

## TP 02 : Comparaison entre TFD (DFT), TFR (FFT)

### Note

Chaque groupe doit rédiger un compte rendu. Pour se faire, chaque groupe doit copier toutes les figures dans un document Word. De plus, un compte rendu doit contenir tous les programmes, le calcul analytique, les commentaires, les démonstrations, les résultats, ainsi que les conclusions. Les comptes rendus doivent être dressés à l'enseignant lors de la prochaine séance de TP.

### 1. DFT

La TFD d'une séquence discrète  $s_1(n)$  est donnée par :

$$S_1(k) = \sum_{n=0}^{N_1-1} s_1(n) \exp\left(-j2\pi \frac{k}{N_1} n\right) \quad \text{pour } k = 0, 1, \dots, N_1$$

1. Créer une fonction « *discret2* » permettant le calcul de la TFD des séquences discrètes.

A. Le nombre d'échantillons dans le domaine temporel ( $N$ ) est égale au nombre d'échantillons dans le domaine spectral ( $N_I$ ).

- Représenter graphiquement l'amplitude de la TFD de la séquence  $s_1(n)$ .

$$s_1(n) = \frac{\pi}{2} \sin\left(2\pi \frac{n}{10}\right) \quad \text{pour } n = 0, 1, \dots, N-1, N=20,$$

- Répéter la procédure pour  $N=10, 20, 40, 50, 100$  et  $1000$

- Répéter la procédure pour les séquences suivantes :

- $s_2(n) = \frac{\pi}{2} \cos\left(\pi \frac{n}{10}\right) \quad n = 0, 1, \dots, N-1,$

- $s_3(n) = \text{rect}_N(n) \quad n = 0, 1, \dots, N-1,$

- $s_4(n) = \text{tri}_N(n), \quad n = 0, 1, \dots, N-1$

- Commentez vos résultats.

- Quelle est votre conclusion ?

B. Maintenant on prend le nombre d'échantillons dans le domaine temporel ( $N$ ) différent de celui du domaine spectral ( $N_I$ ).

- Représenter graphiquement la DFT de la séquence  $s_1(n)$  pour  $N=100$  et respectivement  $N_I=200, 300, 500, 1000, 5000$ .

- Répéter la procédure pour les séquences suivantes :

- $s_2(n) = \frac{\pi}{2} \cos\left(\pi \frac{n}{10}\right) \quad n = 0, 1, \dots, N-1,$

- $s_3(n) = \text{rect}_N(n) \quad n = 0, 1, \dots, N-1,$

- $s_4(n) = \text{tri}_N(n), \quad n = 0, 1, \dots, N-1$

- Commentez vos résultats.
- Quelle est votre conclusion ?

## 2. FFT

L'algorithme de la FFT a été proposé pour réduire le nombre d'opérations de multiplications et d'additions en exploitant la structure particulière des exponentielles dans le calcul de la DFT.

On considère toujours la séquence  $s_1(n)$  et on veut calculer la DFT de cette séquence en utilisant l'algorithme de la FFT.

- Utilisez la fonction *fft* pour calculer la DFT de cette séquence, pour  $N=10, 20, 40, 50, 100$  et  $1000$  et  $N = N_I$ .
- Répéter le calcul précédent pour respectivement  $N_I=2N_I, 4N_I, 6N_I$ .
- Répétez la procédure pour les séquences suivantes :

- $s_2(n) = \frac{\pi}{2} \cos\left(\pi \frac{n}{10}\right) \quad n = 0, 1, \dots, N-1,$

- $s_3(n) = \text{rect}_N(n) \quad n = 0, 1, \dots, N-1,$

- $s_4(n) = \text{tri}_N(n), \quad n = 0, 1, \dots, N-1$

- Commentez vos résultats.
- Quelle est votre conclusion ?
- Utiliser les commandes « *tic* » et « *toc* » pour faire une comparaison du temps de calcul de la **DFT** classique et la **FFT**.
- Calculer le rapport entre ces deux temps.

### **%Programme 01**

```
% Déclarer la valeur de N;
N=100;
% Déclarer le vecteur temps, n
n=0:N-1;
% Déclarer la valeur de N1;
N1=100;
% Déclarer la séquence s1(n)
s1=(pi/2)*sin(2*pi*n/10);
% Représenter graphiquement S1(k), la DFT de s1(n)
for k=0:N1-1
    psi=exp(j*2*pi*n*k/N1);
    S1(k+1)=sum(s1.*psi);
end
k=0:N1-1;
stem(k,abs(S));
xlabel('Fréquence discrète')
ylabel('Module de S1(k)')
```

### **%Programme 02**

```
% Déclarer la valeur de N;
N=10;
N1=N ;
% Déclarer le vecteur temps, n:
n=0:N-1;
% Déclarer la séquence s1(n)
s1=(pi/2)*sin(2*pi*n/10);
% Calculer S1(k), la FFT de s1(n)
S1=fft(s1,N1);
% Ici N1 représente le nombre de points de la FFT
% Déclarer le vecteur de la fréquence
k=0:N1-1;
% Représenter graphiquement le spectre d'amplitude
stem(k,abs(S1))
xlabel('Fréquence discrète')
ylabel('Module de S1(k)')
```

### **%Programme 03**

```
for l=1:100000
    tic
    N=10;
    N1=10;
    n=0:N-1;
    s1=(pi/2)*sin(2*pi*n/10);
for k=0:N1-1
    psi=exp(j*2*pi*n*k/N1);
    S(k+1)=sum(s1.*psi);
end
    t1(l)=toc;
end
t_DFT=mean(t1)
clear all
close all
for l=1:100000
    tic
    N=10;
    n=0:N-1;
    S=fft((pi/2)*sin(2*pi*n/10),20);
    t2(l)=toc;
end
t_FFT=mean(t2)
```