Transformée de Fourier

nov. 26, 2019

1 Transformée de Fourier

1.1 Transformée de Fourier continue

— Transformée de Fourier continue

$$\tilde{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}\mathrm{d}t$$

— Transformée de Fourier inverse

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \tilde{f}(\omega)e^{+i\omega t}\mathrm{d}\omega$$

1.2 Série de Fourier

Elle sont introduite dans le cas d'un signal périodique (T). Seules les harmoniques de la fréquence fondamentale sont présentes :

$$f(t) = \sum_{n=-\infty}^{+\infty} c_n e^{i2\pi\frac{n}{T}t}$$

avec

$$c_n = \frac{1}{T}\int_0^T f(t)e^{-i2\pi\frac{n}{T}t}dt$$

Les formules donnants $f(x)$ et les coeficients c_n ont été écrites pour une fonction périodique. Cependant elles restent toujours valable pour toute fonction si on se restreint à des valeurs de x entre 0 et T. Si une fonction est définie sur un interval $[0, T]$, alors cette fonction peut être décrite par une nombre infini mais discret de coeficients c_n correspondant aux fréquences n/T.

La TF étant sa propre réciproque, on peut en déduire que si une fonction est définie pour une ensemble discret de temps (séparé par Δt), alors elle aussi décrite par des fréquence comprise entre 0 et $\frac{1}{\Delta t}$ (ou entre $-\frac{1}{2\Delta t}$ et $\frac{1}{2\Delta t}$).

1.3 Transformée de Fourier Discrète

Dans le cas d'un signal défini sur une ensemble discret de temps (échantillonage) et sur une durée finie, on définit la transformée de Fourrier de la façon suivante :

$$\tilde{x}_k = \sum_{i=0}^{N} x_j e^{2i\pi\frac{kj}{N}}$$

Cette formule donne ce qui est calculé par les libraires de transformée de Fourrier.

On a le lien suivant avec la TF continue :

$$\tilde{x}_k = \sum x_j e^{2i\pi(j\Delta t)\frac{k}{N\Delta t}} = \frac{1}{\Delta t}\sum f(t_j)e^{2i\pi t_j\frac{k}{T}}\Delta t = \frac{1}{\Delta t}\tilde{f}\left(\frac{k}{T}\right)$$

Cette formule nous permet d'utiliser correctement les librairies, à la fois pour l'échelle horizontale et verticale.

La transformée de Fourier discrète (Discret Fourier Transform - DFT) est souvent calculée en utilisant l'algorithme de FFT (Fast Fourier Transform). L'usage confond les deux acronymes et utilise FFT pour désigné la DFT.

1.4 Convolution

La convolution de deux fonctions est un simple produit dans l'espace de Fourier. Cette propriété reste valable pour la DFT en utilisant la convolution circulaire (les indices sont prix modulo N :

$$(x * y)_n = \sum_l x_l y_{n-l}$$

On alors :

$$\widetilde{x * y} = \tilde{x}\tilde{y}$$

Notons que le calcul de la convolution de deux tableaux en utilisant la FFT sera beaucoup plus rapide car l'algorithme de FFT en en $N \log N$ alors que la formule de convolution est en N^2

1.5 Quelques pièges

— Repliement du spectre : si on regarde le signal $f(t_j) = e^{2i\pi f_0 t_j}$ avec $t_j = j\Delta t$, alors ce signal est le même si on ajoute (ou retranche) la fréquence $1/\Delta t$ à f_0. Conséquence : un signal de fréquence élevée apparaîtra comme un signal de plus basse fréquence si il est sous échantillonné ($2\Delta t > 1/f_0$). dans la formule donnant $\tilde{x}_k$, la fréquence pour k proche de N correspond en fait à $(k-N)/\Delta t$.

— Dans le cas d'une fonction continue enregistrée uniquement sur une durée T, les coefficients c_n permettent de recalculer cette fonction uniquement pendant la durée d'enregistrement. Les coefficients c_n coïncident avec la TF uniquement si f est périodique de période T. Ce n'est pas le cas si f a une période différente de T. En particulier un signal sinusoïdal ne produira pas une fonction δ et apparaîtra comme ayant des composantes pour toutes les fréquences.

— Toujours dans le cas d'une fonction continue enregistrée sur une durée T, les coefficients c_n correspondent à la transformée de Fourier de la fonction multipliée par une fenetre temporelle $w(t)$, avec $w(t)$ nulle sauf pour $t \in [0, T]$ où elle vaut 1. Si au lieu de choisir une fenêtre carré, on choisit une fenêtre plus lisse (par exemple un triangle, un arc de cosinus, ...) alors les différentes composantes d'un signal sinusoïdal seront regroupées autour du pic.

2 Exercices

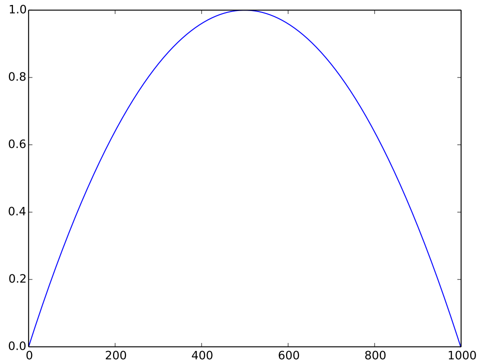
2.1 Transformée de Fourier simple

Tracer la transformée de Fourier de $1/(1 + 0.99\cos(2\pi t))$, pour des fréquences inférieures à 50. Pour cela on utilisera la fonction *fft* du module *numpy.fft*.

— Quel taux d'échantillonnage faut-il utiliser ?
— Sur quelle durée T suffit-il de calculer f ?
— Tracer la TF ?
— Que se passe-t-il si on prend une durée 10 fois plus longue que la période ?

2.2 Introduction aux fenêtres

On considère un signal sinusoïdal de 250 Hz, enregistré pendant 1 s et ayant 1000 échantillons.
— Tracer la valeur absolue de la transformée de Fourier de ce signal entre 200 et 300 Hz
— Sur le même graphique, tracer la TF d'un signal de même amplitude et de fréquence proche mais dont la période est différente de la durée de mesure (par exemple 250.112334245 Hz).
— Que se passe-t-il ?
— Mêmes questions en utilisant une fenêtre parabolique (figure ci-dessous)



2.3 Filtres en Python

Il est possible de simplifier le calcul de la transformée de Fourier lorsque le signal est réél (puisque $\tilde{x}(-f) = \tilde{x}(f)^*$). On utilise alors les fonctions `rfft` et `irfft` pour effectuer la FFT. La fonction `rfftfreq` va donner les fréquences.

On crée le signal suivant

```
samplerate = 44100
signal = zeros(samplerate*3)
signal[samplerate:samplerate*2)] = 1
plot(signal)
```

1. Vérifier que l'on obtient le même signal après avoir appliqué `rfft` et `irfft`

```
import numpy as np
signal_tilde = np.fft.rfft(signal)
signal_2 = np.fft.irfft(signal_tilde)
plt.plot(signal_2)
```

2. Utiliser la fonction `rfftfreq`. Est-ce que ce qu'elle renvoie vous semble cohérent ?
3. On veut appliquer un filtre passe bas de fréquence de coupure $f_c = 3$ Hz

$$H(f) = \frac{1}{1 + j(f/f_c)}$$

Pour écrire la fonction passe_bas suivante

```python
def passe_bas(signal, f_c, samplerate=44100):
    """ Filtre passe base d'ordre 1

    Filtre passe bas implémenté en utilisant la DFT.

    Paramètres :
        signal : numpy array représentant le signal d'entrée
        f_c : fréquence de coupure
        samplerate : taux d'échantillonnage.

    Résultats :
        out_signal : le signal filtré
    """
```

Tracer le signal filtré.

4. Faire de même avec un filtre passe haut en définissant la fonction `passe_haut` :

$$H(f) = \frac{j(f/f_c)}{1 + j(f/f_c)}$$

2.4 Machine à laver

Nous avons enregistré le bruit d'une machine à laver le linge pendant la phase d'essorage. Il s'agit du fichier `machine_a_laver.wav`. On entend un son aigu correspondant au moteur. Le tambour tourne beaucoup plus lentement.

1. Quel est la fréquence typique de rotation du tambour d'une machine à laver le linge ?
2. On va lire les fichiers `wav` en utilisant la fonction `read` de la librairie *scipy.io.wavfile*. On enregistrera l'amplitude du son dans la variable *amplitude*

```python
from scipy.io.wavfile import read
samplerate, amplitude = read('data/machine_a_laver.wav')
```

Quel est le nombre N de points ? Que vaut Δt ? Quelle est la durée T de l'enregistrement ? Tracer l'amplitude du son pendant la dernière seconde.

3. Filtrer le son de la machine à laver avec un filtre passe bas de fréquence 30 Hz. Enregistrer le son dans un fichier wav pour l'écouter ensuite !
4. Quel est la fréquence de rotation du tambour ?