

Révisions : structures de données, bibliothèques et applications

① Structures de données

Plan

- ① Structures de données
- ② Bibliothèques

Plan

- ① Structures de données
- ② Bibliothèques
- ③ Exemples d'algorithmes

Structures de données

- 1 Listes
- 2 Tableaux

- ① définir une liste L
- ② longueur d'une liste L
- ③ accéder à l'élément en position i
- ④ concaténer deux listes L et M
- ⑤ obtenir la sous-liste $L_i \cdots L_j$
- ⑥ supprimer un élément de la liste L .
- ⑦ copier une liste

Matrice

- ① définir un tableau ($M=np.zeros((p,q))$)
- ② modifier une valeur dans la matrice
- ③ nombre de colonnes, nombre de lignes.

- ① numpy (import numpy as np)
- ② matplotlib (import matplotlib.pyplot as plt)
- ③ random (import random as rnd)

Exemples d'algorithmes

- ① méthode d'Euler.
- ② simulation d'une variable aléatoire.
- ③ statistiques descriptives.

Méthode d'Euler

Problème :

$$\begin{cases} y' = f(y) \\ y_0 = f(a) \end{cases}$$

f, a, y_0 données. Trouver une solution g de cette équation différentielle sur un intervalle $[a, b]$ avec $a = t_0$.

Méthode d'Euler : trouver une approximation \tilde{g} de la solution théorique g .

Méthode d'Euler

Trois étapes :

- 1 Subdiviser l'intervalle $[a, b]$ en $a = t_0 < t_1 < \dots < t_n = b$
- 2 Déterminer pour tout $i \in \{0, \dots, n\}$ la valeur de $\tilde{g}(t_i)$.
- 3 Tracer la courbe représentative de \tilde{g} .

Méthode d'Euler

syntaxe générale :

```
import numpy as np
import matplotlib.pyplot as plt
def Traceeuler(f,a,b,y0,n) :
    L=[a+i*(b-a)/n for i in range(n+1)]
    T=[y0]
    for i in range(n) :
        y=T[i]+((b-a)/n)*f(T[i]) # formule à justifier
        T.append(y)
    plt.plot(L,T)
```

Exemple :

$$\begin{cases} y' &= \frac{1}{y^2+1} \\ y(0) &= 1 \end{cases}$$

Méthode d'Euler sur le segment $[0, 1]$

```
import numpy as np
import matplotlib.pyplot as plt
def Traceeuler(y0,n) :
    L=[i/n for i in range(n+1)]
    T=[y0]
    for i in range(n) :
        y=T[i]+(1/n)*(1/(T[i]**2 +1)) # formule à justifier
        T.append(y)
    plt.plot(L,T)
```

Simulation d'une variable aléatoire finie

Simuler une variable aléatoire X avec $P(X = i) = p_i$.

syntaxe générale :

```
import random as rnd
# L contient la liste des probas P(X=i)=L[i]
def variable_aleatoire(L) :
    x=rnd.random()
    S=0
    for i in range(len(L)) :
        S=S+L[i]
        if x<= S :
            return i
```

Simulation d'une variable aléatoire finie

Exemple : $P(X = 0) = \frac{1}{6}$, $P(X = 1) = \frac{1}{3}$, $P(X = 2) = \frac{1}{2}$

```
import random as rnd
# L contient la liste des probas P(X=i)=L[i]
def variable_aleatoire() :
    x=rnd.random()
    L=[1/6,1/3,1/2]
    for i in range(len(L)) :
        S=S+L[i]
        if x<= S :
            return i
```

- ① Calculer la moyenne d'une liste de valeurs.
- ② Calculer la variance d'une liste de valeurs.
- ③ Calculer la médiane d'une liste de valeurs.
- ④ Calculer la covariance d'une liste de valeurs.
- ⑤ Construire la liste des occurrences à partir de la liste des valeurs.

Statistiques descriptives

- 1 `import numpy as np np.mean(L)`
- 2 `import numpy as np np.var(L)`
- 3 `import numpy as np np.median(L)`

(Fonctions à savoir implanter, autrement dit, connaître ses formules!)