

# **TP2- Jeux de mots**

## **Synthèse et analyse spectrale d'une gamme de musique**

Aya Habiballah

Mr Alae Ammour

29 janvier 2023

## Objectifs

Comprendre comment manipuler un signal audio avec Matlab, en effectuant certaines opérations classiques sur un fichier audio d'une phrase enregistrée via un smartphone.

Comprendre la notion des sons purs à travers la synthèse et l'analyse spectrale d'une gamme de musique.

## Jeux de mots

1: Sauvegardez ce fichier sur votre répertoire de travail, puis chargez-le dans MATLAB à l'aide de la commande « audioread ».

**On enregistre l'audio sur notre machine, puis on le convertit en fichier .wav**

```
[y,Fs]=audioread('audi1.wav');  
dt = 1/Fs;  
t = 0:dt:(length(y)-1)*dt;
```

2: Tracez le signal enregistré en fonction du temps, puis écoutez-le en utilisant la commande « sound ».

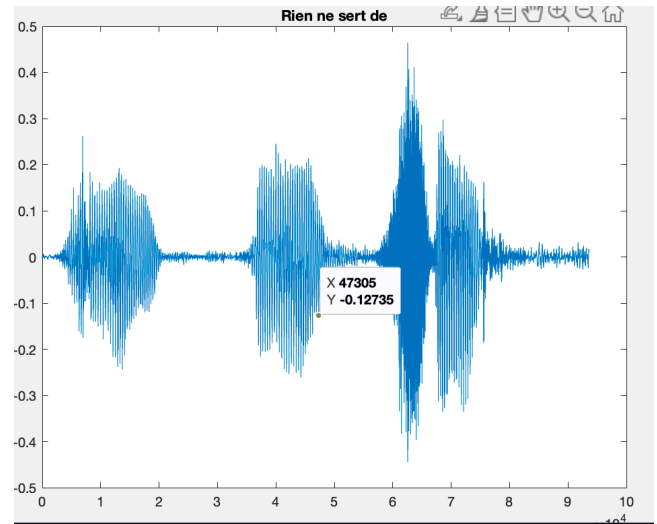
```
[x,fs]=audioread("audio1.au");  
  
Taille = length(x);  
ts=1/fs;  
T = (0:Taille-1)*ts;  
  
sound(x,fs);  
plot(T,x);  
legend("Représentation du signal");  
xlabel("t");  
ylabel("x");
```

3:

```
%Ecoutez la phrase en modifiant la fréquence d'échantillonnage qui va  
%rendre le son grave/aigus  
sound(x,1/8*fs);  
sound(x,2*fs);
```

4:Tracez le signal en fonction des indices du vecteur x, puis essayez de repérer les indices de début et de fin de la phrase « Rien ne sert de ».

```
seg1 = x(36500:130106);  
plot(seg1);
```



5:Créez ce vecteur, puis écoutez le mot segmenté.

```
%créer le vecteur  
sound(seg1, fs);
```

5:Segmentez cette fois-ci toute la phrase en créant les variables suivantes : riennesertde, courir, ilfaut, partirapoint.

```
%segmentation1 'rien ne sert de'
seg1 = x(36500:130106);
plot(seg1);
title('Rien ne sert de');

%segmentation2 'courir'
seg2=x(130107:190006);
plot(seg2);
title('courir');

%segmentation3 'il faut'
seg3=x(190007:250006);
plot(seg3);
title('il faut');

%segmentation4 'partir a point'
seg4=x(250007:394240);
plot(seg4);
title('partir a point');
```

7:la phrase synthétisée « Rien ne sert de partir à point, il faut courir ».

```
%phrase complete|
sound([seg1;seg4;seg3;seg2],fs);
```

## Synthèse et analyse spectrale d'une gamme de musique

### – Synthèse d'une gamme de musique

1:Créez un programme qui permet de jouer une gamme de musique.

```
clear all
close all
clc

fe = 8192;
te = 1/fe;
t = [0:te:0.5];
%les signaux
Do = sin(2*pi*t*261.62);
Dod = sin(2*pi*t*277.18);
Re = sin(2*pi*t*293.66);
Red = sin(2*pi*t*311.12);
Mi = sin(2*pi*t*329.62);
Fa = sin(2*pi*t*349.22);
Fad = sin(2*pi*t*370);
Sol = sin(2*pi*t*392);
Sold = sin(2*pi*t*415.30);
La = sin(2*pi*t*440);
Lad = sin(2*pi*t*466.16);
Si = sin(2*pi*t*494.88);
Do2 = sin(2*pi*t*523.25);

doremifasol_solfamiredo= [Do,Re,Mi,Fa,Sol,La,Si,Do2,Do2,Si,La,Sol,Fa,Mi,Re,Do];
Gamme=[Do,Re,Mi,Fa,Sol,La,Si,Do2];
```

## -Spectre de la gamme de musique :

2:

```
%analyse du spectre|
signalAnalyzer(Gamme);
spectrogram(Gamme)
a = length(Gamme);
fshift = (-a/2:(a/2)-1)*(fe/a);
y = fft(Gamme);
```

3:Tracez le spectrogramme :

