Méthodes numériques
Résolution numérique d'équations différentielles

Compétences

- ❏ À l'aide d'un langage de programmation, simuler l'évolution temporelle d'un signal généré par un oscillateur.
- ❏ À l'aide d'un langage de programmation, résoudre l'équation de la diffusion thermique à une dimension par une méthode des différences finies dérivée de la méthode d'Euler explicite de résolution des équations différentielles ordinaires.
- ❏ À l'aide d'un langage de programmation, simuler la propagation d'un paquet d'ondes dans un milieu dispersif et visualiser le phénomène d'étalement.

Exercices

❏ Exercice 1 ❏ Exercice 2

Résumé du cours

1. Résolution numérique d'équations différentielles

La méthode d'Euler est une procédure numérique permettant de résoudre numériquement et approximativement des équations différentielles à partir d'une condition initiale.

1.1. Discrétisation

La mémoire des ordinateur étant finie, il est indispensable de discrétiser le problème pour le résoudre numériquement. Discrétiser consiste à associer à une fonction $y(t)$ une suite $y_i = y(i \cdot \Delta t)$ où Δt est appelé pas de temps. Le pas de temps est l'équivalent de la période d'échantillonnage.

1.2. Problème d'Euler

Un problème d'Euler est une équation différentielle d'ordre 1 muni d'une condition initiale : on cherche la fonction y telle que $y'(t) = f(y,t)$ et $y(t=0) = y_0$. La fonction recherchée peut éventuellement être un vecteur et donc avoir plusieurs composantes.

Exemple

$$\frac{\mathrm{d}u}{\mathrm{d}t} = -\frac{1}{\tau}u$$

$$\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = \vec{g} - \frac{k}{m}\vec{v}$$

Les équations différentielle d'ordre supérieur peuvent être mises sous la forme de problème d'Euler en introduisant un vecteur dont les coordonnées sont des dérivées successives.

Application ✍[1]

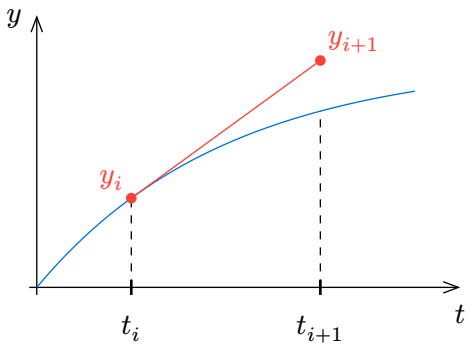
Mettre sous forme de problème d'Euler les équations différentielles suivante.

$$m\frac{\mathrm{d}^2x}{\mathrm{d}t^2} = -kv^2$$

$$m\frac{\mathrm{d}^2\overrightarrow{OM}}{\mathrm{d}t^2} = -k\vec{v} + m\vec{g}$$

$$a\frac{\mathrm{d}^3y}{\mathrm{d}t^3} + b\frac{\mathrm{d}^2y}{\mathrm{d}t^2} + c\frac{\mathrm{d}y}{\mathrm{d}t} + dy = e$$

1.3. Méthode d'Euler



La méthode d'Euler consiste à approximer la courbe localement par sa tangente. Cette approximation s'appuie sur la formule de Taylor à l'ordre 1.

Schéma d'Euler ♥ ✍²

$$y_{i+1} = y_i + \Delta t \cdot f(y_i, t)$$

2. Résolution numérique d'équations aux dérivées partielles}

Il est possible d'adapter la méthode d'Euler pour résoudre des équations aux dérivées partielles.

2.1. Discrétisation

Une double discrétisation spatiale et temporelle pour résoudre numériquement une équation aux dérivées partielle. On pose $f_{i,j} = f(i \cdot \Delta t, j \cdot \Delta x)$ où Δt est le pas de temps et Δx le pas d'espace.

2.2. Résolution numérique de l'équation de diffusion

Dans cette partie, on s'appuie sur l'exemple de l'équation de diffusion $\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2}$ mais la méthode peut être adaptée à toute équation aux dérivées partielles.

Schéma pour résoudre numériquement une équation de diffusion ✍³

$$T_{i+1,j} = D\frac{\Delta t}{\Delta x^2}T_{i,j+1} + D\frac{\Delta t}{\Delta x^2}T_{i,j-1} + \left(1 - 2\frac{D(\Delta t)}{\Delta x^2}\right)T_{i,j}$$

Afin que ce schéma soit stable, il est nécessaire que $2D\frac{\Delta t}{\Delta x^2} < 1$.

On peut interpréter $T_{i,j}$ sous la forme d'une matrice. Chaque ligne i correspond alors à la température à un instant $i \cdot \Delta t$ partout dans le milieu. Chaque colonne j correspond à la température à une position $j \cdot \Delta x$ à tous les instants.

Exercices

1. 💻 Oscillateur de Wien

Pour rappel, un oscillateur de Wien est constitué d'un montage amplificateur non-inverseur et d'un filtre de Wien.

On note
- $u(t)$ la tension en entrée de l'amplificateur non-inverseur (et donc en sortie du filtre de Wien)
- $v(t)$ la tension en sortie de l'amplificateur non-inverseur (et donc en entrée du filtre de Wien)

Les fonctions de transfert du filtre de Wien et de l'amplificateur non-inverseur sont respectivement données par :

$$H_{\text{ANI}}(p) = \frac{V(p)}{U(p)} = \frac{1 + \frac{R_2}{R_1}}{1 + \left(1 + \frac{R_2}{R_1}\right)\frac{\tau}{A_0}p}$$

$$H_{\text{Wien}}(p) = \frac{U(p)}{V(p)} = \frac{\frac{1}{3}}{1 + \frac{1}{3}\left(\frac{1}{RCp} + RCp\right)}$$

On pose $w = \frac{\mathrm{d}u}{\mathrm{d}t}$. On cherche à mettre le problème sous la forme d'un problème d'Euler de la forme $\frac{\mathrm{d}\vec{Y}}{\mathrm{d}t} = F\left(t, \vec{Y}\right)$ avec $\vec{Y} = \begin{pmatrix} u \\ v \\ w \end{pmatrix}$.

1/ Exprimer les dérivées de u, v et w uniquement en fonction de u, v, w, R, C, R_1, R_2, A_0 et τ.

Les questions suivantes se font en ligne, sur Capytale : 2253-2586522

Avec Python, on représente le vecteur $\vec{Y}$ par un array numpy à 3 éléments.

2/ Écrire une fonction Python qui prend en entrée Y et renvoie sa dérivée. On pourra supposer les variables R, C, R_1, R_2, A_0 et τ déjà définies.

Dans un premier temps, on utilise la fonction `scipy.integrate.solve_ivp` qui résout numériquement des équations différentielles mises sous la forme d'un problème d'Euler. Cette fonction utilise des variantes de la méthodes d'Euler la rendant plus précise.

La fonction solve_ivp prend en argument
- la fonction `F` définie précédemment
- le couple $\left(t_i, t_f\right)$ correspondant à l'intervalle de temps sur lequel on souhaite résoudre l'équation différentielle
- le array $\vec{Y}(t_i)$ correspondant aux conditions initiales

La fonction `solve_ivp` renvoie un objet. Si on le stocke dans la variable `solution`,
- `solution.t` contient les temps
- `solution.y` contient les valeurs successives de $\vec{Y}$ à ces temps

3/ Définir et affecter les variables $R = 1\,\text{k}\Omega$, $C = 10\,\text{nF}$, R_1, R_2, $A_0 = 100\,000$ et $\tau = 0.01\,\text{s}$ avec des valeurs vraisemblables satisfaisant la condition de démarrage des oscillations.

4/ Définir `Y0` avec des très petites valeurs pour u, v et w (0.000 1 par exemple).

5/ Définir tf pour observer une dizaine d'oscillations.

6/ Tracer u et v en fonction du temps.

7/ Vérifier la condition de démarrage des oscillations.

8/ Vérifier la valeur de la période des oscillations.

Dans la suite, on souhaite se passer de la fonction et implémenter nous-même la méthode d'Euler.

On note $Y_i = Y(i \cdot \Delta t)$ où Δt est la durée entre deux échantillons (période d'échantillonnage).

9/ Dans le cas général, exprimer Y_{i+1} en fonction de Y_i, de i, Δt et de la fonction F.

10/ Implémenter la méthode d'Euler pour simuler l'évolution des tensions pour un oscillateur de Wien. Δt sera choisi de sorte à ce qu'il y ait environ 50 échantillons par période.

11/ Adapter le code précédent pour prendre en compte la saturation de l'ALI.

2. 💻 Propagation de la chaleur

On cherche à modéliser l'évolution de la température dans un barreau cylindrique en aluminium ($D = 99 \cdot 10^{-6}\,\mathrm{m^2\,s^{-1}}$) de longueur $l = 10\,\mathrm{cm}$. L'évolution de la température est donnée par l'équation de diffusion $\frac{\partial T}{\partial t} = D\frac{\partial^2 T}{\partial x^2}$.

On discrétise spatialement la barre en prenant $N_x = 20$ points.

Le code Python pourra être saisi directement sur Capytale : 1285-5360170

1/ Sachant que le schéma d'Euler est stable ssi $2D\frac{\Delta t}{\Delta x^2} < 1$, quel pas temporel maximal peut-on choisir ? On choisira dans toute la suite $\Delta t = 0.1s$. Compléter le code suivant.

```
D = 99e-6
l = 10e-2
N_x = 20
Delta_x = ... # pas spatial
Delta_t = 0.1 # pas temporel
```

2/ La température dans le barreau à chaque instant sera stockée dans une matrice de sorte que $T_{i,j} = T(i \cdot \Delta t, j \cdot \Delta x)$. On souhaite simuler l'évolution de la température durant 4 minutes. Combien de lignes doit comporter la matrice ? Combien de colonnes ? Compléter le code suivant.

```
import numpy as np
T = np.zeros((...,...))
```

3/ On initialise la simulation en supposant la température égale à $298K$ dans le barreau au début. Compléter le code suivant.

```
T[...] = 298 # Température initiale de la barre
```

4/ L'extrémité gauche du barreau ($x = 0$) est maintenu à une température de 350 K tandis que son extrémité droite ($x = 10\,\mathrm{cm}$) est maintenue à 298 K. Compléter le code suivant.

```
T[...] = 350 # Température de la barre en x=0
T[...] = 298 # Température de la barre en x=10cm
```

5/ Montrer que l'équation de diffusion peut donner lieu à un schéma d'Euler $T_{i+1,j} = D\frac{\Delta t}{\Delta x^2}T_{i,j+1} + D\frac{\Delta t}{\Delta x^2}T_{i,j-1} + \left(1 - 2D\frac{\Delta t}{\Delta x^2}\right)T_{i,j}$. Compléter le code suivant.

```
for i in range(len(T)-1):
    for j in range(1, N_x-1):
        T[i+1,j] = ...
```

6/ Expliquer le choix des bordes des deux boucles du code précédent.

7/ Tracer sur le même graphe le profil de température dans la barre au bout de 15 s, 30 s, 1 min, 2 min et 4 min.

8/ Tracer la température du point central de la barre en fonction du temps.