TP 3 – Série de Fourier, transformée de Fourier

S.Gibet

Année 2019-2020

Objectifs : Comprendre les bases des séries de Fourier - Comprendre la transformée de Fourier sur des exemples de synthèse puis sur des exemples réels.

Vous ferez un compte rendu de ce TP (intégré dans le programme Python), en prenant soin d'écrire pour chaque programme et chaque fonction les cartouches et les commentaires :

1 Série de Fourier

Cet exercice a déjà été traité en cours. L'idée est de revoir la théorie sur les séries de Fourier et de comprendre par la pratique le programme qui en découle (*serieFourier.py*). On considère le signal périodique carré vu en cours, nommé s, et défini par :

$$\left\{ \begin{array}{rll} s(x) = -1 & si & -\pi < x < 0 \\ s(x) = 1 & si & 0 < x < \pi \end{array} \right. \tag{1}$$

1. Quelle est la période T_0 de ce signal ? Que valent la pulsation ω_0 et la fréquence f_0 ?
2. On échantillonne ce signal. Comment doit-on choisir la fréquence d'échantillonnage ? Tracez cette fonction sur une période (de 0 à $2.\pi$), en choisissant une fréquence d'échantillonnage en fonction du nombre d'échantillons choisi N par période T_0.
3. Réécrire s à partir de la décomposition de s en série de Fourier (somme infinie de sinus et cosinus). Quelles conditions doit respecter s ?
4. Comment s'écrit cette décomposition précédente si l'on tronque la somme à un certain ordre (somme finie de termes) ?
5. Donnez les équations générales qui permettent de calculer les coefficients de Fourier a_k et b_k
6. Donnez les résultats des calculs de a_0, a_k et b_k pour le signal s.
7. Donnez la suite d'instructions en python qui permettent de calculer un terme de la série de Fourier.
8. Ecrivez le programme *serieFourier*. Vous effectuerez le calcul du *terme* b_k suivant que k est pair ou impair.
9. Testez la fonction avec plusieurs paramètres (nombre de points N et nombre de termes en sinus *ordre*) ; visualisez le résultat. A partir de quel ordre retrouve-t-on le signal carré ?

10. Ecrire un programme qui permet d'afficher de manière superposée les décompositions avec ordre = 1, 2, ..., N

11. Qu'observez-vous ?

2 Transformée de Fourier sur des signaux de synthèse

Le spectre d'un signal échantillonné peut être obtenu en utilisant la transformée de Fourier discrète (TFD). L'implémentation classique de la TFD est la FFT (Fast Fourier Transform). Quelque soit l'implémentation de la FFT, nous avons la relation suivante :

$$X[k] = FFT\{x[n]\} \tag{2}$$

La variable indépendante n est l'indice temporel, tandis que la variable indépendante k fait référence à l'indice fréquentiel. La FFT est évaluée sur un vecteur de N échantillons temporels et produit en sortie un vecteur de N échantillons fréquentiels. Si $x[n]$ contient une composante spectrale pour la fréquence f, alors, l'indice fréquentiel correspondant k vérifie la relation :

$$\frac{k}{N} = \frac{f}{f_s} \tag{3}$$

où f_s est la fréquence d'échantillonnage.

2.1 Transformée de Fourier discrète sur des signaux de synthèse

Dans les questions suivantes vous utiliserez les bibliothèques suivantes :

```
import numpy as np
from scipy.fftpack import fft
import matplotlib.pyplot as plt
```

1. Créez un signal sinusoïdal s de 16 Hz et vérifiez son allure en le traçant avec matplotlib. Vous prendrez pour fréquence d'échantillonnage 256 Hz, et vous construirez un vecteur de 256 échantillons (vous pourrez utiliser la fonction *genSin* du TP2 pour générer le signal sinusoïdal).

2. On souhaite changer d'espace de représentation en utilisant la transformée de Fourier du signal s.

3. Calculez le spectre du signal précédant en utilisant la fonction *fft*. Vous remarquerez que le vecteur TFD généré est un vecteur de complexes. Il est possible d'afficher l'amplitude, la phase, la partie réelle et imaginaire de chacun des éléments de ce vecteur (voir le code qui suit).

```
TFD = fft(echantillons)            #spectre du signal
A = np.abs(TFD/N)                 #amplitude du spectre
An = A/A.max()                      # amplitude normalisée
P = np.angle(TFD/N)             # phase normalisée
Re = np.real(TFD)                 # partie reelle de TFD
Im = np.imag(TFD)                # partie imaginaire TFD
```

4. Affichez graphiquement ce spectre (module du spectre). Pour l'abscisse, vous prendrez une échelle en Hz plutôt qu'une échelle adimensionnelle (de 0 à 511). Vérifiez le tracé du spectre : une raie doit être visible à la fréquence k=16. Qu'y a-t-il d'autre dans le spectre ? Expliquez.

```
F =np.linspace(0,fs,N)                # creation du vecteur des fréquences
plot(F, abs(TFD))                        # Affichage du spectre
```

5. L'exercice précédent considère une fréquence d'échantillonnage qui est un multiple du nombre d'échantillons ($N = f_s$). Reprenez les questions précédentes avec pour fréquence d'échantillonnage 200 Hz. Qu'observez-vous ?

6. Créez un signal numérique sinusoïdal de 128 Hz et vérifier son allure en le traçant. Vous prendrez pour fréquence d'échantillonnage 200 Hz, et construirez un vecteur de 256 échantillons. Quel phénomène observez vous ? Expliquez.

7. En choisissant soigneusement la fréquence d'échantillonnage, créez le signal numérique suivant :
$$s(t) = 3.cos(50.\pi.t) + 10.sin(300.\pi.t) - sin(100.\pi.t) \tag{4}$$

8. Tracez son spectre en amplitude et en phase. Qu'observez-vous ?