

---

**Travaux pratiques n° 2**

INITIATION À LA TRANSFORMÉE DE FOURIER DISCRÈTE

---

**Consigne :** Au plus tard le **2 avril 2024**, les étudiants doivent envoyer leur fichier `Jupyter` (extension `.ipynb`) à

- `luis-miguel.rodrigues@univ-rennes.fr` pour les groupes Beaulieu ;
- `Adrien.Laurent@inria.fr` pour le groupe Magistère.

Les questions portant un astérisque \* ne sont pas utiles à la programmation et peuvent donc être sautées dans un premier temps.

---

On se propose de découvrir par les exemples la transformée de Fourier discrète comme outil de discrétisation des séries de Fourier et de leurs inverses.

Pour tout  $N \in \mathbf{N}$ , on introduira  $x^{(N)} := (x_j^{(N)})_{j \in [0, N]}$  où, pour  $j \in [0, N]$ ,  $x_j^{(N)} := j/(N + 1)$ .

**Exercice 1.**

1. Écrire un programme qui calcule le vecteur  $y^{(N)} := (f(x_j^{(N)}))_{j \in [0, N]}$  quand  $N = 100$  et

$$f : \mathbf{R} \rightarrow \mathbf{C}, \quad x \mapsto \sum_{k=-10}^{20} (20k + k^2) e^{2i\pi k x}.$$

2. On note  $z^{(N)}$  le vecteur obtenu en divisant par  $(N + 1)$  le résultat de la commande `numpy.fft.fft` appliquée à  $y^{(N)}$ , et  $u^{(N)}$  le résultat de la commande `numpy.fft.fftshift` appliquée à  $z^{(N)}$ . Tracer sur une même figure les parties réelles de  $z^{(N)}$  et  $u^{(N)}$  en fonction des entiers de  $[-50, 50]$ . On veillera à indiquer en légende quelle courbe est avec `shift`.
3. On note  $v^{(N)}$  le vecteur obtenu en multipliant par  $(N + 1)$  le résultat de la commande `numpy.fft.ifft` appliquée à  $z^{(N)}$ . Tracer sur une même figure  $y^{(N)}$  et la partie réelle de  $v^{(N)}$  en fonction de  $x^{(N)}$ . On veillera à tracer un des graphes en pointillés.

**Exercice 2.**

1. Écrire un programme qui calcule le vecteur  $y^{(N)} := (f(x_j^{(N)}))_{j \in [0, N]}$  quand  $N = 1000$  et

$$f : \mathbf{R} \rightarrow \mathbf{C}, \quad x \mapsto \frac{1}{10} + e^{-200(x - \frac{1}{2})^2} \cos(80\pi x).$$

2. On note  $u^{(N)}$  le vecteur obtenu en appliquant à  $y^{(N)}$  d'abord `numpy.fft.fft` puis `numpy.fft.fftshift` et en divisant le tout par  $(N + 1)$ . On note  $u^{(N)} =: (u_k^{(N)})_{k \in [-N/2, N/2]}$  les coordonnées de  $u^{(N)}$ . Tracer la partie réelle de  $(u_k^{(N)})_{k \in [-60, 60]}$  en fonction des entiers de  $[-60, 60]$ .

3. Pour  $k \in \mathbf{Z}$  et  $a : \mathbf{R} \rightarrow \mathbf{C}$ , on note

$$\theta_{a,k} : \mathbf{R} \times \mathbf{R} \rightarrow \mathbf{C}, \quad (t, x) \mapsto a(t) e^{2i\pi kx}.$$

(a) Pour  $k \in \mathbf{Z}$ , déterminer l'unique fonction  $a$  telle que  $\theta_{a,k}(0, \cdot) = e^{2i\pi k \cdot}$ ,  $\partial_t \theta_{a,k}(0, \cdot)$  soit nul et  $\partial_t^2 \theta_{a,k} = \partial_x^2 \theta_{a,k}$ . On notera  $A_k(t)$  la valeur en  $t$  de ce  $a$ .

(b)\* Pour  $k \in \mathbf{Z}$ , déterminer l'unique fonction  $a$  telle que  $\partial_t \theta_{a,k}(0, \cdot) = e^{2i\pi k \cdot}$ ,  $\theta_{a,k}(0, \cdot)$  soit nul et  $\partial_t^2 \theta_{a,k} = \partial_x^2 \theta_{a,k}$ .

4. On pose  $t = 1/4$ . Multiplier  $u^{(N)}$  coordonnée par coordonnée par  $(A_k(t))_{k \in [-N/2, N/2]}$ , puis appliquer au résultat `numpy.fft.ifftshift` puis `numpy.fft.ifft`. On multiplie encore par  $N + 1$  le résultat et l'on note  $v^{(N)}$  le vecteur ainsi obtenu.

Tracer sur une même figure  $y^{(N)}$  et la partie réelle de  $v^{(N)}$  en fonction de  $x^{(N)}$ .