

L'ACOUSTIQUE DES SALLES



**Comment prévoir et améliorer l'acoustique
des salles ?**



Sommaire

- Introduction
- I) Prévoir l'acoustique d'une salle et modélisation
 - II) Amélioration de l'acoustique d'une salle
- Conclusion

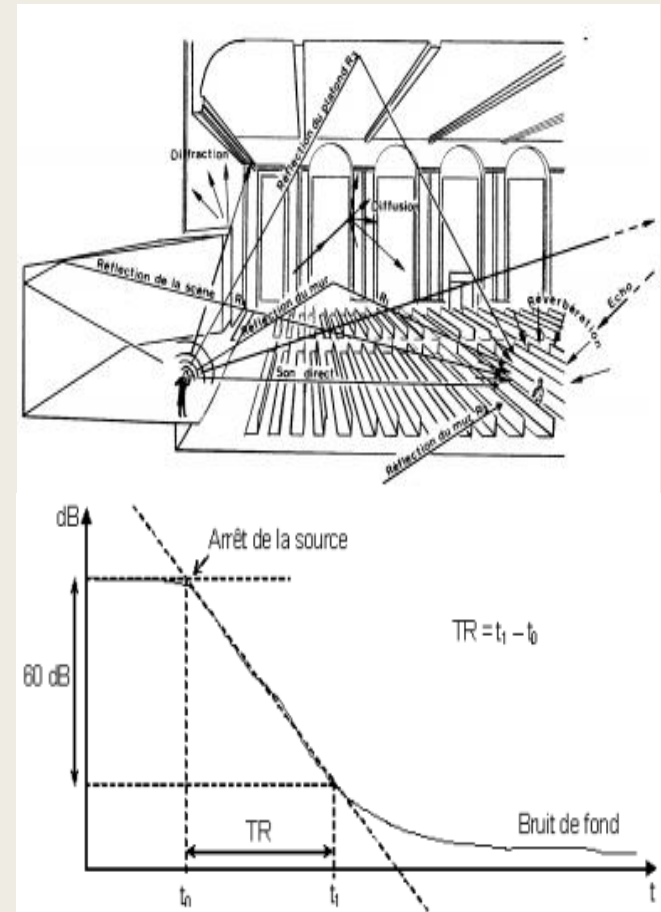
Introduction

■ Définition:

- Temps de réverbération
- Coefficient d'absorption (ou de réflexion)

■ But du TIPE:

- Mesurer le coefficient d'absorption d'une salle de classe et savoir l'améliorer



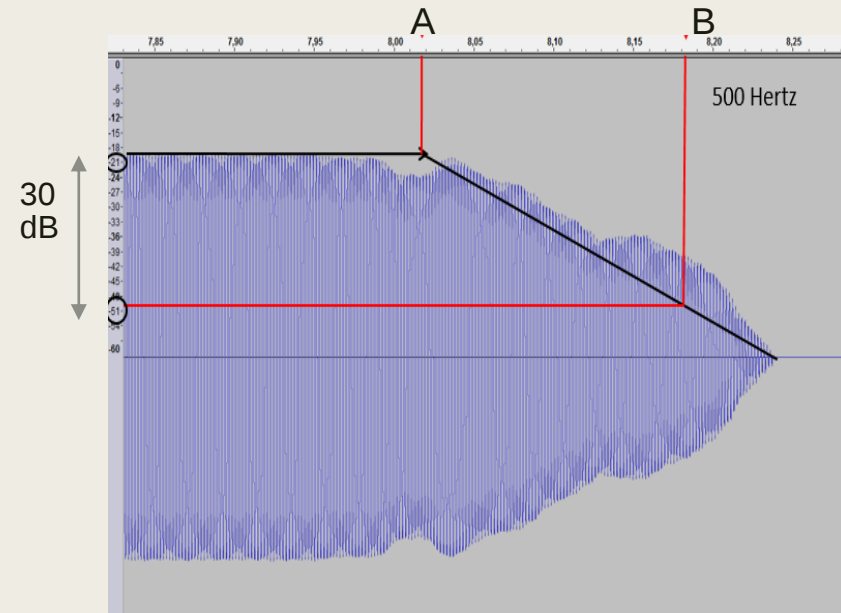
Source: Cours d'acoustique et mécanique
aléatoire
I.U.P GDP Option GET

I) Prévoir le temps de réverbération d'une salle de classe

■ Expérience 1:

Mesure du temps de réverbération par la méthode de la « chute de - 60 dB »

Salle étudiée →



■ Résultats:

Fréquence (Hz)	145	506	751	1000
Temps de réverbération (s)	0,62	0,38	0,17	0,09

Approche théorique

Loi de Sabine :

$$TR = 0,16 \times \frac{V}{A}$$

Avec : V : volume (m³)

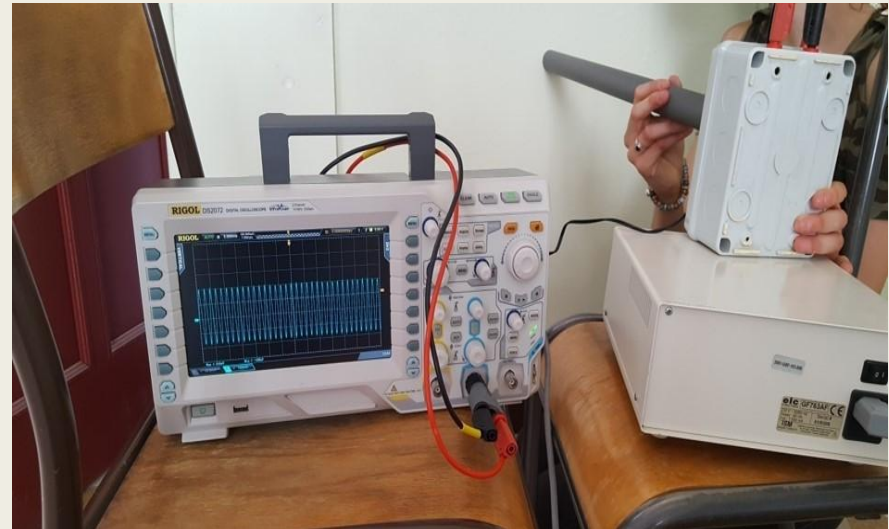
$$A = \sum \alpha_i^* S_i \text{ (m}^2\text{)}$$

α_i : coefficient d'absorption du matériau i

S_i : surface occupée par ce matériau (m²)

Calcul des coefficients d'absorption

- Expérience 2:



- Calcul des coefficients d'absorption des matériaux présents dans la pièce pour la fréquence de 500 Hz

- $\alpha = 1 - r = 1 - \left(\frac{U_{max} - U_{min}}{U_{max} + U_{min}} \right)^2$

Résultats :

Matériaux	Carrelage	Bois	Vitrage	Dalle de plafond	Mur
Coefficient d'absorption	0,5	0,7	0,1	0,7	0,4

- Temps de réverbération avec la formule de Sabine:

$$\underline{\text{TR} = 0,35 \text{ secondes}}$$

Modélisation

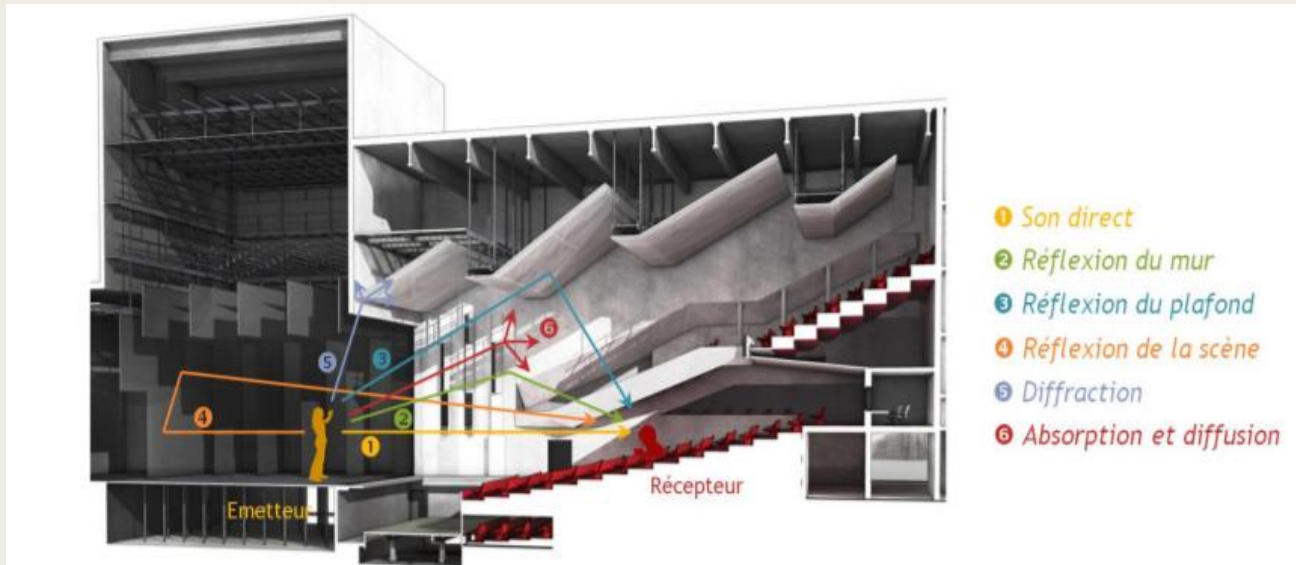


Figure 1 : Propagation d'une onde sonore dans un auditorium
D'après « Pour la science », dossier hors série n° 32, juil/oct 2001, Image de l'auditorium [1]

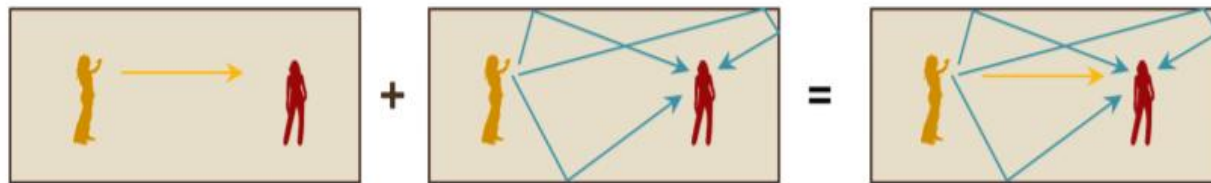
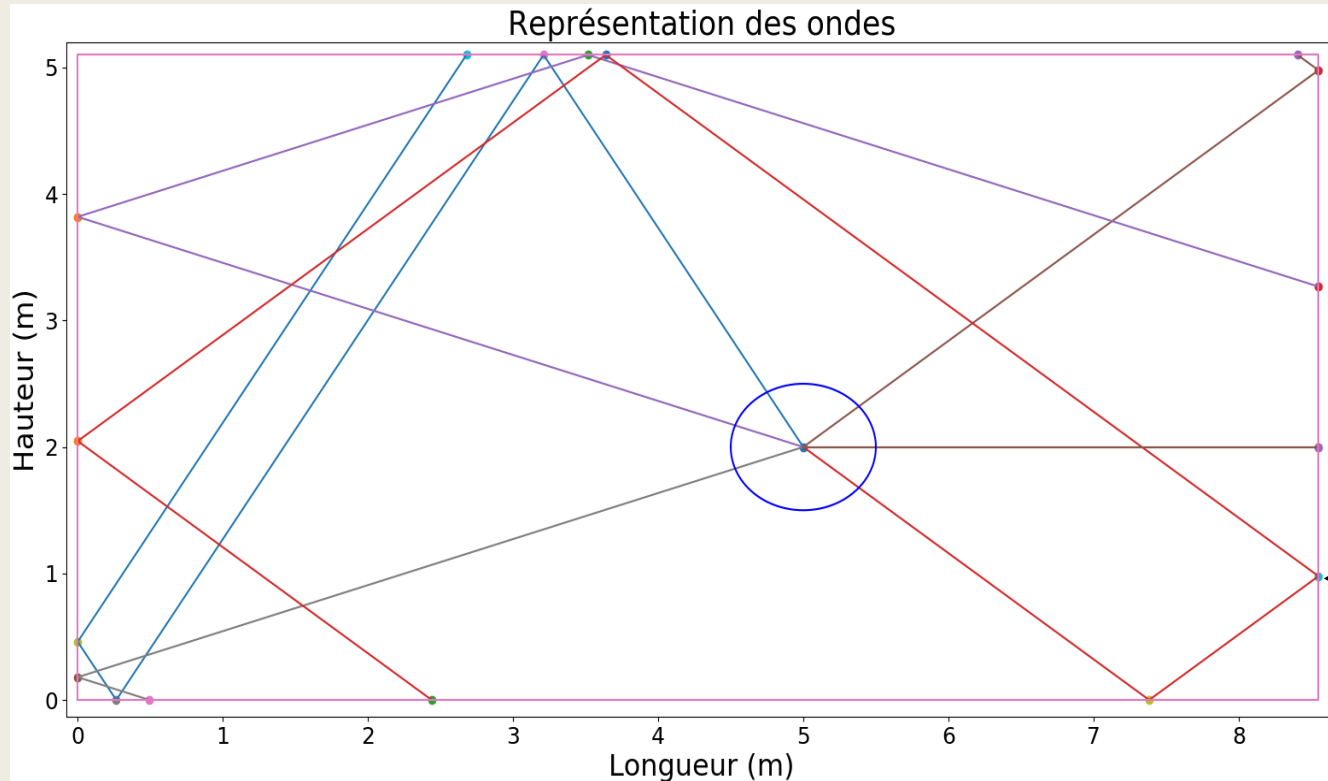


Figure 2 : L'onde directe et les ondes réfléchies se superposent et contribuent à la qualité du son perçu

- **But** → Obtenir le temps de réverbération à partir d'une modélisation numérique

Modélisation

■ Méthode des rayons pour représenter le déplacement des ondes

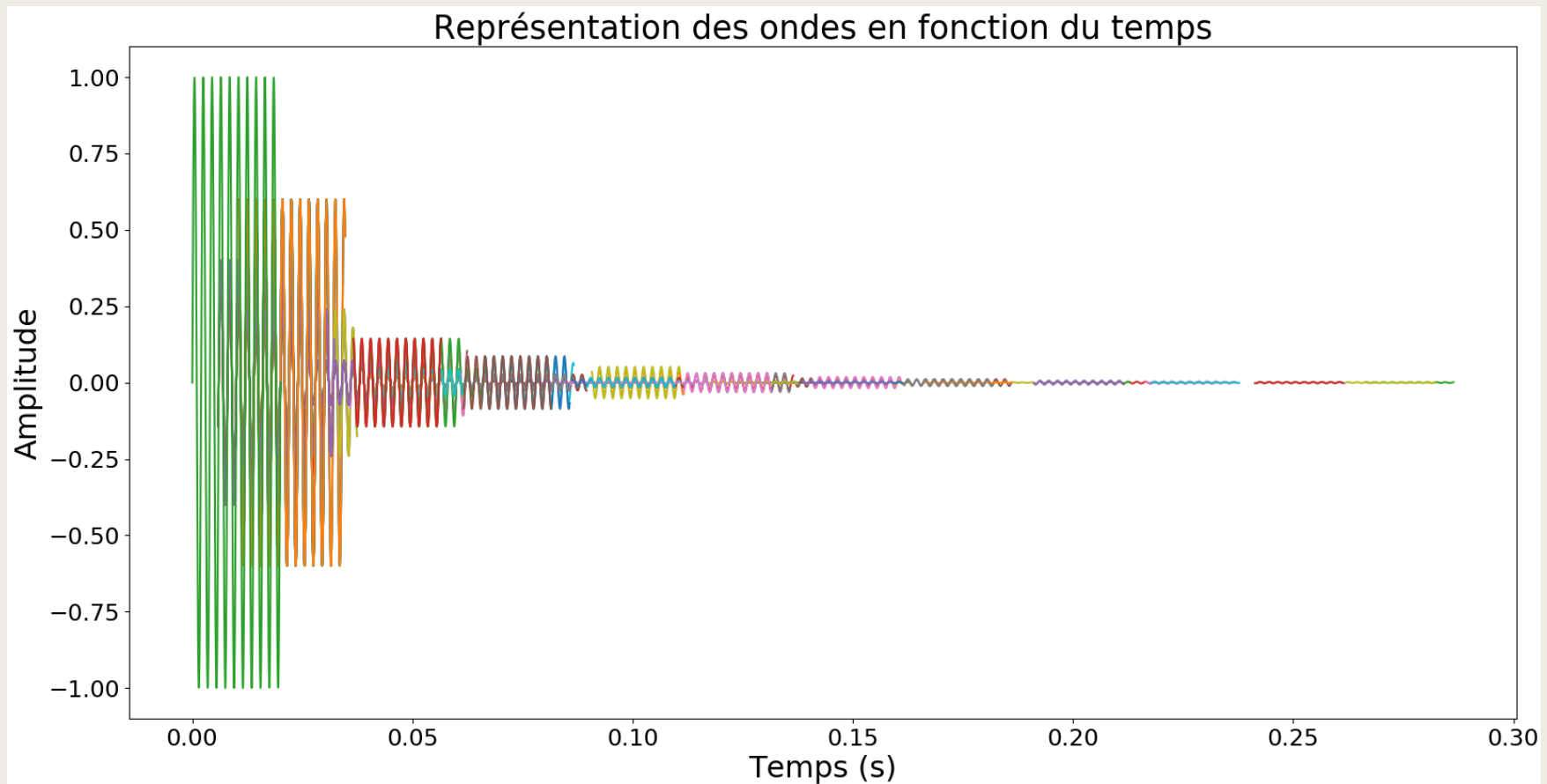


Exemple pour
10 vecteurs

- Coordonnées du nouveau point
- Vecteur après réflexion
- Nouvelle amplitude
- Distance

Ne prend en compte que les rayons ayant une amplitude supérieure à **1/1000** en arrivant dans l'oreille

Résultats



- Tracé de l'ensemble des ondes résultantes en fonction de leur temps de parcours et de leur amplitude

➡ **TR = 0,28 secondes**

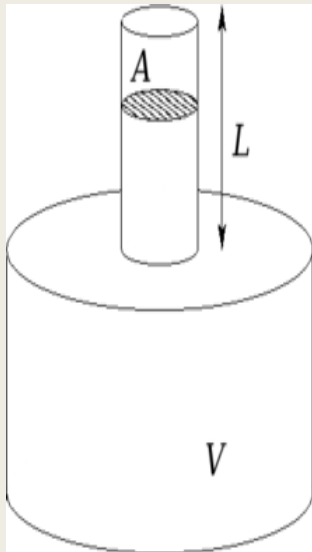
Améliorations de l'algorithme

- Modélisation réalisée seulement en deux dimensions
- Modélisation simplifiée: pas de prise en compte des matériaux occupant des petites surfaces et des objets présents
- Longueur de la pièce supposée assez petite pour éviter une perte d'amplitude dans l'air

➡ Avec les deux méthodes: TR même ordre de grandeur

II) Amélioration de l'acoustique

Etude en basse fréquence : Résonateurs d'Helmholtz



Fréquence théorique du résonateur:

$$f = c \times \sqrt{\frac{A}{L \times V}}$$

Avec A : aire de la surface du goulot

V : Volume de la cavité du résonateur ①

L: longueur du goulot

①



②



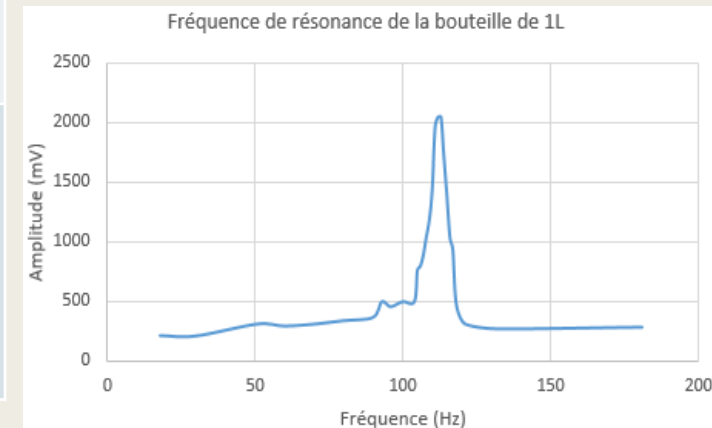
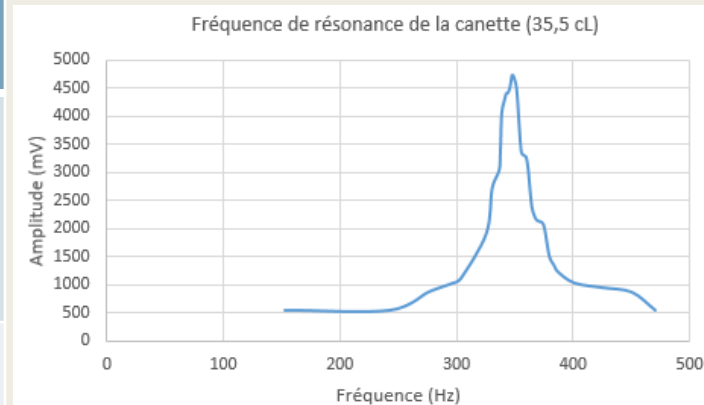
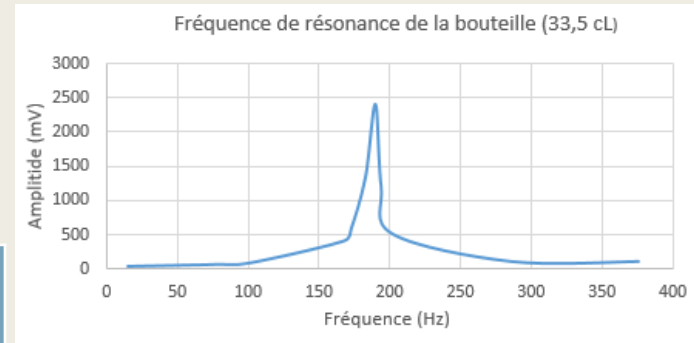
③



Résonateur utilisé	Fréquence de résonance théorique (Hz)
Bouteille de volume 33,5 cL	161 ± 25
Canette de volume 35,5 cL	344 ± 6
Bouteille de volume 1L	115 ± 14

Fréquence de résonance des résonateurs utilisés

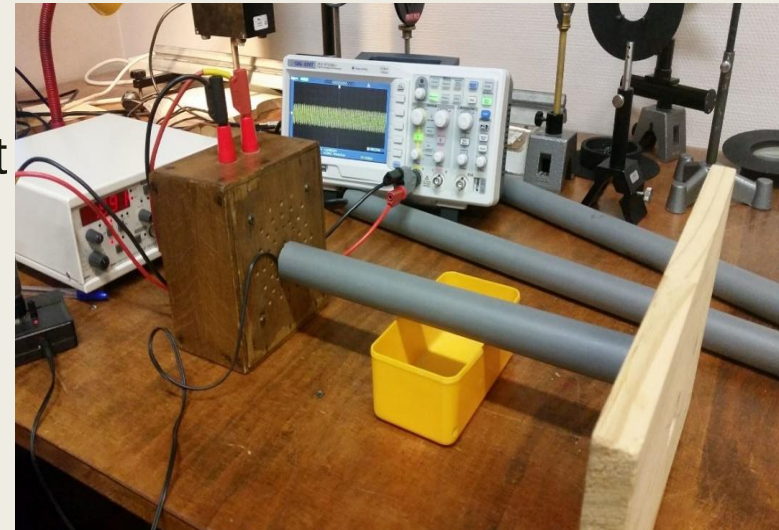
Résonateur	Fréquence de résonance théorique (Hz)	Fréquence de résonance (Hz)	Facteur de qualité	Bande passante (Hz)
Bouteille (33,5 cL)	161 ± 25	188	31	$\Delta f=6$ (192-186)
Canette (35,5 cL)	344 ± 6	343	19	$\Delta f=18$ (355-337)
Bouteille (1L)	115 ± 14	113	22,6	$\Delta f=5$ (110-115)



Influence du résonateur sur le coefficient d'absorption

Expérience:

- Mesure de l'amplification du coefficient d'absorption en fonction du résonateur utilisé



Résultats:

- Pour la canette (35,5 cL):

Fréquence de
résonance →

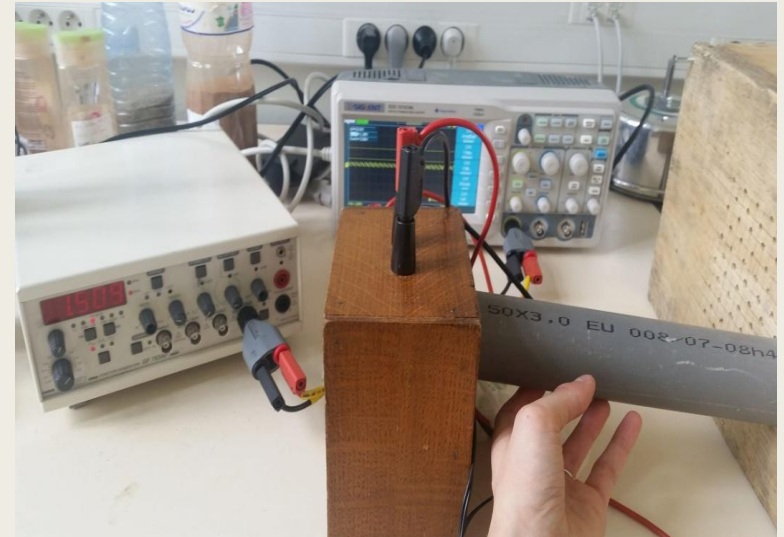
Fréquence (Hz)	α sans résonateur	α avec résonateur
274	0,52	0,52
343	0,46	0,69
395	0,49	0,57

■ Pour la bouteille de 33,5 cL:

Fréquence de résonance	Fréquence (Hz)	α sans résonateur	α avec résonateur
	171	0,43	0,45
	→ 188	0,42	0,54
	197	0,37	0,40



Etude en haute fréquence: Panneau perforé avec/sans laine de verre



Expérience:

- Mesure de l'amplification du coefficient d'absorption en fonction du résonateur utilisé

Fréquence (Hz)	α sur surface non perforée	α sur surface perforée + laine de verre	α sur surface perforée sans laine de verre
883	0,62	0,64	0,64
1000	0,59	0,65	0,58
1416	0,62	0,68	0,57
1511	0,59	0,83	0,60

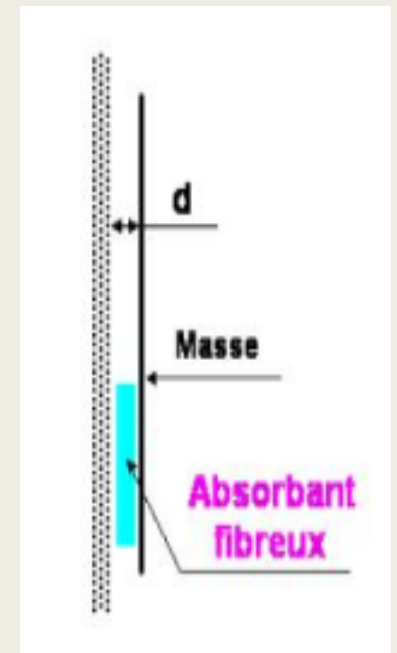


Schéma du principe du résonateur fait à partir d'un panneau perforé

Conclusion

- Obtention du temps de réverbération avec plusieurs méthodes
- Résonateur permettent de diminuer le temps de réverbération
- Valeurs obtenues avec la modélisation numérique:

Fréquence (Hz)	TR sans résonateur (s)	TR avec résonateur (s)
188	0,32	0,27
343	0,3	0,2
1511	0,28	0,13

Bibliographie

- [1] Caroline DE SA – Hélène HORSIN MOLINARO : *Conception acoustique d'une salle – Intérêt du prototypage et principe de conception de maquette*
<http://eduscol.education.fr/sti/sites/eduscol.education.fr/sti/files/ressources/pedagogiques/8440/8440-conception-acoustique-dune-salle-interet-du-prototypage-et-principes-de-conception-de-maquette.pdf>
- [2] GERMAN Arnaud - RAVERA Vincent : *La réverbération en acoustique des salles* <http://physique.unice.fr/sem6/2012-2013/PagesWeb/PT/Reverberation/rapports/rapport-acoustique.pdf>
- [3] I.U.P. GDP Option GET : *Cours d'acoustique et mécanique ondulatoire*
<http://www.gipsa-lab.grenoble-inp.fr/~yo.fujiso/acoustique/cours.pdf>
- [4] WALLACE-C. SABINE. : *Architectural Acoustics. Part. I. Reverberation*
<https://hal.archives-ouvertes.fr/jpa-00240524/document>
- [5] Ricardo ATIENZA Suzel BALEZ : *Acoustique des salles*
<http://www.grenoble.archi.fr/cours-en-ligne/atenzia/RA05-Correction-acoustique.pdf>
- [6] Frédéric POIRRIER - <http://www.conseils-acoustique.com/index.php>

ANNEXE

Fréquence théorique de résonance des différents résonateurs utilisés

Soit $P(t)$ la pression de l'air contenue dans le volume V .
 P_0 = pression atmosphérique.

$p(t) = P(t) - P_0$ surpression dans le volume

$\rightarrow$ force exercée sur l'air située dans le col $F = (P - P_0)s = B$.

(PFD) $m \ddot{x} = F$

pour les mouvements de faible amplitude
 $F \approx$ force de rappel élastique
 $F = -kx$.

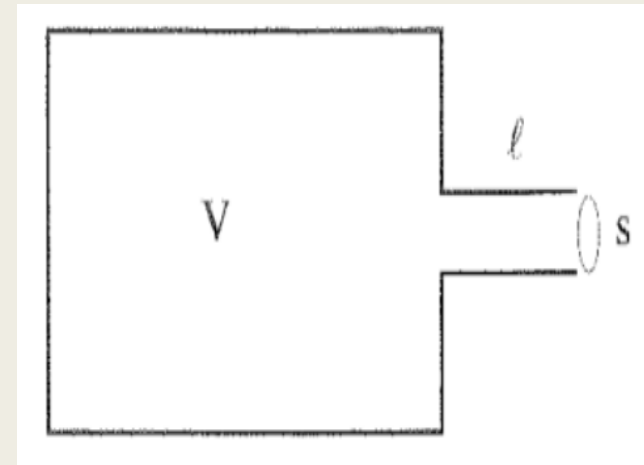
dans le col (par def compression adiabatique de l'air $\chi_B = -\frac{1}{V} \frac{\partial V}{\partial P}$),
 $(\Delta V = \Delta x s \text{ et } \Delta P = p)$

$\Rightarrow p = -\frac{\Delta x s}{\chi_B V}$ et $k = \frac{-F}{\Delta x} = + \frac{s^2}{\chi_B V}$

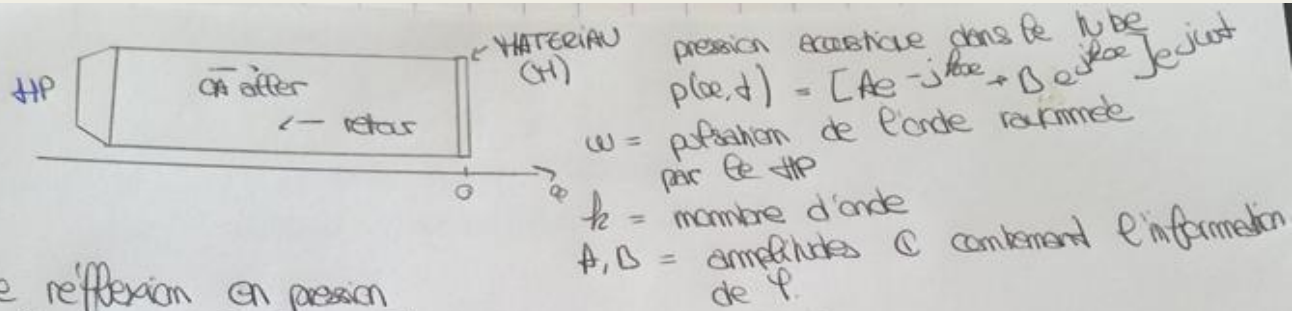
par analogie

$\omega_0 = \sqrt{\frac{k}{m}} = \sqrt{\frac{s^2}{\chi_B V \rho l s}} = \frac{1}{\sqrt{\chi_B \rho l s}} \times \sqrt{\frac{s}{V \rho l}}$

$c = \text{celle-ci du son.}$



Démonstration formule coefficient alpha



coeff de réflexion en pression
 à l'interface du (M) [$x=0$]

$$R_p = \frac{A}{B} = re^{j\varphi}$$

donc
$$p(x,t) = A[e^{-jkx} + R_p e^{jkx}]e^{j\omega t}$$

$$= A e^{-jkx} [1 + r e^{j(2kx + \varphi)}] e^{j\omega t}$$

$$P_{max} = |A| (1+r)$$

$$P_{min} = |A| (1-r)$$

en introduisant le rapport des pressions (TOS)

$$|TOS| = \left| \frac{P_{max}}{P_{min}} \right| = \frac{1+r}{1-r}$$

$$\Rightarrow r = \frac{P_{max} - P_{min}}{P_{max} + P_{min}}$$

Le coefficient d'absorption sous incidence normale est défini en énergie par

$$\alpha = 1 - |R_p|^2 = 1 - r^2 = 1 - \left(\frac{P_{max} - P_{min}}{P_{max} + P_{min}} \right)^2$$

$$= 1 - \left(\frac{U_{max} - U_{min}}{U_{max} + U_{min}} \right)^2$$

```

from pylab import *
import numpy as np
import matplotlib.pyplot as plt
import math
import random

tab=[0.4, 0.7, 0.4, 0.6] #Coefficient d'absorption [gauche, haut, droite, bas]
Lx = 8.55
Ly = 5.10

###Fonction créant une liste de 1000 vecteurs
def vecteur():
    L=[]
    teta=np.linspace(0,2*pi,1000) #Crée une liste de 1000 vecteurs
    for k in teta:
        L.append((cos(k),sin(k)))
    return(L)

Vecteur=vecteur()

###Fonction renvoyant le point d'intersection avec une paroi
def mouvement(x,u,amp,tab):
    global Lx, Ly
    ux,uy = u
    x0,y0 = x
    d=0
    if uy==0:
        if ux>0: #Tape à droite
            x1=Lx
            y1=y0
            u=(-ux,uy)
            amp=amp*(1-tab[2])
        else: #Tape à gauche
            x1=0
            y1=y0
            u=(-ux,uy)
            amp=amp*(1-tab[0])
    elif ux==0:
        if uy>0: #Tape en haut
            x1=x0
            y1=Ly
            u=(ux,-uy)
            amp=amp*(1-tab[1])
        else: #Tape en bas
            x1=x0
            y1=0
            u=(ux,-uy)
            amp=amp*(1-tab[3])

```

```

if ux>0:
    if uy>0:
        if (Lx-x0)/ux < (Ly-y0)/uy: #tape à droite
            x1 = Lx
            y1=((Lx-x0)/ux)*uy + y0
            u = (-ux, uy)
            amp= amp*(1-tab[2])
        else: #tape en haut
            y1= Ly
            x1=((Ly-y0)/uy)*ux + x0
            u = (ux, -uy)
            amp=amp*(1-tab[1])
    else:
        if ux!=0 and uy!=0 and (Lx-x0)/ux < y0/(-uy): #tape à droite
            x1=Lx
            y1=((Lx-x0)/ux)*uy + y0
            u=(-ux,uy)
            amp=amp*(1-tab[2])
        elif uy!=0 and ux!=0: #tape en bas
            x1=(y0/(-uy))*ux + x0
            y1= 0 #x
            u=(ux, -uy)
            amp=amp*(1-tab[3])
    else:
        if uy>0:
            if ux!=0 and uy!=0 and x0/(-ux) < (Ly-y0)/uy: #tape à gauche
                x1= 0
                y1= (x0/ux)*(-uy) + y0
                u = (-ux,uy)
                amp=amp*(1-tab[0])
            elif ux!=0 and uy!=0: #tape en haut
                x1= ((Ly-y0)/uy)*ux + x0
                y1= Ly
                u =(ux, -uy)
                amp=amp*(1-tab[1])
        else:
            if ux!=0 and uy!=0 and x0/(-ux) < y0/(-uy): #tape à gauche
                x1= 0
                y1= (x0/ux)*(-uy) +y0
                u = (-ux,uy)
                amp=amp*(1-tab[0])
            elif ux!=0 and uy!=0: #tape en bas
                x1= (y0/uy)*(-ux) +x0
                y1= 0
                u= (ux, -uy)
                amp=amp*(1-tab[3])
d=sqrt((y1-y0)**2 + (x1-x0)**2)
return([x1,y1],u,amp,d) #Retourne le point suivant, le nouveau vecteur,
                        #l'amplitude et la distance parcouru

```

```

def distance(point, droite):
    (x,y) = point
    (a,b,c) = droite
    d = abs(a*x + b*y+c)/sqrt(a*a + b*b)
    return(d)    #Calcul la distance d'un point à une droite

###Fonction qui renvoie les rayons avec une amplitude supérieur à 1/1000
def liste(pt_depart, rayon,liste_vecteur):
    global tab
    amp=1
    N=len(liste_vecteur)
    L=[]
    pt=pt_depart
    for k in range(N):
        L.append([])
        vect=liste_vecteur[k]
        pt_suivant,(ux,uy),amp,d=mouvement(pt_depart,vect,1,tab)
        D=d
        L[k].append([pt_suivant,(ux,uy),amp,d])
        while distance (pt_depart, (uy,-ux,ux*pt_suivant[1]-uy*pt_suivant[0])) > rayon: #Boucle tant que l'onde ne revient pas
            pt_suivant,(ux,uy),amp,d=mouvement(pt_suivant,(ux,uy),amp,tab)                #vers l'oreille
            D+=d
            L[k].append([pt_suivant,(ux,uy),amp,D])    #Rajoute à la liste l'ensemble des points suivants
    M=[]
    test=[]
    for k in range (len(L)):
        n=len(L[k])
        amp=L[k][n-1][2]
        if amp>1/1000:    #Ne garde que les vecteurs arrivant vers l'oreille avec une amplitude supérieur à 1/1000 car def du TR
            M.append(L[k])
    return(M)

liste_final=liste((5,2),0.1,Vecteur)

```

###Fonction qui trace les rayons

```
def tracer(pt_depart, rayon, liste_vecteur):      #représentation visuelle du trajet des rayons
    M=liste(pt_depart, rayon, liste_vecteur)
    m=len(M)
    plt.plot(pt_depart[0], pt_depart[1], 'o')
    for k in range(m):
        ligne_x = [pt_depart[0]]
        ligne_y = [pt_depart[1]]
        n=len(M[k])
        for j in range(n):
            pt_suivant=M[k][j][0]
            plt.plot(pt_suivant[0], pt_suivant[1], 'o')
            ligne_x.append(pt_suivant[0])
            ligne_y.append(pt_suivant[1])
        plt.plot(ligne_x, ligne_y)
    axeX = np.array([0, Lx, Lx, 0, 0])
    axeY = np.array([0, 0, Ly, Ly, 0])
    plt.plot(axeX, axeY)
    plt.xlim(-1, Lx+1)
    plt.ylim(-1, Ly+1)
    r = rayon
    x= np.linspace((pt_depart[0]-r), (pt_depart[0]+r), 1000)
    y = np.sqrt(r**2 - (x-pt_depart[0])**2)+pt_depart[1]
    plt.plot(x, y, 'b')
    plt.plot(x, -y +2*pt_depart[1], 'b')
    plt.title('Représentation des ondes', fontsize=25)
    plt.xlabel("Longueur (m)", fontsize=23)
    plt.ylabel("Hauteur (m)", fontsize=23)
    plt.show()
```

###Fonction qui renvoie l'ensemble des amplitudes des ondes reçues au niveau de l'oreille en fonction du temps

```
def fonction(L):
    test=[]
    for k in range (len(L)):
        n=len(L[k])
        test.append(n)
        temps=L[k][n-1][3]/340
        amp=L[k][n-1][2]
        x1=np.linspace(temps, (10/500)+temps, 500)
        y1=sin(2*pi*(x1)*500)*amp
        plt.plot(x1, y1)
    x=np.linspace(0, 10/500, 500)
    y=np.sin(2*pi*x*500)
    plt.title('Représentation des ondes en fonction du temps', fontsize=25)
    plt.xlabel("Temps (s)", fontsize=23)
    plt.ylabel("Amplitude", fontsize=23)
    plt.plot(x, y)
    plt.show() #Renvoie le graphique de l'ensemble des ondes en fonction du temps
```

Obtention des TR numériquement avec et sans résonateur

