TP1 Traitement du signal

Ibrahim ALAME

18/03/2024

La transformation de Fourier discrète (TFD), outil mathématique, sert à traiter un signal numérique. Elle constitue un équivalent discret de la transformation de Fourier (continue) utilisée pour traiter un signal analogique :

Soit (s_n) un signal discret périodique de période N. Sa transformation (TFD) est un signal discret périodique de même période N défini par :

$$S_p = \frac{1}{N}\sum_{n=0}^{N-1} s_n \mathrm{e}^{-2\mathrm{i}\pi p \frac{n}{N}} \qquad \text{pour} \qquad p \in \mathbb{Z}$$

Sa transformation inverse est donnée par

$$s(n) = \sum_{p=0}^{N-1} S_p \mathrm{e}^{2\mathrm{i}\pi n \frac{p}{N}} \qquad \text{pour} \qquad n \in \mathbb{Z}$$

La transformation de Fourier rapide **fft** est un algorithme particulier de calcul de la transformation de Fourier discrète :

$$\mathbf{fft}(s)[p] = \sum_{n=0}^{N-1} s_n \mathrm{e}^{-2\mathrm{i}\pi p \frac{n}{N}} \qquad \text{pour} \qquad 0 \leq p < N$$

Sa transformation inverse **ifft** est donnée par :

$$\mathbf{ifft}(S)[n] = \frac{1}{N}\sum_{p=0}^{N-1} S_p \mathrm{e}^{2\mathrm{i}\pi n \frac{p}{N}} \qquad \text{pour} \qquad 0 \leq n < N$$

Nous avons alors

$$S = \frac{1}{N} \times \mathbf{fft}(s) \qquad \text{et} \qquad s = N \times \mathbf{ifft}(S)$$

ou bien

$$N \times S = \mathbf{fft}(s) \qquad \text{et} \qquad s = \mathbf{ifft}(N \times S)$$

En python :

— La fonction `fft(signal)` du paquetage `numpy.fft` renvoie à un coefficient de normalisation $\frac{1}{N}$ près, la transformée de Fourier discrète d'un signal à une dimension éventuellement complexe. La fonction abs de `numpy` renvoie le module d'un tableau de complexes. Sa réciproque à un coefficient multiplicatif N près est la fonction `ifft(signal)` du même paquetage `numpy.fft`.

— La fonction `fftfreq(N, d)`de la bibliothèque `numpy.fft` renvoie un tableau contenant les fréquences associées à la transformée de Fourier pour un signal contenant N échantillons espacés d'un intervalle d(d est la période d'échantillonnage noté T_e).
— La fonction `fftshift(X)` de `numpy.fft` associe à la restriction $X_{/[\![0,N-1]\!]}$ d'un signal N-périodique, la restriction $X_{/[\![-\frac{N}{2},\frac{N}{2}-1]\!]}$ du même signal X.

Transformation de Fourier discrète (TFD)

Soit $x=(x_n)$ un signal 4-périodique qui coïncide sur l'intervalle $[\![0,3]\!]$ avec la liste $[1,1,0,1]$.

1. Tracer x sur $[\![-10,10]\!]$.
2. Calculer analytiquement puis avec python sa transformée de Fourier discrète X.
3. Calculer par la formule de la définition puis par python $\mathscr{F}^{-1}(X)$.
4. Tracer X sur un intervalle de fréquence convenable.
5. Soit $y=(y_n)$ un signal défini par $y_n=x_{n-1}$ pour tout $n\in\mathbb{Z}$. Calculer par trois méthodes le spectre $Y=\mathscr{F}(y)$ transformation de Fourier de y.
6. Tracer sur deux graphiques y et Y.

Transformation de Fourier

1. *Calcul analytique :* Soit x le signal continu défini par $x(t)=e^{-|t|}$. Montrer que la transformation de Fourier de x s'écrit pour $f\in\mathbb{R}$:

$$\hat{x}(f)=\frac{2}{1+4\pi^2f^2}$$

2. *Approximation par FFT de python :* On approche le domaine d'intégration $]-\infty,+\infty[$ par l'intervalle $[-\frac{A}{2},+\frac{A}{2}]$, où A est un réel positif assez grand :

$$\hat{x}(f)=\int_{-\infty}^{\infty}x(t)e^{-2i\pi ft}\,\mathrm{d}t\simeq\int_{-\frac{A}{2}}^{\frac{A}{2}}x(t)e^{-2i\pi ft}\,\mathrm{d}t$$

   (a) On discrétise l'intervalle $[-\frac{A}{2},+\frac{A}{2}]$ en $N+1$ points $(t_n)_{n=-\frac{N}{2},\frac{N}{2}}$ régulièrement espacés avec un pas $T_e=\frac{A}{N}$, on a donc $t_n=nT_e$. On pose $x_n=x(t_n)$, montrer en utilisant une méthode d'intégration approchée que l'on a :

$$\hat{x}(f)\simeq T_e\sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1}x_ne^{-2i\pi fnT_e}$$

   (b) On pose $f_k=k\frac{1}{NT_e}$ et $\hat{x}_k=\hat{x}(f_k)$. Montrer que

$$\hat{x}_k=\frac{A}{N}\sum_{n=0}^{N-1}x_ne^{-2i\pi\frac{kn}{N}},\quad k=0,...,N-1$$

   Donc $\hat{x}_k\simeq A\times X_k$. La FFT décrit alors, à une constante près, la TF de $x(t)$, notée $\hat{x}_k=\hat{x}(f_k)$ pour f_k entre $-\frac{1}{2T_e}$ et $\frac{1}{2T_e}$ par pas de $F_e=\frac{1}{NT_e}$ où N est le nombre de sous intervalles subdivisant $[-\frac{A}{2},+\frac{A}{2}]$.

(c) La fonction `numpy.fft.fftfreq` renvoie les fréquences du signal calculé dans la FFT. La liste `f` renvoyé contient les fréquences discrètes en Hz. Si le signal contient N pas de temps et que le pas de temps vaut Te :
— si n est pair, $f = [0, 1, \ldots, \frac{n}{2} - 1, -\frac{n}{2}, \ldots, -1] \times \frac{1}{N\,Te}$
— si n est impair, $f = [0, 1, \ldots, \frac{n-1}{2}, -\frac{n-1}{2}, \ldots, -1] \times \frac{1}{N\,Te}$
Justifier l'algorithme suivant :

```
t=np.arange(-A,A,Te)
x=s(t)
xtilde = np.fft.fftshift(x)
Xtilde = np.fft.fft(xtilde)/N
ftilde = np.fft.fftfreq(N,Te)
plt.plot(ftilde,Xtilde)
```

(d) Soit X le spectre de x centré à l'origine (déshifté), et f l'intervalle des fréquences déshifté. tracer X en fonction de f et comparer avec le graphe précédent :

```
X = np.fft.ifftshift(Xtilde)
f = np.fft.ifftshift(ftilde)
plt.plot(f,X)
```

On prendra $A = 5$ et $N = 64$.

3. *Décalage fréquentiel :* Illustrer la propriété de décalage fréquentiel de la TF en représentant le module de la TF de $x(t) \times e^{i2\pi f_0 t}$. En déduire que la transformation de Fourier de $s : t \mapsto e^{-|t|}\cos(2\pi t)$ est donnée par :

$$\hat{s} : f \mapsto \frac{1}{1 + 4\pi^2(f-1)^2} + \frac{1}{1 + 4\pi^2(f+1)^2}$$

Tracer $\hat{x}$ et $\hat{s}$ sur l'intervalle $[-3, 3]$. Comparer graphiquement avec la transformée de Fourier rapide obtenue à l'aide de `fft`.

Signal périodique

Considérer le signal x suivant :

$$t \mapsto x(t) = 2\cos(4\pi t) + 3\cos(8\pi t)$$

1. Quelle est sa période que l'on note T ?
2. Écrire le signal $x(t)$ sous forme exponentielle (formule d'Euler !). En déduire les coefficients de Fourier de $x(t)$.
3. On définit $\tilde{x} : n \mapsto \tilde{x}[n] = x(t_n)$ le signal en temps discret résultant de l'échantillonnage de x à la fréquence $f_e = 8$ Hz. Quel est le pas de discrétisation correspondant à une telle fréquence ?
4. Calculer le nombre d'échantillons N sur une période en fonction de T et f_e.
5. Dessiner sur le même graphique la fonction $x(t)$ sur l'intervalle $[0, T]$ et le signal précédent $\tilde{x}[n]$ tronqué à N échantillons correspondant aux temps t_n, $n = 0, ..., N-1$. On utilisera les deux fonctions python : `plot` et `stem` de la bibliothèque `matplotlib.pyplot`.

6. Calculer et tracer la transformée de Fourier discrète (TFD) du signal évalué aux temps t_n, $n = 0, ..., N-1$. Comparer avec le spectre de la question 2.
7. Refaire les questions 4 et 5 pour $fe = 10, 12, 20, ...$