

# Calcul par la méthode des images de la réflexion spéculaire dans des enceintes prismatiques convexes

**Benoit Beckers**

Architecture et Physique Urbaine - ISA BTP

Université de Pau et des Pays de l'Adour

Allée du Parc Montaury, 64600 Anglet (France)

[benoit.beckers@univ-pau.fr](mailto:benoit.beckers@univ-pau.fr), [www.heliodon.net](http://www.heliodon.net), [+33 \(0\)5 59 57 44 25](tel:+33559574425)

## Table des matières

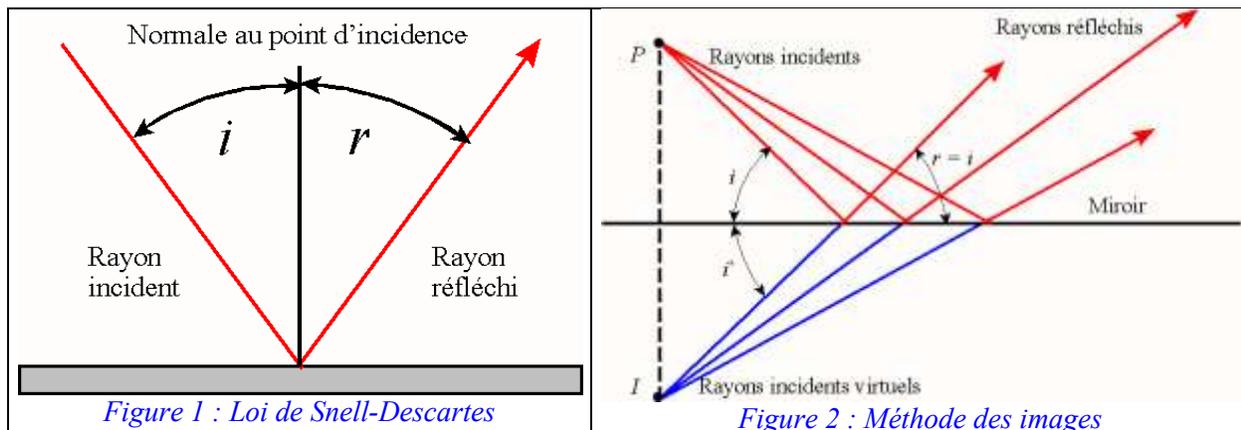
<b>1. Introduction</b> .....	1
<b>2. Réflexion spéculaire</b> .....	2
<b>3. Formules de base</b> .....	3
<b>4. Définition de la géométrie de l'enceinte</b> .....	6
<b>5. Analyse de deux enceintes hexaédriques</b> .....	6
2.1 Chemins comportant une seule réflexion .....	6
2.2 Chemins comportant deux réflexions.....	8
<b>6. Validation des chemins comportant 5 réflexions</b> .....	9
<b>7. Evaluation de chemins comportant <math>n</math> réflexions</b> .....	11
<b>8. Conclusion</b> .....	14
<b>9. Lectures utiles</b> .....	14
<b>10. Annexe I : La procédure de base</b> .....	15
<b>11. Annexe II : Visualisation des résultats</b> .....	19
<b>12. Liste des figures et des tables</b> .....	23

## 1. Introduction

L'objectif de ce travail est de présenter tous les aspects du calcul de la réflexion spéculaire utilisant la méthode des images [Allen 1979, Borish 1984, Beckers 1992], ainsi que les procédures permettant d'effectuer ces calculs. La procédure de calcul des images concernant un rayon se réfléchissant sur les parois 1, 2, ...,  $k$  d'une enceinte prismatique convexe constitue le cœur de la méthode (procédure *Rechem.m*). Les tests conduisent à la conclusion que la méthode est rapide, fiable et facile à mettre en œuvre. Ce travail ne fait pas appel à des références spécifiques, mais nous proposons aux lecteurs intéressés une série de lectures utiles (§ 9).

## 2. Réflexion spéculaire

En optique géométrique, lorsque la réflexion est spéculaire, selon les lois de Snell-Descartes, le plan du rayon incident et du rayon réfléchi est perpendiculaire au plan du miroir et l'angle d'incidence  $i$  formé par le rayon incident et la normale au plan est égal à l'angle de réflexion  $r$  formé par le rayon réfléchi et la normale (*Figure 1*).



Au lieu de réaliser un calcul d'angles par les méthodes de la trigonométrie, on peut créer la symétrique  $I$  d'un point  $P$  du rayon incident par rapport au plan du miroir. En liant ce point au point d'incidence du rayon, on obtient le rayon réfléchi sans effectuer de calcul d'angle (*Figure 2*).

En 3D, le raisonnement est le même (*Figure 3*). Nous appellerons cette méthode : méthode des images. Elle permet de résoudre de manière très élégante le problème des réflexions spéculaires multiples.

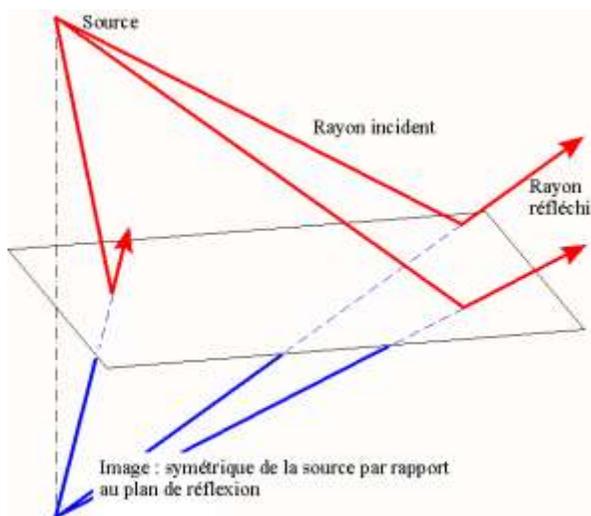


Figure 3 : Illustration de la méthode des images en 3D

Le problème illustré à la *Figure 4* est formulé de la manière suivante. Il s'agit de déterminer le trajet d'un rayon qui part de la source et se réfléchit spéculairement sur les parois 6, 3 et 5 de l'enceinte, avant d'atteindre le récepteur.

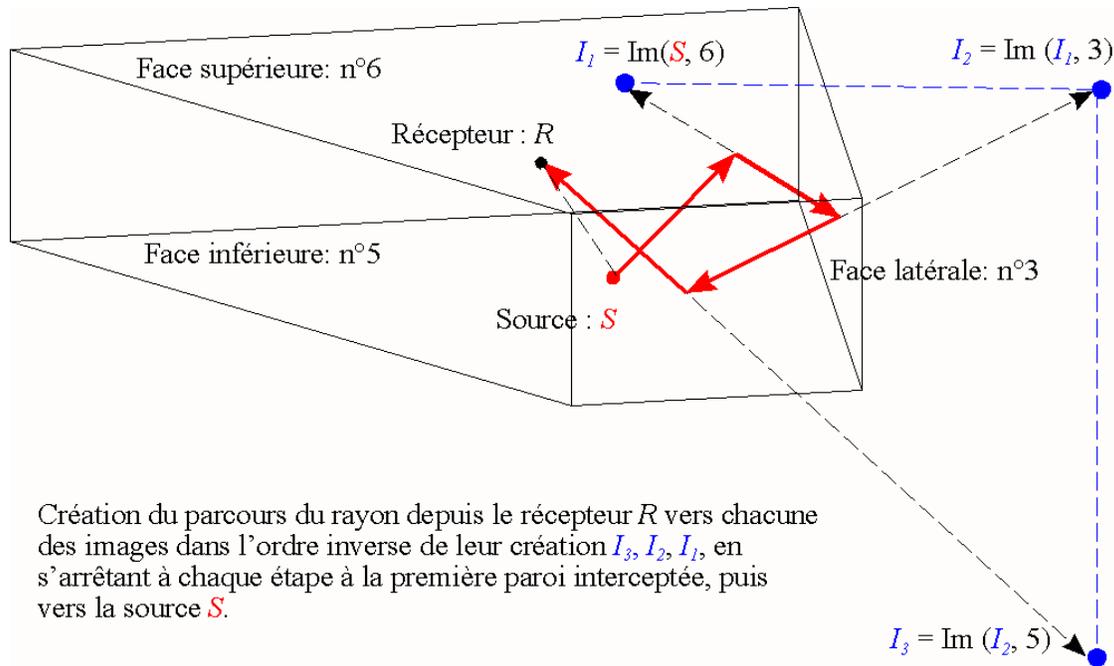


Figure 4 : Réflexion multiple dans une enceinte hexaédrique.

Le processus se déroule en deux étapes.

**Étape 1 :** On crée tout d'abord les images sur la suite des parois sur lesquelles se réfléchit le rayon, en commençant par l'image de la source sur la première paroi. Dans l'exemple traité, il s'agit de la paroi 6. On crée donc le point  $I_1 = \text{Im}(S, 6)$ , image de la source par rapport à la paroi 6. Ensuite on crée l'image de cette image par rapport à la paroi 3 :  $I_2 = \text{Im}(I_1, 3)$  et enfin l'image de cette dernière par rapport à la paroi 5 :  $I_3 = \text{Im}(I_2, 5)$ . Ces points sont représentés sur les extrémités et au sommet intérieur de la ligne brisée en pointillés bleus de la Figure 4.

**Étape 2 :** Cette étape consiste à créer le trajet du rayon (flèches épaisses rouges commençant soit sur une paroi, soit sur la source). On part donc du récepteur vers l'image  $I_3$  en s'arrêtant sur la paroi 5 (on effectue le test de validité : vérifier que la paroi 5 est la première interceptée), le rayon se dirige ensuite vers l'image suivante  $I_2$  en s'arrêtant sur la paroi 3 (test de validité : on vérifie que la paroi 3 est la première interceptée), puis vers la suivante  $I_1$  en s'arrêtant sur la paroi 6 (test de validité : on vérifie que la paroi 6 est la première interceptée) et enfin le rayon rejoint la source  $S$ . Si un test de validité n'est pas vérifié, on doit arrêter le processus car il n'y a pas de solution au problème.

### 3. Formules de base

On présente une méthode rigoureuse pour calculer de manière complète toutes les réflexions d'ordre  $n$ . Les parois sont définies par les points  $W_i$  situés à leurs intersections avec leurs normales  $N_i$  passant par l'origine des coordonnées. A la Figure 5, en 2D, on a représenté les données des parois n° 1, 2 et 3 (points bleus) tandis qu'à la Figure 6, en 3D, on a dessiné en bleu les 6 normales aux parois issues de l'origine.

$$N_i = \frac{W_i}{|W_i|} \quad (1.1)$$

Ceci constitue une façon très compacte de définir une paroi comme une surface infinie, mais il est important de développer les outils permettant de manipuler les plans, limites de l'enceinte.

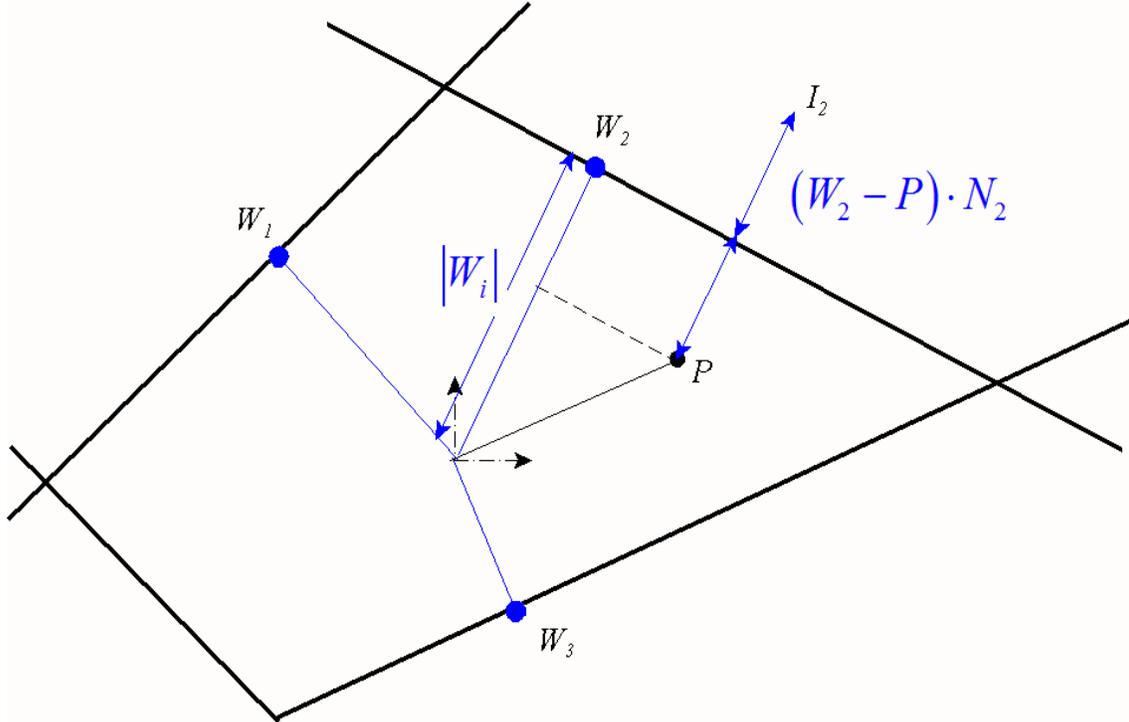


Figure 5: Schéma pour le calcul de l'image d'un point  $P$

La distance du plan  $W_i$  à l'origine est égale à  $|W_i|$ .

Comme la distance du point  $P$  à la paroi  $W_2$  est égale à  $(W_2 - P) \cdot N_2$ , le symétrique  $I_2$  d'un point  $P$  par rapport au plan  $W_2$  se trouve en :

$$I_2 = P + 2 \left( (W_2 - P) \cdot N_2 \right) N_2 \quad (1.2)$$

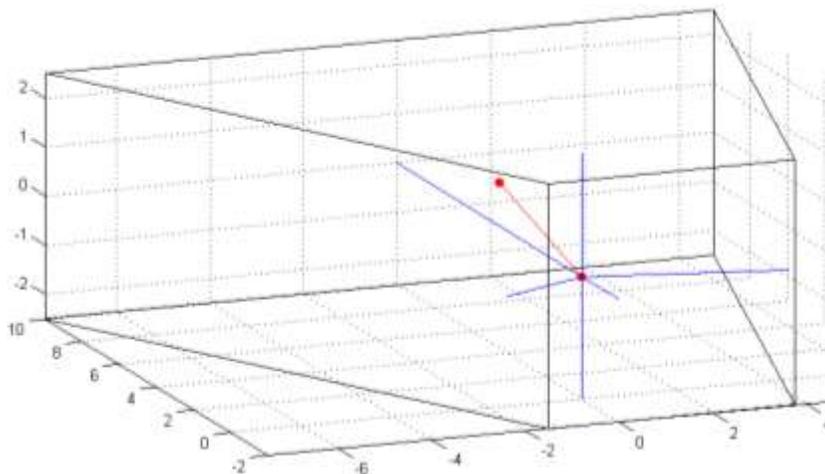


Figure 6: Définition de l'enceinte par les pieds des perpendiculaires abaissées de l'origine

Soit un rayon émis depuis la source  $S$  et arrivant au récepteur  $R$ . Il peut y arriver directement ou après un certain nombre de réflexions. Dans la suite nous calculons les différents trajets. Dans le cas du direct, la longueur du rayon est:

$$|R - S| \quad (1.3)$$

En présence d'une seule réflexion dans une enceinte composée de  $n$  parois, on calcule les  $n$  images de la source  $S$  par rapport aux  $n$  parois. Le premier indice de la variable  $I$  indique qu'il s'agit d'une image d'ordre 1, c'est-à-dire de l'image de la source. Les images de ces images (*Figure 4*) portent les indices 2, 3, ...

$$I_{li} = S + 2 \left( (W_i - S) \cdot N_i \right) N_i \quad ; \quad i = 1, n \quad (1.4)$$

Le point de la paroi où le rayon se réfléchit pour atteindre le récepteur  $R$ , est situé en  $C$  sur la droite liant l'image au récepteur :

$$C = R(1-t) + I_{li} t \quad (1.5)$$

On exprime que le point est situé sur la paroi, en écrivant que sa distance à l'origine est  $|W_i|$  :

$$C \cdot N_i = |W_i| \quad (1.6)$$

Donc, par (1.7) :

$$R \cdot N_i (1-t) + I_{li} \cdot N_i t = |W_i| \quad (1.8)$$

La distance paramétrique où se trouve le point d'intersection est donc :

$$t = \frac{|W_i| - R \cdot N_i}{(I_{li} - R) \cdot N_i} \quad (1.9)$$

Finalement, le point de contact du rayon est situé en :

$$C = R + \frac{|W_i| - R \cdot N_i}{(I_{li} - R) \cdot N_i} (I_{li} - R) \quad (1.10)$$

Pour calculer le rayon, il suffit de viser la dernière image depuis le récepteur. On calcule l'intersection de ce rayon avec la paroi concernée par la dernière image. Dans le cas d'une seule réflexion, on tire vers l'émetteur depuis ce point et le processus est terminé. Sinon, on tire vers l'avant-dernière image, on calcule le point de contact et ainsi de suite.

Le processus n'est valide que si le rayon coupe effectivement la paroi concernée par l'image vers laquelle le tir est réalisé. C'est le cas s'il n'est pas empêché par une paroi intermédiaire, ce qui invalide la séquence d'images sur laquelle on s'appuie.

Lorsqu'on cherche un point de contact, on calcule donc l'intersection du rayon avec la paroi la plus proche. Si c'est la paroi souhaitée, on continue le processus, sinon, on invalide la séquence.

La longueur du rayon est donc égale à la somme des longueurs des segments intérieurs de la ligne brisée joignant la source au récepteur, elle-même égale à la longueur du segment joignant la dernière image au récepteur.

$$|I_i - R| = |C_i - R| + \sum |C_i - C_{i-1}| + |C_1 - S| \quad (1.11)$$

## 4. Définition de la géométrie de l'enceinte

Pour définir l'enceinte, on ne procède pas de manière classique en décrivant la forme polyédrique par des carreaux et leurs sommets. On introduit plus simplement chaque face du polyèdre comme un plan infini. Sa définition comporte uniquement le pied de la perpendiculaire abaissée de l'origine. Chaque plan est limité par les plans des faces adjacentes selon des arêtes qui ne sont pas calculées explicitement. Cette définition est suffisante pour le calcul des images et des rayons. Il faut cependant effectuer quelques calculs supplémentaires pour réaliser la représentation de l'objet (voir fonction *Desal.m* de la *Table 8*). La version actuelle est limitée à des polyèdres convexes pour le calcul des images et des rayons ainsi qu'à des prismes à bases horizontales pour le dessin de l'enceinte. Les enceintes utilisées dans ce document sont reprises dans la fonction *Salle.m* de la *Table 7*.

## 5. Analyse de deux enceintes hexaédriques

La première enceinte est hexaédrique (*Figure 6*), elle est définie par les points  $W_i$  de la *Table 1* au moyen de l'instruction : `[S,R,W] = Salle(1)` qui peut être introduite directement dans Matlab.

Points de la salle de la <i>Figure 6</i> . Cette liste peut être directement obtenue, avec la définition de la source et du récepteur, en entrant l'instruction:				
$W$	1	-2.0000	-1.0000	0
	2	0	-2.0000	0
	3	4.0000	-1.0000	0
	4	0	10.0000	0
	5	0	0	-2.5
	6	0	0	2.5

*Table 1: Définition de la première enceinte hexaédrique (fonction *Salle.m*)*

Les enceintes utilisées dans ce document sont reprises dans la procédure *Salle.m* de la *Table 7*.

### 2.1 Chemins comportant une seule réflexion

Les images de la source située à l'origine des coordonnées par rapport aux 6 parois sont calculées selon (1.2) ou :  $I_i = 2 W_i$ . Dans le cas particulier où la source est à l'origine, cela donne le résultat de la *Table 2*.

Images :	1	-4	-2	0
	2	0	-4	0
	3	8	-2	0
	4	0	20	0
	5	0	0	-5
	6	0	0	5

*Table 2: Images de la source de la salle 1 par rapport aux 6 parois*

En utilisant la procédure *Dechem.m*, avec un récepteur situé en `[1 7 0.2]`, on obtient les délais, pour les rayons réfléchis. Après l'exécution de *Reche.m*, il suffit d'entrer l'instruction :

```
sortrows ([seq(1:nchev,:) delay(1:nchev)],nr+1)
```

Face où se situe le point de contact (gauche) et délai en ms (droite).	
6	25
5	26
1	30
2	32
3	34
4	38

Table 3: Contacts et délais pour 1 réflexion dans l'enceinte n° 1 de la Table 7 (fonction Salle.m)

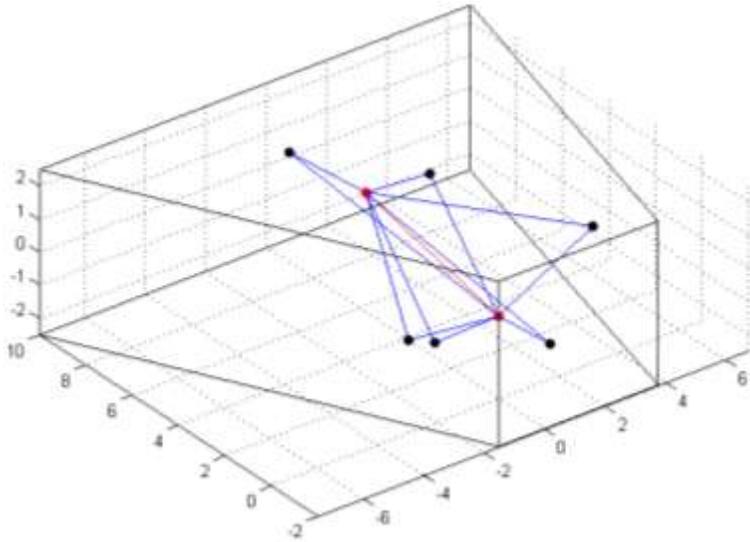


Figure 7: Salle 1, 1 rebond, 6 chemins et rayon direct

Les 6 rayons visualisés à la Figure 7 sont situés dans les 6 plans perpendiculaires aux 6 parois passant par la source et le récepteur. Le rayon direct est également représenté (en rouge). On voit aussi les directions d'arrivée des rayons sur le récepteur à la Figure 13.

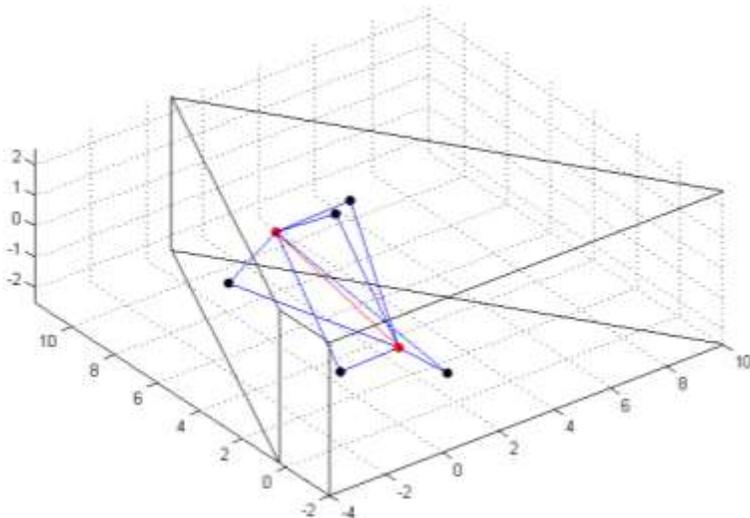


Figure 8: Salle 2, 1 rebond, 5 chemins et rayon direct

Dans l'enceinte de forme hexaédrique (salle 2 de la Table 7), la réflexion d'ordre 1 sur la paroi n°1 est impossible (Figure 8).

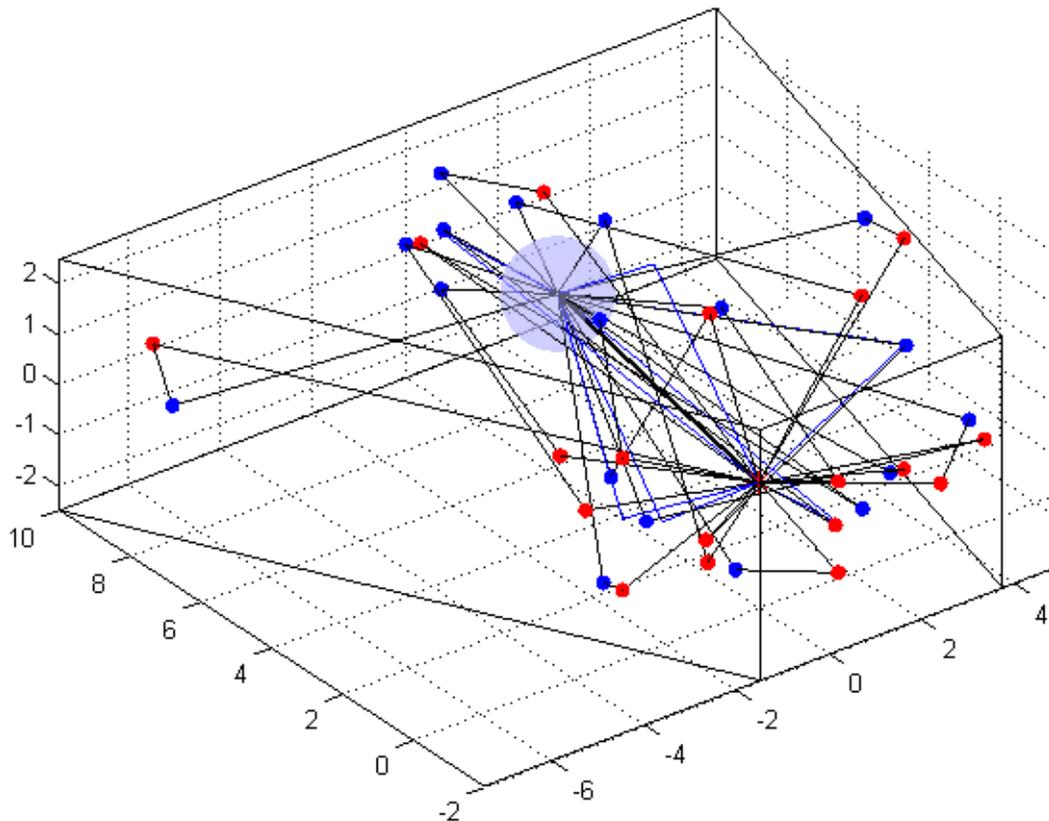
## 2.2 Chemins comportant deux réflexions

Pour ce calcul, on doit disposer des images de la source par rapport aux  $n$  parois ainsi que de leurs images par rapport aux  $(n-1)$  parois restantes. Pour l'enceinte considérée, on a donc au maximum  $n(n-1)$  chemins comportant 2 réflexions. Les rayons sont construits de la manière suivante :

1. On choisit une séquence quelconque de parois sur lesquelles se réalisent les 2 réflexions.
2. On calcule l'image de la source par rapport à la première paroi et ensuite l'image de cette image par rapport à la seconde paroi.
3. On relie ce point au récepteur par un rayon.
4. On calcule l'intersection de ce rayon avec toutes les parois. S'il les coupe après avoir coupé la seconde paroi de la séquence imposée, cette étape est validée. Si ce n'est pas le cas, le chemin est invalidé et on arrête le calcul.
5. On unit le point de contact qui vient d'être calculé à l'image de la source par un nouveau rayon.
6. On calcule l'intersection de ce rayon avec toutes les parois. S'il les coupe après avoir coupé la première paroi de la séquence imposée, le chemin est validé. Si ce n'est pas le cas, le chemin est invalidé.

La *Figure 9* présente les chemins comportant deux réflexions ainsi que leurs points de contact : rouges coté source et bleus coté récepteur. On y représente la source par un astérisque rouge et le récepteur, bien qu'il soit ponctuel, par une sphère transparente de rayon unitaire.

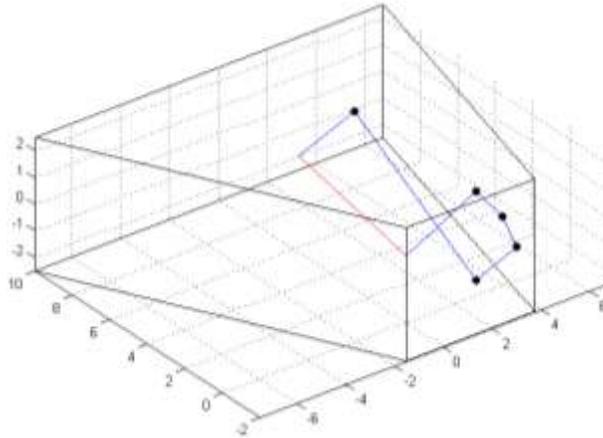
Evaluer les délais revient à calculer les longueurs des rayons. Cela peut être réalisé de deux manières : soit on calcule les distances entre les images de second ordre et le récepteur, soit, on calcule la longueur des rayons eux-mêmes. Les points bleus de la *Figure 9* sont également affichés en tant que directions dans la plate carrée de la *Figure 14*.



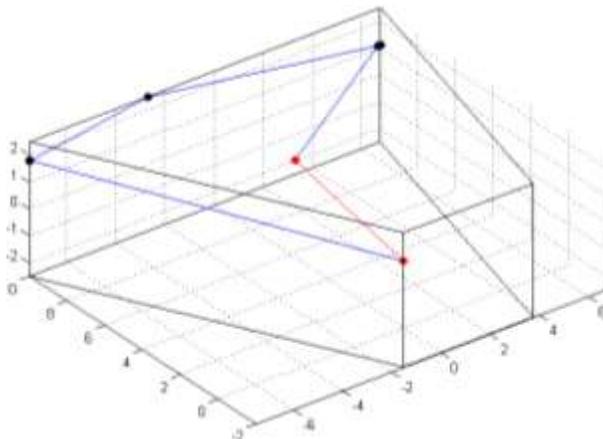
*Figure 9: Salle 1, 2 rebonds, 18 chemins incluant le coloriage des 2 rebonds*

## 6. Validation des chemins comportant 5 réflexions

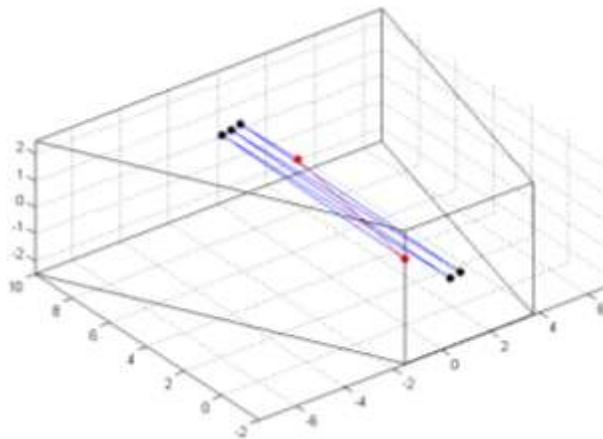
A l'aide de la procédure Rechem.m (*Table 6*), on calcule et valide les 100 chemins de la *Table 4* comportant 5 réflexions ou rebonds.



*Figure 10: Le plus court des 100 chemins à 5 rebonds, séquence 6 2 3 5 6, délai 59 ms (Deche.m)*



*Figure 11: Un chemin quasi singulier parmi les 100 détectés, séquence 4 1 6 3 4, délai : 99 ms*



*Figure 12: Le plus long des 100 chemins à 5 rebonds, séquence : 4 2 4 2 4, délai 179 ms*

Le chemin le plus court est représenté à la *Figure 10*. Le chemin [4 1 6 3 4] dont le délai est égal à 99 ms comporte deux doubles réflexions proches des arêtes verticales bordant la face 4 (*Figure 11*). La procédure de calcul est extrêmement robuste et peut prendre en compte des situations quasiment dégénérées.

Enfin, le dessin de la *Figure 12* correspond au plus long chemin. Ici, le rayon parcourt la salle entre les faces 2 et 4 éloignées l'une de l'autre de 12 mètres. Connaissant les positions de la source et du récepteur (fonction *Salle* de la *Table 7*), on en déduit la distance approximative, elle est de 61 m :  $340 \text{ m/s} \times 1000 = 179.4 \text{ ms}$

Les 100 chemins à 5 rebonds				
1	3	1	4	2
1	3	4	1	3
1	3	4	5	2
1	3	4	6	2
1	3	5	4	1
1	3	6	4	1
1	4	1	2	3
1	4	1	3	1
1	4	3	1	3
1	4	3	6	1
1	4	5	2	4
1	4	5	3	1
1	4	6	2	4
1	5	3	6	4
1	5	4	2	6
1	5	4	3	6
1	5	6	3	5
1	6	3	5	4
1	6	4	2	5
1	6	4	3	5
1	6	5	3	6
2	1	4	1	3
2	1	5	4	1
2	1	6	4	1
2	3	4	1	3
2	3	4	2	4
2	3	4	5	2
2	3	4	6	2
2	4	1	2	3
2	4	1	3	1
2	4	1	6	3
2	4	2	4	2
2	4	3	1	3
2	4	5	1	3
2	4	5	2	4
2	4	6	2	4
2	5	3	6	4
2	5	4	1	6
2	5	4	2	6
2	5	6	5	4
2	6	3	5	4
2	6	4	1	5
2	6	4	5	2
2	6	5	4	6
3	1	4	3	1
3	1	4	6	2
3	1	5	4	2
3	1	5	4	3
3	1	6	4	3
3	4	1	3	4
3	4	5	1	3
3	4	5	2	4
3	4	6	1	3
3	4	6	2	4
3	5	1	6	4
3	5	4	1	6
3	6	1	5	4

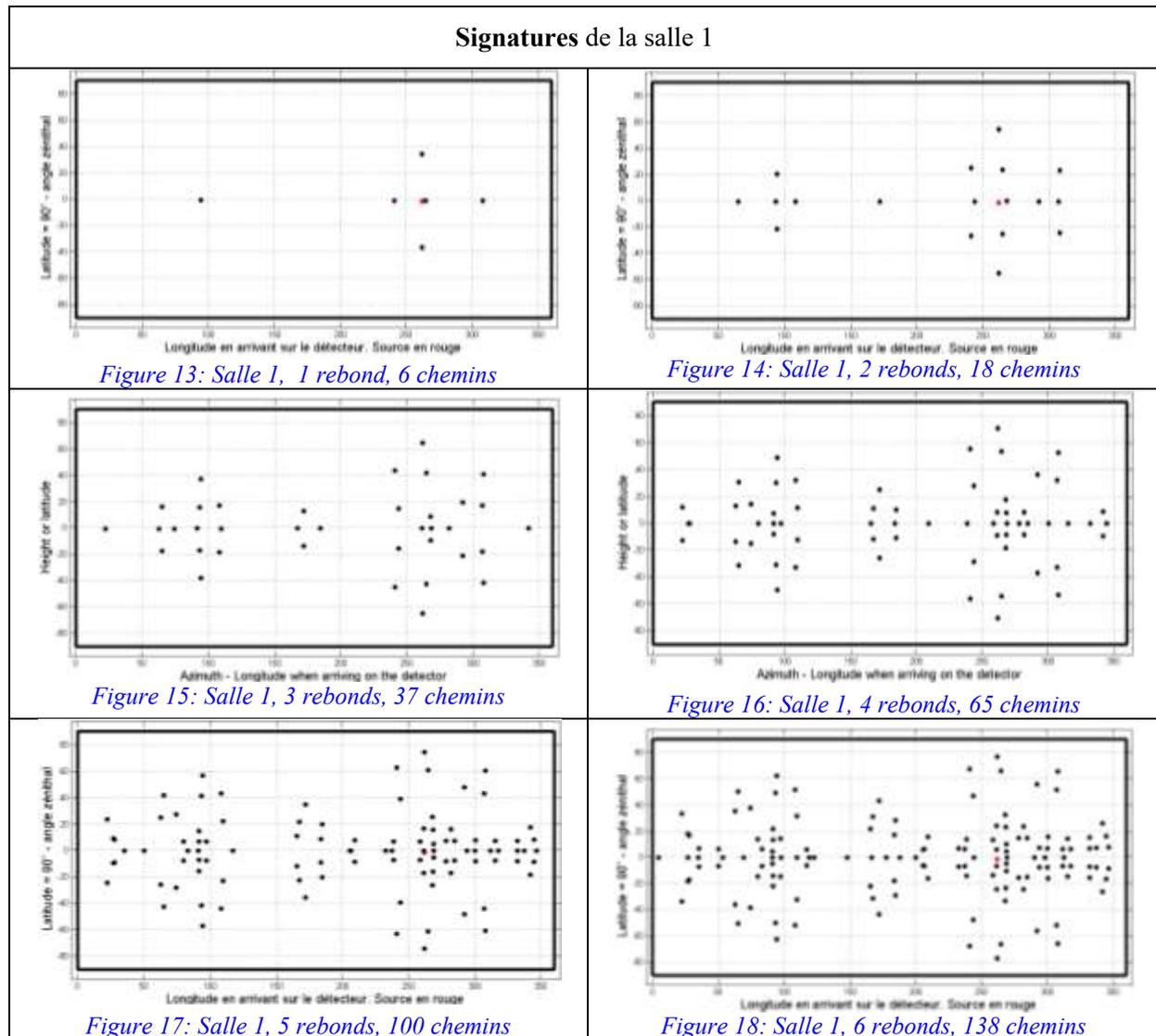
3	6	4	5	1
4	1	3	1	4
4	1	3	4	1
4	1	5	3	1
4	1	5	3	4
4	1	6	3	1
4	1	6	3	4
4	2	4	2	4
4	2	5	4	2
4	2	6	4	2
4	3	1	3	4
4	3	5	1	3
4	3	6	1	3
4	5	1	2	3
4	6	1	2	3
5	1	6	4	5
5	1	6	5	6
5	2	3	6	5
5	2	6	5	6
5	3	4	2	6
5	3	6	1	5
5	3	6	4	5
5	4	1	6	3
5	4	2	6	4
5	4	6	2	5
5	6	3	5	6
5	6	4	1	5
5	6	5	4	6
5	6	5	6	5
6	1	5	4	6
6	1	5	6	5
6	2	3	5	6
6	2	5	6	5
6	3	4	2	5
6	3	5	1	6
6	3	5	4	6
6	4	1	5	3
6	4	2	5	4
6	4	5	2	6
6	5	3	6	5
6	5	4	1	6
6	5	6	4	5
6	5	6	5	6

Table 4: Les 100 chemins comportant 5 rebonds, procédure Rechem.m (Table 6)

## 7. Evaluation de chemins comportant $n$ réflexions

Enceinte n°1, (Table 7)					
Nombre de rebonds	Chemins validés	Délai minimum en ms	Délai maximum en ms	Cpu – pc1	Cpu – pc2
1	6	25	38	0.0505	
2	18	33	91	0.0628	
3	37	42	109	0.1279	
4	65	49	162	0.5324	
5	100	59	179	2.5	2
6	138	71	232	14.531	8.77
7	183	81	250	83.467	49.98
8	236	92	303	404.42	274.22
9	292	94	324		1520
10	347	114	374	16063	10006

La visualisation des directions des rayons à leur arrivée sur le récepteur dans les situations où l'on considère de 1 à 6 rebonds est fournie pour la salle 1. On utilise la projection en “plate carrée” [Beckers 2014] car elle permet de lire directement, en degrés, les deux coordonnées de ces directions. Ce type de graphique fournit une signature de la salle étudiée.



<b>Enceinte n°2, (Table 7)</b>				
Nombre de rebonds	Chemins validés	Délai minimum en ms	Délai maximum en ms	Cpu – pc l
1	5	25	32	0.0292
2	13	30	65	0.0603
3	28	27	89	0.1264
4	49	30	101	0.5338
5	77	56	135	2.5

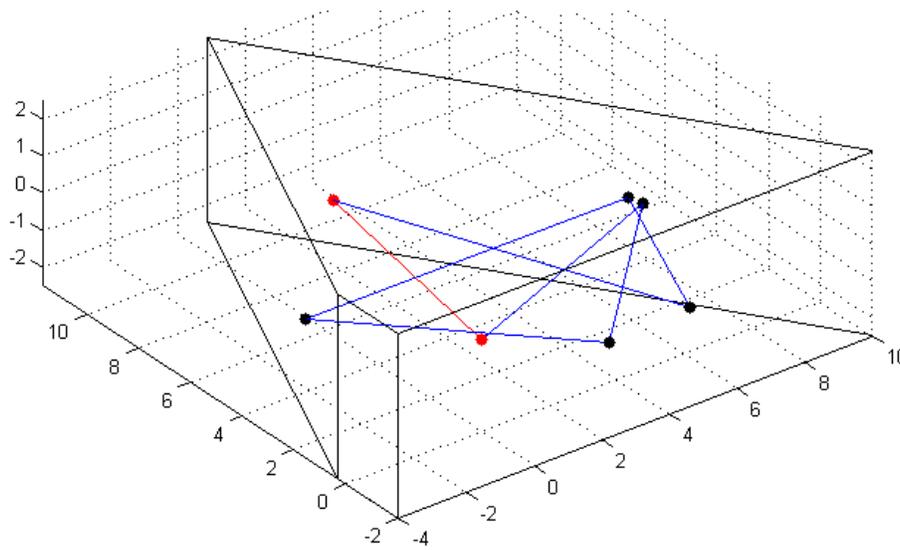


Figure 19: Salle 2 : 135 ms = plus long délai pour le cas de 5 rebonds sur 3 2 4 3 2

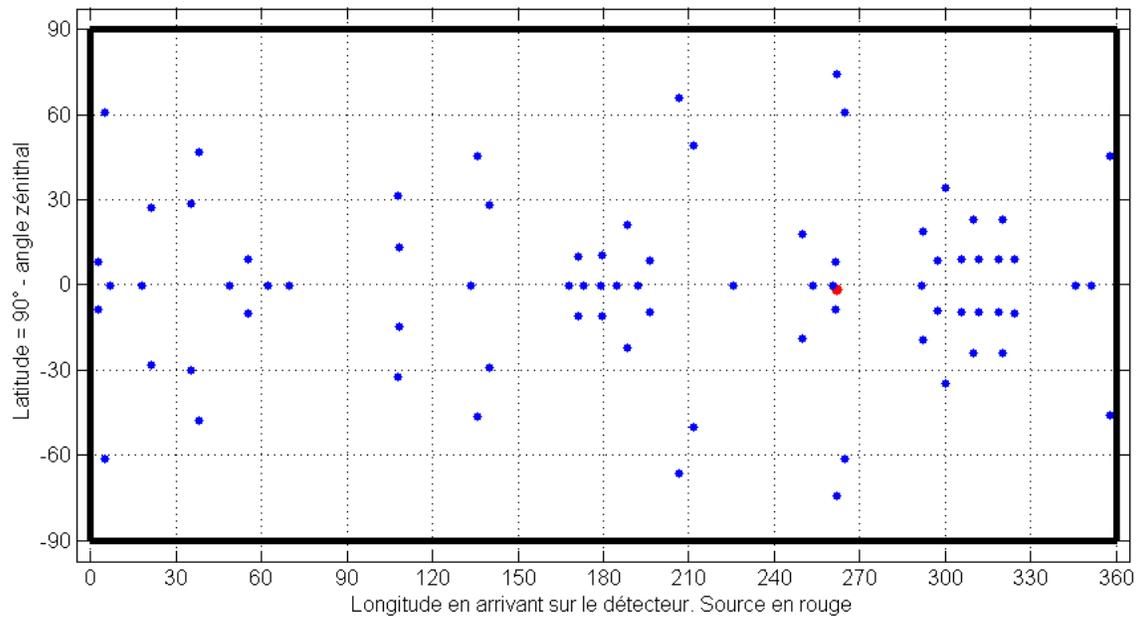


Figure 20: Salle 2, 5 rebonds, 77 chemins détectés

Enceinte n°3, (Table 7)				
Nombre de rebonds	Chemins validés	Délai minimum en ms	Délai maximum en ms	Cpu – pc1
1	5	25	32	0.0495
2	13	30	65	0.0670
3	25	39	89	0.1901
4	39	47	110	1.1156
5	59	56	136	6.6371

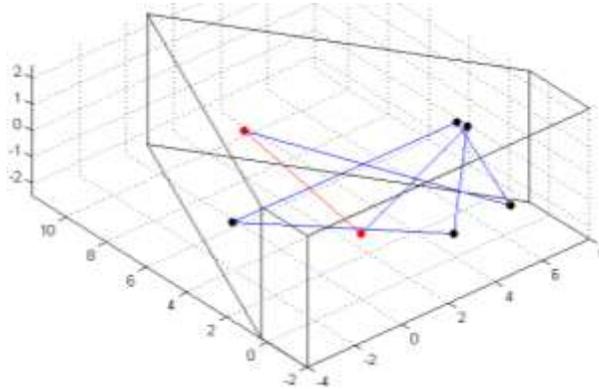


Figure 21: Salle 3 : 135 ms = plus long délai pour le cas de 5 réflexions. Chemin : 4 2 5 4 2

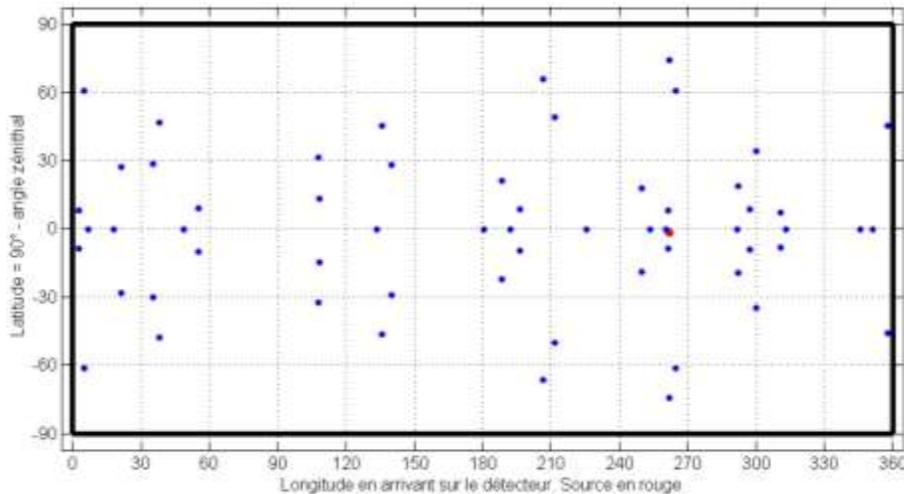


Figure 22: Salle 3, 5 rebonds, 59 chemins

## 8. Conclusion

La méthode présentée s'applique aux problèmes comportant  $n$  réflexions, mais le calcul en devient rapidement prohibitif. Pour  $n$  réflexions et  $k$  faces, il faut évaluer  $k(k-1)^{(n-1)}$  situations.

On peut cependant imaginer de trouver des chemins candidats par lancer de rayons et de les valider comme chemins construits sur des images par la méthode proposée dans la procédure.

## 9. Lectures utiles

[Allen 1979] Jont B. Allen and David A. Berkley, Image method for efficiently simulating small-room acoustics, *J. Acoust. Soc. Am.*, 65(4), Apr. 1979

[Beckers 1992] Benoit Beckers, Acoustique des salles Etude numérique et expérimentale de l'aspect temporel, Mémoire de fin d'études, Université de Liège, 1992, [[heliodon](#), [lien](#)]

[Beckers 2003] Beckers B., Masset L., Diseño de la acustica de salas con la teoria de las imagenes en 2D, Guia del usuario del programa Radit2D, 2003, [[heliodon](#), [enlace](#)]

[Beckers 2009] Beckers B., Borgia N., The Acoustic Model of the Greek Theatre, Protection of Historical Buildings - PROHITECH 2009, June 21-24 2009, Roma, Italy, [[heliodon](#), [lien](#), [link](#)]

[Beckers 2014] Beckers B., Beckers P., Reconciliation of Geometry and Perception in Radiation Physics, FOCUS Series in Numerical Methods in Engineering, Wiley-ISTE, 192 pages, July 2014

[Borish 1984] Borish J., “Extension of the image model to arbitrary polyhedra”, *J. Acoust. Soc. Am.*, Vol 75, No.6, June 1984

[Kandelman 2007] Mikaël Kandelman, Simulation acoustique par la méthode des sources images, Mémoire de fin d’études, Ecole nationale supérieure Louis-Lumière, 2007

[Lee 1988] Heewon Lee, Byung-Ho Lee, An Efficient Algorithm for the Image Model Technique, *Applied Acoustics* 24 (1988) 87-115

[McGovern 2009] Stephen G. McGovern, Fast image method for impulse response calculations of box-shaped rooms, *Applied Acoustics* 70 (2009) 182–189

## 10. Annexe I : La procédure de base

Les procédures de création de rayons joignant une source à un récepteur au travers d’éventuelles réflexions sur les parois de la salle ou de l’enceinte sont basées sur la méthode des images. Cette méthode est une de plus anciennes car elle n’exige pas de très grandes ressources informatiques, mais aujourd’hui elle est fortement concurrencée par la méthode des rayons. Dans ce document les algorithmes sont présentés sous forme de fonctions qui peuvent ainsi être utilisées de manière très souple.

Procédure <i>Propri.m</i> : les instructions qu’elle contient peuvent être introduites séparément dans la fenêtre d’exécution de Matlab ou ensemble via la procédure	
1	<code>tini = tic;h =3;b = 5;epsil=1.e-3; % hall &amp; bounces numbers, tolerance</code>
2	<code>[S,R,W,images,seq,X,Y,Z,vat] = Rechem(h,b,epsil);</code>
3	<code>disp(['===== Enclosure : ',num2str(h),' ====='])</code>
4	<code>suite = seq(1:vat,:);imf = images(1:vat,:);del = zeros(1,size(imf,1));</code>
5	<code>Cx=X(1:vat,:);Cy= Y(1:vat,:);Cz= Z(1:vat,:);</code>
6	<code>for i=1:vat;del(i)=round(norm(imf(i,:)-R)/340*1000);end;delais=sort(del);</code>
7	<code>disp(['Distance &amp; delay (ms) S - R: ',num2str(round(norm(S-R))),...</code>
8	<code>',' ,num2str(round(norm(S-R)/340*1000))])</code>
9	<code>disp(['Délais minimum, maximum(ms): ',num2str([min(delais) max(delais)])])</code>
10	<code>Signature (S,R,imf,vat)% Representation of directions incoming on receiver</code>
11	<code>disp(['CPU.....: ',num2str(toc(tini))])</code>
12	<code>title(['Hall n° ',num2str(h),' , number of images: ',num2str(size(imf,1))])</code>
13	<code>disp(['Délais en ms.....: ',num2str(delais)]) % Print the delays</code>
14	<code>figure;Desal(W);grid on;axis equal;hold on % Drawing detected valid rays</code>
15	<code>plot3(S(1),S(2),S(3),'.r','MarkerSize',20);hold on</code>
16	<code>plot3(R(1),R(2),R(3),'.r','MarkerSize',20);hold on</code>
17	<code>plot3([S(1) R(1)],[S(2) R(2)],[S(3) R(3)],'r','LineWidth',2);hold on</code>
18	<code>Rayon(S,R,Cx,Cy,Cz) % End draw "vat"(= number lines matrix Cx) valid rays</code>
19	<code>disp(['Hall volume.....: ',num2str(Volpri(h))])</code>
20	<code>axis([-10 10 -12 12 -2 2]); axis equal</code>
21	<code>figure;Desal(W);grid on;axis equal;hold on;la=4;</code>
22	<code>plot3([0 la],[0 0],[0 0],'.r','LineWidth',2)% Drawing the 3 axes x-r,y-g,z-b</code>
23	<code>plot3([0 0],[0 la],[0 0],'.g','LineWidth',2)</code>
24	<code>plot3([0 0],[0 0],[0 la],'.b','LineWidth',2)</code>

Table 5: Instructions qui peuvent être entrées directement dans Matlab

La première fonction *Rechem.m* appelée à la ligne 1 de la Table 5 est celle qui permet de calculer les rayons produisant  $n$  réflexions dans une salle répertoriée dans la fonction *Salle.m* appelée par cette fonction. Cette salle est définie par le premier argument de *Rechem.m* Le résultat du calcul est mémorisé dans le tableau *seq* transféré à la ligne 2 dans la matrice *suite* qui a les dimensions exactes du nombre de réflexions détectées, c'est-à-dire :  $vat \times nr$  (où  $nr$  est le deuxième argument de la fonction *Rechem.m*). Les résultats de la fonction se situent au premier membre entre crochets de la ligne 1.  $S$ ,  $R$ ,  $W$  correspondent aux 3 coordonnées de la source, du récepteur et des pieds des normales abaissées sur les parois ; la matrice *images* contient les coordonnées des dernières images des rayons, *seq* : la séquence des faces où ont lieu les rebonds,  $X$ ,  $Y$  et  $Z$  : respectivement les 3 coordonnées des points où ils ont lieu et *vat* le nombre de

rayons validés. La représentation de la salle ne nécessite que la matrice  $W$  et est réalisée à la [ligne 21](#) de la [Table 5](#). A la [ligne 18](#), la fonction [Rayon.m](#) permet de superposer au dessin les trajets de rayons détectés. Les données sont  $S$ ,  $R$ ,  $Cx$ ,  $Cy$  et  $Cz$  correspondant aux coordonnées  $X$ ,  $Y$  et  $Z$  dont les dimensions ont été validées à la [ligne 5](#) et le tableau [suite](#) validant la matrice [seq](#) à la [ligne 4](#) de la [Table 5](#).

Le principe de base de la méthode des images consiste à examiner toutes les possibilités de séquences de faces sur lesquelles s'effectuent les rebonds (ou réflexions) et d'en éliminer celles qui sont invalides. Cette opération est entièrement réalisée dans la fonction [Rechem.m](#) qui fait appel à la fonction [Salle.m](#) pour la définition de la géométrie.

Le troisième argument [epsil](#) utilisé à l'appel de la fonction [Rechem.m](#) correspond à un paramètre de précision ou de tolérance. Quand la procédure génère des chemins sortant de l'enceinte, il faut en général diminuer la valeur de ce paramètre. Sa valeur par défaut est de  $10^{-3}$ .

<b>Fonction <a href="#">Rechem.m</a> (recherche des chemins)</b>	
Cette procédure réalise la détection de la séquence des faces correspondant aux chemins comportant $nr$ réflexions. L'affichage du tableau <a href="#">suite</a> ( <a href="#">ligne 4</a> de la <a href="#">Table 5</a> ) donne le résultat de la <a href="#">Table 4</a> .	
1	<code>function[S,R,W,images,seq,X,Y,Z,vat]=Rechem(Sa,nr,epsil)%hall&amp;refl numb,tol</code>
2	<code>[S,R,W]= Salle(Sa); % Introducing the geometry</code>
3	<code>nf = size(W,1); % Number of walls</code>
4	<code>disp(['Position of the source S...: ',num2str(S)])</code>
5	<code>disp(['Position of the receptor R.: ',num2str(R)])</code>
6	<code>disp(['Number of walls .....: ',num2str(nf)])</code>
7	<code>disp(['Number of reflections .....: ',num2str(nr)])</code>
8	<code>np = zeros(nf,3);dipa=zeros(1,nf);diRe=zeros(1,nf);% General parameters</code>
9	<code>for i = 1:nf;</code>
10	<code>    dipa(i) = norm(W(i,:)); % Distances origin - walls</code>
11	<code>    np(i,:) = W(i,:)/dipa(i); % Walls normals</code>
12	<code>    diRe(i) = (W(i,:)-R)*np(i,:); % Distances receptor - walls</code>
13	<code>end</code>
14	<code>seq = zeros(nf^nr,nr);images=zeros(250,3); % Max possible number of paths</code>
15	<code>X = zeros(250,nr);Y = zeros(250,nr);Z = zeros(250,nr);</code>
16	<code>for ir = 1 : nr % Definition of the full set of possibilities</code>
17	<code>    k = 0.;n = 0.;</code>
18	<code>    for i = 1:nf^ir</code>
19	<code>        n = n + 1;if n &gt; nf ;n=1 ;end</code>
20	<code>        for j = 1:nf^(nr-ir);k = k+1; seq(k,ir) = n; end</code>
21	<code>    end</code>
22	<code>end</code>
23	<code>nche = 0;% Elimination of paths with same consecutive wall numbering</code>
24	<code>for i = 1:nf^nr;</code>
25	<code>    val = 1;</code>
26	<code>    for j = 1:nr-1;</code>
27	<code>        if seq(i,j+1) == seq(i,j);</code>
28	<code>            val =0;</code>
29	<code>    end;</code>
30	<code>end</code>
31	<code>if val == 1;nche = nche+1;seq(nche,:) = seq(i,);end</code>
32	<code>end</code>
33	<code>vat = 0; % *****</code>
34	<code>for ich = 1: nche % Validation of the "nche" paths by the image method</code>
35	<code>    C = zeros(nr,3);dico=zeros(1,nf);</code>
36	<code>    valid = 1;</code>
37	<code>    im = zeros(nr,3);% Images of analyzed path, initializ. &amp; source image</code>
38	<code>    im(1,:) = S+(W(seq(ich,1),:)-S)*np(seq(ich,1),:)*2*np(seq(ich,1),:);</code>
39	<code>    for i = 2 : nr; % Images of order &gt; 1</code>
40	<code>        im(i,:)=im(i-1,)+(W(seq(ich,i),:)-im(i-1,))*np(seq(ich,i),:)'...</code>
41	<code>        *2*np(seq(ich,i),:);</code>
42	<code>    end % End of building the (nr-1) remaining images</code>
43	
44	<code>t=diRe(seq(ich,nr))/((im(nr,:)-R)*np(seq(ich,nr),:));%Int R-I(nr) with Wi</code>

```

45 if t < 0;valid = 0; % 1st validity test: wall crossed with t > 0
46 else
47     tt = zeros(1,nf); % Starting the second validity test
48     for k = 1:nf;
49         tt(k) = diRe(k)/((im(nr,:)-R)*np(k,:)'+epsil);
50         if tt(k) < 0.;tt(k) = 100;end % Eliminate negative values
51     end
52     if min(tt) == 100; valid=0;end % Eliminating impossible solution
53     if min(tt) + epsil < t;valid=0;end % End of the second test
54 end % End of the final segment and the last bounce wall
55 C(nr,:) = (1-t)*R + t*im(nr,:); % Last bounce contact point
56 for j = nr-1 : -1 : 1; % Going back along the ray
57     for i=1:nf;
58         dico(i) = (W(i,:) - C(j+1,:))*np(i,:)'; % Distance contact - walls
59     end
60     t=dico(seq(ich,j))/((im(j ,:)-C(j+1,:))*np(seq(ich,j),:))'; % Int. C I
61     if t < 0;valid=0; % 1st validity test: wall crossed with t > 0
62     else
63         tt = zeros(1,nf); % Starting the second validity test
64         for k = 1:nf;
65             tt(k) = dico(k)/((im(j,:)-C(j+1,:))*np(k,:)'+epsil);
66             if tt(k)-epsil < 0.;tt(k) = 100;end % Eliminate neg. values
67         end
68         if min(tt) == 100; valid=0;end % Eliminating impossible solution
69         if min(tt)+epsil < t;valid=0;end % End of second test
70     end
71     if t > 1;valid=0;end % Third validity test *****
72     C(j ,:) = (1-t)*C(j+1,:) + t*im(j,:); % Next contact point
73 end
74 if valid > 0
75     vat = vat+1;seq(vat,:)=seq(ich,:); % Inserting the new validated path
76     images(vat,:) = im(nr,:);
77     X(vat,:) = C(:,1);Y(vat,:) = C(:,2);Z(vat,:) = C(:,3);
78 end
79 end
80 disp(['Number of detected paths ... ', num2str(vat)])
81 end

```

Table 6: Fonction Rechem.m pour la détermination des chemins d'ordre n

### Fonction Salle

Dans cette fonction sont répertoriées toutes les enceintes testées dans cette étude. On y définit la forme de la salle ainsi que les positions de la source et du récepteur. Les résultats attendus y sont mis en commentaire. Il en est de même pour la tolérance suggérée (ligne 1 de la procédure Proprin.m).

```

1 function [S,R,W] = Salle(Sa)
2 if Sa == 1 % Hall 1, prop epsil: 0.001, valid paths: 6+18+37+65+100 = 226
3     xi = -2;xs=4;yi=-2;ys=10;zi=-2.5;zs=2.5;
4     W = [xi -1 0;0 yi 0;xs -1 0;0 ys 0;0 0 zi;0 0 zs];
5     S = [0 0 0]; % Position of the source
6     R = [1 7 0.2]; % Position of the Receptor
7 end
8 if Sa == 2 % Hall 2, prop epsil: 0.001,valid paths: 5+13+27+48+77 = 170
9     W = [-4 0 0;0 -2 0;6 4 0;-3.5 1.5 0;0 0 -2.5;0 0 2.5];
10    S = [0 0 0];
11    R = [1 7 0.2];
12 end
13 if Sa == 3 % Hall 3, prop epsil: 0.001,valid paths: 5+13+25+39+59 = 141
14    W = [-4 0 0;0 -2 0;8 0 0;6 4 0;-3.5 1.5 0;0 0 -2.5;0 0 2.5];
15    S = [0 0 0];
16    R = [1 7 0.2];

```

```

17 end
18 if Sa == 4 % Hall 4, prop epsilon: 0.001, valid paths: 10+31+56+85+128 = 310
19     W         = [-4 0 0; -3 -1 0; 0 -2 0; 4 -2 0; 6 0 0; 5 8 0; 0 10 0; -3 9 0;
20                 0 0 -2.5; 0 0 2.5];
21     S         = [0 0 0];
22     R         = [1 7 0.2];
23 end
24 if Sa == 5 % Hall 5, prop epsilon: 1e-17, valid: 8+20+33+39+48 = 148
25     Nso       = 6;
26     W         = zeros(Nso+2,3); incr=2*pi/Nso;
27     for i     = 1:Nso;
28         W(i,:) = 5*[cos(-pi/2+(i-1)*incr) sin(-pi/2+(i-1)*incr) 0];
29     end
30     W(Nso+1,:) = [0 0 -2.5];
31     W(Nso+2,:) = [0 0 2.5];
32     S         = [0 -4.5 0];
33     R         = [0 4.5 0];
34 end
35 if Sa == 6 % Hall 6, prop epsilon: 0.001, valid paths: 6+22+42+78+106 = 254
36     W         = [0 -1 0; 1 0 0; 0 1 0; -1 0 0; 0 0 -1; 0 0 1];
37     S         = [0 -0.8 0];
38     R         = [0 0.8 0];
39 end
40 if Sa == 7 % Hall 7, prop epsilon: 1.e-16, valid paths: 6+16+40+68+98 = 228
41     W         = [0 -1.2 0; 1.0 0.8 0; 0 1.8 0; -1.0 0.8 0; 0 0 -1.; 0 0.1.];
42     S         = [0 0 0];
43     R         = [0 1.6 0];
44 end
45 if Sa == 8 % Hall 8, prop epsilon: 0.001, val. paths: 10+34+68+112+170 = 394
46     Nso       = 8;
47     W         = zeros(Nso+2,3); incr=2*pi/Nso;
48     for i     = 1:Nso;
49         W(i,:) = 5*[cos(-pi/2+(i-1)*incr) sin(-pi/2+(i-1)*incr) 0];
50     end
51     W(Nso+1,:) = [0 0 -2.5];
52     W(Nso+2,:) = [0 0 2.5];
53     S         = [0 -4.5 0];
54     R         = [0 4.5 0];
55 end
56 if Sa == 9 % Hall 9, prop epsilon: 0.001, valid paths: 6+22+46+92+150 = 316
57     W         = [-5 0 0; 0 -2 0; 5 0 0; 0 18 0; 0 0 -2; 0 0 3];
58     S         = [0 0 .5];
59     R         = [0 16 .5];
60 end
61 if Sa ==10 % Hall 10, prop epsilon: 0.001, valid paths: 6+18+34+56+88 = 202
62     W         = [-1 0 0; -1.5 -5 0; 20 0 0; -1.5 5 0; 0 0 -2; 0 0 2];
63     S         = [2 0 0];
64     R         = [18 0 0];
65 end
66 if Sa ==11 % Hall 11, prop epsilon: 0.001, valid paths: 6+18+34+56+88 = 202
67     W         = [0 -1 0; -5 -1.5 0; 0 20 0; 5 -1.5 0; 0 0 -2; 0 0 2];
68     S         = [0 2 0];
69     R         = [0 18 0];
70 end
71 if Sa ==12 % Hall 12, prop epsilon: 0.0001 valid paths: 6+18+34+56+86 = 200
72     W         = [0 -1 0; 5 -1.5 0; 0 20 0; -5 -1.5 0; 0 0 -2; 0 0 2];
73     S         = [0 0 0];
74     R         = [0 18 0];
75 end
76 if Sa ==13 % Hall 13, prop epsilon: 0.001 valid paths: 6+18+34+56+86 = 200
77     W         = [0 -1.05 0; 5 -1.5 0; 0 20 0; -5 -1.5 0; 0 0 -2; 0 0 2];
78     S         = [0 0 0];
79     R         = [0 18 0];

```

80	end
81	end

Table 7: Fonction Salle pour la définition des enceintes utilisées dans le document

## 11. Annexe II : Visualisation des résultats

On peut envisager d'autres opérations que le calcul des chemins. La première opération est la représentation de la salle étudiée, elle est réalisée dans la fonction *Desal.m*. Cette fonction permet de visualiser les salles enregistrées dans la fonction *Salle.m* appelée à la [ligne 13](#) de la procédure *Proprin.m*. L'argument *W* de la fonction a été calculé à la [ligne 2](#).

Cette fonction, très compacte, ne nécessite que la définition de la géométrie des faces de la salle étudiée. On peut facilement y inclure la représentation de la source et du récepteur qui sont fournis par la fonction *Salle.m* ainsi que le tracé du rayon direct ([lignes 14 à 16](#)) de la procédure *Proprin.m*.

Pour réaliser la représentation de la salle, il faut passer d'une définition du polygone de base par les pieds des normales aux cotés à une définition basée sur ses sommets. Pour les obtenir, on calcule les points d'intersections de 2 cotés consécutifs.

En désignant les pieds des perpendiculaires par  $(x_i, y_i)$  pour le côté  $i$ ,  $(x_{i+1}, y_{i+1})$  pour le côté  $i+1$ , etc... et les tangentes :  $(s_i, t_i)$  pour celle qui coïncide avec le côté  $i$ ,  $(s_{i+1}, t_{i+1})$ , celle qui coïncide avec le côté  $i+1$ , etc... on exprime tout d'abord l'intersection de deux cotés successifs  $i$  et  $i+1$  comme un point appartenant aux deux cotés.

$$\begin{aligned} x_i + us_i &= x_{i+1} + vs_{i+1} \\ y_i + ut_i &= y_{i+1} + vt_{i+1} \end{aligned} \quad (1.12)$$

Les membres de droite correspondent à la [ligne 12](#) de la fonction *Desal.m*. C'est un système de 2 équations à 2 inconnues  $u$  et  $v$ . En détaillant les étapes, on calcule  $v$ , à partir de la seconde ligne :

$$v = \frac{(y_i - y_{i+1}) + ut_i}{t_{i+1}} \quad (1.13)$$

En remplaçant dans la première relation de (1.12), on obtient le paramètre  $u$  :

$$\begin{aligned} x_i + us_i &= x_{i+1} + \frac{s_{i+1}}{t_{i+1}} [(y_i - y_{i+1}) + ut_i] \\ x_i t_{i+1} + us_i t_{i+1} &= x_{i+1} t_{i+1} + s_{i+1} [(y_i - y_{i+1}) + ut_i] \\ u &= \frac{t_{i+1}(x_i - x_{i+1}) + s_{i+1}(y_i - y_{i+1})}{s_i t_{i+1} - s_{i+1} t_i} \end{aligned} \quad (1.14)$$

Ensuite, on calcule explicitement les composantes des tangentes (produits vectoriels des normales par l'axe vertical  $z$ ) :

$$[s_i \ t_i] = \frac{[-y_i \ x_i]}{\sqrt{x_i^2 + y_i^2}} ; \quad [s_{i+1} \ t_{i+1}] = \frac{[-y_{i+1} \ x_{i+1}]}{\sqrt{x_{i+1}^2 + y_{i+1}^2}} \quad (1.15)$$

En remplaçant dans (1.14), on obtient :

$$\begin{aligned}\frac{u}{\sqrt{x_i^2 + y_i^2}} &= \frac{[x_{i1}(x_{i1} - x_i) - y_{i1}(y_i - y_{i1})]}{x_{i1}y_i - x_iy_{i1}} \\ &= \frac{[x_{i1}^2 + y_{i1}^2 - x_ix_{i1} - y_iy_{i1}]}{x_{i1}y_i - x_iy_{i1}}\end{aligned}\quad (1.16)$$

On obtient ensuite le sommet  $(i - i+1)$  en reportant le paramètre  $u$  de (1.16) dans (1.12) :

$$\begin{aligned}x_i + us_i &= x_i - \left( \frac{u}{\sqrt{x_i^2 + y_i^2}} \right) y_i \\ y_i + ut_i &= y_i + \left( \frac{u}{\sqrt{x_i^2 + y_i^2}} \right) x_i\end{aligned}\quad (1.17)$$

A partir de (1.13) et de (1.15), on a également la possibilité d'exprimer le paramètre  $v$  en fonction des coordonnées et de  $u$ :

$$v = \frac{(y_i - y_{i1}) - \frac{ux_i}{\sqrt{x_i^2 + y_i^2}}}{\frac{-x_{i1}}{\sqrt{x_{i1}^2 + y_{i1}^2}}}\quad (1.18)$$

$$v = \frac{(y_i - y_{i1})(x_{i1}y_i - x_iy_{i1}) - x_i[x_{i1}(x_{i1} - x_i) - y_{i1}(y_i - y_{i1})]}{\frac{-x_{i1}(x_{i1}y_i - x_iy_{i1})}{\sqrt{x_{i1}^2 + y_{i1}^2}}}\quad (1.19)$$

On trouve la solution de la *ligne 11* de la fonction *Desal.m* :

$$v^* = \frac{v}{\sqrt{x_{i1}^2 + y_{i1}^2}} = \frac{y_i^2 + x_i^2 - y_iy_{i1} - x_ix_{i1}}{(x_{i1}y_i - x_iy_{i1})}\quad (1.20)$$

A partir des seconds membres des équations de on obtient finalement l'expression de la *ligne 12* de la fonction *Desal.m* :

$$\begin{aligned}x_{i1} + vs_{i1} &= x_{i1} - \left( \frac{v}{\sqrt{x_{i1}^2 + y_{i1}^2}} \right) y_{i1} = x_{i1} - v^* y_{i1} \\ y_{i1} + vt_{i1} &= y_{i1} + \left( \frac{v}{\sqrt{x_{i1}^2 + y_{i1}^2}} \right) x_{i1} = y_{i1} + v^* x_{i1}\end{aligned}\quad (1.21)$$

Dans la fonction *Desal.m*, la variable  $v$  correspond à la variable  $v^*$  des deux relations précédentes.

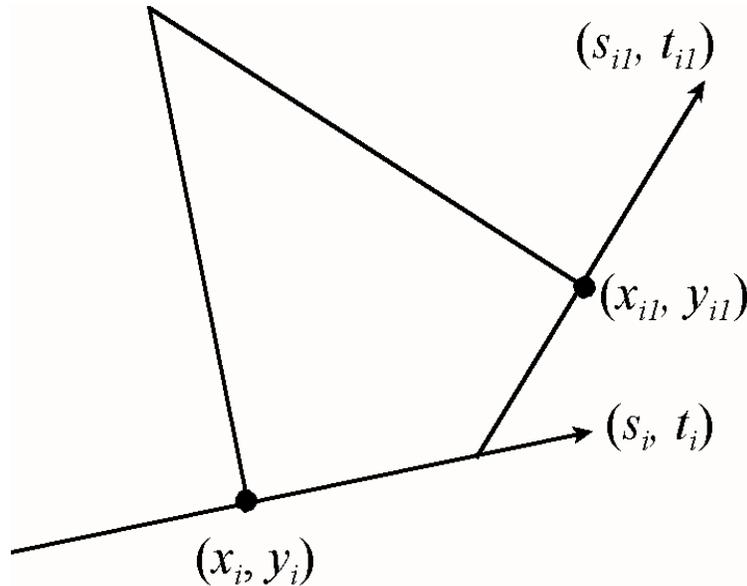


Figure 23: Schéma du polygone de base de la salle étudiée

<b>Fonction <i>Desal.m</i> (dessin d'une salle)</b>	
Les salles sont données par des matrices de points $W$ définies dans la procédure <i>Salle.m</i>	
1	<code>function [] = Desal(W) % Drawing a prismatic enclosure</code>
2	<code>nf = size(W,1); % Any number of sides for the bases polygons</code>
3	<code>zi = W(nf-1,3);zs = W(nf,3);% Heights of horizontal ground &amp; roof</code>
4	<code>normales = zeros(nf,3);tangentes = zeros(nf-2,3);sop = zeros(nf-1,3);</code>
5	<code>for i=1:nf ;normales(i,:) = W(i,:)/norm(W(i,:)) ;end</code>
6	<code>for i=1:nf-2;tangentes(i,:) = cross([0 0 1],normales(i,:));end</code>
7	<code>for i=1:nf-2;</code>
8	<code>xi = W(i,1);yi = W(i,2);</code>
9	<code>if i &lt; nf-2;i1 = i+1;else i1 = 1; end</code>
10	<code>xil = W(i1,1); yil=W(i1,2);</code>
11	<code>v = (xi^2+yi^2-xi*xil-yi*yil)/(yi*xil-xi*yil);</code>
12	<code>sop(i,1)= xil-v*yil; sop(i,2) = yil+v*xil;</code>
13	<code>end</code>
14	<code>sop(nf-1,:) = sop(1,:); % Drawing</code>
15	<code>plot3(sop(:,1)',sop(:,2)',sop(:,3)+zi','k','LineWidth',1);hold on;%2 bases</code>
16	<code>plot3(sop(:,1)',sop(:,2)',sop(:,3)+zs','k','LineWidth',1);hold on;</code>
17	<code>for i=1:nf-2 % Drawing the nf-2 vertical edges</code>
18	<code>plot3([sop(i,1) sop(i,1)],[sop(i,2) sop(i,2)],[zi zs],'k',...</code>
19	<code>'LineWidth',1);hold on;</code>
20	<code>end</code>
21	<code>end</code>

Table 8: Fonction *Desal* pour le dessin d'une enceinte prismatique

Les opérations suivantes consistent à représenter les rayons, soit en les dessinant entièrement soit en dessinant leur orientation lorsqu'ils arrivent sur le récepteur. Dans ce but on utilise soit la fonction *Rayon.m* où on récupère les points de contacts lors des rebonds, soit la fonction *Signature.m* pour laquelle on a récupéré la dernière image (celle qui conduit au récepteur) utilisée par chaque rayon.

<b>Fonction <i>Rayon.m</i> (tracé des rayons)</b>	
1	<code>function [] = Rayon(S,R,Cx,Cy,Cz)</code>
2	<code>vat = size(Cx,1);</code>
3	<code>for nray = 1:vat</code>
4	<code>plot3([S(1) Cx(nray,:) R(1)],[S(2) Cy(nray,:) R(2)],...</code>
5	<code>[S(3) Cz(nray,:) R(3)],'b','LineWidth',1) %Drawing the full ray</code>

```

6     plot3(Cx(nray,:),Cy(nray,:),Cz(nray,:),'.k','MarkerSize',10);
7 end
8 % axis([-10 20 -12 12 -2 2])
9 title(['Number of validated rays: ',num2str(vat)])
10 end

```

Table 9: Fonction Rayon pour le tracé des rayons

<b>Fonction <i>Signature.m</i></b> (visualisation des directions des rayons arrivant sur le récepteur)	
1	<code>function [] = Signature(S,R,imf,vat)</code>
2	<code>figure('Position',[100 100 1024 512]) % Rays directions in "plate carree"</code>
3	<code>co = 90-acos((S(3)-R(3))/norm(S-R))*180/pi ; % Direct ray</code>
4	<code>lo = atan2(S(2)-R(2),S(1)-R(1))*180/pi;</code>
5	<code>if lo &lt; 0;lo = lo + 360;end</code>
6	<code>plot(lo,co,'r','MarkerSize',20);hold on;</code>
7	<code>x = [0 360 360 0 0];y=[-90 -90 90 90 -90]; % Dessin du cadre</code>
8	<code>plot(x,y,'k','LineWidth',4);hold on; grid on</code>
9	<code>axis([-5 365 -91 91]);axis equal</code>
10	<code>for i = 1: vat% For each possible path issued from the walls combinations</code>
11	<code>co = 90-acos((imf(i,3)-R(3))/norm(imf(i,:)-R))*180/pi ;</code>
12	<code>lo = atan2(imf(i,2)-R(2),imf(i,1)-R(1))*180/pi;</code>
13	<code>if lo &lt; 0;lo = lo + 360;end</code>
14	<code>plot(lo,co,'.b','MarkerSize',15);hold on;grid on</code>
15	<code>end</code>
16	<code>set(gca,'fontsize',12);set(gca,'XTick',0:30:360);set(gca,'YTick',-90:30:90)</code>
17	<code>title(['Number of images ', num2str(vat)])</code>
18	<code>xlabel('Longitude en arrivant sur le détecteur. Source en rouge');</code>
19	<code>ylabel('Latitude = 90° - angle zénithal');</code>
20	<code>end</code>

Table 10: Fonction Signature : directions des rayons à leur arrivée sur le récepteur

<b>Fonction <i>Volpri.m</i></b> (calcul du volume de l'enceinte prismatique)	
1	<code>function [volume] = Volpri(En)</code>
2	<code>[~,~,P] = Salle(En); % 2 dernières faces = plancher et plafond</code>
3	<code>nf = size(P,1); % Cotés de la base : ordre horaire ou trigonométrique</code>
4	<code>n = zeros(nf,3);</code>
5	<code>nfl = nf-2;t = zeros(nfl,3);</code>
6	<code>for i = 1:nfl;n(i,:) = P(i,:)/norm(P(i,:));end</code>
7	<code>for i = 1:nfl;t(i,:) = cross([0 0 1],n(i,:));end</code>
8	<code>So = zeros(nfl,3);</code>
9	<code>for i = 1:nfl;</code>
10	<code>if i &lt; nfl;i1 = i+1;else i1 = 1;end</code>
11	<code>xi = P(i,1) ;yi = P(i,2);</code>
12	<code>xil = P(i1,1);yil = P(i1,2);</code>
13	<code>u=(t(i1,2)*(xil-xi)+t(i1,1)*(yi-yil))/(t(i,1)*t(i1,2)-t(i1,1)*t(i,2));</code>
14	<code>So(i,1) = xi+u*t(i,1);So(i,2) = yi+u*t(i,2);</code>
15	<code>end</code>
16	<code>tr = 0.;</code>
17	<code>for i = 1:nfl;</code>
18	<code>if i &lt; nfl;i1 = i+1;else i1 = 1;end</code>
19	<code>tr = tr + cross(So(i,:),So(i1,))*[0;0;1]/2;</code>
20	<code>end</code>
21	<code>volume = norm(tr*(P(nf,3)-P(nf-1,3)));</code>
22	<code>end</code>

Table 11: Fonction Volpri: volume de la salle

## 12. Liste des figures et des tables

Figure 1 : Loi de Snell-Descartes .....	2
Figure 2 : Méthode des images .....	2
Figure 3 : Illustration de la méthode des images en 3D .....	2
Figure 4 : Réflexion multiple dans une enceinte hexaédrique. ....	3
Figure 5: Schéma pour le calcul de l'image d'un point P .....	4
Figure 6: Définition de l'enceinte par les pieds des perpendiculaires abaissées de l'origine .....	4
Figure 7: Salle 1, 1 rebond, 6 chemins et rayon direct .....	7
Figure 8: Salle 2, 1 rebond, 5 chemins et rayon direct .....	7
Figure 9: Salle 1, 2 rebonds, 18 chemins incluant le coloriage des 2 rebonds.....	8
Figure 10: Le plus court des 100 chemins à 5 rebonds, séquence 6 2 3 5 6, délai 59 ms (Deche.m) .....	9
Figure 11: Un chemin quasi singulier parmi les 100 détectés, séquence 4 1 6 3 4, délai : 99 ms.....	9
Figure 12: Le plus long des 100 chemins à 5 rebonds, séquence : 4 2 4 2 4, délai 179 ms.....	9
Figure 13: Salle 1, 1 rebond, 6 chemins .....	12
Figure 14: Salle 1, 2 rebonds, 18 chemins .....	12
Figure 15: Salle 1, 3 rebonds, 37 chemins .....	12
Figure 16: Salle 1, 4 rebonds, 65 chemins .....	12
Figure 17: Salle 1, 5 rebonds, 100 chemins .....	12
Figure 18: Salle 1, 6 rebonds, 138 chemins .....	12
Figure 19: Salle 2 : 135 ms = plus long délai pour le cas de 5 rebonds sur 3 2 4 3 2 .....	13
Figure 20: Salle 2, 5 rebonds, 77 chemins détectés .....	13
Figure 21: Salle 3 : 135 ms = plus long délai pour le cas de 5 réflexions. Chemin : 4 2 5 4 2.....	14
Figure 22: Salle 3, 5 rebonds, 59 chemins .....	14
Figure 23: Schéma du polygone de base de la salle étudiée .....	21
Table 1: Définition de la première enceinte hexaédrique (fonction Salle.m).....	6
Table 2: Images de la source de la salle 1 par rapport aux 6 parois.....	6
Table 3: Contacts et délais pour 1 réflexion dans l'enceinte n° 1 de la Table 7 (fonction Salle.m) .....	7
Table 4: Les 100 chemins comportant 5 rebonds, procédure Rechem.m (Table 6) .....	11
Table 5: Instructions qui peuvent être entrées directement dans Matlab .....	15
Table 6: Fonction Rechem.m pour la détermination des chemins d'ordre n .....	17
Table 7: Fonction Salle pour la définition des enceintes utilisées dans le document .....	19
Table 8: Fonction Desal pour le dessin d'une enceinte prismatique .....	21
Table 9: Fonction Rayon pour le tracé des rayons .....	22
Table 10: Fonction Signature : directions des rayons à leur arrivée sur le récepteur .....	22
Table 11: Fonction Volpri: volume de la salle.....	22