papiers et manuscrits traitant de ce sujet. Physiquement (et très basiquement résumé), la lumière est l'énergie transportée par les *photons*, qui se déplacent selon une onde. Cette onde n'a pas toujours la même longueur λ et c'est cette différence qui produit un large spectre de lumière.

Ces photons interagissent avec toutes les particules qu'ils vont croiser. Certaines interactions sont extrêmement faibles, c'est par exemple le cas de l'air qui ne modifie pas (sur une distance pas trop importante) notre perception des couleurs. D'autres à l'inverse vont avoir un impact plus important, telles les interactions dues au passage de la lumière dans de la fumée.

Ce rapport résume ce que j'ai pu lire et comprendre autour de la propagation de la lumière dans un milieu participant, afin de proposer un algorithme simple permettant de calculer la radiance d'un rayon récupérée à la sortie d'un milieu participant (plus précisément de la fumée).

Dans un premier temps je présente rapidement la méthode de Monte Carlo qui permet d'algorithmer une intégrale sur un domaine donné, en utilisant l'aléatoire et les probabilités. Je ne m'attarde ni sur les démonstrations ni sur les optimisations que je ne compte pas appliquer pour le moment.

Puis je présente l'équation du transfert de la lumière (et plus précisément l'équation en potentiel) qui sera la base de notre démarche future.

Ensuite se trouve un court résumé expliquant ce que sont les fonctions de phase puis quelques fonctions de phase usuelles.

Le troisième chapitre se penche sur les différentes interactions qui s'appliquent sur la radiance d'un rayon lorsque celui-ci passe dans un milieu participant.

Dans le dernier chapitre, qui est très court, est introduite l'équation de transfert, qui combine l'ensemble des interactions. C'est avec cette équation de transfert et l'équation du transport de la lumière qu'il me faudra réfléchir à une solution.

Chapitre 1

Monte Carlo

1.1 La méthode

La méthode de Monte Carlo permet de trouver l'approximation d'une intégrale grâce à un échantillonnage aléatoire sur le domaine d'intégration. Soit I une intégrale sur son domaine Ω :

$$I = \int_{\Omega} f(x) dx. \quad (1.1)$$

Prenons un échantillon de N variable aléatoires x_i avec leur probabilités d'échantillonnage $p(x_i)$. Alors nous avons une approximation de l'intégrale avec \hat{I} , l'**estimateur** de Monte Carlo :

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)}. \quad (1.2)$$

$p(x_i)$ est appelée la *pdf* (Probability Density Function) de x_i . Nous avons les conditions suivantes sur une pdf :

$$\forall x \in \Omega, \quad p(x) \geq 0 \quad \text{et} \quad \int_{\Omega} p(x) dx = 1. \quad (1.3)$$

\hat{I} n'est qu'une approximation de I . Puisque nous travaillons avec un échantillon aléatoire, il est possible d'en tirer un nouveau et d'obtenir une nouvelle approximation. Aussi, il suffit de calculer plusieurs estimations et d'en faire la moyenne pour s'approcher de plus en plus de la valeur de I . L'estimateur \hat{I} de Monte Carlo converge vers I :

$$E \left[\hat{I} \right] = I. \quad (1.4)$$

Après des calculs basiques de probabilités, il est facile de trouver l'écart-type de \hat{I} :

$$\sigma \left[\hat{I} \right] = \frac{1}{\sqrt{N}} \sigma \left[\frac{f(x)}{p(x)} \right]. \quad (1.5)$$

Ceci indique que l'erreur diminue avec l'augmentation de la racine carrée de la taille de l'échantillonnage. C'est-à-dire que pour diviser l'erreur par 2, il faut multiplier le nombre d'échantillons par 4. Cette erreur n'est pas dépendante de la dimension du domaine d'intégration, ce qui est un avantage principal de la méthode de Monte Carlo.

1.2 L'échantillonnage

En informatique il est courant de générer des variables pseudo-aléatoires selon une loi uniforme sur $[0, 1]$. De nombreux algorithmes plus ou moins robustes permettent de tirer des séquences de nombres qui passeront les tests d'aléatoire.

Cependant, comme nous l'avons vu précédemment, la méthode de Monte Carlo utilise des variables aléatoires distribuées selon une loi de densité $p(x)$, qui est propre au domaine d'intégration. Il nous faut donc pouvoir générer de tels échantillons, en partant de variables aléatoires uniformes.

1.2.1 Méthode d'inversion

La première méthode, la plus efficace si la *pdf* le permet, est la méthode d'inversion. Celle-ci se base sur le calcul de la CDF (Cumulative Distribution Function) puis son inversement.

La CDF $P(x)$ associée à la *pdf* $p(x)$ est :

$$P(x) = p(y \in \Omega, y \geq x). \quad (1.6)$$

$$= \int_{-\infty}^x p(y) dy. \quad (1.7)$$

A partir de cette CDF et d'une variable aléatoire uniforme ξ_i , nous pouvons générer une variable aléatoire x_i selon la *pdf* avec la relation suivante :

$$x_i = P^{-1}(\xi_i). \quad (1.8)$$

Il suffit donc de réaliser N fois ce calcul à partir d'un échantillon uniforme de taille N , et nous aurons notre échantillon suivant la densité $p(x)$.

1.2.2 Méthode de rejet

Cependant une *pdf* ne permet pas toujours facilement de trouver la CDF associée. Ou bien cette dernière n'est pas toujours inversible (ou cette transformation est lourde à faire d'un point de vue algorithmique). Il existe donc une autre méthode pour produire un échantillonnage adéquat.

Pour cette méthode, il faut qu'en plus de notre densité $p(x)$ difficilement utilisable, nous ayons une densité $q(x)$ pour laquelle nous savons générer un échantillonnage (par exemple avec la méthode d'inversion), telle que :

$$\forall x \in \Omega, \quad p(x) \leq c \times q(x). \quad (1.9)$$

avec c une constante positive choisie judicieusement (voir ci-après).

Avec cette densité $q(x)$, il faut générer des échantillons x_i et, pour chacun, vérifier que

$$\xi \times c \times q(x_i) \leq p(x_i). \quad (1.10)$$

où ξ est une variable aléatoire qui suit une loi uniforme $U(0, 1)$.

Si la condition 1.10 est vérifiée, alors x_i est un échantillon accepté et sera utilisé. Sinon, une nouvelle paire (x_i, ξ) est générée et testée à nouveau.

Plus l'écart entre les deux densités $p(x)$ et $cq(x)$ est grand, plus il y a des chances que l'échantillon x_i soit rejeté. Aussi, pour optimiser la vitesse de génération de l'échantillonnage, il faut chercher à prendre la constante c la plus petite possible.

1.2.3 Direction aléatoire en coordonnées sphériques

Dans le cadre de la propagation de la lumière, nous aurons à échantillonner sur un hémisphère pour lancer le rayon, puis sur des sphères lorsque nous calculerons le comportement de la lumière sur une particule du milieu participant. L'échantillonnage d'une direction est donc plutôt à faire en coordonnées sphériques où elle est représentée par deux données (θ, ϕ) qui représente un point sur la surface de la sphère. θ est appelé le *zenith* (ou *élévation*) du point. Et ϕ est appelé l'*azimuth* du point.

Nous avons les intervalles de définition suivants :

$$\theta \in \begin{cases} [0, \pi] & \text{sur une sphère.} \\ [0, \frac{\pi}{2}] & \text{sur un hémisphère.} \end{cases}$$

$$\phi \in [0, 2\pi].$$

Mais il ne suffit pas de tirer deux variables aléatoires uniformes sur l'intervalle correspondant pour générer un bon échantillon (θ, ϕ) . En effet, imaginons que l'on coupe des tranches de notre sphères, orthogonalement à l'axe de révolution de l'angle ϕ . Alors, peu importe la valeur de θ et donc l'aire de notre cercle, ϕ sera distribué de la même manière. De manière uniforme, nous aurons donc autant d'échantillons sur un cercle large (lorsque θ est proche de $\frac{\pi}{2}$) que sur un cercle bien plus petit (lorsque θ est proche de 0). il en découlera une mauvaise répartition des directions échantillonnées. Et l'échantillonnage ne sera plus uniforme sur la sphère.

Il faut échantillonner la direction $\vec{\omega}$ selon la *pdf* $p(\vec{\omega})$. C'est-à-dire qu'il faut échantillonner chacune des deux variables θ et ϕ selon leur *pdf* respectives $p(\theta)$ et $p(\phi)$ associée à $p(\vec{\omega})$. Plus précisément, il ne faut pas échantillonner les deux variables indépendamment, mais en prenant en compte la valeur trouvée pour θ lorsque l'on va vouloir générer ϕ .

Avec des calculs en partant de l'angle solide $d\vec{\omega}$ et en sachant que $p(\vec{\omega})$ est constante (puisque nous voulons une distribution uniforme sur la sphère ou l'hémisphère), nous arrivons aux fonctions suivantes.

Pour un hémisphère.

$$p(\vec{\omega}) = \frac{1}{2\pi}. \quad (1.11)$$

$$p(\theta) = \sin(\theta). \quad (1.12)$$

$$p(\phi \mid \theta) = \frac{1}{2\pi}. \quad (1.13)$$

$$P(\theta) = 1 - \cos\theta. \quad (1.14)$$

$$P(\phi \mid \theta) = \frac{\phi}{2\pi}. \quad (1.15)$$

$$P^{-1}(\xi_\theta) = \cos^{-1}(1 - \xi_\theta). \quad (1.16)$$

$$P^{-1}(\xi_\phi) = 2\pi\xi_\phi. \quad (1.17)$$

où ξ_θ et ξ_ϕ sont des variables aléatoires suivant la loi uniforme.

Pour une sphère.

$$p(\vec{\omega}) = \frac{1}{4\pi}. \quad (1.18)$$

$$p(\theta) = \frac{\sin(\theta)}{2} \quad (1.19)$$

$$p(\phi \mid \theta) = \frac{1}{2\pi}. \quad (1.20)$$

$$P(\theta) = \frac{1 - \cos\theta}{2}. \quad (1.21)$$

$$P(\phi \mid \theta) = \frac{\phi}{2\pi}. \quad (1.22)$$

$$P^{-1}(\xi_\theta) = \cos^{-1}(1 - 2\xi_\theta). \quad (1.23)$$

$$P^{-1}(\xi_\phi) = 2\pi\xi_\phi. \quad (1.24)$$

Chapitre 2

LTE

2.1 Deux équations

L'équation du transport de la lumière *LTE* (Light Transport Equation) décrit la distribution de la radiance dans toute la scène. Elle permet de connaître la radiance réfléchie en un point d'une surface, en fonction de l'émission de cette surface, sa BRDF et toutes les autres surfaces de la scène pouvant venir perturber le rayon.

Il existe deux LTE différentes : l'équation de *rendu* et l'équation en *potentiel*. Dans l'équation de rendu, les rayons étudiés partent du récepteur et on cherche des chemins remontant jusqu'aux sources. L'équation de potentiel partage le même principe mais dans l'autre sens, les rayons étant tiré de la source et on calcule leur contributions sur le récepteur.

Dans les deux cas, il s'agit de trouver l'éclairage global de la scène et non pas juste l'éclairage direct qui ne prend en compte que les rayons provenant directement d'une source de lumière. Ici, on souhaite également tenir compte de la radiance provenant de réflexions sur des surfaces n'étant pas forcément auto-émittante.

L'équation de rendu s'exprime par la formule :

$$L_r(x, \vec{\omega}_r, t) = L_e(x, \vec{\omega}_r, t) + \int_{\Omega_x} f_r(x, \vec{\omega}_i \longrightarrow \vec{\omega}_r) L_i(x, \vec{\omega}_i, t) |\vec{\omega}_i, \vec{n}| d\vec{\omega}_i, \quad (2.1)$$

où $L_e(x, \vec{\omega}_r, t)$ est la lumière auto-émise par la surface,

$L_i(x, \vec{\omega}_i, t)$ est la radiance incidente dans la direction $\vec{\omega}_i$,

\vec{n} est la normale à la surface au point x ,

$f_r(x, \vec{\omega}_i \longrightarrow \vec{\omega}_r)$ est la BRDF de la surface au point x dans le sens de propagation $\vec{\omega}_i \longrightarrow \vec{\omega}_r$.

L'équation en potentiel s'exprime par la formule :

$$W(x, y, \vec{\omega}_x, t) = W_e(x, y, \vec{\omega}_x, t) + \int_{\Omega_z} f_r(z, \vec{\omega}_x \longrightarrow \vec{\omega}_z) W(z, y, \vec{\omega}_z, t) |\vec{\omega}_z, \vec{n}_z| d\vec{\omega}_z, \quad (2.2)$$

où $W(x, y, \vec{\omega}_x, t)$ est l'action potentielle du point x sur le point y dans la direction $\vec{\omega}_x$,

$W_e(x, y, \vec{\omega}_x, t)$ est l'action potentielle directe de la source au point x sur le point y suivant $\vec{\omega}_x$.

Dans ce rapport et pour ce qui suit, je me place dans le cas de l'équation en potentiel.

2.2 Equation en potentiel avec réflexions

L'éclairement reçu au niveau d'un récepteur positionné en M_{rx} pour un rayonnement émis par une source positionnée en M_{tx} est donné par :

$$E(M_{tx}, M_{rx}, t) = \int_{\Omega_{tx}} L_e(M_{tx}, \vec{\omega}_0, t) W(M_{tx}, M_{rx}, \vec{\omega}_0, t) |\vec{\omega}_0 \cdot \vec{n}_{tx}| d\vec{\omega}_0, \quad (2.3)$$

où $L_e(M_{tx}, \vec{\omega}_0, t)$ désigne le rayonnement sortant dans la direction $\vec{\omega}_0$.

Cela dit, la plupart des rayons lancés de la source ne vont pas directement au récepteur et sont d'abord réfléchis par des surfaces. Aussi, après k réflexions, nous avons l'équation suivante :

$$E_k(M_{tx}, M_{rx}, t) = \int_{\Omega_{tx}} \cdots \int_{\Omega_k} f_k(\bar{x}, t) d\vec{\omega}_0 \cdots d\vec{\omega}_k, \quad (2.4)$$

où \bar{x} correspond au trajet de propagation de la lumière entre M_{tx} et M_{rx} rebondissant sur les x_i points de réflexions ($i \in \llbracket 1, k \rrbracket$). Et

$$f_k(\bar{x}, t) = |\vec{\omega}_0 \cdot \vec{n}_{tx}| L_e(M_{tx}, \vec{\omega}_0, t) \text{rect} \left(\frac{|\overrightarrow{x_k M_{rx}} \cdot \vec{n}_{rx}|}{\cos(FOV) \|\overrightarrow{x_k M_{rx}}\|} \right) W_l(x_k, M_{rx}, \vec{\omega}_k t) \\ \times \prod_{i=1}^k f_r(x_i, \overrightarrow{\omega_{i-1}} \longrightarrow \vec{\omega}_i) |\vec{\omega}_i \cdot \vec{n}_i|, \quad (2.5)$$

avec la fonction *rect* qui vérifie que x_k est dans le champ de vision du récepteur et $W_l(x_k, M_{rx}, t)$ qui est l'action potentielle directe du dernier point de réflexion x_k sur le récepteur.

2.3 Equation en potentiel optimisée

L'équation précédente permet seulement de suivre un chemin de k réflexions puis de regarder l'action potentielle sur le récepteur du dernier point x_k du chemin. De ce fait, les actions potentielles des points de réflexions x_i ($1 \leq i < k$) doivent être calculés indépendamment les uns des autres, et nous aurons à lancer k fois le même chemin (de plus en plus long) pour avoir les k contributions.

Une optimisation possible de l'équation cherche donc à ne pas répéter inutilement les réflexions. Elle consiste à calculer, pour chaque point x_i son action potentielle sur le récepteur dès le premier lancé de rayon. Cette technique s'appelle l'*estimation du prochain évènement* (NEE "Next Event Estimation"). Ces contributions sont représentées dans la figure 2.3 par les traits en pointillés reliant un point de réflexion x_i à un récepteur M_{rx} .

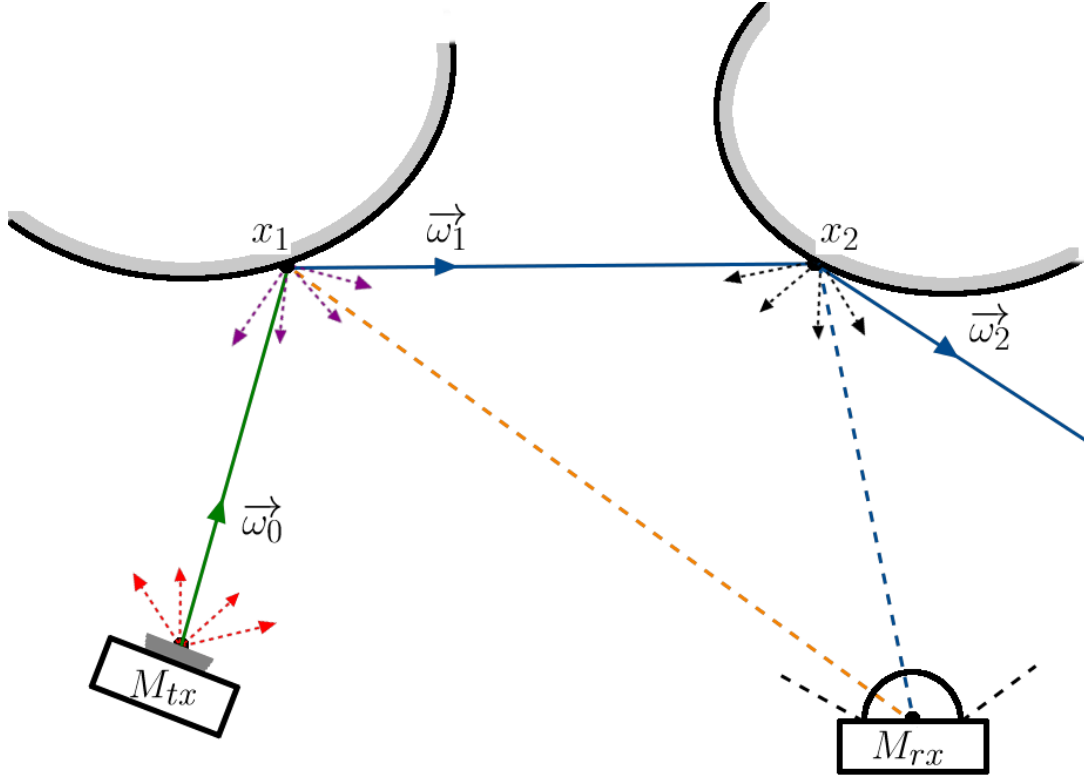


FIGURE 2.1 – Équation en potentiel avec la technique NEE

$$E(M_{tx}, M_{rx}, t) = \int_{\Omega_{tx}} \underbrace{L_e(M_{tx}, \vec{\omega}_0, t)}_{\text{rayonnement sortant de } M_{tx} \text{ dans la direction } \vec{\omega}_0} \underbrace{W(M_{tx}, M_{rx}, \vec{\omega}_0, t)}_{\text{action potentielle de la source } M_{tx} \text{ sur le récepteur } M_{rx} \text{ dans la direction } \vec{\omega}_0} |\vec{\omega}_0 \cdot \vec{n}_{tx}| \underbrace{d\vec{\omega}_0}_{\text{intégrale sur le domaine des directions}} . \quad (2.6)$$

On tire un rayon de la source M_{tx} dans la direction $\vec{\omega}_0$. Bien évidemment nous avons le rayonnement $L_e(M_{tx}, \vec{\omega}_0, t)$ sortant de la source, rayonnement atténué selon son inclinaison par rapport à la normale \vec{n}_{tx} de la source. Ce rayonnement n'atteignant pas forcément le récepteur M_{rx} immédiatement, il faut donc calculer l'action potentielle $W(M_{tx}, M_{rx}, \vec{\omega}_0, t)$ que la source a sur le récepteur par la direction $\vec{\omega}_0$.

$$E(M_{tx}, M_{rx}, t) = \int_{\Omega_{tx}} L_e(M_{tx}, \vec{\omega}_0, t) |\vec{\omega}_0 \cdot \vec{n}_{tx}| \times \left(\underbrace{W_l^*(x_1, M_{rx}, t)}_{\substack{\text{contribution directe} \\ \text{de } x_1 \text{ sur } M_{rx}, \text{ en} \\ \text{tenant compte de la} \\ \text{BDRF}}} + \int_{\Omega_1} \underbrace{W^*(x_1, M_{rx}, \vec{\omega}_1, t)}_{\substack{\text{contribution dans la direction} \\ \vec{\omega}_1 \text{ en repartant de } x_1 \\ (\text{\`a d\'erouler})}} d\vec{\omega}_1 \right) d\vec{\omega}_0, \quad (2.7)$$

avec

$$W^*(x_1, M_{rx}, \vec{\omega}_1, t) = f_r(x_1, \vec{\omega}_0 \longrightarrow \vec{\omega}_1) W(x_1, M_{rx}, \vec{\omega}_1, t) |\vec{\omega}_1 \cdot \vec{n}_1|, \quad (2.8)$$

et

$$W_l^*(x_1, M_{rx}, t) = f_r(x_1, \vec{\omega}_0 \longrightarrow \overrightarrow{x_1 M_{rx}}) W_l(x_1, M_{rx}, t) |\vec{\omega}_1 \cdot \vec{n}_1|. \quad (2.9)$$

L'action potentielle $W(M_{tx}, M_{rx}, \vec{\omega}_0, t)$ se calcule en deux termes. Premièrement, sachant qu'en partant dans la direction $\vec{\omega}_0$ à partir du point M_{tx} le rayon intercepte une surface au point de réflexion x_1 , il faut regarder la contribution directe sur le récepteur de la lumière réfléchie en ce point ; il s'agit du terme $W_l^*(x_1, M_{rx}, t)$. Ce premier terme ne traite donc que la direction de réflexion $\overrightarrow{x_1 M_{rx}}$.

Hors il y a tout un hémisphère de directions $\vec{\omega}_1$ possibles qu'il faut regarder. C'est ce que se charge de faire le deuxième terme $\int_{\Omega_1} W^*(x_1, M_{rx}, \vec{\omega}_1, t) d\vec{\omega}_1$. Ce terme là continue donc les chemins possibles du rayon et parmi toutes les directions, certaines iront intercepter d'autres surfaces (la direction ω_1 qui intercepte une surface au point x_2 par exemple, sur la figure 2.3). C'est donc sur celui-ci qu'il faudra dérouler l'équation pour avoir $E_k(M_{tx}, M_{rx}, t)$.

$$E(M_{tx}, M_{rx}, t) = \int_{\Omega_{tx}} L_e(M_{tx}, \vec{\omega}_0, t) W_l^*(x_1, M_{rx}, t) |\vec{\omega}_0 \cdot \vec{n}_{tx}| d\vec{\omega}_0 + \int_{\Omega_{tx}} L_e(M_{tx}, \vec{\omega}_0, t) |\vec{\omega}_0 \cdot \vec{n}_{tx}| \int_{\Omega_1} W^*(x_1, M_{rx}, \vec{\omega}_1, t) d\vec{\omega}_0 d\vec{\omega}_1. \quad (2.10)$$

En séparant l'intégrale sur Ω_{tx} en deux intégrales, avec une seule à dérouler, nous pouvons facilement écrire l'équation pour k réflexions.

$$E_k(M_{tx}, M_{rx}, t) = \int_{\Omega_{tx}} f_1(M_{tx}, x_1, M_{rx}, t) d\vec{\omega}_0 + \int_{\Omega_{tx}} \int_{\Omega_1} f_2(M_{tx}, x_1, x_2, M_{rx}, t) d\vec{\omega}_0 d\vec{\omega}_1 + \vdots + \int_{\Omega_{tx}} \cdots \int_{\Omega_k} f_k(\bar{x}, t) d\vec{\omega}_0 \cdots d\vec{\omega}_k. \quad (2.11)$$

Avec les intégrales sur les domaines de chemin, il est facile de voir que cette équation est très facilement transformable avec la méthode de Monte Carlo pour des facilités algorithmiques :

$$E_k(M_{tx}, M_{rx}, t) = \underbrace{\frac{1}{N} \sum_{s=1}^N \frac{|\vec{\omega}_0 \cdot \vec{n}_{tx}| L_e(M_{tx}, \vec{\omega}_{tx}, t)}{p_0(M_{tx}, \vec{\omega}_0)}}_{\text{On tire une première direction pour } \vec{\omega}_0} \times \sum_{i=1}^k \left(\underbrace{W_l(x_i, M_{rx}, t)}_{\text{Contribution directe du point } x_i \text{ sur le récepteur}} \underbrace{\prod_{j=1}^i \frac{f_r(x_j, \vec{\omega}_{j-1} \rightarrow \vec{\omega}_j) |\vec{\omega}_j \cdot \vec{n}_j|}{p_j(x_j, \vec{\omega}_j)}}_{\text{Prise en compte de toutes les BRDF des surfaces interceptées par le chemin aux points } x_i} \right). \quad (2.12)$$

Cependant cette équation ne prend pour l'instant pas du tout en compte les milieux participants. Il s'agira donc ensuite de réfléchir à une solution pour injecter la distribution de la lumière par la fumée.

2.4 Equation de rendu avec réflexions

Rappelons l'équation de rendu (equation 2.1) :

$$L_r(x, \vec{\omega}_r, t) = L_e(x, \vec{\omega}_r, t) + \int_{\Omega_x} f_r(x, \vec{\omega}_i \rightarrow \vec{\omega}_r) L_i(x, \vec{\omega}_i, t) |\vec{\omega}_i \cdot \vec{n}| d\vec{\omega}_i,$$

où $L_e(x, \vec{\omega}_r, t)$ est la lumière auto-émise par la surface,
 $L_i(x, \vec{\omega}_i, t)$ est la radiance incidente dans la direction $\vec{\omega}_i$,
 \vec{n} est la normale à la surface au point x ,
 $f_r(x, \vec{\omega}_i \rightarrow \vec{\omega}_r)$ est la BRDF de la surface au point x dans le sens de propagation $\vec{\omega}_i \rightarrow \vec{\omega}_r$.

Similairement à l'équation en potentiel, on va pouvoir en déduire l'éclairement reçu en un point x après k réflexions.

Après k réflexions, on connecte le point de surface x_k au récepteur. Ceci est représenté par la luminance rayonnée depuis l'émetteur M_{tx} et reçue au dernier point de réflexion x_k :

$$L_l(M_{tx}, x_k, t) = L_e(M_{tx}, \overrightarrow{M_{tx}x_k}, t) V(x_k, M_{tx}) \frac{|\overrightarrow{M_{tx}x_k} \cdot \vec{n}_{tx}|}{\|\overrightarrow{M_{tx}x_k}\|^3}. \quad (2.13)$$

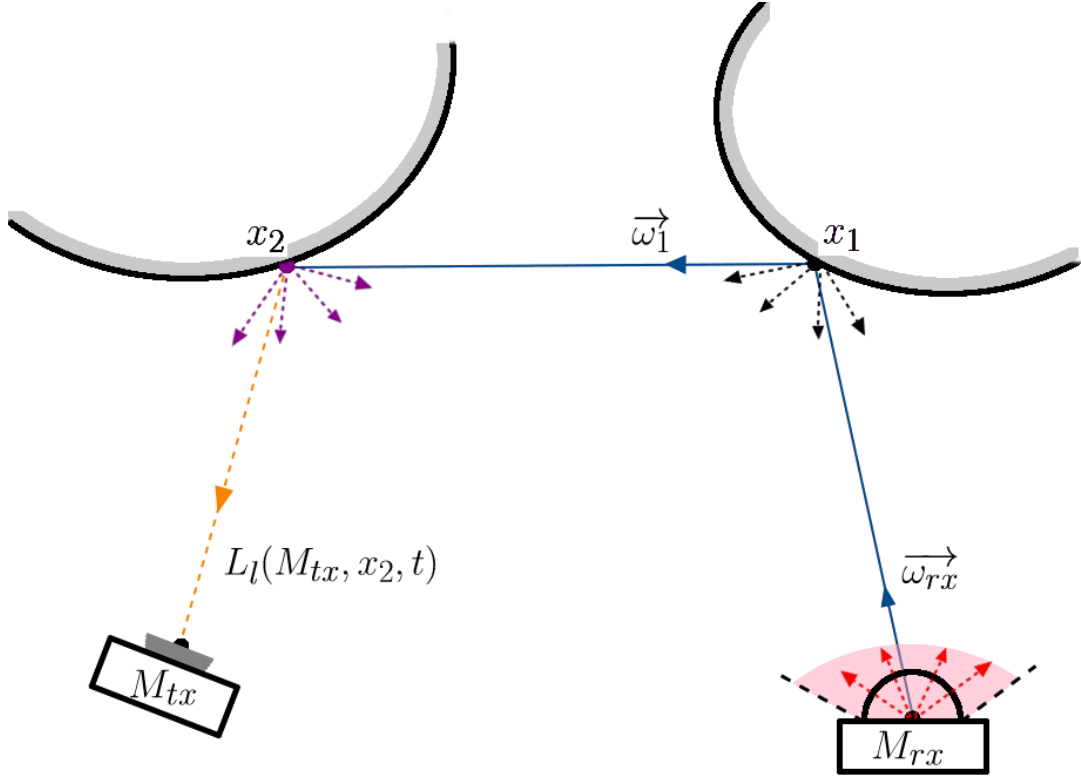


FIGURE 2.2 – Algorithme de rendu pour $k = 2$ réflexions

On rappelle que $V(x_k, M_{tx})$ est la fonction de visibilité qui vaut 1 si les points x_k et M_{tx} sont visibles l'un l'autre, 0 sinon.

Nous voulons connaître l'éclairement reçu au récepteur, par le point de réflexion x . L'équation de rendu nous permet de modéliser la luminance réfléchi au point x_1 dans la direction $\overrightarrow{x_1 M_{rx}}$. En déroulant cette équation, nous arrivons pour k réflexions à l'équation suivante :

$$E_k(x_1, M_{rx}, t) = \int_{\Omega_0} \cdots \int_{\Omega_k} g_k(\bar{x}, t) d\vec{\omega}_0 \cdots d\vec{\omega}_k, \quad (2.14)$$

où $\bar{x} = (M_{rx}, x_1, x_2, \dots, x_k, M_{tx})$ est un trajet partant du récepteur M_{rx} et passant par k points de réflexions avant d'atteindre la source M_{tx} . Et la fonction g_k est :

$$g_k(\bar{x}, t) = |\vec{\omega}_0 \cdot \vec{n}_{rx}| \text{rect} \left(\frac{|\vec{\omega}_0 \cdot \vec{n}_{rx}|}{\cos(\text{FOV})} \right) \times L_l(M_{tx}, x_k, t) \prod_{i=1}^k f_r(x_i, \vec{\omega}_{i-1} \longrightarrow \vec{\omega}_i) |\vec{\omega}_i \cdot \vec{n}_i|, \quad (2.15)$$

où la fonction rect vérifie que la direction $\vec{\omega}_0$ au champ de vision du récepteur, le produit des BRDF $f_r(x_i, \vec{\omega}_{i-1} \longrightarrow \vec{\omega}_i)$ à chaque point x_i du trajet est un coefficient multiplicateur du dernier rayonnement en x_k .

2.5 Equation de rendu optimisée

Dans cet algorithme trivial on doit itérer plusieurs fois sur le même chemin si l'on souhaite avoir les contributions de chaque point de réflexion x_i ($1 \leq i \leq k$) recevant directement de la lumière l'émetteur. On va donc, tout comme pour l'équation en potentiel, relier chaque point de réflexion à l'émetteur. Pour chaque point de réflexion x_i , si celui-ci peut recevoir une lumière directe de la source, la contribution de celle-ci est ajoutée avant de tirer une nouvelle direction aléatoire pour aller trouver un autre point de réflexion x_{i+1} .

L'estimateur de Monte Carlo de cette version améliorée est le suivant :

$$E_k(M_{tx}, M_{rx}, t) = \frac{1}{N} \underbrace{\sum_{s=1}^N \frac{|\vec{\omega}_{rx} \cdot \vec{n}_{rx}|}{p_0(M_{rx}, \vec{\omega}_{rx})}}_{\substack{\text{On tire une première} \\ \text{direction pour } \vec{\omega}_{rx}}} \times \sum_{i=1}^k \left(\underbrace{L_l(M_{tx}, x_i, t)}_{\substack{\text{Contribution de} \\ \text{la source sur le} \\ \text{point } x_i}} \underbrace{\prod_{j=1}^i \frac{f_r(x_j, \vec{\omega}_{j-1} \rightarrow \vec{\omega}_j) |\vec{\omega}_j \cdot \vec{n}_j|}{p_j(x_j, \vec{\omega}_j)}}_{\substack{\text{Prise en compte de toutes} \\ \text{les BRDF des surfaces interceptées} \\ \text{par le chemin aux points } x_i}} \right). \quad (2.16)$$

Chapitre 3

Fonctions de phase

3.1 Présentation

Une BSDF (Bidirectional Scattering Distribution Function) permet de décrire et simuler la façon dont la lumière est dispersée par une surface. Une **fonction de phase** $\rho(\omega_i \longrightarrow \omega_o)$ est son analogue volumétrique. Elle permet de décrire la façon dont la lumière se disperse *à l'intérieur* du volume. On peut la lire comme une pdf donnant la probabilité qu'un photon arrivant sur une particule dans la direction ω_i en ressorte par la direction ω_o . De ce fait, une fonction de phase possède une contrainte de normalisation :

$$\int_{\Omega_{4\pi}} \rho(\omega_i \longrightarrow \omega_o) d\omega' = 1. \quad (3.1)$$

Dans beaucoup de milieux participants, la fonction de phase est **isotropique**. Celle-ci ne dépend que de l'*angle de phase* θ entre le rayon incident ω_i et le rayon sortant ω_o . On se contente alors d'écrire $\rho(\cos\theta)$. La dispersion est identique dans toutes les directions pour un angle donné.

Il est également intéressant de noter que les fonctions de phases sont des fonctions réciproques, les deux directions incidentes et sortantes peuvent être interchangées et la valeur de la fonction de phase restera la même :

$$\rho(\omega_i \longrightarrow \omega_o) = \rho(\omega_o \longrightarrow \omega_i). \quad (3.2)$$

Tout comme les BSDF, il existe de nombreux modèles de fonctions de phase qui produisent des effets différents. Certains peuvent facilement être décrits avec un seul paramètre, mais d'autres plus complexes demandent une équation bien plus grosse et difficilement utilisables dans un algorithme. Il est donc parfois plus simple de diviser une fonction de phase $\rho(\omega_i \longrightarrow \omega_o)$ complexe en plusieurs fonctions de phases $\rho_k(\omega_i \longrightarrow \omega_o)$ plus simples. Nous avons l'égalité suivante :

$$\rho(\omega_i \longrightarrow \omega_o) = \sum_{k=1}^n w_k \rho_k(\omega_i \longrightarrow \omega_o), \quad (3.3)$$



FIGURE 3.1 – Backscattering, $g < 0$



FIGURE 3.2 – Forwardscattering, $g > 0$

où chacune des fonctions ρ_k possède un poids w_k dans l'équation, avec la somme de tous les w_k qui égale 1 afin de maintenir la *condition de normalisation* (formule 3.1).

3.2 Henyey-Greenstein

Henyey et Greenstein (1941) ont introduit une fonction de phase qui est aujourd'hui beaucoup utilisée car elle ne dépend que d'un paramètre $-1 \leq g \leq 1$ et décrit les effets de *backscattering* tout comme ceux de *forwardscattering* :

$$\rho_{HG}(\cos\theta) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 - 2g(\cos\theta))^{3/2}}.$$

Le paramètre g n'est pas arbitraire et peut être calculé avec la formule suivante :

$$g = \int_{\Omega_{4\pi}} \rho(\omega_i \rightarrow \omega_o)(\omega_i \cdot \omega_o) d\omega_o = 2\pi \int_0^\pi \rho(\cos\theta) \cos\theta \sin\theta d\theta.$$

La valeur de g induit certains effets de dispersion dans la fonction de phase ρ_{HG} . Si $g < 0$ alors la fonction de phase décrit principalement des effets de *backscattering* (Figure 3.1). Si $g > 0$, on a à l'inverse principalement du *forwardscattering* (Figure 3.2). Enfin, comme on s'y attend, si $g = 0$, la fonction de phase est isotropique.

3.3 Schlick

La fonction de phase de Henyey et Greenstein est cependant lourde à calculer à cause de la puissance présente au dénominateur. Blasi, Le Saëc et Schlick (1993) ont développé une approximation de ρ_{HG} qui évite cette puissance et qui est donc plus utilisée en informatique graphique, où la forme exacte de la fonction de phase de Henyey et Greenstein n'est pas nécessaire.

$$\rho_{Schlick}(\cos\theta) = \frac{1}{4\pi} \frac{1 - k^2}{(1 - k\cos\theta)^2},$$

où le paramètre k joue le même rôle que le terme g : $k = -1$ un backscattering total, $k = 0$ implique l'isotropie du milieu, $k = 1$ correspond à un forwardscattering total. Pharr et Humphreys

(2004) ont trouvé que définir k comme :

$$k = 1,55g - 0,55g^2,$$

était une bonne façon d'avoir une relation directe entre k et g avec laquelle $\rho_{Schlick}$ avoisine très bien ρ_{HG} .

3.4 Rayleigh

Le modèle de Rayleigh décrit une dispersion de la lumière dans un milieu composé de petites particules, jusqu'à un dixième de la longueur d'onde de la lumière.

$$\rho_R(\cos\theta) = \frac{3}{16\pi}(1 + \cos^2\theta).$$

Ce modèle permet par exemple de simuler la diffusion de la lumière dans le ciel. En le suivant, on retrouve ainsi sa couleur bleue.

3.5 Mie

La théorie de Rayleigh ne fonctionne que pour des particules de petites tailles. Lorsque celles-ci dépassent le dixième de leur longueur d'onde (aérosols, poussière, goutellettes d'eau), il faudrait passer à la théorie de Mie. Cela dit, son application est compliquée. La figure 3.3 présente une fonction de phase pour un nuage, trouvée avec la théorie de Mie.

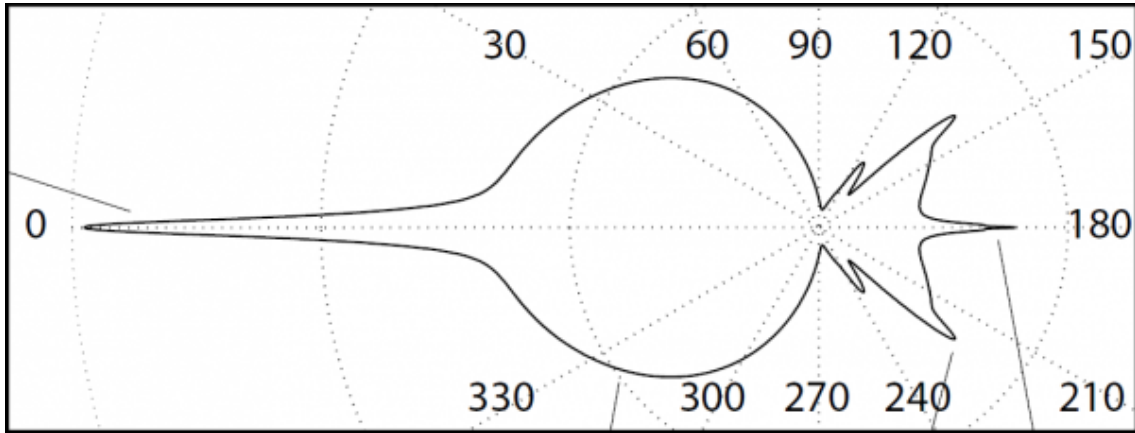


FIGURE 3.3 – Fonction de phase d'un nuage

Aussi, plutôt que d'utiliser cette fonction de phase compliquée, on choisit généralement une composition de plusieurs fonctions de phase de Henyey et Greenstein (ou Schlick). C'est ici, par exemple, qu'est utilisée la formule 3.3 :

$$\rho(\omega_i \longrightarrow \omega_o) = \sum_{k=1}^n w_k \rho_k(\omega_i \longrightarrow \omega_o).$$

Chapitre 4

Interactions

Lorsque la lumière traverse un milieu participant, de multiples interactions peuvent avoir lieu au fur et à mesure que le rayon de lumière avance dans le milieu et rencontre des particules. La fréquence et la puissance de ces interactions vont dépendre de plusieurs propriétés du milieu participant, comme la densité par exemple. Ces propriétés varient dans le milieu, celui-ci étant rarement globalement homogène. Lorsque nous en serons à l'algorithme, il sera donc possible de décomposer le milieu en une grille 3D de fragments homogènes où les propriétés du milieu seront constantes.

4.1 Absorption

Lorsque la lumière arrive sur une particule, celle-ci fait obstruction et un certains nombres de photons ne parviennent donc pas à continuer leur chemin, c'est l'**absorption** (Figure 4.1).



FIGURE 4.1 – Absorption par une particule

Un milieu participant homogène possède un *coefficient d'absorption* σ_a qui correspond à la probabilité que la lumière soit absorbée lors de son passage dans le milieu. Une estimation simple et constante de ce coefficient est la suivante :

$$\sigma_a(p, \omega) = \pi r^2 N,$$

où r est le rayon des particules (m) et N la densité des particules (m^{-3}). σ_a est donc en m^{-1} .

La valeur qui nous intéresse est la variation de radiance entre le faisceau arrivant sur la particule et celui en partant. Cette différence ne dépend que de la radiance initiale et du coefficient d'absorption du milieu traversé :

$$\Delta L(p, \omega) = L_o(p, \omega) - L_i(p, \omega) = -\sigma_a(p, \omega) L_i(p, \omega) dt. \quad (4.1)$$

4.2 Out-Scattering

Lorsque le faisceau de lumière rencontre une particule dans le milieu, les photons qui le composent peuvent être absorbés (cf. Absorption) mais également renvoyés dans d'autres directions. C'est ce qu'on appelle l'**out-scattering** (figure 4.2).

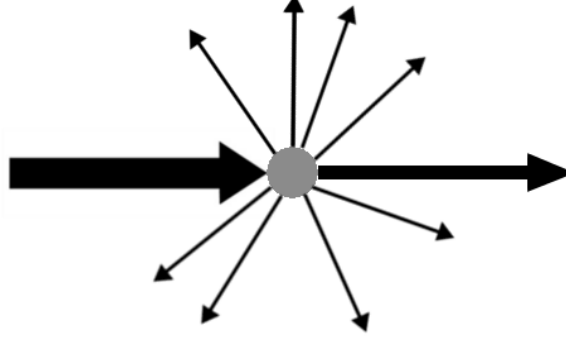


FIGURE 4.2 – Out-scattering causé par une particule

Tout comme il existe un coefficient d'absorption, le milieu possède un *coefficient de dispersion* σ_s correspondant à la probabilité que le faisceau de lumière soit dispersé.

Son impact est semblable au coefficient d'absorption :

$$\Delta L(p, \omega) = -\sigma_s(p, \omega)L_i(p, \omega)dt. \quad (4.2)$$

4.3 Extinction

Nous avons deux interactions qui atténuent la lumière lorsque celle-ci passe dans un milieu participant : l'absorption et l'out-scattering. Ces deux effets combinés donnent l'**extinction**. Un milieu participant homogène possède donc, de manière analogue aux deux interactions vues précédemment, un *coefficient d'atténuation*, ou *coefficient d'extinction* σ_t . Celui-ci est tout simplement donné par :

$$\sigma_t = \sigma_a + \sigma_s.$$

Ce qui conduit à :

$$\Delta L(p, \omega) = -\sigma_t(p, \omega)L_i(p, \omega)dt. \quad (4.3)$$

Nous pouvons maintenant calculer non pas un delta de radiance mais une **proportion**. Soient p et p' deux points sur un rayon de direction normalisée ω , à une distance d l'un de l'autre. Alors nous avons :

$$T_r(p \longrightarrow p') = e^{-\int_0^d \sigma_t(p+t\omega, \omega)dt} = e^{-\tau}. \quad (4.4)$$

où T_r est la *transmittance*, la proportion de radiance qui est transmise de p à p' sur un rayon traversant le milieu participant. τ est appelé l'*épaisseur optique* du milieu. Sachant que dans un milieu homogène σ_t est constant, on a $T_r(p \longrightarrow p') = e^{-\sigma_t d}$: c'est la *loi de Beer*.

Ainsi, soit $L_o(p, \omega)$ la radiance sortant d'un point p d'une surface dans une direction normalisée ω . Ce rayon traverse un milieu participant et arrive sur un autre point p' . Alors la radiance incidente à ce point p' est donnée par (cf. figure 4.3) :

$$L_i(p', -\omega) = T_r(p \longrightarrow p') L_o(p, \omega). \quad (4.5)$$

Il va donc de soit que dans le vide, $T_r(p \longrightarrow p') = 1$.

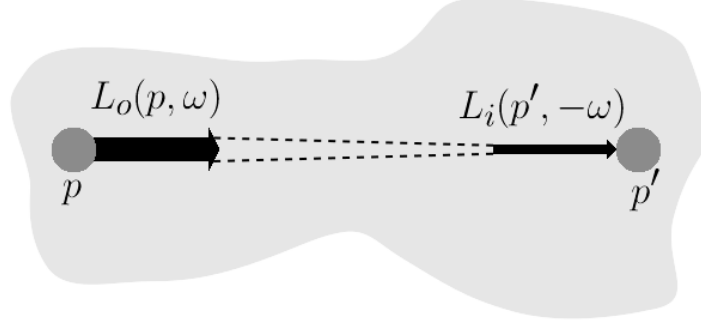


FIGURE 4.3 – Atténuation d'un rayon grâce à la transmittance

Une propriété importante de la transmittance, qui nous sera utile pour l'écriture de l'algorithme, dit que la transmittance est multiplicative le long des points sur un rayon. Soient p et p'' deux points sur un rayon. Et soit p' un point entre p et p'' . Alors on a :

$$T_r(p \longrightarrow p'') = T_r(p \longrightarrow p') T_r(p' \longrightarrow p''). \quad (4.6)$$

4.4 Émission

Lors de son passage dans un milieu participant, la radiance du rayon ne fait pas que diminuer. Certains milieux participants peuvent auto émettre de la lumière. C'est ce qu'on appelle l'**émission** (figure 4.4). Lorsque le faisceau de lumière passe sur une particule émettant de la lumière, celui-ci gagne en intensité.

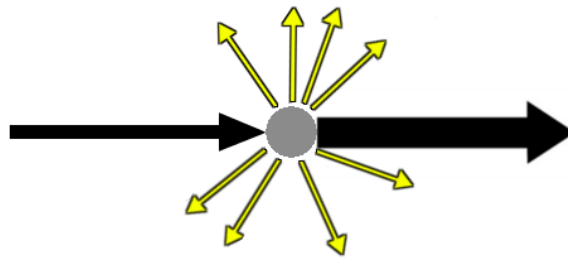


FIGURE 4.4 – Émission d'une particule

La lumière émise, qui ne dépend ici pas du rayon incident au point p s'ajoute à la radiance :

$$\Delta L(p, \omega) = L_{ve}(p, \omega). \quad (4.7)$$

4.5 In-Scattering

Nous avons vu que lors d'une dispersion de la lumière (out-scattering), des photons étaient déviés de leur trajectoire, ne contribuant donc plus à la radiance du rayon initialement lancé. Ces photons ne disparaissent cependant pas et vont contribuer ailleurs, ils vont s'ajouter à un autre rayon avec qui ils partagent la trajectoire. Cela dit, il serait lourd de calculer, à chaque point p et chaque événement d'out-scattering, les contributions de chaque nouveau rayon de contributions individuellement faibles. Pour simuler cet effet, nous avons donc l'interaction **in-scattering** (figure 4.5).

Lorsque le faisceau rencontre une particule et que nous voulons calculer les contributions de chacune des interactions présentées précédemment, nous ajoutons la contribution des ces radiances perdues.

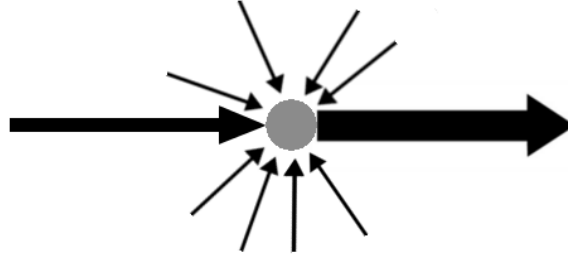


FIGURE 4.5 – In-scattering sur une particule

Cet ajout dépend naturellement du *coefficient de dispersion* σ_s , puisque l'*out-scattering* et l'*in-scattering* sont dûs à la même réaction.

Le calcul de cette interaction est cependant un peu plus complexe. En effet, là où pour l'out-scattering nous n'avons pas besoin de savoir ce qu'il se passait de la radiance perdue, nous devons ici prendre en compte que nouveaux rayons qui ont voyagé dans le milieu avant d'arriver jusqu'à notre particule, et dont la direction a changé. C'est ici que nous utilisons la **fonction de phase** ρ de notre milieu :

$$\Delta L(p, \omega) = \sigma_s(p, \omega) \int_{\Omega_{4\pi}} \rho(p, -\omega' \rightarrow \omega) d\omega' dt. \quad (4.8)$$

Le domaine d'intégration est ici la sphère complète autour du point p . Puisque nous ne sommes pas sur une surface, il n'y a en effet pas d'obstruction faite ne laissant qu'une hémisphère de possibilités.

4.6 Source

Au final, nous avons deux interactions qui viennent augmenter la radiance d'un rayon lors de son passage dans un milieu participant : l'émission et l'in-scattering. Cela nous donne la **source**

L_s de radiance qui va venir s'ajouter à la radiance initiale :

$$L_s(p, \omega) = L_{ve}(p, \omega) + \sigma_s(p, \omega) \int_{\Omega_{4\pi}} \rho(p, -\omega' \longrightarrow \omega) L_i(p, \omega') d\omega'. \quad (4.9)$$

En découle :

$$\Delta L(p, \omega) = L_s(p, \omega) dt. \quad (4.10)$$

Chapitre 5

Équation de transfert

Avec toutes ces interactions causées par le passage de la lumière dans un milieu participant, nous pouvons écrire l'**équation de transfert**. Cette équation va nous permettre de décrire le comportement et la distribution de la lumière dans le milieu. L'équation du transport de la lumière (LTE) est assez proche de ce que nous allons écrire, puisqu'il s'agit en fait d'une équation de transfert mais sans milieu participant.

La différence entre la radiance incidente à un point p selon la direction ω et la radiance sortante de ce point p est tout simplement calculable grâce à l'extinction et la source causées par le milieu :

$$\Delta L(p, \omega) = [-\sigma_t(p, \omega)L_i(p, \omega) + L_s(p, \omega)]dt. \quad (5.1)$$

Rappelons que nous avons donc dans cette équation l'ensemble des quatre interactions :

$$\begin{aligned} \frac{\Delta L(p, \omega)}{dt} = & - \underbrace{\underbrace{\sigma_a(p, \omega)L_i(p, \omega)}_{\text{absorption}} - \underbrace{\sigma_s(p, \omega)L_i(p, \omega)}_{\text{out-scattering}}}_{\text{extinction}} \\ & + \underbrace{\underbrace{L_{ve}(p, \omega)}_{\text{emission}} + \underbrace{\sigma_s(p, \omega) \int_{\Omega 4\pi} \rho(p, -\omega' \rightarrow \omega) L_i(p, \omega') d\omega'}_{\text{in-scattering}}}_{\text{source}}. \end{aligned} \quad (5.2)$$

De là, il y a plusieurs moyens d'utiliser cette équation. Supposons qu'aucune surface ne vienne bloquer le rayon et que celui-ci soit donc de longueur infinie. Alors la radiance incidente à un point p dans une direction ω est augmentée par la radiance S ajoutée à chaque point p' le long du rayon. Cette quantité de radiance ajoutée est réduite par la transmittance T_r . Nous aurons donc :

$$L_i(p, \omega) = \int_0^\infty T_r(p' \rightarrow p) L_s(p', -\omega) dt.$$

Supposons maintenant qu'il y ait présence de surfaces. Alors d'une part ces surfaces vont interagir avec les radiances du milieu par réflexion et/ou émission et donc ajouter une contribution dans l'équation finale. Et d'autre part, elles peuvent empêcher complètement un rayon de continuer son chemin et donc enlever la contribution directe de ce rayon de l'autre côté de la surface.

Soit un rayon d'origine p et de direction ω qui intercepte une surface au point p_0 (on a $p_0 = p + t\omega$ où t est la distance entre les deux points). Alors l'équation de transfert est la suivante :

$$L_i(p, \omega) = T_r(p_0 \longrightarrow p)L_o(p_0, -\omega) + \int_0^t T_r(p' \longrightarrow p)L_s(p', -\omega)dt'. \quad (5.3)$$

où $p' = p + t'\omega$ sont les points le long du rayon entre p et p_0

Le premier terme de cette équation, $T_r(p_0 \longrightarrow p)L_o(p_0, -\omega)$, correspond à la radiance réfléchie par la surface en p_0 dans la direction $-\omega$. C'est dans ce terme que la contribution de la surface est ajoutée (réflexion et émission sont compris dans L_o). Le second terme est celui bien connu maintenant, causé par les différentes interactions ayant eu lieu lors du passage de la lumière dans le média.

Chapitre 6

Algorithme 1

6.1 Présentation

Cet algorithme reprend l'algorithme MCS (ou MCG) classique avec la technique NEE, si ce n'est qu'entre deux points de réflexions on prend en compte l'atténuation de la fumée. Cette prise en compte est également faite entre la source et les différents points de surface, ou entre le récepteur et les différents points de surface.

6.2 Intégration continue

Entre deux points p et p' , dans la direction ω ($p' = p + t\omega$), le rayon passe dans de la fumée et de ce fait nous avons la formule suivante :

$$L_i(p', \omega) = T_r(p \rightarrow p') L_o(p, \omega). \quad (6.1)$$

Rappelons ici la définition de la transmittance T_r (eq. 4.4) :

$$T_r(p \longrightarrow p') = e^{-\int_0^d \sigma_t(p+t\omega, \omega) dt}. \quad (6.2)$$

Dans cet algorithme, c'est le seul calcul que nous ferons, pour chaque luminance sortant d'un point de réflexion (ou de la source) et arrivant sur un autre point de réflexion (ou sur l'émetteur).

Rappelons l'équation 2.10 pour l'algorithme MCS avec NEE :

$$\begin{aligned} E(M_{tx}, M_{rx}, t) = & \int_{\Omega_{tx}} L_e(M_{tx}, \vec{\omega}_0, t) W_l^*(x_1, M_{rx}, t) |\vec{\omega}_0 \cdot \vec{n}_{tx}| d\vec{\omega}_0 \\ & + \int_{\Omega_{tx}} L_e(M_{tx}, \vec{\omega}_0, t) |\vec{\omega}_0 \cdot \vec{n}_{tx}| \int_{\Omega_1} W^*(x_1, M_{rx}, \vec{\omega}_1, t) d\vec{\omega}_0 d\vec{\omega}_1, \end{aligned}$$

avec

$$W_l^*(x_1, M_{rx}, t) = f_r(x_1, \vec{\omega}_0 \longrightarrow \overrightarrow{x_1 M_{rx}}) W_l(x_1, M_{rx}, t) |\vec{\omega}_1 \cdot \vec{n}_1|,$$

et

$$W^*(x_1, M_{rx}, \vec{\omega}_1, t) = f_r(x_1, \vec{\omega}_0 \longrightarrow \vec{\omega}_1) W(x_1, M_{rx}, \vec{\omega}_1, t) |\vec{\omega}_1 \cdot \vec{n}_1|.$$

En ajoutant la prise en compte de la fumée, nous aurons :

$$E(M_{tx}, M_{rx}, t) = \int_{\Omega_{tx}} \mathbf{T}_r(M_{tx} \rightarrow x_1) L_e(M_{tx}, \vec{\omega}_0, t) W_l^*(x_1, M_{rx}, t) |\vec{\omega}_0 \cdot \vec{n}_{tx}| d\vec{\omega}_0 \\ + \int_{\Omega_{tx}} \mathbf{T}_r(M_{rx} \rightarrow x_1) L_e(M_{tx}, \vec{\omega}_0, t) |\vec{\omega}_0 \cdot \vec{n}_{tx}| \int_{\Omega_1} W^*(x_1, M_{rx}, \vec{\omega}_1, t) d\vec{\omega}_0 d\vec{\omega}_1, \quad (6.3)$$

avec

$$W_l^*(x_1, M_{rx}, t) = \mathbf{T}_r(x_1 \rightarrow M_{rx}) f_r(x_1, \vec{\omega}_0 \rightarrow \overrightarrow{x_1 M_{rx}}) W_l(x_1, M_{rx}, t) |\vec{\omega}_1 \cdot \vec{n}_1|,$$

et

$$W^*(x_1, M_{rx}, \vec{\omega}_1, t) = f_r(x_1, \vec{\omega}_0 \rightarrow \vec{\omega}_1) W(x_1, M_{rx}, \vec{\omega}_1, t) |\vec{\omega}_1 \cdot \vec{n}_1|,$$

où

$$W(x, y, \vec{\omega}_x, t) = \mathbf{T}_r(x \rightarrow y) W_e(x, y, \vec{\omega}_x, t) + \int_{\Omega_z} f_r(z, \vec{\omega}_x \rightarrow \vec{\omega}_z) W(z, y, \vec{\omega}_z, t) |\vec{\omega}_z \cdot \vec{n}_z| d\vec{\omega}_z.$$

6.3 Implémentation discrète

Cette intégrale peut être transformée avec Monte Carlo :

$$E_k(M_{tx}, M_{rx}, t) = \frac{1}{N} \sum_{s=1}^N \underbrace{\frac{|\vec{\omega}_0 \cdot \vec{n}_{tx}| \mathbf{T}_r(M_{tx} \rightarrow x_1) L_e(M_{tx}, \vec{\omega}_{tx}, t)}{p_0(M_{tx}, \vec{\omega}_0)}}_{\text{On tire une première direction pour } \vec{\omega}_0} \\ \times \sum_{i=1}^k \left(\underbrace{W_l(x_i, M_{rx}, t)}_{\text{Contribution directe du point } x_i \text{ sur le récepteur}} \underbrace{\prod_{j=1}^i \frac{\mathbf{T}_r(x_j \rightarrow x_{j-1}) f_r(x_j, \vec{\omega}_{j-1} \rightarrow \vec{\omega}_j) |\vec{\omega}_j \cdot \vec{n}_j|}{p_j(x_j, \vec{\omega}_j)}}_{\text{Prise en compte de toutes les BRDF des surfaces interceptées, et de la fumée entre les différents sommets du chemin}} \right). \quad (6.4)$$

6.4 Algorithme

Dans ma classe **ParticipatingMedia**, j'ai ajouté la méthode **multiplyByTransmittance** qui, pour une distance d donnée, multiplie des champs par la transmittance $tr = e^{-\sigma_t d}$ (possible car nous sommes dans un milieu homogène).

Ensuite, cette méthode est appliquée sur chaque rayon produit par l'algorithme de lancer de rayons.

6.5 Résultats

Dans cet algorithme, puisque le seul impact de la fumée est l'extinction par la transmittance, nous n'avons pas à différencier les coefficients d'atténuation σ_a et de dispersion σ_s . En effet, le seul qui nous intéresse est $\sigma_t = \sigma_a + \sigma_s$.

Les résultats qui suivent ont été réalisés avec 10^8 rayons lancés, 3 réflexions, et 4 valeurs différentes de σ_t . Ils sont présentés dans un premier temps, par réflexion (seulement le trajet direct, seulement la première réflexion, ...) puis toutes les contributions sont rassemblées.

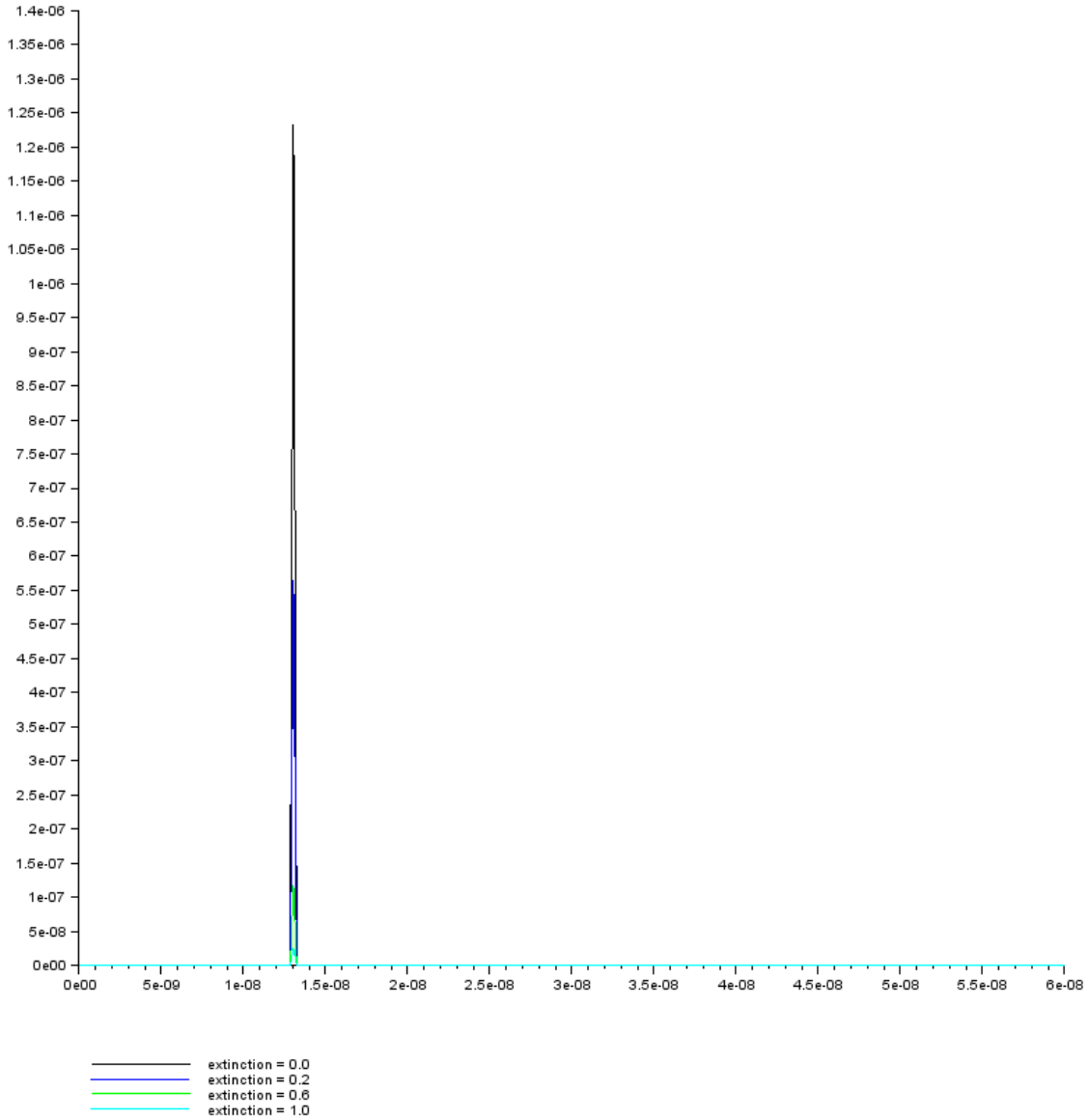


FIGURE 6.1 – $k = 0$

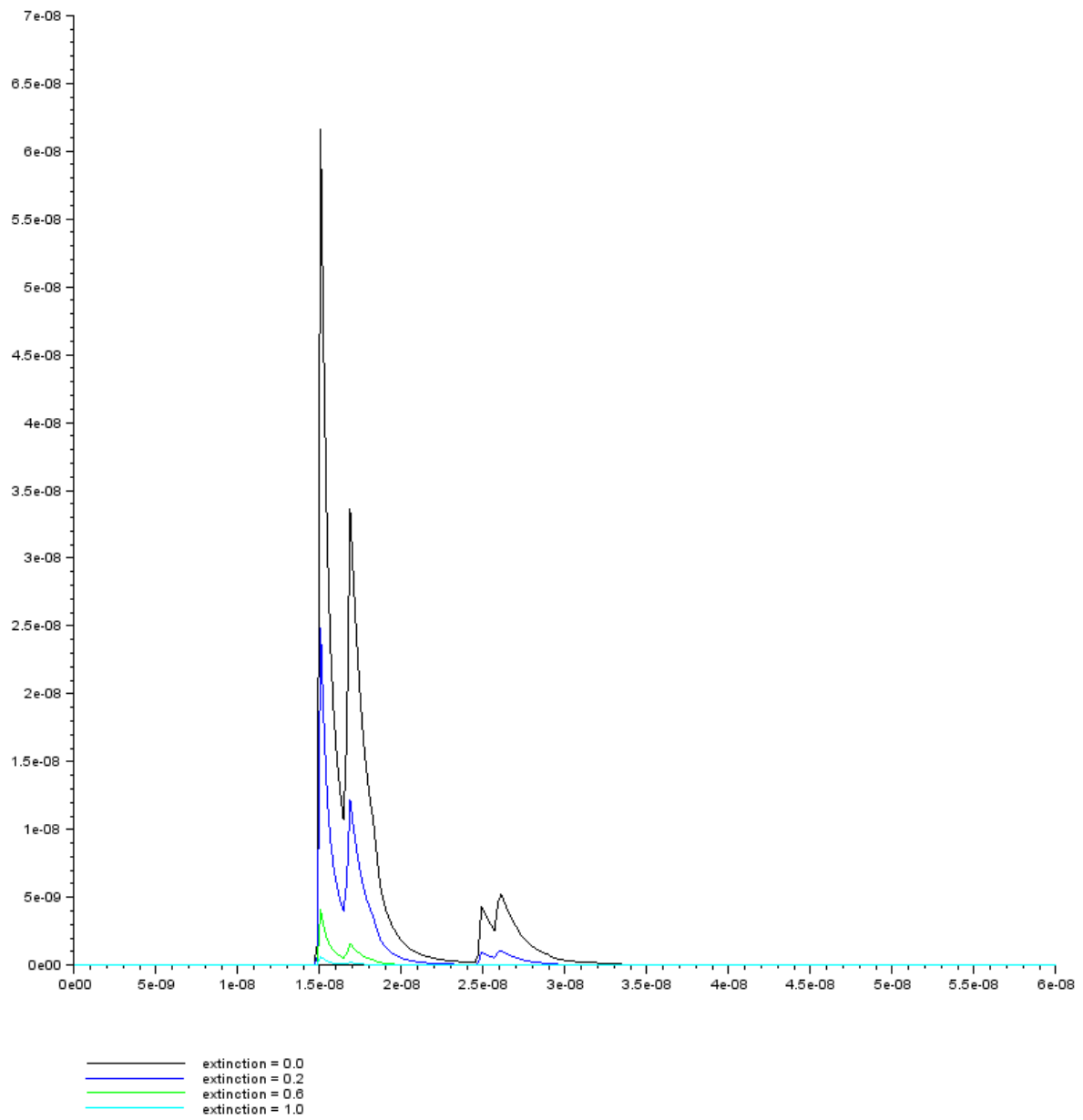


FIGURE 6.2 – $k = 1$

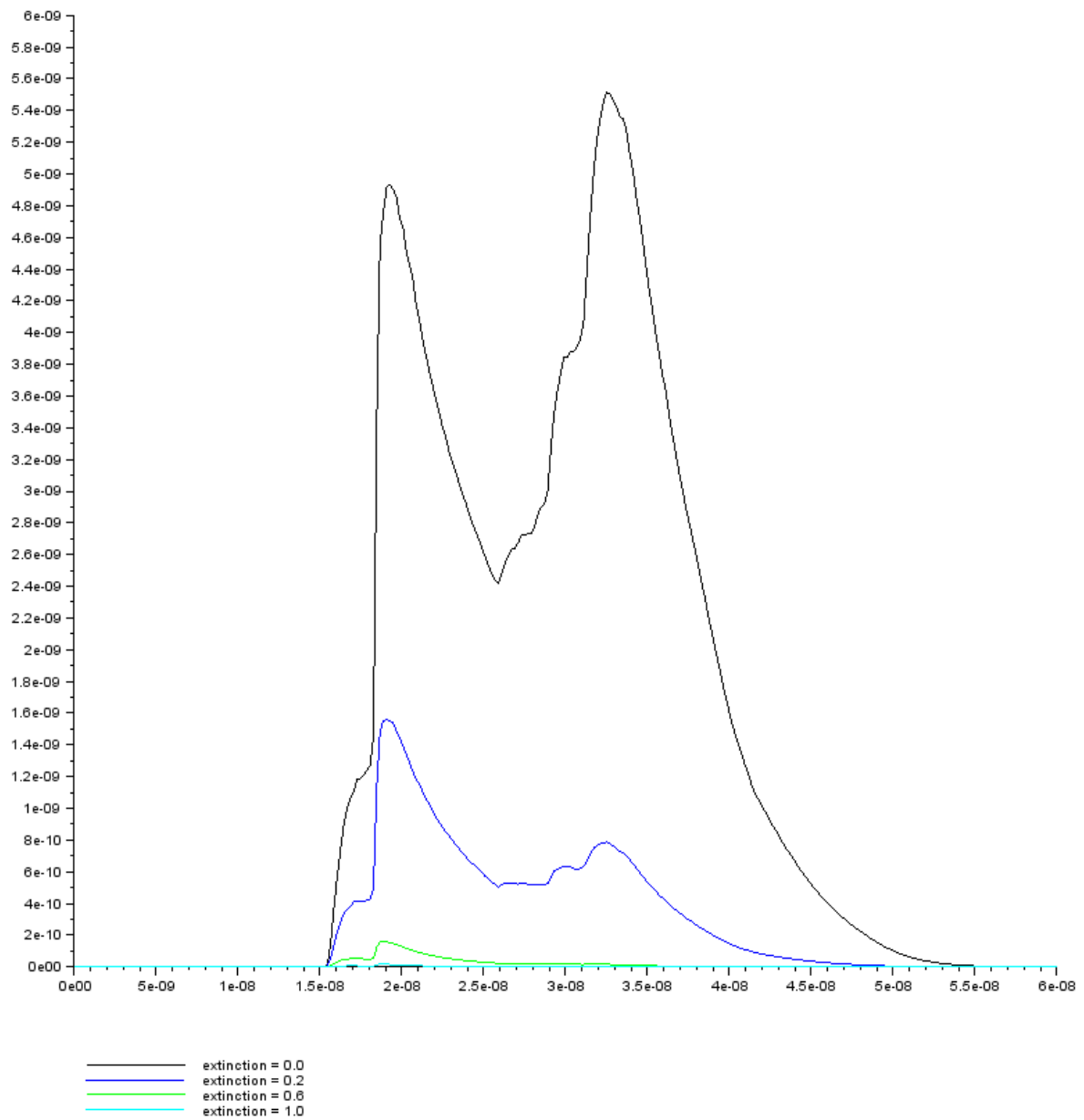


FIGURE 6.3 – $k = 2$

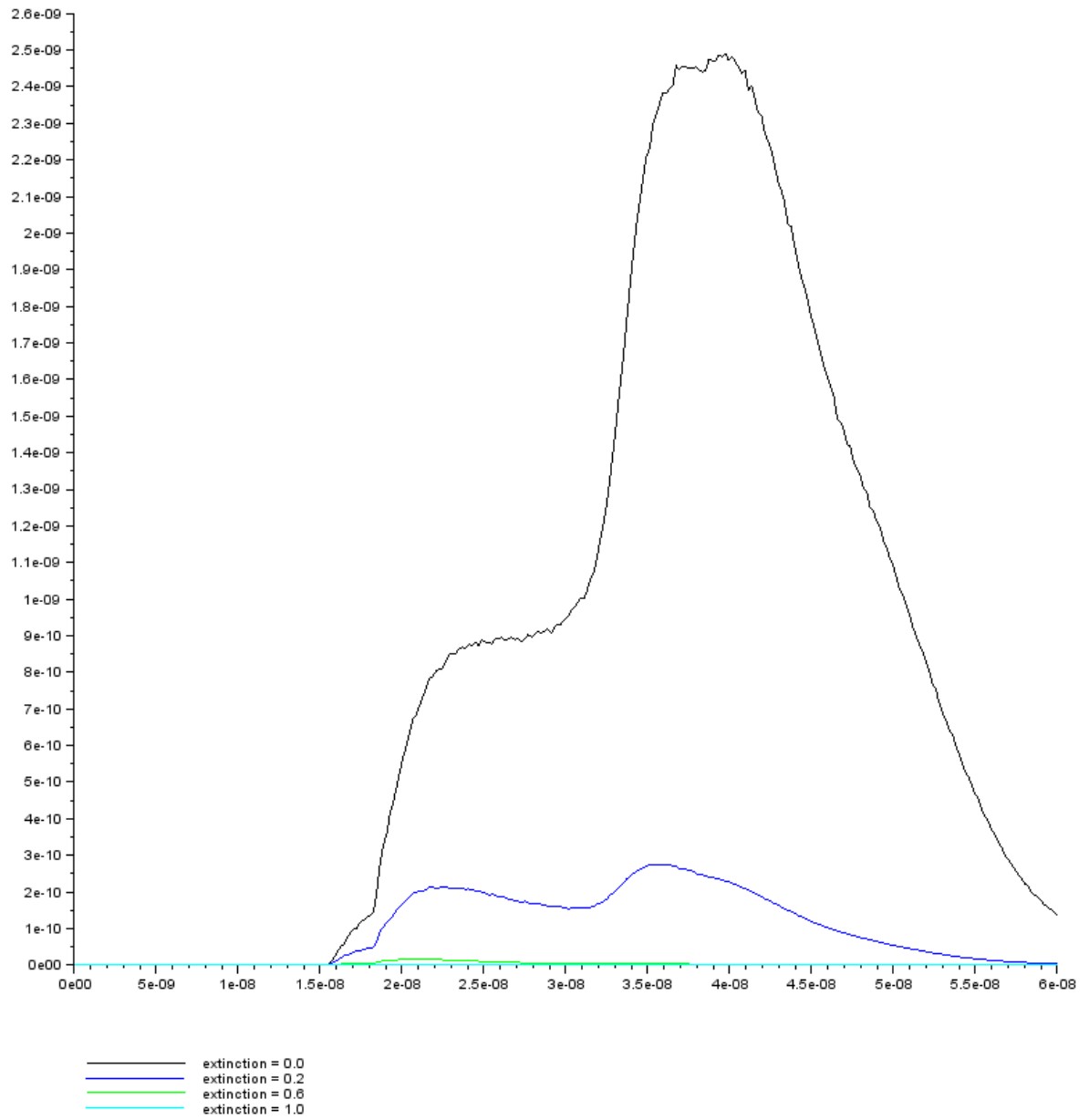


FIGURE 6.4 – $k = 3$

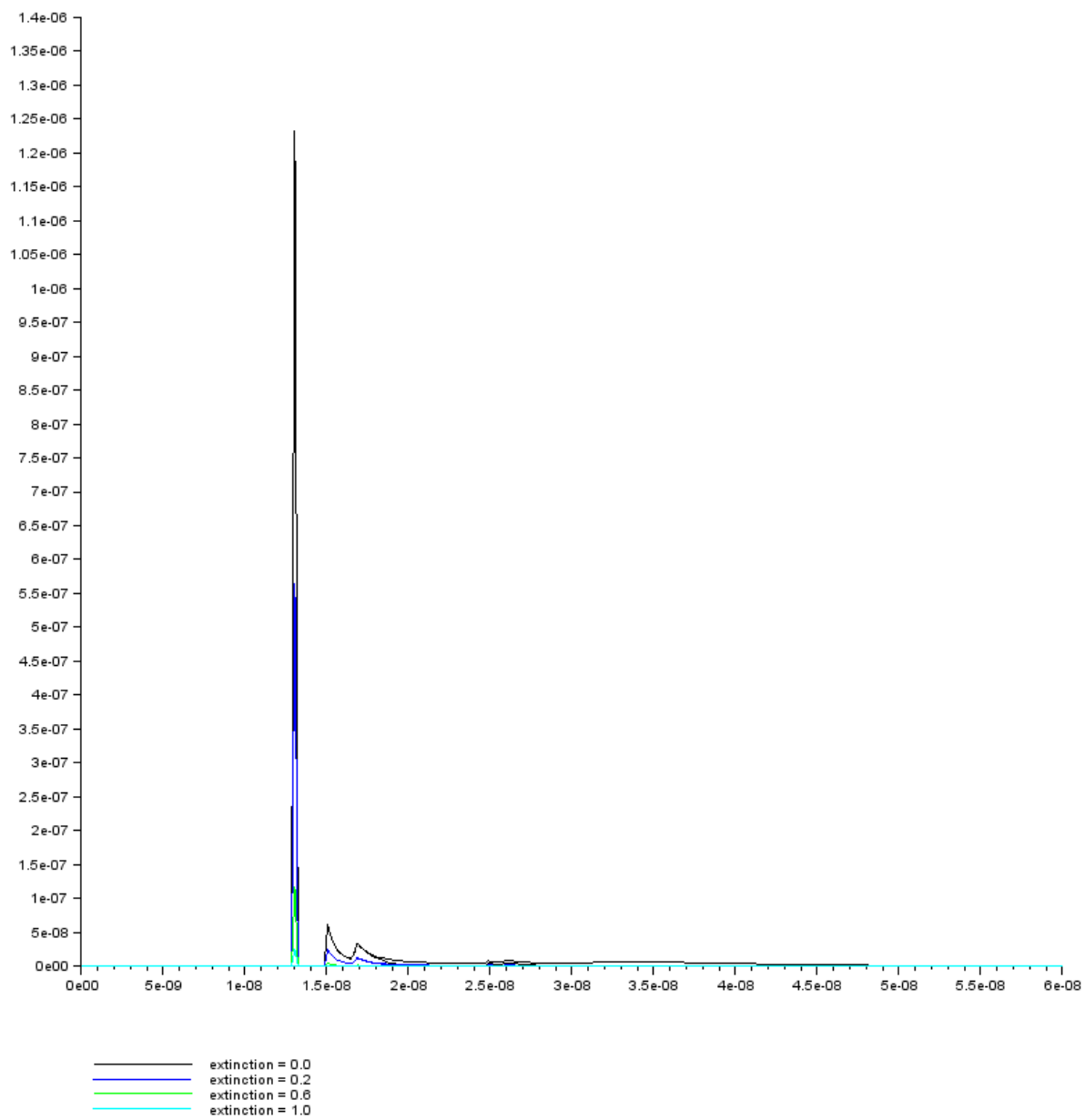


FIGURE 6.5 – $k = 0, 1, 2, 3$

Chapitre 7

Algorithme 2

7.1 Présentation

Le deuxième algorithme reprend le premier algorithme mais décompose chaque rayon entre deux points de surface (ou émetteur/récepteur) en plusieurs segments afin de prendre en compte l'in-scattering provenant de la source (et uniquement de la source) à plusieurs endroits dans la fumée.

7.2 Intégration continue

Prenons un rayon qui après un rebond sur une surface au point x_i , repart dans une direction ω_i pour arriver au point x_{i+1} . Alors la luminance arrivant sur le point x_{i+1} et portée par ce rayon est donnée par (eq. 5.3) :

$$L_i(x_{i+1}, \omega_i) = T_r(x_i \longrightarrow x_{i+1})L_o(x_i, \omega_i) + \int_0^t T_r(x_i \longrightarrow p')L_s(p', \omega_i)dt'.$$

Dans notre cas, nous n'avons pas de lumière émise par la fumée. Seul l'in-scattering intervient et le terme L_s devient donc :

$$L_s(p, \omega) = \sigma_s(p, \omega) \int_{S^2} \rho(p, \omega' \longrightarrow \omega) L_i(p, \omega') d\omega'.$$

De plus, les L_i venant des différentes directions ω_i ne sont pas connues dans un premier temps. La seule direction incidente pour laquelle il est possible d'avoir une valeur (autre que la direction ω) est la direction $\overrightarrow{M_{tx}p_k}$ de la source au point p_k . Nous n'avons plus qu'un seul choix de direction incidente, il n'est plus nécessaire d'avoir du Monte Carlo et la source devient donc :

$$L_s(p, \omega) = \sigma_s(p, \omega) \rho(p, \overrightarrow{M_{tx}p} \longrightarrow \omega) L_d(p), \quad (7.1)$$

où $L_d(p)$ est la luminance venant directement de la source :

$$L_d(p) = T_r(M_{tx} \longrightarrow p) L_e(M_{tx}, \omega). \quad (7.2)$$

7.3 Découpe du rayon

Le lumière traversant la fumée va rencontrer de nombreuses particules avant d'arriver sur une surface. Il serait lourd de réaliser autant de calculs qu'il n'y a de particules sur le chemin du rayon, mais d'un autre côté ne réaliser qu'un seul calcul serait très inexact. Il faut donc découper ce rayon en plusieurs points p_i séparés d'une distance à trouver de manière cohérente. Mon rayon, lancé au point p et touchant une surface au point p' est donc décomposé comme le montre la figure 7.1.

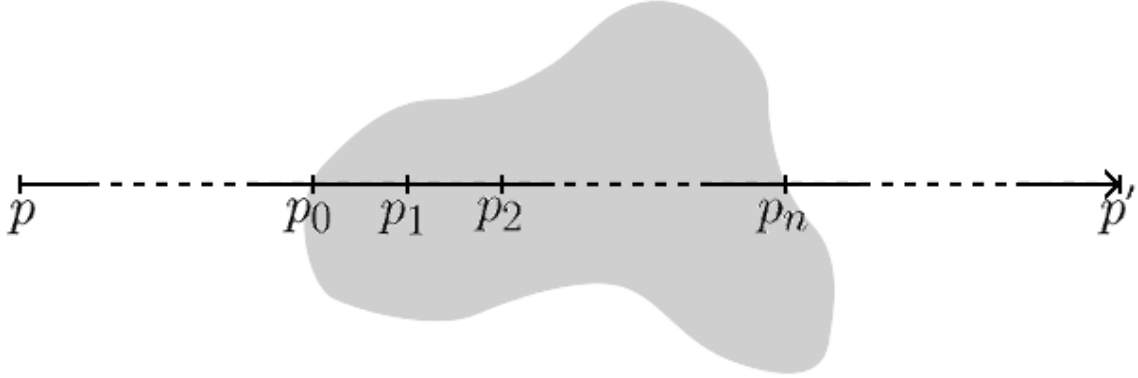


FIGURE 7.1 – Décomposition du rayon

Nous connaissons le point de départ p et le point d'arrivée p' du rayon. Nous pouvons donc connaître le temps t qui correspond à l'arrivée de la lumière sur le récepteur, après avoir effectué le chemin (M_{tx}, p, M_{rx}) . Il en va de même pour le temps t' pas le chemin (M_{tx}, p, p', M_{rx}) . Avec ces deux données temporelles, il est possible de déduire les positions des points intermédiaires p_k , et plus précisément leur distance à p , afin d'obtenir des potentiels répartis régulièrement entre t et t' , selon une durée Δt qui est un paramètre de la simulation. (cf. figure 7.2)

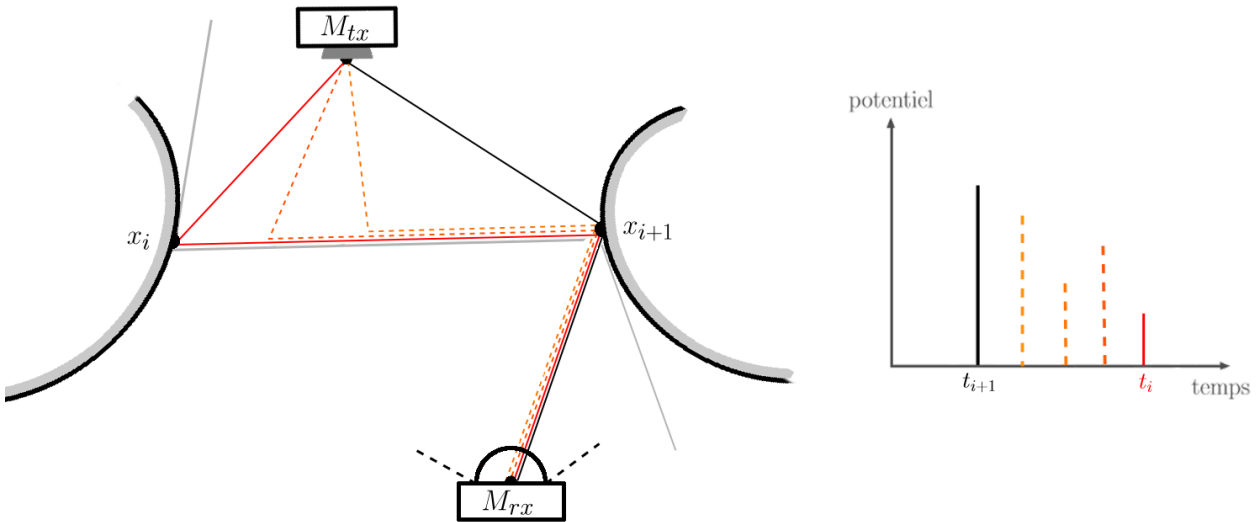


FIGURE 7.2 – Décomposition du rayon et réponse impulsive correspondante

Pour réaliser la découpe de ce rayon, j'utilise les équations de l'ellipse. En effet, l'inconnue

que nous cherchons c'est la distance entre x_i et un point intermédiaire p sur le rayon de direction $\overrightarrow{x_i x_{i+1}}$. Nous connaissons la distance entre x_i et M_{tx} , notons-la $d1$, qui sont des points fixes. Nous connaissons également la longueur totale $L = x_i p + p M_{tx}$, que l'on peut trouver en calculant les temps t_i et t_{i+1} . Cette distance étant fixée, nous sommes dans un problème d'ellipse : deux foyers (x_i et M_{tx}) et un paramètre (la longueur L). Connaissant l'angle $\theta = \angle M_{tx} x_i p$, l'inconnue $r(\theta) = x_i p$ est donc calculable car il s'agit de la distance entre un point de l'ellipse et un foyer,

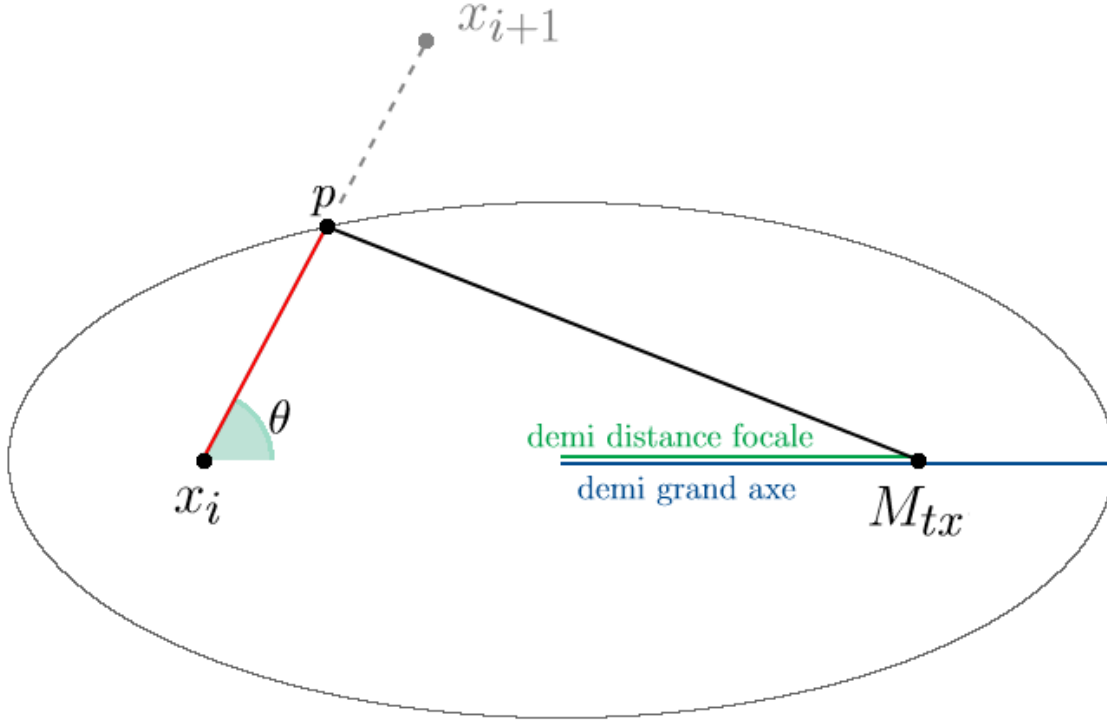


FIGURE 7.3 – Utilisation de l'ellipse pour trouver la distance $x_i p$

Voici les équations utilisées pour cela :

- La longueur du chemin passant par le point p est notée $L = x_i p + p M_{tx}$
- La longueur du demi grand-axe est noté $a = L/2$
- La distance séparant le centre de l'ellipse et un des foyers, notée $c = x_i M_{tx}/2$ est la demi-distance focale
- L'excentricité est $e = c/a$
- Le "paramètre" de l'ellipse est $\gamma = a(1 - e^2)$
- Enfin, l'équation polaire d'une ellipse nous donne $r(\theta) = \frac{\gamma}{1 + e \cos \theta}$

Ce qui amène l'algorithme suivant :

```

1 private double[] computeStepSizeIn(Ray ray, double stepTime){
2     // distance de la source au debut du rayon (de M_tx a x_i)
3     double d1 = Ray.createFromFromAndTo(this.irf.transmitter.getPosition(), ray.
from).getDistance();
4     // distance de la source a la fin du rayon (de M_tx a x_{i+1})
5     double d2 = Ray.createFromFromAndTo(this.irf.transmitter.getPosition(), ray.to)
.getDistance();

```

```

6
7 // Il va maintenant falloir trouver l'angle theta.
8 // Pour cela, on va d'abord chercher la hauteur du triangle (x_i - p - M_tx)
9 // en utilisant la formule de Heron
10 double perimeter = d1 + d2 + ray.getDistance();
11 double p = perimeter/2.0;
12 double area = Math.sqrt(p * (p-d1) * (p-d2) * (p-ray.getDistance()));
13 double h = area * 2.0 / ray.getDistance(); // la hauteur provenant de la source
14 M_tx
15 // On peut maintenant trouver l'angle theta avec la formule de trigonometrie
16 // suivante :
17 // dans un triangle rectangle, nous avons
18 // sin(angle) = (longueur du cote oppose)/(longueur de l'hypotenuse)
19 // Ici, l'angle theta est celui du sommet x_i
20 double theta = Math.asin(h/d1);
21 // On peut egalement calculer les temsp t1 et t2
22 double t1 = d2/physics.getSpeed();
23 double t2 = (d1 + ray.getDistance())/physics.getSpeed();
24
25 // et en deduire le nombre de points intermediaires qu'il doit y avoir
26 // entre x_i et x_{i+1}
27 int nbPoints = (int)Math.floor((t2 - t1)/stepTime);
28 double[] distancesToRayStart = new double[nbPoints+1];
29
30 double[] L = new double[nbPoints+1];
31 distancesToRayStart[0] = 0D;
32 // pour chaque point, on va pouvoir derouler les equations presentees plus haut
33 for (int i = 1; i < distancesToRayStart.length - 1; ++i){
34     L[i] = (t1 + i*stepTime) * physics.getSpeed();
35
36     // Ellipse
37     double a = L[i]/2D; // grand axe
38     double c = d1/2D; // demi-distance focale
39     double e = c/a; // excentricite
40     // utilisation de l'equation polaire
41     double A_X = (a*(1-e*e))/(1+e*Math.cos(theta));
42     distancesToRayStart[i] = A_X;
43 }
44 distancesToRayStart[distancesToRayStart.length - 1] = ray.getDistance();
45 return distancesToRayStart;
46 }

```

7.4 Problème de temporalité

Je choisis intentionnellement de commencer ma décomposition à l'entrée du milieu participant. En effet, entre p et p_0 , aucune particule ne viendra modifier la radiance et nous avons donc :

$$L_o(p, \omega) = L_i(p_0, \omega).$$

De la même manière, à la sortie du milieu participant il ne sert plus à rien de décomposer.

Nous aurons :

$$L_i(p', \omega) = L_o(p_n, \omega).$$

Bien évidemment, il est tout à fait possible de fusionner p et p_0 si dans un premier temps on considère que la fumée remplit toute la scène. Dans ce cas, p' se trouvera aussi dans la fumée et on calculera $L_i(p', \omega)$ comme n'importe quel $L_i(p_i, \omega)$.

Avec ce que nous savons, nous pourrions décomposer le calcul de la radiance en $n + 1$ itérations, en suivant ce schéma :

$$\begin{aligned} L_i(p_0, \omega) &= L_o(p, \omega), \\ L_o(p_0, \omega) &= L_i(p_0, \omega) - \sigma_t(p_0, \omega)L_i(p_0, \omega) + L_s(p_0, \omega), \\ L_i(p_1, \omega) &= T_r(p_0 \longrightarrow p_1)L_o(p_0, \omega), \\ L_o(p_1, \omega) &= L_i(p_1, \omega) - \sigma_t(p_1, \omega)L_i(p_1, \omega) + L_s(p_1, \omega), \\ &\dots \\ L_i(p_n, \omega) &= T_r(p_{n-1} \longrightarrow p_n)L_o(p_{n-1}, \omega), \\ L_o(p_n, \omega) &= L_i(p_n, \omega) - \sigma_t(p_n, \omega)L_i(p_n, \omega) + L_s(p_n, \omega), \\ L_i(p', \omega) &= L_o(p_n, \omega). \end{aligned}$$

Plus généralement, mise à part éventuellement aux extrémités de la décomposition, nous aurions les deux calculs suivants sur lesquels il faudrait itérer (cf. figure 7.4) :

$$L_i(p_k, \omega) = T_r(p_{k-1} \longrightarrow p_k)L_o(p_{k-1}, \omega), \quad (7.3)$$

$$L_o(p_k, \omega) = L_i(p_k, \omega)(1 - \sigma_t(p_k, \omega)) + L_s(p_k, \omega). \quad (7.4)$$

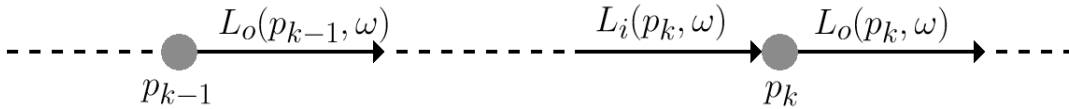


FIGURE 7.4 – Zoom sur la décomposition

Mais il faut prendre en compte que la lumière provenant directement de la source et correspondant à notre in-scattering, ne parcourt pas le même chemin que le rayon initialement lancé qui a effectué plusieurs réflexions (figure 7.5). Le récepteur ne recevra pas leurs contribution au même moment (figure 7.6). Il n'est donc pas possible d'itérer comme présenté précédemment puisque $L_s(p_3, \omega)$ ne correspond pas au même pic de potentiel que $L_o(p_2, \omega)$ par exemple.

Il faudra compléter la réponse impulsionnelle pour chaque point intermédiaire p_k à des temporalités différentes. Et non pas sommer toutes les contributions, dues à l'in-scattering, sur une même temporalité.

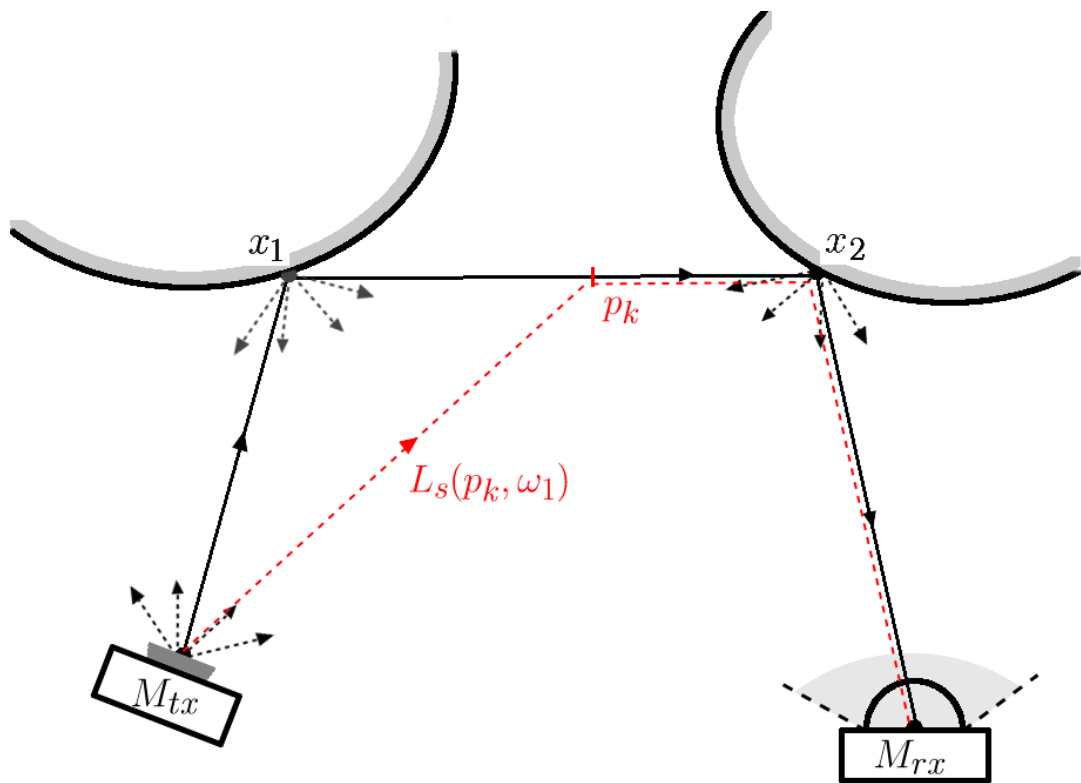


FIGURE 7.5 – Le chemin des réflexions (ligne pleine) est plus long que celui débuté par l'in-scattering (pointillés)

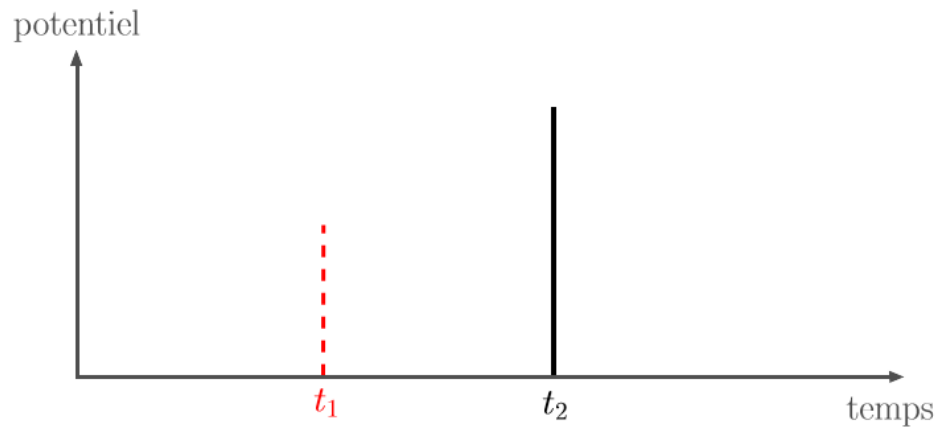


FIGURE 7.6 – La contribution de l'in-scattering est donc reçue plus tôt en M_{rx}

7.5 Algorithme

Dans ma classe *ParticipatingMedia*, j'ajoute une méthode qui sera utilisée pour trouver la réponse impulsionnelle finale.

inScatteringAlongARay Cette méthode calcule la contribution de la source en plusieurs points le long du rayon de direction ω : c'est l'in-scattering. Pour chaque point intermédiaire p_i (appelons-

les *point de milieu* en opposition aux *points de surface*, je calcule la source grâce à la fonction de phase appliquée au point p_i sur le rayon ω . Je multiplie cette source $L_s(p_i, \omega)$ par la transmittance car elle passe également dans la fumée avant d’atteindre le point p_i . Puis finalement, j’ajoute cette contribution lumineuse à la réponse impulsionnelle qui sera réutilisée dans une nouvelle itération de la méthode **RayLaunching**.

J’utilise toujours la méthode suivante :

multiplyByTransmittance Cette méthode ne s’occupe pas de l’in-scattering mais de l’extinction le long du rayon. Puisque ceci correspond au trajet de base du rayon (entre deux points de réflexions trouvé dans la méthode **RayLaunching**), il n’y a pas besoin de s’occuper de la réponse impulsionnelle ici. En effet, ceci est déjà géré dans la méthode **RayLaunching**.

Il aura également fallut que je modifie la classe *ImpulseResponse* en y ajoutant deux méthodes.

addTime Après avoir calculé une réponse impulsionnelle potentielle comprenant tout l’in-scattering entre deux points de réflexions x_i et x_{i-1} , l’algorithme réalise un rebond vers un autre point de réflexion x_{i+2} . Ce nouveau rayon est une distance supplémentaire parcourue par les contributions d’in-scattering sur tous les rayons précédents. Il faut donc, dans la réponse impulsionnelle potentielle, décaler tous les champs en fonction de cette nouvelle longueur. Cette méthode prend donc en paramètre un temps. Il s’agit de la distance du nouveau rayon obtenu après rebond, divisé par la vitesse de la lumière. Et met à jour la réponse impulsionnelle.

addDepth De la même manière, lorsque ce rebond est effectué, la distance du chemin augmente mais le nombre de réflexions également. L’in-scattering, lorsqu’il est calculé sur le rayon de direction ω_i , est ajouté à une profondeur de 1 puisqu’il ne correspond à aucune réflexion (il vient directement de la source sur un point de milieu sur le rayon), et ceux peut importe le rayon ω_i sur lequel on travaille. A chaque rebond, il faut donc indiquer que ces contributions ajoutées préalablement dans la réponse impulsionnelle potentielle, effectuent un rebond de plus.

7.6 Résultats

Incorrects pour l’instant. Travail en cours.

Chapitre 8

Algorithme 3

8.1 Présentation

Le troisième algorithme applique la technique du NEE en chaque point intermédiaire. Ainsi, chaque point p_k sera connecté au récepteur où sa contribution sera prise en compte.

8.2 Découpe du rayon

Ici, il faut procéder comme pur l'algorithme 3, mais les foyers de l'ellipse seront les point x_i et M_{rx} puisque nous voulons connecter chaque point intermédiaire p au récepteur afin d'appliquer le NEE (et non plus à l'émetteur).

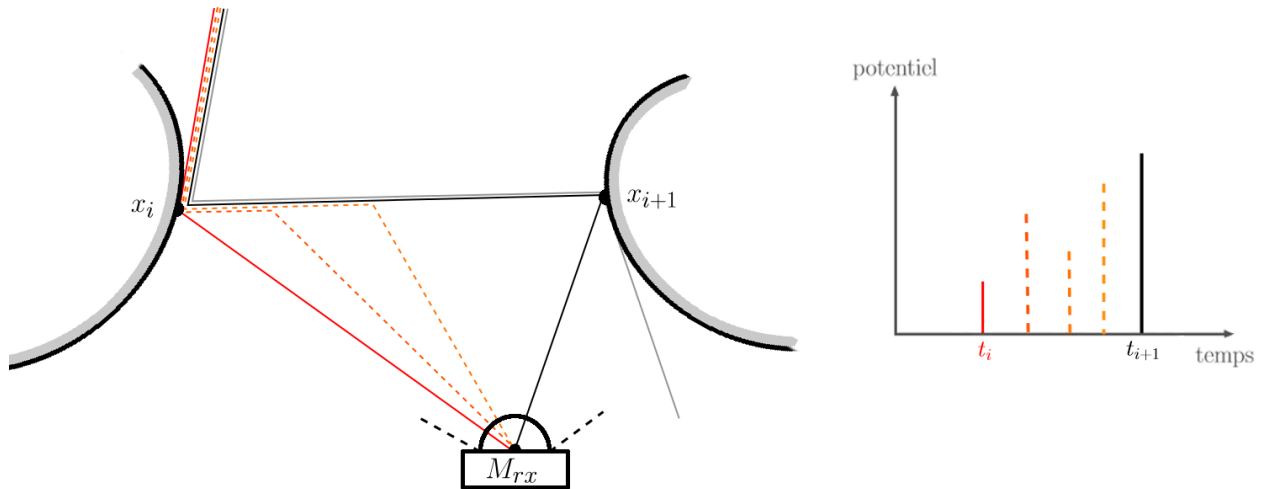


FIGURE 8.1 – Décomposition du rayon et réponse impulsive correspondante

8.3 Algorithme

Je rajoute, une méthode **outScatteringAlongARay** qui gère la connexion au récepteur M_{rx} pour chaque point intermédiaire (point de milieu), en simulant la réception par le récepteur, et en

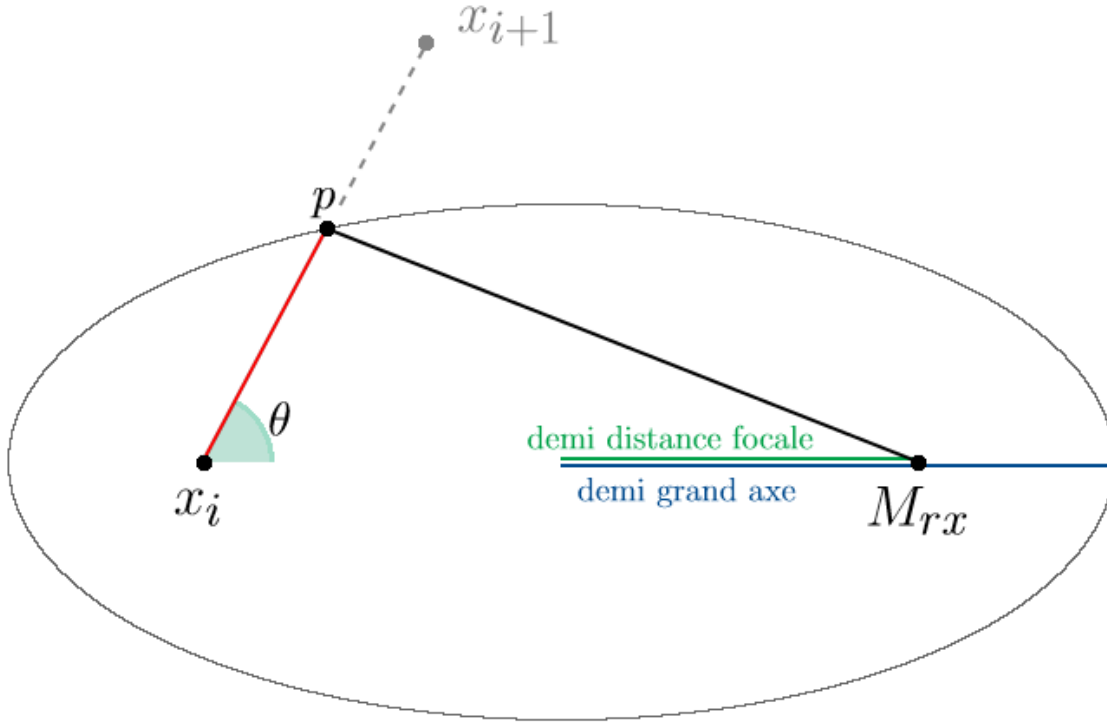


FIGURE 8.2 – Utilisation de l'ellipse pour trouver la distance $x_i p$

complétant la réponse impulsionnelle.

8.4 Résultats

Suite à un problème avec Scilab, je n'ai pas pu réaliser des graphes comme à l'algo 1, c'est-à-dire un graphe par profondeur.

8.4.1 Coefficient de dispersion

Dans un premier temps, nous regardons l'impact du coefficient de dispersion sur les contributions lumineuses. Ainsi :

- le coefficient d'atténuation σ_a vaut 0 ;
- le paramètre g vaut 0 : le milieu est complètement isotropique.
- la densité du milieu est de 1 ;
- 3 réflexions.

En lançant 10^7 rayons, nous obtenons le résultat de la figure 8.3.

On remarque que l'out-scattering produit dans un premier temps un effet d'amplification de la lumière. Le récepteur reçoit en permanence des rayons qui viennent contribuer, même faiblement. Cela dit, au bout d'un moment le coefficient d'extinction devient trop fort et la lumière est tellement

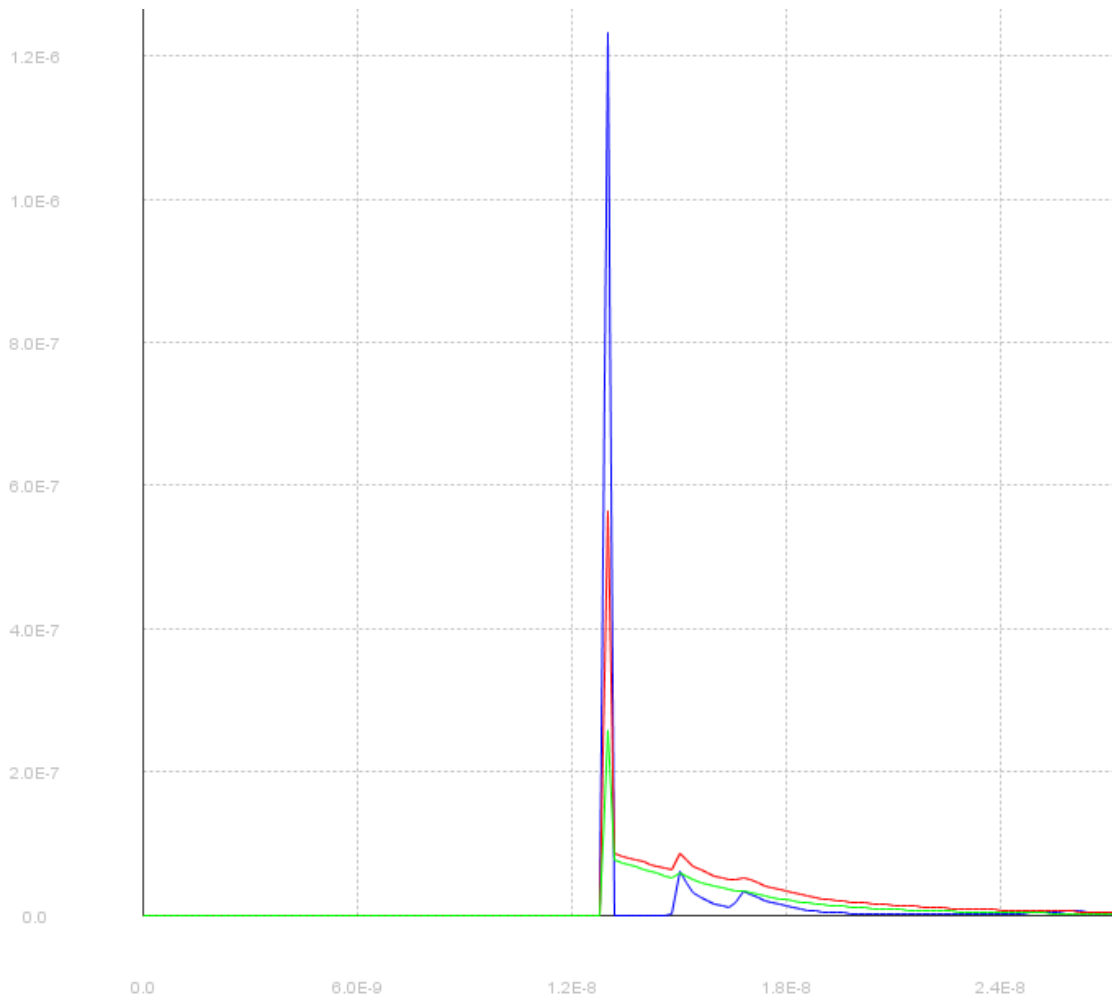


FIGURE 8.3 – bleu : $\sigma_s = 0.0$ rouge : $\sigma_s = 0.2$ vert : $\sigma_s = 0.4$

atténuée que les contributions de l'out-scattering ne sont plus suffisantes pour avoir un gain de lumière.

8.4.2 Paramètre g

Rappelons que le paramètre g d'un milieu participant indique si celui-ci est isotropic ($g = 0$), induit une majorité de backward-scattering ($-1 < g < 0$) ou une majorité de forward-scattering ($0 < g < 1$) (cf section 3.2).

Pour les résultats suivants, je n'ai fait varier que le paramètre g et les autres valeurs étaient :

- le coefficient d'atténuation σ_a vaut 0 ;
- le coefficient de dispersion σ_s vaut 0.2 ;
- la densité du milieu est de 1 ;
- 3 réflexions.

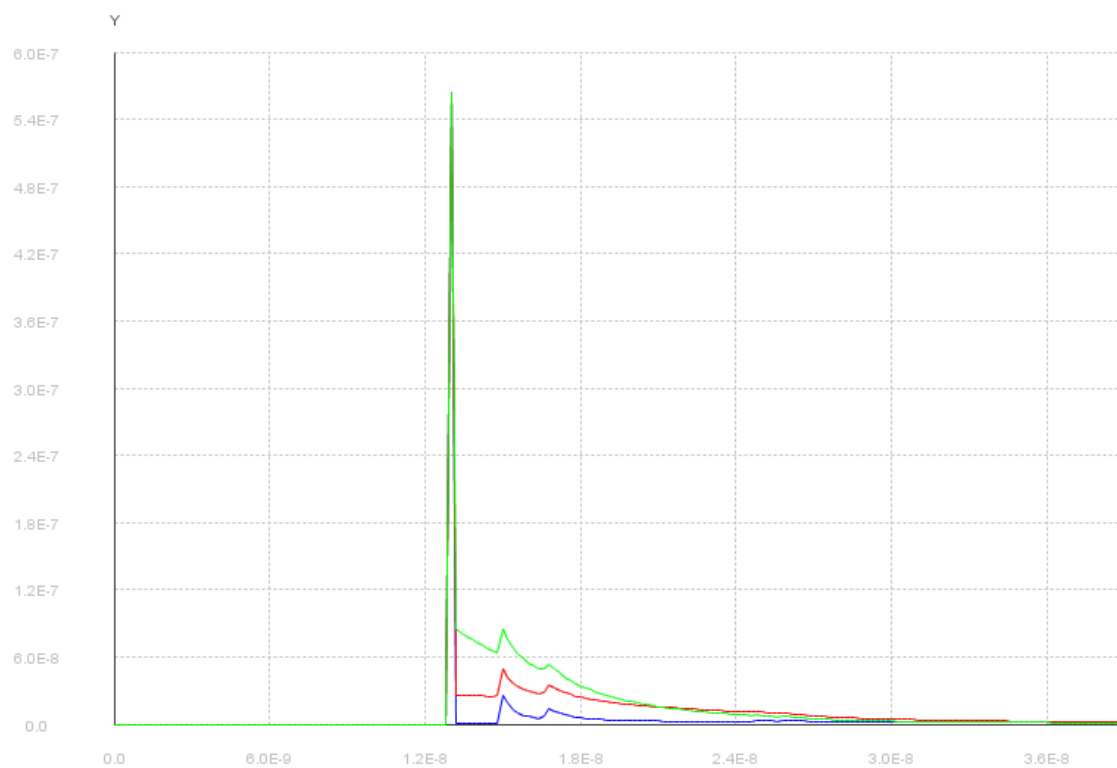


FIGURE 8.4 – backward-scattering : vert : $g = 0$ rouge : $g = -0.5$ bleu : $g = -0.9$

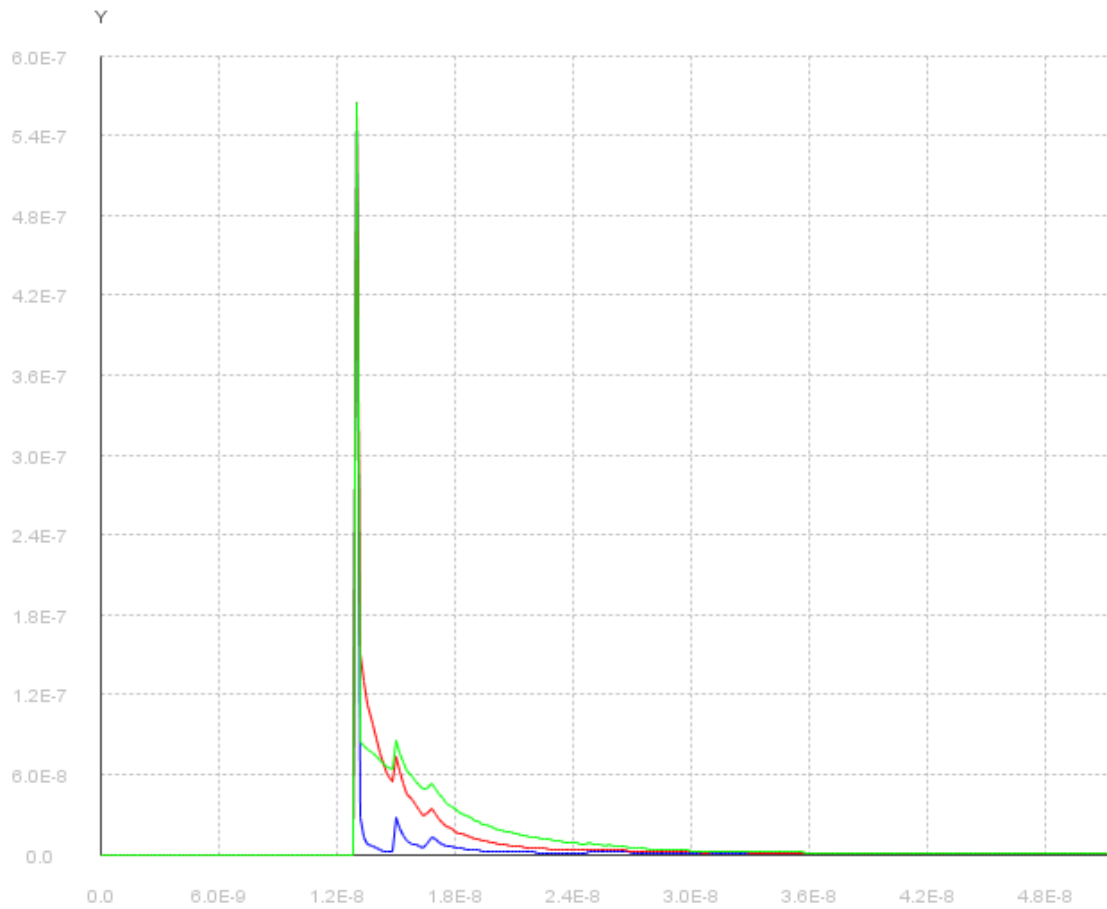


FIGURE 8.5 – forward-scattering : vert : $g = 0$ rouge : $g = 0.5$ bleu : $g = 0.9$

Chapitre 9

Algorithme 4

9.1 Présentation

Le quatrième algorithme fusionne les algo 2 et 3. Il faut effectuer deux découpes de chaque rayon, l'une pour l'in-scattering et l'autre pour le NEE. Mais la contribution de l'in-scattering en un point p doit être prise en compte sur chaque point où l'on appliquera la NEE, si celui-ci est situé après le point p .

9.2 Découpes du rayon

Attention, pour cet algorithme il y aura besoin de deux découpes différentes : une pour appliquer l'in-scattering, l'autre pour l'out-scattering. Ces deux découpes ne créeront pas les mêmes points, mais les contributions de tous les in-scattering le long du rayon sur des points situés avant un point p_o devront être prises en compte lorsqu'on y calculera l'out-scattering.

Chapitre 10

Algorithme 5

10.1 Présentation

Celui-ci est différent des autres. Au lieu de tirer un rayon en ligne droite et de simuler les interactions avec la fumée grâce à des coefficients d'absorption ou de dispersion, nous allons en chaque point p_k (point dans le volume) estimer la probabilité de dévier dans une certaine direction. Cette approche est très semblable à l'approche actuelle (sans milieu participant) où sur chaque point surface rencontré nous tirons une nouvelle direction. Ici, régulièrement dans la fumée, nous tirons éventuellement une nouvelle direction sur la sphère complète (et non plus juste l'hémisphère). Ceci est faisable grâce à la fonction de phase, tout comme on le faisait avec la BRDF jusqu'à présent. Bien évidemment, la technique du NEE y est aussi applicable en chaque point p_k .

10.2 Idée

Ici, il faudra dévier le rayon à l'intérieur du milieu et plus nécessairement sur des points de surface. Pour cela, il faudra utiliser la fonction de phase, qui joue un rôle semblable à la BRDF mais pour un milieu participant. Bien évidemment, les échantillonnages de nouvelles direction ne se feront plus sur un hémisphère mais sur une sphère complète. De plus, la probabilité de dévier dépend des paramètres du milieu, de ce fait un rayon ne dévie pas régulièrement selon un pas fixé.

Je n'ai pas eu le temps de m'y mettre mais il serait très intéressant de créer une interface pour les fonctions de phase, tout comme il en existe déjà une pour les BRDF.