TD 4 : Recherche des zéros, transformation de Fourier

Exercice 4.1 : Méthode de la sécante

Implémentez la méthode de la sécante du cours. Appliquez votre code à la fonction $f(x) = x - \cos(x^2)$ avec les valeurs de départ $x_0 = 1$ et $x_1 = 0.5$.

Exercice 4.2 : Diffraction par une fente

On considère la diffraction d'une onde lumineuse monochromatique de longeur d'onde λ par une fente de largeur a. La figure de diffraction obtenue sur un écran à distance d de la fente est décrite par l'intensité $I(x)$

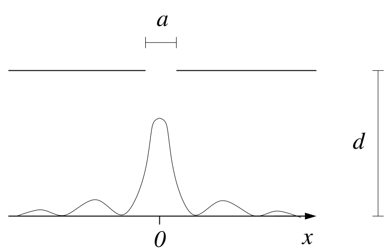
$$I(x) = I_0 \,\mathrm{sinc}^2 \frac{\pi a\, x}{\lambda d}\,, \qquad \text{avec } \mathrm{sinc}\, t \equiv \frac{\sin t}{t}\,.$$

On prend $a = 5$ μm, $\lambda = 600$ nm et $d = 20$ cm.

(a) *Sur papier :* Rendez-vous compte que les maxima d'intensité sont données par $x = \frac{\lambda d}{\pi a} t$, où t vérifie l'équation transcendante $\tan(t) - t = 0$.

(b) Le premier maximum est évidemment à $x = 0$. Calculez les positions des deux maxima suivantes avec une précision numérique relative d'au moins 10^{-3}. Utilisez une méthode appropriée de votre choix.

Indication : Repérez d'abord la position approximative des zéros, par exemple avec l'aide d'un graphique de la fonction, afin de trouver des bons points ou intervalles de départ.



Exercice 4.3 : Transformation de Fourier discrète

1. Réalisez une fonction `sft(phi)` (pour "slow Fourier transform") qui renvoie la TFD du vecteur à N composantes $\vec{\phi}$, calculée par le produit matriciel $F\vec{\phi}$. On rappelle que les éléments F_{mn} de la matrice de Fourier sont $F_{mn} = \omega^{mn}$, où $\omega = e^{-2\pi i/N}$.
2. Améliorez la fonction `fft(phi)` du cours pour qu'elle puisse aussi fonctionner avec un signal $\vec{\phi}$ dont la longueur n'est pas une puissance de 2. Pour ce faire, la fonction ajoutera d'abord autant de zéros à la fin du signal qu'il faut pour obtenir 2^q composantes.

   Indication : il convient de se servir de la fonction `numpy.log2` (logarithme de base 2).
3. La fonction `numpy.random.random_sample(n)` renvoie un tableau 1-dimensionnel de `n` nombres pseudo-aléatoires entre 0 et 1. Générez des séquences de 100, 1000 et 10 000 nombres pseudo-aléatoires et transformez-les avec la fonction `sft` et avec la fonction `fft` améliorée. Comparez le temps d'exécution (si vous voulez, vous pouvez le mesurer précisément avec `time.perf_counter()`).

Exercice 4.4 : Traitement de signal

1. Le fichier `sunspots.txt`, disponible sur Moodle, contient le nombre de taches solaires observées tous les mois depuis l'année 1750 environ (première colonne : mois, deuxième colonne : moyenne du nombre de taches).

   Creez un programme qui charge ces données et les trace. Vous remarquerez une périodicité dans l'activité solaire ; estimez la période avec l'aide du graphique. Puis, calculez la TFD des données et tracez le spectre de puissance, qui vous permettra de déterminer la période plus précisément.
2. Le fichier `piano.txt` (disponible sur Moodle) contient un échantillon audio d'une note jouée par un piano, enrégistrée avec un taux d'échantillonnage de 44.1 kHz. Trouvez la note que joue le piano.

Indication : En principe, vous avez le choix d'utiliser soit votre fonction `sft` d'exercice 4.3.1, soit la fonction `fft` du cours ou votre version améliorée d'exercice 4.3.2, soit la fonction `scipy.fft.fft` de la bibliothèque `SciPy`. Cependant, en fonction du signal, certains choix ne conviennent pas. Réfléchissez en amont quelle méthode peut être utilisée pour quel problème, et s'il y a plusieurs options, comparez les résultats.