25 Transformée de Fourier rapide (FFT)

I) Introduction

Dans le module numpy, on va utiliser `fft.rfftfreq` pour avoir les fréquences des raies calculées ;

Et utiliser `fft.rfft` pour pouvoir trouver les hauteurs des raies du spectre en amplitude à ces fréquences (il faudra penser à prendre la **valeur absolue** de `fft.rfft`).

- La première instruction (`fft.rfftfreq`) a pour syntaxe `fft.rfftfreq(Ne,Te)`
- La seconde instruction (`fft.rfft`) travaille sur le **tableau** des valeurs temporelles successives d'un signal

II) Idée de base

L'idée de base de la transformée de Fourier discrète est de transformer une liste de valeurs discrètes d'une fonction temporelle en une liste de valeurs discrètes de son spectre en amplitude.

Plus précisément, si le signal temporel est échantillonné aux instants $t_n = nT_e$, n variant de 0 à $(N_e - 1)T_e$, alors le spectre en amplitude est échantillonné aux fréquences $k\frac{f_e}{N_e}$ (c'est-à-dire $k\frac{1}{N_e T_e}$), k variant de 0 à $E\left(\frac{N_e}{2}\right)$.

En conséquence, une sinusoïde présente dans le signal ne donne une raie fine et de bonne hauteur que si sa fréquence correspond à $k\frac{f_e}{N_e}$, c'est-à-dire $k\frac{1}{N_e T_e}$

Attention, si on crée les instants des échantillons temporels avec `np.linspace(0,Ne*Te,Ne)`, le pas d'échantillonnage n'est pas T_e mais $T_e\frac{N_e}{N_e-1}$, et les fréquences du spectre sont $k\frac{N_e-1}{N_e^2 T_e}$, k variant de 0 à $E\left(\frac{N_e}{2}\right)$.

Pour que les instants des échantillons temporels soient bien $t_n = nT_e$, n variant de 0 à $(N_e - 1)T_e$, il faut les créer avec `np.linspace(0,(Ne-1)*Te,Ne)` , ou bien avec `np.array([n*Te for n in range(Ne)])`, ou bien éventuellement `np.linspace(0,Ne*Te,Ne, endpoint=False)`.

III) Prise en main des instructions `fft.rfftfreq` et `fft.rfft`

- Importer le module `numpy`, en tant que `np` et, pour pouvoir tracer des courbes, importer le module `matplotlib.pyplot` en tant que `plt` ou bien le module `pylab` complet.
- Choisir un nombre d'échantillons `Ne=10`, une fréquence d'échantillonnage `fe=1000`, une période d'échantillonnage `Te=1/fe`, une fréquence de signal `f1=200`.
- Créer un **tableau** `t_ech` d'instants, comprenant `Ne` valeurs équiréparties, de `0` à `(Ne-1)Te`.
- Définir une **fonction** s(t), constituée de la somme d'une constante (de valeur 1), et d'une sinusoïde de fréquence f1 et d'amplitude 1.
- Créer un **tableau** `s_ech` comprenant les valeurs de s(t) pour chacun des instants de `t_ech`.
- Avec `np.fft.rfftfreq`, créer (voir la syntaxe au tout début de cette feuille) un tableau `f` des valeurs de fréquences $\frac{k}{N_e T_e}$, k variant de 0 à $E\left(\frac{N_e}{2}\right)$.
- Avec `abs` et `np.fft.rfft`, créer un tableau `spec` des amplitudes des raies pour les fréquences $\frac{k}{N_e T_e}$, k variant de 0 à $E\left(\frac{N_e}{2}\right)$.
- Sur une première figure, tracer la fonction s(t), en utilisant les tableaux `t_ech` et `s_ech`.
- Sur une seconde figure, tracer le spectre en amplitude de s(t), en utilisant les tableaux `f` et `spec`, sans relier les points par un trait
  Rappel : pour n'afficher que les points, sous forme de carrés rouges, faire `plot(f,spec,'sr')`.
- Changer `Ne=10` en `Ne=20` et interpréter.

IV) **Normalisation**

1°) Hauteur de la raie de fréquence nulle :

- Quand on veut calculer la valeur moyenne (ou « raie de fréquence nulle ») d'un signal, sur une durée T_a, on fait $< s(t) > = \frac{1}{T_a}\int_0^{T_a} s(t)\, dt$.
- Et puisque $T_a = N_e T_e$, et qu'on fait du calcul numérique, donc approché, on obtient une bonne estimation de l'intégrale en faisant $\frac{1}{N_e T_e}\sum_{n=0}^{N_e-1} s(nT_e) \times T_e$, c'est-à-dire $\frac{1}{N_e}\sum_{n=0}^{N_e-1} s(nT_e)$.
- Mais pour la raie de fréquence nulle, l'instruction `fft.rfft` calcule uniquement la somme, sans la multiplier par $\frac{1}{N_e}$.
  Pour avoir la bonne raie à fréquence nulle, il faut donc la multiplier par (1/Ne)

2°) Hauteur des autres raies :

- On admet que, si un signal $s(t)$ périodique de période T comporte une sinusoïde d'amplitude a_k à la fréquence $f_k = \frac{k}{N_e T_e}$, on peut trouver cette valeur a_k en faisant $\frac{2}{T}\int_0^{T} s(t) \times \cos(2\pi f_k t)\ dt$. En effet, si $s(t)$ se décompose en une somme de sinusoïdes de différentes fréquences, pour toutes celles de fréquence autre que f_n, le calcul donne 0, et celle de fréquence f_n, le calcul revient à faire la valeur moyenne de $2a_n \cos^2(2\pi f_n t)$, ce qui donne bien a_n car la valeur moyenne d'un cosinus carré vaut $\frac{1}{2}$.
- Pour un signal acquis sur une durée T_a, on fait donc $\frac{2}{T_a}\int_0^{T_a} s(t) \times \cos(2\pi f_k t)\, dt$
- Et puisque $T_a = N_e T_e$, et qu'on fait du calcul numérique, donc approché, on estime l'intégrale en faisant $\frac{2}{N_e T_e}\sum_{n=0}^{N_e-1} s(nT_e) \times \cos\left(2\pi \frac{k}{N_e T_e} nT_e\right) \times T_e$, c'est-à-dire $\frac{2}{N_e}\sum_{n=0}^{N_e-1} s(nT_e) \times \cos\left(\frac{2\pi k n}{N_e}\right)$.
- Mais pour les raies autres que celles de fréquence nulle, l'instruction `fft.rfft` calcule uniquement la somme, sans la multiplier par $\frac{2}{N_e}$.
  Pour avoir la bonne raie à fréquence f_k, il faut donc la multiplier par (2/Ne).
- Créer un **tableau**, nommé `C`, contenant `len(f)` éléments, le premier de valeur `1/Ne`, tous les autres de valeur `2/Ne`.
- Dans le calcul du spectre `spec`, introduire `C` pour normaliser le spectre, c'est-à-dire pour que les raies soient de bonne hauteur.
- Tracer ce spectre pour `Ne=20`.
- Garder Ne=20 mais changer `f1=200` en `f1=220` et interpréter de deux façons différentes, l'une du point de vue fréquentiel, l'autre du point de vue temporel.

Suite de la séance : 3 programmes à écrire et à faire tourner :

1. Premier programme : un signal de type sinusoïde pure, de fréquence f1
    - Choisir un grand nombre d'échantillons, par exemple 1000, et une fréquence d'échantillonnage telle que le critère de Shannon-Nyquist soit largement respecté, et faire aussi en sorte que `Ta`, c'est-à-dire `Ne*Te` soit un multiple entier de la période de la sinusoïde.
    - Afficher le signal temporel dans une figure(1).
    - Afficher son spectre en amplitude dans une figure(2), en faisant en sorte que les raies calculées soient de bonne hauteur (cf plus haut, normalisation).
    - Choisir f1 ne correspondant pas à $k\frac{f_e}{N_e}$ (c'est-à-dire $k\frac{1}{N_e T_e}$) et observer la hauteur et la forme du spectre. Expliquer.

2. Second programme : un signal somme d'une composante continue et de 3 sinusoïdes à f1, f2, f3
    - Afficher son spectre en amplitude dans une figure(1), en faisant en sorte que les raies soient de bonne hauteur (cf plus haut), même pour la raie de fréquence nulle.
    - Vérifier qu'il y a repliement(s) si Shannon-Nyquist n'est pas respecté.

3. Troisième programme un signal en forme de créneau
    - Créer un créneau au moyen d'une sinusoïde et de l'instruction np.sign qui renvoie 1, 0, ou -1 selon que la variable sur laquelle elle travaille est positive, nulle ou négative.
    - Afficher le signal temporel dans une figure(1)
    - Afficher son spectre en amplitude dans une figure(2), en faisant en sorte que les raies soient de bonne hauteur (cf plus haut, normalisation), même pour la raie de fréquence nulle.
    - Vérifier qu'il y a repliements pour certaines raies.