

compte rendu du :

TP2- Jeux de mots

Synthèse et analyse spectrale d'une gamme de musique

Réalisé par : Hiba QOUIQA

filière : IA

Encadré par : Mr. Alae AMMOUR

2022-2023



Sommaire :

1. Buts du Tp.
2. Jeux de mots.
3. Synthèse et analyse spectrale d'une gamme de musique :
 - a.- Synthèse d'une gamme de musique
 - b. - Spectre de la gamme de musique
 - c. - Approximation du spectre d'un signal sinusoïdal à temps continu par FFT.
4. Conclusion.

1. Buts du Tp :

- 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.

2. Jeux de mots :

1.« Centre.wave » est un fichier audio qu'on a enregistré à l'aide d'un smartphone, en prononçant lentement la phrase :

- « Rien ne sert de courir, il faut partir à point ».

```
filename = 'Centre.wav';  
[y,Fs] = audioread(filename);
```

[y,Fs] = audioread(filename) reads data from the file named filename, and returns sampled data, y, and a sample rate for that data, Fs.

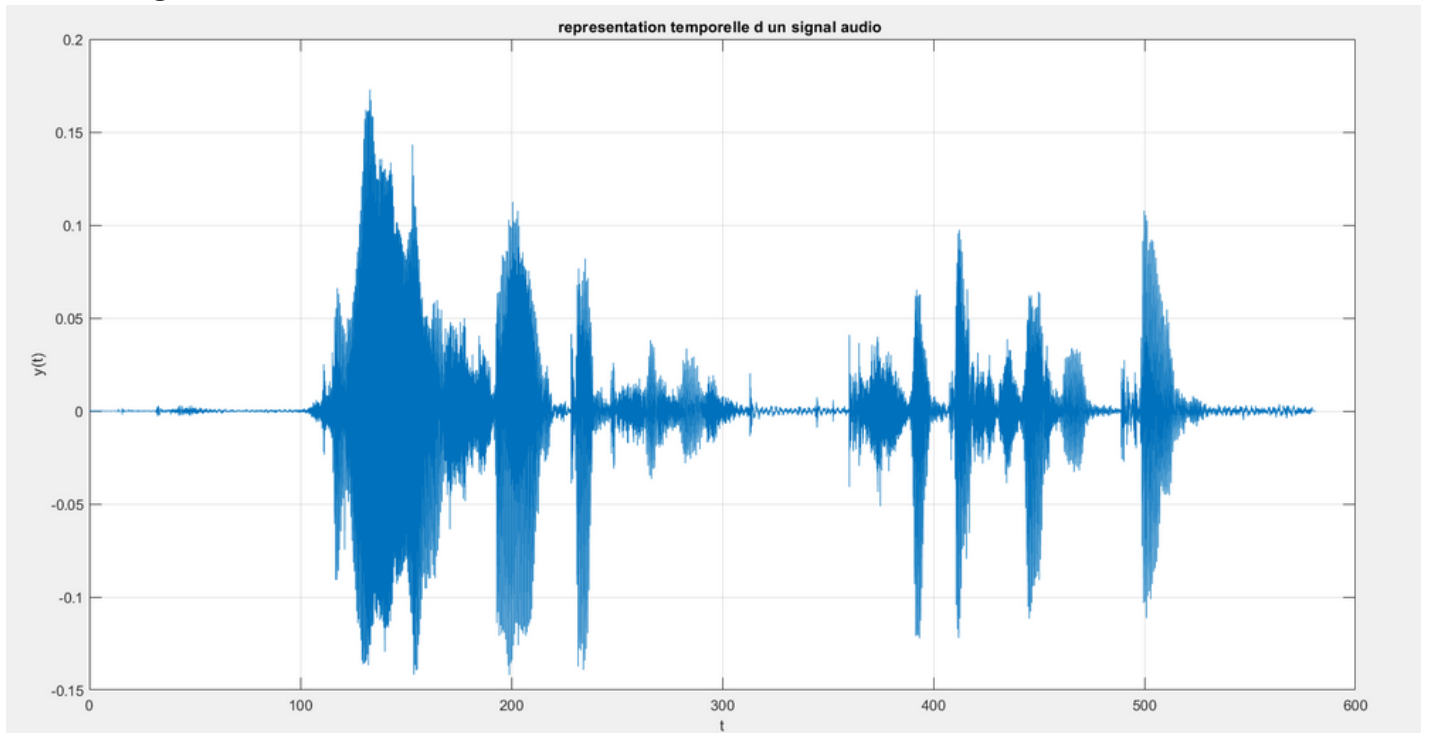
2. Pour tracer le signal enregistré on a utilisé une fréquence d'échantillonnage d'enregistrement ($F_s = 48000\text{Hz}$) et on a généré un vecteur de même taille que le vecteur y pour qu'il donne le même nombre d'échantillons.

le script Matlab :

```
Ts= 1/Fs;  
N=length(y); % le nbr d'échantillons égal à la taille du vecteur y  
  
t=0:Ts:(N-1)*Ts;  
  
plot(t,y)  
title('représentation temporelle d un signal audio ')  
xlabel('t')  
ylabel('y(t)')  
grid on  
  
sound(y,Fs)
```

la commande sound(y,Fs) nous a permis d'entendre l'audio qu'on vient d'enregistrer.

- la figure:



3. Dans Cette étape on modifie la fréquence d'échantillonnage à double ou deux fois plus petite que la fréquence d'origine . En effet, cette modification de la fréquence d'échantillonnage revient à appliquer un changement d'échelle $x(t) = y(at)$ avec ($a=2$) la valeur du facteur d'échelle, cela revient à opérer une dilatation du spectre initial d'où la version plus grave du signal écouté.

```
Fe=Fs/2; % fréquence d'echantillonnage est deux fois plus petite que la fréquence d'origine
sound(y,Fe)
```

- Remarque:**

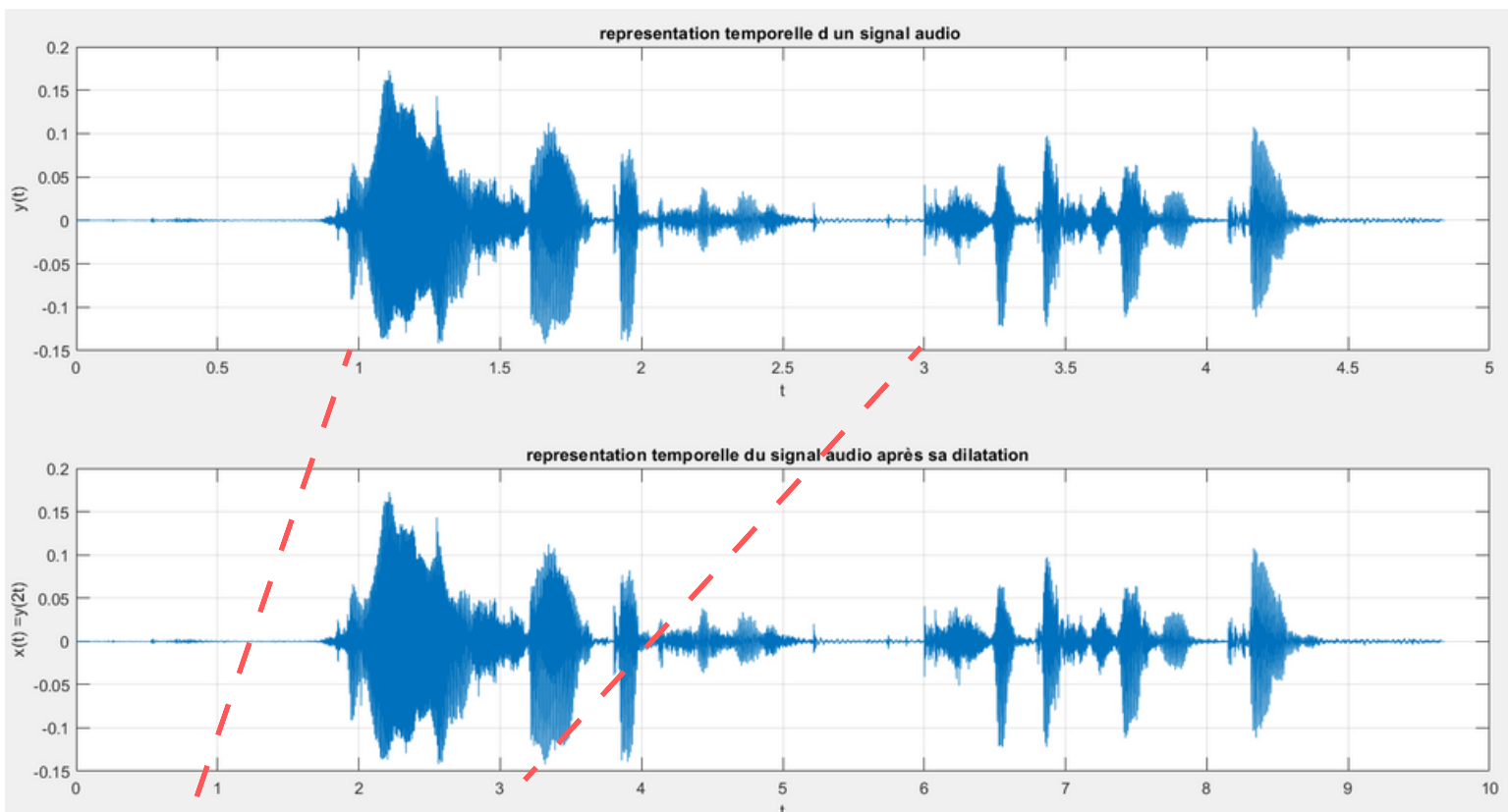
-Le signal entendu est une version plus longue que la version d'origine.
 -si on trace les deux signaux on va bien remarquer que le deuxième signal est la version dilatée du premier signal , comme il est affiché dans les figures suivantes :

```
subplot(2,1,1)
plot(t,y)
title('representation temporelle d un signal audio ')
xlabel('t')
ylabel('y(t)')
grid on

sound(y,Fs)

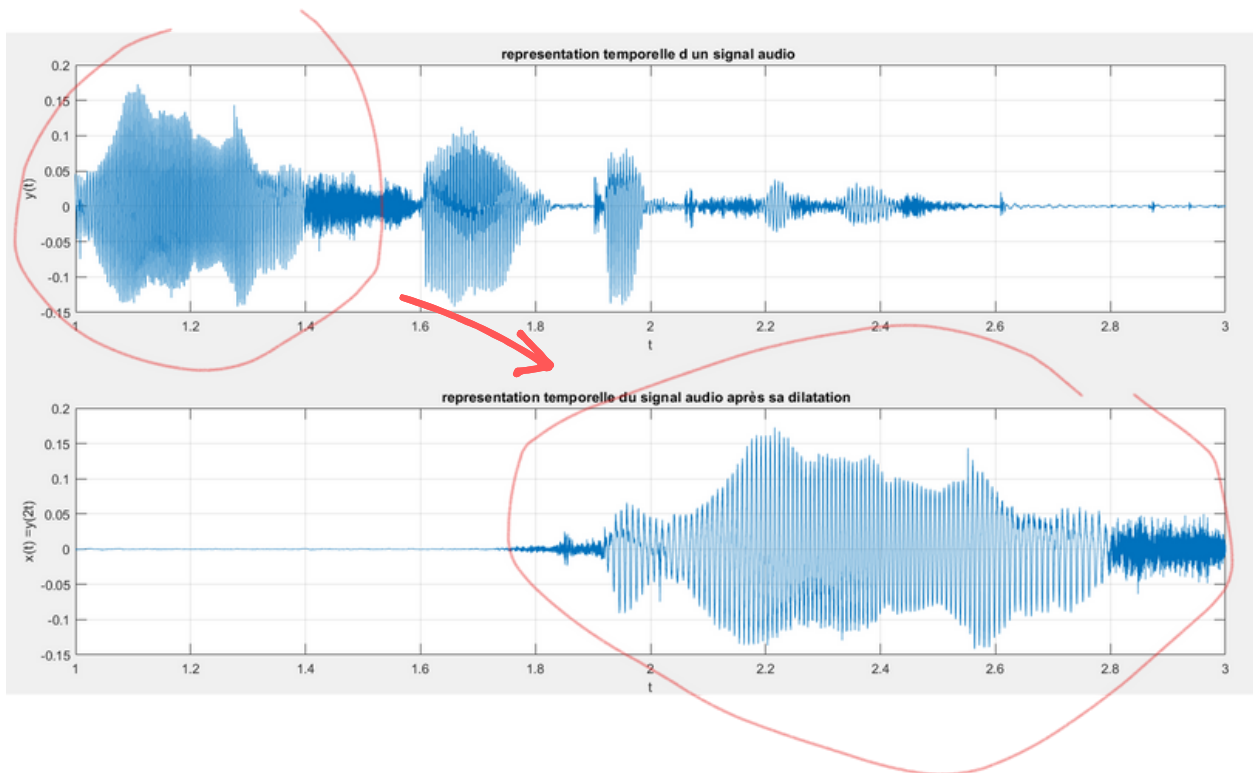
Fe=Fs/2; % fréquence d'echantillonnage est deux fois plus petite que la fréquence d'origine
Te= 1/Fe;
t2=0:Te:(N-1)*Te;
sound(y,Fe)

subplot(2,1,2)
plot(t2,y)
title('representation temporelle du signal audio après sa dilatation ')
xlabel('t')
ylabel('x(t) =y(2t)')
grid on
```

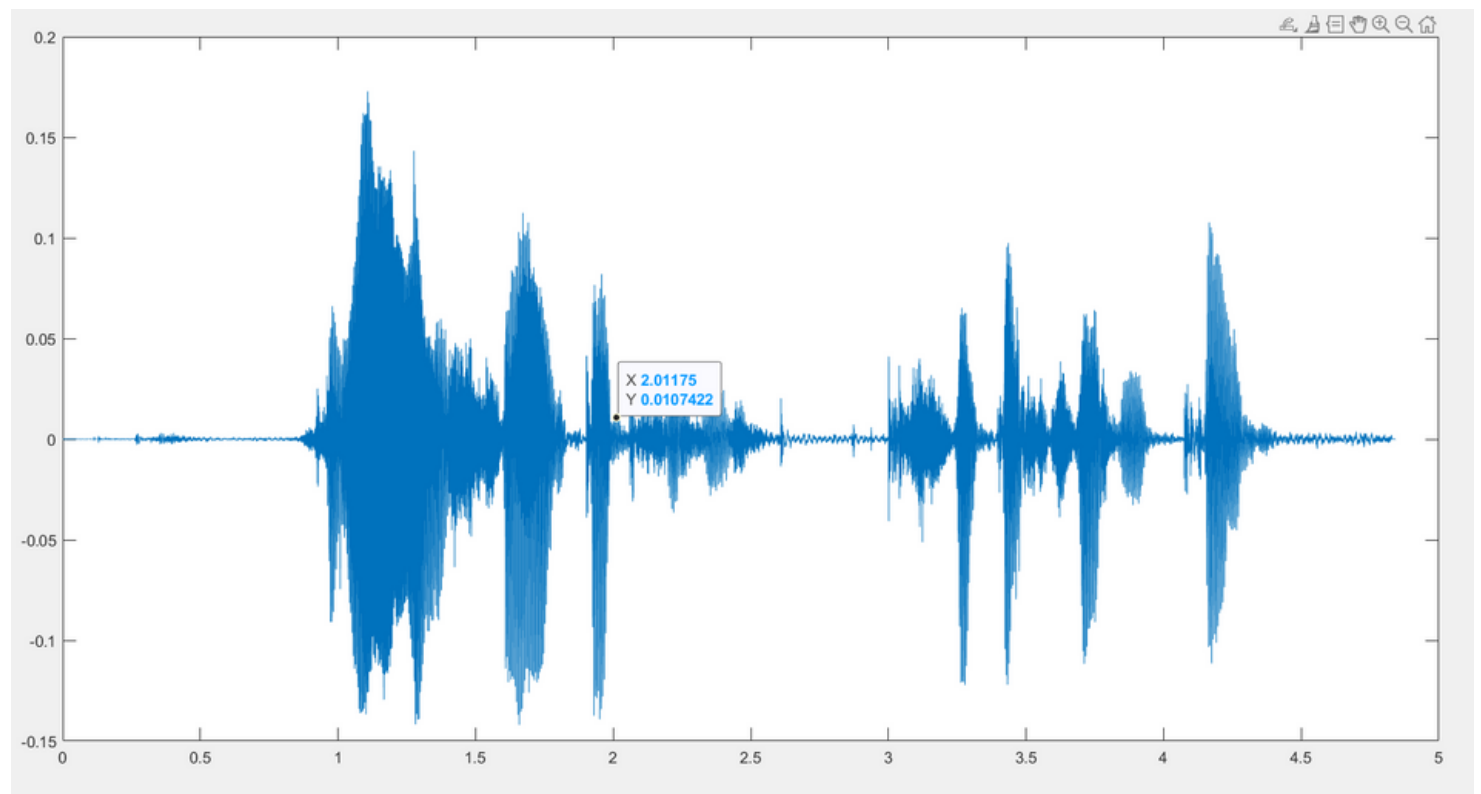


- Remarque:

- pour une vision plus claire on utilise la commande : `xlim([1 3]);`



4- Après le traçage du signal en fonction des indices du vecteur x, on remarque que la phrase "rien ne sert de " se termine dans la seconde 2.01



5- On crée un vecteur « riennesertde » contenant les n premières valeurs du signal enregistré x qui correspondent à ce morceau.

```
%% question 5
riennesertde=y(1:2.01*Fs);
sound(riennesertde,Fs)
```

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

```
%% question 6
courir=y(2.01*Fs:2.6*Fs);
sound(courir,Fs)

ilfaut=y(3*Fs:3.4*Fs);
sound(ilfaut,Fs)

partirapoint=y(3.4*Fs:4.5*Fs);
sound(partirapoint,Fs)
```

8-on crée un vecteur "parole"qui présente le signal parole réarrangé de cette façon « Rien ne sert de partir à point, il faut courir ».

%% Question 7:

```
parole=[rien nesert de ,partir a point ,il faut,courir];  
sound(parole,Fs)
```

2. Synthèse et analyse spectrale d'une gamme de musique :

- Synthèse d'une gamme de musique

1- on va créer un programme qui permet de jouer une gamme de musique. en utilisant les fréquence des notes dans le tableau ci-dessous. Chaque note aura une durée de 1s. La durée de la gamme sera donc de 8s. La fréquence d'échantillonnage f_e sera fixée à 8192 Hz.

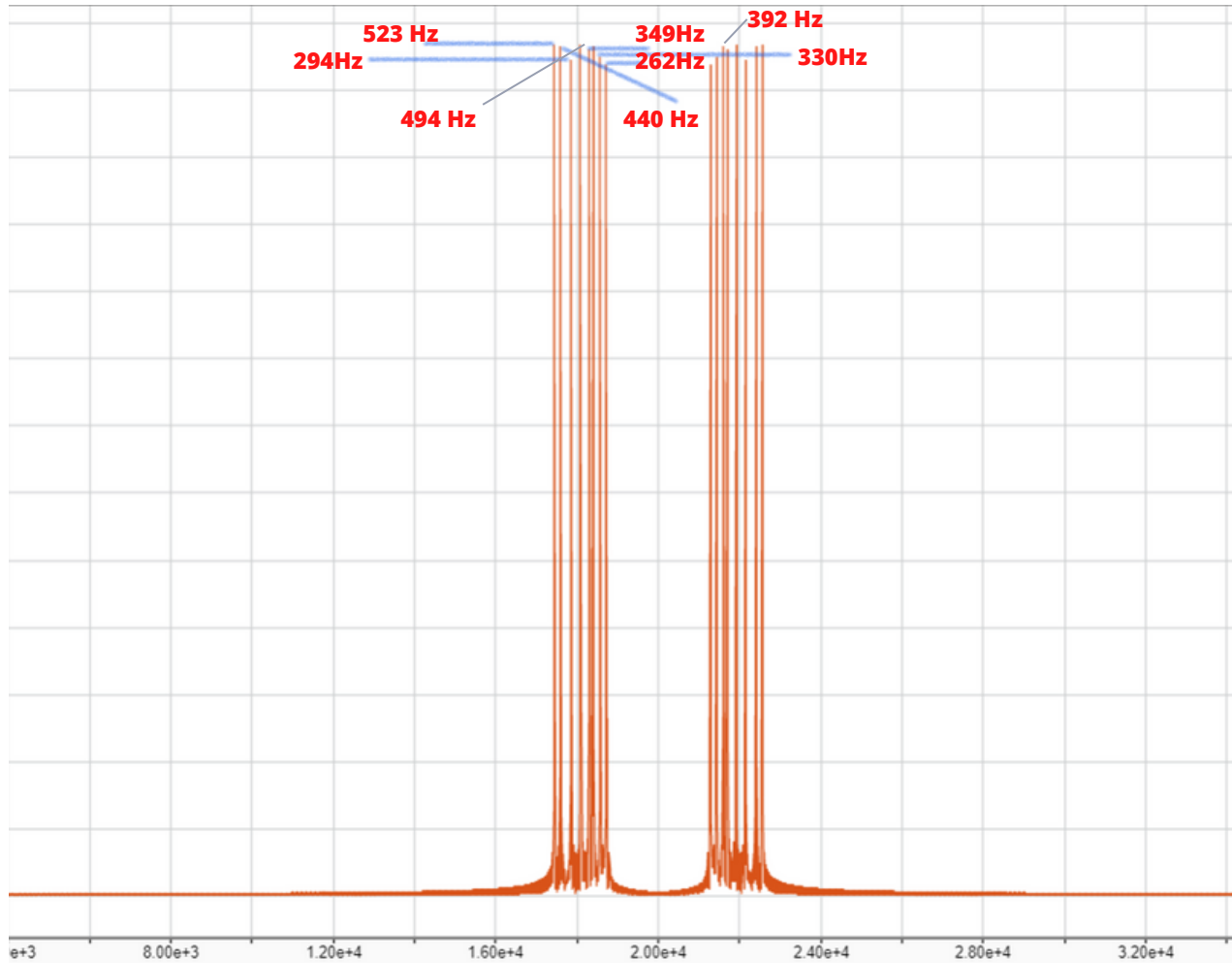
Do1	Ré	Mi	Fa	Sol	La	Si	Do2
262 Hz	294 Hz	330 Hz	349 Hz	392 Hz	440 Hz	494 Hz	523 Hz

le programme utilisé :

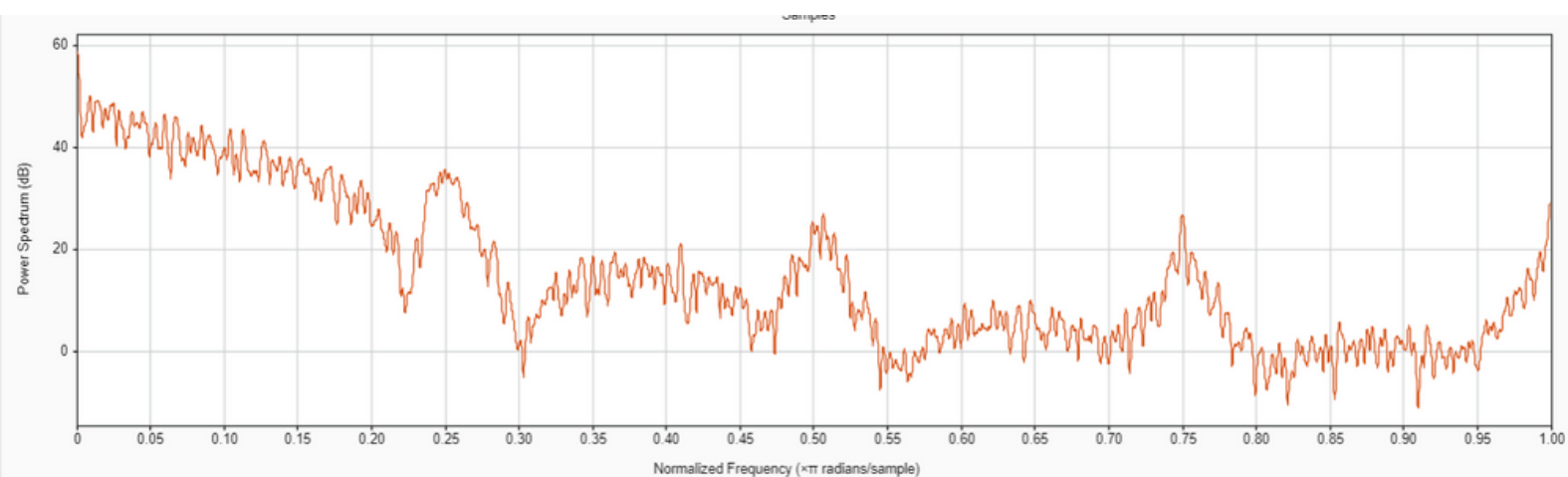
```
%%  
% Question 1:  
  
Fe=8192;% la fréquence d'échantillonnage  
te=1/Fe;  
N=5000;%nbr 'échantillons'  
t=(0:N-1)*te;  
  
%la note Do  
f_do=262;%frequence de la note Re  
Do=10*cos(2*pi*f_do*t);  
sound(Do,Fe)  
  
%la note Re  
f_Re=294;%frequence de la note Re  
Re=10*cos(2*pi*f_Re*t);  
sound(Re,Fe)  
  
%la note Mi  
f_Mi=330;%frequence de la note Mi  
Mi=10*cos(2*pi*f_Mi*t);  
sound(Mi,Fe)  
  
%la note Fa  
f_fa=349;%frequence de la note Fa  
Fa=10*cos(2*pi*f_fa*t);  
sound(Fa,Fe)  
  
%la note sol  
f_sol=392;%frequence de la note sol  
sol=10*cos(2*pi*f_sol*t);  
sound(sol,Fe)  
  
%la note La  
f_la=440;%frequence de la note La  
La=10*cos(2*pi*f_la*t);  
sound(La,Fe)  
  
%la note si  
f_si=494;%frequence de la note si  
si=10*cos(2*pi*f_si*t);  
sound(si,Fe)  
  
%la note Do2  
f_Do2=523;%frequence de la note do2  
Do2=10*cos(2*pi*523*t);  
sound(Do2,Fe)  
  
musique=[Do,Re,Mi,Fa,sol,La,si,Do2];  
%sound(musique,Fe)  
  
musique2=[Do,Re,Mi,Do,Do,Re,Mi,Do2];  
sound(musique2,Fe)
```

- Spectre de la gamme de musique

2- Pour visualiser le spectre de la gamme de musique qu'on a créé il faut utiliser "signalAnalyzer" mais d'abord on faut transformer notre signal du domaine temporel au domaine fréquentiel à l'aide de la transformée de fourier, comme il est fait dans le programme ci-dessous :



3- On trace le spectrogramme qui permet de visualiser le contenu fréquentiel du signal au cours du temps:



- Approximation du spectre d'un signal sinusoïdal à temps continu par FFT

4- Le spectre d'un signal à temps continu peut être approché par transformée de Fourier discrète (TFD) ou sa version rapide (Fast Fourier Transform (FFT)). Afficher le spectre de fréquence de la gamme musicale créée en échelle linéaire, puis avec une échelle en décibel.

Conclusion :

En conclusion, ce TP nous a permis de comprendre comment manipuler un signal audio avec Matlab. Nous avons effectué certaines opérations classiques sur un fichier audio enregistré via un smartphone. Nous avons également compris la notion des sons purs à travers la synthèse et l'analyse spectrale d'une gamme de musique. Nous devons être conscients des différences entre le temps continu et le temps discret en raison de l'utilisation de Matlab. Nous devons également veiller à tracer les figures avec les axes et les légendes appropriées. Ce TP nous a demandé de créer un script Matlab commenté pour documenter notre travail et nos observations. Nous avons manipulé un fichier audio "phrase.wav" pour le segmenter en mots distincts, puis nous avons réarrangé ce vecteur pour synthétiser une nouvelle phrase. Nous avons également synthétisé une gamme de musique en utilisant des fréquences spécifiques pour chaque note.