Algorithmique et Programmation 3
TP 3: Racine Carrée et La Bifurcation de Feigenbaum

12/11/2020

1 Racine Carrée

On peut calculer la racine carrée d'un nombre A par l'algorithme suivant :

Algorithm 1: Racine carrée

Choisir au hasard un nombre X
while $\varepsilon < |X - \sqrt{A}|$ **do**
    $X = \frac{1}{2}\left(X + \frac{A}{X}\right)$
Retourner X

1. Écrivez cet algorithme.
2. Comparez l'algorithme avec la fonction *sqrt* de Python. Ensuite, calculez la racine carrée de tous les entiers entre 1 et 1 milliard et voyez le nombre d'itérations utilisées. Attention, pour utiliser la fonction *sqrt* il faut d'abord importer la librairie *math* : **from math import ***.
3. Pourquoi est-ce que cet algorithme fonctionne ?

2 La bifurcation de Feigenbaum

Considérez la suite suivante,

$$x_{n+1} = 1 - \mu x_n^2, \; x_0 \in [0, 1] \tag{1}$$

La suite $(x_n)_n$ représente l'évolution de la taille (en proportion) d'une population biologique au fil des générations. Selon la valeur de μ l'équation (1) peut générer une suite convergente, une suite soumise à oscillations ou une suite chaotique.

1. Écrivez une fonction qui calcule la suite de Feigenbaum étant donné μ et x_0.
2. À partir du même point initial x_0, étudiez les différentes suites obtenues quand on considère
   (a) $0 <= \mu <= 0.75$
   (b) $0.75 <= \mu <= 1.25$
   (c) $1.25 <= \mu <= 1.368$
   (d) $1.368 <= \mu <= 1.401$
   (e) $1.401 <= \mu <= 2$

   Pour cela, pensez à utiliser la librairie *matplotlib*.