TP d'informatique n°09
Filtrage à l'aide de numpy

PCSI 2020 – 2021

I À faire *avant* le TP

Le TP suivant porte sur le filtrage. **Avant de venir en TP**, il faut avoir compris le filtrage. Pour cela, réfléchissez à la question suivante : soit un signal d'entrée $e(t) = 3 + \cos(5t) + 2\cos(100t + 1{,}2)$ (on ne se préoccupe exceptionnellement pas d'unité et d'homogénéité) passant par un filtre de fonction de transfert

$$\underline{H}(\omega) = \frac{\underline{u_s}(\omega)}{\underline{u_e}(\omega)} = \frac{1}{1 + j\frac{\omega}{5}}$$

Donner l'expression du signal de sortie $s(t)$ (sous une forme similaire à celle de $e(t)$).

II Synthèse de Fourier

Faites les différents exercices de cette section dans **un même programme** pour pouvoir réutiliser les fonctions.
On utilisera les propriétés des tableaux numpy autant que possible (calcul d'un cosinus élément par élément par exemple).

Exercice n°1 Faire une fonction qui
- Définit un tableau correspondant au temps t allant de 0 à 12 et ayant 5000 points.
- Calcule la somme (sous forme d'un tableau ayant autant de point)

$$s = \cos(t) + \frac{1}{3^2}\cos(3 \times t) + \frac{1}{5^2}\cos(5 \times t)$$

  (s est en fait le tableau dont chaque élément vaut $s_i = \cos(t_i) + \frac{1}{3^2}\cos(3 \times t_i) + \frac{1}{5^2}\cos(5 \times t_i)$ pour chaque $t_i \in t$ correspondant)
- Trace le graphique correspondant

Exercice n°2 Améliorer la fonction précédente en une nouvelle fonction qui prend un argument N puis calcule et affiche cette fois

$$s = \sum_{k=0}^{N} \frac{1}{(2k+1)^2}\cos\left((2k+1)t\right)$$

Exercice n°3 Pour généraliser le calcul de l'exercice précédent, réaliser une fonction qui réalise la synthèse d'un signal connaissant son spectre. Plus précisément cette fonction prendra 4 arguments ;
- une liste d'amplitude $C = [c_1, c_2, c_3...]$
- une liste de phase $\Phi = [\varphi_1, \varphi_2, \varphi_3...]$ de même longueur que C
- une liste de fréquence $F = [f_1, f_2, f_3...]$ de même longueur que C
- une liste de temps $T = [t_1, t_2, t_3, ...]$ de longueur (a priori) différente de C

Les trois premières listes correspondent au spectre du signal dont on veut calculer la représentation temporelle (amplitude et phase en fonction de la fréquence) et la quatrième aux différents instants pour lesquels on veut renvoyer le signal.
Testez votre fonction sur un signal dont le spectre est, pour $n > 0$:

— $f_n = n$;
— $c_n = \frac{2}{n}$;
— $\varphi_n = -\frac{\pi}{2} + n\pi$;
— et prennez pour T le tableau suivant : `np.linspace(0,3,3000)`.

On rappelle que $s(t) = \sum_{n=1}^{+\infty} c_n \cos(2\pi f_n t + \varphi_n)$.

III Filtrage

On se propose ici de filtrer un signal.
On considère le signal obtenu à l'exercice 2, dont la décomposition fréquentielle donne (en fonction d'une fréquence f_0 en paramètre) : $f_0, 3f_0, 5f_0, \dots (2n+1)f_0$. Les amplitudes c_n et les phases correspondantes φ_n sont : $c_0 = 1, c_1 = 1/3^2, c_2 = 1/5^2, \dots, c_n = 1/(2n+1)^2$ et $0, 0, 0, \dots \varphi_n = 0$.

Exercice n°4 Créez trois listes ou tableaux numpy $freq$, $ampli$ et $phase$ contenant les fréquences, les amplitudes et les phases pour n allant de 0 à 10. On prendra $f_0 = 2$.
Conseil : créer préalablement un tableau contenant les nombres $0, 1, 2, 3, ...10$, en déduire les tableaux $freq$, $ampli$ et $phase$. Ainsi, si l'on veut faire varier le nombre de point, ce sera plus facile.

Exercice n°5 On considère un filtre passe-haut dont la fonction de transfert est $\underline{H} = \frac{j\frac{f}{f_c}}{1+j\frac{f}{f_c}}$ avec f_c la fréquence de coupure. On donne $|\underline{H}| = \frac{\frac{f}{f_c}}{\sqrt{1+\left(\frac{f}{f_c}\right)^2}}$ et $\arg \underline{H} = \frac{\pi}{2} - \arctan \frac{f}{f_c}$.

1. Créer une fonction module qui prend en argument un tableau numpy de fréquence et un réel correspondant à la fréquence de coupure et renvoie le tableau des modules de H correspondant.
2. Créer une fonction argument qui prend les mêmes arguments que ci-dessus et renvoie le tableau des arguments de H correspondant.

On rappelle que le filtrage agit de la façon suivante, pour un signal sinusoïdal de pulsation ω et d'amplitude c, alors le signal de sortie est sinusoïdal et de pulsation ω, mais son amplitude vaut $|\underline{H}(f)|c$ et il est déphasé par rapport au signal d'entré de $\arg\underline{H}$.
D'après le principe de superposition, pour un signal complexe, décomposé en somme de sinus, il suffit d'appliquer cette méthode à chaque harmonique.
Ainsi si le signal d'entrée est

$$e(t) = \sum_{n=1}^{+\infty} c_n \cos(2\pi f_n t + \varphi_n)$$

alors le signal de sortie est

$$s(t) = \sum_{n=1}^{+\infty} |\underline{H}(f_n)| c_n \cos\left(2\pi f_n t + \varphi_n + \arg(\underline{H}(f_n))\right)$$

3 Créez une fonction $filtre$ qui prend en argument la fréquence de coupure et trois tableaux $freq$, $ampli$ et $phase$ correspondant au signal $e(t)$ et renvoie deux tableaux contenant les amplitudes et les phases du signal filtré $s(t)$.

Exercice n°6 En vous inspirant de la première partie, reconstituez le signal filtré et affichez le dans le cas $f_c \gg f_0$, $f_c \ll f_0$ et $f_c = 1$. Le résultat est-il conforme à vos attente ?

IV Analyse de Fourier

Cette partie est optionnelle.

Exercice n°7 On se propose de réaliser l'analyse de Fourier d'un signal. C'est-à-dire tracer la courbe $c(f)$.
Créez tout d'abord un tableau de temps équi-échantillonné t allant de a à b avec N points (qui seront des paramètres que l'on fera varier par la suite).
Créez ensuite un signal $s(t)$ (pour commencer, on prendra $s(t) = \cos(t)$)
Il faudra ensuite utiliser successivement les fonctions

1. `sfft=np.fft.fft(s)` qui calcule la transformé de Fourier d'un tableau de nombre et renvoie un tableau de nombres (complexes)
2. `sfft=np.abs(sfft)` qui calcule le module (pour avoir des nombres réels)
3. pour avoir les fréquences (notre axe des abscisses), il faudra utiliser `f=np.fft.fftfreq(len(s),t[1]-t[0])`. (On lui donne le nombre de points et le pas de temps et il se débrouille).

Tracez la courbe avec f en abscisse et $sfft$ en ordonnée. On n'affichera que la moitié des points (`[0,len(f)//2]`) le résultat est-il conforme à votre intuition ? Faites varier les paramètres suivant : a, b, N, s (on pourra par exemple essayer de programmer soi-même une fonction créneau).
On pourra éventuellement tracer la courbe avec une échelle logarithmique en y en utilisant `semilogy` à la place de `plot`.

V À faire pour la prochaine fois

Exercice n°8 Rappeler le principe de la méthode des rectangles. Un schéma et une formule sont nécessaires (mais non suffisant).

Exercice n°9 Écrire le programme informatique permettant de calculer par la méthode des rectangles l'intégrale suivante :

$$\int_0^5 2^{-x^2}\,\mathrm{d}x$$

Exercice n°10 Rappeler le principe de la méthode des trapèzes. Un schéma est nécessaire

Exercice n°11 Écrire le programme informatique permettant de calculer la même intégrale par la méthode des trapèzes et comparer votre résultat au précédent.

Exercice n°12 Calculer la même intégrale en utilisant la fonction adaptée d'une bibliothèque scientifique.