



Acoustique : simulation en acoustique des salles

François DERRIDA, Yaël GOSSEC

Plan de la présentation :



I - Description du projet

II - Focus sur deux scripts

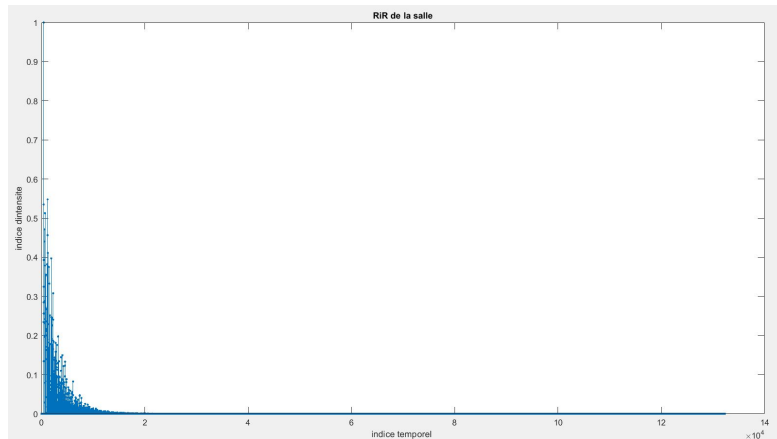
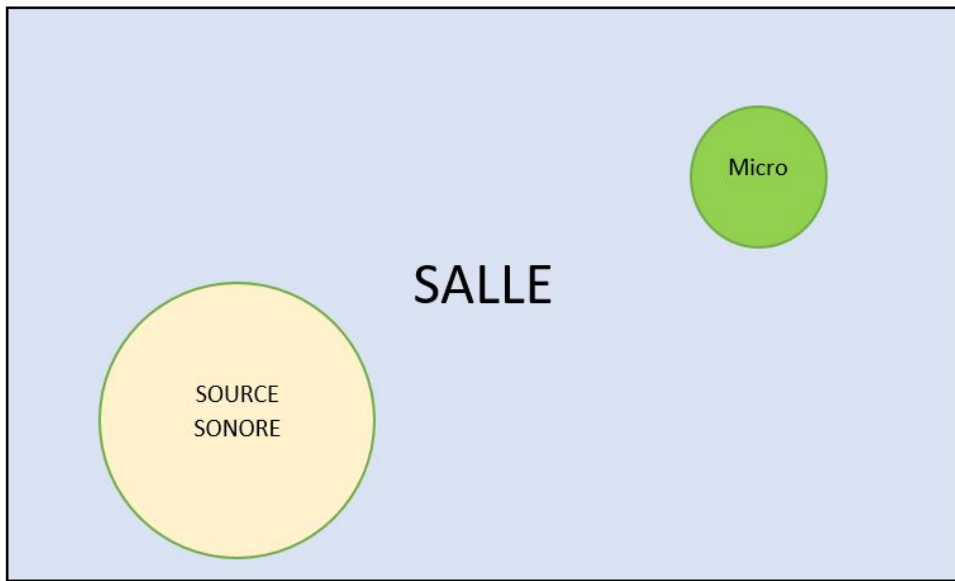
III - Conclusion



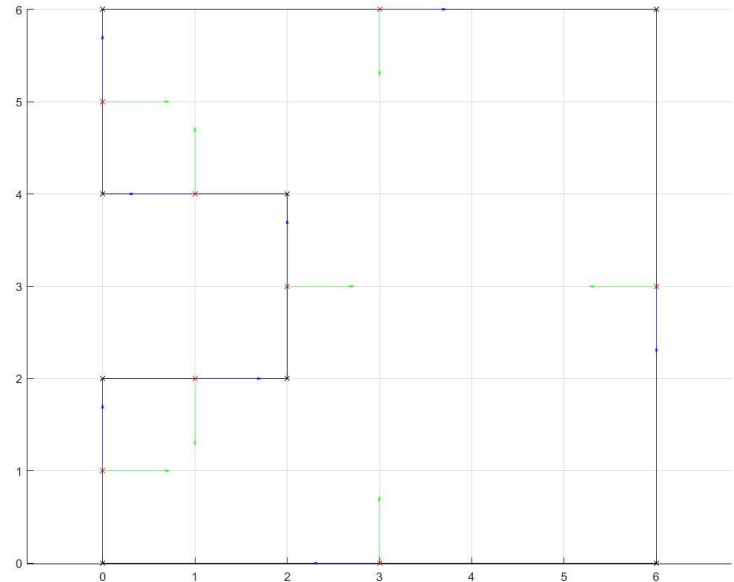
Description du projet

Description du projet:

Objectif : Calculer numériquement la réponse impulsionnelle d'une salle.



```
% Paramètres de la salle
vtx = [0,0;0,2;2 2; 2 4; 0 4; 0 6; 6 6; 6 0];
elt = [1,2;2,3;3,4;4,5;5,6;6,7;7,8;8,1];
```

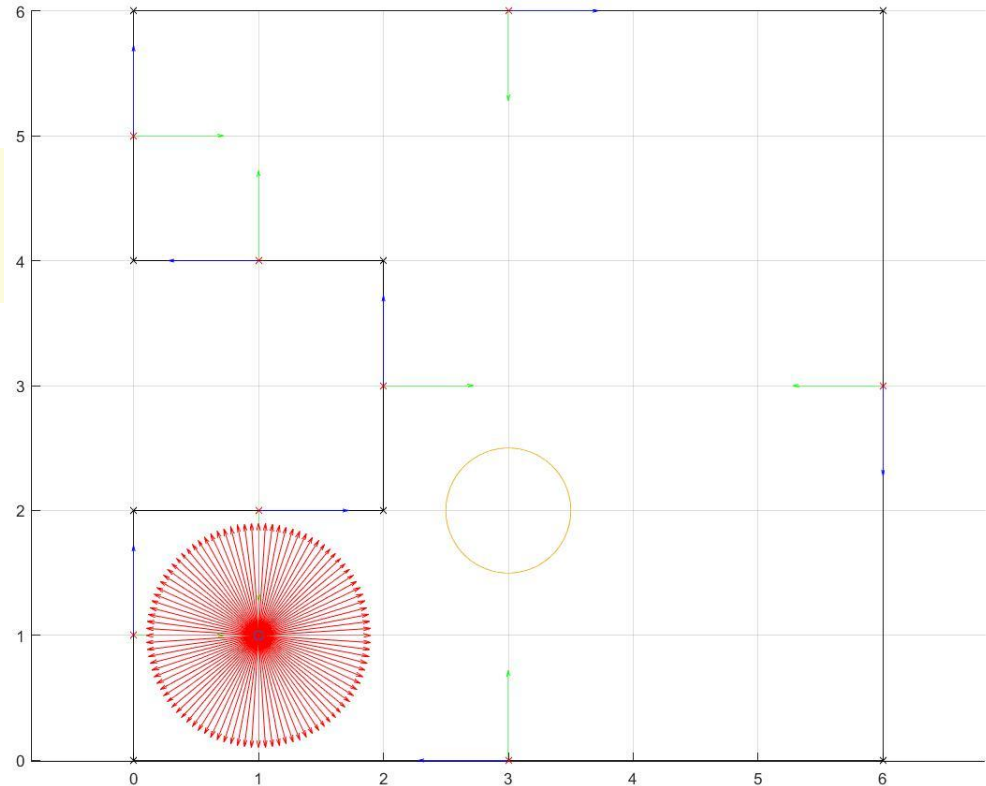


Description du projet:

-> créer la source sonore et le micro

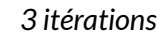
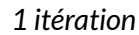
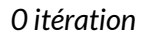
```
% Paramètres des rayons  
src = [1,1]; % Coordonnées de la source  
N_ray = 100; % Nombre de rayons
```

```
% Micro  
xm = 3; % Coordonnée X du micro  
ym = 2; % Coordonnée Y du micro  
rm = 0.5; % Rayon du micro
```



```
for i=1:n_iter

    new_ray = rebond(ray,mesh);
```



Description du projet:

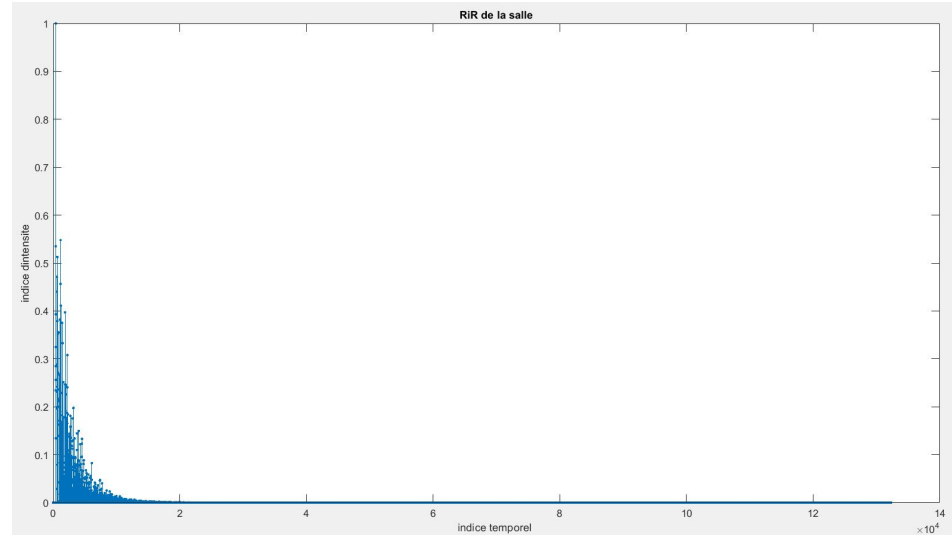
-> création de l'objet RiR (Room Impulsional Response)

```
function rir = initialisation_rir(fs,duree,celerite)

rir.celerite = celerite;
rir.ordonnee = zeros(duree*fs,1);
rir.taille = duree*fs;
rir.freq = fs;

end
```

Fréquence d'échantillonnage : $fs = 44\,100$ Hz



Description du projet:

-> création d'un script `mesure` qu'on appelle à chaque itération, qui mesure le nombre de rayons ayant intersectés le micro lors de la dernière itération et les range temporellement au bon endroit dans la rir.

```
for i=1:n_iter % Boucle qui met à jour le tableau rir

    new_ray = rebond(ray,mesh);
    rir = mesure(ray,new_ray,mic,rir);
    plot_lignes(ray,new_ray);
    ray = new_ray;

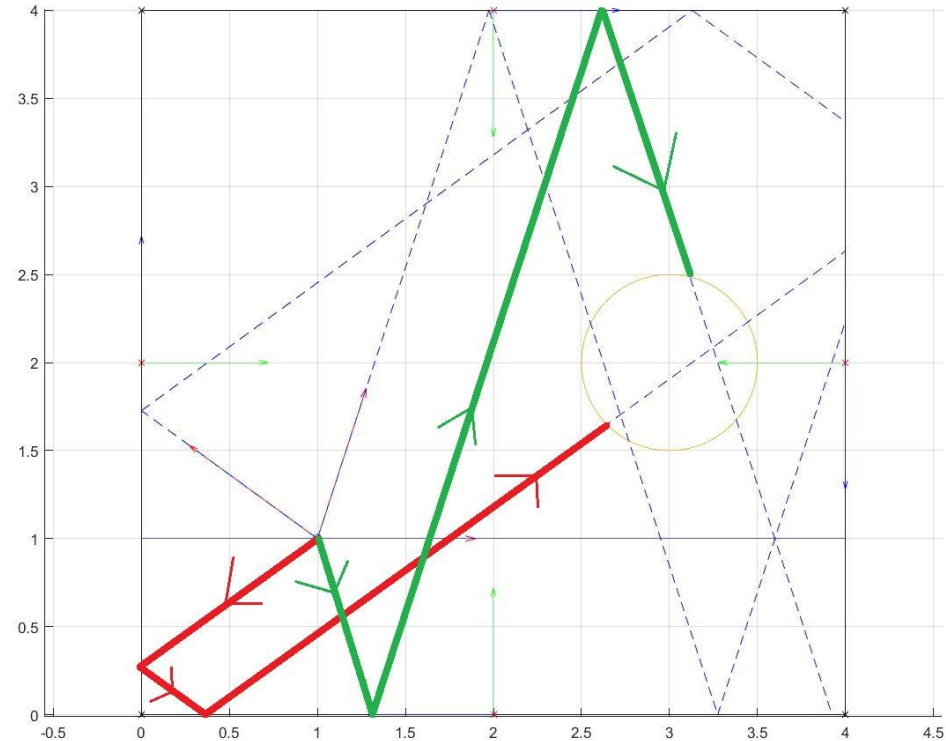
end
```

Description du projet: (Yaël)

“ranger temporellement au bon endroit dans la rir”.
→ arrivée à la même itération != arrivée au même instant.

résumé de la fonction mesure:

```
for rayon in rayons :  
    if (rayon croise micro):  
        distance = rayon.distance  
        indice_temporel = distance/vitesse * f  
        rir[indice_temporel] += 1
```





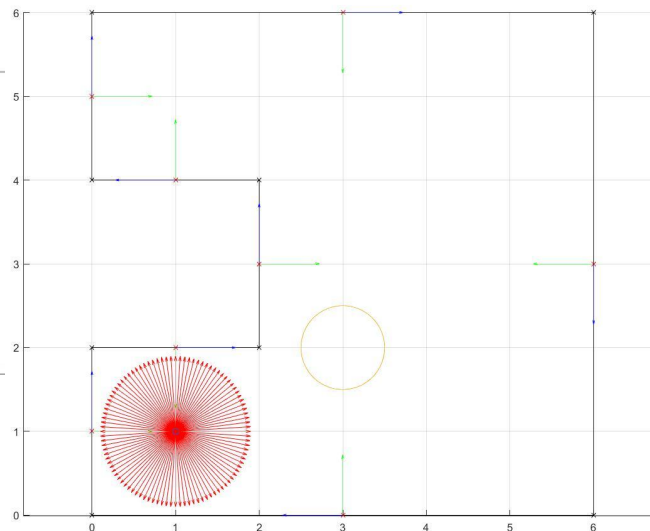
Focus sur le script global

```
1  %% Nettoyage de l'environnement matlab
2 - clear all
3 - close all
4
5  %% Users
6  % Paramètres de la salle
7 - vtx = [0,0;0,2;2 2; 2 4; 0 4; 0 6; 6 6; 6 0];
8 - elt = [1,2;2,3;3,4;4,5;5,6;6,7;7,8;8,1];
9
10 % Paramètres des rayons
11 - src = [1,1]; % Coordonnées de la source
12 - N_ray = 3; % Nombre de rayons
13 - ammortissement_mur = 1;
14 - ammortissement_micro = 1;
15
16 % Micro
17 - xm = 3; % Coordonnée X du micro
18 - ym = 2; % Coordonnée Y du micro
19 - rm = 0.5; % Rayon du micro
20
21 % Paramètres d'enregistrement
22 - fs = 44100; %freq echantillonnage
23 - duree = 4;
24 - n_iter = 1000;
25 - celerite = 340; %célérité du son
26
```

```

28  %% initialisation de la Géométrie
29  mesh = geometry(vtx,elt);
30  ray = rayons(N_ray,src,ammortissement_mur,ammortissement_micro);
31  mic = micro(xm,ym,rm);
32
33
34  %% Affichage
35  figure(1)
36  plotmaillage(mesh);
37  plotray(ray);
38  plotmicro(mic);
39

```



```
40 %% Création et mise à jour de la Réponse Impulsionnelle de la Salle
```

```
41  
42 rir = initialisation_rir(fs,duree,celerite); %initialise l'objet RiR
```

```
43  
44 for i=1:n_iter % Boucle qui met à jour le tableau rir
```

```
45  
46     new_ray = rebond(ray,mesh);
```

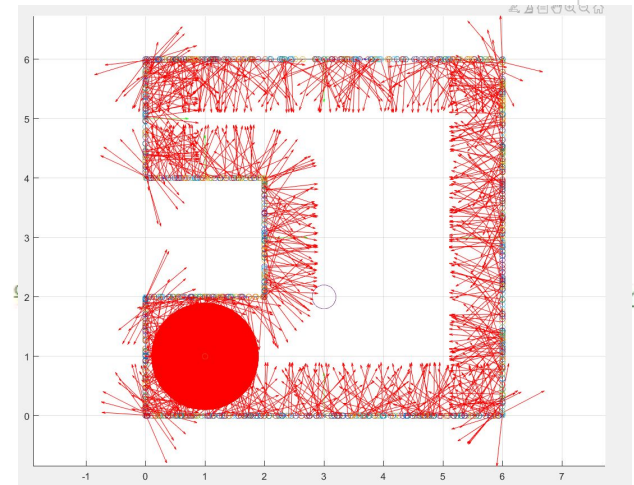
```
47     rir = mesure(ray,new_ray,mic,rir);
```

```
48     ray = new_ray;
```

```
49 end
```

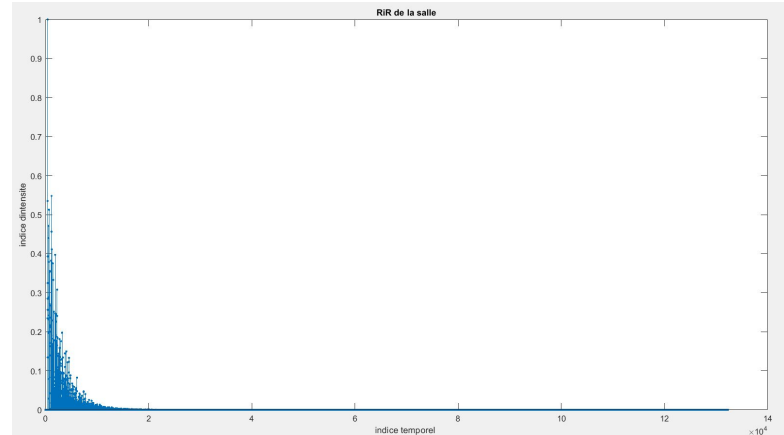
```
50 plotray(ray);
```

```
51  
52 rir.ordonnee = rir.ordonnee / max(rir.ordonnee);
```



```
%% Affichage de la Réponse Impulsionnelle de la Salle  
figure(2)
```

```
stem(rir.ordonnee, '.');
```



Description de la fonction rayon

```
function [ray] = rayons(N,source,ammortissement_mur,ammortissement_micro)

    %cette fonction initialise l'objet ray
    x_s = source(1,1);
    y_s = source(1,2);
    ray.pos = ones(N,2);
    ray.dir = ones(N,2);
    ray.dist = ones(N,1); %pour garder en mémoire la distance parcourue par le rayon
    ray.last_dist = ones(N,1) ; %pour garder en tête la dernière longueur
    ray.intensity = ones(N,1); %l'intensité vaut 1 au début puis elle sera modifiée au fur et a mesure
    ray.ammortissement_mur = ammortissement_mur;
    %valeur comprise entre 0 et 1 par laquelle l'intensité est multipliée lorsque le rayon rebondit sur un mur
    ray.ammortissement_micro = ammortissement_micro;
    ray.nb = N;

    ray.pos(:,1) = ray.pos(:,1)*x_s ;
    ray.pos(:,2) = ray.pos(:,2)*y_s;
    for k=1:N
        ray.dir(k,1) = cos(2*pi*k/N);
        ray.dir(k,2) = sin(2*pi*k/N);
    end
end
```




Merci pour votre attention

