TP 2 : ingéniérie numérique et simulation

Approximation d'intégrales par la méthode des trapèzes

Le module `scipy.integrate` propose une fonction `quad(f,a,b)` qui calcule $I = \int_a^b f(x)\,\mathrm{d}x$ et donne un majorant de l'erreur commise :

```
import scipy.integrate as integr
I = integr.quad(f,a,b)
```

1. Écrire une fonction `trapezes` de paramètres `f`, `a`, `b` et `n` qui calcule $I = \int_a^b f(x)\,\mathrm{d}x$ en utilisant la méthode des trapèzes sur `n` intervalles : I est approchée par $\sum_{i=1}^{n}(x_i - x_{i-1})\dfrac{y_i + y_{i-1}}{2}$ avec $x_i - x_{i-1} = \dfrac{b-a}{n}$, $x_0 = a$ et $y_i = f(x_i)$.
   La fonction doit aussi renvoyer le résultat de l'intégrale fourni par `quad(f,a,b)`.
2. Vérifier que le programme fonctionne en calculant $\int_0^{\pi} \sin(t)\mathrm{d}t$.
3. Une majoration de l'erreur due à la méthode des trapèzes est $\varepsilon = \dfrac{b-a}{12}\max|y_{i+1} + y_{i-1} - 2y_i|$. Écrire une fonction `trapezes_2` qui renvoie, en plus de la valeur approchée de I, la majoration de l'erreur commise.
   Tracer $f(t) = \dfrac{\mathrm{e}^{-t}}{1+t^2}$ sur $[0,4]$ et lui appliquer la fonction précédente.

Dynamique d'un sytème prédateurs / proies

L'évolution des effectifs des proies $x(t)$ et des prédateurs $y(t)$ est modélisé par un système de deux équations différentielles du premier ordre couplées : $\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = x(t)\,[\alpha - \beta y(t)] \\ \dfrac{\mathrm{d}y}{\mathrm{d}t} = y(t)\,[-\gamma + \delta x(t)] \end{cases}$

α est le taux de reproduction des proies, dont l'effectif en l'absence de prédateurs et avec des ressources en nourriture supposées inépuisables, augmenterait exponentiellement. $\beta y(t)$ est le taux de mortalité des proies, dû essentiellement aux prédateurs (on néglige les morts « naturelles »). γ est le taux de mortalité des prédateurs, dont l'effectif diminuerait exponentiellement en l'absence de proies. $\delta x(t)$ est le taux de reproduction des prédateurs, dû essentiellement à leur consommation de proies.

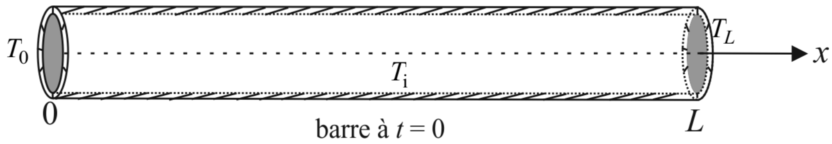
On cherche à déterminer l'évolution des deux populations entre les instants t_0 et t_f.
On travaillera avec la bibliothèque `numpy` et des objets de type `np.array`. On utilisera un tableau `L` de $N+1$ lignes et 2 colonnes dont la ligne i contient les valeurs x_i, y_i des fonctions $x(t)$ et $y(t)$ aux instants $t_i = t_0 + i * \mathrm{d}t$, où $\mathrm{d}t = (t_\mathrm{f} - t_0)/N$ est le pas temporel d'intégration. La tableau `L` est initialisé de la façon suivante :

```
L = np.zeros((N+1,2))
```

1. En utilisant la méthode d'Euler, déterminer l'évolution des deux populations entre les instants $t = 0$ et $t = 10$, avec $\alpha = 2$, $\beta = 0,01$, $\gamma = 10$, $\delta = 0,5$, en prenant les populations initiales $x_0 = 10$ et $y_0 = 100$.
2. Représenter les courbes $x(t)$ et $y(t)$ ainsi que la trajectoire de phase : courbe paramétrique $\begin{cases} x(t) \\ y(t) \end{cases}$.

Équation de la chaleur à une dimension

On considère une barre d'axe Ox constituée d'un solide homogène de masse volumique ρ, de conductivité thermique λ et de capacité thermique c. La barre est de longueur L et de section très inférieure à L^2, si bien que l'on peut négliger les variations de température dans une section droite de la barre. La température ne dépend donc que de l'abscisse x et du temps t : $T(x,t)$. La barre est calorifugée sur sa surface latérale.



Initialement, la barre est en équilibre avec le milieu extérieur : sa température est uniforme et égale à `Ti`.
À partir de $t = t_0 = 0$, on porte l'extrémité $x = 0$ de la barre à la température `T0` et l'extrémité $x = L$ à `TL`. La température dans la barre est solution de l'équation de la chaleur à 1D : $\dfrac{\partial^2 T}{\partial x^2} = \dfrac{1}{a}\dfrac{\partial T}{\partial t}$, avec $a = \dfrac{\lambda}{\rho c}$, diffusivité thermique du solide.

1. Nous allons, afin de déterminer l'évolution de la température dans la barre pour $t \in [0, \texttt{tmax}]$, approximer la dérivée partielle seconde par rapport à x grâce à la formule : $\dfrac{\partial^2 T}{\partial x^2} \simeq \dfrac{T(x+\Delta x,t) - 2T(x,t) + T(x-\Delta x,t)}{(\Delta x)^2}$.
   Le pas Δx sera pris égal à L/M, où M est un entier.
   Justifier cette expression. À quelle condition sur M la simulation peut-elle s'approcher de la solution exacte du problème ?
2. On discrétise de même l'intervalle de temps étudié : $\Delta t = t_\text{max}/N$, où N est un entier. On note `T[n,i]` la température au point $x_i = i\Delta x$, $i \in [\![0, M]\!]$ et à l'instant $t_n = n\Delta t$, $n \in [\![0, N]\!]$.
   Une ligne de la matrice `Temperature` dont les coefficients sont les `T[n,i]` contient les valeurs de température aux points de la barre d'abscisses x_i à un instant fixé ; une colonne de la matrice contient les valeurs de température en un point fixé de la barre aux différents instants t_n.
   Montrer que l'on peut calculer `T[n,i]` en fonction de `T[n-1,i-1]`, `T[n-1,i]` et `T[n-1,i+1]` et du nombre de Fourier $\text{Fo} = a\dfrac{\Delta t}{(\Delta x)^2}$. On parle de schéma *explicite* en temps.
3. On montre que le schéma est stable si $\text{Fo} \leqslant \dfrac{1}{2}$. Donner à l'aide d'une analyse dimensionnelle une durée caractéristique de la diffusion thermique sur une longueur Δx et justifier la condition de stabilité.
4. On cherche maintenant à calculer sous Python la matrice des `T[n,i]` et à afficher le profil de température dans la barre à différents instants. On introduit les valeurs numériques suivantes :

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

a = 1E-5 #diffusivité thermique du solide
L = 0.1 #longueur de la barre
Ti = 20 #température initiale dans la barre
TL = 20. #température de l'extrémité droite
T0 = 100. #température de l'extrémité gauche
tmax = 250 #durée de l'étude
M = 200 #nombre d'intervalles d'absisses
```

   Créer à l'aide de `np.linspace` la matrice `x` des $M+1$ abscisses ainsi que le pas spatial Δx noté `dx` et un pas temporel `dt_stab` assurant la stabilité du schéma. On prendra pour cela une marge en multipliant par `beta=0.999` la valeur maximale de Δt. Calculer la valeur correspondante du nombre de Fourier `Fo` ainsi que le nombre `N` d'intervalles temporels.
5. Créer la matrice des températures `Temperatures` de $N+1$ lignes et $M+1$ colonnes, respectant la condition initiale et les conditions aux limites.
6. Remplir la matrice pour obtenir la solution approchée à tous les instants t_n, puis tracer sur le même graphe les courbes donnant le profil de température à une dizaine d'instants différents.
7. Commenter la forme que prend le profil de température pour `tmax` assez grand.
8. Prendre `beta=1.0003` puis des valeurs plus grandes et commenter le résultat obtenu.
9. Revenir à `beta=0.999` et diminuer la valeur de `M`. Commenter.