

Physique - Modélisation

I) Analyse spectrale numérique

a) Principe de l'algorithme de FFT sous numpy

On utilise la biblio. FFT $\rightarrow$ TF rapide et discrète

* nb de échantillons N et T_e la période d'acquisition

* $\text{fft_signal} = \text{calculer la TF du signal obtenu avec l'instruct}^{\circ}$
 $\text{fft}(\text{signal})$
 tableau

Paramètre d'entrée $\rightarrow$ tableau des valeurs du signal numérisé

* freq_signal sera les freq du spectre (ou modes) obtenus avec l'instruction $\text{fft_freq}(N, T_e)$

Paramètre d'entrée $\rightarrow N$
 $\rightarrow T_e$

Elle renvoie un tableau des freq de l'enregistrement
 $[0, F_0/N, \frac{2F_0}{N}, \dots, F_{\frac{N}{2}}, -F_{\frac{N}{2}}, \dots, -F_0/N, \dots, -F_0/2]$

Trace du spectre

$\rightarrow$ abscisses $\rightarrow \text{freq_signal} \rightarrow \text{freq du spectre}$

$\rightarrow$ ordonnées $\rightarrow \text{fft_signal} \rightarrow \text{spectre}$
 (hauteurs des raies)

$\hookrightarrow \text{plt.plot}(\text{freq_signal}, \text{fft_signal})$

Remarque : • il est très souvent utile de se débarrasser de la freq nulle composante de freq
 $\text{freq} \rightarrow \text{plt.plot}(\text{tableau}[1:], \text{tableau}[1:])$

• on peut tracer le module de la FFT $\text{abs}(\text{fft})$

b) Utilisation d'un signal échantillonné ou construct^o d'un signal numérique par ex

$\rightarrow$ Utilisation du signal échantillonné :

On récupère un tableau de valeurs numériques, on trouve N : $N = \text{len}(\text{signal})$
 tableau

On applique directement l'algorithme en connaissant T_e

$\rightarrow$ Ex de la construct^o d'un signal numérique :

Pour simuler les échantillons numériques, construire un signal sinusoïdal de période $T = 1 \text{ ms}$ avec $T_e = T/16 \rightarrow$ respect du critère de Shannon et $N = 512$.
 amplitude = 1

Écrire la routine et représenter le spectre.

construction du signal ② numérisé

$T = 10^{-3}$

$T_e = T/16$ $N = 512$

$\rightarrow$ liste des échantillons $\rightarrow \text{tab}$

$\text{t_e} = \text{linspace}(0, N T_e, N)$

$\rightarrow$ simulation/calcul du signal

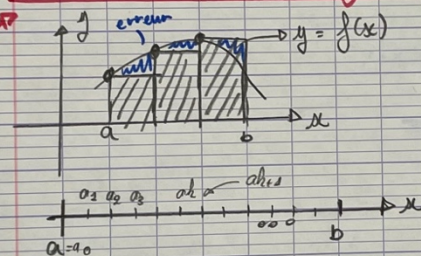
def signal(t_e):

return $\text{np.cos}(2 \times \text{np.pi} \times t_e / T)$

représentat° du spectre
 $\text{freq} = \text{fft_freq}(N, T_s)$
 $\text{spectre} = \text{fft}(\text{signal}(t_s)) \times 2/N$ permet de normaliser
 $\text{plt.plot}(\text{freq}, \text{spectre}, \text{label}='spectre')$ le spectre dont
 l'amplitude ne dépend pas de N

Remarque: On représentera en général de $[0; \pi/2]$
 $\hookrightarrow$ axis $([0, \max(\text{freq})])$

II) Méthode d'intégration numérique = méthode des rectangles



$$\text{But} \rightarrow I = \int_a^b f(x) dx$$

$f \in C^0$

On divise l'intervalle de calcul $[a, b]$
 $a_k = a + k \frac{b-a}{n} \quad k \in [0, n]$
 avec $\begin{cases} a_0 = a \\ a_n = b \end{cases}$

L'intégrale (= surface sous la courbe) est approchée par la somme des rectangles!

$$I = \int_a^b f(x) dx = \sum_{k=0}^{n-1} \int_{a_k}^{a_{k+1}} f(x) dx$$

$$\int_{a_k}^{a_{k+1}} f(x) dx \approx f(a_k) \times \frac{a_{k+1} - a_k}{1} = f(a_k) \times \frac{b-a}{n}$$

$$I = \sum_{k=0}^{n-1} f(a_k) \times \frac{b-a}{n} = \underbrace{\left(\frac{b-a}{n} \right)}_{\text{longueur d'un rect. unitaire}} \underbrace{\sum_{k=0}^{n-1} f(a_k)}_{\text{suite des valeurs de } f \text{ de } [a, b]} = I$$

longueur d'un rect. unitaire suite des valeurs de f de $[a, b]$

2) Comment implémenter en python

Ex à partir des valeurs de f de $[a, b]$ et de N nbr de segments:

def rectangle(f, a, b, n):

 somme = 0

 for k in range(0, n):

 somme += f(a_k)

 return (b-a)/n * somme