

TP 6. Suites récurrentes 2

November 18, 2021

1 Rappel

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

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

Le but de ce TP est d'étudier une suite au comportement complexe, la **suite logistique**. C'est une suite récurrente définie par

$$0 \leq u_0 \leq 1; \quad \forall n \in \mathbb{N}, \quad u_{n+1} = au_n(1 - u_n),$$

où dans tout le TP, $a \in [0, 4]$.

2 Première partie

2.1 Coder une fonction f qui prend en arguments deux réels a et x , et qui renvoie $ax(1 - x)$.

2.2 Afficher sur un même graphique les courbes de f

pour $a = 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4$ et $x \in [0, 1]$, en affichant les légendes pour distinguer les courbes.

2.3 Deux questions de maths :

(i) Montrer par récurrence que

$$\forall n \in \mathbb{N}, \quad 0 \leq u_n \leq 1$$

(ii) Résoudre l'équation

$$ax(1 - x) = x, \quad x \in [0, 1]$$

- 2.4 Coder une fonction suite qui prend en arguments deux réels a, u_0 et un entier n , et qui renvoie la liste des valeurs u_0, u_1, \dots, u_n . On pourra s'aider du TP 3.
- 2.5 Toujours en s'appuyant sur le TP 3, coder une fonction colimacon qui prend en arguments deux réels a, u_0 et un entier n , et qui dessine le diagramme en colimaçon correspondant à la suite récurrente étudiée.
- 2.6 Ajouter les deux lignes suivantes à votre code :

```
[ ]: plt.scatter(doubles[0:-1], doubles[1:], c = np.arange(2*n+1), cmap = 'viridis')
plt.colorbar()
```

- 2.7 Interprétation ?
- 2.8 Tester la fonction colimacon avec diverses valeurs de u_0 et de a . On pourra en particulier tester avec $a = 3.2$ et $a = 3.8$. Peut-on constater un comportement général de la suite $(u_n)_n$?

3 Deuxième partie

- 3.1 Coder une fonction diagramme1 qui prend en argument deux réels a, u_0 et un entier n , et qui dessine sur un graphe les points de coordonnées (a, u_k) pour $k \in \{0, \dots, n\}$.
- 3.2 Tester la fonction diagramme1 avec diverses valeurs de $a \in [0, 4]$ pour $u_0 = 0.7$ et $n = 250$.
- 3.3 À partir de la fonction diagramme1, coder une fonction diagramme2 qui prend en argument deux réels a, u_0 et deux entiers $N_1 < N_2$, et qui dessine sur un graphe les points de coordonnées (a, u_k) pour $k \in \{N_1, \dots, N_2\}$.
- 3.4 Que peut-on espérer mettre en évidence avec la fonction diagramme2 ?
- 3.5 Reprendre les tests précédents avec la fonction diagramme2. On pourra fixer $N_1 = 200$ et $N_2 = 300$.
- 3.6 En s'inspirant de diagramme2, coder une fonction bifurcation qui prend en arguments un réel u_0 et deux entiers $N_1 < N_2$, et qui dessine sur un graphe les points de coordonnées (a, u_k) pour $a \in [0, 4]$ et $k \in \{N_1, \dots, N_2\}$.