TP SIMULATION NUMÉRIQUE : NEWTON VECTORIEL ET ÉQUATION DE LA CHALEUR

Objectifs du TP

- Refaire de la simulation numérique.
- Appliquer la méthode de Newton à des fonctions vectorielles.
- Résoudre numériquement l'équation de la chaleur par différences finies et par séries de Fourier.

```
import numpy as np
import matplotlib.pyplot as plt
```

1 Méthode de Newton vectorielle

On peut appliquer la méthode de Newton (ou Newton-Raphson) à une fonction $f : \Omega \to \mathbb{R}^m$ de classe $\mathscr{C}^2$ sur un ouvert Ω de $\mathbb{R}^m$.

On peut donc écrire $f(x_1,\dots,x_m) = \big(f_1(x_1,\dots,x_n),\dots,f_m(x_1,\dots,x_n)\big)$ où $f_1,\dots,f_m$ sont à valeurs réelles.

On cherche à résoudre numériquement l'équation $f(x) = 0$, c'est-à-dire $f(x_1,\dots,x_m) = (0,\dots,0)$.

L'idée est, comme en dimension 1, de créer récursivement une suite avec

$$X_{n+1} = X_n - J_f(X_n)^{-1} \cdot f(X_n)$$

où X_n est un élément (vecteur) de $\mathbb{R}^m$ et J_f est la matrice jacobienne de f, c'est-à-dire la matrice $\left(\dfrac{\partial f_i}{\partial x_j}\right)_{1\leqslant i,j\leqslant m}$ supposée inversible en tout point où elle est calculée.

On peut démontrer que si f est de classe $\mathscr{C}^2$, $f(a) = 0$ et $J_f(a)$ est inversible, alors a est un point fixe superattractif de $\varphi : x \mapsto x - J_f(x)^{-1} \cdot f(x)$ ce qui permet d'obtenir une convergence rapide pour peu que l'on ne parte (avec X_0) pas trop loin de a.

Comme inverser J_f est coûteux et peut poser des problèmes numériques, on préfère en général résoudre le système linéaire

$$J_f(X_n) \cdot (X_n - X_{n+1}) = f(X_n)$$

pour déduire X_{n+1} de X_n.

1. Écrire une fonction `distance(X, Y)` renvoyant la distance euclidienne entre deux vecteurs représentés par des tableaux numpy de même dimension m.

2. Écrire une fonction `Newton(X0, f, Jf, epsilon)` prenant en entrée

- un tableau numpy `X0` de taille un certain m,
- `f` une fonction prenant en entrée un tableau de taille m
- `Jf` une fonction prenant en entrée un tableau de taille m et renvoyant un tableau de taille $m \times m$

et renvoyant un tableau de taille m correspondant au terme de la suite obtenue par la méthode de Newton en s'arrêtant dès que deux termes successifs sont à distance au plus `epsilon`.

3. On s'intéresse à l'équation $z^3 = 1$. Pour cela, nous allons appliquer la méthode de Newton appliquée à la fonction $f : (x, y) \mapsto \left(\mathfrak{Re}\left((x+\mathrm{i}y)^3 - 1\right), \mathfrak{Im}\left((x+\mathrm{i}y)^3 - 1\right)\right)$.

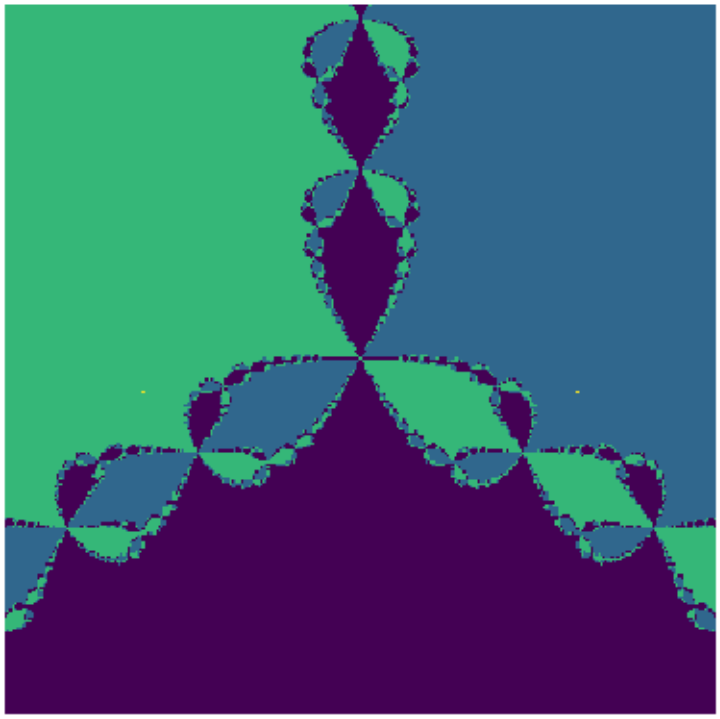
3.a) Calculer à la main $f(x, y)$ puis $J_f(x, y)$.

3.b) Définir des fonctions `f(X)` et `Jf(X)` correspondant (on pourra écrire `x, y = X`.)

3.c) Dans une fonction `attraction(h, epsilon)`, en travaillant avec une grille de pas h couvrant le carré $[-1{,}5, 1{,}5]^2$, Résoudre $f(x, y)$ par la méthode de Newton initialisée en chacun des points (x_i, y_j) de cette grille. Renvoyer un tableau `T` le numéro k entre 0 et 2 de la racine $e^{\frac{2ik\pi}{3}}$ vers laquelle l'algorithme converge (à `epsilon` près).

3.d) Tracer à l'aide de `plt.imshow(T)` pour $h = 0{,}1$.

3.e) Recommencer en réduisant h.



2 Équation de chaleur[1]

On considère l'équation de la chaleur

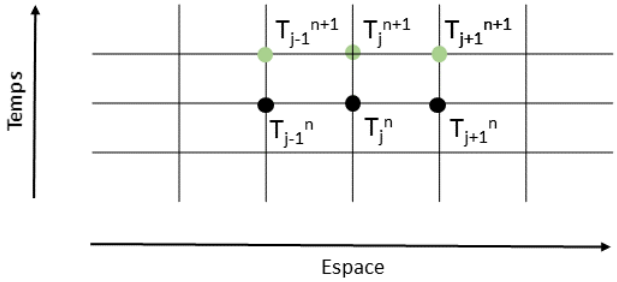
$$\begin{cases} \dfrac{\partial^2 T}{\partial x^2}(x,t) - \dfrac{1}{D}\dfrac{\partial T}{\partial t} = 0 & \\ T(x,0) = T_0(x) & \text{pour tout } x > 0 \\ T(0,t) = T(L,t) = T_{ext} & \text{pour tout } t > 0 \end{cases}$$

où T est définie sur $[0,L] \times [0,+\infty[$ et T_0 est définie sur $[0,L]$ représentant la température d'une tige dont les extrémités sont maintenues à température constante.

2.1 Discrétisation : schéma de Crank-Nicholson

Pour résoudre numériquement l'équation de la chaleur, on discrétise le temps et l'espace pour former une subdivision régulière $(x_j, t_n)_{\substack{1 \leqslant j \leqslant N_x \\ 1 \leqslant n \leqslant N_t}}$ de $[0,L] \times [0,t_f]$ de pas $\delta = L/N_x$ en espace et $\tau = t_f/N_t$ en temps.

On note[2] T_j^n la valeur approchée de $T(x_j, t_n)$.



La dérivée seconde en espace $\dfrac{\partial^2 T}{\partial x^2}(x_j, t_n)$ va être approchée en utilisant la formule vue en début d'année pour un schéma aux différence finies (avec u de classe $\mathscr{C}^2$) :

$$u''(x) = \frac{u(x-h) - 2u(x) + u(x+h)}{h^2} + \underset{h \to 0}{o}(1)$$

mais on prendra la moyenne aux temps t_n et t_{n+1} :

$$\frac{T_{j-1}^{n+1} - 2T_j^{n+1} + T_{j+1}^{n+1}}{2\delta^2} + \frac{T_{j-1}^{n} - 2T_j^{n} + T_{j+1}^{n}}{2\delta^2}$$

et la dérivée première en temps $\dfrac{\partial T}{\partial t}(x_j, t_n)$ est approchée classiquement par $\dfrac{T_j^{n+1} - T_j^n}{\tau}$.

4. Donner alors une équation reliant T_{j-1}^n, T_j^n, T_{j+1}^n, T_{j-1}^{n+1}, T_j^{n+1}, T_{j+1}^{n+1}, τ, D et δ.

1. Librement inspiré du livre *Informatique - Programmation et calcul scientifique - en Python et Scilab* de T. Audibert et A. Oussalah (ed. ellipses)
2. Ce n ne désigne évidemment pas une puissance.

5. Vérifier qu'en posant en posant $\alpha = \dfrac{D\tau}{2\delta^2}$, pour tout $j \in [\![1, N_x - 1]\!]$,

$$-\alpha T_{j-1}^{n+1} + (1+2\alpha) T_j^{n+1} - \alpha T_{j+1}^{n+1} = \alpha T_{j-1}^n + (1-2\alpha) T_j^n + \alpha T_{j+1}^n.$$

6. Écrire une fonction `matrice(m, a, b)` renvoyant un tableau de taille $(m+1) \times (m+1)$ de la forme :

$$\begin{pmatrix} 1 & 0 & & & (0) \\ a & b & a & & \\ & \ddots & \ddots & \ddots & \\ & & a & b & a \\ (0) & & & 0 & 1 \end{pmatrix}$$

7. On note $T^{(n)} = \begin{pmatrix} T_0^n & T_1^n & \dots & T_{N_x}^n \end{pmatrix}^\intercal = \begin{pmatrix} T_0^n \\ T_1^n \\ \vdots \\ T_{N_x}^n \end{pmatrix}$ la distribution de températures au temps t_n. Que valent le premier et le dernier cœfficients ? Que vaut $T^{(0)}$?

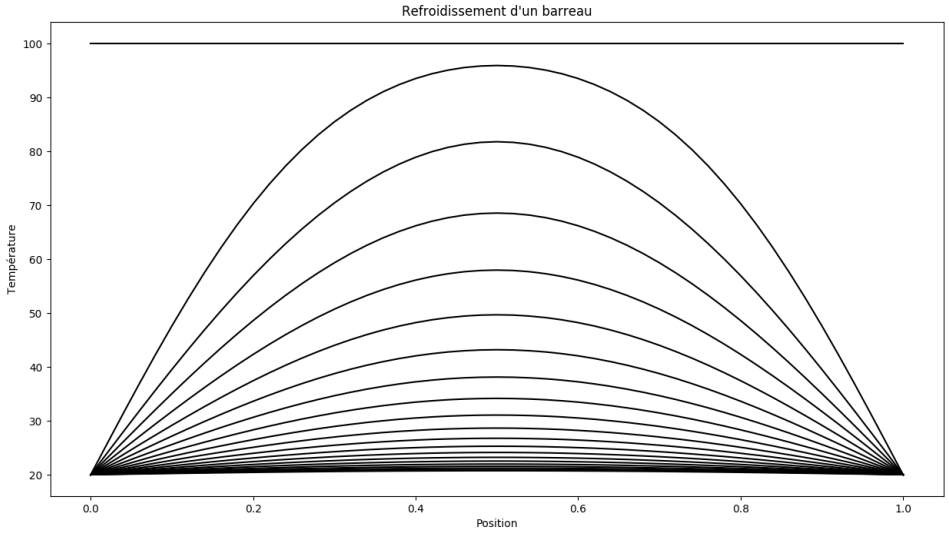
8. Montrer que, si on connaît $T^{(n)}$, on détermine $T^{(n+1)}$ en résolvant un système $AT^{(n+1)} = BT^{(n)}$ où A et B sont de la forme :

$$A = \begin{pmatrix} 1 & 0 & & & (0) \\ -\alpha & 1+2\alpha & -\alpha & & \\ & \ddots & \ddots & \ddots & \\ & & -\alpha & 1+2\alpha & -\alpha \\ (0) & & & 0 & 1 \end{pmatrix} \quad \text{et} \quad B = \begin{pmatrix} 1 & 0 & & & (0) \\ \alpha & 1-2\alpha & \alpha & & \\ & \ddots & \ddots & \ddots & \\ & & \alpha & 1-2\alpha & \alpha \\ (0) & & & 0 & 1 \end{pmatrix}$$

en posant $T_0^0 = T_{Nx}^0 = T_{ext}$.

9. Écrire une fonction `CrankNicholson(L, tf, D, T0, Text, Nx, Nt)` qui retourne le tableau T dont les colonnes sont les $T^{(n)}$.

10. Tracer les solutions approchées pour $x \in [0, L]$ et $t \in [0, t_f]$, en prenant $T_0 \equiv 100$, $L = 1$, $D = 1$, $t_f = 0{,}5$, $T_{ext} = 20$, $N_t = 1000$, $N_x = 50$.



2.2 Solution exacte

Le problème admet comme unique solution

$$T(x,t) = T_{ext} + \sum_{k=1}^{+\infty} c_k \sin\left(\frac{k\pi}{L}x\right) \mathrm{e}^{-\left(\frac{k\pi}{L}\right)^2 Dt}$$

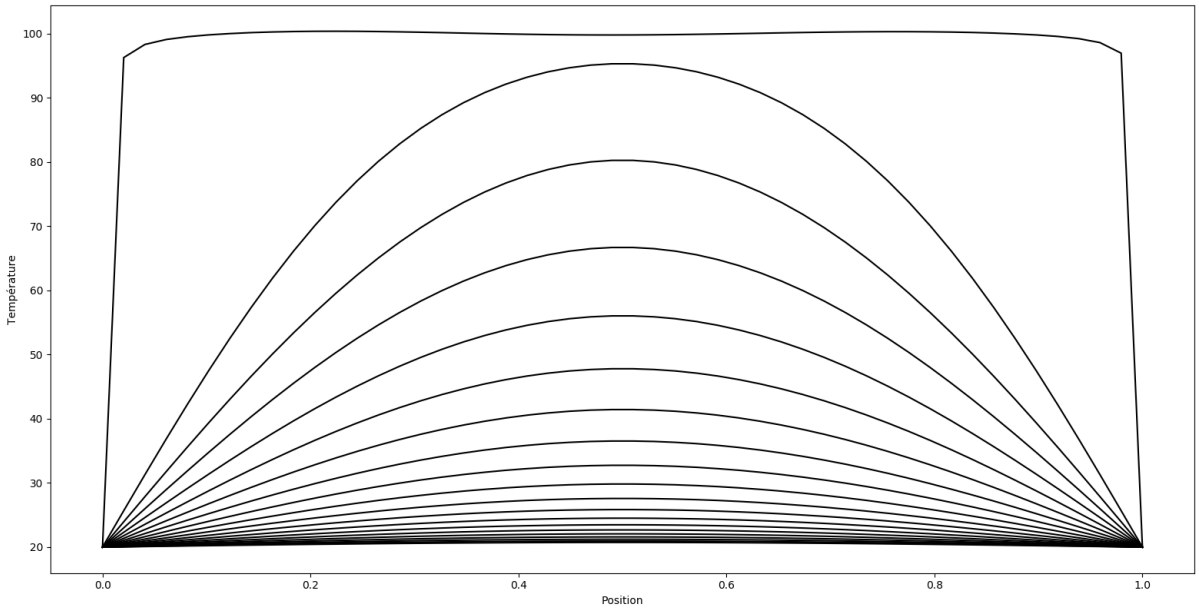
avec [3]

$$c_k = \frac{2}{L}\int_0^L \sin\left(\frac{k\pi}{L}s\right)(T_0(s) - T_{ext})\,\mathrm{d}s.$$

11. Écrire une fonction `c(k, T0, L, Text)` calculant une valeur approchée de c_k : l'intégration se fera par la méthode des trapèzes, avec un pas suffisamment petit.

12. Écire une fonction `T(T0, n, L, D, x, t, Text)` renvoyant une approximation de $T(x,t)$ correspondant à la somme partielle d'indice n.

13. Tracer sur un même graphe la température en fonction de $x \in [0, L]$ pour divers temps $t \in [0, t_f]$, en prenant $T_0 \equiv 100$, $L = 1$, $D = 1$, $T_{ext} = 20$ et $t_f = 0{,}5$.



3. Ce sont les cœfficients de Fourier du prolongement impaire de T_0 en une fonction $2L$-périodique.