TD 3 : NumPy et matplotlib

Exercice 3.1 : Manipulation des tableaux

(a) Créer un tableau de nombres flottants 1.5, 2, 2.5 ...10.5, 11.

(b) Réarranger dans un tableau 4×5 avec la fonction `numpy.reshape()`.

(c) De ce tableau, extraire les colonnes aux indices impairs.

(d) Extraire les lignes à partir de l'indice 2.

(e) De la dernière ligne, extraire les éléments aux indices pairs à partir de 2.

Exercice 3.2 : Calcul matriciel avec NumPy (système à trois niveaux)

Pour un système quantique à trois niveaux, on donne le hamiltonien

$$\hat{H} = \begin{pmatrix} E_1 & \epsilon & \delta \\ \epsilon^* & E_2 & \eta \\ \delta^* & \eta^* & E_3 \end{pmatrix}$$

où $E_{1,2,3}$ et ϵ, δ, η sont des constantes de dimension d'énergie. Ici on prendra $E_1 = 1, E_2 = 2, E_3 = 4, \epsilon = -1/3, \delta = -1/2, \eta = 0$ en unités où $\hbar = 1$.

(a) Réaliser une fonction `hmoyenne(psi)` qui calcule $\langle \hat{H} \rangle$ dans un état quelconque $|\psi\rangle = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{pmatrix}$, avec $\psi_{1,2,3}$ des nombres complexes. Calculer $\langle \hat{H} \rangle$ dans l'état $|\psi_0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$.

(b) Pour rappel, si l'état du système au temps $t = 0$ est $|\psi(0)\rangle$, alors au temps t il sera donné par

$$|\psi(t)\rangle = \exp\left(-\frac{i}{\hbar}\hat{H}t\right)|\psi(0)\rangle$$

selon l'équation de Schrödinger. Ici l'exponentielle d'une matrice est définie par la série de Taylor

$$\exp(\hat{A}) = \mathbb{1} + \hat{A} + \frac{1}{2}\hat{A}^2 + \frac{1}{3!}\hat{A}^3 + \ldots + \frac{1}{n!}\hat{A}^n + \ldots$$

où les puissances des matrices sont définies avec le produit matriciel.

Réaliser une fonction `expmat(A, N=40)` qui renvoie une approximation $\sum_{n=0}^{N} \hat{A}^n/n!$ de l'exponentielle de la matrice `A`. On suppose qu'au temps $t = 0$ le système soit dans l'état $|\psi(0)\rangle = |\psi_0\rangle$ comme défini ci-dessus. Calculer $|\psi(t)\rangle$ à $t = 3$.

(c) La fonction `eig(A)`, définie dans la bibliothèque `scipy.linalg`, prend comme argument une matrice carrée `A`. Elle renvoie un tableau 1-dimensionnel `w` contenant ses valeurs propres et un tableau 2-dimensionnel `vr` dont les vecteurs de colonne sont ses vecteurs propres. Se servir de cette fonction pour calculer l'énergie de l'état fondamental du système. Afficher le résultat à l'écran.

Exercice 3.3 : Graphisme

On rappelle la fonction de Gauss de l'exercice 2.4,

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \, .$$

Pour $\mu = 0$ et $\sigma = 1, 2, 3$, tracer les trois courbes de $f(x)$ sur l'intervalle $[-5,\, 5]$ dans un seul graphique. Celui-ci incluera une légende et des étiquettes pour les axes. (Il convient d'implémenter la fonction f de nouveau dans `NumPy`.)

Exercice 3.4 : Importer et visualiser des données

Dans le fichier `647_Global_Temperature_Data_File.txt` vous trouvez les déviations ΔT de la température moyenne globale par rapport à T_0, où T_0 est la température moyenne entre 1951 et 1980. Première colonne : année, deuxième colonne : ΔT en °C, troisième colonne : $\overline{\Delta T} = (\Delta T$ moyennée sur 5 ans).

(Source : `https://climate.nasa.gov/vital-signs/global-temperature/`)

(a) Tracer ΔT et $\overline{\Delta T}$ en fonction du temps dans un seul graphique.
(b) Faire un histogramme de ΔT.

Exercice 3.5 : Diagramme de bifurcation ($\star$)

La *suite logistique* est récursivement définie par

$$x_0 = \frac{1}{2}\,, \quad x_{n+1} = r\, x_n(1 - x_n)\, .$$

Ici $r \geq 0$ est un paramètre constant dont la valeur détermine le comportement asymptotique de la suite. Plus précisément, pour $0 \leq r \leq 3$, la suite converge ; pour $3 < r \lesssim 3.6$, elle se compose d'un nombre fini de sous-suites convergentes ; pour $r \gtrsim 3.6$, le comportement est largement chaotique, à part certains intervalles qui donnent lieu aux structures plus régulières. L'objectif de cet exercice est d'étudier la suite logistique numériquement par son *diagramme de bifurcation* ou *diagramme de Feigenbaum*.

(a) Pour 1000 valeurs de r entre 2.5 et 4, calculer les x_n pour $0 \leq n \leq 200$. Enregistrer ces 201 000 nombres dans un fichier `logistique.dat`.
(b) Avec un deuxième programme, importer les données de `logistique.dat` et tracer $x_{100}, x_{101}, x_{102} \ldots x_{200}$ en fonction de r entre $r = 2.5$ et $r = 4$, tout dans le même graphique. Pour l'affichage, utiliser des marqueurs pixel créés avec une virgule '`,`' dans l'appel de `matplotlib.pyplot.plot()`.

Exercice 3.6 : Graphiques des fonctions de deux variables

La fonction d'onde de l'état $|n = 4,\, \ell = 3,\, m = 1\rangle$ de l'atome d'hydrogène est donnée en coordonnées cartésiennes par

$$\psi_{431}(x, y, z) = N\; \exp(-r/4) \frac{(x + i\, y)\, z\, (7\, z^2 - 3\, r^2)}{r}$$

où $r = \sqrt{x^2 + y^2 + z^2}$, N est une constante de normalisation (on prendra $N = 1$), et toutes les quantités sont en unités du rayon de Bohr a_0.

Visualiser la densité de probabilité $||\psi_{431}||^2$ dans le plan $(x,\, z)$ pour x et z entre $-20\, a_0$ et $20\, a_0$

(a) par des courbes de niveau,
(b) par une carte de chaleur.