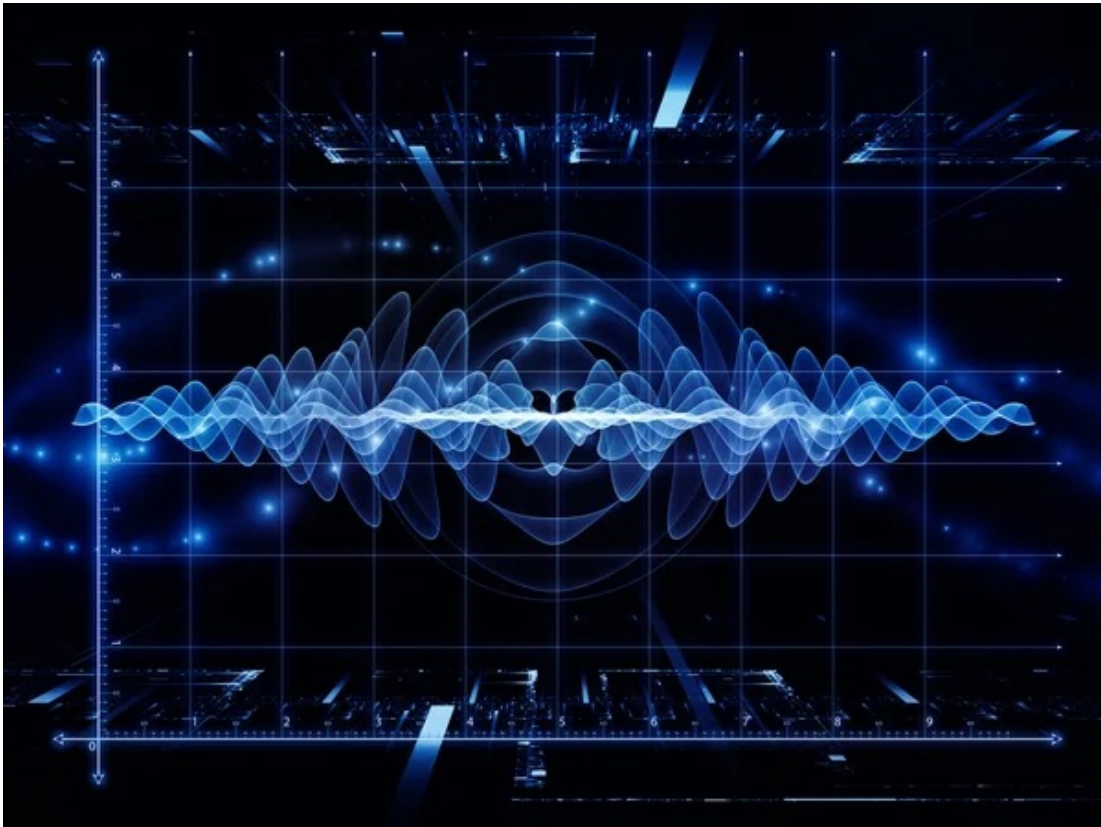


# Rapport Tp2

Encadré par : M Alae  
Ammour



## Objectifs

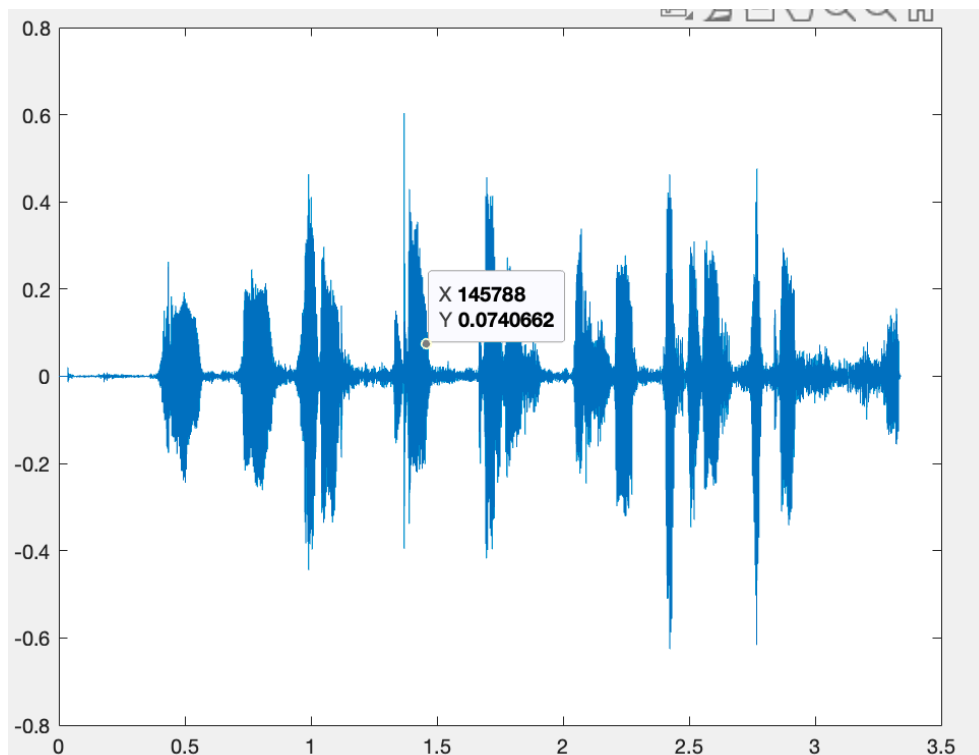
*Comprendre comment manipuler un signal audio avec Matlab, en effectuant certaines opérations classiques sur un fichier audio d'une phrase enregistrée via un smartphone.*

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

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

```
[y,Fs]=audioread('phrase.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("phrase.au");
```

```
Taille = length(x);
```

```
ts=1/fs;
```

```
T = (0:Taille-1)*ts;
```

```
sound(x,fs);
```

```
plot(T,x);
```

```
legend("Représentation du signal du son");
```

```
xlabel("t");
```

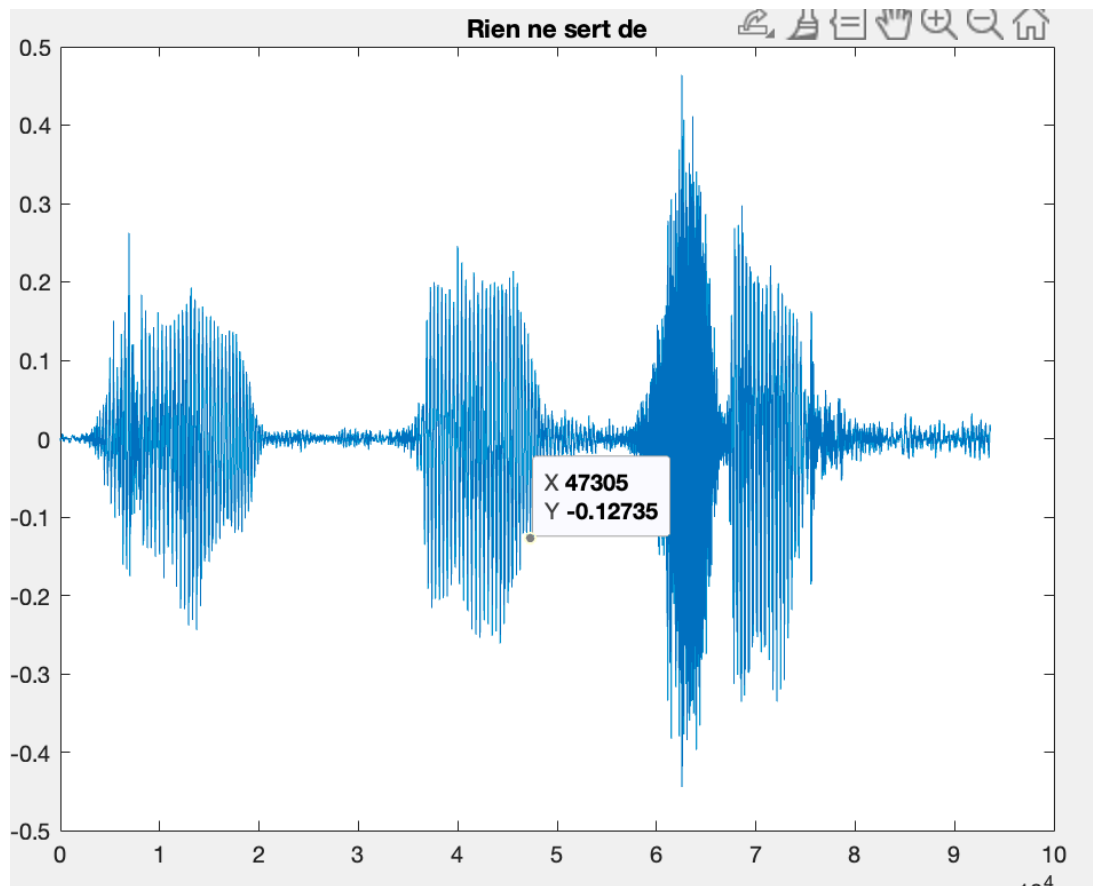
```
ylabel("x(t)");
```

3)

```
sound(y,2*Fs); %Donald Duck
```

```
sound(y,Fs/2); %Terminator
```

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);
title('Rien ne sert de');
```

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

```
sound(seg1,fs);
```

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

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

%second segmentation 'courir'
seg2=x(130107:190006);

%third segmentation 'il faut'
seg3=x(190007:250006);

%4th segmentation 'partir a point'
seg4=x(250007:394240);
```

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

```
sound([seg1;seg4;seg3;seg2],fs);
```

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

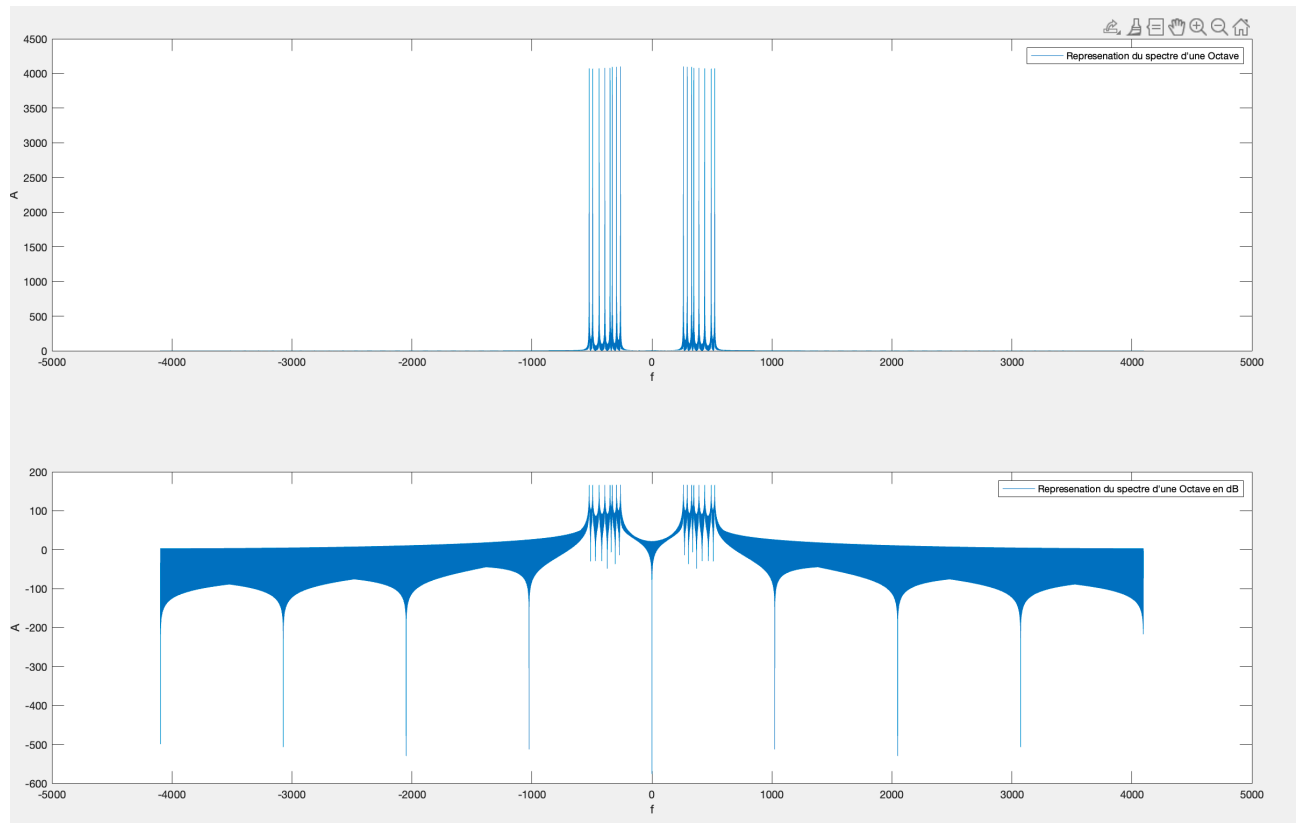
1) Créez un programme qui permet de jouer une gamme de musique. La fréquence de chaque note est précisée dans le tableau ci-dessous.

```
m_Fs=8192;
Ts=1/m_Fs;
t=[0:Ts:1];
F_A=440;
F_dol=262;
F_re=294;
F_m=330;
F_fa=349;
F_sol=392;
F_si=494;
F_do2=523;
A=sin(2*pi*F_A*t);
Dol=sin(2*pi*F_dol*t);
re=sin(2*pi*F_re*t);
mi=sin(2*pi*F_m*t);
fa=sin(2*pi*F_fa*t);
so=sin(2*pi*F_sol*t);
la=sin(2*pi*F_A*t);
si=sin(2*pi*F_si*t);
do=sin(2*pi*F_do2*t);
doremifasol_solfamiredo= [Dol,re,mi,fa,so,la,si,do,do,si,la,so,fa,mi,re,Dol];
faded = [fa,fa,fa,si,mi,mi,re,si,si,si,fa,fa,fa,mi];
doremifa = [Dol,re,mi,fa,so,la,si,do];
inv = [do,si,la,so,fa,mi,re,do];
```

2)- Utilisez l'outil graphique d'analyse de signaux signalAnalyzer pour visualiser le spectre de votre gamme.

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

3)

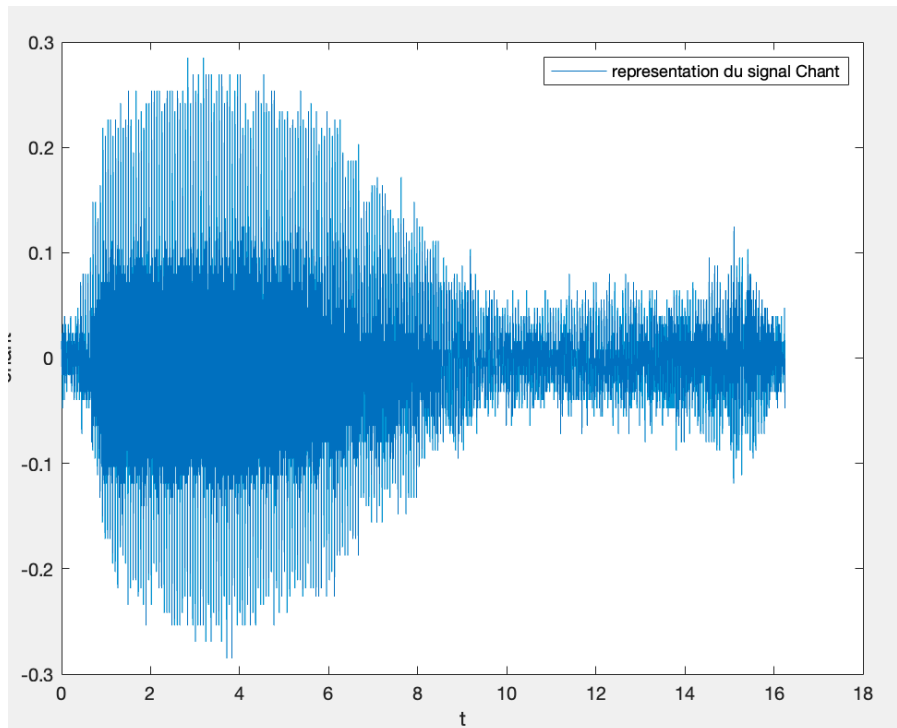


```
subplot(2,1,1)
plot(fshift,fftshift(abs(y)));
legend("Représentation du spectre d'une Octave");
xlabel("f");
ylabel("A");
subplot(2,1,2)
sig = 20*log(fftshift(abs(y)));

plot(fshift,sig);
legend("Représentation du spectre d'une Octave en dB");
xlabel("f");
ylabel("A");
```



1) Chargez, depuis le fichier 'bluewhale.au', le sous-ensemble de données qui correspond au chant du rorqual bleu du Pacifique. En effet, les appels de rorqual bleu sont des sons à basse fréquence, ils sont à peine audibles pour les humains. Utiliser la commande `audioread` pour lire le fichier. Le son à récupérer correspond aux indices allant de  $2.45 \times 10^4$  à  $3.10 \times 10^4$ .



```
sound(chant,fs);  
plot(t,chant);  
legend("representation du signal Chant");  
xlabel("t");  
ylabel("chant");
```

2) Ecoutez ce signal en utilisant la commande `sound`, puis visualisez-le.

