

# TP1

# 3DIS

TP fait avec Matlab 2012a

## I.Manipulation de fichiers numériques

### 1.Signal audio

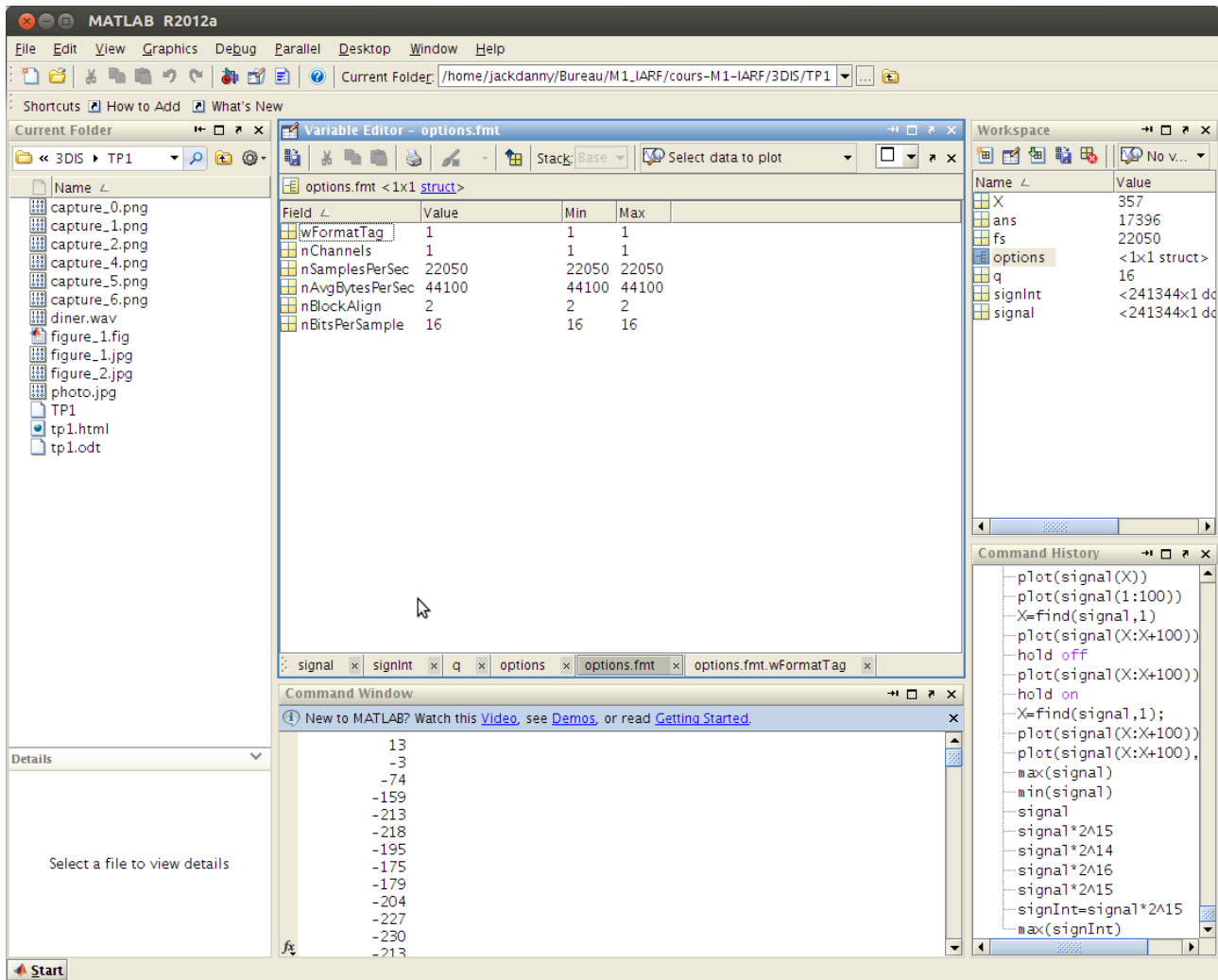
Chargement du signal

```
>> [signal,fs,q,options] = wavread('diner.wav');
```

Des informations sont contenues dans le paramètre options. Options est visiblement une structure au format fmt. Pour récupérer les informations, il suffit d'exécuter la commande :

```
>> options.fmt  
ans =  
  
    wFormatTag: 1  
    nChannels: 1  
    nSamplesPerSec: 22050  
    nAvgBytesPerSec: 44100  
    nBlockAlign: 2  
    nBitsPerSample: 16
```

On peut aussi afficher les données en cliquant sur la structure en question dans la fenêtre Workspace comme ci-dessous :

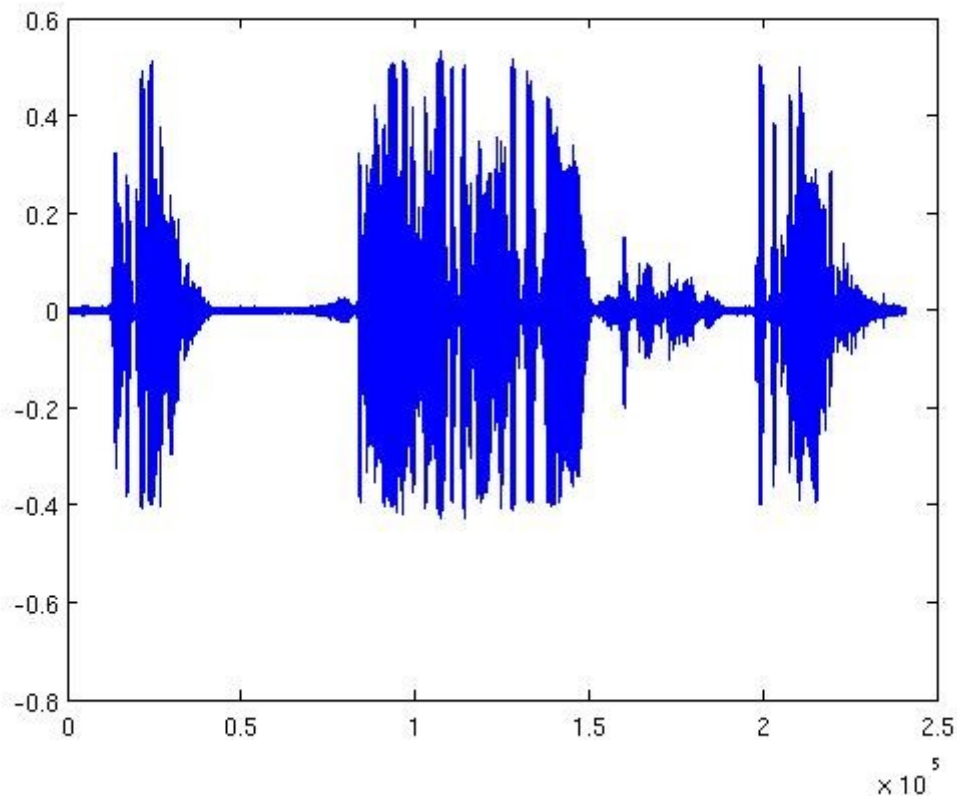


On a donc une fréquence d'échantillonnage de 22050 Hz, une quantification de 16 bits et 1 canal.

On affiche ensuite le signal :

```
>> plot(signal)
```

On obtient la figure suivante:



On souhaite afficher les 10 premières valeurs différentes de 0. Graphiquement c'est assez galère on y voit rien.

`find(X,K)` renvoie les K premiers indices non nuls. Il y a deux manières de comprendre la question

**Cas 1 :**

soit on affiche les 10 valeurs à partir de la première différente de zéro:

```
>> X=find(signal,1)
```

```
X =
```

```
357
```

```
>> signal(X:X+9)
```

```
ans =
```

```
1.0e-04 *
```

```
-0.3052
```

```
-0.3052
```

```
-0.3052
```

```
-0.3052
```

```
-0.3052
```

```
-0.3052
```

```
0
```

```
0
```

```
0
```

```
0
```

### Cas 2 :

Soit on affiche les 10 premières valeurs chacune non nuls

```
>> X=find(signal,10)
```

```
X =
```

```
357  
358  
359  
360  
361  
362  
377  
378  
379  
380
```

```
>> signal(X)
```

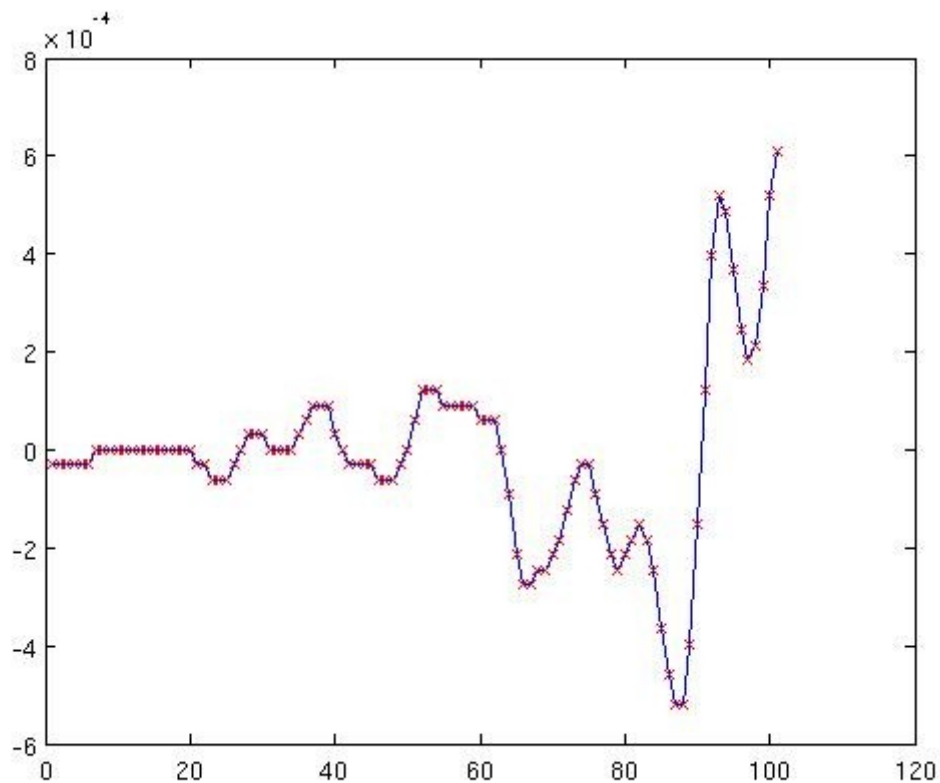
```
ans =
```

```
1.0e-04 *  
-0.3052  
-0.3052  
-0.3052  
-0.3052  
-0.3052  
-0.3052  
-0.3052  
-0.3052  
-0.6104  
-0.6104
```

On veut superposer à l'affichage standard le signal en croix rouge, comme le signal a un fréquence d'échantillonnage importante, on va juste afficher les 100 premières valeurs du signal à partir de la première non nulle pour y voir quelque chose

```
>> hold on  
>> X=find(signal,1);  
>> plot(signal(X:X+100))  
>> plot(signal(X:X+100),'rx')
```

hold on sert à superposer sur l'affichage standard sans réinitialiser la figure. hold off fera l'inverse. On obtient la figure suivante :



Les données sont échantillonnées sur 16 bits. Il faut donc multiplier les données par  $2^{15}$  (ce sont des entier signés) pour avoir des entiers.

```
>> signal*2^15;
```

## 2.Image

L'image est une matrice 595\*900\*3 de uint8

On affiche les valeurs des 100 premiers pixels de l'image dans les composantes rouge vert et bleu :

```
>> img(1:10,1:10,1:3)
```

```
ans(:,:,1) =
```

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

```
ans(:,:,2) =
```

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 19 | 19 |
| 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 18 | 18 |

```
ans(:,:,3) =
```

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 |
| 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 |
| 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 |
| 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 |
| 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 |
| 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 |
| 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 |
| 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 |
| 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 93 | 93 |
| 91 | 91 | 91 | 91 | 91 | 91 | 91 | 91 | 92 | 92 |

## II Quantification, échantillonnage

Pour sous quantifier un fichier audio sur  $q$  bits, il faut multiplier le signal par  $2^{q-1}$ , puis arrondir avec la fonction floor(partie entière) (il ne faut pas utiliser la fonction round(entier le plus près) ça se perçoit pour la quantification 4 bits on a une mauvaise qualité)

```
>> signal12b=floor(signal*2^11)/2^11;
>> wavwrite(signal12b,fs,q,'diner_12b');
>>
>> signal8b=floor(signal*2^7)/2^7;
>> wavwrite(signal8b,fs,q,'diner_8b');
>>
>> signal4b=floor(signal*2^3)/2^3;
>> wavwrite(signal4b,fs,q,'diner_4b')
```

On ne perçoit aucune différence entre la quantification sur 16 bits et celle sur 12. Sur celle sur 8 bits on perçoit du bruit, un léger grésillement, et sur celle sur 4, le bruit couvre en partie la voix des acteurs. On peut remarquer que même avec une quantification sur 1 bit, on comprend le dialogue.

Pour sous échantillonner le signal audio \*X, il faut garder une valeur sur X.

```
>> signalx2=signal(1:2:size(signal));
>> wavwrite(signalx2,fs./2,'diner_se*2');
>>
>>
>> signalx4=signal(1:4:size(signal));
>> wavwrite(signalx4,fs./4,'diner_se*4');
```

Lors de l'enregistrement du signal, il ne faut pas oublier de diviser la fréquence d'origine par X puisqu'on a X fois moins de valeurs.

En sous échantillonnant\*2 il est difficile de constater une différence avec le signal original. En sous échantillonnant par 4 on s'aperçoit que le son est beaucoup plus grave. En effet en diminuant la fréquence on supprime les hautes fréquences et on obtient plus que des basses fréquences. On peut sous échantillonner jusqu'à 16 fois mais il faut faire attention avec les valeurs des fréquences et préférer utiliser floor(fs/X) pour avoir une fréquence entière sinon le fichier peut devenir illisible. On ne peut pas descendre en dessous de 1000Hz. Fs étant égal à 22050Hz on peut descendre à une quantification \*22. Le son ainsi produit donne l'impression d'écouter



du fond d'une piscine.

```
>> signalx22=signal(1:22:size(signal));  
>> wavwrite(signalx22,floor(fs/22),'diner_se*22');  
..
```

On cherche à sous quantifier une image en ne gardant que nbits.

Notre image est par défaut une matrice d'uint8. Il nous faut la convertir en matrice double pour que les divisions se passent bien.

```
>>  
>> img=imread('photo.jpg');  
imgaux=double(img);  
  
img2=2*floor(imgaux./2);  
img2=uint8(img2);  
imwrite(img2,'img_sq2.jpg','jpg');  
  
img4=2^2*floor(imgaux./2^2);  
img4=uint8(img4);  
imwrite(img4,'img_sq4.jpg','jpg');  
  
img8=2^3*floor(imgaux./2^3);  
img8=uint8(img8);  
imwrite(img8,'img_sq8.jpg','jpg');  
  
img16=2^4*floor(imgaux./2^4);  
img16=uint8(img16);  
imwrite(img16,'img_sq16.jpg','jpg');  
  
img32=2^5*floor(imgaux./2^5);  
img32=uint8(img32);  
imwrite(img32,'img_sq32.jpg','jpg');  
  
img64=2^6*floor(imgaux./2^6);  
img64=uint8(img64);  
imwrite(img64,'img_sq64.jpg','jpg');  
  
img128=2^7*floor(imgaux./2^7);  
img128=uint8(img128);  
imwrite(img128,'img_sq128.jpg','jpg');
```

Ci-dessus dans l'ordre quantification sur 7,6,5,4,3,2 et 1 bits (les noms des variables sont trompeurs attention).

On peut vérifier qu'il n'y a pas d'erreur en faisant `length(unique(img_a_veref))`, qui compte le nombre d'éléments différents. Ainsi on peut vérifier qu'on a bien  $2^n$  valeurs différentes.

On obtient les images suivantes :



*Illustration 1: Image originale*



*Illustration 2: Quantification 4 bits*



*Illustration 3: Quantification 2 bits*



*Illustration 4: Quantification 1bit*

En diminuant le nombre de bits on assombrit au fur et à mesure l'image et on crée des régions homogènes au niveau de la texture de plus en plus grande.

Veysseire Daniel TP1 3DIS

Qu'est-ce que fmt ?

Script matlab ?

### **Contact**

Si il y a des erreurs, des remarques, des ajouts à faire, etc.

Veillez m'en faire part à cette adresse :

wedg@hotmail.fr