Informatique - TP 14

Révisions, méthodes d'Euler et de Newton, ensembles fractals

Le but de ce TD est d'étudier quelques exemples de suites récurrentes, et de réviser certaines de leurs utilisations en informatique.

I. Énoncé

I.1. Suites récurrentes

1. On définit la suite A par $A_0 = 2$ et $\forall n \in \mathbb{N}, A_{n+1} = 1 - e^{A_n}$.
   Écrire une fonction `A(n)` renvoyant la valeur de A_n pour la valeur de n donnée en paramètre.
2. Écrire une fonction `lstA(n)` renvoyant la liste des valeurs de A_i pour $i \in [\![0; n]\!]$.
3. On rappelle que pour tracer la représentation de points $P_0, P_1, ..., P_n$, on importe le module `matplotlib.pyplot` puis on utilise la fonction
   `plt.plot(X,Y,'+')`
   où `X` est la liste des abscisses des points et `Y` la liste de leurs ordonnées.
   Tracer pour $n = 100$ la liste des points d'abscisses $i \in [\![0; n]\!]$ et d'ordonnées A_i.
   La suite A semble-t-elle convergente ?
4. On définit les suites B et C par $B_0 = 0, C_0 = 1$ et $\forall n \in \mathbb{N}, \left\{\begin{array}{lcl} B_{n+1} & = & -B_n + \frac{C_n}{2} \\ C_{n+1} & = & B_n - C_n \end{array}\right.$.
   Écrire une fonction `BC(n)` renvoyant le couple (B_n, C_n) pour la valeur de n donnée en paramètre.
5. Écrire une fonction `lstBC(n)` renvoyant la liste des couples de (B_i, C_i) pour $i \in [\![0; n]\!]$.
6. Tracer pour $n = 10$ la liste des points d'abscisses $i \in [\![0; n]\!]$ et d'ordonnées B_i, puis la liste des points d'abscisses $i \in [\![0; n]\!]$ et d'ordonnées C_i.
   Les suites B et C semblent-t-elles bornées ?
   Remarque : pour tracer des points de forme différente pour la suite C, on pourra utiliser
   `plt.plot(X,Y,'o')`
   au lieu de
   `plt.plot(X,Y,'+')`.
7. Refaire les deux questions précédentes en remplaçant la définition des suites B et C par
   $$B_0 = 0, C_0 = 1 \quad \text{et} \quad \forall n \in \mathbb{N}, \left\{\begin{array}{lcl} B_{n+1} & = & -B_n + \frac{C_n}{2} \\ C_{n+1} & = & B_{n+1} - C_n \end{array}\right.$$
   et en traçant les 1001 premiers points ($n = 1000$).
   Les suites semblent-elles bornées cette fois-ci ?

I.2. Méthode d'Euler et systèmes chaotiques

On rappelle que le principe de la méthode d'Euler pour une équation différentielle du type $y' = f(y, t)$ (où f est une fonction continue), est d'effectuer l'approximation

$$y' = \lim_{h \to 0} \frac{y(t+h) - y(t)}{h} \approx \frac{y(t+dt) - y(t)}{dt}$$

pour une valeur de dt « suffisamment petite ».
En partant de $y_0 = y(0)$, on obtient alors une suite récurrente où $y_{n+1} = y_n + dt \times y' = y_n + dt \times f(y_n, n \times dt)$.

1. Écrire une fonction `euler(y0,T,dt=1e-2)` renvoyant la liste des valeurs obtenues à l'aide de la méthode d'Euler pour :
   - l'équation différentielle $y' = y^2 + y - \cos(y)$
   - avec la condition initiale $y(0) = y_0$
   - en prenant une valeur de $dt = 10^{-2}$ par défaut modifiable dans un paramètre optionnel
   - la liste renverra toutes les valeurs de y_k correspondant à des temps $t_k < T$.
2. Tester votre fonction pour $y_0 = 0,55$ et $T = 10$ et représenter graphiquement y en fonction de t.
   Puis la tester à nouveau pour une valeur de y_0 légèrement supérieure (avec représentation graphique).
   Remarque ?
3. On considère le système différentiel (issu d'un problème de cinétique chimique) suivant :

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} & = & -y - z \\ \dfrac{\mathrm{d}y}{\mathrm{d}t} & = & x + \frac{y}{5} \\ \dfrac{\mathrm{d}z}{\mathrm{d}t} & = & \dfrac{1 + z(10x - 57)}{10} \end{cases}$$

   avec la condition initiale $x(0) = -10$, $y(0) = z(0) = 0$.
   Écrire une fonction `eulerXYZ(V0,T,dt=1e-2)` implémentant la méthode d'Euler pour ce système différentiel - ici `V0=(x0,y0,z0)`.

4. Représenter graphiquement les points de coordonnées $(x(t); y(t))$ obtenus pour $T = 100$.
   Refaire des représentations graphiques en changeant uniquement la valeurs de x_0. Remarque ?

I.3. Méthode de Newton et fractals

Soit f une fonction définie et dérivable sur $\mathbb{R}$ (ou sur $\mathbb{C}$, ou sur une partie I de $\mathbb{R}$ ou de $\mathbb{C}$) et s'annulant en au moins un point.
La méthode de Newton consiste à tenter d'obtenir une solution de $f(x) = 0$ comme limite de la suite v définie par $v_0 \in \mathbb{C}$ et $v_{n+1} = v_n - \dfrac{f(v_n)}{f'(v_n)}$.

1. Écrire une fonction permettant d'obtenir, ***si la suite de Newton associée à*** $z \mapsto z^3 - 1$ ***et*** $v_0 \in \mathbb{C}$ ***est définie et converge***, la meilleure approximation de sa limite à l'aide de flottants.
   En pratique, on arrêtera la recherche de cette approximation au bout de N termes si elle n'est toujours pas obtenue (on peut prendre par exemple $N = 10$).
2. Tester votre fonction avec $v_0 = \frac{-1}{2} + i\frac{1}{10}$ puis avec $v_0 = \frac{-1}{2} + i\frac{1}{5}$. Pouvait-on s'attendre aux limites obtenues ?
3. Pour chaque complexe c du pavé $[-1; 1] \times [-1; 1]$ du plan complexe subdivisé en 200×200 pixels, tracer le pixel correspondant dans l'une des trois couleurs rouge/vert/bleu suivant la racine cubique de l'unité vers laquelle la suite a convergé en prenant $v_0 = c$.
   Tracer le pixel en noir si la suite n'a pas convergé.
   Rappels : pour créer une image noire de 200×200 pixels, on utilise le code suivant

```
import numpy as np
image=np.zeros((200,200,3),dtype=np.uint8)
```

   Pour modifier le pixel à l'intersection de la ***ligne*** numéro i et de la ***colonne*** numéro j, on écrit

```
image[i,j]=[R,V,B]
```

où [R,V,B] donne l'intensité de *R*ouge, *V*ert et *B*leu.
Enfin, pour afficher l'image, on écrit

```
import matplotlib.pyplot as plt
plt.imshow(image)
```

4. Recommencer la question précédente pour la fonction $f : z \in \mathbb{C} \mapsto (z-1)\,(z^3-1)$.