

# 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

- 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
```

# Statistiques descriptives

- 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!)