

TP n°1 : TFD et fenêtrage (en python)

Rappels

1. Transformée de Fourier Discrète

La TFD d'ordre N d'un signal numérique $s(kT_e)$, $k=0\dots N-1$ est définie par :

$$X(k) = X\left(\frac{k}{N}f_e\right) = \sum_{n=0}^{N-1} x(nT_e) e^{-2\pi jkn/N}, k = -N/2, \dots, N/2$$

Sa transformée inverse est donnée par:

$$x(n) = x(nT_e) = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} X(kf_e/N) e^{2\pi jkn/N}, n = 0, \dots, N-1$$

Python fournit dans **scipy.fftpack** les fonctions nécessaires pour calculer la transformée de Fourier complexe d'un vecteur de longueur N.

- La fonction **scipy.fftpack.fft(x)** ou **numpy.fft.fft(x)** permettra de calculer la TFD (respectivement la TFD⁻¹) sur les N points, soit de 0 à N-1 correspondant à l'intervalle $[0, f_e]$ (respectivement $[0, NT_e]$). Le spectre obtenu sera un vecteur complexe que l'on pourra représenter en coordonnées polaires (module et phase, fonction **numpy.abs** et **numpy.angle** ou **numpy.unwrap**).

- Pour une représentation entre $-f_e/2$ et $f_e/2$ de la TFD, il suffira d'écrire **X=numpy.fft.fftshift(X)**

- Pour augmenter la précision, il suffit d'écrire **numpy.fft.fft(x,NF)** où $NF > N$. La transformée inverse est donnée alors par **numpy.fft.ifft(x,NF)**.

Quelques astuces

1. Pour des figures séparées, aller Outils -> Préférences -> Console IPython -> Graphique -> Sortie: Automatique. Fermer spyder et ré-ouvrir.
2. Pour effacer toutes les variables avant exécution Préférences -> Exécuter -> Supprimer toutes les variables avant exécution
3. Ctl+L efface la console

2. Exemple d'utilisation de la FFT sous Spyder

```
import numpy as np
import matplotlib.pyplot as plt
N = 100; f0=1000; fe=8000; Te=1/fe
t = np.linspace(0.0, N*Te, N)
x = np.sin(2.0*np.pi*f0*t)
y = np.fft.fft(x)
y = np.fft.fftshift(y)
freq = np.arange(-N/2, N/2)*fe/N
plt.figure(1)
plt.subplot(311)
plt.plot(freq, np.abs(y))
plt.title('Module de la T.F.D.');
```

plt.grid(True); plt.xlabel('Fréquence (Hz)');

plt.ylabel('Amplitude')

```
plt.subplot(312)
plt.plot(freq, np.angle(y))
plt.title('Phase de la T.F.D.');
```

plt.grid(True); plt.xlabel('Fréquence (Hz)');

plt.ylabel('Amplitude')

```
plt.subplot(313)
plt.plot(freq, 20*np.log10(np.abs(y)/N))
```

```
plt.title('Module de la T.F.D. en db'); plt.grid(True);plt.xlabel('Fréquence (Hz)');plt.ylabel('Amplitude')
plt.show()
```

3. Programmes à réaliser

I. Reprendre l'exemple vu en cours $x(n) = [1 \ 0 \ 0 \ 1]$, calculer, afficher et visualiser sa TFD pour $NF=4$ puis $NF=32$ et commenter.

```
import numpy as np
import matplotlib.pyplot as plt
NF = 4; Te = 1.0; fe=1.0/Te;
x = np.array([1,0,0,1])
y = scipy.fftpack.fft(x,NF)
y = np.fft.fftshift(y)
freq = np.arange(-NF/2,NF/2)*fe/NF
plt.figure(2)
plt.subplot(211)
plt.stem(freq, np.abs(y),use_line_collection=True)
plt.title('Module de la T.F.D. '); plt.grid(True); plt.xlabel('Fréquence (Hz)');plt.ylabel('Amplitude')
plt.subplot(212)
plt.stem(freq, np.angle(y),use_line_collection=True)
plt.title('Phase de la T.F.D. '); plt.grid(True); plt.xlabel('Fréquence (Hz) ');
plt.ylabel('Amplitude')
plt.show()
```

II. TFD d'une porte (à compléter)

```
import numpy as np
import matplotlib.pyplot as plt
from math import pow
N=6; Te = 1.0; fe=1.0/Te; NF=64; f0=0.2
x=np.ones(N)
```

```
E=0.0
for i in x:
    E+=pow(i,2)
```

```
t = np.linspace(0.0, N*Te, N)
z = np.exp(2*np.pi*1j*f0*t)
```

- Expliquer par un dessin le rôle de `fftshift`.
- Calculer la TFD théorique et comparer avec celle obtenue par ce programme.
- Prendre $NF = 12$ puis 24 puis encore 128 et commenter.
- Vérifier que l'énergie se conserve $\sum_0^{N-1} |x(n)|^2 = \frac{1}{N} \sum_0^{N-1} |X(k)|^2$
- Illustrer la propriété de décalage fréquentiel de la TF en calculant (théoriquement) et en représentant le module de la TFD de $x(n) \times e^{2\pi j f_0 n}$ avec $f_0=0.2$ Hz.
- Reprendre le signal original et mesurer la largeur Δf du lobe principal à partir de sa TF. Sa taille est-elle fonction de N ou NF ?
- Afficher le module de la transformée de Fourier en décibels en écrivant puis vérifier que l'on a bien environ 13 dB entre le max du lobe principal et le max du lobe secondaire pour tout N (prendre $N=10$ et $N=40$).

III. TFD d'un signal illimité et Effet du fenêtrage (à compléter)

```
import numpy as np
import matplotlib.pyplot as plt

N = 50; f0=1680; fe=8000; Te=1/fe; NF=1024
t = np.linspace(0.0, N*Te, N)
x = np.exp(2*np.pi*1j*f0*t)
#help(scipy.signal.hamming)
window = sp.windows.hann(N);
```

- Déterminer la TFTD théorique de $s(n) = A_0 e^{2\pi j f_0 n}$
- Que se passe-t-il si l'on calcule cette TFTD sur N ?
- Pourquoi parle-t-on de fenêtre rectangulaire ? A-t-on vraiment multiplié par une fenêtre ?
- Que se passe-t-il si l'on remplace f_0 par 1600?
- On remet f_0 à 1680 et on teste différentes fenêtres
- Calculer la largeur du lobe principale pour chaque fenêtre en fonction de f_e et N.
- Comparer les fenêtres (amplitude principale, largeur des lobes, amplitudes des lobes).
- Tester d'autres fenêtres (Hamming, Blackman)
- Modifier ce programme pour calculer la TFD de la somme de 2 exponentielles de fréquences $f_0=1680$ Hz et $f_1=1800$ de même amplitude 1. Comparer alors les résolutions fréquentielles théoriques et pratiques des différentes fenêtres
- Refaire le même travail avec $f_0=1680$ Hz et $f_1=2100$ avec des amplitudes différentes ($A_0=1$ et $A_1=0.2$) et commenter.

Remarque Deux raies d'un spectre sont considérées comme séparables, si le maximum de l'une correspond au premier minimum nul de l'autre soit $|f_1 - f_2| > \text{Largeur du lobe principal} / 2 = L \Delta f / 2$ (Voir tableau page 31) .