TP 3. Suites récurrentes

October 3, 2021

1 Rappel

Pour faire des calculs efficacement et afficher des graphiques, on importera toujours les modules numpy et matplotlib via les commandes :

[3]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Dans ce TP, on s'intéresse à l'étude des suites définies par une relation de récurrence.

2 Suites récurrentes d'ordre 1

On commence par l'étude des deux suites suivantes :

$$u_0 = \frac{1}{2}, \quad \forall n \in \mathbb{N},\ u_{n+1} = f(u_n), \quad f(x) = \sqrt{1+x}$$

et

$$v_0 = \frac{1}{2}, \quad \forall n \in \mathbb{N},\ v_{n+1} = g(v_n), \quad g(x) = \mathrm{e}^{-\frac{5x}{7}}.$$

2.1 Calcul de termes successifs

2.1.1 Afficher simultanément :

- le graphe de la courbe $y = f(x)$, $x \in [-1, 3]$.
- le graphe de la courbe $y = x$, $x \in [-1, 3]$.

N'hésitez pas à mettre des couleurs et des légendes !

2.1.2 **Calculer “à la main” u_1, u_2 et u_3.**

2.1.3 **Écrire une fonction terme, qui prend en argument un entier n, et qui renvoie u_n à l'aide d'une boucle.**

2.1.4 **Tester en observant les valeurs u_{10}, u_{100}, u_{1000}...**

2.1.5 **Modifier légèrement le code de la fonction terme pour afficher les valeurs de la suite à chaque pas de la boucle.**

2.1.6 **À partir de la fonction terme, coder une fonction suite, qui prend en argument un entier n et qui renvoie la liste des n premières valeurs de la suite $(u_n)_n$.**

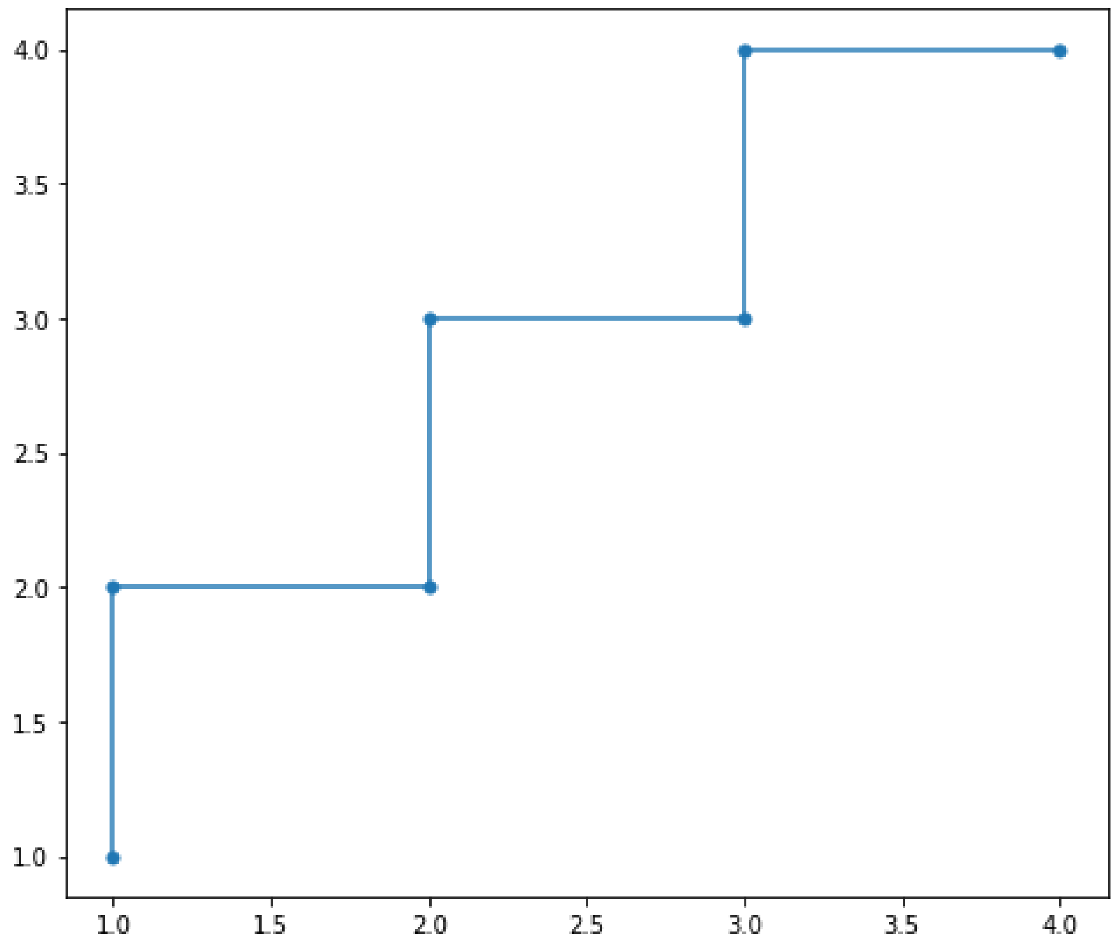
2.1.7 **En s'inspirant du code qui suit, coder une fonction visualisation, prend en argument un entier n, et qui dessine le diagramme reliant les points $(u_0, u_0), (u_0, u_1), (u_1, u_1), ..., (u_n, u_n)$. On superposera les graphes des courbes $y = f(x)$ et $y = x$.**

[1]:
```python
u = [1, 2, 3, 4]
print(u)

doubles = [valeur for valeur in u for i in range(2)]
print(doubles)
```

```
[1, 2, 3, 4]
[1, 1, 2, 2, 3, 3, 4, 4]
```

[4]:
```python
plt.figure(figsize = (8, 7))

plt.plot(doubles[0:-1], doubles[1:], marker="o", markersize = 5)
```

[4]: `[<matplotlib.lines.Line2D at 0x1cab675ef88>]`



2.1.8 Comment s'interprète le graphique obtenu ?

2.2 Reprendre les questions précédentes avec la suite $(v_n)_n$. On tracera le graphe des fonctions sur $[0, 1]$.

3 Une suite récurrente d'ordre 2 : la suite de Fibonacci

3.1 Coder une fonction fibo, qui prend en arguments deux réels a, b, un entier n, et qui renvoie la liste des n premiers termes de la suite de Fibonacci :

$$x_0 = a; \quad x_1 = b; \quad \forall n \in \mathbb{N},\ x_{n+2} = x_{n+1} + x_n.$$

3.2 On choisit $a = 1$ et $b = \dfrac{1-\sqrt{5}}{2}$. Observer le comportement de $\dfrac{x_{n+1}}{x_n}$.

3.3 Même question avec $a = 1$ et $b = \dfrac{1+\sqrt{5}}{2}$.

3.4 Chercher les suites de Fibonacci qui sont de la forme $x_n = q^n$ avec $q \neq 0$ (vous devez en trouver deux, associées à des réels q_1 et q_2).

3.5 Revenir sur les tests numériques et les expliquer.

4 Un exemple en dimension 2

On considère un milieu écologique constitué de proies (lapins) et de prédateurs (renards). Au temps t = 0, le nombre de lapins est L_0 et on a R_0 renards. On étudie l'évolution annuelle de ces populations : L_k et R_k sont respectivement les nombres de proies et de prédateurs la $k-$ième année. On essaie de modéliser cette évolution via la suite récurrente :

$$\begin{cases} L_{k+1} = 1.2 \times L_k - 0.1 \times R_k \\ R_{k+1} = 0.9 \times R_k + 0.2 \times L_k. \end{cases}$$

4.1 Coder une fonction interaction qui prend en arguments deux réels L_0, R_0, un entier n, et qui renvoie les deux listes des n premières valeurs des suites $(L_k)_k$ et $(R_k)_k$.

4.2 On choisit $L_0 = 20$ et $R_0 = 10$. Représenter sur un même graphe les suites $\dfrac{L_k}{L_k + R_k}$ et $\dfrac{R_k}{L_k + R_k}$.